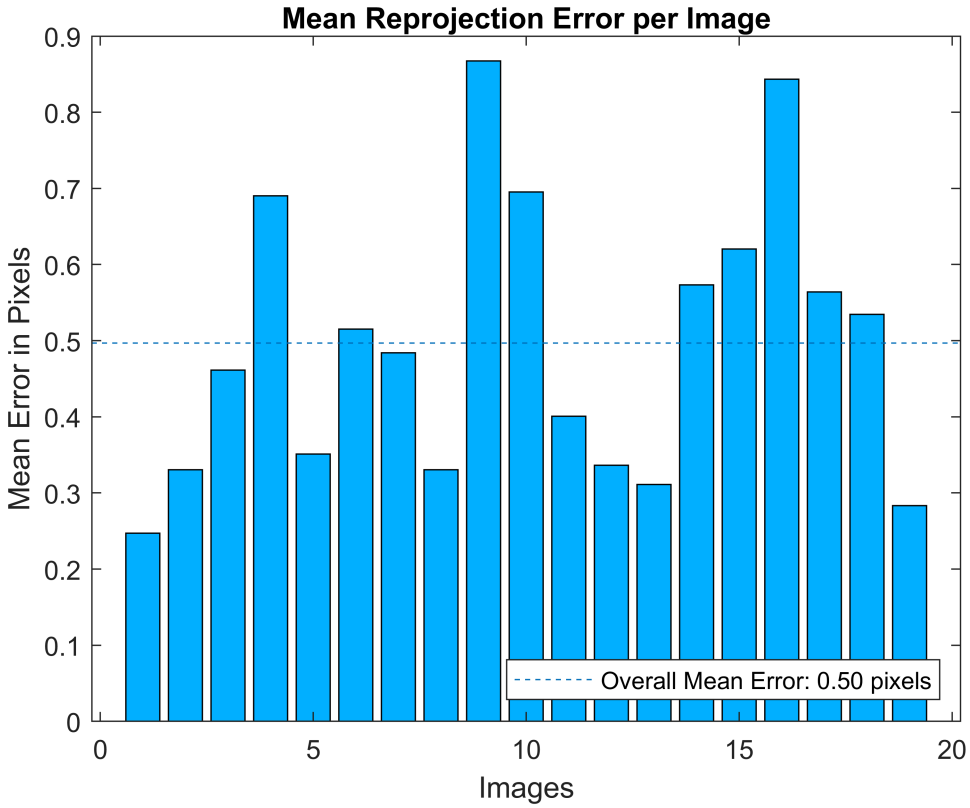
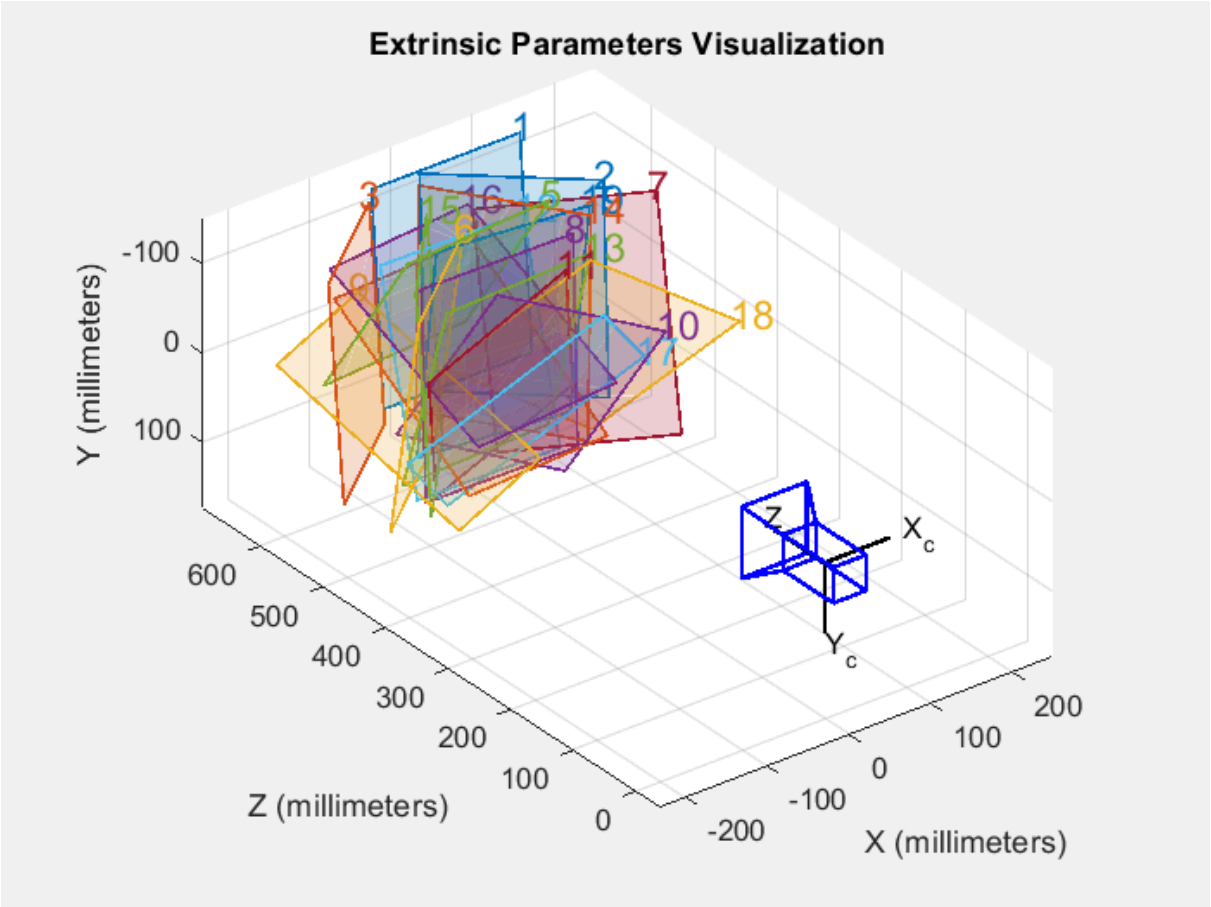
```matlab
% Auto-generated by cameraCalibrator app on 16-Apr-2021
%-------------------------------------------------------


% Define images to process
imageFileNames = {
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    'F:\UNIVERSITY\4TH_SEM\3 EN2550 - Fundamentals of Image Processing and Machine Vision\Compu
    };
% Detect checkerboards in images
[imagePoints, boardSize, imagesUsed] = detectCheckerboardPoints(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.

```matlab
imageFileNames = imageFileNames(imagesUsed);

% Read the first image to obtain image size
originalImage = imread(imageFileNames{1});
[mrows, ncols, ~] = size(originalImage);

% Generate world coordinates of the corners of the squares
squareSize = 30;  % in units of 'millimeters'
worldPoints = generateCheckerboardPoints(boardSize, squareSize);

% Calibrate the camera
[cameraParams, imagesUsed, estimationErrors] = estimateCameraParameters(imagePoints, worldPoint
    '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');
```




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

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

Intrinsics
----------
Focal length (pixels):   [  950.2569 +/- 1.2369      953.0092 +/- 1.2973  ]
Principal point (pixels):[  631.3673 +/- 0.8360      372.1849 +/- 0.7246  ]
Radial distortion:       [    0.0233 +/- 0.0053       -0.0725 +/- 0.0227  ]

Extrinsics
----------
Rotation vectors:
                         [   -0.0461 +/- 0.0043        0.0667 +/- 0.0043        1.5634 +/- 0.0004  ]
                         [    0.5741 +/- 0.0017        0.6283 +/- 0.0016        1.4713 +/- 0.0005  ]
                         [   -0.6167 +/- 0.0019       -0.4756 +/- 0.0020        1.5209 +/- 0.0005  ]
                         [   -0.8125 +/- 0.0013        0.6973 +/- 0.0013        1.4055 +/- 0.0006  ]
                         [    0.4280 +/- 0.0018       -0.5052 +/- 0.0018        1.5017 +/- 0.0005  ]
                         [   -0.4390 +/- 0.0017       -0.6215 +/- 0.0017        1.5408 +/- 0.0005  ]
                         [   -0.0644 +/- 0.0016        1.0093 +/- 0.0016        1.5767 +/- 0.0006  ]
                         [    0.0309 +/- 0.0028        0.0395 +/- 0.0028        1.5517 +/- 0.0003  ]
                         [   -1.0991 +/- 0.0012        0.4071 +/- 0.0012        1.2318 +/- 0.0006  ]
                         [    0.8040 +/- 0.0013       -0.0119 +/- 0.0013        1.7545 +/- 0.0005  ]
                         [    0.2549 +/- 0.0016       -0.3666 +/- 0.0014        1.1606 +/- 0.0004  ]
                         [   -0.2520 +/- 0.0027        0.1956 +/- 0.0028        1.5901 +/- 0.0004  ]
                         [    0.2312 +/- 0.0019       -0.2626 +/- 0.0019        1.5376 +/- 0.0004  ]
                         [    0.7247 +/- 0.0013        0.8910 +/- 0.0013        1.4414 +/- 0.0006  ]
```

```
                        [   -0.7524 +/- 0.0014       -0.5976 +/- 0.0014        1.5082 +/- 0.0006  ]
                        [   -0.8933 +/- 0.0010        0.8045 +/- 0.0011        1.3384 +/- 0.0007  ]
                        [    0.1613 +/- 0.0014       -0.6802 +/- 0.0015        2.3963 +/- 0.0005  ]
                        [    0.9277 +/- 0.0015       -0.1362 +/- 0.0016        1.9253 +/- 0.0006  ]
                        [    0.1202 +/- 0.0031       -0.0250 +/- 0.0031        1.5987 +/- 0.0004  ]

Translation vectors (millimeters):
                        [  145.8074 +/- 0.5970     -121.3390 +/- 0.5100      676.0938 +/- 1.0632  ]
                        [  137.9534 +/- 0.4795     -136.5588 +/- 0.4045      532.6326 +/- 0.8843  ]
                        [  -49.0962 +/- 0.6010     -118.0805 +/- 0.4947      662.8433 +/- 0.8580  ]
                        [   91.5265 +/- 0.5974      -18.9985 +/- 0.5295      681.7230 +/- 0.7658  ]
                        [   58.9465 +/- 0.4564     -149.9076 +/- 0.4223      515.6692 +/- 0.8454  ]
                        [  -17.1872 +/- 0.5047     -121.5132 +/- 0.4060      554.1500 +/- 0.7400  ]
                        [  248.7995 +/- 0.5252      -58.4581 +/- 0.4616      592.1318 +/- 0.8858  ]
                        [   84.9836 +/- 0.4572     -104.5741 +/- 0.3935      512.0889 +/- 0.8163  ]
                        [  -71.2107 +/- 0.5840      -29.6229 +/- 0.5057      651.9883 +/- 0.7292  ]
                        [   99.5131 +/- 0.3455      -49.0262 +/- 0.3006      382.8515 +/- 0.6776  ]
                        [   45.0222 +/- 0.4168      -95.7146 +/- 0.3741      471.7432 +/- 0.7280  ]
                        [  103.3608 +/- 0.5338      -74.0788 +/- 0.4627      615.4756 +/- 0.8418  ]
                        [   81.0972 +/- 0.4227     -100.5044 +/- 0.3817      477.4138 +/- 0.7397  ]
                        [  111.6614 +/- 0.4649     -112.8383 +/- 0.3921      518.3726 +/- 0.8436  ]
                        [    3.7700 +/- 0.5723      -95.8925 +/- 0.4794      639.8148 +/- 0.7774  ]
                        [   69.4059 +/- 0.5781      -75.5297 +/- 0.5072      659.4204 +/- 0.7176  ]
                        [  135.1463 +/- 0.3992       28.1838 +/- 0.3515      463.7275 +/- 0.6412  ]
                        [  198.8869 +/- 0.3756      -19.9028 +/- 0.3070      394.7524 +/- 0.7580  ]
                        [  127.1845 +/- 0.4900     -108.2221 +/- 0.4264      540.7813 +/- 0.8949  ]
```

```
% 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')
```