

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

%-----

% Define images to process

```
imageFileNames = {'C:\Users\pooja\OneDrive\Documents\MATLAB\CV\Image1.png',...  
    'C:\Users\pooja\OneDrive\Documents\MATLAB\CV\Image2.png',...  
    'C:\Users\pooja\OneDrive\Documents\MATLAB\CV\Image3.png',...  
    'C:\Users\pooja\OneDrive\Documents\MATLAB\CV\Image4.png',...  
    'C:\Users\pooja\OneDrive\Documents\MATLAB\CV\Image5.png',...  
    'C:\Users\pooja\OneDrive\Documents\MATLAB\CV\Image6.png',...  
    'C:\Users\pooja\OneDrive\Documents\MATLAB\CV\Image7.png',...  
    'C:\Users\pooja\OneDrive\Documents\MATLAB\CV\Image8.png',...  
    'C:\Users\pooja\OneDrive\Documents\MATLAB\CV\Image9.png',...  
    'C:\Users\pooja\OneDrive\Documents\MATLAB\CV\Image10.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

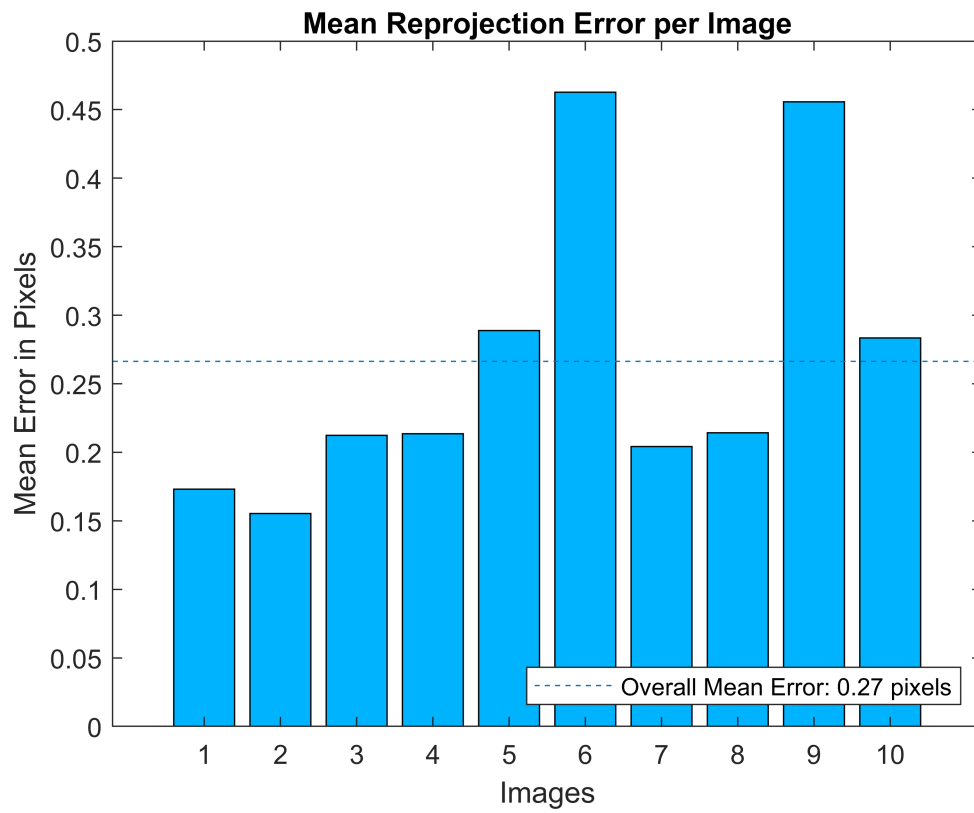
```
squareSize = 25; % in units of 'millimeters'  
worldPoints = generateWorldPoints(detector, 'SquareSize', squareSize);
```

% Calibrate the camera

```
[cameraParams, imagesUsed, estimationErrors] = estimateCameraParameters(imagePoints, worldPoints,  
    'EstimateSkew', false, 'EstimateTangentialDistortion', false, ...  
    'NumRadialDistortionCoefficients', 2, 'WorldUnits', 'millimeters', ...  
    'InitialIntrinsicMatrix', [], 'InitialRadialDistortion', [], ...  
    'ImageSize', [mrows, ncols]);
```

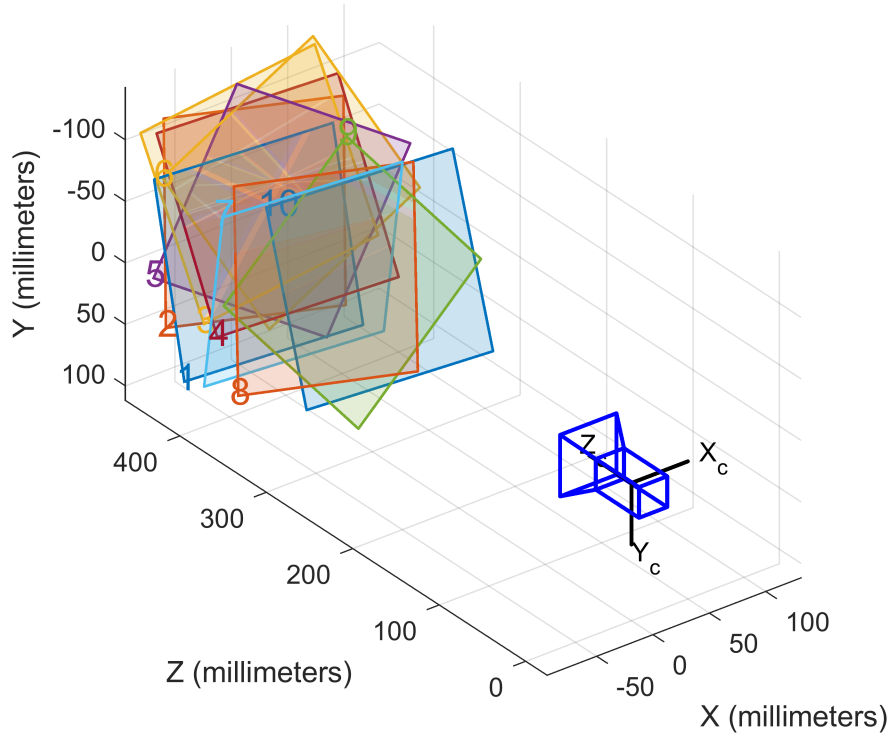
% View reprojection errors

```
h1=figure; showReprojectionErrors(cameraParams);
```



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

Extrinsic Parameters Visualization



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

Standard Errors of Estimated Camera Parameters

Intrinsics

```
-----
Focal length (pixels): [ 385.5608 +/- 5.4678    385.5531 +/- 5.6885 ]
Principal point (pixels): [ 343.6063 +/- 1.3959    155.1538 +/- 1.6810 ]
Radial distortion: [ -0.0995 +/- 0.0129    0.2816 +/- 0.0670 ]
```

Extrinsics

Rotation vectors:

```
[ -0.2756 +/- 0.0064    -0.1524 +/- 0.0042    -1.5420 +/- 0.0010 ]
[ -0.2828 +/- 0.0063    -0.1509 +/- 0.0038    -1.3729 +/- 0.0010 ]
[ -0.3190 +/- 0.0061    -0.1823 +/- 0.0043    -1.7086 +/- 0.0012 ]
[ -0.5045 +/- 0.0061    -0.3352 +/- 0.0038    -1.5238 +/- 0.0017 ]
[ -0.2533 +/- 0.0061     0.0998 +/- 0.0038    -1.0211 +/- 0.0007 ]
[ -0.2461 +/- 0.0058     0.0607 +/- 0.0035    -0.4710 +/- 0.0008 ]
[  0.1516 +/- 0.0043     0.0888 +/- 0.0037     0.0016 +/- 0.0006 ]
[ -0.2863 +/- 0.0061    -0.1599 +/- 0.0036    -1.3791 +/- 0.0009 ]
[ -0.2056 +/- 0.0047     0.1996 +/- 0.0049     0.9083 +/- 0.0006 ]
[ -0.3307 +/- 0.0056     0.1567 +/- 0.0037    -0.0156 +/- 0.0007 ]
```

Translation vectors (millimeters):

```
[ -90.2111 +/- 1.4450    69.0291 +/- 1.7562    399.8133 +/- 5.7281 ]
[ -94.0936 +/- 1.5007    31.4406 +/- 1.8191    416.0946 +/- 6.0046 ]
[ -66.5724 +/- 1.4688    34.1223 +/- 1.7852    407.6936 +/- 5.8467 ]
[ -71.1466 +/- 1.3938    33.4809 +/- 1.6874    386.5686 +/- 5.7189 ]
```

| | | |
|-----------------------|---------------------|-----------------------|
| [-89.0576 +/- 1.5652 | 2.1910 +/- 1.9018 | 436.8670 +/- 6.1669] |
| [-78.6263 +/- 1.5876 | -74.0717 +/- 1.9084 | 440.1598 +/- 6.3868] |
| [-78.6100 +/- 1.3328 | -73.9192 +/- 1.6353 | 370.4380 +/- 5.3841] |
| [-83.3922 +/- 1.2516 | 58.4011 +/- 1.5193 | 346.2548 +/- 5.0014] |
| [41.4546 +/- 1.3848 | -91.5153 +/- 1.6625 | 383.8948 +/- 5.5515] |
| [-17.8378 +/- 1.4343 | -47.3891 +/- 1.7389 | 399.2909 +/- 5.5863] |

% For example, you can use the calibration data to remove effects of lens distortion.

```
undistortedImage = undistortImage(originalImage, cameraParams);
```

% See additional examples of how to use the calibration data. At the prompt type:

```
% showdemo('MeasuringPlanarObjectsExample')
```

```
% showdemo('StructureFromMotionExample')
```