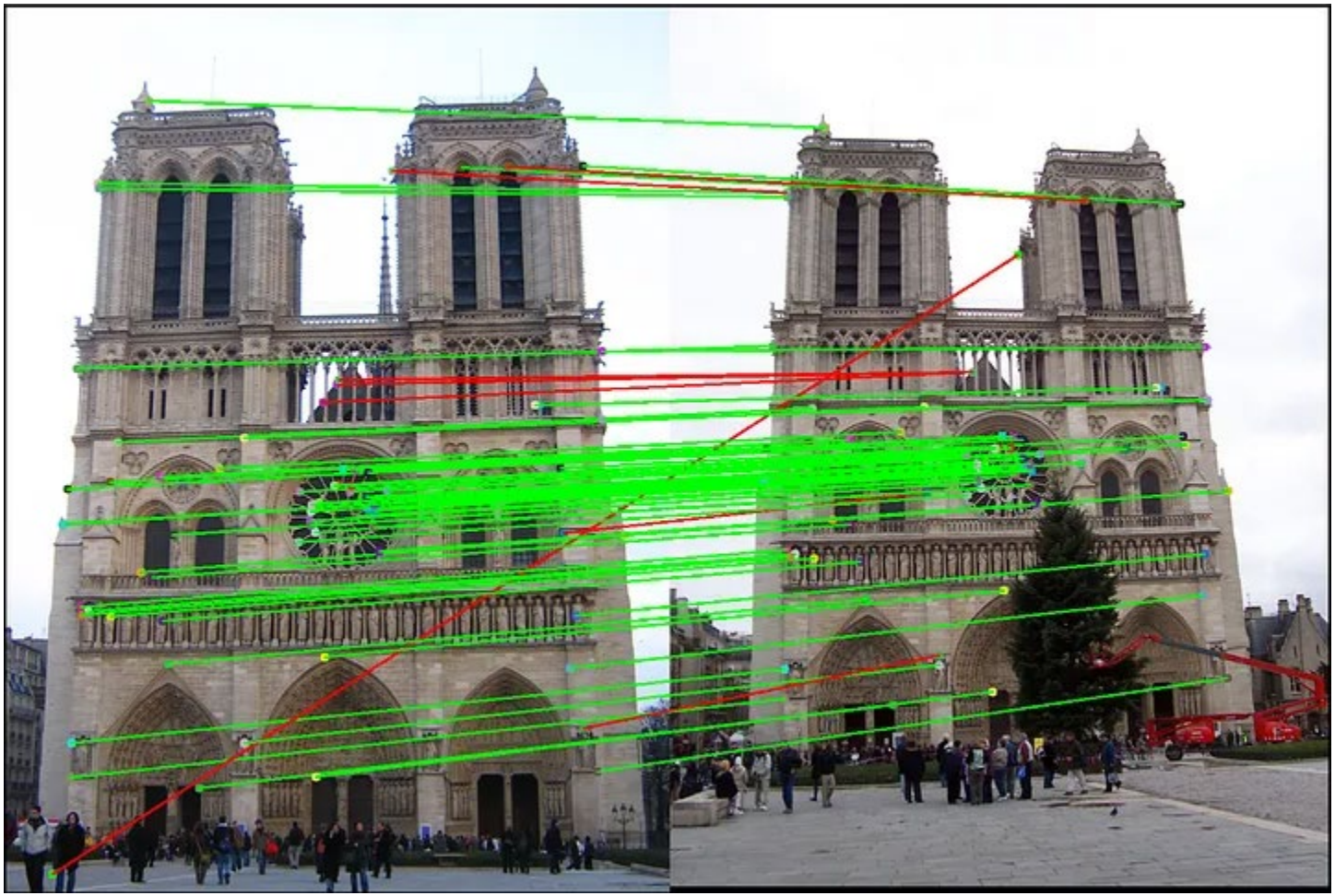
Visión por Computador

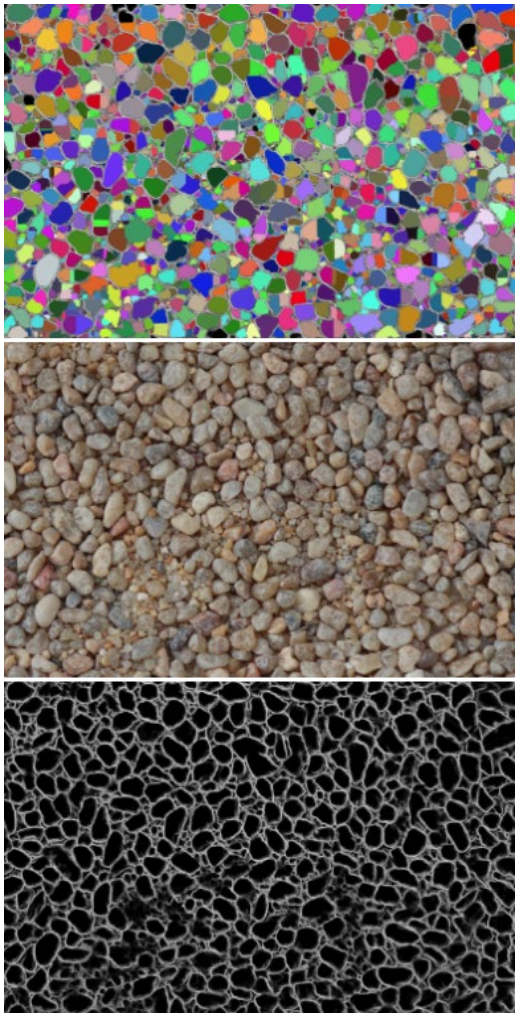
Tema 4. Detección de características



EII, GII. *Modesto Castrillón Santana*

Contenidos

- Introducción
- Puntos
- Bordes
- Líneas
- Segmentación

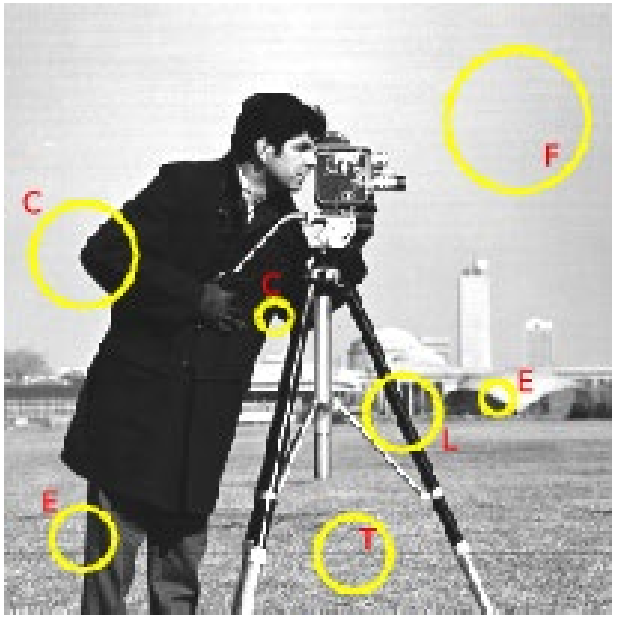


Introducción

Detección de características y encaje (*matching*)

Fundamental en múltiples aplicaciones VC

Puntos, esquinas, líneas, contornos y parches (*patches*)

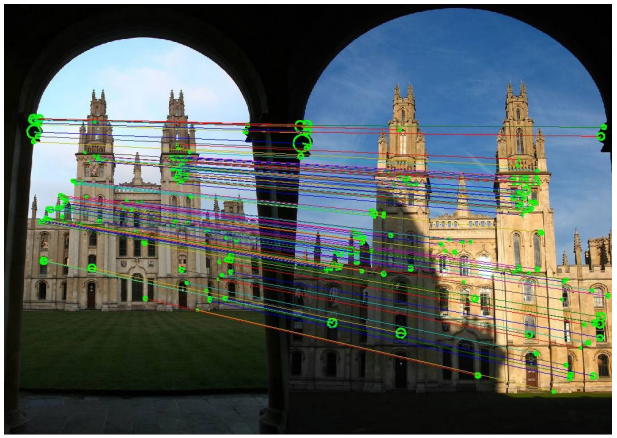


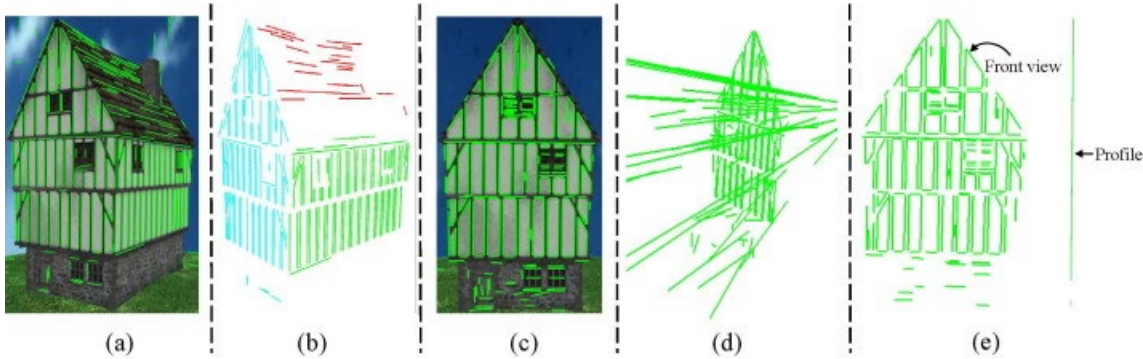
Aplicaciones

Detección de elementos y formas

Aplicaciones

Alineamiento/encaje de imágenes desde puntos o líneas





Aplicaciones

Alineamiento/encaje de imágenes desde puntos o líneas

¿Qué es un punto de interés?

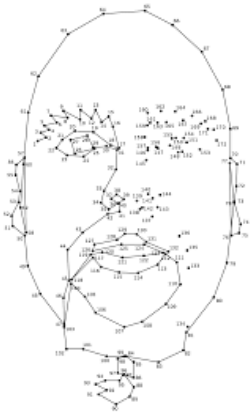


Imagen original o preprocesada, p.e. bordes

Presencia de elementos singulares/característicos:

- Extremos
- Cambios de dirección

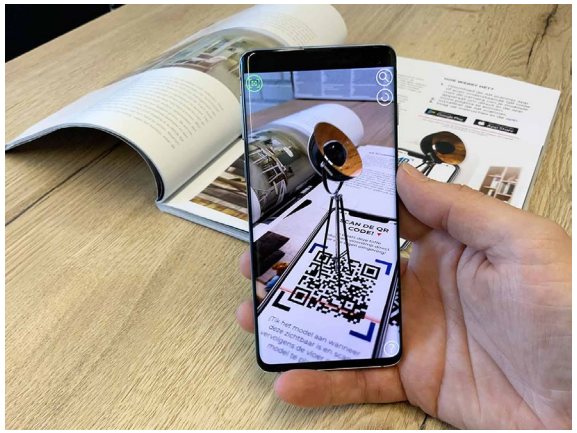
¿Qué es un punto de interés?

Correspondencia entre imágenes
¿Es el mismo objeto?

Aplicaciones

Alineamiento de imágenes:

- Calibración de cámara
- Encaje estéreo
- Reconocimiento de patrones
- *Stitching* de imágenes (mosaicos)
- Consulta de imágenes (*image retrieval*)
- Seguimiento
- Estabilización de vídeo (cámaras, AR)



Buen punto de interés

¿Cuál(es) es/son más útil(es) para la correspondencia entre imágenes?

Buen punto de interés

¿Cuál(es) es/son más útil(es) para la correspondencia entre imágenes?

Distintivo y repetible

Buen punto de interés

Correspondencia de puntos:

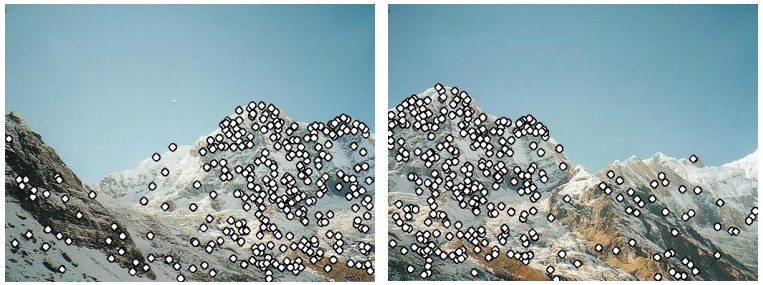
- Detectar en una imagen
  Vídeo, elementos próximos, imágenes sucesivas
  Encaje por correlación en zonas cercanas (seguimiento)

- Detectar en imágenes, encajar en base a apariencia local
  Más general

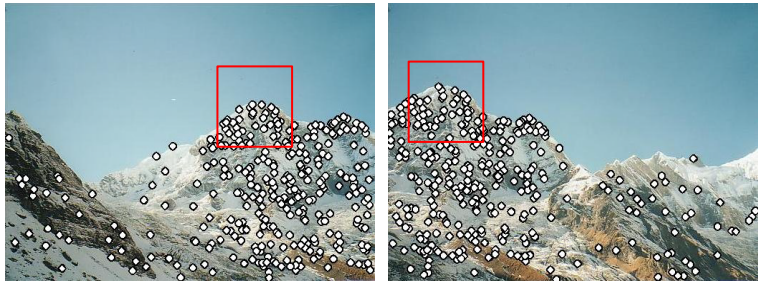
Proceso

Etapas:

- Detección
  Localizar puntos de interés

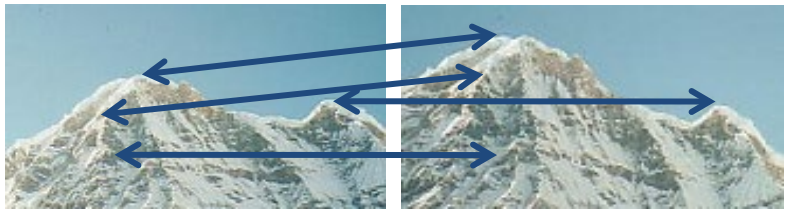


- Descripción
  Caracterizar puntos de interés



$$x_1 = [x_1^1, x_2^1, \dots, x_d^1] \qquad x_2 = [x_1^2, x_2^2, \dots, x_d^2]$$

- Encaje
  Determinar la correspondencia

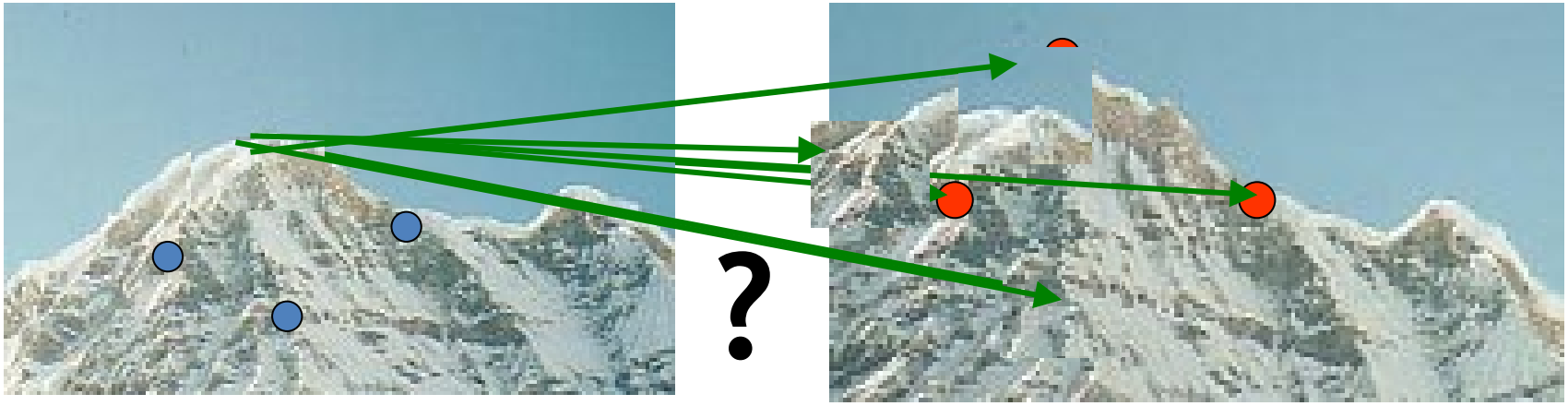


Szeliski

Etapas

Determinar correspondencia de puntos en varias imágenes

Dificultad en contextos con repetición de elementos

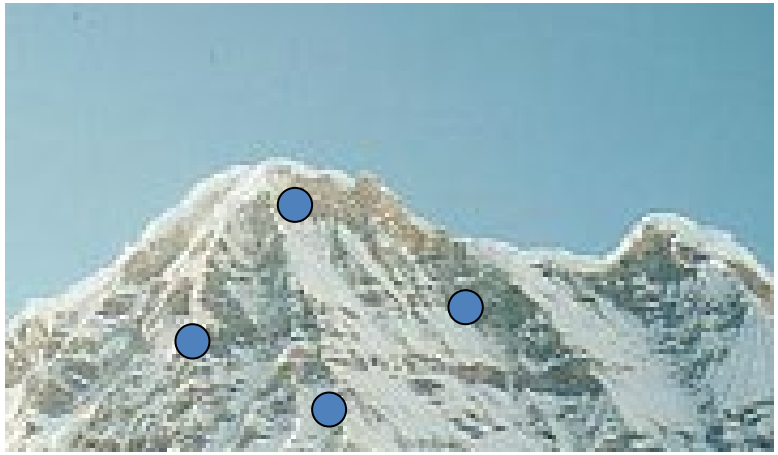


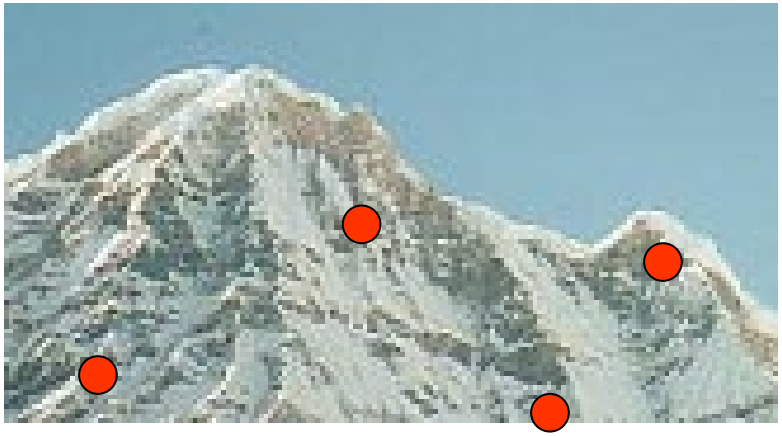
Szeliski

Detección

Características:

- **Distintivos**: Cada elemento único
- **Repetible**: Localizables en múltiples imágenes, aún con variaciones geométricas y fotométricas
- **Compacidad**: Descripción reducida << que la imagen
- **Eficiencia**: Al comparar/encajar



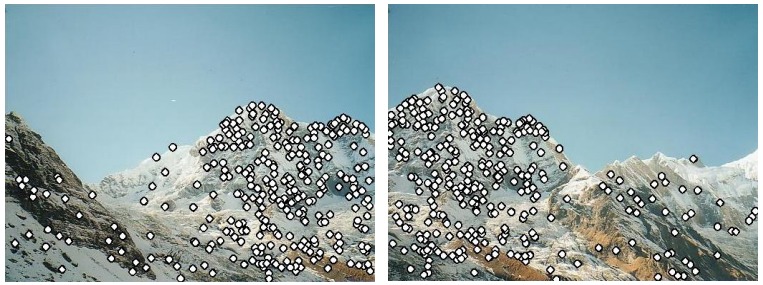


Szeliski

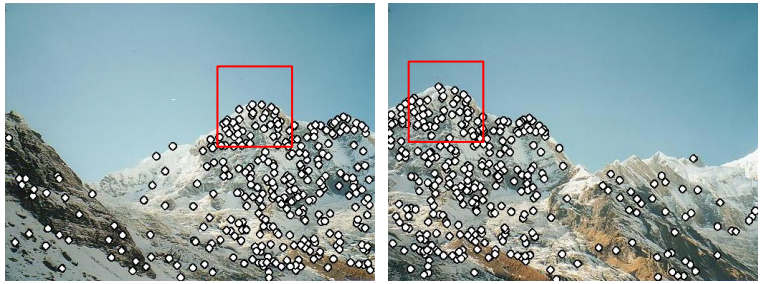
Proceso

Etapas:

- **Detección**
  Localizar puntos de interés

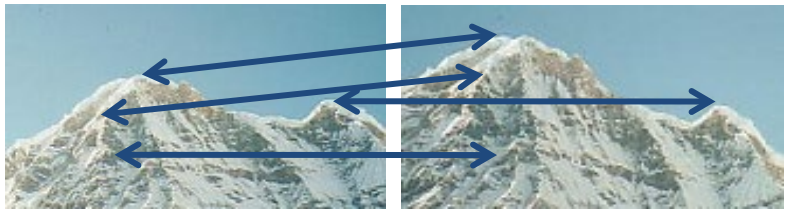


- Descripción
  Caracterizar puntos de interés



$$x_1 = [x_1^1, x_2^1, \dots, x_d^1] \qquad x_2 = [x_1^2, x_2^2, \dots, x_d^2]$$

- Encaje
  Determinar la correspondencia



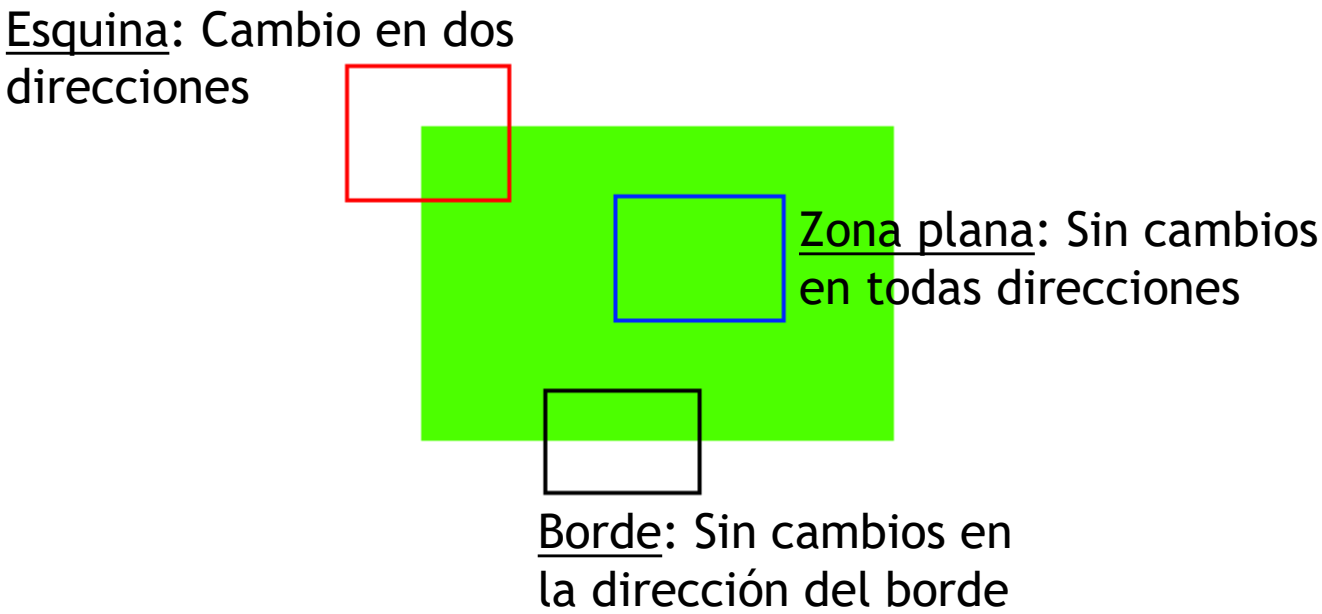
Szeliski

Detección

Un punto de interés presenta cambios en su entorno

Variaciones de intensidad en la vecindad

Estabilidad local frente a pequeños desplazamientos

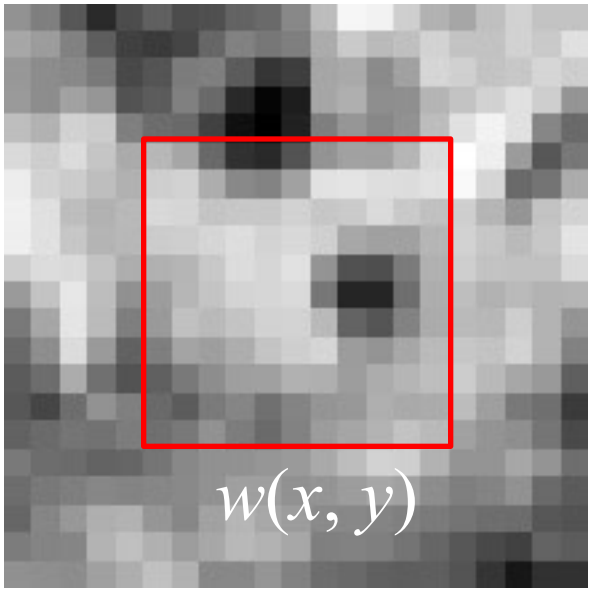


Detección

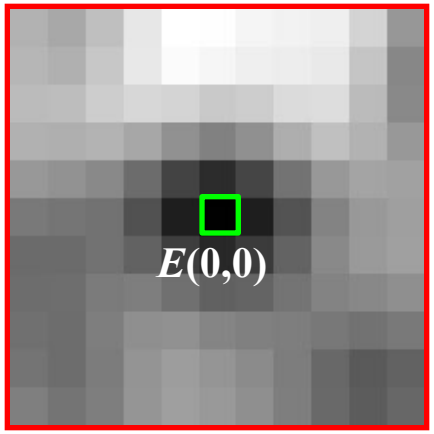
Autocorrelación en ventana *w* con desplazamiento *(u,v)*
Suma de las diferencias entre dos trozos de imagen

$$E(u,v) = \sum_{x,y} w(x,y)[f(x+u, y+v) - f(x,y)]^2$$

$f(x, y)$



$E(u, v)$



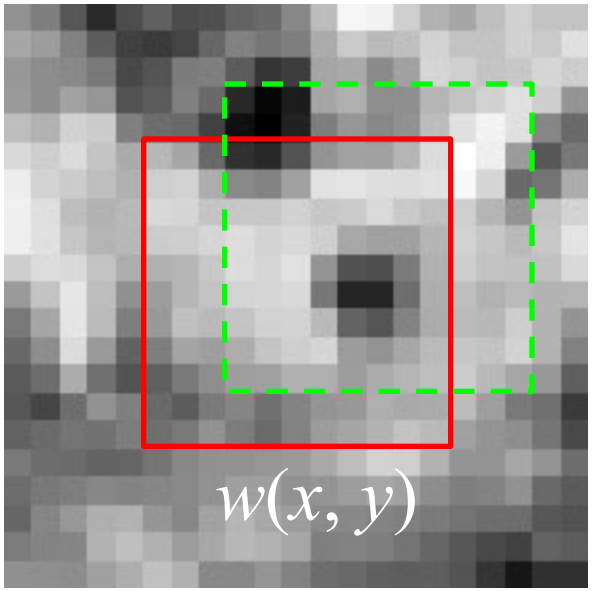
Negro, mínimo

Detección

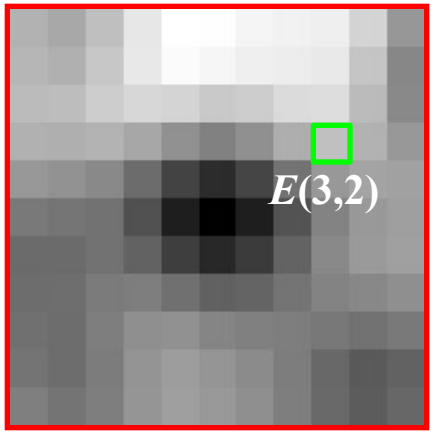
Autocorrelación en ventana *w* con desplazamiento *(u,v)*
Suma de las diferencias entre dos trozos de imagen

$$E(u,v) = \sum_{x,y} w(x,y)[f(x+u, y+v) - f(x,y)]^2$$

$f(x, y)$



$E(u, v)$



Negro, mínimo

Szeliski

Detección

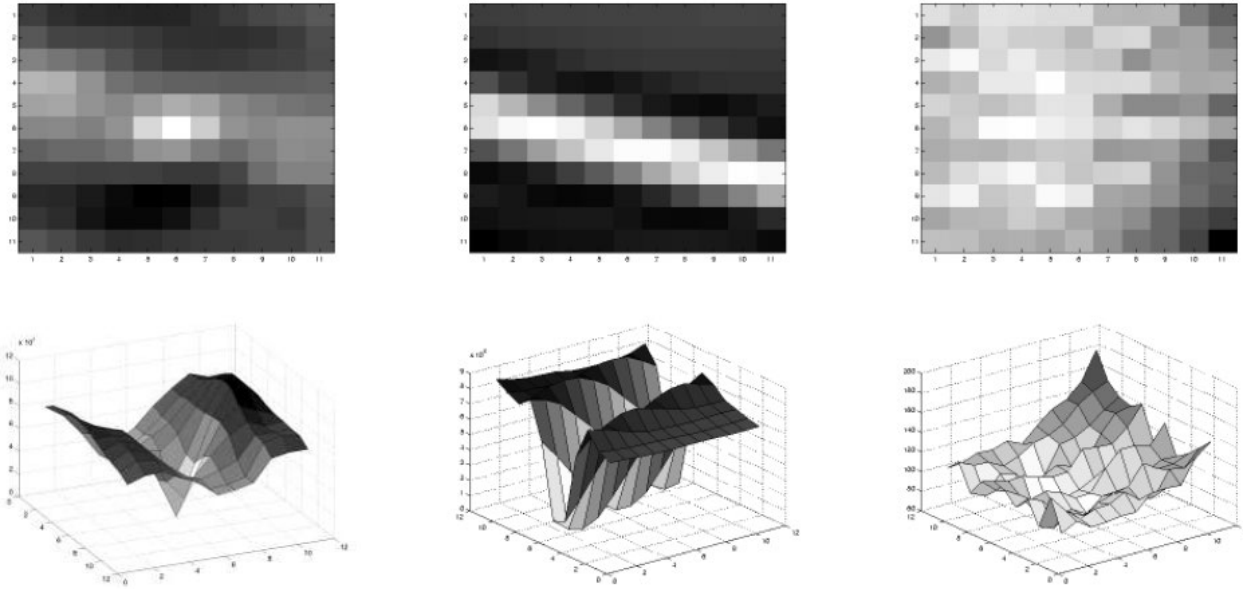
Salida, matriz 2D (superficie)

$$E(u,v) = \sum_{x,y} w(x,y)[f(x+u, y+v) - f(x,y)]^2$$

Szeliski

Puntos

Ojo, blanco es mínimo



(b) (c) (d)

Szeliski

Buenos puntos

Conocida como SSD (*sum of squared differences*)

$$E(u,v) = \sum_{x,y} w(x,y)[f(x+u, y+v) - f(x,y)]^2$$

Coste elevado

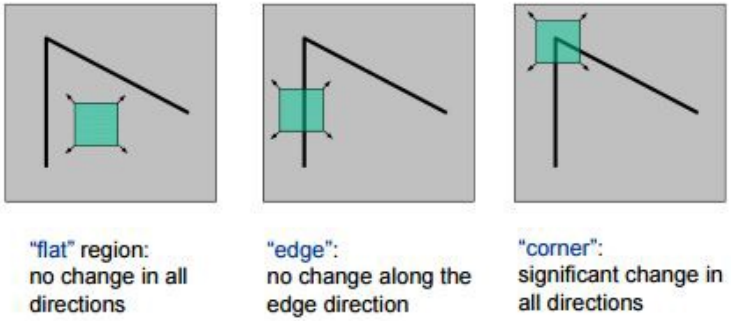
*o(ancho_w^2 * rango_desp^2 * ancho_img^2)*

Buenos puntos

Esquina: gradiente local con dos direcciones dominantes

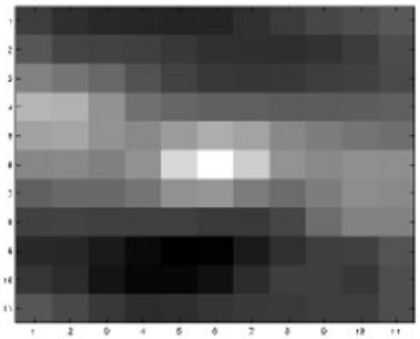
Esquinas repetibles y distintivas

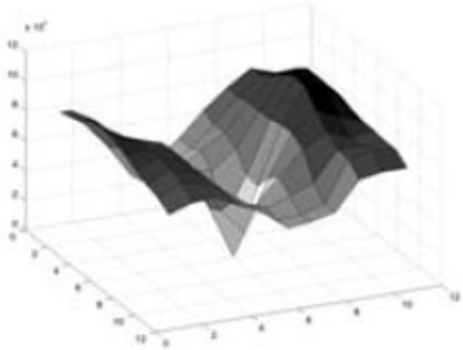
Estabilidad apariencia local



Busca poca estabilidad de *E(u,v)*

Localiza mínimo(s) local(es) marcado(s) en *E(u,v)* (blanco en gráfica)





Szeliski

Harris

Técnica para localizar esquinas (1988)

Explora vecindad del píxel

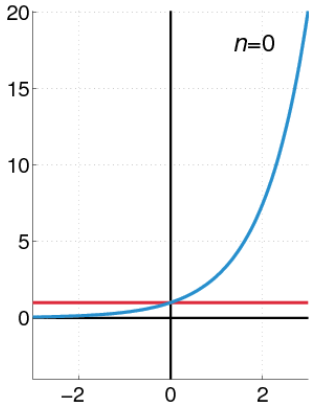
$$E(u,v) = \sum_{x,y} w(x,y)[f(x+u, y+v) - f(x,y)]^2$$

Interesan *E(u,v)* con diferencias (segundo término)

Alto coste, aproxima *E(u,v)*

C.Harris and M.Stephens. "A Combined Corner and Edge Detector" *4th Alvey Vision Conference*:

Harris



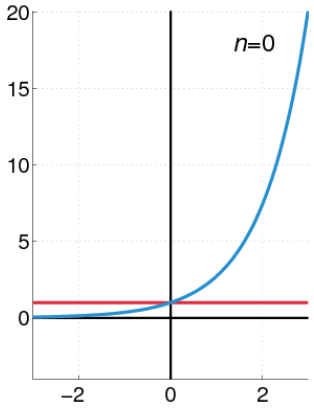
Aproximación local superficie SSD

Expansión con serie de Taylor

Función como serie infinita de sus derivadas en un punto

$$f(a) = \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots$$

Harris



Aproximación local de la superficie SSD

Expansión con serie de Taylor

Función como serie infinita de sus derivadas en un punto

$$f(a) = \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots$$

E(u,v) cerca de (0,0)

$$E(u,v) \approx E(0,0) + [u \quad v]\begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2}[u \quad v]\begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{vu}(0,0) & E_{vv}(0,0) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}$$

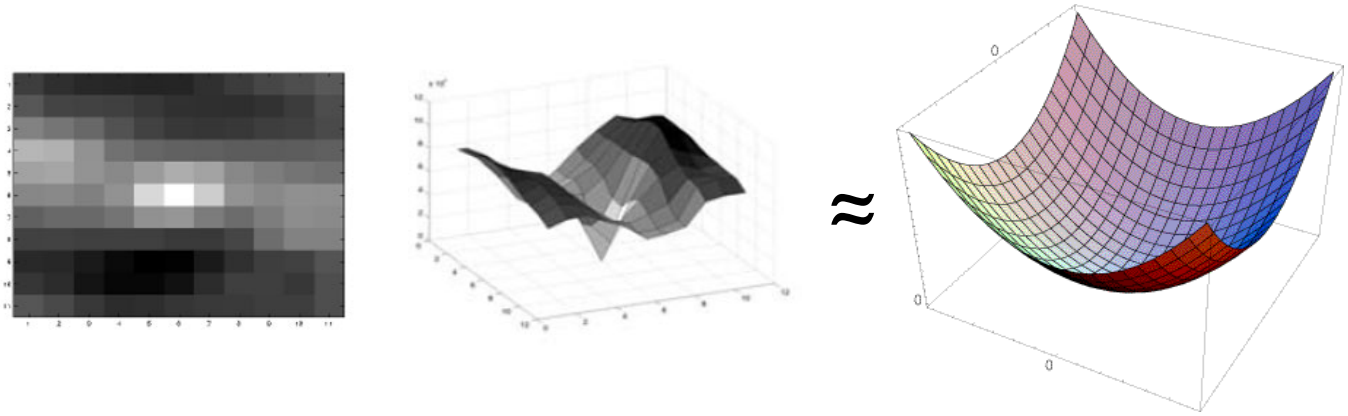
Harris

Expansión con serie de Taylor

Asumiendo $E(0,0)=0$, $E_u(0,0)=0$ y $E_v(0,0)=0$

$$E(u,v) \approx (u \quad v) M \begin{pmatrix} u \\ v \end{pmatrix} \qquad M = \sum_{x,y} w(x,y) \begin{pmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{pmatrix}$$

Segundo orden de la serie, una superficie cuadrática

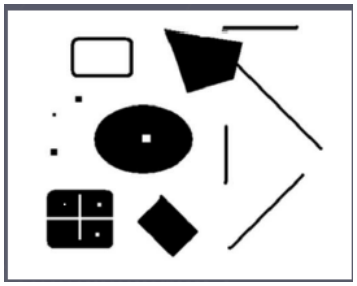


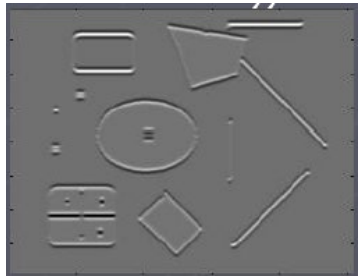
James Hays

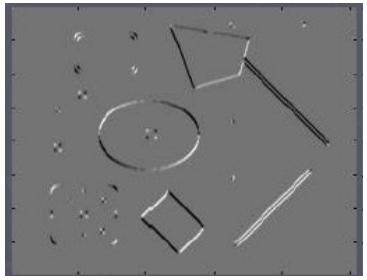
Harris

$$M = \sum_{x,y} w(x,y) \begin{pmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{pmatrix}$$

Gradientes o diferencias de intensidades en imagen







Donde: $I_x \Leftrightarrow \frac{\partial I}{\partial x}$ $I_y \Leftrightarrow \frac{\partial I}{\partial y}$ $I_x I_y \Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}$

James Hays

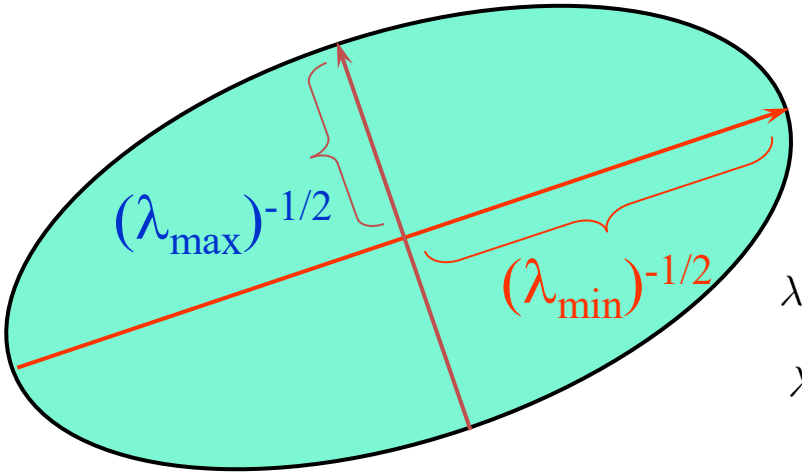
Harris

M es:

- Simétrica, autovectores ortogonales
- Cuadrada, puede ser diagonalizable

$$M = \begin{pmatrix} I_x I_x & I_x I_y \\ I_y I_x & I_y I_y \end{pmatrix} = C^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} C$$

- Los autovalores (λ_1 y λ_2) describen una elipse



$$\lambda_{max} = \frac{1}{a^2} \implies a = (\lambda_{max})^{-1/2}$$

$$\lambda_{min} = \frac{1}{b^2} \implies b = (\lambda_{min})^{-1/2}$$

James Hays

Harris

$$M = \begin{pmatrix} I_x I_x & I_x I_y \\ I_y I_x & I_y I_y \end{pmatrix} = C^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} C$$

En una matriz diagonal:

$$\det(M) = \prod_{i=1}^{n} \lambda_i \qquad \text{traza}(M) = \sum_{i=1}^{n} \lambda_i$$

Obtener *det* y *traza* sin diagonalizar

Criterio de decisión en base al valor de S:

$$S = \det(M) - k\big(\text{traza}(M)\big)^2 \text{ con } k = 0.04\text{-}0.06$$

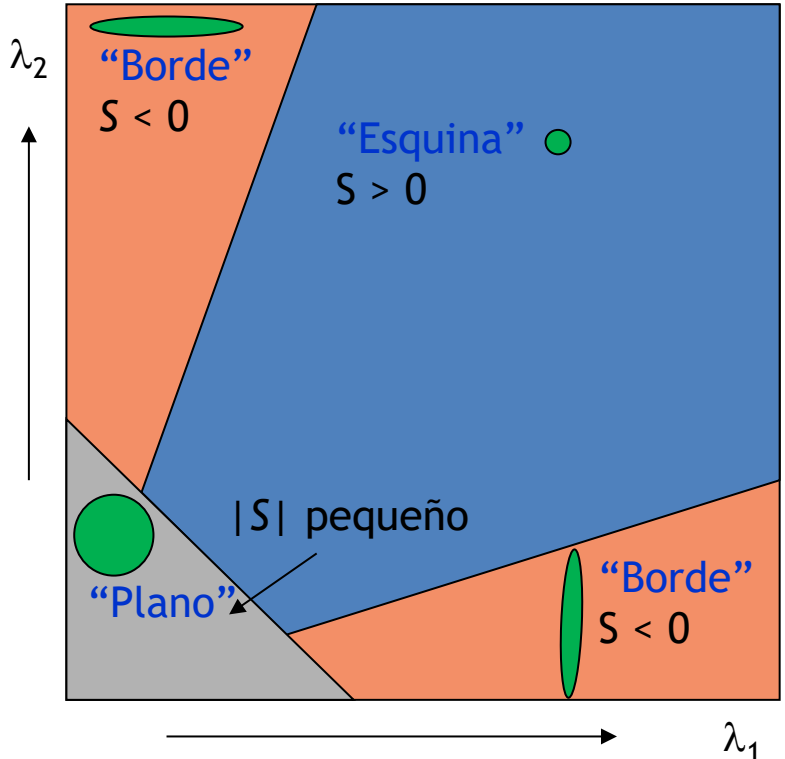
k empírico, 0.04 más sensible, 0.06 menos sensible

Harris

$$S = \det(M) - k\left(traza(M)\right)^2$$

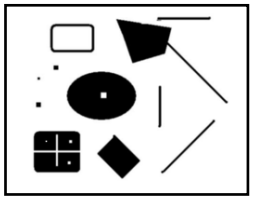
Si:

- |S| es pequeño, λ_1 y λ_2 pequeños: zona plana
- S<0, $\lambda_1 >> \lambda_2$ o viceversa: borde
- S>0, λ_1 y λ_2 grandes y similares $\lambda_1 \sim \lambda_2$: esquina



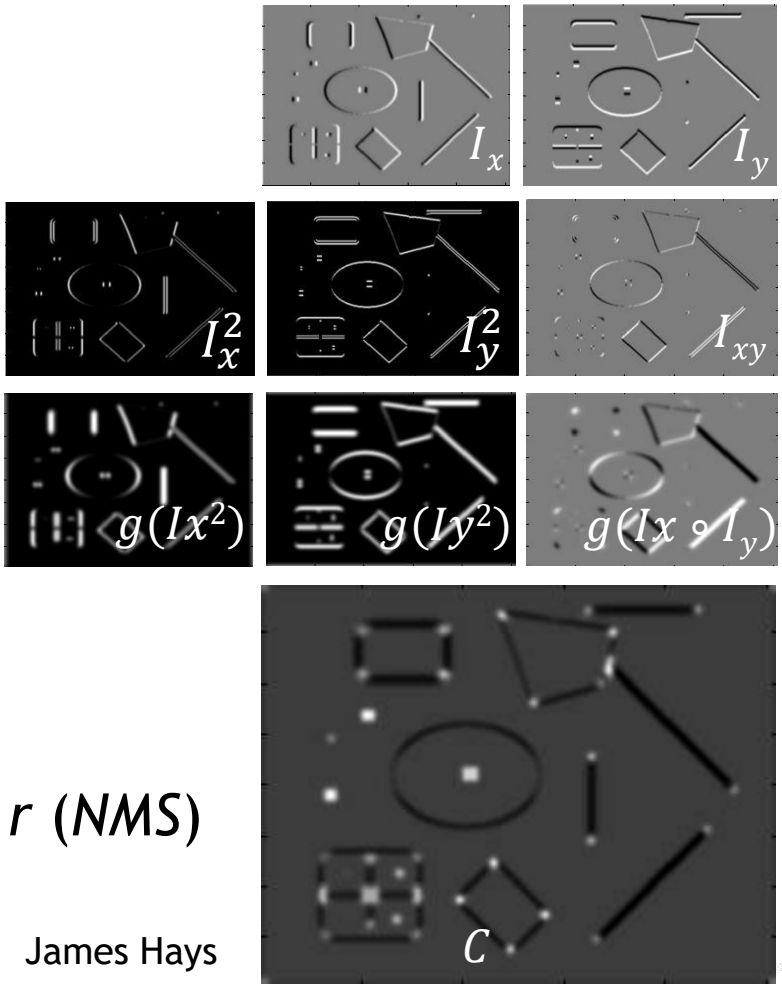
Adaptado de Szeliski

Harris



Cálculo de M en cada píxel:

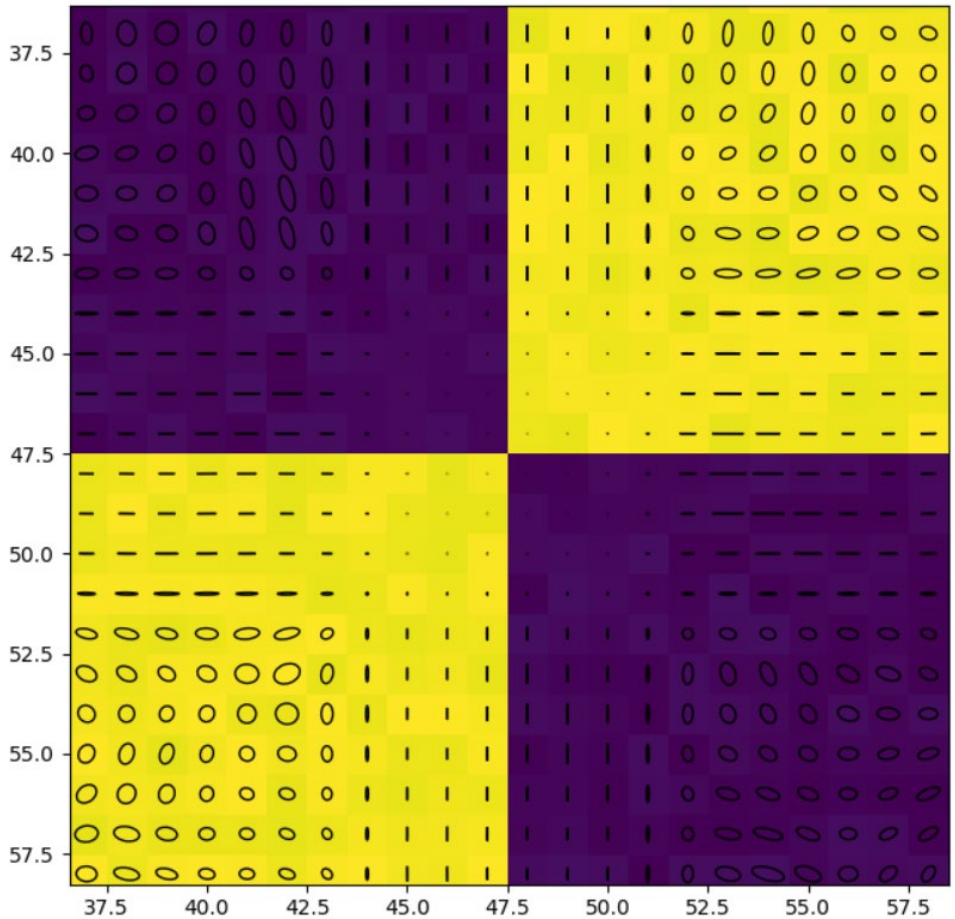
- Gaussiana
- Gradientes I_x e I_y
- Componentes de M
- Calcular S
- Desechar valores pequeños (≈0)
- Eliminar redundancias en cierto r (NMS)



James Hays

Harris

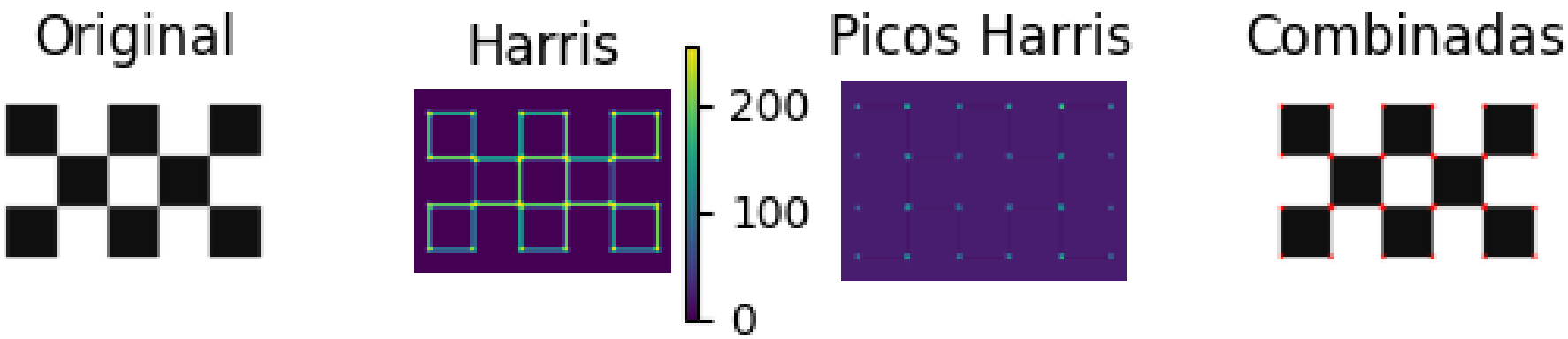
$$S = \det(M) - k\left(traza(M)\right)^2$$



Harris

Resultado

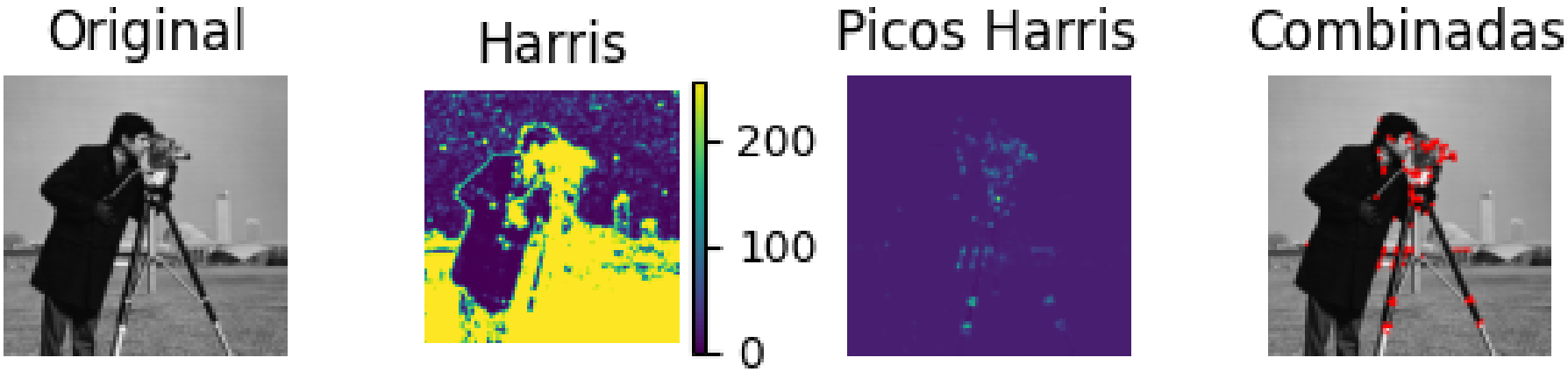
```
gris = np.float32(gris) #Opera con 32 bits
dst = cv2.cornerHarris(gris,2,3,0.04) #Calcula Harris
dst = cv2.dilate(dst, None) #Dilata para resaltar
cth=0.1
rgb[dst>cth*dst.max()]=[255,0,0] #Remarca valores altos
```



Harris

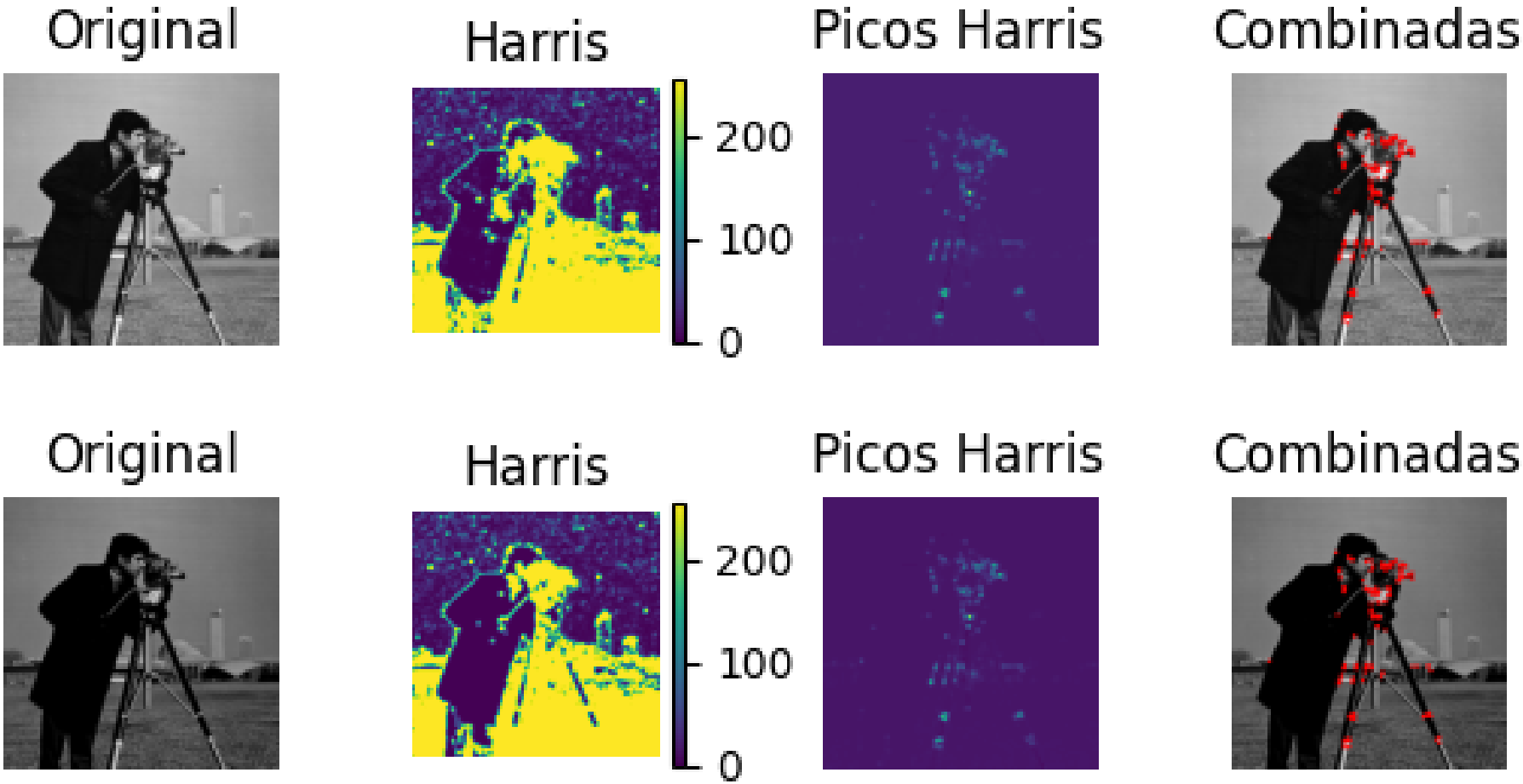
Resultado

```
gris = np.float32(gris) #Opera con 32 bits
dst = cv2.cornerHarris(gris,2,3,0.04) #Calcula Harris
dst = cv2.dilate(dst, None) #Dilata para resaltar
cth=0.1
rgb[dst>cth*dst.max()]=[255,0,0] #Remarca valores altos
```



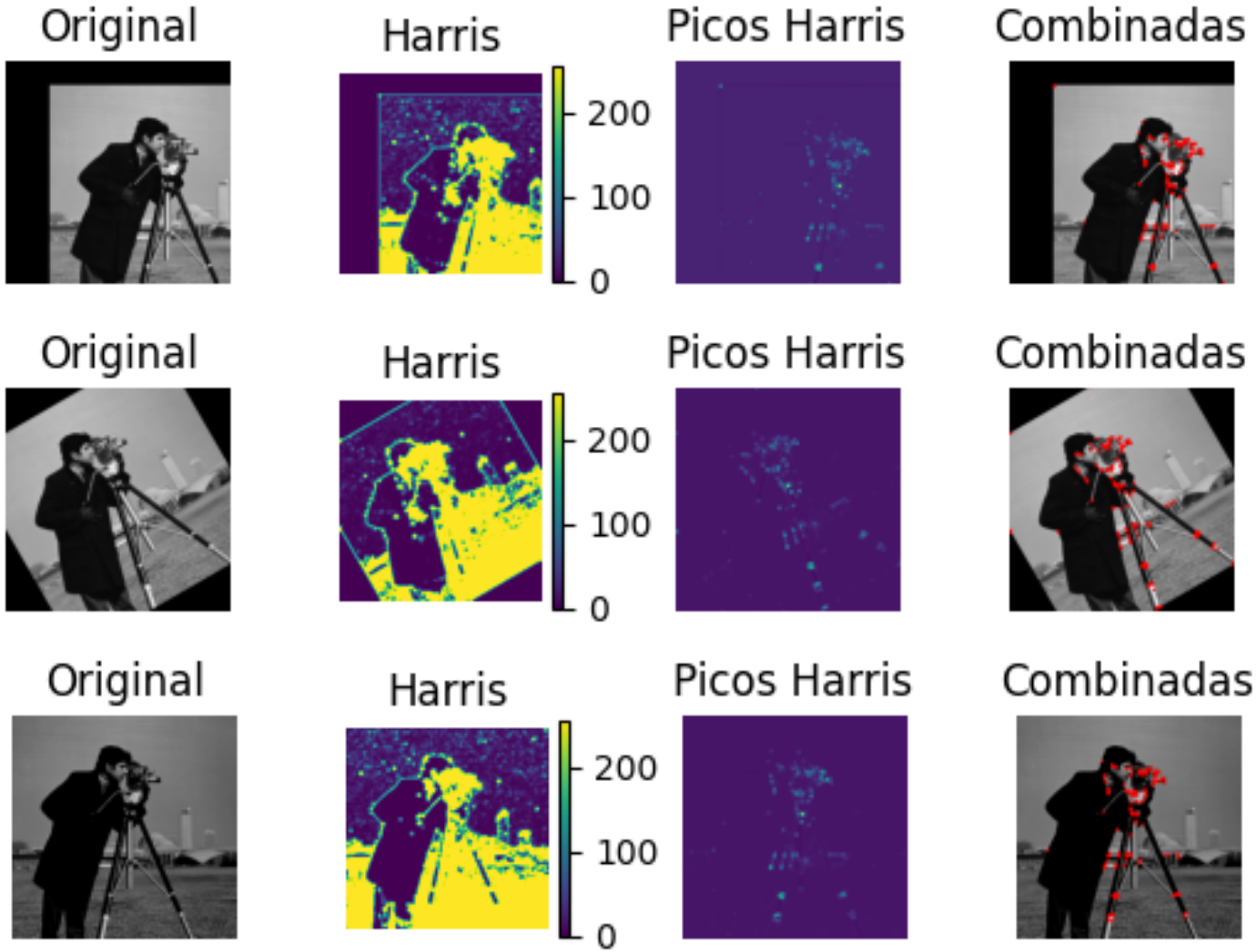
Harris

Invarianza a traslaciones, rotaciones y cambios de brillo



Harris

Invarianza a traslaciones, rotaciones y cambios de brillo



Harris

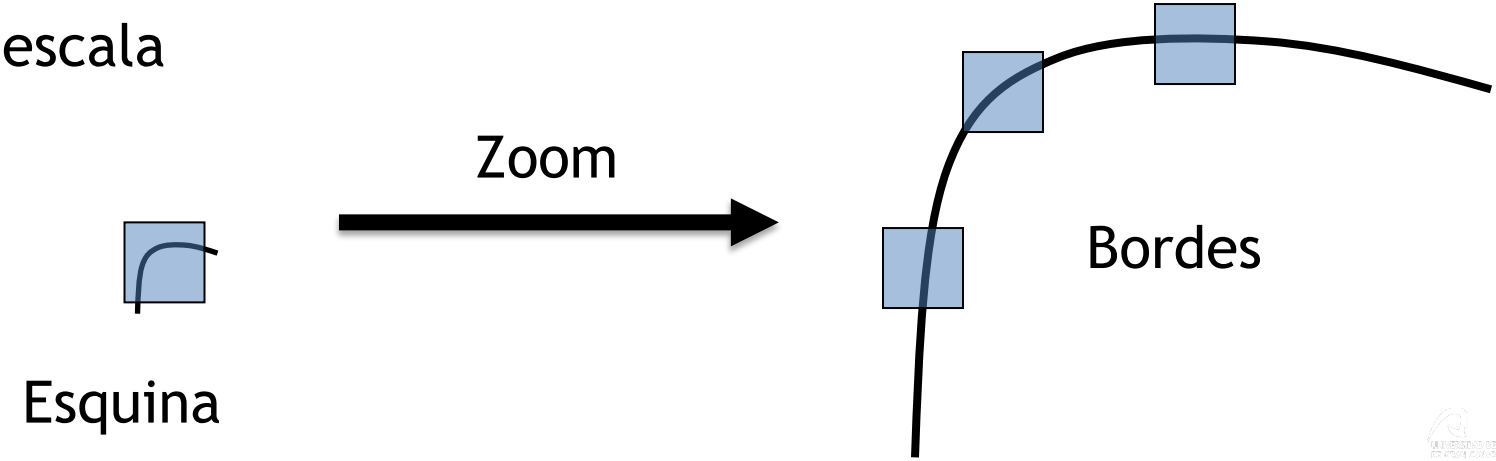
Invariante

Cálculo dependiente de gradientes/derivadas

No afectadas por:

- Traslaciones
- Rotaciones
- Iluminosidad

Sí por escala



Harris

¿Invarianza a escala?

Calcular a múltiples resoluciones (pirámide)

Extraer características estables a escala y posición

LoG se adapta bien a múltiples resoluciones (Harris-Laplace)

Difference of Gaussians (DoG) como aproximación de LoG

Histogramas de gradientes orientados locales (SIFT)

Maximally Stable Extremal Region (MSER)

Comparativa

Table 7.1 Overview of feature detectors.

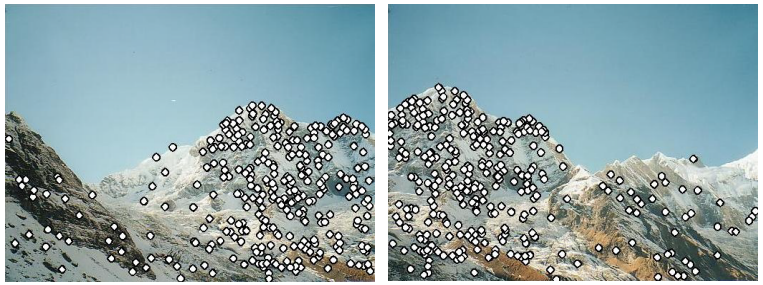
Feature Detector	Corner	Blob	Region	Rotation invariant	Scale invariant	Affine invariant	Localization Repeatability	Localization accuracy	Robustness	Efficiency
Harris	√			√			+++	+++	+++	++
Hessian		√		√			++	++	++	+
SUSAN	√			√			++	++	++	+++
Harris-Laplace	√	(√)		√	√		+++	+++	++	+
Hessian-Laplace	(√)	√		√	√		+++	+++	+++	+
DoG	(√)	√		√	√		++	++	++	++
SURF	(√)	√		√	√		++	++	++	+++
Harris-Affine	√	(√)		√	√	√	+++	+++	++	++
Hessian-Affine	(√)	√		√	√	√	+++	+++	+++	++
Salient Regions	(√)	√		√	√	(√)	+	+	++	+
Edge-based	√			√	√	√	+++	+++	+	+
MSER			√	√	√	√	+++	+++	++	+++
Intensity-based			√	√	√	√	++	++	++	++
Superpixels			√	√	(√)	(√)	+	+	+	+

Tuytelaars y Mikolajczyk 2008

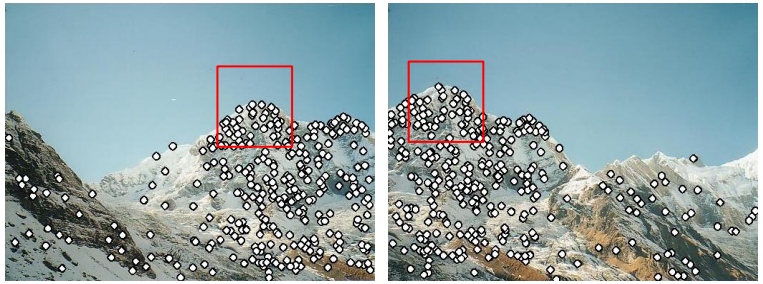
Proceso

Etapas:

- Detección
  Localizar puntos de interés

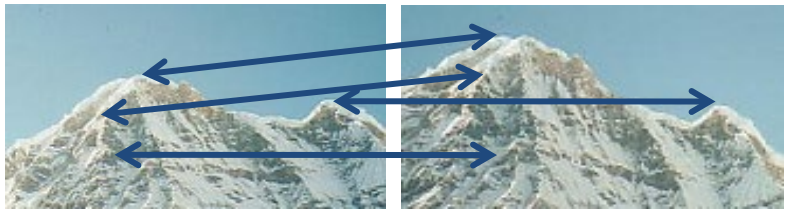


- **Descripción**
  Caracterizar puntos de interés



$$x_1 = [x_1^1, x_2^1, \dots, x_d^1] \qquad x_2 = [x_1^2, x_2^2, \dots, x_d^2]$$

- Encaje
  Determinar la correspondencia



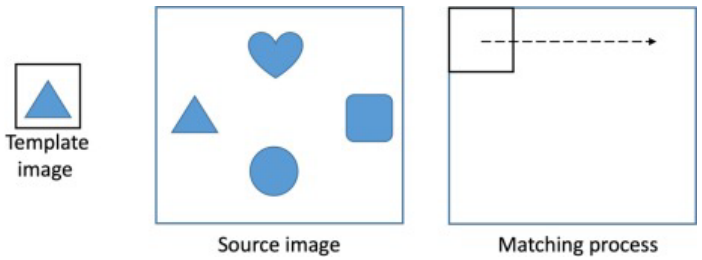
Szeliski

Descripción

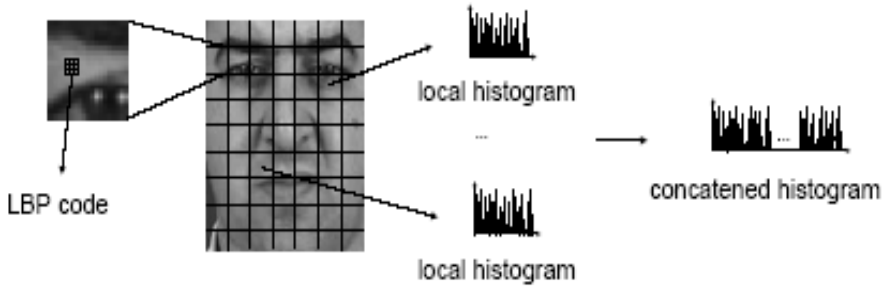
Tras localizar puntos singulares
Describir para encajarlos

Posibilidades de descripción:

- Plantilla: Tolera traslaciones
  SSD o similar



- Histograma: Tolera también cambios de escala y orientación
  Color, textura, SIFT, etc.
  Distancia de histogramas

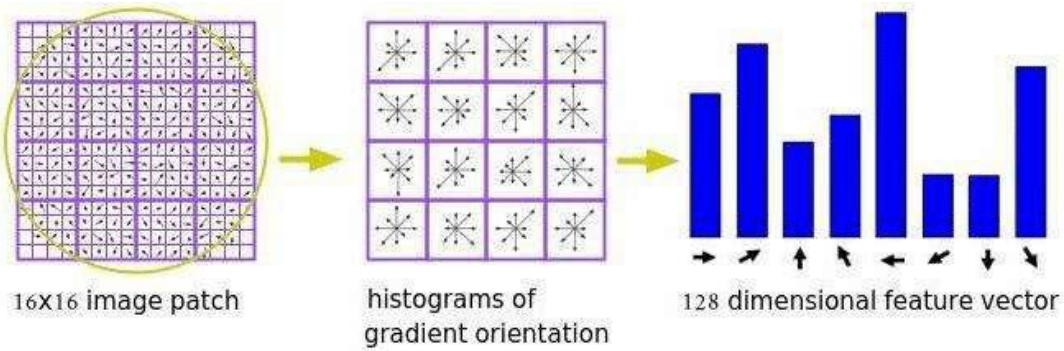


Descripción

Descriptores (alternativas según aplicación):

- *Binary Robust Independent Elementary Features* (BRIEF)
- *A Fast Local Descriptor for Dense Matching* (DAISY)
- *Gradient location-orientation histogram* (GLOH)
- *Scale Invariant feature transform* (SIFT patente expiró en 2020)
- PCA-SIFT
- RootSIFT
- *Bias and gain normalization* (MOPS)
- Oriented FAST and Rotated BRIEF (ORB)
- *Speeded-Up Robust Features* (SURF)
- …

Ilustración con SIFT:



SIFT

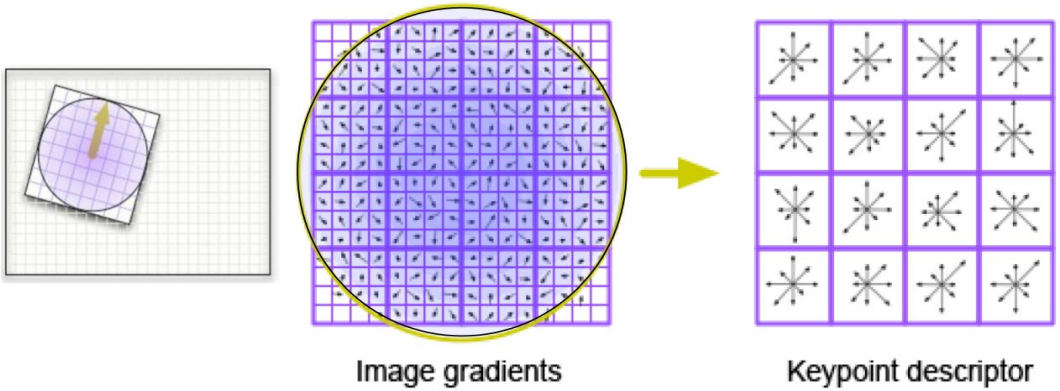
Invariante a cambios de escala (D. Lowe, 2004)

Textura como distribución de diferencias de la imagen

Orientación gradiente dominante en la vecindad

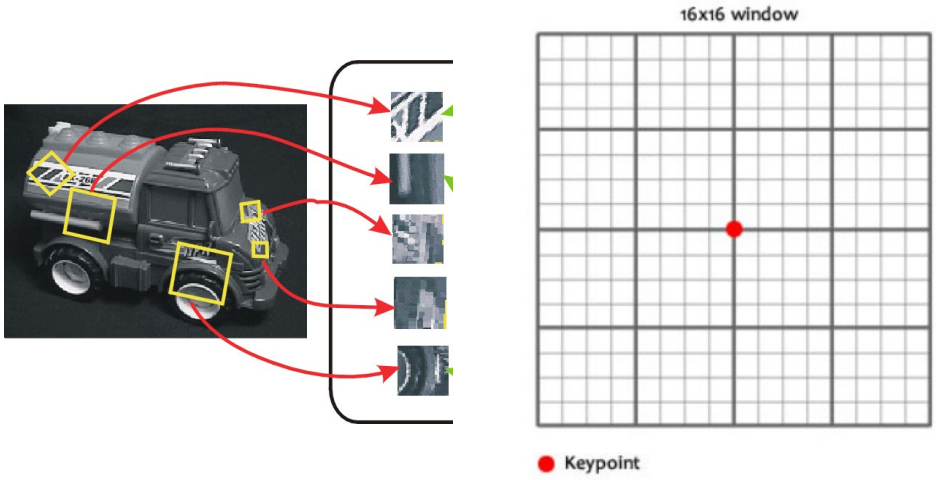
Cálculo del punto de interés en pirámide

Histograma local de gradientes orientados



SIFT

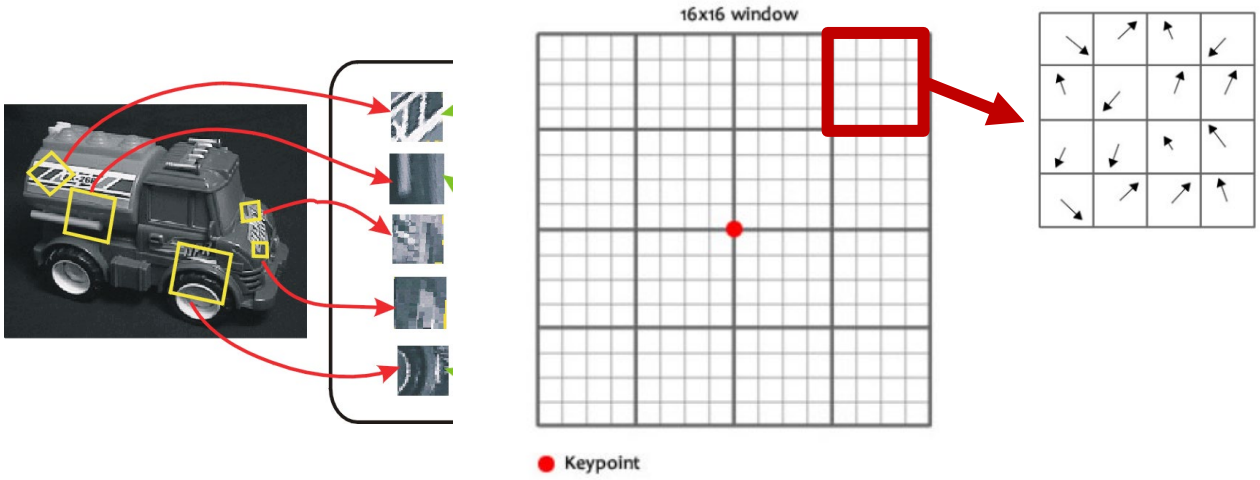
Ventana 16x16 (amarilla) en puntos de interés (rojos)



Adaptado de James Hays y Utkarsh Sinha

SIFT

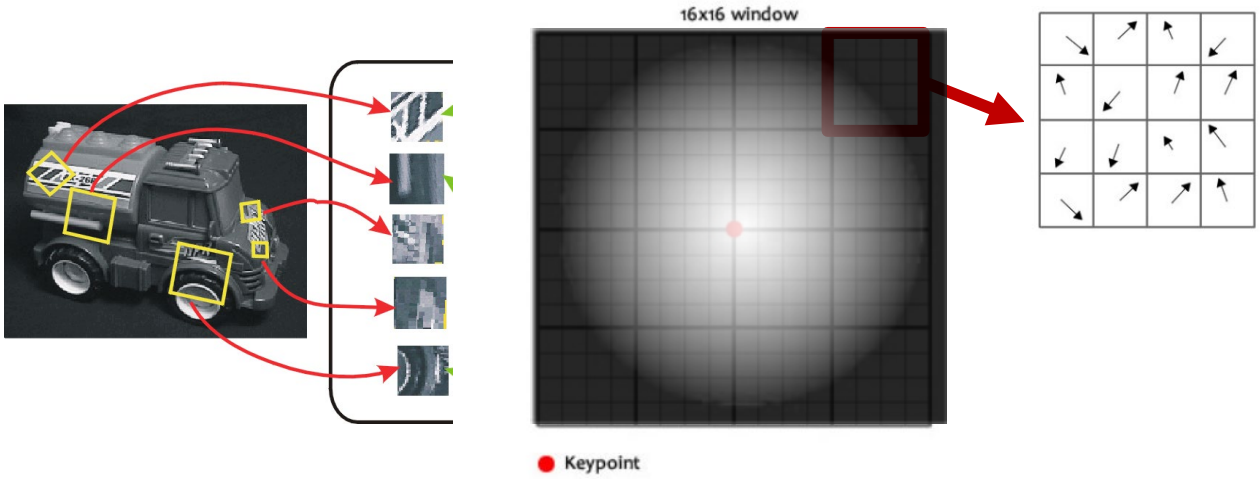
Ventana 16x16 (amarilla) en puntos de interés (rojos)
Magnitud y orientación del gradiente por píxel



Adaptado de James Hays y Utkarsh Sinha

SIFT

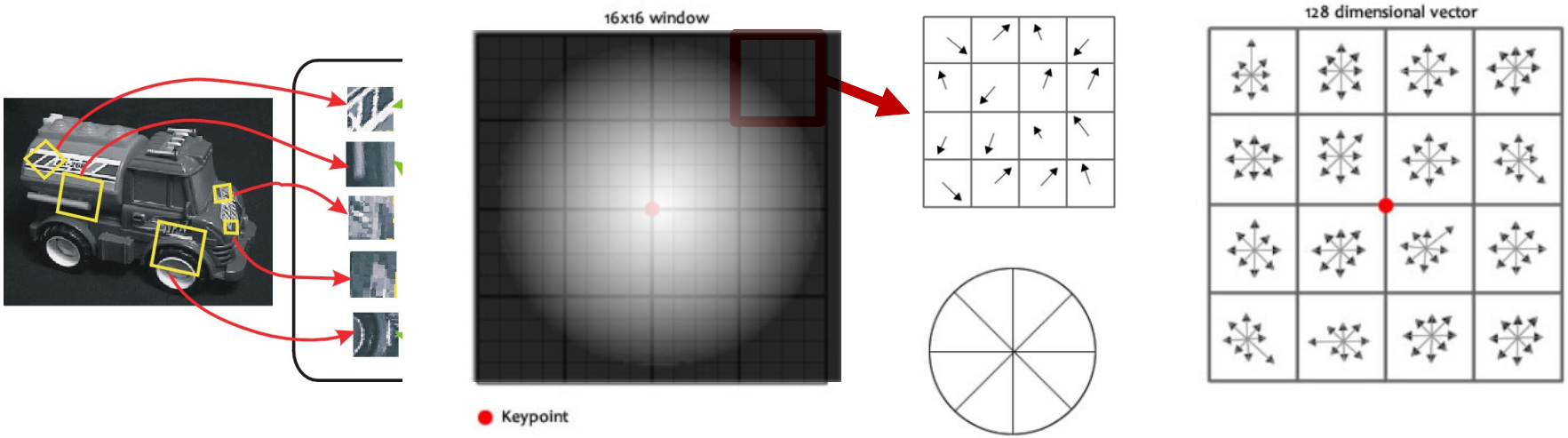
Ventana 16x16 (amarilla) en puntos de interés (rojos)
Magnitud y orientación del gradiente por píxel
Ponderado con Gaussiana



Adaptado de James Hays y Utkarsh Sinha

SIFT

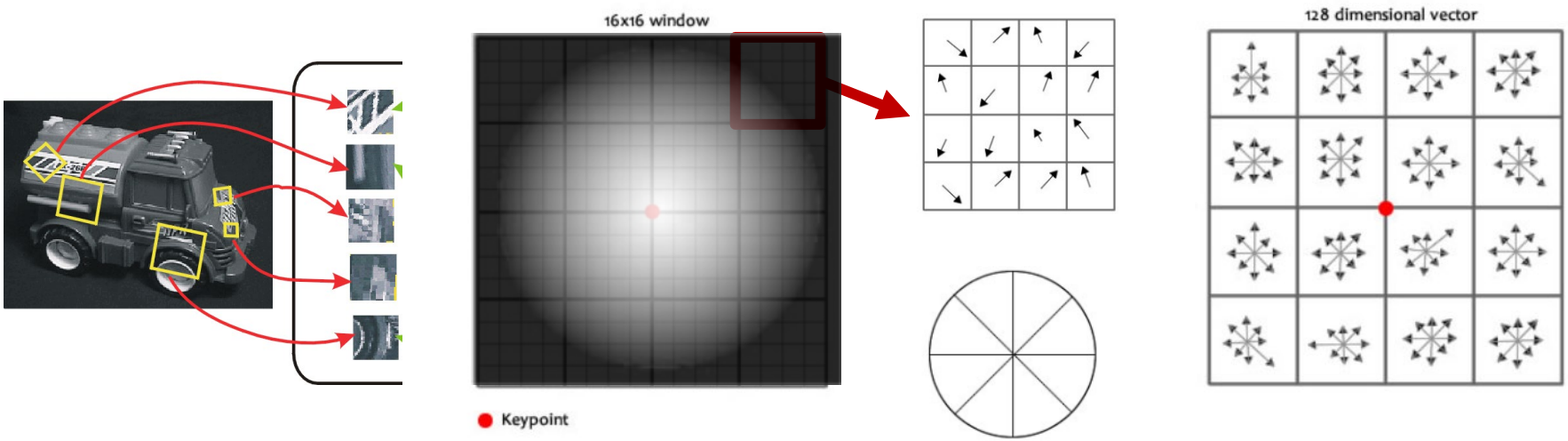
Ventana 16x16 (amarilla) en puntos de interés (rojos)
Magnitud y orientación del gradiente por píxel
Ponderado con Gaussiana
Discretiza a 8 orientaciones (*bins*)
4x4 histogramas -> 16*8=128 valores



Adaptado de James Hays y Utkarsh Sinha

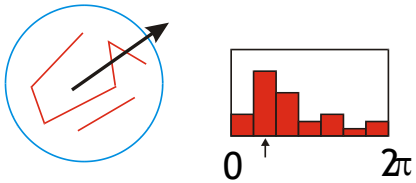
SIFT

128 valores positivos
Normaliza para reducir efectos de contraste, ganancia
Satura gradientes > 0.2 (destellos, no lineal), y normaliza



Adaptado de James Hays y Utkarsh Sinha

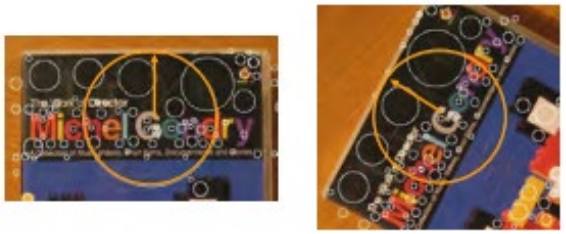
SIFT



[Lowe, SIFT, 1999]

Parte de puntos de interés (p.e. Harris o DoG):

- Gradientes, orientación dominante
- Normaliza orientación

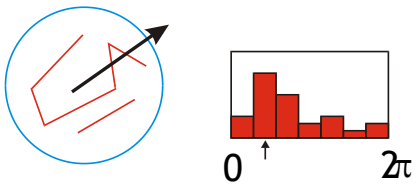


S. Nayar

```
sift = cv2.SIFT_create()
kp = sift.detect(gris,None)
imgkp=cv2.drawKeypoints(gris,kp,img)
```

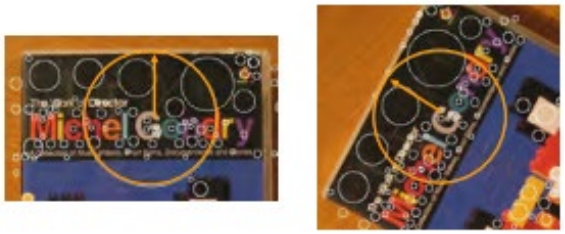
SIFT

Parte de puntos de interés (p.e. Harris o DoG):



[Lowe, SIFT, 1999]

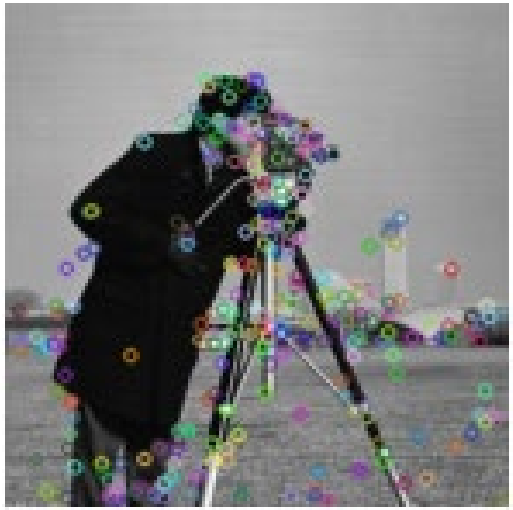
- Gradientes, orientación dominante
- Normaliza orientación



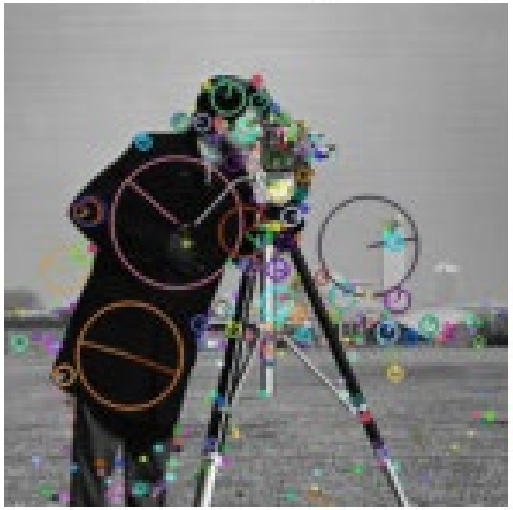
S. Nayar

Original

SIFT



SIFT (tam. y ori.)



SIFT

Gradiente, invarianza intensidad

Orientación dominante en 16x16 para invarianza

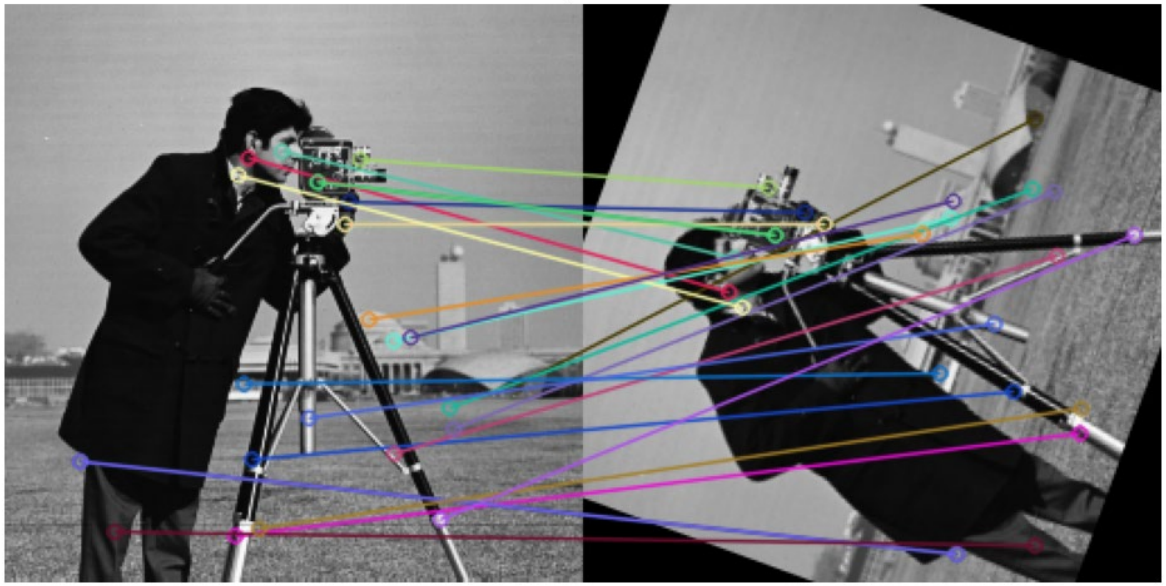
Discretiza orientaciones

Textura depende de vista, histograma para invarianza

Satura gradiente evitando brillos/destellos

Múltiples escalas

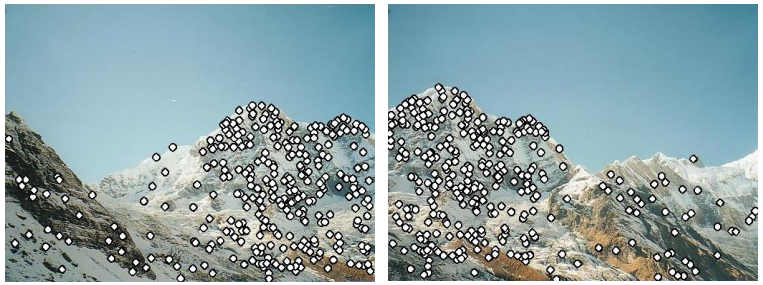
Encajes SIFT



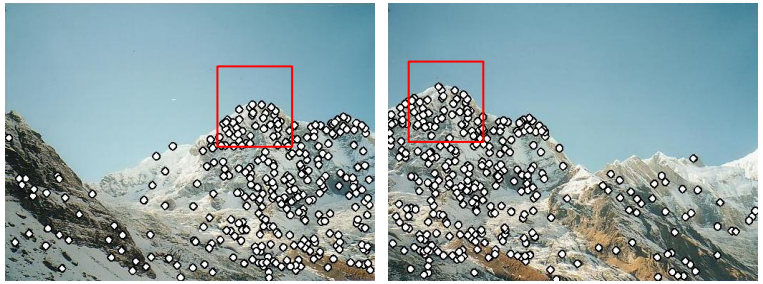
Proceso

Etapas:

- Detección
  Localizar puntos de interés

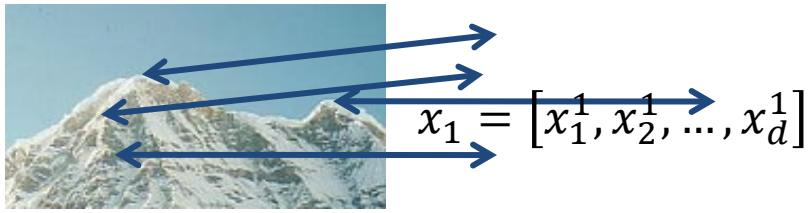


- Descripción
  Caracterizar puntos de interés



$$x_1 = [x_1^1, x_2^1, \dots, x_d^1] \qquad x_2 = [x_1^2, x_2^2, \dots, x_d^2]$$

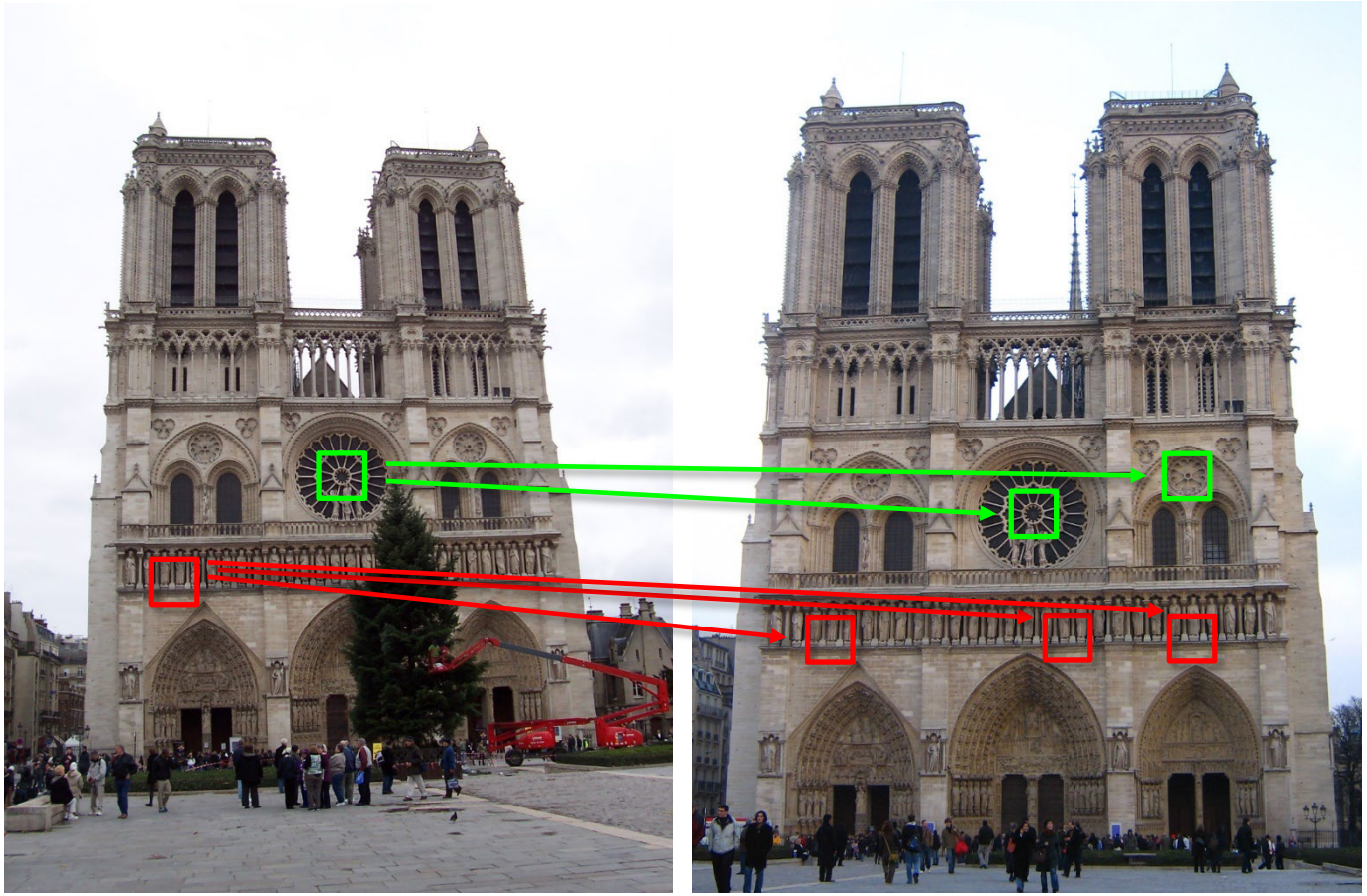
- **Encaje**
  Determinar la correspondencia



Szeliski

Encaje

Decidir si dos puntos se corresponden



Distancia: 0.34, 0.30, 0.40

Distancia: 0.61, 1.22

Szeliski

Encaje

Decidir si dos puntos se corresponden

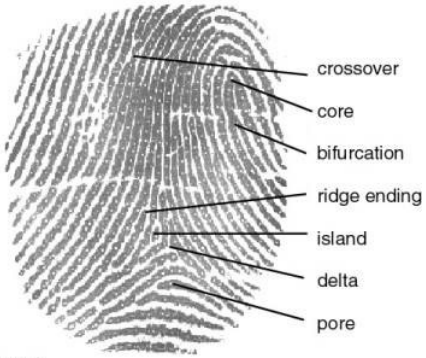
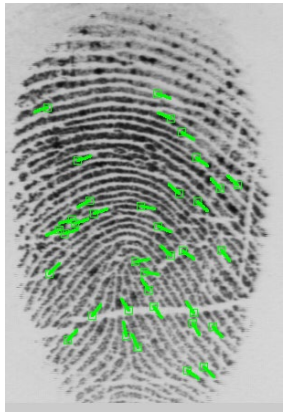


Figure 1

Calidad = 0.96
Falsas minucias=0

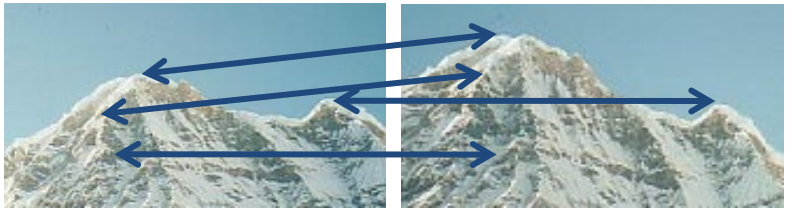


Calidad = 0.53
Falsas minucias=7

Encaje

Dos imágenes, dos conjuntos de puntos con sus descriptores

$$x_1 = \left[x_1^1, x_2^1, \dots, x_d^1\right]$$



Szeliski

Plantillas: SSD o similar

Histogramas: Comparación

Encaje

Descriptores SIFT, 128 valores

Distancias entre los puntos de dos imágenes

Menor distancia

Vecino más cercano (*Nearest Neighbor* NN) es NN_1

$$d_2(a,b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

Encaje

¿Todos con todos? Buf

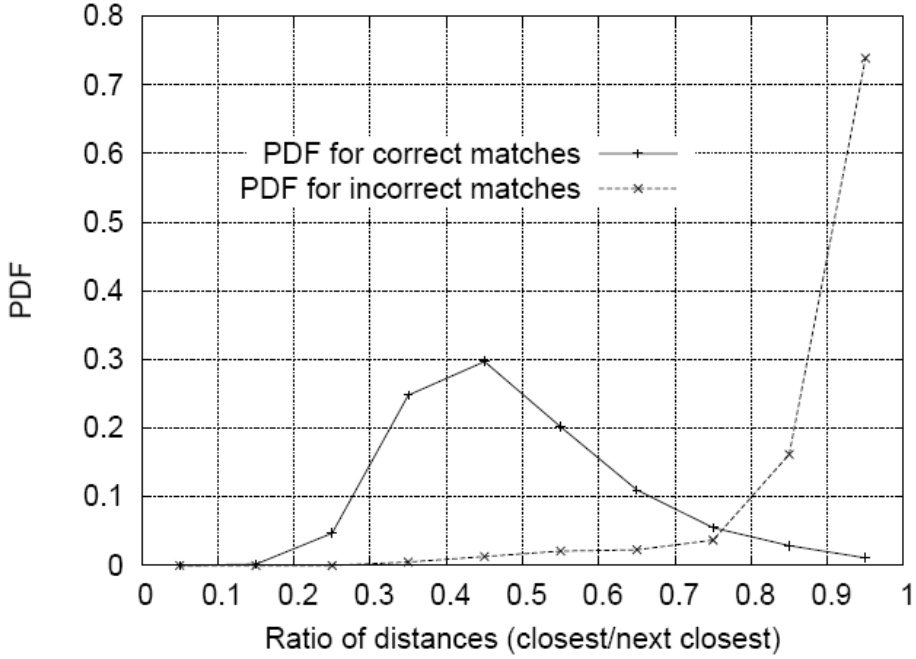
$$d_2(a,b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

Costoso
Bucles mala idea, cálculo matricial
Estructura en árbol o tabla *hash*
Vídeo, limitar búsqueda a zonas próximas

Encaje

¿Y si el segundo más cercano tiene distancia similar?

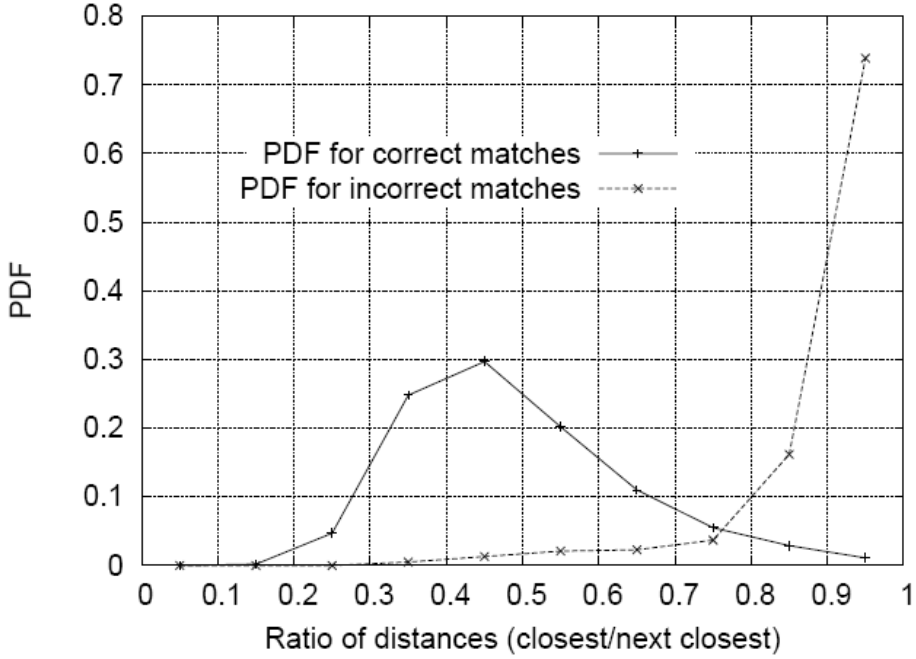
D. Lowe analizó ratio NN_1/NN_2 de 40000 puntos
Valores próximos a 1 *peligrosos*
Valor de confianza



D. Lowe

Encaje

¿Tienen todos encaje?
Umbral de distancia mínima aceptable
Valor de confianza según aplicación, compromiso FP y TP
¿Qué son FP y TP?



D. Lowe

Encaje

Evaluación

- ¿Aciertos?
- ¿Fallos?

Encaje

Evaluación (*scores*):

- Verdadero positivo (TP *true positive*): Encajes correctos
- Falso negativo (FN *false negative*): Encajes no detectados
- Falso positivo (FP *false positive*): Encajes incorrectos
- Verdadero negativo (TN *true negative*): No encajes bien rechazados

Matriz de confusión

		Predicho		
		Encajes predichos	No encajes predichos	Total
Real	Encajes	TP	FN	P
	No encajes	FP	TN	N
	Total	P’	N’	

Encaje

Matriz de confusión

		Predicho		
		Encajes predichos	No encajes predichos	Total
Real	Encajes	TP	FN	P
	No encajes	FP	TN	N
	Total	P’	N’	

$$precisión\ o\ PPV = \frac{TP}{TP + FP} = \frac{TP}{P'}$$

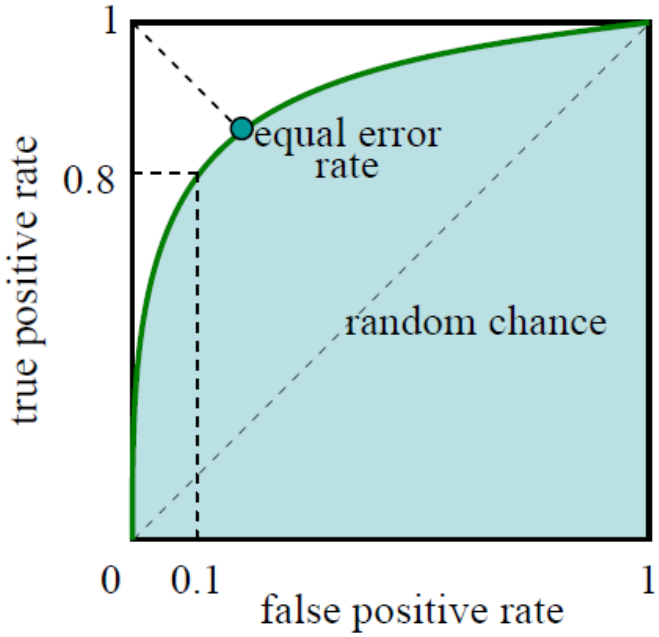
$$FPR = \frac{FP}{FP + TN} = \frac{FP}{N}$$

$$recall/sensitivity\ o\ TPR = \frac{TP}{TP + FN} = \frac{TP}{P}$$

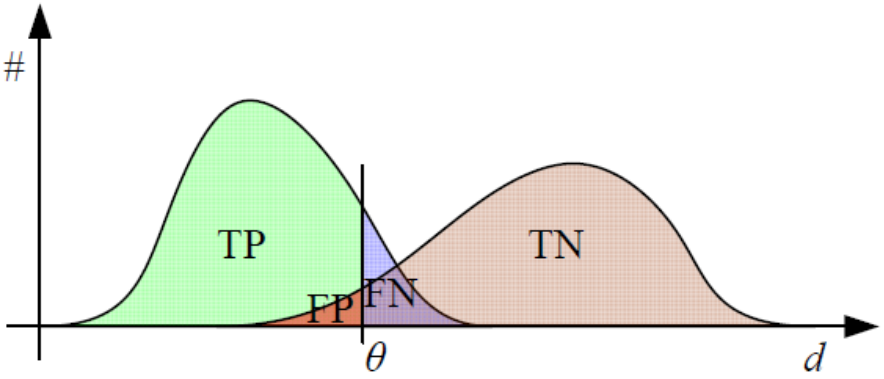
$$ACC = \frac{TP + FN}{P + N}$$

Encaje

Dada una distancia decidir si hay encaje o no
Umbral de decisión (θ)
Elección no exenta de problemas
Receiver operating characteristic (ROC) enfrenta TPR a FPR
Comparativa de modelos con curva ROC



(a)



(b)

Szeliski

Encaje

Distancia euclídea

No la única

Elección dependiente del problema

$$d_2(a,b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

Encaje

Distancia euclídea
No la única
Elección dependiente del problema

$$d_2(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

Distancias entre vectores de características

Dos muestras a y b con n componentes

- Distancia de **Minkowski:**

$$d(a,b) = \left(\sum_{i=1}^{n}|a_i - b_i|^r\right)^{\frac{1}{2}}, r \geq 1$$

- Distancia de **Mahalanobis:**
  - Interviene la correlación de las variables
  - No afecta la escala de la medida
  - Σ matriz de covarianza

$$d_M(a,b) = \sqrt{(a-b)^T \Sigma^{-1}(a-b)}$$

Encaje

Concreciones de Minkowski:

- Manhattan o city block (r=1)
- Euclídea o L2 (r=2)
- Chebyshev o tablero de ajedrez (r -> ∞)

$$d_1(a,b) = \sum_{i=1}^{n} |a_i - b_i|$$

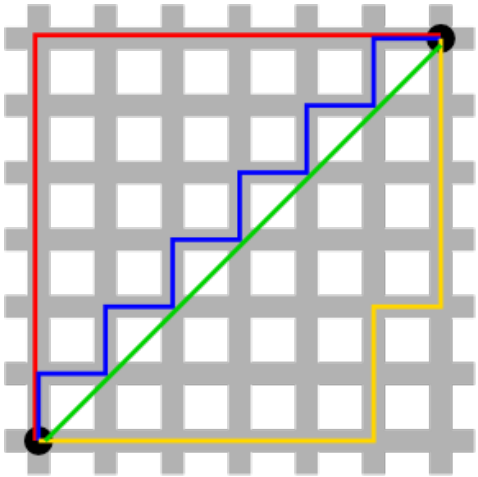
$$d_2(a,b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

$$d_\infty(a,b) = máx_{i=1..n} |a_i - b_i|$$

Manhattan = 12

Euclídea ≈ 8.5

Chebyshev = 6



d = 0 para el mejor encaje

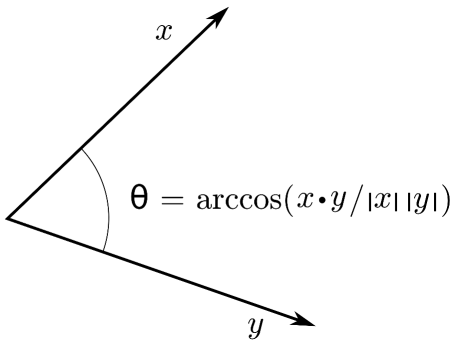
sklearn.neighbors.DistanceMetric

Encaje

Similitud coseno

Coseno del ángulo formado entre dos vectores

$$\cos\theta = \frac{\vec{x}\cdot\vec{y}}{|\vec{x}||\vec{y}|}$$



Rango [-1, 1]

Resulta 1 cuando son idénticos

No es métrica/distancia, no cumple la desigualdad triangular

Encaje

Comparación de histogramas de *n bins*, en OpenCV:

- Correlación (1 idénticos)

$$d(h_1, h_2) = \frac{\sum_i \left(h_1(i) - \overline{h}_1\right)\left(h_2(i) - \overline{h}_2\right)}{\sqrt{\sum_i \left(h_1(i) - \overline{h}_1\right)^2 \sum_i \left(h_2(i) - \overline{h}_2\right)^2}} \quad \text{donde} \quad \overline{h}_k = \frac{1}{n}\sum_j h_k(j)$$

- Chi-square (0 idénticos)

$$d(h_1, h_2) = \sum_i \frac{\left(h_1(i) - h_2(i)\right)^2}{h_1(i)}$$

- Intersección (mayores más similares)

$$d(h_1, h_2) = \sum_i min(h_1, h_2)$$

- Bhattacharyya (0 idénticos)

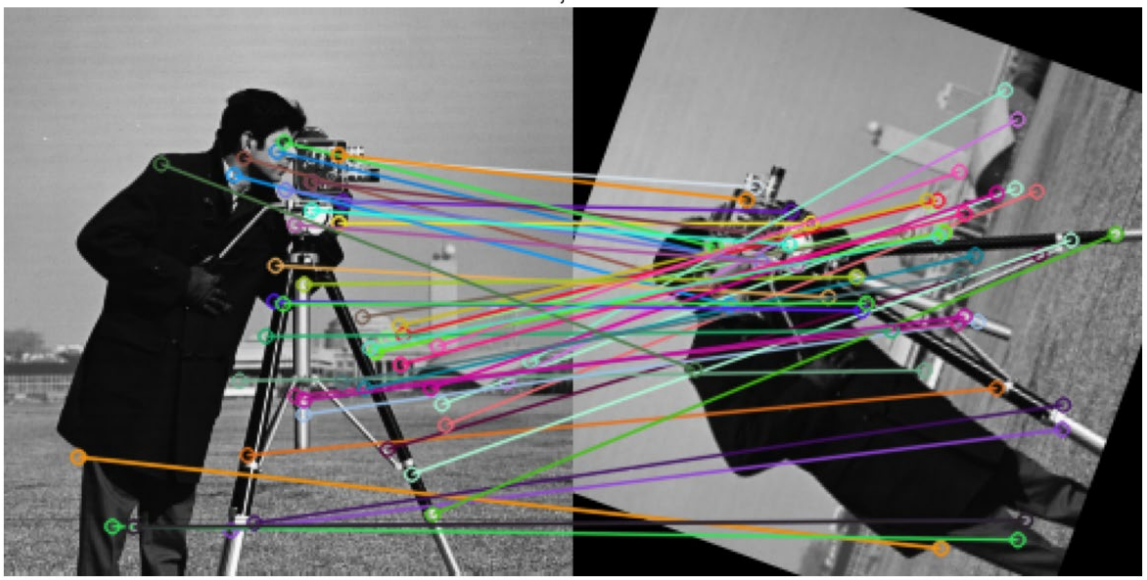
$$d(h_1, h_2) = \sqrt{1 - \frac{1}{\sqrt{\overline{h}_2 \overline{h}_2 n^2}} \sum_i \sqrt{h_1(i) \cdot h_2(i)}}$$

Encaje

Encaje puntos SIFT en OpenCV

```
sift = cv2.SIFT_create()
kp1,de1 = sift.detectAndCompute(gris1,None)
kp2,de2 = sift.detectAndCompute(gris2,None)
bf = cv2.BFMatcher(cv2.NORM_L2, crossCheck=True)
matches = bf.match(de1, de2)
```

Encajes SIFT



Encaje

Encaje puntos SIFT en OpenCV

```
sift = cv2.SIFT_create()
kp1,de1 = sift.detectAndCompute(gris1,None)
kp2,de2 = sift.detectAndCompute(gris2,None)
bf = cv2.BFMatcher(cv2.NORM_L2, crossCheck=True)
matches = bf.match(de1, de2)
img_matches = cv2.drawMatches(img1, kp1, img2, kp2, matches[:50], None,
matchColor=(0, 0, 0),flags=cv2.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS)
```

Encajes SIFT

Encaje

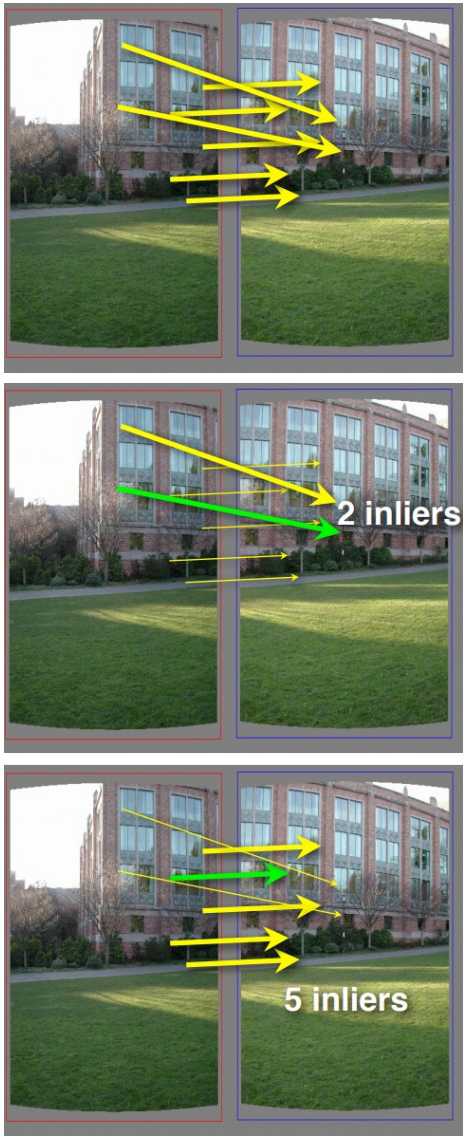
Encajes no coherentes, *outliers*

Método RANSAC (*random sample consensus*):

- Escoger par(es) aleatoriamente
- Contar *inliers*
- Repetir el proceso

Repeticiones según número muestras

Escoge la de mayor número de *inliers*



I. Gkioulekas

Encaje

Eliminar encajes erróneos (*outliers*) con [cv.findHomography]

```
# Extrae los puntos de encaje
pts1 = np.float32([kp1[m.queryIdx].pt for m in matches]).reshape(-1, 1, 2)
pts2 = np.float32([kp2[m.trainIdx].pt for m in matches]).reshape(-1, 1, 2)
# RANSAC para filtrar outliers
H, mask = cv2.findHomography(pts1, pts2, cv2.RANSAC, 5.0)
matches_f = mask.ravel().tolist()
# Filtra correspondencias con RANSAC
matchesf = [matches[i] for i in range(len(matches)) if matches_mask[i]]
```

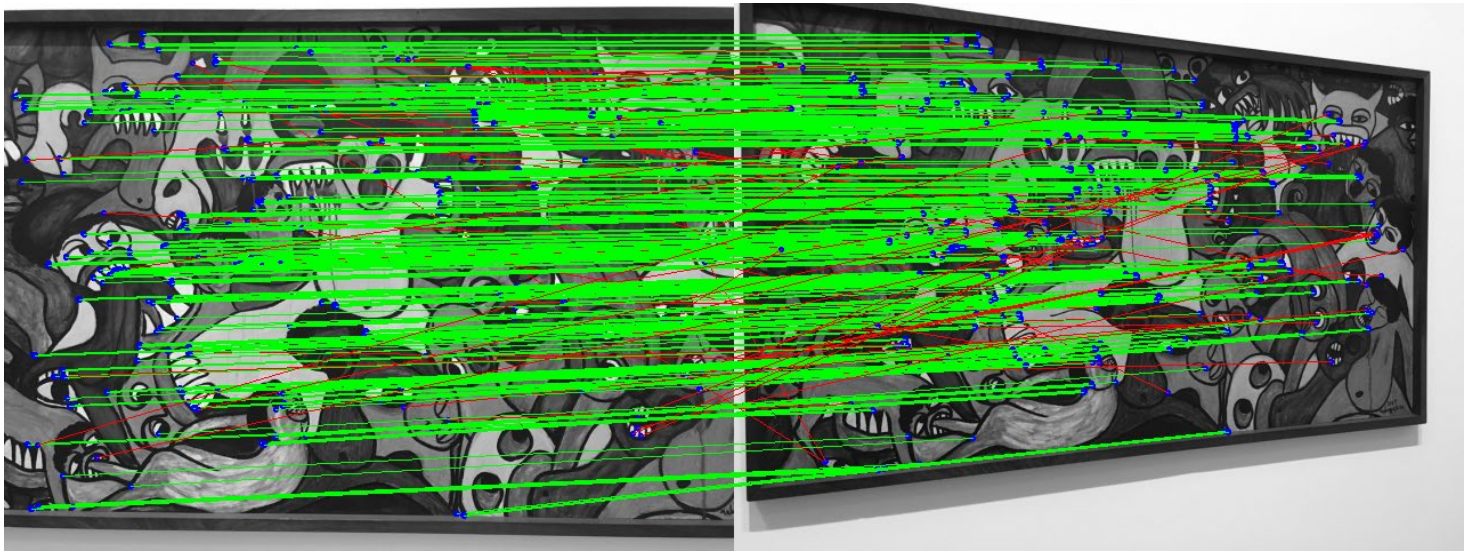
Encajes SIFT

¿Y en tiempos de DL?

SuperPoint (2018)
Detección y descripción similar a SIFT
Basado en VGG
70 fps

¿Y en tiempos de DL?

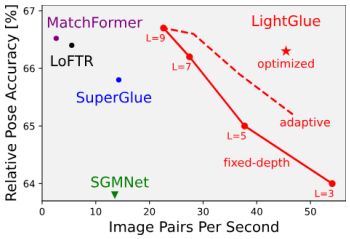
SiLK - Simple Learned Keypoints (2023)
Dos GPUs para entreno
30 fps

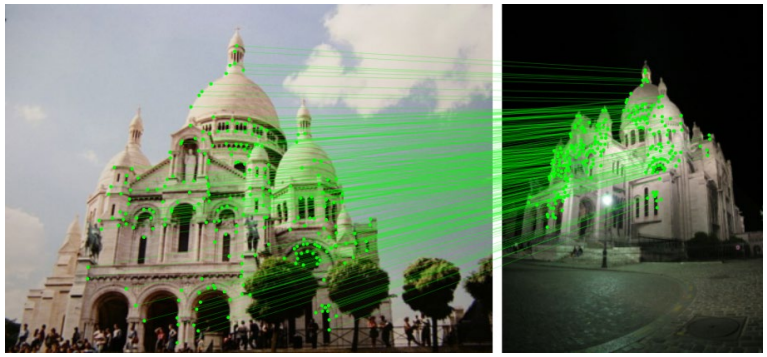


¿Y en tiempos de DL?

Encaje

LightGlue (2023)
Puntos





GlueStick (2023)
Puntos y líneas