Augmented Photogrammetry: 3D Object Scanning and Appearance Editing in Mobile Augmented Reality

Daniel Lohn
University of California
Santa Barbara, CA, USA
dlohn@ucsb.edu

Tobias Höllerer
University of California
Santa Barbara, CA, USA
holl@cs.ucsb.edu

Misha Sra
University of California
Santa Barbara, CA, USA
sra@ucsb.edu

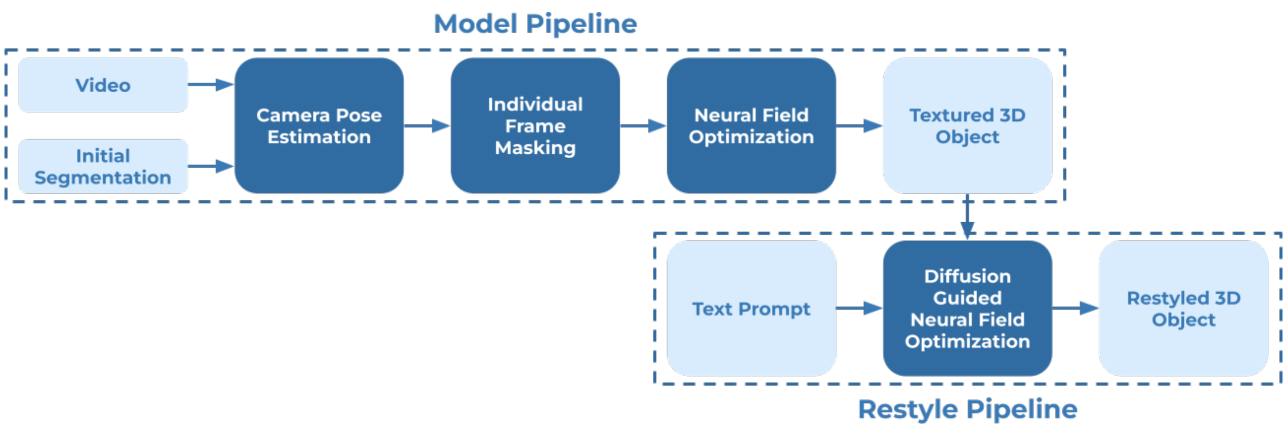


Figure 1: Pipeline overview showing how an input video is converted into a stylized 3D model in AR. The restyling stage can be triggered as often as needed.

ABSTRACT

We present a novel approach, Augmented Photogrammetry, for scanning and editing the appearance of physical objects in augmented reality (AR). Our work provides a user-friendly and efficient technique for enabling customizable appearance modifications in real time on arbitrary objects scanned from a user's physical environment. We accomplish this by integrating Structure from Motion (SfM), instance segmentation, and machine learning into a unified pipeline. Our streamlined process enables users to easily select a physical object and specify its desired appearance. We believe our mobile AR approach holds promise for applications in interior design, virtual prototyping, and content creation.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; **Mixed / augmented reality**.

KEYWORDS

Augmented Reality, Photogrammetry, Style transfer

UIST '23 Adjunct, October 29–November 01, 2023, San Francisco, CA, USA

ACM ISBN 979-8-4007-0096-5/23/10.
https://doi.org/10.1145/3586182.3616638

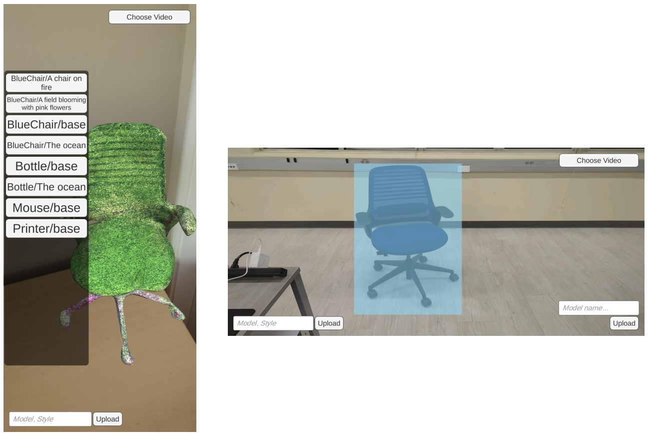
ACM Reference Format:
Daniel Lohn, Tobias Höllerer, and Misha Sra. 2023. Augmented Photogrammetry: 3D Object Scanning and Appearance Editing in Mobile Augmented Reality. In *The 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23 Adjunct), October 29–November 01, 2023, San Francisco, CA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3586182.3616638

1 INTRODUCTION

As we continue to bridge the gap between the physical and the digital world, three-dimensional (3D) models of real-world objects have become increasingly important in several applications such as video games, virtual reality, augmented reality (AR), and digital asset libraries. Existing 3D scanning methodologies often involve complex equipment such as depth-sensing cameras[1], laser scanners, or time-consuming manual labor in sculpting the object digitally. Recent work [4, 6, 11] has explored fitting a neural field representation of a scene using only image data and camera poses, followed by extracting an explicit representation (e.g. triangle mesh). Although all the required data for neural field models can be captured using a mobile video camera, these models need to be trained on a remote host due to high computational cost. If no access to the remote host is allowed, editing applications using model inference become inaccessible to the end user.

Motivated by these shortcomings, our work introduces an application for 3D object scanning that enables editing the appearance of arbitrary real world objects in AR. For example, we can change

[1]https://developer.apple.com/documentation/avfoundation/additional_data_capture/capturing_depth_using_the_lidar_camera/



(a) Stylized 3D Object Placement **(b) Segmentation Interface**

Figure 2: Application interfaces for (a) placing a stylized physical chair using our method, and (b) selecting the original physical chair to scan and stylize.

the surface color of a physical coffee mug to make it appear like it's made of different materials, such as metal, wood, or Van Gogh's Starry Night. The customization process utilizes a latent diffusion model [8] to style the object based on a text prompt, allowing for greater flexibility and a user-friendly interface. Previous approaches have either used preset objects with UV maps[2] or simply painted texture images onto flat 3D objects [5].

To scan an object, we start by leveraging the camera and AR capabilities of a smartphone, recording a video that captures a physical object from multiple angles. This video is then processed in a pipeline (Figure 1) to transform the captured video into a detailed 3D model.

2 METHOD

Our method utilizes a front end interface in an AR app, and a back end server which handles two separate computation steps shown in Figure 1: (1) The model pipeline, which turns a video and an object's bounding box on the first frame into a textured 3D model, and (2) the restyle pipeline which takes the output 3D model from step 1 and restyles the texture map according to a text prompt provided by the user. The AR interface (Figure 2a) is used to trigger either the model pipeline by uploading a video and initial segmentation, or the segmentation pipeline by uploading a text prompt and the name of the model to be restyled.

Video Frame Extraction. We split the input video into individual frames and process them with the Structure from Motion (SfM) pipeline COLMAP [9] to generate camera poses for each frame and a sparse point cloud.

Per-Frame Segmentation. We use the Segment Anything Model (SAM) [3] to generate masks for each frame. This model produces instance segmentations from a single image, and is promptable with both bounding boxes and points. The user provides the bounding box for the first frame using the interface shown (Figure 2b). The bounding box for the first frame is fed into SAM, which generates a mask for the object. The non-occluded points in the sparse point cloud that project within the mask are projected into the next frame as inputs to SAM.

[2]https://geenee.ar/

(a) Reference Image **(b) Unstyled Model** **(c) "A field blooming with pink flowers"** **(d) "The ocean"** **(e) "A chair on fire"**

Figure 3: Results of Restyling Method

Mesh Generation. The masked video frames and camera poses are inputted into the Nerf2Mesh model [11]. This model optimizes a neural signed distance field with color, from which a textured triangle mesh can be extracted. We introduce an additional loss term when optimizing this model during the restyling stage.

Loss Formulation. We employ a unique loss function for style transfer that incorporates two components: a *style loss*, and a *content loss*. The *style loss* employs the score distillation sampling technique introduced in Dreamfusion [7], a method of using a pretrained diffusion model to score any image (in our case, a 3D rendering) against a text prompt. The diffusion model chosen for our implementation is the freely-available Stable Diffusion [8].The *content loss* ensures that updates to the texture map preserve visual landmarks. To accomplish this, we draw from previous work in neural style transfer [1] and restyling neural fields [2], and use features extracted from the intermediate layers of a pretrained VGG network [10] as a representation of content. The content loss is the mean squared error between the VGG features of the masked input frames and the VGG features of renders of the model from the same poses.

3 DISCUSSION AND FUTURE WORK

In this work, we introduced a novel approach for the creation and customization of 3D models in augmented reality. We built a pipeline composed of state-of-the-art techniques to output a 3D model that can be stylized based on a user's preferences expressed with text prompt. See Figure 3 for our stylization results.

While our method offers a promising avenue for users to create and customize 3D models in AR, we acknowledge certain limitations. Our approach relies heavily on the performance of SAM for accurate segmentation, which might be influenced by factors such as indoor lighting, object complexity, and video quality. Furthermore, our object editing technique does not support deformations or modifications that cannot be easily translated into a text prompt.

Future work will aim to improve these aspects with potential directions including the integration of more robust video segmentation techniques and refinement of our style transfer process to handle more complex artistic styles and textures.

REFERENCES

[1] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* (2015).

[2] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. 2022. StylizedNeRF: Consistent 3D Scene Stylization as Stylized NeRF via 2D-3D Mutual Learning. In *Computer Vision and Pattern Recognition (CVPR)*.

[3] 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. 2023. Segment Anything. *arXiv:2304.02643* (2023).

[4] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. 2023. Neuralangelo: High-Fidelity Neural Surface Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[5] Marwan Mashra, David Maruscsak, and Christian Sandor. 2023. Diffusion-Based Content Generation for Augmented Reality. In *IEEE VR 2023*. 1–2.

[6] 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 *ECCV*.

[7] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv* (2022).

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752 [cs.CV]

[9] Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[10] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

[11] Jiaxiang Tang, Hang Zhou, Xiaokang Chen, Tianshu Hu, Errui Ding, Jingdong Wang, and Gang Zeng. 2022. Delicate Textured Mesh Recovery from NeRF via Adaptive Surface Refinement. *arXiv preprint arXiv:2303.02091* (2022).