NeRFlame: Flame-based conditioning of NeRF for 3D face rendering

Wojciech Zając[1⋆[0009−0009−1948−9532]], Joanna Waczyńska[1⋆[0009−0003−8593−9307]], Piotr Borycki[1[0000−0002−5715−4428]], Jacek Tabor[1[0000−0001−6652−7727]], Maciej Zieba[2[0000−0003−4217−7712]], and Przemysław Spurek[1,3[0000−0003−0097−5521]]

[1] Jagiellonian University, Faculty of Mathematics and Computer Science, Krakow, Poland
[2] Wroclaw University of Science and Technology, Wroclaw, Poland
[3] IDEAS NCBR {przemyslaw.spurek}@uj.edu.pl

Abstract. Traditional 3D face models are based on mesh representations with texture. One of the most important models is Flame (Faces Learned with an Articulated Model and Expressions), which produces meshes of human faces that are fully controllable. Unfortunately, such models have problems with capturing geometric and appearance details. In contrast to mesh representation, the neural radiance field (NeRF) produces extremely sharp renders. However, implicit methods are hard to animate and do not generalize well to unseen expressions. It is not trivial to effectively control NeRF models to obtain face manipulation.
The present paper proposes a novel approach, named NeRFlame, which combines the strengths of both NeRF and Flame methods. Our method enables NeRF to have high-quality rendering capabilities while offering complete control over the visual appearance, similar to Flame. In contrast to traditional NeRF-based structures that use neural networks for RGB color and volume density modeling, our approach utilizes the Flame mesh as a distinct density volume. Consequently, color values exist only in the vicinity of the Flame mesh. Our model's core concept involves adjusting the volume density based on its proximity to the mesh. This Flame framework is seamlessly incorporated into the NeRF architecture for predicting RGB colors, enabling our model to represent volume density explicitly and implicitly capture RGB colors.

Keywords: NeRF · Flame · Avatar 3D.

1 Introduction

Methods to automatically create fully controllable human face avatars have many applications in VR/AR and games [18]. Traditional 3D face models are based on fully controllable mesh representations. Flame [25] is a method used for mesh-based avatars [23,42]. Flame integrates a linear shape space trained using 3800 human head scans with articulated jaw, neck, eyeballs, pose-dependent corrective blend shapes, and extra global expression blend shapes. In practice, we can easily train Flame on the 3D scan (or 2D image) of human faces and then manipulate basic behaviors like jaw, neck, and eyeballs.

⋆ equal contribution

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

We can also produce colors for mesh by using textures. Unfortunately, such models have problems with capturing geometric and appearance details.

In contrast to the classical approaches, we can use implicit methods that represent avatars using neural networks. NeRFs [29] represent a scene using a fully connected architecture. As input, they take a 5D coordinate (spatial location $\mathbf{x} = (x, y, z)$ and viewing direction $\mathbf{d} = (\theta, \Psi)$) and return an emitted color $\mathbf{c} = (r, g, b)$ and volume density σ. NeRF extracts information from unlabelled 2D views to obtain 3D shapes. NeRF allows for the synthesizing of novel views of complex 3D scenes from a small subset of 2D images. Based on the relations between those base images and computer graphics principles, such as ray tracing, this neural network model can render high-quality images of 3D objects from previously unseen viewpoints. In contrast to the mesh representation, NeRF captures geometric and appearance details. However, it is not trivial to effectively control NeRF to obtain face manipulation.

There are many approaches to controlling NeRF, including generative models [21,38], dynamic scene encoding [22], or conditioning mechanisms [4]. However, our ability to manipulate NeRF falls short compared to our proficiency in controlling mesh representations.

This paper proposes NeRFlame, a hybrid approach for 3D face rendering that uses implicit and explicit representations; see Fig. 1. Our model is based on two components: NeRF and mesh, with only points in the mesh surroundings treated as NeRF inputs. Our method inherits the best features from the above approaches by modeling the quality of NeRF rendering and controlling the appearance as in Flame. We combine those two techniques by showing how to condition the NeRF model by mesh effectively. Our model's fundamental idea is conditioning volume density by distance to mesh. The volume density is non-zero only in the ε neighborhood of Flame mesh. Therefore, we use the NeRF-based architecture to model volume density and RGB colors only in the ε neighborhood of the mesh. Such a solution allows one to obtain renders of similar quality to NeRF and a level of control mesh similar to Flame (Tab. 1). In contrast to Dynamic Neural Radiance Fields, NeRFlame undergoes training using a single position of the human face rather than relying on sequences from various positions in movies. Despite this distinction, our model exhibits comparable functionality. We have the capability to generate a range of facial expressions and novel perspectives for newly encountered face positions facilitated by the Flame backbone. This enables us to model

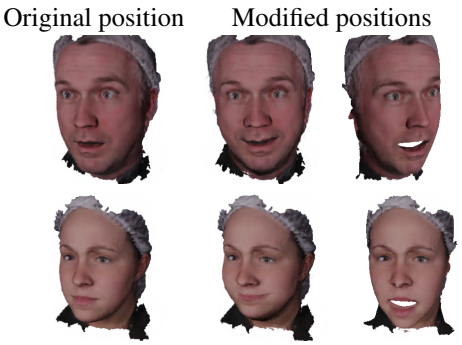

Fig. 1. Our model facilitates the manipulation of facial attributes in the context of human visage. NeRFlame use Flame as a conditioning factor in the NeRF-based model. We can produce novel views in training positions as well as in modified facial expressions.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

previously unseen facial expressions. Consequently, we conduct a comparison between our model and the traditional static Neural Radiance Fields.

The contributions of this paper are significant and are outlined as follows:

- We introduce NeRFlame – an innovative NeRF model conditioning be Flame that combines the best features of both methods, namely the exceptional rendering quality of NeRF and the precise control over appearance as in Flame.
- We demonstrate the ability to condition model volume density in NeRF by employing mesh representation, which represents a significant advancement over traditional NeRF-based approaches that rely on neural networks.
- We train our model on a single position of the human face rather than using entire movies, thereby highlighting the versatility and practicality of our approach.

Overall, our contributions offer a substantial advancement in the realms of 3D facial modeling and rendering, providing a foundation for future exploration and research in this domain.

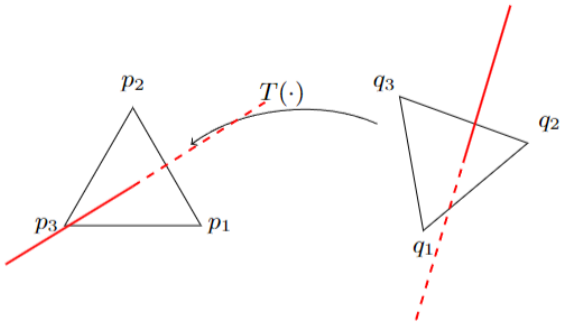

Fig. 2. Visualization of transformation in NeRFlame. We aim to aggregate colors along the ray during rendering in the new position (see the red line in the right image). NeRFlame uses Flame mesh, therefore we can localize the face's vertex, which is crossed with the ray p_1, p_2, p_3 and the corresponding triangle in the initial position mesh q_1, q_2, q_3. Thanks to such pairs of points, we estimate affine transformation T, which is used to find the ray in the initial position (see the red line in the left image).

2 Related Works

NeRFlame is a model of controllable human face avatars trained on a single 3D face represented by a few 2D images. Given our model's training on 2D facial images, we naturally refer to Static Neural Radiance Fields. However, our ability to adapt NeRF is noteworthy due to the utilization of Flame as a backbone. Consequently, in our exploration of related works, we incorporate considerations for Dynamic Neural Radiance Fields.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

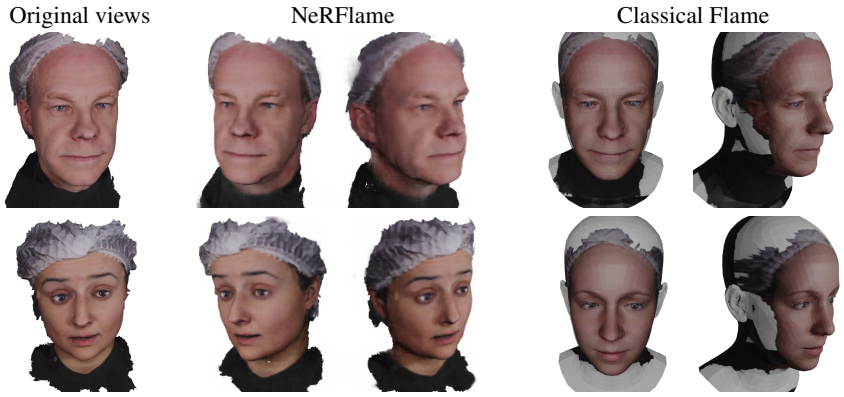

Fig. 3. Competition between NeRFlame and classical Flame fitting. NeRF-based model better fits human expression of the face.

Static Neural Radiance Fields 3D objects can be represented by using many different approaches, including voxel grids [10], octrees [19], multi-view images [3,27], point clouds [1,34,41], geometry images [35], deformable meshes [17,25], and part-based structural graphs [24].

The above representations are discreet, which causes some problems in real-life applications. In contrast to such apprehension, NeRF [29] represents a scene using a fully-connected architecture. NeRF and many generalizations [7,8,26,31,32,36,39] synthesize novel views of a static scene using differentiable volumetric rendering.

One of the largest limitations is training time. To solve such problems in [14], authors propose Plenoxels, a method that uses a sparse voxel grid storing density and spherical harmonics coefficients at each node. The final color is the composition of tri-linearly interpolated values of each voxel. In [30], authors use a similar approach, but the space is divided into an independent multilevel grid. In [9], authors represent a 3D object as an orthogonal tensor component. A small MLP network, which uses orthogonal projection on tensors, obtains the final color and density. Some methods use additional information to NeRF, like depth maps or point clouds [6,12,32,40].

In our paper, we produce a new NeRF-based representation of 3D objects. As input, we use classical 2D images. However, RGB colors and volume density are conditioned by distance to Flame mesh.

Dynamic Neural Radiance Fields Current solutions for implicit reprehension of human face avatars are trained on image sequences. We assume that we have external tools for segmenting frames in the movie. We often use additional information like each frame's camera angle or Flame representation.

In [15], authors implicitly model the facial expressions by conditioning the NeRF with the global expression code obtained from 3DMM tracking [37]. In [44], authors leverage the idea of dynamic neural radiance fields to improve the mouth region's rendering, which is not represented by the face model motion prior. The IMAvatar [42] model learns the subject-specific implicit representation of texture together with expression. In

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

[16] authors use neural graphics primitives, where for each of the blend shapes, a multi-resolution grid is trained. In RigNeRF [5] authors propose a model that changes head pose and facial expressions using a deformation field that is guided by a 3D morphable face model (3DMM).

Diverging from the previously mentioned approach, certain researchers adopt an explicit apprehension strategy. In the case of [2], the authors introduced ClipFace, a method facilitating text-guided editing of textured 3D face models. In [23], a one-shot mesh-based model reconstruction is presented, while in [45,11], a model is proposed that draws upon a blend of 2D and 3D datasets.

Our method is situated between the above approaches. Similarly to the explicit approach, we use a single 3D object instead of movies to train. We also use Flame mesh to edit the avatar's shape end expressions. On the other hand, we use an implicit representation of the colors of objects.

3 NeRFlame: Flame-based conditioning of NeRF for 3D face rendering

In this subsection, we introduce NeRFlame - the novel 3D face representation that combines the benefits of Flame and NeRF models. We first provide the details about the Flame and NeRF approaches and further describe the concept of NeRFlame and how it can be used to control face mesh.

3.1 Flame

Flame (Faces Learned with an Articulated Model and Expressions) [25] is a 3D facial model trained from thousands of accurately aligned 3D scans. The model is factored in that it separates the representation of identity, pose, and facial expression, similar to the human body approach. It is represented by low polygon count, articulation, and blend skinning that is computationally efficient, compatible with existing game and rendering engines, and simple in order to maintain its practicality. The parameters of the model are trained by optimizing the reconstruction loss, assuming a detailed temporal registration of our template mesh with three unconnected components, including the base face and two eyeballs.

Formally, the Flame is a function from human face parametrization $\mathcal{F}_{Flame}(\beta, \psi, \phi)$ where β, ψ and ϕ describe shape, expression, and pose parameters to a mesh with n vertices:

$$\mathcal{F}_{Flame}(\beta, \psi, \phi) : \mathbb{R}^{k_\beta \times k_\psi \times k_\phi} \rightarrow \mathbb{R}^{n \times 3},$$

where k_β, k_ψ, and k_ϕ are the numbers of shape, expression, and pose parameters. In the classical version, we can fit our model to 3D scans or 2D images by using facial landmarks. Many strategies exist to choose landmarks and parameters for its training [25]. However, in the high level of generalization for input I – image 3D scan (or 2D image) and arbitrarily chosen method for estimation facial landmark points $\mathcal{LP}$ we minimize L2 distance

$$\min_{(\beta,\psi,\phi)} \|\mathcal{LP}(\mathcal{F}_{Flame}(\beta, \psi, \phi)) - \mathcal{LP}(I)\|_2$$

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

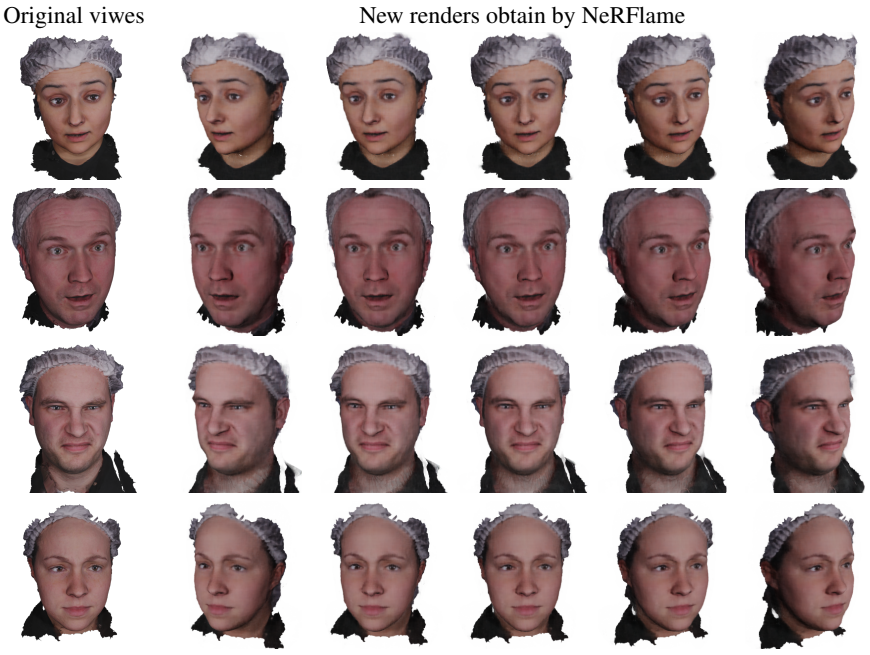

Fig. 4. Reconstruction of 3D object obtained by NeRFlame. As we can see, NeRFlame model the detailed appearance of the 3D face.

Such an approach is effective, but there are a few limitations. Localizing landmarks and choosing which parameters to optimize first is not trivial. On the other hand, for 2D images, the results are not well-qualified. We can use a pre-trained auto-encoder-based model DECA [13] for face reconstruction from 2D images to solve such problems.

In this paper, we train the Flame-based model in a NeRF-based scenario. As input, we take a few 2D images. As an effect, we obtain a correctly fitted Flame model and NeRF rendering model for new views.

3.2 NeRF

NeRF representation of 3D objects NeRFs [29] are the model for representing complex 3D scenes using neural architectures. In order to do that, NeRFs take a 5D coordinate as input, which includes the spatial location $\mathbf{x} = (x, y, z)$ and viewing direction $\mathbf{d} = (\theta, \Psi)$ and returns emitted color $\mathbf{c} = (r, g, b)$ and volume density σ.

A classical NeRF uses a set of images for training. In such a scenario, we produce many rays traversing through the image and a 3D object represented by a neural network. NeRF parameterized by $\mathbf{\Theta}$ approximates this 3D object with an MLP network:

$$\mathcal{F}_{NeRF}(\mathbf{x}, \mathbf{d}; \mathbf{\Theta}) = (\mathbf{c}, \sigma).$$

The model is trained to map each input 5D coordinate to its corresponding volume density and directional emitted color.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

The loss of NeRF is inspired by classical volume rendering [20]. We render the color of all rays passing through the scene. The volume density $\sigma(\mathbf{x})$ can be interpreted as the differential probability of a ray. The expected color $C(\mathbf{r})$ of camera ray $\mathbf{r}(t) = \mathbf{o} + td$ (where $\mathbf{o}$ is ray origin and d is direction) can be computed with an integral.

In practice, this continuous integral is numerically estimated using a quadrature. We use a stratified sampling approach where we partition our ray $[t_n, t_f]$ into N evenly-spaced bins and then draw one sample t_i uniformly at random from within each bin. We use these samples to estimate $C(\mathbf{r})$ with the quadrature rule [28], where $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i, \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right),$$

This function for calculating $\hat{C}(\mathbf{r})$ from the set of $(\mathbf{c_i}, \sigma_\mathbf{i})$ values is trivially differentiable.

We then use the volume rendering procedure to render the color of each ray from both sets of samples. Contrary to the baseline NeRF [29], where two "coarse" and "fine" models were simultaneously trained, we use only the "coarse" architecture. Our loss is simply the total squared error between the rendered and true pixel colors

$$\mathcal{L} = \sum_{\mathbf{r}\in R} \|\hat{C}(\mathbf{r}) - C(\mathbf{r})\|_2^2 \tag{1}$$

where R is the set of rays in each batch, and $C(\mathbf{r})$, $\hat{C}(\mathbf{r})$ are the ground truth and predicted RGB colors for ray $\mathbf{r}$ respectively.

3.3 NeRFlame

We introduce NeRFlame the 3D face model that combines the benefits of mesh representations from Flame and NeRF implicit representation of 3D objects. Thanks to the application of NeRFs, we can estimate the parameters of Flame directly from 2D images without using landmarks points. On the other hand, we obtain NeRF model, which can be editable similarly to Flame. In order to achieve that, we introduce the function that approximates the volume density using the Flame model.

Consider the distance function $d(\mathbf{x}, \mathcal{M})$ between point $\mathbf{x} = (x, y, z) \in \mathbb{R}^3$ and the mesh $\mathcal{M} := \mathcal{M}_{\beta,\psi,\phi}$ created by $F_{Flame}(\beta, \psi, \phi)$, with parameters β, ψ, ϕ. Note, that edges between vertices in Flame model can be directly taken from the template mesh (see [4] for details). We define the volume density function as:

$$\sigma(\mathbf{x}, \mathcal{M}) = \begin{cases} 0, & \text{if } d(\mathbf{x}, \mathcal{M}) > \varepsilon \\ (1 - \frac{1}{\varepsilon} d(\mathbf{x}, \mathcal{M})), & \text{otherwise,} \end{cases} \tag{2}$$

where ε is a hyperparameter that defines the neighborhood of the mesh surface. In practice, the values of the density volume function are non-zero only in the close neighborhood of the mesh.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

The NeRFlame can be represented by the function:

$$\mathcal{F}_{NeRFlame}(\mathbf{x};\beta,\psi,\phi,\Theta) = (\mathcal{F}^{c}_{\Theta}(\mathbf{x}), \sigma(\mathbf{x},\mathcal{M})), \tag{3}$$

where $\mathcal{F}^{c}_{\Theta}$ is the MLP that predicts the color, similar to the NeRF model.

The model is trained in an end-to-end manner directly, optimizing the criterion (1) with respect to the MLP parameters Θ, and Flame parameters β, ψ, ϕ, which describe shape, expression, and pose. In NeRFlame, we utilize the original loss function used to train NeRF models. Therefore, the structure of colors on rays must be consistent. During training, the model modified the mesh structure to be consistent with the 3D structure of the target face. The simultaneous neural network $\mathcal{F}$ produces colors for the rendering procedure.

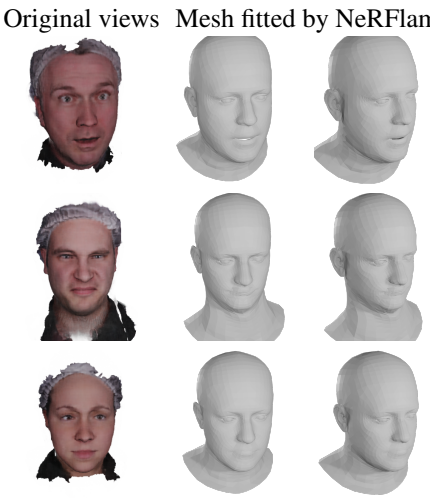

Fig. 5. During the training of NeRFlame, we simultaneously model Flame mesh and NeRf dedicated to colors. In the above figure, we present the meshes fitted by NeRFlame.

During training, we can see the trade-off between the level of mesh fitting and the quality of renders. The main reason is that we must train our model with small ε, and color must be encoded in a small neighborhood of the mesh. Therefore, the quality of the model is lower than classical NeRF. With larger ε we obtain mesh which is not correctly fitted.

Therefore, we use the MLP $\mathcal{F}^{\sigma}_{\Theta}$ that predicts the volume density analogical to NeRF. In the first 10000 epoch we train our model with volume density given by formula (2). Then, Flame parameters are frozen, and we train only NeRF component with volume density given by

$$\sigma(\mathbf{x},\mathcal{M},\mathcal{F}^{\sigma}_{\Theta}) = \begin{cases} 0, & \text{if } d(\mathbf{x},\mathcal{M}) > \varepsilon \\ \mathcal{F}^{\sigma}_{\Theta}(\mathbf{x}), & \text{otherwise,} \end{cases}$$

where ε is a hyperparameter that defines the neighborhood of the mesh surface and F^{σ}_{Θ} is MLP that predicts volume density. The value of ε increases gradually over time during training after 10 000 epochs. The value of ε increases up to $\varepsilon = 0.1$ at the end of the training procedure.

3.4 Controlling NeRF models to obtain face manipulation

The classical NeRF method is known to generate highly detailed and realistic images. However, it can be challenging to manipulate NeRF models to achieve precise facial modifications. Several techniques, such as generative models, dynamic scene encoding, and conditioning mechanisms, have been proposed to address this challenge. Nonetheless,

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

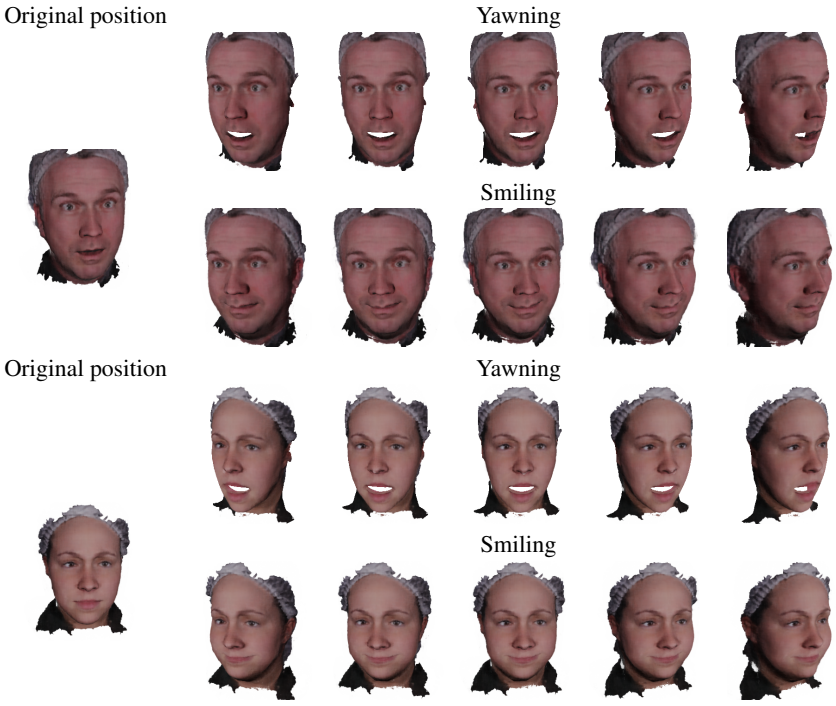

Fig. 6. Our model allows producing manipulation of the human face. In NeRFlame, we use Flame for volume density rendering. Therefore, we can manipulate Flame features and modify NeRF representation. In the figure, we show two faces and their versions with open mouths and changing expressions from different views.

controlling NeRF models to the same extent as mesh representations remains elusive. In contrast, the Flame is a straightforward model with three parameters, namely β, ψ, and ϕ, representing shape, expression, and pose, respectively. By performing simple linear operations on these parameters, it is possible to rotate the avatar, change facial expressions, and adjust facial features to a certain degree.

Our NeRFlame is built on the Flame model so that we can manipulate our Flame model to control density prediction σ. However, accurately predicting the RGB colors of the modified object is challenging. To solve this problem, we use a simple technique. To predict color after transformation, we return to the initial position where the color is known, see Fig. 2.

Let us consider NeRFlame model, which is already trained. We have parameters $\beta_1, \psi_1, \phi_1, \Theta, \varepsilon$, function

$$\mathcal{F}_{NeRFlame}(\mathbf{x}; \beta_1, \psi_1, \phi_1, \Theta)$$

and fitted mesh $\mathcal{M}_1$, created by the Flame from parameters β_1, ψ_1, ϕ_1. Let us assume that we apply some modification of Flame parameters, which means that we obtain new β_2, ψ_2, ϕ_2. Using these parameters, we can create the modified mesh $\mathcal{M}_2$, simply using

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

the Flame model. Instead of retraining the NeRF model for new parameters requiring the 2D images for a new pose, we propose applying a simple transformation between the modified and original space.

Let's take a mesh $\mathcal{M}_2$ representing the new pose. We postulate using the affine transformation $T(\cdot)$, which transforms the point $\mathbf{x}_2$ on the mesh $\mathcal{M}_2$ to the original pose: $T(\mathbf{x}_2) = \mathbf{x}_1$, where $\mathbf{x}_1 \in \mathcal{M}_1$ is the corresponding point on original pose. After the transformation, we can identify an element in the original pose $\mathcal{M}_1$ for each element on the mesh $\mathcal{M}_2$. In practice, finding the transformation T is not a trivial task since it depends on the local transformation of the mesh.

However, for a given point $\mathbf{x}_1 \in \mathcal{M}_1$, we can find a face triangle described by vertices $(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) \in V_1$ that contains $\mathbf{x}_1$, where V_1 is the set of vertices of the mesh $\mathcal{M}_1$. Because Flame model is shifting the vertices of the model keeping the connections unchanged, we can locate the corresponding triangle $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) \in V_2$ of the mesh $\mathcal{M}_2$. For each of the triangles that create the mesh, we define affine transformation:

$$T(\mathbf{p}_i) = \mathbf{q}_i, \text{ for } i = 1, 2, 3.$$

In such a situation, we assume that such transformations $T(\cdot)$ are affine, and we can use Least-Squares Conformal Multilinear Regression [33] to estimate the parameters. Practically, finding the transformation for each of the triangles is extremely fast, requires inverting a fourth-dimensional matrix, and can be parallelized. Having the parameters of transformations estimated, we can apply them directly to the $\mathbf{x}$ in NeRFlame model given by formula (3) and calculate the colors as in an unshifted pose.

Our approach is sensitive to some particular facial manipulations. When we open the mouth of our avatar, we obtain artifacts. Three main reasons cause such problems. First, it is difficult to fit the mouth around the mesh to images since it is very sensitive to perturbations. Additionally, the mesh lacks internal content, and it cannot represent the inside of the mouth and tongue. The third problem is that rays go through the open mouth, cut the mesh back of the head, and render some artifacts.

To solve such a problem, we remove rays through an open-mouth region. Such a solution is simple to implement since we can easily filter rays that do not cross mesh. On the other hand, allows for reducing most of the artifacts.

4 Experiments

In this section, we describe the experimental results of the proposed model. To our knowledge, it is the first model that obtains editable NeRF trains on a single object in one position. Most of the methods use movies to encode many different positions of the face. We can produce novel views in training positions and in modified facial expressions using knowledge only from one fixed position. Therefore, it is hard to compare our results to other algorithms. In the first subsection, we show that our model produces high-quality NeRF representations of the objects by comparing our model with our baseline classical NeRF and classical textured Flame. In the second subsection, we present meshes obtained by our model. Finally, we show that our model allows facing manipulations.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

	PSNR ↑			SSIM ↑			LPIPS ↓		
	NeRF	NeRFlame	Flame	NeRF	NeRFlame	Flame	NeRF	NeRFlame	Flame
Face 1	33.37	27.89	9.67	0.96	0.95	0.76	0.05	0.09	0.26
Face 2	33.39	29.79	12.44	0.96	0.96	0.80	0.05	0.06	0.24
Face 3	33.08	29.70	12.97	0.97	0.95	0.82	0.04	0.08	0.20
Face 4	31.96	25.78	12.51	0.96	0.92	0.79	0.04	0.10	0.23
Face 5	33.15	32.59	11.30	0.96	0.96	0.77	0.05	0.05	0.26
Face 6	32.42	29.18	11.45	0.96	0.95	0.76	0.06	0.07	0.26

Table 1. Comparison of PSNR, SSIM, and LPIPS matrices between our model and NeRF and Flame baselines. While NeRF achieves better results, it lacks manipulation capabilities. In contrast, the Flame model produces inferior outcomes as it is trained solely on landmark points for mesh and texture.

Since the current literature does not provide suitable data sets for evaluating the NeRF-based model for modeling 3D face avatars, we created a data set using 3D scenes. We create a classical NeRF training data set. We construct 200×200 transparent background images from random positions.

4.1 Reconstruction Quality

In this subsection, we show that NeRFlame can reconstruct a 3D human face with similar quality as classical NeRF. Since we train our model on a single position, it is difficult to compare our model to dynamic neural radiance fields. Therefore we show that our model has a slightly lower quality than classical NeRF but allows dynamic modification. On the other hand, we show that we obtain better results than textured Flame, which cannot capture the geometry and appearance details of the human face.

In Fig 3, we compare NeRFlame and textured Flame. As we can see, NeRFlame can reproduce facial features and geometry. On the other hand, Flame produces well-suited textures, but the mesh is not well-suited. In Tab. 1, we present a numerical comparison. We compare the metric reported by NeRF called PSNR (*peak signal-to-noise ratio*), SSIM (*structural similarity index measure*), LPIPS (*learned perceptual image patch similarity*) used to measure image reconstruction effectiveness. As we can see NeRF gives essentially better since do not allow manipulation. On the other hand, Flame model gives inferior results since we train mesh and texture only on landmark points. In Fig. 4, we present new renders of the model obtained by NeRFlame. As we can see, NeRFlame model the detailed appearance of the 3D face.

4.2 Mesh fitting

The RGB colors generated by NeRF are present only in the ε vicinity of the mesh. This approach allows for a precise fitting of the mesh to the human face, which is critical for generating animated models. In Fig. 5, we present the rendered faces and corresponding meshes produced by our NeRFlame approach. The results demonstrate that our method can accurately capture the underlying mesh structure.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

Fig. 7. The renders obtained by NeRFlame on MoFaNeRF dataset.

4.3 Face manipulation

Our approach enables the manipulation of human facial features. Leveraging Flame as a backbone, NeRFlame offers the ability to manipulate Flame features and modify NeRF representations. In Fig. 6, we showcase three faces and their modification including open mouths and changing expressions, that can be manipulated using our model in a manner similar to the classical Flame model.

NeRFlame simultaneously train mesh and NeRF components for color. After training, we can exchange the produced mesh in our model to obtain a modification of the final look of the avatar, see Fig. 1.

4.4 Comparison with MoFaNeRF [43]

Comparison between MoFaNeRF [43] and NeRFlame is not possible directly. MoFaNeRF is trained on a large dataset. MoFaNeRF is a generative model witch ability to control face position and expression. NeRFlame is a classical NeRF-based model trained individually on each element of the dataset separately.

This experiment shows that our NeRFlame reconstruct objects with a PSNR value similar to that of MoFaNeRF. In Fig. 7, we present faces trained on elements from the MoFaNeRF dataset. In Tab. 2, we present numerical experiments, but it should be noted that the models were trained entirely differently on different parts of the data set. It only states that we have similar render quality.

Model	PSNR(dB)	SSIM	LPIPS
FaceScape	27.96±1.34	0.932±0.012	0.069±0.009
i3DMM	24.45±1.58	0.904±0.014	0.112±0.015
MoFaNeRF	31.49±1.75	0.951±0.010	0.061±0.011
MoFaNeRF-fine	30.17±1.71	0.935±0.013	0.034±0.007
NeRFlame*	29.57	0.955	0.060

Table 2. Quantitative evaluation of representation ability. Results are from MoFaNeRF experiments [43]. * Comparison between MoFaNeRF [43] and NeRFlame is not possible directly. It should be highlighted that models were trained entirely differently on different dataset parts. It only states that we have similar-quality of renders.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

5 Conclusions

In this work, we introduce a novel approach called NeRFlame, which combines NeRF and Flame to achieve high-quality rendering and precise pose control. While NeRF-based models use neural networks to model RGB colors and volume density, our method utilizes an explicit density volume represented by the Flame mesh. This allows us to model the quality of NeRF rendering and accurately control the appearance of the final output. As a result of offering complete control over the model, the quantitative performance of our approach is marginally inferior to that of a static NeRF model. We believe that future work should prioritize advancements in mesh fitting techniques. By doing so, we can maximize the potential for extensive modifications.

6 Acknowledgments

The work of P. Spurek was supported by the National Centre of Science (Poland) Grant No. 2021/43/B/ST6/01456. The work of M. Zieba was supported by National Science Centre of Poland, grant no 2020/37/B/ST6/03463. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016357. The research of P. Borycki was funded by the program Excellence Initiative–Research University at the Jagiellonian University in Kraków.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: International conference on machine learning. pp. 40–49. PMLR (2018)
2. Aneja, S., Thies, J., Dai, A., Nießner, M.: Clipface: Text-guided editing of textured 3d morphable models. arXiv preprint arXiv:2212.01406 (2022)
3. Arsalan Soltani, A., Huang, H., Wu, J., Kulkarni, T.D., Tenenbaum, J.B.: Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1511–1519 (2017)
4. Athar, S., Shu, Z., Samaras, D.: Flame-in-nerf: Neural control of radiance fields for free view face animation. arXiv preprint arXiv:2108.04913 (2021)
5. Athar, S., Xu, Z., Sunkavalli, K., Shechtman, E., Shu, Z.: Rignerf: Fully controllable neural 3d portraits. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20364–20373 (2022)
6. Azinović, D., Martin-Brualla, R., Goldman, D.B., Nießner, M., Thies, J.: Neural rgb-d surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6290–6301 (2022)
7. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5855–5864 (2021)
8. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

9. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII. pp. 333–350. Springer (2022)
10. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European conference on computer vision. pp. 628–644. Springer (2016)
11. Daněček, R., Black, M., Bolkart, T.: Emoca: Emotion driven monocular face capture and animation. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20279–20290. IEEE (2022)
12. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12882–12891 (2022)
13. Feng, Y., Feng, H., Black, M.J., Bolkart, T.: Learning an animatable detailed 3d face model from in-the-wild images. ACM Transactions on Graphics (ToG) **40**(4), 1–13 (2021)
14. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5501–5510 (2022)
15. Gafni, G., Thies, J., Zollhofer, M., Nießner, M.: Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8649–8658 (2021)
16. Gao, X., Zhong, C., Xiang, J., Hong, Y., Guo, Y., Zhang, J.: Reconstructing personalized semantic facial nerf models from monocular video. ACM Transactions on Graphics (TOG) **41**(6), 1–12 (2022)
17. Girdhar, R., Fouhey, D.F., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. In: European Conference on Computer Vision. pp. 484–499. Springer (2016)
18. Grassal, P.W., Prinzler, M., Leistner, T., Rother, C., Nießner, M., Thies, J.: Neural head avatars from monocular rgb videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18653–18664 (2022)
19. Häne, C., Tulsiani, S., Malik, J.: Hierarchical surface prediction for 3d object reconstruction. In: 2017 International Conference on 3D Vision (3DV). pp. 412–420. IEEE (2017)
20. Kajiya, J.T., Von Herzen, B.P.: Ray tracing volume densities. ACM SIGGRAPH computer graphics **18**(3), 165–174 (1984)
21. Kania, A., Kasymov, A., Zięba, M., Spurek, P.: Hypernerfgan: Hypernetwork approach to 3d nerf gan. arXiv preprint arXiv:2301.11631 (2023)
22. Kania, K., Yi, K.M., Kowalski, M., Trzciński, T., Tagliasacchi, A.: Conerf: Controllable neural radiance fields. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18602–18611. IEEE (2022)
23. Khakhulin, T., Sklyarova, V., Lempitsky, V., Zakharov, E.: Realistic one-shot mesh-based head avatars. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II. pp. 345–362. Springer (2022)
24. Li, J., Xu, K., Chaudhuri, S., Yumer, E., Zhang, H., Guibas, L.: Grass: Generative recursive autoencoders for shape structures. ACM Transactions on Graphics (TOG) **36**(4), 1–14 (2017)
25. Li, T., Bolkart, T., Black, M.J., Li, H., Romero, J.: Learning a model of facial shape and expression from 4d scans. ACM Trans. Graph. **36**(6), 194–1 (2017)
26. Liu, L., Gu, J., Zaw Lin, K., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. Advances in Neural Information Processing Systems **33**, 15651–15663 (2020)
27. Liu, Z., Zhang, Y., Gao, J., Wang, S.: Vfmvac: View-filtering-based multi-view aggregating convolution for 3d shape recognition and retrieval. Pattern Recognition **129**, 108774 (2022)
28. Max, N.: Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics **1**(2), 99–108 (1995)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24

29. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
30. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)
31. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S., Geiger, A., Radwan, N.: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5480–5490 (2022)
32. Roessle, B., Barron, J.T., Mildenhall, B., Srinivasan, P.P., Nießner, M.: Dense depth priors for neural radiance fields from sparse input views. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12892–12901 (2022)
33. Schmid, K.K., Marx, D.B., Samal, A.: Tridimensional regression for comparing and mapping 3d anatomical structures. Anatomy Research International **2012** (2012)
34. Shu, D.W., Park, S.W., Kwon, J.: Wasserstein distributional harvesting for highly dense 3d point clouds. Pattern Recognition **132**, 108978 (2022)
35. Sinha, A., Bai, J., Ramani, K.: Deep learning 3d shape surfaces using geometry images. In: European Conference on Computer Vision. pp. 223–240. Springer (2016)
36. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-nerf: Scalable large scene neural view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8248–8258 (2022)
37. Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., Niessner, M.: Face2face: Real-time face capture and reenactment of rgb videos. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2387–2395. IEEE Computer Society (2016)
38. Trzciński, T.: Points2nerf: Generating neural radiance fields from 3d point cloud. arXiv preprint arXiv:2206.01290 (2022)
39. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-nerf: Structured view-dependent appearance for neural radiance fields. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5481–5490. IEEE (2022)
40. Wei, Y., Liu, S., Rao, Y., Zhao, W., Lu, J., Zhou, J.: Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5610–5619 (2021)
41. Yang, F., Davoine, F., Wang, H., Jin, Z.: Continuous conditional random field convolution for point cloud segmentation. Pattern Recognition **122**, 108357 (2022)
42. Zheng, Y., Fernández Abrevaya, V., Bühler, M., Chen, X., Black, M.J., Hilliges, O.: Im avatar: Implicit morphable head avatars from videos. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13535–13545. IEEE (2022)
43. Zhuang, Y., Zhu, H., Sun, X., Cao, X.: Mofanerf: Morphable facial neural radiance field. In: European Conference on Computer Vision. pp. 268–285. Springer (2022)
44. Zielonka, W., Bolkart, T., Thies, J.: Instant volumetric head avatars. arXiv preprint arXiv:2211.12499 (2022)
45. Zielonka, W., Bolkart, T., Thies, J.: Towards metrical reconstruction of human faces. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII. pp. 250–269. Springer (2022)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63749-0_24