

Vision Artificial. GIEC.

Vision Artificial. GIEC.

Sistemas de Vision Artificial. GIC.

Miguel Angel Garcia, Juan Manuel Miguel, Sira Palazuelos.

Departamento de Electrónica. Universidad de Alcalá.

Tema 4.2: ejercicio 05 - Local Features

El objetivo de esta práctica es mostrar el potencial de los denominados "Local Feature", para ello se va a buscar un objeto dentro de una escena utilizando los detectores y descriptores de características locales.

```
clear all  
close all
```

Step 1) Leer la imagen de referencia que contiene el objeto de interés.

```
Object1Image = imread('elephant.jpg');  
  
figure;  
imshow(Object1Image);  
title('Image of an Object1');
```

Image of an Object1



Se lee la escena que contiene al objeto de interés

```
sceneImage = imread('clutteredDesk.jpg');  
figure;  
imshow(sceneImage);  
title('Image of a Cluttered Scene');
```

Image of a Cluttered Scene

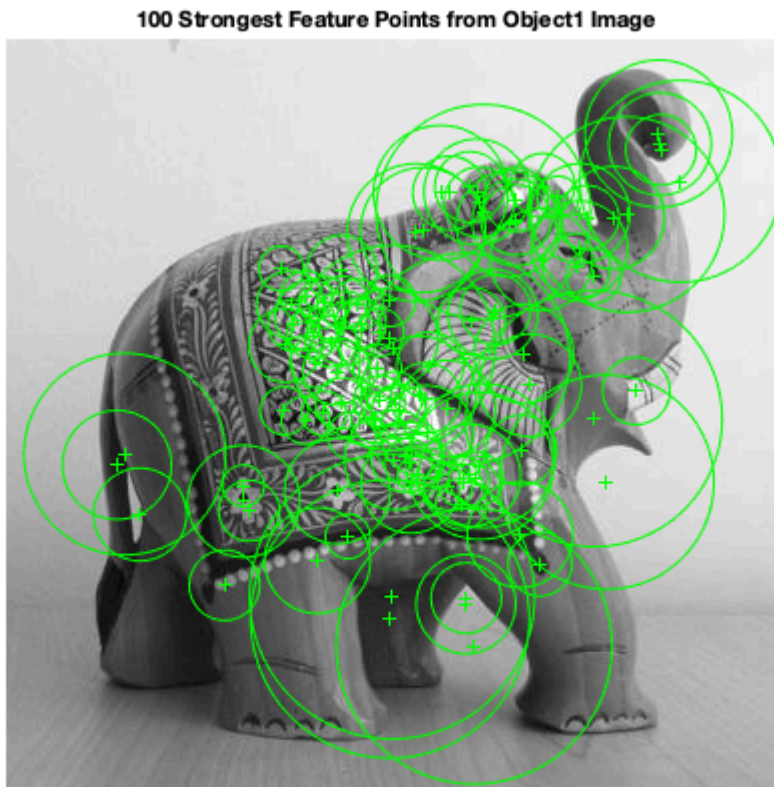


Step 2) Se detectan las características locales del objeto y de la escena

```
% Detect feature points in both images.  
Object1Points = detectSURFFeatures(Object1Image);  
scenePoints = detectSURFFeatures(sceneImage);
```

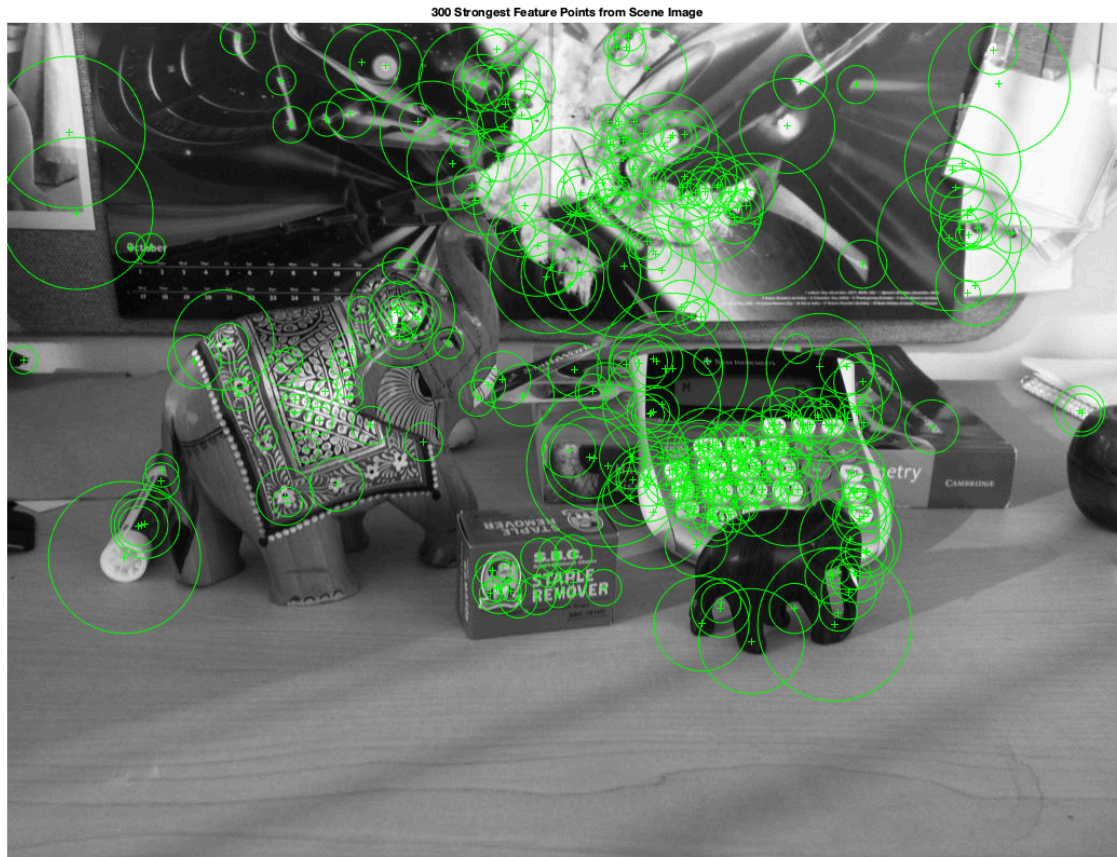
se visualizan los puntos más fuertes que se encuentran en la imagen del objeto de referencia

```
figure;  
imshow(Object1Image);  
title('100 Strongest Feature Points from Object1 Image');  
hold on;  
plot(selectStrongest(Object1Points, 100));
```



se visualizan los puntos más fuertes que se encuentran en la imagen de la escena

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



Step 3) Se extraen los descriptores de características de ambas imágenes

```
[Object1Features, Object1Points] = extractFeatures(Object1Image, Object1Points);
[sceneFeatures, scenePoints] = extractFeatures(sceneImage, scenePoints);
```

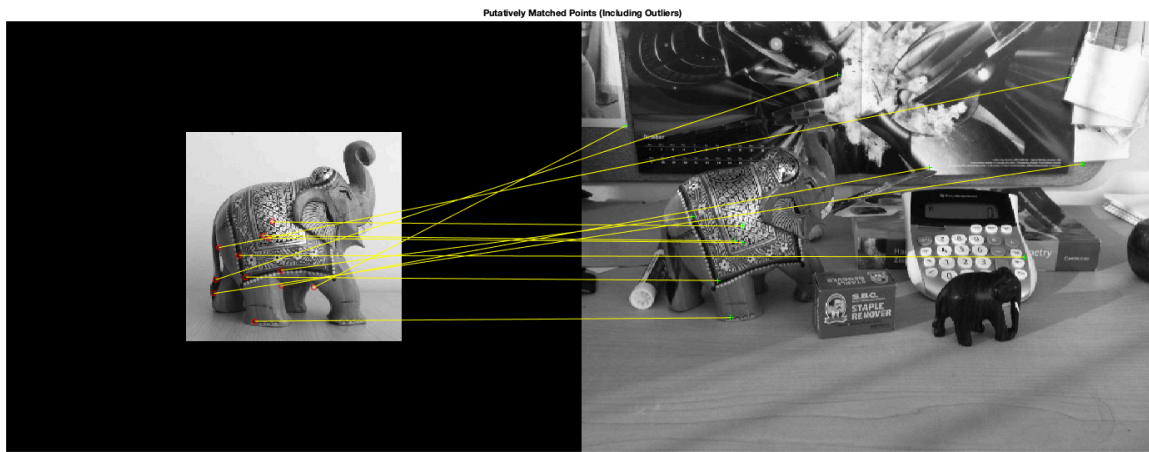
Step 4) Se encuentran las coincidencias (Matches) y se visualizan

```
Object1Pairs = matchFeatures(Object1Features, sceneFeatures, 'MaxRatio', 0.9);

matchedObject1Points = Object1Points(Object1Pairs(:, 1), :);
matchedScenePoints = scenePoints(Object1Pairs(:, 2), :);
figure;
showMatchedFeatures(Object1Image, sceneImage, matchedObject1Points, ...
    matchedScenePoints, 'montage');
```

Warning: Image is too big to fit on screen; displaying at 67%

```
title('Putatively Matched Points (Including Outliers)');
```



Step 5) Se localiza el objeto en la escena usando las coincidencias.

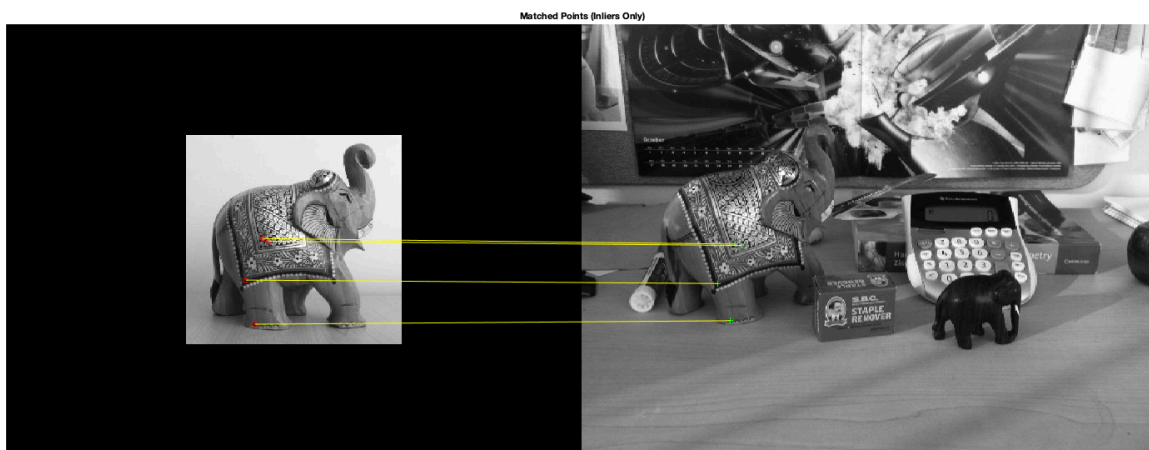
estimateGeometricTransform calcula la transformación relacionando los puntos coincidentes, a la vez que elimina los valores outliers. Esta transformación nos permite localizar el objeto en la escena.

```
[tform, inlierObject1Points, inlierScenePoints] = estimateGeometricTransform(matchedObj
figure;
showMatchedFeatures(Object1Image, sceneImage, inlierObject1Points, inlierScenePoints, '

```

Warning: Image is too big to fit on screen; displaying at 67%

```
title('Matched Points (Inliers Only)');
```

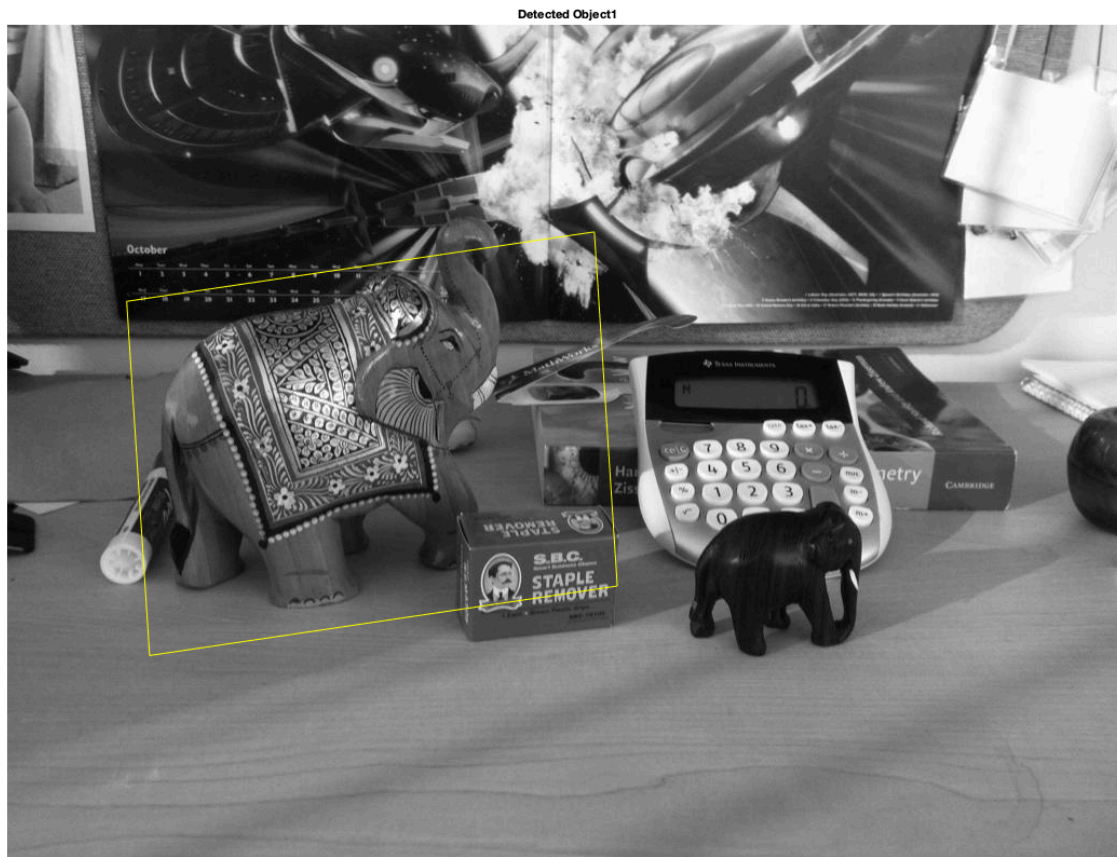


Se obtiene el polígono delimitador de la imagen de referencia.

```
Object1Polygon = [1, 1;... % top-left
                  size(Object1Image, 2), 1;... % top-right
                  size(Object1Image, 2), size(Object1Image, 1);... % bottom-right
                  1, size(Object1Image, 1);... % bottom-left
                  1, 1]; % top-left again to close the polygon
```


Se transforma el polígono en el sistema de coordenadas de la imagen de destino. El polígono transformado indica la ubicación del objeto1 en la escena.

```
newObject1Polygon = transformPointsForward(tform, Object1Polygon);  
  
figure;  
imshow(sceneImage);  
hold on;  
line(newObject1Polygon(:, 1), newObject1Polygon(:, 2), 'Color', 'y');  
title('Detected Object1');
```



Se pide:

1. Repita el proceso anterior pero buscando otro objeto de la escena, en este caso la caja, cuya imagen es 'stapleRemover.jpg'. Además utilice el detector y descriptor BRISK ([detectBRISKFeatures](#)).