Language-Guided Visual Navigation in a Neural Radiance World

Yinpei Dai
Umich EECS
daiyp@umich.edu

Onur Bagoren
Umich Robotics
obagoren@umich.edu

Abstract

We present an visual-language guided end-to-end trajectory optimization framework that operates on a Neural Radiance Field (NeRF) representation of the world. The proposed method utilizes visual and language based prompting for waypoint generation, and uses prior work on probabilistic trajectory optimization on NeRFs for generating a continuous path. We perform experimental validation in both simulation and real world navigation environments and show that our method is able to produce both safe and correct paths.

1. Introduction

Human-robot interaction (HRI) becomes more prevalent as a research topic as we find more daily tasks automated and completed by mobile robotic platforms. It's common to use robots for purposes varying from completing daily tasks such as vacuuming our floors to automating production lines in factory settings. This integration of robots into everyday life has led to the important research area of HRI, where we study ways to improve the ease and quality of robot involvement in human tasks.

In this project, we specifically address the problem of using natural language communication between a person and a robot operating in real-world, indoor environments. We specifically utilize recent advancements in natural language processing (NLP), along with visual representations of real-world environments, to propose a method that can guide the navigation of robots in such environments. We propose that our method fuse these language instructions that can come from human prompts with a robot planning a path in a Neural Radiance Field (NeRF) representation of the world.

Neural Radiance Fields have gained increasing interest in recent years due to their impressive ability to encode detailed 3D geometry and color compactly. With the innate continuous representations, NeRFs are well-suited for robot motion planning using gradient-based methods, serving as implicit density fields to dynamically plan collision-free trajectories for the mobile robot navigation [1]. Such a fully-differentiable method has great potential for large-scale navigation optimization and pushes the performance limit of traditional planning algorithms [2]. However, previous work has not considered more fined-gained motion planning conditioned on language instructions with NeRFs.

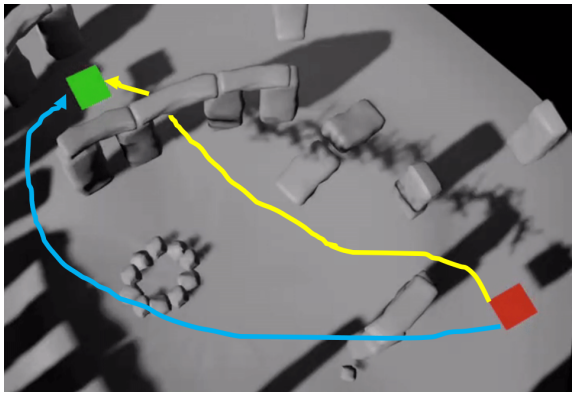


Figure 1. Given a 3D occupancy map reconstructed from a NeRF, a robot drone started from the red square is tasked to navigate to the green square through different paths given language instructions. The yellow line indicates a suitable path conditioned on "go to the goal by traversing the door". The blue line indicates a suitable path conditioned on "move to the goal by taking a detour from the left side". Image adapted from [1]

In this work, we present an end-to-end framework for trajectory optimization based on a NeRF world representation and diverse language instructions so that the generated navigation paths can vary according to different language guidance, as shown in Figure 1. Such language-guided robot navigation is useful for developing robots that support human-robot language interactions in the future. We specifically study the ability of natural language guidance to understand directional instructions, such as **"Plan a path from the left of the objects"**.

2. Related Work

NeRF-based Navigation aims to apply the neural implicit representations to the robotic navigation tasks. One line of the work is to estimate the agent state by jointly optimizing the pose and the NeRF models for rendering the envi-

ronment pictures [5, 7, 9] and construct the map simultaneously [12, 14, 19]. Another line of work is to plan suitable navigation trajectory based on the NeRF representations, for example, [1] preprocess the density output of the NeRF into an occupancy map to plan collision-free trajectory via an MPC controller, [2] build fully differentiable end-to-end trajectory planner based on the NeRF representations and perform the SLAM at the same time.

Language-guided planning in robot navigation has been well studied. Popular methods include: 1) using imitation and reinforcement learning to train an embodied agent [6] and 2) applying Large-Language Models (LLMs) to generate codes to execute low-level actions [4]. To the best of our knowledge, none of the previous work applies language-guided planning to NeRF-based robot navigation.

3. Methodology

3.1. Neural Radiance Fields

For an input 3D location $\mathbf{x} = (x, y, z)$ and viewing direction $\mathbf{d} = (\theta, \phi)$, Neural Radiance Fields (NeRFs) return an emitted color $\mathbf{c} = (r, g, b)$ and volume density σ. The composition of the emitted color and volume density form a continuous scene representation, which is typically performed via a network $\mathbf{F}_\theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$.

A discrete approximation of the traditional volumetric rendering method renders color from rays cast through the 3D space. For a given epipolar ray r, the numerical integration process is performed as shown in equation (1), where $\hat{C}(r)$ is the estimated pixel color, σ_i the MLP-computed color density at the i-th sample, $\mathbf{c}_i$ the MLP-computed color value at the i-th sample, and δ_i represents the distance between sample i and its adjacent sample.

$$\hat{C}(r) = \sum_{i=1}^{N} T_i \left(1 - \exp\left(-\sigma_i \delta_i\right)\right) \mathbf{c}_i \tag{1}$$

$$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) \tag{2}$$

The model parameters are then optimized by computing the mean-squared error (MSE) loss between the rendered pixels and those in the ground truth image.

For our implementation, we use a NeRF constructed from a multiresolution hash encoding as input into the model [10, 11] for fast optimization and rendering.

3.2. NeRF-based Navigation

We use the navigation method in [1] to plan a path for the robot to follow in a NeRF representation of the world.

3.2.1 Trajectory Planning in NeRF

The trajectory optimization formulation is proposed as an unconstrained optimization problem, where the cost function is composed of two terms: a collision penalty and a control penalty. This cost function is shown in equation (3), where $\sigma(\cdot)$ denotes the density of the NeRF at a 3D position, $\mathbf{u}$ the control, and $(\mathbf{R}, \mathbf{p})$ denote the orientation and position of the robot at time τ over the optimization horizon h. It is important to point out that the points considered as part of the collision penalty are denoted as the set $\mathcal{B}$, which represents the 3D points defining the robot's bounding box.

$$J = \sum_{\tau=0}^{h} \overbrace{\sum_{\mathbf{b_i} \in \mathcal{B}} \sigma\left(\mathbf{R}_\tau \mathbf{b}_i + \mathbf{p}_\tau\right) s\left(\mathbf{b}_i\right)}^{\text{collision penalty}} + \overbrace{\mathbf{u}_\tau^T \Gamma \mathbf{u}_\tau}^{\text{control penalty}} \tag{3}$$

Minimizing this cost function gives us a set of waypoints $W = \{w_0, \ldots, w_h\}$ for the robot to navigate to over its planning horizon. The optimization is performed using gradient descent.

3.2.2 State Estimation in NeRF

A probabilistic state estimation is performed with an optimization-based approach. The state estimator estimates the state as a Gaussian $\mathbf{x}_t \sim \mathcal{N}(\mu_t, \Sigma_t)$ at time t. The estimator takes the previous action $\mathbf{u}_t$, previous state $\mathbf{x}_{t-1}$ and image I_t as inputs, and estimates $\mathbf{x}_t$.

Additionally, the estimator uses a model of the robot dynamics to produce a predicted state that gets updated with the measurement of image I_t, such that $\mathbf{x}_{t|t-1} = f(\mathbf{x}_{t-1}) \sim \mathcal{N}(\mu_{t|t-1}, \Sigma_{t|t-1})$.

For the vision-based state estimation, feature-based photometric consistency is used. Initially, a set of pixels $\mathcal{J} \in I_t$ are extracted using ORB [13] features. These pixels are then compared against the NeRF rendered image at the pose $\mathbf{T}_t$, $C_\mathcal{J}(\mathbf{T}_t)$ and used to minimize the objective function shown in equation (4). Here, $\mathbf{S}_t$ is the measurement model noise covariance, and $\|x\|_M^2 = x^T M x$.

$$J = \|C_\mathcal{J}(\mathbf{T}_t) - I_t(\mathcal{J})\|_{\mathbf{S}_t}^2 + \|\mu_{t|t-1} - \mu_t\|_{\Sigma_{t|t-1}^{-1}}^2 \tag{4}$$

The minimization of this cost function is with respect to the pose $\mathbf{T}_t \in SE(3)$, which is optimized for using an auto-differentiating library LieTorch [15].

3.3. Language-guided NeRF-based Navigation

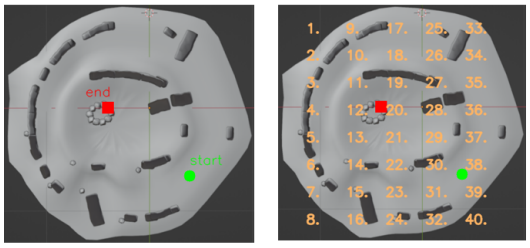
In this work, due to the lack of training data, we use LLM/VLM to perform zero-shot trajectory planning by prompting them to predict an intermediate waypoint between the start and end positions based on language instructions, to implement language-guided navigation. Our

You are supposed to give a robot intermediate waypoint to plan its trajectory correctly following the language instructions. You will be given the start position and the end position, denoted as [X, Y, Z], then you should give a guess about the middle waypoint [X, Y, Z] based on the language description.

Note:
1. The coordinate scope is bounded into -1.0 and 1.0.
2. You are looking at towards the y positive axis from a bird-view.
3. Although you can not see, but you can make a reasonable guess based on geometry. Imaging there is a stone in front of you at the start position, you may need to plan left or right a little bit around the start position as the waypoint according to the task.

Here is an example:
Q: The start position is [0.52, -0.6, 0.2]. The end position is [-0.4, 0, 0.2]. There is a stone in front of you, you need to plan a middle waypoint to move the drone robot from the left side of the stone. Where should it be?
A: To make the drone a little bit more on the left to avoid the stone, we can make the X value more negative, where Y value a little bit higher than the starting pose. The middle waypoint can be [-0.33, -0.67, 0.2].

Are you ready?



You are supposed to give a robot intermediate waypoint to plan its trajectory correctly following the language instructions. Each time I will give you two pictures. The first picture is an occupancy world; there are some obstacle black stones in the world that you should avoid, the start and the goal position of the drone is shown as the green circle and the red square, respectively. The second picture labels some possible candidate intermediate waypoint for you to choose conditioned on the to the language-described tasks.

Here is an example:
Q: Given the images above. Now move the robot to the goal from the left side of the stone in front of the drone at the start position. which waypoint do you want to choose for the drone?
A: According to the image, a good choice should be the waypoint 23.

Are you ready?

Figure 2. The prompt example input in the LLM and VLM. The left is the prompt for GPT-4 and the right is the prompt for GPT-4V. After input the prompt, we will iteratively send the evaluated data into LLM/VLM to get the waypoint prediction results.

work is built based upon the work of NeRF-based navigation [1], and our key challenge becomes how to predict suitable waypoints given the language instructions and relevant images of the environment. Once the waypoint is chosen, we will run the original NeRF-based navigation to connect those pieces of sub-trajectories for a complete navigation path. Figure 3 shows an example of using waypoints for language-conditioned trajectory planning.

3.3.1 LLM-based Waypoint Prediction

Large language models (LLM) such as GPT-4 has been shown to have impressive capability of spatial reasoning over the 3D geometry [3, 8], which is important to predict reasonable waypoint coordinate for our model to navigate. Following the previous work [3], we design prompt to include the task description, the environmental information, the start and end positions, then use the chain-of-thought technique [16] to generate thoughts of reasoning and the waypoint numbers. The left part of Figure 2 shows an example of our prompts using GPT-4. Note that, LLM-based waypoint prediction only relies on language tokens and does not take the image modality into account.

3.3.2 VLM-based Waypoint Prediction

Previous work [17, 18] has demonstrated that the state-of-the-art Vision-Language Models (VLM) possess great ability in website design, math problem solution, and decision

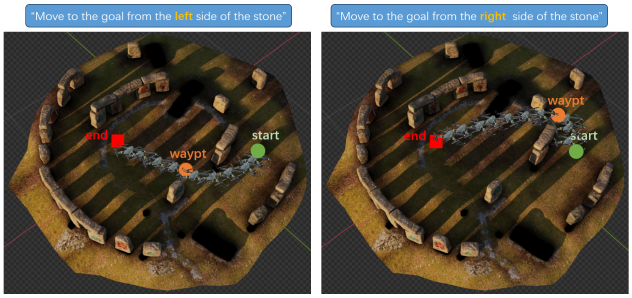


Figure 3. Given the same start and goal pose, we prompt the LLM/VLM to get the middle waypoint conditioned on different language instructions, then run nerf-based navigator to connect those pieces to get a complete trajectory.

making in complicated scenarios like autonomous driving and embodied environment. In this work, we choose to prompt GPT-4V to generate waypoint prediction as shown in the right part of Figure 2. We add labels to indicate the start and goal position of the robot, and also add a grid of waypoints as candidates for GPT-4V to select for the waypoint, which makes the prediction more reliable.

4. Experimental Results

4.1. Settings

We conducted our experiment in both a simulated scene (the Stonehenge in Blender) and a real-world scene (a laboratory in CSE department). For both scenes, we chose 20 sets of start and end pose pairs for evaluation. For simplicity, we

considered two types of language-guided tasks: move to the goal by taking a detour around the obstacle in front of the drone in terms of either the left side or the right side.

4.1.1 Metrics

Correctness. After planning the complement trajectory, we will examine whether the drone robot moves following the language instructions or not. For the Stonehenge scene, we only consider the nearest stone in front of the start pose as the reference object, for the lab scene, we only consider the table located in the room center as the reference object.
Collision-free. We will also examine whether the trajectory is collision-free in the Blender. If the trajectory has intersections with any obstacles, it will be considered as collision.
Time cost. We report the total planned steps of the robot from the start to the end position. Generally, a good robot should move to the goal correctly without collision as fast as possible.

4.1.2 Compared Methods

For the waypoint prediction, we report both LLM-based and VLM-based methods, and a random generation method for reference. To compare with language-guided approaches, we also report the unconditional trajectory planning in [1] .

4.2. Results in Simulated Environment

Model	Correctness ↑	Collision-Free ↑	Time ↓
unconditional	0.35	0.8	17.8
Random	0.4	0.8	31.1
LLM-based	0.5	0.7	**22.5**
VLM-based	**0.6**	**1.0**	25.3

Table 1. Results in Stonehenge simulated scene. *unconditional* denotes the unconditional trajectory planning. The rest denotes different methods for language-guided trajectory planning.

Table 1 shows the evaluation results of all the methods in the simulated scene. As we can see, the VLM-based trajectory planning achieves the highest correctness and collision-free rates among all methods, indicating that multi-modal input brings superior performance. Comparatively, LLM-based trajectory planning tends to make more conservative waypoint prediciton and obtains less time costs and is not as good as VLM-based method in correctness.

4.3. Results in Real World

4.3.1 Real World Neural Radiance Field Representations for Navigation

In this section, we present our results on a NeRF-based representation of a real-world environment in that we use our language-guided visual navigation method for. We used a series of images taken from an indoor environment and trained a NeRF using a Tiny Cuda NN (tcnn) [10] implementation of instant-ngp [11]. Figure 4 illustrated the bird-view picture of the NeRF we trained on a 74s video.

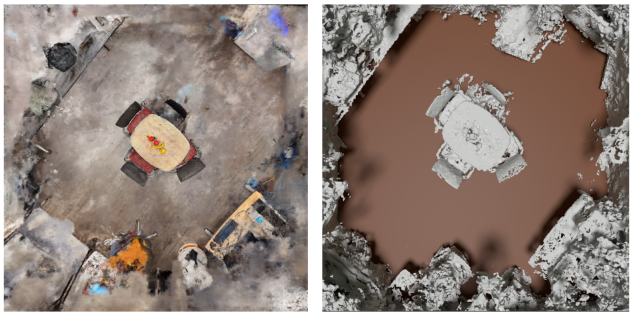


Figure 4. Birdseye view of the NeRF reconstruction of the real world indoor scene. The left is RGB picture, the right is the mesh object rendered in Blender.

4.4. Final Results

Table 2 gives the results on the lab scene. Since the real-world scene has much more noise than the simulated scene, the planned trajectories are more likely to collide and take more time steps to achieve. Interestingly, the VLM-based method does not perform better then LLM-based method in terms of correctness, this is because the layout of the lab is quite simple (only the table in the room center as the main obstacles) so that LLM without vision input can already acquire adequate performance. The incorrect cases are mainly because the waypoint can not make enough detour for the drone robot to take from the left or right sides.

Model	Correctness ↑	Collision-Free ↑	Time ↓
unconditional	0.15	0.8	23.25
random	0.1	0.5	36.2
LLM-based	**0.5**	0.5	**31.25**
VLM-based	**0.5**	**0.55**	37.2

Table 2. Results in real-world lab scene. *unconditional* denotes the unconditional trajectory planning. The rest denotes different methods for language-guided trajectory planning.

5. Conclusion and Discussion

We present a method that uses Vision-Language Models (VLMs) for the purpose of indoor navigation for robotic platforms. We build this end-to-end visual-language guided trajectory optimization framework to such that the navigation is capable to be performed in a NeRF, and show that compared to language-only, or unconditioned navigation attempts, the VLMs produce safe and correct paths.

References

[1] Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2): 4606–4613, 2022. 1, 2, 3, 4

[2] Benjamin Bolte, Austin Wang, Jimmy Yang, Mustafa Mukadam, Mrinal Kalakrishnan, and Chris Paxton. Usa-net: Unified semantic and affordance representations for robot memory. *arXiv preprint arXiv:2304.12164*, 2023. 1, 2

[3] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023. 3

[4] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE, 2023. 2

[5] Lin Huang, Tomas Hodan, Lingni Ma, Linguang Zhang, Luan Tran, Christopher Twigg, Po-Chen Wu, Junsong Yuan, Cem Keskin, and Robert Wang. Neural correspondence field for object pose estimation. In *European Conference on Computer Vision*, pages 585–603. Springer, 2022. 2

[6] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*, 2020. 2

[7] Tsung-Yi Lin, Peter Raymond Florence, Yen-Chen Lin, and Jonathan Tilton Barron. Inverting neural radiance fields for pose estimation, 2023. US Patent App. 18/011,601. 2

[8] Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models. *arXiv preprint arXiv:2307.04738*, 2023. 3

[9] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Conference on Robot Learning*, pages 1347–1356. PMLR, 2022. 2

[10] Thomas Müller. tiny-cuda-nn, 2021. 2, 4

[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, 2022. 2, 4

[12] Joseph Ortiz, Alexander Clegg, Jing Dong, Edgar Sucar, David Novotny, Michael Zollhoefer, and Mustafa Mukadam. isdf: Real-time neural signed distance fields for robot perception. *arXiv preprint arXiv:2204.02296*, 2022. 2

[13] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011. 2

[14] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6229–6238, 2021. 2

[15] Zachary Teed and Jia Deng. Tangent space backpropagation for 3d transformation groups. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[16] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022. 3

[17] Licheng Wen, Xuemeng Yang, Daocheng Fu, Xiaofeng Wang, Pinlong Cai, Xin Li, Tao Ma, Yingxuan Li, Linran Xu, Dengke Shang, et al. On the road with gpt-4v (ision): Early explorations of visual-language model on autonomous driving. *arXiv preprint arXiv:2311.05332*, 2023. 3

[18] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9:1, 2023. 3

[19] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. 2