**Planning**

**Create How-To Guide for Robotic Arm**

Due to there not being a how-to guide currently for the robotic arm, I could create a how-to guide for it since it whenever I was trying to learn how to use it, it took me about a week to learn since the codebase isn’t well documented or commented and on the GitHub there is very minimal explanation of the setup or how the robotic arm works. In my how-to guide I could include areas such as what software is needed, the commands, how to use and setup the HTTP API, how to create demos and how to use the different kinematics functions such as inverse and forward kinematics.

**Create GoPro Attachment for End Effector of Robotic Arm**

I could 3D print a part that would allow me to connect the GoPro to the end effector of the robotic arm so it can be rotated 360 degrees and also be moved to any possible position within the limits of the arm. This could very easily be done since I already have a plastic case that the GoPro came in (which I’m assuming is to waterproof it) but I could modify this to fit on the end effector of the arm.

**Create System to Remotely Control GoPro / PiCamera**

To perform photogrammetry with the robotic arm, I’ll need a camera attached to the end effector of the robotic arm. To control the camera, I’ll need to create or find an API such as goprocam or goprowifi-hack so I can take images / videos etc. via WiFi. If I decide to use an API such as goprocam, I’ll have to implement this into the current robotic arm codebase so I can control it directly from the demos. I also need to find a way to create a web server to connect the GoPro from any device on a network. However, to remotely control the GoPro it’ll require some programming knowledge using this method.

**Create Web Interface for Camera**

As an additional deliverable even though it won’t be directly used with the robotic arm, I can create a web interface which could be accessed via any device on the same network as the same web server as explained above. This web interface should allow any user on the network to make the GoPro take photos, videos, turn on, turn off and change the resolution so it would be extremely beneficial for testing of the GoPro and controlling it from far distances. For example, if I was able to set it up with the Makerspace Wi-Fi, that could mean it could be accessed in many places throughout the Computer Science Building.

Using this, I could also potentially use the API used for the remote control of the GoPro above, to try to livestream the current GoPro’s feed onto the web interface, however the latency and resolution of this solution might not be good. This as well as the previous deliverable will be very useful to future QLabs, computer science challenges and any students throughout the world since I haven’t seen a proper, easy-to-use, well documented web interface for controlling GoPro’s before, and as I continually improve on this, I can include more and more features for newer GoPro’s as well as any other cameras for different brands such as Canon or Pentax.

**Create Programmable Turntable**

To perform photogrammetry, I could either make the arm move in a circular motion, or the arm be still and create a turntable to spin around at a programmable speed which I could also make a web interface for to adjust the speed and turn it on/off. I could create this turntable using an Arduino and stepper motors along with a couple of 3D printed parts. However, it may be a more efficient use of my time to just buy a turntable, but I like the idea of being able to adjust the speed etc either via programming or via a web interface so that whenever I’m performing photogrammetry, I’d be able to adjust the speed depending on how many different pictures of the object I need (depending on the detail of the object) and how large the object is. If I decided to create one of my own, I could easily make it in approximately a week if I had no issues along the way since there are many guides online with the 3D printing files of a turntable, along with an assembly guide and programming for it, meaning all I’d have to do is solder some components and 3D print some parts.

Another problem with doing this however is that I would have to make quite a large turntable to allow for a large variation of sizes of objects for photogrammetry. This could be very useful for a variety of different projects for the future since it would be able to be adapted to any project in most programming languages with Arduino support, whether it be python or C++.

**Programmable Lighting System**

Due to the fact that for photogrammetry, very good lighting is needed, I could very easily create a system to control the brightness etc. of the lights in a lightbox needed for objects on the turntable. This could again, be very easily done with a basic Arduino and some dimmer LEDs or light strips which can easily be purchased meaning this part of the project would be both cheap and quick to accomplish. I could also make a web interface for this if needed, or combine all of the different components (lighting, turntable and GoPro) into one web interface with different pages on a website meaning it would reduce to cost of since I would only need to run one web server to run all 3 of these systems. Since the lights would be connected to an Arduino, I’d be also able to easily control them using the IO pins on the Arduino or raspberry pi. This could be very useful for a variety of different projects as well that require large amounts of lighting that can be programmed based on scope of the project.

**Performing Photogrammetry on a Small Object (Testing)**

I could attempt photogrammetry by attaching the GoPro to the end effector of the robotic arm and creating a path for it to go in a sphere-like shape around any object and take a large number of photos before inserting the photos into the photogrammetry software and see what the result is since I haven’t attempted photogrammetry of any sort yet with Agisoft Metashape Pro. This would allow me to get a rough idea on how much I’m going to need to focus on the post-processing stage of the project to try and make the model look as good as possible. Having developed a system to be able to complete photogrammetry on smaller objects, a future student, or myself, could work on it in the future in order to perform photogrammetry on larger objects as well with more detail.

**API to Check Brightness of Object**

I could develop software which uses the GoPro to take pictures of each side of the object and ensure that the lighting is adequate eg. No glare, no shadows etc. to allow the photogrammetry process to achieve the best product possible to reduce the need for long amounts of time spent on post-processing of the mesh. This could be accomplished using existing APIs that I would just need to implement into the GoPro system, however it could be a challenge for the GoPro to take the best quality photos to ensure an accurate reading of the brightness of the object.

**Performing Photogrammetry**

One of the end goals of this project is to be able to input the dimensions of an object, and be able to calculate a sphere of points to move the robotic arm to as well as calculate how many pictures will be needed to be taken based on the size of the object, input this into the GoPro interface which will adjust the speed of the turntable and brightness of the lights to be able to ensure the object isn’t too bright or too dark. Then, the arm should move in a circular motion around the object, taking pictures and inputting them into the photogrammetry software to produce a mesh model of the object.

This model that it produces will probably be relatively unsatisfactory since no post-processing will have been applied to it at this stage to fix things like holes.

**Post-Processing of the Mesh Model**

After the photogrammetry has been performed, there will most likely be a large number of issues with the model such as holes in it and materials not being accurate etc. To fix this, I can spend a good period of time focusing on the post-processing of the model to make it look as good as possible. Creating a post-processing algorithm could be extremely useful for future users since they could easily improve upon it and use it for their own photogrammetry.

**3D printing mesh model**

I could attempt to create software which takes a mesh model as an input and converts it into a 3D printable file to 3D print a small version, or actual version of any model created. This could be very useful if there isn’t open source, free software out there like this right now to allow consumers to easily 3D print a model.

**Turn iPhones into remote cameras**

I’m going to create iPhone apps so that a user could turn their iPhone into a remote camera such as a GoPro since they have great cameras with great processing power which means I have unlimited possibilities on what I could expand this project into.

**Create web interface for iPhone camera app**

I could create a web interface, like the interface I’ve made with the GoPro and PiCamera to allow the user to view the constant feed of their iPhone cameras from anywhere by going onto a web server, however if this product went commercial, some companies may have issues regarding cyber-security meaning I may just have to store the footage locally.

**Perform labelling using ARKit**

I could label different objects that the iPhone sees through its camera using ARKit and other AI APIs to allow me to label people or objects going through a system in combination with Mechanical Turk to get humans to label them initially.

**Perform Data Analysis**