Due: Monday, October 26, 2020, 8:00 AM (submit to BlackBoard, under Assignments)

File Type: PDF

Note: Blue sections should be updated versions from the Initial Project Description.

Note: Black sections are new sections for the Project Proposal.

Team No. 22

Team Members: Brandon Wheat, Zack Khazraeinazmpour, Ian Hierl, William Starks, Chase Odgers

Project Name (1-3 words): PyScan

Project Synopsis (1-25 words)

PyScan will implement photogrammetry techniques to generate 3D models of real-life objects from photographs, and ML integration will allow for objection recognition and model generation from labels.

Project Description (150-250 words)

- Why is the project being undertaken?

The project is being undertaken to learn more about machine learning properties like object recognition and the possibilities of AI-based generation of 3D models. Other benefits of the project include learning about the applications of computer vision with Python and how it could possibly be used in other industries.

- Describe an opportunity or problem that the project is to address.

PyScan will function as a utility for creating digital representations of physical objects, which has applications in many machine vision problems, such as path finding and environmental navigation. Incorporating object recognition into our model will increase the potential for those applications, as differentiating between object kinds is necessary in predicting safe and collision-free paths. Moreover, the project provides a method for digitizing physical objects, which could be highly beneficial for preserving digital records of historical artifacts. Such an application would provide benefits for the preservation of artifacts and for collaborative study of artifacts through virtual environments.

- What will be the end result of the project?

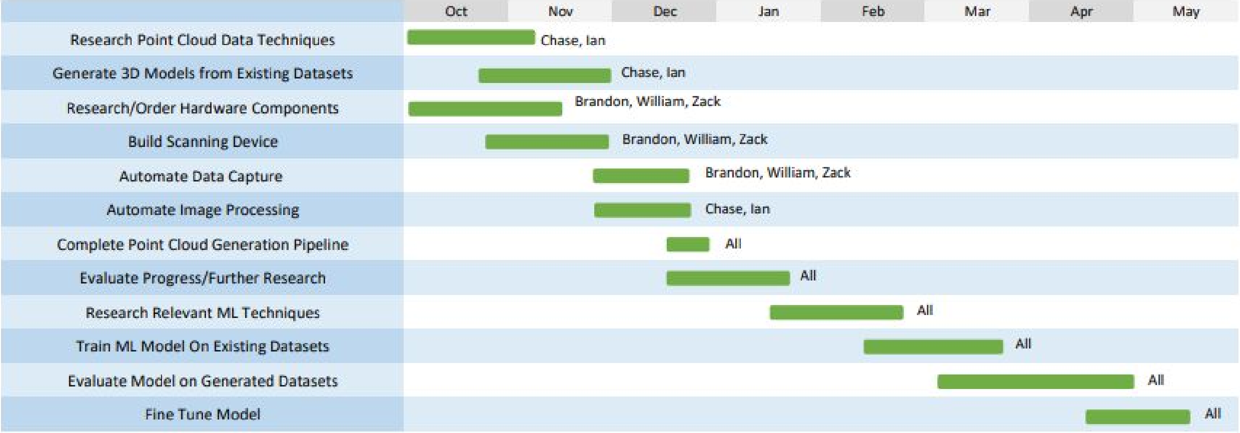
The end result for this project would be to have a program that rotates an object and takes a series of images of this object and then converts it to a 3D model. Moreover, if we have enough data on a particular object, we would like to use machine learning to predict what the object is based on its point cloud representation.

Project Milestones

- 3-5 specific and measurable objectives per semester for first & second semester.

First Semester	Second Semester
Practice generating 3D models from existing point cloud datasets - **10/16/2020**	Research best machine learning models and methods for our goals - **First 2 weeks of February**
Scanning/Image capture device completed and image taking process automated - **11/6/2020**	Apply machine learning model to existing point cloud data **- Last 2 weeks of February**
Automated image processing completed - **11/13/2020**	Apply machine learning model to our data - **Third month of second semester**
Point Cloud generation from images completed - **11/27/2020**	Fine tune ML model and continue to make adjustments as necessary - **Fourth month of second semester**

- Gantt Chart including who will do what



Project Budget

- Hardware, software, and/or computing resources
  - Cloud AI computing service - $50-100 over the full year
    - AWS, Google Cloud?

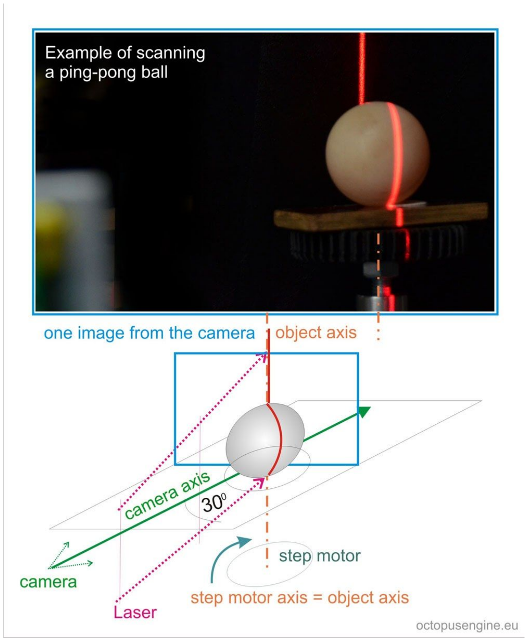
- Cameras, sensor components and device body - $100-150
    - Amazon for camera and other miscellaneous components
- Arduino Microcontroller - $25
- Stepper motor and motor controller for controlling platform

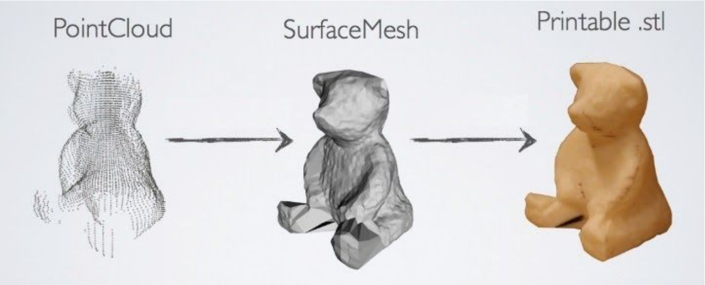
Preliminary Project Design

The preliminary design of the software is based around the generation of 3D point clouds from a series of 2D images. A point cloud is a dataset made up of X, Y, and Z coordinates and optional RGB values. Each point has its own X, Y, and Z value and together represent a 3D object.

Using a camera and rotating turntable, the software will automatically capture images at certain increments to form a comprehensive set of images representing the object. This set of images will then be used to generate a point cloud that can be manipulated using various open-source libraries and programs such as Open3D and Meshroom. The algorithm used to perform this computation is typically referred to as a *structure from motion* algorithm.

These datasets can then be passed to a machine learning model that will be trained to recognize objects based on the point cloud dataset, and with enough data, eventually be able to generate its own point cloud dataset that can be converted into a 3D object entirely from the output of the machine learning model. (see below for diagrams of this process)





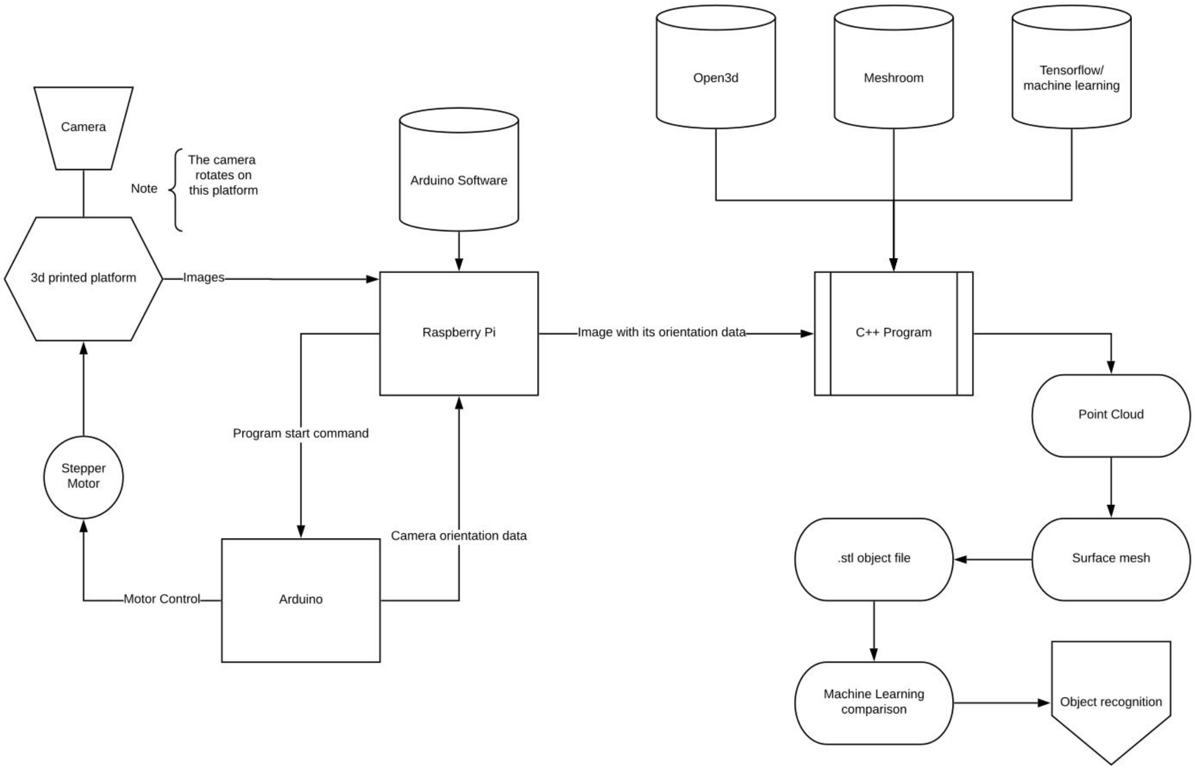
Technical constraints:

For this project we will be using two different programming languages, Python and C++. We are constrained to using C++ for a variety of reasons. To start with, the majority of our team is very familiar with the C++ programming language. On top of this, the hardware that we will be using, specifically the arduino microcontroller, operates using C++. We will also be using C++ for rendering our 3D models because it has some great open source libraries that are available for it. C++ has very fast execution time for image processing which is just another perk for using C++ to handle the processing and rendering part of our project.

In this project we will also be using Python. Python is a great language that is commonly used in a lot of real world machine learning applications. It has so many great machine learning libraries made available to it. So we will be using python primarily to build a machine learning model to predict what our 3D model is. Another benefit is that most of the team has at least a little experience with the python programming language.

Operating system was an easy choice for our group. We have all spent our college careers developing skills in c++ development on linux distributions, naturally we will use this to do our development on the pyScan project. Because our project will be python based it should be able to be portable to any operating system capable of handling python. One issue that may present when running pyScan on a non-linux system may be the communication between the program and the computer hardware. If the camera/rotor assembly we are making brings about driver compatibility issues on various systems we will have to prioritize linux potentially ruling out the possibility of windows compatibility. The arduino may offer a solution to this dilemma as it has preexisting libraries that could offer multiple OS support by handling the camera data through the arduino first and then to the OS.

Arduino and Raspberry pi will be utilized in this project. We will use the arduino to handle the rotation of the camera. The camera rotation should be relatively simple to implement, using a stepper motor, power source, and a 3d printed platform we will be able to accurately increment the stepper motor giving us precise understanding of camera orientation. In order to handle the data we will have to use a computer, if we want pyScan to be portable this means the computation involved with pyScan must be portable. Raspberry pi offers a unique solution to this problem by enabling lightweight portable computation. By running the raspberry pi in cooperation with the arduino the incoming camera data can be combined with the incoming arduino data on the raspberry pi with our python script. After cleaning our inputs we can pass the relevant data obtained to our c++ program that will regenerate the point cloud on the computer and run the machine learning and video output handling functions to better represent and determine the objects scanned in.



Above is the process flow of our project. Summarizing this diagram, you can see the Raspberry Pi initiates the program. Following the “start” signal the stepper motor begins rotating incrementally, following each increment it passes its orientation back to the raspberry pi. The raspberry pi takes the data from the arduino and then captures an image on the camera. The raspberry pi collates the image from the camera with the orientation data from the arduino and stores that in a file. The number of scans necessary to generate a high resolution point cloud will be determined during implementation. Following the completion of the scan using the camera the file that has been generated contains all of the scan data. This scan data file is passed to the C++ program to be operated on. The first operation of the c++ program is to generate the point cloud. The program utilizes necessary functions from open3d, meshroom, and artificial intelligence/ machine learning libraries (tensorflow for example) to generate the point cloud. Following the generation of the cloud the same libraries are implemented to generate a surface mesh of polygons on top of the point cloud. This polygonal mesh is used to generate a 3d object file. The object file can then be compared against known data sets using machine learning to generate a guess at the object. Our goal will be to refine this generated guess to obtain as accurate guesses as possible.

While our process will undoubtedly require refining and fine tuning the above description depicts our end goal and a rough outline of how we intend to get there. The biggest challenges will come from optimization, finding the correct libraries, and compatibility between our chosen resources. While there are potentially more streamlined methods to completing this task we have chosen to create our own hardware in order to fully develop our understanding of the process, opting out of a plug and play 360 camera in order to learn the intricacies of working with images, hardware, and process scheduling.

Ethical Issues -

Inherent to any project using digital cameras is an ethical responsibility to ensure best practices are followed. The most prominent concern when using cameras is personal privacy. With our project we may come to points in development that require us to leave our camera running for a longer than usual duration, which may involve us not supervising the camera for a period of time. Because of this we must ensure that individuals in the area of the camera are aware they are being recorded, and respect their wishes to remove them from our data should they request it. This allows us to stay in compliance of ACM rule 1.2 “Avoid harm” specifically the disclosure of information subpoint.

Another ethical issue we face is our use of open source libraries. Using open source libraries means that we will be co-opting another individual's work into our project at certain points in time. It is our ethical duty to ensure the individual who wrote the code is credited. Open source libraries also blur the line of “who did what work” and with that in mind it is our responsibility to ensure we are extra transparent on where we have used the libraries, what the libraries contribute to our project, and why we chose to use the library rather than generate our own code. Following these two best practices puts us in compliance of ACM 1.2 “Avoid Harm”,1.3 “Be honest and trustworthy”, and 2.5 “Give comprehensive and thorough evaluations of systems and their impacts, including analysis of possible risks. This ethical issue also arises as an intellectual property issue, as demonstrated below.

Intellectual Property Issues -

As with any machine learning project, we will require access to large datasets to train our models on. Much of this data will have to be pre-labelled as training data, and it won’t be feasible to capture and label the data ourselves, so we will have to rely on pre-existing datasets. This presents an issue with regards to our rights to use this data. Much of the data we have already sourced comes from universities and other academic sources, and the use of that data is restricted to academic or other non-commercial use. Thus, if we were to attempt to commercialize our product later, we would be obligated to recreate our models with data we either generate ourselves or that we have obtained with permission for commercial use.

Additionally, we acknowledge that the project name "PyScan" is derived from that of the Python programming language. While using the part of the name of an existing property could require explicit permissions, the Python Software Foundation states in its trademark usage policy that "stating accurately that software is written in the Python programming language, that it is compatible with the Python programming language, or that it contains the Python programming language is always allowed. In those cases, you may use the word "Python" or the unaltered logos to indicate this, without our prior approval". Although this approval may not be needed, it would never hurt to seek it should this project ever see commercial use.

Change Log -

- Project description now includes a specific application for the project outside of the general umbrella of machine learning and machine version, namely artifact digitization.
- Scanning/Image capture device completed and image taking process automated project milestone was pushed back from its original date of 10/23/2020. The new date to complete this milestone is 11/6/2020. This was changed because we did not receive the hardware necessary to complete this task in time.
- Automated image processing completed and Point Cloud generation from images completed milestones were pushed back a week due to previous milestone not being completed on schedule. Second semester milestones remain unchanged.