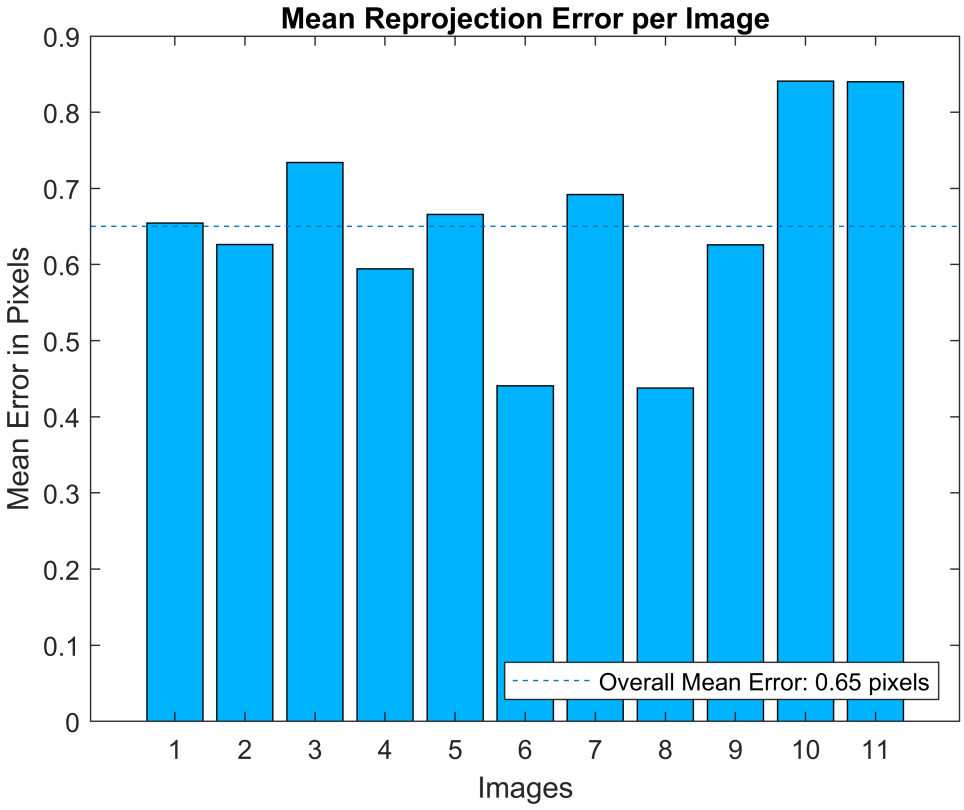
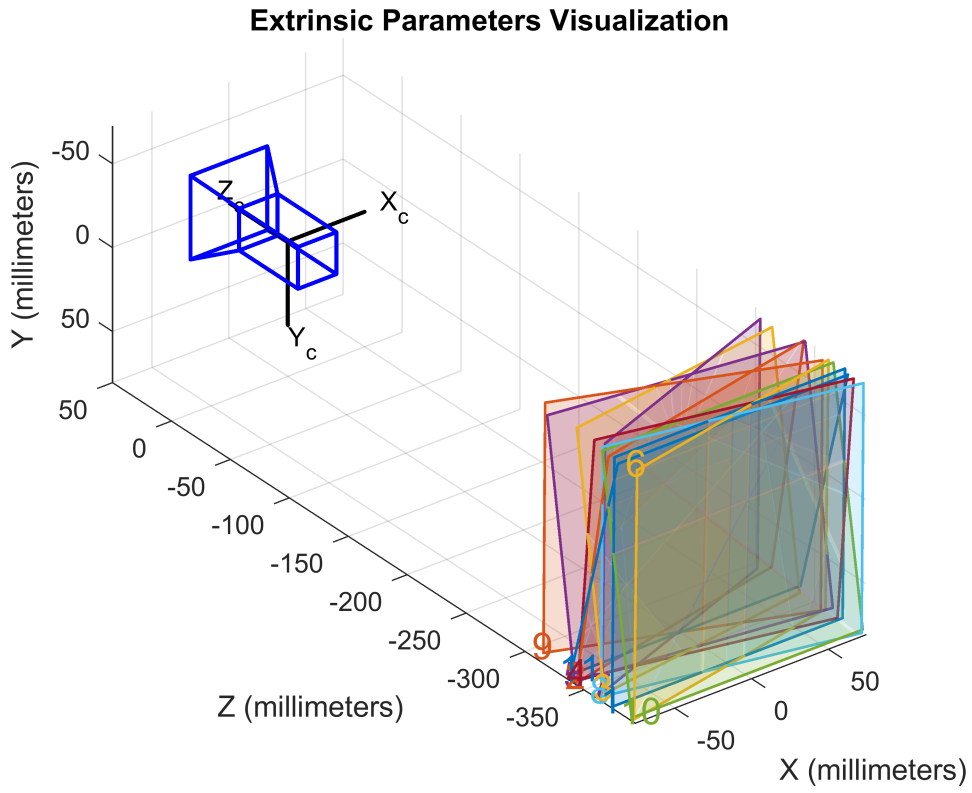
```
% Auto-generated by cameraCalibrator app on 20-Feb-2022
%------------------------------------------------------


% Define images to process
imageFileNames = {'C:\Users\DELL\OneDrive\Desktop\Spring\CV\cv\New folder\chess 1.jpeg',...
    'C:\Users\DELL\OneDrive\Desktop\Spring\CV\cv\New folder\chess 2.jpeg',...
    'C:\Users\DELL\OneDrive\Desktop\Spring\CV\cv\New folder\chess 3.jpeg',...
    'C:\Users\DELL\OneDrive\Desktop\Spring\CV\cv\New folder\chess 4.jpeg',...
    'C:\Users\DELL\OneDrive\Desktop\Spring\CV\cv\New folder\chess 5.jpeg',...
    'C:\Users\DELL\OneDrive\Desktop\Spring\CV\cv\New folder\chess 6.jpeg',...
    'C:\Users\DELL\OneDrive\Desktop\Spring\CV\cv\New folder\chess 7.jpeg',...
    'C:\Users\DELL\OneDrive\Desktop\Spring\CV\cv\New folder\chess 8.jpeg',...
    'C:\Users\DELL\OneDrive\Desktop\Spring\CV\cv\New folder\chess 9.jpeg',...
    'C:\Users\DELL\OneDrive\Desktop\Spring\CV\cv\New folder\chess 10.jpeg',...
    'C:\Users\DELL\OneDrive\Desktop\Spring\CV\cv\New folder\chess 11.jpeg',...
    };
% 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);

% Calibrate the camera using fisheye parameters
[cameraParams, imagesUsed, estimationErrors] = estimateFisheyeParameters(imagePoints, worldPoi
    [mrows, ncols], ...
    'EstimateAlignment', false, ...
    'WorldUnits', 'millimeters');

% View reprojection errors
h1=figure; showReprojectionErrors(cameraParams);
```




```
% Visualize pattern locations
h2=figure; showExtrinsics(cameraParams, 'CameraCentric');
```




```
% Display parameter estimation errors
displayErrors(estimationErrors, cameraParams);
```

```
            Standard Errors of Estimated Camera Parameters
            ----------------------------------------------

Intrinsics
----------
Mapping coefficients:    [-1245.0232 +/- -5.0174      -0.0001 +/- -0.0000      -0.0000 +/- 0.0000        0.0000 +/-
Distortion center (pixels):[  514.2935 +/- 1.1308      493.8862 +/- 1.1094  ]
Stretch matrix parameters:[    1.0000 +/- 0.0000        0.0000 +/- 0.0000        0.0000 +/- 0.0000  ]

Extrinsics
----------
Rotation vectors:
                         [    0.0114 +/- 0.0029        0.0131 +/- 0.0029       -1.5723 +/- 0.0003  ]
                         [    0.2860 +/- 0.0020        0.0024 +/- 0.0018       -1.5791 +/- 0.0003  ]
                         [    0.0210 +/- 0.0018       -0.2612 +/- 0.0019       -1.5500 +/- 0.0003  ]
                         [   -0.0737 +/- 0.0022        0.2218 +/- 0.0023       -1.5577 +/- 0.0004  ]
                         [   -0.2640 +/- 0.0020        0.0195 +/- 0.0018       -1.5774 +/- 0.0003  ]
                         [    0.0187 +/- 0.0021       -0.1969 +/- 0.0021       -0.0030 +/- 0.0003  ]
                         [    0.2901 +/- 0.0016       -0.2635 +/- 0.0016       -1.5571 +/- 0.0004  ]
                         [   -0.1453 +/- 0.0024        0.1650 +/- 0.0024       -1.5653 +/- 0.0004  ]
                         [   -0.2685 +/- 0.0016        0.2857 +/- 0.0016       -1.5520 +/- 0.0004  ]
                         [   -0.1334 +/- 0.0024       -0.1270 +/- 0.0024       -1.5683 +/- 0.0004  ]
                         [    0.2162 +/- 0.0020        0.2126 +/- 0.0020       -1.5620 +/- 0.0004  ]

Translation vectors (millimeters):
                         [  -75.5677 +/- 0.3412       79.8777 +/- 0.3356     -373.6116 +/- 1.5119  ]
                         [  -74.3810 +/- 0.3039       80.4873 +/- 0.3064     -341.7585 +/- 1.3533  ]
                         [  -74.4064 +/- 0.3187       76.3810 +/- 0.3121     -365.3639 +/- 1.3900  ]
```

```
[  -74.1271 +/- 0.3241       78.0597 +/- 0.3149     -342.9847 +/- 1.4068  ]
[  -74.2794 +/- 0.3117       80.5508 +/- 0.2973     -340.3271 +/- 1.3499  ]
[  -74.4641 +/- 0.3467      -69.9273 +/- 0.3430     -393.4595 +/- 1.5288  ]
[  -71.0626 +/- 0.3154       79.6397 +/- 0.3107     -359.4385 +/- 1.3386  ]
[  -74.0329 +/- 0.3400       79.6876 +/- 0.3277     -362.2431 +/- 1.4736  ]
[  -70.7853 +/- 0.2900       78.9165 +/- 0.2778     -309.1847 +/- 1.2840  ]
[  -75.6256 +/- 0.3486       78.8312 +/- 0.3377     -390.3298 +/- 1.5313  ]
[  -75.5683 +/- 0.3105       76.9353 +/- 0.3113     -338.4065 +/- 1.3851  ]
```

```
% For example, you can use the calibration data to remove effects of lens distortion.
undistortedImage = undistortFisheyeImage(originalImage, cameraParams.Intrinsics);

% See additional examples of how to use the calibration data.  At the prompt type:
% showdemo('MeasuringPlanarObjectsExample')
% showdemo('StructureFromMotionExample')
```