3D Reconstruction of Endoscopy Images with NeRF

THESIS

Submitted in Partial Fulfillment of

the Requirements for

the Degree of

MASTER OF SCIENCE (Computer Science)

at the

NEW YORK UNIVERSITY
TANDON SCHOOL OF ENGINEERING

by

Qin Ying Chen

January 2023

3D Reconstruction of Endoscopy Images with NeRF

THESIS

Submitted in Partial Fulfillment of

the Requirements for

the Degree of

MASTER OF SCIENCE (Computer Science)

at the

NEW YORK UNIVERSITY
TANDON SCHOOL OF ENGINEERING

by

Qin Ying Chen

January 2023

Approved:

Qi Sun (Aug 16, 2022 11:03 EDT)

Advisor Signature

Aug 16, 2022

Date

Lisa Hellerstein (Aug 18, 2022 07:52 CDT)

Department Chair Signature

Aug 18, 2022

Date

University ID: **N10625169**
Net ID: qyc206

Approved by the Guidance Committee:

Major: Computer Science

Qi Sun
Qi Sun (Aug 10, 2022 12:01 PDT)

Qi Sun
Assistant Professor
NYU Tandon School of Engineering

Aug 10, 2022

Date

Saad Nadeem
Saad Nadeem (Aug 10, 2022 15:43 EDT)

Saad Nadeem
Assistant Professor
Memorial Sloan Kettering Cancer Center

Aug 10, 2022

Date

Yi-Jen Chiang
Yi-Jen Chiang (Aug 10, 2022 18:12 EDT)

Yi-Jen Chiang
Associate Professor
NYU Tandon School of Engineering

Aug 10, 2022

Date

Vita

Qin Ying Chen was born in Fuzhou, Fujian, China, on September 20, 1998. She was raised in New York City, where she attended P.S. 33 Chelsea Prep elementary school, Robert F. Wagner Middle School, and Eleanor Roosevelt High School. She studied Computer Engineering during her undergraduate years at NYU Tandon, and she was awarded her Bachelor of Science degree in May 2021. After receiving her undergraduate degree, she continued her education at NYU Tandon to pursue her Master of Science degree in Computer Science. During this time, she joined the NYU Immersive Computing Lab led by Professor Qi Sun, where she was introduced to her thesis project. Professor Qi Sun became her advisor for the project, and Qin Ying closely collaborated with Yujie Wang, a PhD student from Shandong University and Peking University, and Praneeth Chakravarthula, a research scholar in the Princeton Computational Imaging Lab and research assistant professor at UNC Chapel Hill. This thesis work spanned from September 2021 to August 2022. Qin Ying defended her thesis and submitted her thesis paper to fulfill her graduation requirements in August 2022. She will be awarded her Master of Science degree in January 2023.

Acknowledgements

I would like to first thank my advisor, Professor Qi Sun, for introducing me to my thesis topic and for being so understanding and supportive throughout the process. I would also like to thank Praneeth Chakravarthula, a research scholar in the Princeton Computational Imaging Lab and research assistant professor at UNC Chapel Hill, for always checking in with me and always being available to any questions that I may have. In addition, I would like to extend my gratitude to Yujie Wang, a PhD student from Shandong University and Peking University. She spent many hours helping me with the challenges that I faced. Her guidance played an important role in my understanding and completion of my thesis project. I would also like to thank my thesis committee who took time out of their busy schedules during the summer to attend my defense. And last but not least, I would like to thank all my friends and family who showed so much support and encouragement throughout my thesis journey.

ABSTRACT

3D Reconstruction of Endoscopy Images with NeRF

by

Qin Ying Chen

Advisor: Prof. Qi Sun, Ph.D.

Submitted in Partial Fulfillment of the Requirements for

the Degree of Master of Science (Computer Science)

January 2023

During endoscopic procedures, a camera is inserted into the human body to capture images of an inner organ. However, with a single scan of the organ, the camera is unable to capture all of the images needed to provide a complete assessment of the situation in the body. As a result, polyps, or protrusions on the surface of the examined organ, could be missed during the flythrough. The various factors that contribute to polyp misses include camera blind spots, haustral folds hiding polyps, and the inability of the camera angle to capture the polyp shape. Previous works that attempt to lower the polyp miss rate focus on improving the segmentation of the haustral folds in hopes of guiding the camera to areas where polyps could be hidden. The goal of this research, however, is to use Neural Radiance Fields (NeRF), a neural rendering model, to fill in the gaps of missing views in captured endoscopy images and reconstruct a scene that provides information to assist radiologists in identifying anomalies. When trained on a set of real stomach images, the NeRF reconstruction of the scene reveals a protrusion that is unidentifiable when observing the set of images alone. This suggests the potential in using this technique to recover polyps that are lost with the missed or uncaptured views.

Table of Contents

List of Figures

List of Tables

Chapter One: Introduction

Endoscopy is the procedure in which a thin, flexible tube with a camera is inserted into the human body to capture images of an inner organ for radiologists to examine for any potential abnormalities [3]. An anomaly that radiologists often look for during an endoscopy is a polyp, or a protrusion on the surface of an organ. Examples of endoscopic procedures include colonoscopy—an examination of the large intestine and rectum—and gastroscopy—an examination of the stomach and the beginning of the small intestine. The issue with endoscopic procedures is that the camera only gets one chance to fly through the organ and capture images. With this comes several problems that can lead to high polyp miss rates. While there is also the option of virtual endoscopy where computed tomography (CT) scans are taken and used in the 3D reconstruction of the captured organ, this procedure shows neither the texture nor color of the organ, and it is still possible to miss small polyps or misclassify something as a polyp.

Traditional endoscopy is commonly used by radiologists to identify diseases and cancers in the inner organs [3]. Radiologists typically examine these endoscopy images to look for polyps on the surfaces of the captured organ. However, as previously mentioned, there are several concerns that come with endoscopic camera flythroughs that result in high polyp miss rates. One such problem are camera blind spots, where the camera misses a spot because it is not in the camera's view. Another problem is that not all camera angles can

capture the shape or geometry of a polyp. For instance, it is possible for a polyp to be captured in an image with the camera aiming directly at it such that the polyp blends in with the background; however, if this image is taken from a different camera angle, the polyp would be revealed to the viewer. This inability to identify polyps caused by missing views could also be due to the limited number of frames that are available for examination. Since the camera has only a single flythrough to capture images, it is unlikely for the entire scene to be captured from all viewpoints. Finally, haustral folds, or folds that exist in the colon, could also hide polyps from the camera's view, and therefore hide the polyp from the radiologist during examination.

To resolve some of these problems and reduce polyp miss rates, the medical image computing community has been exploring ways to apply computer vision and deep learning techniques to guide the navigation of the camera during the procedures to cover more surface area in the organ and spot potentially hidden polyps [1, 4, 8, 9, 10, 15]. The most common approach is to segment and identify haustral folds such that the camera can be navigated towards areas with high density of folds to search for hidden polyps. However, this is a difficult task due to uncontrollable factors such as the organ's texture and lighting variation, specular reflections, fluid motion, and movements.

Previous works by Mathew et al. on the XDCycleGAN [8] and FoldIt [10] models explored the use of Generative Adversarial Networks (GANs) [5] to achieve accurate detection and segmentation of haustral folds in colonoscopy images and videos. The XDCycleGAN model has been shown to more effectively

handle texture and lighting variation and specular reflections when inferring scale-consistent depth maps than previous supervised approaches. While it is shown that XDCycleGAN can be used to segment haustral folds, it is still unable to achieve feature consistency between consecutive frames. The FoldIt model showed further improvement by demonstrating the ability to not only accurately detect and segment the folds but also achieve feature-consistency in the annotations between consecutive frames. Even though these models demonstrate promising results, they require both optical and virtual colonoscopy data for training. Obtaining large amounts of traditional and virtual colonoscopy data to cover all possible scenarios may be a difficult task since patients will typically opt for only one of these two types of procedures, and these procedures are completed every five to ten years by patients who are over 45 years old. Another limitation is that these solutions only target one part of the cause for polyp miss rates, which is haustral folds potentially hiding polyps.

Instead of looking to segment haustral folds, the research presented in this paper evaluates the use of a popular neural rendering model, Neural Radiance Fields or NeRF [11], to reconstruct an endoscopic scene that contains both the seen and unseen viewpoints. NeRF is a fully-connected neural network that takes a set of 2D images and their corresponding camera poses as input for training. After training, this model is able to generate novel views of the complex 3D scene that is captured by the 2D images. Unlike the previously mentioned approach to reduce polyp miss rates, this approach could help recover the polyps that are lost due to camera blind spots and camera angles that cause view gaps

in the frames that are captured. The 3D reconstruction of endoscopy images using NeRF could provide more information for radiologists to examine and assist them in identifying anomalies.

For this research, a NeRF model is trained using a dataset from EndoSLAM [13] containing real stomach images. EndoSLAM is a public repository that contains real and synthetic endoscopy image datasets. The result from testing the customly trained NeRF model is rendered as a gif showing the scene from different camera viewpoints. In addition to the gif, depth maps are also rendered for evaluation. If NeRF is proven to be effective, this research would introduce a new possibility for medical professionals in their search for low-cost alternatives to reduce polyp miss rates.

Chapter Two: Related Works

2.1 Segmentation of Haustral Folds

One of the main causes for high polyp miss rates during colonoscopy procedures are the haustral folds, also known as colon wall protrusions. These folds can easily hide polyps from the camera's view during the flythrough. Since haustral folds are one of the main causes for high polyp miss rates in colonoscopy procedures, they are highly focused on in the medical image computing community. The general approach to solving this issue is to segment the folds and identify areas of high occlusion such that the camera can be guided towards those areas to uncover potentially hidden polyps. However, segmenting the folds is a difficult task due to several uncontrollable factors: texture and lighting variations, specular reflections, fluid motion, and organ movements. Many existing approaches [1, 4, 8, 9, 10, 15] use deep learning techniques such as Generative Adversarial Networks (GAN) [5] to achieve surface coverage, depth estimation, and 3D reconstruction tasks using the information obtained from endoscopic procedures.

2.1.1 Generative Adversarial Networks (GAN)

Generative Adversarial Networks or GAN [5] is a deep learning technique that can be used to generate images of interest from scratch or transform images from one domain to another. The architecture of GAN consists of two networks: a generator network that learns to generate realistic-looking images, and a

discriminator network that attempts to differentiate between the generated images and the real image examples. The goal of a GAN is for the generator to be able to produce realistic images that can trick the discriminator into deciding that these images are from the pool of real images. However, mode collapse may occur during training that makes GANs difficult to use in practice. This problem arises when the generator figures out how the discriminator works and constantly produces the same output for any given input, knowing that the discriminator will always accept this output.

To combat the mode collapse problem, CycleGAN [22] is introduced. CycleGAN is a two GAN system, where the first system takes an input image and transforms it into something that looks like a realistic target image, and the second system transforms this result back into something that looks like the original input image. This two GAN system ensures that the generator is not just producing the same image every time that has nothing to do with the input image, which therefore solves the mode collapse problem. This way, CycleGAN also allows for unpaired image-to-image translation, making this a technique that can be used to leverage the desirable aspects found separately in optical and virtual colonoscopy data. Prior works such as XDCycleGAN [8] and FoldIt [10] make use of this technique to produce more desirable endoscopy images and achieve segmentation of haustral folds.

2.1.2 XDCycleGAN and FoldIt

XDCycleGAN is a variation of CycleGAN introduced by Mathew et al. [8] that uses the geometric information extracted from 3D virtual colonoscopy data

along with the optical colonoscopy video frames to infer scale-consistent depth maps. Part of its design is a lossy image-to-image translation model that removes texture, color, and specularity from virtual colonoscopy images, and a Directional Discriminator that allows for the stronger removal of specular reflections and textures when linking optical colonoscopy images to virtual colonoscopy images. The XDCycleGAN model has been shown to be effective in handling texture and lighting variations and specular reflections. This suggests that XDCycleGAN is a technique that can be used to make haustral fold segmentation an easier task. While the XDCycleGAN model could be used to segment haustral folds, it is shown to lack in feature-consistency between consecutive frames.

Not long after the publication of XDCycleGAN, FoldIt was introduced by Mathew et al. [10] as a model that can translate images from the optical colonoscopy domain to the virtual colonoscopy domain, and vice versa, with haustral fold annotations. This model is shown to not only be able to accurately detect and segment the haustral folds but also achieve feature-consistency in the annotations between consecutive frames. Unlike its predecessor, FoldIt is a semi-supervised model that leverages the ground truth information to preserve the haustral fold features in the resulting domain by using a transitive loss. The limitation of FoldIt, however, is that it cannot handle scenarios with large amounts of fluids and blurriness because it has not yet been trained to handle those types of data.

A comparison of the results from training and testing XDCycleGAN and FoldIt models on various colonoscopy video sequences show that the annotations made by the FoldIt model are generally closer to the ground truth and more consistent across frames than that of the XDCycleGAN model. Even though these approaches show promising results, they require not only optical colonoscopy images but also virtual colonoscopy images for training. This suggests that to properly train these models, large amounts of optical and virtual colonoscopy data with several variations are needed. However, obtaining both types of data that cover all possible scenarios may be difficult and unlikely because patients will typically opt for only one of these two types of procedures, and these procedures are completed every five to ten years by patients over 45 years old.

2.2 Neural Radiance Fields (NeRF)

The Neural Radiance Fields (NeRF) model introduced by Mildenhall et al. [11] is a rapidly growing technique in the computer vision and deep learning community. The goal of NeRF is to render both the seen and unseen views of a 3D scene when given only a set of 2D images captured with the corresponding camera poses. Different from traditional neural networks, NeRF aims to overfit a neural network to a single data point such that the weights of the network is the actual scene. Since the model itself represents the scene, rendering a part of the scene means to query the model from different viewing positions, including the unseen viewpoints in the given set of input images. This rendering process uses

a concept called ray tracing, where for every pixel, a ray is sent into the bounded region that the scene lives in. For every point that is sampled along this ray, it is sent as input into the trained NeRF model to obtain a color and opacity value. The alpha composing portion of ray tracing works backwards from the end of the ray back to the camera to determine the final color and opacity of the pixel. The limitation of NeRF, however, is that it can take a long time to fit a neural network because it is a computationally-intensive algorithm.

As previously stated, NeRF has been quickly gaining popularity in the computer vision and deep learning community, and there have been several extended research and developments on this neural network. These research efforts generally aim to improve its results and performance, find new applications, and build on top of the existing framework to achieve another related goal [2, 7, 12, 18, 19, 21]. One example of such research is mip-NeRF [2], a variation where point-based ray tracing used by the traditional NeRF model is replaced by cone tracing for smooth multi-scale rendering of a scene. This modeling of a whole volume of space that is visible by each pixel provides a more accurate representation of scenes with varying camera distances. Additionally, mip-NeRF is found to reduce the model size and increase training speed. Another example is NeRF in the Wild [7], an extension of NeRF that takes a collection of unconstrained images of a scene and produces novel views of that scene where the camera can be moved around and the appearance of the scene can be changed to one that exists in any of the input images. A flythrough comparison of NeRF and NeRF in the Wild shows that more transient objects

can be seen in NeRF's rendered scene than in NeRF in the Wild's rendered scene. This is because the NeRF implementation does not have a way to detect and ignore the transient parts of the scene like that of NeRF in the Wild.

In addition to improving the rendered scene using the NeRF model, another challenge that is explored in the community is how NeRF handles sparse input images. Niemeyer et al. [12] observes that the quality of NeRF renderings drops significantly when trained using sparse input views. The results show that the rendered views contain several floating artifacts and color shifts. When using the mip-NeRF implementation, the quality drop is found to be caused by incorrect scene geometry and training divergence. The goal of RegNeRF [12] is to obtain 3D-consistent representations and render realistic novel views despite having sparse input images. This approach consists of three parts: a patched-based regularizer, a normalizing flow model, and an annealing strategy for sampling points along the ray. The results of RegNeRF show much improvement in consistency and reduction of noise in the rendered views. With only three input views, the rendered views have smoother surfaces with a clear depth map and without any floating artifacts or clouds. RegNeRF is also shown to outperform similar models such as PixelNeRF [20] and DietNeRF [6]. The limitation of this approach, however, is that there are blurry predictions in unobserved areas with fine geometric structures.

As of now, most, if not all, applications found for NeRF are for entertainment purposes, but none explore NeRF's potential in critical applications, such as a medical setting. With there being several promising

advancements in NeRF research, it is clear that there is potential for this technique to help medical professionals who may be limited by their resources. This technique could allow radiologists to observe more realistic views and make more accurate conclusions from their observations, even if they only have a few images of an endoscopic scene. Furthermore, NeRF models are able to provide depth estimations for each corresponding rendered view, providing even more information for radiologists to examine and consider. This is useful when a polyp is identified and its depth and size are needed to determine whether the polyp needs to be removed. NeRF has the potential to become an important tool for radiologists to use during endoscopy image analysis.

Chapter Three: Methodology

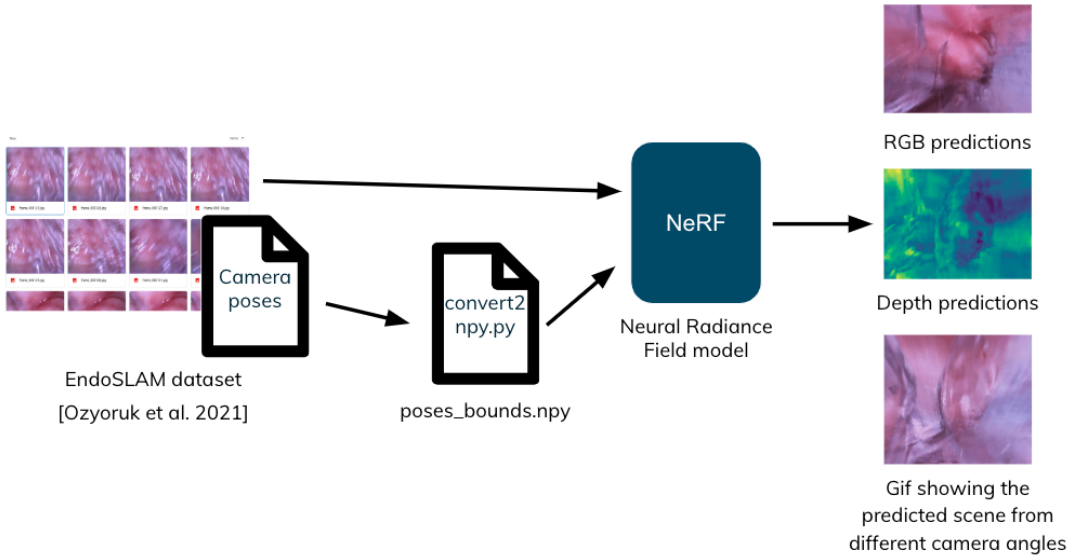


Figure 1: *Pipeline showing the general steps taken to train a NeRF model with custom data.*

Figure 1 shows the general pipeline for this research. First, images captured from an endoscopic procedure with corresponding camera poses are acquired. If there are no camera poses provided, the poses would need to be estimated. One way to do so is to use COLMAP [16, 17], a pipeline that can be used for the reconstruction of image collections. The images used to train the NeRF model for this research are from the EndoSLAM [13] repository, a public repository containing datasets of real endoscopy images taken in a controlled setting as well as datasets of synthetic endoscopy images. The datasets found on EndoSLAM contain not only the endoscopy image frames but also the corresponding camera pose information, which are both crucial information needed for NeRF training. The dataset that is used in this research is of a

stomach. These stomach frames were taken in a very controlled manner: the stomach was cut open, sewn to a foam-like board, and images were taken by a camera that was attached to a human-controlled robotic arm that moved around above the stomach.

After obtaining the dataset, the camera poses were converted into the format that NeRF expects. The expected camera pose file is a Nx17 numpy array where each row is a concatenation of a flattened 3x3 rotation matrix, a 1x3 translation matrix, a 1x3 vector containing image data, and a 1x2 vector containing the close and far depths of the camera. It can be observed that the provided camera poses are quaternion coordinates while the expected poses are euler coordinates, and so the conversion from quaternion to euler coordinates is applied to every pose and formatted such that the three axes are [down (-y), right (x), backward (z)] to obtain the rotation matrix. The 3x3 rotation matrix is then flattened into a 1x9 vector. The translation matrix consists of values provided in the excel file containing camera poses. The image data is a vector containing image height, image width, and focal length. It should be noted that the code written and used to create the camera pose file uses arbitrary values for the stomach dataset's image data and close and far depths of the camera.

Once the camera pose file is created in the format that the NeRF model expects, this file along with the corresponding raw images are used as input to train the model [14]. After training, the test views are then rendered as images, and the corresponding depth predictions are made and stored in files. A gif is also produced to show the predicted scene from different viewpoints. The results

discussed in this paper are obtained from training a NeRF model using 20 consecutive image frames of the stomach for 10 epochs. The resulting gif is linked and further discussed in the following results section.

Chapter Four: Results and Evaluation

4.1 Results

After training the NeRF model using the stomach dataset obtained from EndoSLAM for 10 epochs, the result is rendered as a gif to show the scene from different camera viewpoints. Sample images showing the rendered scene prediction are shown in Figure 2, and the gif of the rendered scene is also attached in this figure.

Figure 2: *Sample images of the NeRF-rendered prediction of the real stomach scene showing different viewpoints obtained after NeRF training. Follow this link to view the complete gif: stomach.gif.*

It can be observed in Figure 2 that there is a protrusion in the rendered scene. This protrusion is especially apparent in the gif that is linked in the figure. This observed protrusion could very likely be a polyp that could not be identified with just the input images alone. In addition to the resulting gif, the corresponding depth predictions are also visualized, as shown in Figure 3.

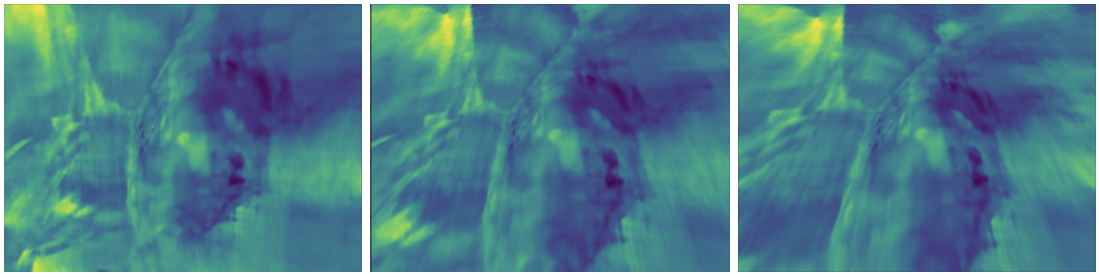


Figure 3: *Sample images of the predicted depth visualizations obtained from training NeRF using the real stomach dataset from EndoSLAM.*

It can be observed in the depth prediction images shown in Figure 3 that there is a protrusion in the rendered predictions, and this is consistent with the observations made for Figure 2. These promising results suggest that this custom-trained NeRF model can be used to recover geometry that would otherwise be lost along with the missing unseen views of this stomach scene. Furthermore, this suggests that NeRF is a neural rendering framework that can be used to fill in the gaps of missing views of a camera-captured endoscopic scene, and therefore assist radiologists in identifying anomalies. However, to be able to make this conclusion confidently, the accuracy of these prediction results must be verified and evaluated.

4.2 Evaluation

During evaluation, the rendered predicted depth is compared to the ground truth depth and the quality of the image reconstruction is evaluated. The comparison of the predicted depth and the ground truth depth involves comparing the shape of the depth maps and measuring the depth accuracy using

the mean squared error (MSE). The quality of the image reconstruction is measured using two metrics: peak signal to noise ratio (PSNR) and structural similarity index (SSIM).

To evaluate the accuracy of the NeRF-rendered predicted views, depth prediction for each rendered view and the corresponding ground truth view need to be obtained and compared. Ideally, the resulting depth predictions shown in the results section would be evaluated against the ground truth depth of the corresponding real images. However, the corresponding ground truth depth is not provided in the dataset from EndoSLAM, and so there is no ground truth to evaluate the predictions against. The EndoSLAM repository did provide a 3D mesh of the stomach for the corresponding used dataset. When attempting to visualize and capture the corresponding scenes and render the corresponding depths using this mesh and the camera poses, it is observed that the camera position is very far from the mesh. The mesh would need to be moved closer to the camera to capture the views and render the corresponding depths. Doing this, however, would make the ground truth inaccurate since this setup would no longer be consistent with the original experimental setup used to collect the real images.

This issue calls for the need of an alternative method to evaluate the results. The solution is to create a custom dataset using the 3D mesh on a rendering software, such as Blender. Since the dataset is customly collected, there is more control over this dataset and the ground truth depth can be rendered to use for evaluation. The idea is to train a NeRF model using this

custom dataset and compare the resulting NeRF depth estimations with the corresponding ground truth depths that are rendered during the collection of the synthetic data.

To prepare for the collection of the synthetic dataset, the camera distribution of the real image dataset from EndoSLAM is visualized by using the provided camera pose information and plotting the camera positions and rotations. Figure 4 shows the visualized camera distribution for this real image dataset. It can be seen in Figure 4 that there is much overlap between the consecutive frames or cameras, and the cameras are generally facing the z direction, as indicated by the blue arrow. This camera distribution makes sense for NeRF because the input views need to have some overlap for NeRF to reconstruct scenes. Therefore, when rendering the synthetic data, the camera distribution for this synthetic data should also show similar overlaps to create a scene that closely represents the original real image scene.

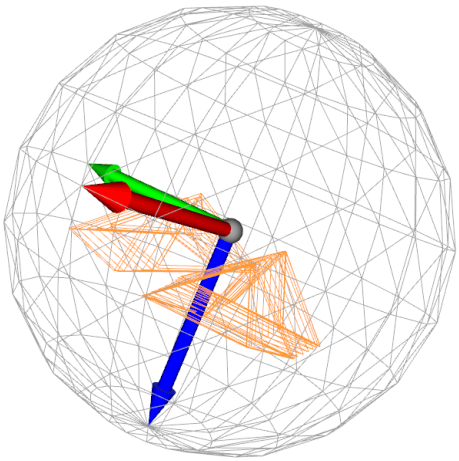


Figure 4: *Camera distribution for the stomach dataset from EndoSLAM.*

A python script was written to render the synthetic dataset to move a camera around the 3D stomach mesh on Blender and render the frames and corresponding depth maps. The camera is initially positioned at a specified, hard-coded position on the mesh and the remaining camera poses are sampled roughly on a plane with the same z-value. Figure 5 shows the visualized camera distribution for the synthetic data. It can be observed that although the camera distribution is not exactly a copy of the one for the real image dataset, it shows similar overlaps in one focused area and the cameras are all facing the z direction, which is consistent with the cameras of the stomach dataset. Once the synthetic dataset is collected and the camera poses are prepared in a file that NeRF expects for training, a new NeRF model is trained using this synthetic data to use for evaluation. The synthetic dataset that is rendered contains 30 consecutive frames, and the NeRF model is trained using this dataset for 150,000 epochs.

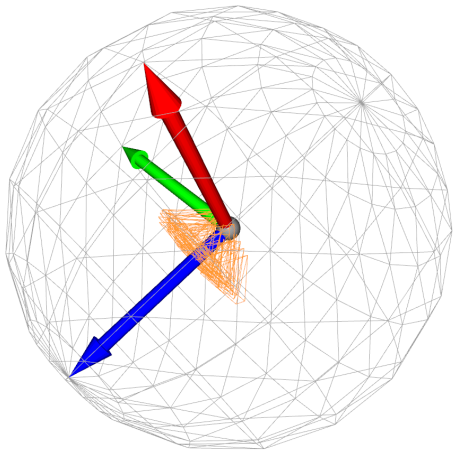


Figure 5: *Camera distribution for the rendered synthetic dataset.*

Figure 6 shows a comparison of the rendered predicted view using the new NeRF model that is trained with the synthetic data and the corresponding ground truth view. It can be observed that the predicted view matches the ground truth view very well. The rendered gif that shows the predicted scene from different viewpoints is also linked in Figure 6. The next step in evaluating the results is to compare the predicted and ground truth depths. Figure 7 shows the predicted depth compared with the corresponding ground truth depth. Bright spots can be spotted in the predicted depth map that is not present in the ground truth depth map, as circled in red in Figure 7. These bright patches that are observed in the depth prediction are likely to be caused by the specular reflection in the input views. To test this claim, specularity removal in the real image dataset is first explored to determine whether removing specular reflections in input views would have an effect on the resulting depth predictions. Specularity removal in the real image dataset is tested first because while specular reflections can be easily removed during the synthetic data rendering process with the use of Blender, the goal is to ultimately determine whether removing specular reflections is effective for real images to be used in real applications.

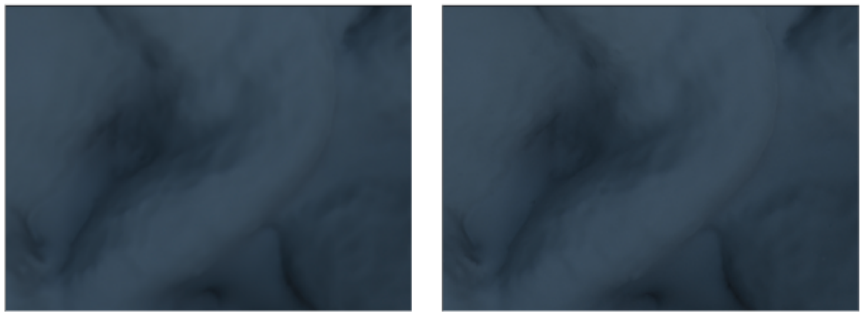


Figure 6: *Comparison of a predicted view using the NeRF model trained with synthetic data and the corresponding ground truth view. Follow this link to view the rendered gif: stomach_synthetic.gif.*

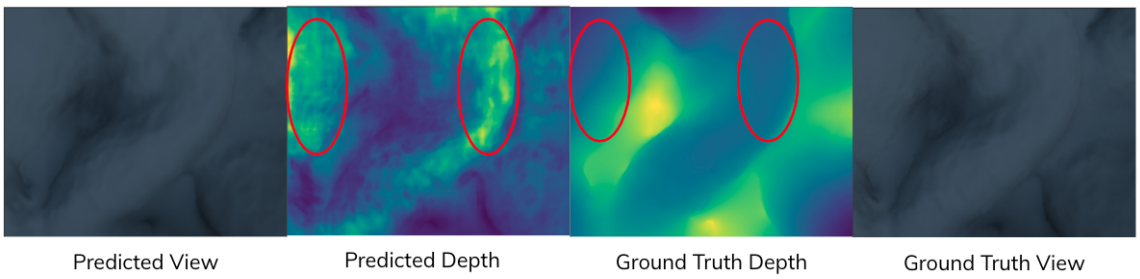


Figure 7: *Comparison of a predicted depth map using the NeRF model trained with the synthetic data and the corresponding ground truth depth map.*

For specularity removal in the real image dataset of the stomach, the pre-trained XDCycleGAN model that is made publicly available on the Computation Endoscopy Platform (CEP) is used to preprocess the images before using them as input views for NeRF training. To reiterate, XDCycleGAN is a variation of CycleGAN that leverages the strengths found separately in the virtual and optical colonoscopy image domains to infer scale-consistent depth maps. Part of the XDCycleGAN design involves the stronger removal of textures and lighting, which would help achieve the goal of specular reflection removal from

the input views. This model has been shown to more effectively remove specular reflections and textures from optical images than other similar models.

Figure 8 shows the workflow of using the pre-trained XDCycleGAN model to preprocess the real images of the stomach and remove the specular reflection from these images. During testing, the pre-trained XDCycleGAN model expects two folders, “testA” and “testB”, each containing images from the optical and virtual domains. Since only the images from the optical domain are available and the major goal is to remove specular reflections, these real images are placed into both test folders for testing. The resulting images are grayscale with the color and specularity removed from the original images. Figure 9 shows the resulting predictions of the NeRF model trained with the original stomach dataset compared with that of the NeRF model trained with the new preprocessed dataset. It can be seen in the depth map comparison shown in Figure 9 that there is improvement in the depth prediction; the depth map on the left shows more bright patches than the depth map on the right, as shown by the area circled in red. This shows that using the pre-trained XDCycleGAN model to preprocess the real images and remove the specular reflections from these input views is indeed effective in removing the bright patches that were previously observed. Now that specularity removal in input views is proven to be effective in reducing the bright spots in NeRF’s depth predictions, specular reflections can be removed in the 3D stomach mesh using Blender and the synthetic dataset can be re-rendered.

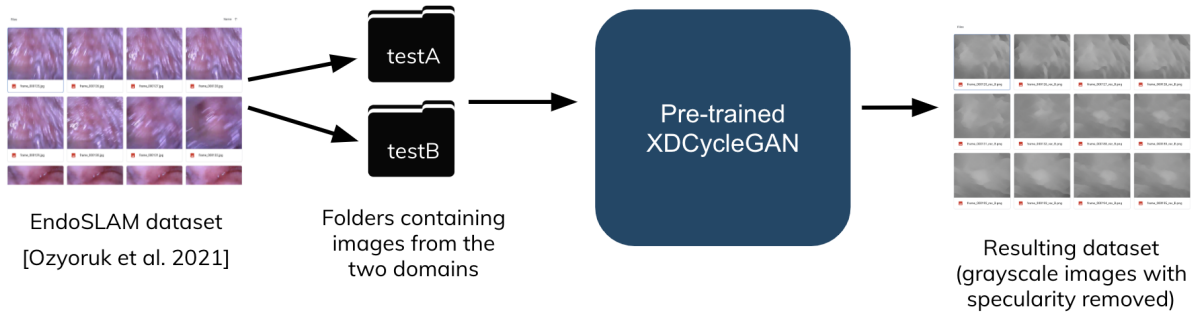
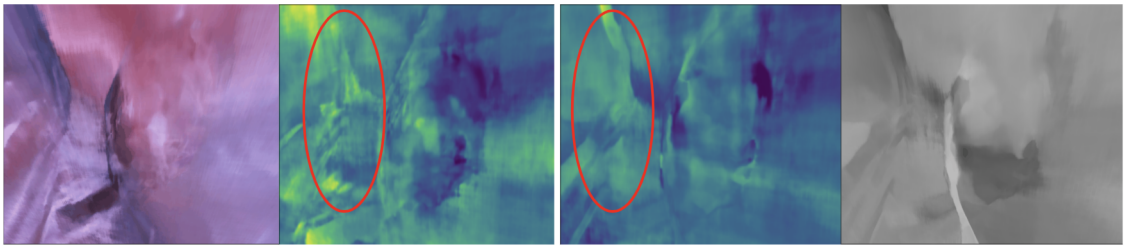


Figure 8: *Pipeline showing the steps taken to preprocess the real images of the stomach using the pre-trained XDCycleGAN model that is publicly available on the Computation Endoscopy Platform (CEP).*



Prediction (view and depth) using the NeRF model trained with the original images from EndoSLAM

Prediction (depth and view) using the NeRF model trained with the new pre-processed images

Figure 9: *Comparison of the rendered results using the NeRF model trained with the original dataset of the stomach from EndoSLAM versus the one trained with the preprocessed dataset of the stomach. Follow this link to view the rendered gif for the NeRF model trained with preprocessed dataset: stomach_real_image_noSpecularity.gif.*

Figure 10 displays the results after repeating the process of re-rendering the synthetic dataset without specular reflection, training the NeRF model, obtaining the results, and visualizing and comparing the depth a few times. It can be observed in Figure 10 that the predicted view matches well with the ground

truth view, which is consistent with the previous results (when NeRF is trained using synthetic data that contains specular reflections). Figure 11 shows the depth prediction compared with the corresponding ground truth depth. It can be observed that there are no longer bright patches in the predicted depth map that do not exist in the ground truth depth map, which is consistent with the observations made when specular reflection is removed from the real image stomach dataset before training the NeRF model. The shape of the predicted depth map is also found to match that of the ground truth depth map. However, the predicted depth map image still does not entirely match the ground truth, indicating that there are still inaccuracies in the predicted depth.

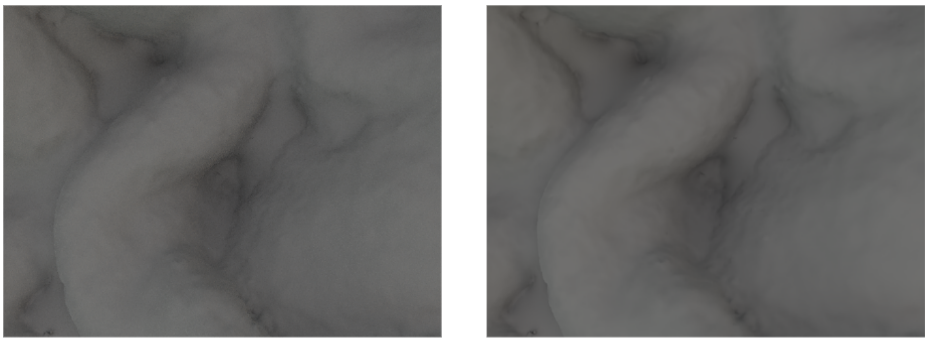


Figure 10: *Comparison of a predicted view using the NeRF model trained with synthetic data (without specular reflection) and the corresponding ground truth view. Follow this link to view the rendered gif: stomach_synthetic_noSpecularity.gif.*

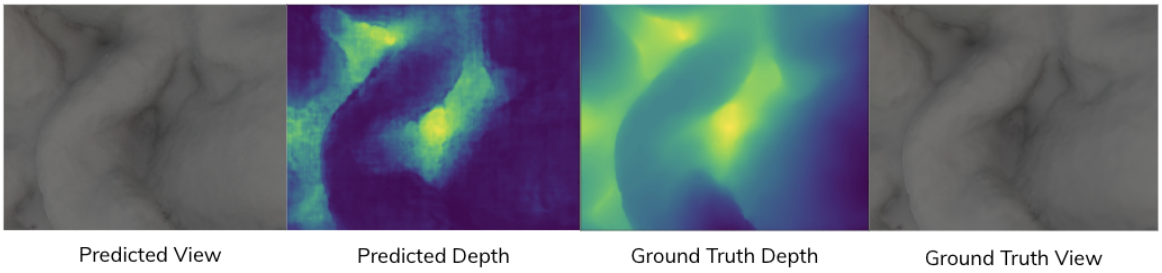


Figure 11: *Comparison of a predicted depth map using the NeRF model trained with the synthetic data with the specular reflection removed and the corresponding ground truth depth map.*

The point clouds generated from the predicted and ground truth RGB-D images are compared to further investigate the differences in the predicted and ground truth depth maps. Figure 12 shows a comparison of these generated point clouds. Note that the color difference in the point clouds does not provide any insight for the depth difference; this coloring may have been caused by the subtle color difference in the predicted and ground truth RGB-D images. In the point clouds shown in Figure 12, it can be observed that the shape of the predicted depth roughly aligns with that of the ground truth depth. However, the thickness or coverage of the depth values do not match very well. This can be seen especially when comparing the border areas of the point clouds, as shown in the red circles in Figure 12. This observation makes sense for the border areas because for NeRF, center areas are always reconstructed with much higher quality than border areas. This is because the center areas are covered by nearly every input view while the border areas are typically only seen by a few views. In the areas circled in blue in Figure 12, depth misalignment can also be observed. This misalignment may be due to the fact that the captured images are all facing

the scene and not enough information of the side views is provided to reveal the thickness or coverage that exists in the scene. For NeRF to improve its depth predictions, more side views would need to be taken to reduce occlusion and better capture the shape and thickness that are present in the scene.

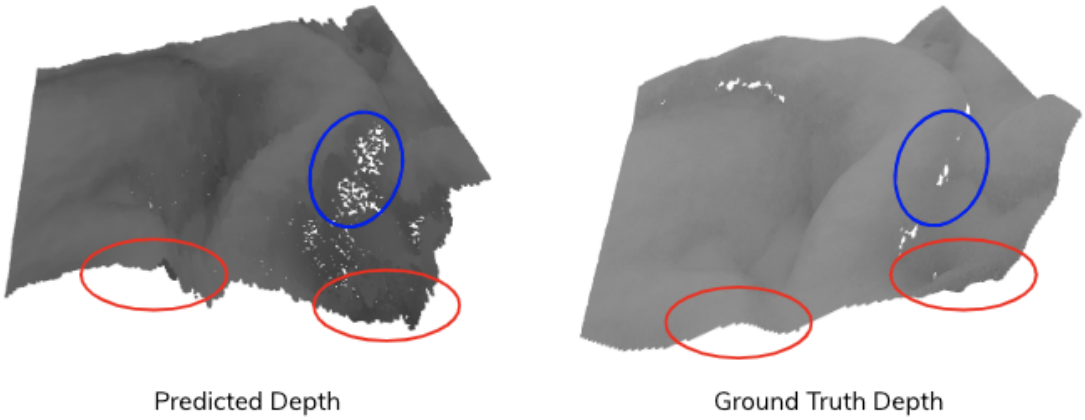


Figure 12: *Comparison of point clouds generated using the predicted and ground truth RGB-D images. These RGB-D images are generated using the NeRF model that is trained with the synthetic dataset that does not contain specular reflections in its images.*

The final step in evaluating the results is to calculate metrics as well as the error of the depth predictions. Table 1 shows the results from calculating the metrics and error for evaluation. The two metrics that are calculated for evaluating the reconstructed images are peak signal to noise ratio (PSNR) and structural similarity index (SSIM). The average PSNR for view synthesis is calculated to be around 39.6 dB. This indicates that the image reconstruction reaches satisfaction as generally 30 dB is considered to be desirable. The average SSIM score is calculated to be around 0.9. SSIM measures how similar

two images are, and a score of 0.85 is generally considered to be reasonable. Similar to the PSNR metrics, the SSIM score of 0.9 also suggests that the image reconstruction is very similar to the ground truth. Finally, after normalizing and scaling the predicted depth and ground truth depth to the same range, the average mean squared error (MSE) is calculated to be around 1.4 m. Note that the unit is in meters because both the script that rendered the ground truth views and the depths on Blender represented the depth values in meters.

Table 1: *Numerical metrics and error calculated for evaluation of the NeRF model trained with synthetic data (without specular reflection).*

Average Peak Signal To Noise Ratio (PSNR)	Average Structural Similarity Index (SSIM)	Average Mean Squared Error (MSE)
39.6 dB	0.9	1.4 m

Chapter Five: Conclusion and Limitations

When the NeRF model is trained on endoscopy images, it is able to produce a shape-accurate reconstructed scene, but the accuracy of the depth prediction still needs some improvement. With depth accuracy being an important factor for radiologists to know, especially when they find a polyp that they should remove, the MSE should be much lower than the calculated 1.4 m for this research's evaluation. Clearly, this is a limitation of rendering endoscopy images with the NeRF model that needs to be further explored in future research and practice.

Another limitation that is found with using NeRF to render endoscopy images is that NeRF does not seem to work well when trained using tunneled views as input. When NeRF is trained using the synthetic dataset provided by EndoSLAM, which contains tunneled views, the rendered view predictions are found to contain shape distortions and the predicted depth does not provide any helpful or usable information, as shown in Figure 13. Since most endoscopy images that are captured during real endoscopic procedures are captured in a tunnel-like setting, it is important for NeRF to be able to handle these types of scenarios. However, the results shown in Figure 13 suggest that NeRF may not work well for 3D reconstruction of most if not all real-world captured endoscopy images.

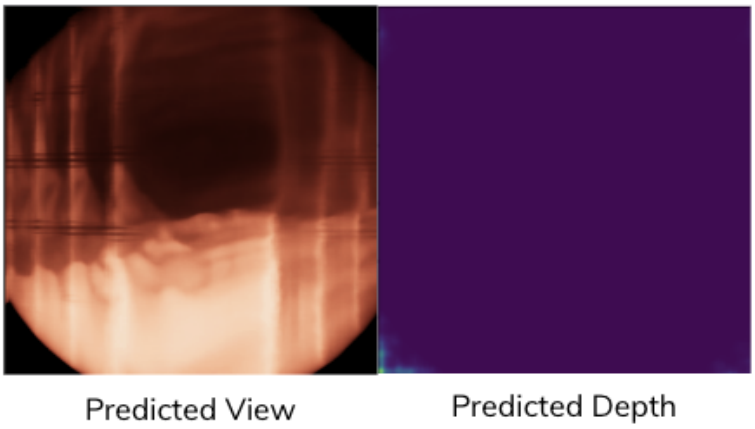


Figure 13: *Sample view and depth prediction using a NeRF model that is trained on tunneled views. Follow this link to view the rendered gif: tunneled_views.gif.*

NeRF traditionally expects the camera distribution to be moving around a scene rather than moving through the scene. Thus, the shape distortions observed in Figure 13 are likely caused by NeRF's inability to reconstruct the scene using cameras that move through the colon scene. This highlights the important role that camera poses have on the performance of NeRF when using endoscopy images. As shown in the results for the stomach dataset, NeRF has the capability to recover polyps that are lost due to missing camera views. This demonstrates NeRF's potential to be a helpful tool for endoscopy image analysis. However, the camera must take images of the organ surface of interest rather than the entire tunneled view for NeRF to provide usable information.

Chapter Six: Future Research

With depth being a crucial piece of information for radiologists to obtain, it is important to improve the accuracy of NeRF's depth predictions. One way to possibly improve the depth accuracy is to test and evaluate the more recent implementations of the NeRF model. With NeRF being a model that is often researched, there are several versions of the model implementation that can be found and tested. It is possible that a more recent model implementation may have better depth prediction capabilities. Another way to improve depth prediction is to add a depth awareness loss to NeRF during training to lower the mean squared error for depth predictions. A third way is to make it possible for the endoscopic cameras to capture more views during the flythrough, especially the side views. Future research should study the possibility of a protrusion-detecting camera that can detect protrusions in real-time and capture multiple views of the detected protrusion from various viewpoints. NeRF can then be used to predict the depth of the protrusion more accurately and provide more information for radiologists to examine and consider.

Another possibility for future research is to study the 3D reconstruction of scenes with endoscopy images that contain tunneled views. As shown in the previous section, NeRF does not perform effectively when the image contains a tunneled view; there are shape distortions that exist in the scene prediction and the depth prediction does not provide any usable information. Since cameras in real world endoscopic procedures often fly through colons or other tunnel-like

portions, it is more likely to see these types of scenes captured than the ones used in this research (i.e. the stomach dataset from EndoSLAM).

Finally, it would be worthwhile to find and evaluate a neural rendering model that can render and produce equally as good or better scene and depth predictions with sparse input views for training. Since endoscopic procedures only allow a single camera flythrough, it can be difficult for the camera to capture all possible images. Thus, there is a limited number of images that can be used to study the organ that is in the scene. This also means that there are less images that can be used to reconstruct the scene. RegNeRF [12], as previously mentioned in prior works, is a NeRF extension that has been recently published and has been shown to be able to render accurate scene predictions and clear depth predictions with as few as three input views. It would be interesting to test and evaluate this model using endoscopy images as well.

References

1. Bae, G., Budvytis, I., Yeung, C.K., & Cipolla, R. (2020). Deep Multi-view Stereo for Dense 3D Reconstruction from Monocular Endoscopic Video. MICCAI.
2. Barron, J. T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., & Srinivasan, P. P. (2021). Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 5855-5864).
3. *Endoscopy - Better Health Channel*. (n.d.). Better Health Channel. Retrieved August 4, 2022, from https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/endoscopy
4. Freedman, D., Blau, Y., Katzir, L., Aides, A., Shimshoni, I., Veikherman, D., Golany, T., Gordon, A., Corrado, G.S., Matias, Y., & Rivlin, E. (2020). Detecting Deficient Coverage in Colonoscopies. IEEE Transactions on Medical Imaging, 39, 3451-3462.
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.
6. Jain, A., Tancik, M., & Abbeel, P. (2021). Putting nerf on a diet: Semantically consistent few-shot view synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 5885-5894).

7. Martin-Brualla, R., Radwan, N., Sajjadi, M. S., Barron, J. T., Dosovitskiy, A., & Duckworth, D. (2021). Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7210-7219).
8. Mathew, S., Nadeem, S., Kumari, S., & Kaufman, A. (2020). Augmenting colonoscopy using extended and directional CycleGAN for lossy image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4696-4705).
9. Mathew, S., Nadeem, S., & Kaufman, A.E. (2021). Visualizing Missing Surfaces In Colonoscopy Videos Using Shared Latent Space Representations. 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), 329-333.
10. Mathew, S., Nadeem, S., & Kaufman, A. (2021, September). FoldIt: Haustral Folds Detection and Segmentation in Colonoscopy Videos. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 221-230). Springer, Cham.
11. Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2020, August). Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision* (pp. 405-421). Springer, Cham.
12. Niemeyer, M., Barron, J. T., Mildenhall, B., Sajjadi, M. S., Geiger, A., & Radwan, N. (2022). Regnerf: Regularizing neural radiance fields for view

synthesis from sparse inputs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5480-5490).

13. Ozyoruk, K. B., Gokceler, G. I., Bobrow, T. L., Coskun, G., Incetan, K., Almalioglu, Y., ... & Turan, M. (2021). EndoSLAM dataset and an unsupervised monocular visual odometry and depth estimation approach for endoscopic videos. *Medical image analysis*, *71*, 102058.
14. Quei-An, C.. (2020). Nerf_pl: a pytorch-lightning implementation of NeRF.
15. Rau, A., Edwards, P.J., Ahmad, O.F., Riordan, P., Janatka, M., Lovat, L.B., & Stoyanov, D. (2019). Implicit domain adaptation with conditional generative adversarial networks for depth prediction in endoscopy. International Journal of Computer Assisted Radiology and Surgery, 14, 1167 - 1176.
16. Schönberger, J., & Frahm, J.M. (2016). Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
17. Schönberger, J., Zheng, E., Pollefeys, M., & Frahm, J.M. (2016). Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*.
18. Srinivasan, P. P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., & Barron, J. T. (2021). Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7495-7504).
19. Yen-Chen, L., Florence, P., Barron, J. T., Rodriguez, A., Isola, P., & Lin, T. Y. (2021, April). inerf: Inverting neural radiance fields for pose estimation.

In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1323-1330). IEEE.

20. Yu, A., Ye, V., Tancik, M., & Kanazawa, A. (2021). pixelnerf: Neural radiance fields from one or few images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4578-4587).
21. Zhang, K., Riegler, G., Snavely, N., & Koltun, V. (2020). Nerf++: Analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492.
22. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).