3D Scene Reconstruction and Virtual Tour

Abdul Rafey
26100077@lums.edu.pk

Abdul Hadi
26100075@lums.edu.pk

Abstract

This project demonstrates 3D scene reconstruction from photographs and interactive 3D visualization. We captured multiple images of an indoor scene and attempted Structure from Motion (SfM) reconstruction using SIFT feature detection and matching. Initial results were unsatisfactory, so we used Agisoft MetaShape for photogrammetry-based reconstruction. To improve the point cloud quality, we captured additional images and reprocessed them with MetaShape, resulting in a significantly better reconstruction. The final high-quality point cloud was used to develop an interactive web-based virtual tour application using Three.js. The application allows users to navigate the 3D scene with mouse and keyboard controls. This project demonstrates the practical workflow of 3D reconstruction from image capture to interactive visualization.

Project Repository: https://github.com/Rafey16504/3D-Scene-Reconstruction-and-Virtual-Tour

1 Introduction

3D scene reconstruction from photographs is useful for creating virtual tours and digital models of physical spaces. In this project, we worked on reconstructing an indoor scene from images and creating an interactive web-based viewer for the resulting 3D model.

Our project workflow involved:

1. Capturing photographs of an indoor scene
2. Attempting 3D reconstruction using Structure from Motion techniques
3. Using professional photogrammetry software (Agisoft MetaShape) for better results
4. Capturing additional images to improve reconstruction quality
5. Building an interactive Three.js application for 3D visualization

1.1 Project Overview

The goal of this project was to go from raw photographs to an interactive 3D virtual tour application. We explored both traditional computer vision techniques and professional photogrammetry tools to achieve the best possible 3D reconstruction, then implemented a web-based viewer with intuitive controls for exploring the reconstructed environment.

2 Methodology

2.1 Initial Approach: Structure from Motion

We implemented a basic Structure from Motion pipeline to reconstruct 3D from two images. The process involved:

- **Feature Detection**: Using SIFT to detect keypoints in images
- **Feature Matching**: Matching keypoints between image pairs using FLANN-based matching with Lowe's ratio test
- **Essential Matrix**: Estimating the Essential Matrix using RANSAC to find the geometric relationship between camera views
- **Camera Pose**: Recovering rotation and translation matrices from the Essential Matrix decomposition
- **Triangulation**: Computing 3D point positions from matched features across views

The initial two-view reconstruction produced a sparse point cloud, but the quality was not satisfactory. The reconstruction had gaps and inconsistencies that made it unsuitable for visualization.

2.2 Improved Approach: Using Professional Photogrammetry

Instead of relying solely on our SfM implementation, we switched to Agisoft MetaShape, a professional photogrammetry software. MetaShape provides:

- Simultaneous processing of multiple images (not just pairs)

- Automatic feature detection and matching optimized for photogrammetry
- Bundle adjustment that jointly optimizes all camera poses and 3D points
- Filtering of outliers and automatic removal of duplicates
- Dense point cloud generation

Using MetaShape on our initial image set produced better results, but we found the point cloud quality could be further improved.

2.3 Final Approach: Capturing More Images

To achieve higher reconstruction quality, we captured additional photographs of the scene from more viewpoints. We then reprocessed all images (both original and new) using MetaShape. This larger and more diverse image set resulted in a significantly higher-quality point cloud with better coverage and fewer holes. The final point cloud was saved in PLY format with per-vertex color information.

3 Virtual Tour Application

3.1 Three.js Visualization

We developed a web-based 3D viewer using Three.js, a JavaScript library for 3D graphics. The application loads the PLY point cloud and displays it in a WebGL-based 3D view.

3.2 User Controls

The application provides mouse and keyboard controls for navigation:

- **Mouse controls**: Left-click and drag to rotate the view around the point cloud
- **Zoom**: Mouse scroll wheel to zoom in and out
- **Keyboard navigation**: Arrow keys and other keyboard shortcuts for movement

These controls allow users to freely explore the reconstructed 3D environment from different angles and distances.

3.3 Implementation

The application is built with:

- **Frontend**: Three.js for 3D rendering, HTML5 for UI
- **Backend**: Express.js server for serving the application
- **Data format**: PLY point cloud files with vertex colors

The application runs smoothly on our desktop system with responsive interaction and real-time rendering of the point cloud.

4 Results and Implementation

The relevant figures have been attached and can be seen in the next page.

4.1 Point Cloud Quality

The reconstruction quality improved significantly through our workflow:

1. **Initial SfM attempt**: Produced a sparse, low-quality point cloud with many gaps and inconsistencies
2. **MetaShape on original images**: Generated a much better point cloud, but still had some areas with poor coverage
3. **MetaShape on expanded image set**: The additional images provided better scene coverage and more redundancy for reconstruction. The final point cloud was significantly better in density and consistency

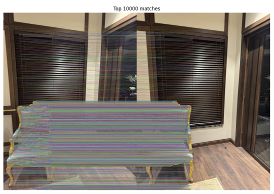
4.2 Virtual Tour Application

The Three.js application successfully displays the point cloud and allows interactive navigation. The application features:

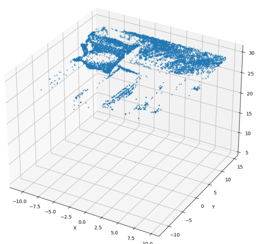
- Real-time 3D rendering of the point cloud
- Mouse and keyboard controls for camera movement
- Smooth interaction on our desktop system
- Responsive zoom and rotation controls

4.3 Visualization Quality

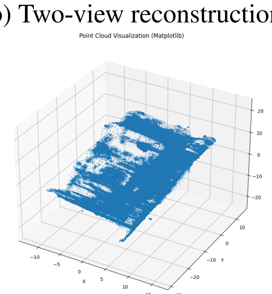
We also examined the reconstructed scene from different perspectives. However, we encountered an issue with mirror reflections in the 3D viewer:



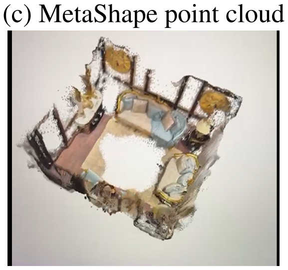
(a) Feature Mapping



(b) Two-view reconstruction



(c) MetaShape point cloud



(d) MetaShape 3D model

Figure 1: Reconstruction pipeline: (a) SIFT feature matches between consecutive images. (b) Sparse point cloud from two-view reconstruction. (c) Final point cloud after MetaShape processing on expanded image set. (d) 3D model generated by MetaShape.

Figure 2: Virtual tour visualization showing the reconstructed point cloud. Mirror surfaces are not rendered correctly in the 3D viewer; reflective surfaces appear distorted or fragmented. This is likely due to the point cloud representation capturing mirror reflections as separate geometry rather than coherent surfaces, a known challenge in photogrammetry of reflective materials.

5 Discussion

5.1 What We Learned

Our initial attempt at Structure from Motion gave us insight into the complexities of 3D reconstruction from images. While our implementation worked in principle, the results were not satisfactory. This led us to use professional photogrammetry software, which handled many details automatically.

We found that:

- Professional tools like Agisoft MetaShape are significantly better than basic two-view reconstruction for practical use
- More images and better coverage lead to higher quality results
- The quality of the 3D reconstruction heavily depends on image quality and camera coverage

5.2 Why Initial Results Were Poor

Our initial SfM pipeline produced a sparse, low-quality point cloud because:

- We only used two-view reconstruction instead of multi-view
- There was no global optimization or bundle adjustment
- The limited number of images meant poor coverage of the scene

These issues were resolved by using MetaShape and capturing more images.

5.3 Impact of More Images

Capturing additional images from diverse viewpoints significantly improved the reconstruction quality. More images provided:

- Better coverage of the entire scene
- More constraints for bundle adjustment
- Redundancy that helps filter out errors

6 Conclusion

In this project, we built a system for 3D scene reconstruction and interactive visualization. We captured photographs of an indoor scene, reconstructed a 3D point cloud using professional photogrammetry software, and created a web-based viewer for the result.

Our approach involved:

1. Implementing a basic Structure from Motion pipeline to understand 3D reconstruction
2. Recognizing the limitations of basic SfM and switching to professional tools (Agisoft MetaShape)
3. Capturing additional images to improve reconstruction quality
4. Building a Three.js application for interactive 3D visualization with mouse and keyboard controls

6.1 Key Takeaways

- Professional photogrammetry tools produce much better results than basic computer vision implementations
- More images and diverse viewpoints improve reconstruction quality
- Web-based 3D visualization is practical and accessible

6.2 Limitations

- The application was developed and tested only on our desktop system
- No formal user testing or evaluation was performed
- The quality of reconstruction depends heavily on image capture and camera coverage