```
%Step 1: Read Images

arucoImage = imread('Frame0.jpg');
figure;
imshow(boxImage);
title('Image of aruco Marker');
```


```
sceneImage = imread('Frame1.jpg');
figure;
imshow(sceneImage);
title('Image of the wall');
```

Image of the wall

```
%Step 2: Detect Feature Points

arucoImageGS = rgb2gray(arucoImage)
```

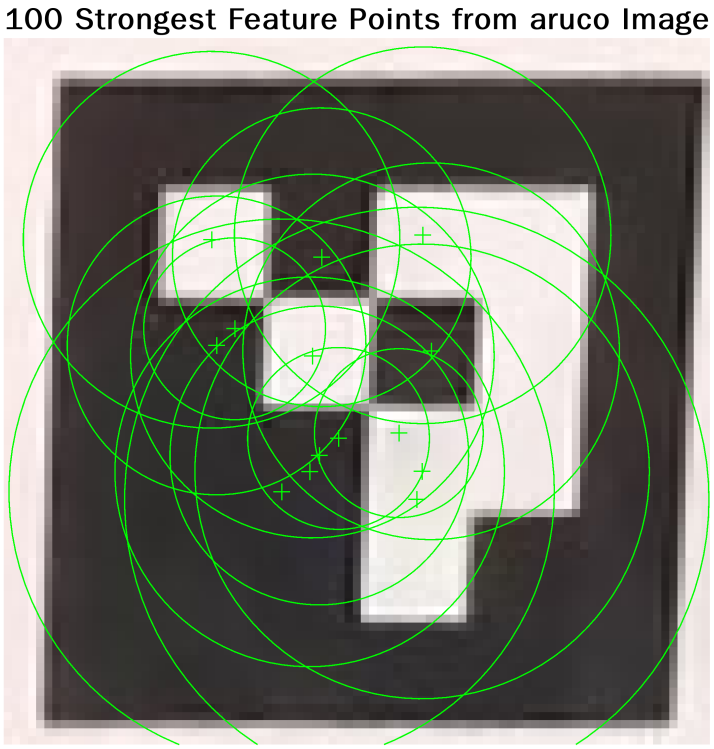
```
arucoImageGS = 87×87 uint8 matrix
   238   244   242   237   238   240   239   239   241   241   242   241   240 ···
   240   240   239   241   242   239   235   236   243   242   242   241   242
   242   238   239   243   243   239   240   246   245   243   242   242   243
   243   240   241   241   240   241   245   247   245   246   247   248   249
   241   242   242   240   241   242   225   196   193   197   203   205   205
   240   241   242   241   246   242   187   114    95   100   105   106   103
   241   241   241   240   249   243   166    66    46    49    51    50    46
   243   242   240   239   248   244   168    64    56    57    57    57    56
   242   239   241   241   248   243   154    68    56    56    56    56    56
   242   239   242   241   248   243   150    65    56    56    56    56    55
     ⋮
```

```
sceneImageGS = rgb2gray(sceneImage)
```

```
sceneImageGS = 480×848 uint8 matrix
   111   114   116   119   119   113   114   124   114   113   116   113   100 ···
   127   123   125   122   116   123   124   109   114   113   113   116   107
    94    88    65    59    84   104   109   116   113   113   115   112   114
    28    19    20    19    13    25    55    73   113   113   121   110   115
    24    25    22    19    18    16    18    25    62   107   118   101   120
    32    29    24    21    21    18    15    15    24    62   105   121   110
    38    30    27    28    23    20    18    17    12    16    55   104   117
    36    35    33    31    31    27    22    20    17    13    16    47   116
    15    20    35    43    47    28    26    15     9    16    18    10    61
```

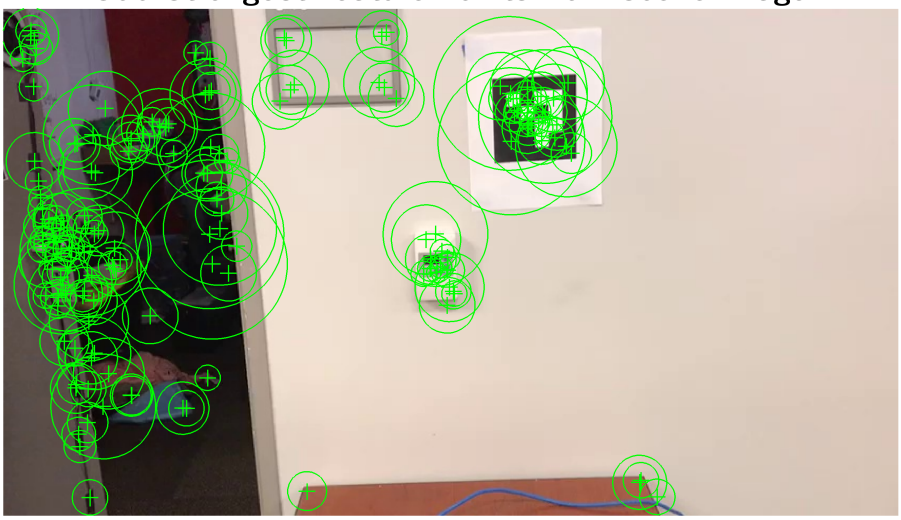
```
    30    17    20    35    47    41    32    25    17    16    13    12    29
     ⋮
     ⋮
```

```
arucoPoints = detectSURFFeatures(arucoImageGS);
scenePoints = detectSURFFeatures(sceneImageGS);

figure;
imshow(arucoImage);
title('100 Strongest Feature Points from aruco Image');
hold on;
plot(selectStrongest(arucoPoints, 100));
```



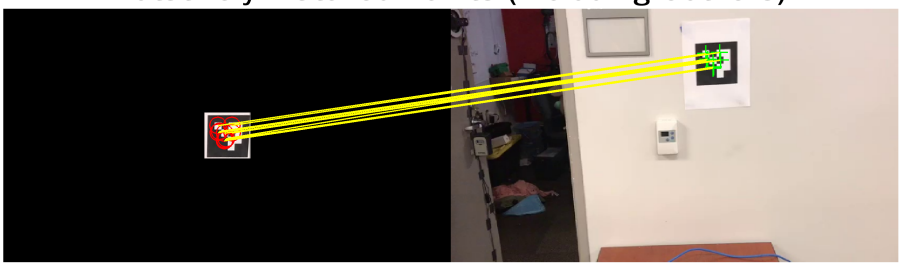

```
figure;
imshow(sceneImage);
title('300 Strongest Feature Points from Scene Image');
hold on;
plot(selectStrongest(scenePoints, 300));
```

300 Strongest Feature Points from Scene Image

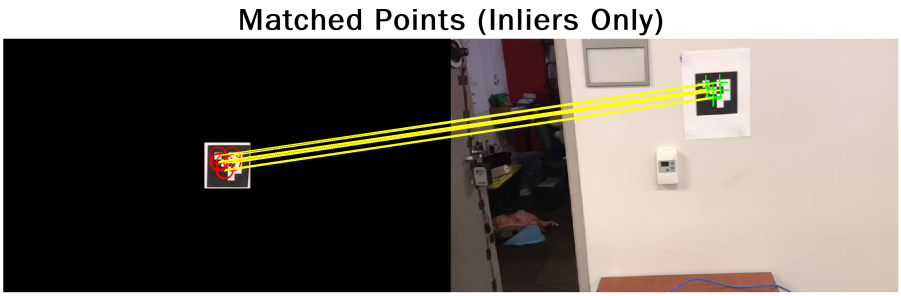


```
%Step 3: Extract Feature Descriptors

[arucoFeatures, arucoPoints] = extractFeatures(arucoImageGS, arucoPoints);
[sceneFeatures, scenePoints] = extractFeatures(sceneImageGS, scenePoints);

%Step 4: Find Putative Point Matches

arucoPairs = matchFeatures(arucoFeatures, sceneFeatures);

matchedarucoPoints = arucoPoints(arucoPairs(:, 1), :);
matchedScenePoints = scenePoints(arucoPairs(:, 2), :);
figure;
showMatchedFeatures(arucoImage, sceneImage, matchedarucoPoints, ...
    matchedScenePoints, 'montage');
title('Putatively Matched Points (Including Outliers)');
```

Putatively Matched Points (Including Outliers)

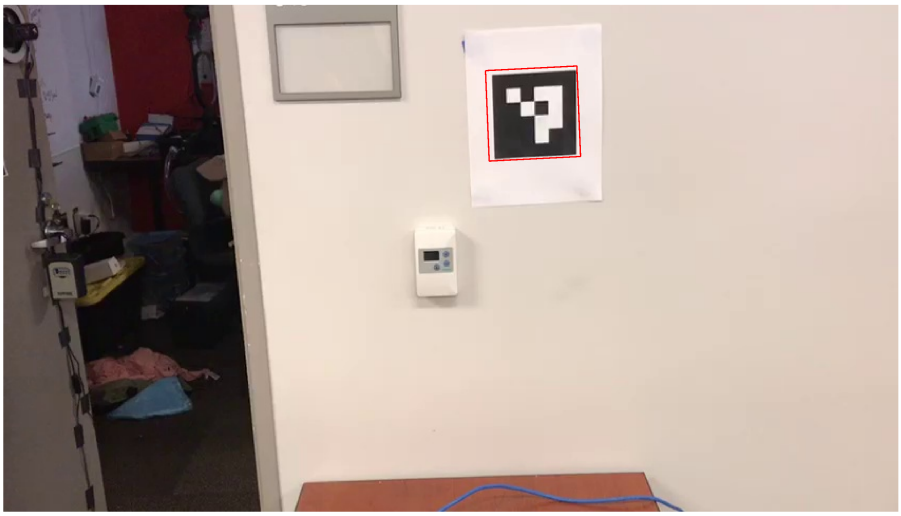


```
%Step 5: Locate the Object in the Scene Using Putative Matches

[tform, inlierIdx] = ...
    estimateGeometricTransform2D(matchedarucoPoints, matchedScenePoints, 'affine');
inlierarucoPoints   = matchedarucoPoints(inlierIdx, :);
inlierScenePoints = matchedScenePoints(inlierIdx, :);

figure;
showMatchedFeatures(arucoImage, sceneImage, inlierarucoPoints, ...
    inlierScenePoints, 'montage');
title('Matched Points (Inliers Only)');
```




```
arucoPolygon = [1, 1;...                                % top-left
        size(arucoImage, 2), 1;...                      % top-right
        size(arucoImage, 2), size(arucoImage, 1);... % bottom-right
        1, size(arucoImage, 1);...                      % bottom-left
        1, 1];                  % top-left again to close the polygon

newarucoPolygon = transformPointsForward(tform, arucoPolygon);

figure;
imshow(sceneImage);
hold on;
line(newarucoPolygon(:, 1), newarucoPolygon(:, 2), 'Color', 'r');
title('Detected aruco');
```

Detected aruco



```
%Step 7: Detect Another Object

controllerImage = imread('Frame3.jpg');
figure;
imshow(controllerImage);
title('Image of an controller');
```

Image of an controller

```
controllerImageGS = rgb2gray(controllerImage)
```

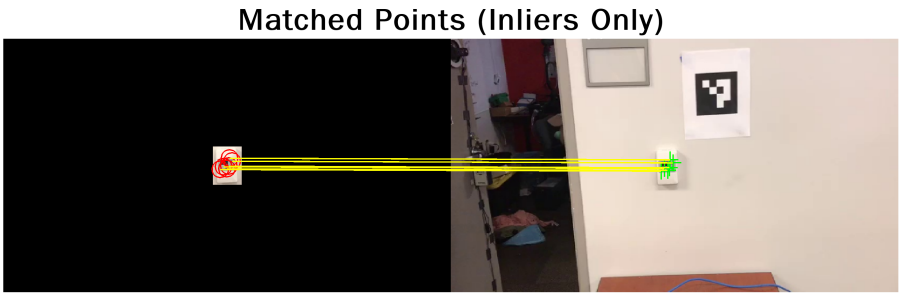
```
controllerImageGS = 73×55 uint8 matrix
   226   228   230   231   232   234   236   238   239   241   243   244   245 ···
   227   228   229   230   232   234   236   238   242   243   245   246   246
   227   228   229   230   232   234   236   238   240   241   243   244   244
   228   228   230   232   233   234   237   240   240   241   243   244   244
   229   230   232   234   234   234   238   243   245   246   247   248   248
   230   231   233   234   232   231   238   247   245   246   247   247   247
   228   228   229   228   223   224   235   247   243   244   245   245   244
   225   224   224   221   215   217   231   246   246   247   247   247   247
   220   220   220   222   209   215   244   243   248   248   248   248   249
   220   220   219   220   207   214   244   244   247   247   247   247   247
     ⋮
```

```
controllerPoints = detectSURFFeatures(controllerImageGS);

figure;
imshow(controllerImage);
hold on;
plot(selectStrongest(controllerPoints, 100));
title('100 Strongest Feature Points from controller Image');
```

100 Strongest Feature Points from controller Image

```
[controllerFeatures, controllerPoints] = extractFeatures(controllerImageGS, controllerP

controllerPairs = matchFeatures(controllerFeatures, sceneFeatures, 'MaxRatio', 0.9);

matchedcontrollerPoints = controllerPoints(controllerPairs(:, 1), :);
matchedScenePoints = scenePoints(controllerPairs(:, 2), :);
figure;
showMatchedFeatures(controllerImage, sceneImage, matchedcontrollerPoints, ...
    matchedScenePoints, 'montage');
title('Putatively Matched Points (Including Outliers)');
```


```
[tform, inliercontrollerPoints, inlierScenePoints] = ...
    estimateGeometricTransform(matchedcontrollerPoints, matchedScenePoints, 'affine');
figure;
showMatchedFeatures(controllerImage, sceneImage, inliercontrollerPoints, ...
    inlierScenePoints, 'montage');
title('Matched Points (Inliers Only)');
```




```
controllerPolygon = [1, 1;...                                    % top-left
        size(controllerImage, 2), 1;...                          % top-right
        size(controllerImage, 2), size(controllerImage, 1);...   % bottom-right
        1, size(controllerImage, 1);...                          % bottom-left
        1,1];                          % top-left again to close the polygon

newcontrollerPolygon = transformPointsForward(tform, controllerPolygon);

figure;
imshow(sceneImage);
hold on;
line(newarucoPolygon(:, 1), newarucoPolygon(:, 2), 'Color', 'r');
line(newcontrollerPolygon(:, 1), newcontrollerPolygon(:, 2), 'Color', 'g');
title('Detected Elephant and Box');
```

Detected Elephant and Box