

Camera Tracking Marshmallow Shooter

Spencer Witte

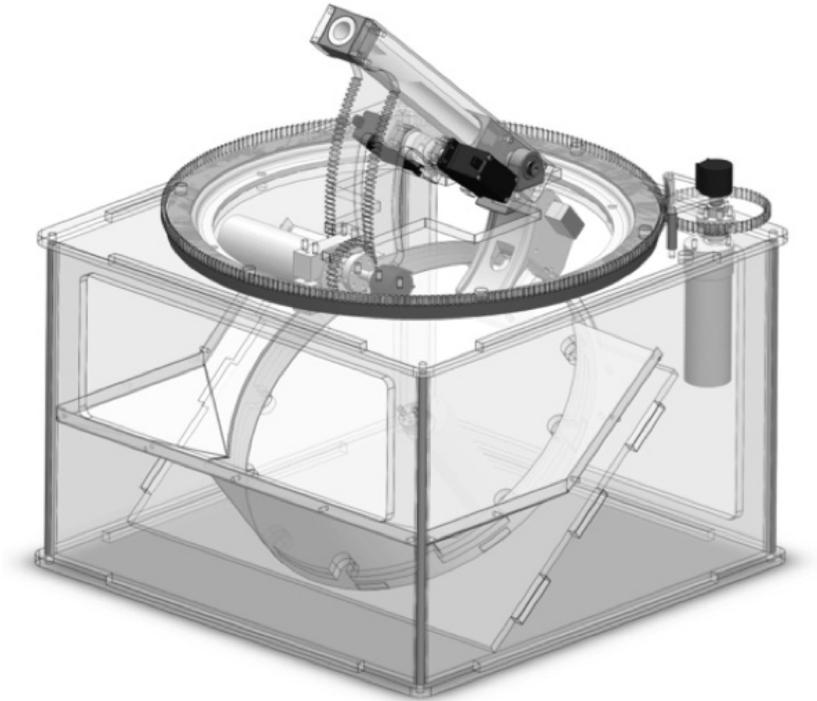


Figure 1: Camera tracking marshmallow shooter.

1 Abstract

The concept of this project is to build a device capable of launching a marshmallow at a target. This entails a turreted marshmallow gun with some sort of tracking ability. While the task may be essentially pointless, it provides a challenging design task. The purpose of this project is to be ambitious and to explore a field that I normally do not

This paper was written for Dr. James Dann's Applied Science Research class in the spring of 2015.

touch. The Marshmallow Turret Project is not about the end product so much as the task of designing it. A task this complex will require clever engineering, intricate geometry and complex software, but it should be a lot of fun and a tremendous learning experience. The idea originated from other projects such as the Confectionary Cannon built by students at Olin College, and the Popinator from Popcorn, Indiana; however, I would like to deviate from their designs and add my own unique features [1, 2]. This project is not about copying what others have made in the past, but about thinking through my own ideas.

2 Introduction

Below I have listed two of the projects from which I have gained the most inspiration. The Popinator was a stunt that a popcorn company tried to use to generate interest in their product. The idea was that it would recognize the word “pop” and then triangulate your location based upon the sound of the word. It would then shoot a piece of popcorn in your direction. It turns out, however, that the machine was a hoax and the company actually had no intention of making it beyond their commercials. I enjoyed the consumer’s standpoint where they would just pour in a bunch of popcorn and the machine would magically shoot one out, and I tried to incorporate the idea into my own project.



Figure 2: *Popinator popcorn launcher.*

A group of Olin College undergraduates took on a marshmallow cannon as their final project. It was a cannon mounted on a turret with a camera. The camera located human faces and then shot a marshmallow towards them. While I liked their use of camera tracking to find their target, I wanted to build more features into the design of my marshmallow shooter.

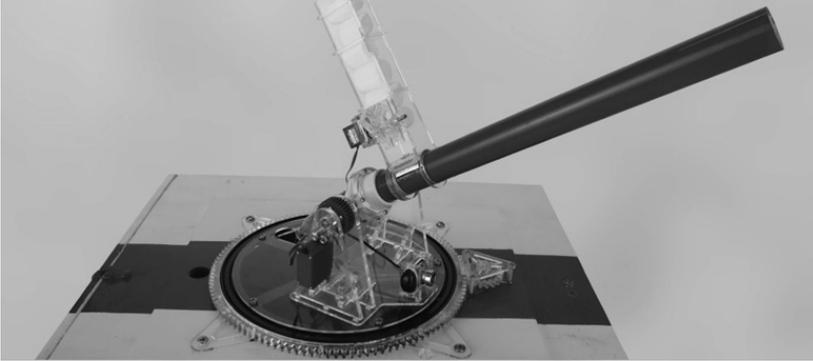


Figure 3: *Confectionary cannon.*

While both of these projects had aspects that I incorporated into my final design, I also had to design mechanisms on my own. The individualizing and loading system are both completely custom designed. The second reason why this is difficult is that I am unfamiliar with the software aspect of the project. I have always found camera recognition an interesting field to study, however, I have not had the opportunity to explore this field. This marshmallow turret will allow me to research and learn about an unfamiliar field while also providing a challenge in a field I am very comfortable in.

3 History

When I first set out on this project, I hoped to use facial tracking to determine the target. While I currently do not have this feature on the final product, I found that learning about how it functions helped me find a different solution.

Facial detection is the ability to identify the presence of a face in a digital image. Woody Bledsoe, Helen Chan, and Charles Bisson are recognized as the pioneers of facial detection for their work in 1964 and 1965. Most of the work they did was unpublished due to their funding coming from an anonymous secret intelligence agency;

however, their challenge was to design an algorithm that could identify people in an image. They were tasked with facial recognition, the ability to identify a specific human from an image. This early system worked geometrically, first locating facial features in the image and then defining a group of facial features as a person's face. For example, the algorithm could identify eyes, mouths, and noses, and, by their location in reference to each other, could identify the location of the entire faces. While Bledsoe's, Chan's and Bisson's goals were to identify specific people in photographs, they had, along the way, laid down the framework for facial tracking.

Afterwards, research continued at Stanford Research Institute and by 1997 other universities around the world began to publish their own algorithms, and work still continues today. Through these recent efforts, practical requirements have become apparent for real world use. A facial tracking system must have a small dependency on lighting and changes in facial expression, and must report as much information as possible [3].

If I were to use facial tracking in my product, I would get a coordinate location as to where the face in the image is located. The turret and angle adjust would have to react accordingly to aim at the target. This led me to the solution I am currently using, blob tracking. Blob tracking tracks a colored blob instead of a face. In essence, it operates similarly in that I use the coordinates of a colored object to aim at my target, but brings down the computing power required, allowing me to do all the processing on an onboard processor instead of my computer.

4 Design

4.1 Shooter

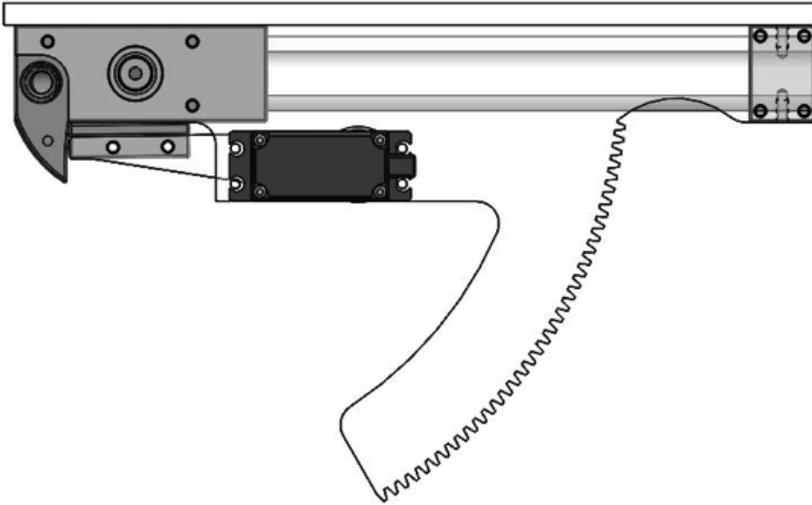


Figure 4: *Shooter, side view.*

The first prototype I made was a simple pneumatic launcher. I used a pneumatic solenoid, a power supply, and pneumatic tubing, tanks and fittings to mock up a quick prototype. I modified tube lengths and pressure and did not see much change in accuracy. I did notice that I needed a tank after the regulator because air was previously trying to rush through the regulator, creating inconsistent pressure. I also learned that the shape of the marshmallow has a huge effect on the shooter's accuracy. I found a great increase in accuracy when I used a jig to roll the marshmallows into a consistent shape. I also learned that I can control shot power by adjusting how long the solenoid is open. Using an Arduino, I triggered the solenoid using a time delay and was able to adjust how far the marshmallow was shot based on the time delay between when the solenoid was open and when it was closed.

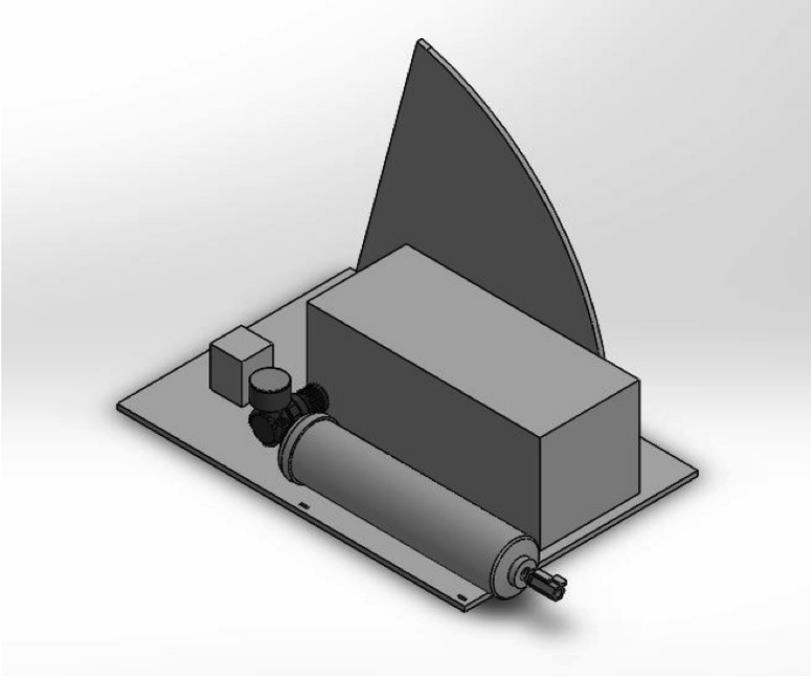


Figure 5: *Shooter prototype, isometric view.*

The design now uses a 9 in. long, 0.5 in. inner diameter tube that is held in place using two 3D printed blocks. The 3D printed block is an easy way to manufacture parts that would normally take a mill and several operations and allows me to make awkward contours that conventional tools would not be able to make. The front block is mainly used to contain the tube while the rear block contains the loading mechanism of the shooter which will be operated by a servo. The shooter will be mounted on top of a turret that has yet to be designed, allowing it to rotate to aim at a target and adjust the shot angle of the cannon. I plan to laser cut a gear tooth profile into the base plate to rotate the shooter, and another gear tooth profile will be laser cut into the plates alongside the shooter tube to adjust the shooting angle.

4.2 Individualizer

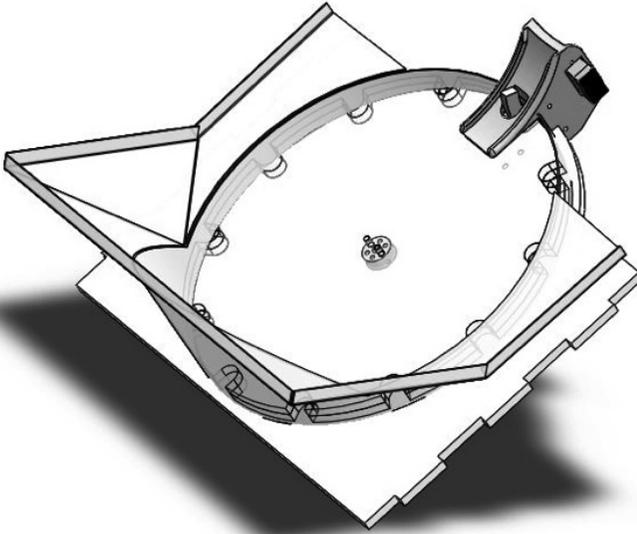


Figure 6: *Individualizing mechanism, isometric view.*

I also prototyped the individualizing mechanism. This mechanism pulls out one marshmallow from the pile of marshmallows. It uses a rotating disk that is mounted at an angle, and it has slots that each fit a single marshmallow. As the disk rotates, the marshmallows fall into the slot and then are lifted out of the pile of marshmallows. I prototyped this pretty quickly and it worked like a charm. I learned that the shallower the angle of the device, the more likely the marshmallow would fall into the slot. However, if taken to the extreme of being perfectly level, the rotating disk would lift more than a single marshmallow, so a balance had to be struck. Using the prototype, I decided that a 40 degree angle was the optimal angle for the individualizer.

The individualizer will use a 10 in. diameter laser cut disk to individualize the marshmallows. The marshmallows will be poured into the trough of the mechanism and the rotating disk will lift them on up onto the loading mechanism. A reflective infrared sensor (also dubbed a top hat sensor) in the loading mechanism will detect when

there is a marshmallow in the slot in the disk of the individualizer and tell the rotor to stop in the correct position for the loader. From there, a servo pushes the marshmallow out of the individualizer mechanism and into the loader mechanism.

4.3 Turret

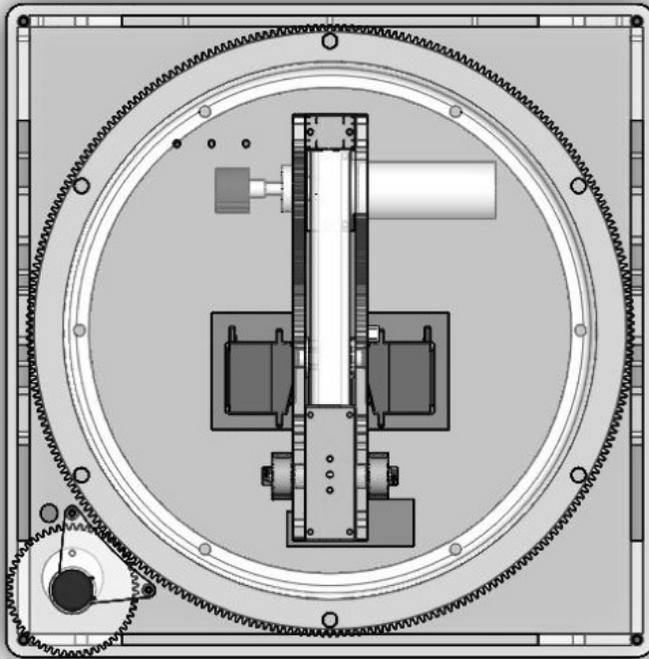


Figure 7: Turret, top view.

The shooter is mounted on a turret, allowing the shooter to aim at its target. The turret is mounted on top of a lazy susan ring, allowing the top of it to rotate freely. The plate on top of the turret has a gear profile on its outside. Another laser cut gear is then attached to the motor and potentiometer. The motor is in the bottom left corner of Figure 7 and uses a gear ratio of 55:260 to power the turret. I'm expecting the gear ratio to be too large and thus make the rotation too small. I am looking to spec

a different motor, which has a different gearbox head on it. Connected to the motor is a 10 turn potentiometer that will help the turret locate its home position and its max rotation positions. I expect to rotate the turret a maximum of 180 degrees, but, mechanically speaking, it is able to rotate over 720 degrees based on the limits of the potentiometer.

4.4 Angle Adjust

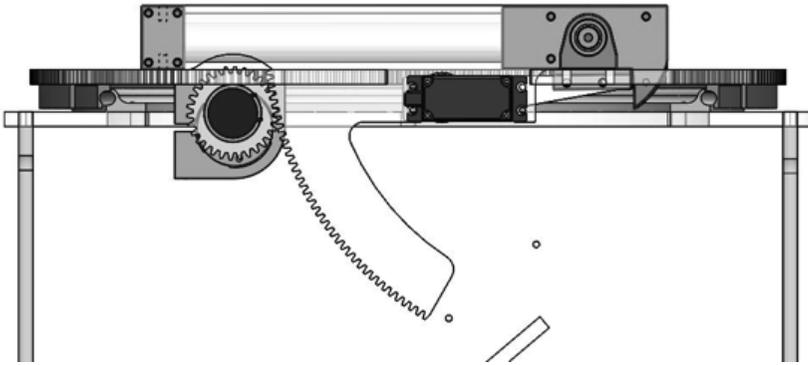


Figure 8: Angle adjust down.

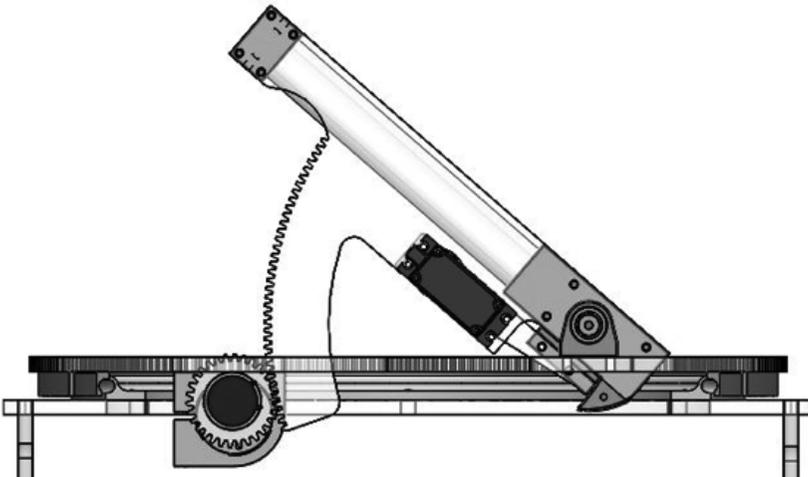


Figure 9: Angle adjust up.

The angle adjust mechanism allows for the shooter to adjust for the height of the target. The shooter is mounted to pivot blocks that contain bearings and shoulder bolts. A gear profile was laser cut into the shooter side plate and meshes with a laser cut gear. The gear then mounts to a motor and potentiometer so that the angle can be powered and controlled. Using the potentiometer, we can calculate the angle of the shooter by knowing the rotational angle of the motor and the distance from the gear profile to the pivot axis. The angle adjust is designed to have 45 degrees of rotation, which should be more than enough for the desired shots.

4.5 Loading Mechanism

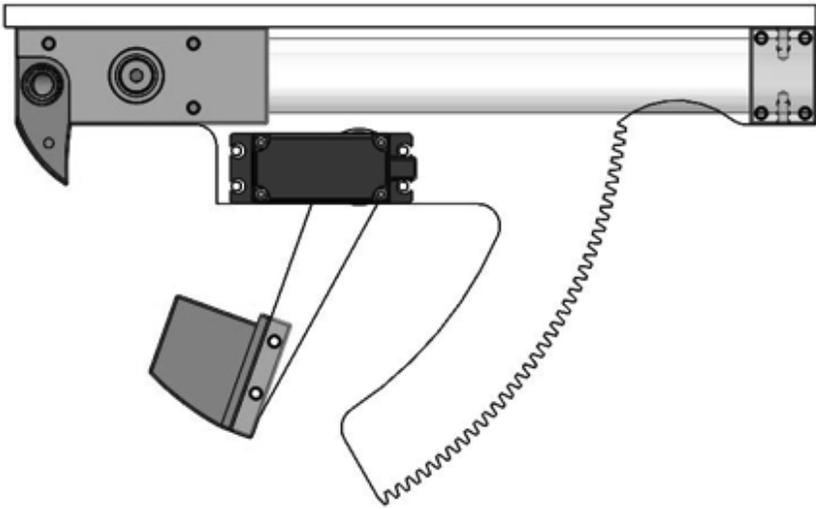


Figure 10: Loader, down side view.

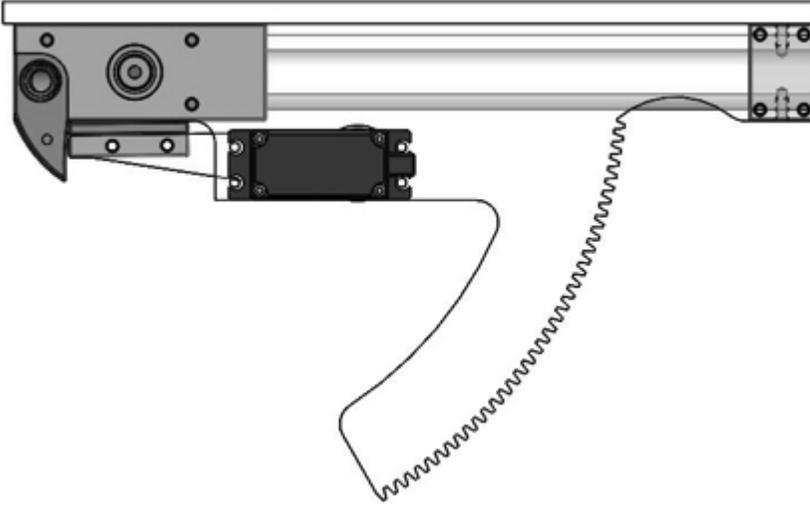


Figure 11: *Loader, up side view.*

The loader mechanism has undergone several iterations since its initial design. The goal of this mechanism is to place the marshmallow in the shooter and to compress the marshmallow to increase shooter consistency. The current design uses a lever that lifts the marshmallow from the individualizer to the shooter. The marshmallow is then compressed between shooter tube and the individualizer lever arm. I have had to prototype the compressing profile. The nub that pushes against the marshmallow has gone through several different profiles. The rectangular profile would squeeze the marshmallow between the two parts. The round profile, on the other hand, took up too much cross-sectional area for the marshmallow to fit. Currently I am using a hybrid between the two profiles. While I have not found a geometry I like, the part is designed to be adjustable. The shooter also includes a ramp for the marshmallow to slide on in order to be moved to the shooter.

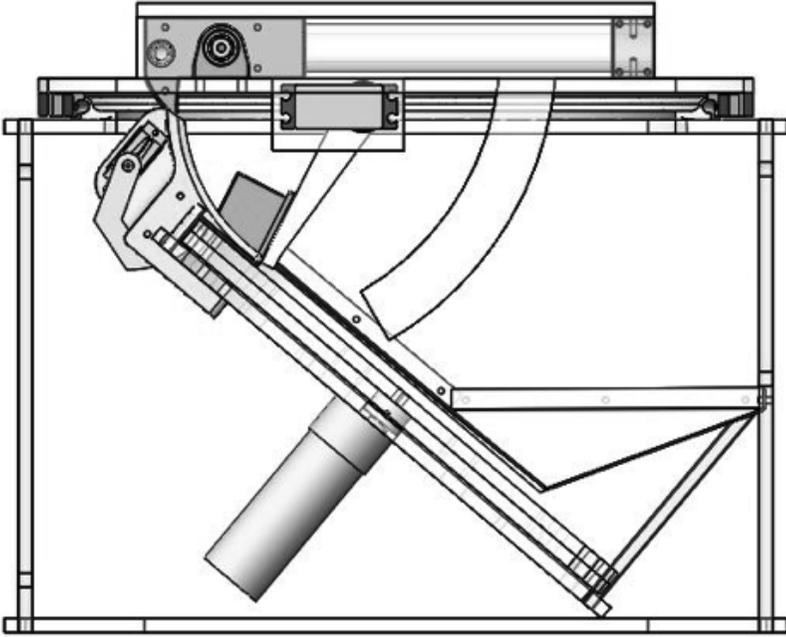


Figure 13: Loader compression hybrid profine part (grey).

4.6 Blob Tracking

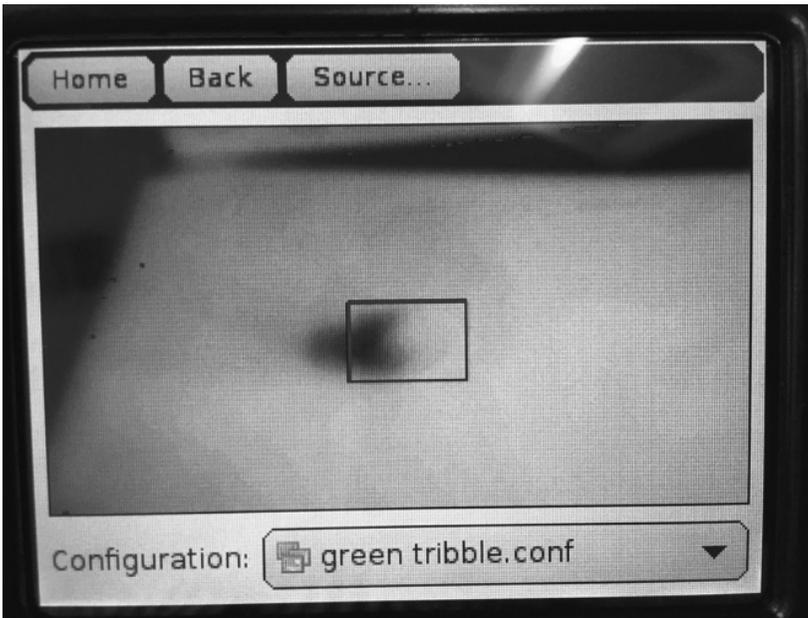


Figure 14: Blob tracking a green pom-pom.

My first step in the vision tracking process is to perform blob tracking. Blob tracking is the ability to identify a group of pixels as an object. I first input the color configuration of the object I want to track. I currently set it to track a green pom-pom as seen in Figure 14. The camera then looks for adjacent pixels that match the range of color and saturation described in the preset configuration. The adjacent pixels form a blob and can then be identified as an object. From there, the coordinates and the size of the object can be obtained in terms of pixels.

5 Theory

5.1 Compressed Air

My design utilizes compressed air to shoot the marshmallow. I found that I could consistently control the shot power for my marshmallow shooter. Compressed air is a method of storing energy. A compressed

gas, if not contained, will decompress, turning its potential energy into kinetic energy as it moves. I am using air as my gas because it is the most readily available gas. The machine is designed to direct the airflow of the decompressing gas through several tubes and fittings to the shooter tube which holds a marshmallow. On its way to reaching equilibrium with the outside pressure, the air pushes the marshmallow out of the shooter tube, giving it kinetic energy.

If we look at the physics of the shooter, we can understand why it is consistent. Both the pressure (through the use of a pressure regulator and a high pressure supply) and the cross sectional area of the shooter tube are constant (assuming the loader compresses the marshmallow to fill up the shooter tube cross sectional area). The equation $P=F/A$, where P is pressure, F is force, and A is cross sectional area, tells us that if we have constant pressure and cross sectional area, then we will apply a constant force to the marshmallow. Consistent force means that the marshmallow should undergo a constant acceleration due to $F=MA$ where F is force, M is mass, and A is acceleration, assuming all other forces acting on the marshmallow (e.g., friction) are also constant or negligible. This also tells us that the acceleration, and thus the exit velocity, are dependent on the mass of the marshmallow.

I plan to perform tests to measure the acceleration and the exit velocity using an Arduino and a beam break sensor. I will place the beam break at the muzzle of the shooter tube so that it triggers just as the marshmallow leaves the tube. I will be able to measure the time it took from the solenoid opening to when the marshmallow leaves the shooter tube, and by knowing the distance between the two sensors, I will be able to calculate the exit velocity. Using the initial and final velocities, I can calculate the average acceleration.

5.2 Solenoid

A solenoid is a device that turns electric current into linear motion. It uses a coil of wire to produce a magnetic field and then manipulates the magnetic plunger. When current flows through the coil, a magnetic field is produced and the plunger is either repelled or attracted. In this design,

the solenoid is a valve for the pneumatics. This allows me to control when the compressed air is released and for how long. Controlling how long the valve is open is another way to control shot power.

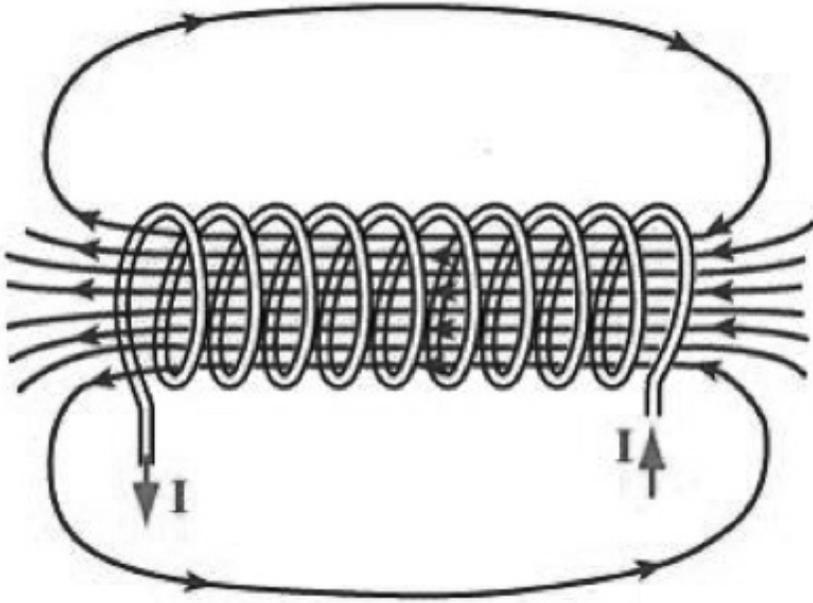


Figure 15: Solenoid diagram.

5.3 Blob Tracking

Before I get to facial tracking, I want to be able to track a target first. Blob detection is the ability to recognize an object based on its uniform appearance. Blob detection groups pixels that have similar brightness and color, and if they are adjacent to each other, forms a blob. Using blob tracking, I can define a specific color and then get the location of that object on the screen. I can make a target that a camera could follow. The x coordinate of the object will give me the left-right direction of the object, the y coordinate will give me its height, and the size of the blob will tell me the distance to the blob since I will know the size of the target. I plan to mount the camera on the turntable. The turntable will rotate until the the target is centered in the camera. Using the size

and y coordinate location, the machine will adjust the angle and shot power of the shooter.

5.4 Projectile Physics

Gravity causes any projectile launched on Earth to follow a parabolic path. From the camera we can obtain the target x position, the y position, and how far away it is given the actual size of the object. Using the x position, the turret is able to aim the shooter at the target. The size of the target will tell us how far away the object is from the shooter, and the y coordinate tells us how tall the object is. We also know that the exit velocity will depend on the pressure of the compressed air. From here we can use projectile physics to find the angle at which the turret needs to be set.

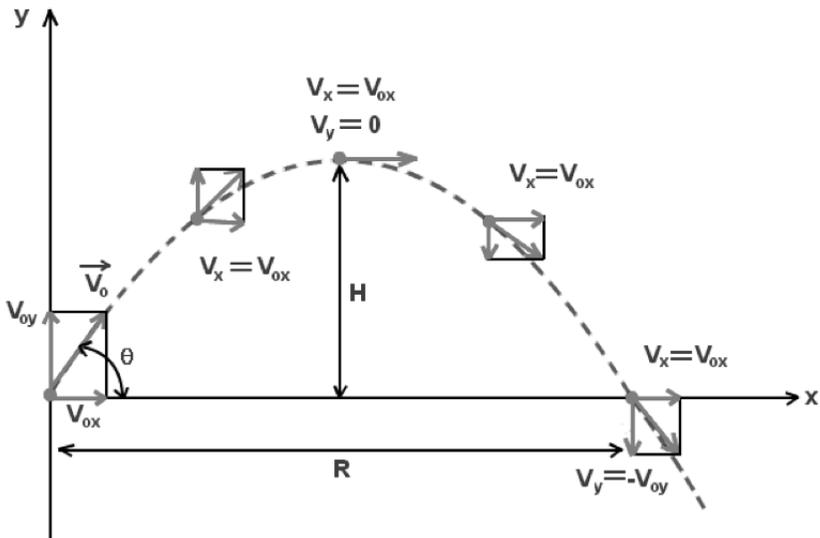


Figure 16: Projectile path and velocity vectors.

When a projectile is launched, it is easiest to split the velocity into two different vectors, one horizontal and the other vertical. In Figure 16, the initial velocity vector V_0 is split into the vertical vector V_{0y} and the horizontal vector is V_{0x} ($V_{0x} = V_0 \cos\theta$ and $V_{0y} = V_0 \sin\theta$). V_{0x}

is constant because there is no force acting on the projectile in the horizontal direction, however, the vertical vector V_{oy} varies due to the force of gravity. For the horizontal component we use $d = vt$ and replace v with V_{ox} . Solving for t , we get $t = d/(V_o \cos\theta)$. For the vertical component we use the equation $h = vt + 0.5at^2$ and replace V with V_{oy} , t with $d/(V_o \cos\theta)$ and a with g , giving us $h = V_o \sin\theta (d/(V_o \cos\theta)) + 0.5g (d/(V_o \cos\theta))^2$. Knowing h and V_o , we can solve for θ , the desired launch angle. This doesn't take into account air resistance. Air resistance would apply a force on the horizontal velocity vector and produce a negative acceleration on the projectile. However, I expect air resistance to be so low as to be negligible.

5.5 Control Loops

The turret and angle adjust both use control loops to control their motions. A control loop is software that measures the distance between the end position and its current position to determine what the motor's movements should be. The benefit of this system is that you can motion profile the rotation of the motor. In the case of the turret, a potentiometer measures the rotation of the motor. A goal rotation will be set by either the camera or by a preset position. The controller then computes the difference in ticks between the current position and the goal position. The difference in ticks is then inputted into a function specific to the mechanism that returns the motor speed. The function is what determines the profiling of the motion. Currently it is a linear equation that slows down as the turret approaches its target to minimize the amount the turret overshoots the target. It also makes the turret move back in the case it does overshoot the target.

6 Results

For my setup, I placed my marshmallow shooter on the ground at the end of a long hallway. I then ran a tape measure from the front surface of the marshmallow shooter to the end of the hallway. I used a compressor for the compressed air, and initially set it to 75 PSI and the burst time

to 25 ms. The shooter was then set to 20 degrees from horizontal and then I proceeded to shoot 30 marshmallows with each marshmallow being loaded with the automatic reloading system. After getting 30 data points, I switched to 45 PSI and 80 ms burst time and measured another 30 data points. I have included my data in the appendix.

After taking my data, the first thing I wanted to do was confirm that both sets of data were approximately normal. Below are several qq plots, which I used to determine whether the samples were approximately normal.

The plot in Figure 17 is from the 75 PSI sample. It shows a clear linear trend, proving that this sample is approximately normal.

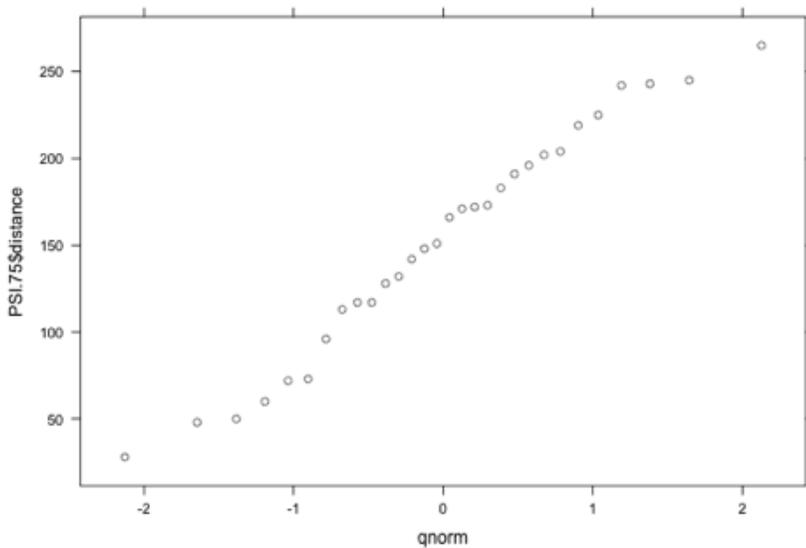


Figure 17: Graph of 75 PSI sample.

This next plot in Figure 18 is of the 45 PSI sample. There is an obvious outlier that causes the data to be skewed, so I then removed the outlier in the next plot.

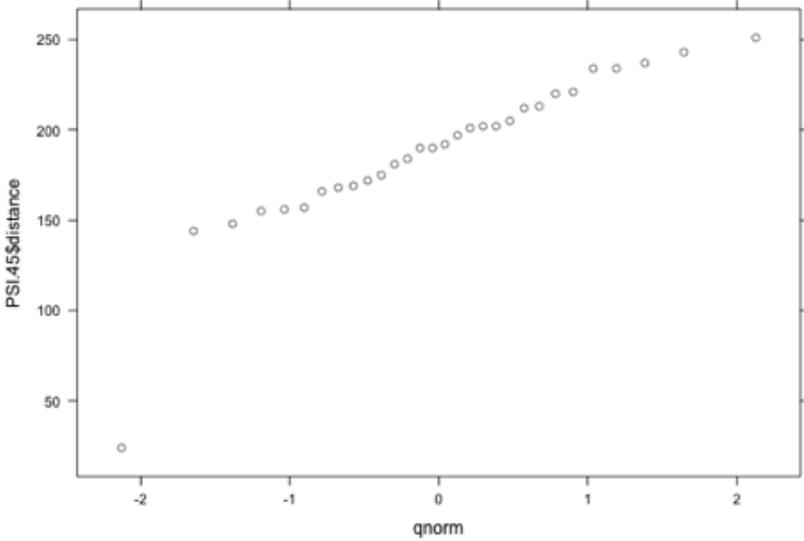


Figure 18: Graph of 45 PSI sample.

The plot in Figure 19 features the 45 PSI sample without the outlier. After removing the outlier, the sample resumes a linear trend that shows the sample to be approximately normal.

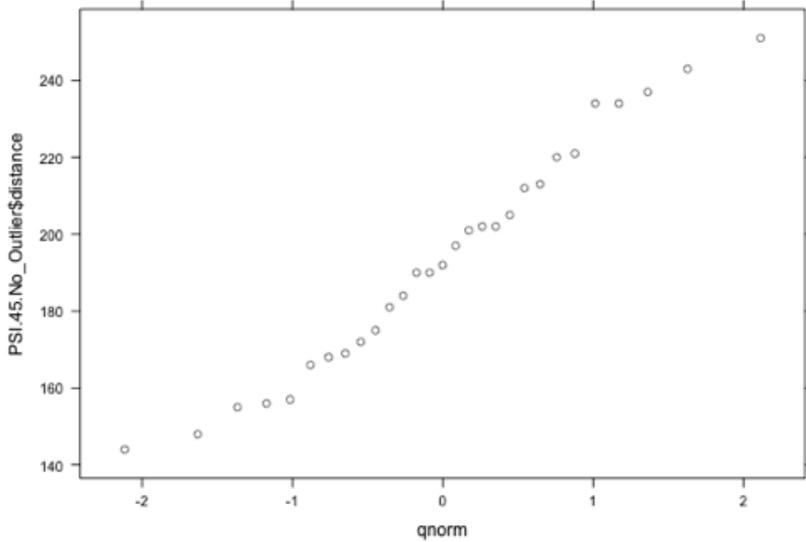


Figure 19: Improved graph of 45 PSI sample.

Assuming that each trial was independent and the variation was random, I proceeded to compare the two samples. I created a box and whisker plot (see Figure 20). The 45 PSI sample has a much smaller spread than the 75 PSI sample.

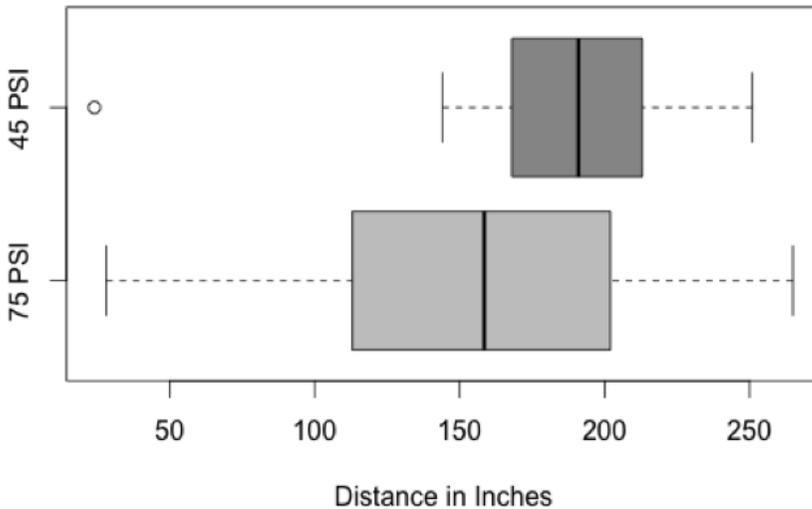


Figure 20: Comparison graph of 45 and 75 PSI shot distances.

I then went on to perform an f test comparing the variances of the 75 PSI sample and the 45 PSI sample without the outlier. In an f test, the null hypothesis is that the variances of the two samples are the same. I include the R output below.

The test found a p value of $8.675e^{-5}$, well below the 0.05 rejection threshold, allowing us to reject the null hypothesis that the variances of the 75 PSI sample and the 45 PSI sample without the outlier were the same.

I found that the 45 PSI shots with the longer burst time were much more consistent in range than the 75 PSI shots with a shorter burst time. If I were to do it again I would like to test more than just the two different pressures and would like to do a range of pressures and burst times. I think my project was limited by how unrefined the shooter was. If I were to do it again, I would have to make the machine more consistent. ●

7 Appendix

7.1 Hardware

Part Description	Need	Cost per unit	Quantity	Total Cost	Vendor
Clear PVC tube 2'	For shooter tube	7.56 each	1	7.56	McMaster
Ring turntable	For shooter rotation	33 each	1	33	McMaster
5/8in 8-32 button head screws	Mounting loading ramp	3.55 per box of 25	1	3.55	McMaster
1/4in 6-32 button head screws	Mounting individualizer chute	3.55 per box of 100	1	3.55	McMaster
24in x 24in 1/32in polycarbonate sheet	Individualizer chute	10.50 per sheet	1	10.50	McMaster
R4ZZ bearing	Shooter pivot	5.76 each	4	23.04	McMaster
1/4in diameter, .5in long shoulder bolt	Shooter pivot	2.41	2	4.82	McMaster
2-56 3/8in button head screw	Mounting micro servo	5.66 per box of 100	1	5.66	McMaster
2-56 locknut	Mounting micro servo	5.66 per box of 100	1	5.66	McMaster
5/64in diameter 7/16 roll pin	Mounting adaptors to motor shafts	6.91 per box of 100	1	6.91	McMaster
1/2in 6-32 button head screws	Mounting 3D printed parts to acrylic plate	3.85 per box of 100	1	3.85	McMaster
10mm M3 flat head screw	Mounting motors	4.67 per box of 100	1	4.67	McMaster
1/2in 5/16-18 button head screw	Mounting turntable	5.54 per box of 25	1	5.54	McMaster
1/2in ¼-20 button head screw	Mounting turntable	5.64 per box of 50	1	5.64	McMaster
6061 Al 1/4in OD tube 3' long	Box standoffs	14.44 each	1	14.44	McMaster
1in 6-32 standoff	Turret pot mount	0.60 each	4	2.40	McMaster
Red and black paired 24 wire	Motor wires	1.20 per 10'	1	1.20	McMaster
1/2in 10-32 flat head screws	Box standoffs	3.35 per box of 10	1	3.35	McMaster

7.2 *Raw Data*

Distance with 75 PSI and 25 ms burst time (in.)	Distance 45 PSI and 80 ms burst time (in.)
265	243
148	175
60	190
196	148
242	144
117	24**
72	157
142	155
183	184
202	156
204	166
96	251
128	169
132	234
191	221
245	201
166	234
225	213
73	197
173	237
113	181
219	212
48	168
50	205
151	172
171	220
172	190
243	202
117	192
28	202

** = Outlier

8 Notes

1. <http://confectionerycannon.com/#myCarousel>
2. <http://www.popcornindiana.com/popinator-project>
3. <http://cbcl.mit.edu/people/poggio/journals/brunelli-poggio-IEEE-PAMI-1993.pdf>

9 Citations

http://en.wikipedia.org/wiki/Facial_recognition_system

<http://cbcl.mit.edu/people/poggio/journals/brunelli-poggio-IEEE-PAMI-1993.pdf>