3D Scanning of Large Spaces

February 12, 2021 - Winter Term

Prepared By

CS 461 - Group CS11

Casey Boomer - Seika Mahmud - Brandon Withington

Abstract

The current industry standard for photogrammetry is the Matterport camera. However, the matterport camera is very expensive, and it requires the company to save their scans directly to Matterport servers, which could be a security issue. This project will tackle this problem by making a software that has the power of the Matterport, while being significantly cheaper and more secure.

Table of Contents

1 Background

With COVID-19 becoming more of a daily life rather than a short quarantine, the need for 3D-scanned rooms has drastically increased. From real estate agents using 3D-scanning to make virtual walkthroughs, to schools using to create virtual classrooms and learning spaces, the technology is making it easier during this pandemic. The problem with this is the range of technologies out there for the common person who would want to use it. Either the hardware or software is extremely expensive to use, or it requires a high level of technical background to set up and develop on your personal computer. Both routes involve a plethora of photos that need to be taken and a lot of moving around to frame the images, especially in large spaces.

The current status quo for creating three-dimensional models of rooms in industry are using matterport cameras. These cameras are created to capture multiple images at a time, and stitch them into a 3d model. These cameras make it easier to create 3D-scans at a high resolution and good detail. The problem with matterport cameras however is that they are very expensive. Listed at around $2995, the average user would have to make an investment to be able to use it. Along with this, any scans created are put into the database that matterport runs. This causes questions of security for companies or people who are taking the images for their own use.

There are other options besides the matterport camera. There are many open source projects that are used as well. The benefits for these are that they do not cost any money other than the camera that is used, which will generally be cheaper than the matterport, and the scanned images can be accessed through any media that the user wants to use. The problem with using open source software is that it is not as user friendly as the matterport camera, and it requires more experience in computers to use.

2 Vision Statement

The vision for this project is to create a 3D-scanning software that is both affordable and user friendly. The project will be taken on any camera and should be able to be done on a notebook laptop. When the software is used, the 3D model that is created will be uploaded directly to the users computer rather than on an online database. Once the software is created we will make a marketing video with Intel that will demonstrate the software's ability to work with a notebook laptop and an Intel RealSense Camera.

3 Success Measures

3.1 Cost and Accessibility

One of the main goals of this project is to create a product that is affordable and accessible to many different types of users - realtors, construction workers, and professors to name a few. Throughout this project, we have researched and tested various tools and software and evaluated whether or not they align with our goal of developing a product that can produce high-quality models without becoming unaffordable to the average user. Our current solution, while no longer open source, features tools that are relatively inexpensive compared to their performance.

3.2 Tools and Methods Used

Our project partner has requested that we use primarily Intel hardware to develop our final product. We originally aimed to only use software that is fully open source. However, we have since switched to using Dot3D for point cloud scanning, which does require a license to use, as well as a specialized depth camera from Intel. We started exploring this solution at the request of our project partner and quickly found that it shows quite a bit of potential. Because of Intel's connections to DotProduct, we were also able to obtain free trial licenses through June. Neither the software nor the depth camera are prohibitively expensive (at least compared to the Matterport camera), so we decided that it would be acceptable to include Dot3D and the Intel RealSense camera in our final solution. We intend on utilizing public spaces, such as the forestry lab, to test our scanning software. At this point, it seems that we have more or less settled on our choice of scanning equipment; however, we have yet to find an effective method for converting the point clouds Dot3D produces into high-quality textured meshes.

3.2 Security

Our project partner specifically requested that the data of each scan must be stored locally to ensure user privacy and security. We are also aware that any cybersecurity attacks resulting in the theft or corruption of 3D models could have serious consequences. Throughout the project, we will make sure that our system is designed such that the user maintains total control over their data while also taking measures to minimize vulnerabilities. The software we are currently using does not send or store user scan data on external servers.

4 Prioritized Project Constraints

With our project, there are certain constraints we are currently working with. We are on a limited budget. This budget extends to the equipment we can request. It also extends the software we can utilize in our development process. Our project partner has expressed to us that he wants us to develop this application using as much Intel technology as we can. This includes multiple Intel Realsense depth cameras and laptops that have enough RAM to run Dot3D, an open source photogrammetry software we are considering, effectively. None of us currently possess an Intel RealSense camera or a matterport camera to develop this application yet. Obtaining physical equipment from both our technical advisor, Dr. Raffaele de Amicis, and our project partner, Mike Premi, poses a unique challenge to us as a group as in the wake of the COVID-19 epidemic. Under normal circumstances, we could consolidate our physical supplies and utilize them as a group together. However, we are limited in how we can work together. We will have to utilize primarily collaborative online software building tools such as GitHub to share and document code. Our last constraint is time. As students, we have other classes that are currently running which will further complicate how much time each of us can individually spend developing this application. This time constraint could potentially result in features not being completed to our own team's and our project partner's liking. In general, we will prioritize maintaining a low budget and minimal overhead so long as we are able to implement the minimum required features and our product remains a viable alternative to other products on the market.

5 Stakeholders

- **Mike Premi**
  Our project partner who gave us the overall vision and scope of this project. He works with the marketing, sales and strategic planning departments. He could potentially give our project an unprecedented amount of reach once it is completed. He has experience with managing various VR development projects similar to our current project.

- **Raffaele de Amicis**
  Our technical project advisor and coordinator, overseeing the development of our software. Raffaele de Amicis will provide us with equipment to utilize for this project.

- **Intel Customers**
  Our project partner, Mike Premi has informed us that many of Intel's consumer base would be interested in more efficient 3D environment scanning software. They may be using this software if it gets released.

- **3D Scanning Team - Seika Mahmud, Brandon Withington, Casey Boomer**
  Develops, hosts and maintains the features of the 3D scanning application and the photogrammetry pipeline between the software and the Unreal Engine.

- **Virtual Studio Team - Xindi Guo, Sowmya Jujjuri, Evan Medinger, Joelle Perez**
  Develops, hosts and maintains the features of the back-end of the application.

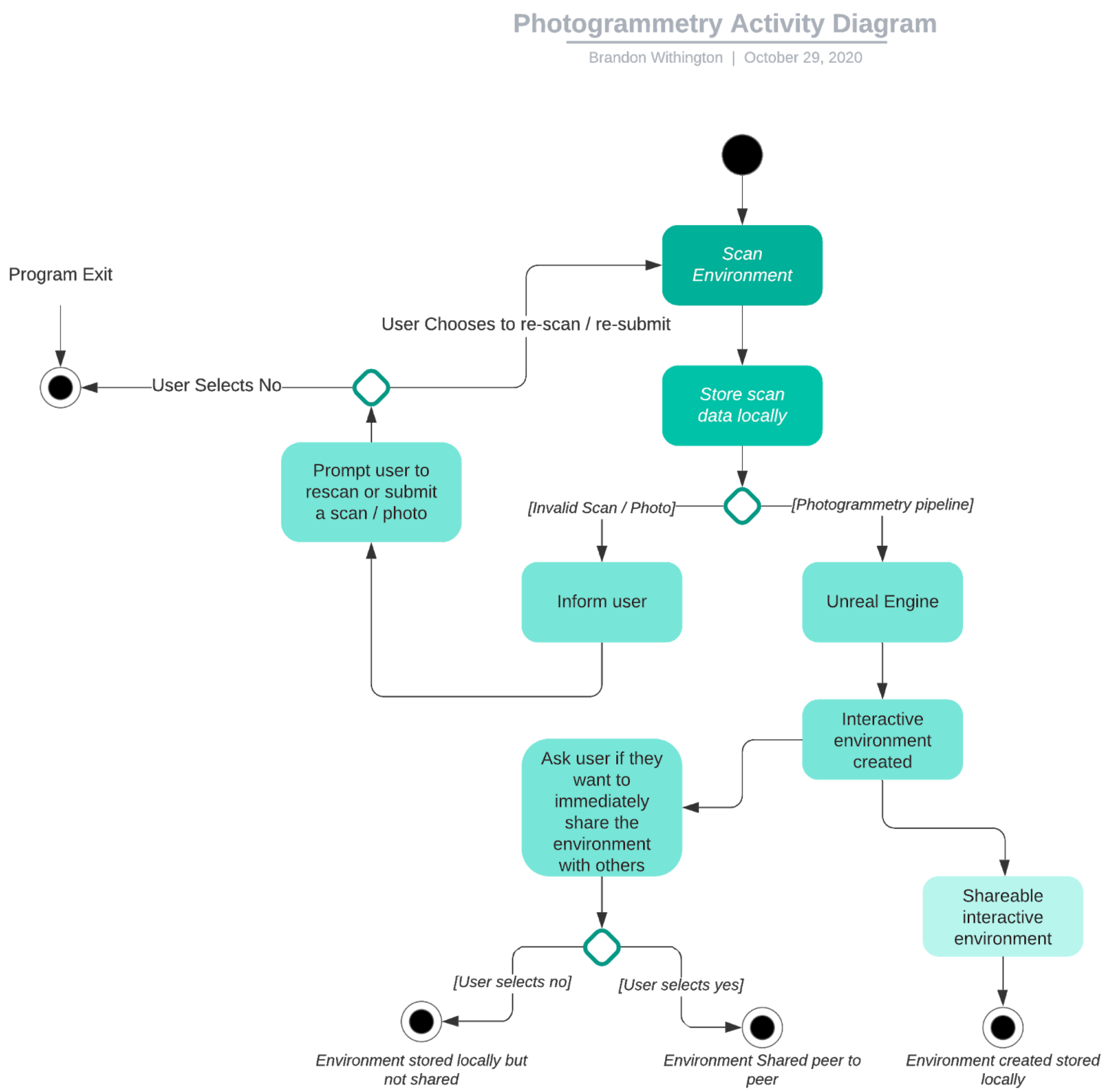
6 Risks

Risk	Likelihood	Impact	Mitigation Strategy	Early Detection	Consequence
It may not be possible to implement the minimum requirements by the deadline.	Unlikely	High	Each sprint, we will review our progress to make sure we are on track to complete at least the minimum required scope of the project by the deadline. If we are behind schedule but unable to scale the project down any further, we will consider bringing in additional resources to help complete the requirements more quickly.	The projected completion date for the project extends past the deadline.	If we are unable to complete the minimum requirements on time, we may not be able to deliver a complete final product.
The scale of the project might be too large.	Somewhat Likely	Medium	We will evaluate the scale of the project as we continue our research and planning, and we will communicate with our project partner about progress and any issues that arise.	Struggling to reach sprint goals.	If the scale of the project remains too large, the final product might lack functionality or fall short of our project partner's expectations.
The deliverables do not meet the expectations of our project partner.	Unlikely	High	We will be sure to communicate our struggles and to provide deliverables regularly and on time.	Project partner disapproves of deliverables.	Having our project partner scale the project down.

7 Scope

7.1 Process Flows

Activity Diagram:



UML Use Case Diagram:

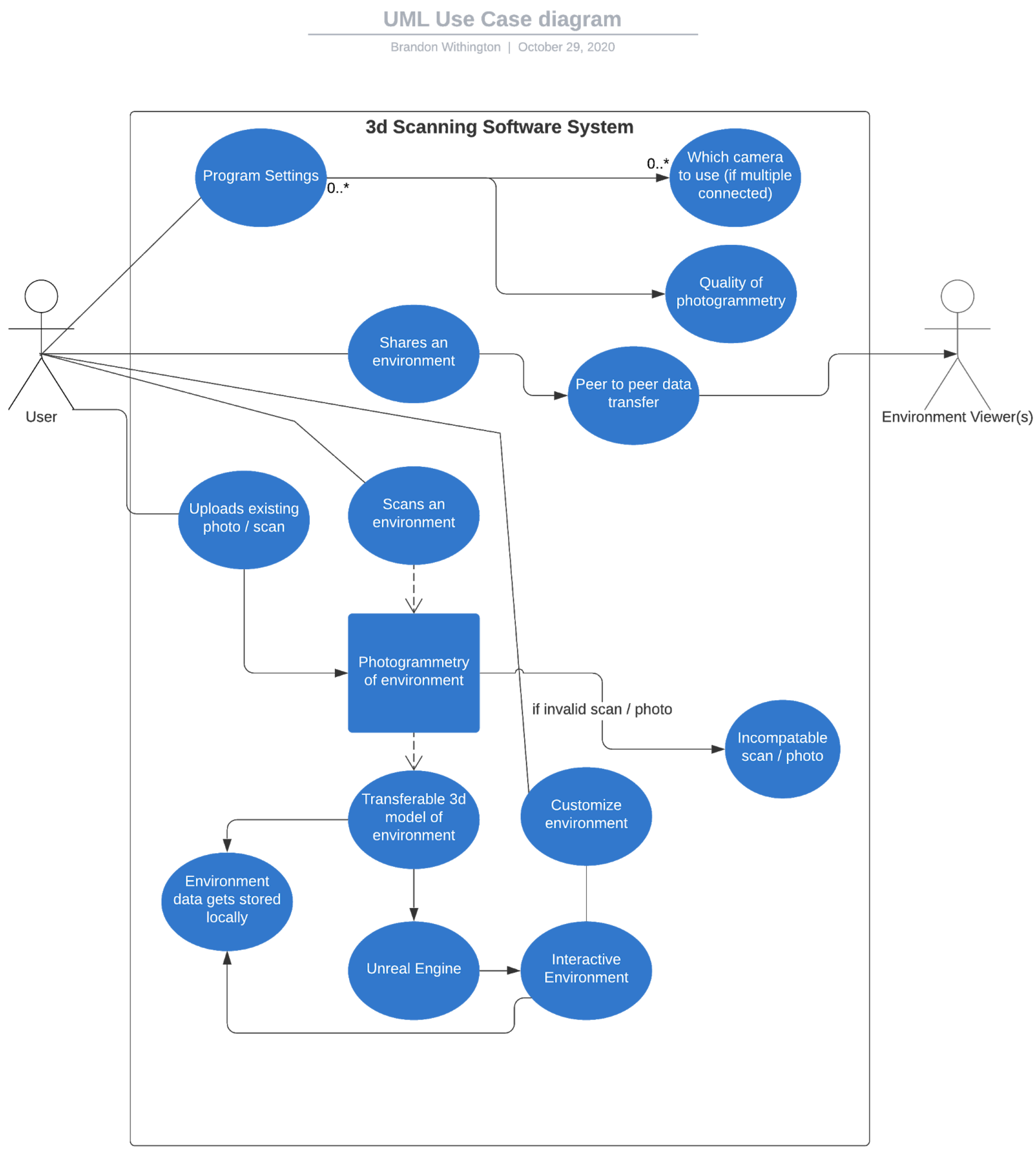


Diagram Summaries

Activity Diagram

This diagram covers general activities that we envision our application utilizing. It has basic user interactions accounted for, such as allowing the user to scan an environment, submit photos, or share an environment. It then either accepts the scan or tells the user that it is an invalid scan. In the case of an invalid scan, the program will then ask the user if they would like to rescan or submit another photo. Once a scan is accepted, we plan on using the Unreal Engine to turn that scanned environment into a shareable and interactable environment. Once that process is complete, the program will then ask the user if they want to share their newly created environment with other people. If they do not, the environment gets stored on the user's system locally. They will have an option to share that environment at their discretion. If the user selects yes to this prompt, they will immediately be able to share their environment with other individuals.

UML Use Case Diagram

This use-case diagram covers the system requirements of our envisioned system and how a user might interact with the system we intend to create.

7.2 User Stories

7.2.1 Epic

As a user, I want to be able to scan a large area.
This epic will cover the implementation of 3D scanning and related functionality.

7.1.1.1 Feature

As a user, I want to be able to upload and use existing photos such that I don't need to scan an environment at the time of creating a model.
This feature will allow users to choose whether to scan the environment or use preexisting photos of the desired space.

7.1.1.2 Feature

As a user, I want to be able to thoroughly scan a large space in a way that is quick and easy.
This feature will focus on simplifying the 3D scanning process while maintaining the necessary functionality.

7.1.1.3 Feature

As a user, I want a 3D scanning system such that I can use equipment that is already at my disposal.
This feature will focus on developing a scanning system that is compatible with common or easily accessible devices.

7.2.2 Epic

As a user, I want a system that can turn a scanned area into a 3D model.
This epic will focus on the process of generating a 3D model given photos of a large space.

7.2.2.1 Feature

As a user, I want to be able to create a model that is detailed and accurate.
This feature will focus on maximizing the quality of 3D models whilst remaining within the scope of the project.

7.2.2.2 Feature

As a user, I want to be able to create large-scale 3D models without too much overhead.
This feature will focus on optimizing the 3D modeling functionality for large areas.

7.2.3 Epic

As a user, I want a system that is closed and secure.
This epic will focus on security and making sure users have control over their own data.

7.2.3.1 Feature

As a user, I want to be able to securely save my scanned rooms locally on my computer such that they do not need to be stored anywhere else.
This feature will allow users to store their work without interacting with a remote server.

7.2.3.2 Feature

As a user, I want my files to be protected such that they cannot be corrupted or stolen. This feature will add security measures to protect users' data from outside attacks.

7.2.4 Epic

As a user, I want a system that is easy to use.
This epic will include features concerning the system's user interface and user experience.

7.2.4.1 Feature

As a user, I want a system with an interface that allows me to work quickly and efficiently.
This feature will focus on streamlining the 3D scanning and modeling process.

7.2.4.2 Feature

As a user, I want to use an interface that is easy and intuitive to navigate.
This feature will focus on creating a user-friendly interface that makes screens and features easy to find.

7.2.4.3 Feature

As a user, I want to have access to a wide variety of tools and features that serve my specific needs.
This feature will include user stories specifying the various tools that users would want or find useful.

Iteration Plan and Estimate

The first iteration of the project was spent planning and preparing. We researched different cameras, hardware, and software that we believed would best apply to our product vision. Based on our findings, we received laptops and an Intel RealSense camera from our project partner and recently were granted extended free licenses for Dot3D. We also researched user interface designs for current 3D-scanning software and understanding the user experiences of people that have used the Matterport camera before. The estimated amount of time for this iteration was for the duration of the first sprint to the beginning of the third sprint; we were able to complete most of our research and preparations during this time period.

The next iteration focuses on creating an application framework that will work with the 3D scanning software that was chosen. This will include getting the framework of the interface set up by getting a system that generally works. The goal of this iteration is to be able to at least navigate the interface, and successfully create 3D-scanned models using the provided hardware. This iteration will be a much longer time than the research section. The estimated amount of time for this iteration is from two to three months. So far, we have successfully produced point clouds and basic meshes using the available tools, and are currently working on creating high-quality textured meshes that can be displayed in Unreal Engine.

Once the framework of the application is complete, the next iteration will be designing the user interface to work with the application. This will apply the research that was done earlier about user interfaces. This section will be making the framework more usable to a general audience. This iteration will take from three weeks to about two months depending on issues that arise in development.

Solution Architecture

The architecture that best fits this project is the blackboard model. This architecture is when there is a knowledge source that evaluates updates and data that is added to the software and decides whether the update is applicable to the project. There would be one source for the application and individuals that would input their partial solutions for problems with the application. Because this project involves mostly open source projects, and the project itself will become open source, this architecture will benefit the software.

Other architecture that was taken into consideration is the model view controller method. This architecture has three separate entities: the model, the viewer and the controller. The viewer is the person that can see the model and control attributes of it with the controller. The controller takes the input from the viewer and applies it to the model, and the model takes the input from the controller, modifies itself, and displays the new results to the viewer. This architecture would work for this project because it has a lot of focus on the user experience and interface, which is a key component of the project's vision. However, this architecture is mostly centered around a continuous application that constantly changes itself for the viewer, and this project is more related to input from the viewer, then output from the software.

8 Alpha Functionality

Casey Boomer

Casey is an applied CS major that is focused on Simulain and game design. He has an interest in developing applications, and has experience in C, C++, Python, OpenGL and web development. He especially has an interest in Computer graphics, and GIS, and hopes to be working on mapping software in the future. Casey researched different cameras that would work well for the software, and created the architecture that is currently being used. During the winter term, he has been the team lead, which involves communication with the project partner, organizing meetings and creating agendas. He has worked on developing a user interface with QML, searching for methods to connect the User interface to Dot3D, and testing our scanning pipeline by working with PointCloudPlugin. He generated scans using Dot3D,

Seika Mahmud

Seika is an Applied CS major focusing on web and app development. Most of her experience to date has involved both front- and back-end development, particularly Python-based frameworks such as Flask and Django, as well as graphics and info visualization. Since starting this project, she has learned to use a variety of scanning and modeling software such as Meshlab, PointFuse, Blender, Unreal Engine, and Dot3D. During fall term, she took on the role of team leader, which involved coordinating meetings, planning agendas, and communicating with project partners. She also served as scrum master for the first sprint. Since then, she has focused on testing different scanning and 3D modeling methods with the goal of developing a robust and efficient pipeline that can be used to generate interactive, photorealistic visualizations. Currently, Seika is researching potential solutions for creating high-quality textured meshes that can be displayed in Unreal Engine.

Brandon Withington

Brandon is an Applied CS major with a focus on Graphics, Simulation, and Game Programming. He is experienced with 3d graphical engines such as Unity, the Unreal Engine, and AutoDesk's 3ds Max and Maya. He is also familiar with graphical and video-game industry standard languages such as C#, C++ and Java. In addition to that, he has a background in web-development with PHP, SQL and node.js. Since beginning this project, Brandon has learned how to use LiDAR camera equipment to scan interior areas and utilize point-cloud editing software to create 3d objects from scans. He has been able to import a high fidelity scan of Oregon State's College of Forestry into the Unreal Engine utilizing both PointFuse and a LiDAR point-cloud plugin. He has served as the scrum master for the second sprint. He has currently been working on improving the quality of meshes and textures generated from Meshlab.