Final Report - Lab #3 - Projective Geometry

Daniel Engelsman - 300546173
Tom Smadar - 203374061

Part 1:

(a) We calculated the waypoints and landmark according to the given scenario and fed them to the envelope script:

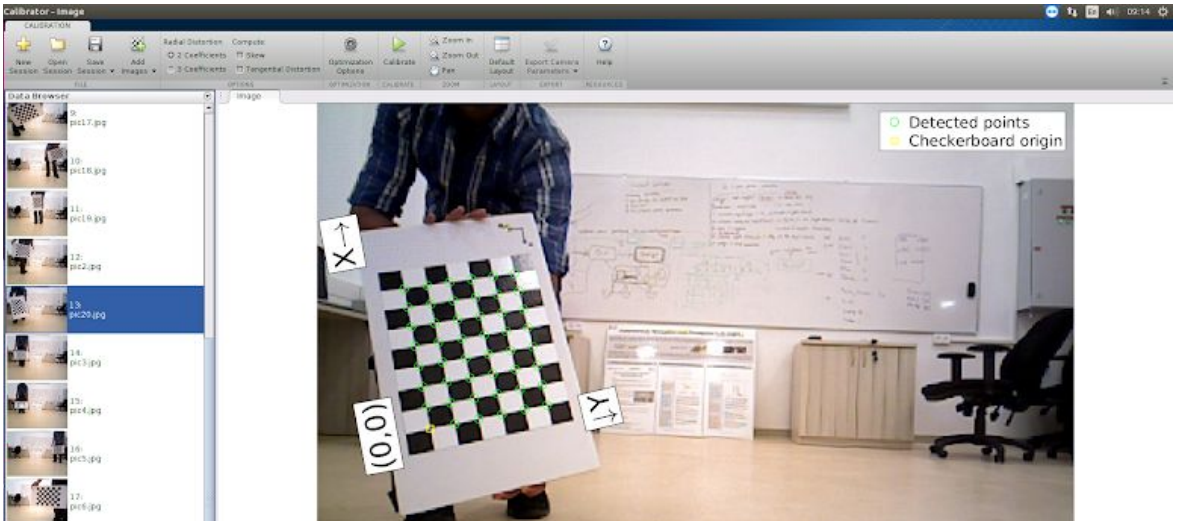
```
%% Goals
Goal(1).loc = [1.244 0 0]';
Goal(1).Orien = deg2rad(45);
Goal(2).loc = [2.584 1.011 0]';
Goal(2).Orien = deg2rad(315);
Goal(3).loc = [3.572 0 0]';
Goal(3).Orien = deg2rad(0);
Goal(4).loc = [4.761 0 0]';
Goal(4).Orien = deg2rad(135);
Landmark1.loc = [2.656 1.83 0.3]';
Landmark2.loc = [3.794 1.495 0.3]';
```

(b) We've updated the pose and velocity topics to match the ones in the real Pioneer (not the Gazebo)

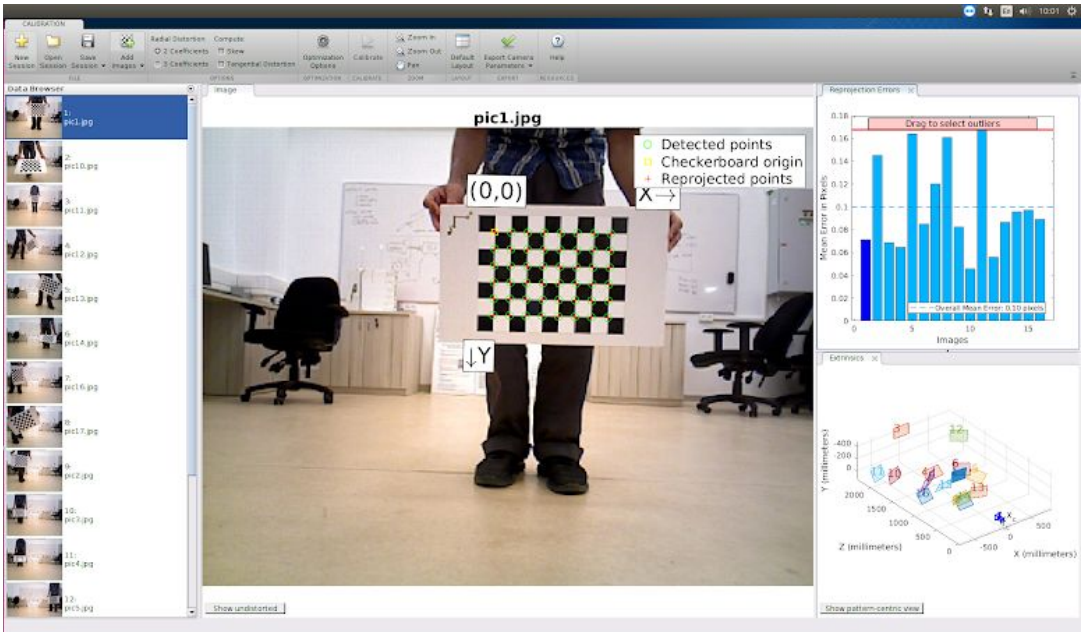
(c)

Part 2:

We've used Matlab's Camera Calibration Tool to find the parameters needed for the Pioneer's calibration matrix. We've also chucked out three images in which the calibration tool misread the checkered origin:



And ran the Calibration:



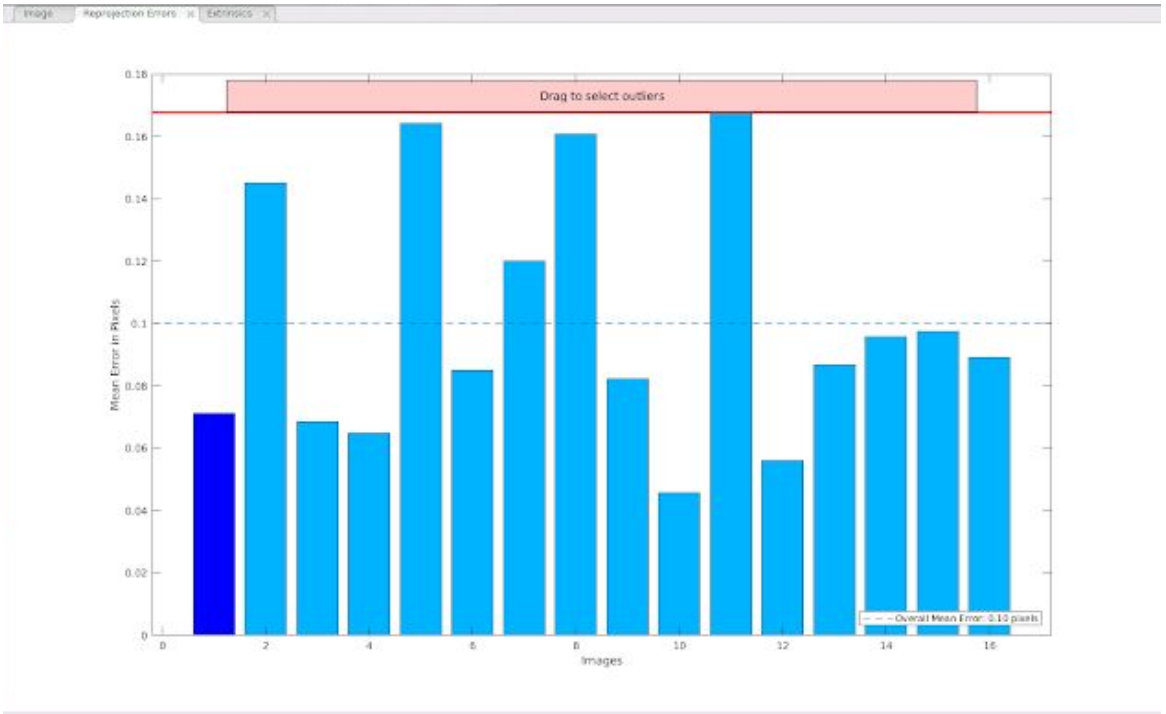


Image
Reprojection Errors
Extrinsics
Drag to select outliers
Mean Error in Pixels
Images
Overall Mean Error: 0.10 pixels

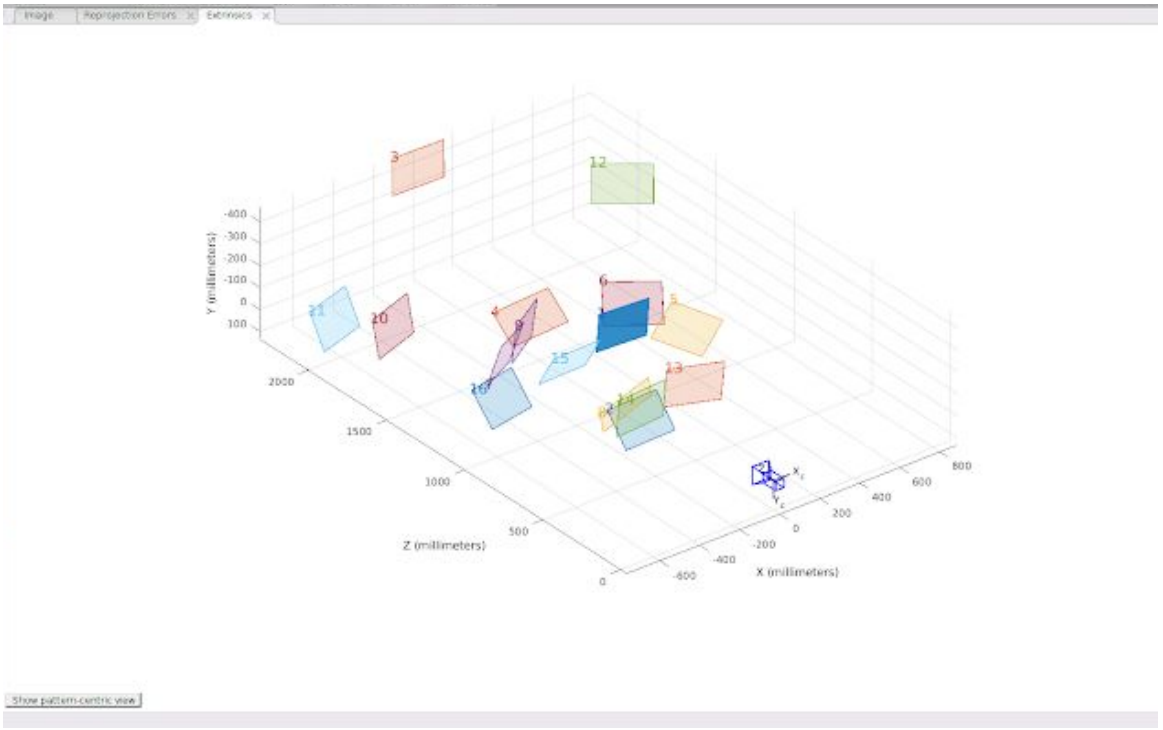
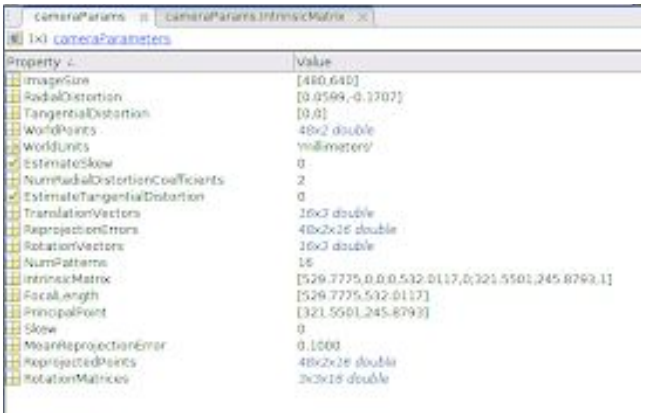


Image
Reprojection Errors
Extrinsics
Y (millimeters)
Z (millimeters)
X (millimeters)
Show pattern-centric view

We used the “Export Camera Parameters” option in the calibration tool and got:



So we derived:

$f_x = 529.7775\ [px]$

$f_y = 532.0117\ [px]$

$P_x = 322[px]$

$P_y = 246[px]$

And so our estimated calibration matrix will be:

$$K = \begin{pmatrix} fx & s & Px \\ 0 & fy & Py \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 529.777 & 0 & 322 \\ 0 & 532.012 & 246 \\ 0 & 0 & 1 \end{pmatrix}$$

Part 3:

(a) We've selected image number 96 and loaded it into matlab:

```
%% Read image from ROSbag
bag = rosbag('rosbag_ac.bag')
bagselect1 = select(bag,'Topic','/camera/rgb/image_raw/compressed')
imgMsg = readMessages(bagselect1);
C = imgMsg{96}.Data;
fileID = fopen('test.jpg','w');
fwrite(fileID,C);
fclose(fileID);
imshow('test.jpg')
```

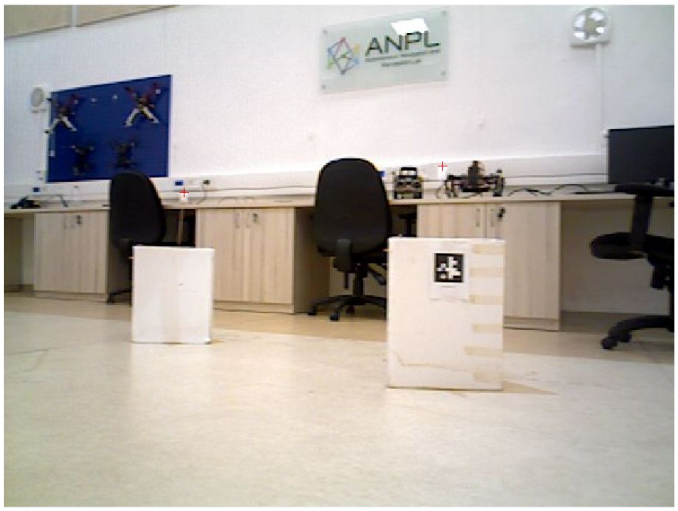
(b) We've not used "Ground Truth", but odometry reading instead:

```
%% Calculate landmark location
bagselect2 = select(bag,'Topic','/RosAria/pose');
poseMsg = readMessages(bagselect2);
P = poseMsg{96}.Pose.Pose.Position;
alpha = poseMsg{96}.Pose.Pose.Orientation;
quat = [alpha.X alpha.Y alpha.Z alpha.W];
alpha = quat2eul(quat);
alpha = alpha(3);
Rob = [P.X P.Y 0]'; % Robot Position (Global)
Rc = [0 -1 0;0 0 -1;1 0 0]; % Rotation from Global to Camera
tC2O = [0.018 0 -0.192]'; % Translation from Camera to Odometry in Camera Frame
Rob = Rc*Rob + tC2O;

% Landmarks:
Landmark1 = Rc*[2.656 1.83 0.3]';
Landmark2 = Rc*[3.794 1.495 0.3]';
% Relative:
t1 = Landmark1-Rob;
t2 = Landmark2-Rob;
```

(c) And derived their “2D” coordinates from the image:

```
%% Derive coordinates:
figure(1)
imshow('test.jpg')
hold on
[Px,Py]= ginput;
tru = scatter(Px,Py,80,'+','r');
```

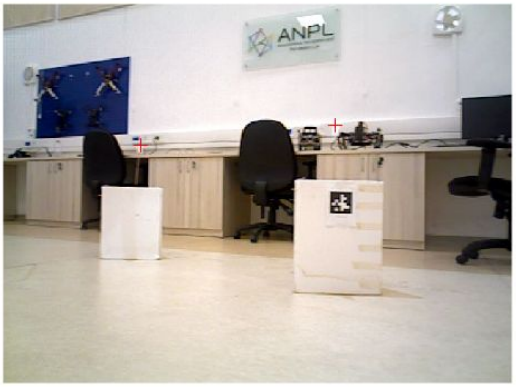


(d) Projecting the 3D landmarks into the camera frame:

```
%% Project Coordinates:
R = angle2dcm(0,alpha,0,'XYZ');
K = [529.777 0 322;0 532.012 246;0 0 1];

% Projection Matrices:
T1 = R;
T1(:,4) = t1;
T2 = R;
T2(:,4) = t2;
M1 = K*T1;
M2 = K*T2;

% Homogenous:
Landmark1(4) = 1;
Landmark2(4) = 1;
L(:,1) = M1*Landmark1;
L(:,1) = L(:,1)./L(3,1);
L(:,2) = M2*Landmark2;
L(:,2) = L(:,2)./L(3,2);
scatter(L(1,:),L(2,:),80,'+','b');
set(gca,'Xlim',[0 900])
```



We can see that the projected points (marked blue) are quite far from the manually selected ones (marked red). There is a “height” difference that might pertain to the Y translation between the odometry sensor and the camera, and an “Azimuth” difference that might pertain to the Pioneer’s angular state measurement error.

(e) The re-projection error for the landmarks was calculated as such:

```
%% Re-Projection Error:
errx = Px' - L(1,:);
erry = Py' - L(2,:);
landErr1 = norm([errx(1) erry(1)])
landErr2 = norm([errx(2) erry(2)])
```

Landmark	**Error [px]**
Landmark 1	540
Landmark 2	511