Gaussian Splatting Platform

Complete User Manual (WSL + FastAPI Local Dashboard)

---

1. Overview

The Gaussian Splatting Platform is a local web-based dashboard that allows you to:

- Run your `gs_cli.py` reconstruction pipeline
- Manage multiple reconstruction jobs
- Monitor live logs
- Configure COLMAP + training parameters
- Resume training from checkpoints
- Organize workspaces visually

The platform runs entirely locally on:

- WSL Ubuntu
- Conda environment (e.g. `gs`)
- Your Gaussian Splatting repository

It acts as a visual control panel for:

Video → Frame Extraction → COLMAP → Undistortion → 3D Gaussian Splatting Training

---

2. System Architecture

The platform consists of:

- `server.py` (FastAPI backend)
- `templates/` (UI files)
- `jobs_db/` (stores job metadata and logs)
- Your external `gs_cli.py` pipeline

When you press "Run":

1. The UI collects form inputs.
2. FastAPI validates parameters.
3. A job folder is created in `jobs_db/`.
4. The command is constructed.
5. A background process runs `gs_cli.py run ...`.
6. Logs stream live to the browser.

---

3. Starting the Platform

Step 1 – Open WSL

Open terminal and enter WSL.

Step 2 – Activate Environment

```
conda activate gs
```

Step 3 – Navigate to Platform

```
cd ~/gs_platform/app
```

Step 4 – Start Server

```
uvicorn server:app --host 0.0.0.0 --port 7860
```

If successful, you will see:

"Uvicorn running on http://0.0.0.0:7860"

Step 5 – Open in Browser

Open:

http://127.0.0.1:7860

---

4. Understanding the Interface

The interface contains two main panels:

LEFT: New Job Configuration RIGHT: Jobs Dashboard

---

5. Creating a New Reconstruction Job

You must configure:

- Workspace
- GS Repo Path
- Input (Video OR Images Directory)
- COLMAP Settings
- Training Settings

---

6. Workspace

The workspace is your project folder.

Example:

/mnt/c/gs_data/projects/horse_auto

This folder will contain:

- Extracted images (if using video)
- COLMAP outputs
- Undistorted images
- Gaussian splatting outputs
- Checkpoints

Each project should have its own workspace.

---

7. GS Repo

This must be the folder that contains:

- gs_cli.py

Example:

/home/aditya/gaussian-splatting

If this is wrong, the job will fail immediately.

---

8. Input Options

You must choose ONLY ONE:

Option A – Video Option B – Images Directory

If both are provided, the platform will reject the job.

---

9. Running from Video (Recommended)

Step-by-step:

1. Choose Workspace
2. Choose Video file
3. Leave Images dir empty
4. Set FPS
5. Set JPG quality
6. Configure COLMAP
7. Configure Training
8. Click Run

Video must be in WSL path format:

/mnt/c/Users/Guest_/Downloads/Auditorium.mp4

---

10. Extraction Settings

FPS:

- Controls frame extraction rate
- Higher FPS = more frames = slower COLMAP
- Recommended: 1–3

JPG Quality:

- 1 = best quality
- 2–3 typical

---

11. Running from Images Directory

Use this if frames are already extracted.

Steps:

1. Choose Workspace
2. Leave Video empty
3. Set Images dir
4. Configure COLMAP
5. Configure Training
6. Click Run

Example images directory:

/mnt/c/gs_data/projects/bultt/images

---

12. COLMAP Settings

Matcher Options:

Sequential:

- Best for videos
- Matches nearby frames
- Uses Overlap parameter

Exhaustive:

- Matches all pairs
- Very slow for many frames

Overlap:

- Controls frame matching distance
- Typical: 15–30

Loop Detection:

- Enable if you walk in a circle
- Helps close loops

Camera Model:

SIMPLE_RADIAL (Recommended) PINHOLE OPENCV

Single Camera:

- Enable for single phone camera

Use GPU:

- Enable if COLMAP CUDA build installed

13. Training Settings

Iterations:

- Total training steps
- Typical: 30,000–60,000

Save Iters:

- Saves point clouds at specified iterations

Checkpoint Iters:

- Saves training checkpoints

Resume:

- Continue training from last checkpoint

14. Monitoring Jobs

Jobs panel auto-refreshes.

Each job shows:

- Status badge (color-coded)
- PID
- Return code
- Workspace path
- Input source

Click "Open" to view:

- Command used
- Live log
- Status

15. Job States

queued – waiting to start running – actively processing done – finished successfully error – process failed stopped – manually terminated

16. Live Log

The log auto-refreshes every few seconds.

You can observe:

- Frame extraction progress
- COLMAP progress
- Bundle adjustment
- Training iterations
- Errors

17. Resuming Training

To continue from 30k → 60k:

1. Select same workspace
2. Set iterations to 60000
3. Enable resume

4. Ensure checkpoint exists
5. Run job

18. Opening Result in Viewer (Windows)

Example:

```
cd "C:\Users\Guest_\OneDrive\Desktop\viewers\bin" .\SIBR_gaussianViewer_app.exe --model-path "C:\gs_data\projects\horse_auto\gs_out" --iteration 60000
```

19. Troubleshooting

Problem: Both Video and Images provided Solution: Use only one

Problem: gs_repo missing Solution: Provide correct path

Problem: Job fails immediately Solution: Check paths exist in WSL

Problem: COLMAP fails Solution:

- Reduce FPS
- Increase overlap
- Try SIMPLE_RADIAL

Problem: JSON decode error Solution:

- Delete corrupted job folder
- Re-run

20. Best Recommended Preset

For phone walkthrough:

FPS: 2 JPG Quality: 1 Matcher: sequential Overlap: 20–25 Loop Detection: Off (unless circular motion) Camera Model: SIMPLE_RADIAL Single Camera: Enabled Iterations: 30000 → Resume to 60000 Save Iters: 7000 10000 15000 20000 30000 Checkpoint Iters: 10000 30000 60000

21. Advanced Notes

- Always keep each project in separate workspace
- Do not reuse workspace for different scenes
- Lower FPS for long videos
- Use GPU build of COLMAP for best speed

22. Workflow Summary

1. Start server
2. Configure workspace
3. Choose input
4. Configure settings
5. Run
6. Monitor
7. Open result in viewer

End of Manual