Alleviating the Uncanny Valley Problem in Facial Model Mapping Using Direct Texture Transfer

Kaylee Andrews*
Augusta University

Jeffrey Benson†
Augusta University

Jason Orlosky‡
Augusta University

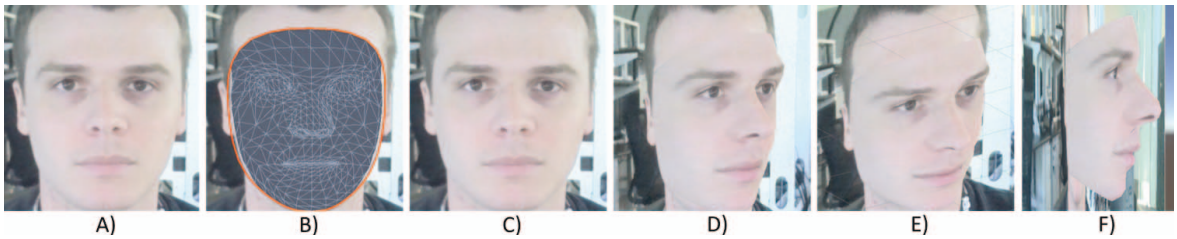

Figure 1: Images showing A) a source image from a single monocular web camera, B) the corresponding coarse reconstruction of a 3D mesh shown as a wireframe, C) the rendered mesh with our direct texture transfer approach applied (with the camera positioned directly in front of the generated model), D), E), and F) side-profile views showing the model geometry and feature persistence.

Abstract

Though facial models for telepresence have made significant progress in recent years, most model reconstruction techniques still suffer from artifacts or deficiencies that result in the uncanny valley problem when used for real-time communication. In this paper, we propose an optimized approach that makes use of direct texture transfer and reduces the inconsistencies present in many facial modeling algorithms. By mapping the source texture from a 2D image to a rough 3D facial mesh, detailed features are preserved, while still allowing a 3D perspective view of the face. Moreover, we accomplish this in real time with a single, monocular camera.

1 Introduction and Prior Work

The accurate modeling of 3D facial features and reproduction of skin properties are often considered to be essential tasks for achieving high-fidelity telepresence [3]. To date, most research has focused on generating a high-fidelity 3D model of a person's face from source video and recoloring or re-texturing that model for use in 3D telepresence applications. However, one major drawback with these reconstruction systems is that even minor inconsistencies in the facial model can have a disproportionately strong affect on a viewer's perception of the resulting model. More specifically, viewers often perceive the model to be unnatural or even grotesque, which is often referred to as the Uncanny Valley problem [1].

A major problem with 3D model construction is that minute features such as shading, coloration, skin reflectance, and other parameters result in a model that is easily detectable as fake. This is similar to problems in the modeling of object lighting, optical effects, or other perceptual cues. Moreover, while some techniques such as that of Zhang et al. provide correction of facial models that look increasingly realistic, these still require complex hardware setups or are not viewable from all angles [2]. In this paper, we propose an approach that makes use of real time texture mapping applied to a coarse model that can help alleviate the Uncanny Valley problem. To do so, we take advantage of the fact that almost all of the nuances of facial features are transmitted with the 2D texture in video systems. By mapping this 2D texture data onto a rough 3D model generated by TensorFlow, the coarse model appears to take on all of the nuanced features of the 2D texture, while still retaining the essential 3D data necessary for viewing in Augmented and Virtual Reality (AR/VR) telepresence applications. By doing so, we can generate convincing facial replicas that are easy to deploy in telepresence systems with a monocular camera.

1.1 Avoiding the Uncanny Valley

Mori et al. discuss how movement can result in an increase in the volatility of the curve that describes the uncanny valley [1]. With respect to 3D modeling, minute movements or coloration that differs from the original face often result in that face being considered unrealistic or inaccurate during telepresence. Reconstructing a face and recoloring the resulting model using in-situ lighting still has a negative effect on its perceived eeriness.

Our approach alleviates this problem by striking a balance between the transfer of minute facial features that are ingrained into the source texture and a coarse 3D model generated from a readily available neural network. The key to our convincing models relies on the fact that the source texture can be stretch over the coarse model without the user noticing perceptual changes in consistency. For example, if the pixels representing a nostril are stretch slightly upwards, assuming that the user's view of the texture is not completely off axis, their view of nostril will essentially remain perspective-correct.

Another essential aspect to consider is the importance of preserving the intricacies and dynamics of facial expressions. Although current technology can replicate high-level structures and textures, capturing the minute details of human emotions on the face remains a significant challenge in 3D modeling alone. By maintaining the original facial texture, we ensure that even minor changes in expressions are retained, minimizing the uncanny effect.

2 Implementation and Texture Mapping Process

To accomplish this, our approach makes use of several existing libraries, including Mediapipe's machine learning framework to generate dense 3D facial landmarks and Unity's VR plugin to create our environment. In addition to integrating these two components, we provide a customized method for constructing a Mesh from 3D facial landmarks and real time texturing using UV mapping.

To implement our direct texture-mapping approach, we use Tensorflow's Mediapipe plugin, which overlays facial features onto a 2D plane. We customized this plugin to convert the depth information (z-axis coordinates), to Unity coordinates, which produced a set of

*e-mail: kandrews@augusta.edu
†e-mail: jebenson@augusta.edu
‡e-mail: jorlosky@augusta.edu

2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW) | 979-8-3503-7449-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/VRW62533.2024.00172

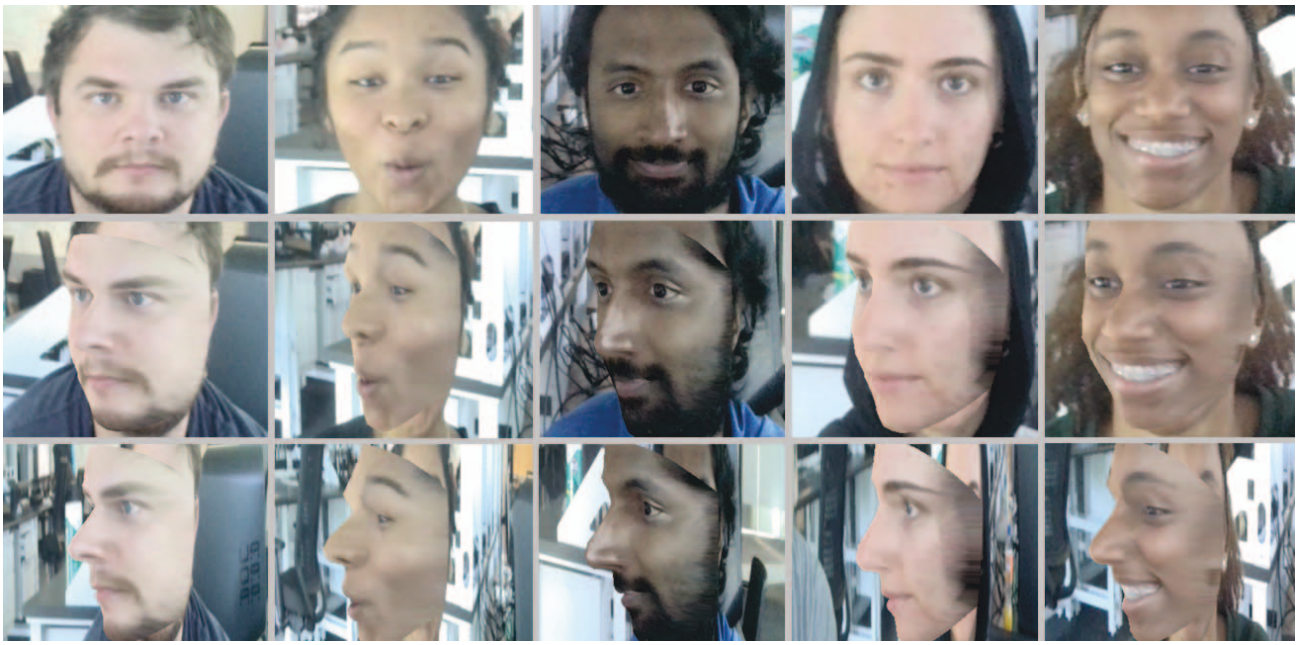

Figure 2: Images from our benchmark tests showing five different individuals (rows) with different expressions, including a) the source webcam texture and b) our custom mesh overlaid directly onto the source texture face. The following images show the textured mesh from c) a right profile, d) a left-oblique profile, e) head-on from the camera's perspective, f) a right-oblique profile, and g) a right profile.

3D facial landmarks. To produce a coarse 3D facial structure, we stored the 468 facial landmarks available from the Mediapipe API to an array of unconnected vertices. These were segmented into primary facial features, such as the eyes, nose, cheeks, and mouth, which streamlined the connection of these points. This also meant that each point had to be re-indexed according to its respective facial feature, allowing us to logically construct the mesh while ensuring consistency between mesh points and 3D coordinates.

To determine the order of indices for each triangle, we had to manually determine each position since Mediapipe does not explicitly provide neighbor landmark coordinates. The result of this process was an interconnected mesh, as shown in B) of Figure 1, offering a more detailed and accurate representation of facial features. The detailed triangle structure of these meshes can be seen by zooming into column b) of Figures 1.

2.1 Texturing and UV Generation

Once the mesh has been generated, we need to accurately map the correct pixels from the source texture onto the coarse model. To compute UV values, we first identified the pixel location of each landmark within the source image by back-calculating from the plugin-provided 3D position in Unity to a rectangle transform on which the video texture was mapped. Though iterating through all facial landmarks and remapping their vertices to triangles via script is somewhat computationally intensive, the small number of landmarks still makes this easy to do in real time.

Another benefit of this approach is that the color and other features of the transferred texture can still be modified to better resemble the on-site scene, much like the brightness, contrast, or white balance of a webcam image can be modified without a significant effect on perceived realism. This has the benefit of allowing some flexibility to better match scene conditions while still preserving the facial features needed to avoid the uncanny valley problem.

2.2 Benchmarking

To evaluate the ability of our texture mapping approach to alleviate the uncanny valley problem, we recorded images and videos of our resulting model for visual inspection. Figure 2 shows a series of these images from different angles, with different faces, and with different expressions to provide a wide range of facial input. We have also attached a video as supplementary material that shows a number of faces in real time from different angles, one of which is also presented as a VR Telepresence call through the HTC Vive Pro.

These benchmarks provide visual evidence that the texture mapping can 1) transfer the nuanced features of the face, 2) provide a 3D mapping that effectively preserves these features at multiple viewing angles, and 3) alleviate the uncanny valley problem. We also tested these in a real time virtual environment, which is shown in the video in our supplementary materials, to show how the mesh appears from different viewing angles. This provides further evidence of its scalability, adaptability, and textural integrity in 3D space.

3 Conclusion

In this work, we explore an alternative approach to facial modeling that can help alleviate the uncanny valley problem for real time telepresence. By taking advantage of the features present in 2D textures taken directly from a monocular camera, we can transfer an individual's minute facial textures onto a coarse 3D model and retain the features necessary to reproduce natural characteristics. We then provide a visual evaluation of different faces, viewing angles, and expressions to validate the extent to which our approach can accurately reproduce facial features under varied conditions. This work was supported in part by NSF grant #2223035.

References

[1] M. Mori, K. F. MacDorman, and N. Kageki. The uncanny valley [from the field]. *IEEE Robotics & automation magazine*, 19(2):98–100, 2012.

[2] Y. Zhang, J. Yang, Z. Liu, R. Wang, G. Chen, X. Tong, and B. Guo. Virtualcube: An immersive 3d video communication system. *IEEE Transactions on Visualization and Computer Graphics*, 28(5):2146–2156, 2022.

[3] R. Zheng, P. Li, H. Wang, and T. Yu. Learning visibility field for detailed 3d human reconstruction and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 216–226, 2023.