Updated Project Report: Computer-Generated Pen-and-Ink Illustration

Adarsh

December 12, 2023

1 Introduction

The primary objective of this project is to implement a rendering system utilizing a 3D Binary Space Partition (BSP) tree and PlanarMap for generating computer-generated pen-and-ink illustrations. While successful in implementing BSP trees for simple geometric shapes like a cube and a square, challenges persist in rendering more intricate shapes such as the bunny and Buddha. Additionally, the project has made significant progress in creating textures on the PlanarMap. Achieving fully realistic rendering remains an ongoing effort. The rendering pipeline, driven by robust OpenGL rendering techniques, stands out as a notable achievement.

2 Project Overview

2.1 Setting Up Basic Rendering

Accomplishments:

- Configured a windowing system and initialized OpenGL for basic rendering.
- Successfully established the rendering pipeline to display simple geometric shapes like a cube and a square.

2.2 Implementing 3D BSP Tree

Accomplishments:

- Successfully implemented a 3D BSP tree for efficient sorting and culling of polygons.
- Achieved successful rendering of simple geometric shapes, such as a cube and a square.
- Applied texture through mapping the texture successfully.
- Introduced partitioning in the BSP tree to enhance handling of complex shapes.

Challenges Faced:

- Encountered difficulties in rendering more complex shapes like the bunny and Buddha.
- Addressed challenges in achieving realistic rendering for intricate models.
- Ongoing efforts in optimizing partitioning for complex shapes.

2.2.1 Code Snippet - BSP Tree Implementation

```
// Code snippet for 3D BSP tree implementation
BSPtree cubeBSP(cubeVertices, cubeTextureCoords);
cubeBSP.render(modelTransform, textureID);
```

2.3 Creating PlanarMap

Accomplishments:

- Implemented a PlanarMap to encode polygon adjacencies, establishing a 2D partition in normalized device coordinates (NDC).
- Successfully integrated textures into the PlanarMap.

Challenges Faced:

- Challenges persist in achieving fully realistic rendering.
- Ongoing efforts to synchronize PlanarMap structures with more complex models.

2.3.1 Code Snippet - Planar Mapping

```
// Code snippet for Planar Mapping with improved strokes
PlanarMap planarMap;
planarMap.addFace(cubeVertices);
planarMap.generateOutlineStrokes(shaderProgram, textureID);
```

2.3.2 Code Snippet - Stroke Coordination

```
// Code snippet for coordinating strokes with transformed matrices
coordinatedStrokePositions(modelTransform, viewMatrix);
```

Figure 1: Caption for Image 1 (Cube)

2.4 Adding Strokes to PlanarMap

Accomplishments:

- Successfully integrated strokes into the PlanarMap for improved visual representation.

- Coordinated stroke positions with transformed model and view matrices.

Challenges Faced:

- Ongoing efforts to manage stroke colors and textures for enhanced realism.
- Addressing challenges in coordinating strokes with complex models.

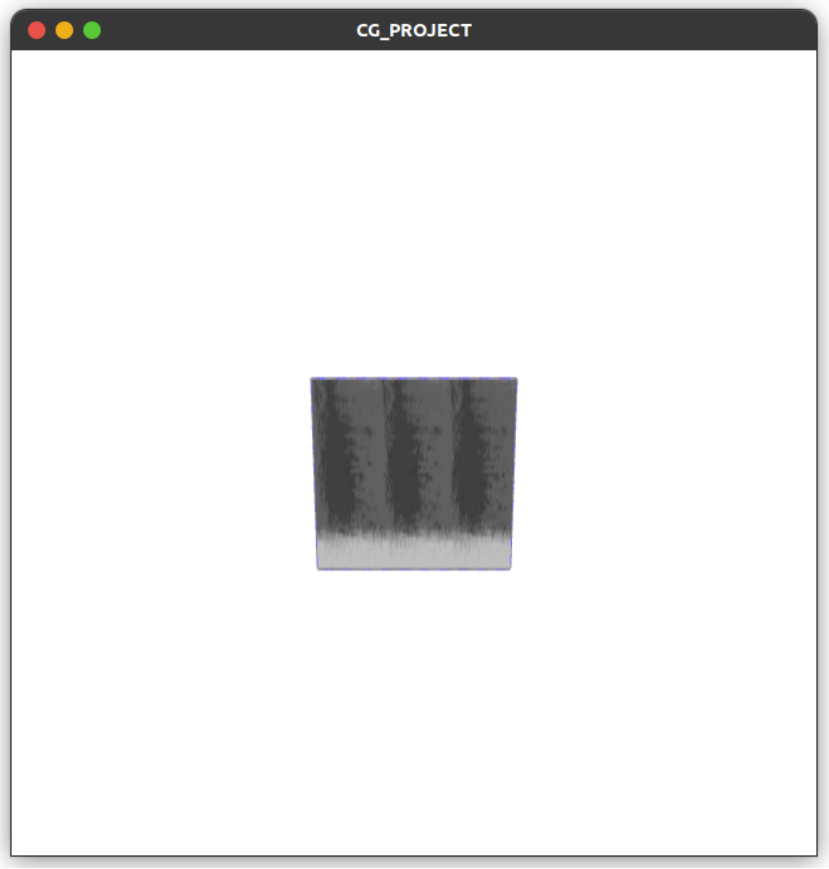

Figure 2: Caption for Image 2

3 Conclusion

While successful in implementing the BSP tree for simple geometric shapes and creating textures on the PlanarMap, challenges persist in rendering more complex shapes and achieving fully realistic rendering. The addition of partitioning in the BSP tree addresses some challenges, providing a foundation for

further enhancements. The project, powered by a robust rendering pipeline, continues to evolve, opening avenues for further exploration in computer-generated illustration techniques within the realm of computer graphics.

The ongoing efforts in this project aim to overcome challenges and elevate the capabilities of computer-generated pen-and-ink illustration techniques.