# Neurally Supported Avatar Generation and Rendering

Dr. Thomas Pock, Thomas Neff  
Graz University of Technology, Austria

A project proposal for Huawei Collaboration

## Introduction

A screenshot of a video game

Description automatically generated with medium confidenceThe last two years have shown a research explosion around the topic of Neural Radiance Fields (NeRFs) [1]. For a short overview of NeRFs, please see the NeRF explosion website [2]. NeRFs are an incredible compact representation that capture the radiance of a scene or objects in a simple multi-layer perceptron (MLP). Combining traditional volume rendering with the sampling from the MLP, allows to reconstruct scenes with complex shading information, see Figure 1. Extensions to NeRF include the splitting of a scene into multiple NeRFs, using NeRFs for dynamic content, integrating mip mapping into NeRFs, or using them to relight scenes. However, they require significant amounts of time to train and evaluate, which we could reduce with our work on DONeRF [3].

Figure 1 Schematics of the basic NeRF setup [1]. During ray marching, an MLP is queried multiple times for each pixel and the sampling results are accumulated to yield a final pixel color.

A concurrent trend in machine learning are transformers [4], which are able to outperform various learning approaches, first shown for sequence to sequence learning. Recently, they have also been shown to outperform CNNs for vision applications, such as classification [5] and have now been used for many computer vision tasks [6]. Transformers applied to sequential data work with the concept of attention and allow to pull information from data points further away in the sequence. Applying the concept to images, global information can be gathered early on in a network and thus lead the better performance on image processing tasks [5], see Figure 2.

Recently a variety of methods started to use neural approaches for avatar generation and rendering of human faces, including Deep Appearance Models [6], Neural Volumes [7], and Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction [8]. Using neural methods greatly increases the possibilities for reconstructing and rendering human heads and faces. The two biggest challenges for creating high quality avatars of human faces to the masses, is (1) the fact that previous approaches require large, expensive capturing systems and (2) that high quality rendering is costly. In this project we aim to lift these limitations, relying on a combination of neural radiance fields, surface extraction and efficient neural rendering.

## Goals

To lift these limitations, we propose the following pipeline:

1. Capturing the human head and face with a simple mobile phone
2. Training of a multi-expression NeRF
3. Surface extraction of the base shape NeRF
4. Surface NeRF generation
5. Animation extraction through latent codes
6. Encoding of the geometry and surface NeRFs
7. Rendering of animations using latent codes for surface-NeRFs

## Multi-expression NeRF and animations

One of the key steps is to establish a common capturing protocol and generating a multi-expression NeRF. To this end, we imagine a user, capturing her own head similarly to NeRFies [9]. Whereas the user is instructed to hold a couple of key expressions as well as arbitrary free expressions. For each of the key expressions which need to be captured from various angles, we start by creating a traditional NeRFie, which leads to a core-NeRF representation and warping fields. From the core NeRF representation, we can enforce a compact representation similarly to UNISURF [10], i.e., by compacting the signed distance field of the NeRF. In the next step we can align the base NeRFs for all key expressions and create a new base NeRF with different warp fields mapping the expressions. In this way, we can generate a canonical base NeRF with each expression being captured in a warp field. Note that this two step approach could also be carried out in a single training step. However, this may complicate training as too many complex networks need to be trained concurrently and there is ambiguity between the morphing network of NeRFies and the expression network.

In any case, having expression networks or one expression network conditioned on the expression allows to capture expressions in 3D space. Still, the base NeRF can be used to create a compact signed distance field and the extraction of an actual surface for efficient rendering.

With the base shape in place and the warp fields for different expressions and captured images, we can start learning surface NeRFs to find a compact representation of the shading for all parts of the head. We can use multiple NeRFs (see below) for different parts of the surface. However, the interaction with the warp fields is best described with a single on-surface NeRF. Instead of sampling in 3D space to get the output color, we want to sample based on the surface. To this end, we first create an inverse of the warp field to allow projecting the surface onto the screen of any given camera. We then take sample points around the surface location, project them back to 3D space and use the forward warp field to sample in space of the original base NeRF.

To allow for expression dependent shading, we want to have a common NeRF that gets different input variables (latent codes) depending on the expression. To this end, we train a common shading NeRF across all expressions, but use a few layers that are separately trained for each expression with a low number of connections to the shading NeRF. In this way, we can establish latent codes for each expression. We do the same to establish the inverse and forward warp field from the same latent codes, creating a full representation of the warping and shading from the latent codes.

After initial training, we can extend the trained networks with additional untagged captures, i.e., expressions that do not specifically follow the predefined base expressions. To this end, we can use a simple landmark tracker or facial pose estimator to align the head with the base NeRF (i.e. to register the pose). We can then fix all weights of the shading NeRF and warp field network and only learn the latent code, i.e., we find the latent code that describes the expression as good as possible. After deriving the latent codes, we can jointly optimize the warp fields and shading NeRFs alongside the found latent codes (including the original training data set). In this way, we can adjust the latent codes as well as the warping networks and shading NeRFs to the extended data set. Note that only the base geometry remains unchanged from this point forward.

The same latent codes could be generated by other means. For example, we could use a simple CNN to go from an input feed (e.g. webcam) to the right latent codes for driving the avatar. The mouth region could be driven from audio input. Or we could use cameras mounted on a head mounted display to drive the entire avatar.

An extension to the entire approach, includes the sharing of latent codes between users, i.e., for face to face expression transfer. While general parametric face models work on top of a common parameter space, we currently do not consider such a space during our avatar generation, but rather learn a new latent space encoding for every avatar. We could establish a general base-NeRF that can be modified with a warp field to capture different heads and faces. This approach may be interesting in the future. As a start, we can simply match avatars through their base expressions, or by using a landmark tracker to establish correspondences between different avatars and their expressions. We can enforce common latent codes and (partially) retrain the expression and shading networks to work with the common latent codes. Alternatively, we could also establish a translation network that translates from one avatar’s latent codes to another.

## Multi-Surface-NeRFs and efficient rendering

While individual NeRFs are well suited to represent individual objects, they do not scale well to high details. Adding additional layers to a NeRF shows diminishing returns and high quality rendering in high resolution are typically not suited for NeRFs. We tackle this issue with a multi-surface-NeRF approach. Instead of using a single NeRF for an entire avatars, we want to use attuned NeRFs for different parts of the head. For example, one NeRF for the skin, one for the eyes, one for the hair, one for lips etc. In this way, we can go significantly beyond the quality of having only a single NeRF.

Furthermore, this also allows to make use of symmetries in the data, or to reuse computation. For example. The left and the right eye could use the same NeRF with a different spatial encoding that remaps one eye to the other. Similarly the left and right ear could be handled by the same NeRF. However, in humans symmetry is never perfect. Thus, a one to one mapping is likely not feasible, especially when considering different expressions.

Similarly, manually tagging different parts of the face models is no long term solution to split the computation into multiple NeRFs. To this end, we propose a multi-headed transformer architecture where each head corresponds to a full NeRF. The transformers goal is to predict which NeRF should be sampled and adjust the domain between these NeRFs, e.g., map left ear to the right ear or place beauty spots at different locations of the face.

Furthermore, we can use a transformer architecture to mix different MLPs for shading, i.e., the transformer can learn to apply a base skin color, add subsurface scattering through a different network and add reflections through another network. Through this kind of splitting, we can add the computational power of multiple NeRFs on top of another. Furthermore, the transformer can decide how many samples are necessary for specific materials and where they should be placed, i.e., for hair a volumetric sampling with multiple steps may be needed. For skin, one sample on the surface and potentially one or two samples inside the surface for subsurface scattering may be useful.

Diagram

Description automatically generated

Figure 2 Simplified representation of our transformer structure. The input to the transformer encoder is a sequence of samples along the ray, whereas the previous output forms the next input to the transformer. The MLP selector and MLP pool can be realized in different forms. The MLP pool can either comprise multiple large MLPs or MLPs that only contribute parts of the learned representation; in the simplest case it may be a single MLP. The MLP selector may choose to mix multiple MLPs to determine the outputs or select a single from the pool.

This approach leads to multiple further use cases. We can pretrain MLPs for features typically found in human faces and use NTMs to choose the most fitting and adjust their appearance. This can significantly lower training time and lead to higher quality results. By injecting meta parameters into the specific MLPs, it may even be possible to adjust the avatars, i.e., beautify them or add special effects to their appearance.

The approach also lends itself for efficient rendering. Using multiple surface NeRFs, each NeRF can be compact and such their individual evaluations can be carried out efficiently. Relying on the surface representations allows us to only place very few samples for image generation. And finally, the separation into multiple NeRFs to generate complex shading may allow us to use a combination of view-dependent and view-independent as well as expression independent and dependent NeRFs. As such, we can cache the computations of some sub steps over time. I.e., the view independent, expression independent network can be fully caches for a specific surface location. View-dependent networks only need to be reevaluated when the view direction changes, and expression-dependent networks are recomputed only when the expression changes. Caches can be established on the surfaces themselves, allowing for an easy reprojection and interpolation.

For the network evaluations we can write custom CUDA or compute shaders that enable efficient evaluation of the multiple smaller NeRFs. This will lead to a large performance boost compared to evaluating the MLPs in a layer by layer manner, as memory transfer can be reduced significantly. It will be key to integrate the transformer architecture into the local evaluations as well. The entire evaluation can be paired with reduce precision evaluation and potential network pruning.

## Initial timeline

Clearly the entire described project requires a multi-year effort to be completed as a whole. In this statement of work, we focus on four subsets of the entire pipeline, looking at the feasibility of the entire approach:

* Training of a multi-expression NeRF
* Surface NeRF generation
* Animation extraction through latent codes
* Efficient rendering

In the first part, we evaluate whether a multi-expression NeRF can be generated from synthetic data that mimic a simple capturing environment. To this end, we will use a virtual model, like e.g. virtual Lousie [11] or the Unreal Engine Digital Human [12] to create a virtual data set that we can use for the entire pipeline.

Relying on a virtual head model, we can directly use the virtual model’s surface representation to represent the base mesh (with some manual adjustments to fit to the distortions of the base NeRF). We can thus disentangle the surface extraction from the rest of the pipeline.

Instead of creating a full multi-NeRF representation automatically, we first focus on using a single NeRF but rather establish the latent codes to drive the NeRF rendering and the different warp fields of the expressions. To evaluate the ability to use different on-surface NeRFs, we manually mark the mesh for different NeRFs (i.e., manually segment out those parts of the mesh the we believe should be captured by one and the same shading NeRF, like skin, lips, teeth, tongue, hair, etc).

For the latent code generation and extraction, we follow the approach described above, we will then explore the space of latent codes and evaluate the ability to interpolate between different expressions. Our evaluation will show that avatars can efficiently be driven by transmitting only little information between the source and the renderer. Finally, we tune the setup for efficient rendering, as described above.

## Team

Principal Investigator: Dr. Marc Masana will be the PI of the project. Marc Masana is a Post Doc at Graz University of Technology, working in the group of Prof. Thomas Pock. His interests include Deep Neural Networks, Object Detection, Network Compression and Continual Learning. His expertise in DNNs and network compression will be essential for all workpackages of the planned project.

PhD student: Thomas Neff will be one of the PhD students working on the project. Thomas is the main author of DONeRF and has worked at the intersection of computer graphics and computer vision, with a focus on machine learning and streaming rendering.

PhD student: to be hired

Student support: to be hired

Supporting Professor: Prof. Thomas Pock is a renowned expert in image processing, computer vision, inverse problems, convex and non-smooth optimization, and machine learning. He will lend his expertise on learning algorithms.

Supporting Professor: Prof. Dieter Schmalstieg is considered one of the pioneers of augmented reality, with additional focus on computer graphics, virtual reality, visualization and efficient parallel computing.

## Deliverables

#### Single expression NeRF generation @4 months after start

In the first deliverable, we will investigate the ability to reconstruct multiple expression NeRFs from synthetic human head data simulating the capture with a monocular camera and minimal expression changes to simulate that users will not hold expressions firmly throughout the capture. Repeating the process for multiple expressions for the same virtual human model, we will compare the generated base-NeRFs, determine correlations between the base-NeRFs and evaluate the possibility to generate a single base-NeRF from the expression NeRFs. The deliverable will contain the learned models, the training data set, and a report on the expression NeRFs.

#### Multi Expression NeRF generation @8 months after start

Following the results of the single expression NeRFs we will investigate the possibility to learn mappings between the different NeRFs. For this deliverable we will focus on different training schemes to learn consistent and smooth warping fields for the different expressions, and evaluating whether a consistent base-NeRF can be learned. evaluating the number of expressions that are needed, to establish a consistent interpolatable warpings. In a final step, we will evaluate how to combine the training of the different warp-fields to establish a common expression sub-network that may form the basis for driving the animations with latent codes. The deliverable will contain the learned models, the training data set, videos for the different strategies and a report on the results.

#### Surface-NeRF rendering @4 months after start

In this deliverable, we will first evaluate the setup for surface-NeRFs for avatars. Starting with the synthetic mesh data, we first annotate the mesh to supply different NeRFs for the individual mesh parts and investigate the most appropriate sampling strategy and how the setup performs for various sample counts, various NeRF sizes, and configuration parameters. In this test, we still focus on a single expression and evaluate the tradeoffs between memory, performance, and quality when switching to a surface NeRF strategy. The deliverable will contain the learned models, the training data set, a comparison between a complete NeRF, a single surface NeRF and multiple surface NeRFs. We will provide videos for the different strategies and a report on the results.

#### Multi-Expression Surface-NeRF rendering @8 months after start

Generalizing to multi-expression rendering, we need to established surface-NeRFs that can provide accurate rendering for different expressions. To this end, we will evaluate different parameter encodings on the surface and look into the severity of color changes between expressions. One way to establish the lighting differences between expressions is to encode the differences into a separate NeRF that is conditioned on expression latent codes, whereas the base shading NeRFs are independent of them. Alternatively, we could also have expression dependent inputs only and share the entire surface NeRFs between all expressions. Evaluations will show which strategy works best. The deliverable will contain the learned models, the training data set, videos for the different strategies and a report on the results.

#### Avatar generation and Rendering from latent codes @12 months

In the final work package, we combine all previous parts to generate a complete avatar that can be driven by latent codes and rendered efficiently. We will focus on how to establish the latent space variables such that they are compact and allow for interpolation. We will evaluate the required density of the expression space and investigate whether additional expressions only present in individual frames help the network to learn a better latent space. For efficient rendering we will provide a custom real-time renderer that implements the pipeline completely on the GPU. The deliverable will contain the learned models, the training data sets, videos for the different strategies, the source code for the renderer and a report on the results.

[1] Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." European conference on computer vision. Springer, Cham, 2020.

[2] Frank Dellaert, “NeRF Explosion 2020” <https://dellaert.github.io/NeRF/>

[3] Neff, Thomas, et al. "DONeRF: Towards real-time rendering of neural radiance fields using depth oracle networks." *arXiv e-prints* (2021): arXiv-2103.

[4] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*. 2017.

[5] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).

[6] Lombardi, Stephen, et al. "Deep appearance models for face rendering." *ACM Transactions on Graphics (TOG)* 37.4 (2018): 1-13.

[7] Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., & Sheikh, Y. (2019). Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*.

[8] Gafni, Guy, et al. "Dynamic neural radiance fields for monocular 4d facial avatar reconstruction." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

[9] Park, Keunhong, et al. "Nerfies: Deformable neural radiance fields." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.

[10] Oechsle, Michael, Songyou Peng, and Andreas Geiger. "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction." *arXiv preprint arXiv:2104.10078* (2021).

[11] https://www.eisko.com/louise/virtual-model

[12] https://www.unrealengine.com/en-US/digital-humans