Weekly Report 2: Voxel Grid Creation and Visualization

Project Number: 18

1 Introduction

This week, we continued our 3D reconstruction workflow by learning about and implementing **voxel grid creation and visualization**. A **voxel** (volume element) is a 3D equivalent of a pixel, representing a small cube in a volumetric grid. Voxel grids form the basis for representing and analyzing 3D objects before performing operations such as carving and meshing.

2 Methodology

We implemented a voxel grid in Python using `NumPy` and visualized it in 3D using `Matplotlib`. The code consists of two main parts: grid creation and visualization.

2.1 Voxel Grid Creation

We created a 3D coordinate grid using `numpy.meshgrid`. The grid represents a cubic volume where each intersection point corresponds to the center of a voxel.

Implementation in Python:

```
import numpy as np

def create_voxel_grid(resolution=32):
    x = np.linspace(-1, 1, resolution)
    y = np.linspace(-1, 1, resolution)
    z = np.linspace(-1, 1, resolution)
    X, Y, Z = np.meshgrid(x, y, z, indexing='ij')
    return X, Y, Z
```

Explanation:

- The range from -1 to 1 defines a normalized 3D space.
- `np.meshgrid` generates 3D coordinate matrices.
- The `resolution` parameter controls how many points (voxels) the grid contains.

Learning outcome:

- Understood that voxels represent discrete samples of 3D space.
- Learned how the resolution affects spatial granularity.

2.2 Voxel Grid Visualization

To visualize the voxel grid, we sampled grid points and plotted them as 3D scatter points using `matplotlib`.

Implementation in Python:

```
import matplotlib.pyplot as plt

def visualize_voxel_grid(X, Y, Z, sample_rate=4):
    fig = plt.figure(figsize=(6, 6))
    ax = fig.add_subplot(111, projection='3d')

    ax.scatter(
        X[::sample_rate, ::sample_rate, ::sample_rate],
        Y[::sample_rate, ::sample_rate, ::sample_rate],
        Z[::sample_rate, ::sample_rate, ::sample_rate],
        color='royalblue', s=10
    )

    ax.set_title("Voxel Grid (3D Sampling of Space)")
    ax.set_xlabel("X-axis")
    ax.set_ylabel("Y-axis")
    ax.set_zlabel("Z-axis")
    plt.show()
```

Explanation:

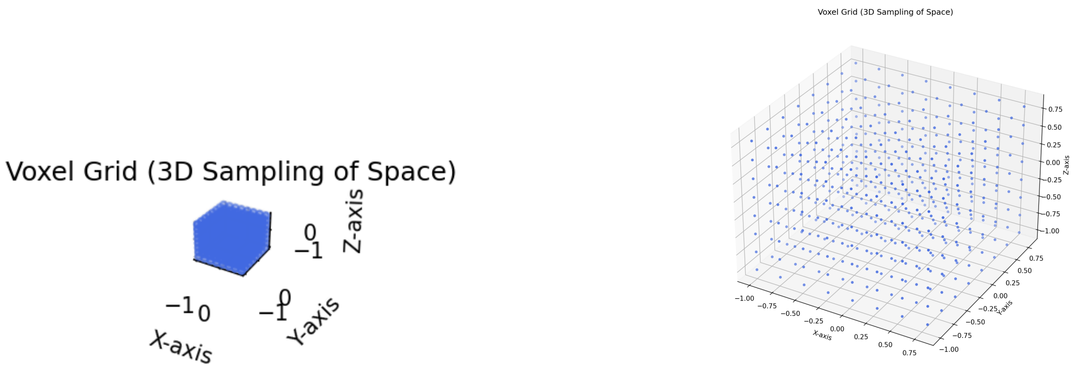
- Sampling every few points (`sample_rate`) makes visualization clearer.
- 3D scatter plotting provides a conceptual view of how voxels fill the space.
- Color and labeling make the structure easier to interpret.

2.3 Learning Outcomes

- Gained a clear understanding of voxel-based representation of 3D space.
- Learned how to create structured 3D coordinate grids using NumPy.
- Visualized spatial voxel distributions and their density.

3 Results

The voxel grid visualization produced by the code shows a 3D array of evenly spaced points representing voxel centers in a cubic space. The figure below shows a sample visualization of the voxel grid.



(a) 3D voxel grid visualization

(b) Close-up view showing voxel density

Figure 1: Voxel grid points generated and visualized using Matplotlib.

4 Conclusion

This week, we successfully:

- Created a voxel grid representation using NumPy.
- Visualized 3D voxel points to understand their distribution in space.
- Understood the importance of voxel grids as the foundation for future steps such as visual hull carving and surface meshing.

This marks the transition from 2D silhouette data to 3D volumetric representation in our reconstruction pipeline.