Assignment 4

Cinematic Navigation

Intelligent Autonomous Video Generation from 3D Gaussian Splatting

Introduction to Computer Vision

Fall Semester 2025

Made by Mahmoud Mousatat

"Think like a filmmaker, engineer like a computer scientist, and create like an artist."

Contents

1 A Vision for Tomorrow

1.1 The Problem

Imagine you're a documentary filmmaker tasked with creating a stunning visual tour of the world's most inaccessible places—ancient ruins, underwater caves, distant planets. Traditional filmmaking requires expensive equipment, dangerous expeditions, and massive crews. But what if we could capture these environments once and then explore them infinitely, creating perfect cinematic experiences without ever returning?

Enter **3D Gaussian Splatting**: a revolutionary technique that captures real-world environments in stunning photorealistic detail. Museums, archaeologists, and real estate companies are using it to preserve and showcase spaces. But there's a catch—navigating these vast 3D reconstructions to create compelling videos is tedious, manual, and requires artistic expertise that doesn't scale.

1.2 The Solution

You are building an **Intelligent Cinematic Agent**—an AI-powered virtual cinematographer that can:

- Autonomously explore and understand 3D environments
- Identify interesting objects and subjects
- Plan smooth, professional camera paths
- Avoid obstacles while maintaining cinematic quality
- Generate multiple video perspectives automatically

Your agent will democratize professional filmmaking in virtual spaces, enabling anyone to create museum-quality virtual tours, real estate showcases, or documentary footage from 3D reconstructions.

2 Assignment Overview

In this assignment, you will develop a complete pipeline for autonomous cinematic navigation and video generation from 3D Gaussian Splatting scenes. Your system must work with **any** 3D Gaussian Splatting scene, demonstrating robustness and generalizability.

2.1 Core Components

2.1.1 Scene Exploration Agent

- Autonomous exploration strategy
- Spatial understanding and mapping
- Coverage optimization

2.1.2 Computer Vision Pipeline [Optional - 15% Bonus]

- Object detection and recognition
- 3D localization of detected objects (or maybe 2D localization is enough)
- Semantic understanding of the scene

2.1.3 Path Planning System

- Professional camera trajectory generation
- Obstacle detection and avoidance
- Smooth motion with cinematic qualities (ease-in/ease-out, proper speed)

2.1.4 Video Generation

- Rendering engine integration
- Frame interpolation and stabilization
- Professional video export

3 Input Data Format

Your system will work with **3D Gaussian Splatting models** stored in `.ply` format. These files contain the 3D scene representation with Gaussian splats that can be rendered in real-time.

Visualization: You can preview and explore your `.ply` models using SuperSplat, an online viewer for Gaussian Splatting scenes. This tool is helpful for understanding the scene structure before running your pipeline.

Important Notes:

- The PLY files are in compressed format, which might not be editable in other software. Export them from SuperSplat as PLY without compression to handle them properly.
- You are **not allowed** to edit or change orientation, translation, or scale of the 3D scene using any editor.
- You are allowed to make transformations on the PLY after calculating the matrix manually using software.

4 Technical Requirements

4.1 Phase 1: Scene Understanding

Objective: Your agent must intelligently explore and understand the 3D scene.

Requirements:

- Implement an exploration strategy that systematically discovers the environment
- [Optional - 15% Bonus] Use computer vision techniques to detect objects in the scene

4.2 Phase 2: Path Planning & Navigation

Objective: Generate professional, smooth camera paths for scene exploration.

Requirements:

- Plan smooth camera paths to explore and cover the entire scene
- Implement obstacle avoidance (don't pass through walls, objects, etc.)
- Ensure smooth transitions (no jerky movements)
- [Optional - 15% Bonus] Randomly select **3 or more objects** from your detected catalog and navigate between them cinematically
- [Optional - 15% Bonus] Maintain proper framing of detected objects during navigation

4.3 Phase 3: Video Generation

Objective: Produce high-quality video output(s). Video 1 is required; Video 2 is optional for bonus points.

4.3.1 Video 1: Comprehensive Panorama Tour (360° Environment) [Required]

Duration: 60–120 seconds
Purpose: Give a complete overview of the environment

Requirements:

- Cover major areas of the scene
- Smooth, professional camera movements
- Showcase the full scope of the environment
- Cinematic pacing (not too fast, not too slow)

4.3.2 Video 2: Object-Focused Cinematic Tour [Optional - 15% Bonus]

Duration: 45–90 seconds
Purpose: Navigate between and showcase selected objects

Requirements:

- Visit each selected object in sequence
- Professional approach and framing for each object

- Smooth transitions between objects
- Highlight the objects in an aesthetically pleasing way
- Avoid obstacles during navigation

5 Creative Freedom

This assignment encourages creativity! Consider:

- **Camera Styles:** Drone-like movement, handheld simulation, crane shots, orbits, etc.
- **Pacing:** Slow and dramatic vs. dynamic and energetic
- **Object Selection:** Random, or biased towards interesting objects (manual selection will be a plus) [Optional - 15% Bonus]
- **Additional CV Techniques:** Depth-based effects, attention mechanisms, saliency detection [Optional - 15% Bonus]
- **Visualization:** Show detected objects, paths, or just the final cinematic result [Optional - 15% Bonus]
- **Audio:** Add sound effect design (optional bonus!)

6 Deliverables

6.1 Submission Structure

Your submission should follow this directory structure:

```
your_project/
        Report.pdf                # How did you solve the problem?
                                # Highlight novelty and proud tricks
        README.md                 # Setup and usage instructions
        requirements.txt          # Python dependencies
        src/                      # Below is an example but not obligatory
             explorer.py           # Scene exploration agent
             detector.py           # Object detection module
             path_planner.py       # Path planning algorithms
             renderer.py           # Video rendering
             main.py               # Main pipeline
        configs/
             config.yaml           # Configuration file (optional)
        outputs/
           scene_1/
               panorama_tour.mp4       # Video 1 (Required)
               object_tour.mp4         # Video 2 (Optional - 15% Bonus)
               detected_objects.json   # Optional - 15% Bonus
               selected_objects.json   # Optional - 15% Bonus
```

6.2 Documentation

6.2.1 README.md

Must include:

- Installation instructions
- Usage guide with examples
- Algorithm descriptions
- Dependencies and requirements

- Known limitations

6.2.2 Technical Report

Separate document covering:

- How did you solve the problem?
- Highlight novelty and the most proud tricks
- Challenges faced and solutions
- Results and evaluation
- **[IMPORTANT REQUIREMENT]** Your vision for future improvements: Discuss how this solution could be enhanced, extended, or improved in future work. Include both technical improvements and creative directions.

7 Grading Criteria

Your submission will be evaluated based on the following criteria:

Weight	Criteria
40%	**Fully Functional Pipeline** • End-to-end system integration • Robustness and reliability • Proper error handling
+15% Bonus	**Object Detection and Navigation [Optional Bonus]** • Accuracy of object detection • Quality of 3D/2D localization • Navigation between detected objects • Video 2: Object-Focused Cinematic Tour
30%	**Path Planning and Scene Exploration** • Effectiveness of exploration strategy • Quality of camera trajectories • Obstacle avoidance implementation • Smoothness of motion
30%	**Artistic/Professional/Innovative/Creative Result Videos** • Cinematic quality • Professional presentation • Creative approach • Visual appeal

Table 1: Grading breakdown for Assignment 4 (Total: 100% + 5% Bonus)

8 Testing Your Implementation

Your code will be tested on **multiple 3D Gaussian Splatting models** with different characteristics:

- Indoor scenes
- Outdoor environments

Important (as bonus but optional): Your solution should be robust and generalizable across different scene types. Test early and often with diverse scenes!

9 Tips for Success

1. **Start Simple:** Get basic rendering working first, then add intelligence
2. **Modular Design:** Separate exploration, detection, planning, and rendering
3. **Incremental Development:** Test each component independently
4. **Visualization:** Create debug visualizations to understand your agent's behavior
5. **Parameter Tuning:** Make hyperparameters configurable (speed, detection threshold, etc.)
6. **Error Handling:** Handle edge cases gracefully (no objects detected, obstacles everywhere, etc.)
7. **Performance:** Consider computational efficiency—you may need to process large scenes
8. **Technical Stack:** Choose the engine, and already built algorithms and libraries wisely

10 Bonus Challenges

Do all the [Optional - 15% Bonus] and you will get this bonus on the HW4

For extra [5% Bonus] credit on the entire course, consider:

- **Semantic scene understanding:** Group objects by room or area
- **Interactive mode:** Allow user to select objects instead of random selection
- **Style transfer:** Apply different cinematic filters or moods
- **Real-time preview:** Show live navigation preview
- **Professional orientation:** 2D Cinematic professional rendering instead of 360 (Panorama)

Conclusion

Good luck, and create something amazing! 🎞 ★

Remember: The goal isn't just to complete the requirements—it's to build something you're proud to showcase. Think like a filmmaker, engineer like a computer scientist, and create like an artist.