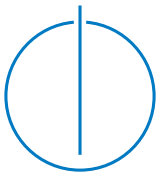
SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Endoscopic NeRF: Neural Rendering for Dynamic Endoscopic Scenes

Florian Philipp Stilz



SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

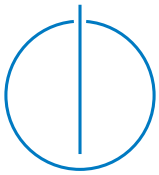
TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Endoscopic NeRF: Neural Rendering for Dynamic Endoscopic Scenes

Endoskopie NeRF: Neurales Rendering für dynamische Endoskopie Szenen

Author:	Florian Philipp Stilz
Supervisor:	Prof. Dr. Nassir Navab
Advisor:	Mert Asim Karaoglu and Felix Tristram
Submission Date:	February 3, 2024



I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

München, February 3, 2024 Florian Philipp Stilz

Abstract

Addressing the challenge of Novel View Synthesis in highly deformable surgical scenes with dynamic camera movements is crucial for medical applications. This thesis aims at modeling highly deformable endoscopic scenes. We achieve this by utilizing an implicit Dynamic Neural Radiance Fields method called HexPlane [6] and expand on it by separating it into several smaller models by need. This ensures a higher representational capacity under constant GPU memory consumption and independent of the scene size and length. Furthermore, we aim to tackle the issue of suboptimal camera poses by optimizing for camera poses from scratch in parallel to scene reconstruction. We call our approach "FLex" for "Flow-Optimized Local HexPlanes". Our method demonstrates high-quality reconstruction results, surpassing existing approaches, all while eliminating the need for pre-processing camera poses and enabling on-the-fly pose optimization.

Acknowledgments

First and foremost, I would like to thank my supervisor Prof. Dr. Nassir Navab for giving me the opportunity to participate in an extremely exciting research project. A special acknowledgment is owed to my two advisors, Mert Asim Karaoglu and Felix Tristram, for their unwavering support and invaluable feedback throughout the thesis. In addition, I am grateful for the opportunity to conduct my thesis in a friendly and competent environment at ImFusion, which helped me excel academically. For this, I extend my thanks to both Dr. Wolfgang Wein and Dr. Alexander Ladikos. Furthermore, I would also like to thank Alexander Schwer for the inspiration to solve several computer vision problems. Last but most importantly, I would like to thank my parents for all their support throughout the years.

To my grandparents who gave me so much love and inspiration

Meinen Großeltern, die meinen Werdegang immer wieder liebevoll inspiriert haben

Contents

1 Introduction

The reconstruction of surgical scenes, particularly those involving highly deformable tissues in endoscope stereo videos, presents a challenging yet crucial task. Achieving high-quality reconstruction of surgical scenes opens avenues for creating virtual in-vitro environments, which can be leveraged for training medical personnel through augmented reality (AR) or virtual reality (VR). It also serves as a valuable database for training surgical robots, a pivotal step in advancing automated surgical procedures. Real-time reconstruction offers the potential for more precise navigation of surgical instruments during in-vitro operations or screenings. Instead of relying solely on endoscope stereo video images, surgeons can analyze real-time 3D visualizations of surgical scenes, enhancing the efficiency and accuracy of surgical procedures. The main challenges within surgical scenes are strong deformations, often caused by cutting or pulling with surgical tools. There are also illumination highlights, blurriness, and visual occlusions. Additionally, there is limited scene coverage, caused by fast camera movements yielding limited information about the 3D nature of the scene.

Novel View Synthesis is one method to obtain reconstruction by training methods based on image sequences. The state-of-the-art approach for Novel View Synthesis within the Computer Vision community, achieving exceptional results, is Neural Radiance Fields [44]. They can model scene geometry and view-dependent effects by utilizing both coordinates and viewing direction as input. However, several aspects still need further improvements, especially the dynamic scene modeling for highly deformable scenes, such as endoscopic scenes. Recently, several approaches have been addressing the issue of improving Dynamic Neural Radiance Fields by either modeling for an explicit representation [53, 50, 51] with shared geometry information but limited capability of modeling topological changes or an implicit representation approach [28, 6, 16] with more outstanding representational capabilities but also highly unconstrained. Additionally, several methods adopt an implicit modeling approach while predicting scene flow [30, 18, 31]. Some of these works have also been converted to be better suited for endoscopic scenes. The first Neural Radiance Fields method for surgical scenes was EndoNeRF [70], applying an explicit Dynamic Neural Radiance Fields, in which a deformation field models the topological changes. Several other works extended on EndoNeRF, such as EndoSurf [78], paying greater attention to the geometrical consistency within scenes via a Signed Distance Field and LerPlane [76] improved on both rendering speed and reconstruction quality by utilizing an advanced implicit Dynamic Neural Radiance Fields. However, all of these mentioned Endoscopic Neural Radiance Fields only showcase their performance on shorter deformable scenes

(max. 200 frames) without any camera movement. Static camera scenes are unlikely for real-world applications, especially for longer video sequences as in actual surgical operations.

Introducing camera movement to the problem increases its difficulty level drastically and introduces another potential problem for real-world scenes: the need for well-calibrated camera poses beforehand. Camera poses are an essential requirement for Neural Radiance Fields, and for highly dynamic scenes, standard methods often yield noisy poses, significantly impacting reconstruction quality. Several approaches have been targeting unreliable poses by jointly optimizing for reconstruction and poses for static scenes [32, 65, 43], achieving significant improvements over non-pose optimizing methods. In some cases, it is possible to generate poses from scratch for large scenes as in LocalRF [43]. Generating an entire camera pose trajectory from scratch makes it possible to remove the pre-processing step typically required for Neural Radiance Fields. However, no method tackles the issue of modeling for large dynamic scenes while optimizing poses from scratch.

This thesis presents a method aiming to provide Neural Radiance Fields capable of representing large, highly deformable endoscopic scenes while avoiding the necessity of pre-processing camera poses beforehand and obtaining camera poses on the fly. For this, the presented method combines ideas from HexPlane [6] to model for highly deformable scenes, Mip-NeRF 360 [4] for an efficient scene representation and modeling of large scenes, Urban Radiance Fields [56] for depth supervision and to achieve greater geometric consistency. LocalRF [43] to model for large scenes spatial-wise and, more importantly, temporal-wise. Additionally, we incorporate progressive optimization from LocalRF to jointly optimize for poses and reconstruction. Our method is evaluated on several scenes from the StereoMIS dataset [20] for both reconstruction quality and camera pose trajectory correctness.

The structure of the thesis is outlined as follows: The second chapter provides a comprehensive review of related work, followed by Chapter 3, which elucidates our approach and the selected methodologies. Chapter 4 delves into the available data and offers additional implementation details along with the ultimate results. Chapter 5 concludes with a discussion of the limitations inherent in our approach, a summarizing conclusion, and the presentation of potential extensions for future projects.

2 Related Work

Contents

This chapter will introduce several related studies and foundational terms for this project. It starts with an explanation of Neural Rendering and Novel View Synthesis, before introducing several related methods.

2.1 Neural Rendering

The goal of rendering in computer graphics is to synthesize photo-realistic images given a 3D scene. Traditional computer graphics can generate controllable and high-quality images when provided with physical parameters such as camera position, illumination, and the object's materials. Those physical parameters are required as traditional rendering methods try to imitate the image formation model of cameras, such as the global illumination and processing of complex materials via simulating the light transport from the light source to the virtual camera. The main limitation of traditional rendering methods is the difficulty of generating controllable images for real-world scenes due to the fact that the physical parameters need to be estimated. The typical solution is to perform inverse rendering based on existing images of a scene. Inverse rendering is, however, naturally very challenging.

An alternative approach is Neural Rendering, where the main idea is to combine classical computer graphics ideas with Neural Networks. The main task of Neural Rendering methods is, similarly to its classical counterpart, to represent or render 3D scenes based on real-world images. The Input can be a set of structured or unordered images that can be taken from one or several cameras simultaneously. An important aspect of 3D neural rendering is the disentanglement of the projection and image formation and the 3D scene representation during training. The main advantage of this disentanglement is the fact that it leads to a high level of 3D consistency during the synthesis of images, e.g., for novel viewpoint synthesis. Furthermore, Neural Networks are incorporated into the classical pipelines to learn specific parts of the rendering pipeline, e.g., scene representation from vast quantities of images. Noteworthy is that the entire rendering pipeline must be differentiable to optimize the Neural Networks via gradient descent. Two main approaches that are taken from classical computer graphics and need to be adjusted to be differentiable are differentiable volume rendering [61], [47], like in the case of Neural Radiance Fields [44], and the much faster approach during inference with differentiable rasterization [24], [73], [64].

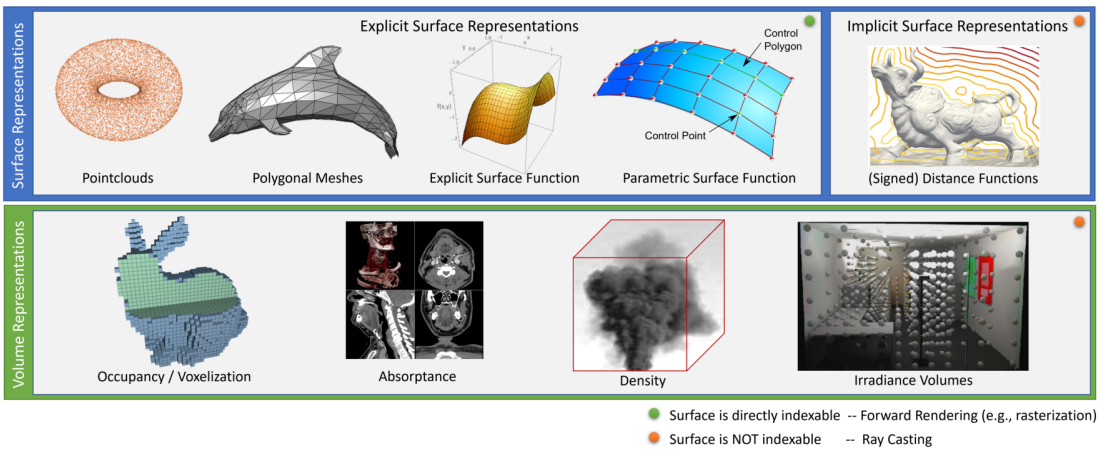


Figure 2.1: **Traditional 3D Scene Representations:** Categorized into surface and volume representations. This figure is taken from [63].

2.1.1 3D Scene Representation

Traditionally, there are two central representations of 3D scenes: volumetric and surface representations. Volumetric representations can also represent surfaces and contain volumetric information like densities, occupancies, and opacities. They might also contain additional properties such as colors as multidimensional features. Conversely, surface representations cannot represent volumes and can commonly be represented either explicitly or implicitly. Point clouds or meshes are common explicit representations of surfaces, whereas a signed distance function is an example of an implicit surface representation. Further details are depicted in Figure 2.1. These representations can be approximated via, e.g., Gaussian mixture models like, e.g., Radial basis functions [7]. Alternatively, Neural Scene Representations describes the methods, where volumetric and surface representations are approximated via Neural Networks, e.g., via Multi-Layer Perceptrons [55].

2.1.2 Novel View Synthesis

Novel View Synthesis is an application of Neural Rendering and covers the rendering of novel images for a given static scene from an unseen camera pose. This formulation can even be extended to dynamic scenes, where not only the camera pose changes but also objects within a scene are moving or deforming. Novel View Synthesis methods must

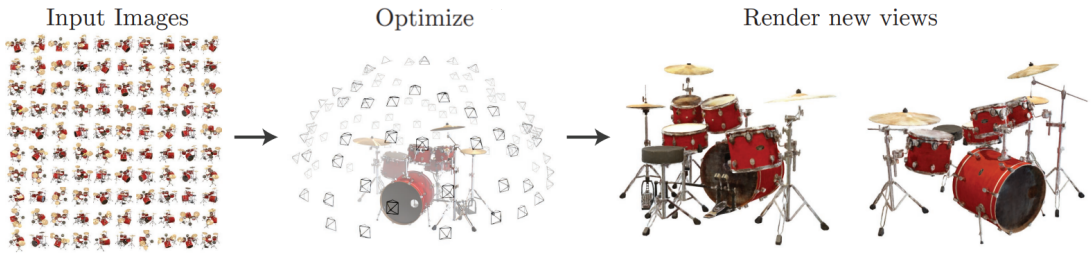


Figure 2.2: **Novel View Synthesis Pipeline:** It commonly starts by gathering initial images and optimizes the method according to the pixel values. Finally, the trained method is able to render photo-realistic images for novel views. This figure is adapted from [44].

be capable of not only providing realistic-looking novel images but, more importantly, ensuring 3D consistency. A lack of 3D consistency can, e.g., yield artifacts within the images and or flickering from image to image when moving through the scene with the camera.

For dense view coverage of static scenes, simple light field interpolations can be utilized to obtain photo-realistic results [27], [11], [39]. For sparser view coverage, other methods were developed, e.g., using mesh-based representations of a scene with either view-dependent [74], [5], [12] or diffuse [68] appearance. These representations can be further optimized using gradient descent methods and differentiable rasterizers [38], [19], [35], [10] or path-tracers [48], [29]. The two main issues with optimizing mesh representations for image re-projections, as needed for Novel View Synthesis, are poor conditioning of the training loss landscape, making it difficult to optimize, and the unavailability of initial mesh templates for real-world scenes.

3D Voxel Grid Representations

A different approach is to use Deep Learning and Neural Networks to tackle the problem of Novel View Synthesis, called Neural Scene Representation methods. The first method using Deep Learning for Novel View Synthesis was DeepStereo [15]. It

utilized a Convolutional Neural Network (CNN) to observe input images as a plane sweep volume (PSV). In a PSV, each nearby input is reprojected to a set of candidate depth planes. This way, the network evaluates how well the reprojections match for each pixel at each candidate depth. The model outputs are passed through a softmax function representing a probability distribution over depths. Finally, the distribution is used to combine the input images. The biggest drawback of DeepStereo is the individual computation of output frames, resulting in a slow running time and primarily a low 3D consistency.

Stereo Magnification [81] addresses the challenges encountered by DeepStereo using a different CNN architecture that processes a PSV directly into a multiplane image. A multiplane image is an output persistent 3D voxel grid representation. Rendering new viewpoints involves using alpha compositing on the RGB-alpha grid from a different location. To enhance image quality, Stereo Magnification deliberately distorts the parameterization of its 3D grid, biasing it towards the frame of reference of one of the two input views. While this strategy significantly reduces memory consumption in the dense grid, it comes at the cost of restricting the rendering of new views to the immediate vicinity of the input stereo pair.

Both DeepStereo and Stereo Magnification require large datasets of input/output pairs. A different approach adopted by, e.g., DeepVoxels [60] and Neural Volumes [36] is to optimize a method exclusively on the desired scene. DeepVoxel's method combines a learned renderer with an optimizable 3D voxel grid of features. Furthermore, its parametrization of the 3D voxel grid is not biased towards a particular viewing direction, which in contrast to Stereo Magnification, allows for rendering novel views from entirely different directions. The same applies to Neural Volumes, which employs a 3D CNN to compute a volumetric representation for a single scene initially captured via multi-view videos. Although voxel-based representations are effective for view synthesis, they suffer from memory issues overall and do not scale well to scenes at higher resolutions.

Scene Representation Networks

Scene Representation Networks (SRNs) [61] try to resolve the memory and resolution issues of 3D voxel grid representations. Similar to Neural Radiance Fields, these methods adopt Multi-Layer Perceptrons in combination with a sphere-tracing based

neural renderer. The task of the Multi-Layer Perceptron in SRNs is to learn to depict scenes. The overall task of SRNs is generalization across scenes to enable few-shot reconstruction. Another method is Differentiable Volumetric Rendering (DVR) [46], which similarly leverages an approach to render surfaces and reveals that overfitting on single scenes enables the reconstruction of more complex appearance and geometry.

2.2 Neural Radiance Fields

Neural Radiance Fields [44] can be considered a Scene Representation Network; more precisely, it is a Coordinate-Based Neural Representation (CBNR). Neural Radiance Fields approximate a continuous volume integral utilizing a Gaussian quadrature to render a desired pixel from multiple 3D points along a ray cast into the scene from the camera position. This ray-casting is done by querying a Multi-Layer Perceptron, which, for each query, outputs color and density values that are then accumulated to produce a pixel value for this ray. One significant contribution to achieving photo-realistic results is the employment of Positional Encodings (PE) on the input values before passing them to the Multi-Layer Perceptron. Using PEs allows Neural Radiance Fields' Multi-Layer Perceptron to represent much higher frequency signals without increasing its network capacity.

2.2.1 Advancements in Neural Radiance Fields

Several modifications to Neural Radiance Fields prioritize the enhancement of view synthesis quality. This section underscores crucial models aimed at refining both the photometric and geometric aspects of Neural Radiance Fields view synthesis and 3D scene representation, improving the quality of synthesized images.

NeRF++ [79] innovates in generating novel views for unbounded scenes by partitioning the scene with a sphere. The sphere encapsulates all foreground objects and artificial camera views, while the background lies outside the sphere, reparameterized in an inverted sphere space. Two distinct NeRF models are trained, one for the sphere's interior and one for the exterior. The evaluation of the camera ray integral is also conducted separately. NeRF++ additionally addresses the tendency of overfitting to high-frequency components under incorrect density predictions regarding the viewing

angles by enforcing lower-frequency components due to introducing the viewing angles in later layers of the Multi-Layer Perceptron. Doing this leads to smoother color predictions.

NeRF in the Wild (NeRF-W) [40] tackles two major challenges in the original Neural Radiance Fields. Real-world photos of the same scene may exhibit per-image appearance variations due to distinct lighting conditions and transient objects that differ in each image. While the density Multi-Layer Perceptron remains constant for all images in a scene, NeRF-W introduces conditioning of their color Multi-Layer Perceptron on a per-image appearance embedding. Additionally, another Multi-Layer Perceptron, conditioned on per-image transient embedding, predicts transient objects' color and density functions. These latent embeddings are constructed through Generative Latent Optimization. It is worth noting that NeRF-W's effective approach results in a slower running time compared to the original Neural Radiance Fields [44].

Mip-NeRF [3] tackles the challenge of aliasing effects observed in the standard Neural Radiance Fields [44] by incorporating a multi-scale representation. Departing from ray tracing in the original Neural Radiance Fields volume rendering, Mip-NeRF employs cone tracing. This involves casting a cone from the camera's center along the viewing direction through the pixel's center. To facilitate cone tracing, Mip-NeRF introduces Integrated Positional Encoding (IPE), representing the cone as a multivariate Gaussian. The mean vector and variance matrix are tailored for the desired geometry, giving rise to the Integrated Positional Encoding.

To adapt Mip-NeRF [3] for unbounded scenes, Mip-NeRF 360 [4] is introduced with pivotal technical enhancements. These include integrating a proposal network employing a Multi-Layer Perceptron, a revamped scene parametrization, and a novel regularization approach. The proposal network undergoes supervision solely from the Neural Radiance Fields Multi-Layer Perceptron, predicting volumetric density for determining optimal sampling intervals without delving into color prediction. The explicit construction of the scene parametrization is tailored for the Gaussians in Mip-NeRF and performs a scene contraction by following the idea of the classic extended Kalman filter. Additionally, the innovative regularization method prevents artifacts related to geometric floaters and background collapse better than the original Neural Radiance Fields [44] anti-floater measurements that inject noise.

Speed Improvements for Neural Radiance Fields

One major concern of Neural Radiance Fields is its rendering speed. This is due to the fact that the network needs to be queried for every pixel in an image t times, where t stands for the sampling size of each ray. Thus the higher the desired image resolution the slower the network's rendering speed.

Neural Sparse Voxel Fields [33] builds upon Neural Radiance Fields by introducing a sparse voxel field representation, effectively enhancing rendering speed. This technique merges a sparse voxel octree with a neural network to depict scene appearance. Feature representations are obtained by interpolating learnable features stored at voxel vertices, then processed by the Multi-Layer Perceptron. This combination allows for increased empty space skipping and an early ray termination. Those concepts facilitate the efficient reconstruction and rendering of large-scale scenes. However, it is more memory-intensive due to storing feature vectors on a voxel grid.

KiloNeRF [55] adapts the concept of empty space skipping and early ray termination from Neural Sparse Voxel Fields [33] and combines it with the idea of using thousands of much smaller and faster Multi-Layer Perceptrons instead of one large one like in [44]. The smaller Multi-Layer Perceptrons are assembled in a dense 3D grid where each Multi-Layer Perceptron represents a tiny fraction of the scene. Furthermore, KiloNeRF employs teacher-student distillation during training to optimize each tiny Multi-Layer Perceptron and thereby avoids sacrificing visual quality. The speed improvements of KiloNeRF make it more suitable for real-time applications while maintaining high-quality results.

PlenOctree [77] has further optimized running time. Instead of directly predicting the color function, it employs a spherical harmonic Neural Radiance Fields (NeRF-SH), forecasting the spherical harmonic coefficients of the color function. An octree was constructed using pre-computed spherical harmonic coefficients of the Multi-Layer Perceptron's colors. Voxelization of the scene occurred during octree building, eliminating low transmissivity voxels. This voxelization process could also be applied to standard Neural Radiance Fields (non-NeRF-SH models) through Monte Carlo estimations of the spherical harmonic components. Fine-tuning of PlenOctrees with initial training images was performed, faster than Neural Radiance Fields training.

The method achieves an impressive 3000 times faster inference time than the original Neural Radiance Fields [44].

Numerous contemporary strategies have surfaced [75], [14], [52], [8], employing classical data structures like grids, sparse grids, trees, and hashes to enhance rendering speed and streamline training. One notable example is Instant Neural Graphics Primitives (Instant-NGP) [45], which achieves NeRF training in seconds through the utilization of a multi-resolution hash encoding, departing from the traditional explicit grid structure. Furthermore, Instant-NGP also employs sophisticated ray marching techniques, including exponential stepping, empty space skipping, and sample compaction. This new multi-resolution hash encoding and associated optimized implementation of a Multi-Layer Perceptron significantly improved the training and inference speed and scene reconstruction accuracy of the resulting Neural Radiance Fields model.

Geometrical Constraints for Neural Radiance Fields

Employing depth supervision, utilizing point clouds from LiDAR or SfM, facilitates faster convergence, attains higher final quality, and demands fewer training views compared to the original Neural Radiance Fields model. Additionally, these methods demonstrate superior 3D consistency.

Deng et al. introduce Depth-Supervised Neural Radiance Fields (DS-NeRF) [13], leveraging depth supervision from point clouds. Alongside color supervision through volume rendering and photometric loss, DS-NeRF incorporates depth supervision using sparse point clouds extracted from training images via COLMAP [58]. Depth is represented as a normal distribution centered around the depth recorded by the sparse point cloud. A KL divergence term is introduced to minimize the divergence between the ray's and noisy depth distributions.

Another method that uses depth supervision and outperforms DS-NeRF is Urban Radiance Fields [56]. Urban Radiance Fields seeks to leverage Neural Radiance Fields for novel view synthesis and 3D reconstruction of urban environments, utilizing sparse multi-view images and LiDAR data. Beyond the standard photometric loss, they incorporate LiDAR-based depth loss $\mathcal{L}_{depth}$, sight loss $\mathcal{L}_{sight}$, and skybox-based segmentation loss $\mathcal{L}_{seg}$. The depth loss aligns the estimated depth with LiDAR-acquired depth, while the sight loss concentrates radiance at the surface of the measured depth. The segmentation loss ensures zero density for point samples along rays through sky pixels.

Other approaches like NeuS [69] try to achieve higher 3D consistency via employing several new geometric constraints on Neural Radiance Fields. NeuS suggests employing a Signed Distance Field (SDF) representation instead of volume densities for reconstructing scenes from RGB-D data. They associate volume density with a Signed Distance Field, reparameterizing the transmittance function to ensure its maximal slope precisely at the zero-crossing of this SDF. This configuration enables an unbiased estimate of the corresponding surface. Using root finding on the SDF allows for defining explicit surface geometry for the scene. HF-NeuS [71] enhances NeuS by segregating low-frequency details into a base SDF and high-frequency details into a displacement function, substantially improving reconstruction quality. Simultaneously, Geo-NeuS [17] introduces new multi-view constraints, such as a multi-view geometry constraint for the SDF supervised by sparse point clouds and a multi-view photometric consistency constraint. SparseNeus [37], another concurrent approach, advances NeuS by concentrating on sparse-view SDF reconstruction through a geometry encoding volume with learnable image features, employing a hybrid representation method.

2.2.2 Dynamic Neural Radiance Fields

All previously discussed Neural Radiance Fields methods represent static scenes and objects. However, some approaches can additionally handle dynamically changing content. These methods for Dynamic Neural Radiance Fields can be categorized into two types of representations, implicit or explicit. Implicit with regards to this case means conditioning the Neural Radiance Fields on a representation of the time input, and explicit methods use a separate deformation field that maps from the deformation to a canonical space from where the standard Neural Radiance Fields can operate as before.

Explicit Methods

Explicit methods decouple deformations from geometry and appearance by separating deformations into a distinct function layered onto a static canonical scene. Unlike implicit modeling, these techniques inherently share geometry and appearance information across time by incorporating the static canonical scene. Deformations are achieved by directing rays straight into deformed space and bending them into the canonical scene, typically by determining per-point offsets for points on the straight ray through a Multi-Layer Perceptron conditioned on the deformation. This concept can be considered as space warping or scene flow. This construction ensures rigid correspondences, thereby eliminating drift. However, constrained by this rigid structure, explicit deformation methods struggle with topological changes and are most effective in scenes with considerably smaller motions than their implicit counterparts.

One of the first explicit Dynamic Neural Radiance Fields is D-NeRF [53]. D-NeRF captures time-varying appearance and geometry by conditioning the neural radiance field on a time-varying latent code. The technique incorporates an extra loss term to ensure temporal smoothness in the acquired radiance field, facilitating the synthesis of high-quality views in dynamic scenes with temporal consistency. This methodology can reconstruct dynamic scenes from multi-view videos, generating novel views that sustain appearance and motion continuity.

Another explicit method is Nerfies [50], in which the deformable Neural Radiance Fields condition deformations and appearance with an auto-decoded latent code per input view. The bent rays are regularized using an elastic regularization that penalizes deviations from piecewise rigid scene configurations. Furthermore, it includes background regularization and coarse-to-fine deformation regularization with the help of adaptive masking of the positional encodings. This adaptive masking scheme ensures that the model concentrates initially on learning low-frequency content, thereby avoiding overfitting to high-frequency components.

Nerfies sees enhancement in HyperNeRF [51], which employs a canonical hyperspace rather than a singular canonical time, enabling the handling of scenes with topological changes. HyperNeRF augments the deformation field that bends rays into canonical space via a Multi-Layer Perceptron regression task with an ambient slicing surface

network. This field network selects a canonical subspace for each input RGB view, indirectly conditioning the canonical scene on the deformation. It represents a hybrid model, combining explicit and implicit deformation modeling, allowing for the management of topological changes at the cost of sacrificing hard correspondences.

Implicit Methods

Implicit methods for Dynamic Neural Radiance Fields tackle the problem of representing deformable scenes by adding time as an additional input besides the spatial coordinates and the viewing direction. However, this approach is highly unconstrained, and therefore most methods try to tackle the problem by introducing geometric scene priors for regularization purposes.

DyNeRF's [28] Dynamic Neural Radiance Field is based on temporal latent codes. Each temporal latent embedding represents one timestep/frame of the scene, and all latent codes are jointly with the Multi-Layer Perceptron optimized during training. It also includes a novel training strategy based on hierarchical training and importance sampling in the spatiotemporal domain, which boosts training speed significantly and leads to higher-quality results for longer sequences. Uniformly sampling rays in dynamic scenes cause an imbalance between time-invariant and time-variant observations, making sampling highly inefficient and negatively impacting reconstruction quality. Therefore, rays around regions of higher temporal variance are preferred in importance sampling. Hierarchical training follows the idea of optimizing data over a coarse-to-fine frame selection.

Several concurrent works that are much faster than existing Dynamic Neural Radiance Fields have emerged. Two of those are HexPlane [6] and K-Planes [16], which both utilize an explicit scene parametrization via a learnable 4D feature grid volume. The features from the 4D volume are jointly optimized with smaller Multi-Layer Perceptrons that produce the emitted color and volume density. The reduction of the Multi-Layer Perceptrons helps in boosting training and inference speed. Both methods utilize factorization similarly as in TensoRF [8] to model the 4D feature grid as memory efficient as possible. Utilizing an explicit data structure helps boost training speed and considerably improve reconstruction quality, allowing HexPlane to outperform DyNeRF by a significant margin.

Optical Flow Supervision For Dynamic Neural Radiance Fields

An alternative approach involves modeling scene flow mappings between temporally adjacent time steps to provide additional regularization for Dynamic Neural Radiance Fields. The rationale behind incorporating scene flow is to promote the consistency of reflectance and opacity across different moments. The training of scene-flow mapping entails reconstruction losses that warp the scene from other time steps into the current time step, ensuring consistency between the estimated optical flow and the 2D projection of the scene flow. This can be achieved by tracking back-projected key points in 3D. Additional regularization losses are often applied to further constrain the scene flow, encouraging spatial or temporal smoothness and forward-backward cycle consistency.

One method called Neural Scene Flow Fields (NSFF) [30] incorporates scene flow by warping rays into the forward and backward time steps and rendering at these times for the warped points to obtain the warped pixels, which are optimized with the photometric reconstruction loss using the original pixel values at the initial time step. Furthermore, NSFF predicts occlusion weights for the warped photometric losses to avoid incorrect supervision. Additionally, NSFF includes several regularization losses, e.g., a smoothness loss, slow scene flow loss, and a cycle consistency loss for forward and backward scene flow. It also contains an induced optical flow loss to provide additional supervision for the scene flow.

Dynamic-NeRF [18] improves over NSFF by introducing additional regularization losses. It includes spatial and temporal smoothness losses and an entropy loss on the network density weights since scene flow is not an intrinsic property, unlike color, meaning it is impossible to see through it. Therefore, the amount of impactful weights is encouraged to be reduced. The most significant contribution of Dynamic-NeRF is the utilization of two Neural Radiance Fields models, one modeling dynamic parts and the other static parts of the scene.

A recent advancement known as DynIBaR [31] builds upon the concept introduced by NSFF, utilizing scene flow to compute a warped photometric consistency loss, referred to as cross-time rendering, for achieving temporal consistency. Unlike the NSFF's approach of predicting occlusion weights using a Multi-Layer Perceptron, DynIBaR computes them by taking the difference of accumulated alpha weights between frames.

Additionally, DynIBaR incorporates 2D features generated by a CNN network as an additional input. Like Dynamic-NeRF, DynIBaR divides the model into two Neural Radiance Fields, one dedicated to static parts and the other to dynamic elements. The partition is enhanced by leveraging motion segmentation masks for supervision, computed through an additional CNN model before the actual training.

2.2.3 Pose Optimization With Neural Radiance Fields

All the previously mentioned methods require camera poses as input for frame supervision. These poses are typically acquired during pre-processing using techniques like Structure from Motion (SfM), such as COLMAP [58]. However, real-world scenes often present challenges, including complete failure cases of COLMAP, leading to either the absence of generated poses or the generation of incorrect ones. These issues significantly impact the reconstruction performance of Neural Radiance Fields. Consequently, several Neural Radiance Fields approaches tackle these challenges by incorporating pose optimization into their training process, simultaneously optimizing for poses while training for Novel View Synthesis.

The Bundle-Adjusted Neural Radiance Field (BARF) [32] is a method that concurrently estimates poses while training Neural Radiance Fields. Employing a coarse-to-fine optimization strategy, BARF dynamically masks positional encodings, akin to the approach seen in Nerfies. This adaptive masking of positional encodings contributes to a smoother optimization process, enhancing joint camera registration and Novel View Synthesis.

SPARF [65] adopts the idea of using a coarse-to-fine positional encoding from BARF. It includes a multi-view correspondence loss targeting to learn globally consistent poses across multiple views. The approach is similar to NSFF's cross-time rendering to supervise via mapping 3D correspondences into the correspondence frames and using those pixel values for the photometric loss. Notably, in contrast to NSFF, this assumes a static scene as it does not model deforming objects. Additionally, SPARF introduces a depth consistency loss utilizing the predicted depth for one view and warping it to an unseen view for pseudo-depth supervision of the unseen view.

However, both methods of optimizing for poses are incapable of learning reasonable poses from scratch for 360-degree scenes, especially not with dynamic motion within the scenes.

The "Progressively Optimized Local Radiance Fields for Robust View Synthesis" method, also called LocalRF [43], aims to tackle large unbounded scenes. This study tries to optimize for poses from scratch while also optimizing for the reconstruction. Besides using TensoRf as its main building block, it also utilizes a progressive training scheme in which new frames are added iteratively to an existing model. This progressive optimization helps immensely in generating meaningful trajectories. Furthermore, it separates the scene into smaller local models trained one after another with an inevitable overlap in training views to remain connected. This method yields high-quality reconstruction and reasonable pose trajectories and avoids the need for pre-processing poses even for 360-degree scenes. However, this approach does not handle deforming objects.

2.2.4 Neural Radiance Fields For Endoscopic Scenes

Several works are dedicated to endoscopic scene reconstruction, but they often focus on specific scenarios, such as deforming scenes with a fixed camera or static scenes with a moving camera.

One notable work in this domain is EndoNeRF [70], which introduces the EndoNeRF dataset featuring deforming endoscopic scenes with a stationary camera. Leveraging the D-NeRF [53] architecture, an explicit Dynamic Neural Radiance Field model, EndoNeRF employs a tool mask to exclude surgical instruments from the reconstruction. The approach incorporates tool-guided ray marching to selectively sample points behind surgical tools when casting rays for tool pixels, enhancing the reconstruction process. Depth supervision is also utilized to improve geometric reconstruction accuracy.

EndoSurf [78] is a specialized endoscopic scene reconstruction method, drawing inspiration from NeuS [69]. It tackles the challenge of achieving smoother surfaces by adopting a Signed Distance Field (SDF) representation. The architecture is structured into three models: two for the conventional extrinsic Dynamic Neural Radiance Fields pipeline and one dedicated to surface representation. With the incorporation of various regularization losses to constrain surface representation, EndoSurf demonstrates notable improvements in 3D reconstructions compared to EndoNeRF.

LerPlane [76] emerges as the latest state-of-the-art method for visual reconstruction quality on the EndoNeRF dataset. It integrates the architecture of K-Planes [16] with ray importance sampling, leveraging DyNeRF's importance sampling scheme along with tool masking to prioritize rays during training. Additionally, LerPlane incorporates a proposal network similar to Mip-NeRF 360's, a separate model predicting volume density to determine the sampling range for each ray. Notably, LerPlane outperforms both EndoNeRF and EndoSurf on the EndoNeRF dataset while exhibiting a significantly faster performance than its predecessors.

3 Method

Contents

This chapter describes the methods and architectures used to generate novel views for strongly deformable endoscopic scenes with moving cameras.

3.1 Pose Pre-processing

Before explaining the detailed concept of Neural Radiance Fields, it is beneficial to take a closer look at a standard pre-processing step for Neural Radiance Fields, which is acquiring the camera trajectory of a scene. Camera poses are important for Neural Radiance Fields since they are required for putting the supervising images into relation. Thus allowing a model to learn meaningful scenes. The most common approach is to use Structure from Motion (SfM). We also detail a recent method called Robust Camera Pose Estimation [20] as it is used for obtaining camera poses in this work. Alternatively, acquiring camera poses by following a SLAM approach is possible.

3.1.1 Structure From Motion

The core idea of SfM lies in reconstructing the three-dimensional structure of a scene through the analysis of two-dimensional images. The process involves identifying distinctive features within each image and establishing correspondences between these features across multiple frames. By leveraging the changing perspectives captured in the image sequence, the relative positions and orientations of the cameras are estimated. Triangulation is then employed to compute the spatial coordinates of the identified features in three dimensions. Bundle adjustment, a crucial optimization step, refines the entire reconstruction by iteratively adjusting the 3D scene structure and camera poses to minimize discrepancies between predicted and observed feature locations. The ultimate goal is to obtain an accurate and detailed three-dimensional representation of the scene based on the information extracted from the sequence of two-dimensional images. The extracted camera trajectory is an essential input for Neural Radiance Fields. One problem that arises from using SfM poses is that those are often unreliable or incomplete for deformable scenes. This strongly impacts the performance of Neural Radiance Fields.

3.1.2 Robust Camera Pose Estimation

The Robust Camera Pose Estimation model [20] tries to estimate relative poses on a frame-to-frame level via a 2D residual function and a 3D residual function. The 2D residual targets to more accurately predict rigid motion, while the 3D residual is designed for non-rigid movement. Both residual functions consider depth maps and optical flow information and strike a balance between input information for the relative pose estimation. Additionally, weight maps are learned by utilizing a 3-layer U-Net architecture [57] and a sigmoid activation function to keep the output between 0 and 1. These weight maps aim for each image pixel to prioritize one of the two residual terms for relative pose estimation and thus focus more on non-rigid or rigid movement assumptions. The Robust Camera Pose Estimation model showcased superior performance for endoscopic scenes on the StereoMIS dataset [20] even for challenging scenes with significant deformations.

3.2 Neural Radiance Fields

The main body of this methodology revolves around Neural Radiance Fields, first introduced by Mildenhall et al. [44], as well as further developments with a specific focus on Dynamic Neural Radiance Fields. The main idea of Neural Radiance Fields is to generate novel views based on implicitly representing a continuous 3D scene via a 5D function. The function takes as input the spatial locations $\mathbf{x} = (x, y, z)$ and the 3D Cartesian viewing direction $\mathbf{d} = (\theta, \phi)$ indicating the direction from the camera position to the pixel location. The function outputs the emitted color $\mathbf{c} = (r, g, b)$ and volume density σ. The parameterization of the function is conducted via a Multi-Layer Perceptron (MLP) $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$. The function weights Θ are optimized to correctly predict the volume density and emitted color for each 5D input. The general pipeline of obtaining novel views from this 5D continuous function is to perform ray marching. The emitted colors and volume densities along each ray are then accumulated via the classical volume rendering [23]. The final accumulated color for the volume rendered ray is the corresponding pixel value in 2D. This procedure has to be repeated for all pixels in an image to visualize a novel view.

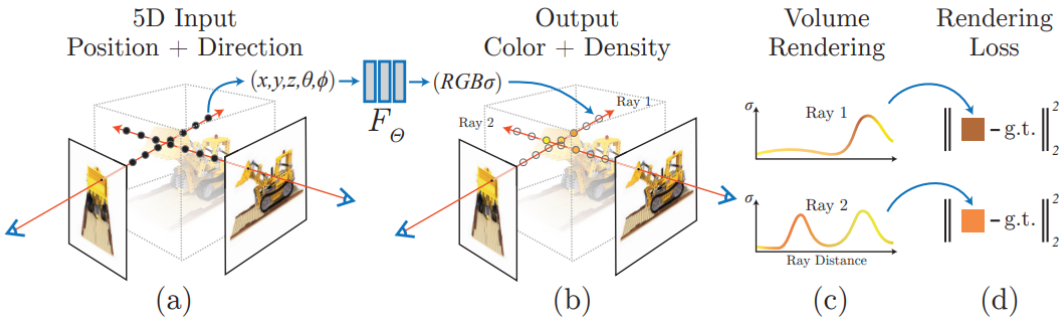


Figure 3.1: **Overview Of Neural Radiance Fields Pipeline [44]:** (**a**) The model takes spatial coordinates $\mathbf{x} = (x, y, z)$ and viewing direction $\mathbf{d} = (\theta, \phi)$ as input for each sample point; (**b**) These 5D inputs are then fed into a 6-layered Multi-Layer Perceptron F_Θ with skip-connections to generate the volume density . Simultaneously, the remaining part of the Multi-Layer Perceptron is combined with the viewing direction **d** through another Multi-Layer Perceptron to attain the emitted color **c**; (**c**) The differentiable volume rendering is applied to the model's output to receive pixel color values; (**d**) The model is optimized on these pixel color values by comparing it to the ground truth pixels. This figure is taken from [44].

3.2.1 Sampling

To obtain novel views, Neural Radiance Fields first perform ray marching. A ray $\mathbf{r}$ has the camera location as its origin $\mathbf{o} = (x, y, z)$ and a direction $\mathbf{d}$ which spans it from $\mathbf{o}$ to the desired pixel location. The ray is thus defined as $\mathbf{r}(t) = \mathbf{o} + \mathbf{d} \cdot t$, where t is the distance to the ray origin. However, in Neural Radiance Fields, t is constrained to be within the near plane t_n and the far plane t_f. Sample points are then uniformly generated along the ray and combined with their respective direction $\mathbf{d}$ as input to the model.

Scene Contraction

Additionally, scene contraction is performed on the actual 3D sample points, following the idea of Mip-NeRF 360 [4]. Scene contraction is primarily conducted because this work tries to model endoscopic scenes as unbounded. Scene contraction tries to perform an extended Kalman filter on the samples by separating them into near and far samples. The near samples are treated as in the original Neural Radiance Fields [44], and the far samples are sampled according to the inverse distance, meaning disparity. Sampling linearly in disparity allows for a more efficient sampling of the scene.

$$contract(\mathbf{x}) = \begin{cases} \mathbf{x}, & \text{if } \|\mathbf{x}\| \leq 1 \\ \left(\frac{2 \cdot \mathbf{x} - 1}{\|\mathbf{x}\|_2^2} \cdot \mathbf{x}\right), & \text{otherwise} \end{cases} \tag{3.1}$$

The first case in equation (3.1) represents the uncontracted near space and the second case covers the contracted far space for each ray.

3.2.2 Volumetric Rendering

After obtaining the sample points along the ray and their respective directional vectors using ray marching, the next step in the process is typically volume rendering. Volume rendering is the technique used to generate a 2D image from the information gathered along the ray. The exact formula for computing the expected color for ray $\mathbf{r}$ is given in (3.2).

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})\, dt, \quad \text{where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\, ds\right) \tag{3.2}$$

In Equation (3.2) function $T(t)$ describes the accumulated transmittance along the ray $\mathbf{r}$ from t_n to t. It indicates the probability of the ray to hit a surface in between t_n to t. Due to the numerical infeasibility of the continuous integral in (3.2), Neural Radiance Fields estimate the integral via quadrature. The idea is to use a stratified sampling approach in which a random sample is drawn for each bin of N evenly-spaced bins generated by partitioning $[t_n, t_f]$.

$$t_i \sim U\left[t_n + \frac{i-1}{N}(t_f - t_n), t_n + \frac{i}{N}(t_f - t_n)\right] \tag{3.3}$$

Stratified sampling allows for a continuous scene representation and avoids discretization, like in deterministic quadrature. The quadrature rule used in (3.4) is discussed by [41].

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp\left(-\sigma_i\delta_i\right))\, \mathbf{c}_i, \quad \text{where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j\delta_j\right) \tag{3.4}$$

Note that $\delta_i = t_{i+1} - t_i$ signifies the distance between two consecutive samples, while $\mathbf{c}_i$ and σ_i represent the emitted color and volume density for sample i. $\hat{C}(r)$ denotes a differentiable continuous integral that allows for the optimization of the parametrized 5D function.

3.2.3 Optimization

Following the generation of pixel values from 5D inputs, the subsequent stage involves optimizing the model with respect to the ground truth images. This optimization is accomplished by calculating a photometric loss, characterized as an L2 loss measuring the disparity between the predicted pixel values and the corresponding ground truth pixel values, as illustrated in Equation (3.5).

$$\mathcal{L}_{pho} = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r}\in\mathcal{R}} \|\hat{C}(\mathbf{r}) - C(\mathbf{r})\|_2^2 \tag{3.5}$$

In (3.5) $\mathcal{R}$ describes the set of rays in a batch. After computing the loss, the model is optimized via Gradient Descent, e.g., Stochastic Gradient Descent (SGD). This is possible since all functions contained in the forward pass of Neural Radiance Fields are differentiable. SGD first backpropagates the loss to the parameters to obtain their gradients and then stepwise adapts the weights in the negative direction of the gradient.

3.2.4 Depth Supervision

A typical problem of Neural Radiance Fields is predicting incorrect geometry when trained on insufficient or weakly overlapping images. This particular issue arises because the optimization becomes underconstrained, and the depth of hardly seen points in 3D is highly ambiguous. This, again, can also lead to artifacts in novel views due to the incorrect geometry. The network is simply overfitting to the training views. A way to handle this problem is by additionally constraining the density prediction of the model on depth, first introduced by [13], emphasizing a more precise geometry. This work will follow the estimated depth prediction of the Urban Neural Radiance Field [56].

For depth supervision, revisiting volume rendering from (3.2) is necessary for understanding how to acquire a depth prediction from the density prediction. Recall that $T(t)$ is the accumulated transmittance along a ray, t is the sampling distance, and $\sigma(\mathbf{r}(t))$ the density of a ray at length t. This part of the integral in (3.2) can be reinterpreted as the weighting of the emitted color prediction at a distance t along a ray, see (3.6).

$$w(\mathbf{r}, t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\, ds\right) \cdot \sigma(\mathbf{r}(t)) \tag{3.6}$$

Combining the weights $w(\mathbf{r}, t)$ with the distance t on ray $\mathbf{r}$ will yield the estimated depth value $\hat{z}(\mathbf{r})$ along ray $\mathbf{r}$.

$$\hat{z}(\mathbf{r}) = \int_{t_n}^{t_f} (w(\mathbf{r}, t) \cdot t) dt \tag{3.7}$$

The Neural Radiance Field can then be optimized via comparing to the Ground truth depth value $z(\mathbf{r})$ along ray $\mathbf{r}$ via computing the L2 loss.

$$\mathcal{L}_z = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \|\hat{z}(\mathbf{r}) - z(\mathbf{r})\|_2^2 \tag{3.8}$$

Additionally, this project adopts the idea of the line-of-sight loss from the Urban Radiance Fields [56], with the assumption that a pixel's color is almost exclusively determined by the surface tissue color. The idea is to provide two additional regularization losses besides the expected depth, called near loss and empty loss. The near loss encourages the density weights of an area around the surface, determined by the ground truth depth, to follow a Dirac distribution. Due to computational feasibility, the Dirac distribution is represented by a Kernel that sums up to 1.

$$\mathcal{L}_{near} = \mathbb{E}_{\mathbf{r} \sim \mathcal{D}} \left[\int_{z-\epsilon}^{z+\epsilon} \left(w(t) - \mathcal{K}_\epsilon(t - z) \right)^2 dt \right], \quad \text{where } \mathcal{K}_\epsilon(x) = \mathcal{N}(0, (\epsilon/3)^2) \tag{3.9}$$

In Equations (3.9) and (3.10) ϵ represents a noise value around the actual surface and thereby defining the surface thickness. This also helps to make the near loss more robust as small values for ϵ reduce the reconstruction quality. $\mathcal{K}_\epsilon$ is the kernel function, represented by a truncated normal distribution $\mathcal{N}$ with mean 0 and variance $(\epsilon/3)^2$. The empty loss encourages the density weights up to the previously defined surface area to be as small as possible since this space should be empty.

$$\mathcal{L}_{empty} = \mathbb{E}_{\mathbf{r} \sim \mathcal{D}} \left[\int_{t_n}^{z-\epsilon} \left(w(t) \right)^2 dt \right] \tag{3.10}$$

3.2.5 Positional Encoding

One essential contribution to the success of Neural Radiance Fields is Positional Encoding. Using (x, y, z, θ, ϕ) as the only input to the Multi-Layer Perceptron gives inferior results for Neural Radiance Fields even considering that Multi-Layer Perceptrons are universal function approximators [22]. Furthermore, Rahman et al. [54] have highlighted that deeper Neural Networks are biased towards learning lower frequency functions, meaning functions that vary globally without local fluctuations. Therefore, Neural Radiance Fields utilize Positional Encodings to map the input into higher dimensional space, similar to [67]. The Positional Encoding function is a high-frequency Fourier function. It is applied to each input coordinate, i.e., each sample

point $\mathbf{x} = (x, y, z)$ as well as the viewing direction $\mathbf{d}$ independently to perform the mapping (3.11):

$$\gamma(p) = (\sin(2^0\pi p), \cos(2^0\pi p), ..., \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p)) \tag{3.11}$$

Note L in (3.11) represents the used number of frequencies. The positionally encoded input coordinates are then similarly to before processed by the Neural Network.

3.3 Dynamic Neural Radiance Fields

In contrast to the initial Neural Radiance Fields [44], this project cannot rely on static scene assumptions. Introducing dynamic elements into a scene makes the task of Novel View Synthesis much more complicated, and acquiring multi-view scenes is non-trivial. This leads to sparse observations, which are firmly underconstrained. Therefore, sharing as much information across different time steps as possible is essential for high-quality reconstruction.

3.3.1 HexPlane

Our work builds upon several ideas in HexPlane [6]. The general concept in HexPlane explicitly represents a dynamic scene via a 4D feature grid. However, the naive approach would naturally scale by the power of four memory-wise and is, therefore, infeasible for long sequences. Because of that, one can represent a 3D volume, highly inspired by [8], $\mathbf{V} \in \mathbb{R}^{XxYxZxF}$ as a sum of vector-matrix multiplications, see (3.12).

$$\mathbf{V} = \sum_{r=1}^{R_1} \mathbf{M}_r^{XxY} \otimes \mathbf{v}_r^Z \otimes \mathbf{v}_r^1 + \sum_{r=1}^{R_2} \mathbf{M}_r^{XxZ} \otimes \mathbf{v}_r^Y \otimes \mathbf{v}_r^2 + \sum_{r=1}^{R_3} \mathbf{M}_r^{YxZ} \otimes \mathbf{v}_r^X \otimes \mathbf{v}_r^3 \tag{3.12}$$

In equation (3.12) $\otimes$ is the matrix outer product, $M_r^{XxY} \in \mathbb{R}^{XxY}$ is a matrix spanning the X and Y axes, and $\mathbf{M}_r^{XxY} \circ \mathbf{v}_r^Z \circ \mathbf{v}_r^1$ is a low-rank component of the volume $\mathbf{V}$. The vectors $\mathbf{v}^Z \in \mathbf{Z}$ and $\mathbf{v}^1 \in \mathbf{F}$ are along the $\mathbf{Z}$ and $\mathbf{F}$ dimensions, where the $\mathbf{F}$ axis is the feature axis of the 3D volume. The factorization approach from (3.12) helps reduce memory consumption but does not represent the temporal dimension. However,

decoupling temporal and spatial parts of the scene will not be capable of correctly representing a dynamic scene as time and space are strongly entangled. Henceforth, the final 4D volume used in HexPlane uses joint piecewise linear functions of the temporal axis **T** and any of the spatial axes $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ leading to equation (3.13).

$$\mathbf{V} = \sum_{r=1}^{R_1} \mathbf{M}_r^{XxY} \otimes \mathbf{M}_r^{ZxT} \otimes \mathbf{v}_r^1 + \sum_{r=1}^{R_2} \mathbf{M}_r^{XxZ} \otimes \mathbf{M}_r^{YxT} \otimes \mathbf{v}_r^2 + \sum_{r=1}^{R_3} \mathbf{M}_r^{YxZ} \otimes \mathbf{M}_r^{XxT} \otimes \mathbf{v}_r^3 \quad (3.13)$$

In HexPlane every $\mathbf{M}_r^{XxY}$ is a learnable feature plane. Furthermore, as can be seen from equation (3.13) and its architecture in Figure 3.2, it contains three pairs of feature planes. In the first step, the pair plane features are combined via multiplication after sampling points from these feature planes. In the second step, all combined pair plane features are concatenated in one big feature vector representing a sampling point along ray **r**. There are two 4D volumes, one comprising density features and the other containing color features. These features are then passed through subsequent 3-layered Multi-Layer Perceptrons, which return the emitted color **c** and volume density value σ for each sampling point $\mathbf{x} = (x, y, z)$ in space and for time **t** as well as for the corresponding viewing direction $\mathbf{d} = (\theta, \phi)$.

3.3.2 Local HexPlanes

Although HexPlane displays great results for deformable scenes, even with solid deformations, it is bounded in its capacity to represent large deforming scenes as the 4D volume increases rapidly with an increasing scene scale and image sequence length. Due to that reason and strongly inspired by LocalRF[43], this work separates scenes into local submodels capable of representing larger scenes locally at the exact spatial resolution instead of one big HexPlane globally for the entire scene while also being independent of the actual scene dimension. Furthermore, it allows for an infinite sequence length representation without sacrificing temporal grid resolution and at identical GPU memory consumption during training and inference. We call this approach "FLex", which stands for "Flow-Optimized Local HexPlanes". Overall, FLex possesses the advantage compared to HexPlane of representing scenes with moving cameras and/or large videos with constant GPU memory consumption and with a higher representational capability.

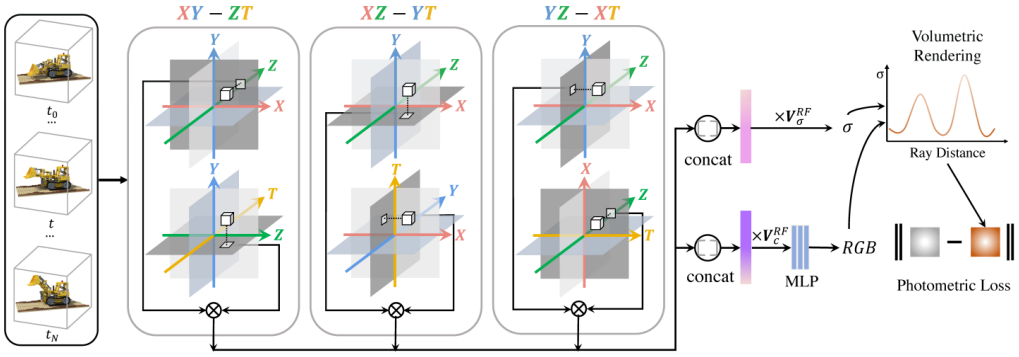


Figure 3.2: **HexPlane Architecture Overview [6]:** The model comprises six feature planes in total and is pre-split into three feature plane pairs. Each pair contains all four dimensions (**X**, **Y**, **Z**, **T**). Those feature plane pairs are combined via multiplication before concatenating with all other pairs. Different to this Figure is that both the color features as well as the density features are passed through 3-layered Multi-Layer Perceptrons to acquire the emitted color **c** and volume density σ. This Figure is taken from [6].

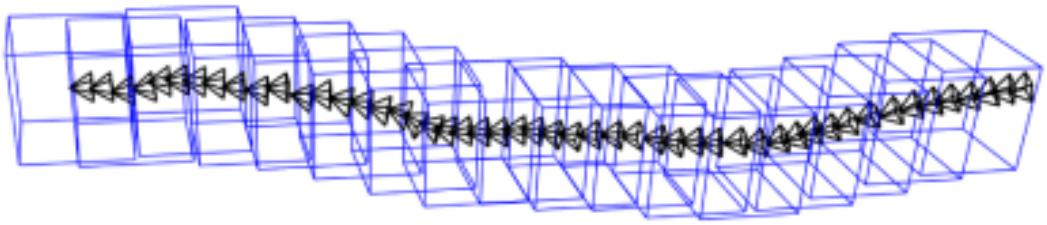


Figure 3.3: **FLex Scene Partitioning Overview:** The model progressively optimizes for both poses and reconstruction. Whenever the trajectory reaches a certain distance or exceeds its temporal dimension, a new local HexPlane model is generated. This Figure is taken from [43].

In FLex, new models are generated based on several aspects, the first being the number of training views. Since the image sequence length could be substantial, generating a new local model according to the number of frames to train on is crucial. Therefore we split the latest after every 100 frames. The second aspect is the camera trajectory's position change, meaning a new model is created whenever the camera trajectory moves far enough away from the current model's center.

$$D_{last}(i, j) = \|trans_i - center_j\|_2^2 \tag{3.14}$$

The computation of this distance measure can be seen in Equation (3.14). In this equation $trans_i$ represents the camera location at frame i, while $center_j$ represents model j's center location. Note that the camera poses are represented as the relative poses to the initial frame. Likewise, the center locations are expressed as the relative distance to the origin equivalent to the initial frame's camera pose location. The initial camera center for the first submodel is set to the scene origin, while the following submodel centers are located at the corresponding camera locations. The distance threshold for a new submodel generation is set to 1, which is chosen as it represents the threshold for a submodel leaving the uncontracted space. Scene contraction is more thoroughly explained in section 3.2.1. To ensure consistency across all neighboring submodels, an overlap region is required, and the newly generated submodel's center is at the last camera's location $trans_i$ of the previous model. Internally, each submodel is 0-centered, which means that during ray tracing, the global submodel's center is subtracted from the ray origin, which naturally shifts each sample point along this ray to be shifted accordingly. Additionally, to further strengthen consistency, blending weights are used to combine submodel outputs for overlapping frames. The blending weights increase linearly within the overlapping region the closer a frame is temporally to a model's center. Moreover, the temporal grid dimension is set to six initially for each local model, and the ultimate temporal resolution is configured to be half the number of encompassing frames. The precise number varies depending on the maximum number of frames the submodel covers. The adjustment for the final temporal resolution is made dynamically during training for each model to maintain a consistently accurate temporal resolution.

3.3.3 Optical Flow Supervision

Optical flow supervision is an additional regularization loss that can be utilized to improve scene geometry. Since optical flow supervision has proven to be helpful in

several Novel View Synthesis models like, e.g., NSFF [30] or LocalRF [43]. This project adapts LocalRF's approach for optical flow supervision.

Predicting the optical flow $\hat{\mathcal{F}}_{k\to k+1}$ from frame k to frame $k+1$ can be computed by predicting the difference between the new pixel location of a ray $\mathbf{r}$ from frame k in frame $k+1$. The approach is to utilize the predicted depth and a ray direction to find the surface point for a given pixel coordinate of frame k in 3D and then map this 3D point into the new pixel coordinate of frame $k+1$ using the relative camera pose and the camera intrinsics.

$$\hat{\mathcal{F}}_{k\to k+1}(\mathbf{r},\hat{z},R,trans) = \mathbf{p}(\mathbf{r}) - \pi_{2D}\left([R|trans]_{k\to k+1}\,\pi_{3D}(\mathbf{r},\hat{z})\right), \quad \text{where } \pi_{3D}(\mathbf{r},\hat{z}) = \mathbf{d}\cdot\hat{z} \tag{3.15}$$

$$\pi_{2D}(\mathbf{x}) = (\frac{x}{z}\cdot f_w + c_w, \frac{y}{z}\cdot f_h + c_h) \tag{3.16}$$

In equation (3.15) π_{2D} is the projection from a 3D coordinate back to the 2D pixel coordinate and π_{3D} displays the re-projection from the 2D pixel coordinate to the 3D coordinate given the predicted depth $\hat{z}$ and the ray direction $\mathbf{d}$ for ray $\mathbf{r}$. Furthermore, $\mathbf{p}(\mathbf{r})$ is the pixel coordinate of ray $\mathbf{r}$ and $[R|trans]_{k\to k+1}$ is the relative camera pose from frame k to $k+1$. f_w, f_h, c_w, and c_h are the focal point's dimensions along the image height and width, as well as the camera center along the width and height respectively in equation (3.16).

Similarly to $\hat{\mathcal{F}}_{k\to k+1}(\mathbf{r})$, meaning the forward optical flow, backward optical flow $\hat{\mathcal{F}}_{k\to k-1}(\mathbf{r})$ can also be computed. The total optical flow loss is computed by calculating the L1 difference between the ground truth optical flow and the predictions as can be seen in Equation (3.17) and (3.18) for forward optical flow and backward optical flow loss respectively.

$$\mathcal{L}_{fwd-opt} = \frac{1}{|\mathcal{R}|}\sum_{(\mathbf{r},\hat{z},R,trans)\in\mathcal{R}} \|\hat{\mathcal{F}}_{k\to k+1}(\mathbf{r},\hat{z},R,trans) - \mathcal{F}_{k\to k+1}(\mathbf{r})\|_1 \tag{3.17}$$

$$\mathcal{L}_{bwd-opt} = \frac{1}{|\mathcal{R}|}\sum_{(\mathbf{r},\hat{z},R,trans)\in\mathcal{R}} \|\hat{\mathcal{F}}_{k\to k-1}(\mathbf{r},\hat{z},R,trans) - \mathcal{F}_{k\to k-1}(\mathbf{r})\|_1 \tag{3.18}$$

$$\mathcal{L}_{total-opt} = \mathcal{L}_{fwd-opt} + \mathcal{L}_{bwd-opt} \tag{3.19}$$

3.3.4 Total Variational Loss

In order to ensure spatial-temporal consistency, we apply Total Variational Loss as in HexPlane [6] and TensoRF [8]. The Total Variational Loss measures the difference in features across neighboring feature plane locations, thus encouraging the feature grids to be similar in the direct neighborhood.

$$\mathcal{L}_{TV}(\lambda) = \frac{2}{|\mathcal{R}|} \sum_{i,j} \left(\frac{1}{D_1}(x_{i,j} - x_{i+1,j})^2 + \frac{\lambda}{D_2}(x_{i,j} - x_{i,j+1})^2 \right) \tag{3.20}$$

$$\mathcal{L}_{TV,s} = \mathcal{L}_{TV}(1) \tag{3.21}$$

$$\mathcal{L}_{TV,ts} = \mathcal{L}_{TV}(2) \tag{3.22}$$

$$\mathcal{L}_{TV,total} = \mathcal{L}_{TV,s} + \mathcal{L}_{TV,ts} \tag{3.23}$$

In Equation (3.20) x represents the features on the plane location i, j, D_1 and D_2 are the total number of differences along the first and second dimension of the feature plane, respectively. Equation (3.21) represents the Total Variational Loss for the spatial feature planes, and Equation (3.22) represents the spatial-temporal feature planes. Combining spatial and spatial-temporal variational losses yields the final Total Variational Loss in Equation (3.23).

3.4 Pose Optimization

As mentioned earlier, pre-processing of camera poses is required for most Neural Radiance Fields and, especially for dynamic scenes with moving cameras. These poses can be unreliable and thus hurt the reconstruction quality of Neural Radiance Fields. This problem can be addressed by optimizing jointly for poses and reconstruction. However, it is a challenging task in highly deformable scenes like many endoscopic scenes, and becomes even more difficult without prior poses, meaning predicting poses from scratch. Nevertheless, this work aims to tackle precisely this problem by using progressive optimization similar to LocalRF [43]. Subsequently, it is vital to use local

models and not train one big HexPlane model alone since wrong poses locally can otherwise affect the global reconstruction capability. The idea of progressive optimization is to initialize the first local model with a handful of frames and progressively append new frames until one of the two conditions from section 3.3.2 is reached. Every other model is initialized using the frames within the overlap region shared with the previous model and then following the same principle by appending new frames. Without progressive optimization, training poses and reconstruction spread out the poses significantly, lacking coherence and often getting poses stuck in local minima. This problem is neglectable for small amounts of frames but too significant for larger endoscopic scenes.

The first handful of poses are initialized with the identity matrix. Following views are initialized to be identical as the prior pose. This sustains a local prior and helps the trajectory consistency. Furthermore, each camera extrinsic is represented as the relative pose to the first pose location. Additionally, the rotations are represented as 6D poses as suggested in Zhou et al. [82] for their continuous representation of the rotations in contrast to Euler angles or quaternions. Having a continuous representation is beneficial for optimization, and furthermore, it reduces the rotation to a $3x2$ matrix and thus shrinks its parameter number, making the optimization slightly easier. A crucial step for maintaining a consistent trajectory is the Local-to-Global optimization approach. The local optimization takes place during the progressive optimization of new frames and ensures that the latest appended poses are optimized. During the global optimization frames from the entire trajectory of the current submodel are selected at random. Not performing the local optimization initially leads to incoherence and poses being stuck in local minima. The poses are only optimized based on the optical flow loss in contrast to LocalRF [43]. Using only optical flow improves the trajectory quality, subsequently leading to an improved reconstruction. Furthermore, the optical flow loss is decreased over the course of the training to ensure a decrease in pose changes per iteration. This ensures an earlier pose convergence and enables the model to fully concentrate on the scene reconstruction task.

4 Experiments & Results

Contents

4.1 Data

4.1.1 StereoMIS Dataset

The evaluation of our method and several baselines is performed on the in-vivo StereoMIS dataset [20]. StereoMIS was released in early 2023 and consists of three porcine (P1, P2, and P3) and three human subjects (H1, H2, and H3) with a total of 16 endoscopic video sequences. However, only the three porcine scenes are publicly available. The videos were generated via a da Vinci Xi surgical robot, and ground-truth camera trajectories were computed from the endoscope forward kinematics and synchronized with the video images. Additionally, the dataset contains tool masks for surgical instruments, which were obtained using a trained DeepLabv3+ [9] on the EndoVis2018 dataset [2]. The video lengths within StereoMIS range from 50 seconds to 30 minutes and are named according to the following pattern P_{x_y}, where x stands for the subject number and y for the sequence.

Given that most Neural Radiance Fields models face challenges with processing lengthy 30-minute videos, we further segment these sequences in our approach, adopting an extended pattern: $P_{x\_y\_z}$, where z represents the numbered subsection within sequence y. Instead of relying on ground-truth poses, we pre-process input poses using the Robust Camera Pose Estimation method [20], the same method that introduced the StereoMIS dataset. This choice is motivated by the impracticality of knowing camera trajectories in real-world scenarios, necessitating an alternative method for pose acquisition. Ground-truth trajectories are exclusively used to assess the correctness of predicted poses. To prevent reliance on overfitted poses from the trained Robust Camera Pose Estimation model, we exclude P1, a part of its training set. Consequently, all evaluations are presented solely for the P2 subject sequences.

To introduce regularization in optical flow and depth, we employ the RAFT method [62], which is trained on the FlyingThings 3D dataset [42]. Optical flow regularization utilizes all adjacent frames for computation, while depth is obtained using the left and right stereo images as inputs to RAFT. All scenes used in the evaluation are separated into three categories: (1) static camera and deforming scenes, including breathing motion and pulling or cutting tool deformations (2) almost no scene deformations under a moving camera (3) a moving camera, while displaying deforming content also including breathing motions and tool caused deformations.

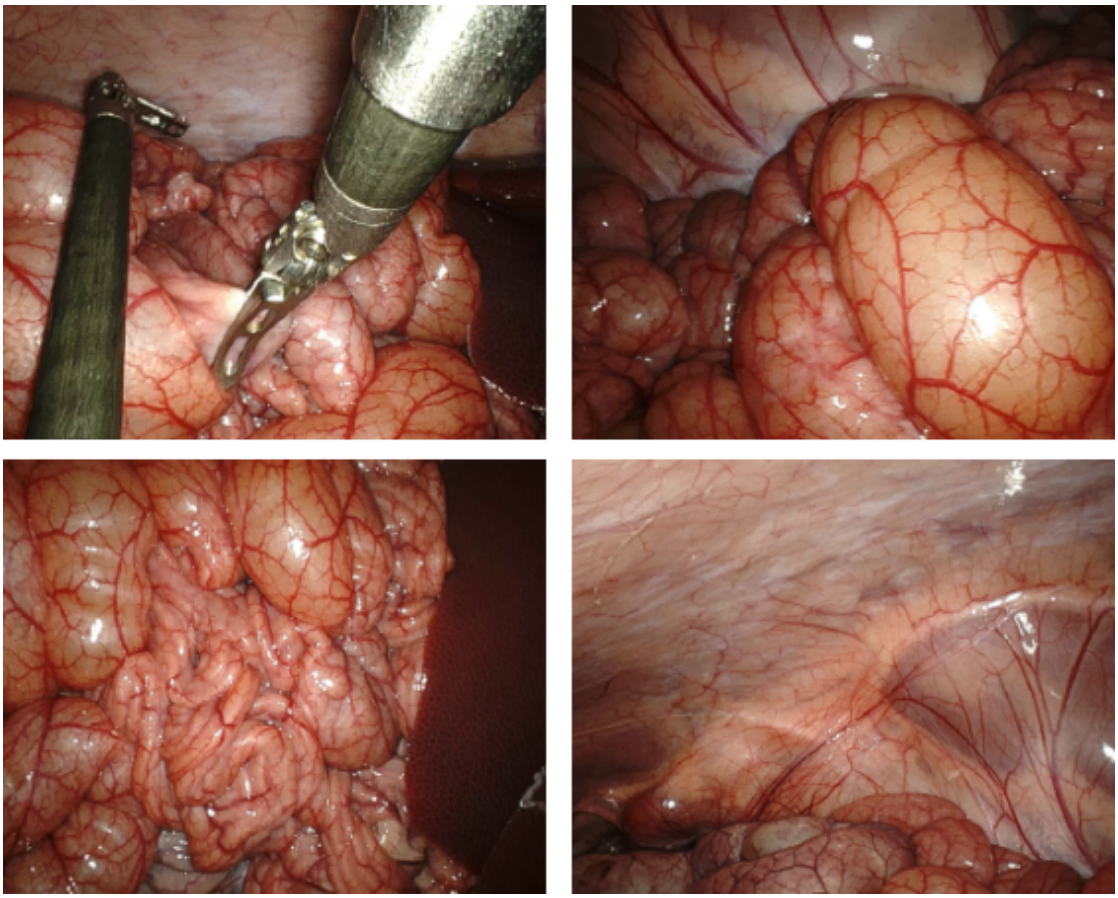


Figure 4.1: **StereoMIS Image Examples:** This Figure shows 4 different images from the P2 subject. All images are also contained in the scenes used for the evaluation in section 4.

4.2 Evaluation Metrics

The following standard metrics for Novel View Synthesis are used to evaluate and compare our method with other baselines: PSNR, SSIM [72], and LPIPS [80].

Peak-to-noise ratio (PSNR) indicates the ratio between the predicted image's maximal noise and the maximal achievable noise. It is used to quantify the perceptual quality of the predicted image. Its precise formulation can be seen in Equation (4.1), where $\mathcal{L}_{pho,i}$ stands for the average photometric loss between the predicted image and the ground-truth image for frame i. Note $\mathcal{L}_{pho}$ is explained in more detail in Equation (3.5) in section 3.2.3.

$$PSNR(i) = -10 \cdot \frac{\log(\mathcal{L}_{pho,i})}{\log(10)} \tag{4.1}$$

SSIM stands for Structure Similarity Index Method [72] and measures the similarity between the ground-truth image and the predicted one by considering image degradation as the change of perception in structural information.

$$SSIM(\hat{C}(\mathbf{r}), C(\mathbf{r})) = \left[l\left(\hat{C}(\mathbf{r}), C(\mathbf{r})\right)\right]^{\alpha} \cdot \left[c\left(\hat{C}(\mathbf{r}), C(\mathbf{r})\right)\right]^{\beta} \cdot \left[s\left(\hat{C}(\mathbf{r}), C(\mathbf{r})\right)\right]^{\gamma} \tag{4.2}$$

In Equation (4.2) $l\left(\hat{C}(\mathbf{r}), C(\mathbf{r})\right)$ indicates the luminance term, computing the difference in brightness between the two images. $c\left(\hat{C}(\mathbf{r}), C(\mathbf{r})\right)$ symbolizes the contrast term, which measures the differences between the most extreme value (brightest and darkest) differences within the images. The final term of the equation, $s(\hat{C}(\mathbf{r}), C\left(\mathbf{r}\right))$ is the structure term, measuring the similarities for the local luminance patterns across the images. In our case α, β, and γ are set to 1 each. More details to Equation (4.2) can be found in the original paper [72].

The last visual metric is Learned Perceptual Image Patch Similarity (LPIPS) [80], which measures the similarity between two images using Deep Neural Network features. The Neural Network computes image features for both the predicted and ground-truth image and compares the features after normalization by calculating the distance in feature space. For the following experiments, both Alex Net [26] and VGG [59] are utilized for computing the LPIPS metric.

Additionally, we use the L1 Distance for measuring 3D reconstruction quality. The L1 Distance measures the distance between the predicted depth and the ground-truth depth. The precise equation is displayed in Equation (4.3).

$$D_{L1} = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \|\hat{z}(\mathbf{r}) - z(\mathbf{r})\|_1 \tag{4.3}$$

We not only assess the perceptual quality of the generated images but also conduct an evaluation of our method based on the predicted poses. Three metrics, as outlined in [49], are employed to gauge the accuracy of the predicted poses. The first is the absolute trajectory error (ATE-RMSE) to evaluate the overall trajectory shape. The ATE-RMSE is drift-sensitive and impacted by the sequence length. The other two metrics are the relative pose errors (RPE-Trans and RPE-Rot), employed to assess the relative pose error from frame to frame.

4.3 Training Details

In the following experiment results, we train all methods for the same number of ray samples, equivalent to 100,000 iterations, on a batch size of 4,096 rays. All baseline methods follow their initial settings, and if a hyperparameter is dependent on the max iteration size, then it is linearly adjusted to the change of max iterations. The Feature grid dimensions for both LerPlane [76] and LocalRF [43] are adjusted to match those of HexPlane and our method to ensure a fair comparison. The following abbreviation (HexPlane^{†_2}) stands for the optimized version of HexPlane, including scene contraction as in Mip-NeRF 360 [4], depth loss supervision, and optical flow supervision. HexPlane^{†_1} represents an optimized version of HexPlane that only includes scene contraction but no additional scene regularization.

FLex without pose optimization is implemented on Pytorch 12.1.1 on CUDA 12.1 and is built on top of the Pytorch implementation of HexPlane [6] with several Adam optimizers [25]. The learning rate for the feature grids starts initially at $2 \cdot 10^{-2}$, while the Multi-Layer Perceptrons learning rates are initially set to $1 \cdot 10^{-3}$. All learning rates follow an exponential learning rate decay with a final value of $\frac{1}{10}$th of the initial learning rate. The dimension of the spatial feature grids is 512 for (x, y, z), and the

temporal dimension is 1/2 the actual image sequence covered by the model. The feature dimension is 72 in total for both density and color. Additionally, we adopt a coarse-to-fine approach as in HexPlane [6] to start with a lower grid resolution and increase over time to the settings mentioned above. The Multi-Layer Perceptron for the density consists of one hidden layer and ReLU [1] non-linear activation functions. The input to this Multi-Layer Perceptron comprises only the density features obtained from the feature grid, and additionally, positional encoding is applied on those beforehand. The Multi-Layer Perceptron for the color also contains only one hidden layer with ReLU [1] non-linear activation functions and receives positionally encoded color features and positionally encoded viewing directions. All positional encodings use a maximal frequency of 2. The final loss is depicted in Equation (4.4). Note that the optical flow loss $\mathcal{L}_{total-opt}$ is decreased discretely by $\frac{1}{10}$ after every 10% of the total training.

$$\mathcal{L}_{total} = \mathcal{L}_{pho} + 0.01 \cdot \left(\mathcal{L}_z + \mathcal{L}_{near} + \mathcal{L}_{empty}\right) + 1.0 \cdot \mathcal{L}_{total-opt} + 0.0001 \cdot \mathcal{L}_{TV,total} \quad (4.4)$$

For FLex with pose optimization from scratch, the identical settings as our other method are used. Additionally, the starting learning rate for the pose rotation parameters is at $5 \cdot 10^{-3}$, while the learning rate for the pose translation parameters is set to $5 \cdot 10^{-4}$. Both learning rates are also decreased exponentially to a final value of $\frac{1}{10}$th of the initial learning rate. Similar to LocalRF [43], our method with pose optimization contains a progressive optimization scheme in which 100 iterations are conducted before appending a new frame, with the first model being initialized with five frames and every other with the overlap region fixed at 30 frames. After the progressive optimization, each submodel is fine-tuned for 100 iterations per covering frame. The final loss term used for optimization is also described in Equation 4.4. The optical flow supervision is likewise discretely decreased after the progressive optimization is complete by $\frac{1}{10}$ after every 10% of the total fine-tuning steps.

4.4 Results

All results in the following sections are conducted on five scenes from the StereoMIS dataset for the P2 subject. All scenes contain 1000 frames, from which every eighth frame, starting with the first frame, is part of the test set. Every other frame not in the test set is used for training the models. The following quantitative and qualitative

results are performed on the test frames, except that the pose metrics are computed over the entire 1000-element trajectory. Additionally, we first apply the Umeyama algorithm [66] on the predicted poses to align the predicted poses to the ground-truth poses and ensure that both trajectories are in the identical coordinate system to conduct a proper comparison. Unlike in EndoNeRF [70], EndoSurf [78], and LerPlane [76], we are not using tool masks for removing surgical tools from the scenes and, for fairness, do not utilize tool masks for any method including the baseline models shown in this section. The main reason is that the necessity of removing tools from surgical scenes strongly depends on the specific use case at hand and might not always be desired. Additionally, we observe that tool masks within the StereoMIS dataset are only partially correct, leading to many artifacts within each model, including the baselines. Thus making the use of those tool masks obsolete, and we do not want to depend on hand-labelled tool masks.

Both scenes P2_8_1 and P2_7_1 contain camera movement and breathing motion. However, more extreme motion is displayed in scene P2_8_1. Scene P2_8_2 shows camera motion but hardly any scene motion and can be considered almost static. The last two scenes, P2_7_1 and P2_6_1 contain hardly any camera motion but, therefore, extreme tissue deformations caused by pulling operations from surgical tools.

Based on both Table 4.1, Table 4.2, and Figure 4.2, our final methods are capable of producing high-quality reconstruction even for extreme deformations, e.g., caused by pulling tissue with surgical tools. Our method without camera pose optimization outperforms all baselines, including the prior Endoscopic Neural Radiance Fields, by a significant margin. Our method with pose optimization achieves slightly worse results but still outperforms the endoscopic baseline models. It is only inferior in its reconstruction quality to ours with pre-processed poses and the optimized HexPlane baseline with pre-processed poses on average. The endoscopic baseline models struggle especially with more extreme camera movements, as shown in Figure 4.2, where the first image example is during such a camera motion.

4.4.1 Ablation Studies

Table 4.3 showcases several ablation studies conducted on HexPlane and on our method. Although the visual quality is highest for HexPlane without any geometrical regularization loss, the 3D reconstruction suffers immensely. As geometric consistency is essential

Model	Scene	PSNR ↑	SSIM ↑	LPIPS_a ↓	LPIPS_v ↓	L1 Distance ↓
EndoNeRF	P2_8_1	25,28	0,6279	0,5065	0,4996	–
	P2_8_2	11,22	0,5460	0,5165	0,5726	–
	P2_7_1	23,87	0,5689	0,5006	0,5166	–
	P2_7_2	26,56	0,6360	0,4094	0,4558	–
	P2_6_1	23,03	0,5694	0,5448	0,5241	–
EndoSurf	P2_8_1	25,14	0,6189	0,5157	0,5330	1,1432
	P2_8_2	29,81	0,7664	0,5130	0,5517	1,8945
	P2_7_1	23,42	0,5818	0,4927	0,5045	14,0252
	P2_7_2	25,39	0,6085	0,4821	0,5051	17,1465
	P2_6_1	22,14	0,5328	0,6179	0,5670	6,3175
LerPlane	P2_8_1	29,47	0,7661	0,1821	0,2916	8,1750
	P2_8_2	36,17	0,9002	0,1389	0,2725	20,8750
	P2_7_1	27,28	0,7102	0,2718	0,3551	30,7017
	P2_7_2	32,70	0,8496	0,1489	0,2139	30,3250
	P2_6_1	26,11	0,6900	0,2956	0,3730	28,5105
LocalRF	P2_8_1	29,02	0,8182	0,1768	0,2329	3,9256
	P2_8_2	35,07	0,8946	0,1721	0,2353	2,7013
	P2_7_1	22,54	0,7010	0,3177	0,3523	6,4239
	P2_7_2	31,22	0,8405	0,1655	0,2093	4,4184
	P2_6_1	19,21	0,6486	0,3940	0,4093	5,4113
HexPlane^{†_2}	P2_8_1	31,64	0,8567	0,1640	0,2283	1,1173
	P2_8_2	37,10	0,9166	0,1771	0,2511	2,4287
	P2_7_1	27,17	0,7471	0,2582	0,3155	1,3583
	P2_7_2	33,14	0,8739	0,1619	0,2060	1,0792
	P2_6_1	25,91	0,7103	0,2883	0,3588	1,5026
Ours w/o. Pose Optim.	P2_8_1	32,26	0,8842	0,1418	0,1836	0,9529
	P2_8_2	37,42	0,9226	0,1605	0,2263	1,0130
	P2_7_1	27,24	0,7734	0,2744	0,2161	1,5426
	P2_7_2	33,84	0,8908	0,1514	0,1823	1,0187
	P2_6_1	25,66	0,7240	0,2773	0,3171	1,6905
Ours w. Pose Optim.	P2_8_1	31,35	0,8491	0,1460	0,2037	6,2794
	P2_8_2	35,28	0,8868	0,1920	0,2727	7,2101
	P2_7_1	27,85	0,7929	0,2002	0,2573	19,1607
	P2_7_2	31,42	0,7983	0,1711	0,2317	12,3202
	P2_6_1	26,24	0,7365	0,2424	0,3099	16,1940

Table 4.1: **Quantitative Comparisons on StereoMIS dataset:** Blue indicates the best result and red the second best, respectively.

Model	PSNR ↑	SSIM ↑	LPIPS_a ↓	LPIPS_v ↓	L1 Distance ↓
EndoNeRF	21,99	0,5896	0,4956	0,5137	–
EndoSurf	25,18	0,6217	0,5277	0,5288	8,1054
LerPlane	30,35	0,7832	0,2075	0,3012	23,7174
LocalRF	27,41	0,7806	0,2452	0,2878	4,5761
HexPlane^{†_2}	30,992	0.8209	0.2099	0.2719	1,4972
Ours w/o. Pose Optim.	**31,28**	**0,8390**	0,1946	**0,2367**	**1,2435**
Ours w. Pose Optim.	30,43	0,8127	**0,1903**	0,2551	12,2329

Table 4.2: **Average Quantitative Comparisons on StereoMIS dataset:** Average results for results shown in Table 4.1. HexPlane^{†_2} is the vanilla HexPlane with scene contraction, depth loss, and optical flow loss. L1 Distance is measured in mm. The best result is marked in bold.

Model	PSNR ↑	SSIM↑	LPIPS_a↓	LPIPS_v↓	L1 Distance ↓
HexPlane^{†_1}	**31,98**	**0,8639**	**0,1580**	**0,2175**	104,2718
HexPlane^{†_1} + $\mathcal{L}_z$	31,63	0,8571	0,1640	0,2276	1,1418
HexPlane^{†_1} + $\mathcal{L}_z$ + $\mathcal{L}_{opt}$	31,64	0,8567	0,1640	0,2283	**1,1173**
(Ours) FLex +$\mathcal{L}_z$	32,25	**0,8843**	**0,1411**	**0,1831**	0,9752
(Ours) FLex +$\mathcal{L}_z$ + $\mathcal{L}_{opt}$	**32,26**	**0,8843**	0,1418	0,1836	**0,9529**

Table 4.3: **Geometric Regularization Ablation Studies On StereoMIS dataset:** All ablations in this table are conducted on scene P2_8_1. HexPlane^{†_1} is the optimized vanilla HexPlane meaning the use of scene contraction. HexPlane^{†_1} + $\mathcal{L}_z$ represents the vanilla HexPlane^{†_1} + depth loss, while HexPlane^{†_1} $\mathcal{L}_z$ + $\mathcal{L}_{opt}$ also includes an optical flow loss. FLex + $\mathcal{L}_z$ is FLex + depth loss, and our final method also includes optical flow loss. All methods use the predicted poses from the Robust Camera Pose Estimation model [20]. The best result for each metric is marked in bold separately for the HexPlane studies in the upper rows and the FLex results in the lower rows. The L1 Distance is measured in mm.

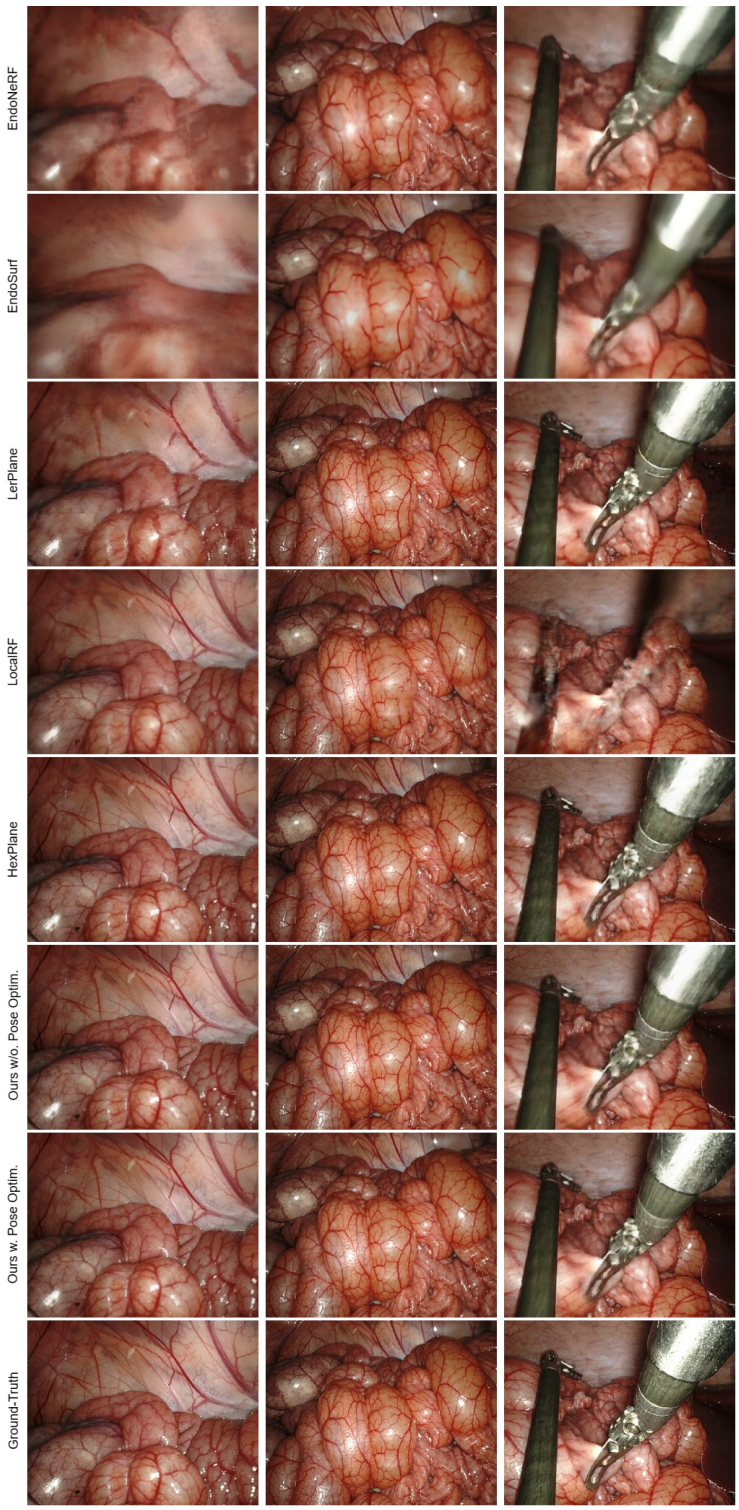


Figure 4.2: **Qualitative Results on StereoMIS dataset**

Model	Scene	ATE-RMSE ↓	RPE-Trans ↓	RPE-Rot ↓
Robust Pose Estimation	P2_8_1	2,334	**0,0746**	**0,0007**
	P2_8_2	**2,638**	**0,0803**	**0,0009**
	P2_7_1	34,597	**0,8512**	**0,0069**
	P2_7_2	**1,444**	**0,0708**	**0,0006**
	P2_6_1	33,804	**0,7246**	**0,0054**
	Average	**14,963**	**0,3603**	**0,0029**
LocalRF	P2_8_1	8,206	0,1543	0,0024
	P2_8_2	8,879	0,1979	0,0025
	P2_7_1	34,665	0,8527	0,0071
	P2_7_2	6,016	0,1285	0,0014
	P2_6_1	**29,911**	0,8443	0,0065
	Average	17,533	0,4355	0,0040
Ours w. Pose Optim.	P2_8_1	**2,328**	0,1038	0,0016
	P2_8_2	3,370	0,1182	0,0022
	P2_7_1	**33,511**	0,9989	0,0070
	P2_7_2	3,178	0,1852	0,0020
	P2_6_1	33,36	0,7809	0,0058
	Average	15,149	0,4374	0,0037

Table 4.4: **Quantitative Results for Poses on StereoMIS dataset:** Results for optimizing poses from scratch for the identical scenes as in Table 4.1 and Table 4.2. The ATE-RMSE and the RPE-Trans are in mm, and the RPE-Rot is in degrees.

for Novel View Synthesis robustness, we proceed using depth and optical flow loss. It can also be observed that for both HexPlane and FLex, using only depth supervision is equally favorable than using depth and optical flow supervision in terms of visual quality. However, we decided to proceed with both supervision losses as optical flow supervision is crucial for optimizing camera poses in our approach, and we want to maintain comparable results. Additionally, it appears that adding optical flow loss supervision improves 3D reconstruction.

4.4.2 Pose Results

With pose optimization from scratch, our method showcases high-quality results and outperforms LocalRF [43] by a great margin, as seen in Table 4.1 and Table 4.2 visually.

This is expected as LocalRF does not model for deformations and thus is limited in reconstructing endoscopic scenes. Furthermore, when comparing the actual trajectories of both LocalRF and our method with the ground-truth camera poses, we can observe that ours are also better than LocalRF's. This statement is not only correct for strongly deformable scenes but for scenes with nearly static content, as in scene P2_8_2 as well, underlining not only the improvement for modeling deformable scenes more accurately but, in general, providing improved camera trajectories for all scenarios. Figure 4.6 supports the statement by showcasing that our predicted camera poses are more similar to ground-truth poses. Our predicted poses are also more coherent than LocalRF, where one can observe several bigger incorrect jumps within the trajectory. Our method is, however, still a bit worse than the Robust Pose Estimation model [20].

Furthermore, we observe that training with non-perfect tool masks, as contained in the StereoMIS dataset, leads to poor predicted camera poses, decreasing reconstruction quality significantly, as shown in Figure 4.3. This is because the masks do not cover all the tools, which lets the models learn random artifacts for partial tools and prevents the learning of correct camera poses. Additionally, it is crucial to follow a Local-To-Global Optimization approach when optimizing poses from scratch to acquire coherent trajectories and high-quality reconstruction. This phenomenon can be observed in Figure 4.4, where the image quality difference is immense. The reasons for this are derivable from Figure 4.5, which shows that utilizing a global optimization approach leads to a big spread of the camera poses across the scene and, thus, an incoherent trajectory.

According to Figure 4.5, using only optical flow supervision for pose optimization yields better camera poses and thus results in a higher reconstruction quality. This result aligns with our hypothesis that using only optical flow supervision improves camera pose optimization, postulated in section 3.4. It is noteworthy that even though the optical flow supervision assumes no scene dynamics, the camera trajectories achieve great results. We believe this is because, in most cases, the camera motion is stronger than the scene deformation, leading to a more significant impact on the optical flow loss than the actual deformations. However, this assumption does not hold for scenes P2_7_1 and P2_6_1, leading to higher errors and more incorrect camera trajectories in Table 4.4.

Figure 4.3: **Pose Optimization With And Without Tool Mask:**

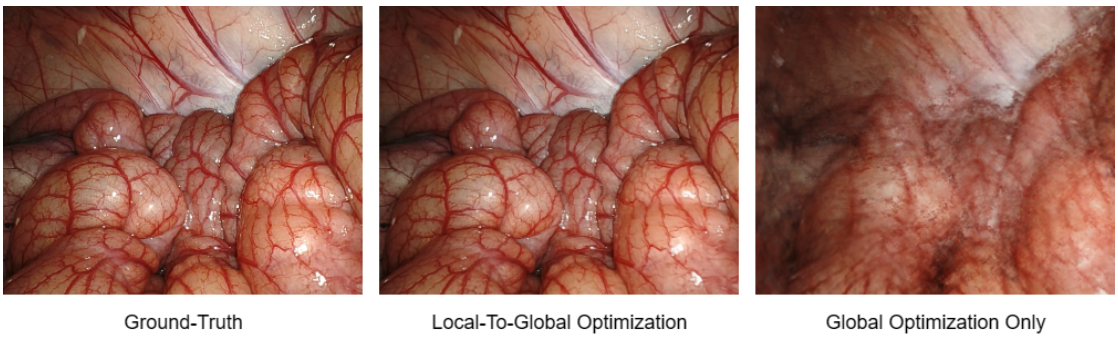


Figure 4.4: **Different Pose Optimization Strategies:** Not optimizing first locally leads to an incoherent trajectory, resulting in poor reconstruction quality.

Model	PSNR ↑	SSIM↑	LPIPS_a↓	LPIPS_v↓	RTE-RMSE↓	RPE-Trans ↓	RPE-Rot ↓
All Losses	31,19	0,8433	0,1590	0,2152	2,705	0,1052	0,001639
Ours	**31,35**	**0,8491**	**0,1416**	**0,2037**	**2,328**	**0,1038**	**0,001616**

Table 4.5: **Pose Optimization Ablation Studies On StereoMIS dataset:** All ablations in this table are conducted on scene P2_8_1 and trained with no prior pose knowledge, starting the pose optimization from scratch. The best result for each metric is marked in bold.

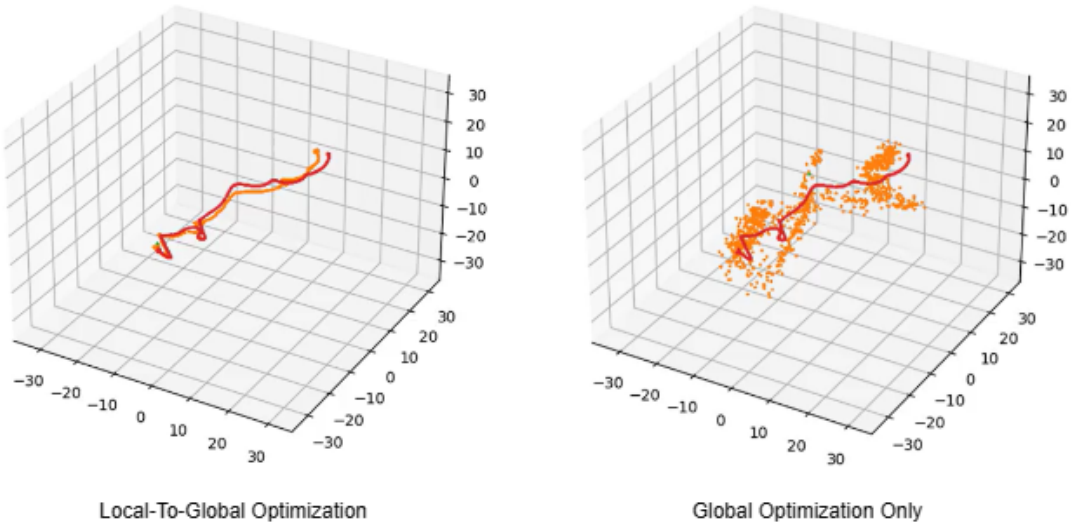


Figure 4.5: **Different Pose Optimization Strategies:** The red trajectory is the ground truth trajectory for the entire scene. Orange showcases the predicted trajectory. Without the Local Optimization step, poses tend to spread out and showcase incoherent behavior.

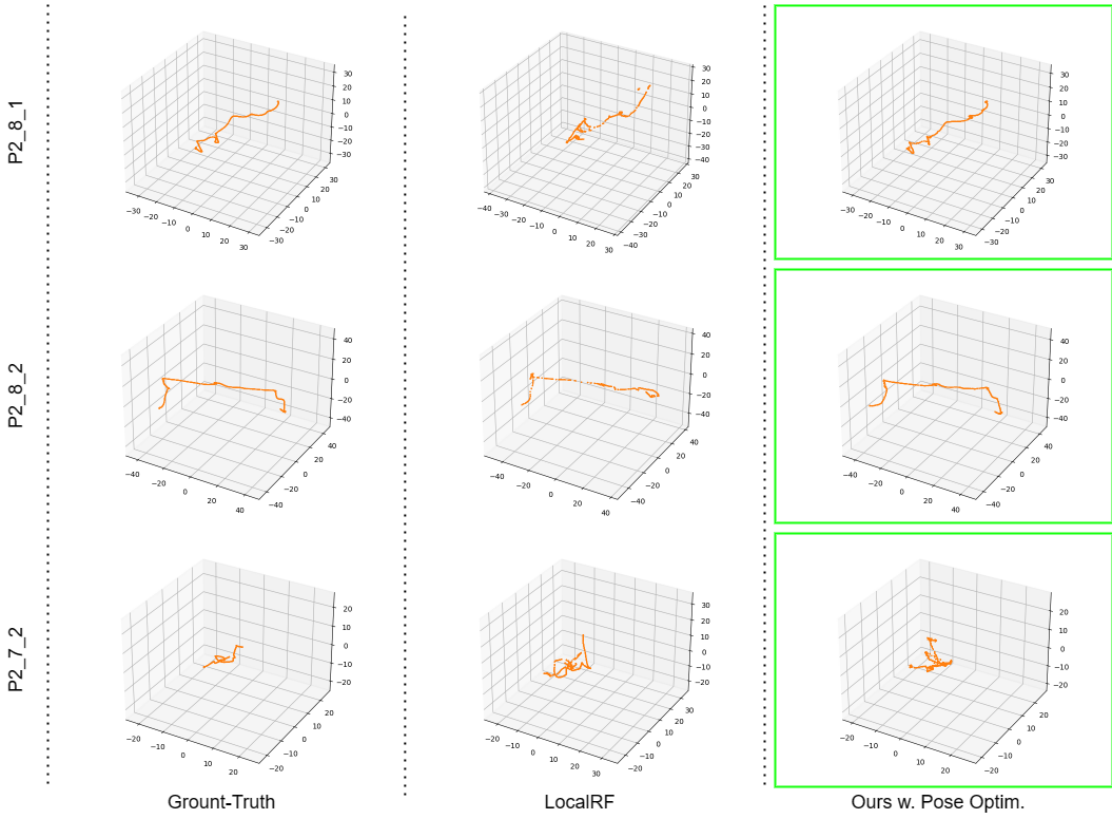


Figure 4.6: **Qualitative Pose Results:** The green bounding box around a predicted trajectory highlights the best trajectory quantitatively for the corresponding scene according to the ATE-RMSE, RPE-Trans, and RPE-Rot errors.

5 Discussion

Contents

5.1 Limitations & Future Work

As shown in the results, our work clearly improves upon common methods. However, as in any work, nothing is perfect, and there are some limitations that we will address in this section.

FLex has a running time issue since it requires approximately 4-5 hours for a 1000-frame scene. The running time issue is even more extreme for our method that optimizes for reconstruction and poses from scratch, which requires around 13-15 hours for a 1000-frame scene. The model, in some cases, generates blurry parts within the rendered images for highly deformable scenes, e.g., when deforming tissue via surgical tools. Both problems can be addressed by improving sampling efficiency. A possible approach is to use a proposal network as in Mip-NeRF 360 [4], which predicts tighter sampling range bounds for each ray, helping the model to concentrate its representational capacity on the most relevant section around the surface tissue. Another improvement for dynamic scenes is using ray importance weights similar to DyNeRF [28] to sample more often from regions with the most substantial deformations. This allows the model to emphasize learning the dynamics behind the most challenging regions within the scene instead of a uniform sampling approach as deployed in this work. Combining both ideas could significantly improve convergence, thus reaching an equivalent performance in much fewer total iterations or improving the overall reconstruction quality in an equivalent amount of time.

Another potential weakness of our current modeling approach is the incorrect tool masking system since parts of the tool are reconstructed for an incorrect/partially correct tool mask while others are not. This type of modeling is meaningless as it neither removes tools entirely from the reconstruction nor models them correctly, generating artifacts within the depth and optical flow supervision that lead to artifacts in the reconstruction. Therefore, we avoided modeling masking out tools and included them in our reconstructions. Suppose one wants to model without tool masks for the StereoMIS dataset or any newly observed surgical scenes. In that case, much-improved tool masks should be obtained via a new state-of-the-art method.

Furthermore, the predicted pose trajectory can be improved, especially when strong deformations are contained in the scene. These stronger deformations lead to pose optimization ambiguities between camera movement and object deformation. A simple solution would be to predict a mask using a state-of-the-art model like Mask R-CNN [21] to separate approximately static regions from highly deformable ones. Alternatively, one could follow the approach of the Robust Dynamic NeRF [34] by combining Mask R-CNN's output with a Sampson binary motion mask obtained by thresholding the Sampson distance for the fundamental matrix that was generated from optical flow. Applying this segmentation could reduce ambiguities and strictly improve camera trajectory convergence, yielding better reconstruction results as well. Alternatively, one could enforce a more robust geometric regularization on the density weights to concentrate more on the surface area, resulting in higher weight peak values and fewer variant weight distributions for time-invariant regions. This information could let the model internally separate between time-invariant and time-variant parts within the scene in a self-supervised fashion without adding another pre-processing step.

5.2 Conclusion

This project introduces an innovative approach for Novel View Synthesis in infinitely long and highly deformable surgical scenes, eliminating the need for pre-processed camera poses. The method partitions extensive 3D scenes into localized submodels by leveraging recent advancements in Neural Radiance Fields. This ensures robust representation irrespective of camera movements and sequence lengths. A progressive training scheme is implemented to jointly optimize scene reconstruction and camera poses from scratch, making pose pre-processing unnecessary. With pre-processed poses, the proposed method attains state-of-the-art results on the StereoMIS dataset, surpassing previous endoscopic Neural Radiance Fields. Although the method without pre-processed poses achieves slightly lower results, it still produces high-quality reconstructions with reasonable camera trajectories, outperforming previous methods even with pre-processed poses.

List of Figures

List of Tables

Bibliography

[1] A. Agarap. "Deep learning using rectified linear units (relu)." In: *arXiv preprint arXiv:1803.08375* (2018).

[2] M. Allan, S. Kondo, S. Bodenstedt, S. Leger, R. Kadkhodamohammadi, I. Luengo, F. Fuentes, E. Flouty, A. Mohammed, M. Pedersen, A. Kori, V. Alex, G. Krishnamurthi, D. Rauber, R. Mendel, C. Palm, S. Bano, G. Saibro, C.-S. Shih, H.-A. Chiang, J. Zhuang, J. Yang, V. Iglovikov, A. Dobrenkii, M. Reddiboina, A. Reddy, X. Liu, C. Gao, M. Unberath, M. Kim, C. Kim, C. Kim, H. Kim, G. Lee, I. Ullah, M. Luna, S. H. Park, M. Azizian, D. Stoyanov, L. Maier-Hein, and S. Speidel. "2018 Robotic Scene Segmentation Challenge." In: *arXiv preprint arXiv:2001.11190* (2020).

[3] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan. "Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2021).

[4] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. "Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).

[5] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. " Unstructured lumigraph rendering." In: *SIGGRAPH* (2001).

[6] A. Cao and J. Johnson. "HexPlane: A Fast Representation for Dynamic Scenes." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).

[7] J. Carr, R. Beatson, J. Cherrie, T. Mitchell, W. Fright, B. McCallum, and T. Evans. " Reconstruction and representation of 3d objects with radial basis functions." In: *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001).

[8] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su. "TensoRF: Tensorial Radiance Fields." In: *European Conference on Computer Vision (ECCV)* (2022).

[9] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation." In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2018).

[10] W. Chen, H. Ling, J. Gao, E. Smith, J. Lehtinen, A. Jacobson, and S. Fidler. "Learning to predict 3D objects with an interpolation-based differentiable renderer." In: *Advances in neural information processing systems (NEURIPS)* (2019).

[11] A. Davis, M. Levoy, and F. Durand. "Unstructured light fields." In: *Computer Graphics Forum* (2012).

[12] P. Debevec, C. Taylor, and J. Malik. "Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach." In: *sIGGGRAPH* (1996).

[13] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan. "Depth-supervised nerf: Fewer views and faster training for free." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).

[14] J. Fang, L. Xie, X. Wang, X. Zhang, W. Liu, and Q. Tian. "Neusample: Neural sample field for efficient view synthesis." In: *arXiv preprint arXiv:2111.15552* (2021).

[15] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. "Deepstereo: Learning to predict new views from the world's imagery." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[16] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa. "K-Planes: Explicit Radiance Fields in Space, Time, and Appearance." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).

[17] Q. Fu, Q. Xu, Y. S. Ong, and W. Tao. " Geo-Neus: Geometry-Consistent Neural Implicit Surfaces Learning for Multi-view Reconstruction." In: *Advances in Neural Information Processing Systems 35 (NeurIPS)* (2022).

[18] C. Gao, A. Saraf, J. Kopf, and J.-B. Huang. "Dynamic View Synthesis From Dynamic Monocular Video." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (ICCV).

[19] K. Genova, F. Cole, A. Maschinot, A. Sarna, D. Vlasic, and W. T. Freeman. "Unsupervised training for 3d morphable model regression." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

[20] M. Hayoz, C. Hahne, M. Gallardo, D. Candinas, T. Kurmann, M. Allan, and R. Sznitman. "Learning how to robustly estimate camera pose in endoscopic videos." In: *International Conference on Information Processing in Computer-Assisted Interventions (IPCAI)* (2023).

[21] K. He, G. Gkioxari, P. Dollar, and R. Girshick. "Mask R-CNN." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2017).

[22] K. Hornik, M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators." In: *Neural Networks, Elsevier* (1989).

[23] J. T. Kajiya and B. P. V. Herzen. "Ray tracing volume densities." In: *Computer Graphics (SIGGRAPH)* (1984).

[24] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis. "3D Gaussian Splatting for Real-Time Radiance Field Rendering." In: *ACM Transactions on Graphics* (2023).

[25] D. Kingma and J. Ba. "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980* (2014).

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks." In: *Advances in Neural Information Processing Systems 25 (NIPS)* (2012).

[27] M. Levoy and P. Hanrahan. "Light field rendering." In: *SIGGRAPH* (1996).

[28] T. Li, M. Slavcheva, M. Zollhöfer, S. Green, C. Lassner, C. Kim, T. Schmidt, S. Lovegrove, M. Goesele, R. Newcombe, and Z. Lv. "Neural 3D Video Synthesis From Multi-View Video." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).

[29] T. Li, M. Aittala, F. Durand, and J. Lehtinen. "Differentiable monte carlo ray tracing through edge sampling." In: *ACM Transactions on Graphics (TOG)* (2018).

[30] Z. Li, S. Niklaus, N. Snavely, and O. Wang. "Neural scene flow fields for space-time view synthesis of dynamic scenes." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).

[31] Z. Li, Q. Wang, F. Cole, R. Tucker, and N. Snavely. "DynIBaR: Neural Dynamic Image-Based Rendering." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).

[32] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey. "BARF: Bundle-Adjusting Neural Radiance Fields." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2021).

[33] L. Liu, J. Gu, K. Z. Lin, T.-S. Chua, and C. Theobalt. "Neural Sparse Voxel Fields." In: *Advances in Neural Information Processing Systems 33 (NeurIPS)* (2020).

[34] Y.-L. Liu, C. Gao, A. Meuleman, H.-Y. Tseng, A. Saraf, C. Kim, Y.-Y. Chuang, J. Kopf, and J.-B. Huang. "Robust Dynamic Radiance Fields." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).

[35] S. Liu, T. Li, W. Chen, and H. Li. "Soft rasterizer: A differentiable renderer for image-based 3d reasoning." In: *Proceedings of the IEEE/CVF* (2019).

[36] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh. "Neural volumes: Learning dynamic renderable volumes from images." In: *SIGGRAPH* (2019).

[37] X. Long, C. Lin, P. Wang, T. Komura, and W. Wang. "SparseNeuS: Fast Generalizable Neural Surface Reconstruction from Sparse Views." In: *European Conference on Computer Vision (ECCV)* (2022).

[38] M. Loper and M. Black. "OpenDR: An approximate differentiable renderer." In: *Computer Vision–ECCV* (2014).

[39] R. G. M. Cohen S.J. Gortler and R. Szeliski. "The lumigraph." In: *SIGGRAPH* (1996).

[40] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth. "NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).

[41] N. Max. " Optical models for direct volume rendering." In: *IEEE Transactions on Visualization and Computer Graphics* (1995).

[42] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[43] A. Meuleman, Y.-L. Liu, C. Gao, J.-B. Huang, C. Kim, M. H. Kim, and J. Kopf. "Progressively Optimized Local Radiance Fields for Robust View Synthesis." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).

[44] B. Mildenhall, P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis." In: *ECCV* (2020).

[45] T. Müller, A. Evans, C. Schied, and A. Keller. "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding." In: *ACM Transactions on Graphics* (2022).

[46] M. NIEMEYER, L. MESCHEDER, M. OECHSLE, and A. GEIGER. "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[47] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger. "Differentiable Volumetric Rendering: Learning Implicit 3D Representations Without 3D Supervision." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).

[48] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob. "Mitsuba 2: A retargetable forward and inverse renderer." In: *ACM Transactions on Graphics* (2019).

[49] K. B. Ozyoruk, G. I. Gokceler, T. L. Bobrow, G. Coskun, K. Incetan, Y. Almalioglu, F. Mahmood, E. Curto, L. Perdigoto, M. Oliveira, H. Sahin, H. Araujo, H. Alexandrino, N. J. Durr, H. B. Gilbert, and M. Turan. "EndoSLAM dataset and an unsupervised monocular visual odometry and depth estimation approach for endoscopic videos." In: *Medical Image Computing and Computer Assisted Intervention – MICCAI* (2021).

[50] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla. "Nerfies: Deformable Neural Radiance Fields." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2021).

[51] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz. "HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields." In: *ACM Transactions on Graphics, vol. 40* (2021).

[52] M. Piala and R. Clark. "Terminerf: Ray termination prediction for efficient neural rendering." In: *International Conference on 3D Vision (3DV)* (2021).

[53] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer. "D-NeRF: Neural Radiance Fields for Dynamic Scenes." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).

[54] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. "On the spectral bias of neural networks." In: *International Conference on Machine Learning* (2019).

[55] C. Reiser, S. Peng, Y. Liao, and A. Geiger. "KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs." In: *International Conference on Computer Vision (ICCV)* (2021).

[56] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari. "Urban Radiance Fields." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).

[57] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2015).

[58] J. L. Schonberger and J.-M. Frahm. "Structure-from-Motion Revisited." In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[59] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition." In: *arXiv preprint arXiv:1409.1556* (2014).

[60] V. Sitzmann, J. Thies, F. Heide, M. Niessner, G. Wetzstein, and M. Zollhofer. "DeepVoxels: Learning Persistent 3D Feature Embeddings." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[61] V. Sitzmann, M. Zollhoefer, and G. Wetzstein. "Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations." In: *Advances in Neural Information Processing Systems 32 (NeurIPS)* (2019).

[62] Z. Teed and J. Deng. "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow." In: *European Conference on Computer Vision (ECCV)* (2020).

[63] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi, T. Simon, C. Theobalt, M. Nießner, J. T. Barron, G. Wetzstein, M. Zollhöfer, and V. Golyanik. "Advances in Neural Rendering." In: *EUROGRAPHICS* (2022).

[64] J. Thies, M. Zollhöfer, and M. Nießner. "Deferred neural rendering: image synthesis using neural textures." In: *Acm Transactions on Graphics (TOG)* (2019).

[65] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari. "SPARF: Neural Radiance Fields From Sparse and Noisy Poses." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).

[66] S. Umeyama. "Least-Squares Estimation of Transformation Parameters Between Two Point Patterns." In: *IEEE Transactions on Pattern Analysis Machine Intelligence* (1991).

[67] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need." In: *Conference on Neural Information Processing Systems (NeurIPS)* (2017).

[68] M. Waechter, N. Moehrle, and M. Goesele. "Let there be color! Large-scale texturing of 3D reconstructions." In: *Computer Vision–ECCV* (2014).

[69] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. "NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction." In: *Advances in Neural Information Processing Systems* (2021).

[70] Y. Wang, Y. Long, S. Fan, and Q. Dou. "Neural Rendering for Stereo 3D Reconstruction of Deformable Tissues in Robotic Surgery." In: *Medical Image Computing and Computer Assisted Intervention – MICCAI* (2022).

[71] Y. Wang, I. Skorokhodov, and P. Wonka. "Hf-neus: Improved surface reconstruction using high-frequency details." In: *Advances in Neural Information Processing Systems 35 (NeurIPS)* (2022).

[72] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. "Image quality assessment: from error visibility to structural similarity." In: *IEEE Transactions on Image Processing* (2004).

[73] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson. "Synsin: End-to-end view synthesis from a single image." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[74] D. Wood, D. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle. "Surface light fields for 3D photography." In: *SIGGRAPH* (2000).

[75] L. Wu, J. Lee, A. Bhattad, Y. Wang, and D. Forsyth. "Diver: Real-time and accurate neural radiance fields with deterministic integration for volume rendering." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).

[76] C. Yang, K. Wang, Y. Wang, X. Yang, and W. Shen. "Neural LerPlane Representations for Fast 4D Reconstruction of Deformable Tissues." In: *Medical Image Computing and Computer Assisted Intervention – MICCAI* (2023).

[77] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa. "PlenOctrees for Real-Time Rendering of Neural Radiance Fields." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2021).

[78] R. Zha, X. Cheng, H. Li, M. Harandi, and Z. Ge. "EndoSurf: Neural Surface Reconstruction of Deformable Tissues with Stereo Endoscope Videos." In: *Medical Image Computing and Computer Assisted Intervention – MICCAI* (2023).

[79] K. Zhang, G. Riegler, N. Snavely, and V. Koltun. "NeRF++: Analyzing and Improving Neural Radiance Fields." In: *arXiv preprint arXiv:2010.07492* (2020).

[80] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. "The unreasonable effectiveness of deep features as a perceptual metric." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

[81] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. "Stereo magnification: Learning view synthesis using multiplane images." In: *SIGGRAPH* (2018).

[82] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. "On the continuity of rotation representations in neural networks." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).