University of Applied Sciences and Arts Northwestern Switzerland
School of Architecture, Civil Engineering and Geomatics

MASTER OF SCIENCE IN ENGINEERING

Towards freely navigable street view services

Mobile mapping services like Google Street View, Cyclomedia's Street Smart, and iNovitas' infra3D provide intuitive virtual navigation with panoramic images. Some, like infra3D, enable direct measurements using depth maps. While users can navigate easily, they are restricted to the original data acquisition path, usually limited to street-level views. In contrast, point cloud viewers allow full six degrees of freedom (6DOF) movement but often lack the detail of images, making small features harder to recognize.

To enhance virtual representations, textured 3D meshes or novel view synthesis methods like NeRF or 3D Gaussian Splatting (3DGS) can be used, combining multiple datasets from different capture methods (e.g., cars, backpacks, drones). This thesis aims to integrate 6DOF navigation into a street view service while maintaining image-level detail, focusing on web-based visualization and evaluating libraries for production use.

Data processing

The raw dataset was processed into three object representations: point cloud, textured mesh, and 3D Gaussian Splats, each evaluated for visualization and interaction in a freely navigable street-view context.

- **Point Cloud:** Converted to an XYZ file and a 3D tile set for compatibility. The dataset initially contained 12.2 million points, down sampled to 6 million for efficiency. The 3D tile set, generated using py3dtiles, comprised 1,750 tiles for smooth web rendering.
- **Textured Mesh:** Generated in Agisoft Metashape, covering 200×200 m, resulting in a 92M-triangle mesh (2.2 GB). Optimized versions with 10M and 5M triangles (268 MB & 145 MB) were created for web-based visualization.
- **Gaussian Splats:** Computed using Jawset Postshot, including 3 million splats after 4 hours of processing. Alignment and coordinate system adjustments were required for proper integration.

Evaluation of Object Representation

The evaluation of object representations (point cloud, textured mesh, and Gaussian splats) was conducted using both qualitative and quantitative measures. Screenshots from identical viewpoints were captured for comparison, and an unused image was used as ground truth. Quantitative assessment was performed using Mean Squared Error (MSE), Structural Similarity Index (SSIM), and Peak Signal-to-Noise Ratio (PSNR) to measure accuracy and visual fidelity. The images used for evaluation are shown in Figure 1, while the results are summarized in Table 1.

	MSE ↓	SSIM ↑	PSNR ↑	Artifacts ↓	FPS ↑	Processing Time in min ↓
Point Cloud	11005	0.21	7.71 dB	Low	60	5
Mesh	10542	0.33	7.90 dB	High	60	180
Gaussian Splats	1117	0.57	17.65 dB	High	8	240

Table 1: Qualitative and quantitative results of the comparison of the different object representations. There is not a single object representation, which is the best in all the metrics.

Development and Evaluation of Viewers

Interactive viewers were developed using three.js and Unity to explore point clouds, meshes, and Gaussian splats. The three.js viewer leverages the 3D-Tiles specification via the 3DTilesRendererJS library (NASA AMMOS, 2024a), enabling efficient streaming of large point cloud datasets. Gaussian splats were integrated using the GaussianSplats3D plugin (Kellogg, 2025), allowing for high-fidelity visualization directly in the browser.

The Unity viewer, while offering advanced rendering capabilities, relies on WebAssembly for web deployment. It initially attempted to use the Unity3DTiles plugin (NASA AMMOS, 2024b) for 3D-Tiles support, but differences in coordinate systems posed challenges. Gaussian splats were visualized using the UnityGaussianSplatting plugin (Pranckevičius, 2024), but its WebGL limitations restricted real-time rendering performance. Due to these constraints, only the visualization and measurement in the point cloud is supported.

Fig. 1: Comparison of 3D representation methods from the same viewpoint and extent. (a) Ground truth image for reference, not used for the calculation of the other object representations. (b) Point cloud visualization showing discrete points and lower structural density. (c) Mesh representation providing continuous surfaces but with texture artifacts. (d) Gaussian splatting approach achieving smooth surfaces and high detail, albeit with higher computational costs and floaters (blurred parts).

Integration in infra3D

Due to the difficulties with the Unity Web-Build, the integration in infra3D has been done using three.js. The integration into infra3D has introduced two viewing modes: a side-by-side viewer for comparing street-level imagery with 3D representations and an integrated viewer for immersive exploration. While both implementations enhance usability, challenges remain regarding cognitive load, performance optimization, and transition smoothness. Future improvements, such as better rendering techniques and usability refinements, will further enhance the system's scalability and user experience.

Conclusion & Outlook

This thesis explored data processing techniques, viewer implementations, and object representations for a freely navigable street-view service. It highlighted the trade-offs between point clouds, meshes, and Gaussian splats, as well as the strengths and limitations of three.js and Unity for web-based visualization. The findings demonstrate that integrating a hybrid approach and extending infra3D with a 3D viewer could enhance navigation, measurement, and urban planning applications, with future advancements in Gaussian splatting and web rendering frameworks further improving performance and usability.

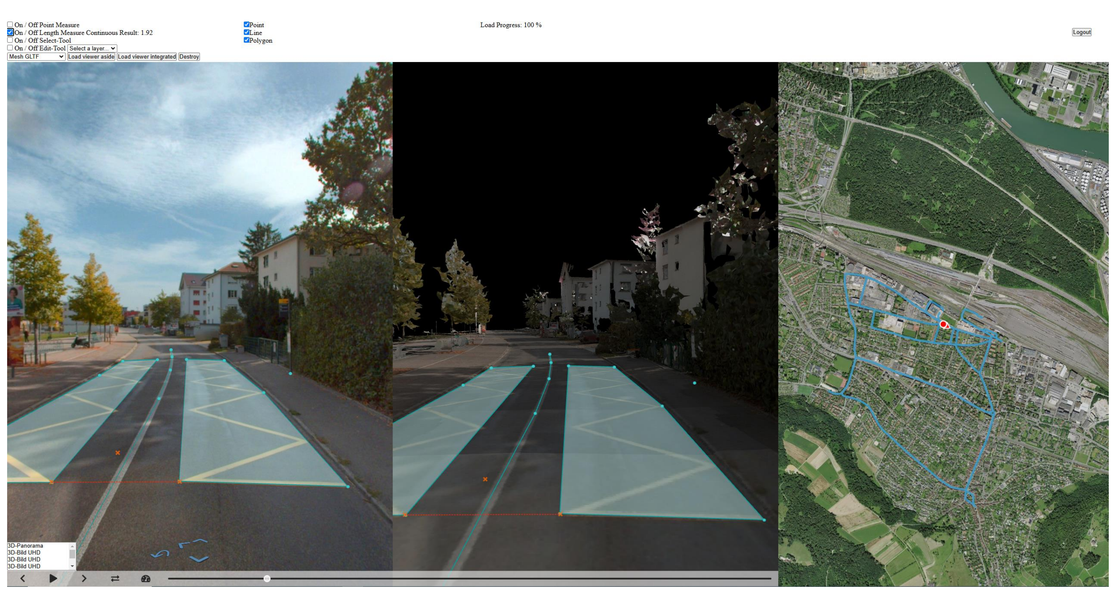

Fig. 2: Side-by-side viewer implementation in infra3D, showing the split-screen layout with street-level imagery and the three.js viewer. Further layers (blue objects) and measurements (orange line) are synchronized in real time between both viewers.

Student: Andreas Eisenring, andi2604@hotmail.com

Supervisor: Prof. Dr. Stephan Nebiker, stephan.nebiker@fhnw.ch
Expert: Robert Wüest, robert.wueest@fhnw.ch
Tutor: Jonas Meyer, jonas.meyer@fhnw.ch
Project Partner: Joel Burkhard, joel.burkhard@inovitas.ch

Literature:
Kellogg, M. (2025). 3D Gaussian splatting for Three.js. Retrieved September 19, 2024, from https://github.com/mkkellogg/GaussianSplats3D
Pranckevičius, A. (2024). Aras-p/UnityGaussianSplatting. Retrieved October 11, 2024, from https://github.com/aras-p/UnityGaussianSplatting
NASA AMMOS. (2024a). NASA-AMMOS/3DTilesRendererJS. Retrieved October 21, 2024, from https://github.com/NASA-AMMOS/3DTilesRendererJS
NASA AMMOS. (2024b). NASA-AMMOS/Unity3DTiles. Retrieved October 21, 2024, from https://github.com/NASA-AMMOS/Unity3DTiles