Senior Design Project Report

Robotic Makeup Application System

Alejandro Varela

Jimmy Jason Ramsarran

Dillon Naidu

Jason Calle

Junmin Xu

Chun Kit Wong

The City College of New York

Department of Mechanical Engineering

Richard LaGrotta ME-47300

May 14th, 2019

Robotic Makeup Application System

Table of Contents

1.0 Executive Summary

The multibillion dollar cosmetic industry at an estimate of around $170 billion is an ever increasing and continually relevant field, not only today but also in the foreseeable future. For our senior project we were tasked with taking part in this industry by inducing a combination of innovative technologies to develop an apparatus that would automate cosmetic application.

Specifically, the given problem is to design a system that will take a picture of an individual's face and then allow the individual to modify the picture to their liking, with respects to makeup application. The user should ideally be able to perform his or her selection through the utilization of custom-built software. After the photo is taken and modified, the modifications should be performed on the user through a robotic apparatus.

To address the requirements of the given problem, we decided to utilize a uArm Swift Pro robotic arm after multiple trade studies, in conjunction with an Intel® RealSense camera. With a built-in Arduino Mega 2560 microcontroller in combination with pressure and proximity sensors, the majority of the workload is based on the software and coding of these units for components integration, facial detection and capture, processing, and application. The final product should be easily mobile and be able to be readily mounted on a typical table of at least 2.5x2.5 ft. The user will be able to sit in a comfortable position within operational distance of the robotic arm's base. After the user's photo is taken and the modifications chosen from pre-set options, the working medium is manually inserted into the arm holder and the arm performs the desired path with no further human interaction required. The apparatus will be able to detect relatively slight movements of the user's face and compensate accordingly in a safe and efficient manner. A motion sensor can detect if the user is ready by using a minimum

distance measurement. Collaborative features enable the user to have more control and feel more secure while the robot carries out its commanded path.

2.0 Problem Description

Understanding the nature of the given problem is paramount as it should be with any engineering problem, though in this case it is readily apparent as a result of operating directly on a human's face. The problem given does not dictate strict guidelines, procedures or requirements itself. We are required simply to design a robotic system that uses a user's photo which is then analyzed, after which user-chosen makeup modifications are applied. Additionally, we are required to ensure that our robotic system incorporates real time feedback. With this basic set of requirements and the overall problem outlined, we analyzed the problem in much greater detail and formed a series of refined requirements which were used as guidelines for the development of our project. The table on the following page outlines these requirements along with a basic reasoning for each.

2.1 Requirements

Table 2.1.1. Requirements and Reasons

Priority	Requirement	Reason
1	The apparatus should be able to detect unsafe operational conditions through the use of sensors and react accordingly	User safety is of top priority
2	The software should be able to capture, modify and command the robotic arm to perform the desired operations	Without software functionality, the apparatus fails its purpose
3	The system should incorporate real time feedback with respects to movements of the user's head/face.	To perform satisfying makeup application, the natural movements of the human body must be taken into account
4	The speed of the arm should not exceed 2 cm/sec	Estimate necessary for the material properties of the applicative medium
5	The user should have an easy-to-access user interface for makeup design selections	An interface is necessary for visualization and style selection
6	A mirror or form of progress indication	For ease of use
7	The applicant of the makeup faces the camera	The facial recognition works best when the person is facing the camera
8	Applicant does not wear glasses	Glasses may pose an issue when the robot moves near the eyes
9	Room is well lit	The camera sometimes has trouble detecting darker skin tones in lower light

The requirements listed in *Table 2.1.1* are listed in order of importance, with safety given the utmost priority since a well-designed and highly efficient product can still be rendered absolutely useless if it causes any form of harm to a client. It is necessary to consider the most sensitive parts of the human face such as the eyes and the skin around the eyes during application. Touching the eyes, permeating or applying excessive pressure to the skin and causing any sort of discomfort should be avoided at all costs. The criticality of this requirement governs every design choice and decision that follows. In addition to preventing the physical dangers of using a robotic arm near sensitive areas, it is necessary to plan for scenarios in which sensors or code structure malfunctions may occur. With this in mind, the final product may contain either manual or verbally operated shutdown/kill switches, or a combination of both.

With respects to performance, the ideal operation should be relatively fast in comparison to human application timing and also allow for the user to move slightly during the application, without having such movements hinder the outcome. Additionally, if the user needs to take a break or something of similar nature, the system should be able to pause and continue exactly where it left off.

2.2 Interfaces

For our system to have a high ease-of-access factor, it is desirable to have a user interface though a computer program. The user should be able to intuitively navigate menus and select the desired look without confusion or further human instruction. A secondary but less flexible configuration would be to utilize a set of presets that require only the press of a

button with respect to it's preset. The software would then apply this configuration specifically to the user's 3D model of the face, which is accomplished by the use of the mounted Intel® RealSense camera. With respect to the interfacing of the components of the apparatus with each other, the camera and sensors will be integrated with the arm through the Arduino Mega 2560 mainboard, which is built into the arm assembly.

3.0 Summary of Solutions

For our approach to the given problem the bulk of our components can be purchased and assembled together with relative ease, with the programming and integration of components proving to be the greater challenge. The governing thought process involves a solution in which we minimize the production cost of our apparatus/product, but only to an extent where the requirements are met satisfactorily and where the operation of our final result is safe and efficient. The design utilizes a robotic arm made by the robotics company uArm: the uArm swift Pro model, and an Intel® RealSense depth camera. An ultrasonic sensor, controller and universal holder as an end effector— all manufactured by uArm— were also chosen for our design.

Operationally, the apparatus first takes a picture of the users face when they are within range (operation is manually initialized) of the arm, which is at about 1 ft or 320 mm. The user must be seated in a chair with the apparatus at the edge of a table. Once the user makes the desired selection through a physical button on a controller and the desired makeup type is attached to the universal arm holder, the physical and autonomous operation begins. Facial landmarks are used to pinpoint 68 distinct locations on the face, using libraries provided in

Python software. An image obtained from the Intel® RealSense camera is used for the application of the facial landmarks. By mapping the landmarks in a loop, a real time view of the 68 points can be obtained as the picture is refreshed quickly enough to simulate a video. Using the information obtained from the Intel® RealSense camera, depth information is provided alongside the coordinates given from the landmark tracking. This gives x, y, and z coordinates which the robot arm can be set to move towards. This operates in conjunction with the uArm ultrasonic sensor, which gives the system a distance reading between the robot's base and the user. If for any reason the user leaves the operational limit of 1 foot, the system will pause and resume only when the user is once again within range. The continuation requires manual confirmation which is required as a safety precaution. A very basic schematic of system integration logic is shown in the figure below.

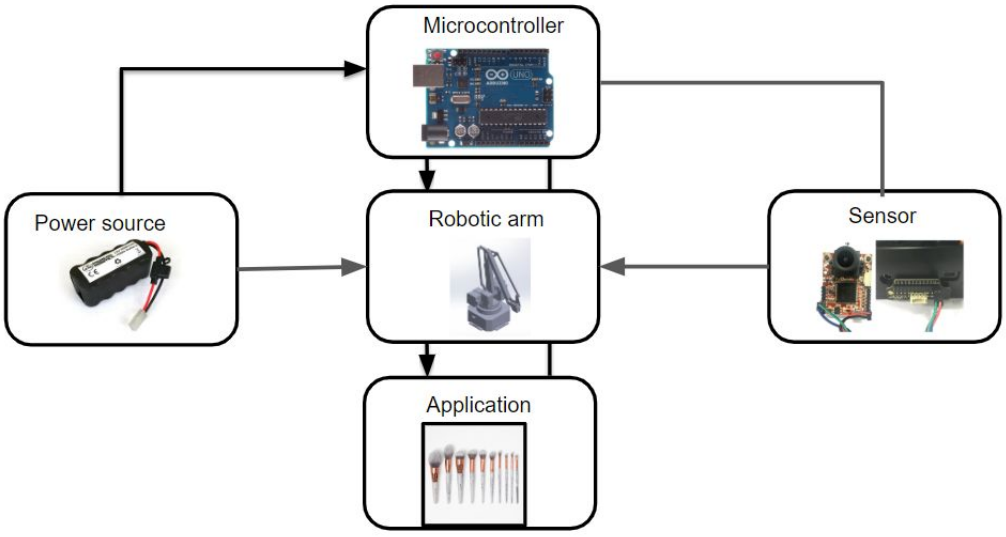


Figure 3.0.1 *Component Operational logic*

3.1 Trade Studies and conclusions

In the selection of components and parts used to design our project, we researched a variety of sensors, arms and controller boards for comparison. Using multiple categories of interest for comparison, we adopted a scale of 1 to 3 for scoring with 1 being the lowest and 3 being the highest. These results of these trade studies are shown in the following tables.

Table 3.1.1 Robotic arm trade study

Weight	3	3	3	2	2	2	2	2
Robotic Arm	**Affordability**	**On-board software**	**User-friendly**	**Degrees of Freedom**	**Repeatability**	**Expandability**	**Appearance**	**Compactness**
uArm Swift Pro	2	3	3	2	3	3	3	3
Universal Robots UR3	1	3	3	3	3	2	3	1
Braccio	3	1	2	3	1	1	2	2
Weighted score:	**uArm Swift Pro**	Universal Robots UR3	Braccio					
	52	45	36					

The uArm swift pro was chosen due to its overall outperformance or tie with the other arms, with the only negatives being the price and one less degree of freedom. In all other areas the uArm receives excellent ratings and it was determined that this arm is ideal for the requirements and our project needs. Additionally, the uArm is popular amongst robotics

enthusiasts on the internet, meaning there is a large pool of open source code and expertise.

The robot is also designed to be expandable with a wide variety of sensors and robotic parts.

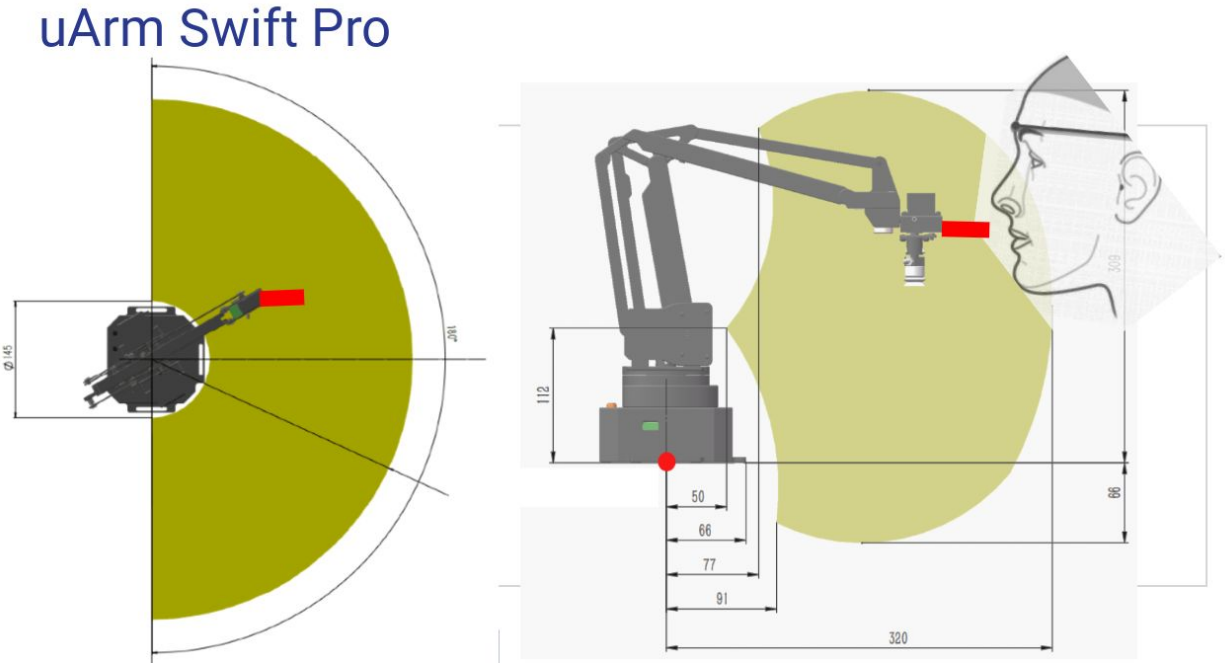


Table 3.1.2 Microcontroller trade study

Type of Microcontrolller	**I/O performance**	**CPU Intensive Performance**	**Web Support and Example Projects**	**Compatibility (with robotic arm and programming language)**
Weight	2	2	1	1
Raspberry Pi	1	2	3	1
Arduino	3	2	3	3
Weighted score	Raspberry Pi	**Arduino**		

	7	**11**		

When comparing the two major types of microcontrollers to use for our project, we decided to use an Arduino for a variety of reasons. The input/output performance is significantly more versatile in comparison to a Raspberry Pi, as is the case with compatibility with our chosen robotic arm. With respects to CPU performance and the availability of code to be found, they are relatively equal.

Table 3.1.3 Camera type trade study

Weight	3	1	2	3
Type of Sensor	**Programming ease**	**Depth tracking**	**Image sensing**	**Affordability**
OpenMV Cam H7	3	1	3	2
Pixy 2	2	1	2	2
Intel RealSense Camera	3	3	3	2
Weighted score	**Intel RealSense Camera**	Pixy camera	OpenMV Cam H7	
	24	17	22	

For selecting the type of camera to fit our 3D face processing and tracking needs, the initial choices were between the OpenMV Cam H7 and the Pixy 2. The research proved that it should be slightly easier to program the OpenMV, in addition to it having better image sensing capabilities. With all other areas at a tie, it was determined that an OpenMV Cam H7 was ideal for our needs. However, after consultation with uArm on their product and recommendations along with observing our budget allowances, we felt that Intel's® RealSense camera was more fit for this project and our design was then modified to use it.

Table 3.1.4 Programming language trade study

Trade Study: Programming language	**Experience with Language**	**Compatibility with Microcontroller**	**Accessibility (in computer labs)**	**Installation Complexity**	**Ease of use**
Weight	1	1	1	1	1
MATLAB	3	2	3	3	3
Python	1	3	2	3	2
C++	2	3	1	1	2
Weighted score:	**MATLAB**	Python	C++		
	14	11	9		

As integral as the components and the assembly of components, is the programming behind the entirety of the operation. With this in mind, a trade study on the ideal form of programming language was necessary. The clear winner in this trade study was MATLAB, due in no small part to the experience, accessibility and ease of use for this group. Although Python and C++ are more ideal when it comes to using Arduino microcontrollers, it is likely that our lack of experience with the languages may result in compiled complications and as a result, MATLAB became the definite choice for our programming needs.

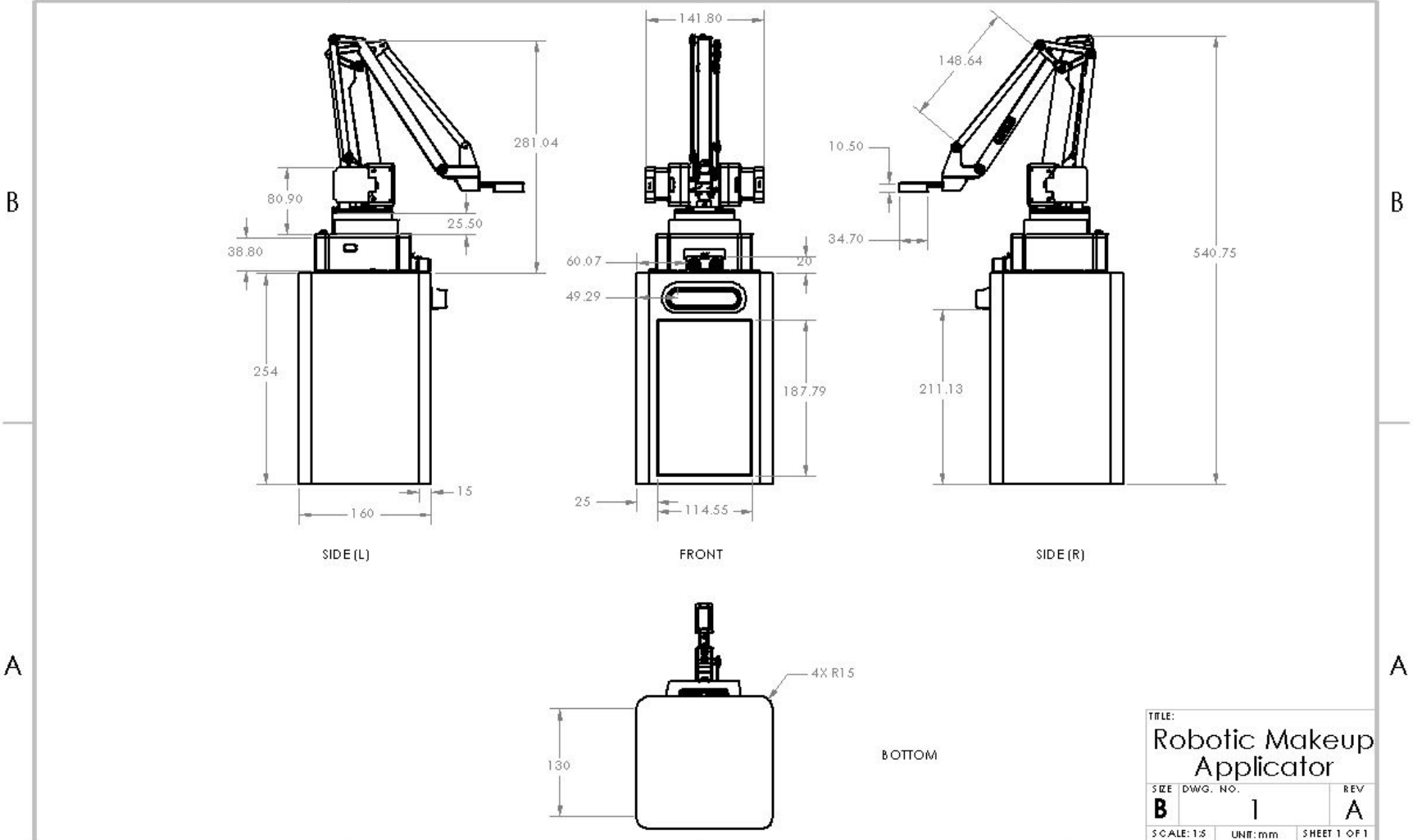
3.2 Cost Analysis of product design

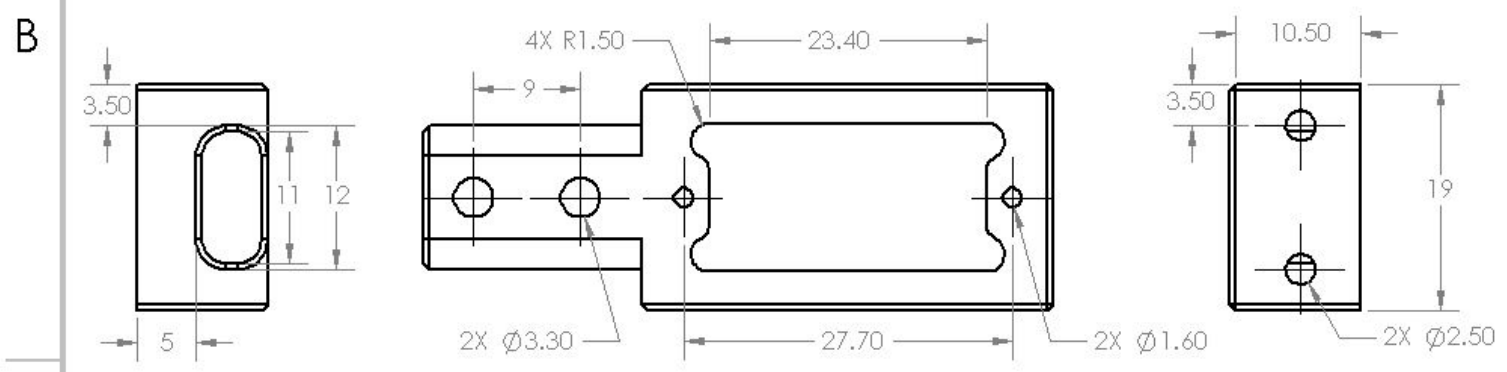
Table 3.2.1 List of components and costs

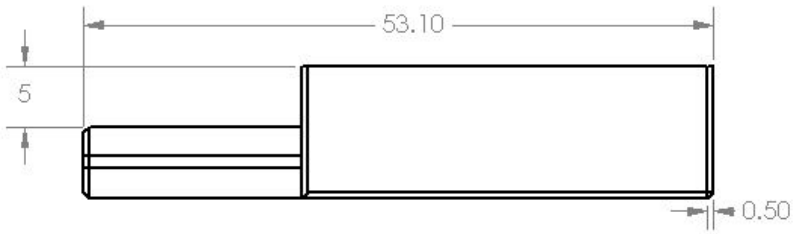
Part	Cost $	Link
uArm Swift Pro	749.00	https://store.ufactory.cc/products/uarm?variant=22044585328758
uArm free of cost	-749.00	
Intel RealSense Depth Camera D435i	199.99	https://store.intelrealsense.com/buy-intel-realsense-depth-camera-d435i.html
uArm Universal Holder	25.00	https://store.ufactory.cc/products/uarm-universal-holder
uArm Ultrasonic Ranger	19.90	https://store.ufactory.cc/products/uarm-ultrasonic-ranger
uArm Controller	99.00	https://store.ufactory.cc/products/uarm-controller
Total	$343.89	

The budget assigned to us was not fixed, but it was desired that we keep costs at or under $1,000 so that we left room for extra purchases or unforeseen events. The component that took a significant portion— around 75% of the allocated budget— was the robotic arm. After confirming our choice as the result of the trade studies, we debated this purchase due to the lack of reserve funding after purchasing the remaining components. Thankfully, we were able to win the exact arm in a giveaway event on social media which freed up our budget, allowing us to choose a better and more expensive camera which improved the performance of our design. With the new overall expenses of $393.89 before shipping and taxes, we have an estimated 55% of the proposed budget limit left, leaving us at a major advantage compared to our initial situation. If we require more sensors or more materials for our design in the future, we have a comfortable amount of flexibility to modify our design to fit those requirements.

3.3 Manufacturing / descriptive drawings


4
3
2
1
B
A
281.04
80.90
25.50
38.80
254
15
160
SIDE (L)
141.80
60.07
20
49.29
187.79
25
114.55
FRONT
148.64
10.50
34.70
540.75
211.13
SIDE (R)
4X R15
130
BOTTOM
TITLE:
Robotic Makeup Applicator
SIZE
B
DWG. NO.
1
REV
A
SCALE: 1:5
UNIT: mm
SHEET 1 OF 1





FRONT

TITLE: Universal Arm Holder		
SIZE A	DWG. NO. 2	REV A
SCALE: 2:1	UNITS: mm	SHEET 2 OF 2

APPLICATION

DO NOT SCALE DRAWING

4.0 Manufacturing and Assembly

With all the necessary components, the manufacturing process consists mostly of assembling the robotic arm with the given instructions. After, the mounting and positioning of the camera and uArm ultrasonic ranger sensor is all that is necessary for the overall assembly to function as intended. The only raw form of manufacturing required for this project is the design and construction of an extension holder to attach to the uArm universal holder, which is necessary because we need the application medium to be perpendicular to the points of application on the user's face. The uArm universal holder is attached to the arm in such a way that the application medium will be parallel to the user's face and therefore it would be impossible to apply makeup. This piece will be constructed through CAD software and will be 3D printed. Due to the relatively small amount of material required for 3D printing this component as shown in the drawings, it is possible to print this part free of cost. As a result, manufacturing and assembling will likely come at no extra cost overall. Factoring in the assembly of the arm with the positioning of the camera and sensors, it should be possible to assemble the apparatus within a few hours. The 3D printing will depend on the chosen material (types of plastic), but the production of the part may take anywhere from half an hour to slightly over an hour.

5.0 Plans for testing

Testing our design to see if our requirements are met satisfactorily is more of an "output analysis" type focus as opposed to a numerical force or strain study. That said, the main forces under consideration are the forces that the makeup applicators (brushes, lipstick or pencils) subject on the arm of the robot, and the forces that are applied to the face of the user. The force or weight of the applicator with respect to arm length, angular acceleration and angle of contact does not matter as long as the applicator weighs 500 grams or less [1]. With this limitation set by the manufacturer, adhering to this restriction negates the need to do any type of force calculation or testing with respect to what is placed in the robotic arm universal holder.

The primary force that needs to be considered is the applied pressure to the human face during makeup application. The force applied will vary based on the shades of makeup for the selected styles, along with the area of application. While a variable force is necessary, it should be limited to a level where human discomfort can never occur. While this may be a matter of modifying the software, this type of force and pressure application is something that should be tested. To do such a delicate form of experimentation, we will first allow our arm to run repeated trials of multiple presets or modifications on different mannequins. Ideally these mannequins will have different facial structures so that we can run repeated trials of every pre-set/modification, on various face types. During experimentation, the mannequin heads will be placed on a platform of sufficient height with only a light support to brace up against, such as a coffee mug or another item of similar weight. During the trials, we will observe for any

excessive head movement which would indicate the occurrence of too much applicator force on the face. One instance of excessive force would count as an automatic trail failure, since the user should never be subjected to discomfort. In addition to head movement, we will observe the accuracy of the application especially around sensitive areas of the face such as the eyes. If our robotic arm applies even a faintly visible mark on a surface that should not be touched, that trial will also be counted as an automatic failure. If we observe the presence of at least one trial failure after performing all presets on every mannequin head, the apparatus cannot be verified as safe for human operation and it will require tuning of some sort.

To test the apparatus for handling of head movement during operation, we can move these mannequin heads during testing at rates that appear reasonable. As there is no readily available data on the average velocity and 3D directional movement of a human head, this movement should be based on our best judgment. As long as the movement is not illogical, the same failure criteria from the previous experiment is used in this experiment.

The costs for testing may range anywhere from $7 to $25 per head, depending on the type and quality of mannequin head that we desire. The time for testing depends on the number of presets and the number of mannequins used. If an average makeup application takes three minutes and we have five pre-sets, we will need fifteen minutes per head. Additionally, time to test requires the consideration of repeat trials if any tuning or code modifications are performed. If we are able to verify extremely high reliability and safety probabilities, self-testing is possible but the trials should be done wearing low profile goggles to protect the eyes. Depending on our results we may be required to state that our product

should not be used around the eyes, that protection should be worn or more hopefully, that our product can be used for anywhere on the face.

References

[1] https://cdn.sparkfun.com/assets/9/8/a/c/0/uArm-Swift-Specifications-en.pdf

[2] https://www.ufactory.cc/#/en/

[3] https://www.youtube.com/watch?v=5DnKot3mMSc - Designing Robot Manipulator Algorithms Matlab

[4] https://www.youtube.com/watch?v=q4mIAX-Rcmg Robot safety control and response to unexpected object