# Hallucinating the World: Towards Geometrically-Aware Generative Cinematography

Timothy Chen
chengine @ stanford.edu

MSL Multi-robot Systems Lab

## Problem

How do we generate novel videos of *real-world unstructured* environments? Can we generate *realistic* videos with little human intervention, while developing a tool that is *interpretable* for humans to use? We want the resulting video stream to be accurate, realistic, and quick to generate. Applications include cinematography, scientific exploration, and search-and-rescue.

## Contribution

We develop a pipeline that automatically generates a **realistic**, **random** video stream of an environment given the following:

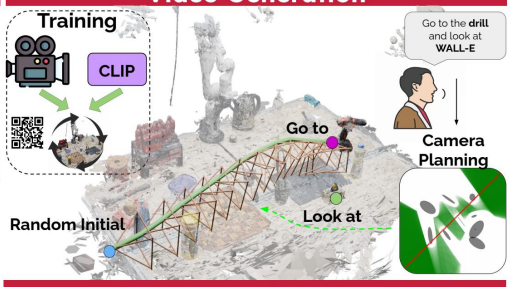
- An input image stream of the environment
- A textual instruction of where to *go to*
- A textual instruction of where to *look at*

The developed pipeline is **semantically meaningful** (the textual instructions will be followed), **accurate** (the resulting video stream is of the same environment as the input stream), and **geometrically-aware** (the camera respects object boundaries).

## Pipeline

We train a semantics-embedded Neural Radiance Field (NeRF) [1] from the input image stream to form a consistent 3D geometry from which to render **accurate** images from. The semantic embedding allows the representation to associate 3D space with object identities, and we develop a novel camera planning technique to respect boundaries.

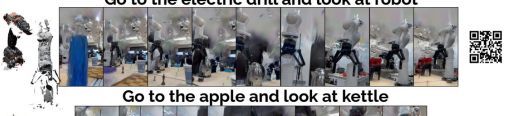
## Video Generation



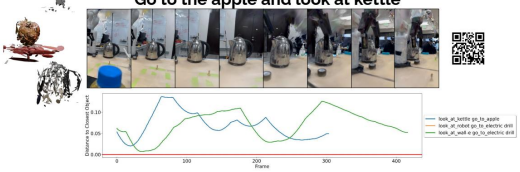

## Results

Go to the electric drill and look at WALL-E

Go to the electric drill and look at robot



Go to the apple and look at kettle



## Polytopes and Planning

We model the camera as a ball and use Gaussian Splatting [2] as the NeRF model. A complete ellipsoid-ellipsoid intersection test is presented in [3].

$$\exists s : K(s) = (\mu_a - \mu_b)^T [\tfrac{1}{1-s}\Sigma_a + \tfrac{1}{s}\Sigma_b]^{-1}(\mu_a - \mu_b) > 1$$

We generate separating hyperplanes based on this constraint. Stacking these constraints form a polytope.

$$\forall j : a_j^T(\Sigma_j, \mu_j, \Sigma^*, x^*)\ x \le b_j(\Sigma_j, \mu_j, \Sigma^*, x^*)$$

We parametrize the camera path as poly-Bezier curves. We optimize for the smoothness of the solution. We constrain the curves using the polytopes and continuity. We then solve a Quadratic Program to find the path.

$$\min\ y^T Q y$$
$$\forall i : A^i(\Sigma_{1:J}, \mu_{1:J}, \Sigma^*, x^{*,i}) y^i \le b^i(\Sigma_{1:J}, \mu_{1:J}, \Sigma^*, x^{*,i})$$
$$y^i(1) = y^{i+1}(0)$$
$$y^0(0) = y_0$$

## Limitations

The method **cannot**:

- hallucinate beyond some volume of the training dataset, where supervision exponentially decreases due to perspective.
- Build a scene where the data quality is poor
- Account for more artistic cinematographic attributes (yet)

## Future Work

The pipeline is **differentiable** from the rendered video to the polytopes. The pipeline can be extended to include the rendered video in a loss function and backpropagate into the polytopes while still preserving geometric-awareness.

[1] B. Mildenhall et al. NeRF: Neural Radiance Fields. ECCV 2020.
[2] B. Kerbl et al. 3D Gaussian Splatting. SIGGRAPH 2023.
[3] T. Chen et al. Splat-Nav. arXiv 2024.