**Planning**

**The Challenge**

The objective of my project is to use cameras such as GoPros which will be attached to the end effector of an already built robotic arm to perform photogrammetry in order to create digital representations of real-life objects

**Green = Making it / Has Lasting Value**

**Red = Not making it / No Lasting Value**

**Week 3**

This week I plan to create the how-to guide for the robotic arm as well and create the GoPro attachment for the end effector of the robotic arm. I feel these two things will be very important to others because it means that there will be documentation for the robotic arm and the end effector can be used by anyone wanting to attach a GoPro to the robotic arm.

By the end of the project, I’m hoping to have a working photogrammetry system where I can pick an object and use the robotic arm to automatically generate a path to go around the object, take pictures, and perform photogrammetry, creating a mesh model of the object.

**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. There is no current how-to guide for how to use the robotic arm, so I’m definitely going to make a how-to guide for it to help other students in the future working on the robotic arm learn how to use it quickly.

**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. This will be useful to other students in the future using the robotic arm who want to attach a GoPro to the end effector of the arm, so I’m going to make it. I could also publish the 3D printable file so that anyone can make it and modify it since it works with the GoPro waterproof case meaning it could be useful to more than just students at QUB.

**Week 4**

This week I plan to create an API to remotely control the GoPro through Python commands. I feel like this will be very useful for others because it means anyone can download this and control their GoPro / PiCamera from python and implement it into their software since it’s all open source.

My plans for the end of my project haven’t really changed this week, however I need to take time to experiment with other cameras to see what the best solution would be.

**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. I’m going to make this system because it means that anyone can implement the system with relative ease into their project since every function such as takePhotos are in methods so they can be called from other classes, meaning it could be useful to anyone and especially students wanting to implement the GoPro into their projects.

**Week 5/ 6**

This week I plan to create the web server / web interface for the remote control of the GoPro and PiCamera to allow users to control them from anywhere. I see this is extremely useful to others and has a lot of lasting value since it’s easy to set up with some technical knowledge and when set up, it’s very easy to use due to the user-friendly UI.

My end objective has slightly changed now, instead of focusing on the photogrammetry part of my project, I’m planning to focus more on the cameras part of the project as I feel like this will have much more lasting value and will be more useful to other students in the future. I’m going to polish the web interface as much as possible and try to make it as user friendly and with as many features as I can.

**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. Due to these reasons that it will be extremely useful to anyone; from photography professionals, to students, to anyone with some technical knowledge to be able to set up the hardware as explained in the how-to guide, I am going to make this.

**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++. I feel like this isn’t really needed and I could spend my time making other things that would be more useful to others since there are already many guides on how to make turntables with an Arduino and also many pre-made semi-programmable turntables out there, so I don’t see the need to make this.

**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. I feel like I could spent my time making more valuable things since there are already many guides on how to make these with Arduinos if you wish to do that and also many you can buy online both commercially and professionally which would provide much more light and functionality than any one I would be able to make, so I’m not going to make this as part of my project.

**Photogrammetry:**

For the photogrammetry part of this project in general for the 5 below ideas, I’m going to focus more on the remote iPhone camera idea below this as I feel that it has a greater business potential and have much more of a lasting impact on people. With the iPhone data analysis / camera project below, it hasn’t really been done before and I’ve found a niche in the market meaning it could be useful to big companies running factories as I’ve explained in its paragraph. So, I’m not going to do any of the photogrammetry projects listed below.

**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.

**Week 7 / 8**

In these weeks, I’m planning to start to work on my new goal for the end of this project, to create an affordable, more powerful and less time consuming way of performing data analysis in factories by using iPhone cameras as an alternative to IP Cameras. I feel like this has a lot of lasting value and business potential. By making systems to integrate Apple’s ARKit and potentially TensorFlow, this may be useful for other students in QLabs in the future.

**Turn iPhones into remote cameras**

I’m going to create an iPhone app 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. Some of the ideas include being able to use ARKit to label objects, using machine learning and AI to measure productivity levels in somewhere like a factory and analyse data which is captured by the cameras to help improve production levels etc.

When googling this idea, I found that there aren’t many products like it other than ones for football, but no ones for places like factories to help analyse data to improve profits etc.

Although I probably won’t be able to complete this app by the end of this project, if I make a good start on it, another student may be able to work on it in the future, or I could work on it over the summer as I feel it has amazing potential. Also, if it’s open-source and I write a how-to guide for how I did it, this would also be very useful for future students.

Due to these reasons, I’m going to make this part of the project.

**Week 9**

This week, I’m going to create the actual web interface

**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. However, I feel like this would also appeal to a lot of companies since it isn’t something all other IP Cameras have and due to the processing and data analysis that I’ll be able to carry out with the footage automatically, this will make it much more appealing.

Having a web interface that is able to access the constant feed of the iPhone camera would be very useful to some other students for other projects as an alternative to something like a Kinect since you would be able to do a lot of processing on the iPhone itself, eliminating the need for it to be connected to something like a PC or Laptop. Due to the fact that I feel this project could be very useful to both students for their projects as well as companies, I’m going to make this part of my project.

**Gather Data using Mechanical Turk**

To get the data needed for the processing of items going through something like a factory or to go gather data about productivity levels, I could use Mechanical Turk to gather a large data set of productivity levels in videos as well as labelling of things like items to feed into a machine learning algorithm so it can detect items going through a factory or for any other uses. This would produce a large data set which could be very useful for other projects in the future of other students as well as other developers around the world that are looking for a data set like this. However, in this project, I’m not going to have time to complete this so I’m not going to do this.

**Perform Data Analysis**

Using the data that I would gain from Mechanical Turk; I could create a system that would analyse data going the system. For example, if the iPhones are installed in a factory, I could measure the productivity levels of people based on how many actions they’re performing every minute, for example and feed this to the management in the factory where they’re installed to help them improve production etc. This system could also be a base for many other different systems for different industries and uses which other students and people could work on since I would make the project open source. This could also be installed in labs in the CSB to measure productivity levels of yourself or others to help improve you’re the amount of work you’re actually doing. However, in this project, I’m not going to have time to complete this so I’m not going to do this.

**Make Graphing Software for the Data**

Using the data from the data analysis stage, I could create a system to create different types of graphs based on the data produced to show the productivity levels of people in a factory, as an example. However, there is already a number of existing APIs which I can use to pursue this goal, so there is no need for me to make my own API for it as it would probably not work as well as if I used a pre-existing one that had been thoroughly bug tested and have much more functionality. So due to these reasons, I’m not going to make this part of my project.