Semantic Reconstruction Using NeRFs

Group 6: Athreyi Badithela, Ryan Diaz, Ivan Radkevich, Arthur Nghiem

December 15, 2023

1 Introduction

3D semantic reconstruction of a scene is a complex task composed of two parts: *reconstruction*, in which the 3D point cloud or mesh of the scene is recovered from a smetricsequence of 2D image observations, and *segmentation*, in which various objects within the 3D geometry are identified and localized per point. Separately, these two tasks have been extensively researched and understood. Open-source libraries such as COLMAP [1] are able to perform per-frame camera pose estimation and dense reconstruction of a point cloud given an input sequence of images observing a scene. On the other hand, 2D semantic segmentation is a well-understood problem with data-driven methods such as Mask-RCNN [2] and more recently Segment Anything (SAM) [3] from the FAIR team at Meta provide extremely precise and robust segmentation masks for images of highly complex scenes.

However, there is yet to exist a universal framework for combining the reconstructed scene mesh with the segmented 2D observations to produce a complete 3D semantic representation of a given scene. Semantic reconstruction has many useful applications in virtual reality systems and mobile robotics setups including self-driving cars, so it is important that a robust and accurate system for semantic reconstruction be developed.

To this end, we propose and explore a pipeline using neural radiance fields (NeRFs) for rendering novel camera trajectories. The rendered views are combined with a method for producing a temporally consistent semantic segmentation of the image sequence using SAM, which is projected onto a 3D mesh of the scene through a majority vote mechanism. The code implemented for our project along with additional results and visualizations outside of this report can be found at `https://github.umn.edu/diaz0329/CSCI5561-G6-Final-Project`.

2 Related Works

2.1 Methods for Semantic Reconstruction

2.1.1 Structure from Motion and Multi-view Stereo

The baseline method for semantic reconstruction begins by applying Structure from Motion (SfM) to a sequence of RGB images. Feature extraction and matching are used to determine camera poses and generate a sparse point cloud. Multi-view stereo (MVS) is then employed to reconstruct surfaces and generate a dense point cloud. Various works offer different implementations of this general pipeline. COLMAP is a well-known solution that improves completeness through methods such as iterative bundle adjustment [1] and pixelwise view selection [4]. Similar software such as Agisoft Metashape and 3DF Zephyr can achieve superior accuracy to COLMAP in some contexts [5]. Our project utilizes COLMAP-SfM, but aims to use NeRFs for dense reconstruction instead of MVS.

2.1.2 iMAP: Implicit Mapping and Positioning in Real-Time

iMAP is an MLP with 4 hidden layers of feature size 256, and two output heads that map a 3D coordinate $p = (x, y, z)$ to a colour and volume density value: $F_\theta(p) = (\mathbf{c}, \rho)$. Instead of trigonometric embedding, it uses the Gaussian positional embedding proposed in Fourier Feature Networks [**?**] to lift the input 3D coordinate into n-dimensional space: $\sin(Bp)$, with B an [n × 3] matrix sampled from a normal distribution with standard deviation σ. Given a camera pose T_{WC} and a pixel coordinate [u, v], camera pose is back-projected to an normalised viewing direction and transformed it into world coordinates $\boldsymbol{r} = T_{WC}K^{-1}[u, v]$, where K is camera's parameters. Then, N random samples (our random initialization of weights) are taken along the ray $\boldsymbol{p_i} = d_i\boldsymbol{r}$ with corresponding depth values $d_1, , d_N$, and query the network for a colour and volume density $(\mathbf{c}_i, \rho_i) = F_\theta(p_i)$. Weight is the ray termination probability at distance $\delta_i = d_{i+1} - d_i$ is defined as $w_i = o_i\Pi_{i-1}^{j=1}(1 - o_j)$, where $o_i = 1 - \exp(-\rho_i\delta_i)$ is an activation function. Depth and colour are calculated as mathematical expectations.

iMAP selects a number of pixels, based on loss statistics, from keyframes, selected based on difference between them and previous keyframes in terms of information gain, and optimizes a loss on them, a loss that consists of geometric and photometric losses, which we minimize with respect to implicit scene network parameters θ which direct the sampling along the ray, and camera poses.

In general, NeRFs are perfect for rendering 3D surfaces on pre-collected datasets and when enough computational power is available, however, in some cases there arises a need to reconstruct a surface without pre-training and on a device with a limited computational ability. Here is where iMAP [6] comes to help: an MLP with a few fully connected layers, it uses Fourier Features [7] for information encoding and requires no pre-training. It jointly optimizes camera poses with rendering, although, for simplicity, it axes angles ϕ and θ of ray looking on pixel. While it was supposed to be able to run on edge devices, computational power of an average notebook is clearly not enough to run it. As iMAP experiments did not give impressive and positive results, we switched to other methods.

2.2 NeRFs for Novel View Synthesis

Neural radiance fields (NeRFs) [8] have garnered much attention over the past few years for its impressive ability to render realistic novel views of a scene encoded in a fairly simple model such as a multilayer perceptron. Specifically, the model learns a *volumetric scene function* (VSF) $F_\Theta(x, y, z, \theta, \phi)$ over the parameters Θ, where (x, y, z) denotes the position in 3D space of the point to render and (θ, ϕ) denotes the view angle of that point, and outputs a volume density σ and radiance (R, G, B). Novel views are generated by sampling VSF values along rays projected from the camera, and employing classical volume-rendering techniques to compose the sampled values into the 2D render. Thus, the VSF serves as an implicit 3D representation of the scene, since it can theoretically be queried to provide any arbitrary 2D view of the scene.

2.2.1 NeRFusion

NeRFusion aims at overcoming the limitations poised by the MLP architecture of the NeRFs: namely, their inability to capture large scens. It retreats to the capabilities of a TSDFs (Truncated Signed Difference Function) and an RNN to produce per-voxel sparse neural representations and local, in terms of a few neighboring frames, features of the objects as rendered, resepctively. For each frame, local features are extracted by a 2D convolutional network. To make local features volumes temporally consistent, the authors use a GRU with a 3D to fuse them into a global representation, simultaneously pruning voxels whose features contribute very little to the global rendered volume.

2.2.2 BARF

BARF stands for Bundle Adjusting Radiance fields, and rests upon the idea of combining NeRFs with the technique of Bundle Adjustment (BA) and to apply a smooth mask on the encoding at different frequency bands (from low to high) over the course of optimization, which acts like a dynamic low-pass dynamic filter. Thus we are doing BA on many levels, where each level of coarseness/fine-ness of encoding has an associated weight, and by manipulating these weights we can adjust the contribution of feature levels to the final outlook. We also learn the weights of these frequency components to produce more realistic outlook.

2.3 Semantic Segmentation

Semantic segmentation of an image involves associating every pixel in an image to a class or a label. It has various applications in object and scene recognition tasks. One of the most well-known algorithms in segmentation is Mask-RCNN [2]. Mask-RCNN utilizes a Feature-Pyramid Network and a ResNet101 backbone. It comes with a pre-trained model for the COCO dataset. Another popular model is the Segment Anything Model (SAM) [9] which boasts zero-shot instance segmentation on any given image, without needing any pre-training.

3 Methods

3.1 Novel View Rendering with NeRFs

Given a training sequence of image observations of a scene and a handcrafted camera trajectory, we train a NeRF to learn an implicit scene representation off of the image observations, and render novel views following the given camera trajectory for segmentation.

3.1.1 Training a NeRF

We train and evaluate our NeRFs on four scenes from the open-sourced Replica [10] dataset: `room_0`, `room_1`, `office_1`, and `office_2`. A training set of 500 images is provided for each scene, with the total training set having relatively thorough visual coverage of the objects within. We utilize COLMAP [1] to estimate the camera pose for each training view, and use the images and their corresponding camera poses as input to the NeRF model for training.

3.1.2 Camera Trajectories for Novel View Rendering

After the scene-specific NeRF is trained, we develop a camera trajectory, represented by a sequence of camera transformations, along which novel views will be rendered. Our novel-view camera trajectory is parameterized by (r, ϕ, n), and for $0 \leq i < n$, the ith camera transformation along the trajectory is given by $C_i = \begin{bmatrix} R_i & t_i \end{bmatrix}$, where

$$t_i = \bar{t} + \begin{bmatrix} \bar{r}\cos\left(\frac{2\pi}{ni}\right) & \bar{r}\sin\left(\frac{2\pi}{ni}\right) & 0 \end{bmatrix}^T$$

$$R_i = R_\gamma\left(\frac{2\pi}{ni} + \frac{\pi}{2}\right) R_\phi(\phi)$$

and R_γ and R_ϕ are the camera yaw and pitch rotation matrices respectively. Here, $\bar{t}$ represents the average camera position over all training observations (corresponding to the center of the scene), and $\bar{r}$ represents the average distance from all of the training observations to the scene center, scaled by the parameter r. Essentially, the camera rotates around the center of the scene in a circle of radius $\bar{r}$ looking down with an angle of ϕ, capturing a smooth 360° view of the scene. This simple trajectory allows for ample coverage of the scene, and also allows for objects to shift very little between each frame, which will aid our strategy for creating a temporally consistent 2D segmentation.

3.2 Semantic Segmentation with Segment Anything

We preferred to use SAM over Mask-RCNN as there is no need to pre-train Segment Anything for zero-shot instance segmentation. However, since SAM only provides instance segmentation, our class labels are inconsistent between multiple frames of our data. To overcome this, we devised a simple threshold—mIoU heuristic to enforce temporal consistency between multiple frames of segmentation. We make two assumptions: images are in sequentional order and that the camera pose does not drastically change. Thanks to our NeRF approach, we are able to generate novel views in sequence to satisfy both of these assumptions.

Given an image t_0 we use SAM to obtain n binary masks in the scene. This number is determined by the automatic mask generator in SAM and can be influenced by some of the hyperparameters. We obtain a segmentation map for t_0 by constructing a 2D array with segmentation ids assigned based on the output binary masks. We then use SAM to obtain binary masks for image t_1. We compare these binary masks to the segmentation map from t_0 and assign the segmentation id that has the highest mIoU with the mask. To account for new objects coming into the scene, we also compare the highest mIoU with a hyperparameter called threshold. If the highest mIoU is not greater than the threshold, we consider this binary masks to be a new object, assigning it a brand new segmentation id. Thus, objects that overlap between frames above a certain threshold will maintain the same segmentation id throughout the frames.

3.3 3D Semantic Reconstruction

Using SAM with our threshold-mIoU heuristic, we obtain segmentations for novel views rendered by the NeRF. We then project these segmentations using the camera poses obtained from NeRF onto the ground truth mesh. For each point in the mesh, we use a majority vote it assign the point with the most frequent segmentation id from all the segmentations that project to that point.

4 Experiments and Results

4.1 Comparison of NeRF Models

To compare the ability of various NeRF models to learn an implicit 3D representation of a complex indoor scene, we conduct a quantitative and a qualitative experiment. From the many NeRF variants out there in the current ecosystem, we elect to explore three: Instant-NGP (INGP) [11] for its advertised lightning-fast training speed, Segment Anything in 3D with NeRFs (SA3D) [3] due to its adaptation for larger-scale unbounded scenes, and TensoRF [12] which serves as the NeRF backbone for SA3D.

Our quantitative experiment consists of evaluating the mean peak signal-to-noise ratio (PSNR) across the first 100 views of the given training image sequence for the four aforementioned Replica scenes. Results and a short analysis can be found in Table 1.

Our qualitative experiment consists of generating 250 novel views for each scene following our novel-view camera trajectory and evaluating the reconstructed image quality. Results and a short analysis can be found in Figure 1.

4.2 Semantic Segmentation of Image Sequence

We try out various thresholds for evaluating SAM with threshold-mIoU heuristic. Due to the excessive run-time of SAM per image and the time constraints at hand, we only tested

	room_0	**room_1**	**office_1**	**office_2**
INGP	23.09	25.52	27.16	20.26
SA3D	**32.86**	**30.72**	20.87	**31.45**
TensoRF	28.26	27.98	**36.04**	27.79

Table 1: Mean PSNR values for NeRF variants evaluated on four scenes over 100 views. In alignment with qualitative observations, INGP performs the worst overall, while SA3D produces the highest-quality renders in most of the scenes.

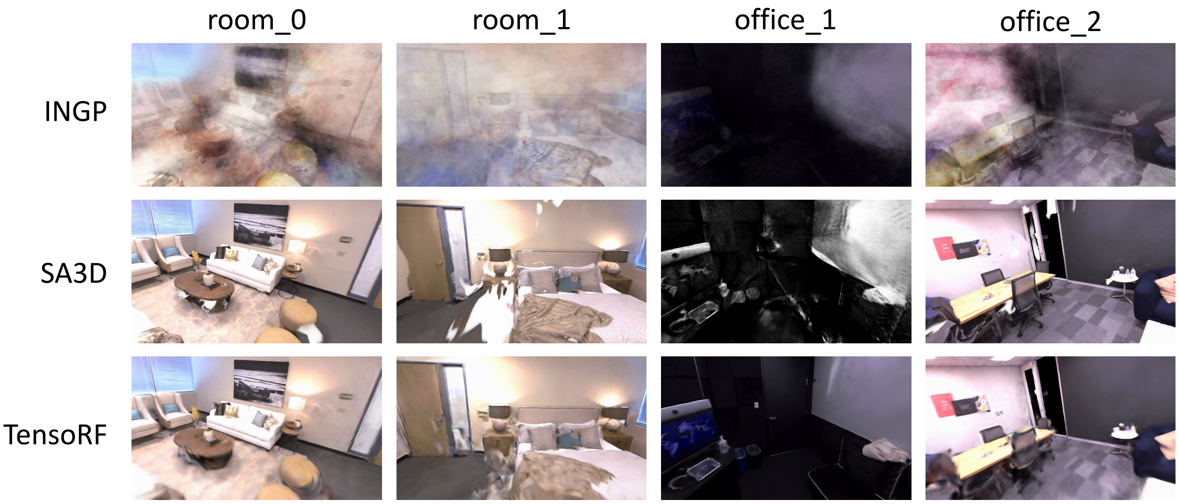

Figure 1: Select renders of novel-view camera trajectory for NeRF variants evaluated on four scenes. INGP produces blurry renders that sometimes make the scene unrecognizable, and thus performs the least well qualitatively on all four scenes. SA3D and TensoRF produce renders of comparable quality, but in spots where SA3D produces image tears in places that are unexplored by the original camera, TensoRF compensates with additional noise in that region.

two thresholds, 0.3 and 0.1, on 20 images from our dataset. We determined that the 0.1 threshold is much more consistent and provides more leniency for smaller objects in the frame. These results can be found in Figure 1.

We also compared the performance of our threshold-mIoU SAM with ground truth segmentations. We ran our model on 20 images from all 8 scenes available in our dataset. We use mIoU as our metric for accuracy. Keep in mind that the number of segments provided by SAM is usually ten times higher than the number of objects in scene – most likely due to not tuning SAM's hyperparameters for our data. Thus, we find the most relevant mask to the corresponding ground truth to perform mIoU on. We find the average mIoU to be around 58%.

One reason for lower accuracy could be due to oversegmentation of the objects. Another reason is that SAM may not be consistent when used on the same frame twice.

scene	mIoU
office_0	53.6%
office_1	56.7%
office_2	49.6%
office_3	57.1%
office_4	60.4%
room_0	66.1%
room_1	66.1%
room_2	61.3%
Avg. mIoU	58.9%

Table 2: Average intersection over union for 20 images of 5 office and 3 room scenes. The average of all these intersection over union is presented in the last row of the table.

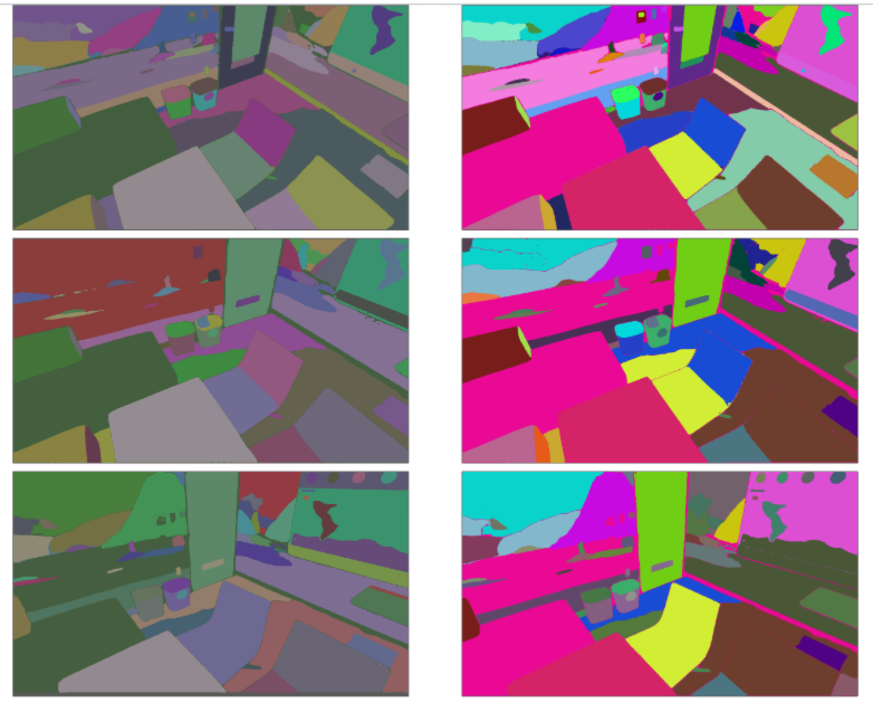

Figure 2: Segmentations from office_0 scene using our threshold-mIoU SAM with a threshold of 0.3 (left column) and 0.1 (right column).

5 Conclusion

This work proposes applying NeRFs in conjunction with SfM to conduct 3D semantic reconstruction. NeRFs produced implicit scene representations to accomplish 3D reconstruction. SA3D provided the strongest performance among the three NeRF varieties evaluated. SAM was used to perform 2D segmentation with moderate success in enforcing temporal consistency across frames. Finally, we were not able to complete the projection of 2D segmentation into 3D. Reconciling 3D reconstruction with 2D segmentation would be the first priority of future research. Tuning the parameters of SAM could also be explored to improve the accuracy of the 2D segmentation.

Some ways to improve the 2D segmentation is to look into other version of SAM such as integrating threshold-mIoU with FastSAM[13] to reduce the amount of time needed for obtaining segmentations. Another paper to explore is the DEVA: Track Anything [14] is a paper that utilizes Grounded-SAM for providing temporal consistent semantic segmentations.

6 Summary of Contributions

Athreyi: I implemented threshold-mIoU SAM for semantic segmentation. I also conducted literature reviews on other semantic segmentation models such as Mask_RCNN, and DEVA: Track Anything. I conducted qualitative and quantitative analysis for threshold-mIoU SAM. This project allowed me to explore the differences and challenges in creating a temporal consistent segmentation algorithm. I usually have projects where I am required to implement an architecture from scratch, or use existing models. However, this project gave me the opportunity to take an existing academic work and add functionality to it. I am definitely interested in exploring how to improve the results I received from threshold-mIoU SAM.

Ryan: I got up and running and adapted the three main NeRF variants used in this project: Instant-NGP, Segment Anything in 3D, and TensoRF. This involved setting up the appropriate environments to run the code, as well as adapting the training and evaluation code for the provided Replica dataset. I developed and implemented our novel-view camera trajectory for rendering scenes from NeRF, and conducted the qualitative and quantitative comparisons for the three NeRF variants. This project taught me *many* things when it came to working with emerging technologies in computer vision and computer science in general. I learned a lot about how NeRFs worked, how they could be used, and how they could be adapted to fill specific niches or needs. I learned (or at least was reaffirmed) that projects won't always look the same from start to finish, and that one should adapt to the evolving state of the project. Above all, I learned not just how to write code but to *read* it; it's one of the most important skills a computer scientist should have, and adapting the three main NeRF repos has taught me a lot about how to do it.

Ivan: Literature review, extensive evaluation, testing and modification of NeRFs, including, but not limited to: BARF, iMAP, NICE-SLAM (the NeRF part of it), vanilla NeRF. Auxiliary, non-primary role in testing and adjusting SAM. I learned that planning and early execution of the plan is one of the few most essential aspects of a success of a project, because stumbling blocks are hard to predict. The hardest part of any project is not creation of the tool, and not its conceptualization, but its integration and creating a pipeline with existing data, removing platform incompatibilities, converting data between formats etc.

Arthur: I worked on adapting Segment Anything in 3D, which proved to be the most effective way to train the neural radiance fields for semantic reconstruction. I conducted much of the group's literature review. This helped me provide insight into which avenues of research to pursue. I prepared the content and layout for most of our final presentation. I learned how heavily academic research relies on previously published work. I already understood this in the abstract, but I had grown accustomed to writing code from the ground up in my previous computer science classes. This wasn't feasible for a problem as complex as semantic reconstruction. I had to shift my focus from figuring out how to solve the problem independently to finding opportunities to improve existing solutions.

References

[1] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[3] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Chen Yang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs. In *NeurIPS*, 2023.

[4] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[5] A. H. Qureshi, W. S. Alaloul, A. Murtiyoso, S. Saad, and B. Manzoor. Comparison of photogrammetry tools considering rebar progress recognition. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2022:141–146, 2022.

[6] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew Davison. iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[7] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *CoRR*, abs/2006.10739, 2020.

[8] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[9] Segment anything.

[10] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis

Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[11] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022.

[12] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022.

[13] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything, 2023.

[14] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *ICCV*, 2023.