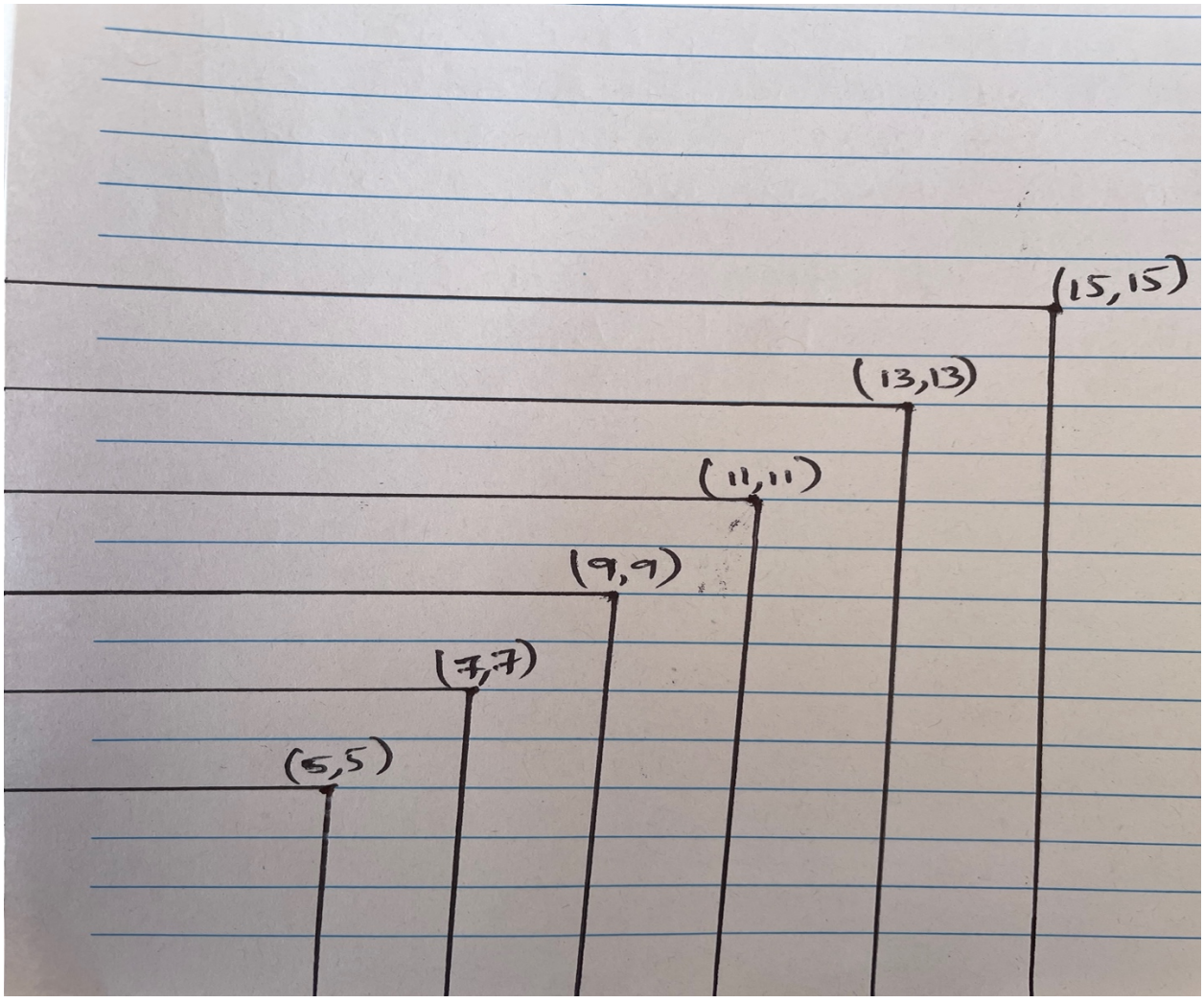
HW-1
COMPUTER VISION

1. Point the camera to a chessboard pattern or any known set of reference points that lie on the same plane. Capture a series of 10 images by changing the orientation of the camera in each iteration. Select any 1 image, and using the image formation pipeline equation, set up the linear equations in matrix form and solve for intrinsic and extrinsic parameters (extrinsic for that particular orientation). You will need to make measurements of the actual 3D world points, and mark pixel coordinates.

Once you compute the Rotation matrix, you also need to compute the angles of rotation along each axis. Choose your order of rotation based on your experimentation setup.



	x	y	u	v
	0	0	1.261×10^3 ≈0	973.20 ≈0
point 1	50	50	1.296×10^3 ≈78	897.40 ≈73.3
point 2	70	70	1.33×10^3 ≈109.2	870.50 102.6
point 3	90	90	1.376×10^3 ≈144	842.62 ≈132
point 4	110	110	1.413×10^3 ≈174	813.26 ≈161
point 5	130	130	1.46×10^3 ≈208	785.50 ≈188
point 6	150	150	1.486×10^3 ≈240.6	755.10 ≈217

Parametric matrix :-

$$a = \begin{bmatrix}
50,50,2,1,0,0,0,0, & -3880, & -3880, & -155, & -82 \\
0,0,0,0,50,50,50,50, & -3675, & -3690, & -144, & -78 \\
70,70,2,1,0,0,0,0, & -7650, & -7650, & -222, & -109 \\
0,0,0,0,70,70,2,1, & -7120, & -7120, & -202, & -101 \\
90,90,2,1,0,0,0,0, & -12954, & -12954, & -280, & -148 \\
0,0,0,0,90,90,2,1, & -11790, & -11790, & -259, & -133 \\
110,110,2,1,0,0,0,0, & -19136, & -19136, & -340, & -170 \\
0,0,0,0,110,110,2,1, & -17826, & -17826, & -322, & -164 \\
130,130,2,1,0,0,0,0, & -27101, & -27101, & -413, & -201 \\
0,0,0,0,130,130,2,1, & -18072, & -18072, & -380, & -187 \\
150,150,2,1,0,0,0,0, & -36220, & -36869 & -482, & -242 \\
0,0,0,0,150,150,2,1, & -32850, & -32850, & -437, & -218
\end{bmatrix}$$

$A^T \cdot AP = \lambda P$

$$\begin{bmatrix}
2500 & 0 & 140 & 0 & 0 & 0 & 0 & 0 & -504400 & 0 & -23250 & 0 \\
0 & 0 & 0 & 0 & 4500 & 0 & 220 & 0 & -477750 & 0 & -21600 & 0 \\
140 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & -15300 & 0 & -444 & 0 \\
0 & 0 & 0 & 0 & 70 & 0 & 2 & 0 & -7120 & 0 & -202 & 0 \\
0 & 4500 & 0 & 70 & 0 & 0 & 0 & 0 & 0 & -1684020 & 0 & -22200 \\
0 & 0 & 0 & 0 & 0 & 8100 & 0 & 110 & 0 & -1532700 & 0 & -19950 \\
0 & 220 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & -38272 & 0 & -340 \\
0 & 0 & 0 & 0 & 0 & 110 & 0 & 1 & 0 & -17826 & 0 & -164 \\
-504400 & -477750 & -15300 & -7120 & 0 & 0 & 0 & 0 & 734464201 & 48976272 & 14958860 & 6602850 \\
0 & 0 & 0 & 0 & -1684020 & -1532700 & -38272 & -17826 & 48976272 & 3265971184 & 13763600 & 6142950 \\
-23250 & -21600 & -444 & -202 & 0 & 0 & 0 & 0 & 14958860 & 13763600 & 232324 & 105754 \\
0 & 0 & 0 & 0 & -22200 & -19950 & -340 & -164 & 6602850 & 6142950 & 105754 & 47524
\end{bmatrix}$$

parametrix matrix

$$\begin{bmatrix} -504400, & -477750, & -15300, & 7126 \\ -1684\text{[illegible]}00, & -1532700, & -38272, & -17826 \\ -504400, & -1684020, & -23250, & -22200 \end{bmatrix}$$

QR factorization

$$\begin{bmatrix} -1296020 & -1283281 & -221979 \\ -10869261 & -10799486 & -193775 \\ -12807620 & -11908060 & -221239 \end{bmatrix}$$

translation matrix

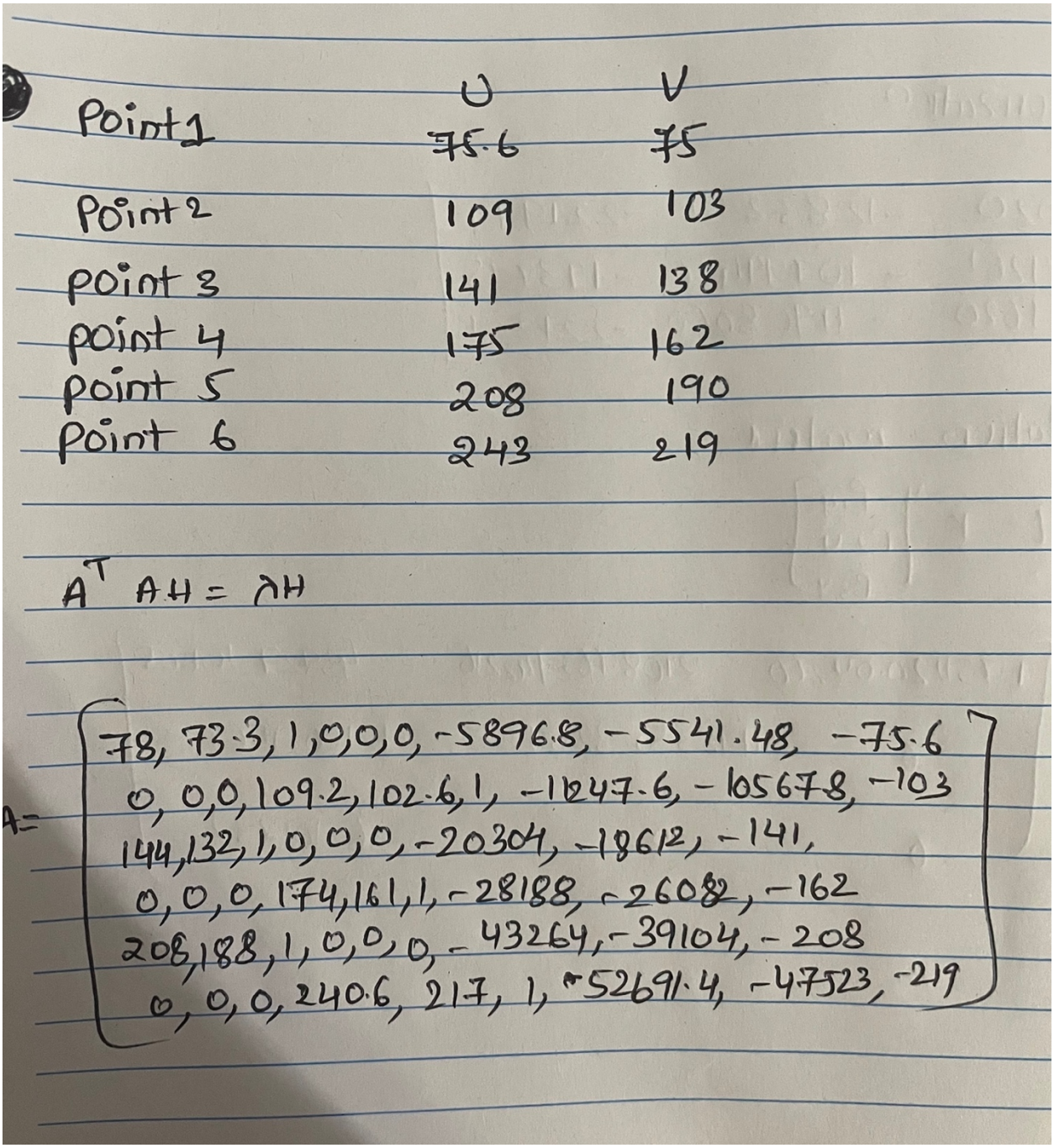
$$t = K^{-1} \begin{bmatrix} P_{14} \\ P_{24} \\ P_{34} \end{bmatrix}$$

$$K = \begin{bmatrix} 1731120.04260 & 2108763.140216 & 43927.10605 \\ 0 & -1172350.3269 & -12160.14676 \\ 0 & 0 & -362101237 \end{bmatrix}$$

final extrinsic vector

$$= \begin{bmatrix} -0.19165132 \\ -0.0490964 \\ 5.0068021 \end{bmatrix}$$

2. Select any pair of images from the set in problem 1 above. Compute the homography between those two images.

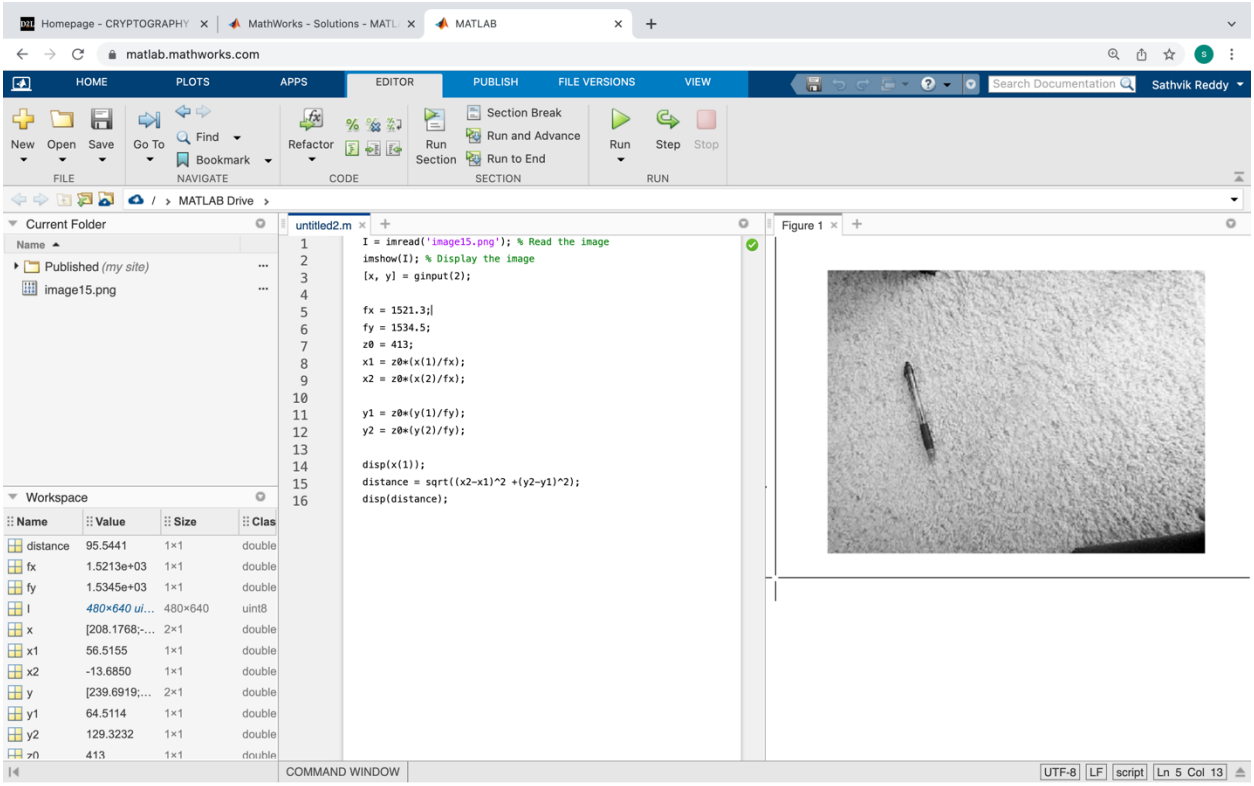


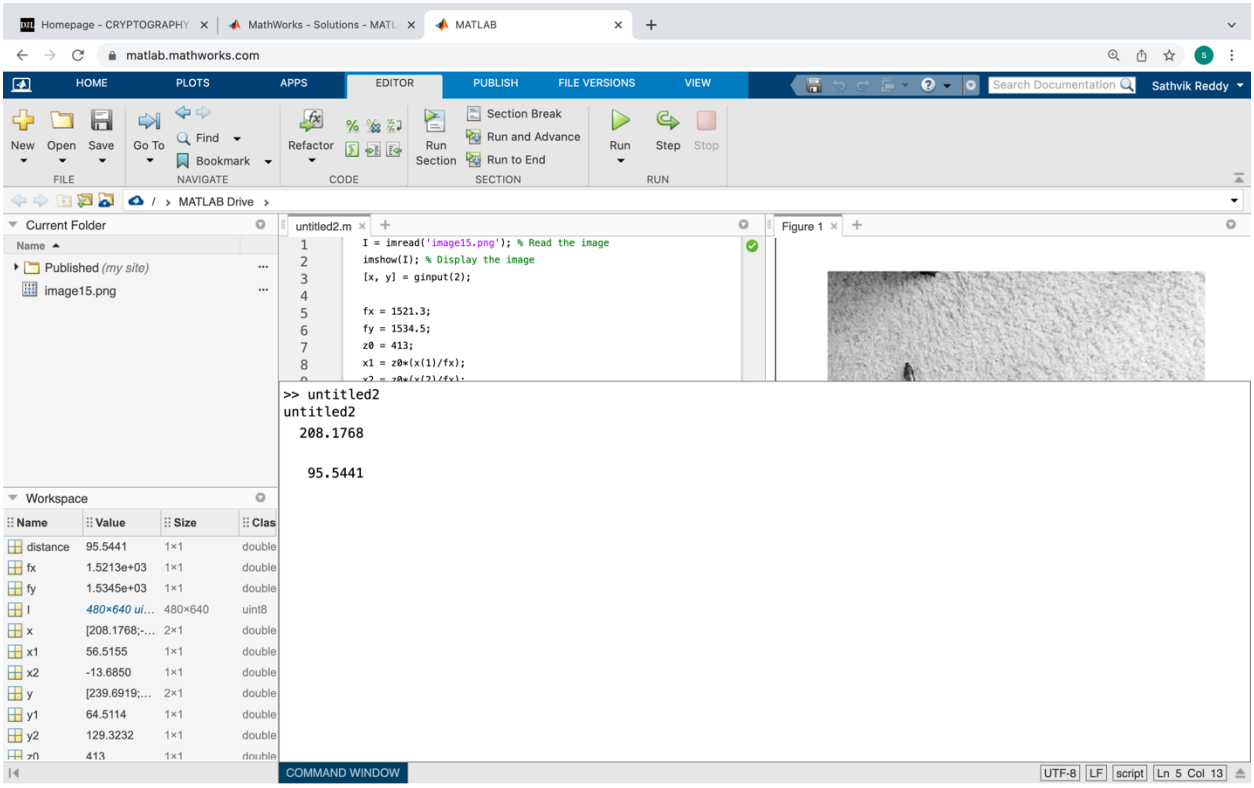
	U	V
Point 1	75.6	75
Point 2	109	103
point 3	141	138
point 4	175	162
point 5	208	190
point 6	243	219

$A^T A H = \lambda H$

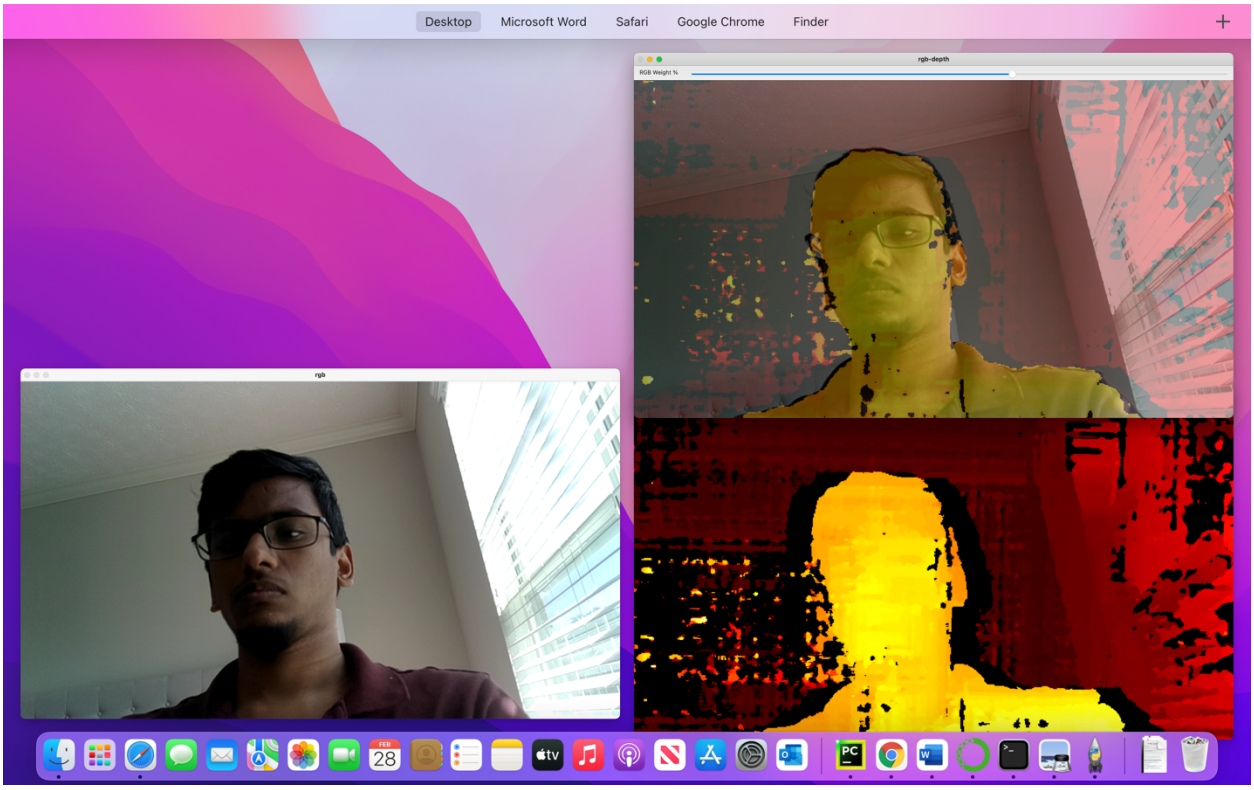
$$A = \begin{bmatrix} 78, 73.3, 1, 0, 0, 0, -5896.8, -5541.48, -75.6 \\ 0, 0, 0, 109.2, 102.6, 1, -11247.6, -10567.8, -103 \\ 144, 132, 1, 0, 0, 0, -20304, -18612, -141, \\ 0, 0, 0, 174, 161, 1, -28188, -26082, -162 \\ 208, 188, 1, 0, 0, 0, -43264, -39104, -208 \\ 0, 0, 0, 240.6, 217, 1, -52691.4, -47523, -219 \end{bmatrix}$$

3. MATLAB SCRIPT



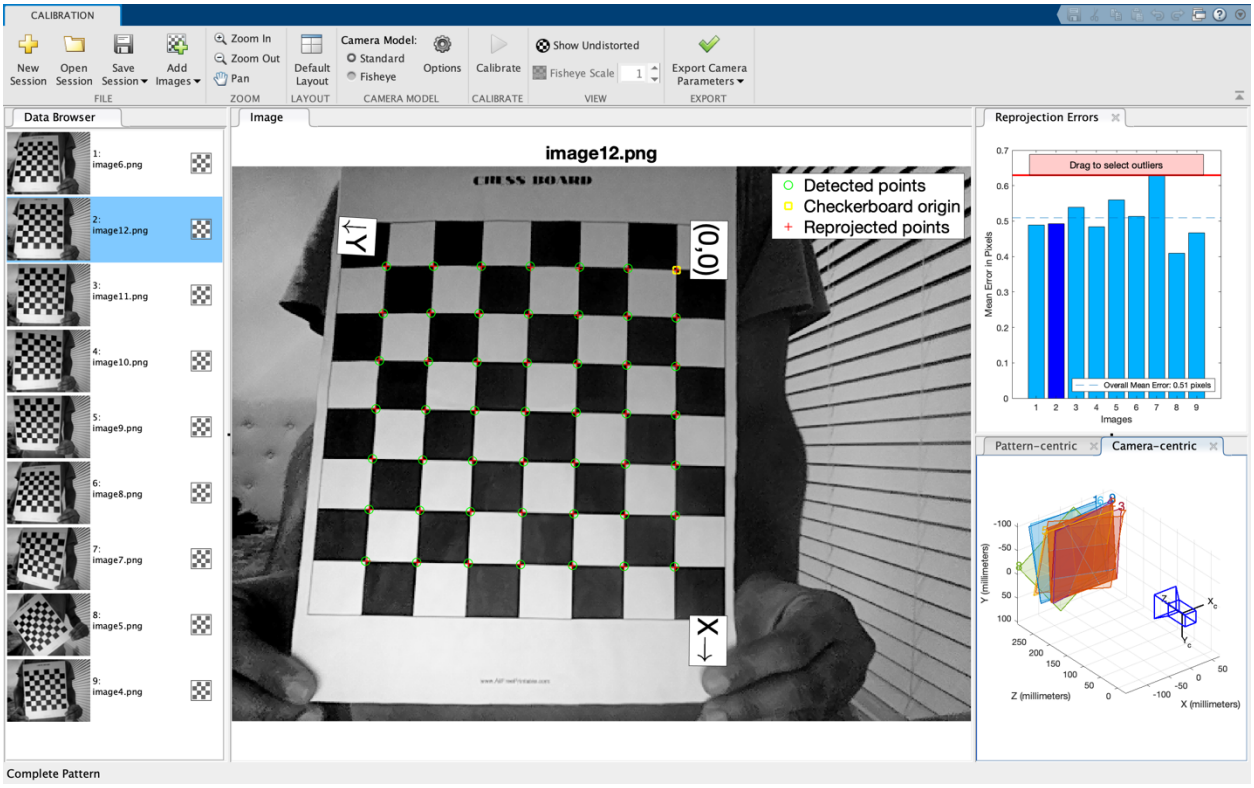


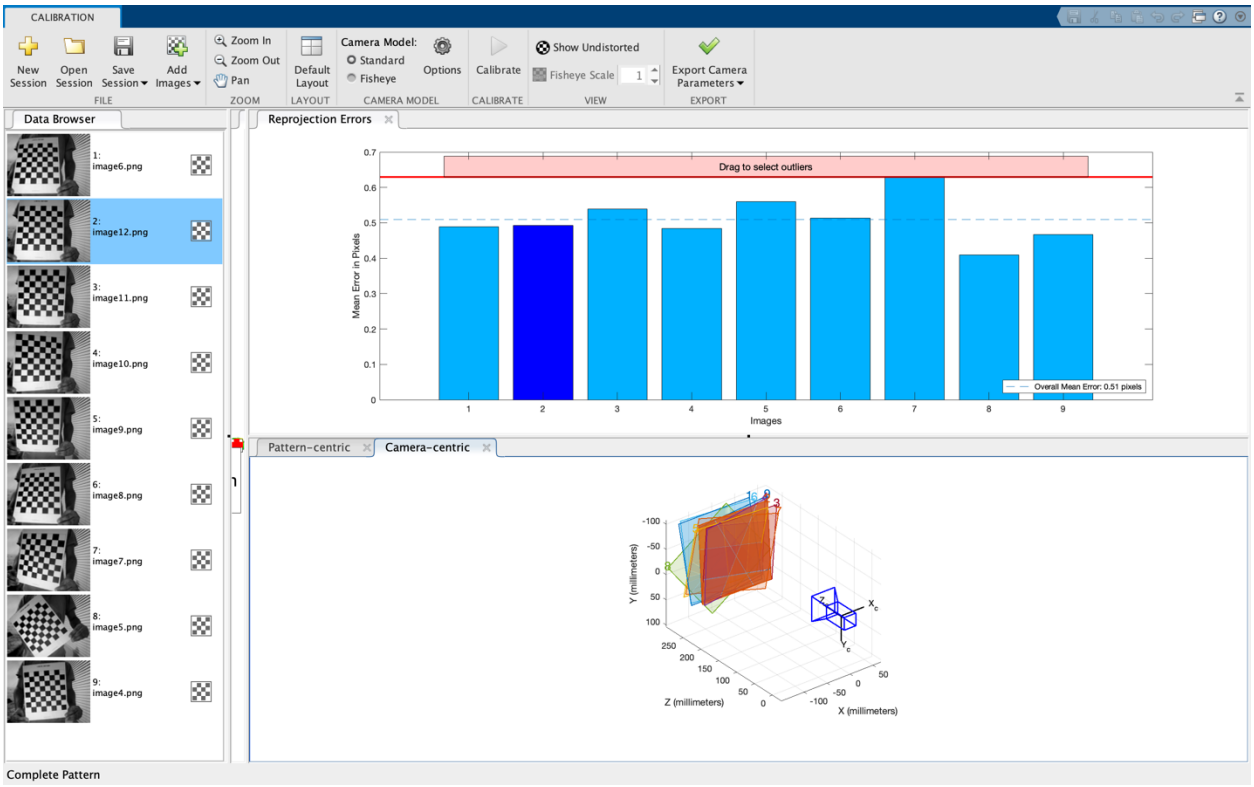
5.RGB STREAM

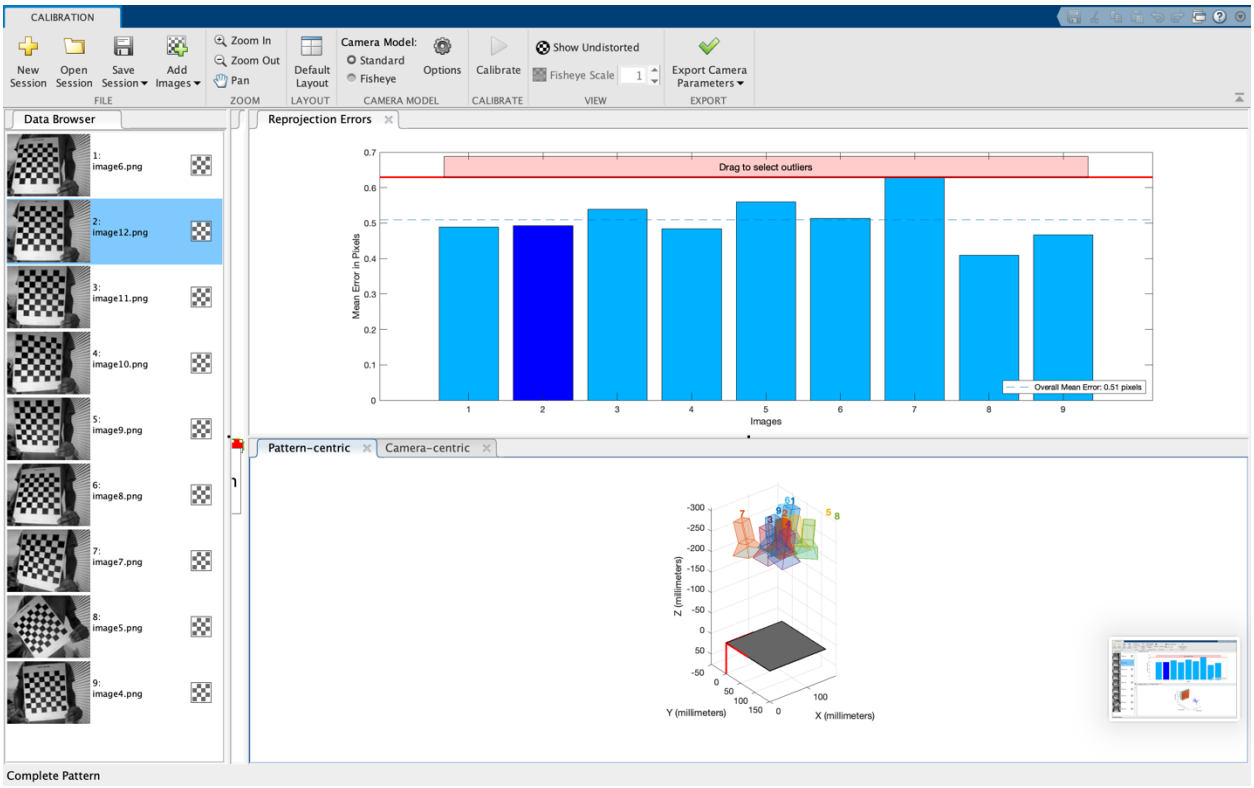


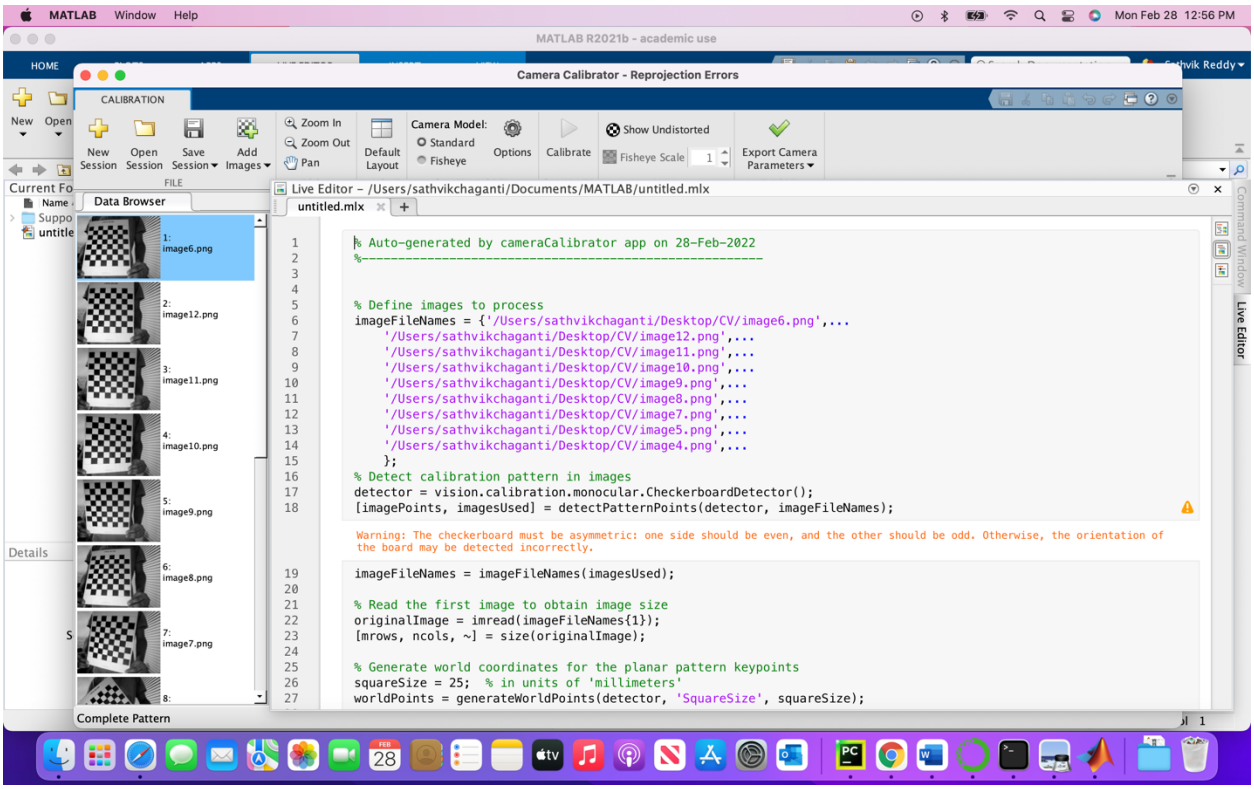
Yes, it is possible to show rgb stream from the mono camera and a depth map stream from the stereo camera.

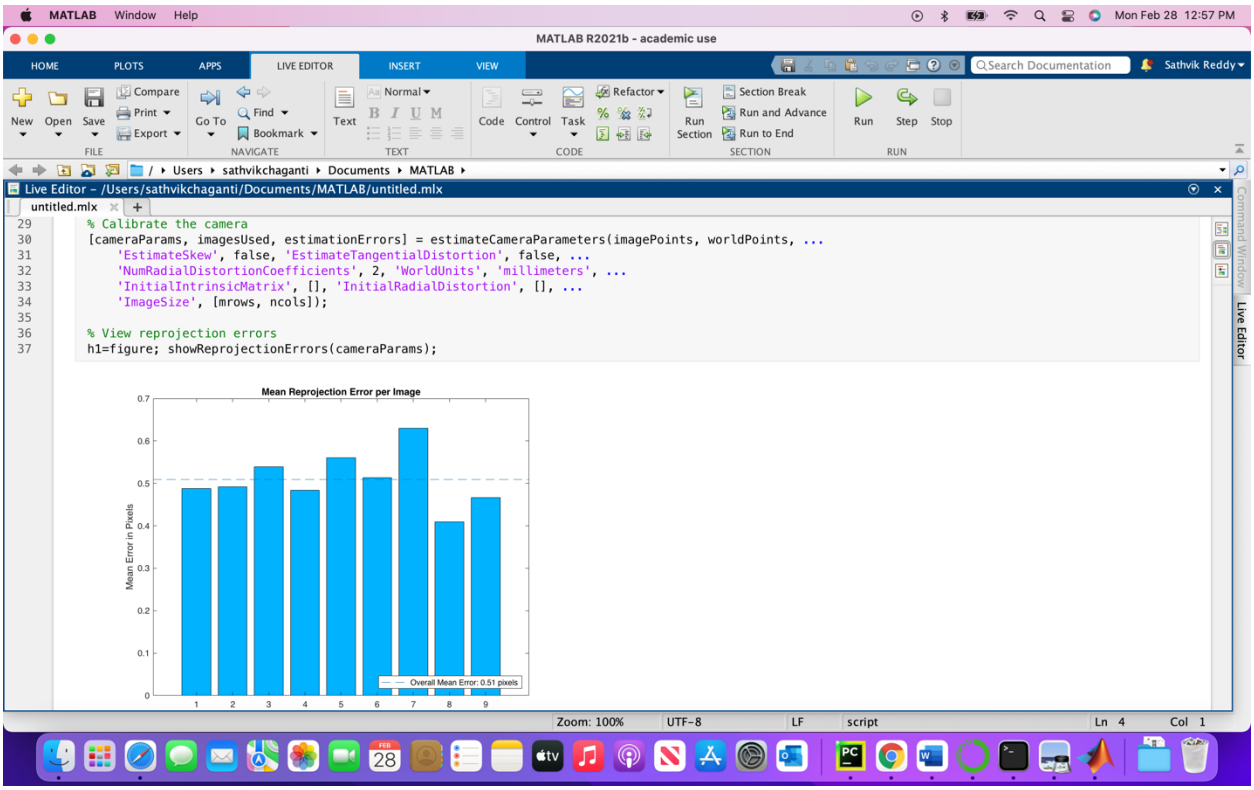
6. CAMERA CALIBRATION

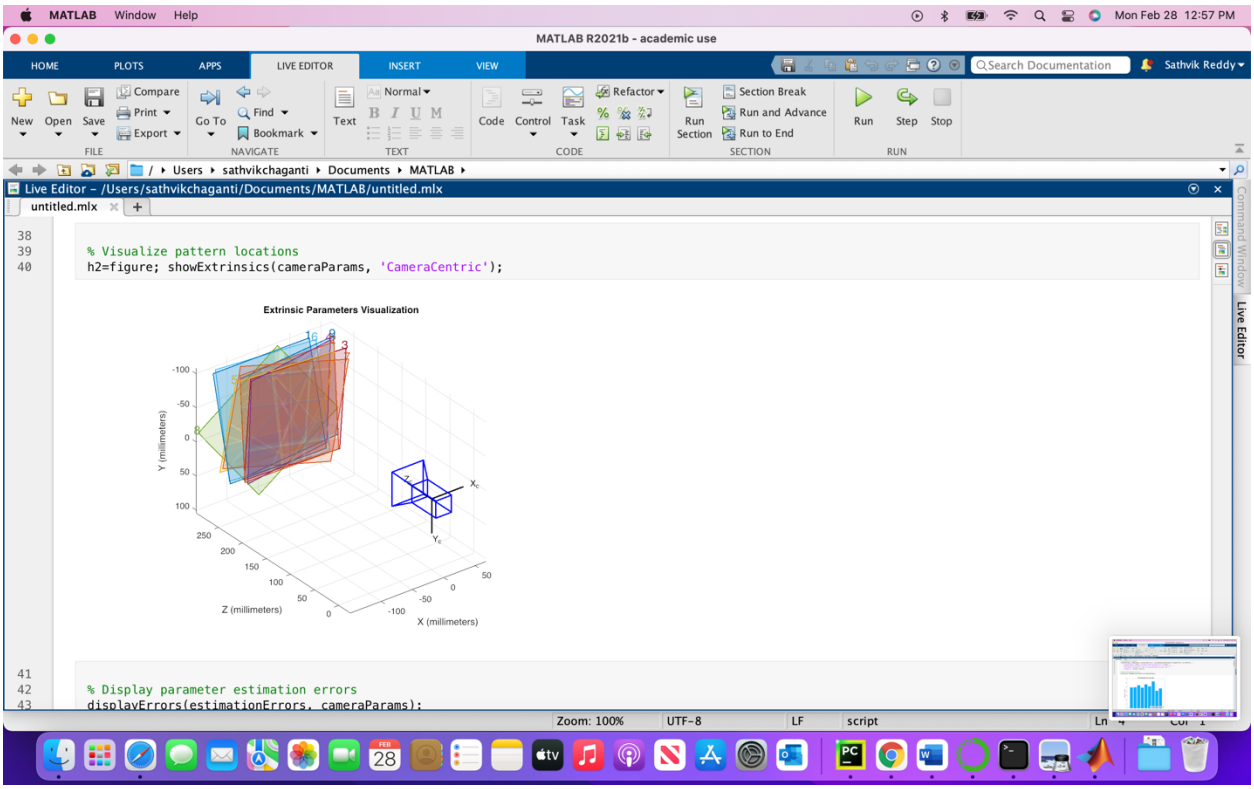


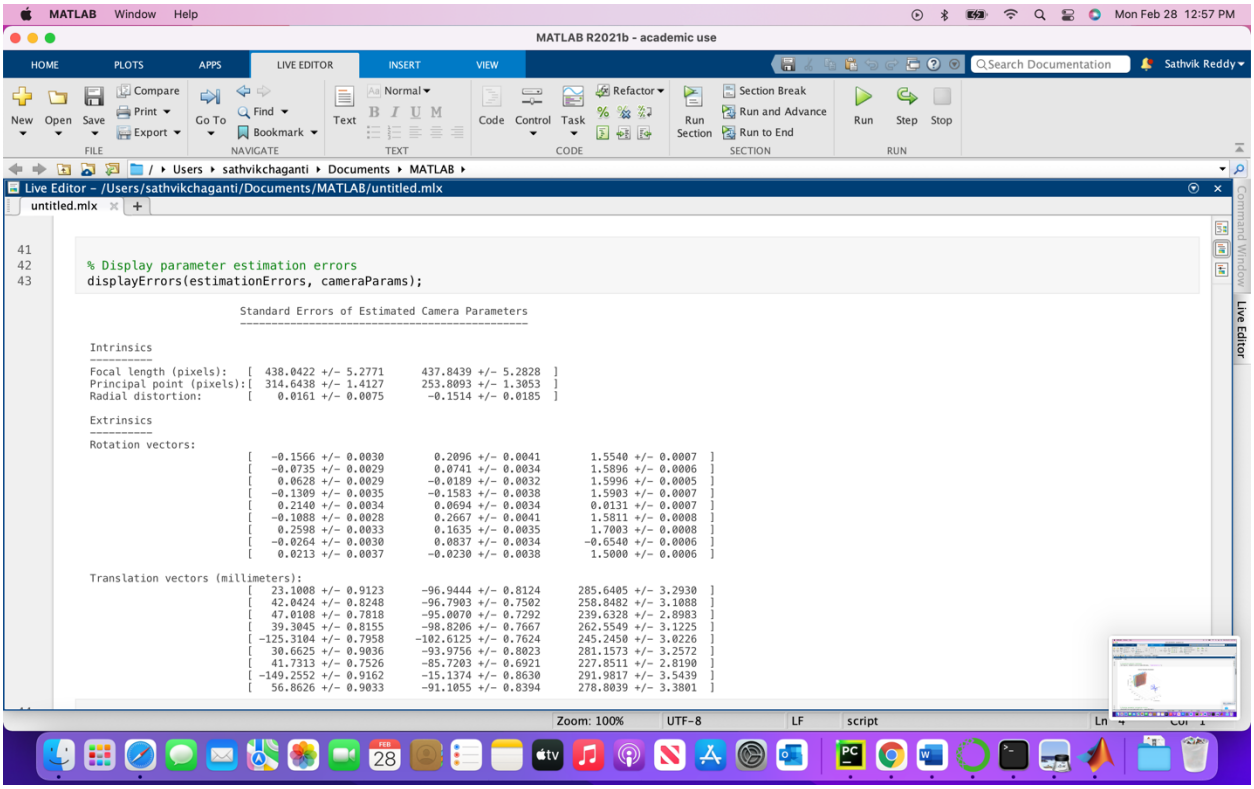

CALIBRATION
New Session
Open Session
Save Session
Add Images
FILE
Zoom In
Zoom Out
Pan
ZOOM
Default Layout
LAYOUT
Camera Model:
Standard
Fisheye
Options
CAMERA MODEL
Calibrate
CALIBRATE
Show Undistorted
Fisheye Scale
1
VIEW
Export Camera Parameters
EXPORT
Data Browser
1: image6.png
2: image12.png
3: image11.png
4: image10.png
5: image9.png
6: image8.png
7: image7.png
8: image5.png
9: image4.png
Reprojection Errors
Drag to select outliers
Mean Error in Pixels
Images
Overall Mean Error: 0.51 pixels
Pattern-centric
Camera-centric
Y (millimeters)
Z (millimeters)
X (millimeters)
Complete Pattern


CALIBRATION
New Session
Open Session
Save Session
Add Images
FILE
Zoom In
Zoom Out
Pan
ZOOM
Default Layout
LAYOUT
Camera Model:
Standard
Fisheye
Options
CAMERA MODEL
Calibrate
CALIBRATE
Show Undistorted
Fisheye Scale
1
VIEW
Export Camera Parameters
EXPORT
Data Browser
1: image6.png
2: image12.png
3: image11.png
4: image10.png
5: image9.png
6: image8.png
7: image7.png
8: image5.png
9: image4.png
Reprojection Errors
Drag to select outliers
Mean Error in Pixels
Images
Overall Mean Error: 0.51 pixels
Pattern-centric
Camera-centric
Z (millimeters)
Y (millimeters)
X (millimeters)
Complete Pattern


MATLAB Window Help
Mon Feb 28 12:56 PM
MATLAB R2021b - academic use
Camera Calibrator - Reprojection Errors
CALIBRATION
New Session
Open Session
Save Session
Add Images
Zoom In
Zoom Out
Pan
Default Layout
Camera Model:
Standard
Fisheye
Options
Calibrate
Show Undistorted
Fisheye Scale
Export Camera Parameters
Data Browser
1: image6.png
2: image12.png
3: image11.png
4: image10.png
5: image9.png
6: image8.png
7: image7.png
8:
Live Editor - /Users/sathvikchaganti/Documents/MATLAB/untitled.mlx
untitled.mlx
% Auto-generated by cameraCalibrator app on 28-Feb-2022
% Define images to process
imageFileNames = {'/Users/sathvikchaganti/Desktop/CV/image6.png',...
'/Users/sathvikchaganti/Desktop/CV/image12.png',...
'/Users/sathvikchaganti/Desktop/CV/image11.png',...
'/Users/sathvikchaganti/Desktop/CV/image10.png',...
'/Users/sathvikchaganti/Desktop/CV/image9.png',...
'/Users/sathvikchaganti/Desktop/CV/image8.png',...
'/Users/sathvikchaganti/Desktop/CV/image7.png',...
'/Users/sathvikchaganti/Desktop/CV/image5.png',...
'/Users/sathvikchaganti/Desktop/CV/image4.png',...
};
% Detect calibration pattern in images
detector = vision.calibration.monocular.CheckerboardDetector();
[imagePoints, imagesUsed] = detectPatternPoints(detector, imageFileNames);
Warning: The checkerboard must be asymmetric: one side should be even, and the other should be odd. Otherwise, the orientation of the board may be detected incorrectly.
imageFileNames = imageFileNames(imagesUsed);
% Read the first image to obtain image size
originalImage = imread(imageFileNames{1});
[mrows, ncols, ~] = size(originalImage);
% Generate world coordinates for the planar pattern keypoints
squareSize = 25; % in units of 'millimeters'
worldPoints = generateWorldPoints(detector, 'SquareSize', squareSize);
Complete Pattern


MATLAB Window Help
Mon Feb 28 12:57 PM
MATLAB R2021b - academic use
HOME
PLOTS
APPS
LIVE EDITOR
INSERT
VIEW
New
Open
Save
Compare
Print
Export
Go To
Find
Bookmark
Text
Code
Control
Task
Refactor
Run Section
Section Break
Run and Advance
Run to End
Run
Step
Stop
Users sathvikchaganti Documents MATLAB
Live Editor - /Users/sathvikchaganti/Documents/MATLAB/untitled.mlx
untitled.mlx
% Calibrate the camera
[cameraParams, imagesUsed, estimationErrors] = estimateCameraParameters(imagePoints, worldPoints, ...
'EstimateSkew', false, 'EstimateTangentialDistortion', false, ...
'NumRadialDistortionCoefficients', 2, 'WorldUnits', 'millimeters', ...
'InitialIntrinsicMatrix', [], 'InitialRadialDistortion', [], ...
'ImageSize', [mrows, ncols]);
% View reprojection errors
h1=figure; showReprojectionErrors(cameraParams);
Mean Reprojection Error per Image
Mean Error in Pixels
Overall Mean Error: 0.51 pixels
Zoom: 100%
UTF-8
LF
script
Ln 4
Col 1


MATLAB Window Help
Mon Feb 28 12:57 PM
MATLAB R2021b - academic use
HOME PLOTS APPS LIVE EDITOR INSERT VIEW
Search Documentation
Sathvik Reddy
New Open Save Compare Print Export Go To Find Bookmark Text Normal Code Control Task Refactor Run Section Section Break Run and Advance Run to End Run Step Stop
FILE NAVIGATE TEXT CODE SECTION RUN
/ Users sathvikchaganti Documents MATLAB
Live Editor - /Users/sathvikchaganti/Documents/MATLAB/untitled.mlx
untitled.mlx
% Visualize pattern locations
h2=figure; showExtrinsics(cameraParams, 'CameraCentric');
Extrinsic Parameters Visualization
Y (millimeters)
Z (millimeters)
X (millimeters)
% Display parameter estimation errors
displayErrors(estimationErrors, cameraParams);
Zoom: 100%
UTF-8
LF
script


MATLAB Window Help
Mon Feb 28 12:57 PM
MATLAB R2021b - academic use
HOME PLOTS APPS LIVE EDITOR INSERT VIEW
Search Documentation
Sathvik Reddy
New Open Save Compare Print Export Go To Find Bookmark Text Normal Code Control Task Refactor Run Section Section Break Run and Advance Run to End Run Step Stop
FILE NAVIGATE TEXT CODE SECTION RUN
/ Users sathvikchaganti Documents MATLAB
Live Editor - /Users/sathvikchaganti/Documents/MATLAB/untitled.mlx
untitled.mlx
% Display parameter estimation errors
displayErrors(estimationErrors, cameraParams);
Standard Errors of Estimated Camera Parameters
Intrinsics
Focal length (pixels): [438.0422 +/- 5.2771 437.8439 +/- 5.2828]
Principal point (pixels):[314.6438 +/- 1.4127 253.8093 +/- 1.3053]
Radial distortion: [0.0161 +/- 0.0075 -0.1514 +/- 0.0185]
Extrinsics
Rotation vectors:
[-0.1566 +/- 0.0030 0.2096 +/- 0.0041 1.5540 +/- 0.0007]
[-0.0735 +/- 0.0029 0.0741 +/- 0.0034 1.5896 +/- 0.0006]
[0.0628 +/- 0.0029 -0.0189 +/- 0.0032 1.5996 +/- 0.0005]
[-0.1309 +/- 0.0035 -0.1583 +/- 0.0038 1.5903 +/- 0.0007]
[0.2140 +/- 0.0034 0.0694 +/- 0.0034 0.0131 +/- 0.0007]
[-0.1088 +/- 0.0028 0.2667 +/- 0.0041 1.5811 +/- 0.0008]
[0.2598 +/- 0.0033 0.1635 +/- 0.0035 1.7003 +/- 0.0008]
[-0.0264 +/- 0.0030 0.0837 +/- 0.0034 -0.6540 +/- 0.0006]
[0.0213 +/- 0.0037 -0.0230 +/- 0.0038 1.5000 +/- 0.0006]
Translation vectors (millimeters):
[23.1008 +/- 0.9123 -96.9444 +/- 0.8124 285.6405 +/- 3.2930]
[42.0424 +/- 0.8248 -96.7903 +/- 0.7502 258.8482 +/- 3.1088]
[47.0108 +/- 0.7818 -95.0070 +/- 0.7292 239.6328 +/- 2.8983]
[39.3045 +/- 0.8155 -98.8206 +/- 0.7667 262.5549 +/- 3.1225]
[-125.3104 +/- 0.7958 -102.6125 +/- 0.7624 245.2450 +/- 3.0226]
[30.6625 +/- 0.9036 -93.9756 +/- 0.8023 281.1573 +/- 3.2572]
[41.7313 +/- 0.7526 -85.7203 +/- 0.6921 227.8511 +/- 2.8190]
[-149.2552 +/- 0.9162 -15.1374 +/- 0.8630 291.9817 +/- 3.5439]
[56.8626 +/- 0.9033 -91.1055 +/- 0.8394 278.8039 +/- 3.3801]
Zoom: 100%
UTF-8
LF
script