Structured Light 3D Scanner
Final Project - Report

Usama Munir Sheikh, Yaser Khalil

ECSE 6650: Computer Vision with Dr. Qiang Ji, RPI

Troy, New York

sheiku@rpi.edu

***Abstract/Goals*— The goal of this project was to use the concepts learned in class to develop a Structured Light 3D Scanner with one Laser line-light and One Camera. Concepts from class that were used in this project include Perspective Projection, Coordinate Frame Transformations, Camera Calibration (Using an External Source), Active Stereo 3D Reconstruction and DoG and LoG for Edge Detection**

Keywords- Structured Light; Active 3D Reconstruction; DoG & LoG Edge Detection; Perspective Projection; Camera Calibration

I. Introduction & Related Work

3D reconstruction is a very important topic in computer vision. It has many applications, ranging from robotics to human computer interaction to exploration to 3D scanning etc.

It is the part of computer vision that allows us to recover the 3D properties of an object or a scene from its 2D images. One can recover many different types of properties of a 3D object including its shape, size and depth.

There are many approaches one can take to 3D reconstruct and object or a scene. Some of these approaches are passive and some active. Passive approaches include techniques such as "Shape from X" that use one image of the object for reconstruction or Stereopsis that uses multiple images of an object for 3D reconstruction. Active approaches of 3D reconstruction include structured light techniques, active stereo, that include that include at least one camera and a projector. You can also use range sensors such as RADAR or LIDAR for active 3D reconstruction.[1]

For this project we have implemented one of the structured light, active, techniques for 3D reconstruction. We shall use a (red) laser line light and a web camera to reconstruct various 3D objects. We will call this 3D scanning. In particular, we shall use two types of reconstruction processes here. The first type of 3D Scanning will move the laser light and shine it on a stationary object from different angles to reconstruct the part of the object that is in the view of the camera. This is, in fact, called a 2.5 D scan since it doesn't scan the object from all angles. The second type of 3D scanning we do in this project involves rotating the object while keeping the laser light stationary and passing through the center of the axis of rotation. A detail look on our final setup can be found in Section III.

Structured light 3D scanning is not that uncommon in the field of computer vision. Our own setup is inspired by and based upon techniques highlighted in Taubin's [2] course for SIGGRAPH 2009. [2] This course was titled, "Build Your Own 3D Scanner: 3D Photography for Beginners" and its documentation on his webpage [2] presents techniques based on many different ideas on 3D reconstruction from papers in the past. One of these methods is called 3D Scanning with Swept Planes where they use the shadow of a stick to 3D reconstruct an object. Our setup is very similar to this technique that was originally proposed by Bouget and Perona. [3]

For a rotating object we have used the same technique and applied it in a different way. Here, we shall use the angular speed of the rotating platform and the frames-per-second count of the video camera to 3D reconstruct an object. Our results for this method work better with symmetrical objects where a little drift in the angular velocity does not add over-lap in our 3D reconstructed object. There are other methods to get the angular speed this however, ones that extract the speed directly from the frames of the video sequence. One such method is, again, found in one of Taubin's courses, this time for SIGGRAPH 2014. [4] It is called the "Laser Slit 3D Scanner" and it 'calibrates' the turn table using CALTag checkerboard patterns that allow them to get the angle/position of the turntable using the 2D images they acquired. A CALTag checkerboard pattern, shown in figure 1, has a unique code assigned to each cell of the pattern that allows them to calibrate the turntable even when the pattern is partially occluded due to the presence of an object on top of the turntable.

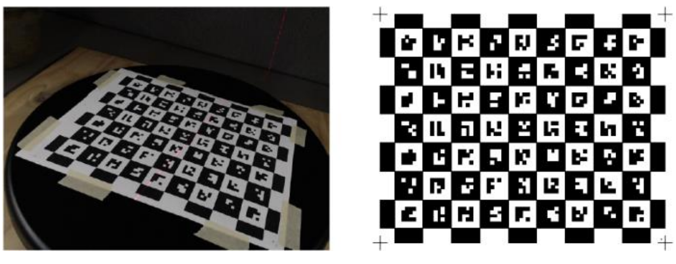


Figure 1. CALTag Checkerboard Pattern (Image from [4])

II. Theory and Methodology

A. *Full Perspective Projection*

In 3D reconstruction, we are interested in the 3D coordinates of a point $P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$. A full perspective projection model, which relates the 3D spatial points to 2D camera points is used. The full perspective projection model is given as

$$\lambda \begin{bmatrix} c \\ r \\ 1 \end{bmatrix} = W[R\ T] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1}$$

where (c, r) are the column and row coordinates of the point in the image, λ is the depth of the object. It is the same as z coordinate of the 3D spatial points. W is the intrinsic camera matrix and R and T are the rotaion and translation matrices of the global frame w.r.t. the camera frame. The intrinsic matrix is defined as

$$W = \begin{bmatrix} fs_x & 0 & c_0 \\ 0 & fs_y & r_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where f is the focal length, s_x, s_y are the sampling frequencies in the x and y direction respectively, (c_0, r_0) are the column and row origins of the camera frame or the principle point. To simplify the computation of the position, the global/reference frame is taken to be the camera frame. This results in the following affine transformation

$$R = I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

And

$$T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Then equation 1 simplifies to

$$z \begin{bmatrix} c \\ r \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & 0 & c_0 \\ 0 & fs_y & r_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3}$$

Expanding the equation above yields

$$z\,c = fs_x\,x + c_0\,z \tag{4.1}$$
$$z\,r = fs_y\,y + r_0\,z \tag{4.2}$$
$$z = z \tag{4.3}$$

Rearranging equation 4 to solve for x, y, and z yields

$$x = z\frac{(c - c_0)}{fs_x} \tag{5.1}$$
$$y = z\frac{(r - r_0)}{fs_y} \tag{5.2}$$

Examining equation 5, it can be noticed that there are three unknowns (x, y, z) with two equations. Therefore, an additional equation is required to completely solve for the position of each 3D point in space. This additional equation can be extracted from the intersection of the laser plane with the object.

When a laser line is projected on an object, it intersects with the object. This results in a (curved) 2D image line that is dependent on the shape of the object. Now, let's assume that the laser plane equation is given as

$$A_l x + B_l y + C_l z + D_l = 0 \tag{6}$$

where A_l, B_l, C_l are the components of the normal vector to the plane and they are known. Also assume that D is known. Combining equations 5 and 6 yields

$$x = z\frac{(c - c_0)}{fs_x} \tag{7.1}$$
$$y = z\frac{(r - r_0)}{fs_y} \tag{7.2}$$
$$A_l x + B_l y + C_l z + D_l = 0 \tag{7.3}$$

It can be noticed that equations 7 are a set of linear equations for the unknowns, (x, y, z). It can be rearranged in the following form to solve for (x, y, z)

$$Lp = M \tag{8}$$

where

$$L = \begin{bmatrix} 1 & 0 & -\dfrac{(c - c_0)}{fs_x} \\ 0 & 1 & -\dfrac{(r - r_0)}{fy} \\ A_l & B_l & C_l \end{bmatrix}$$

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

And

$$M = \begin{bmatrix} 0 \\ 0 \\ -D \end{bmatrix}$$

Now the position $p = [x, y, z]^T$ can be solved for

$$P = L^{-1}M \tag{9}$$

B. Camera Calibration Toolbox for Camera Intrinsic Paramaters

To solve for the 3D position as described earlier, we need the intrinsic parameters of the camera. To get the intrinsic parameters of the camera we have used the 'Camera Calibration Toolbox for MATLAB' from Caltech. [8] The toolbox has a really nice GUI that lets us select checkerboard calibration images from our camera and takes us step by step to compute the intrinsic parameters of the camera and the respective re-projection error. If the re-projection error is too large, the toolbox allows you to analyze and find the guilty images. We can then suppress these images and re-compute the intrinsic matrix to lower the re-projection error. The toolbox also has other options that were not of relevance to our project. We did find out from the professor that the underlying principal of the camera calibration toolbox, using checkerboard images, is homography. We have used 20 checkerboard images from different positions and angles for calibration (shown in figure 2). The intrinsic matrix we got from the toolbox is the following:

$$W = \begin{bmatrix} 1037 & 0 & 642 \\ 0 & 1046 & 333 \\ 0 & 0 & 1 \end{bmatrix}$$

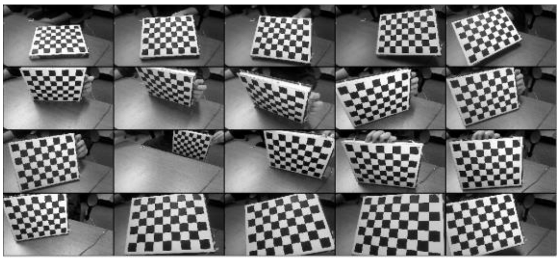


Figure 2. Images for Camera Calibration

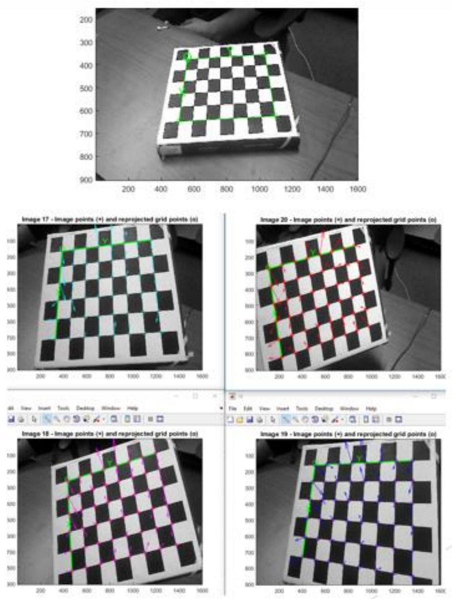


Figure 3. Intermediate steps in calibration showiing (top) region selection by the user and (bottom) Reprojected Points

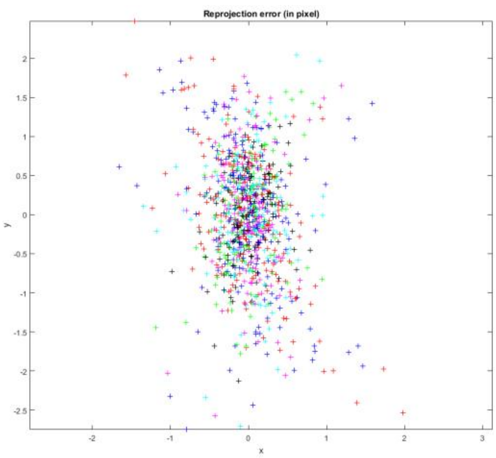


Figure 4. Reprojection Error

C. *Laser Plane*

In order to solve for the position, the laser plane equation must also be known. To compute the normal vector in the laser plane equation, $N = 4$ points were used to solve for the equation of the plane. The procedure for the determination of the laser plane is as follows

1- We have to setup two known planes (assuming plane equations are known)
2- Project the laser line on these planes
3- Pick two points in the image from the intersection of the laser line with first plane and another two points in the image from the intersection of the laser line with the second plane
4- Use equation 9 to solve for the 3D position of each point
5- Solve for the laser plane using these four points
6- 3D reconstruct the object using equations 9

Each of the steps is explained in details in the following sections

D. *Board Planes Setup*

As described earlier, to solve for the laser line equation, two known planes are required. The planes are setup as shown in figure 5 below.

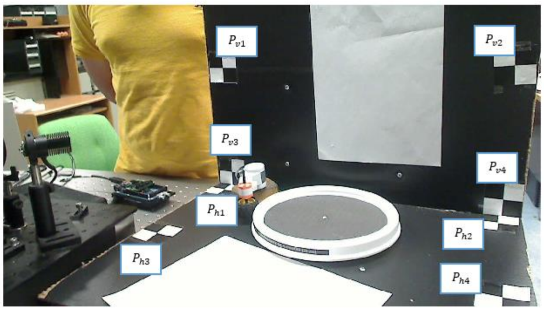


Figure 5. Checkerboard Points on the Board Planes

The planes are somewhat perpendicular. However, we still need to compute their equations with respect to the camera frame. In order to do so, four points on each plane are marked. A rectangle was created with a length of 500 mm which represents the distance between P_1 and P_2 in figure 5. Using equation 5, the position of any point P_i is given by

$$P_i = \begin{bmatrix} z_i \dfrac{(c_i - c_0)}{f s_x} \\ z_i \dfrac{(r_i - r_0)}{f s_y} \\ z_i \end{bmatrix} \; ; \; i \in (1,2,3,4)$$

Then, the vector that represents the difference between P_1 & P_2

$$P_{12} = P_2 - P_1$$

$$P_{12} = \begin{bmatrix} z_2 \frac{(c_2 - c_0)}{f s_x} \\ z_2 \frac{(r_2 - r_0)}{f s_y} \\ z_2 \end{bmatrix} - \begin{bmatrix} z_1 \frac{(c_1 - c_0)}{f s_x} \\ z_1 \frac{(r_1 - r_0)}{f s_y} \\ z_1 \end{bmatrix} \qquad (10)$$

Similarly the difference between P_3 & P_4

$$P_{34} = \begin{bmatrix} z_4 \frac{(c_4 - c_0)}{fs_x} \\ z_4 \frac{(r_4 - r_0)}{fs_y} \\ z_4 \end{bmatrix} - \begin{bmatrix} z_3 \frac{(c_3 - c_0)}{fs_x} \\ z_3 \frac{(r_3 - r_0)}{fs_y} \\ z_3 \end{bmatrix} \tag{11}$$

We also know that the magnitudes of P_{12} & P_{34} are equal since the points form a rectangle this yields to

$$P_{12} = P_{34}$$
$$\begin{bmatrix} z_2 \frac{(c_2 - c_0)}{fs_x} \\ z_2 \frac{(r_2 - r_0)}{fs_y} \\ z_2 \end{bmatrix} - \begin{bmatrix} z_1 \frac{(c_1 - c_0)}{fs_x} \\ z_1 \frac{(r_1 - r_0)}{fs_y} \\ z_1 \end{bmatrix} = \begin{bmatrix} z_4 \frac{(c_4 - c_0)}{fs_x} \\ z_4 \frac{(r_4 - r_0)}{fs_y} \\ z_4 \end{bmatrix} - \begin{bmatrix} z_3 \frac{(c_3 - c_0)}{fs_x} \\ z_3 \frac{(r_3 - r_0)}{fs_y} \\ z_3 \end{bmatrix} \tag{12}$$

it can be noticed that there are three equations with four unknowns. These equations are linear, then they can be rearranged in the following linear form

$$Ax = B \tag{13}$$

where x represents the unknowns z_i. After rearranging and dividing the equations above by z_4. The resulting form is given by

$$\begin{bmatrix} \frac{(c_0 - c_1)}{fs_x} & -\frac{(c_0 - c_2)}{fs_x} & -\frac{(c_0 - c_3)}{fs_x} \\ \frac{(r_0 - r_1)}{fs_y} & -\frac{(r_0 - r_2)}{fs_y} & -\frac{(r_0 - r_3)}{fs_y} \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \frac{z_1}{z_4} \\ \frac{z_2}{z_4} \\ \frac{z_3}{z_4} \end{bmatrix} = \begin{bmatrix} -\frac{c_0 - c_4}{fs_x} \\ -\frac{r_0 - r_4}{fs_y} \\ -1 \end{bmatrix} \tag{14}$$

This is a linear problem with

$$A = \begin{bmatrix} \frac{(c_0 - c_1)}{fs_x} & -\frac{(c_0 - c_2)}{fs_x} & -\frac{(c_0 - c_3)}{fs_x} \\ \frac{(r_0 - r_1)}{fs_y} & -\frac{(r_0 - r_2)}{fs_y} & -\frac{(r_0 - r_3)}{fs_y} \\ 1 & -1 & -1 \end{bmatrix}$$

And

$$x = \begin{bmatrix} \frac{z_1}{z_4} \\ \frac{z_2}{z_4} \\ \frac{z_3}{z_4} \end{bmatrix}$$

And

$$B = \begin{bmatrix} -\frac{c_0 - c_4}{fs_x} \\ -\frac{r_0 - r_4}{fs_y} \\ -1 \end{bmatrix}$$

the vector x is solved for easily by

$$x = A^{-1}B$$

This vector x has four unknowns (z_1, z_2, z_3, z_4). These depths can be written as

$$z_1 = x(1)z_4 \ , \ z_2 = x(2)z_4 \quad , \ z_3 = x(3)z_4 \tag{15}$$

Then, an additional equation is required to solve for all depths. This equation comes from the fact that the distance between two points is measured. So the distance between P1 and P2 is given by

$$P_{12} = \begin{bmatrix} z_2 \frac{(c_2 - c_0)}{fs_x} \\ z_2 \frac{(r_2 - r_0)}{fs_y} \\ z_2 \end{bmatrix} - \begin{bmatrix} z_1 \frac{(c_1 - c_0)}{fs_x} \\ z_1 \frac{(r_1 - r_0)}{fs_y} \\ z_1 \end{bmatrix} \tag{16}$$

substitute for z_1 & z_2 from equation 15 into equation 16 . This will result in P_{12} being a function of z_4

$$P_{12} = \begin{bmatrix} x(2)z_4 \frac{(c_2 - c_0)}{fs_x} \\ x(2)z_4 \frac{(r_2 - r_0)}{fs_y} \\ x(2)z_4 \end{bmatrix} - \begin{bmatrix} x(1)z_4 \frac{(c_1 - c_0)}{fs_x} \\ x(1)z_4 \frac{(r_1 - r_0)}{fs_y} \\ x(1)z_4 \end{bmatrix} \tag{17}$$

then the norm of this vector is given by

$$P_{12} = \sqrt{(P_{12}^T \, P_{12})} \tag{18}$$

And it is also the distance P_{12} which is the length of the rectangle

$$P_{12} = Rectangle\ Length \tag{19}$$

Finally, we have one equation in terms of z_4 given by

$$\sqrt{(P_{12}^T \, P_{12})} = Rectangle\ Length \tag{20}$$

Once z_4 is solved for, then z_1, z_2, and z_3 can also be solved for using the value of z_4 and substituting it back into equation 15

$$z_1 = x(1)z_4 \ , \quad z_2 = x(2)z_4 \quad , \ z_3 = x(3)z_4$$

Now that all depths are known, then the position of each point in the rectangle edges is solved for using equation 5, repeated here for convenience

$$x_i = z_i \frac{(c_i - c_0)}{fs_x} \tag{5.1}$$
$$y_i = z_i \frac{(r_i - r_0)}{fs_y} \tag{5.2}$$

Once all the points P_i are known, then the equation of the plane is obtained using these points. The normal of the plane $[A, B, C]$ is solved for using a cross product between two non-parallel vectors in a plane

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = P_{12} X P_{13} \tag{21}$$

Then, any point can be used and substituted in the plane equation to solve for D

$$D = -[A\ B\ C] \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \tag{22}$$

this completes the derivation for the vertical and horizontal cardboard planes.

E. *Laser Line Plane Equation*

Now that the board plane equations are solved for. The next step is to use these two planes to determine the laser plane equation. The procedure for the determination of the laser plane equation is as follows. First, a laser line is projected on the vertical and horizontal planes. Next, we pick two points from the intersection of the laser plane with the vertical plane in the 2D image, and another two points from the intersection of the laser plane with the horizontal plane and solver for their 3D coordinates using the board plane equations. Let's assume that the vertical and horizontal plane equations were solved for from previous step and they are represented as follows,

$$\begin{aligned} A_v x + B_v y + C_v z + D_v &= 0 \\ A_h x + B_h y + C_h z + D_h &= 0 \end{aligned} \tag{23}$$

where the subscripts v, h denote vertical and horizontal respectively. If the 2D image laser points taken on the vertical plane are $(c, r)_{v1}$ & $(c, r)_{v2}$, then, there 3D spatial position is given by

$$P_{v1} = \begin{bmatrix} z_{v1} \dfrac{(c_{v1} - c_0)}{f s_x} \\ z_{v1} \dfrac{(r_{v1} - r_0)}{f s_y} \\ z_{v1} \end{bmatrix}$$
$$P_{v2} = \begin{bmatrix} z_{v2} \dfrac{(c_{v2} - c_0)}{f s_x} \\ z_{v2} \dfrac{(r_{v2} - r_0)}{f s_y} \\ z_{v2} \end{bmatrix}$$

substituting these two points in the vertical plane yields

$$\begin{aligned} A_v z_{v1} \frac{(c_{v1} - c_0)}{f s_x} + B_v z_{v1} \frac{(r_{v1} - r_0)}{f s_y} + C_v z_{v1} + D_v &= 0 \\ A_v z_{v2} \frac{(c_{v2} - c_0)}{f s_x} + B_v z_{v2} \frac{(r_{v2} - r_0)}{f s_y} + C_v z_{v2} + D_v &= 0 \end{aligned} \tag{24}$$

using the set of equations 24 we can solve for z_{v1} and z_{v2}. These z values can then be used to find P_{v1} and P_{v2}. The other two laser points, on the horizontal plane intersection points, P_{h1} and P_{h2}, are solved for in a similar way. These new 3D points are also the part of the projected laser plane. As discussed earlier, the laser plane is determined using these four points and equations 21 & 22

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = P_{12} \times P_{13} \qquad\qquad D = -[A\ B\ C] \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

F. *3D Reconstruction*

Finally, the equation of the laser plane is computed. Now we can use equation set 9 for each point on the laser line. Let's take a point (c_i, r_i) lying on the object and laser line, then

$$L = \begin{bmatrix} 1 & 0 & -\dfrac{(c_i - c_0)}{f s_x} \\ 0 & 1 & -\dfrac{(r_i - r_0)}{f y} \\ A_l & B_l & C_l \end{bmatrix}; \; p_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \; ; \; M_i = \begin{bmatrix} 0 \\ 0 \\ -D_l \end{bmatrix}$$

then $p = [x, y, z]^T$ is solve for using

$$p_i = L_i^{-1} M_i$$

This is repeated for all desired points on the laser line to be reconstructed. This completes the 3D reconstruction using the laser line.

G. *Edge Detection*

Up to this point the discussion assumes that the points and the laser line are detected. One way to determine the laser line in an image is to manually select each camera 2D point lying on the laser line. This is a tedious and undesired process. Another way, is to use edge detection to determine the laser line points. There are many methods for edge detection and there are many tools available that does edge detection. However, we chose to implement edge detection ourselves using [1]. A canny enhancer was implemented with difference of Gaussian (DoG) and Laplacian of Gaussian (LoG) for edge detection. Edge detection process is described by three steps [1]

1- Edge Enhancement : Designing a filter and convolving it with the images
2- Non-Maximal Suppression : This step is used for removing non-locally maximum edges
3- Edge Thresholding : This step is used for identifying pixels using image gradients

First order derivative of the image intensity is required for a step edge. The gradient of the intensity is defined as

$$\nabla I(c, r) = \begin{bmatrix} \frac{\partial I}{\partial c} \\ \frac{\partial I}{\partial r} \end{bmatrix} \tag{25}$$

detailed information about Canny edge enhancer are available in [1]. However, the Canny edge enhancer is given by

$$h = \nabla G(c, r) \quad (26)$$

where we can obtain a 3x3 DoG w.r.t the vertical and horizontal coordinates (x, y) as,

$$H_x = \begin{bmatrix} 0.0764 & 0.3062 & 0.0764 \\ 0 & 0 & 0 \\ -0.0764 & -0.3062 & -0.0764 \end{bmatrix}$$

$$H_y = \begin{bmatrix} 0.0764 & 0 & -0.0764 \\ 0.3062 & 0 & -0.3062 \\ 0.0764 & 0 & -0.0764 \end{bmatrix}$$

horizontal mask H_x is used to compute the gradient in the horizontal direction G_x and the vertical mask H_y is used to compute the gradient in the vertical direction G_y. Then the gradient magnitude for each pixel is easily determined.

The second order derivative of the intensity can also be used to detect a step edge. For a step edge, the second order derivative has a zero crossing when the first order derivative is maximum. The second order derivative of the intensity which is also known as the laplacian is given by

$$\nabla^2 I = \frac{\partial I}{\partial^2 c} + \frac{\partial I}{\partial^2 r} \quad (27)$$

The resulting laplacian image filter is given by,

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

this filter is applied to all pixel in the image to find the LoG. Once LoG and DoG are determine for pixels, then we can follow the procedure below [1]

- Convolve an image with LoG
- Non-Maximum Suppression : Find zero-crossing pixel using a threshold (ddI)
  - If a pixel $LoG < -ddI$ and the neighboring pixels > ddI
- Thresholding : Apply a threshold on the gradient of the zero-crossing pixels to identify the edge pixels
  - If a zero crossing pixel is has a gradient magnitude greater than the threshold, then define it as an edge pixel

Note that edge detection does not have to be applied for all pixels. If the region in an image where the edge might exist is known, then different search regions can be defined to search for edges. This will reduce the computation power from searching a whole image to searching small regions. This completes the edge detection process. Great details for edge detection using this method and other methods are found in [1].

H. *3D Reconstruction with a Rotating Platform and Stationary Laser Line*

For a rotating platform with a stationary laser plane. The following steps describe the procedure for reconstruction of the object

- Determine the laser plane equation once at the beginning using four points as described earlier.
- Shoot a video and decompose it into a sequence of images.
- 3D reconstruct the object in each frame with respect to the camera frame.
- Measure the angular rotation of the platform (turntable) for each frame.
- Compute the transformation from the global frame to the rotating body frame
- Determine the position of points in the body frame and plot the position after each rotation

the first three steps are straightforward and they are described earlier. Steps 3 through 6 are described below.

Measurement of angular rotation of the platform

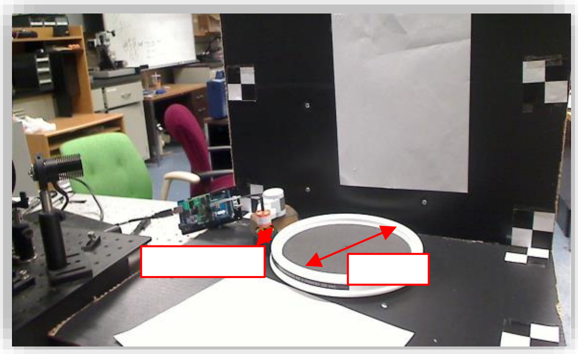


Figure 6. Segway motor setup with the turn table

A NXT Segway motor [6] was used as our driving mechanism to rotate the turntable. The Segway is embedded with an encoder to measure the angular displacement of the Segway wheel. Figure 6 shows the setup that was used.

The angular velocity of the Segway is determined using the encoders. However, we are interested in the angular speed of the table. The equation the relates the angular speed of the Segway to the angular speed of the table is given by

$$\omega_{table} = \frac{d_{wheel}}{d_{table}} \omega_{wheel} \quad (28)$$

where d is the diameter, and ω is the angular speed. The diameter of the wheel was measured to be $41.85mm$ and the diameter of the turntable was measured to be $223.85mm$. Assuming the speed of the Segway is constant. Then the angular displacement of the turntable for each frame is computed using the speed of the video [fps]

$$\frac{\delta\theta}{\delta frame} = \frac{\omega_{table}}{fps} \quad \left[\frac{rad}{frame}\right] \tag{29}$$

Next, the angular displacement with respect to the initial frame is measured using the frame number,

$$\theta = \frac{\delta\theta}{\delta frame} N_{frame} \tag{30}$$

Note that this rotation is about the z-axis in the global\world fixed frame which is centered at the center of the turntable. So we need to find the transformation matrix between the camera frame and global frame.

Transformation Matrix between the Camera and Global Frame

There many ways to compute the rotation between the camera frame and the world frame. One can use MATLAB calibration toolbox to do so. However, it can be computed in the following way as well. The global frame is assumed to be at the center of the turntable which is parallel to the horizontal cardboard (horizontal plane). Then the rotation between the camera frame and the horizontal is determined using the following

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = R_{wc} \begin{bmatrix} A_h \\ B_h \\ C_h \end{bmatrix} \tag{31}$$

where $\begin{bmatrix} A_h \\ B_h \\ C_h \end{bmatrix}$ are the normal of the horizontal plane. This yields

$$r_3 = \frac{[A_h \quad B_h \quad C_h]}{\sqrt{A_h^2+B_h^2+C_h^2}} \tag{32}$$

$$r_2 = \frac{[B_h \quad -A_h \quad 0]}{\sqrt{A_h^2+B_h^2}} \tag{33}$$

$$r_1 = r_2 \times r_3 \tag{34}$$

where r_i are the rows of the rotation matrix R_{wc}.

Next, we compute the translation between the camera and the global frame. This translation is computed by first locating the center of the turntable (c, r) and using 3D reconstruction process described earlier. The translation vector is computed in the camera frame and given as

$$T_{cw} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{35}$$

Now that the rotation and translation between the global and the camera frames are known, then we can determine the transformation matrix H as

$$H_{cw} = \begin{bmatrix} R_{wc}^T & T_{cw} \\ \bar{0} & 1 \end{bmatrix} \tag{36}$$

Computing the Position in the Object frame

At this point, the transformation matrix is known. Now, the position of each pixel on the laser line is reconstructed. In other words, we can reconstruct the object in the camera frame. Then, it is transformed to the global frame using this matrix. Let's assume that the 3D position of a point in the camera frame is given as P_c. Then, the position in the world frame P_w is

$$\begin{bmatrix} P_w \\ 1 \end{bmatrix} = H_{cw}^{-1} \begin{bmatrix} P_c \\ 1 \end{bmatrix} \tag{37}$$

However, the body frame which is fixed to the center of the turntable which is rotating about the z-axis of the global frame. The position of the point in the body frame P_b is governed by

$$P_b = rotz(\theta)\, P_w \tag{38}$$

this position is then plotted for each frame as the turntable is rotating.

Furthermore, if the object geometry was to be viewed in the camera frame, then there is no need to use the transformation matrix H_{cw}. However, the rotation about the z-axis still needs to be transformed to the camera frame. This transformation is given as

$$R = R_{cw} * rot(z, \theta) * R_{cw}^T \tag{37}$$

where $rot(z, \theta)$ is the rotation about z-axis by angle θ. Once the position of the points on the object is determined, it is multiplied by the R matrix as described above to represent it in the camera frame. This completes the derivation and process for 3D reconstruction of an object on a rotating platform and stationary laser line

I. *3D Reconstruction with a stationary platform and moving laser line*

This is a little different from the rotating platform in the sense that the laser plane is changing for all frames. This means that the process of detecting 2 points in the vertical plane and 2 points in the horizontal plane is repeated for each frame. Once we have 4 points, then the laser plane equation is determined from previous sections. Finally, the laser plane equation is used with the projection equations to 3D reconstruct all the points on the laser line.

III. Experimental Procedure, Results & Discussion

A. *The Setup*

Our initial setup for the 3D scanning platform is shown in figure 7.

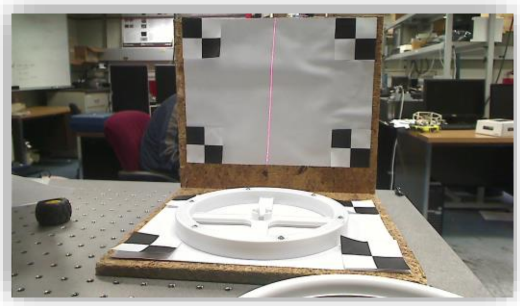


Figure 7. Version 1 of the Setup. We can see the four checkerboard points on the horizontal and vertical plane as well as the turn table at the center. If we look closely, we can also make out the laser line, passing through the center of the turntable's axis of rotation

Version 1 of the setup had a lot of problems associated with it which made it unfit for 3D reconstruction. Although we got some results for the resulting 3D point cloud of some objects, their shapes were not exact.

The reasons why this version of the setup failed were two-fold. The first reason was that the four checkerboard points on the horizontal board plane got occluded once the turn table was completely assembled. This meant that if the set up was moved or the platform was slightly disturbed, we would have to recalibrate the planes by first disassembling the turn-table to remove any occlusions and then getting the now visible checkerboard points.

The second reason this setup failed was because it was not robust. It was a shaky setup and any outside sources of disturbance would have us recalibrate the entire platform.

External sources of disturbances to the platform came in many forms. They could be any movements to the platform when it came in contact with people, while placing an object on top of it or disassembling the turn table. Other sources of disturbances came, when the person performing the experiment, tried to connect to the wires of the motor system that was used to rotate the platform or when the table, where the setup was placed on, moved itself.

Keeping the failures of setup 1 in mind, we built another, more robust, platform on top of an optical table. We added bigger boards for the horizontal and the vertical planes to make our laser and checker point detection easier and we firmly nailed-in the turn-table and the NXT Segway motor[6] to the horizontal board to make them more immune to sources of disturbances.

Figure 8. Version 2 of the setup. We can see the position of the camera and the boards in the top left photograph. At the top-left corner of the second (top-right) photograph we see the laser light. And finally, the bottom left image shows the view from the camera of the platform.

This new setup, version 2, shown in figure 8, had many advantages over the last setup. It had much broader planar boards which didn't occlude any of the checkerboard points and accommodated more of the laser light on the board planes to make its detection easier.

This version of the setup was quite robust, less prone to external disturbances and it offered better traction between the motor and the turn-table that gave us a smoother (almost constant) angular velocity for the turn-table.

Figures 9 and 10 show how the camera viewed the platform and how the laser plane passed through the center of the turntable's axis of rotation.

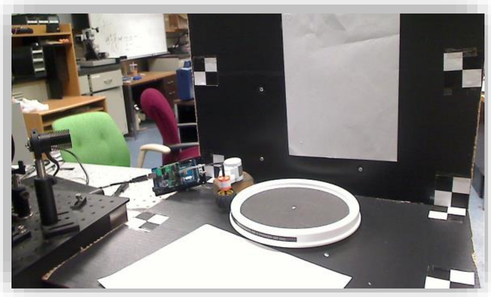
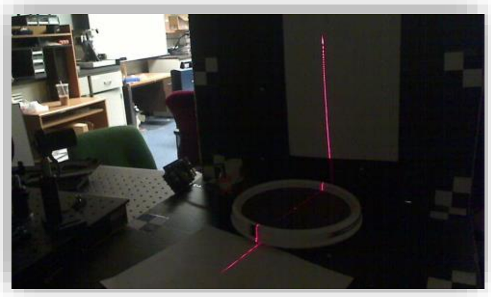


Figure 9. View of the Platform from the Camera



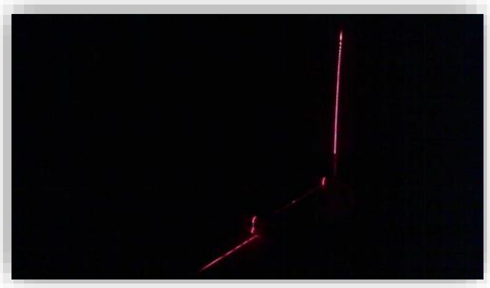


Figure 10. Views of the Camera with the lights dimmed and turned off respectively. We can see where the laser line is over the platform

B. *Experimental Procedure*

As mentioned earlier in the report, we did two different types of 3D scans. 1) Moving Light Scan – where the laser light moves and the object is stationary and the 2) Rotating Object Scan – where the object is rotating and the laser light is fixed, passing through the center of the turn-table's axis of rotation.

The results for these objects are provided ahead, along with images of the original object placed on the setup for a visual comparison. We have also made short videos for a few of the objects and links to those have been provided in the description of the results.

For the Moving Light Scans, we have shown results of two main objects, a small box and a polystyrene cup, and some other objects we found lying around like a ball or decorative ornaments etc. For the Rotating Object Scan, we have 3D reconstructed a polystyrene cup.

C. *Results – Moving Light Scan: BOX*

For our first successful experiment, we 3D reconstructed a box. It was a white carboard box that reflected the laser line well. The setup of the box is shown in figure 11.

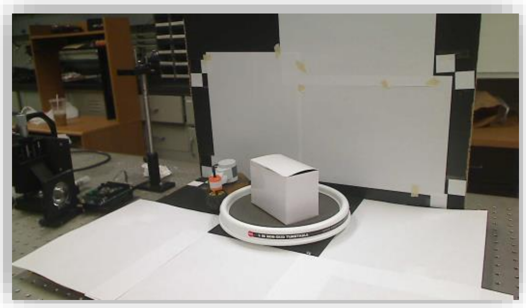


Figure 11. Setup for the Moving Light Box Scan

We got really got good results for the box setup that our visually appealing. To view the results better, we added false colors to the 3D point cloud we got from out from our algorithm before viewing it in meshlab. [5] The false colors show shading getting lighter as we move away from the camera. The results are shown in figure 12 and 13.

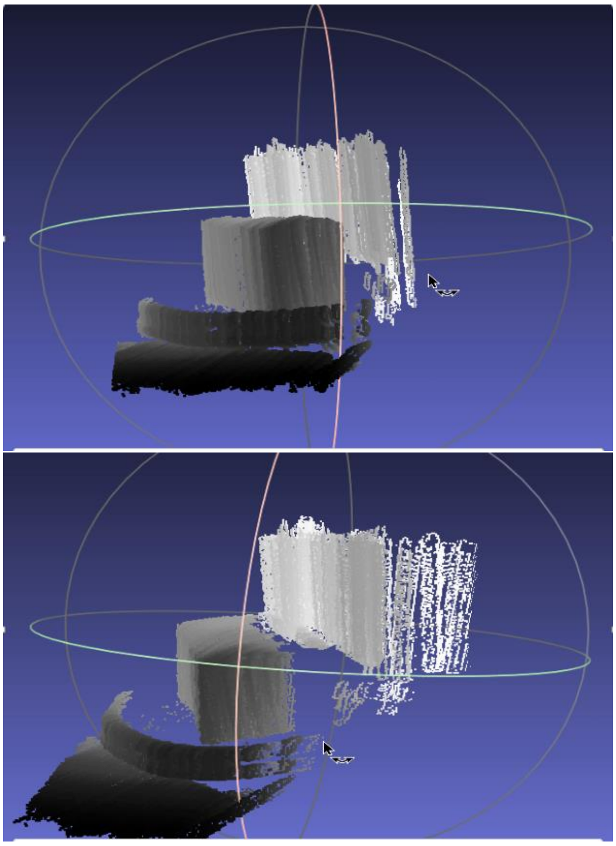


Figure 12. 2.5D Reconstructed Point Cloud of the Box as viewed in Meshlab from different angles. We also see the wall of the board behind the Box and the reconstruction of the turntable in front of it.

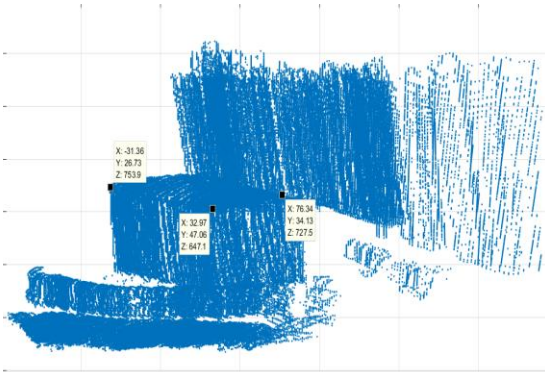


Figure 13. We can see the 3D coordinates of the box in this image. From here we can compute the width and the height of the box

From the 3D vertices of the reconstructed box we found that the length of the box is 125.89 mm, the width is 92.26 mm and the height is 81.65. The actual measurements of the box are length of 11.7 cm, width of 6.5 cm and a height of 8.2 cm. Therefore our results are pretty close within a margin of error. Also we see that the box top was a little curved when we used it on the platform. This can account for the height value of the width we got from the reconstruction result. Also, the box was a little shiny and this might have caused some errors in the reconstruction.

A short video of the meshlab preview of the box may provide greater perspective to the reader on the results. Therefore, you can find the video at the following link

https://www.dropbox.com/sh/8w246erg35tmp3y/AAAQgPuM95Nx_8H5KY1FZRHOa?dl=0

The results of the 2.5D reconstructed box are really good. We can see that the reconstructed box looks really well in the camera coordinates when viewed in meshlab and the reconstructed lengths are very close to the original dimensions of the box.

D. Results – Moving Light Scan: CUP

For our second successful experiment, we 3D reconstructed a polystyrene cup. It was a white cup that reflected the laser line pretty well. The setup of the cup is shown in figure 14

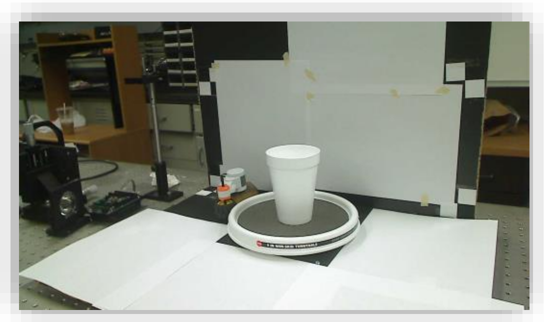


Figure 14. Setup for the Moving Light Cup Scan

We got really got good results for the cup setup as well. It is even more so visually appealing because we can see the actual curved opening of the cup really well. To view the results better, we again added false colors to the 3D point cloud we got from out from our algorithm before viewing it in meshlab. The results are shown in figure 15

A short video of the meshlab preview of the cup is given the following link

https://www.dropbox.com/sh/8w246erg35tmp3y/AAAQgPuM95Nx_8H5KY1FZRHOa?dl=0

For the cup, we get really great results that capture the curves of the cup perfectly. We also see more of the turn table in the reconstruction than we did in the box reconstruction. This is because the cup is a small object and doesn't occlude it as much within our search region.

For the cup, however, we do see one peculiar cluster of points (shown in the red circle) in our reconstruction that is not part of the scene. From the shading we can tell that it is nearer to the camera. This 'noise' is due to the edge detection algorithm that may have picked up some extra edges due the diffusion of the laser light onto other surfaces that shows up in the video image sequence.

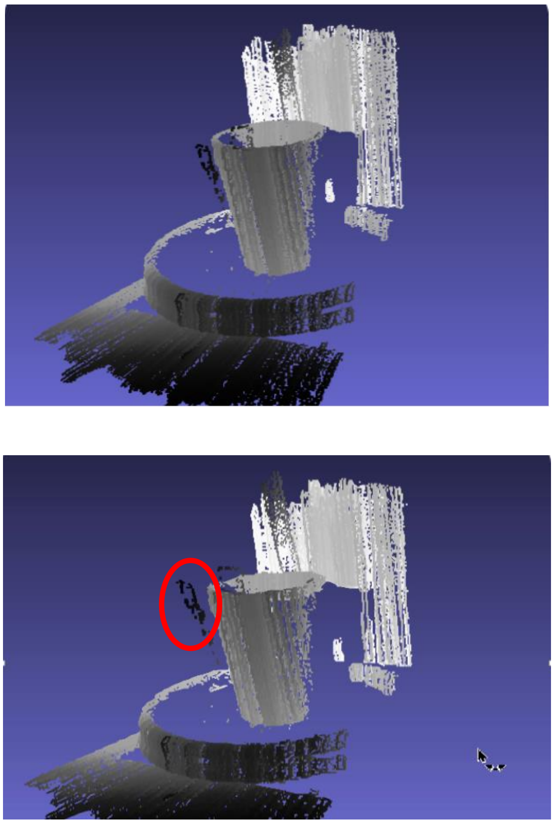
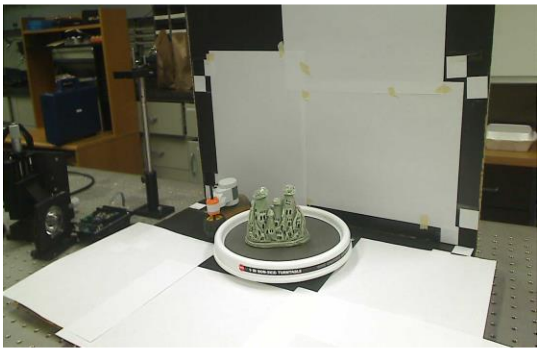


Figure 15. 2.5D Reconstructed Point Cloud of the Cup as viewed in Meshlab from different angles. In the second picture, we see some noisy points due to lower thresholds of the edge detection algorithm,

E. *Results – Other Scans*

We did a bunch of other scans to test our algorithm but not all objects gave good results. We have observed that objects that were darker in color absorbed most of the red laser light and didn't give any results.

Another thing we noticed was that shiny and glossy objects didn't provide us with as good results as the cup or the box. This is because they diffuse the light of the laser, therefore, making edge (laser line) detection noisier. Figure 16 and figure 17 show results of two other objects we scanned.



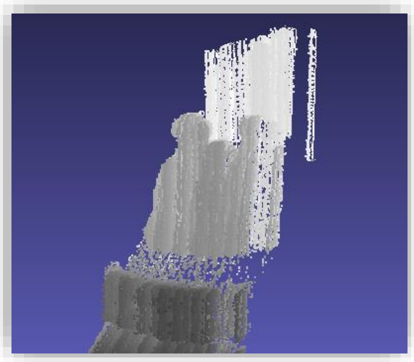


Figure 16. Results of a glossy house hold ornament we are naming 'underwatercity'. You can see that the points are diffused and it is really hard to gauge the details of the object from the point cloud

Figure 16 shows the results of the glossy house hold ornament we called 'underwater city.' We can see that the reconstructed point cloud lacks detail. However, it does get the general shape right.

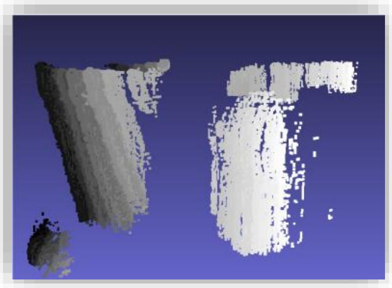


Figure 17. Results of the cup and a ball wrapped in white plastic bag. We can see the comparison between the two. The point cloud of ball has more holes in it.

Figure 17 shows the reconstruction results of a ball and a cup. The ball we had available at the moment was really dark in color so we wrapped a white plastic bag on it. The plastic bag had a really reflective and shiny surface which explains the holes in the reconstructed image. Although, we note that it is not as bad as the glossy underwater city ornament that was made of polished ceramic.

F. *Results – Rotating Object Scan: CUP*

For our final successful result, we did a 3D scan of a cup rotating on top of a turn table. For this experiment, we had to fix the laser light so it stayed stationary. We also made sure that it passed through the center of the axis of rotation of the turn-table.

We used the NXT Segway motor to rotate the turntable. We ran it through an Arduino MEGA 2560[7] board connected to our laptop. The Segway motor is a little jerky which is why we did not get an accurate readings of the angular velocity of the table but averaging the encoder readings overtime gave us a better estimate.

The setup for the Rotating Object Scan is shown in figure 18.

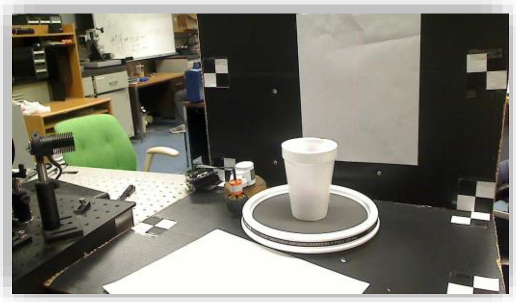


Figure 18. Setup for the Rotating Object Scan of the Cup

We ran the experiment for just over one rotation and then viewed the resulting 3D cloud point on meshlab. The results are shown in figure 19.

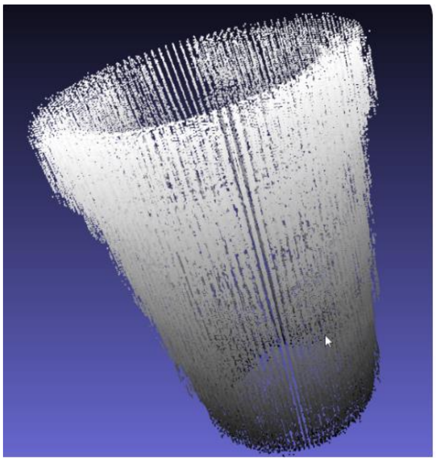


Figure 19. 3D Point Cloud of the Cup using Rotating Object Scan

From figure 19 we see that our 3D point cloud results look quite appealing visually. We also checked our results through actual measurements. We found that the diameter of the reconstructed object is ~8.6 cm while the actual cup diameter was 9 cm. Also the slant height of the cup of the reconstructed object was ~12.3 cm whereas the actual cup had a slant height of 12 cm.

Our results are therefore, quite accurate. However, due to the inaccurate angular velocity measurement we do see a drift in the resulting point cloud. We noticed that one actual rotation of the cup resulted in a cup that wasn't fully round: An actual 360 degree rotation gave around 300 degrees of rotation of the cup. Therefore, we had to take more than one rotation to get the full object. Thus we conclude that due to an error in getting an accurate measurement of the angular velocity, our algorithms well only with symmetric objects.

Finally we tried the Delaunay triangulation on the cup to reconstruct its surface. The results are present in a meshlab preview video on the following link. Also shown in figures 20 and 21.

https://www.dropbox.com/sh/8w246erg35tmp3y/AAAQgPuM95Nx_8H5KY1FZRHOa?dl=0

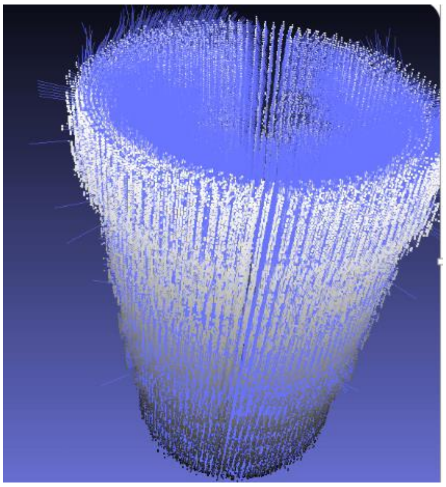


Figure 20. Recomputed Surface Nomals in Meshlab with the 3D point cloud

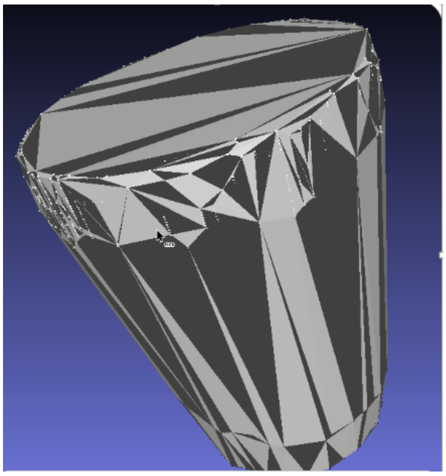


Figure 21. Delaunay Triangulation for Surface Reconstruction of the Cup from the 3D Point Cloud

IV. Future Works

The future works for this project include the following three things:

1) Try rotation scans of other objects. Try non symmetrical, and irregular objects.

2) Add color to the resulting 3D Point Cloud using the concepts learnt in class in the illumation model project.

3) Try the reverse problem: Rotate the laser line-light and keep the object/room stationary. Using this approach we can get the scan of a room, provided we can recognize planes and take some points on them.

We made an attempt on the third point to get the scan of the room but due to hardware limitations we were unable to pursue it further. Specifically, our camera was incapable of detecting the laser light after a specific distance. Figure 22 shows our setup for this.

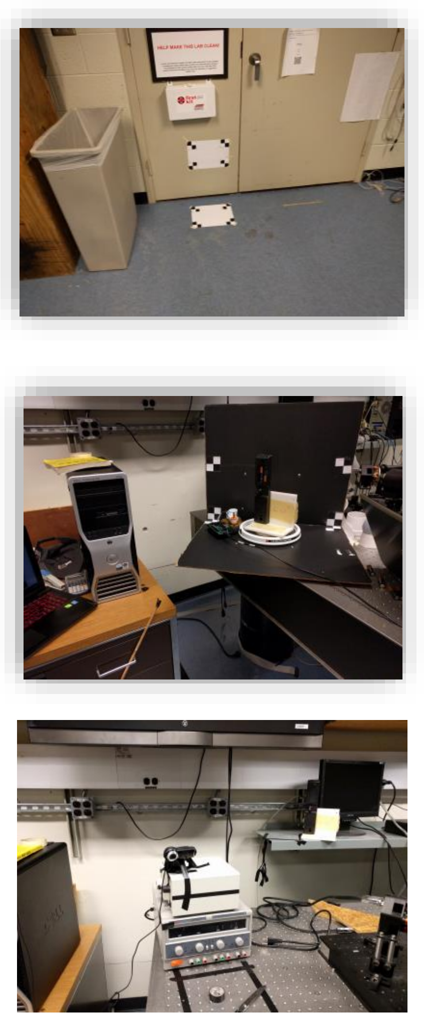


Figure 22. Set up for the room scan

V. Problems Encountered

The first problem we faced was with our initial setup that was very shaky and non-robust. We fixed this by making version 2 of the setup.

Another issue we faced was finding hidden camera modes. We collected a bunch of data before we realized that the video feature of the camera had different intrinsic parameters than the photograph feature. We solved this by calibrating the video feature of the camera by repeating the camera calibration procedure and re-computing the intrinsic parameters.

The third issue was faced with our laser light. The one we bought online wasn't bright enough and therefore it was harder to detect its edge on many types of objects. Also, we noticed that the line it produced was actually multiple, distinct laser spots. This added holes in our reconstruction.

Lastly, for the rotation scan, it was really hard to get a constant angular speed for the turn table. Unfortunately we weren't able to completely negate this and it affected our results. In the future, it can be resolved by using a better motor with an encoder in it that directly controls the turntable.

VI. Conclusions

We have achieved the goals that we set out for the final project to develop a 3D scanner with one laser line-light and one camera.

We have used the following concepts from the Computer Vision class:

- Camera Calibration (External Toolbox Used)
- Perspective Projection & Transformations between coordinate frames
- Actice Stereo 3D Reconstruction
- Difference of Gaussian and Laplacian of Gaussian for Edge Detection
- Application of our scanner in different scenarios

References

[1] Ji, Qiang. "ECSE 6650 Lecture Notes." N.p., n.d. Web. 10 May. 2016. https://www.ecse.rpi.edu/Homepages/qji/CV/ecse6650lecture notes.html

[2] Build Your Own 3D Scanner: 3D Photography for Beginners mesh.brown.edu/**byo3d**/notes/**byo3D.pdf**

[3] Bouguet, Jean-Yves, and Pietro Perona. "3D photography using shadows in dual-space geometry." *International Journal of Computer Vision* 35.2 (1999): 129-149.

[4] SIGGRAPH2014 – 3D Scanning for Personal 3D Printing: Build Your Own Desktop 3D Scanner mesh.br**own**.edu/**desktop3dscan**/SG14-byod3d.pdf

[5] Meshlab **meshlab**.sourceforge.net

[6] NXT Segway motor www.nxtprograms.com/NXT2/**segway**/

[7] Arduino Mega 2560 www.**arduino**.cc/en/Main/**arduino**Board**Mega**2560

[8] Camera Calibration Toolbox for MATLAB www.vision.caltech.edu/bouguetj/calib_doc/