Dibris Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi

Augmented and Virtual Reality

Report

Temporal 3D Reconstruction of Cultural Heritage using Gaussian Splatting

---

Advisor: Gianni Vercelli, Luca Martini

UNIVERSITY OF GENOA, JANUARY 2025

Member list & Workload

No.	Full name	Student ID	Percentage of work
1	Bianca Perasso	4994475	100%
2	Lorenzo Mesi	5070110	100%
3	Martin Martuccio	5061529	100%
4	Samuele Pellegrini	4951487	100%

Contents

1 Concept Overview **5**

2 Target Audience **5**

3 Example Use Case **5**
3.1 Conservation and Environmental Studies . . . 5
3.2 Historical Monument Restoration . . . 5
3.3 Educational Applications . . . 6

4 Technical Information **6**

5 Development Plan **6**

6 State of Art **7**
6.1 Platforms for 3D Cultural Heritage Visualization . . . 7
6.2 Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction . . . 7

7 Tools **8**

8 Description **9**
8.1 Data Collection . . . 9
8.2 Preprocessing . . . 9
8.3 Model Processing . . . 10
8.3.1 CloudCompare . . . 10
8.3.2 MeshLab . . . 11
8.3.3 Coding . . . 11

9 Results **13**
9.1 Model Merging Accuracy . . . 13
9.2 Visualization of Differences . . . 13
9.3 Software Comparisons . . . 13
9.4 Limitations . . . 13

10 Conclusions **14**

1 Concept Overview

- **Brief Description:** The Temporal 3D Reconstruction of Cultural Heritage using Gaussian Splatting project leverages advanced Gaussian Splatting (GS) techniques to create and visualize temporal changes in cultural heritage sites. By modeling transformations caused by natural degradation or environmental factors, the project generates a "time capsule" that digitally preserves and analyzes these sites over time.
- **Objectives:**
  - Reconstruct high-quality 3D models of cultural heritage sites using GS techniques.
  - Capture and highlight temporal changes by either partially retraining GS models or integrating multiple time-based models.
  - Develop visualization methods to effectively showcase changes, ensuring user-friendly analysis.
- **Key Feature:** The innovative use of Gaussian Splatting for temporal visualization, by employing techniques like color coding or transparency effects to indicate areas of change, enables users, through e.g. interactive timelines, to intuitively explore and analyze changes across different time periods.

2 Target Audience

- **Primary Users:** Researchers and professionals in the preservation of cultural heritage who aim to analyze and visualize temporal changes in historical sites. This includes archaeologists, conservationists, and digital humanities scholars.
- **Secondary Users:** Educators, students, and enthusiasts interested in understanding the impact of environmental and temporal factors on cultural heritage through interactive 3D models.

3 Example Use Case

3.1 Conservation and Environmental Studies

A conservationist aims to analyze the temporal erosion caused by environmental factors. They upload historical and recent 3D models and the software generates a composite model that highlights changes. A timeline slider allows them to visualize these changes dynamically. This detailed visualization helps in developing targeted conservation strategies.

3.2 Historical Monument Restoration

A restoration team uses the software to compare the current state with previous models. The application highlights deteriorated or altered areas. This allows restoration to be planned precisely, focusing resources on specific areas. Interactive tools help promote informed decision-making.

3.3 Educational Applications

In an educational environment, students learn about the construction phases of iconic monuments. By interacting with 3D models that show various stages of construction over time, students gain a deeper understanding of architectural evolution.

4 Technical Information

- **Technologies & Software:** Blender 4.3.0, Postshot 0.5.10, CloudeCompare 2.13.2, MeshLab 2023.12, ChatGPT, Claude and DeepSeek V3;
- **Platforms:** ImageToSTL and Polycam;
- **Programming Languages:** Python 3.11;
- **Frameworks:** Anaconda 24.9.2 (Conda 3), Open3D 0.18, NumPy, PLY Format, Node.js Visual Studio Code;
- **Cloud:** GitHub, Overleaf, Google Colab and Microsoft Teams.

5 Development Plan

- **Team and Roles:** The project began with a collective study and research phase focused on 3D formats and their characteristics, followed by an exploration of Gaussian Splatting techniques. The team then collaborated on identifying suitable models, while the search for algorithms and software was conducted individually by each member.

  The simulation of erosion was primarily handled by Bianca Perasso and Samuele Pellegrini. During the development and experimentation phase, Bianca Perasso took the lead in working with MeshLab, while Lorenzo Mesi focused on CloudCompare. The coding tasks were mainly carried out by Martin Martuccio and Samuele Pellegrini. Finally, the report was collaboratively written by all team members, with each person contributing their findings and results.

  It is important to emphasize that all team members actively participated in every phase of the project. The above roles are mentioned to highlight areas where individuals dedicated more effort, but the work was truly a collective effort.

- **Timeline:** The final version must be completed by end of January or February 2025 at the latest.

6 State of Art

In recent years, the digital preservation of cultural heritage has been revolutionized by advanced 3D reconstruction and visualization technologies. These technologies leverage computer vision, machine learning, and graphics to document and analyze historical sites with high precision and detail. Techniques like Gaussian Splatting (GS) have introduced efficient, high-fidelity 3D models, enhancing cultural heritage preservation.

These advancements tackle key challenges in heritage conservation, such as capturing subtle environmental changes and providing comprehensive, non-invasive digital documentation. Beyond mere recording, these technologies help understand the relationships between cultural artifacts and their environments, predict risks, and plan conservation efforts. They enable early detection of structural changes and environmental impacts, promoting proactive management over reactive maintenance.

Our research focuses on developing innovative visualization techniques to document changes in cultural heritage sites over time, creating a "digital time capsule" for future preservation.

6.1 Platforms for 3D Cultural Heritage Visualization

Existing platforms for public 3D visualization of cultural heritage primarily offer point cloud models and 3D scans, allowing researchers and the public to explore digital representations of historical sites and artifacts. The key platforms we have identified and analyzed are:

- **Sketchfab**: A widely-used platform hosting 3D models from museums, researchers, and cultural institutions worldwide.
- **OpenHeritage3D**: A specialized repository focusing on archaeological and historical 3D reconstructions.
- **Europeana Collections**: An official European platform aggregating digital cultural heritage resources from across the continent.
- **Polycam**: A 3D scanning platform and app for creating, visualizing, and sharing 3D models of objects and spaces.

6.2 Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction

The work "Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction" [13] by Liao et al. explores innovative techniques in dynamic scene reconstruction. It presents a method using Deformable 3D Gaussian Splatting to capture and represent structural deformations with high fidelity. This research contributes to efficient modeling of non-rigid scene deformations, allowing for precise visualizations of temporal changes in cultural heritage sites, which is critical for monitoring gradual environmental impacts and structural degradation.

7 Tools

The following tools and technologies were used during the project:

1. **Blender 4.3.0**: For 3D model visualization and manipulation.
2. **Open3D 0.18**: For point cloud processing and model merging.
3. **Anaconda 24.9.2 (Conda 3)**: Environment management and dependency handling.
4. **Python 3.11**: Main programming language for data processing and tool integration.
5. **MeshLab 2023.12**: For point cloud comparison and analysis.
6. **CloudCompare 2.13.2**: For point cloud comparison and analysis.
7. **Postshot 0.5.10**: For extracting splat ply format from video.
8. **Node.js**: For running a local viewer for model visualization imported by GitHub repository.
9. **GitHub**: For version control and collaborative development. The main project repository can be accessed via the following link: 3D Gaussian Splatting Project. Additionally, dedicated repositories for each specific implementation are available at the following links:
   - ICP Merging with Python.
   - ICP Merging with CloudCompare.
   - ICP Merging with MeshLab.
10. **Plyfile**: Python library for handling PLY file formats.
11. **NumPy**: Python library for numerical computations and data manipulation.
12. **Overleaf**: Used for collaborative report writing and LaTeX compilation.

8 Description

The project "Temporal 3D Reconstruction of Cultural Heritage using Gaussian Splatting" focuses on creating high-quality 3D models and highlighting changes over time. The main phases of the project were:

8.1 Data Collection

At the beginning of the project, we were provided with a Gaussian splat PLY model of the Balzi Rossi cliff in Ventimiglia. However, this model alone was not suitable for our purposes, as a second model captured at a different point in time was not available. Additionally, it was not feasible to artificially generate noise or modifications (such as erosion) in an efficient and realistic manner to simulate temporal changes.

To find appropriate datasets, we searched for models online, particularly on the platform **Polycam**. The primary requirement was to obtain models in **SPLAT PLY** format or models that could be converted to this format (without loss of information), representing the same object at different time intervals.

As an example of cultural heritage, we selected the **Satiro e Baccante** sculpture by James Pradier (1834) located at the Louvre Museum in Paris. The source link for the model is provided here: Satiro e Baccante. The link for the viewer used for model visualization are provided here: WebGL 3D Gaussian Splat viewer, WebGL 3D Gaussian Splatting Renderer.

Figure 1: *Satiro e Baccante* using WebGL 3D Gaussian Splat viewer

8.2 Preprocessing

The preprocessing phase was critical to prepare the models for effective comparison and visualization of temporal changes. This phase involved two key steps: format conversion and simulation of erosion effects.

Step 1: Format Conversion. Since the models obtained from online platforms were not initially in **SPLAT PLY** format, we used **Postshot 0.5.10** to convert video captures into SPLAT PLY models. This step

ensured compatibility with the Gaussian Splatting techniques used in the project.

Step 2: Simulation of Erosion Effects. To simulate realistic temporal changes (such as natural degradation or erosion), we employed **Blender 4.3.0**. Using Blender's point cloud editing capabilities, we manually removed points from specific areas of the model to mimic erosion over time. This process allowed us to create a second version of the model with visible differences, which could then be used to highlight changes through different techniques. The result of this preprocessing phase was two models of the same sculpture in splat PLY format, representing different time intervals with observable variations.

8.3 Model Processing

To implement the model merging process between the two resulting models, we explored three different methods: **CloudCompare**, **MeshLab**, and a custom solution developed in **Python**. Each of these methods relies on the **Iterative Closest Point (ICP)** algorithm, a widely used technique for aligning two 3D models by iteratively minimizing the distance between corresponding points.

8.3.1 CloudCompare

To develop the project we used CloudCompare, a point cloud and 3D model management software, to align and join two PLY models of a statue: one complete and one simulating damages. After applying initial transformations to one of the models, two alignment methods were tested.

The first, ICP, automated the alignment by iteratively reducing the distances between points, generating a transformation matrix. The second, Point Pairs Picking, required the manual selection of at least four matching point pairs to calculate the alignment. Finally, the aligned models were merged into a single exportable model.

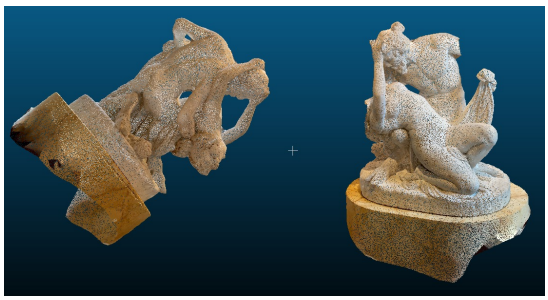

Figure 2: Initial configuration

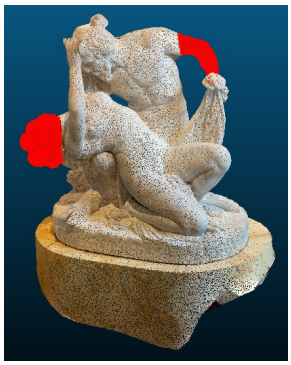

Figure 3: Merged models

During experiments we observed that CloudCompare in general is a powerful software for this kind of projects, but with some imprecisions, in fact multiple tries with the same set up were be done because not always the tool worked properly.

8.3.2 MeshLab

The other software we used for model merging was MeshLab, developed by the ISTI-CNR research center. Initially, MeshLab was created as a course project at the University of Pisa in late 2005. Compared to CloudCompare, MeshLab offers fewer tools and presents several limitations, which led us to discard it as a primary option for model merging. The main limitation is that the ICP algorithm in MeshLab can only be executed after performing a Point Pairs Picking process (Figure 4). This method is prone to high error rates, significantly reducing the accuracy and reliability of the ICP alignment, making it unsuitable for our needs.

The results of the ICP process using MeshLab are shown in Figure 5.

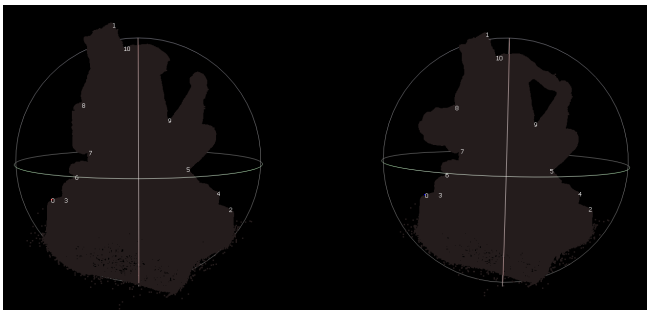

Figure 4: Point Pairs Picking process

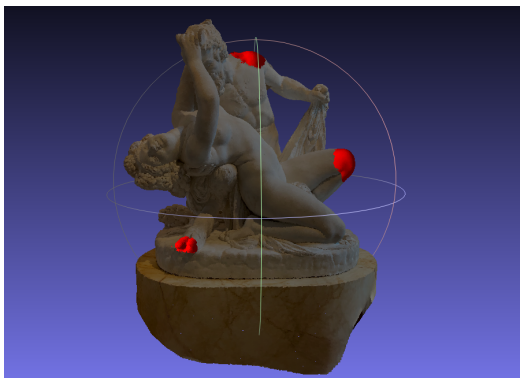

Figure 5: Models after performing ICP

8.3.3 Coding

The implementation of the project involved several key steps, each addressing specific aspects of the temporal 3D reconstruction process. The workflow was carried out in a VS Code environment, with the code organized into distinct modules for clarity and maintainability.

Data Loading and Feature Extraction. The first step involved loading the PLY files containing the 3D models. We read the file contents and extracted essential features such as coordinates and colors. These features were stored in appropriate data structures.

ICP Registration and Merging. To align the two models in a common coordinate system, we implemented the Iterative Closest Point algorithm. This algorithm computed the transformations necessary to align the models, then we applied them to the models using the function called `apply_transformation`.

Figure 6: Before performing ICP

Figure 7: After performing ICP

Difference Visualization. To highlight differences between the models, we implemented a function `compare_and_color`. This function utilized the KDTree structure from Open3D to efficiently find nearest neighbors for each point in the merged model. Points were colored red or green based on whether they represented erosion or new material, respectively. Points with identical coordinates in both models retained their original color, indicating no change.

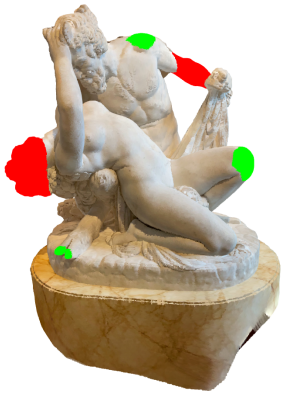

Figure 8: Differences of the two models

PLY File Reconstruction. Finally, we reconstructed the merged model into a new splat ply file using the processed data. This file was then visualized using Open3D, allowing us to inspect all the temporal changes interactively.

9 Results

This section presents the main results achieved throughout the project. The analysis focuses on the accuracy of the merged models and the quality of the visual representation of the differences between them. Furthermore, a comparison of the various implementations is provided, along with a discussion of the limitations encountered during the development process.

9.1 Model Merging Accuracy

The project's outcomes are promising, though several areas for improvement were identified. Regarding accuracy, while the results are satisfactory, there is potential for enhancement. Notably, some inaccuracies were observed, which could be mitigated through advanced coding techniques and a more rigorous grid search for parameter optimization. Unfortunately, such refinements are not feasible with the software tools employed, such as CloudCompare and MeshLab, which sometimes failed to align models correctly.

9.2 Visualization of Differences

The visualization of differences between models is commendable, though more sophisticated methodologies could potentially offer enhanced insights. Nonetheless, the current approaches are efficient and provide a clear understanding of the changes over time.

9.3 Software Comparisons

In comparing the software tools, MeshLab was found to be less performant and less customizable than CloudCompare. The latter offered more tools and flexibility, though the true potential for customization and improvement lies in the code-based approach. This method allows for greater personalization and offers more room for optimization.

9.4 Limitations

Despite these findings, limitations due to computational constraints were evident. The need for substantial computational power to process heavier models led to the use of lighter models. Nevertheless, these models effectively validated the proposed solutions, demonstrating their feasibility and effectiveness.

10 Conclusions

This project successfully demonstrated the feasibility of using Gaussian Splatting for temporal 3D reconstruction of cultural heritage sites, achieving the objective of highlighting differences between models. These capabilities are invaluable for applications such as restoration, specific interventions in historical sites, and beyond. We explored the state-of-the-art in Gaussian Splatting and proposed improvements, including automation of preprocessing and alignment, extension of visualization tools to support additional formats and larger datasets, and implementation of a web-based viewer. Additionally, future enhancements could involve advanced visualization features like dynamic rendering sliders, AI integration for improved algorithm precision, and simulation of eroded part repairs to predict restoration outcomes. Despite challenges such as computational constraints, the project underscores the potential of Gaussian Splatting in cultural heritage preservation and offers a solid foundation for further research and development.

References

[1] Florian Barthel, Arian Beckmann, Wieland Morgenstern, Anna Hilsmann, and Peter Eisert. Gaussian splatting decoder for 3d-aware generative adversarial networks, 2024.

[2] Paul Bourke. *PLY - Polygon File Format*, 2023. Specification and Documentation.

[3] NumPy Developers. *NumPy: Fundamental Package for Scientific Computing in Python*, 2023. Documentation and API Reference.

[4] Blender Foundation. *Blender: Open Source 3D Creation Suite*, 2023. Version 4.3.0.

[5] Daniel Girardeau-Montaut. *CloudCompare: 3D Point Cloud and Mesh Processing Software*, 2023. Version 2.13.2.

[6] ISTI-CNR. *MeshLab: Open Source Mesh Processing Tool*, 2023. Version 2023.12.

[7] Jawset. *Postshot: Video to 3D Model Conversion Tool*, 2023. Version 0.5.10.

[8] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023.

[9] Kishimisu and RR-28023. Gaussian splatting webgl renderer, 2023. GitHub Repository.

[10] Kevin Kwok, xvdp, and Haoyang Ye. Webgl 3d gaussian splat viewer, 2023. GitHub Repository.

[11] Polycam. *Polycam: 3D Scanning and Model Exploration*, 2023. Online Platform.

[12] Hugo van Kemenade. Plydata: A python library for ply file manipulation, 2023. Documentation and API Reference.

[13] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction, 2023.

[14] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing, 2023. Version 0.18.