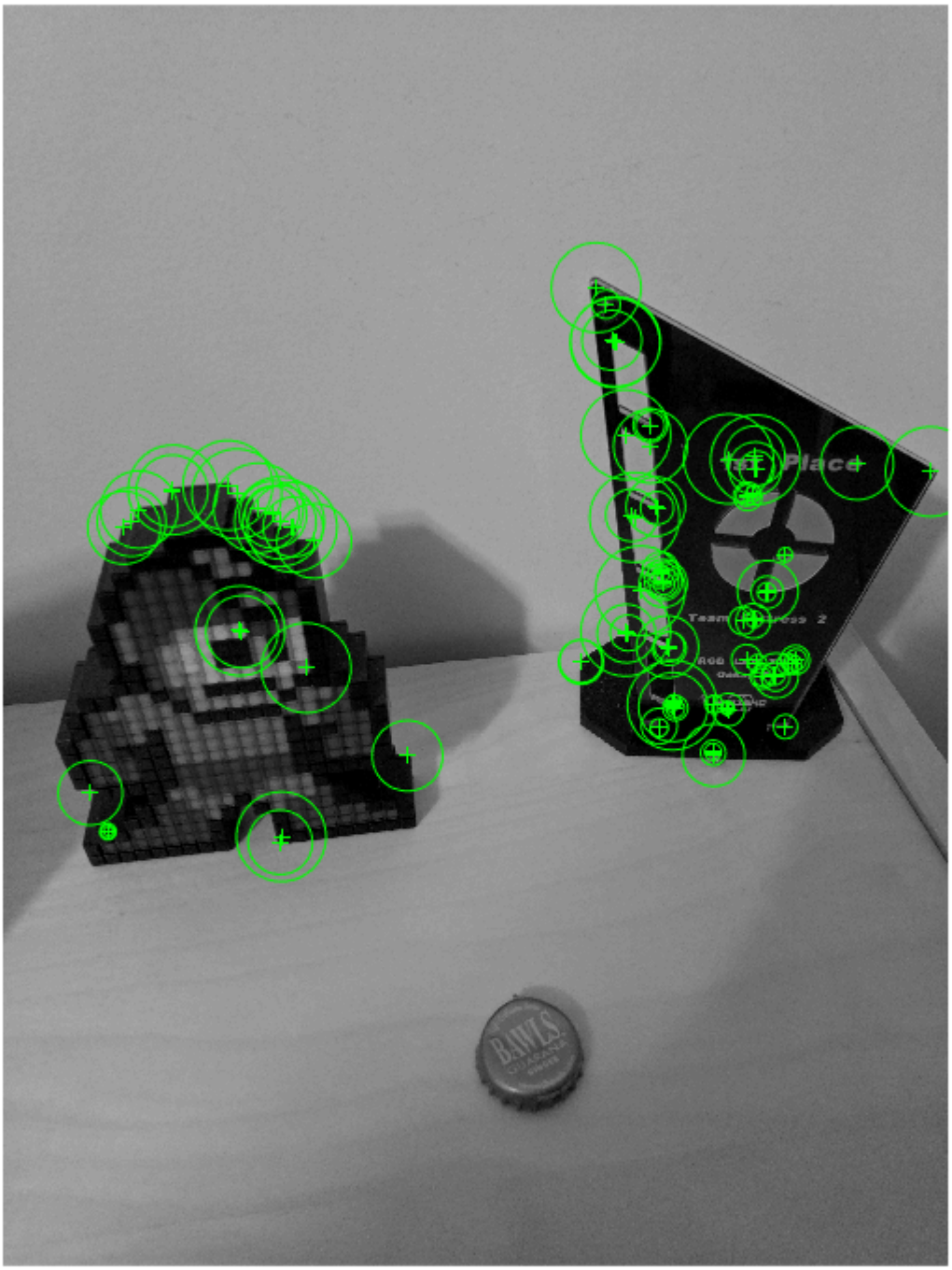
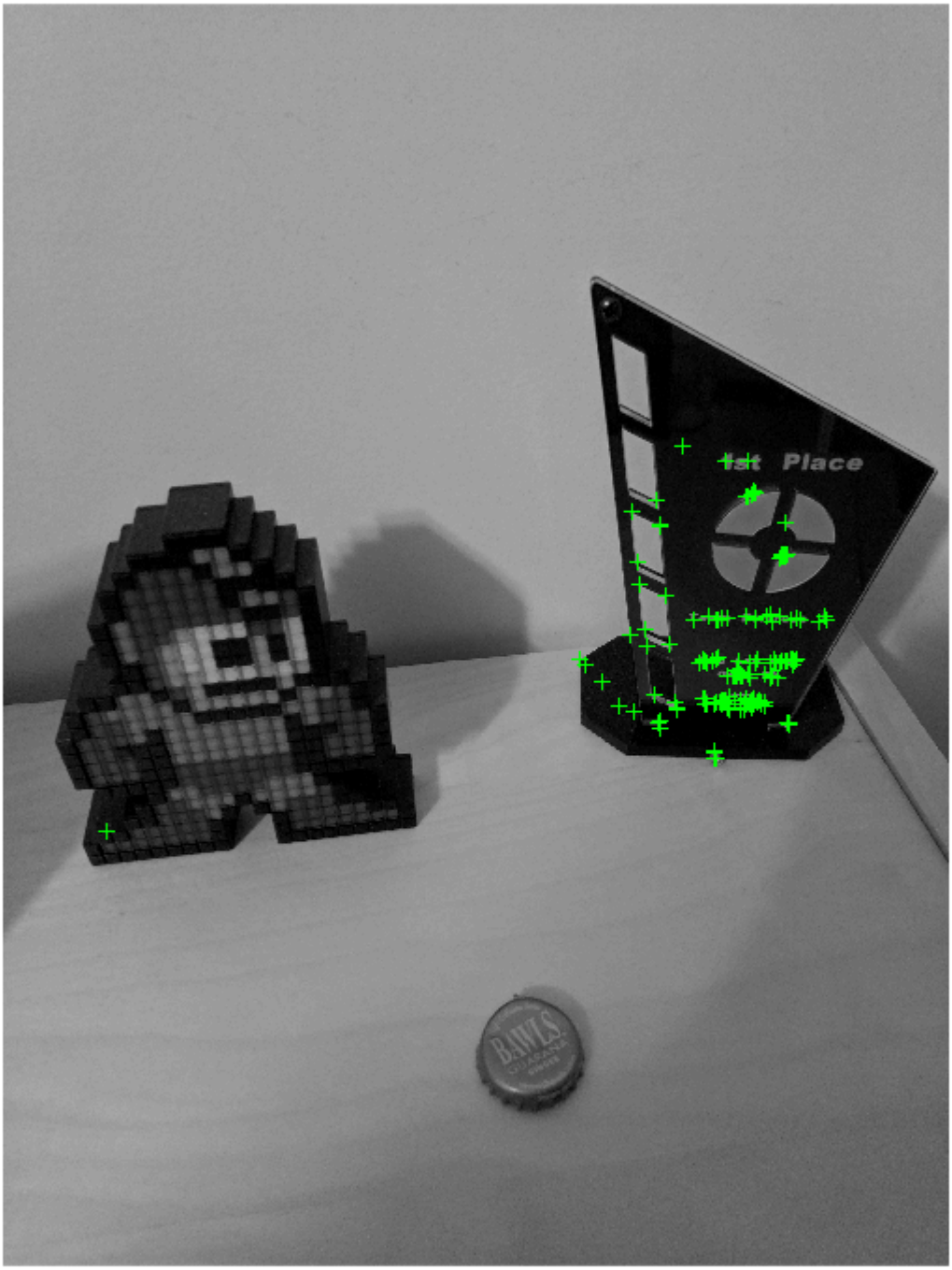
```
addpath 'D:'\homework\'checkerboard pics'\
%1
I = imread('megaman3.jpg');
I = rgb2gray(I);
%brisk
points = detectBRISKFeatures(I);
imshow(I); hold on;
plot(points.selectStrongest(100));
hold off
```



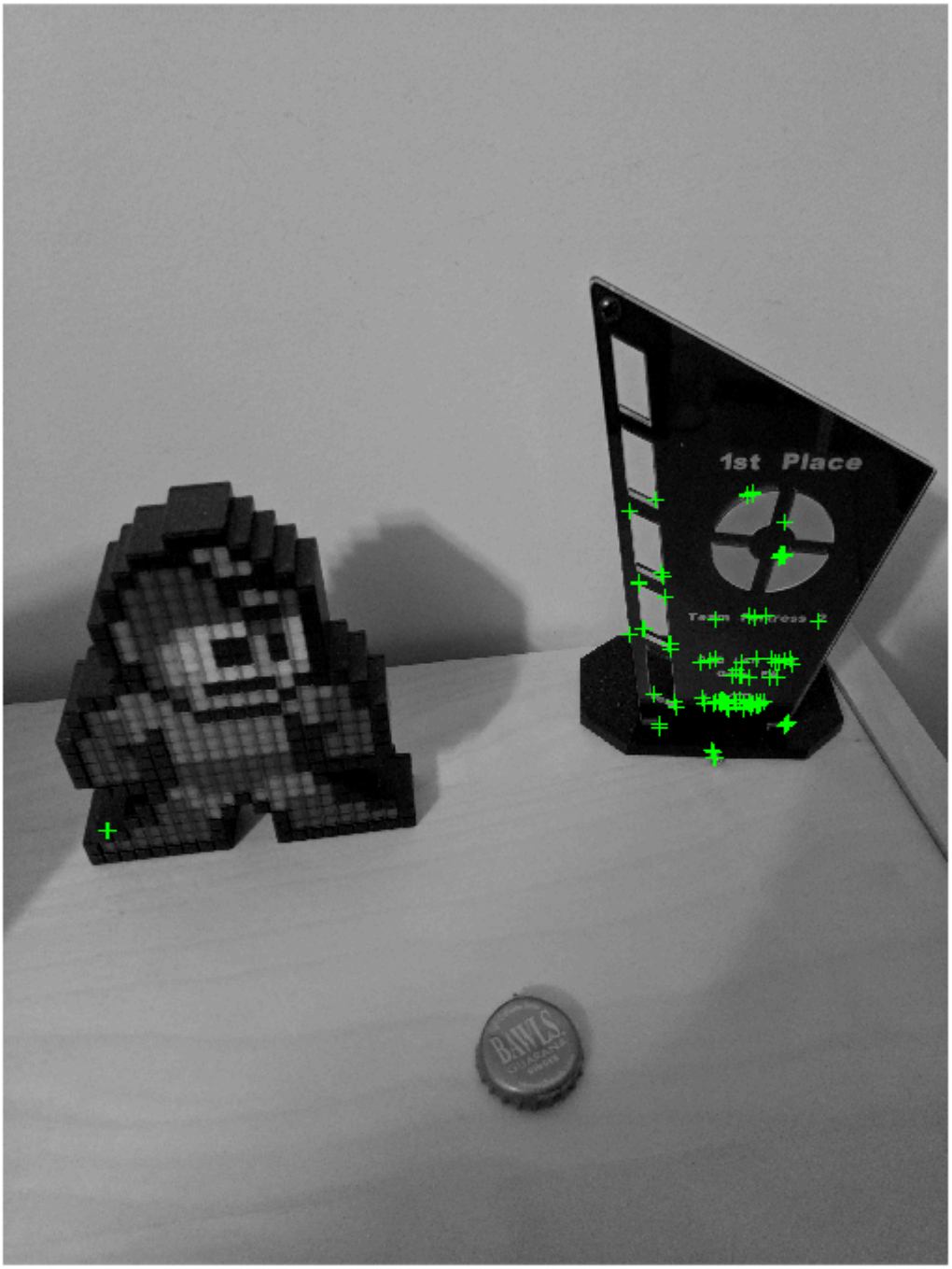
```
%a couple corners mistaken on the body of megaman, but mostly accurate.
%does not notice bottle cap

%fast
corners = detectFASTFeatures(I);
imshow(I); hold on;
plot(corners.selectStrongest(1000));
```

```
hold off
```



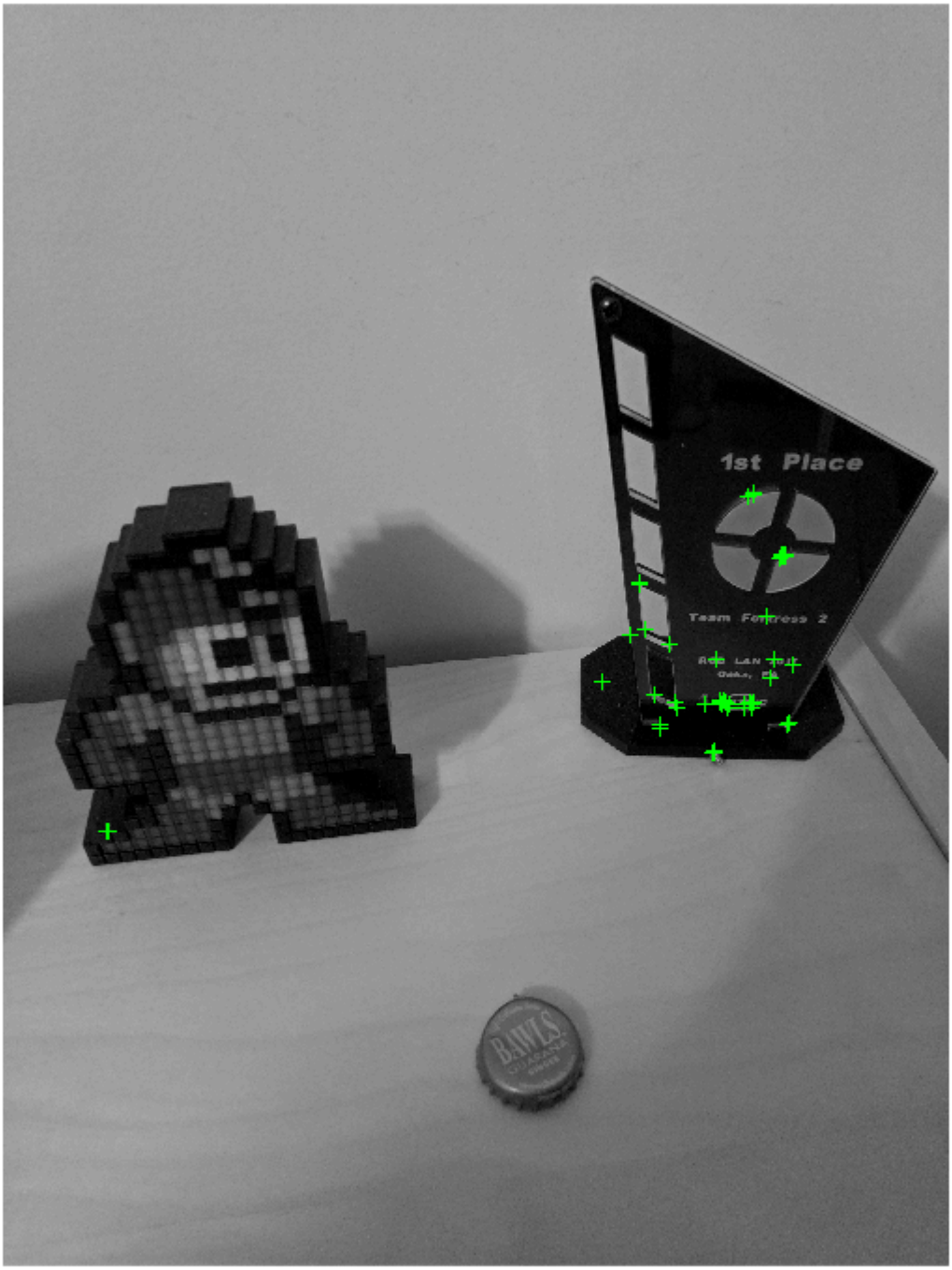

```
%even with 1000 points, only 1 for megaman and none for bottle cap

%harris
corners = detectHarrisFeatures(I);
imshow(I); hold on;
plot(corners.selectStrongest(100));
```

```
hold off
```



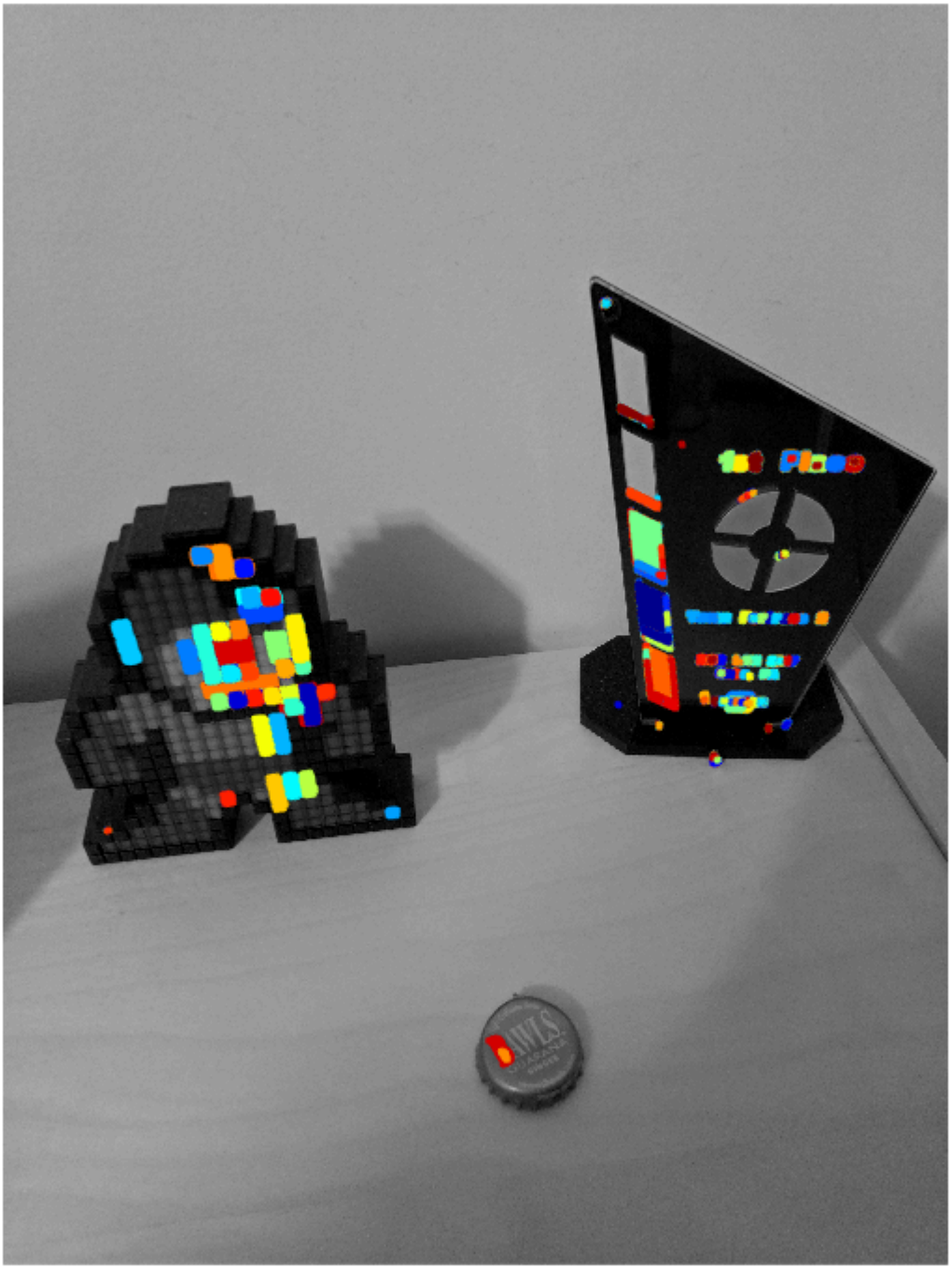

```
%similar results to fast

%mini eigen
corners = detectMinEigenFeatures(I);
imshow(I); hold on;
plot(corners.selectStrongest(50));
```

```
hold off
```



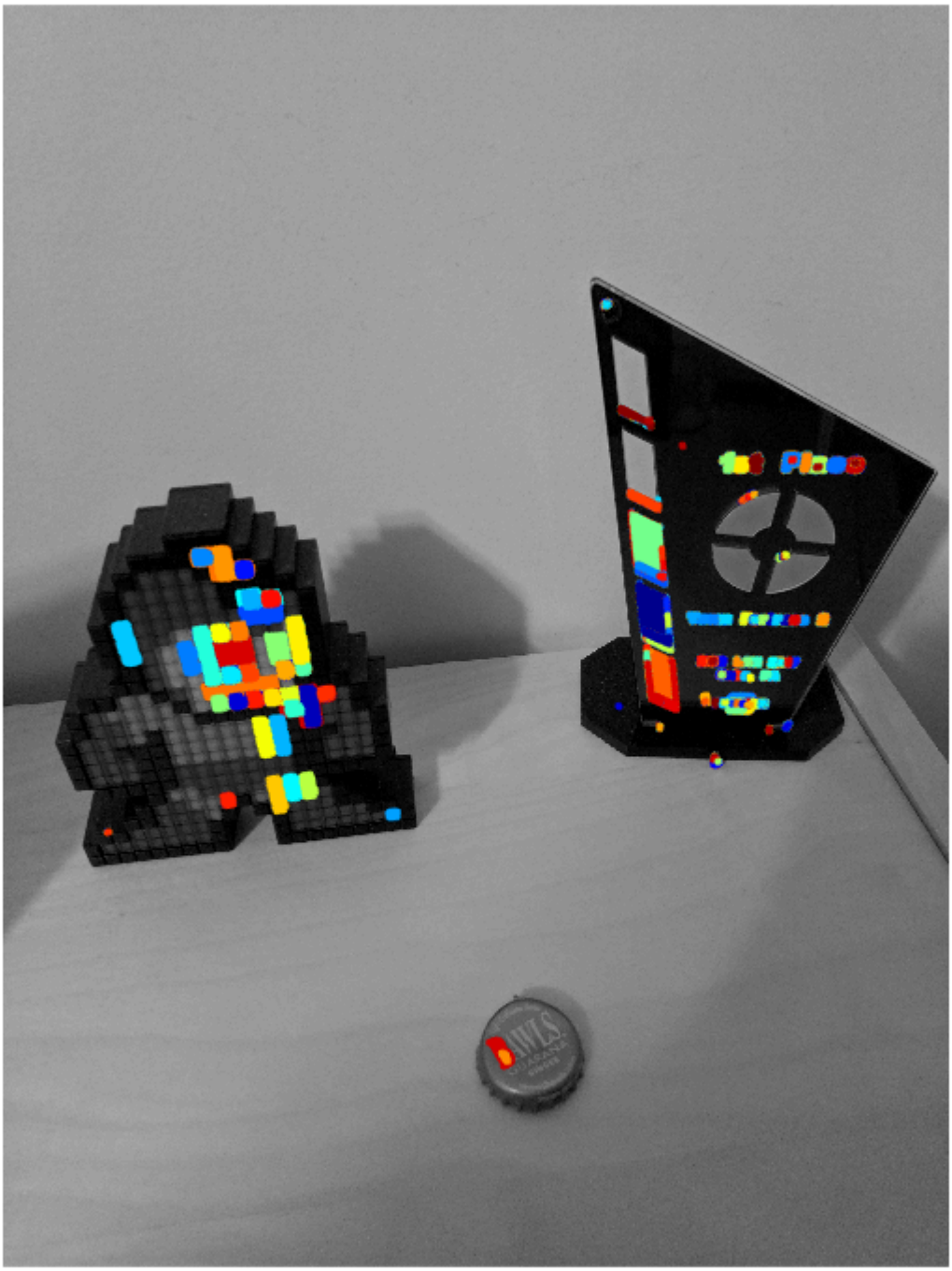

```
%also similar results

%mser
regions = detectMSERFeatures(I);
figure; imshow(I); hold on;
plot(regions, 'showPixelList',true,'showEllipses',false);
```

```
hold off
```



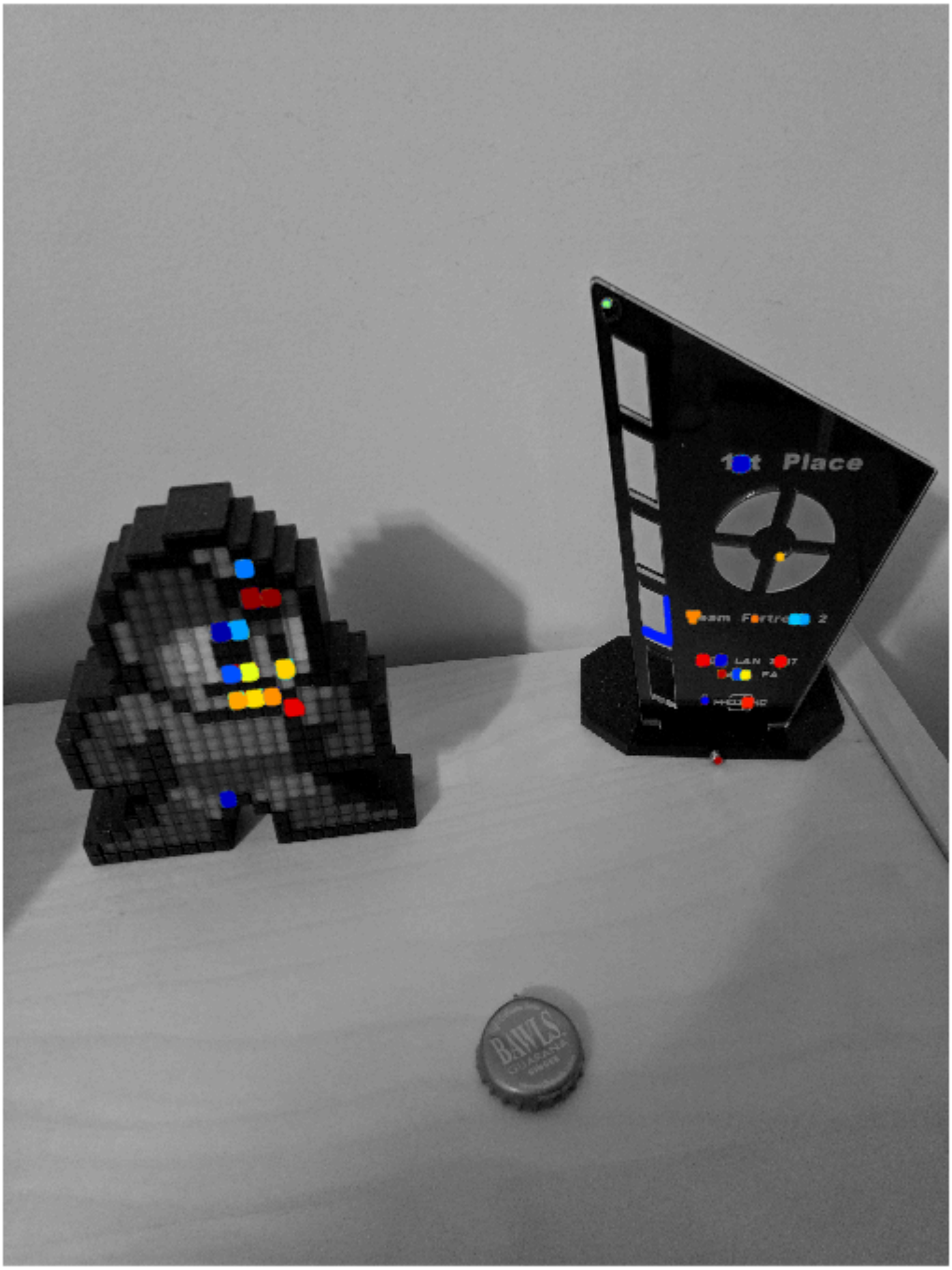
```
%recognizes the bottle cap! first to do so

[regions,mserCC] = detectMSERFeatures(I);
figure
imshow(I)
hold on
```

```
plot(regions,'showPixelList',true,'showEllipses',false);
hold off
```



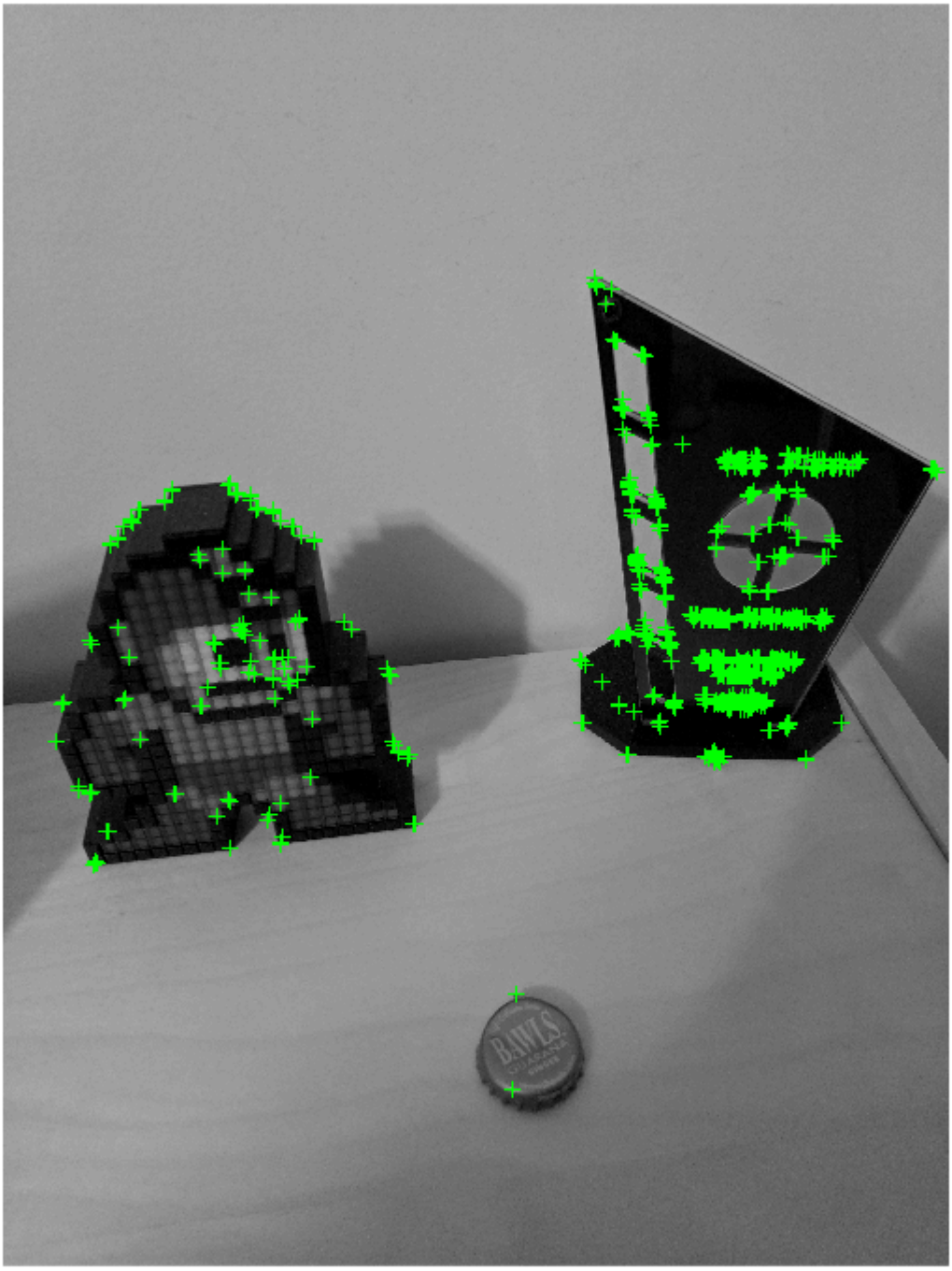
```
stats = regionprops('table',mserCC,'Eccentricity');
eccentricityIdx = stats.Eccentricity < 0.55;
circularRegions = regions(eccentricityIdx);
figure
```

```
imshow(I)
hold on
plot(circularRegions,'showPixelList',true,'showEllipses',false)
hold off
```



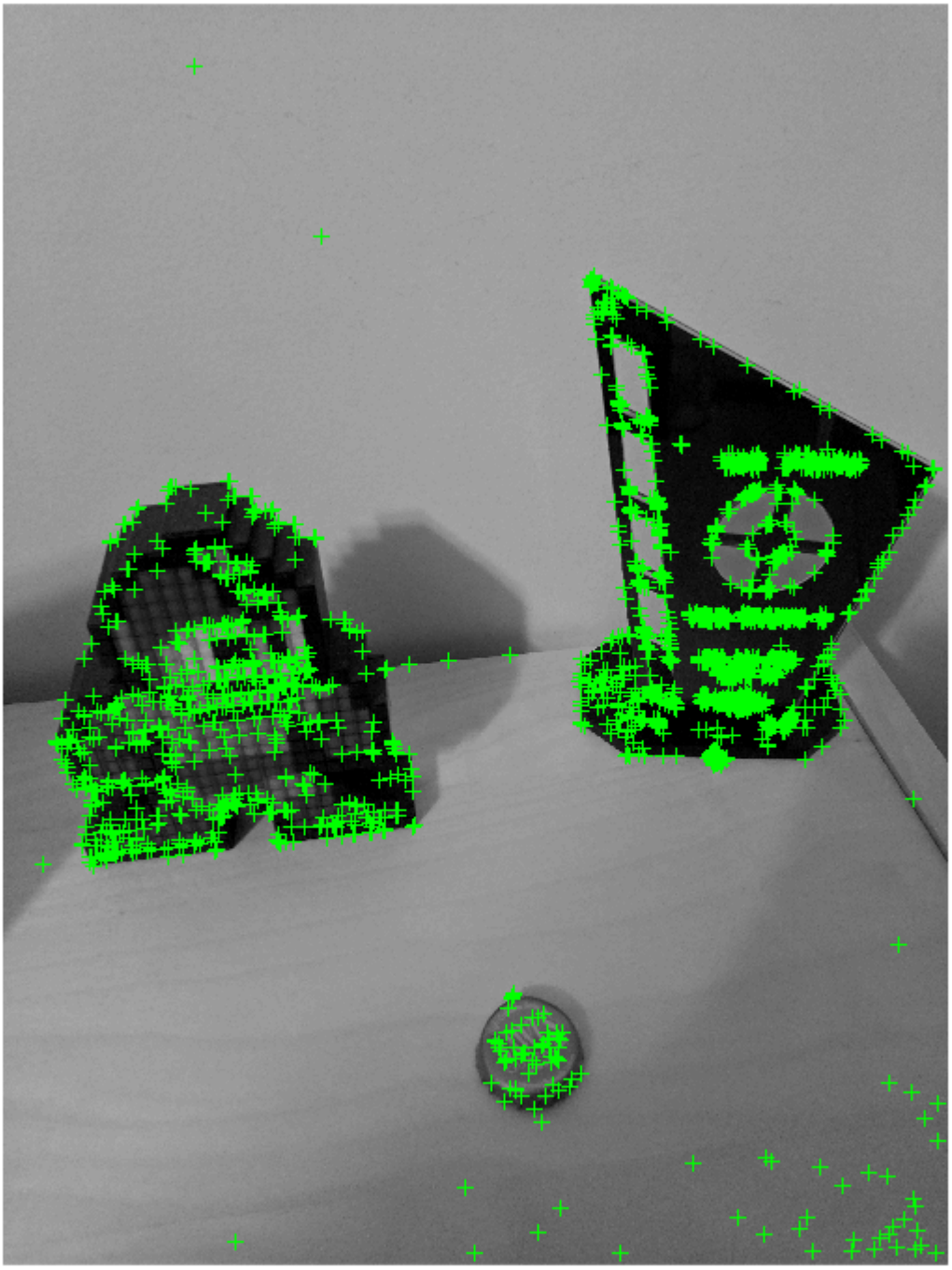

```
%a threshold applied to find most unique(?) points

%random, but works really well?
```

```
imshow(I)
hold on
plot(points,'ShowScale',false)
hold off
```



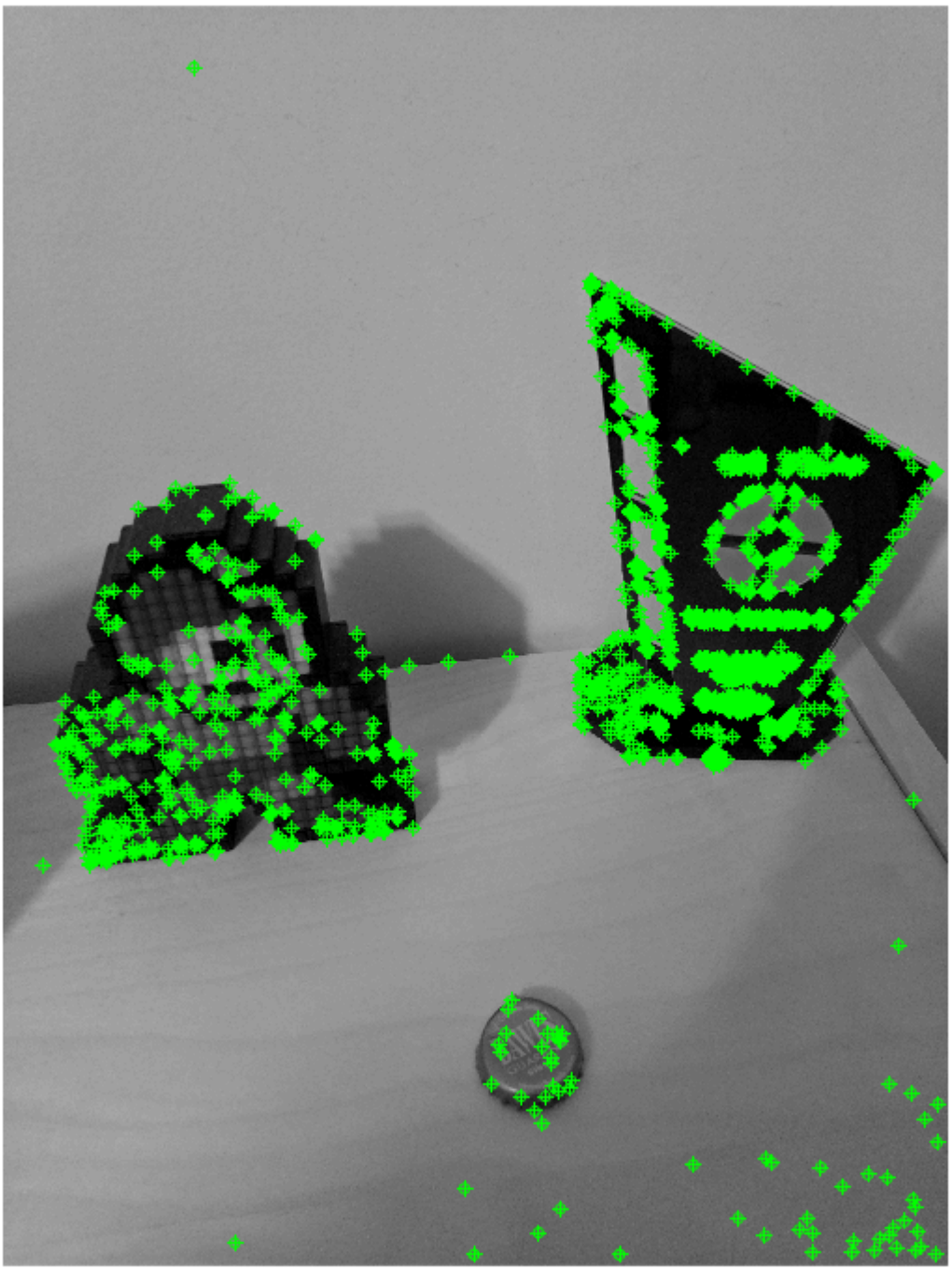
```
%orb
points = detectORBFeatures(I);
```

```
imshow(I)
hold on
plot(points,'ShowScale',false)
hold off
```



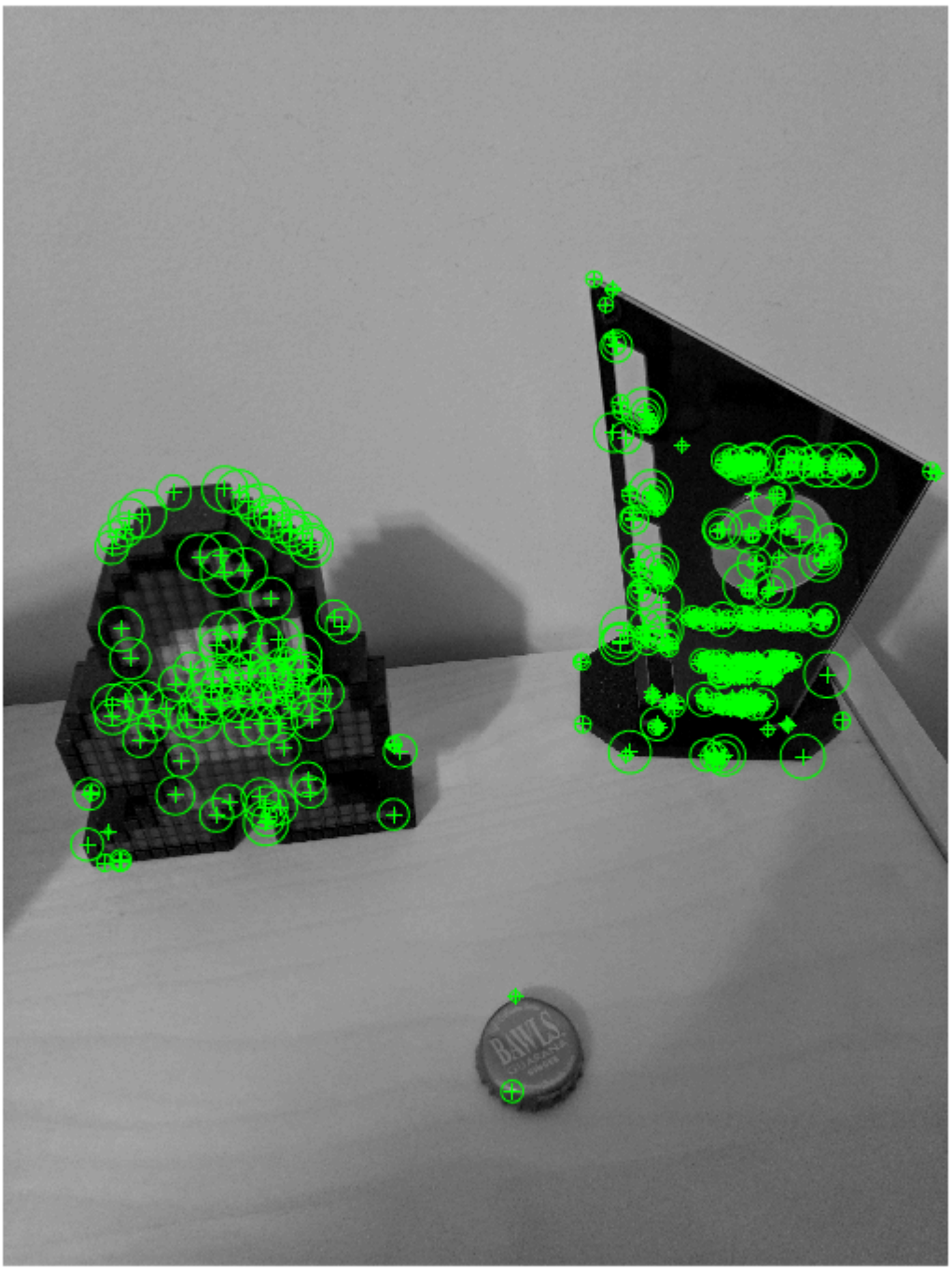
```
%a lot of points shown, happy it found the bottle cap but there are many
%useless points
```

```
points = detectORBFeatures(I,'ScaleFactor',1.01,'NumLevels',3);
imshow(I)
hold on
plot(points)
hold off
```



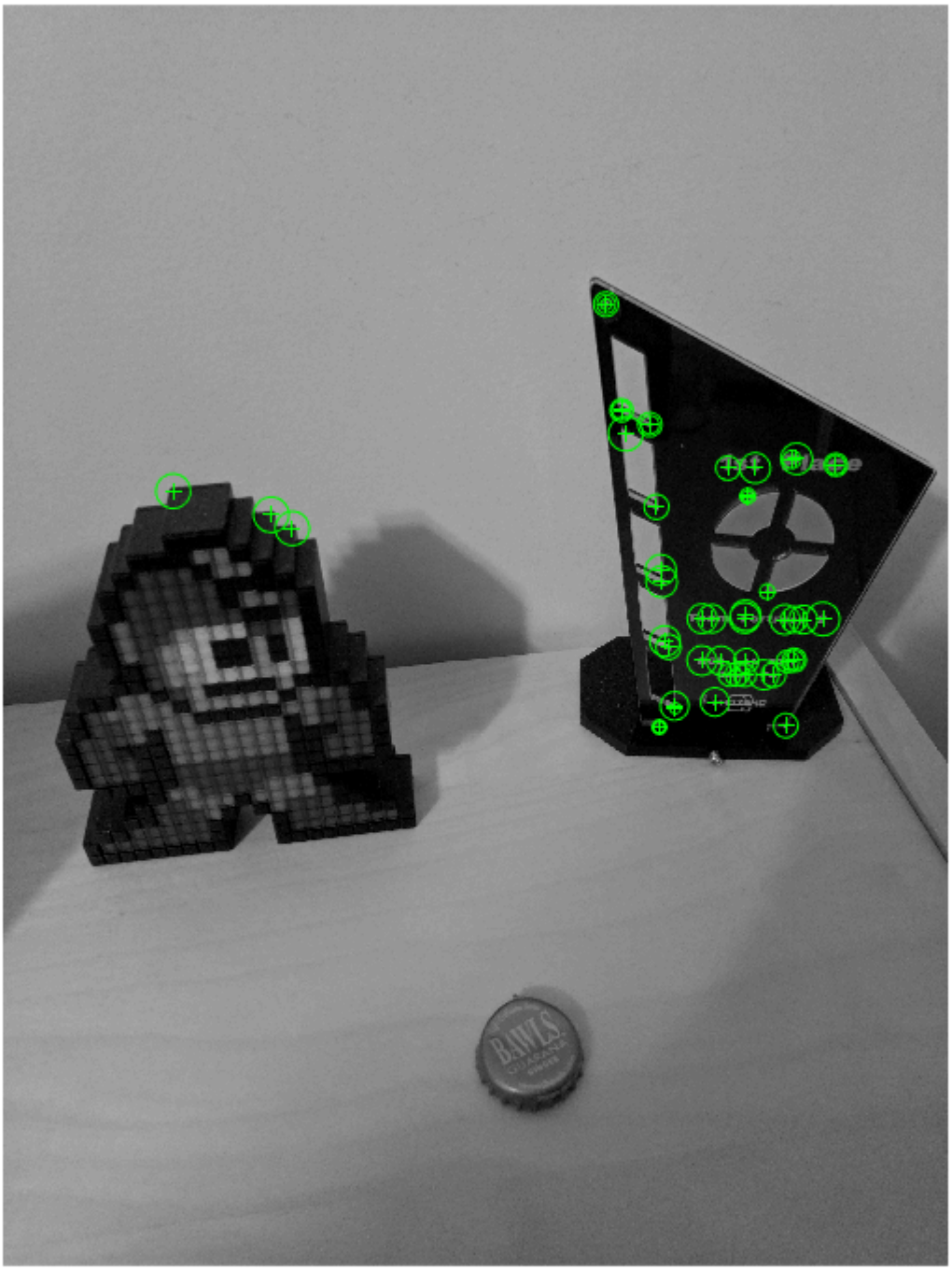
```
%surf
```

```
points = detectSURFFeatures(I);
imshow(I)
hold on
plot(points.selectStrongest(500));
hold off
```



```
%notices all! thanks surf you are great :)
%with 50 points, it's hard to get everything, 500 seems to do the job
```

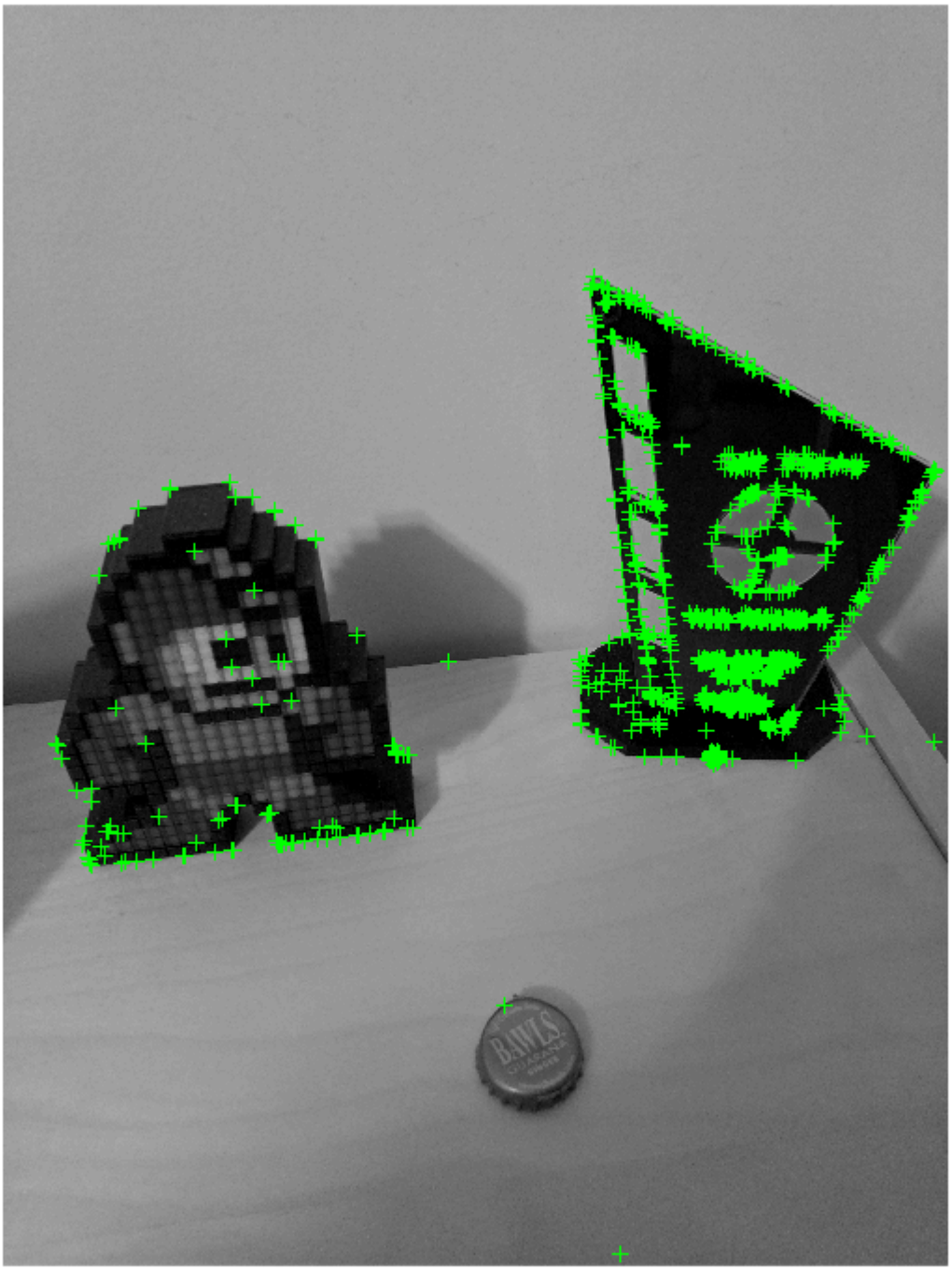
```
%kaze
points = detectKAZEFeatures(I);
imshow(I)
hold on
plot(selectStrongest(points,50))
hold off
```



```
%very similar to surf, with 10x more points it will find everything. kept
%at 50 to show difference

imshow(I)
hold on
plot(points(end-4:end));
hold off
```

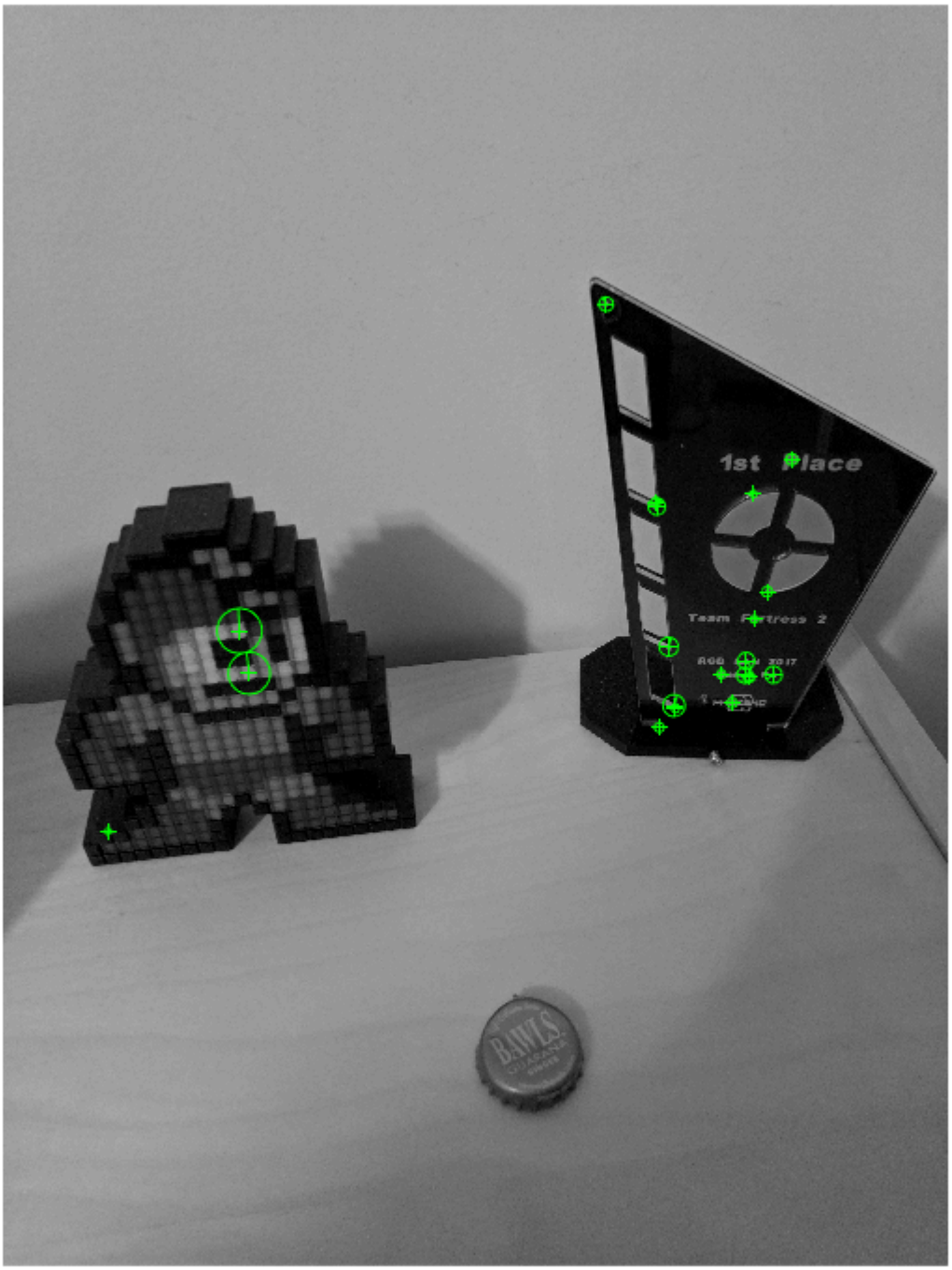
```
%only finds bottle cap for some reason
```

```
addpath 'D:'\homework\'checkerboard pics'\

I = imread('megaman3.jpg');
I = rgb2gray(I);

%harris
corners = detectHarrisFeatures(I);
[features, valid_corners] = extractFeatures(I, corners);
figure
imshow(I)
hold on
plot(valid_corners);
```



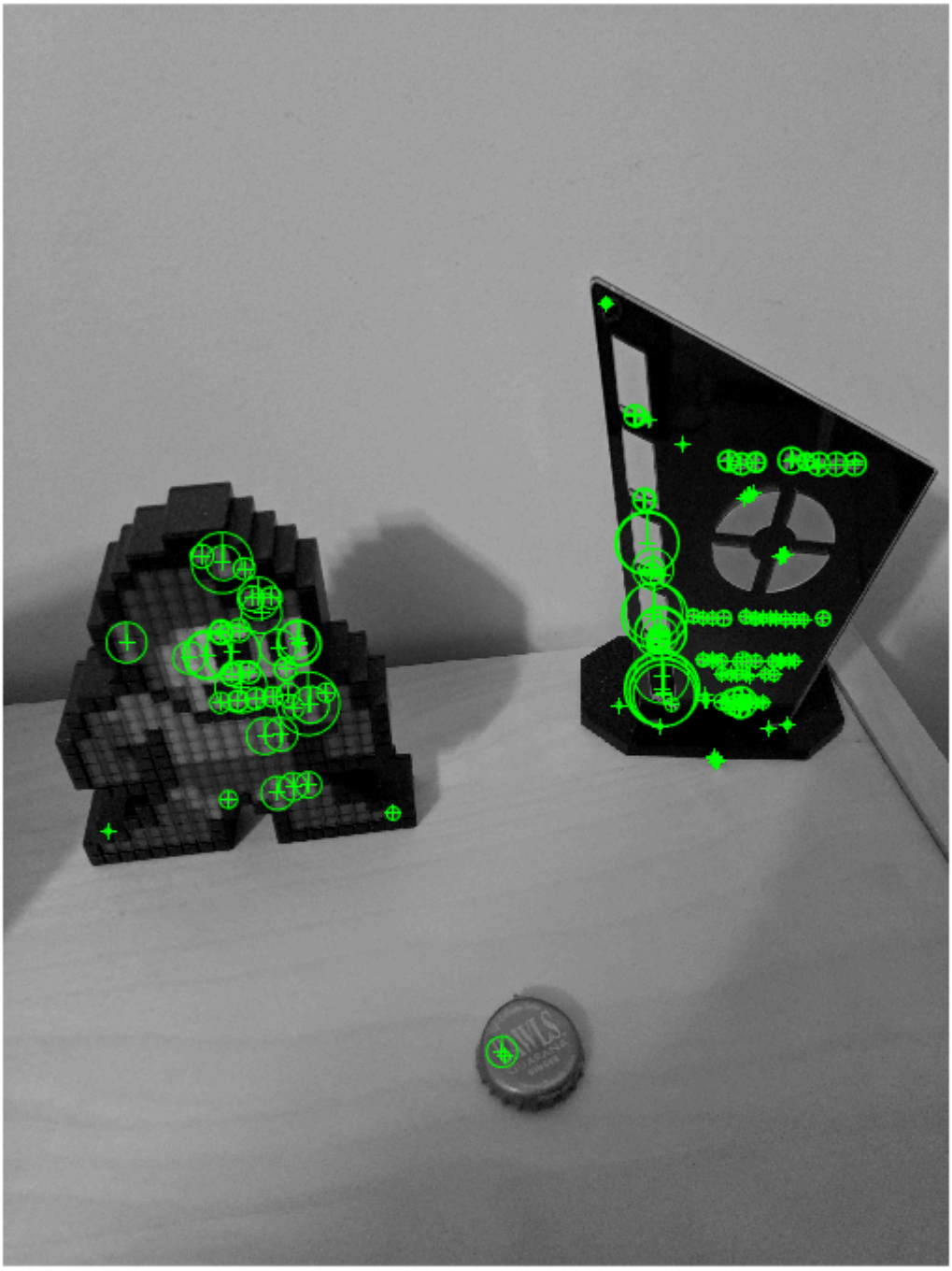

```
%good job at finding corners, again mistaken by the body of megaman. finds
%the bottle cap and even the edge of my bookshelf

%surf
points = detectSURFFeatures(I);
[features, valid_points] = extractFeatures(I, points);
figure
imshow(I)
```

```
hold on
plot(valid_points.selectStrongest(20),'showOrientation',true);
```



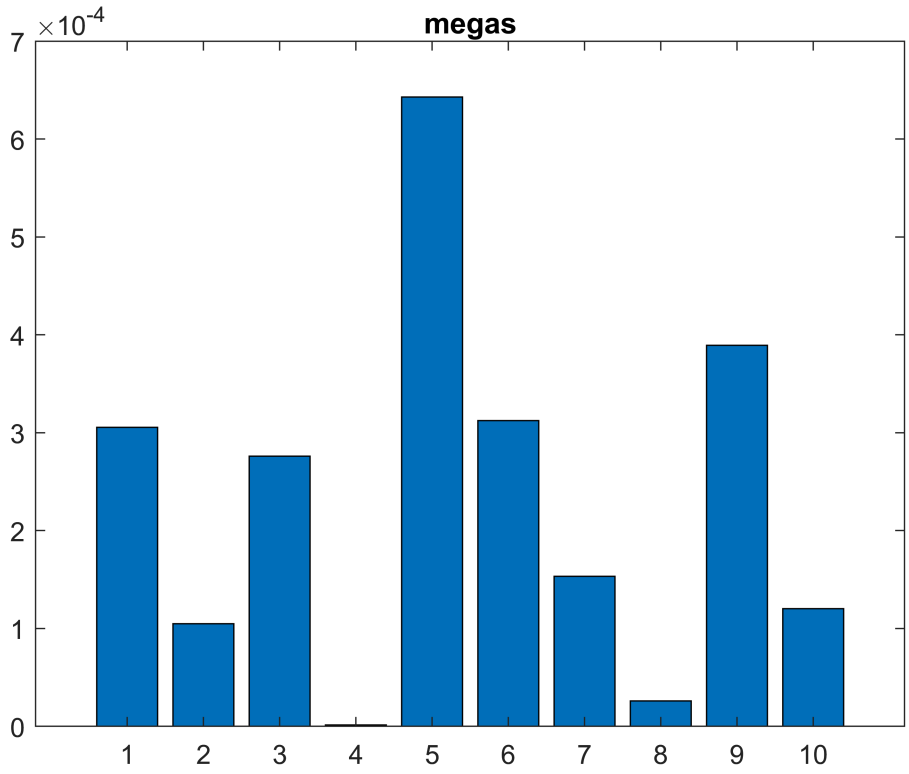

```
%small amount of points, doesn't find bottle cap

%mser
regions = detectMSERFeatures(I);
[features, valid_points] = extractFeatures(I,regions,'Upright',true);
```

```
figure
imshow(I)
hold on
plot(valid_points,'showOrientation',true);
```

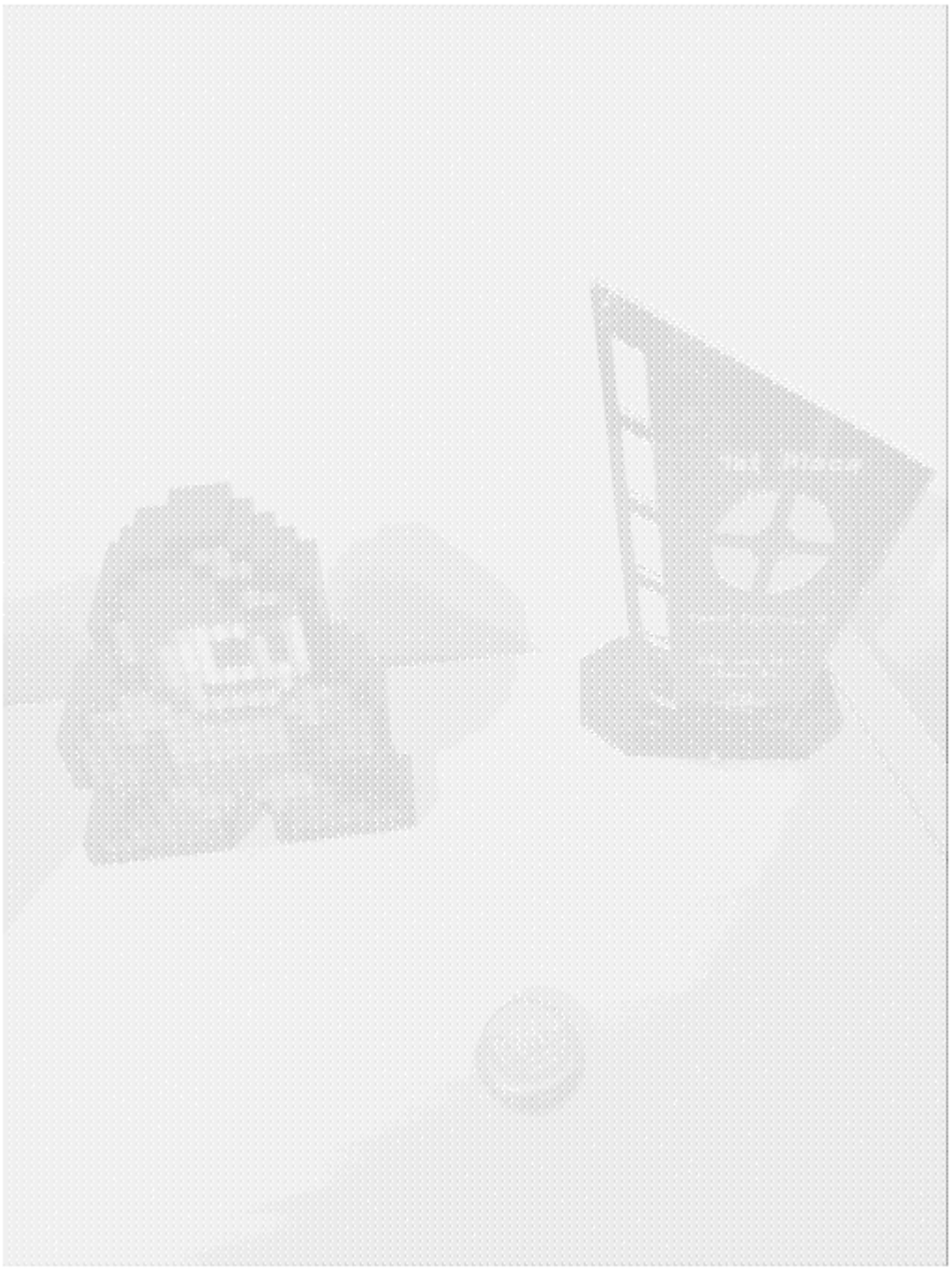


```
%perfect

%lbp
```

```
X = imread('megaman1.jpg');
X = rgb2gray(X);
lbp1 = extractLBPFeatures(I,'Upright',false);
lbp2 = extractLBPFeatures(X,'Upright',false);
megaVsMega = (lbp1 - lbp2).^2;

figure
bar([megaVsMega],'grouped')
title('megas')
```




```
%hog
[featureVector,hogVisualization] = extractHOGFeatures(I);
figure
imshow(I)
hold on
plot(hogVisualization)
```

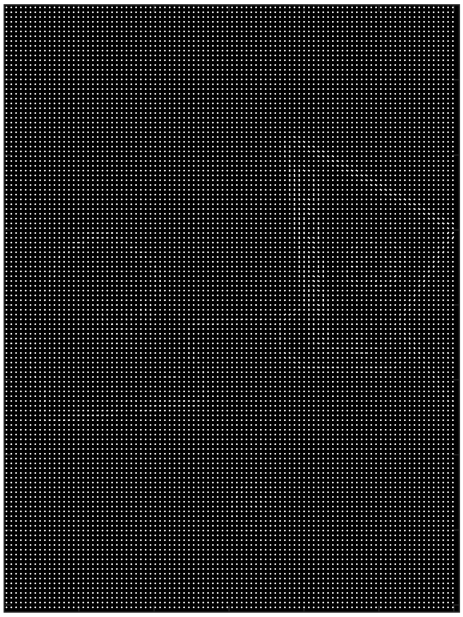


```
[hog1,visualization] = extractHOGFeatures(I,'CellSize',[32 32])
```

```
hog1 = 1×418500 single row vector
    0.1752    0.1527    0.1776    0.1701    0.1919    0.1521    0.1405    0.1529 ···
visualization =
Visualization

   Type plot(visualization) to visualize.
```

```
  Read-only properties:
              CellSize: [32 32]
             BlockSize: [2 2]
          BlockOverlap: [1 1]
               NumBins: 9
    UseSignedOrientation: 0
             BinCenters: [18×1 double]
```

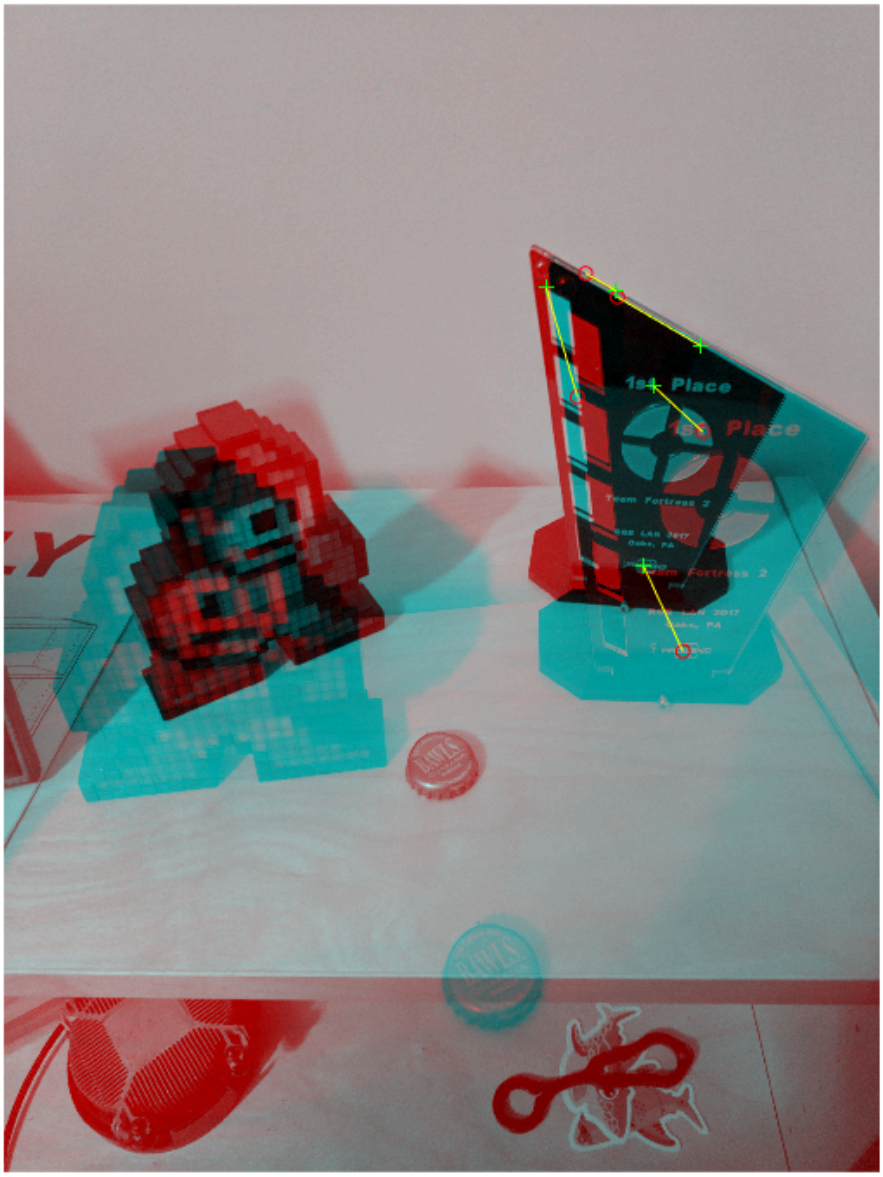
```
subplot(1,2,1);
imshow(I);
subplot(1,2,1);
plot(visualization);
```



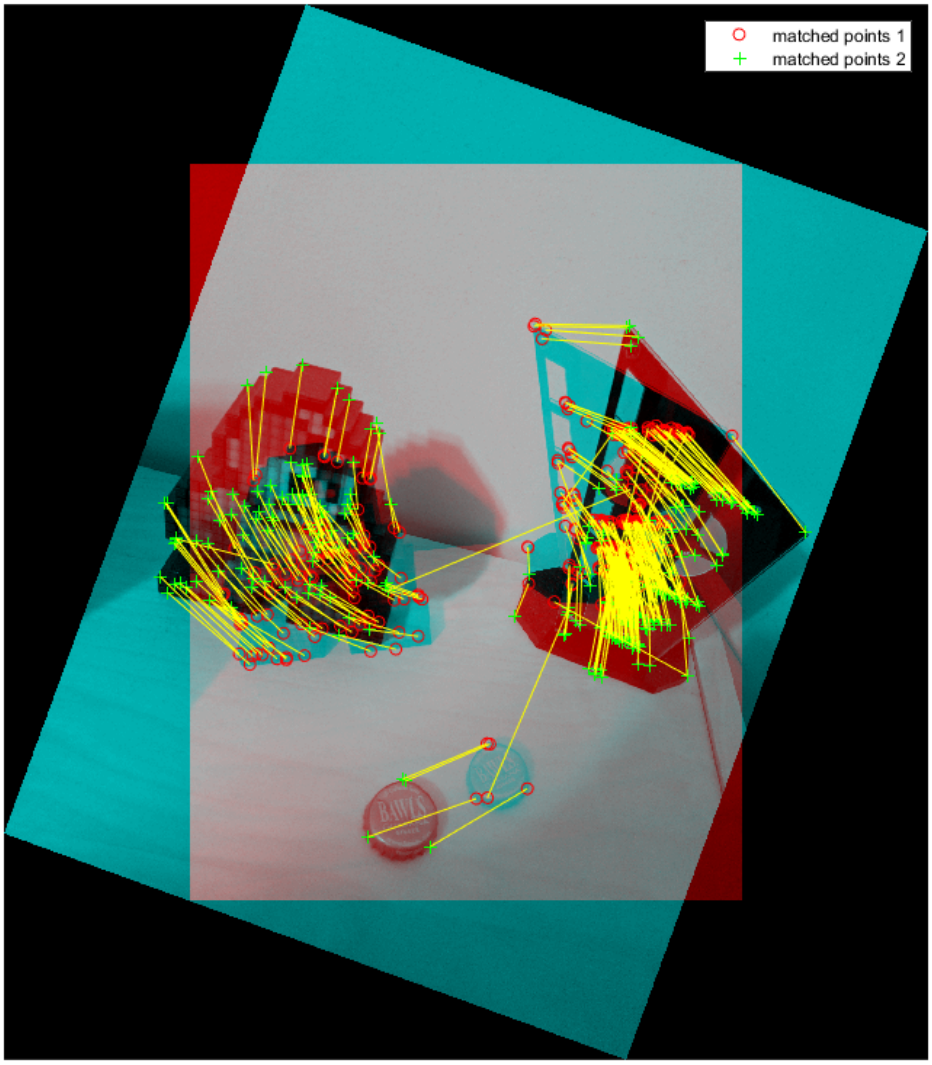
```
%fast
corners = detectFASTFeatures(I);
strongest = selectStrongest(corners,3);
[hog, validPoints,ptVis] = extractHOGFeatures(I,strongest);
figure
imshow(I)
hold on
```

```
plot(ptVis,'Color','green');
```

```
addpath 'D:'\homework\'checkerboard pics'\

I = imread('megaman3.jpg');
I = rgb2gray(I);


Y = imread('megaman2.jpg');
Y = rgb2gray(Y);
%match features
points1 = detectHarrisFeatures(I);
points2 = detectHarrisFeatures(Y);

[features1,valid_points1] = extractFeatures(I,points1);
[features2,valid_points2] = extractFeatures(Y,points2);

indexPairs = matchFeatures(features1,features2);

matchedPoints1 = valid_points1(indexPairs(:,1),:);
matchedPoints2 = valid_points2(indexPairs(:,2),:);

figure; showMatchedFeatures(I,Y,matchedPoints1,matchedPoints2);
```

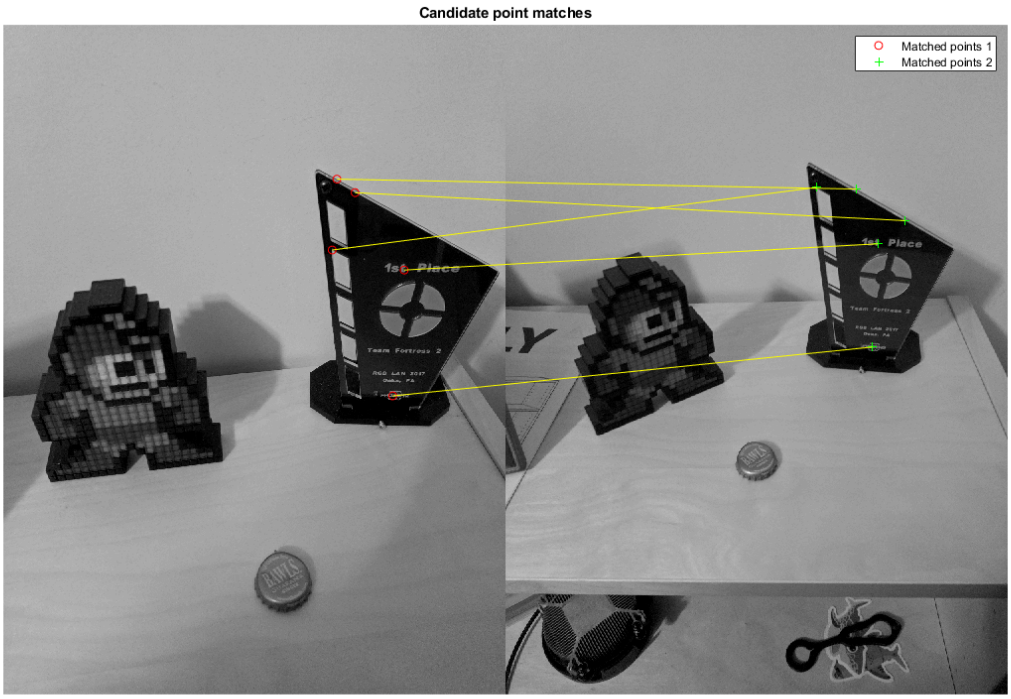



```
I2 = imresize(imrotate(I,-20),1.2);
points1 = detectSURFFeatures(I);
points2 = detectSURFFeatures(I2);
[f1,vpts1] = extractFeatures(I,points1);
[f2,vpts2] = extractFeatures(I2,points2);
indexPairs = matchFeatures(f1,f2) ;
matchedPoints1 = vpts1(indexPairs(:,1));
matchedPoints2 = vpts2(indexPairs(:,2));
figure; showMatchedFeatures(I,I2,matchedPoints1,matchedPoints2);
legend('matched points 1','matched points 2');
```




```
%tracks the base, logo, and top of the trophy, but neither of the other two
%objects

%showMatchedFeatures
points1 = detectHarrisFeatures(I);
points2 = detectHarrisFeatures(Y);
%tracks the location of all 3 objects post rotation. works really well in
%this case

[f1, vpts1] = extractFeatures(I, points1);
[f2, vpts2] = extractFeatures(Y, points2);

indexPairs = matchFeatures(f1, f2) ;
matchedPoints1 = vpts1(indexPairs(1:5, 1));
matchedPoints2 = vpts2(indexPairs(1:5, 2));

figure; ax = axes;
showMatchedFeatures(I,Y,matchedPoints1,matchedPoints2,'montage','Parent',ax);
```

```
title(ax, 'Candidate point matches');
legend(ax, 'Matched points 1','Matched points 2');
```




```
%matches the trophy from one image to another, but neither of the other two
%objects
```

```
addpath 'D:'\homework\'checkerboard pics'\

I = imread('megaman3.jpg');
I = rgb2gray(I);

X = imread('megaman1.jpg');
X = rgb2gray(X);

%binary
%not entirely sure what values to put in for the binary features, just used
%those in the example
features1 = binaryFeatures(uint8([1 8 7 2; 8 1 7 2]));
features2 = binaryFeatures(uint8([8 1 7 2; 1 8 7 2]));

[indexPairs matchMetric] = matchFeatures(features1, features2)
```

```
indexPairs = 2×2 uint32 matrix
   1   2
   2   1
matchMetric = 2×1 single column vector
     0
     0
```

```
%brisk
points = detectBRISKFeatures(I);
location = [200:228;200:228]';
points = BRISKPoints(location);

strongest = points.selectStrongest(10);
imshow(I)
hold on
plot(strongest)
hold off
```


```
%garbage
strongest.Location
```

```
ans = 10×2 single matrix
   200   200
   201   201
   202   202
   203   203
   204   204
```

```
   205   205
   206   206
   207   207
   208   208
   209   209
```

```
%kaze
points = detectKAZEFeatures(I);
strongest = selectStrongest(points,10);
imshow(I)
hold on
plot(strongest)
hold off
```


```
%only finds trophy with this low amount of points, but with a higher amount
%of points it also does well
strongest.Location
```

```
ans = 10×2 single matrix
10^3 ×
    1.9378    0.9609
    1.9379    0.9609
```

```
    2.1579    2.2508
    2.1384    2.0617
    1.9376    0.9614
    2.0999    1.6118
    2.1159    1.8514
    2.4756    2.1507
    2.3831    1.9687
    2.3879    2.1060
```

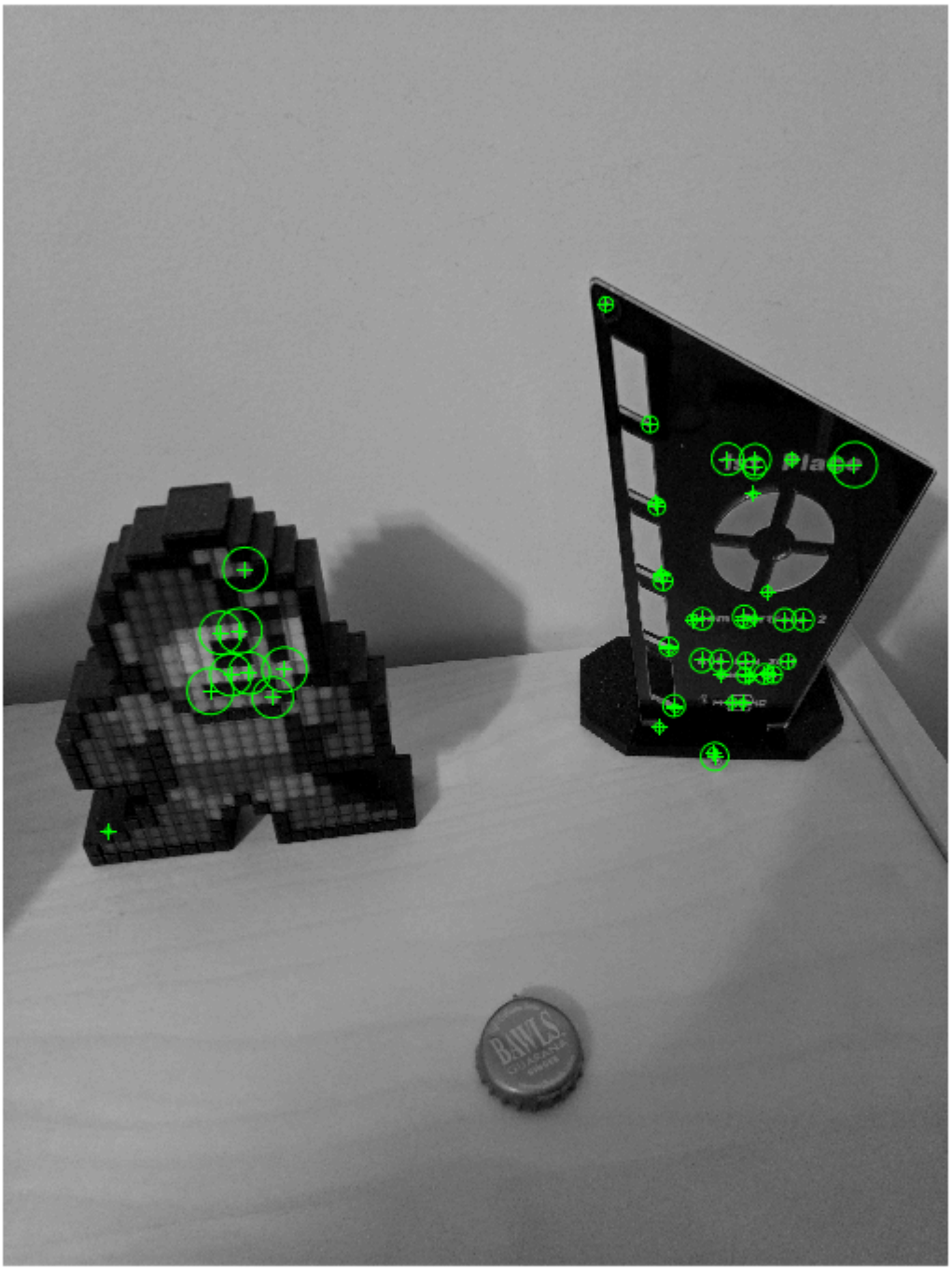
```
imshow(I)
hold on
plot(points(end-4:end)) %whenever i do this notation, i only find the bottlecap
hold off
```


```
%corner
points = detectHarrisFeatures(I);
strongest = selectStrongest(points,50);
imshow(I)
hold on
plot(strongest)
```

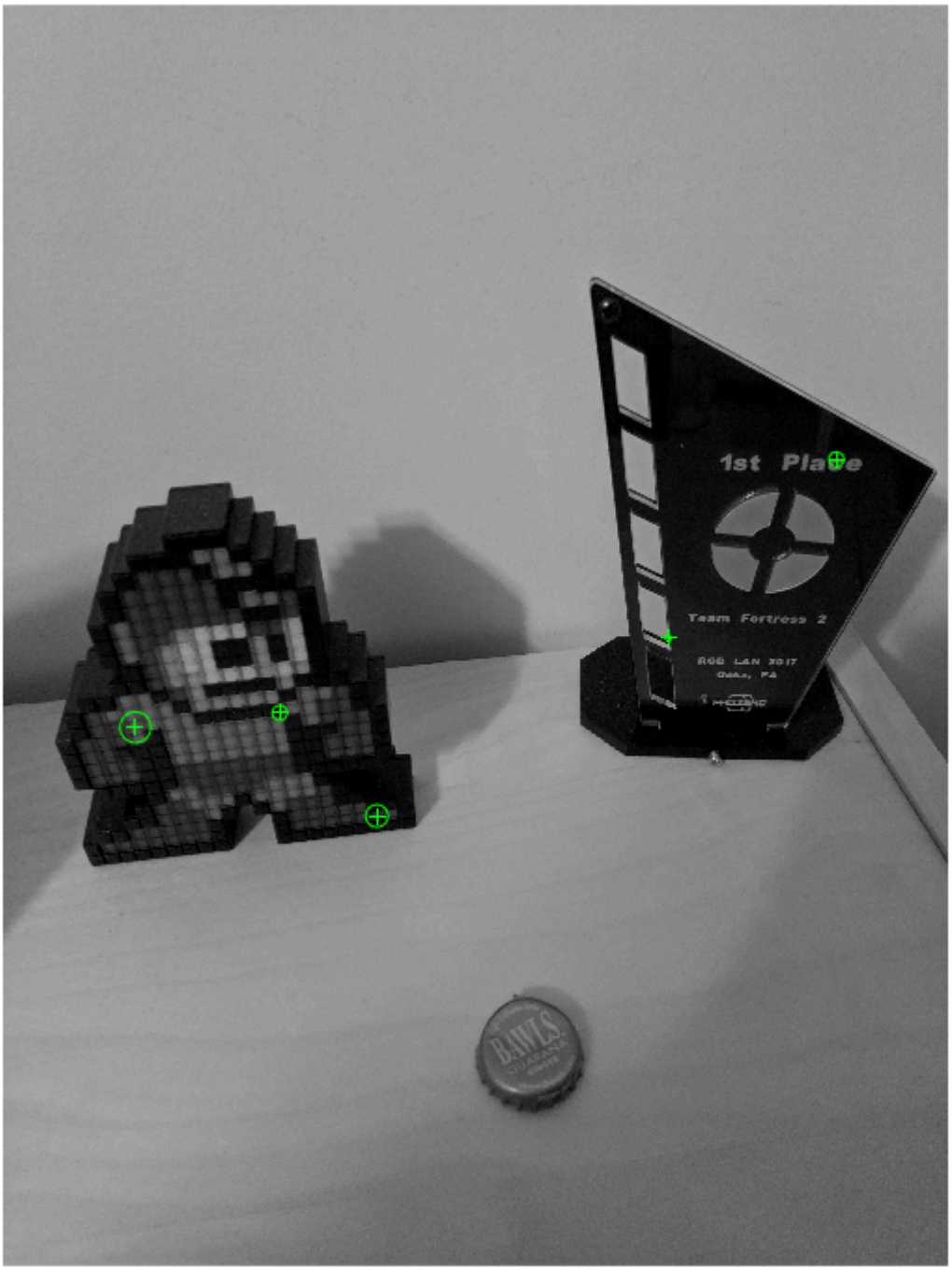
```
hold off
```


```
%surf
points = detectSURFFeatures(I);
strongest = points.selectStrongest(50);
imshow(I); hold on;
```

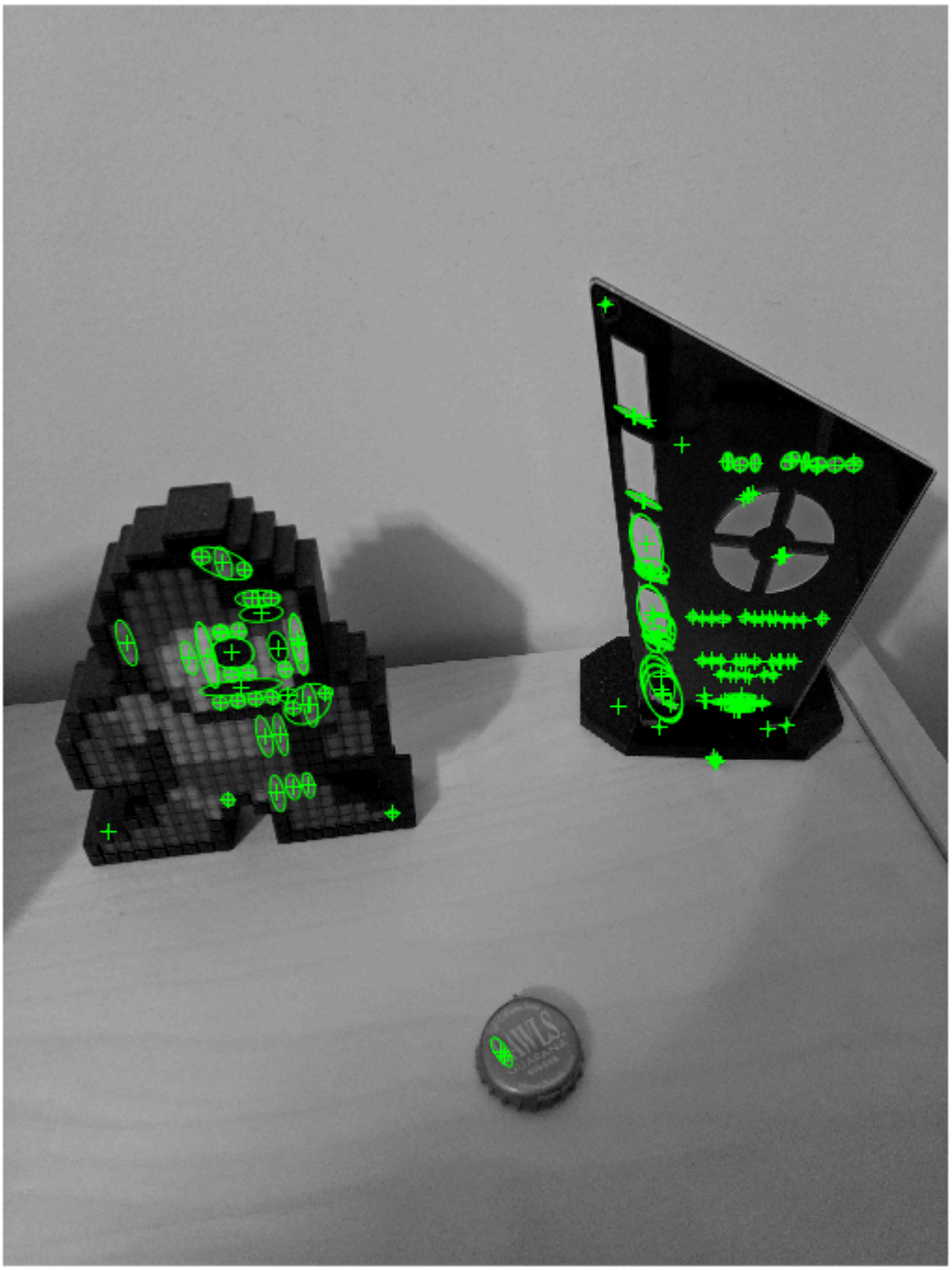
```
plot(strongest);
hold off
```



```
imshow(I); hold on;
plot(points(end-4:end));
hold off
```

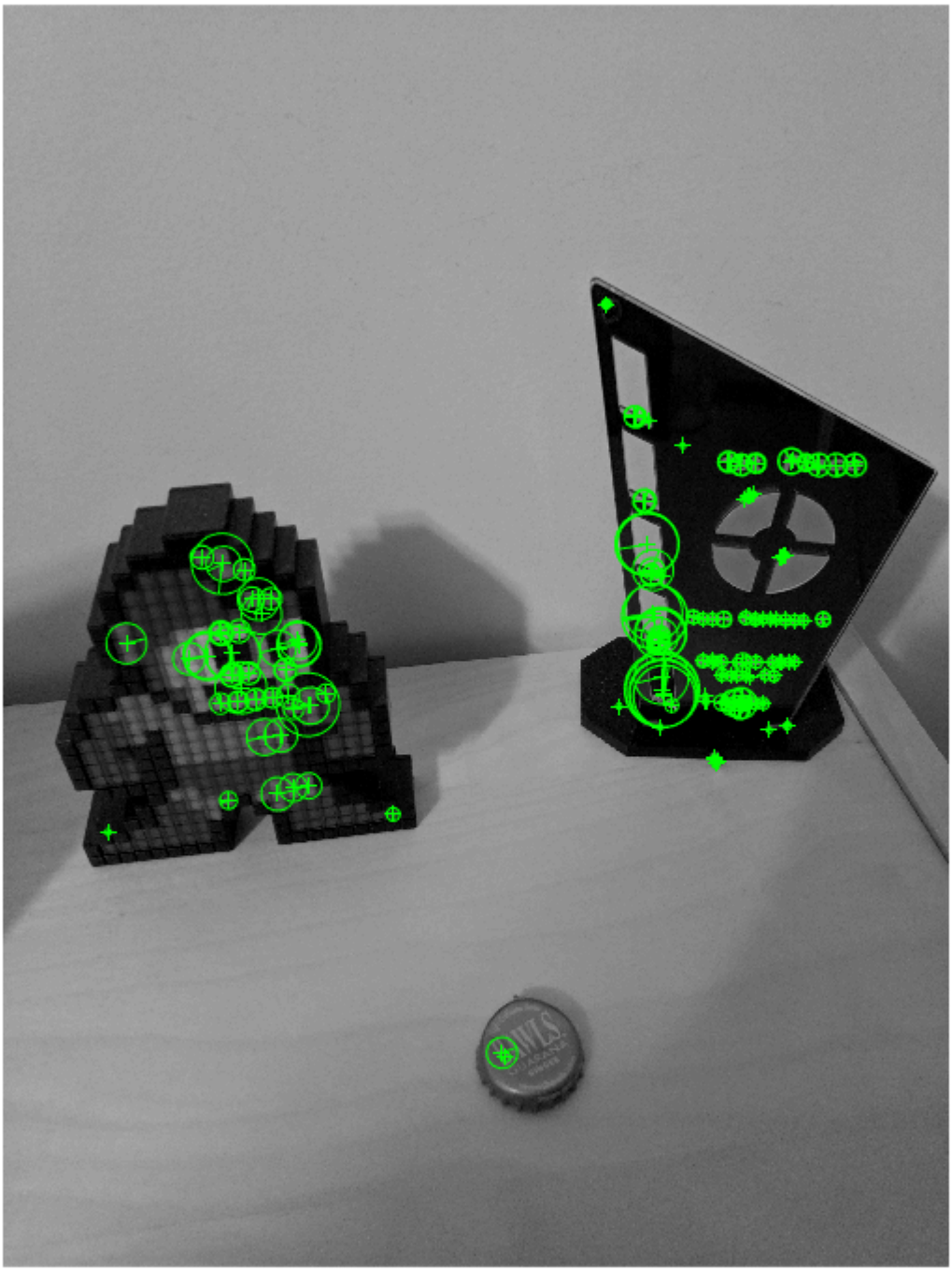



```
%mser
regions = detectMSERFeatures(I);
imshow(I); hold on;
plot(regions);
hold off
```



```
imshow(I); hold on;
plot(regions(1:10),'showPixelList', true);
hold off
```


```
regionsObj = detectMSERFeatures(I);
[features, validPtsObj] = extractFeatures(I, regionsObj);
imshow(I); hold on;
plot(validPtsObj,'showOrientation',true);
hold off
```

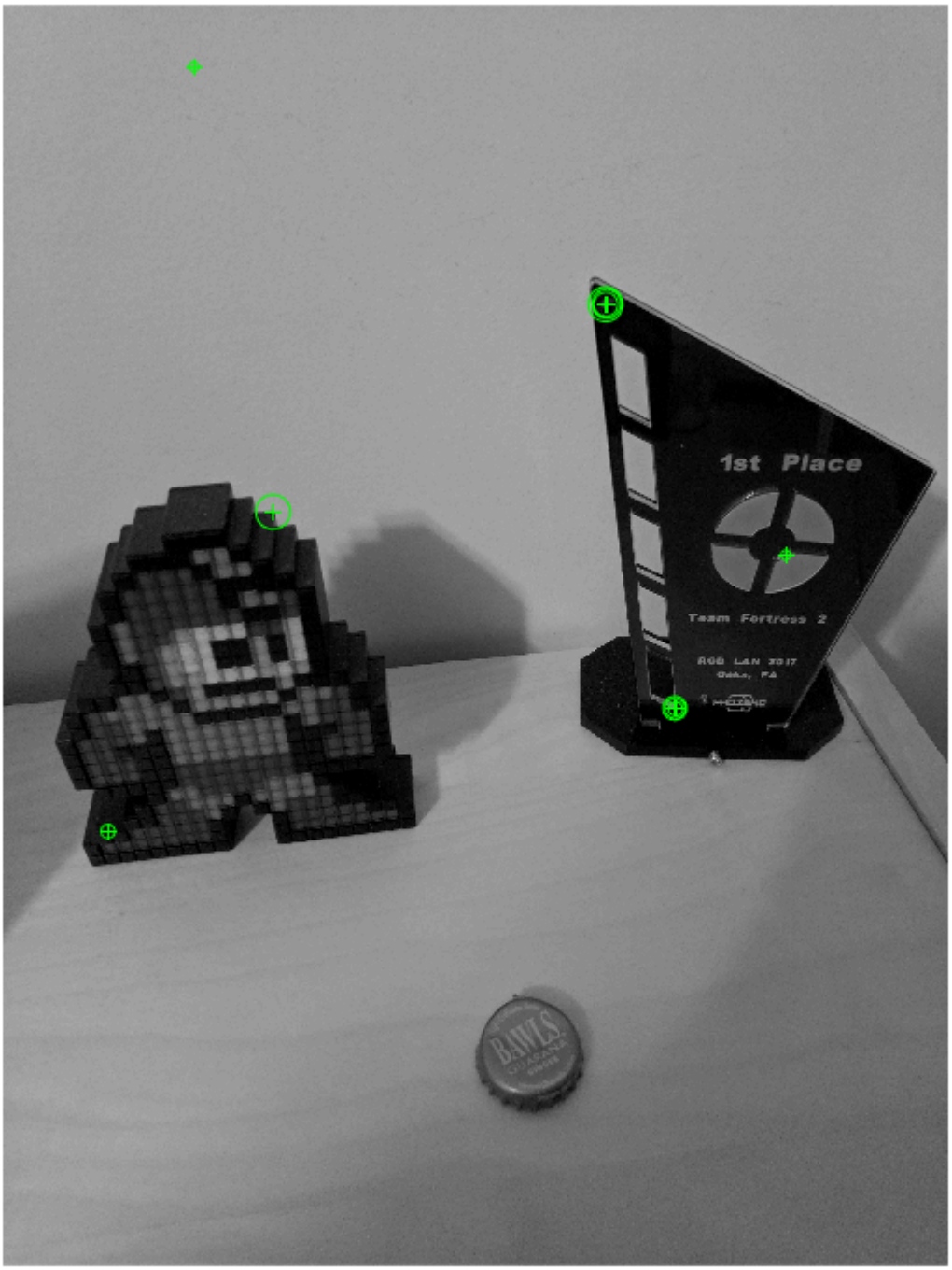


```
%orb
points = detectORBFeatures(I);
newPoints = selectUniform(points,10,size(I))
```

```
newPoints =
  10×1 ORBPoints array with properties:
```

```
       Location: [10×2 single]
         Metric: [10×1 single]
          Count: 10
          Scale: [10×1 single]
    Orientation: [10×1 single]
```

```
figure
imshow(I)
hold on
plot(newPoints)
hold off
```




```
%almost all of these needed more points to find the objects, and would
%require some kind of manual editing for the points that didn't belong to
%an object
```

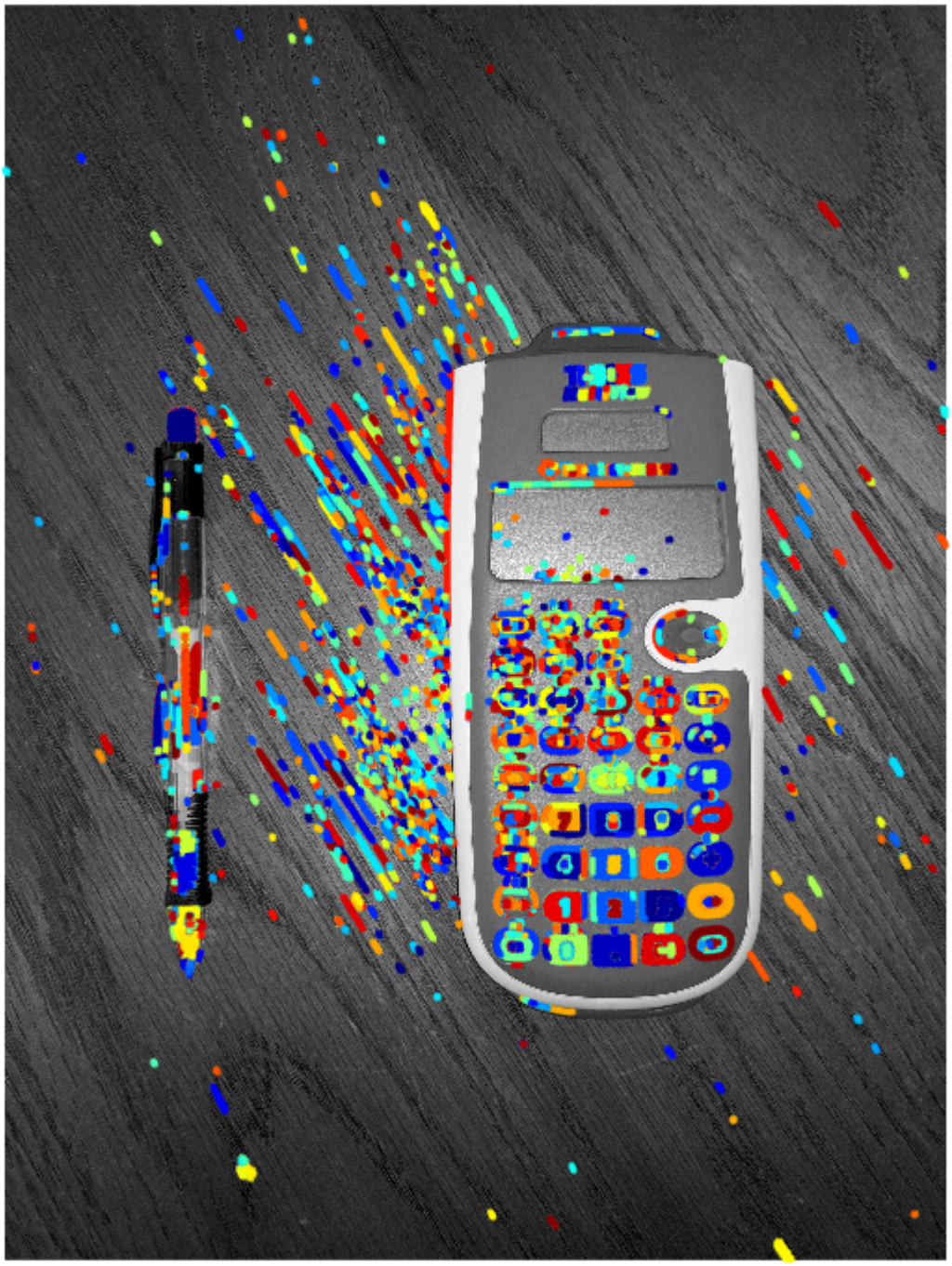
```
addpath 'D:'\homework\'comp vision'\'image pairs'\
%1
I = imread('calc_pencil_1.jpg');
I = rgb2gray(I);
%brisk
points = detectBRISKFeatures(I);
imshow(I); hold on;
plot(points.selectStrongest(100));
hold off
```

```
%fast
corners = detectFASTFeatures(I);
imshow(I); hold on;
plot(corners.selectStrongest(1000));
hold off
```

```
%harris
corners = detectHarrisFeatures(I);
imshow(I); hold on;
plot(corners.selectStrongest(100));
hold off
```

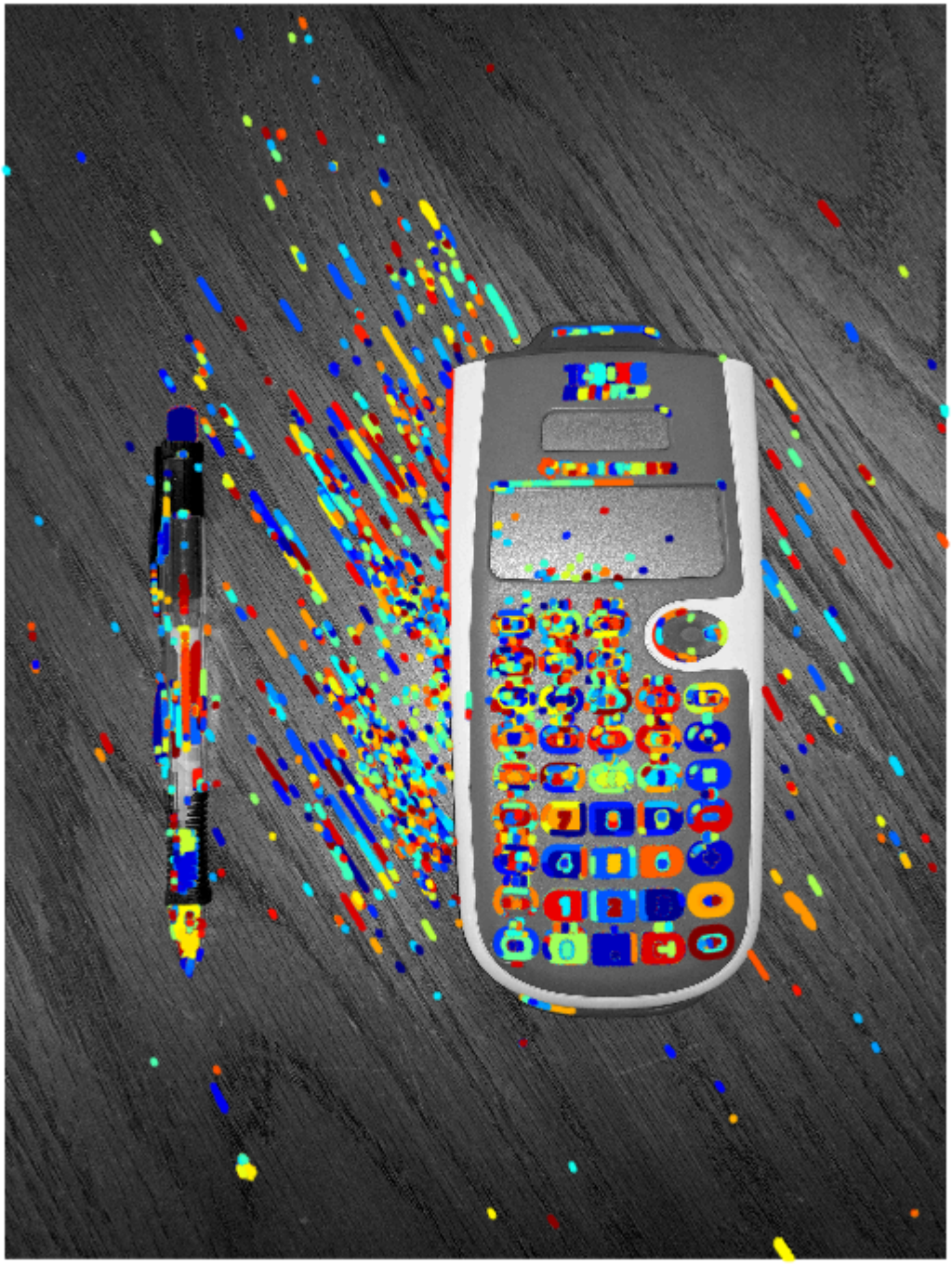
```
%similar results to fast

%mini eigen
corners = detectMinEigenFeatures(I);
imshow(I); hold on;
plot(corners.selectStrongest(50));
hold off
```


```
%also similar results

%mser
regions = detectMSERFeatures(I);
figure; imshow(I); hold on;
plot(regions, 'showPixelList',true,'showEllipses',false);
hold off
```



```
[regions,mserCC] = detectMSERFeatures(I);
figure
imshow(I)
hold on
plot(regions,'showPixelList',true,'showEllipses',false);
```

```
hold off
```



```
stats = regionprops('table',mserCC,'Eccentricity');
eccentricityIdx = stats.Eccentricity < 0.55;
circularRegions = regions(eccentricityIdx);
figure
imshow(I)
```

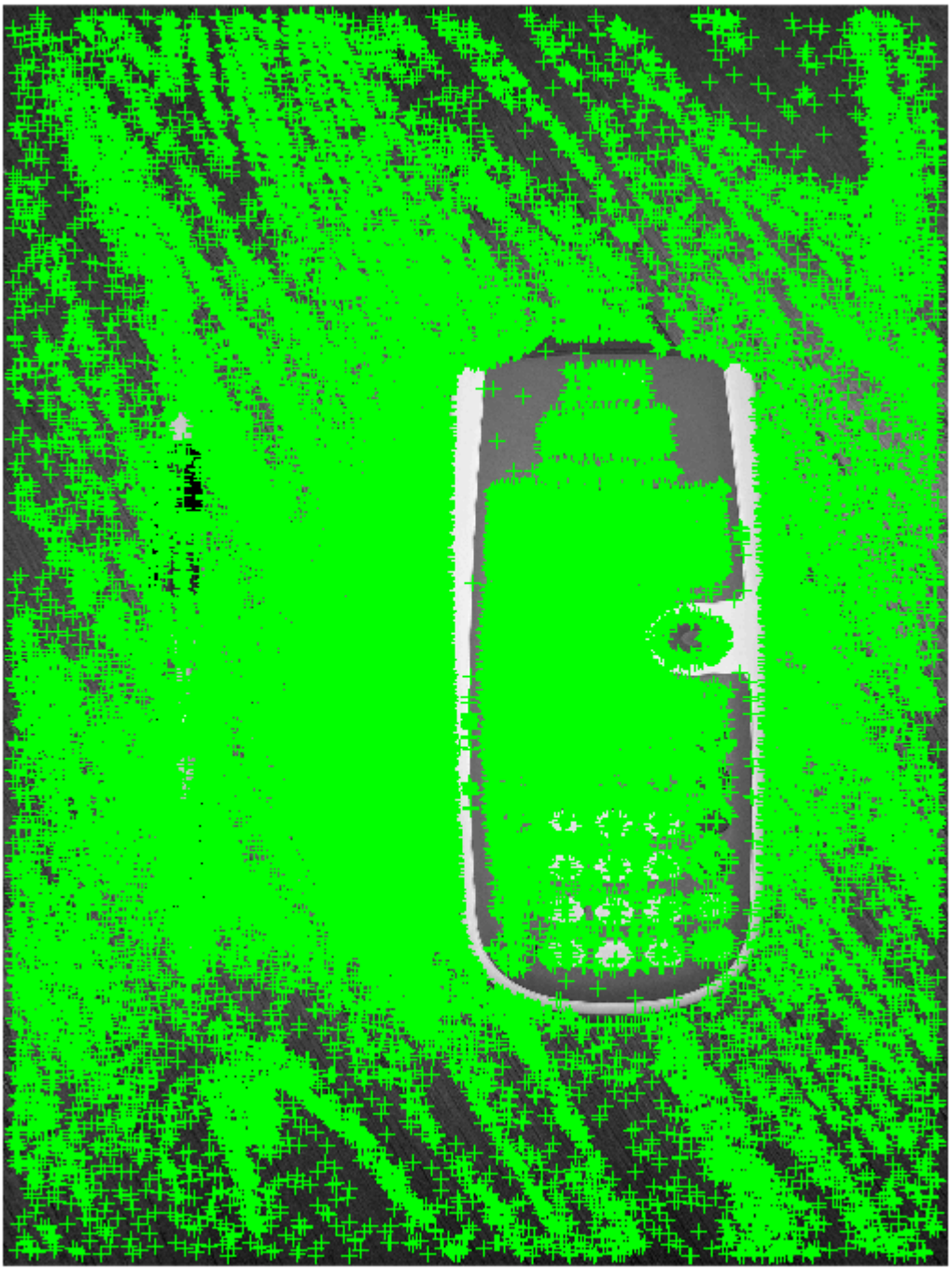
```
hold on
plot(circularRegions,'showPixelList',true,'showEllipses',false)
hold off
```


```
%a threshold applied to find most unique(?) points

%random, but works really well?
imshow(I)
```

```
hold on
plot(points,'ShowScale',false)
hold off
```

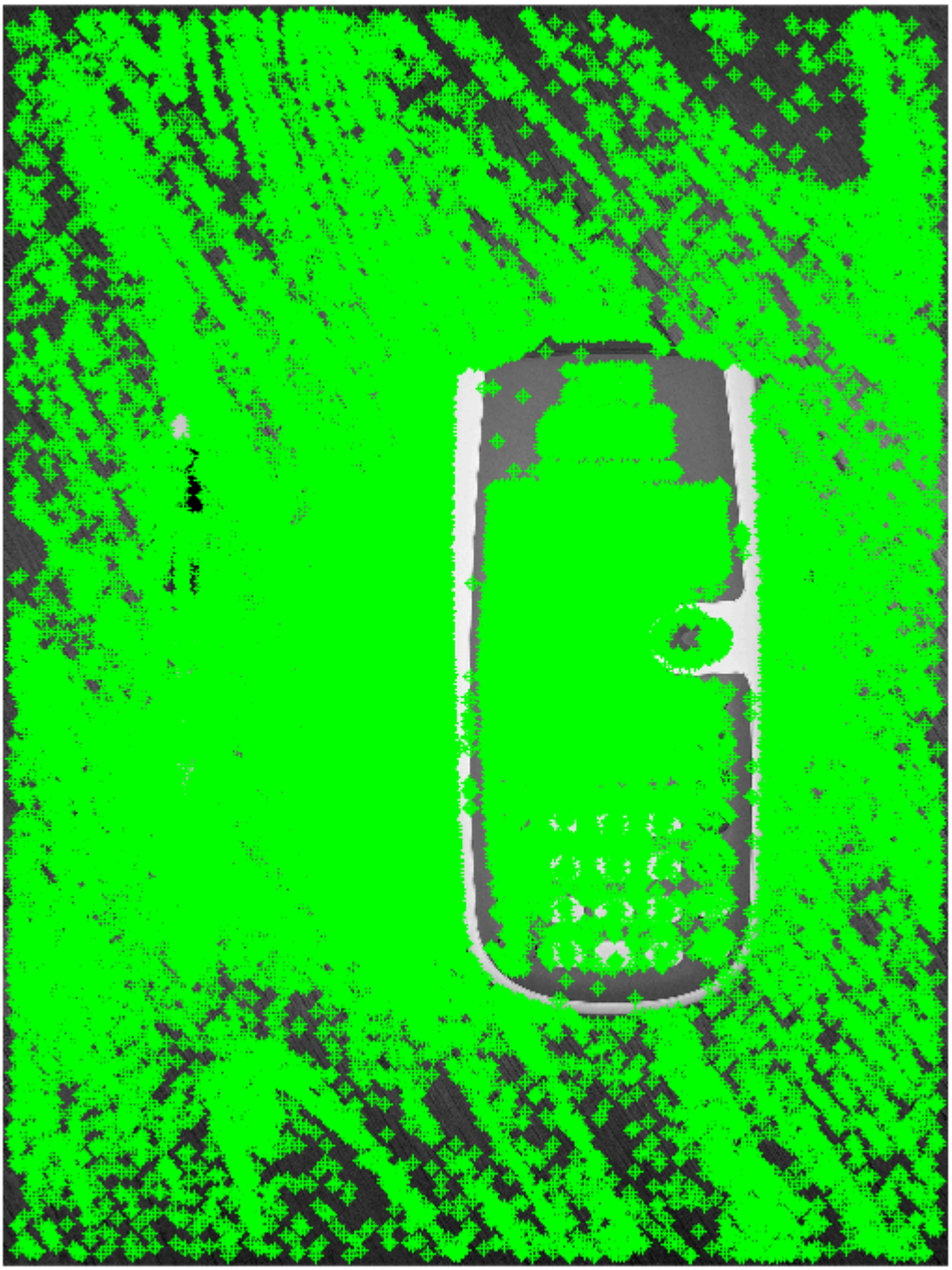
```
%orb
points = detectORBFeatures(I);
imshow(I)
```

```
hold on
plot(points,'ShowScale',false)
hold off
```



```
points = detectORBFeatures(I,'ScaleFactor',1.01,'NumLevels',3);
imshow(I)
```

```
hold on
plot(points)
hold off
```



```
%surf
points = detectSURFFeatures(I);
imshow(I)
```

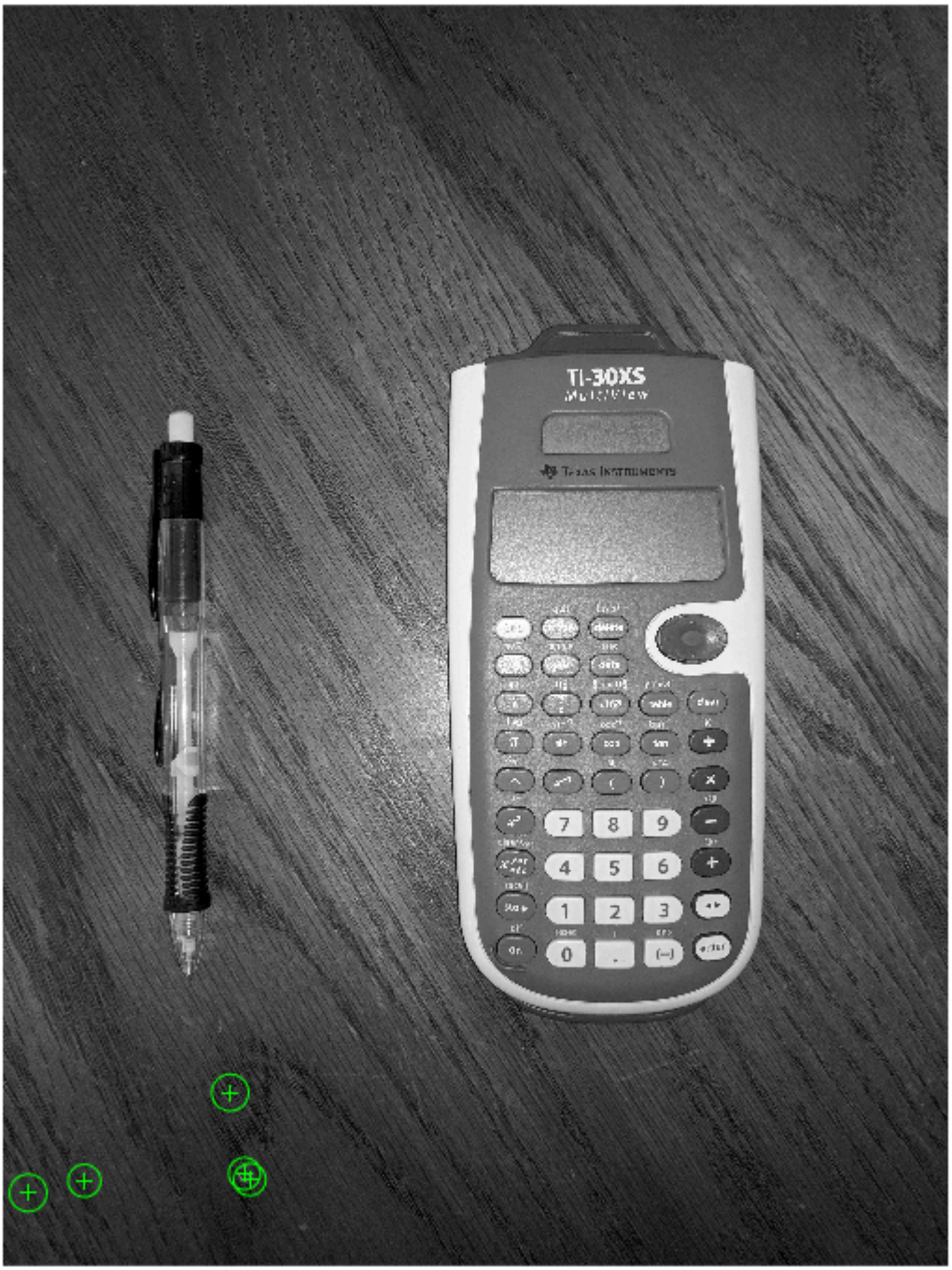
```
hold on
plot(points.selectStrongest(50));
hold off
```

```
%kaze
points = detectKAZEFeatures(I);
```

```
imshow(I)
hold on
plot(selectStrongest(points,50))
hold off
```

```
imshow(I)
```

```
hold on
plot(points(end-4:end));
hold off
```




```
%only finds bottle cap for some reason
```

```
addpath 'D:'\homework\'comp vision'\'image pairs'\

I = imread('calc_pencil_1.jpg');
I = rgb2gray(I);

%harris
corners = detectHarrisFeatures(I);
[features, valid_corners] = extractFeatures(I, corners);
figure
imshow(I)
hold on
plot(valid_corners);
```

```
%surf
points = detectSURFFeatures(I);
[features, valid_points] = extractFeatures(I, points);
figure
imshow(I)
hold on
```

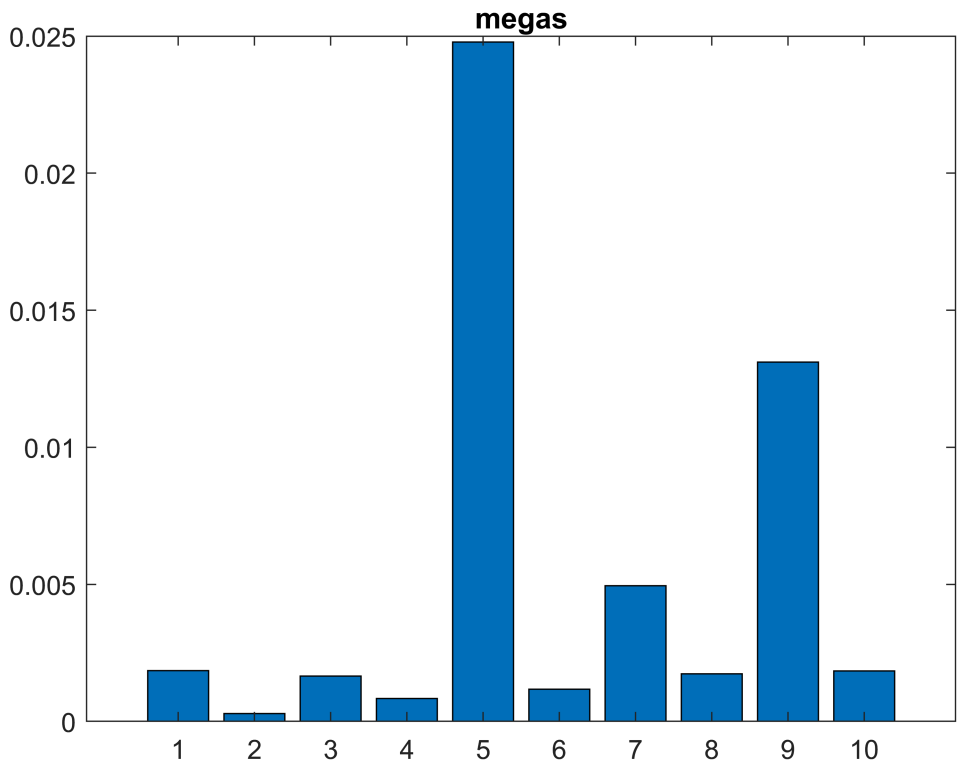
```
plot(valid_points.selectStrongest(20),'showOrientation',true);
```


```
%mser
regions = detectMSERFeatures(I);
[features, valid_points] = extractFeatures(I,regions,'Upright',true);
figure
```

```
imshow(I)
hold on
plot(valid_points,'showOrientation',true);
```

```
%perfect

%lbp
X = imread('calc_pencil_2.jpg');
```

```
X = rgb2gray(X);
lbp1 = extractLBPFeatures(I,'Upright',false);
lbp2 = extractLBPFeatures(X,'Upright',false);
megaVsMega = (lbp1 - lbp2).^2;

figure
bar([megaVsMega],'grouped')
title('megas')
```



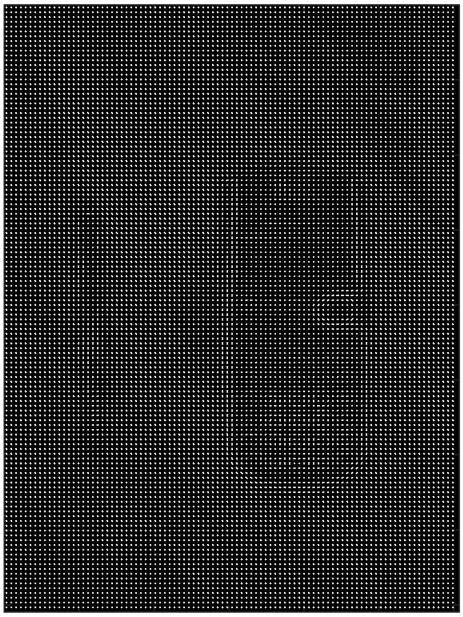

```
%hog
[featureVector,hogVisualization] = extractHOGFeatures(I);
figure
imshow(I)
hold on
plot(hogVisualization)
```

```
[hog1,visualization] = extractHOGFeatures(I,'CellSize',[32 32])
```

```
hog1 = 1×418500 single row vector
    0.1993    0.2112    0.2112    0.1665    0.1570    0.1259    0.1052    0.1260 ···
visualization =
Visualization

   Type plot(visualization) to visualize.
```

```
  Read-only properties:
              CellSize: [32 32]
             BlockSize: [2 2]
          BlockOverlap: [1 1]
               NumBins: 9
    UseSignedOrientation: 0
             BinCenters: [18×1 double]
```

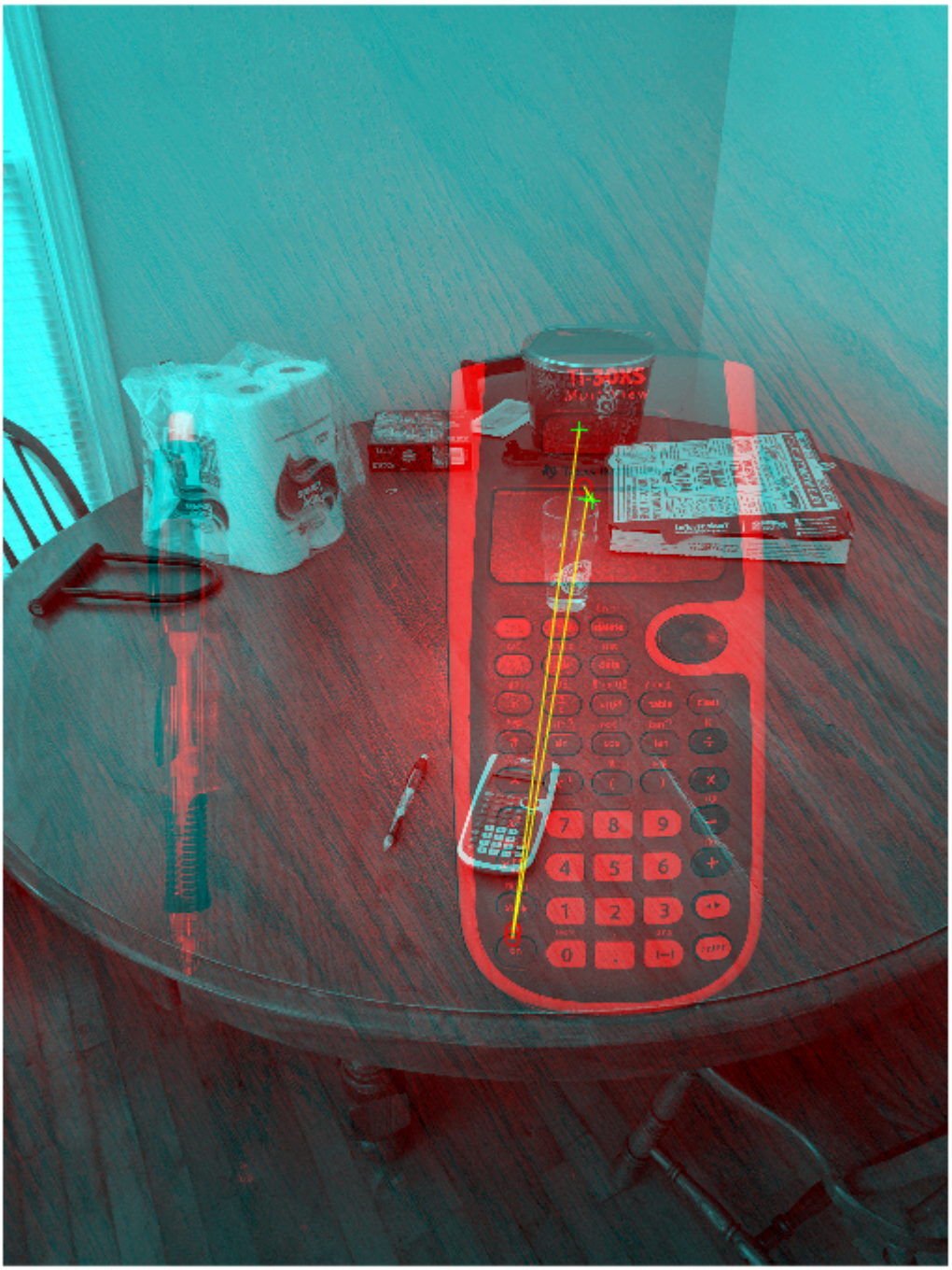
```
subplot(1,2,1);
imshow(I);
subplot(1,2,1);
plot(visualization);
```



```
%fast
corners = detectFASTFeatures(I);
strongest = selectStrongest(corners,3);
[hog, validPoints,ptVis] = extractHOGFeatures(I,strongest);
figure
imshow(I)
hold on
```

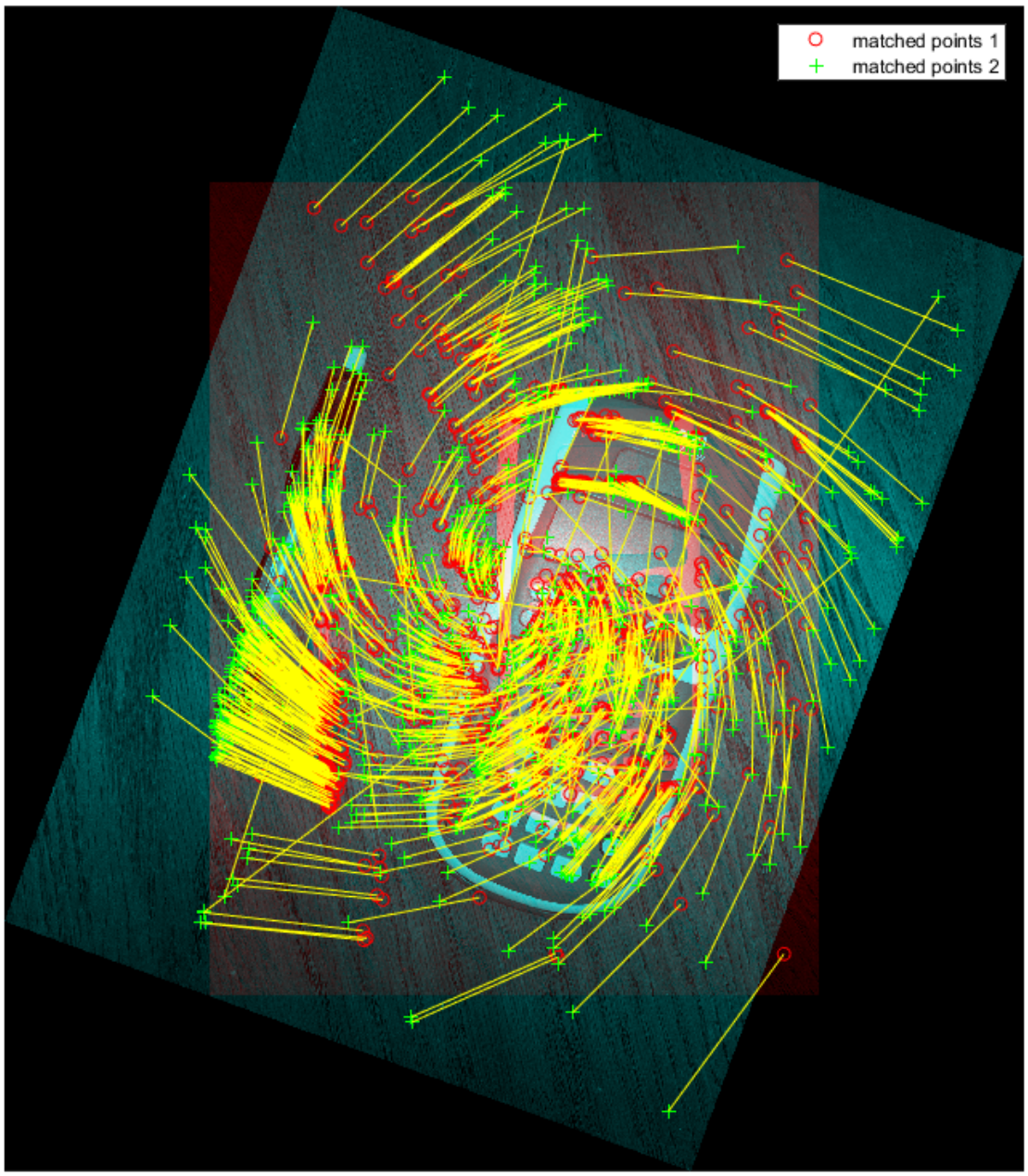
```
plot(ptVis,'Color','green');
```

```
addpath 'D:'\homework\'comp vision'\'image pairs'\

I = imread('calc_pencil_1.jpg');
I = rgb2gray(I);


Y = imread('calc_pencil_2.jpg');
Y = rgb2gray(Y);
%match features
points1 = detectHarrisFeatures(I);
points2 = detectHarrisFeatures(Y);

[features1,valid_points1] = extractFeatures(I,points1);
[features2,valid_points2] = extractFeatures(Y,points2);

indexPairs = matchFeatures(features1,features2);

matchedPoints1 = valid_points1(indexPairs(:,1),:);
matchedPoints2 = valid_points2(indexPairs(:,2),:);

figure; showMatchedFeatures(I,Y,matchedPoints1,matchedPoints2);
```



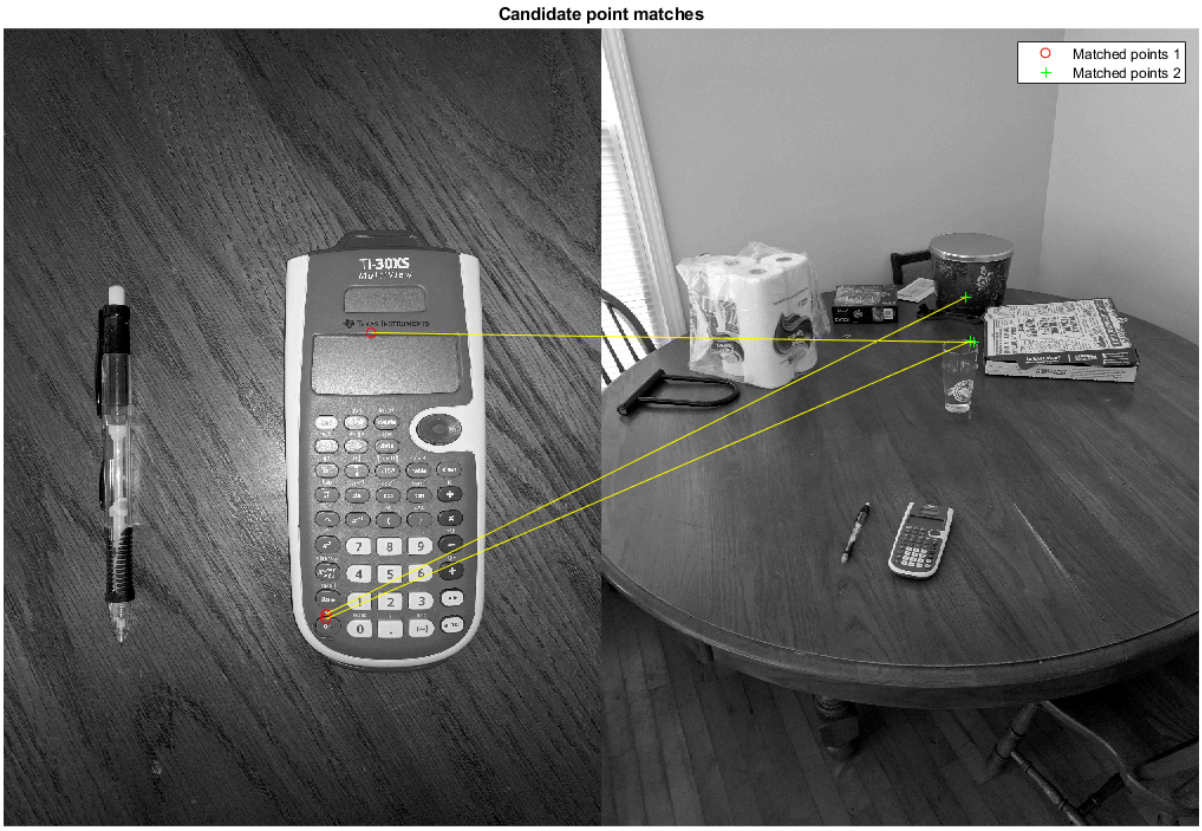
```
I2 = imresize(imrotate(I,-20),1.2);
points1 = detectSURFFeatures(I);
points2 = detectSURFFeatures(I2);
[f1,vpts1] = extractFeatures(I,points1);
[f2,vpts2] = extractFeatures(I2,points2);
indexPairs = matchFeatures(f1,f2) ;
matchedPoints1 = vpts1(indexPairs(:,1));
```

```
matchedPoints2 = vpts2(indexPairs(:,2));
figure; showMatchedFeatures(I,I2,matchedPoints1,matchedPoints2);
legend('matched points 1','matched points 2');
```




```
%showMatchedFeatures
points1 = detectHarrisFeatures(I);
points2 = detectHarrisFeatures(Y);


[f1, vpts1] = extractFeatures(I, points1);
```

```
[f2, vpts2] = extractFeatures(Y, points2);

indexPairs = matchFeatures(f1, f2) ;
matchedPoints1 = vpts1(indexPairs(1:3, 1));
matchedPoints2 = vpts2(indexPairs(1:3, 2));

figure; ax = axes;
showMatchedFeatures(I,Y,matchedPoints1,matchedPoints2,'montage','Parent',ax);
title(ax, 'Candidate point matches');
legend(ax, 'Matched points 1','Matched points 2');
```

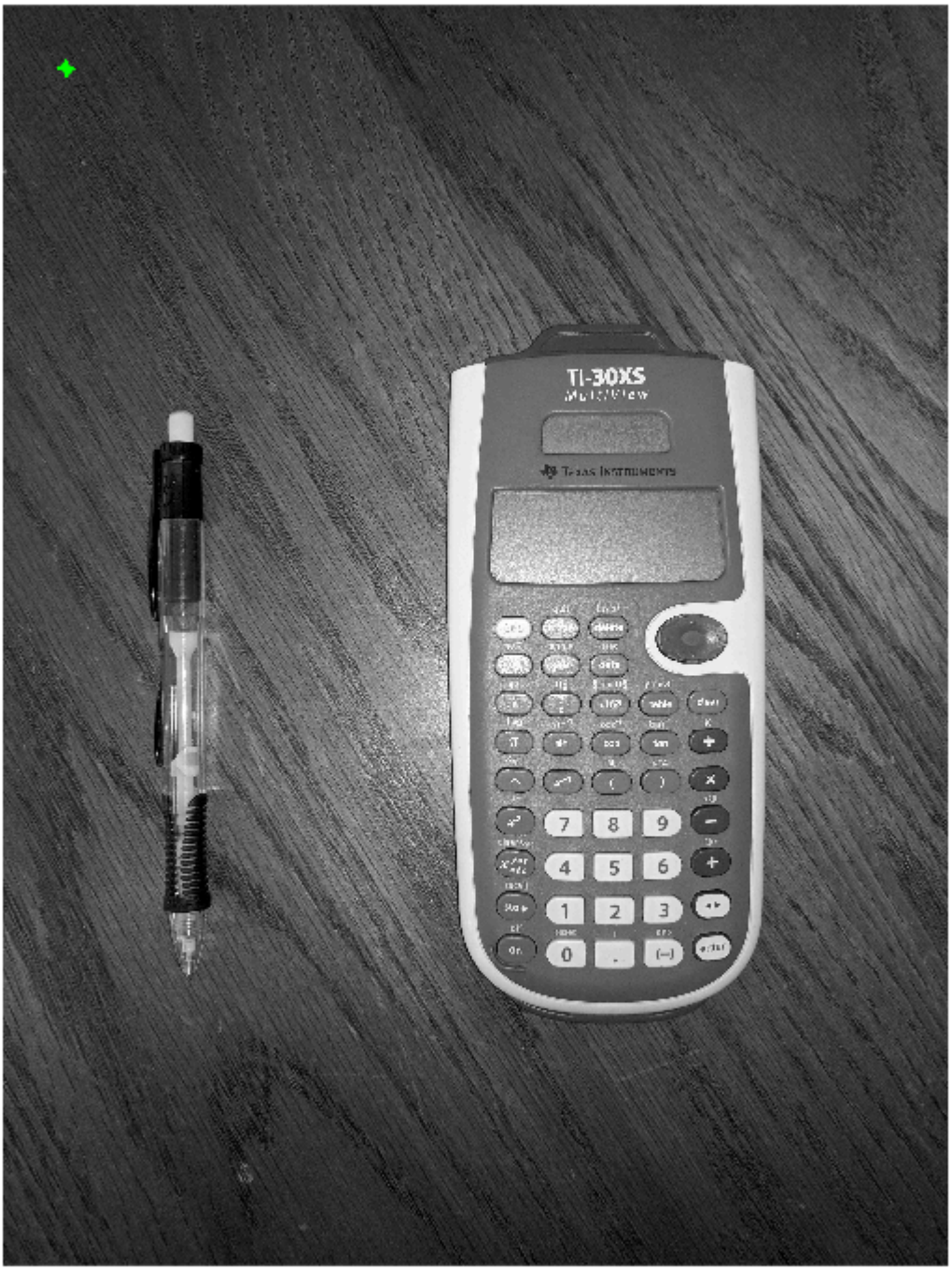



```
%doesn't work
```

```
addpath 'D:'\homework\'comp vision'\'image pairs'\

I = imread('calc_pencil_1.jpg');
I = rgb2gray(I);



%binary
%not entirely sure what values to put in for the binary features, just used
%those in the example
features1 = binaryFeatures(uint8([1 8 7 2; 8 1 7 2]));
features2 = binaryFeatures(uint8([8 1 7 2; 1 8 7 2]));

[indexPairs matchMetric] = matchFeatures(features1, features2)
```

```
indexPairs = 2×2 uint32 matrix
   1   2
   2   1
matchMetric = 2×1 single column vector
     0
     0
```

```
%brisk
points = detectBRISKFeatures(I);
location = [200:228;200:228]';
points = BRISKPoints(location);

strongest = points.selectStrongest(10);
imshow(I)
hold on
plot(strongest)
hold off
```




```
%garbage
strongest.Location
```

```
ans = 10×2 single matrix
   200   200
   201   201
   202   202
   203   203
   204   204
```

```
   205   205
   206   206
   207   207
   208   208
   209   209
```

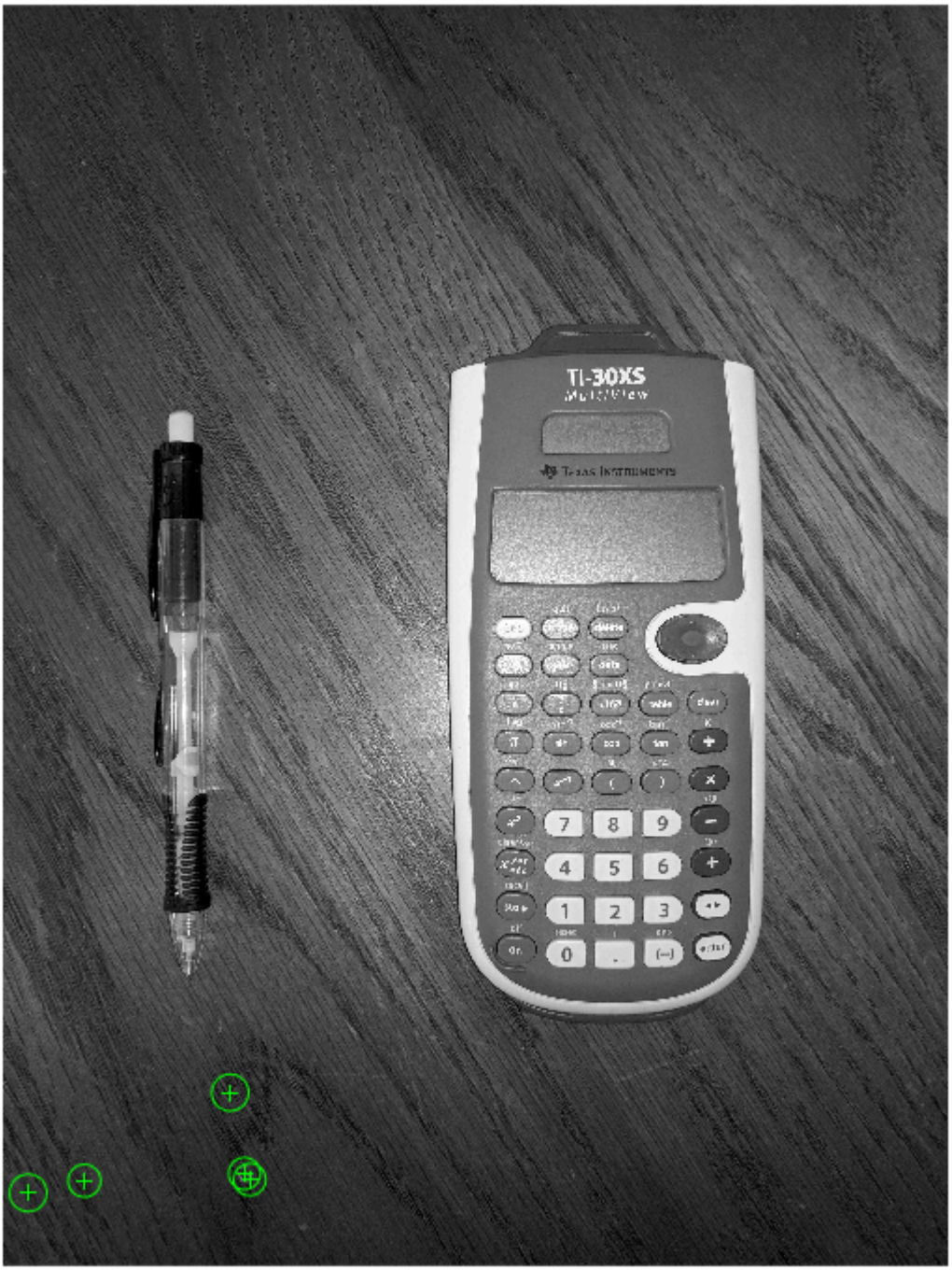
```
%kaze
points = detectKAZEFeatures(I);
strongest = selectStrongest(points,10);
imshow(I)
hold on
plot(strongest)
hold off
```


```
%only finds trophy with this low amount of points, but with a higher amount
%of points it also does well
strongest.Location
```

```
ans = 10×2 single matrix
10^3 ×
    0.4830    1.8931
    2.2694    2.8941
```

```
    0.4829    1.8932
    2.2952    2.8913
    2.2694    2.3777
    2.2696    2.3774
    2.2684    2.3521
    1.6751    2.9033
    2.1200    2.6219
    1.9623    2.6429
```

```
imshow(I)
hold on
plot(points(end-4:end)) %whenever i do this notation, i only find the bottlecap
hold off
```




```
%corner
points = detectHarrisFeatures(I);
strongest = selectStrongest(points,50);
imshow(I)
hold on
plot(strongest)
```

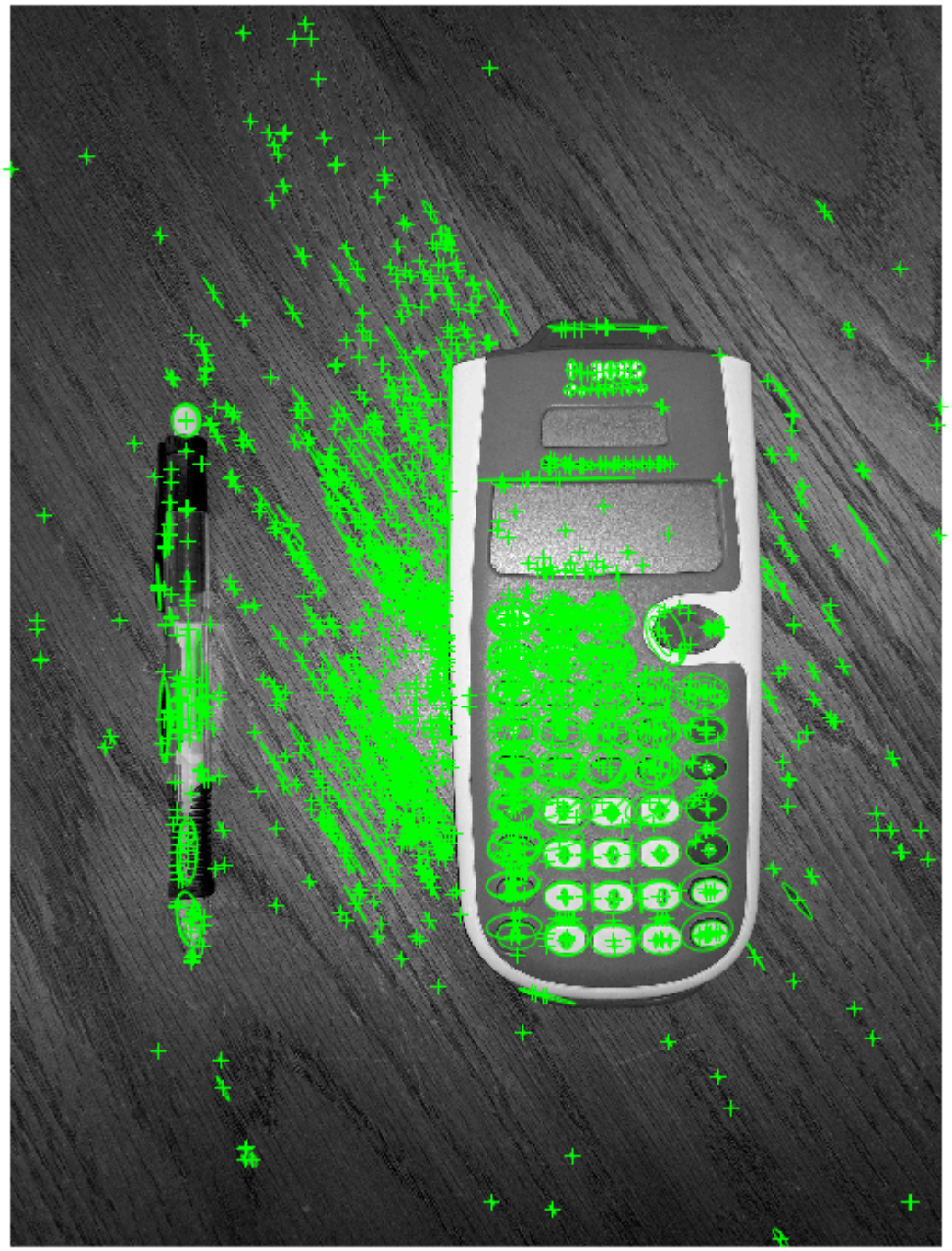
```
hold off
```

```
%surf
points = detectSURFFeatures(I);
strongest = points.selectStrongest(50);
imshow(I); hold on;
```

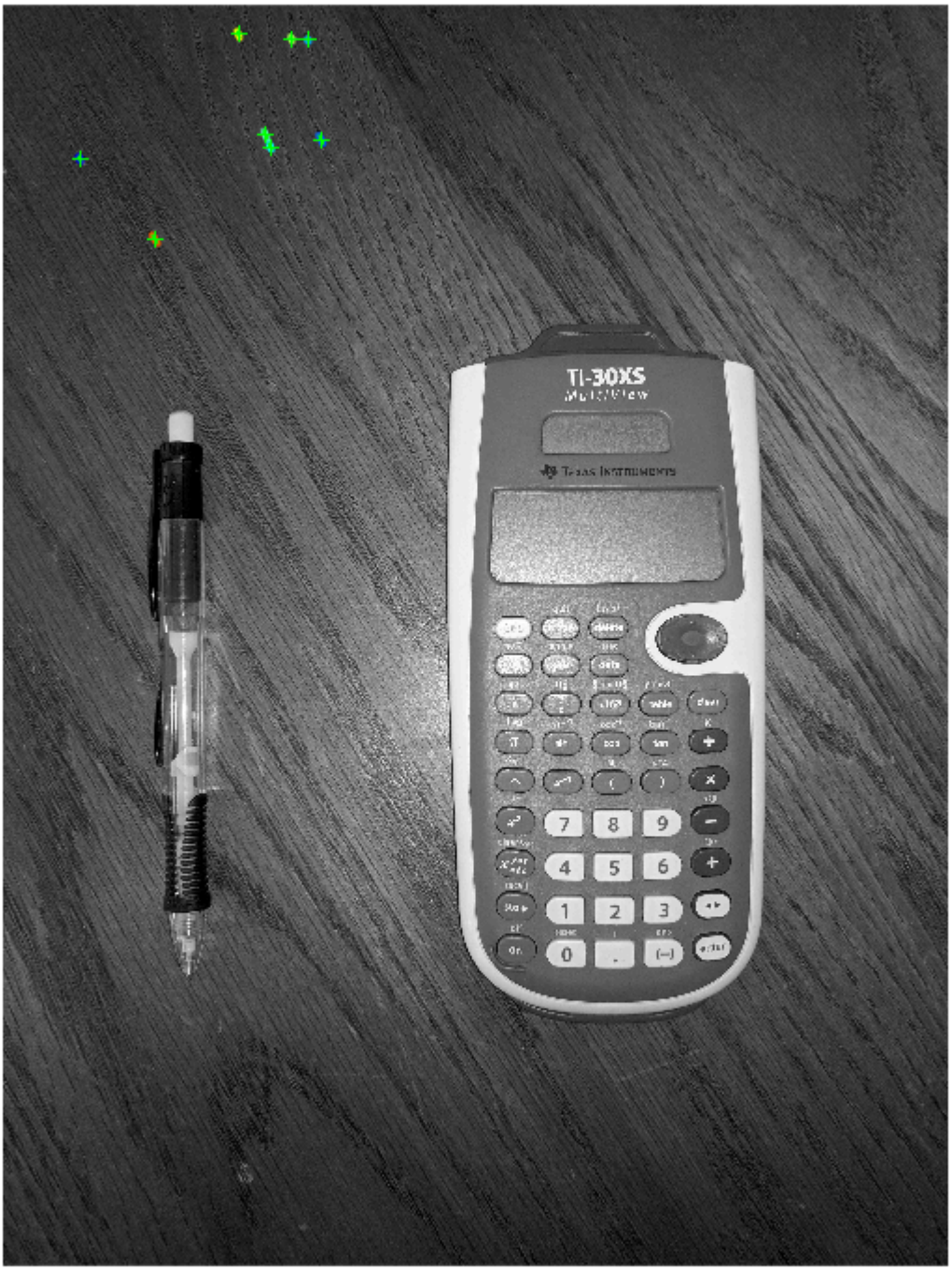
```
plot(strongest);
hold off
```

```
imshow(I); hold on;
plot(points(end-4:end));
hold off
```


```
%mser
regions = detectMSERFeatures(I);
imshow(I); hold on;
plot(regions);
hold off
```



```
imshow(I); hold on;
plot(regions(1:10),'showPixelList', true);
hold off
```




```
regionsObj = detectMSERFeatures(I);
[features, validPtsObj] = extractFeatures(I, regionsObj);
imshow(I); hold on;
plot(validPtsObj,'showOrientation',true);
hold off
```

```
%orb
points = detectORBFeatures(I);
newPoints = selectUniform(points,10,size(I))
```

```
newPoints =
  10×1 ORBPoints array with properties:
```

```
       Location: [10×2 single]
         Metric: [10×1 single]
          Count: 10
          Scale: [10×1 single]
    Orientation: [10×1 single]
```

```
figure
imshow(I)
hold on
plot(newPoints)
hold off
```


TI-30XS
Texas Instruments