Online Semantic Scene Representations Using Neural Fields

Sergio Orozco, Benjamin Spiegel, Xinqian Zhou
Department of Computer Science
Brown University
Providence, RI

Abstract

In the field of robotics, strong perception systems that capture visual and meaningful information about scenes would prove to be useful for many downstream tasks. Furthermore, these scene representations must be generated online in a matter of minutes, as opposed to hours or days. For our final project, we present Online Semantic Scene Representations Using Neural Fields, a method for generating semantic and photometric scene representations that leverages multi-resolution hash encodings to significantly speed up training over previous methods.

1. Introduction

For robots to serve a practical purpose, it is essential that they are able to comprehend their surroundings. This necessitates not only the ability to locate themselves within a given environment and identify distinct objects or individuals but also to have a deeper understanding of the embedded geometry and semantic information present in the scene. This level of understanding forms the basis for a general perception system that can facilitate a broad array of downstream tasks, including object retrieval, manipulation, and scene reorganization, as well as enabling methods for controlling the overall safety of the robotic system by granting it the ability to label parts of the scene as humans, for example.

Furthermore, for robots to be generally useful in novel environments, they must be able to construct geometrically and semantically accurate scene representations on the fly. For example, in a military setting, a robot might need to run or fly through a scene quickly and construct a map of the environment. This task entails avoiding collisions, so being able to quickly construct scene representations is crucial. While traditional neural field methods that do semantic segmentation can require days of training, we adapt the latest advancements and integrate multi-resolution hash encoding into our model, enabling us to learn novel scene representations in a matter of minutes.

2. Background and Related Work

2.1. Volume Rendering and Neural Radiance Fields

At its core, our model is an extension of the Neural Radiance Field (NeRF) model introduced by [3], which is capable of synthesizing novel views of complex scenes given a large data set of images and corresponding camera poses. The method models the radiance field of a scene using a multi-layer perception (MLP), which is then queried using a differentiable volume-rendering method to output pixels in a novel view image. Formally, NeRFs represent scenes as 5D functions F_Θ, parameterized by a 3D spatial coordinate $\mathbf{x} = (x, y, z)$ and a 2D viewing direction (θ, ϕ). The output of the MLP is an emitted color $\mathbf{c} = (r, g, b)$ and volume density σ.

Rendering a scene using its radiance field then involves accumulating the colors emitted along a ray shot through the scene from each pixel of a novel view image. Formally, the expected color of a pixel is the expected color $C(\mathbf{r})$ of a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$:

$$C(\mathbf{r}) = \int T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \quad \text{where } T(t) = \exp\left(-\int \sigma(\mathbf{r}(s))ds\right). \tag{1}$$

Though, this integral is often approximated using a quadrature during the volume rendering process for practical purposes.

2.2. Multiresolution Hash Encoding

While NeRFs provide a remarkable solution for the problem of novel view synthesis, one major drawback of the work is the vast compute and time required to train the parameters of the radiance field. [4] address this problem with their introduction of multiresolution hash encoding, which enables a massive speedup and decrease in the number of overall parameters. Their method, InstantNGP, works by encoding the input coordinates to the radiance field as a vector $\mathcal{E}$ by linearly interpolating the output of an array of

pseudo-hash-tables, each corresponding to a different level of granularity. They then pass $\mathcal{E}$ to an MLP, which outputs a color and volume density. InstantNGP is a drop-in replacement for the large MLPs used in the traditional NeRF paper.

2.3. Semantic Scene Understanding

The methods we have discussed so far are concerned with capturing a photometric representation of a scene, but our application requires a scene representation that contains meaningful information about the items inside it. Semantic-NeRF [9] achieves this by augmenting the classic NeRF architecture with an additional semantic head appended to the end of the radiance field MLP for predicting the logits for a distribution over C semantic labels. The network is trained in a similar fashion to the original NeRF, but with an additional cross-entropy loss term.

2.4. Pre-trained Vision and Language Models

A common strategy for collecting data to train semantically-informed neural field models is to leverage pre-trained vision and language models (VLMs) to segment and label training images with object classes. Models like CLIP [5] are good candidates for image labeling due to them being trained on massive image and text label data sets, as it is likely that they have already seen most items in any given scene. For panoptic segmentation, many existing models such as Detectron 2 [8] and Segment Anything [2] have likewise been trained on massive data sets of segmented images, and are capable of performing this task at fine levels of granularity and accuracy.

3. Online Semantic Scene Representations using Neural Fields

We present Online Semantic Scene Representations using Neural Fields, a method for generating novel photometric and semantic views of scenes online from limited training data. Our method is a natural extension of Semantic-NeRF, which can segment and semantically label 3D scenes, though with the addition of multi-resolution hash encodings as seen in Figure 1 to increase training speed.

3.1. Data Collection

To generate data for our model we collected X images of the scene using the camera on the Spot arm. We then estimated their poses using Colmap [6], as we originally tried calculating their pose using the odometry of the Spot arm but found the translation from the spot coordinate system to that required by NeRF to be too difficult. In order to generate semantic labels for the data, we originally deployed Detectron 2, which is capable of off the shelf panoptic segmentation and labeling, but we found the background labels

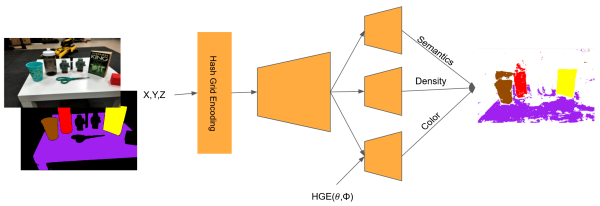

Figure 1: Outline of Model Architecture. A set of positional coordinates are fed into the network in order to estimate the semantics, density, and color of that specific point in 3D space. The viewing direction is attached only to the Color portion of the network.

to be too noisy. We replaced this with a two-stage pipeline of Segment Anything [2] for panoptic segmentation, and CLIP [5] for labeling the segments. The result was a dataset of X RGB+S (S for semantic labels) images that we trained our model on.

3.2. Model Architecture

Our semantic scene model makes use of multiple MLP heads, each equipped with different objectives to implicitly represent appearance, geometry, and semantics. As written in [9], the expected semantic logits $\hat{S}(r)$ of a given pixel can be written as:

$$\hat{S}(r) = \sum_{k=1}^{K} \hat{T}(t_k)\alpha(\sigma(t_k)\delta_k s(t_k)$$

The resulting semantic logits can then be transformed into multi-class probabilities through a softmax normalization layer.

3.3. Training

The network can be trained from randomized weights under the photometric and semantic loss functions, L_p and L_s, respectively:

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

$$L_s = -\sum_{\mathbf{r}\in\mathcal{R}} \left[\phi \sum_{l=1}^{L} p^l(\mathbf{r}) \log \hat{p}^l(\mathbf{r})\right], \text{where} \phi = \begin{cases} 0.5, & \text{if } p^0(r) = 1 \\ 1, & \text{otherwise} \end{cases}$$

The additional ϕ term scales down the loss for NULL labels generated by our automated semantic segmentation pipeline during training. Unlike [9], this simple augmentation of the semantic cross-entropy loss function allows our model to not only learn with sparse data, but it also affords the model the ability to assign 3D points the NULL label if

Figure 2: Qualitative results for automated semantic segmentation using CLIP and Segment Anything. From left to right are input images and resulting segmentations.

no data has been seen by the model yet. This addition allows for possible future work in frontier exploration using neural fields.

The final loss can then simply be calculated as the weighted sum of L_s and L_p:

$$L = \lambda L_s + L_p$$

where λ simply acts as a weighted scalar to reduce the magnitude of loss seen in the semantic head. Though [9] suggests that this value does not affect performance, we found that the weight performs best at a value set to 0.001. Anything higher caused the network to diverge, and anything lower caused it to converge too slowly.

3.4. Implementation

A semantic scene is trained from randomized weights using both input rgb and semantic images. We implemented our model using Pytorch for the network construction and CUDA for the forward/backward ray marching of the network. The model was trained on a single RTX 3080-Ti GPU with 11GB of memory. The batch size of rays was set to 4095 rays in order to prevent OOM exceptions. We trained the network using the Adam optimizer [1] with a learning rate of 1e-2 and a decay rate of 5e-5.

4. Results

In addition to our own sparse data collection method, we made use of Replica [7], a reconstruction-based 3D data set of 18 scenes with geometry, HDR textures, and semantic annotations. Though out of scope for this project, results for this data set were added in order to demonstrate the effectiveness of our model with fully labeled scenes.

4.1. Automated Segmentation Pipeline

Although the pipeline we constructed is an automated process with regards to the fact that no human need to manually add segmentations to the input images, it would be disingenuous to state the process was without human intervention altogether. The labels from which the segmented images are constructed are still hand specified by human operators. The labels used to capture Figure 2 are as follows:

- ”scissors”, ”toy robot”, ”water bottle”, ”book”, ”table”, ”cup”, ”floor”, ”marker”

Figure 2 shows the qualitative results of our automated segmentation pipeline. It is clear that the implemented method frequently misclassifies objects as NULL. This sparseness in data, which varies from image to image, can be partially remedied by the altered semantic loss function detailed in Section 3.3.

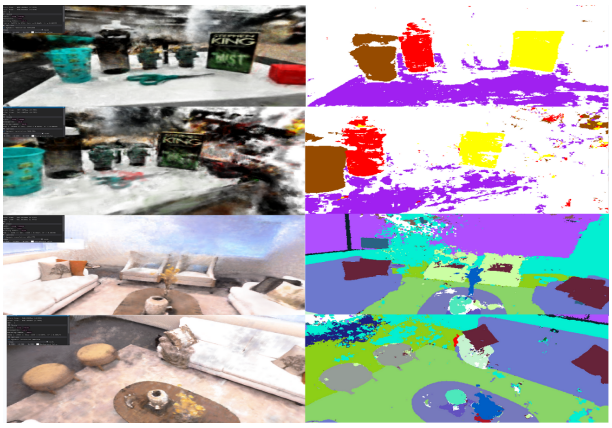

Figure 3: Qualitative results for online semantic scene representation. From left to right are rendered photometric and semantic images from the model. From top to bottom are renderings from our automated data capture and Replica.

4.2. Online Scene Representation

While our method is able to learn semantic scenes in only a matter of minutes, we were faced with a slight issue of artifacting. This erroneous behavior in our model can most likely be attributed to our camera poses being generated by Colmap [6]. Although the general-purpose Structure-From-Motion pipeline can estimate camera poses, it will never beat camera pose estimation captured from odometry data. Although our original intention was to use the on-board sensors available from Spot, we simply could not accurately find a one-to-one correspondence between the robot's coordinate and units system to that required by the Torch-Ngp code base.

Figure 3 demonstrates the effectiveness of our online method even when input data is inconsistent. Additionally, the model is capable of labeling areas as NULL when it doesn't know what is there. This is a simple, yet crucial difference between our method and [9]. It is intuitive to

think that if a robot were to plan within a given state representation, it would be preferred that the robot concede it's inability to detect an object rather than be confident in false positives.

5. Limitations

Though our work demonstrates our ability to train Semantic-NeRF in real-time, there are obviously many issues. First and foremost, our data segmentation and labeling pipeline is not very good. Preferably, we would want an automated labeling tool that could achieve panoptic segmentation, rather than simply segmenting a predefined set of labels specified by humans. Additionally, our model is riddled with artifacts. We believe the root cause of this issue to be Colmap; however, we cannot know this definitively without further testing. Lastly, we did not include any quantitative results of our method; therefore, there is no way to know for certain if our model is comparable to Semantic-NeRF. This is something we wish to do if this were to become a publishable workshop paper, for instance. In place of quantitative results, we have simply added more qualitative results for viewing in Figure 4.

6. Conclusion

The ability of robots to comprehend their surroundings is a critical aspect of their practical use in a wide range of applications. Such comprehension includes not only the ability to locate and identify objects but also to understand the embedded geometry and semantic information present in a scene. This understanding forms the foundation of a general perception system, which can facilitate a wide range of downstream tasks, including object retrieval, manipulation, and scene reorganization. By combining Segment Anything and CLIP as a pre-processing method, NeRFs with Multiresolution Hash Encoding, our model can now construct geometrically and semantically accurate scene representations in just a matter of minutes.

Acknowledgments

Thank you to Professor Srinath Sridhar and Rugved Mavidipalli for their expertise and advice, which made this project possible.

Contributions

- Automated data collection pipeline - Ben, Sergio
- Spot movement and pose configuration - Sergio, Xinqian
- Model Architecture - Sergio

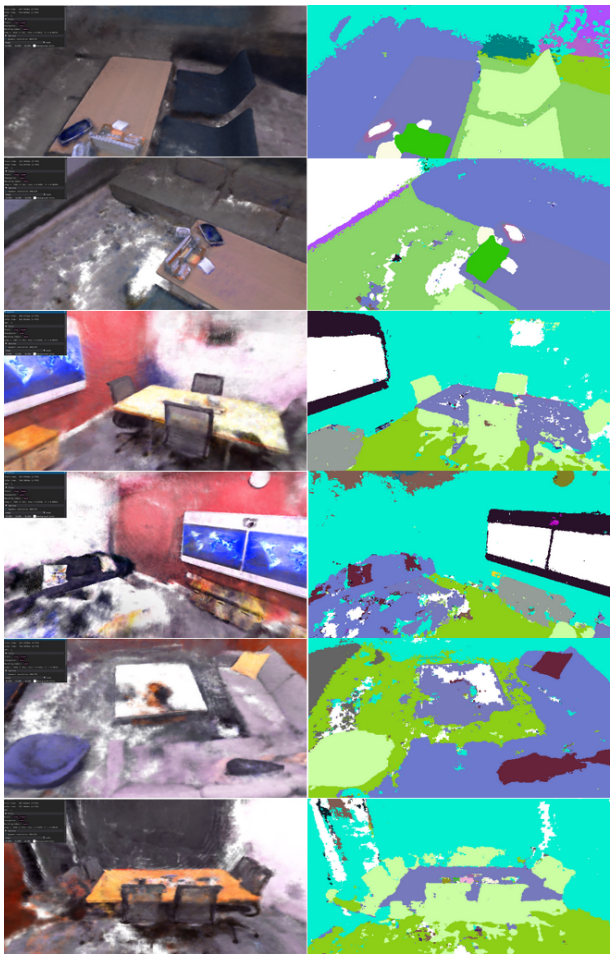

Figure 4: Photo dump of qualitative results. Input images and segmentations for training were provided by the Replica data set, and camera poses were generated via Colmap.

References

[1] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 3

[2] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2

[3] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1

[4] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 1

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learn-

ing transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[6] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3

[7] 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. 3

[8] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 2

[9] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 2, 3