

Características de imagen

Temario

- Definiciones
- Detección de esquinas:
 - Detector de Harris
 - Detector de Shi-Tomasi
- Detección de características:
 - SIFT
 - ORB



Características

- En inglés features.
- Son partes “interesantes” de una imagen.
- Ejemplos: bordes, esquinas, blobs, crestas.

Ejemplo



Fuente: https://docs.opencv.org/4.x/df/d54/tutorial_py_features_meaning.html

Características

- Actividades:
 1. Detección de características.
 2. Descripción / representación de características.
 3. Extracción de características.
 4. Matching.

Detección de características

Common feature detectors and their classification:

Feature detector	Edge	Corner	Blob	Ridge
Canny ^[2]	Yes	No	No	No
Sobel	Yes	No	No	No
Harris & Stephens / Plessey ^[3]	Yes	Yes	No	No
SUSAN ^[4]	Yes	Yes	No	No
Shi & Tomasi ^[5]	No	Yes	No	No
Level curve curvature ^[6]	No	Yes	No	No
FAST ^[7]	No	Yes	Yes	No
Laplacian of Gaussian ^[6]	No	Yes	Yes	No
Difference of Gaussians ^{[8][9]}	No	Yes	Yes	No
Determinant of Hessian ^[6]	No	Yes	Yes	No
Hessian strength feature measures ^{[10][11]}	No	Yes	Yes	No
MSER ^[12]	No	No	Yes	No
Principal curvature ridges ^{[13][14][15]}	No	No	No	Yes
Grey-level blobs ^[16]	No	No	Yes	No

Fuente: [https://en.wikipedia.org/wiki/Feature_\(computer_vision\)#Detection](https://en.wikipedia.org/wiki/Feature_(computer_vision)#Detection)

Detección de características con OpenCV

Método	Funcionalidad
Harris	Esquinas
SIFT	Blobs
SURF	Blobs
FAST	Esquinas
BRIEF	Blobs
ORB	Esquinas y blobs

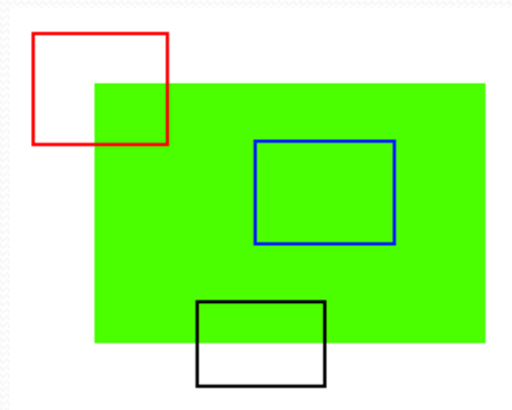


Matching de características con OpenCV

- Fuerza bruta.
- FLANN.

Detector de esquinas de Harris

- Las esquinas son regiones en una imagen con una gran varianza en intensidad en todas las direcciones.

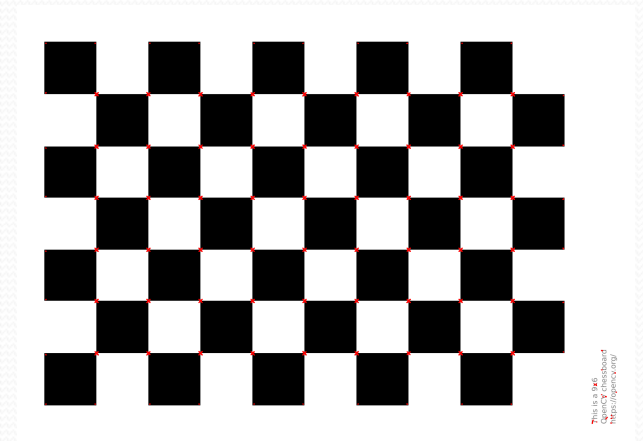
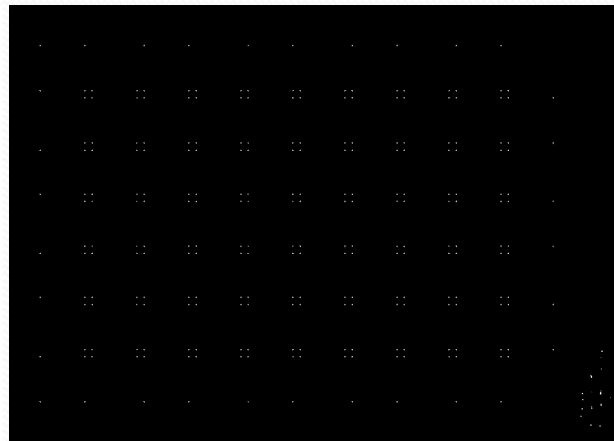
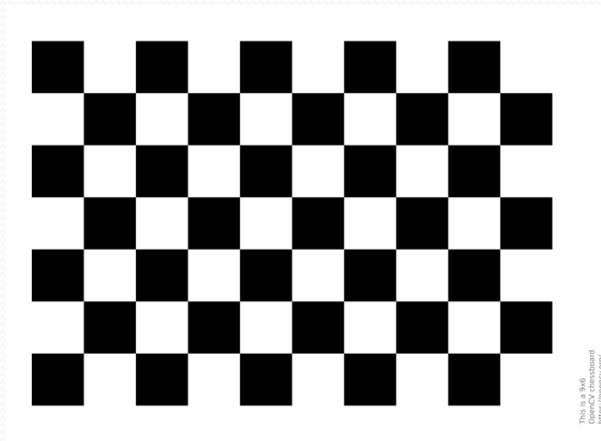


Fuente: https://docs.opencv.org/4.x/df/d54/tutorial_py_features_meaning.html

En OpenCV

- Usar el método: `cv2.cornerHarris()`.
- Ejemplo disponible en: `image_features/corner_harris.py`.

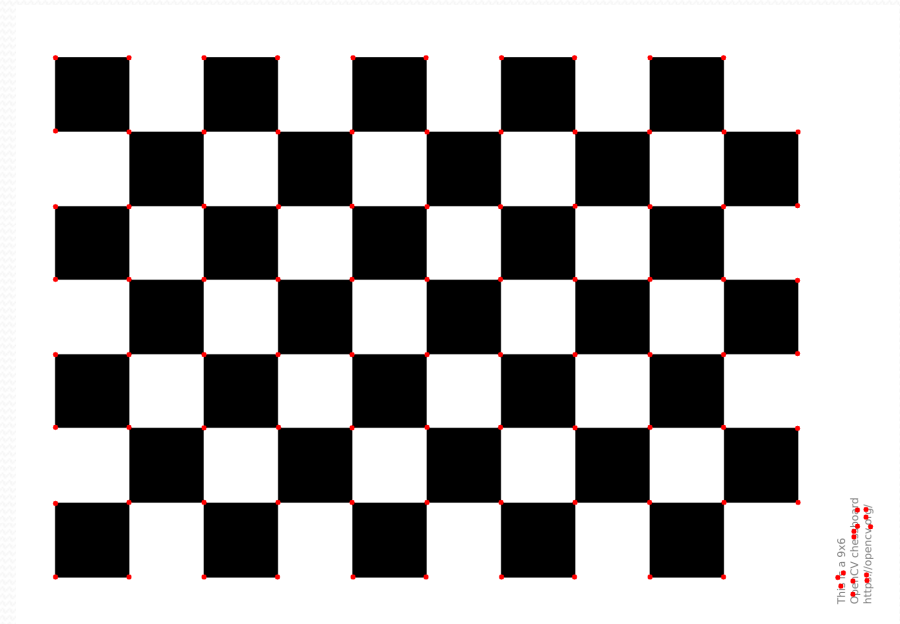
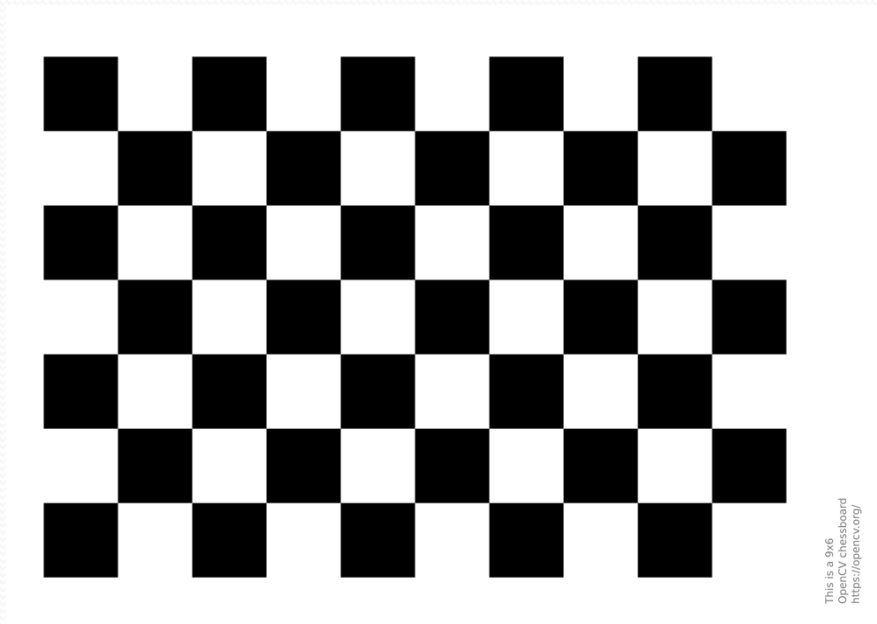
Resultado



Detector de esquinas de Shi-Tomasi

- Generalmente obtiene mejores resultados que el detector de Harris.
- En OpenCV usar el método: `cv.goodFeaturesToTrack()`.
- Ejemplo en: `image_features/corners_shi_tomasi.py`.

Resultado



SIFT

- Scale-invariant feature transform.
- Permite detectar y describir características en imágenes.
- Las características son invariantes a la escala y a la rotación.
- Los detectores de Harris y Shi-Tomasi no son invariantes a la escala.
- SIFT regresa una lista de puntos de interés (keypoints) y una lista con el descriptor de cada keypoint.

SIFT en OpenCV

- El primer paso es crear un objeto SIFT:
`sift = cv.SIFT.create()`
- Luego, obtener los keypoints y, opcionalmente, los descriptores:
`key_points = sift.detect()`
`key_points, descriptors = sift.detectAndCompute()`
- Por último, dibujar los keypoints:
`cv.drawKeypoints(image, keypoints=key_points, outImage=image, color=(51, 163, 236), flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)`

Clase KeyPoint

- angle – orientación calculada del keypoint (-1 si no corresponde)
- class_id – ID de objeto; se puede usar para agrupar keypoints por el objeto al que pertenecen
- octave – capa en la pirámide de la que se ha extraído el keypoint
- pt – coordenadas x , y del keypoint
- response – la fuerza del keypoint
- size – diámetro del área adyacente al keypoint

Imprimir un keypoint

File: print_keypoint.py

prints a keypoint

```
def print_keypoint(kp):
```

```
    print(f'angle: {kp.angle}, class_id: {kp.class_id}, octave: {kp.octave}, \\  
pt: {kp.pt}, response: {kp.response}, size: {kp.size}')
```



Ejemplo

- Disponible en `image_features/shift.py`.

Resultado



Imagen original



Imagen con los keypoints

ORB

- Oriented FAST and rotated BRIEF.
- Es un detector de características.
- Es una alternativa rápida y eficiente a SIFT.
- Se basa en el detector de keypoints FAST y en el descriptor de características BRIEF.
- ORB utiliza una versión orientada de FAST y le llama oFAST.
- ORB utiliza un descriptor binario basado en BRIEF y le llama rBRIEF.

ORB en OpenCV

- Crear un objeto ORB:
`orb = cv.ORB.create()`
- Obtener los keypoints y los descriptores:
`keypoints, descriptors = orb.detectAndCompute()`
- Dibujar los keypoints:
`cv.drawKeypoints()`

Ejemplo

- Disponible en `image_features/orb.py`.



Comparación SIFT vs ORB



SIFT (885 keypoints)



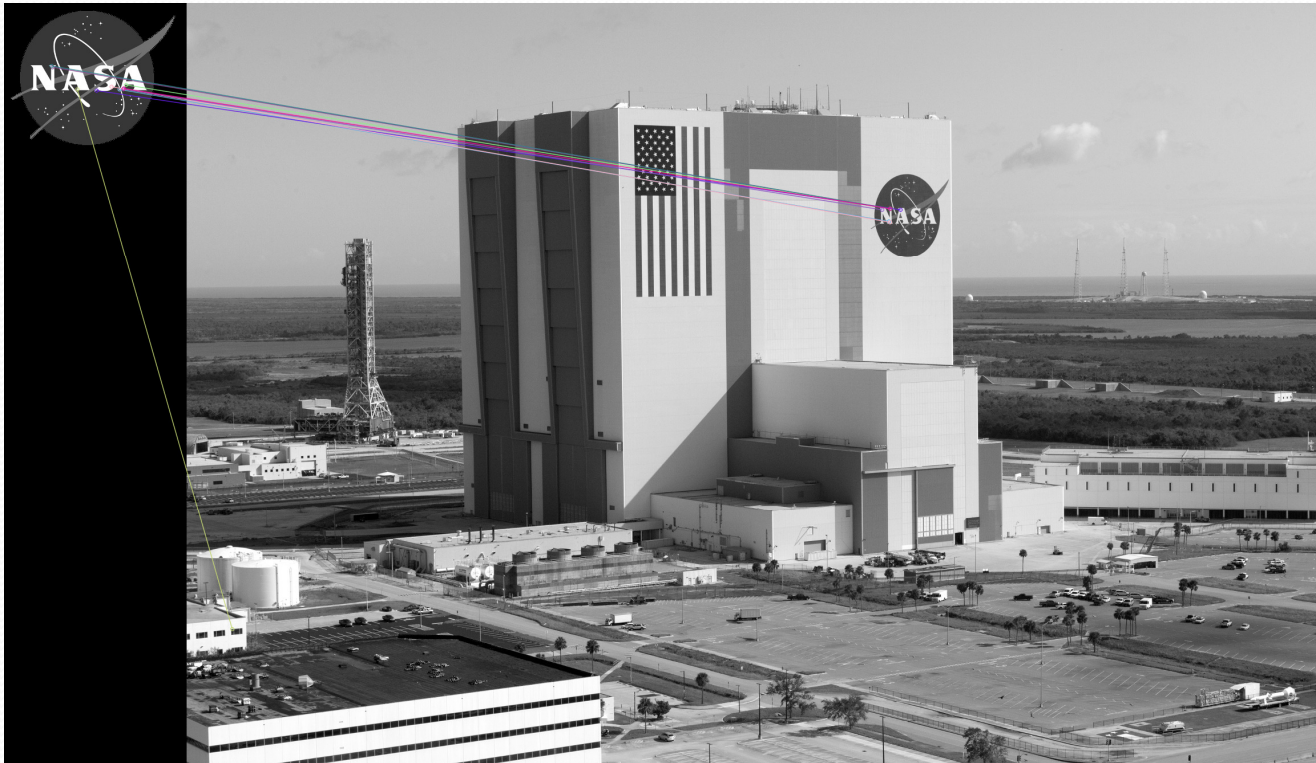
ORB (3,750 keypoints)



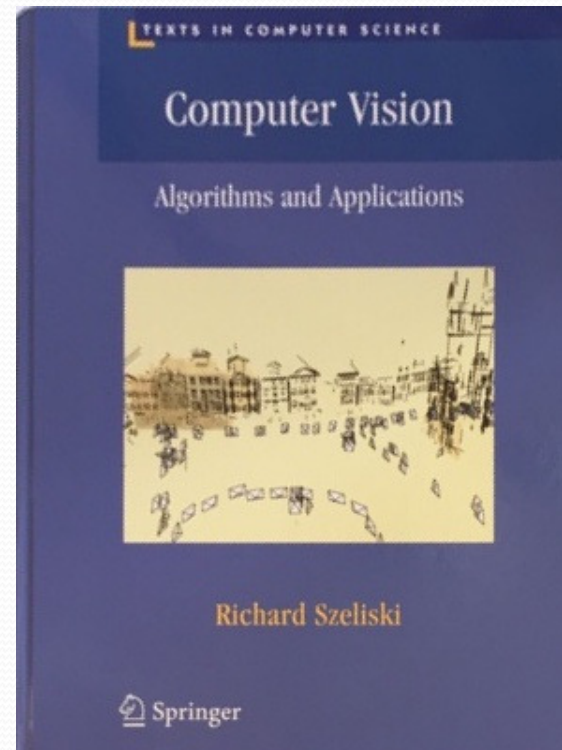
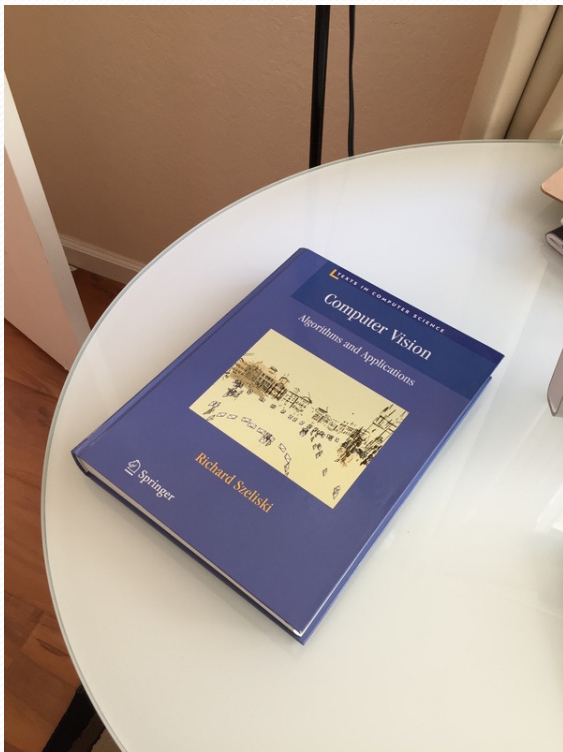
Utilización

- Matching de imágenes.
- Alineación de imágenes.
- Pegado de imágenes.

Matching de imágenes



Alineación de imágenes



Pegado de imágenes

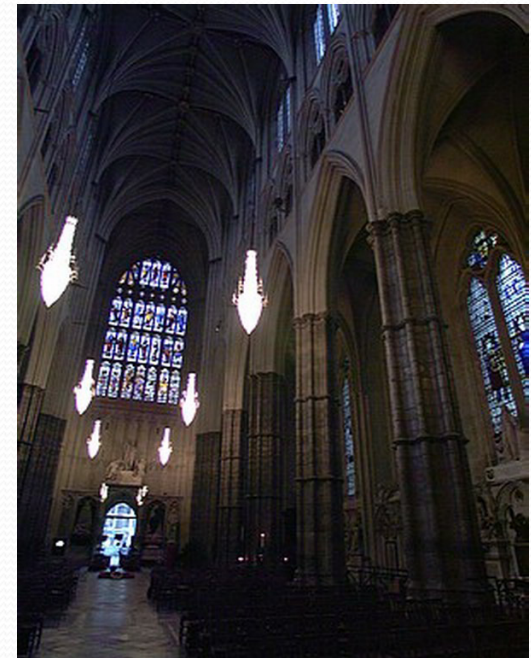
Izquierda



Centro



Derecha



Pegado de imágenes

Panorama

