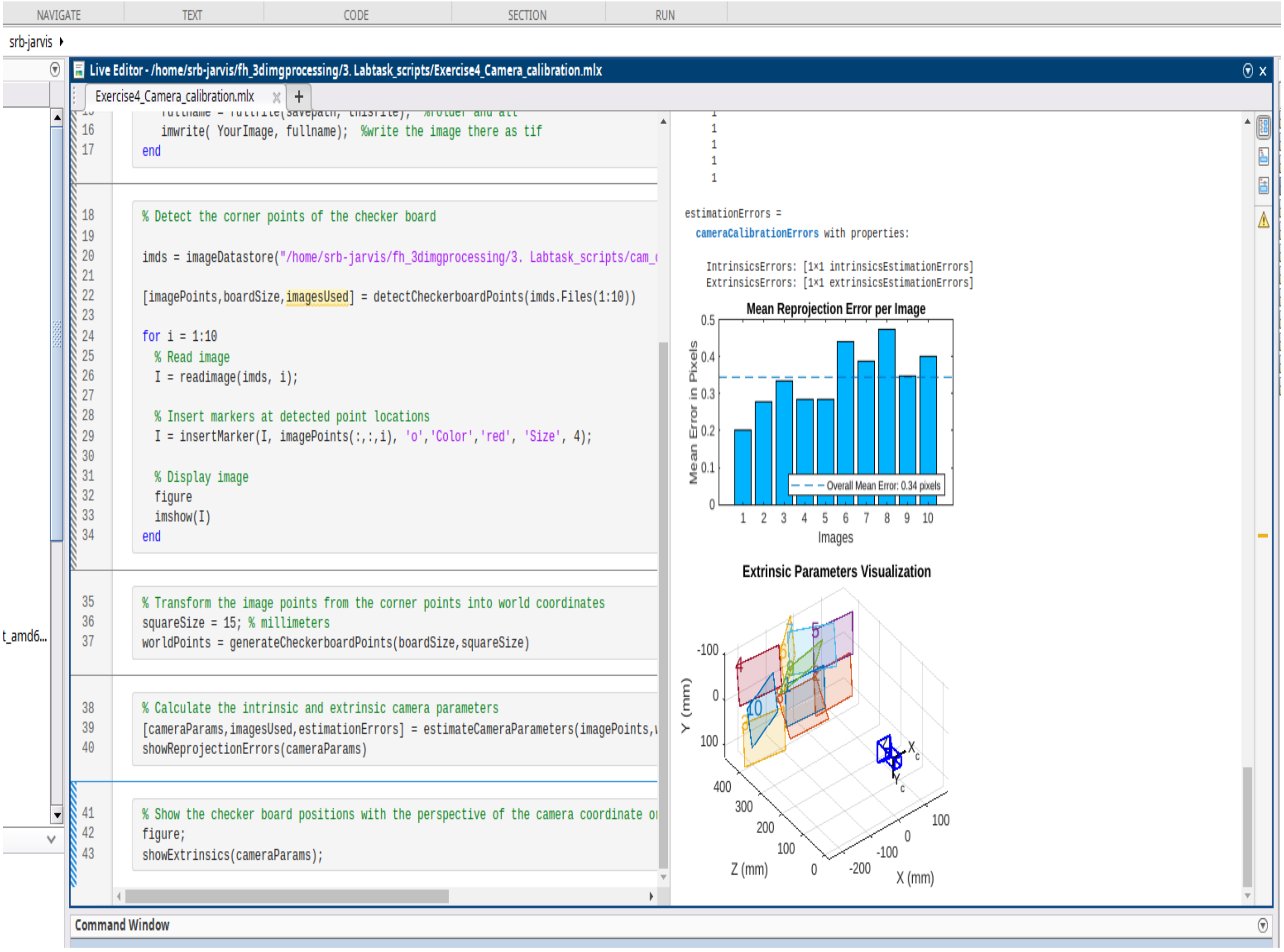
Practicum:1 Mono camera calibration :

output : checker board positions with the perspective of the camera coordinate origin


NAVIGATE
TEXT
CODE
SECTION
RUN
srb-jarvis
Live Editor - /home/srb-jarvis/fh_3dimgprocessing/3. Labtask_scripts/Exercise4_Camera_calibration.mlx
Exercise4_Camera_calibration.mlx
t_amd6...
Command Window
Mean Reprojection Error per Image
Mean Error in Pixels
0
0.1
0.2
0.3
0.4
0.5
1 2 3 4 5 6 7 8 9 10
Images
Overall Mean Error: 0.34 pixels
Extrinsic Parameters Visualization
Y (mm)
-100
0
100
Z (mm)
400
300
200
100
0
X (mm)
-200
-100
0
100
X_c
Y_c
1 2 3 4 5 6 7 8 9 10

```
    fullname = fullfile(savepath, thisfile);  %folder and all
    imwrite( YourImage, fullname);  %write the image there as tif
end

% Detect the corner points of the checker board

imds = imageDatastore("/home/srb-jarvis/fh_3dimgprocessing/3. Labtask_scripts/cam_

[imagePoints,boardSize,imagesUsed] = detectCheckerboardPoints(imds.Files(1:10))

for i = 1:10
  % Read image
  I = readimage(imds, i);

  % Insert markers at detected point locations
  I = insertMarker(I, imagePoints(:,:,i), 'o','Color','red', 'Size', 4);

  % Display image
  figure
  imshow(I)
end

% Transform the image points from the corner points into world coordinates
squareSize = 15; % millimeters
worldPoints = generateCheckerboardPoints(boardSize,squareSize)

% Calculate the intrinsic and extrinsic camera parameters
[cameraParams,imagesUsed,estimationErrors] = estimateCameraParameters(imagePoints,w
showReprojectionErrors(cameraParams)

% Show the checker board positions with the perspective of the camera coordinate or
figure;
showExtrinsics(cameraParams);
```

```
     1
     1
     1
     1
     1

estimationErrors = 
  cameraCalibrationErrors with properties:

    IntrinsicsErrors: [1×1 intrinsicsEstimationErrors]
    ExtrinsicsErrors: [1×1 extrinsicsEstimationErrors]
```

Source code :

```
%step1: List webcam, capture images and store in folder ranging from 1-10 nos.
webcamlist
clear('cam');
cam = webcam('/dev/video0');
preview(cam)

savepath = '/home/srb-jarvis/fh_3dimgprocessing/3. Labtask_scripts/cam_calibration';  %the folder
nametemplate = 'image_%02d.jpg';  %name pattern
for K = 1 : 10    %if you want to do this 50 times
   pause(2)
   [YourImage, t] = snapshot(cam)%capture the image
   thisfile = sprintf(nametemplate,K);  %create filename
   fullname = fullfile(savepath, thisfile);  %folder and all
   imwrite( YourImage, fullname);  %write the image there as tif
end
```

```
% Step2: Detect the corner points of the checker board
imds = imageDatastore("/home/srb-jarvis/fh_3dimgprocessing/3.Labtask_scripts/cam_calibration","FileExtensions",[".jpg",".tif"])
[imagePoints,boardSize,imagesUsed] = detectCheckerboardPoints(imds.Files(1:10))
for i = 1:10
  % Read image
  I = readimage(imds, i);
  % Insert markers at detected point locations
  I = insertMarker(I, imagePoints(:,:,i), 'o','Color','red', 'Size', 4);
  % Display image
  figure
  imshow(I)
end

% Step3: Transform the image points from the corner points into world coordinates
squareSize = 15; % millimeters
worldPoints = generateCheckerboardPoints(boardSize,squareSize)

% Step4: Calculate the intrinsic and extrinsic camera parameters
[cameraParams,imagesUsed,estimationErrors] = estimateCameraParameters(imagePoints,worldPoints)
showReprojectionErrors(cameraParams)

% Step5: Show the checker board positions with the perspective of the camera coordinate origin
figure;
showExtrinsics(cameraParams);
```