Certifiable Safety and Control in Neural Radiance Fields

Timothy Chen, *Stanford University, Stanford, USA, chengine@stanford.edu*

I. INTRODUCTION

Navigating within unknown environments is a key area of study in robotics, from manipulation to autonomous driving to drone flight. In many scenarios, virtually no information is given about the environment apart from on-board perception. In these cases, usually there is no precise localization like GPS or a precisely known map prior to build upon. Example environments include cave systems, hostile environments, or dynamically changing environments (e.g., collapsed infrastructure).

Key aspects of navigating these environments is to learn a map of the environment using on-board perception while safely navigating the environment using this learned representation. Formal certificates on safety are paramount for reliable behavior. Not only do guarantees on safety prevent costly collisions of robots, but they also prevent damage to the environment. For structurally unsound environments like a collapsed building, collision of a search-and-rescue robot into an unstable wall could jeopardize the lives of survivors in the building. Furthermore, safety certificates make for more robust and interpretable control strategies in real-world environments compared to those without such guarantees.

While learned representation techniques of unknown environments have evolved rapidly, using them to navigate should be considered in a probabilistic manner in order to promote robustness and confidence. Probabilistic formulations can also alleviate behavior that might be too conservative, compared to those generated from reachability analysis due to over-approximations. For example, consider a drone and a hole that is slightly bigger than it. Deterministically learned representations and controllers may not realize that the hole is traversable, resulting in more control effort spent on less efficient trajectories. As a result, working on distributions allows planners and controllers to act more intelligently.

In this paper, our contributions are a deterministic and probabilistic guarantee of safety within learned probabilistic environments, along with synthesis of controllers from these guarantees. While works considering collision avoidance within high-fidelity probabilistic environments are already few, to our knowledge, none have yet proposed certifiably safe controllers of the deterministic or probabilistic variety when navigating these learned environments.

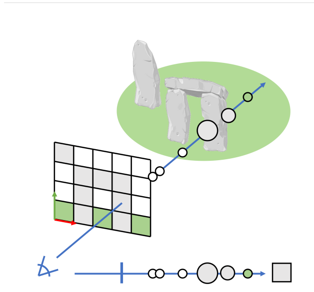
II. RELATED WORK

1. Learned Representations

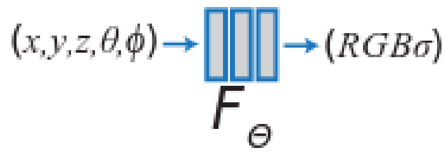
An unknown environment can be represented in many ways, from discrete representations like voxel grids, occupancy grids, and point clouds to continuous implicit functions like signed-distance functions (SDF). However, these representations are typically low-resolution, deterministic representations of the environment. Moreover, while some representations do take into account color, those that do are not able to be queried for a full colored image from arbitrary views. This is unacceptable for missions where the map is of great importance for visual inspection (e.g. scientists looking for specific visual features in the terrain). Moreover, while robots have been shown to navigate through these learned representations, their use in localization solely from RGB sensors is greatly diminished. Because they cannot query a full color image, a convoluted front-end as part of a simultaneous localization and mapping (SLAM) pipeline is necessary to perform feature matching, outlier rejection, etc. Probabilistic formulations are also compelling in that they can ease the learning process and model complex effects in the environment (e.g., light scattering or material properties).

2. Neural Radiance Fields

Neural Radiance Fields (NeRFs) [1] is a key advance to learned representations as they are a high-fidelity, probabilistic, and continuous representation of an environment, while also sporting the ability to query colored images from arbitrary viewpoints. Because this representation has much potential for robotics applications, our work only focuses on safety within NeRFs. The key components of a NeRF is a multi-layer perceptron (MLP) implicitly representing the environment as $(\sigma, r, g, b) \in \mathcal{R}^4$, a density σ and radiance given points in 3D space and view directions along traced rays from the image plane (Figure 1). A differentiable rendering process is then used to recreate the image, allowing training of the MLP from the reconstruction loss between the rendered and observed image.



(a)



(b)

Figure 1: NeRF pipeline from sampling rays shot (a) into scenes to feed as coordinates into an MLP returning radiance and density.

3. Control Barrier Functions

Formal verification on safety is paramount to the success of navigation through unknown environments. A popular technique developed from nonlinear control theory is Control Barrier Functions (CBFs) [2], related to Control Lyapunov Functions (CLFs) for stability. A safe set S can be represented as the 0-level set of the CBF $h(x)$ such that $S \in \{x \in \mathcal{R}^n \mid h(x) \geq 0\}$. Assuming control-affine continuous-time dynamics, a controller can be obtained that guarantees the forward invariance of this safe set, hence the agent is safe $\forall\, t \geq 0$. This controller can be formulated as a quadratic program with affine constraints (Eqn. 1), which can be solved efficiently and is suitable in real-time applications.

The Quadratic Program is given by:

$$\begin{aligned} \min_u \quad & ||u||_Q^2 \\ \text{s.t.} \quad & \frac{\partial h}{\partial x}(f(x) + g(x)u) \geq -\alpha(h(x)) \\ & u \in \mathcal{U} \end{aligned} \tag{1}$$

where α is an extended class-K function and $\mathcal{U}$ is the set of feasible control inputs. The loss is the Mahalanobis distance.

A probabilistic formulation of CBFs is given by [3], which is amenable for use in NeRFs. Not only can it provide less conservative control while ensuring probabilistic safety, but also provides a confidence metric for possibly poorly understood noise models (e.g., poorly trained NeRFs). The probabilistic formulation of the safe set follows a similar definition $Pr(x \in S) \geq \phi$. Using a series of sufficient conditions, this constraint is guaranteed by a QP with polytopic constraints (Eqn. 2). We will not explain variables here, as we will make extensions to Eqn. 2 in our approach and provide definitions there. Again, this program can be solved efficiently and real-time.

$$\begin{aligned} \min_u \quad & ||u||_Q^2 \\ \text{s.t.} \quad & -2e^T g(x)u/\gamma \leq ||e||^2 - d \cdot R_i^2 + 2e^T f(x)/\gamma \\ & u \in \mathcal{U} \end{aligned} \tag{2}$$

III. PROBLEM FORMULATION

The purpose of this work is to enable safe navigation within unknown environments, a task that involves generating a learned representation of the environment and safety certificates that act on this representation. We frame this task probabilistically, both in learning a representation and in formulating a safe controller. Specifically, a map of the environment is represented as a function mapping points in 3D space to a density $\mathcal{F}_\theta : \mathcal{R}^3 \to \mathcal{R}$ (i.e., a NeRF). This density will be treated similar to some distribution of noise corrupting the true object within this density cloud. For points in $\mathcal{R}^d$ where $d = 3$,

$$\begin{aligned} &\dot{x} = f(x) + g(x)u + w,\ w \sim \mathcal{U}(-\Delta w, \Delta w) \longrightarrow \text{Dynamics} \\ &\hat{x} = x + v,\ v \sim \mathcal{U}(-\Delta v, \Delta v) \longrightarrow \text{State Measurement} \\ &\hat{x}_k = x_k + r_k,\ r_k \sim \mathcal{F}_k(x)\ \forall\, x_k \in \mathcal{X} \longrightarrow \text{Obstacle Measurement} \end{aligned} \tag{3}$$

where $\mathcal{X}$ is an infinite set of x_k representing the ground truth surface of a obstacle represented by a NeRF. Note that this noise injected into the true surface location is dependent on the location of the robot in accordance with the definition of the NeRF density. To simplify the discussion, we assume perfect state information about the agent $\Delta v = \Delta w = 0$, so the only uncertainty is in the surface of the obstacle. For simplicity, we assume a 3D single-integrator robot initially, before testing on nonlinear systems.

We assume that the NeRF is already trained, such that the densities are not time-varying $\frac{\partial F_\theta}{\partial t} = 0$. From this model of the environment and Eqn. 2, we can utilize probabilistic CBFs to create a probabilistically safe control in the form of a quadratic program in order to live within the probabilistic safe set $S = \{x \,|\, Pr(\dot{h}_k(x, x_k) + \alpha(h_k(x, x_k)) \geq 0) \geq \sigma \,\forall\, x_k \in \mathcal{X}\}$. This controller will ensure probabilistic safety for some higher-level planner when navigating unknown environments.

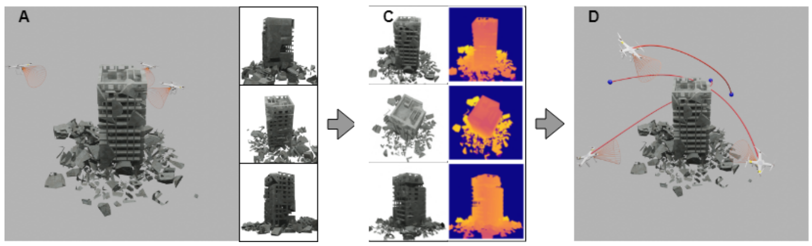


Figure 2: A NeRF is trained from training images (A, C). A probabilistic CBF formulation will be used to ensure safety while agents are navigating the NeRF (D).

IV. APPROACH

The deterministic CBF baseline can be created by treating the negative density as the CBF. In other words, we want the CBF to drive our agent away from regions of positive density to remain safe. This is a valid assumption because the density is highly correlated with the presence of mass or an object at that location in space. However, we make some amendments to Eqn. 1. First, because our controller receives a control from a higher level planner and renders it safe, we change the loss and the dynamics to reflect our amending control input. Moreover, due to the noise present in gradients of neural networks at a single point, we average densities in a box around the agent to get a smoother density signal $\bar{h}(x) = \sum_{j=0}^{N} \frac{1}{N} h(x_j)$. Moreover, we take the average of all gradients of $\bar{h}$ across all points in the box to retrieve a smooth gradient signal. We retrieve Eqn. 4, a minimally invasive, safe controller.

$$\begin{aligned} \min_{u} \quad & ||u||_Q^2 \\ \text{s.t.} \quad & -\{\sum_{i=0}^{N} \frac{1}{N} \frac{\partial \bar{h}(x)}{\partial x}(x_i)\} g(x) u \leq \{\sum_{i=0}^{N} \frac{1}{N} \frac{\partial \bar{h}(x)}{\partial x}(x_i)\}(f(x) + g(x) u_{plan}) + \gamma \alpha(\bar{h}(x)) \\ & u \in \mathcal{U} \end{aligned} \tag{4}$$

In creating the probabilistic variety, we provide some intuition on the NeRF density in creating our controller. The NeRF density is not a joint probability distribution, as these densities are determined by ray-traced lines, rather by the entire 3D domain. It is more enlightening to turn to interpretations of the light rendering equation in [4] and treat the opacity $1 - exp(-\int_0^t \sigma(t) dt)$ as an occupancy probability. This is a ray termination probability in light rendering literature. Because we can model light as a particle, we make the analogy that agents acting like particles should also terminate (i.e., collide) according to the same cumulative density assuming opaque materials. From this assumption, we are able to make analogies to variables in Eqn. 2.

We trace rays in the positive (x, y, z) directions, as well as in the negative directions. We use the normalized cumulative occupancy probability $\Phi_i(x)$ from tracing out the rays in each direction as components of e in Eqn. 2, which defined in [3] as either $e_i^1 = \Phi_i^{-1}(\sigma)$ or $e_i^2 = \Phi_i^{-1}(1 - \sigma)$ - the inverse CDF of the noise distribution along each axis in $\mathcal{R}^3$. These inverse CDFs are retrieved using linear interpolation from our discrete normalized cumulative occupancy probability. Due to the unconventional way the ray-tracing plays a part into the distribution, we will always use e_i^2 as that is the closer point to the agent. Specifically, components of the positive direction $e_+ = (e_{x,+}, e_{y,+}, e_{z,+})$ use $-e_i^2$, while the negative direction e_- uses e_i^2. Because we shoot out rays in the positive and negative directions, we have two constraints with similar forms corresponding to the positive and negative rays.

$$
\begin{aligned}
\min_{u} \quad & ||u||_Q^2 \\
\text{s.t.} \quad & -2e_+^T g(x)u/\gamma \leq ||e_+||^2 - d_+ \cdot R^2 + 2e_+^T f(x)/\gamma + 2e_+^T g(x)u_{plan}/\gamma \\
& -2e_-^T g(x)u/\gamma \leq ||e_-||^2 - d_- \cdot R^2 + 2e_-^T f(x)/\gamma + 2e_-^T g(x)u_{plan}/\gamma \\
& u \in \mathcal{U}
\end{aligned}
\tag{5}
$$

R is the safety radius around the agent along each axis. d_+ and d_- are determined by the information along the positive and negative set of rays, respectively. If along certain directions, there is no density (i.e. the CDF is not well-defined), d is decremented starting from 3. For example, if no density is along the $+z$ axis, while $+x$ and $+y$ have non-zero densities, then $d_+ = 2$. More detailed construction of the probabilistic CBF formulation is found in [3] in the Appendix.

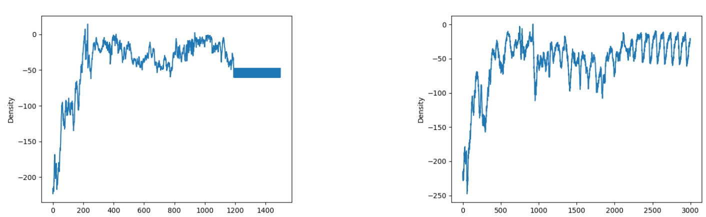
V. RESULTS

Simulation experiments of both the deterministic baseline and probabilistic algorithm are reported across two initializations. Simulations were done in Blender using single-integrator dynamics. The higher level planner was a naive planner pointing the agent directly to the goal location.

We are able to see that qualitatively, both the baseline and proposed approach get to the destination and are nominally safe (i.e., the trajectories do not visually clip the mesh), albeit through different trajectories (Figure 3). It can also be observed that the baseline trajectory has more jitter because it is reactionary only within its small box of sampled points; on the other hand, because the probabilistic CBF performs ray-tracing into the scene, it is able to reason about possible collisions farther away, resulting in smoother trajectories. Furthermore, the smoother trajectory allows for smaller control effort. If both the baseline and probabilistic formulation used the same feasability set $u \in \mathcal{U}$, the baseline would return infeasability in the QP. We also provide quantitative results of the density along these trajectories. While both the baseline and proposed are nominally safe, at single points in the trajectory where they are unsafe, the baseline has a higher violation than the proposed. We urge viewers to see the comparison videos in the supplementary materials.

(a) (b) (c)

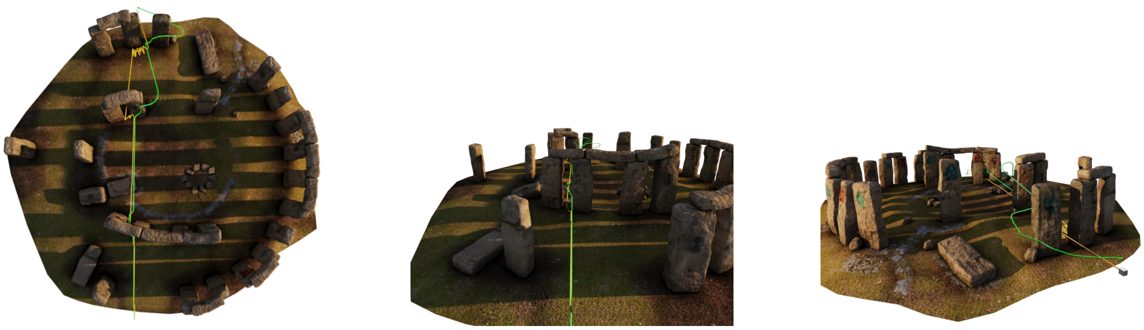
Figure 3: Trajectories of deterministic (yellow) and probabilistic (green) controllers.



(a) Deterministic **(b)** Probabilistic

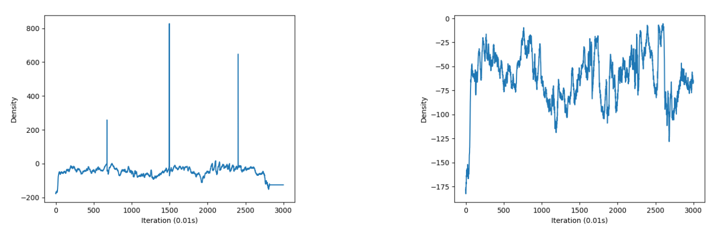
Figure 4: NeRF densities along trajectory.

In this initialization, it is more clear the jitter present in the baseline compared to the proposed when coming head-on with the pillar near the end of the trajectory (Figure 5). This jitter also causes points in the baseline trajectory that are massively unsafe, though it quickly reverts to safe behavior at the expense of high control effort. Meanwhile, the probabilistic method is always safe in this initialization (Figure 6). It should also be noted that while the proposed method displays higher safety than the baseline, it is not conservative. It is still able to go between pillars at the beginning of the trajectory.



(a) **(b)** **(c)**

Figure 5: Trajectories of deterministic (yellow) and probabilistic (green) controllers.



(a) Deterministic **(b)** Probabilistic

Figure 6: NeRF densities along trajectory.

Both methods can be run in real-time. The baseline sampling 27,000 points within the box can be computed at 35 iterations per second. Using a similar batching strategy as the baseline for the proposed method, sampling 125 points in the box yields similar times at 33 iterations per second. However, it should be noted that nearly 300,000 points are passed through the network, so the proposed method is more computationally efficient per point because it does not rely on gradients the way the baseline does. Even when using a single point, the proposed method still maintains similar safety while sporting evaluations at 250 iterations per second. As a result, the proposed work is robust, safe, non-conservative, and very efficient.

VI. FUTURE WORK

We hope to inject more complex, nonlinear dynamics into our experiments, such as quadrotor dynamics. This agent may also be subject to process and measurement noise. Furthermore, extensions to the decentralized multi-robot, multi-NeRF scenario is also possible, as [3] has shown that their method is amenable to a decentralized formulation. Moreover, an ablation study can be performed on NeRFs of varying fidelity to highlight the robustness of our probabilistic formulation. Safety on the ground-truth mesh will also be determined. Finally, safety of this controller will be tested when a robot is simultaneously training a NeRF and navigating within the unknown environment.

VII. CONCLUSION

In this work, we propose a non-conservative, safe, and efficient probabilistic Control Barrier Function-based controller amenable to real-time navigating in a learned Neural Radiance Field. The proposed method is formulated as a quadratic program that can be solved efficiently, amenable to use in real-time applications on hardware. While the baseline is also our proposed work and works decently, the probabilistic algorithm is far superior in most regards. We see this controller as a lower-level controller to a trajectory planner that can render all control actions of the planner safe while being as minimally invasive as possible.

REFERENCES

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, R. Ng, “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis,” in European Conference on Computer Vision, 2020.

[2] A. D. Ames, X. Xu, J. W. Grizzle and P. Tabuada, "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," in IEEE Transactions on Automatic Control, vol. 62, no. 8, pp. 3861-3876, Aug. 2017, doi: 10.1109/TAC.2016.2638961.

[3] W. Luo, W. Sun, A. Kapoor, "Multi-Robot Collision Avoidance under Uncertainty with Probabilistic Safety Barrier Certificates," in 34th Conference on Neural Information Processing Systems (NeurIPS), 2020.

[4] N. Max, "Optical Models for Direct Volume Rendering," in IEEE Transactions on Visualization and Computer Graphics, Vol. 1, No. 2, June 1995.

APPENDIX

Please see the Github link (https://github.com/chengine/Papers) for videos of the baseline and probabilistic CBF experiments. In the title names, exp1 and exp2 correspond to the different initializations, while cbf and pr cbf denote the baseline and proposed work, respectively.