Angular Limits of Image Samples for Novel View Synthesis

Jaroslav Venjarski, Matej Benža , Juraj Cáfal
Slovak University of Technology in Bratislava, Slovakia
jaroslav.venjarski@stuba.sk

***Abstract* – Novel view synthesis (NVS) is a technology that aims to generate new views of a scene from a limited set of images, enabling rendering from unseen perspectives. This thesis explores the development of NVS and tests the capabilities of NVS to create a middle view of a human face with increasingly more difficult sets of images to create the view from. A novel method is proposed to improve visual quality and efficiency, addressing challenges in reconstruction and view consistency. The results demonstrate the potential of the approach for applications in virtual reality, telepresence, and immersive media, contributing to the development of realistic scene generation.**

Keywords – **Novel view synthesis; Triangulation; Sample; Image blending; Image morphing**

I. INTRODUCTION

Novel View Synthesis (NVS) is a large field in computer vision and graphics, concerned with generating new views of a scene that were not captured in the original set of input images. Given a limited set of observations, the goal of NVS is to reconstruct or render images of the scene from novel camera positions with realistic detail and consistency. This task is particularly challenging due to issues such as occlusions, varying lighting conditions, complex scene geometry, and limited information provided by the input views.

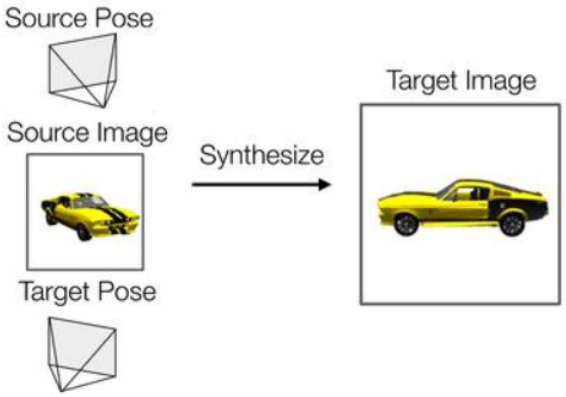

Figure 1. Visualisation of the objective of NVS to create new views of a scene with a lack of data for that specific scene.

Most approaches to NVS often rely on explicit 3D representations, such as point clouds, meshes, or depth maps, to reconstruct the structure of the scene and reproject it to the desired viewpoint. While these methods can produce accurate results in controlled environments, they typically struggle with scenes that lack sufficient texture, contain reflective or transparent surfaces, or have limited viewpoint coverage, therefore NVS can not recreate views from data that is not present.

In recent years, learning-based methods have shown significant progress in overcoming the limitations of classical techniques. Neural rendering approaches, such as Neural Radiance Fields (NeRF) and its variants, model the appearance and geometry of a scene implicitly through deep neural networks. These models are capable of generating highly realistic novel views with fine details and complex lighting effects, even with sparse input data. However, they often require long training times and large amounts of data, which limits their practicality in real-world applications.

This thesis focuses on the ability of Novel View Synthesis (NVS) to generate results from increasingly challenging sets of images, in order to identify the functional limits of NVS for the implementations discussed earlier. Our results do not present a definitive evaluation of the capabilities of NVS, as there are multiple ways to implement it, each with its own strengths and weaknesses. Our code is based on MediaPipe face recognition, a tool developed by Google using a large dataset of human faces. Since there are several ways to implement MediaPipe, analyzing the results of this thesis allows us to determine whether the limitations in generating a facial image are caused by MediaPipe's failure to detect facial features, or by errors introduced in a different stage of the image generation process, such as facial merging. Furthermore, this thesis examines the entire process of generating a novel view using NVS — from dataset creation to the final results [1], [2].

II. CREATION OF THE DATASET

For the purpose of having a realistic, consistent, and convenient dataset, we have opted for a virtual one. To this end, we have created a 3D model of a person, based on the image of one of the authors. This model was created using 3D sculpting with a high polygon count to create a realistic target for NVS. Furthermore, the model incorporates complex hair and skin details to simulate a more realistic use of NVS in the real world. This model was created in Blender, which has the ability to accurately simulate lighting and shadows, which we took advantage of.

Figure 2. Frontal view of the 3D model.

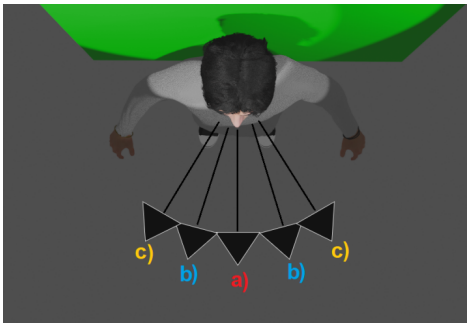

Figure 3. A visual representation of the first dataset, which we have labeled "horizontal". This dataset consists of pairs of images, except for the ground truth image (labeled *a)* in Figure 3), which always has a perpendicular angle relative to the subject. The paired images (labeled *b)* and *c)* for the separate pairs in Figure 3) are taken at the same angle relative to the ground truth but mirrored across it. These pairs of views were used as input data for NVS. After each successful generation of a frontal view, we tested another set of images with a less acute angle relative to the perpendicular view, continuing this process until we reached the failure point of our NVS code.

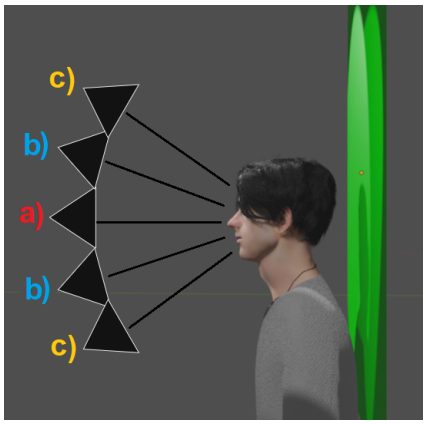

Figure 4. This visual representation of the second dataset, labeled "vertical," shows the positional arrangement of the cameras capturing input images along the vertical axis. It is important to note that for both of these datasets, the angles rotate around a point in the center of the head, rather than on the surface of the head. The cameras are kept at a constant distance from this central point for all images, which causes them to focus on different parts of the head — continually hiding and revealing various features as the angle changes [3].

III. Our Implementation of NVS

This implementation represents an approach via computer vision to Novel View Synthesis (NVS) specifically optimized for the moddeling of a human head. Unlike modern neural rendering techniques that might use implicit neural representations or neural radiance fields, this approach builds on classical image processing techniques with targeted optimizations for facial features. This approach contains following the steps:

- **Facial Landmark Detection and Extension**

Firstly, images from two views are loaded and subsequently processed. A facial landmark mesh is created using the MediaPipe library, this mesh was originally designed for the face area only, and has been extended with additional points in the areas of the hair, sides of the head, ears, and neck. These extended landmarks enable more precise morphing of the entire head, not just the face, leading to a more natural final image when generating the intermediate view between profiles.

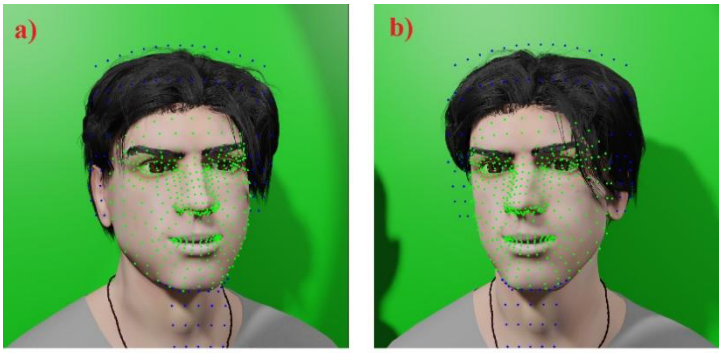

Figure 5. Facial landmark detection and its extension.

- **Background Removal and Segmentation**

Segmentation begins by converting the image to the HSV (Hue, Saturation, Value) color space, where color shades can be identified. Next, a selfie mask is created using MediaPipe Selfie Segmentation. To achieve the most accurate segmentation, a green mask is generated to identify green pixels. This green mask is then inverted to create a foreground mask, in order to suppress the background rather than the person. Finally, the foreground mask and the selfie mask are combined into a final mask using a logical AND operation.

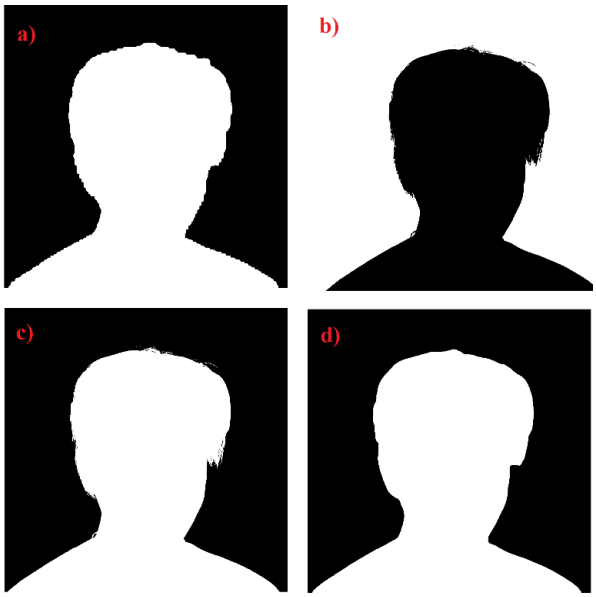

Figure 6. The segmentation process involves creating: a) a selfie mask, b) a green mask, c) a foreground mask, and d) a final mask.

Next, the focus shifts to smoothing the edges of the final mask. Using the Canny edge detection algorithm, edges are identified to create an edges mask. To expand the edges, dilation is applied, resulting in an edges dilated mask. Finally, an edge band mask is created by placing white pixels in areas where the edges dilated mask contains non-zero values. The edge band mask is then used to identify pixels in problematic edge regions that may remain even after applying the main mask. Once detected, these pixels are replaced with black.

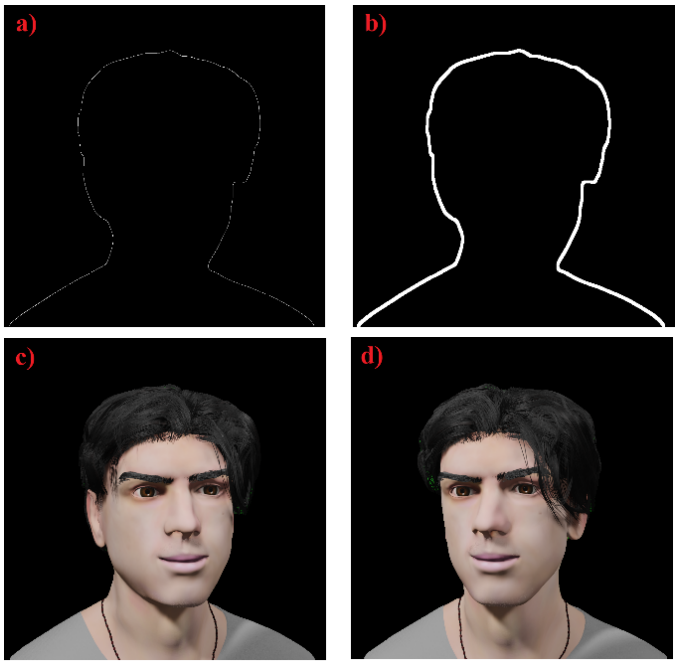

Figure 7. Edge smoothing process consisting of a) edges mask b) edge band mask c) final segmentation of left view d) final segmentation of right view.

- **Face Alignment and Cropping**

This function is used for aligning and cropping the face from the input image. For the sake of speed and better code optimization, it deliberately uses facial landmark detection instead of detecting the entire head. The obtained landmarks are converted from normalized coordinates to pixel coordinates, which are then used to find the minimum and maximum x and y values. From these values, the width and height of the face are determined, along with the size of the additional space around the face. The size of this padding is calculated as a percentage of the face dimensions, 80% of the face width for horizontal padding and 70% of the face height for vertical padding. To ensure the neck area is included, an additional 60% of the face height is added to the bottom. Finally, a precisely defined region is cropped from the original image based on these calculations.

- **Delaunay Triangulation**

The Delaunay triangulation process works with the already created landmark mesh of points and connects them in such a way that the resulting triangles do not overlap.

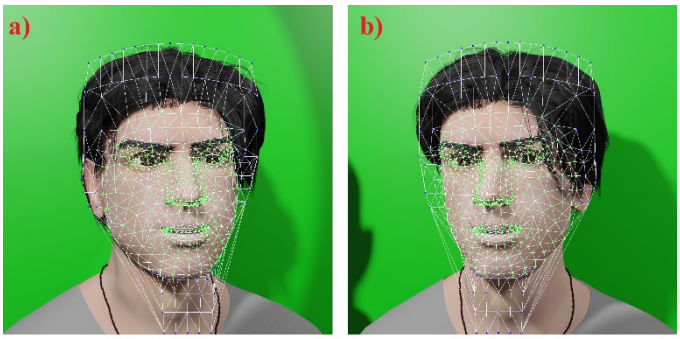

Figure 8. Delaunay triangulation.

- **Adaptive Triangle Morphing**

After triangulation, individual triangles from both views are compared using dynamically changing weights. The overlap ratio is calculated for each triangle. Based on this overlap value, an alpha value is determined. Triangles that reach an overlap of 87.5% or more are considered to contain valid information. These triangles are assigned an alpha value of 0.5 and are morphed in a 50/50 ratio. An alpha value of 0 is assigned when triangles contain data only from the first image, only data from the first image is used. The same applies for an alpha value of 1, with the only difference being that the data comes exclusively from the second image. [4]

- **Result Refinement and Masking**

To refine the results, the final morphed image is smoothed using a combined mask, resulting in a more natural appearance with soft transitions and clear separation from the background.

IV. RESULTS

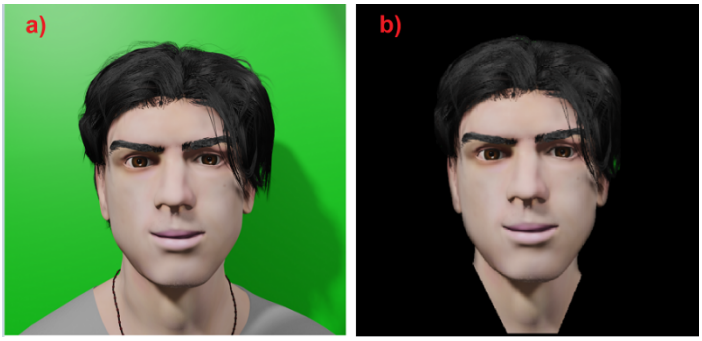

Figure 9. Baseline functionality of our NVS code. In order to compare the results effectively, we first need to establish a baseline and verify the functionality of the code. In Figure 5, we observe the result of NVS with optimal input data, meaning that the camera is positioned completely perpendicular to the face. Image *a)* represents the baseline, while image *b)* shows the result of applying NVS using image *a)* as both input images. This results in image *b)* having no facial differences due to image merging, but it allows us to identify the regions of the head where the NVS code is applied. Image *a)* will serve as a reference for subsequent comparisons.

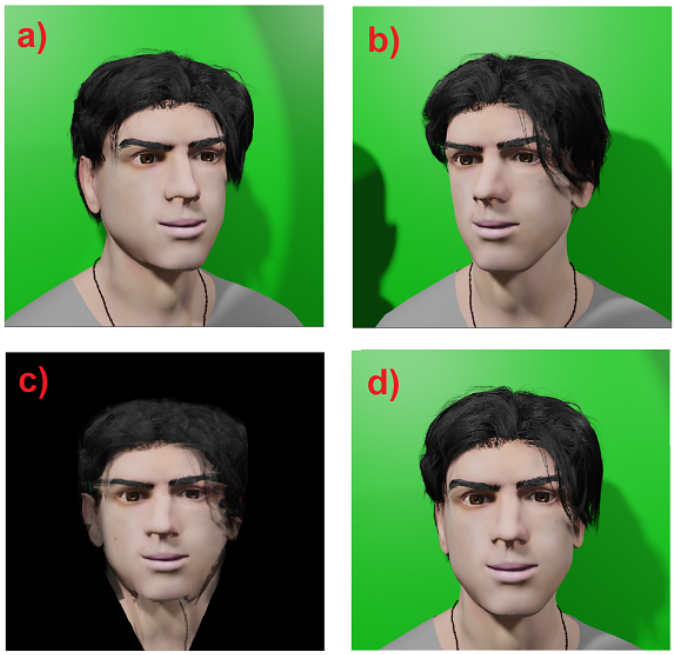

Figure 10. Explanation of Input Data and Results. Both the horizontal and vertical datasets consist of the following pairs of angular offsets: 5°, 10°, 15°, 20°, 25°, 30°, and 35°. For better visualization, we refer to the 15-degree input images shown in Figure 6, marked as *a)* and *b)*. From these input images, NVS attempts to generate a middle view, marked as *c)* in Figure 6, that approximates the ground truth, marked as *d)*, using the available input data. As we can observe, with a 15-degree offset of the face, the resulting image successfully recreates acceptable facial features but struggles with the areas of the head outside the facial region.

Figure 11. Overview of the horizontal and vertical databases used in creation of the resulting NVS images.

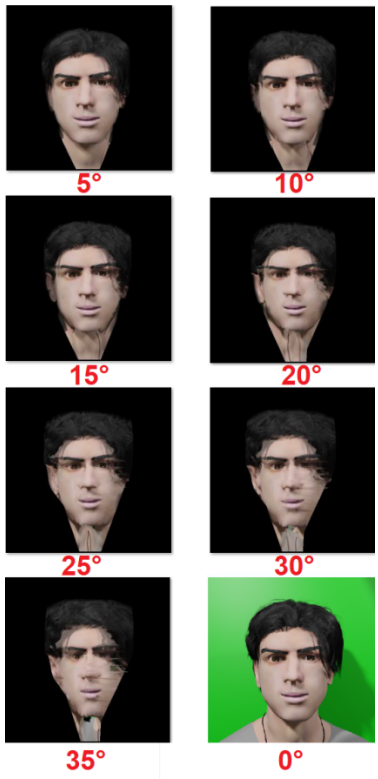

Figure 12. Results of horizontal dataset. As expected, the results progressively worsen with increasing angular offset. For the facial region, which MediaPipe is primarily trained on, NVS is able to reconstruct the face even at high offsets, around 20°–25°. However, at higher angular offsets, facial merging rapidly deteriorates. The rest of the head starts deviating significantly from the ground truth after 15°. The area that produces the most visible errors is the front of the neck, with the necklace being especially problematic.

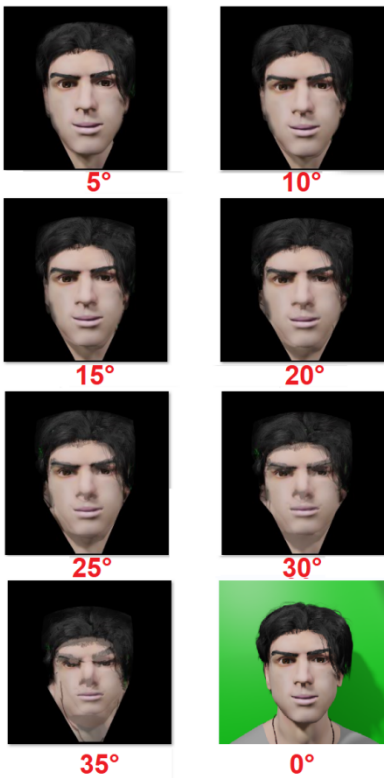

Figure 13. Results of the vertical dataset show fewer visual errors. For facial features, ghosting becomes noticeable than in the horizontal dataset at a lower angular offset, around 25°, but the features remain readable up to 30°. The overall head shape is also better in this dataset, with the main issue being that the neck and shoulder area appears positioned too high on the head. However, even at the highest offset of 35°, the head remains more readable than some results from the horizontal dataset at lower angular offsets.

V. Conclusion

This paper examined the angular limitations of image-based Novel View Synthesis for generating middle views of a human head using an approach with MediaPipe face recognition. Our experiments with horizontal and vertical datasets at increasing angular offsets (5° to 35°) revealed several insights. The facial region can be reasonably reconstructed up to 20°-25° in horizontal offsets, while vertical offsets maintained readable features up to 30°. Areas outside the facial region, particularly the neck and shoulders, showed deterioration at lower angles (around 15°). The vertical dataset produced better overall results with fewer visual errors than the horizontal dataset.

Acknowledgement

Research in this paper was supported by projects DISIC (09I05-03-V02-00077), InteRViR (VEGA 1/0605/23), and NEXT (ERASMUS-EDU-2023-CBHE-STRAND-2, ID: 101129022).

References

[1] Novel View Synthesis. (2024, March 22). 2024W, UCLA CS188 Course Projects. Availible at: https://ucladeepvision.github.io/CS188-Projects-2024Winter/2024/03/22/team05-novel.html

[2] Papers with Code - Novel View Synthesis. (n.d.). Availible at: https://paperswithcode.com/task/novel-view-synthesis

[3] Blender 4.4 Manual. (n.d.). https://docs.blender.org/manual/en/latest/index.html

[4] Media Pipe - https://pypi.org/project/mediapipe/