Warsaw University of Technology

FACULTY OF ELECTRONICS AND INFORMATION TECHNOLOGY

PhD Thesis

in the discipline of Information and Communication Technology

Few-Shot Human Neural Rendering with Partial Information

Kacper Kania, M.Sc.

supervisor
Tomasz Trzciński, Prof. PhD DSc.

assistant supervisor
Marek Kowalski, PhD DSc.

WARSZAWA 2025

Acknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Abstract

This thesis is a series of publications that introduce novel methods for human neural rendering using limited information, focusing on Neural Radiance Fields (NeRFs) and 3D Gaussian Splatting (3DGS). It explores how these models construct 3D representations from 2D images and demonstrates ways to condition these representations for generating high-quality human renderings. We propose techniques that use simple, interpretable inputs derived from sparse training data and extends these methods to perform effectively in few-shot learning scenarios.

We begin by examining the field of neural radiance fields, addressing limitations in existing approaches and presenting contributions to controllable radiance fields. By incorporating partial and sparse data during training, it leverages the smoothness of neural networks to produce controllable, high-quality human images.

To tackle the reliance on extensive, high-quality data annotations from multi-view videos, we introduce a new method for training neural radiance fields in few-shot, multi-view settings. This approach learns internal deformation templates, which blend smoothly during inference, significantly improving image quality compared to existing baselines and enabling effective human rendering from limited input images.

The work also addresses the need for adaptable computational efficiency during inference. It proposes a fine-to-coarse learning strategy for 3D Gaussian Splatting, which upscales a latent 2D grid that stores Gaussian representations. This strategy achieves competitive results while allowing deployment on various computational devices with minimal quality loss.

In addition, we develop a novel model for controlling radiance fields through environmental lighting. By incorporating precomputed radiance transfer, this model enables physically plausible scene relighting and provides users with intuitive control over lighting in reconstructed scenes.

This research advances the state of the art in controllable neural radiance fields and expands their application to few-shot learning scenarios. These innovations enhance the possibilities for human rendering from limited information and open new directions for future research in the field.

Keywords: Neural Rendering, Neural Radiance Fields, Few-Shot Learning, Human Rendering, Partial Information, Gaussian Splatting

Streszczenie

To jest streszczenie. To jest trochę za krótkie, jako że powinno zająć całą stronę.

Słowa kluczowe: A, B, C

Lay Summary

ok

Publications in this thesis

Title	Authors	Venue	Status
CoNeRF: Controllable Neural Radiance Fields	**Kacper Kania**, Kwang Moo Yi, Marek Kowalski, Tomasz Trzciński, Andrea Tagliasacchi	CVPR 2022	Accepted
BlendFields: Few-Shot Example-Driven Facial Modeling	**Kacper Kania**, Stephan J. Garbin, Andrea Taglisacchi, Virginia Estellers, Kwang Moo Yi, Julien Valentin, Tomasz Trzciński, Marek Kowalski	CVPR 2023	Accepted
LumiGauss: High-Fidelity Outdoor Relighting with 2D Gaussian Splatting	Joanna Kaleta, **Kacper Kania**, Tomasz Trzciński, Marek Kowalski	WACV 2025	Accepted
CLoG: Leveraging UV Space for Continuous Levels of Detail	**Kacper Kania**, Rawal Khirodkar, Shunsuke Saito, Kwang Moo Yi, Julieta Martinez	CVPR 2025	Under Review

Contents

List of Abbreviations and Symbols

List of Figures

List of Tables

3 **Ablation study** – First, we check the effect of the neighborhood size $|\mathcal{N}(\mathbf{v})|$ on the results. Below that, we compare the effect of smoothing. The best results are colored in ■ and the second best in ■. For the real dataset, changing the neighborhood size gives inconsistent results, while smoothing improves the rendering quality. In the synthetic scenario, setting $|\mathcal{N}(\mathbf{v})|$=20 and the Laplacian smoothing consistently gives the best results. The discrepancy between real and synthetic datasets is caused by inaccurate face tracking for the former. We describe this issue in detail in Sec. 4.5.4. 27

4 **Number of training expressions** – We ablate over the number of training expressions. We evaluate the model on the captures from the Multiface dataset [86]. We run the model for each possible expression combination for a given K and average the results. The best results are colored in ■ and the second best in ■. Increasing the number of available training expressions consistently improves the results. However, using K=5 expressions saturates the quality and using K>5 brings diminishing improvements. We do not report "Novel Pose Synthesis" for K>5 as we use validation expressions and poses to train those models (refer to Sec. 4.5.1 for more details). 31

5 **Quantitative results without masking** – Similarly to Tab. 2, we compare BlendFields to other related approaches. However, we calculate the results over the whole image space, without removing the background. BlendFields and VolTeMorph [19] model the background as a separate NeRF-based [48] network. The points that do not fall into the tetrahedral mesh are assigned to the background. As the network overfits to sparse training views, it poorly extrapolates to novel expressions (as the new head pose or expression may reveal some unknown parts of the background) and views. At the same time, all other baselines do not have any mechanism to disambiguate the background and the foreground. 31

π Stała matematyczna równa stosunkowi obwodu okręgu do jego średnicy

I Natężenie prądu elektrycznego

Chapter 1

Introduction

With the advent of deep learning, research have been exploring varying ways to apply it to computer graphics. One of the most recent and promising approaches is neural rendering. Neural rendering is a field that combines deep learning and computer graphics to generate realistic images of 3D scenes. The neural radiance field (NeRF) is a popular neural rendering technique that represents a 3D scene as a continuous function that maps 3D coordinates to radiance values. NeRF has shown impressive results in generating photorealistic images of 3D scenes. However, NeRF has limitations in terms of memory and computational requirements, which makes it difficult to scale to large scenes.

To alleviate the problem, Kerbl *et al.* [35] proposed a new technique—3D Gaussian Splatting (3DGS). 3DGS is a neural rendering technique that represents a 3D scene as a set of 3D Gaussian that are splatted to an image space using algorithm proposed by Zwicker *et al.* [107]. In contrast to NeRF, 3DGS is more memory efficient and can be used to render large scenes. It can also render scenes with millions of points in real-time on a single GPU.

In this thesis, we focus on those two milestone techniques in neural rendering and address their fundamental problem—lack of controllability.

1.1 Motivation and challenges

NeRF and 3DGS are both impressive techniques that can generate realistic images. However, a single scene representation needs to be trained on a high-end GPU for hours or even days just to render a novel view at the inference time. However, any type of controllability is difficult to achieve with those models. That includes changing the lighting conditions, subject's attributes or even the scene itself. We see imbuing those models with controllability as a an important step towards making them more useful in practice. Our proposed models are designed to address this issue.

One may ask why the controllability is a feat sought after to be researched. We see the inspiration in how human artists work. Imagine an artist working on 3D game where they

need an asset, like a 3D mesh, to be created. Such a mesh takes much effort since it includes modeling, creating a UV map which can then be textured. After the process is finish, the artist's supervisor may task him to change the model to some extent which requires the artist to redo all the effort again. Such a process is not limited to 3D assets as meshes and could be applied to 3DGS or NeRF. However, 3DGS and NeRFs are volumetric in nature. Our exploited and well-established practices no longer apply to them since volumetric representations do not have the underlying surface representation. For that reason, we see a couple of avenues which we explore in this thesis.

Firstly, Park *et al.* [54] proposed NeRFies, a model that creates a volumetric representation of a person from a self-captured sequence with a phone camera. Since the inception of NeRFs [48], it was among the first works the achieved such a high quality of reconstructions from a casual videos. In its primal form, NeRFies were unable to control the avatar in any other way than by a linear interpolation of latent embeddings that embedded the video's time dimension. The follow-up work, HyperNeRF [55] handles this issue by projecting the learnable embeddings with D onto a lower-dimensional space $\mathbb{R}^d$ where $d \ll D$. After the assumption that the $d=2$ is enough to explain the sequence variability, that projected embedding becomes a 2-dimensional space that can be traversed in an interpretable way. However, that space is not intuitive since the projection is a non-linear operation and one cannot predict how values affect the results. To mitigate that issue, we propose to leverage smoothness of Multilayer Perceptrons (MLPs) [54, 72] to constrain the projection via sparse supervision. We realize our approach as a weakly-supervised MLP that out of many images from the sequence (we assume at least 100 frames in our work) only a few are provided with a coarse annotation. Such annotations denote what values a chosen attribute takes and where its effect spans in the image space. We show that our method, which we dubbed CoNeRF [33] and published at the CVPR 2022 conference, imbues NeRFs with a flexible editability feature without the lose of the rendering quality.

Secondly, approaches such as CoNeRF [33], EditNeRF [41] or FigNeRF [90] focus solely on static elements of the scene, hence their controllability is limited to changing colors or textures in general. HyperNeRF [55] arises as a potential solution due to its ability to model object deformations. However, our initial experiments showed that those changes cannot handle motions that affect a subject globally, *e.g.*, jumping jacks performed by a person. To solve the issue, Fang *et al.* [15] proposes to model the deformation via a multi-scale voxel structure which works well in the synthetic setting, such as the one proposed by Pumarola *et al.* [57].

There exists a plethora of works that approach the problem from the another angle—instead of modeling the motion purely from data, they use a template model in the form of a 3D mesh to canonicalize deformed points [106]. Such methods rely on the accuracy of the *registration*, *i.e.*, fitting the template mesh to subject. Since the registration methods [16, 105] are imperfect estimators, they inherently contain registration errors. Those deviations are exacerbated by learnable radiance field models which assume a perfectly calibrated scene. The authors of those approaches usually mitigate the issue with additional latent space [21, 33, 45] that requires

thousands of video frames to learn an avatar of high-fidelity that reacts correctly to deformations such as wrinkles on the forehead. At the same time, performing the registration on the large scale is costly [7]. In this thesis, we seek a remedy for those obstacles. We propose a method that is data-efficient, easy to improve with a minimal user input and can model realistic deformation dependent changes in the subject. Inspired by classical methods in character texturing [53] and motion modeling [38], we propose BlendFields [30], an *homage* to traditional blendshapes [38]. We build on VolTeMorph's [19] approach to point canonicalization to provide a data-efficient way to control the character. We further introduce a physically-based mixture of predefined, learned from data wrinkle templates that represent expression-dependent skin deformations. Our proposed was acknowledged by the reviewers and was accepted to the CVPR 2023 conference.

Thirdly, having the texture and coarse mesh-based controllability, we strive for control scene settings directly. The inverse rendering of 3D scenes is an ill-posed problem where many different lighting settings may explain the same light effects [56]. To facilitate solving the problem, many approaches use datasets of single object's images captured under different lighting conditions [8, 63, 95]. These approaches cannot decouple albedo from the lighting effects [8, 95] or need additional neural networks to predict correct shadows [63] which limits methods' practicality. We propose to use recently proposed 2D Gaussian Splatting [24] which exhibit remarkable quality of the surface reconstruction. Together with our precomputed radiance transfer from classical computer graphics approaches [58, 64], our LumiGauss achieves state-of-the-art reconstruction quality with the ability to render novel lighting conditions with high fidelity. Our work received positive reviews for the WACV 2025 conference.

Finally, volumetric representation are computationally intensive to render, compared to the traditional mesh representation. For NeRFs [48], it takes seven days on V100 NVidia GPU to train for single scene, and more than 60 seconds to render a single image—way beyond any practical applications. Although many approaches have been proposed to speed up the rendering process [20, 23, 51, 60, 96], they usually make a trade-off between memory requirements, quality, and rendering speed. 3D Gaussian Splatting [35] (3DGS) rose as an alternative to NeRF, offering both high-quality rendering at interactive frame rates. However, those frame rates could be achieved with the most advanced GPU units available at that time. As we see the potential in 3DGS to be a viable canonical representation for 3D data, akin to 3D meshes, a need for its adaptability to different computational resources exists. Meshes can be adapted easily with levels of detail (LoD) approaches that remove detail from meshes that do not affect the general object's perception if necessary.

1.2 Research objectives

In this thesis, we explore different avenues of radiance field controllability. With this goal in mind, we aim at answering the following research questions:

(**RQ 1**) Can we imbue a Neural Radiance Field (NeRF) with a controllability by providing sparse annotations to the training dataset? How many annotations suffice to learn smooth interpolation capabilities between controlled values?

(**RQ 2**) Are extreme facial expressions known from the literature sufficient to learn expression-dependent details that extrapolate to expressions unseen at the training time?

(**RQ 3**) Is it possible to learn an underlying radiance transfer function of a scene from images taken in "in-the-wild" setting? Can such a transfer function generalize to unknown environment maps?

(**RQ 4**) How to learn a single 3D Gaussian Splatting (3DGS) representation that can be adapted to different computational regimes at inference time in a feed-forward mode?

Each of these questions is a representative of possible among many others controllability directions for radiance fields. In case of this thesis, we present summarize our methods as controls of: texture [30, 33], shape [30], lightning [29] and use of resources [31]. To answer (**RQ 1**), we describe our CoNeRF [33]. It is one of the pioneering works that uses sparsely annotated frames to continuously control the subject in a post-hoc manner. We leverage the fact that MLP used in NeRFs are smooth functions biased towards low frequency signals [72]. For that reason, NeRF can learn to interpolate smoothly the annotation signal between frames of high similarity. We show that this assumption is sufficient to obtain both novel view synthesis and novel attribute synthesis with a single model.

We further move towards answering (**RQ 2**). We introduce BlendFields [30], achieving two primary goals: ability to generalize unknown expressions via a predefined face mesh template, and a mixture model of training expressions that can produce spatially coherent, expression-dependent wrinkles on the face from as few as three expressions. In our work, we build on VolTeMorph [19] to achieve to former, and focus on the latter contribution. Inspired by texture maps in classical computer graphics pipelines [53], we define a set of learnable radiance fields, each being overfit to a particular extreme expression from the training set. We define an extreme expression as one of the possible expressions involving the most facial muscles. Building on VolTeMorph [19] allows us to use an underlying tetrahedral mesh to compute physical quantities such as the volume change of tetrahedra under a given expression. We use those quantities to linearly interpolate between the pretrained radiance fields. We mix the tetrahedra independently which makes rendering novel expressions possible. For example, BlendFields can render one of the eyebrows raised while maintaining the other in a natural position, which is a difficult expression to make for majority of people.

Our LumiGauss [29] answers (**RQ 3**). In contrary to common approaches [63], we posit that the radiance transfer function known among computer graphics researchers [58] can be learned from unconstrained photo collection under varying lighting conditions. To this end, we use 2DGS [24] which gives us a smooth shape representation, difficult to achieve when

using 3DGS [35]. With the use of contributed priors, we induce learning Gaussian's Spherical Harmonics such that they correctly react to changing environment maps. Not only our approach is fast to train, but renders realistic scenes and object's shadows even under novel lighting conditions.

All the contributions so far are affected by a specific disadvantage—a single model needs to be trained from scratch and it can be deployed only on high-end hardware. We then ask if we can train a single model that is adaptable at inference time to different hardware regimes (**RQ 4**). We propose CLoG [31] as a potential remedy. Our approach uses the fact that one can constrain the number of Gaussians in 3DGS [35] to a specific number, such that it can be formed as a 2D grid by simple reshape operation. With a specific training coarse-to-fine training protocol we contributed, the model learns a representation that converges to a high-quality volumetric structure. In the second training stage, we leverage the fact that Gaussians's can be spatially sorted in such a manner that their descriptors are placed next to each other if they are similar. That forms a low-frequency image which can be modulated with an off-the-shelf continuous upsampling architecture [75]. The architecture outputs a new grid of the given resolution. We show in our work that such an approach achieves remarkable results and can output any number of Gaussians at inference time with high quality.

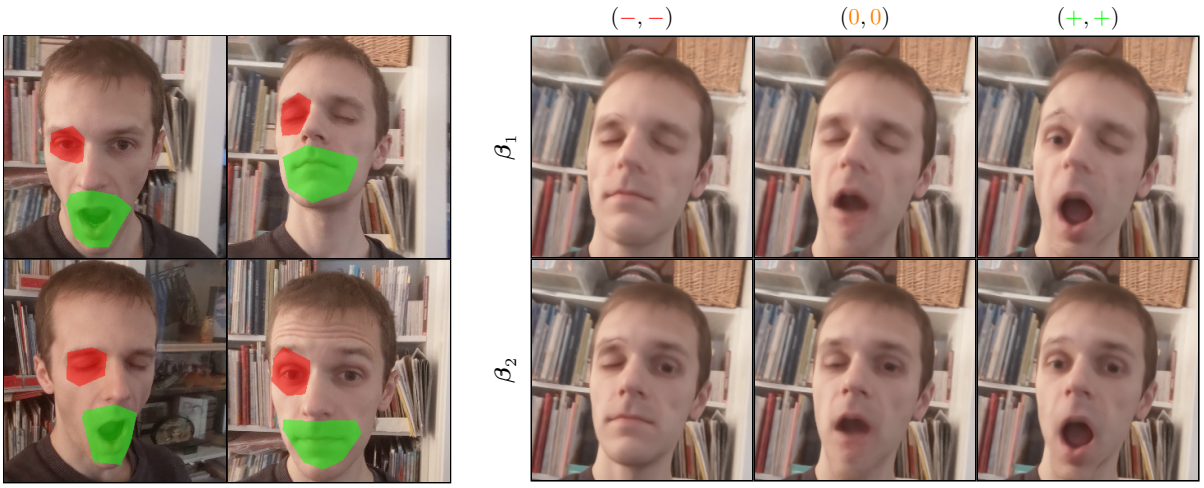
1.3 Contributions

Building on those questions, we structure this thesis in several chapters corresponding to the answers. An answer is in a form of a scientific article where we introduce the following:

- A novel approach for controlling trained radiance fields with the use of sparsely annotated images from a casually captured data.
- A new model capable of blending trained radiance fields from multi-view frames in an interpretable way and extrapolating to novel human expressions for the trained subject.
- The first use of Gaussian Splatting methods that learns a coherent shape representation of a subject and an ability to distill the varying lighting conditions in the data to a radiance transfer function.
- A novel paradigm for learning Gaussian Splatting models as 2D grids to achieve flexibility to adapt the model to different computational regimes at inference time.

1.3.1 Texture from Partial Information

Existing NeRF-based approaches in 2021 were simple models—they were overfit to a single subject, for a new subject the model needed to be retrained from scratch, and the editability



(a) annotated samples **(b)** renderings from the trained model

Figure 1. Example with annotations and controlled attributes – We show an example of how our CoNeRF is capable of controlling attributes selected sparse annotations at the training time. Top row shows possible value combinations ($-$ denoting closed eye, 0 neutral position and $+$ an open eye) and $\{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2\}$ possible values of attributes that are not explicitly learned from the annotations but purely from data (please see Chapter 3 for further explanation).

capabilities were limited [41]. NeRFactor [101] could be considered a more sophisticated model. However, it worked only for simple scene with calibrated scenes and could not handle any motion.

We approach those issues in our CoNeRF [33] published at the CVPR 2021 conference. Inspired by HyperNeRF [55], we propose to revisit the weak supervision in the context of radiance fields. Specifically, under an assumption that one can provide a few of sparse annotations to the dataset, we can leverage the smoothness of the neural networks to propagate the annotation across the dataset. The annotations also consist of what regions in the image space it refers to and hence we can train a semantic segmentation radiance fields that decouples attribute controls. We show an example in Fig. 1 where the left side represents the annotations present in the data and the right one possible manipulations at the inference time. We note that those annotations are easy to make in a matter of a few minutes for a single dataset. However, the complexity of the annotations grows with the number of attributes to control.

1.3.2 Expression from Few-Shot Learning

CoNeRF [33] is capable of rendering complex motions given sufficient amount of data and provided annotations. However, motions as the ones a person performs daily when speaking are infeasible in practice. We propose a solution to target that issue. We introduce BlendFields [30] from the CVPR 2023 proceedings which learns motion-dependent face deformation from data. We build on VolTeMorph [19] which uses existing face template models, such as FLAME [39], to learn a single canonical representation, akin to the canonicalization module in CoNeRF. Internally, BlendFields creates a tetrahedral cage around the face model. For each sample

along the ray in NeRF, it moves the points to a "neutral position", chosen once prior to the training procedure. Such a procedure comes insufficient to model realistic facial features, such as wrinkles. We contribute a novel approach to modeling those deformations. We compute a deformation gradient of each tetrahedra for a given expression which is a physically-based and easy to interpret quantity. The value serves us to smoothly transition face regions textures to appropriate colors. We obtain the colors from NeRFs branches, each overfit to particular expression. In principle, BlendFields predicts face bases which are conditioned on the face expression vector to output the final point color. Our framework works well even when only a few "extreme" expressions are provided such as grinning face with closed eyes and wide open mouth with open eyes.

1.3.3 Light from Unconstrained Images

In both approaches above, we tackle the problem of texture controllability. They assume an ideal case scenario where a capture can be taken in an idealized environment with constant camera exposure and lighting. Moreover, the produced colors are blended together, making the change of light impossible in practice. We then ask the questions if we can decouple an intrinsic color of the subject and change of that color stemming from the environment light. We answer that question with our LumiGauss [29] published at the WACV 2025 conference that, indeed, we can achieve that by learning the radiance transfer function directly from data. For that end, we train a 2DGS [24] model to obtain a smooth and spatially coherent surface of objects. On top of the other attributes known from 3DGS [35], we imbue our Gaussians with additional features corresponding to the radiance transfer, expressed as Spherical Harmonics [22]. We show in our experiments that such a formulation is sufficient to train a model that reacts to changing environment maps in a realistic manner and renders images of higher fidelity than prior approaches.

1.3.4 Levels of Detail in One Model

All the contributions above require considerable computation hardware to be trained on and then run inference in near real-time frame rates. Drawing an inspiration from the gaming industry where Levels of Details (LoD) for meshes is used heavily, we introduce CLoG [31]. Our approach trains a single Gaussian Splatting model such that it can be modulated at inference time and adapted to the target computational requirements with a minimal loss on the rendering quality. That allows us to democratize the use of 3DGS and deploy a single model even on handheld devices. We show later that even under a strict case where only 2,000 Gaussians[1] can be used, the object is still recognizable. The most important contribution of our model is that it is continuous by design while the existing baselines assume prior the training the model how many LoD the model should comprise at inference. To change the size of particular LoD in those

[1] Common 3DGS models can achieve from 10^5 to even 10^6 Gaussians.

baselines requires training the whole model from scratch, imposing a significant computational burden.

1.4 Thesis outline

This thesis is structured as follows. We start by introducing the preliminaries related to the Neural Radiance Fields and Gaussian Splatting in Chapter 2—a common theme in all works that appear later. We then move towards describing CoNeRF [33] in Chapter 3, our approach to control trained neural radiance fields by using sparse annotations. In Chapter 4, we describe BlendFields [30] that can produce realistic expression-dependent texture from just a few multi-view frames of the subjects. **??** introduces the LumiGauss [29], a Gaussian Splatting model that learns a radiance transfer function for novel lighting rendering capabilities. Finally, we bring a general method, CLoG [31], that uses Gaussian Splatting to learn continuous levels of detail while training only a single model. We conclude the thesis in Chapter 5 where we also explore possible future avenues that can be undertaken.

1.5 Publications not included in the thesis

We attach a list of articles that are related to and can be used to in neural rendering approaches:

- **Kania**, **K.**, Zięba, M., and Kajdanowicz, T., "UCSG-NET – Unsupervised Discovering of Constructive Solid Geometry Tree," *NeurIPS*, vol. 33, pp. 8776–8786, 2020,
- **Kania**, **K.**, Kowalski, M., and Trzciński, T., "TrajeVAE: Controllable Human Motion Generation from Trajectories," *arXiv preprint arXiv:2104.00351*, 2021,
- Stypułkowski, M., **Kania**, **K.**, Zamorski, M., Zięba, M., Trzciński, T., and Chorowski, J., "Representing Point Clouds with Generative Conditional Invertible Flow Networks," *Pattern Recognition Letters*, vol. 150, pp. 26–32, 2021,
- Xia, S., Yue, J., **Kania**, **K.**, Fang, L., Tagliasacchi, A., Yi, K. M., and Sun, W., "Densify Your Labels: Unsupervised Clustering with Bipartite Matching for Weakly Supervised Point Cloud Segmentation," *arXiv preprint arXiv:2312.06799*, 2023,
- Esposito, S., Xu, Q., **Kania**, **K.**, Hewitt, C., Mariotti, O., Petikam, L., Valentin, J., Onken, A., and Mac Aodha, O., "GeoGen: Geometry-Aware Generative Modeling via Signed Distance Functions," in *CVPRW*, 2024, pp. 7479–7488,
- Spurek, P., Winczowski, S., Zięba, M., Trzciński, T., **Kania**, **K.**, and Mazur, M., "Modeling 3D Surfaces with a Locally Conditioned Atlas," in *ICCS*, Springer, 2024, pp. 100–115.

Chapter 2

Background

2.1 Neural Rendering

2.2 Neural Radiance Field

2.3 3D Gaussian Splatting

Chapter 3

CoNeRF: Controllable Neural Radiance Fields

Chapter 4

BlendFields: Few-Shot Example-Driven Facial Modeling

4.1 Abstract

Generating faithful visualizations of human faces requires capturing both coarse and fine-level details of the face geometry and appearance. Existing methods are either data-driven, requiring an extensive corpus of data not publicly accessible to the research community, or fail to capture fine details because they rely on geometric face models that cannot represent fine-grained details in texture with a mesh discretization and linear deformation designed to model only a coarse face geometry. We introduce a method that bridges this gap by drawing inspiration from traditional computer graphics techniques. Unseen expressions are modeled by blending appearance from a sparse set of extreme poses. This blending is performed by measuring local volumetric changes in those expressions and locally reproducing their appearance whenever a similar expression is performed at test time. We show that our method generalizes to unseen expressions, adding fine-grained effects on top of smooth volumetric deformations of a face, and demonstrate how it generalizes beyond faces.

4.2 Introduction

Recent advances in neural rendering of 3D scenes [73] offer 3D reconstructions of unprecedented quality [48] with an ever-increasing degree of control [33, 41]. Human faces are of particular interest to the research community [1, 17–19] due to their application in creating realistic digital doubles [44, 73, 98, 103].

To render facial expressions not observed during training, current solutions [1, 17–19] rely on *parametric* face models [6]. These allow radiance fields [48] to be controlled by facial parameters estimated by off-the-shelf face trackers [39]. However, parametric models primarily capture

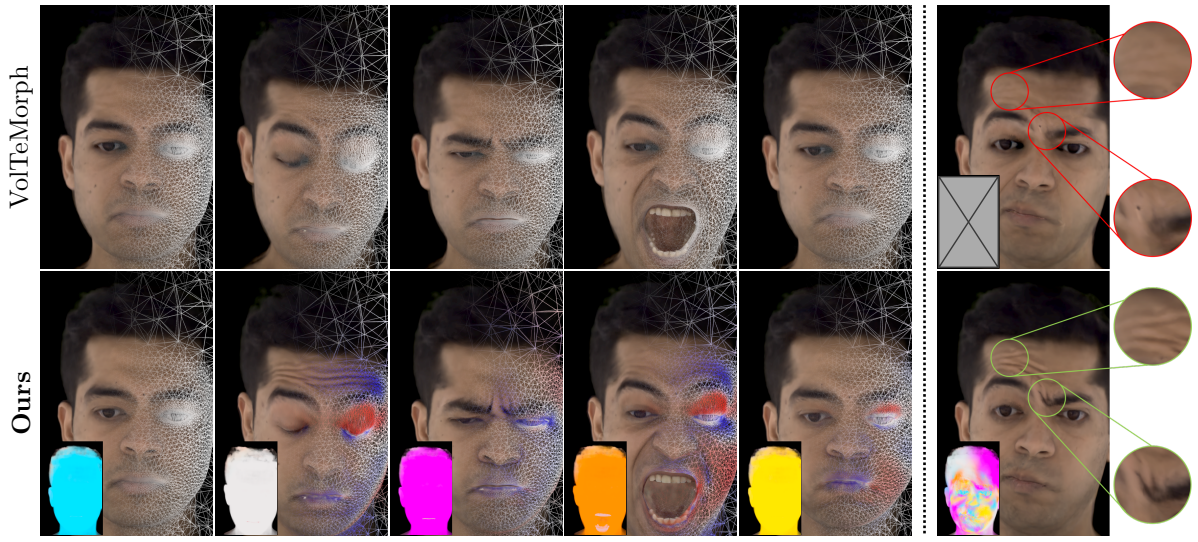


Figure 2. Teaser – Given five multi-view frames of different expressions, our approach generates a model capable of capturing the fine-grained details of a novel expression beyond the resolution of the underlying face model [19] (top right corner). This is achieved by *blending* the radiance fields computed for individual expressions, where the blending coefficients are modulated accordingly to *local* volumetric changes. These volumetric changes are measured as the difference in the tetrahedral volume of a mesh that deforms with the expression (■ increase, ■ decrease, and ■ no change in volume). Such an approach allows *BlendFields* to render sharp, expression-dependent details of the face without increasing the resolution of the mesh (bottom right corner).

smooth deformations and lead to digital doubles that lack realism because fine-grained and expression-dependent phenomena like wrinkles are not faithfully reproduced.

Authentic Volumetric Avatars (AVA) [7] overcomes this issue by learning from a large multi-view dataset of synchronized and calibrated images captured under controlled lighting. Their dataset covers a series of dynamic facial expressions and multiple subjects. However, this dataset remains unavailable to the public and is expensive to reproduce. Additionally, training models from such a large amount of data requires significant compute resources. To democratize digital face avatars, more efficient solutions in terms of hardware, data, and compute are necessary.

We address the efficiency concerns by building on the recent works in Neural Radiance Fields [19, 92, 97]. In particular, we extend VolTeMorph [19] to render facial details learned from images of a sparse set of expressions. To achieve this, we draw inspiration from blend-shape correctives [38], which are often used in computer graphics as a data-driven way to correct potential mismatches between a simplified model and the complex phenomena it aims to represent. In our setting, this mismatch is caused by the low-frequency deformations that the tetrahedral mesh from VolTeMorph [19], designed for real-time performance, can capture, and the high-frequency nature of expression wrinkles.

We train multiple radiance fields, one for each of the K sparse expressions present in the input data, and blend them to correct the low-frequency estimate provided by VolTeMorph [19]; see Fig. 2. We call our method BlendFields since it resembles the way blend shapes are employed in 3DMMs [6]. To keep K small (*i.e.*, to maintain a few-shot regime), we perform local blending to exploit the known correlation between wrinkles and changes in local differential properties [27, 59]. Using the dynamic geometry of [19], local changes in differential properties can be easily extracted by analyzing the tetrahedral representation underlying the corrective blendfields of our model.

Contributions We outline the main qualitative differences between our approach and related works in Tab. 1, and our empirical evaluations confirm these advantages. In summary, we:

- extend VolTeMorph [19] to enable modeling of high-frequency information, such as expression wrinkles in a few-shot setting;
- introduce correctives [6] to neural field representations and activate them according to local deformations [59];
- make this topic more accessible with an alternative to techniques that are data and compute-intensive [7];
- show that our model generalizes beyond facial modeling, for example, in the modeling of wrinkles on a deformable object made of rubber.

	NeRF [48]	NeRFies [54]	HyperNeRF [55]	NeRFace [17]	NHA [21]	AVA [7]	VolTeMorph [19]	**Ours**
Applicability beyond faces	✓	✓	✓	✗	✗	✗	✓	✓
Interpretable control	✗	✗	✗	✓	✓	✗	✓	✓
Data efficiency	✗	✓	✓	✗	✓	✗	✓	✓
Expression-dependent changes	✗	✗	✓	✓	✓	✓	✗	✓
Generalizability to unknown expressions	✗	✗	✗	✓	✓	✗	✓	✓

Table 1. Comparison – We compare several methods to our approach. Other methods fall short in data efficiency and applicability. For example, AVA [7] requires 3.1 million training images while VolTeMorph [19] cannot model expression-dependent wrinkles realistically.

4.3 Related Works

Neural Radiance Fields (NeRF) [48] is a method for generating 3D content from images taken with commodity cameras. It has prompted many follow-up works [4, 5, 25, 45, 47, 54, 61, 67, 71, 76, 89, 99] and a major change in the field for its photorealism. The main limitations of NeRF are its rendering speed, being constrained to static scenes, and lack of ways to control the scene. Rendering speed has been successfully addressed by multiple follow-up works [20, 23, 96]. Works solving the limitation to static scenes[1, 2, 28, 52, 80, 82, 88, 102] and adding explicit control [9, 33, 36, 69, 78, 94, 97] have limited resolution or require large amounts of training data because they rely on controllable coarse models of the scene (*e.g.*, 3DMM face model [6]) or a conditioning signal [55]. Methods built on an explicit model are more accessible because they require less training data but are limited by the model's resolution. Our technique finds a sweet spot between these two regimes by using a limited amount of data to learn details missing in the controlled model and combining them together. Our experiments focus on faces because high-quality data and 3DMM face models are publicly available, which are the key component for creating digital humans.

4.3.1 Radiance Fields

Volumetric representations [77] have grown in popularity because they can represent complex geometries like hair more accurately than mesh-based ones. Neural Radiance Fields (NeRFs) [48] model a radiance volume with a coordinate-based MLP learned from posed images. The MLP predicts density $\sigma(\mathbf{x})$ and color $\mathbf{c}(\mathbf{x}, \mathbf{v})$ for each point $\mathbf{x}$ in the volume and view direction $\mathbf{v}$ of a given camera. To supervise the radiance volume with the input images, each image pixel is associated with a ray $\mathbf{r}(t)$ cast from the camera center to the pixel, and samples along the ray are accumulated to determine the value of the image pixel $C(\mathbf{r})$:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\, \sigma(\mathbf{r}(t))\, \mathbf{c}(\mathbf{r}(t), \mathbf{v}) dt, \tag{1}$$

where t_n and t_f are near and far planes, and

$$T(t) = \exp\left(- \int_{t_n}^{t} \sigma(\mathbf{r}(s)) ds \right), \tag{2}$$

is the transmittance function [70]. The weights of the MLP are optimized to minimize the mean squared reconstruction error between the target pixel and the output pixel. Several methods have shown that replacing the implicit functions approximated with an MLP for a function discretized on an explicit voxel grid results in a significant rendering and training speed-up [20, 23, 40, 68, 96].

4.3.2 Animating Radiance Fields

Several works exist to animate the scene represented as a NeRF. D-NeRF uses an implicit deformation model that maps sample positions back to a canonical space [57], but it cannot generalize to unseen deformations. Several works [17, 54, 55, 74] additionally account for changes in the observed scenes with a per-image latent code to model changes in color as well as shape, but it is unclear how to generalize the latents when animating a sequence without input images. Similarly, works focusing on faces [1, 17, 18, 104] use parameters of a face model to condition NeRF's MLP, or learn a latent space of images and geometry [7, 42–44, 46, 79] that does not extrapolate beyond expressions seen during training.

In contrast to these approaches, we focus on using as little temporal training data as possible (*i.e.* five frames) while ensuring generalization. For this reason, we build our method on top of VolTeMorph [19], that uses a parametric model of the face to track the deformation of points in a volume around the face and builds a radiance field controlled by the parameters of a 3DMM. After training, the user can render an image for any expression of the face model. However, the approach cannot generate expression-dependent high-frequency details; see Fig. 2.

Similarly, NeRF-Editing [97] and NeRF Cages [92] propose to use tetrahedral meshes to deform a single-frame NeRF reconstruction. The resolution of the rendered scenes in these methods is limited by the resolution of the tetrahedral cage, which is constrained to a few thousand elements.

We discuss additional concurrent works in `Supplementary`.

4.3.3 Tetrahedral Cages

To apply parametric mesh models, it is necessary to extend them to the volume to support the volumetric representation of NeRF. Tetrahedral cages are a common choice for their simplicity and ubiquity in computer graphics [19, 92, 94]. For example, VolTeMorph uses dense landmarks [84] to fit a parametric face model whose blendshapes have been extended to a tetrahedral cage with finite elements method [10]. These cages can be quickly deformed and raytraced [49] using parallel computation on GPUs [12] while driving the volume into the target pose and allowing early ray termination for fast rendering. We further leverage the tetrahedral cage and use its differential properties [27], such as a local volume change, to model high-frequency details. For example, a change from one expression to another changes the volume of tetrahedra in regions where wrinkle formation takes place while it remains unchanged

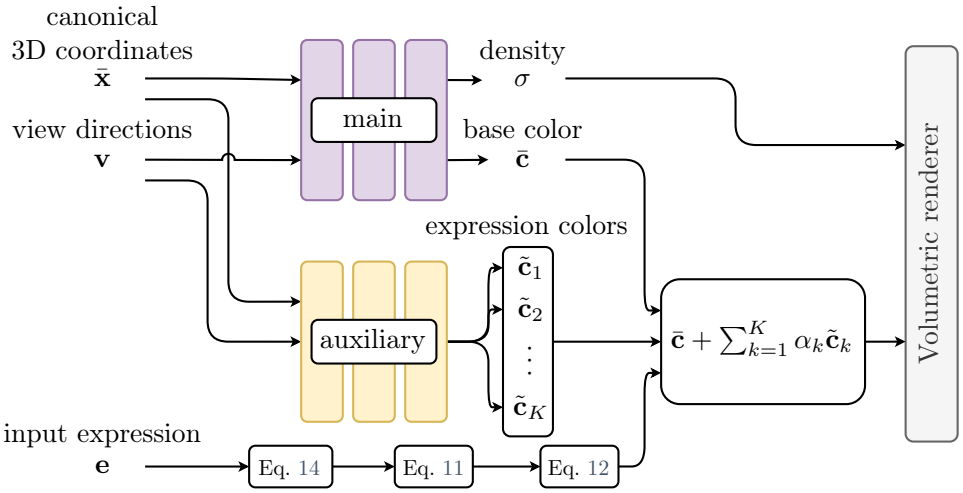


Figure 3. BlendFields – We implement our approach as a volumetric model, where the *appearance* (*i.e.* radiance) is the sum of the main appearance corrected by blending a small set of K expression-specific appearances. These appearances are learnt from extreme expressions, and then blended at test-time according to blend weights computed as a function of the input expression $\mathbf{e}$.

in flat areas. We can use this change in volume to select which of the trained NeRF expressions should be used for each tetrahedron to render high-frequency details.

4.4 Method

We introduce a volumetric model that can be driven by input expressions and visualize it in in Fig. 3. We start this section by explaining our model and how we train and drive it with novel expressions utilizing parametric face models (Sec. 4.4.1). We then discuss how to compute measures of volume expansion and compression in the tetrahedra to combine volumetric models of different expressions (Sec. 4.4.2) and how we remove artifacts in out-of-distribution settings (Sec. 4.4.3). We conclude this section with implementation details (Sec. 4.4.4).

4.4.1 Our model

Given a neutral expression $\bar{\mathbf{e}}$, and a collection of posed images $\{C_c\}$ of this expression from multiple views, VolTeMorph [19] employs a map $\mathcal{T}$ to fetch the density and radiance[1] for a new expression $\mathbf{e}$ from the *canonical* frame defined by expression $\bar{\mathbf{e}}$:

$$\mathbf{c}(\mathbf{x};\mathbf{e}) = \bar{\mathbf{c}}(\bar{\mathbf{x}}), \quad \bar{\mathbf{x}} = \mathcal{T}(\mathbf{x};\mathbf{e}\to\bar{\mathbf{e}}) \tag{3}$$

$$\sigma(\mathbf{x};\mathbf{e}) = \bar{\sigma}(\bar{\mathbf{x}}), \quad \bar{\mathbf{x}} = \mathcal{T}(\mathbf{x};\mathbf{e}\to\bar{\mathbf{e}}) \tag{4}$$

$$\ell_{\text{rgb}} = \mathbb{E}_{C\sim\{C_c\}}\, \mathbb{E}_{\mathbf{r}\sim C}\, \ell^{\mathbf{r}}_{\text{rgb}} \tag{5}$$

$$\ell^{\mathbf{r}}_{\text{rgb}} = \|C(\mathbf{r};\mathbf{e}) - C(\mathbf{r})\|_2^2, \tag{6}$$

[1] We omit view-dependent effects to simplify notation but include them in our implementation.

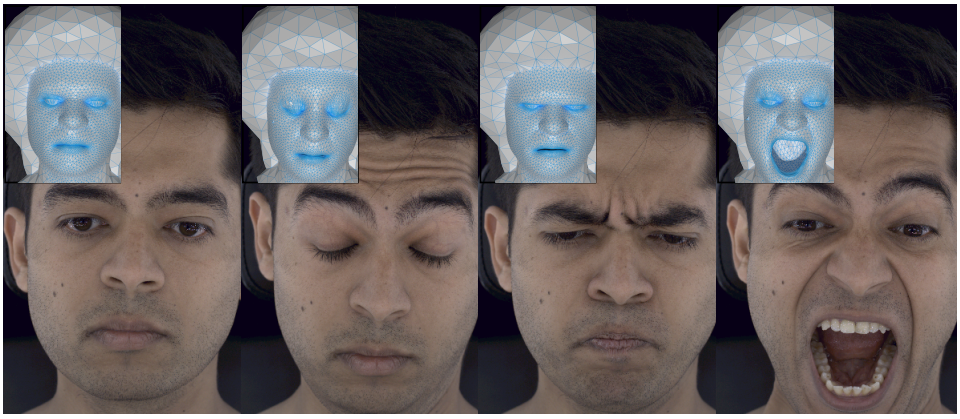


Figure 4. Data – We represent the data as a multi-view, multi-expression images. For each of these images, we obtain parameters of a parametric model, such as FLAME [39] to get: an expression vector $\mathbf{e}$ and a tetrahedral mesh described by vertices $\mathbf{V}(\mathbf{e})$. We highlight that our approach works for any object if a rough mesh and its descriptor are already provided.

where $C(\mathbf{r};\mathbf{e})$ is a pixel color produced by our model conditioned on the input expression $\mathbf{e}$, $C(\mathbf{r})$ is the ground-truth pixel color, and the mapping $\mathcal{T}$ is computed from smooth deformations of a tetrahedral mesh to render unseen expressions $\mathbf{e}$. We use expression vectors $\mathbf{e}$ from parametric face models, such as FLAME [39, 83]. However, as neither density nor radiance change with $\mathbf{e}$, changes in appearance are limited to the low-frequency deformations that $\mathcal{T}$ can express. For example, this model cannot capture high-frequency dynamic features like expression wrinkles. We overcome this limitation by conditioning radiance on expression. For this purpose, we assume radiance to be the sum of a template radiance (*i.e.* rest pose appearance of a subject) and K residual radiances (*i.e.* details belonging to corresponding facial expressions):

$$\mathbf{c}(\mathbf{x};\mathbf{e}) = \bar{\mathbf{c}}(\mathbf{x}) + \sum_{k=k}^{K} \alpha_k(\mathbf{x};\mathbf{e}) \cdot \tilde{\mathbf{c}}_k(\mathbf{x}), \tag{7}$$

We call our model *blend fields*, as it resembles the way in which blending is employed in 3D morphable models [6] or in wrinkle maps [53]. Note that we assume that pose-dependent geometry can be effectively modeled as a convex combination of colors $[\tilde{\mathbf{c}}(\mathbf{x})]_{k=1}^{K}$, since we employ the same density fields as in Eq. (4). In what follows, for convenience, we denote the vector field of blending coefficients as $\boldsymbol{\alpha}(\mathbf{x})=[\alpha_k(\mathbf{x})]_{k=1}^{K}$.

Training the model We train our model by assuming that we have access to a small set of K images $\{\mathbf{C}_k\}$ (example in Fig. 4), each corresponding to an "extreme" expression $\{\mathbf{e}_k\}$, and minimize the loss:

$$\ell_{\text{rgb}} = \mathbb{E}_k \, \mathbb{E}_{\mathbf{r}} \, \|C_K(\mathbf{r};\mathbf{e}_k) - C_k(\mathbf{r})\|_2^2 \tag{8}$$

$$\text{where} \quad \forall \mathbf{x}, \;\; \boldsymbol{\alpha}(\mathbf{x}) = \mathbb{1}_k, \tag{9}$$

where $\mathbb{1}_k$ is the indicator vector, which has value one at the k-th position and zeroes elsewhere, and C_K represents the output of integrating the radiances in Eq. (7) along a ray.

Driving the model To control our model given a novel expression $\mathbf{e}$, we need to map the input expression code to the corresponding blendfield $\boldsymbol{\alpha}(\mathbf{x})$. We parameterize the blend field as a vector field discretized on the vertices $\mathbf{V}(\mathbf{e})$ of our tetrahedral mesh, where the vertices deform according to the given expression. The field is discretized on vertices, but it can be queried within tetrahedra using linear FEM bases [50]. Our core intuition is that when the (local) geometry of the mesh matches the local geometry in one of the input expressions, the corresponding expression blend weight should be locally activated. More formally, let $\mathbf{v}\in\mathbf{V}$ be a vertex in the tetrahedra and $\mathcal{G}(\mathbf{v})$ a local measure of volume on the vertex described in Sec. 4.4.2, then

$$\mathcal{G}(\mathbf{v}(\mathbf{e}))\approx\mathcal{G}(\mathbf{v}(\mathbf{e}_k)) \implies \boldsymbol{\alpha}(\mathbf{v}(\mathbf{e})) \approx \mathbb{1}_k. \tag{10}$$

To achieve this we first define a *local* similarity measure:

$$[\Delta\mathcal{G}_k(\mathbf{v}(\mathbf{e}))] = [\|\mathcal{G}(\mathbf{v}(\mathbf{e}))-\mathcal{G}(\mathbf{v}(\mathbf{e}_k))\|_2^2] \in \mathbb{R}^K \tag{11}$$

and then gate it with softmax (with temperature $\tau{=}10^6$) to obtain vertex blend weights:

$$\boldsymbol{\alpha}(\mathbf{v}(\mathbf{e})) = \text{softmax}_\tau\{\Delta\mathcal{G}_k(\mathbf{v}(\mathbf{e}))\} \in [0,1]^K \tag{12}$$

which realizes Eq. (10), as well as preserves the typically desirable characteristics of blend weights:

- *partition of unity*: $\forall\mathbf{x}\ \boldsymbol{\alpha}(\mathbf{x}) \in [0,1]^K$ and $\|\boldsymbol{\alpha}(\mathbf{x})\|_1{=}1$
- *activations sparsity*: minimizers of $\|\boldsymbol{\alpha}(\mathbf{x})\|_0$

where the former ensures any reconstructed result is a *convex* combination of input data, and the latter prevents destructive interference [26].

4.4.2 Local geometry descriptor

Let us consider a tetrahedron as the matrix formed by its vertices $\mathbf{T}{=}\{\mathbf{v}_i\} \in \mathbb{R}^{3\times4}$, and its edge matrix as $\mathbf{D} = [\mathbf{v}_3 - \mathbf{v}_0, \mathbf{v}_2 - \mathbf{v}_0, \mathbf{v}_1 - \mathbf{v}_0]$. Let us denote $\bar{\mathbf{D}}$ as the edge matrix in rest pose and $\mathbf{D}$ as one of the deformed tetrahedra (*i.e.*, due to expression). From classical FEM literature, we can then compute the change in volume of the tetrahedra from the determinant of its deformation gradient [27]:

$$\Delta\mathcal{V}(\mathbf{T}) = \det(\mathbf{D}\cdot\bar{\mathbf{D}}^{-1}) \tag{13}$$

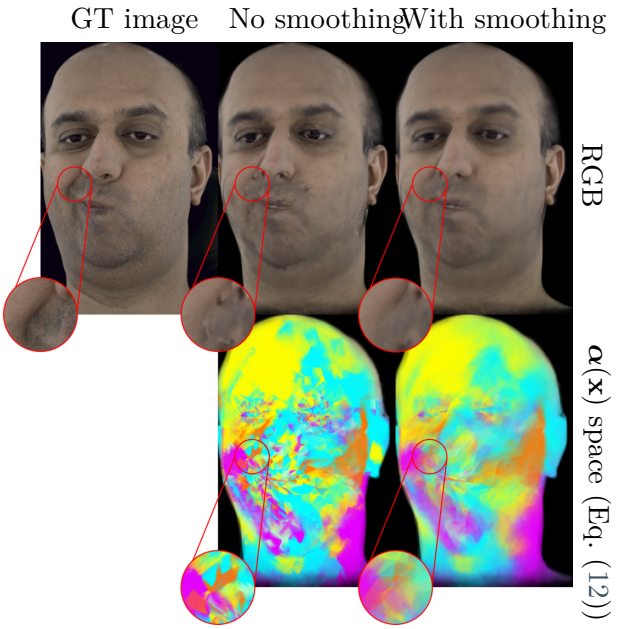


Figure 5. Laplacian smoothing – To combat artifacts stemming from calculating weights α across multiple expressions, which may assign different expressions to neighboring tetrahedra, we apply Laplacian smoothing [13]. As seen in the bottom row, smoothing gives a more consistent expression assignment.

We then build a local volumetric descriptor for a specific (deformed) vertex $\mathbf{v}(\mathbf{e})$ by concatenating the changes in volumes of neighboring (deformed) tetrahedra:

$$\mathcal{G}(\mathbf{v}(\mathbf{e})) = \bigoplus_{\mathbf{T}\in\mathcal{N}(\mathbf{v})} \Delta\mathcal{V}(\mathbf{T}(\mathbf{e})), \tag{14}$$

where $\bigoplus$ denotes concatenation and $\mathcal{N}(\mathbf{v})$ topological neighborhood of a vertex $\mathbf{v}$.

4.4.3 Blend-field smoothness

High-frequency spatial changes in blendfields can cause visual artifacts, see Fig. 5. We overcome this issue by applying a small amount of smoothing to the blendfield. Let us denote with $\mathbf{A}=\{\boldsymbol{\alpha}(\mathbf{v}_v)\}$ the matrix of blend fields defined on all mesh vertices, and with $\mathbf{L}$ the Laplace-Beltrami operator for the tetrahedral mesh induced by linear bases [27]. We exploit the fact that at test-time, the field is discretized on the mesh vertices, execute a diffusion process on the tetrahedral manifold, and, to avoid instability problems, implement it via backward Euler [13]:

$$\mathbf{A}^{\text{diff}} = (\mathbf{I} - \lambda_{\text{diff}}\mathbf{L})^{-1}\mathbf{A}^{n}. \tag{15}$$

4.4.4 Implementation details

We build on VolTeMorph [19] and use its volumetric 3DMM face model. However, the same methodology can be used with other tetrahedral cages built on top of 3DMM face models. The

Method	Real Data						Synthetic Data		
	Casual Expressions			Novel Pose Synthesis			Novel Pose Synthesis		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF [48]	23.6465	0.7384	0.2209	25.6696	0.8127	0.1861	13.7210	0.6868	0.3113
Conditioned NeRF [48]	22.9106	0.7162	0.2029	24.7283	0.7927	0.1682	19.5971	0.8138	0.1545
NeRFies [54]	22.6571	0.7105	0.2271	24.8376	0.7990	0.1884	19.3042	0.8081	0.1591
HyperNeRF-AP [55]	22.6219	0.7087	0.2236	24.7119	0.7931	0.1848	19.3557	0.8132	0.1563
HyperNeRF-DS [55]	22.9299	0.7182	0.2241	24.9909	0.8007	0.1860	19.4637	0.8159	0.1526
VolTeMorph$_1$ [19]	24.9939	0.8358	0.1164	26.7526	0.8749	0.0954	26.7033	0.9500	0.0394
VolTeMorph$_{\text{avg}}$[19]	26.9209	0.8912	0.1105	28.6866	0.9176	0.0982	30.2107	0.9815	0.0387
BlendFields	27.5977	0.9056	0.0854	29.7372	0.9311	0.0782	32.7949	0.9882	0.0221

Table 2. Quantitative results – We compare BlendFields to other related approaches. We split the real data into two settings: one with casual expressions of subjects and the other with novel, static expressions. For the real data, we only compute metrics on the face region, which we separate using an off-the-shelf face segmentation network [83]. Please refer to the `Supplementary` for the results that include the background in the metrics as well. We average results across frames and subjects. VolTeMorph$_{\text{avg}}$ [19] is trained on all frames, while VolTeMorph$_1$ is trained on a single frame. HyperNeRF-AP/-DS follows the design principles from Park *et al.* [55]. The best results are colored in ■ and second best results in ■. BlendFields performs best in most of the datasets and metrics. Please note that HyperNeRF-AP/DS and NeRFies predict a dense deformation field designed for dense data. However, our input data consists of a few static frames only where the deformation field leads to severe overfitting.

face model is created by extending the blendshapes of the parametric 3DMM face model [83] to a tetrahedral cage that defines the support in the neural radiance field. It has four bones controlling global rotation, the neck and the eyes with linear blend skinning, 224 expression blendshapes, and 256 identity blendshapes. Our face radiance fields are thus controlled and posed with the identity, expression, and pose parameters of the 3DMM face model [83], can be estimated by a real-time face tracking system like [85], and generalize convincingly to expressions representable by the face model.

Training. During training, we sample rays from a single frame to avoid out-of-memory issues when evaluating the tetrahedral mesh for multiple frames. Each batch contains 1024 rays. We sample N_{coarse}=128 points along a single ray during the coarse sampling and $N_{\text{importance}}$=64 for the importance sampling. We train the network to minimize the loss in Eq. (8) and sparsity losses with standard weights used in VolTeMorph [19, 23]. We train the methods for 5×10^5 steps using Adam [37] optimizer with learning rate 5×10^{-4} decaying exponentially by factor of 0.1 every 5×10^5 steps.

Inference. During inference, we leverage the underlying mesh to sample points around tetrahedra hit by a single ray. Therefore, we perform a single-stage sampling with N=N_{coarse}+$N_{\text{importance}}$ samples along the ray. When extracting the features (Eq. (14)), we consider $|\mathcal{N}(\mathbf{v})|$=20 neighbors. For the Laplacian smoothing, we set λ_{diff}=0.1 and perform a single iteration step. Geometric-related operations impose negligible computational overhead.

Parameter	Real Data						Synthetic Data		
	Casual Expressions			Novel Pose Synthesis			Novel Pose Synthesis		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
$\lvert\mathcal{N}(\mathbf{v})\rvert = 1$	27.5620	0.9043	0.0893	29.7269	0.9306	0.0815	32.2371	0.9882	0.0234
$\lvert\mathcal{N}(\mathbf{v})\rvert = 5$	27.5880	0.9054	0.0864	29.7548	0.9312	0.0789	32.2900	0.9882	0.0231
$\lvert\mathcal{N}(\mathbf{v})\rvert = 10$	27.5933	0.9054	0.0859	29.7456	0.9312	0.0785	32.3324	0.9882	0.0230
$\lvert\mathcal{N}(\mathbf{v})\rvert = 20$	27.5977	0.9056	0.0854	29.7372	0.9311	0.0782	32.7949	0.9887	0.0221
Without smoothing	27.2535	0.8959	0.0939	29.3726	0.9233	0.0846	32.2452	0.9876	0.0238
With smoothing	27.5977	0.9056	0.0854	29.7372	0.9311	0.0782	32.7949	0.9887	0.0221

Table 3. Ablation study – First, we check the effect of the neighborhood size $|\mathcal{N}(\mathbf{v})|$ on the results. Below that, we compare the effect of smoothing. The best results are colored in ■ and the second best in ■. For the real dataset, changing the neighborhood size gives inconsistent results, while smoothing improves the rendering quality. In the synthetic scenario, setting $|\mathcal{N}(\mathbf{v})|$=20 and the Laplacian smoothing consistently gives the best results. The discrepancy between real and synthetic datasets is caused by inaccurate face tracking for the former. We describe this issue in detail in Sec. 4.5.4.

4.5 Experiments

We evaluate all methods on data of four subjects from the publicly available Multiface dataset [86]. We track the face for eight manually-selected "extreme" expressions. We then select K=5 expressions the combinations of which show as many wrinkles as possible. Each subject was captured with ≈38 cameras which gives ≈190 training images per subject [2]. We use Peak Signal To Noise Ratio (PSNR) [3], Structural Similarity Index (SSIM) [81] and Learned Perceptual Image Patch Similarity (LPIPS) [100] to measure the performance of the models. Each of the rendered images has a resolution of 334×512 pixels.

As baselines, we use the following approaches: the original, static NeRF [48], NeRF conditioned on an expression code concatenated with input points $\mathbf{x}$, NeRFies [54], HyperNeRF[3] [55], and VolTeMorph [19]. We replace the learnable code in NeRFies and HyperNeRF with the expression code $\mathbf{e}$ from the parametric model. Since VolTeMorph can be trained on multiple frames, which should lead to averaging of the output colors, we split it into two regimes: one trained on the most extreme expression[4] (VolTeMorph_1) and the another trained on all available expressions ($\text{VolTeMorph}_{\text{avg}}$)[5]. We use both of these baselines as VolTeMorph was originally designed for a single-frame scenario. By using two versions, we show that it is not trivial to extend it to multiple expressions.

[2] To train BlendFields for a single subject we use ≈ 0.006% of the dataset used by AVA [7].

[3] We use two architectures proposed by Park *et al.* [55].

[4] We manually select one frame that has the most visible wrinkles.

[5] We do not compare to NeRFace [17] and NHA [21] as VolTeMorph [19] performs better quantitatively than these methods.

4.5.1 Realistic Human Captures

Novel expression synthesis. We extract eight multi-view frames from the Multiface dataset [86], each of a different expression. Five of these expressions serve as training data, and the rest are used for evaluation. After training, we can extrapolate from the training expressions by modifying the expression vector $\mathbf{e}$. We use the remaining three expressions: moving mouth left and right, and puffing cheeks, to evaluate the capability of the models to reconstruct other expressions. In Fig. 6 we show that BlendFields is the only method capable of rendering convincing wrinkles dynamically, depending on the input expression. BlendFields performs favorably compared to the baselines (see Tab. 2).

Casual expressions. The Multiface dataset contains sequences where the subject follows a script of expressions to show during the capture. Each of these captures contains between 1000 and 2000 frames. This experiment tests whether a model can interpolate between the training expressions smoothly and generalize beyond the training data. Quantitative results are shown in Tab. 2. Our approach performs best all the settings. See animations in the `Supplementary` for a comparison of rendered frames across all methods.

4.5.2 Modeling Objects Beyond Faces

We show that our method can be applied beyond face modeling. We prepare two datasets containing 96 views per frame of bending and twisting cylinders made of a rubber-like material (24 and 72 temporal frames, respectively). When bent or twisted, the cylinders reveal pose-dependent details. The expression vector $\mathbf{e}$ now encodes time: 0 if the cylinder is in the canonical pose, 1 if it is posed, and any values between $[0, 1]$ for the transitioning stage. We select expressions $\{0, 0.5, 1.0\}$ as a training set (for VolTeMorph$_1$ we use 1.0 only). For evaluation, we take every fourth frame from the full sequence using cameras from the bottom and both sides of the object. We take the mesh directly from Houdini [91], which we use for wrinkle simulation, and render the images in Blender [11]. We show quantitative results in Tab. 2 for the bending cylinder, and a comparison of the inferred images in Fig. 7 for the twisted one[6]. BlendFields accurately captures the transition from the rest configuration to the deformed state of the cylinder, rendering high-frequency details where required. All other approaches struggle with interpolation between states. VolTeMorph$_1$ (trained on a single extreme pose) renders wrinkles even when the cylinder is not twisted.

4.5.3 Ablations

We check how the neighborhood size $|\mathcal{N}(\mathbf{v})|$ and the application of the smoothing influence the performance of our method. We show the results in Tab. 3. BlendFields works best in most

[6] Our motivation is that it is easier to show pose-dependent deformations on twisting as it affects the object globally, while the bending cannot be modeled by all the baselines due to the non-stationary effects.

cases when considering a relatively wide neighborhood for the tetrahedral features[7]. Laplacian smoothing consistently improves the quality across all the datasets (see Fig. 5). We additionally present in the `Supplementary` how the number of expressions used for training affects the results.

4.5.4 Failure Cases

While BlendFields offers significant advantages for rendering realistic and dynamic high-frequency details, it falls short in some scenarios (see Fig. 8). One of the issues arises when the contrast between wrinkles and the subject's skin color is low. In those instances, we observe a much longer time to convergence. Moreover, as we build BlendFields on VolTeMorph, we also inherit some of its problems. Namely, the method heavily relies on the initial fit of the parametric model—any inaccuracy leads to ghosting artifacts or details on the face that jump between frames.

4.6 Conclusions

We present a general approach, BlendFields, for rendering high-frequency expression-dependent details using NeRFs. BlendFields draws inspiration from classical computer graphics by blending expressions from the training data to render expressions unseen during training. We show that BlendFields renders images in a controllable and interpretable manner for novel expressions and can be applied to render human avatars learned from publicly available datasets. We additionally discuss the potential misuse of our work in the `Supplementary`.

4.7 Acknowledgements

The work was partly supported by the National Sciences and Engineering Research Council of Canada (NSERC), the Digital Research Alliance of Canada, and Microsoft Mesh Labs. This research was funded by Microsoft Research through the EMEA PhD Scholarship Programme. We thank NVIDIA Corporation for granting us access to GPUs through NVIDIA's Academic Hardware Grants Program. This research was partially funded by National Science Centre, Poland (grant no 2020/39/B/ST6/01511 and 2022/45/B/ST6/02817).

CoNeRF: Controllable Neural Radiance Fields

Supplementary Material

[7] Larger neighborhood sizes caused out-of-memory errors on our NVIDIA 2080Ti GPU.

4.8 Potential social impact

Our motivation for this work was to enable the creation of 3D avatars that could be used as communication devices in the remote working era. As our approach stems from blendshapes [38], these avatars are easily adjustable via texture coloring and may be used for entertainment. We note, however, that the potential misuse of our work includes using it as deep fakes. We highly discourage such usage. One of our future directions includes detecting fake images generated by our method. At the same time, we highlight the importance of BlendFields—in the presence of closed technologies [7, 44], it is crucial to democratize techniques for personalized avatar creation. We achieve that by limiting the required data volume to train a single model. As history shows, when given an open, readily available technology for generative modeling of images [62], users can scrutinize it with unprecedented thoroughness, thus raising the general awareness of potential misuses.

4.9 Concurrent Works

Gao *et al.* [18] and Xu *et al.* [93] also use an interpolation between known expressions to combine multiple neural radiance fields trained for those expressions. However, their approach interpolates between grids of latent vectors [51] globally. The interpolation weights are taken from blendshape coefficients.

Zielonka *et al.* [106] use a parametric head model to canonicalize 3D points similarly to our ends. However, instead of building a tetrahedral cage around the head, they smoothly assign each face triangle to 3D points. Then they canonicalize points using transformations that each of the assigned triangles undergoes for a given expression. They concatenate 3D points with the expression code from FLAME [39] to model expression-dependent effects.

4.10 Additional results

4.10.1 Ablating number of expressions

We ablate over the number of used expressions during the training. To evaluate the effect of the number of expressions, we add consecutive frames to the training set (starting from a single, neutral one), *i.e.*, the training set has $k<K$ expressions. We train BlendFields for such a set for each subject separately. We then average the results for a given k across subjects. We present the results in Tab. 4. When selecting the training expressions, we aim to choose those that show all wrinkles when combined. We can see from Fig. 10 that if removed, *e.g.*, the expressions with eyebrows raised, then the model cannot render wrinkles on the forehead. In summary, increasing the number of expressions improves the quality results with diminishing returns when $K>5$, while $K=5$ provides a sufficient trade-off between the data capture cost and the quality.

# expr.	Casual Expressions			Novel Pose Synthesis		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
K=1	27.5834	0.9028	0.0834	28.7589	0.9147	0.0806
K=2	27.6783	0.9026	0.0856	29.2859	0.9186	0.0803
K=3	27.9137	0.9054	0.0819	29.8551	0.9279	0.0728
K=4	27.8140	0.9055	0.0815	30.1543	0.9336	0.0701
K=5	28.0254	0.9110	0.0778	30.4721	0.9372	0.0688
K=6	28.0517	0.9091	0.0813	–	–	–
K=7	28.2004	0.9115	0.0823	–	–	–
K=8	28.2542	0.9124	0.0830	–	–	–

Table 4. Number of training expressions – We ablate over the number of training expressions. We evaluate the model on the captures from the Multiface dataset [86]. We run the model for each possible expression combination for a given K and average the results. The best results are colored in ■ and the second best in ■. Increasing the number of available training expressions consistently improves the results. However, using K=5 expressions saturates the quality and using K>5 brings diminishing improvements. We do not report "Novel Pose Synthesis" for K>5 as we use validation expressions and poses to train those models (refer to Sec. 4.5.1 for more details).

Method	Casual Expressions			Novel Pose Synthesis		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF [48]	22.0060	0.6556	0.3222	23.8077	0.7448	0.2779
Conditioned NeRF [48]	21.0846	0.6280	0.3042	22.9991	0.7261	0.2362
NeRFies [54]	20.7004	0.6076	0.3579	23.0123	0.7253	0.2840
HyperNeRF-AP [55]	20.8105	0.6214	0.3504	22.8193	0.7185	0.2689
HyperNeRF-DS [55]	20.8847	0.6111	0.3656	23.0075	0.7259	0.2729
VolTeMorph$_1$ [19]	21.3265	0.7091	0.2706	22.3007	0.7795	0.2281
VolTeMorph$_{avg}$[19]	22.0759	0.7755	0.2615	23.8974	0.8458	0.2302
BlendFields	22.8982	0.7954	0.2256	24.4432	0.8477	0.2052

Table 5. Quantitative results without masking – Similarly to Tab. 2, we compare BlendFields to other related approaches. However, we calculate the results over the whole image space, without removing the background. BlendFields and VolTeMorph [19] model the background as a separate NeRF-based [48] network. The points that do not fall into the tetrahedral mesh are assigned to the background. As the network overfits to sparse training views, it poorly extrapolates to novel expressions (as the new head pose or expression may reveal some unknown parts of the background) and views. At the same time, all other baselines do not have any mechanism to disambiguate the background and the foreground.

4.10.2 Training frames

We present in Fig. 9 example training frames for one of the subjects. Each frame is a multi-view frame captured with ≈35 cameras (the number of available cameras varied slightly between subjects).

4.10.3 Quantitative results with background

We compare BlendFields and the baselines similarly to Sec. 4.5.1. However, in this experiment, we deliberately include the background in metric calculation. We show the results in Tab. 5. In all the cases, BlendFields performs best even though the method was not designed to model the background accurately. Additionally, as HyperNeRF [55], NeRFies [54], and NeRF [48] do not have any mechanism to disambiguate between the foreground and the background, the metrics are significantly worse when including the latter.

4.10.4 Additional qualitative results

We show in Fig. 11 results of baselines that do not rely on parametric models of the face [39]. Compared to BlendFields, they cannot render high-fidelity faces. The issue comes from the assumed data sparsity—those approaches rely on the interpolation in the training data. As we assume access to just a few frames, there is no continuity in the training data that would guide them to interpolate between known expressions. BlendFields presents superior results given novel expressions even with such a sparse dataset. results.

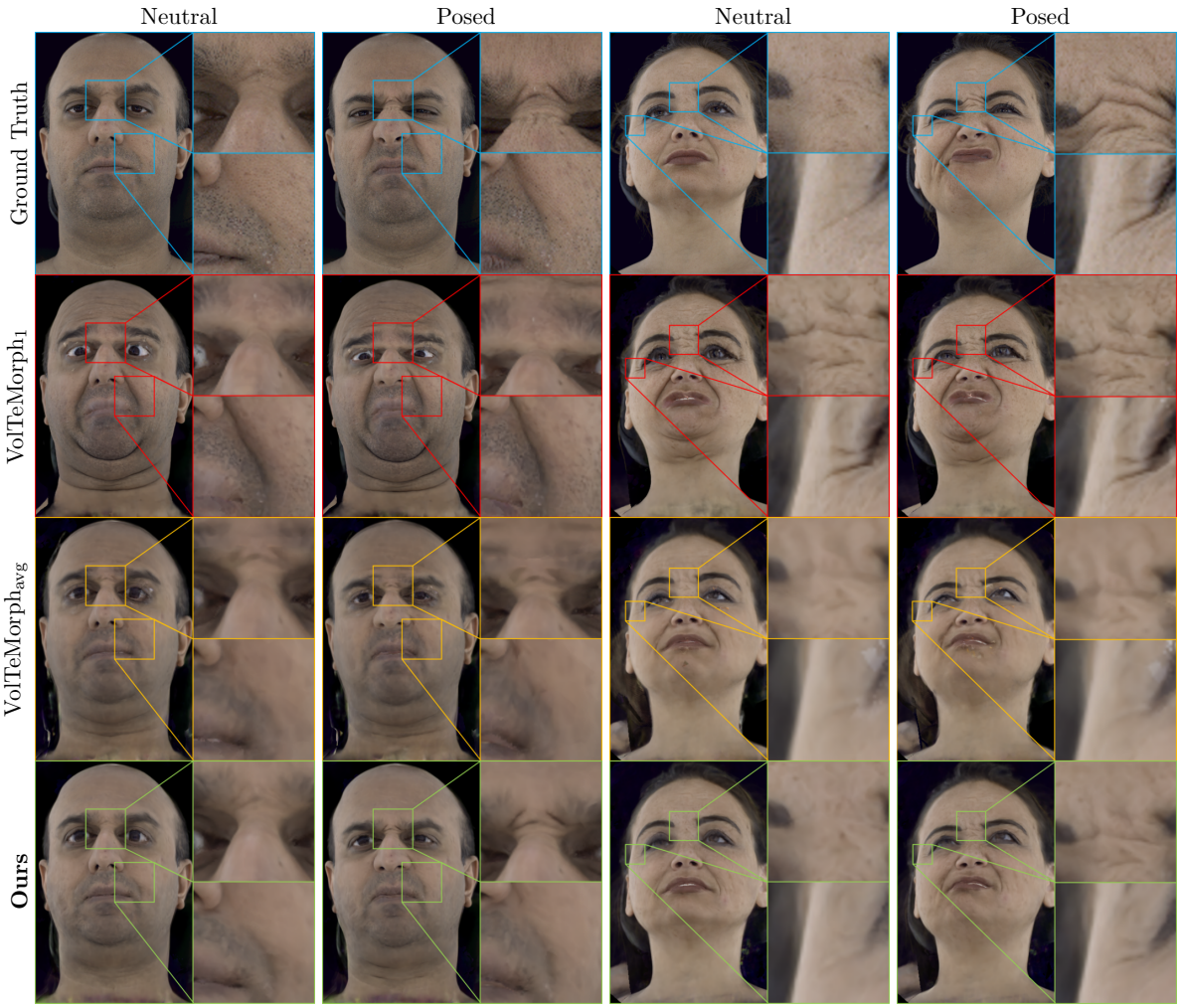


Figure 6. Novel expression synthesis – We compare qualitatively BlendFields with selected baselines (vertical) across two selected subjects (horizontal). Firstly, we show a neutral pose of the subject and then any of the available expressions. To our surprise, VolTeMorph$_{avg}$ trained on multiple frames renders some details but with much lower fidelity. We argue that VolTeMorph$_{arg}$ considers rendering wrinkles as artifacts that depend on the view direction (see Eq. (1)). VolTeMorph$_1$ is limited to producing the wrinkles it was trained for. In contrast to those baselines, **BlendFields** captures the details and generalizes outside of the distribution. Please refer to the `Supplementary` for animated sequences and results for other methods.

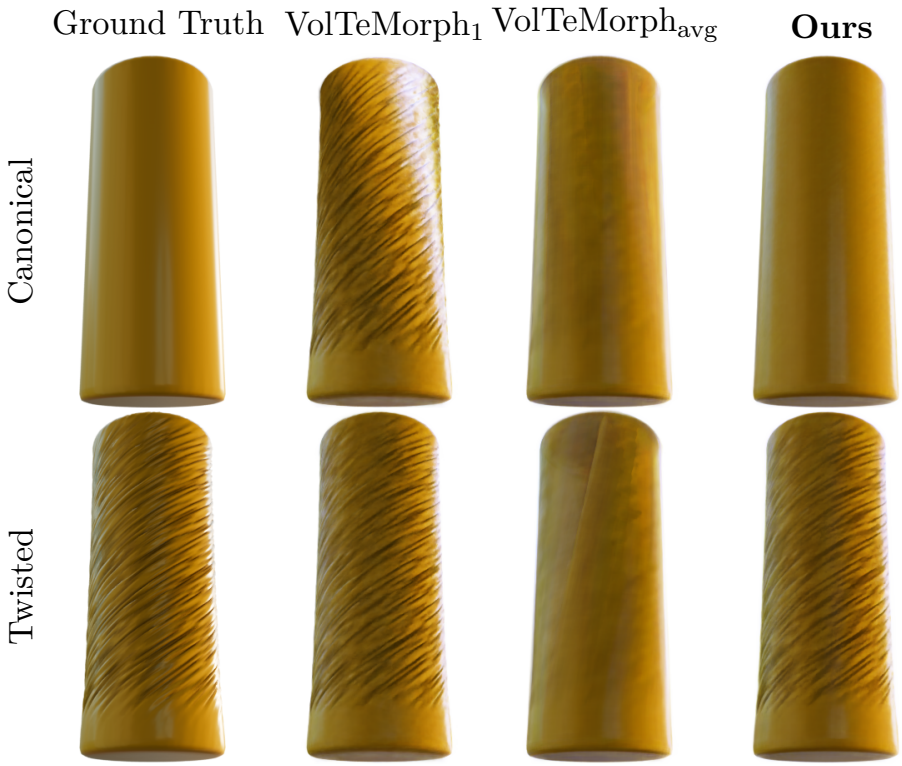


Figure 7. Qualitative results on synthetic dataset – For a simple dataset, baselines cannot model high-frequency, pose-dependent details. VolTeMorph$_1$ renders wrinkles for the straight pose as well, as it is trained for the twisted cylinder only, while VolTeMorph$_{avg}$ averages out the texture.

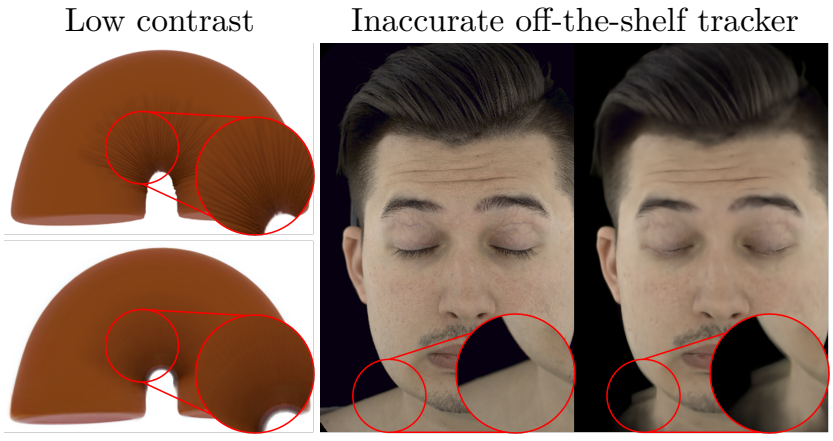


Figure 8. Failure cases – We show failure cases for our proposed approach. *Left:* In the presence of wrinkles in low-contrast images, BlendFields takes longer to converge to make wrinkles visible. We show the ground truth on the top, and rendering after training $7{\times}10^5$ steps on the bottom. In contrast, we rendered images in Fig. 7 after $2{\times}10^5$ steps. *Right:* BlendFields inherits issues from VolTeMorph [19], which relies on the initial fit of the face mesh. If the fit is inaccurate, artifacts appear in the final render.

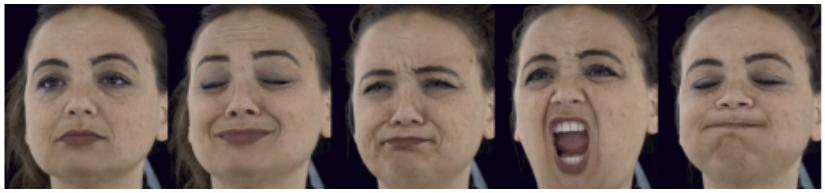


Figure 9. Training frames – In Sec. 4.5, we show results for the BlendFields trained on K=5 expressions. The images represent these expressions for one of the subjects. For each subject, we selected similar expressions to show all possible wrinkles when combined. Please note that we also include a "neutral" expression (the first from the left)—it is necessary to enable the learning of a face without any wrinkles.

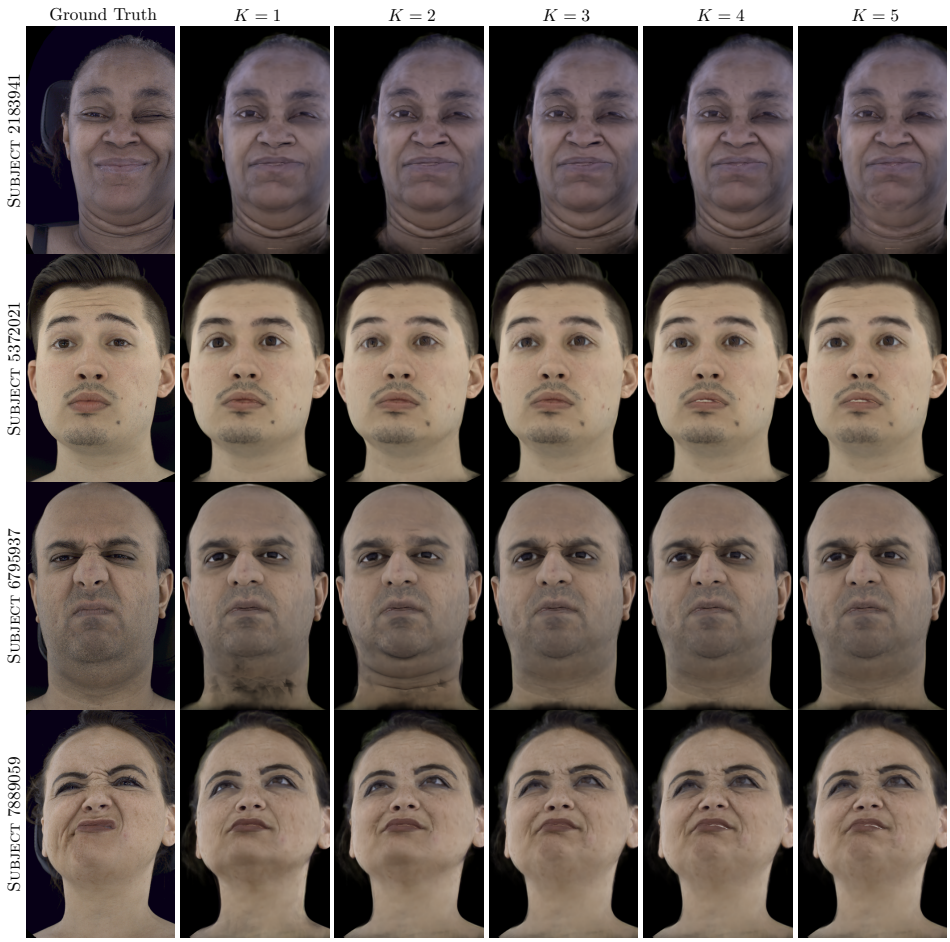


Figure 10. Qualitative ablation over the number of training expressions – We show qualitatively how the number of training expressions K affects the rendering quality. The first row shows the ground truth images. All other consecutive rows show the images rendered with BlendFields while increasing the number of training expressions. The last row, K=5 corresponds to the results presented in the main part of the article. The subject's naming follows the convention introduced in the Multiface repository [86]. Please refer to Tab. 4 for quantitative results.

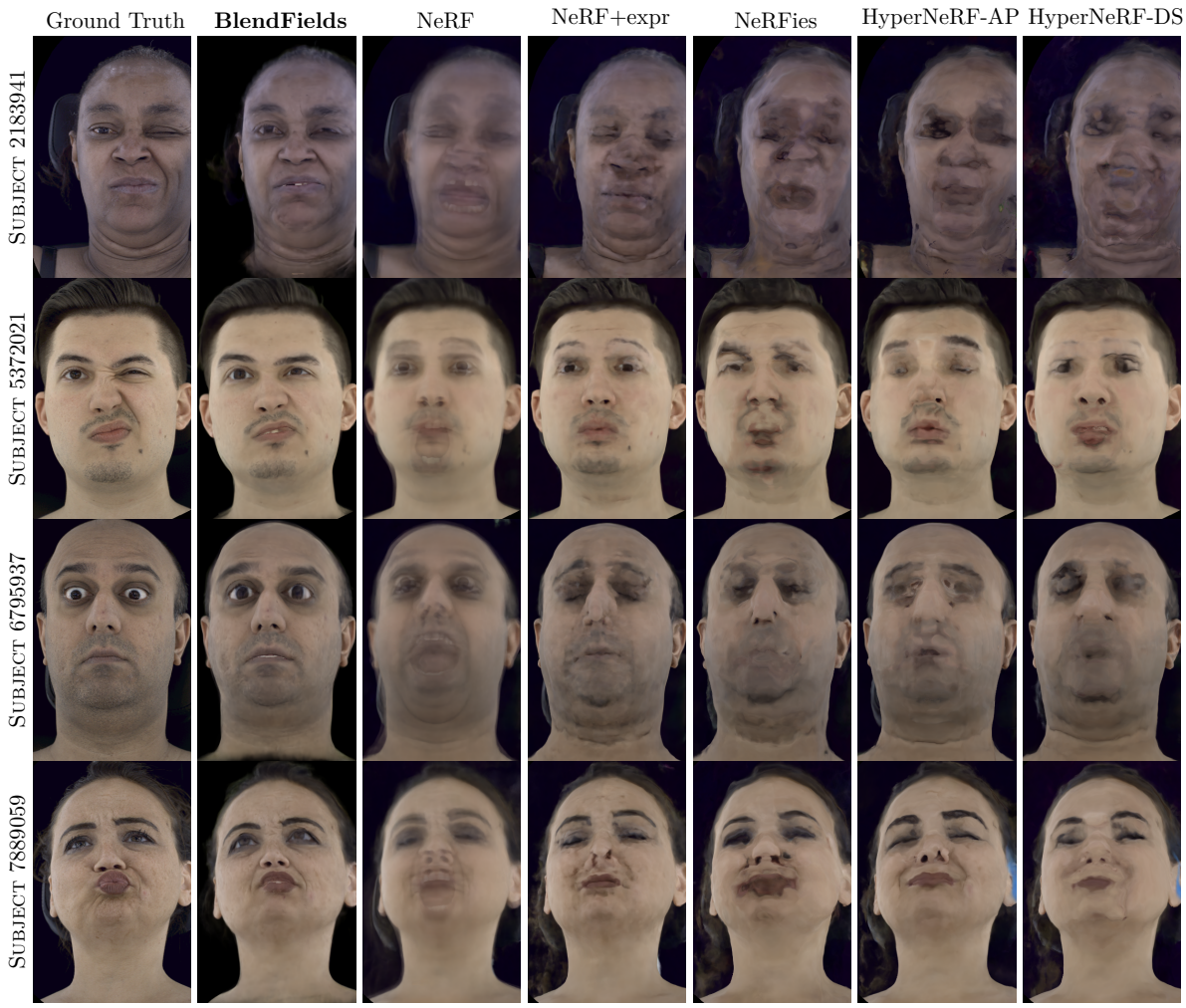


Figure 11. Comparison to strictly data-driven approaches – We compare BlendFields to other baselines that do not rely on mesh-driven rendering: NeRF [48], NeRF conditioned on the expression code (NeRF+expr) [48], NeRFies [54], and HyperNeRF-AP/DS [55]. As a static model, NeRF converges to an average face from available (K=5) expressions. All other baselines exhibit severe artifacts compared to BlendFields. Those baselines rely on the data continuity in the training set (*e.g.*, from a video), and cannot generalize to any other expression.

Chapter 5

Final remarks and discussion

5.1 Conclusions

5.2 Future work

Bibliography

[1] Athar, S., Xu, Z., Sunkavalli, K., Shechtman, E., and Shu, Z., "RigNeRF: Fully Controllable Neural 3D Portraits," in *CVPR*, 2022, pp. 20 364–20 373 (cit. on pp. 17, 20, 21).

[2] Attal, B., Laidlaw, E., Gokaslan, A., Kim, C., Richardt, C., Tompkin, J., and O'Toole, M., "TöRF: Time-of-Flight Radiance Fields for Dynamic Scene View Synthesis," *NeurIPS*, vol. 34, pp. 26 289–26 301, 2021 (cit. on p. 20).

[3] Avcibas, I., Sankur, B., and Sayood, K., "Statistical evaluation of image quality measures," *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 206–223, 2002 (cit. on p. 27).

[4] Barron, J., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., and Srinivasan, P., "Mip-NeRF: A Multiscale Representation for Anti-aliasing Neural Radiance Fields," in *ICCV*, 2021, pp. 5855–5864 (cit. on p. 20).

[5] Barron, J., Mildenhall, B., Verbin, D., Srinivasan, P., and Hedman, P., "Mip-NeRF 360: Unbounded Anti-aliased Neural Radiance Fields," in *CVPR*, 2022, pp. 5470–5479 (cit. on p. 20).

[6] Blanz, V. and Vetter, T., "A Morphable Model For The Synthesis Of 3D Faces," in *CGIT*, 1999, pp. 187–194 (cit. on pp. 17, 19, 20, 23).

[7] Cao, C., Simon, T., Kim, J. K., Schwartz, G., Zollhoefer, M., Saito, S.-S., Lombardi, S., Wei, S.-E., Belko, D., Yu, S.-I., *et al.*, "Authentic Volumetric Avatars from a Phone Scan," *ToG*, vol. 41, no. 4, pp. 1–19, 2022 (cit. on pp. 7, 19–21, 27, 30).

[8] Chen, X., Zhang, Q., Li, X., Chen, Y., Feng, Y., Wang, X., and Wang, J., "Hallucinated Neural Radiance Fields in the Wild," in *CVPR*, 2022, pp. 12 943–12 952 (cit. on p. 7).

[9] Cheng, Z., Chai, M., Ren, J., Lee, H.-Y., Olszewski, K., Huang, Z., Maji, S., and Tulyakov, S., "Cross-Modal 3D Shape Generation and Manipulation," in *ECCV*, Springer, 2022, pp. 303–321 (cit. on p. 20).

[10] Clough, R. W., "The Finite Element Method in Plane Stress Analysis," in *Conference on Electronic Computation*, 1960 (cit. on p. 21).

[11] Community, B. O., *Blender - a 3d modeling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2022. [Online]. Available: `http://www.blender.org` (cit. on p. 28).

[12] Cook, S., *CUDA Programming: A Developer's Guide to Parallel Computing with GPUs*, 1st. 2012, ISBN: 9780124159334 (cit. on p. 21).

[13] Desbrun, M., Meyer, M., Schröder, P., and Barr, A. H., "Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow," in *CGIT*, 1999, pp. 317–324 (cit. on p. 25).

[14] Esposito, S., Xu, Q., **Kania**, **K.**, Hewitt, C., Mariotti, O., Petikam, L., Valentin, J., Onken, A., and Mac Aodha, O., "GeoGen: Geometry-Aware Generative Modeling via Signed Distance Functions," in *CVPRW*, 2024, pp. 7479–7488 (cit. on p. 12).

[15] Fang, J., Yi, T., Wang, X., Xie, L., Zhang, X., Liu, W., Nießner, M., and Tian, Q., "Fast Dynamic Radiance Fields with Time-Aware Neural Voxels," in *SIGGRAPH Asia 2022 Conference Papers*, 2022, pp. 1–9 (cit. on p. 6).

[16] Feng, Y., Feng, H., Black, M. J., and Bolkart, T., "Learning an Animatable Detailed 3D Face Model from In-The-Wild Images," *ToG*, vol. 40, no. 4, pp. 1–13, 2021 (cit. on p. 6).

[17] Gafni, G., Thies, J., Zollhofer, M., and Nießner, M., "Dynamic Neural Radiance Fields for Monocular 4d Facial Avatar Reconstruction," in *CVPR*, 2021, pp. 8649–8658 (cit. on pp. 17, 20, 21, 27).

[18] Gao, X., Zhong, C., Xiang, J., Hong, Y., Guo, Y., and Zhang, J., "Reconstructing Personalized Semantic Facial NeRF Models From Monocular Video," *SIGGRAPH Asia*, vol. 41, no. 6, 2022 (cit. on pp. 17, 21, 30).

[19] Garbin, S. J., Kowalski, M., Estellers, V., Szymanowicz, S., Rezaeifar, S., Shen, J., Johnson, M. A., and Valentin, J., "VolTeMorph: Real-time, Controllable and Generalizable Animation of Volumetric Representations," in *CGS*, Wiley Online Library, vol. 43, 2024, e15117 (cit. on pp. 7, 8, 10, 17–22, 25–27, 31, 34).

[20] Garbin, S. J., Kowalski, M., Johnson, M., Shotton, J., and Valentin, J., "FastNeRF: High-Fidelity Neural Rendering at 200FPS," in *ICCV*, 2021, pp. 14 346–14 355 (cit. on pp. 7, 20, 21).

[21] Grassal, P.-W., Prinzler, M., Leistner, T., Rother, C., Nießner, M., and Thies, J., "Neural Head Avatars From Monocular RGB Videos," in *CVPR*, 2022, pp. 18 653–18 664 (cit. on pp. 6, 20, 27).

[22] Green, R., "Spherical harmonic lighting: The gritty details," in *Archives of the game developers conference*, vol. 56, 2003, p. 4 (cit. on p. 11).

[23] Hedman, P., Srinivasan, P. P., Mildenhall, B., Barron, J. T., and Debevec, P., "Baking Neural Radiance Fields for Real-Time View Synthesis," in *ICCV*, 2021, pp. 5875–5884 (cit. on pp. 7, 20, 21, 26).

[24] Huang, B., Yu, Z., Chen, A., Geiger, A., and Gao, S., "2D Gaussian Splatting for Geometrically Accurate Radiance Fields," in *ACM SIGGRAPH 2024 Conference Papers*, 2024, pp. 1–11 (cit. on pp. 7, 8, 11).

[25] Huang, X., Zhang, Q., Feng, Y., Li, H., Wang, X., and Wang, Q., "HDR-NeRF: High Dynamic Range Neural Radiance Fields," in *CVPR*, 2022, pp. 18 398–18 408 (cit. on p. 20).

[26] Ichim, A. E., Bouaziz, S., and Pauly, M., "Dynamic 3D Avatar Creation from Hand-Held Video Input," *ACM Trans. Graph.*, vol. 34, no. 4, Jul. 2015, ISSN: 0730-0301 (cit. on p. 24).

[27] Irving, G., Teran, J., and Fedkiw, R., "Invertible Finite Elements For Robust Simulation of Large Deformation," in *SIGGRAPH*, 2004, pp. 131–140 (cit. on pp. 19, 21, 24, 25).

[28] Jiang, W., Yi, K. M., Samei, G., Tuzel, O., and Ranjan, A., "NeuMan: Neural Human Radiance Field from a Single Video," in *ECCV*, Springer, 2022, pp. 402–418 (cit. on p. 20).

[29] Kaleta, J., Kania, K., Trzcinski, T., and Kowalski, M., "LumiGauss: High-Fidelity Outdoor Relighting with 2D Gaussian Splatting," 2025 (cit. on pp. 8, 11, 12).

[30] Kania, K., Garbin, S. J., Tagliasacchi, A., Estellers, V., Yi, K. M., Valentin, J., Trzciński, T., and Kowalski, M., "BlendFields: Few-Shot Example-Driven Facial Modeling," in *CVPR*, 2023, pp. 404–415 (cit. on pp. 7, 8, 10, 12).

[31] Kania, K., Khirodkar, R., Saito, S., Yi, K. M., and Martinez, J., *CLoG: Leveraging UV Space for Continuous Levels of Detail*, 2024 (cit. on pp. 8, 9, 11, 12).

[32] **Kania**, **K.**, Kowalski, M., and Trzciński, T., "TrajeVAE: Controllable Human Motion Generation from Trajectories," *arXiv preprint arXiv:2104.00351*, 2021 (cit. on p. 12).

[33] Kania, K., Yi, K. M., Kowalski, M., Trzciński, T., and Tagliasacchi, A., "CoNeRF: Controllable Neural Radiance Fields," in *CVPR*, 2022 (cit. on pp. 6, 8, 10, 12, 17, 20).

[34] **Kania**, **K.**, Zięba, M., and Kajdanowicz, T., "UCSG-NET – Unsupervised Discovering of Constructive Solid Geometry Tree," *NeurIPS*, vol. 33, pp. 8776–8786, 2020 (cit. on p. 12).

[35] Kerbl, B., Kopanas, G., Leimkühler, T., and Drettakis, G., "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *TOG*, vol. 42, no. 4, pp. 139–1, 2023 (cit. on pp. 5, 7, 9, 11).

[36] Kim, M., Ko, J., Cho, K., Choi, J., Choi, D., and Kim, S., "AE-NeRF: Auto-Encoding Neural Radiance Fields for 3D-Aware Object Manipulation," *arXiv preprint arXiv:2204.13426*, 2022 (cit. on p. 20).

[37] Kingma, D. P. and Ba, J., "Adam: A Method for Stochastic Optimization," in *ICLR*, 2015 (cit. on p. 26).

[38] Lewis, J. P., Anjyo, K., Rhee, T., Zhang, M., Pighin, F., and Deng, Z., "Practice and Theory of Blendshape Facial Models," in *Eurographics 2014 - State of the Art Reports*, Lefebvre, S. and Spagnuolo, M., Eds., The Eurographics Association, 2014 (cit. on pp. 7, 19, 30).

[39] Li, T., Bolkart, T., Black, M. J., Li, H., and Romero, J., "Learning a model of facial shape and expression from 4D scans," *SIGGRAPH Asia*, vol. 36, no. 6, 194:1–194:17, 2017 (cit. on pp. 10, 17, 23, 30, 32).

[40] Liu, L., Gu, J., Zaw Lin, K., Chua, T.-S., and Theobalt, C., "Neural Sparse Voxel Fields," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 651–15 663, 2020 (cit. on p. 21).

[41] Liu, S., Zhang, X., Zhang, Z., Zhang, R., Zhu, J.-Y., and Russell, B., "Editing Conditional Radiance Fields," in *ICCV*, 2021, pp. 5773–5783 (cit. on pp. 6, 10, 17).

[42] Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., and Sheikh, Y., "Neural Volumes: Learning Dynamic Renderable Volumes from Images," *ACM Trans. Graph.*, vol. 38, no. 4, Jul. 2019, ISSN: 0730-0301 (cit. on p. 21).

[43] Lombardi, S., Simon, T., Schwartz, G., Zollhoefer, M., Sheikh, Y., and Saragih, J., "Mixture of Volumetric Primitives for Efficient Neural Rendering," *ACM Transactions on Graphics (ToG)*, vol. 40, no. 4, pp. 1–13, 2021 (cit. on p. 21).

[44] Ma, S., Simon, T., Saragih, J., Wang, D., Li, Y., De la Torre, F., and Sheikh, Y., "Pixel Codec Avatars," in *CVPR*, Jul. 2021, pp. 64–73 (cit. on pp. 17, 21, 30).

[45] Martin-Brualla, R., Radwan, N., Sajjadi, M. S., Barron, J. T., Dosovitskiy, A., and Duckworth, D., "NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections," in *CVPR*, 2021, pp. 7210–7219 (cit. on pp. 6, 20).

[46] Mihajlovic, M., Bansal, A., Zollhoefer, M., Tang, S., and Saito, S., "KeypointNeRF: Generalizing Image-based Volumetric Avatars using Relative Spatial Encoding of Keypoints," in *ECCV*, 2022 (cit. on p. 21).

[47] Mildenhall, B., Hedman, P., Martin-Brualla, R., Srinivasan, P., and Barron, J., "NeRF in the Dark: High Dynamic Range View Synthesis from Noisy Raw Images," in *CVPR*, 2022, pp. 16 190–16 199 (cit. on p. 20).

[48] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R., "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021 (cit. on pp. 6, 7, 17, 20, 26, 27, 31, 32, 37).

[49] Molino, N., Bridson, R., and Fedkiw, R., "Tetrahedral Mesh Generation for Deformable Bodies," 2003 (cit. on p. 21).

[50] Monk, P., "Finite Elements on Tetrahedra," in *Finite Element Methods for Maxwell's Equations*, Oxford, United Kingdom: Oxford University Press, Apr. 2003, pp. 99–154, ISBN: 9780198508885 (cit. on p. 24).

[51] Müller, T., Evans, A., Schied, C., and Keller, A., "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding," *ToG*, vol. 41, no. 4, 102:1–102:15, Jul. 2022 (cit. on pp. 7, 30).

[52] Noguchi, A., Iqbal, U., Tremblay, J., Harada, T., and Gallo, O., "Watch It Move: Unsupervised Discovery of 3D Joints for Re-Posing of Articulated Objects," in *CVPR*, 2022, pp. 3677–3687 (cit. on p. 20).

[53] Oat, C., "Animated Wrinkle Maps," in *SIGGRAPH*, ser. SIGGRAPH '07, San Diego, California: Association for Computing Machinery, 2007, pp. 33–37, ISBN: 9781450318235 (cit. on pp. 7, 8, 23).

[54] Park, K., Sinha, U., Barron, J. T., Bouaziz, S., Goldman, D. B., Seitz, S. M., and Martin-Brualla, R., "Nerfies: Deformable Neural Radiance Fields," in *ICCV*, 2021, pp. 5865–5874 (cit. on pp. 6, 20, 21, 26, 27, 31, 32, 37).

[55] Park, K., Sinha, U., Hedman, P., Barron, J. T., Bouaziz, S., Goldman, D. B., Martin-Brualla, R., and Seitz, S. M., "HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields," *ToG*, vol. 40, no. 6, 2021 (cit. on pp. 6, 10, 20, 21, 26, 27, 31, 32, 37).

[56] Patow, G. and Pueyo, X., "A Survey of Inverse Rendering Problems," in *Computer graphics forum*, Wiley Online Library, vol. 22, 2003, pp. 663–687 (cit. on p. 7).

[57] Pumarola, A., Corona, E., Pons-Moll, G., and Moreno-Noguer, F., "D-NeRF: Neural Radiance Fields for Dynamic Scenes," in *CVPR*, 2021, pp. 10 318–10 327 (cit. on pp. 6, 21).

[58] Ramamoorthi, R. and Hanrahan, P., "An efficient representation for irradiance environment maps," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 497–500 (cit. on pp. 7, 8).

[59] Raman, C., Hewitt, C., Wood, E., and Baltrusaitis, T., "Mesh-Tension Driven Expression-Based Wrinkles for Synthetic Faces," in *WACV Workshop on Applications of Computer Vision*, 2023 (cit. on p. 19).

[60] Reiser, C., Peng, S., Liao, Y., and Geiger, A., "KiloNeRF: Speeding Up Neural Radiance Fields With Thousands of Tiny MLPs," in *ICCV*, 2021, pp. 14 335–14 345 (cit. on p. 7).

[61] Rematas, K., Liu, A., Srinivasan, P., Barron, J., Tagliasacchi, A., Funkhouser, T., and Ferrari, V., "Urban Radiance Fields," in *CVPR*, 2022, pp. 12 932–12 942 (cit. on p. 20).

[62] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B., "High-Resolution Image Synthesis with Latent Diffusion Models," in *CVPR*, 2022, pp. 10 684–10 695 (cit. on p. 30).

[63] Rudnev, V., Elgharib, M., Smith, W., Liu, L., Golyanik, V., and Theobalt, C., "Neural Radiance Fields for Outdoor Scene Relighting," in *European Conference on Computer Vision*, Springer, 2022, pp. 615–631 (cit. on pp. 7, 8).

[64] Slomp, M. P. B., Oliveira Neto, M. M. d., and Patrício, D. I., "A gentle introduction to precomputed radiance transfer," *Revista de informática teórica e aplicada. Porto Alegre. Vol. 13, n. 2 (2006), p. 131-160*, 2006 (cit. on p. 7).

[65] Spurek, P., Winczowski, S., Zięba, M., Trzciński, T., **Kania**, **K.**, and Mazur, M., "Modeling 3D Surfaces with a Locally Conditioned Atlas," in *ICCS*, Springer, 2024, pp. 100–115 (cit. on p. 12).

[66] Stypułkowski, M., **Kania**, **K.**, Zamorski, M., Zięba, M., Trzciński, T., and Chorowski, J., "Representing Point Clouds with Generative Conditional Invertible Flow Networks," *Pattern Recognition Letters*, vol. 150, pp. 26–32, 2021 (cit. on p. 12).

[67] Suhail, M., Esteves, C., Sigal, L., and Makadia, A., "Light Field Neural Rendering," in *CVPR*, 2022, pp. 8269–8279 (cit. on p. 20).

[68] Sun, C., Sun, M., and Chen, H., "Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction," in *CVPR*, 2021, pp. 5449–5459 (cit. on p. 21).

[69] Sun, J., Wang, X., Zhang, Y., Li, X., Zhang, Q., Liu, Y., and Wang, J., "FENeRF: Face Editing in Neural Radiance Fields," in *CVPR*, 2022, pp. 7672–7682 (cit. on p. 20).

[70] Tagliasacchi, A. and Mildenhall, B., "Volume Rendering Digest (for NeRF)," *arXiv preprint arXiv:2209.02417*, 2022 (cit. on p. 21).

[71] Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P., Barron, J., and Kretzschmar, H., "Block-NeRF: Scalable Large Scene Neural View Synthesis," in *CVPR*, 2022, pp. 8248–8258 (cit. on p. 20).

[72] Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., and Ng, R., "Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains," in *NeurIPS*, 2020 (cit. on pp. 6, 8).

[73] Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Tretschk, E., Yifan, W., Lassner, C., Sitzmann, V., Martin-Brualla, R., Lombardi, S., *et al.*, "Advances in Neural Rendering," in *CGF*, 2022 (cit. on p. 17).

[74] Tretschk, E., Tewari, A., Golyanik, V., Zollhofer, M., Lassner, C., and Theobalt, C., "Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video," in *ICCV*, 2021 (cit. on p. 21).

[75] Vasconcelos, C. N., Oztireli, C., Matthews, M., Hashemi, M., Swersky, K., and Tagliasacchi, A., "CUF: Continuous Upsampling Filters," in *CVPR*, 2023, pp. 9999–10 008 (cit. on p. 9).

[76] Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J., and Srinivasan, P., "Ref-NeRF: Structured View-dependent Appearance for Neural Radiance Fields," in *CVPR*, 2022, pp. 5481–5490 (cit. on p. 20).

[77] Vicini, D., Jakob, W., and Kaplanyan, A., "Non-exponential transmittance model for volumetric scene representations," *ToG*, vol. 40, no. 4, Jul. 2021, ISSN: 0730-0301 (cit. on p. 20).

[78] Wang, C., Chai, M., He, M., Chen, D., and Liao, J., "CLIP-NeRF: Text-and-Image Driven Manipulation of Neural Radiance Fields," in *CVPR*, 2022, pp. 3835–3844 (cit. on p. 20).

[79] Wang, D., Chandran, P., Zoss, G., Bradley, D., and Gotardo, P., "MoRF: Morphable Radiance Fields for Multiview Neural Head Modeling," in *SIGGRAPH*, ser. SIGGRAPH '22, Vancouver, BC, Canada: Association for Computing Machinery, 2022, ISBN: 9781450393379 (cit. on p. 21).

[80] Wang, L., Zhang, J., Liu, X., Zhao, F., Zhang, Y., Zhang, Y., Wu, M., Yu, J., and Xu, L., "Fourier PlenOctrees for Dynamic Radiance Field Rendering in Real-time," in *CVPR*, 2022, pp. 13 524–13 534 (cit. on p. 20).

[81] Wang, Z., Simoncelli, E. P., and Bovik, A. C., "Multiscale Structural Similarity for Image Quality Assessment," in *Conference on Signals, Systems & Computers*, 2003 (cit. on p. 27).

[82] Weng, C.-Y., Curless, B., Srinivasan, P., Barron, J., and Kemelmacher-Shlizerman, I., "HumanNeRF: Free-viewpoint Rendering of Moving People from Monocular Video," in *CVPR*, 2022, pp. 16 210–16 220 (cit. on p. 20).

[83] Wood, E., Baltrušaitis, T., Hewitt, C., Dziadzio, S., Cashman, T. J., and Shotton, J., "Fake It Till You Make It: Face analysis in the wild using synthetic data alone," in *CVPR*, 2021, pp. 3681–3691 (cit. on pp. 23, 26).

[84] Wood, E., Baltrušaitis, T., Hewitt, C., Johnson, M., Shen, J., Milosavljević, N., Wilde, D., Garbin, S., Sharp, T., Stojiljković, I., *et al.*, "3D Face Reconstruction with Dense Landmarks," in *ECCV*, 2022, pp. 160–177 (cit. on p. 21).

[85] Wood, E., Baltrušaitis, T., Hewitt, C., Johnson, M., Shen, J., Milosavljević, N., Wilde, D., Garbin, S., Sharp, T., Stojiljković, I., *et al.*, "3D Face Reconstruction with Dense Landmarks," in *ECCV*, Springer, 2022, pp. 160–177 (cit. on p. 26).

[86] Wuu, C.-h., Zheng, N., Ardisson, S., Bali, R., Belko, D., Brockmeyer, E., Evans, L., Godisart, T., Ha, H., Hypes, A., Koska, T., Krenn, S., Lombardi, S., Luo, X., McPhail, K., Millerschoen, L., Perdoch, M., Pitts, M., Richard, A., Saragih, J., Saragih, J., Shiratori, T., Simon, T., Stewart, M., Trimble, A., Weng, X., Whitewolf, D., Wu, C., Yu, S.-I., and Sheikh, Y., "Multiface: A Dataset for Neural Face Rendering," in *arXiv*, 2022 (cit. on pp. 27, 28, 31, 36).

[87] Xia, S., Yue, J., **Kania**, **K.**, Fang, L., Tagliasacchi, A., Yi, K. M., and Sun, W., "Densify Your Labels: Unsupervised Clustering with Bipartite Matching for Weakly Supervised Point Cloud Segmentation," *arXiv preprint arXiv:2312.06799*, 2023 (cit. on p. 12).

[88] Xian, W., Huang, J.-B., Kopf, J., and Kim, C., "Space-time Neural Irradiance Fields for Free-viewpoint Video," in *CVPR*, 2021, pp. 9421–9431 (cit. on p. 20).

[89] Xiangli, Y., Xu, L., Pan, X., Zhao, N., Rao, A., Theobalt, C., Dai, B., and Lin, D., "BungeeNeRF: Progressive Neural Radiance Field for Extreme Multi-scale Scene Rendering," in *ECCV*, 2022, pp. 106–122 (cit. on p. 20).

[90] Xie, C., Park, K., Martin-Brualla, R., and Brown, M., "FiG-NeRF: Figure-Ground Neural Radiance Fields for 3D Object Category Modelling," 2021 (cit. on p. 6).

[91] Xu, K. and Campeanuy, D., "Houdini engine: Evolution towards a procedural pipeline," in *Proceedings of the Fourth Symposium on Digital Production*, 2014, pp. 13–18 (cit. on p. 28).

[92] Xu, T. and Harada, T., "Deforming Radiance Fields with Cages," in *ECCV*, 2022 (cit. on pp. 19, 21).

[93] Xu, Y., Wang, L., Zhao, X., Zhang, H., and Liu, Y., "AvatarMAV: Fast 3D Head Avatar Reconstruction Using Motion-Aware Neural Voxels," in *SIGGRAPH*, 2023, pp. 1–10 (cit. on p. 30).

[94] Yang, B., Bao, C., Zeng, J., Bao, H., Zhang, Y., Cui, Z., and Zhang, G., "NeuMesh: Learning Disentangled Neural Mesh-based Implicit Field for Geometry and Texture Editing," in *ECCV*, 2022, pp. 597–614 (cit. on pp. 20, 21).

[95] Yang, Y., Zhang, S., Huang, Z., Zhang, Y., and Tan, M., "Cross-Ray Neural Radiance Fields for Novel-view Synthesis from Unconstrained Image Collections," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 901–15 911 (cit. on p. 7).

[96] Yu, A., Li, R., Tancik, M., Li, H., Ng, R., and Kanazawa, A., "Plenoctrees for Real-time Rendering of Neural Radiance Fields," in *ICCV*, 2021, pp. 5752–5761 (cit. on pp. 7, 20, 21).

[97] Yuan, Y.-J., Sun, Y.-T., Lai, Y.-K., Ma, Y., Jia, R., and Gao, L., "NeRF-Editing: Geometry Editing of Neural Radiance Fields," in *CVPR*, 2022, pp. 18 353–18 364 (cit. on pp. 19–21).

[98] Zhang, J., Jiang, Z., Yang, D., Xu, H., Shi, Y., Song, G., Xu, Z., Wang, X., and Feng, J., "AvatarGen: A 3D Generative Model for Animatable Human Avatars," in *ECCV*, 2022, pp. 668–685 (cit. on p. 17).

[99] Zhang, K., Riegler, G., Snavely, N., and Koltun, V., "NeRF++: Analyzing and Improving Neural Radiance Fields," *arXiv:2010.07492*, 2020 (cit. on p. 20).

[100] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O., "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in *CVPR*, 2018 (cit. on p. 27).

[101] Zhang, X., Srinivasan, P. P., Deng, B., Debevec, P., Freeman, W. T., and Barron, J. T., "NeRFactor: Neural Factorization of Shape and Reflectance Under an Unknown Illumination," 2021 (cit. on p. 10).

[102] Zhao, F., Yang, W., Zhang, J., Lin, P., Zhang, Y., Yu, J., and Xu, L., "HumanNeRF: Efficiently Generated Human Radiance Field from Sparse Inputs," in *CVPR*, 2022, pp. 7743–7753 (cit. on p. 20).

[103] Zhi, Y., Qian, S., Yan, X., and Gao, S., "Dual-Space NeRF: Learning Animatable Avatars and Scene Lighting in Separate Spaces," in *3DV*, IEEE, 2022, pp. 1–10 (cit. on p. 17).

[104] Zhuang, Y., Zhu, H., Sun, X., and Cao, X., "MoFaNeRF: Morphable Facial Neural Radiance Field," in *ECCV*, 2022, pp. 268–285 (cit. on p. 21).

[105] Zielonka, W., Bolkart, T., and Thies, J., "Towards Metrical Reconstruction of Human Faces," in *ECCV*, Springer, 2022, pp. 250–269 (cit. on p. 6).

[106] Zielonka, W., Bolkart, T., and Thies, J., "Instant Volumetric Head Avatars," in *CVPR*, 2023, pp. 4574–4584 (cit. on pp. 6, 30).

[107] Zwicker, M., Pfister, H., Van Baar, J., and Gross, M., "EWA volume splatting," in *Proceedings Visualization, 2001. VIS'01.*, IEEE, 2001, pp. 29–538 (cit. on p. 5).