6-DOF POSE ESTIMATION USING TSDF

Presented By : Team 5

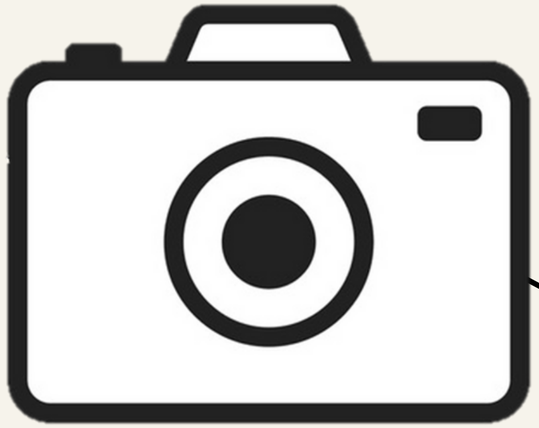
Mobile Robotics Final Evaluation | Monsoon ‘24

SHORT DESCRIPTION

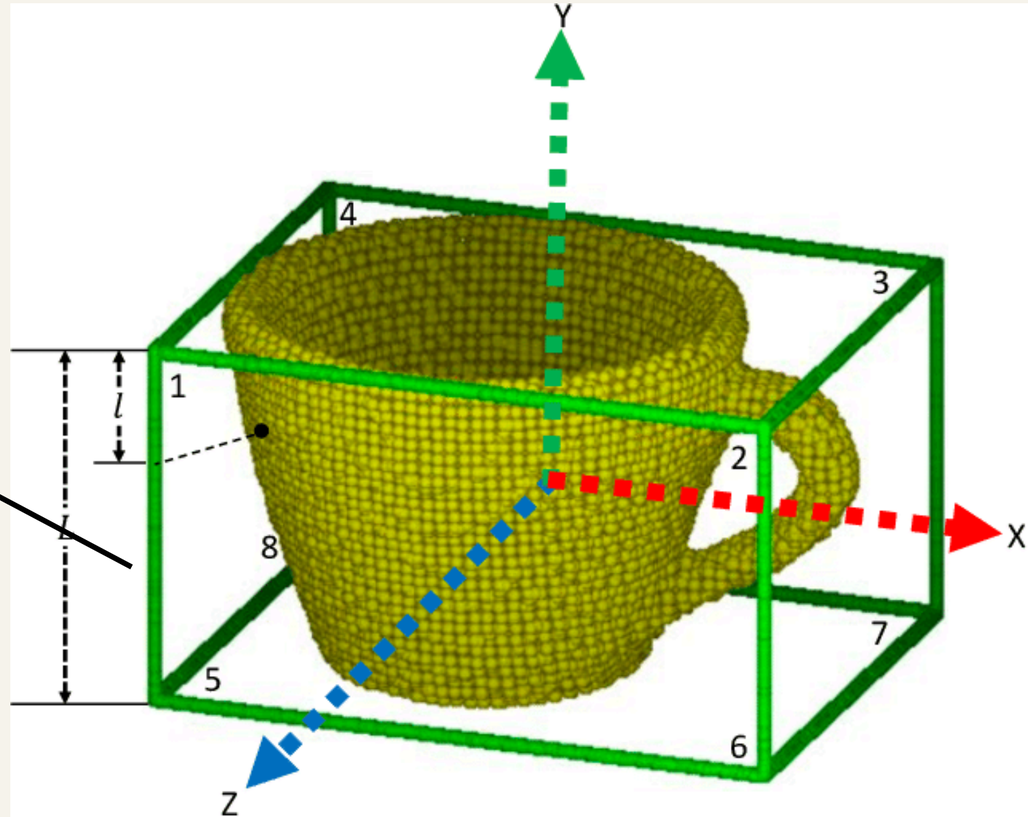
Estimating 6 DOF pose of an object using Truncated Signed Distance Function.

Also, we use the Truncated Signed Distance Function of an object to create its mesh.

Task: To Calculate the 6DOF Pose of an Object



R and T of the object with respect to the camera



RELATED WORK

Here we explore the existing related work in the field.

APPROACHES

CAD BASED

Older methods for 6DOF pose estimation relied on CAD models of objects as ground-truth 3D references

CAD FREE

Newer works try to estimate the 6DOF Pose without using any CAD models. While having still not surpassed the accuracy of CAD, they claim to be close in accuracy to the CAD BASED models.

CAD BASED – GENERAL OVERVIEW

1. Object Detection and Segmentation using CAD Models
2. Feature Extraction and Matching
3. Pose Estimation Using PnP + Pose Refinement

LIMITATIONS OF CAD BASED METHODS

- **Challenges**

  Sensitivity to occlusions, lighting changes, and textures, making it difficult for low-texture or reflective surfaces.

- **Scalability Issues**

  Requires unique CAD models for each object, challenging for large datasets.

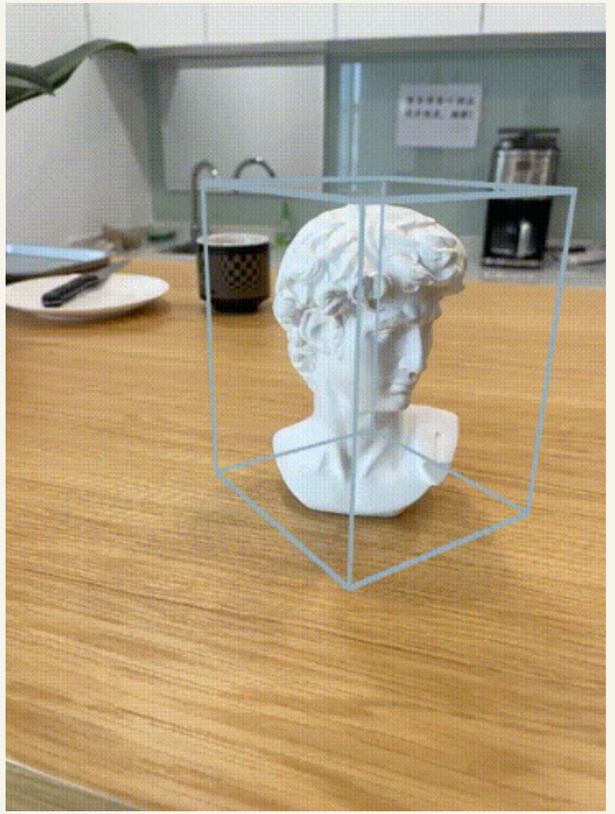
OnePose and OnePose++

These are Model Free Approaches

One Shot Approach

Claim to have results equivalent to Model Based Approaches for Textures Objects

OnePose++ is also able to handle Low Textured Objects using a Keypoint-Free Approach



OnePose and OnePose++

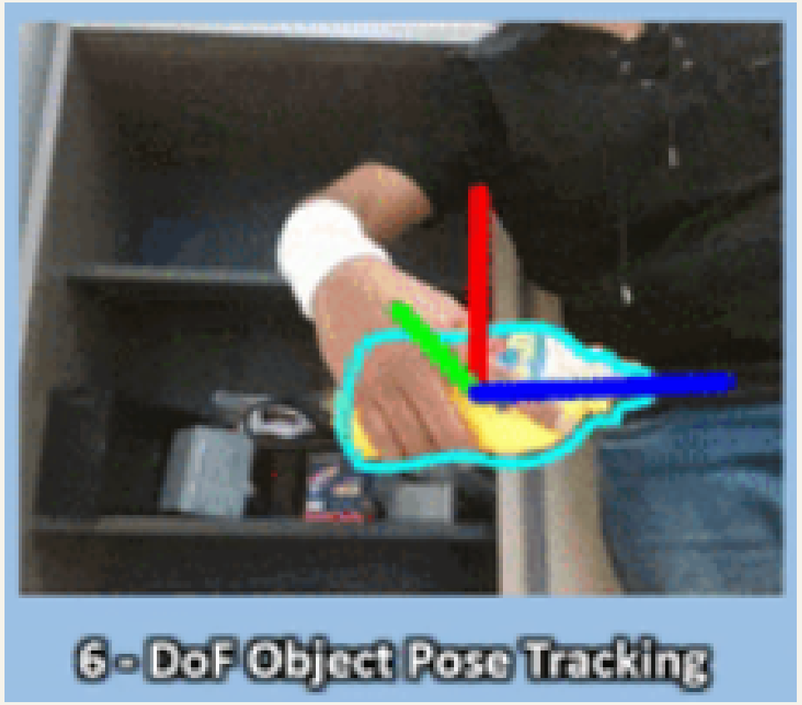
- **Stage I: Keypoint-Free Structure from Motion**

  Reconstructs the object's semi-dense point cloud from a reference image sequence

- **Stage 2: Object Pose Estimation Stage**

  Takes a query image and establishes 2D-3D matches between the reconstructed object point cloud and the image to estimate the object's pose using a coarse-to-fine scheme for efficiency.

BundleSDF



Proposes a novel method for 6-DoF pose tracking and 3D reconstruction of unknown, dynamic objects from monocular RGBD video sequences.

Only requires a 2D object mask in the first frame of the video and assumes the object is rigid

BundleSDF Steps

Coarse pose initialization

Obtains an initial estimate of the object pose in the current frame

Memory Pool

A key-frame memory pool is maintained to alleviate catastrophic forgetting and reduce long-term tracking drift.

Pose Graph Optimization

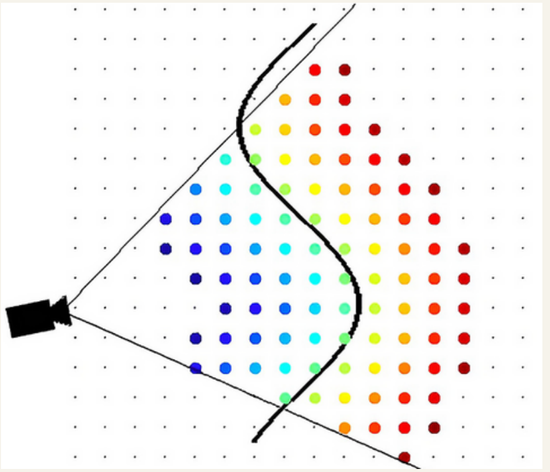
An online pose graph optimization process is used to refine the object pose estimates over time

Neural Object Field

A Neural Object Field is concurrently trained with the pose graph optimization to reconstruct the 3D shape and appearance of the object.

TRUNCATED SIGNED DISTANCE FUNCTION

Volumetric scene representation used for integrating multiple depth images taken from different viewpoints to reconstruct a 3D model of a scene



Favoured for its **efficiency** in terms of both time and space, and its suitability for parallel processing on GPUs

2D TSDF

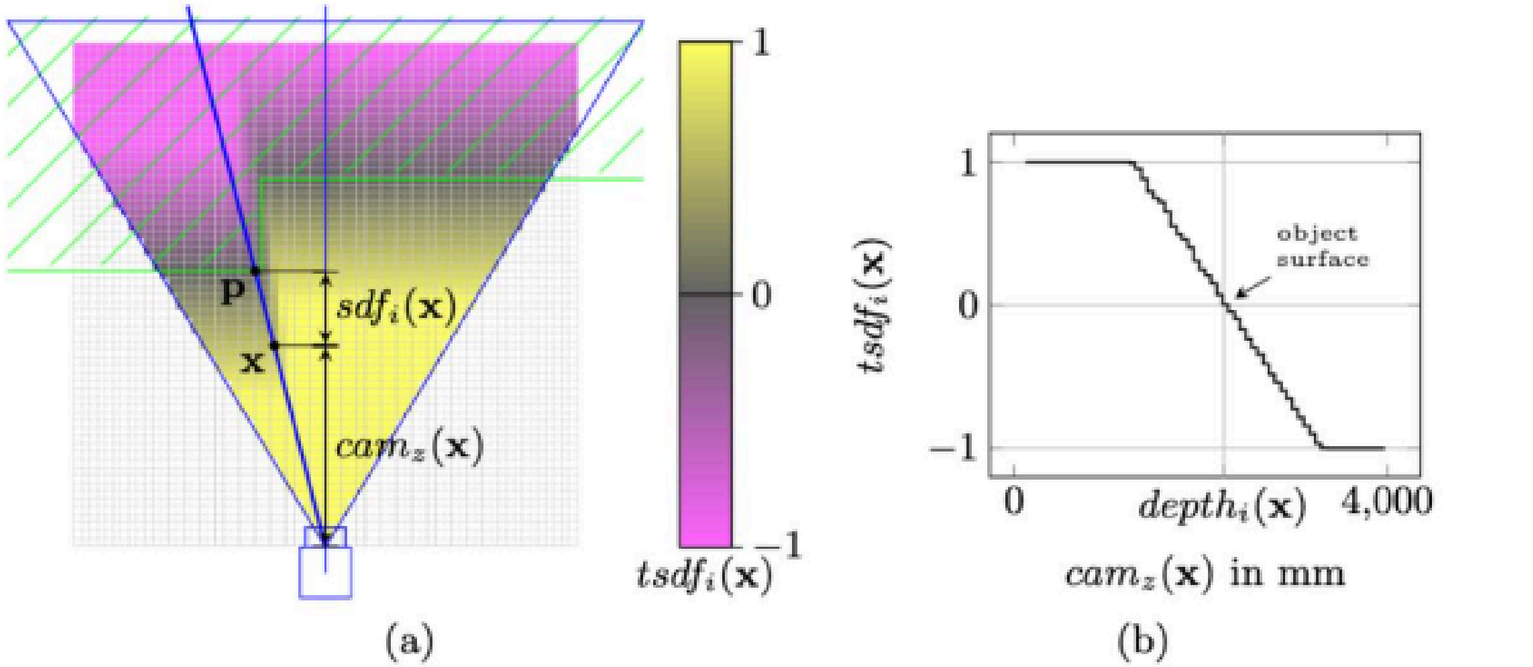
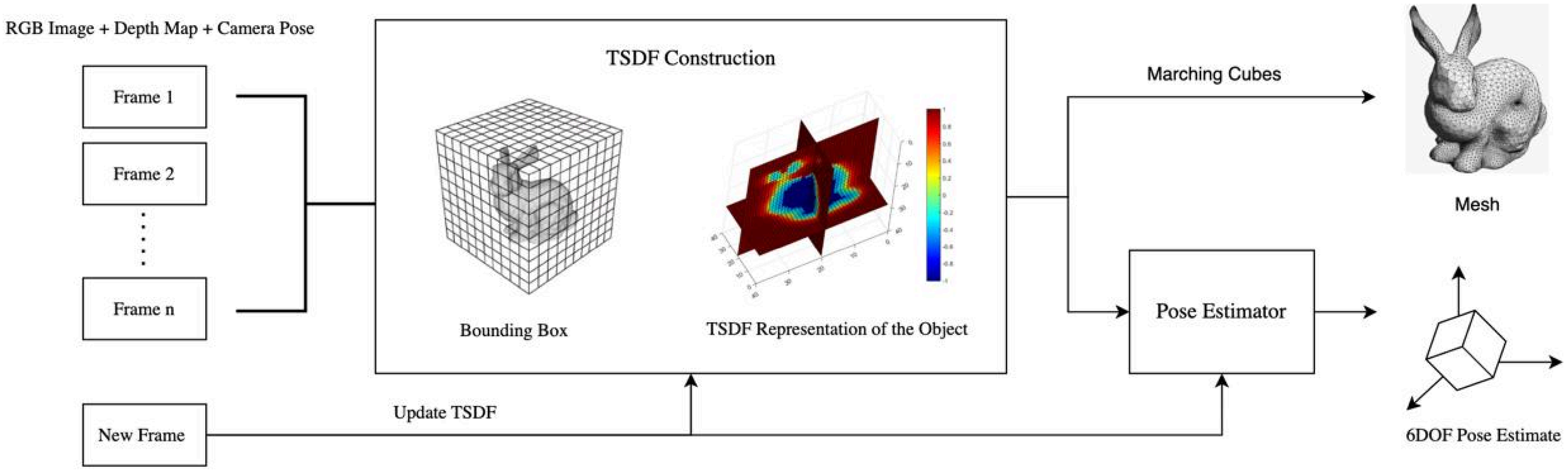


Fig. 3. 2D TSDF example. (a) Solid object (green), camera with field of view, optical axis and ray (blue), and TSDF grid (unseen voxels are white, for others see color bar). The signed distance value of voxel x is determined by the depth of the corresponding surface point p and the voxel's camera distance cam(x). (b) 1D TSDF sampled along the ray through p with t = 1000 mm. Object surface is at zero crossing.

OUR PIPELINE



OUR PIPELINE

TSDF CONSTRUCTION

Obtains an initial estimate of the object pose in the current frame

Mesh Generation

Using the Marching Cubes algorithm, we extract a high-resolution, polygonal mesh from the volumetric TSDF model.

6 DOF Pose Estimation

Estimate 6 DOF pose of an object in each new frame.

METHOD

TSDF Construction

- Volumetric representation where each voxel encodes the signed distance to the nearest surface.
- Using the depth maps and camera parameters, the spatial bounds of the scene are computed.
- Each frame, comprising the depth map, RGB image, and pose, contributes to updating the TSDF and weight grids.
- This process involves voxel-to-camera projection, truncated signed distance calculation, and updating the TSDF, weights, and colors

TSDF FORMULAE

Frame integration:

$$d = D_{\text{pixel}} - D_{\text{voxel}},$$

Truncation Step:

$$t_d = \max(-1, \min(1, d/t)).$$

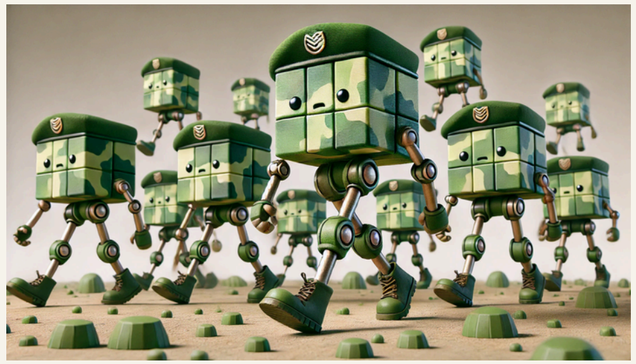
Update Steps:

$$t'_d = \frac{w \cdot t + w_{\text{new}} \cdot t_d}{w + w_{\text{new}}},$$

$$C' = \frac{w \cdot C + w_{\text{new}} \cdot C_{\text{new}}}{w + w_{\text{new}}},$$

$$w' = w + w_{\text{new}}.$$

METHOD



Mesh Generation

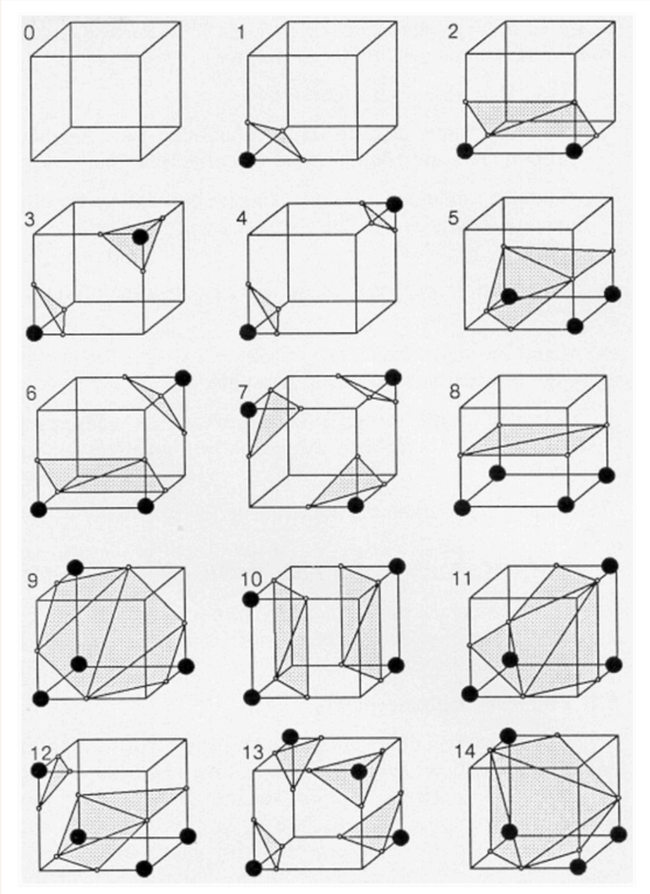
- **Marching Cubes:**

  Generates meshes from TSDFs by interpolating surfaces along cube edges where TSDF values change sign. Using a lookup table, it creates triangles for smooth surfaces but may produce overly detailed meshes at high resolutions.

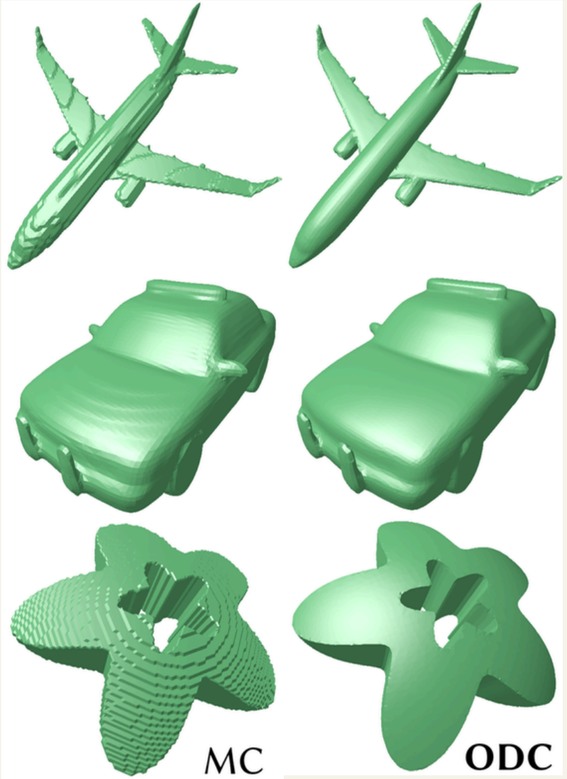
- **Occupancy Dual Contouring (ODC):**

  It preserves sharp features by placing vertices at centroids of voxel faces intersecting the surface, unlike Marching Cubes. It generates efficient meshes with fewer vertices, ideal for high-fidelity 3D models with sharp geometric details.

Comparison of the two methods



Marching Cubes



METHOD

Nearest 6DOF Pose Estimation

- The nearest pose is found by comparing frames used to construct the TSDF and selecting the one with the minimum Mean Squared Error (MSE) loss.

- Only TSDF surface points are compared, focusing on shape and spatial positioning rather than texture.

- The frame with the smallest surface alignment MSE provides the initial 6DOF pose estimate.

DATASETS

7 scenes

Stanford Bunny (Blender)

DATASETS : Synthetic

Suzanne Monkey (Blender)

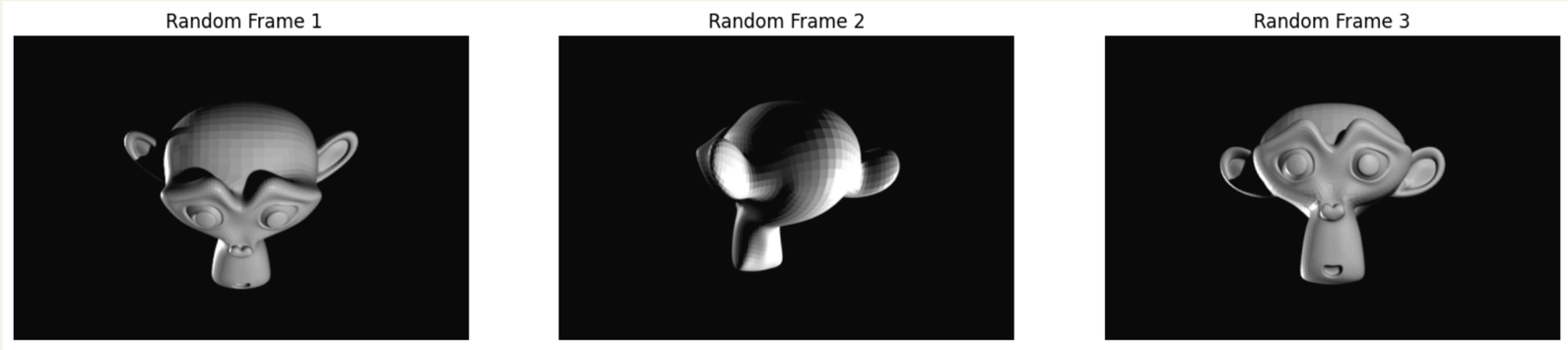
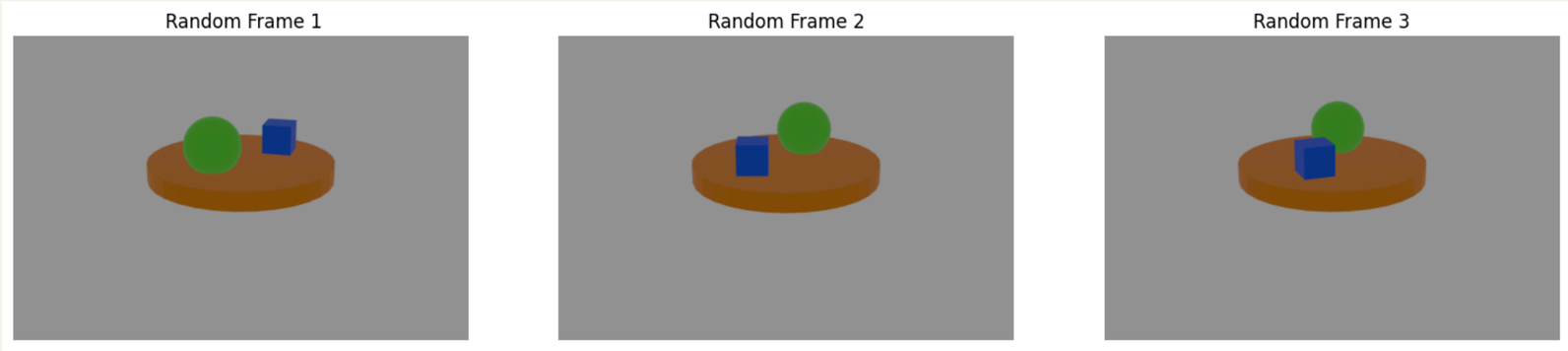
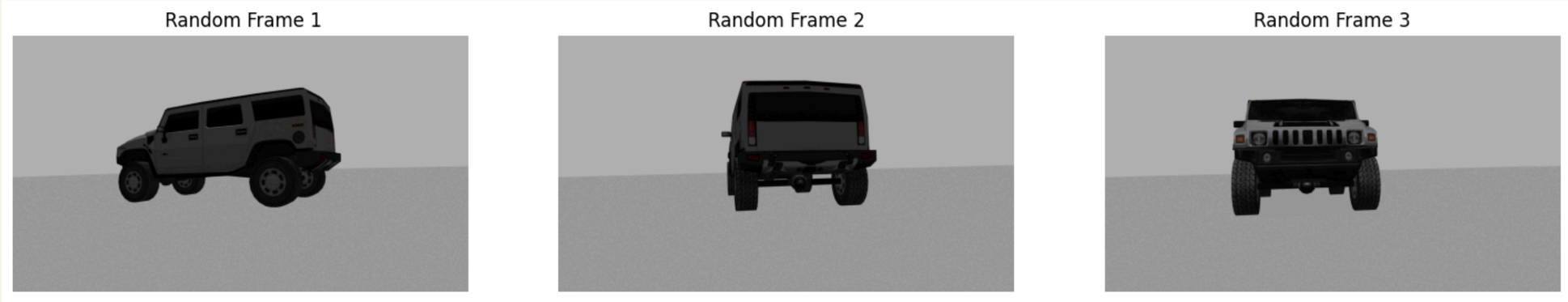


Table with objects (Blender)

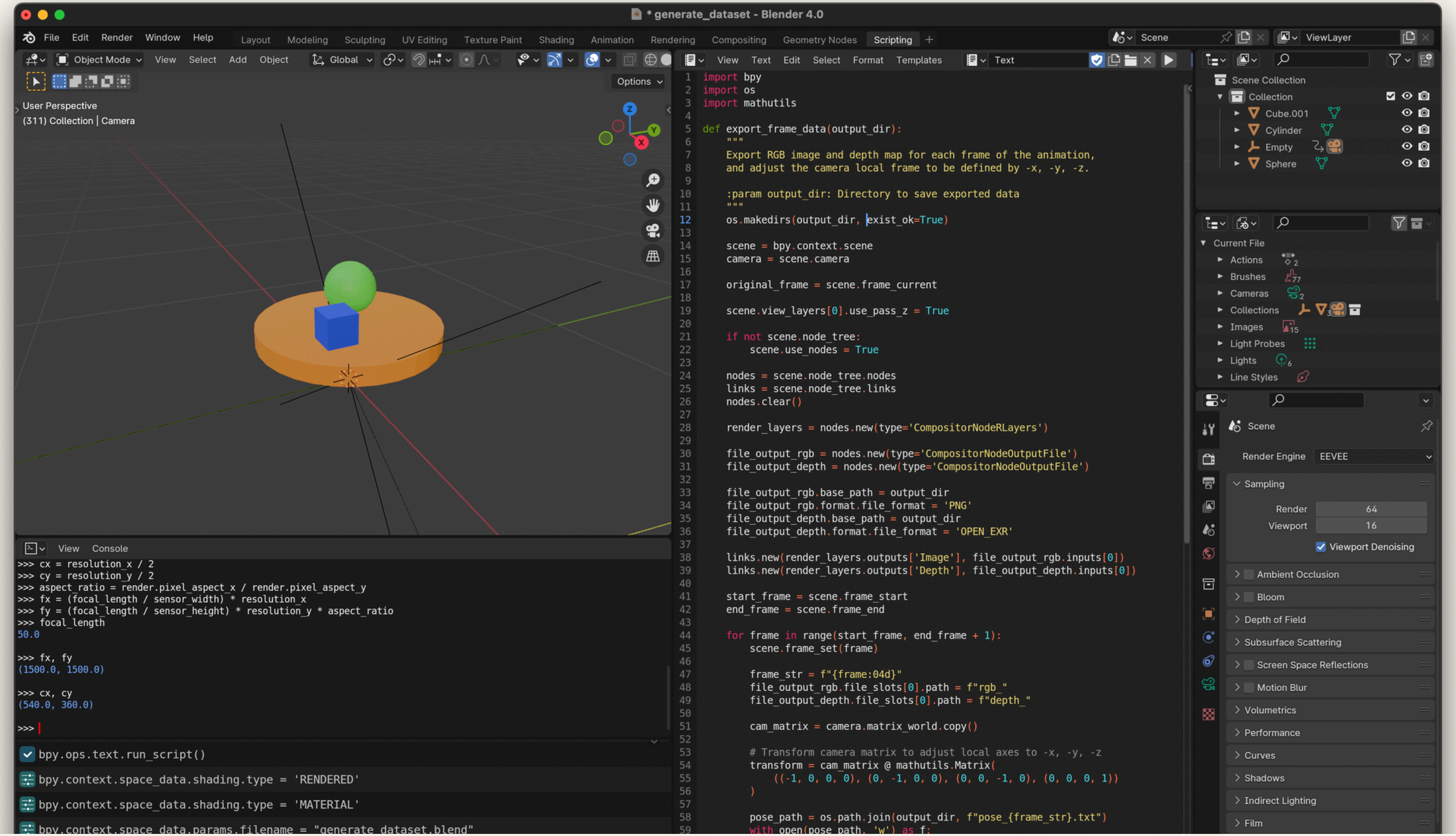


DATASETS : Synthetic

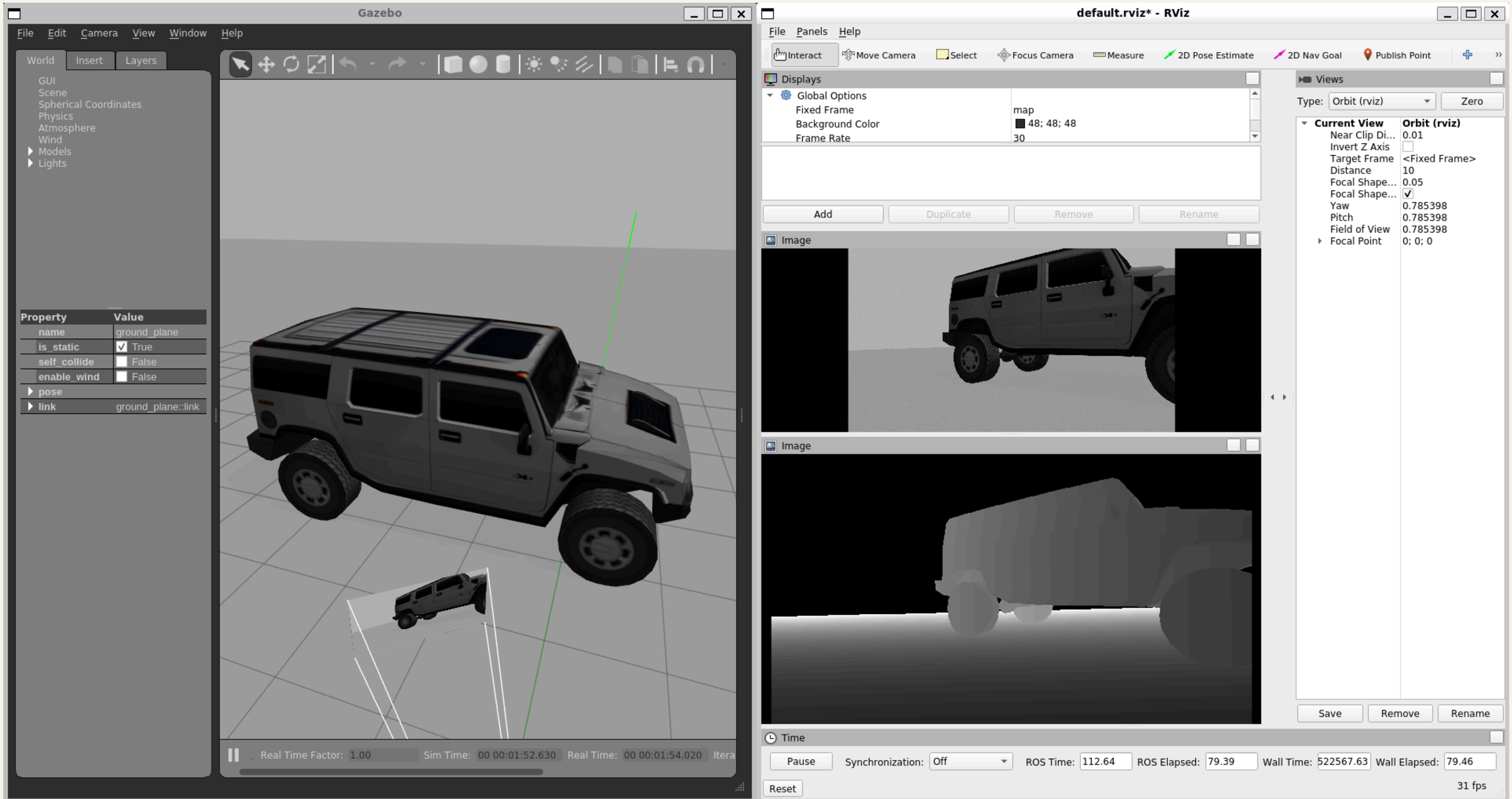
SUV (Gazebo)



Dataset Generation – Blender

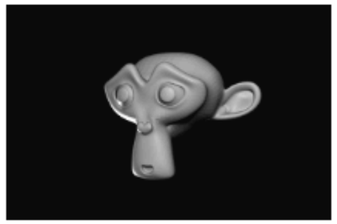


Dataset Generation – Gazebo

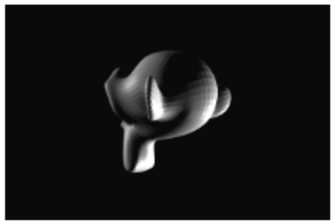


RESULTS

1. Mesh Generation:



(a) RGB Image 1

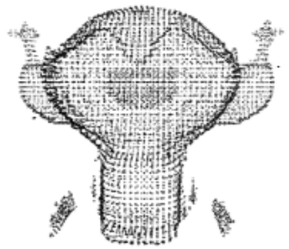


(b) RGB Image 2

(c) Depth Image 1

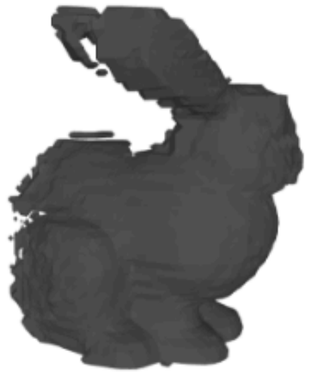
(d) Depth Image 2

(e) Reconstructed Mesh



(f) Reconstructed Point Cloud

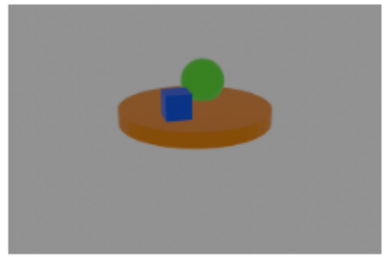
(a) Frames = 10



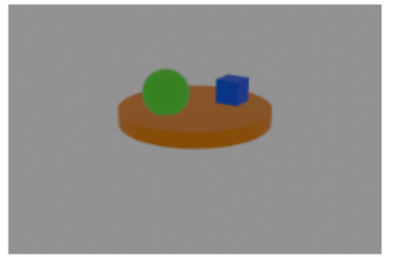
(b) Frames = 50

(c) Frames = 100

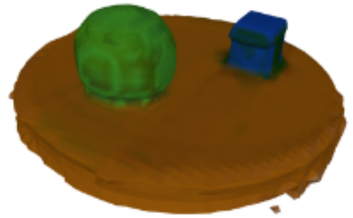
(d) Frames = 200



(a) RGB Image 1



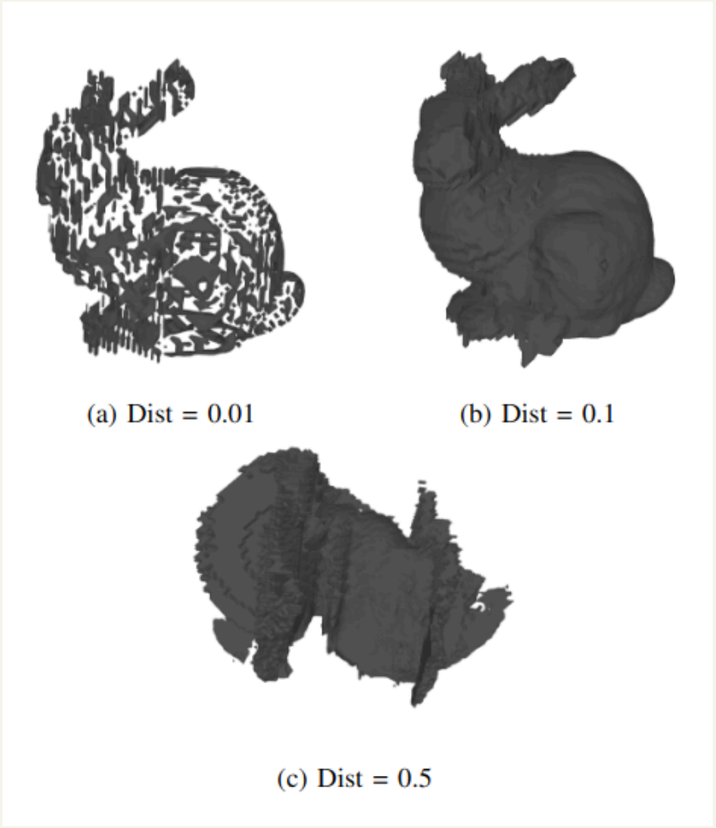
(b) RGB Image 2



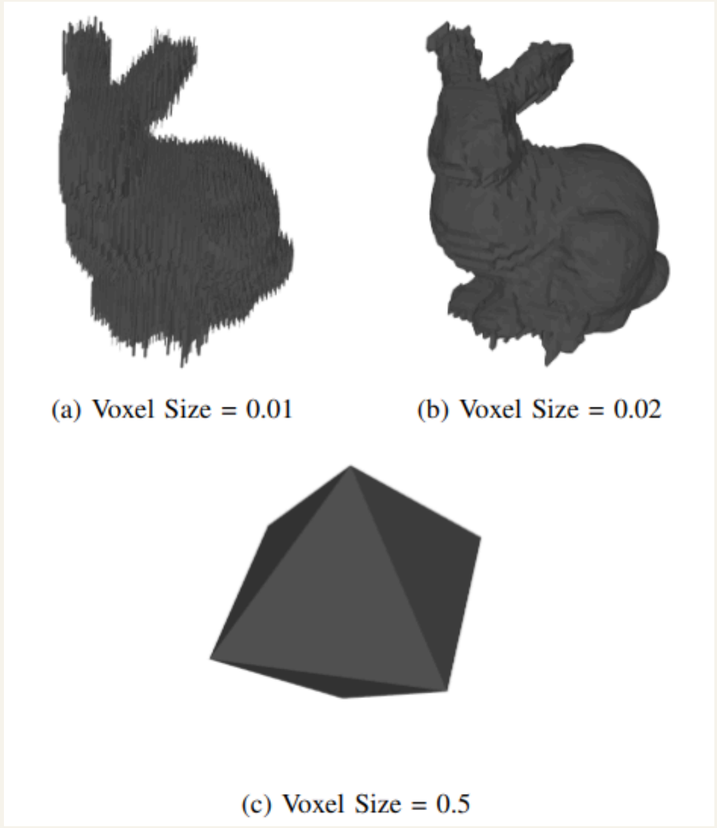
(c) Mesh for Textured Scene

RESULTS

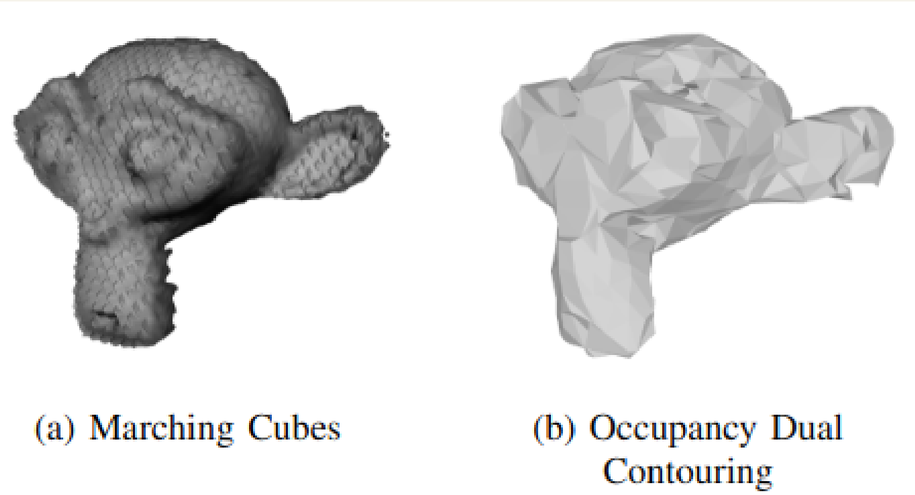
2. Hyperparameter Variations:



Truncation distance



Voxel Sizes



Generation Method

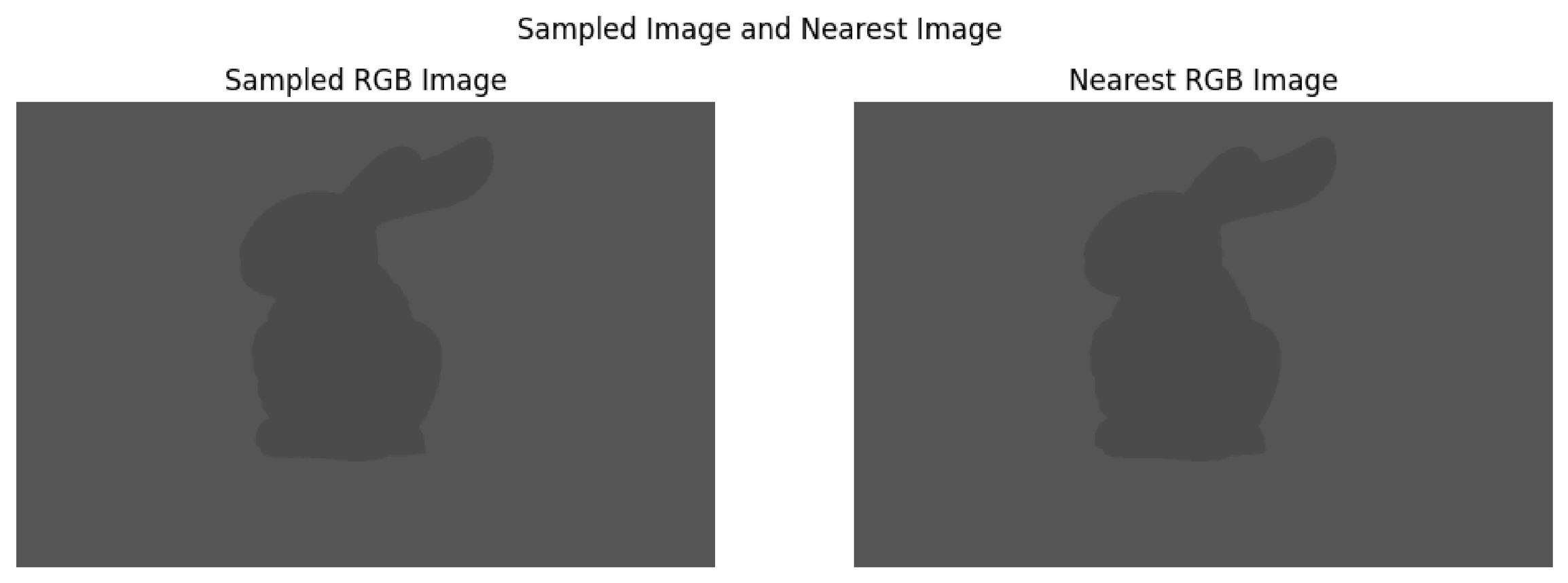
RESULTS

3. Visual Evaluation of Reconstruction:

Reconstructed Mesh

Corresponding Frame from Dataset

Pose Estimation



THANK YOU

GitHub Repo: https://github.com/anshium/mr-project