

```

%Question 1
% DO NOT RUN THIS SECTION MULTIPLE TIMES. EXECUTE THIS SECTION ONLY ONCE
vidReader = VideoReader('C:\Users\aswin\Desktop\CV Assingment 3\Video_1.mp4','CurrentTime',1);
numFrames = vidReader.NumFrames;
n=numFrames;
for i = 1:n
    frames = read(vidReader,i);
    imwrite(frames,['Image' int2str(i), '.jpg']);
    imageIo(i)=image(frames);
end

```



```

Bag_1 = im2gray(imread('C:\Users\aswin\Desktop\CV Assingment 3\Images\Image2.jpg'));
Bag_2 = im2gray(imread('C:\Users\aswin\Desktop\CV Assingment 3\Images\Image309.jpg'));
thisFrame = Bag_2;
imshow(thisFrame);
axis on;
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.
h_rect = drawrectangle('Label','PICK YOUR REGION OF INTEREST','Color',[1 1 1]);% this helps us to draw over the frame

% Rectangle position is given as [x, y, width, height]
pos_rect = h_rect.Position();
% Round off so the coordinates can be used as indices
pos_rect = round(pos_rect);
close(gcf)
thisFrame = imcrop(thisFrame,pos_rect);
Bag_2 = thisFrame

```

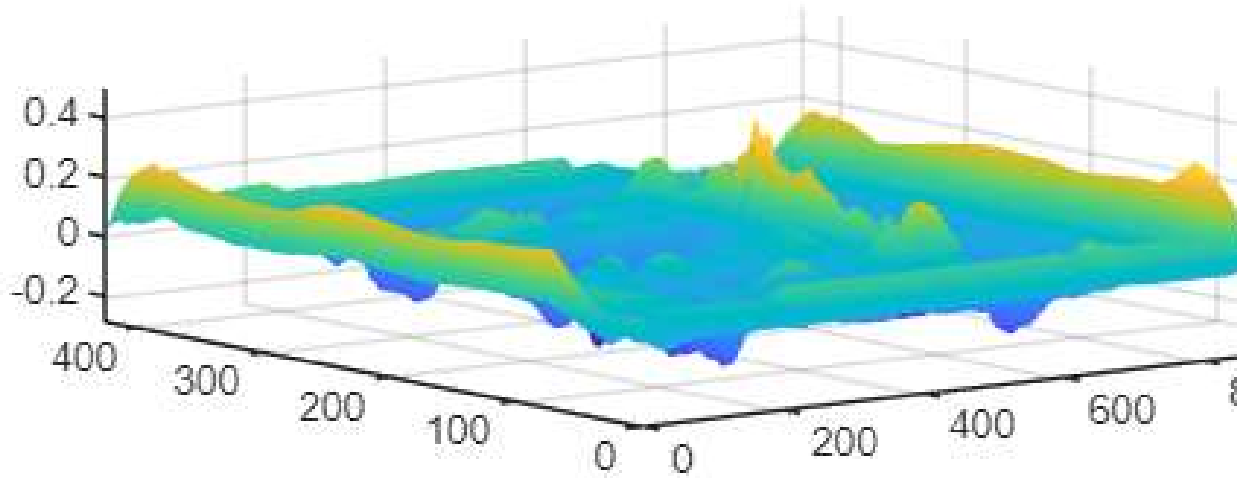
Bag_2 = 79x250 uint8 matrix

94	83	94	87	78	40	48	42	36	37	59	68	69	79	85	80	77	70	70	72	75	7
77	79	89	65	57	43	32	52	44	36	57	54	73	78	72	93	86	75	77	77	74	7
73	70	83	84	86	72	51	46	37	32	53	58	60	56	59	79	84	72	71	75	81	8
71	81	68	67	55	41	35	34	51	33	49	64	48	41	59	73	90	81	74	73	79	7
73	73	61	85	83	71	57	30	45	35	40	60	40	43	74	77	92	88	81	71	67	6
72	75	56	71	66	59	51	29	45	40	34	48	39	56	88	77	78	78	79	69	54	5
68	70	68	86	96	94	76	45	37	48	39	45	52	74	91	76	64	63	72	67	49	4
65	79	73	58	61	64	58	54	41	47	46	44	67	83	79	72	57	55	61	64	55	3
65	64	77	66	83	90	71	60	22	34	43	39	71	80	60	66	54	51	52	61	65	4
64	65	65	68	76	81	76	64	52	39	34	49	42	56	77	51	39	49	62	56	30	2

```
montage({Bag_1,Bag_2})
```



```
c = normxcorr2(Bag_2,Bag_1);
surf(c)
shading flat
```



```
[ypeak,xpeak] = find(c==max(c(:)));
yoffSet = ypeak-size(Bag_2,1);
xoffSet = xpeak-size(Bag_2,2);

imshow(Bag_1)
drawrectangle(gca,'Position',[xoffSet,yoffSet,size(Bag_2,2),size(Bag_2,1)], ...
    'FaceAlpha',0);
```



```
% QUESTION 5 - Part 1
```

```
n = vidReader.NumFrames
```

```
n = 319
```

```
opticFlow = opticalFlowHS
```

```
opticFlow =  
    opticalFlowHS with properties:
```

```
        Smoothness: 1  
        MaxIteration: 10  
        VelocityDifference: 0
```

```
referenceFrame = 1
```

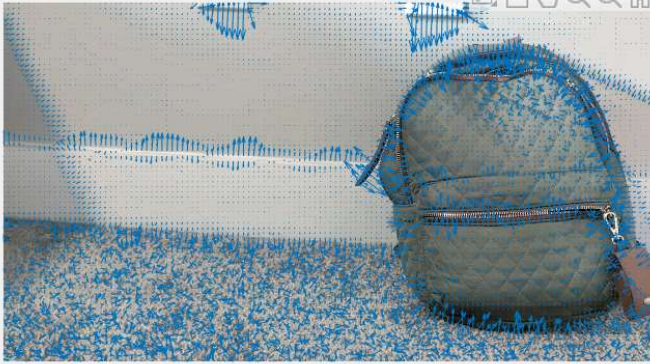
```
referenceFrame = 1
```

```
stepFrame = 1
```

```
stepFrame = 1
```

```
h = figure;  
movegui(h);  
hViewPanel = uipanel(h,'Position',[0 0 1 1],'Title','Plot of Optical Flow Vectors');  
hPlot = axes(hViewPanel);  
  
for frame = referenceFrame:stepFrame:n  
    frameRGB = read(vidReader, frame);  
    frameGray = im2gray(frameRGB);  
    flow = estimateFlow(opticFlow,frameGray);  
    imshow(frameRGB)  
    hold on  
    plot(flow,'DecimationFactor',[5 5],'ScaleFactor',60,'Parent',hPlot);  
    hold off  
end
```

Plot of Optical Flow Vectors



Plot of Optical Flow Vectors



% QUESTION 5 - Part 2

```
opticFlow = opticalFlowHS
```

```
opticFlow =  
    opticalFlowHS with properties:  
        Smoothness: 1  
        MaxIteration: 10  
        VelocityDifference: 0
```

```
referenceFrame = 1
```

```
referenceFrame = 1
```

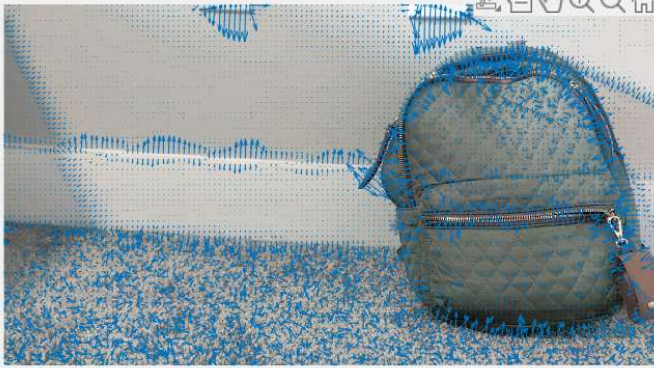
```
stepFrame = 11
```

```
stepFrame = 11
```

```
h = figure;  
movegui(h);  
hViewPanel = uipanel(h,'Position',[0 0 1 1],'Title','Plot of Optical Flow Vectors');  
hPlot = axes(hViewPanel);  
  
for frame = referenceFrame:stepFrame:n  
    frameRGB = read(vidReader, frame);  
    frameGray = im2gray(frameRGB);  
    flow = estimateFlow(opticFlow,frameGray);  
    imshow(frameRGB)  
    hold on  
    plot(flow,'DecimationFactor',[5 5],'ScaleFactor',60,'Parent',hPlot);  
    hold off
```

end

Plot of Optical Flow Vectors



Plot of Optical Flow Vectors



% QUESTION 5 - Part 3

```
opticFlow = opticalFlowHS
```

```
opticFlow =  
    opticalFlowHS with properties:  
  
        Smoothness: 1  
        MaxIteration: 10  
        VelocityDifference: 0
```

```
referenceFrame = 1
```

```
referenceFrame = 1
```

```
stepFrame = 31
```

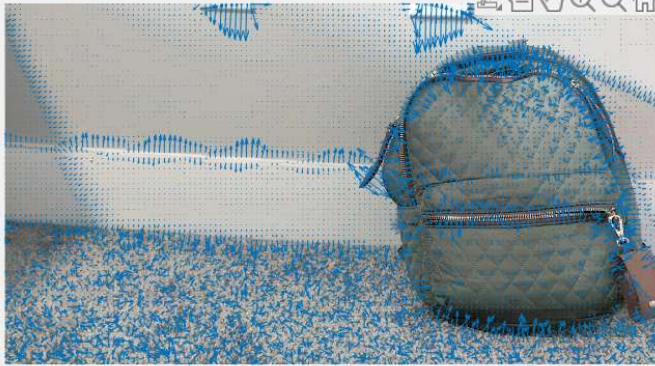
```
stepFrame = 31
```

```
h = figure;  
movegui(h);  
hViewPanel = uipanel(h,'Position',[0 0 1 1],'Title','Plot of Optical Flow Vectors');  
hPlot = axes(hViewPanel);  
  
for frame = referenceFrame:stepFrame:n  
    frameRGB = read(vidReader, frame);  
    frameGray = im2gray(frameRGB);  
    flow = estimateFlow(opticFlow,frameGray);  
    imshow(frameRGB)  
    hold on  
    plot(flow,'DecimationFactor',[5 5],'ScaleFactor',60,'Parent',hPlot);  
end
```



```
hold off  
end
```

Plot of Optical Flow Vectors



Plot of Optical Flow Vectors



```
% QUESTION 6 - images from Question 1
```

```
bI = imread('C:\Users\aswin\Desktop\CV Assingment 3\Images\Image309.jpg');  
boxImage = rgb2gray(bI);  
figure;  
imshow(boxImage);  
title('Image of a Box');
```

Image of a Box



```
sI = imread('C:\Users\aswin\Desktop\CV Assingment 3\Images\Image1.jpg');  
sceneImage = rgb2gray(sI);  
figure;
```

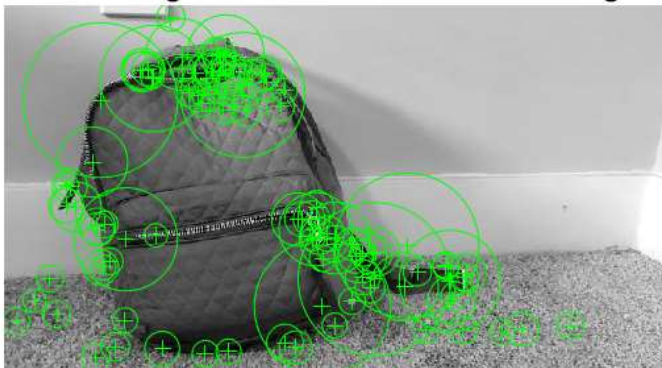
```
imshow(sceneImage);  
title('Image of a scene');
```

Image of a scene



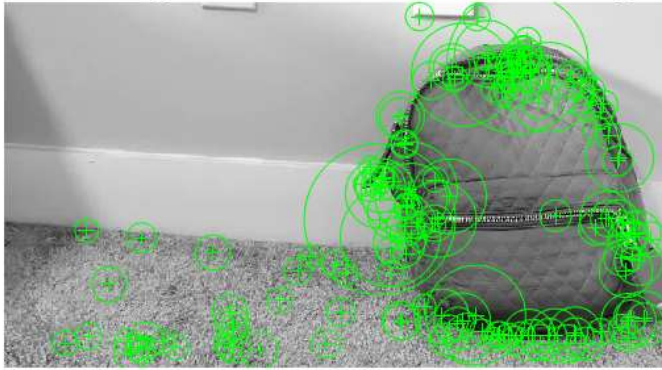
```
boxPoints = detectSURFFeatures(boxImage);  
  
scenePoints = detectSURFFeatures(sceneImage);  
figure;  
imshow(boxImage);  
title('100 Strongest Feature Points from Box Image');  
hold on;  
plot(selectStrongest(boxPoints, 100));
```

100 Strongest Feature Points from Box Image



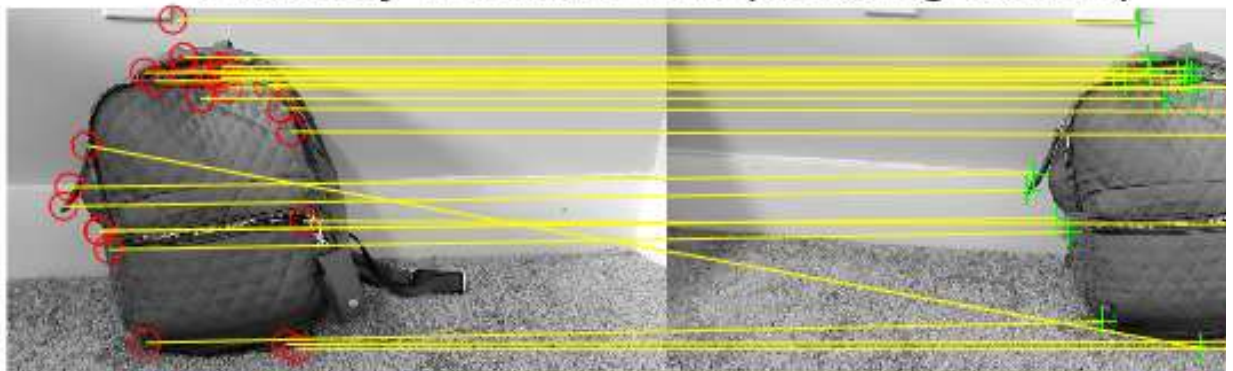
```
figure;  
imshow(sceneImage);  
title('300 Strongest Feature Points from Scene Image');  
hold on;  
plot(selectStrongest(scenePoints, 300));
```

300 Strongest Feature Points from Scene Image



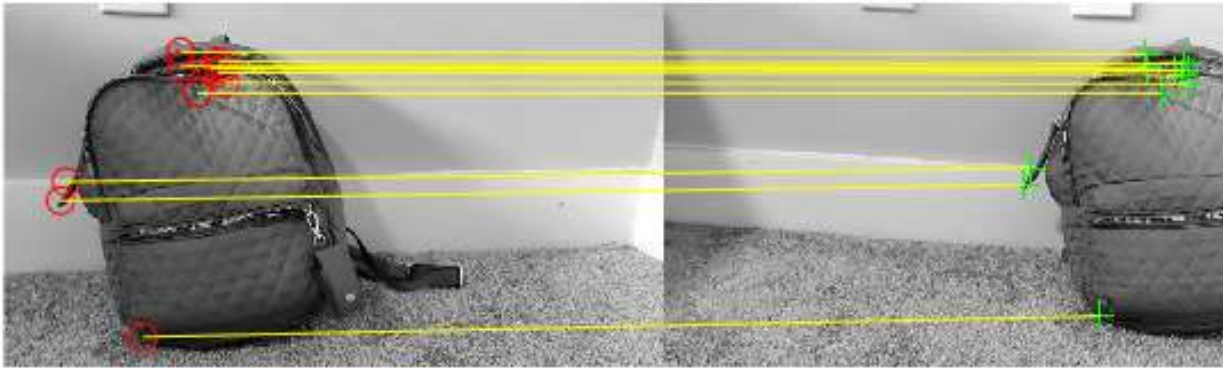
```
[boxFeatures, boxPoints] = extractFeatures(boxImage, boxPoints);
[sceneFeatures, scenePoints] = extractFeatures(sceneImage, scenePoints);
boxPairs = matchFeatures(boxFeatures, sceneFeatures);
matchedBoxPoints = boxPoints(boxPairs(:, 1), :);
matchedScenePoints = scenePoints(boxPairs(:, 2), :);
figure;
showMatchedFeatures(boxImage, sceneImage, matchedBoxPoints, ...
    matchedScenePoints, 'montage');
title('Putatively Matched Points (Including Outliers)');
```

Putatively Matched Points (Including Outliers)



```
[tform, inlierIdx] = ...
    estimateGeometricTransform2D(matchedBoxPoints, matchedScenePoints, 'affine');
inlierBoxPoints = matchedBoxPoints(inlierIdx, :);
inlierScenePoints = matchedScenePoints(inlierIdx, :);
figure;
showMatchedFeatures(boxImage, sceneImage, inlierBoxPoints, ...
    inlierScenePoints, 'montage');
title('Matched Points (Inliers Only)');
```


Matched Points (Inliers Only)



```
boxPolygon = [1, 1;...           % top-left
              size(boxImage, 2), 1;... % top-right
              size(boxImage, 2), size(boxImage, 1);... % bottom-right
              1, size(boxImage, 1);... % bottom-left
              1, 1];           % top-left again to close the polygon
newBoxPolygon = transformPointsForward(tform, boxPolygon)
```

```
newBoxPolygon = 5x2 single matrix
    300.8275   -3.3322
    926.6435    29.6301
    915.0709   355.6786
    289.2549   322.7163
    300.8275   -3.3322
```

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

Detected Box



% QUESTION 7 - Cutlery Images

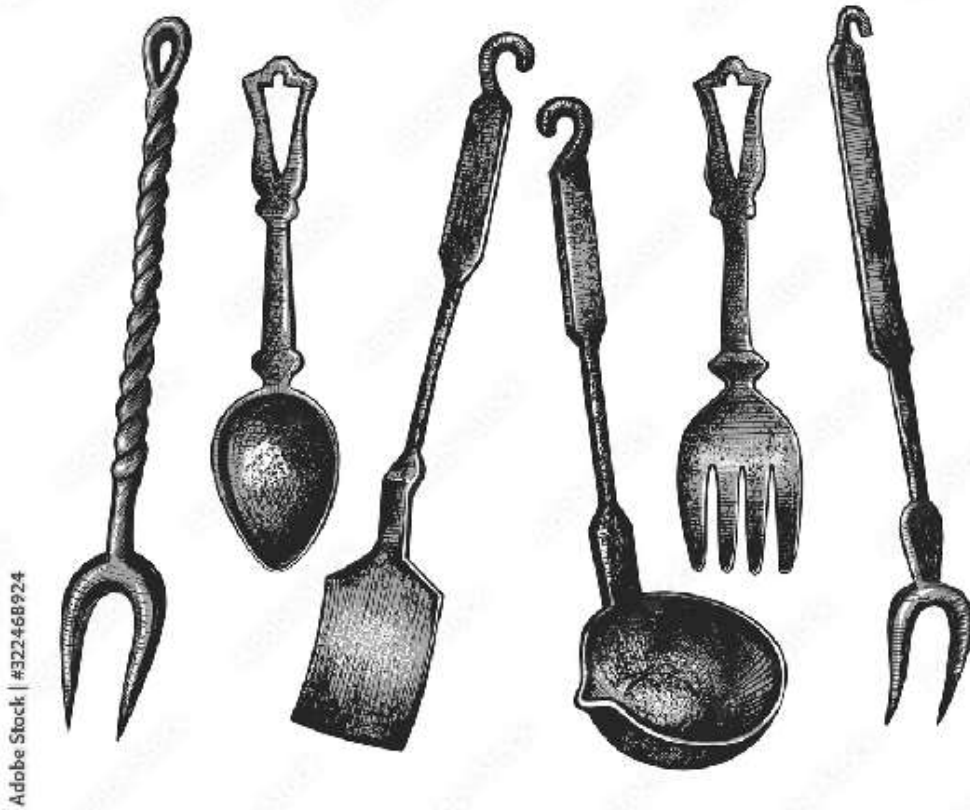
```
bI = imread('C:\Users\aswin\Desktop\CV Assingment 3\ladle.jpg');
boxImage = rgb2gray(bI);
figure;
imshow(boxImage);
title('Image of a fork');
```

Image of a fork



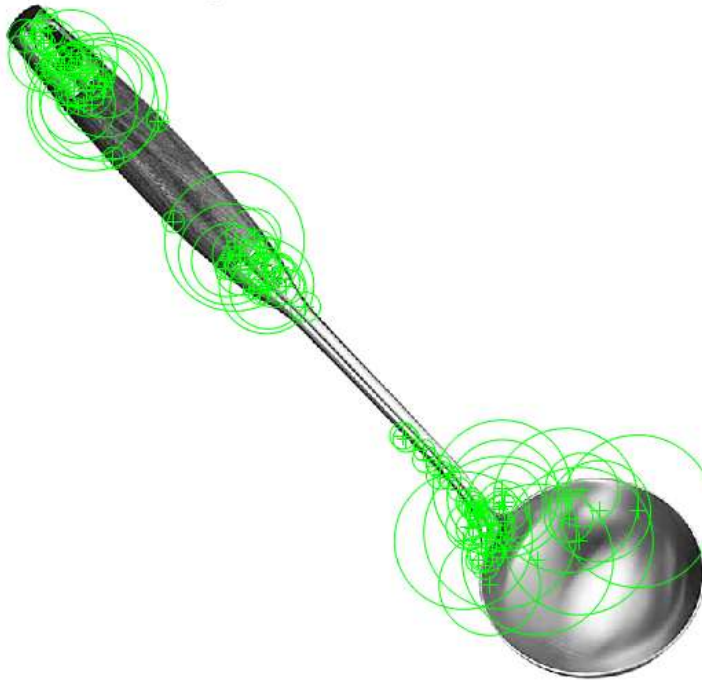
```
sI = imread('C:\Users\aswin\Desktop\CV Assingment 3\Cutlery.jpg');  
sceneImage = rgb2gray(sI);  
figure;  
imshow(sceneImage);  
title('Image of Cutlery');
```

Image of Cutlery



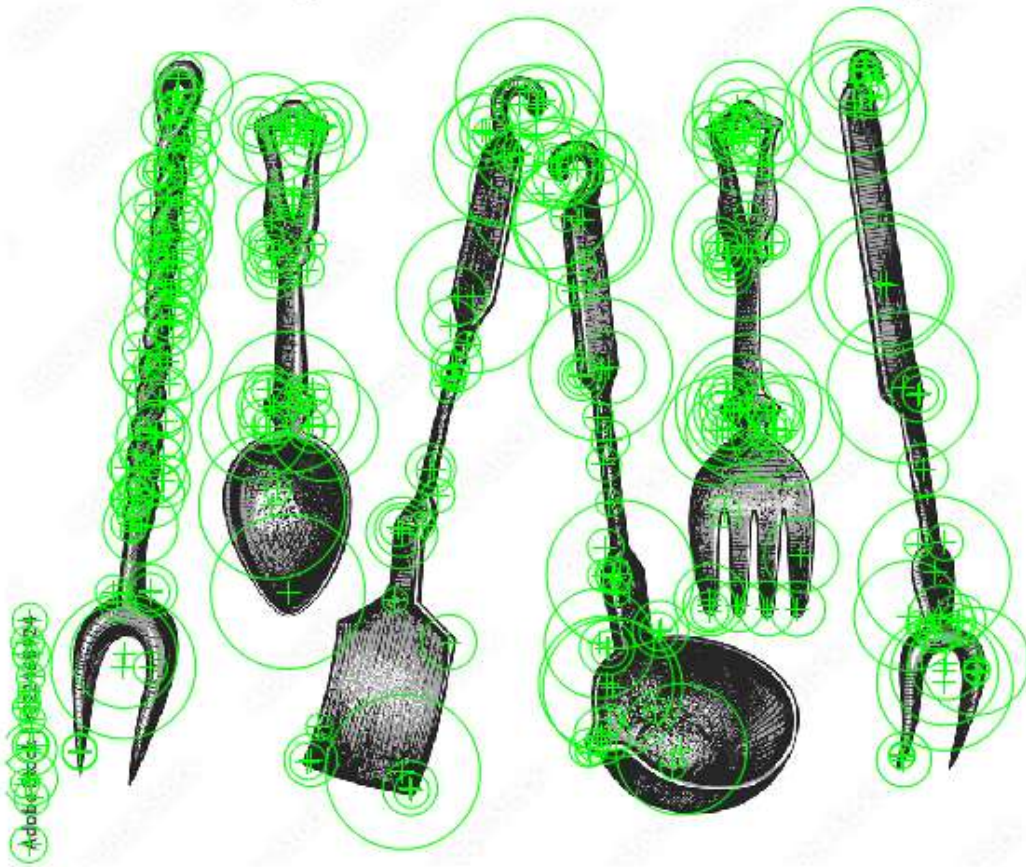
```
boxPoints = detectSURFFeatures(boxImage);  
  
scenePoints = detectSURFFeatures(sceneImage);  
figure;  
imshow(boxImage);  
title('100 Strongest Feature Points from Knife');  
hold on;  
plot(selectStrongest(boxPoints, 100));
```

100 Strongest Feature Points from Knife



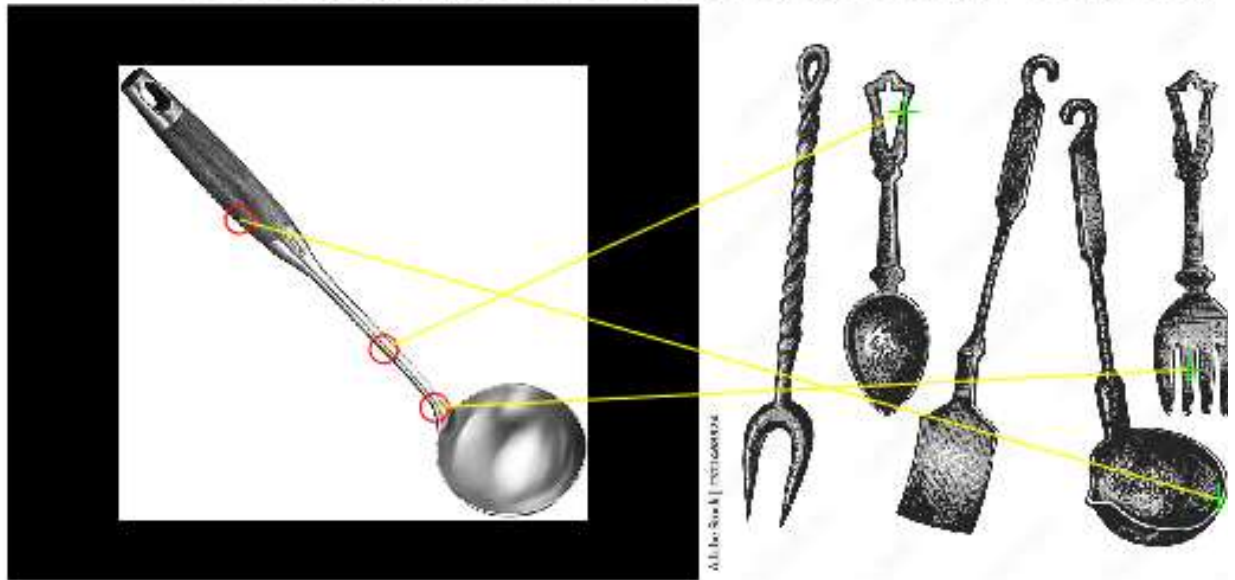
```
figure;  
imshow(sceneImage);  
title('300 Strongest Feature Points from Cutlery');  
hold on;  
plot(selectStrongest(scenePoints, 300));
```

300 Strongest Feature Points from Cutlery



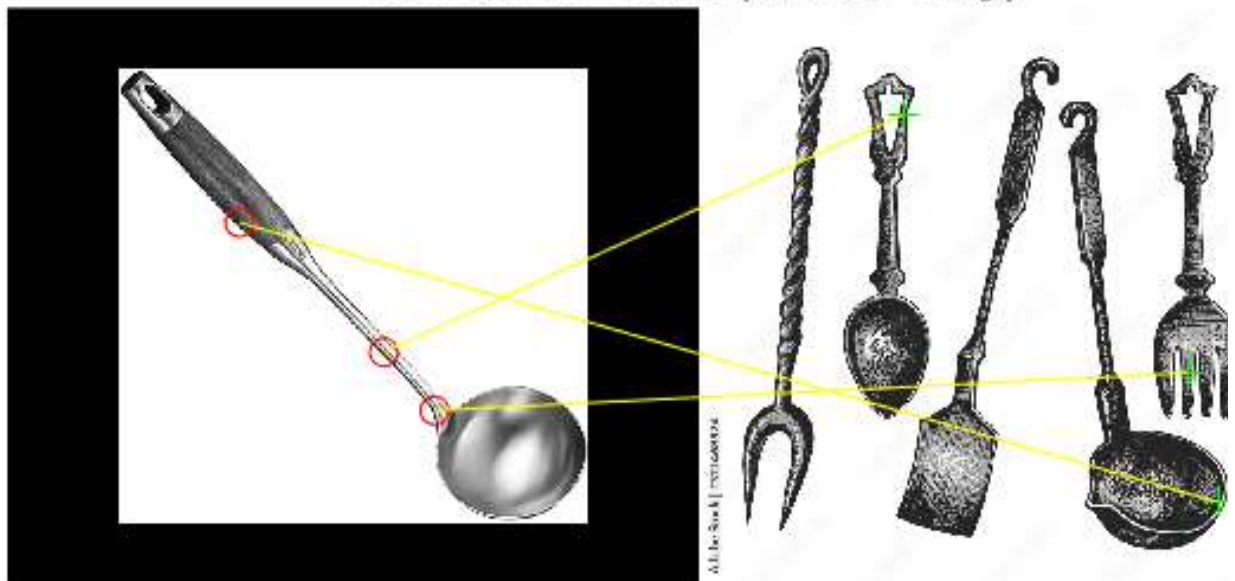
```
[boxFeatures, boxPoints] = extractFeatures(boxImage, boxPoints);  
[sceneFeatures, scenePoints] = extractFeatures(sceneImage, scenePoints);  
boxPairs = matchFeatures(boxFeatures, sceneFeatures);  
matchedBoxPoints = boxPoints(boxPairs(:, 1), :);  
matchedScenePoints = scenePoints(boxPairs(:, 2), :);  
figure;  
showMatchedFeatures(boxImage, sceneImage, matchedBoxPoints, ...  
    matchedScenePoints, 'montage');  
title('Putatively Matched Points (Including Outliers)');
```


Putatively Matched Points (including Outliers)



```
[tform, inlierIdx] = estimateGeometricTransform2D(matchedBoxPoints, matchedScenePoints, 'affine');
inlierBoxPoints = matchedBoxPoints(inlierIdx, :);
inlierScenePoints = matchedScenePoints(inlierIdx, :);
figure;
showMatchedFeatures(boxImage, sceneImage, inlierBoxPoints, ...
    inlierScenePoints, 'montage');
title('Matched Points (Inliers Only)');
```

Matched Points (Inliers Only)



```
boxPolygon = [1, 1;... % top-left
    size(boxImage, 2), 1;... % top-right
    size(boxImage, 2), size(boxImage, 1);... % bottom-right
    1, size(boxImage, 1);... % bottom-left
    1, 1]; % top-left again to close the polygon
newBoxPolygon = transformPointsForward(tform, boxPolygon)
```

newBoxPolygon = 5x2 single matrix

```
104 ×
-0.0860 -0.0801
-2.1107 -2.1286
-0.0905 -0.1187
1.9341 1.9298
-0.0860 -0.0801
```

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

