Infrared Imaging

Building a 3D Scanner

Anne Loverso, Ian Hill, Madeleine Fort, and Tatiana Anthony

May 2014

Introduction

The goal of our project was to build a 3D scanner that would reconstruct in MATLAB a 3D image of an object. We achieved this goal using an infrared distance sensor mounted on a mechanism that rotated and lowered our object to achieve a full scan.

Materials and Methods

Materials Used

Part	Purpose	Quantity
Stepper Motor	Step the platform around in a certain number of degrees per step	1
Sensor cable	Connect the infrared sensor to the circuit	1
NPN transistor	Part of infrared circuit	3
Sharp sensor	Pre-made Sharp infrared distance sensor used to determine distance to object	1
Phototransistor	Digikey number 160-1986-ND determine intensity of returned light	2
arduino	Circuit control and computer interface	1
1N4007 diode	Lowers the voltage from the circuit entering the motor	12
10 kΩ potentiometer	Control the step speed	1
100 μF capacitor	Part of the stepper motor circuit	1
18 by 12 in sheet steel	part of the mechanical frame	1
12 in threaded rod and nut	Mechanical rotation part	2
10-24 screw and nut set	Part of mechanical frame	2 (4 in package)
Polylactic Acid	3d printed gears for rotation of platform	117g
Medium Density Fiberboard	Creating box for mechanical and electrical frame	19 in by 31in
SN754410ne H-Bridge	Used to drive stepper motor	1

Circuit Design

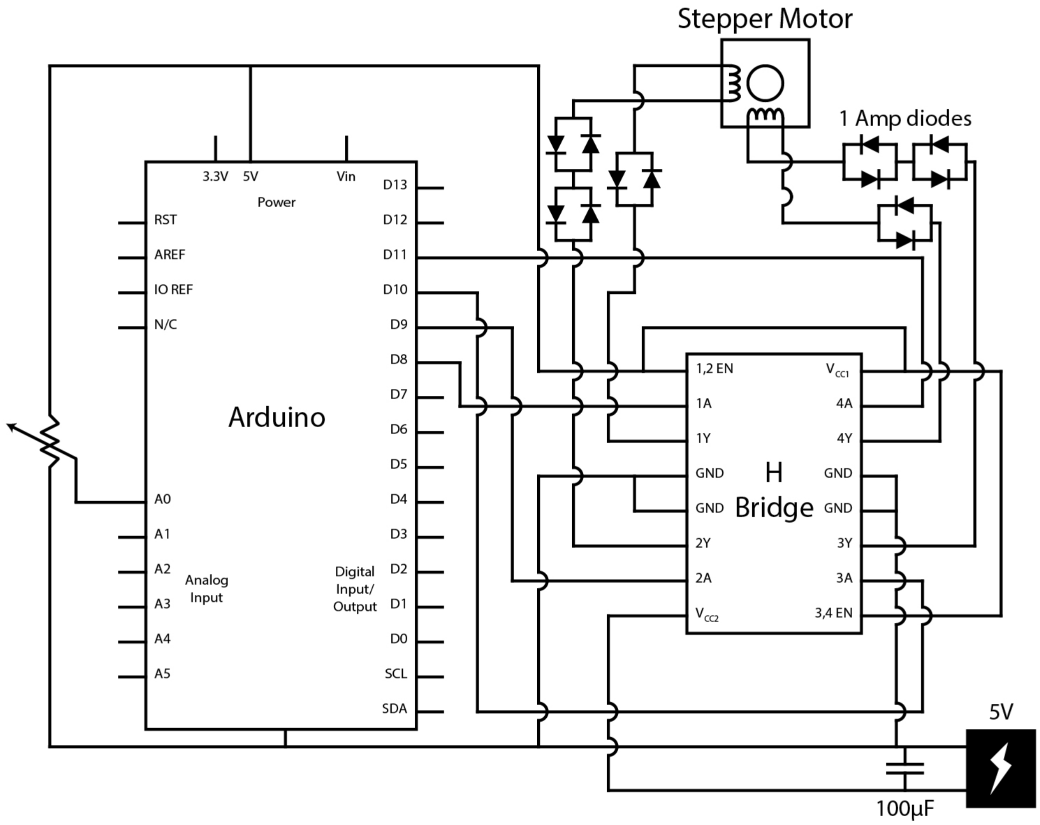


Figure 1: Final diagram of the circuit required to drive the stepper motor.

We found a circuit schematic online[1] for a circuit that could allow an Arduino to operate a stepper motor. However, the circuit was not designed for the same stepper motor as the one we had purchased, so we had to make a few adjustments. Because stepper motors draw current in short bursts, we needed to increase the capacitance of the circuit in order to stabilize the circuit's voltage. Note the 100μF capacitor on the lower right side of the diagram. Also, because the circuit schematic called for a 5V stepper motor and our motor was designed to run on 3.06V we added a series of diodes in parallel to drop to voltage down to a more reasonable 2.9V, figuring that each diode would lower the voltage by 0.7V.

Code Structure

Because we used MATLAB's 3D plotting function to generate our results graphs, we decided to use the Arduino package for MATLAB to write all of our code, so that all variables would be able to be saved in MATLAB format for easy graphing.

The following is our code for running the scanner and collecting data. The final variables we are trying to collect are the vectors X, Y, and Z such that we can use MATLAB's plot3 function to graph them. Here, we record the raw output from the sensor, averaged over 10 collections, in DATA. Then, we use a formula that will be discussed in greater detail later to convert that into

DISTANCE, the radius of the object from the center of the plate. The angle is calculated by how much we know the stepper motor rotates. The DISTANCE and angle are then converted from polar coordinates to cartesian to find X and Y. Z is found by knowing how much the motor turns and that our threaded rod has 13 threads per inch.

```
%a = arduino('COM9');

figure
hold all
axis([-20,20,-20,20,-20,20])

numsteps = 2600 * 6; % the 6 is the height of the object to be scanned

DATA = zeros(numsteps,1);
X = zeros(numsteps,1);
Y = zeros(numsteps,1);
angle = zeros(numsteps,1);
Z = zeros(numsteps,1);
DISTANCE = zeros(numsteps,1);
for i = 1:numsteps
    a.roundTrip(2); % steps the motor by 1.8 degrees
    Z(i) = unitsratio('cm','in')*((2*(i-1))/400)*(1/13); % calculates height travelled
    angle(i) = (1.8*(i-1))*(pi/180); % calculates angle
    sum = 0;
    for n = 1:10
        sum =  sum + a.analogRead(2); % takes 10 data points
    end
    DATA(i) = sum/10; % records average of 10 points
    DISTANCE(i) = 12.24 - (1/((DATA(i) + 53.567)./3313.3)) - 0.42;
    % calculates distance to object in cm based on sensor calibration and
    % linearization
    [X(i),Y(i)] = pol2cart(angle(i),DISTANCE(i)); % converts to cartesian
    plot3(X(i), Y(i), Z(i))
    drawnow()
end

plot3(X,Y,Z)
```

Device Design

Along with the electrical component to our project, we also needed a mechanical system to rotate the object and move it up and down. system is shown in Figure 2.

2.4.1 Structure

The box we used to contain the system is made of $\frac{1}{4}$ inch laser cut MDF. We used no glue to close the box so that we could take it apart if necessary. This proved to be useful, as we needed to screw in the threaded rod from the bottom of the box. Additionally, it allowed us to add a nail through the bottom of the box when we realized the gears were deflecting. This nail was used as a tensioning axle for the longer gear.

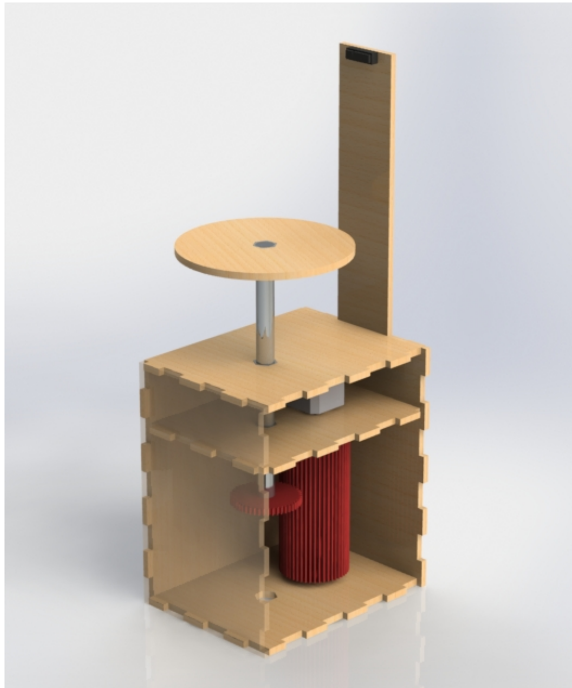


Figure 2: The CAD rendering of our 3D scanner. The box and rotating plate is made of $\frac{1}{4}$" MDF, the gears are 3D printed PLA, and the rod is a $\frac{1}{2}$" threaded rod.

The nut going through the top plate is a drilled out nut. We use it as a shaft to keep the threaded rod from deflecting. It is press fit into the top plate. Similarly, the longer gear is hot glued and press fit on the motor axle. The smaller gear is screwed onto the threaded rod.

The rotating plate is connected to the rod via a nut and sheet metal system shown in Figure 3. The nut is welded to the sheet metal, which is bolted into the rotating plate with 10-24 $\frac{3}{4}$ inch, counter-bored bolts. A second nut-plate piece is bolted into the middle plate. The nut and threaded rod are used as gears to make the plate move.

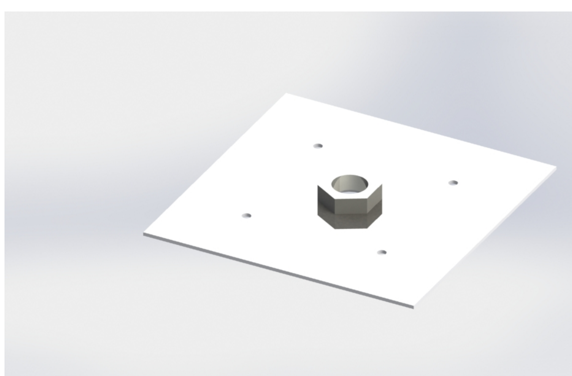


Figure 3: The sheet metal plate used to connect the nuts to the stationary center plate and the rotating circular plate.

Currently, the IR distance sensor is taped onto the plate sticking straight up. The wires are taped to it as well, to prevent them from getting in the way of the rotating plate.

2.4.2 Power

The system is driven by a 400-step, 3.06V stepper motor. It can rotate as little as 0.9 degrees at a time. We power this via a wall plug. The arduino as powered through a laptop. This subsystem is discussed more thoroughly in the Circuit Design section.

2.4.3 Transmission

The transmission is essentially a system of gears. We have two spur gears, each with an outer diameter of 2.5 inches and a pitch diameter of 2.4 inches. One is 6 inches tall, and is connected to the motor, and the other is $\frac{1}{4}$ inches tall, and screwed on to the threaded rod. They are identical except for their heights. This means that the rotating platform will move the same angle as the motor does.

The threaded rod acts as a worm gear. Rotating it through the nut also causes vertical motion by virtue of the threaded rod screwing into it. This means that we collect a spiral of points around the outside of the object, with a height difference of $\frac{1}{13}$ inch per revolution.

Results

Sensor and Initial Calibration

We purchased a Sharp GP2Y0A41SK0F Analog Distance Sensor for range 4-30 cm. In order to interpret its results, we calibrated the sensor by taking a number of measurements with known distances. The resulting curve is shown in Figure 4.

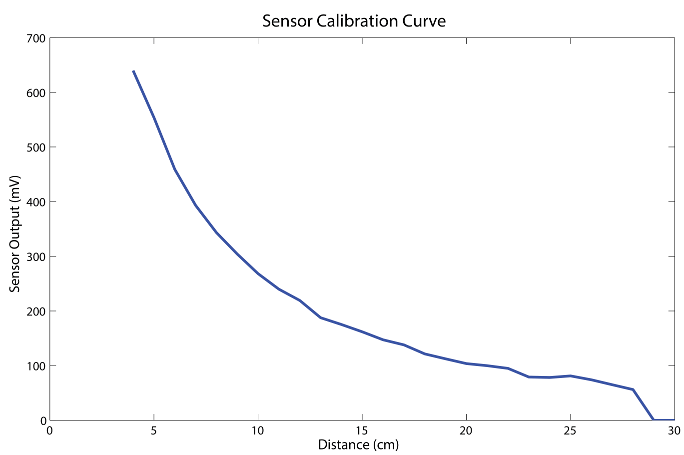


Figure 4: The raw distance vs. output curve for the sensor

We linearized the line using the equation provided on the sensor datasheet: $\frac{1}{L+0.42}$ and used MATLAB's linear fit function to create a line to measure distance as a function of the voltage sensor reading. We achieved the following result, where V is voltage measured by the sensor and D is distance to the object in cm:

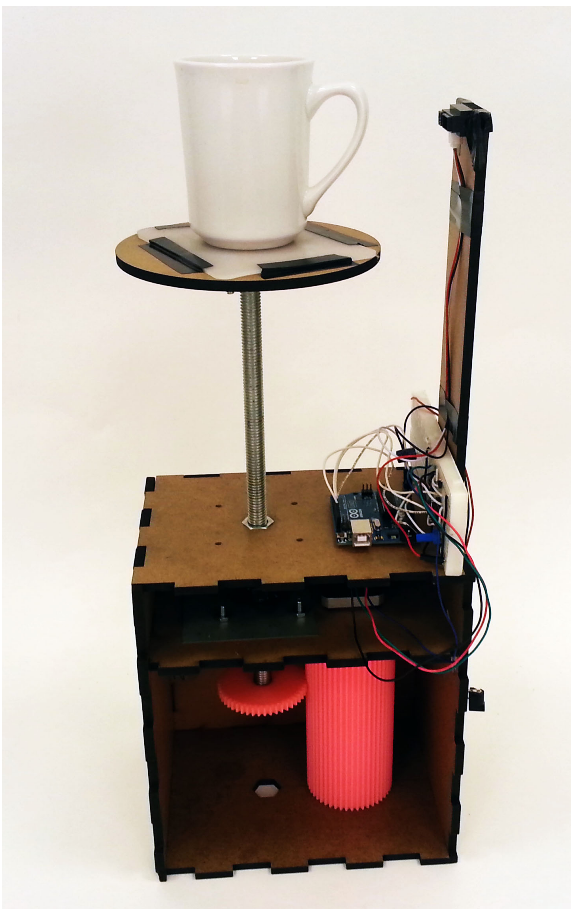
$$D = \frac{1}{\frac{V+53.567}{3313.3}} - 0.42$$

When performing measurements for scanning, we used the equation:

$$D = 12.24 - \frac{1}{\frac{V+53.567}{3313.3}} - 0.42$$

With the added term of 12.24. This gave us not the distance from the sensor to the object but instead the distance from the center of the measuring plate to the object. By having a "radius" term we were able to convert from polar to cartesian coordinates and use that to construct our 3D scans.

Key Results



The scanner works as expected. We went through a few rough patches in construction, in making sure that the motor and gears were turning with the same ratios without slipping. The friction of the threaded rod also affected performance. Remedying these obstacles, we took a scan of a mug from the dining hall. The resulting digital scan constructed in MATLAB is shown in Figure 5.

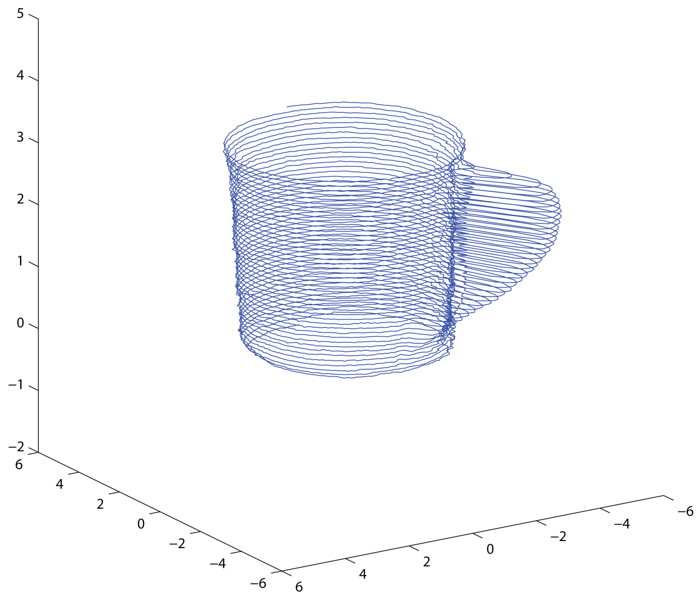


Figure 5: Scan of our cup

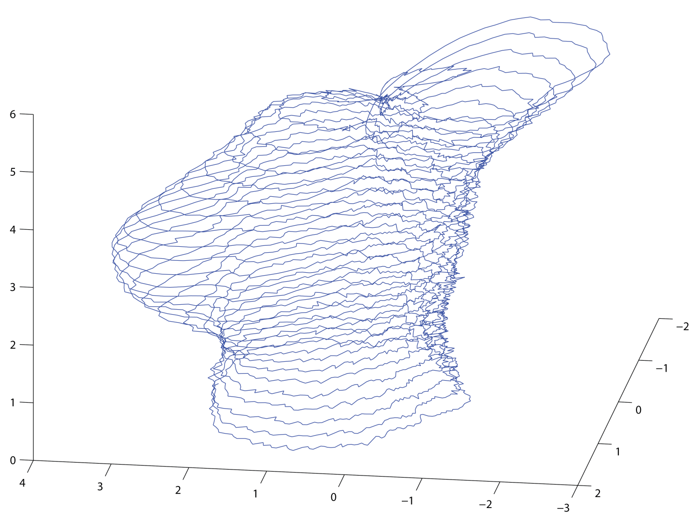


Figure 6: Scan of a chocolate bunny head

Limitations

There are a few limitations associated with the infrared sensor and the scanner that should be acknowledged. First, given that it is an infrared sensor, it will not return an accurate measurement on a translucent or mirrored surface. In addition, it can only scan solid objects. Objects with holes will cause odd discontinuities in the scan because the sensor returns near-infinite data. The best scans come from monocolor opaque objects.

One improvement we could make upon this in the future is faster measuring time. Currently, it takes about 45 minutes to scan a 4-inch tall object.

Acknowledgements

We want thank the teaching staff of Real World Measurements for being excellent mentors and professors. Also, we would like to thank the laser cutting staff for their incredible turn around time.

Works Cited

References

[1] "Arduino - MotorKnob." Arduino - MotorKnob. Arduino, 2014.Web. 10 Apr. 2014.

[2] GP2Y0A41SK0F. N.p.: Sharp, n.d. PDF

[3] Hybrid Stepping Motor Mod. N.p.: Karlson Robotics, n.d. PDF

[4] SN754410 Quadruple Half H-Driver. Dallas: Texas Instruments, n.d. PDF.