



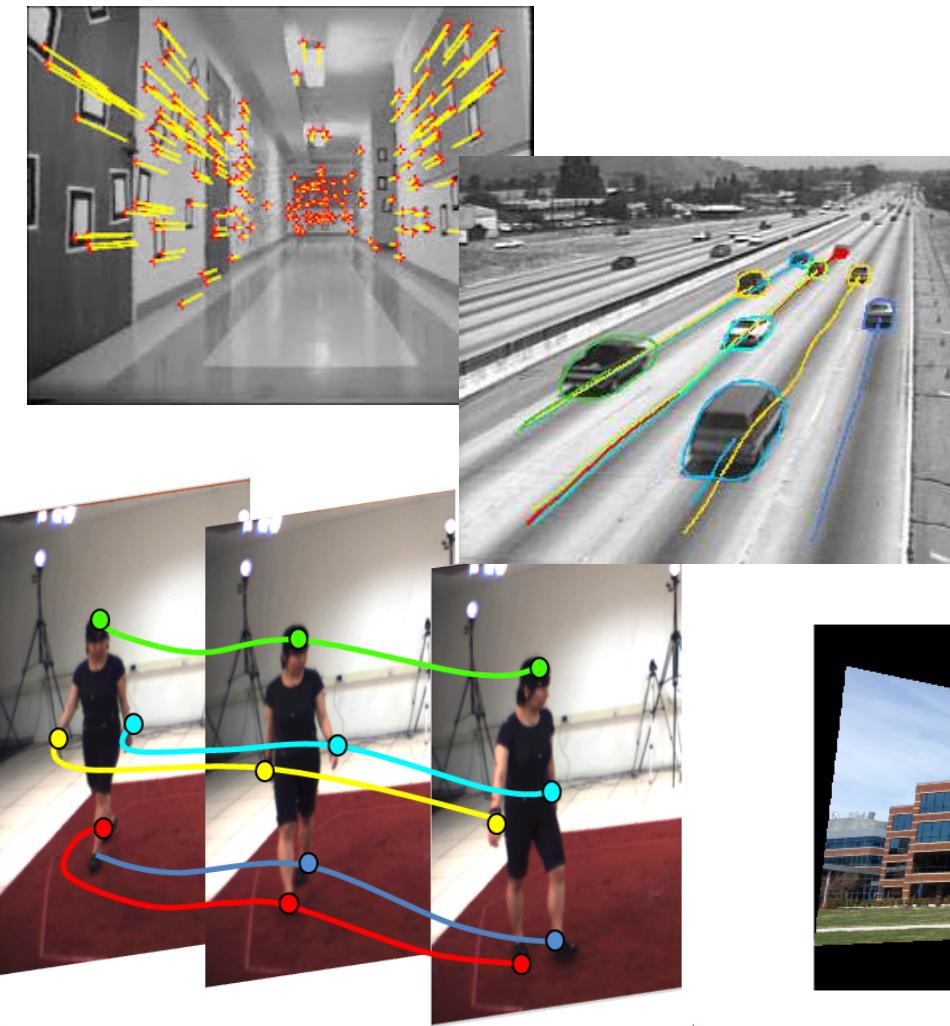
Introducción a visión por Computadoras con GPUs

Detección de características (*features*).

Jorge A. Sánchez
FaMAF, UNC, 2015

Problema: establecer correspondencias

Tracking



Stitching

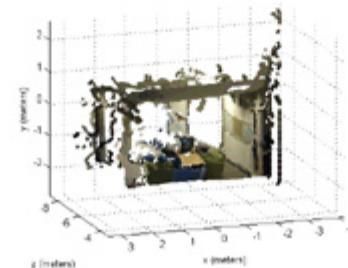
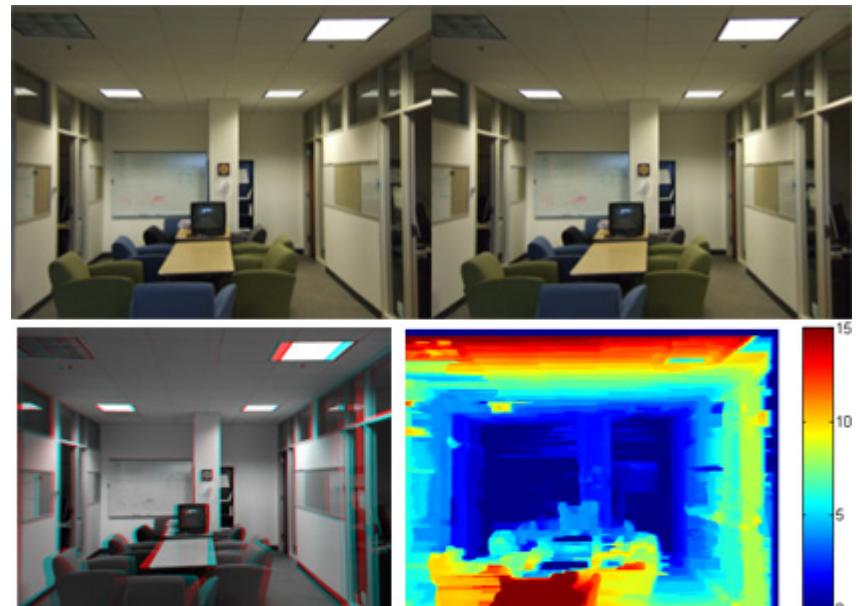


Problema: establecer correspondencias

Reconocimiento / búsqueda

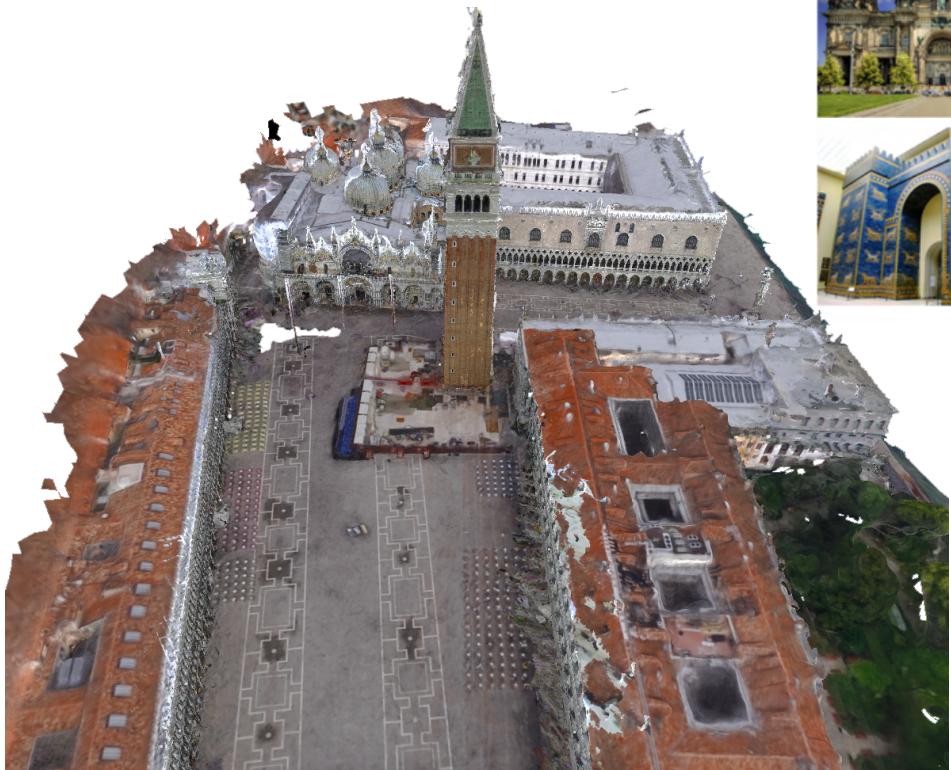


Visión estéreo

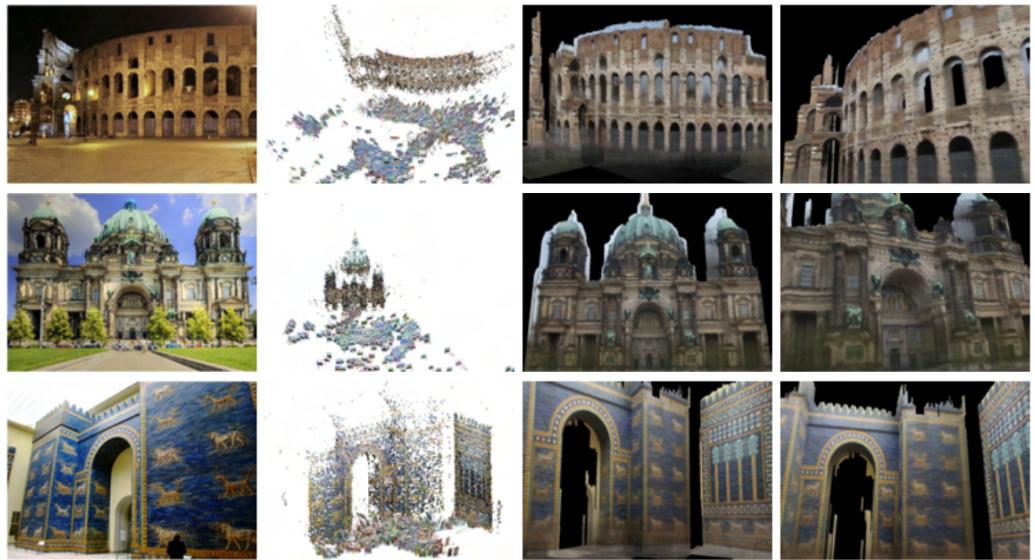


Problema: establecer correspondencias

Reconstrucción 3D



The Visual Turing Test for Scene Reconstruction (3DV'13)

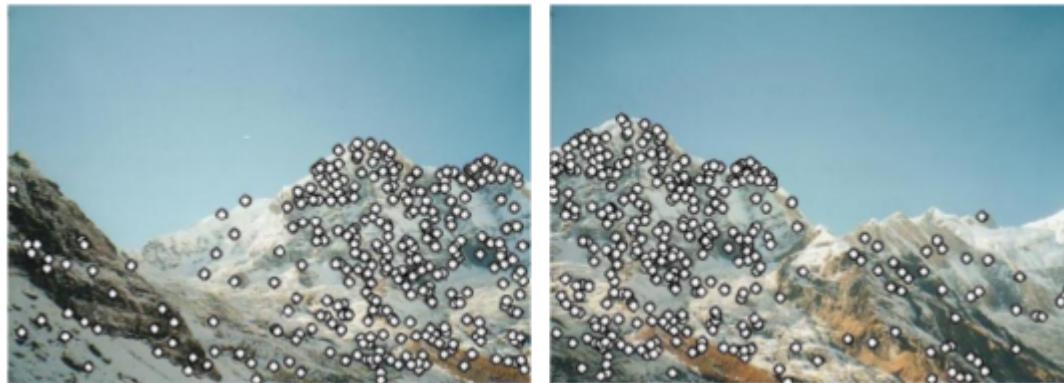


Building Rome in a day (ECCV'10)

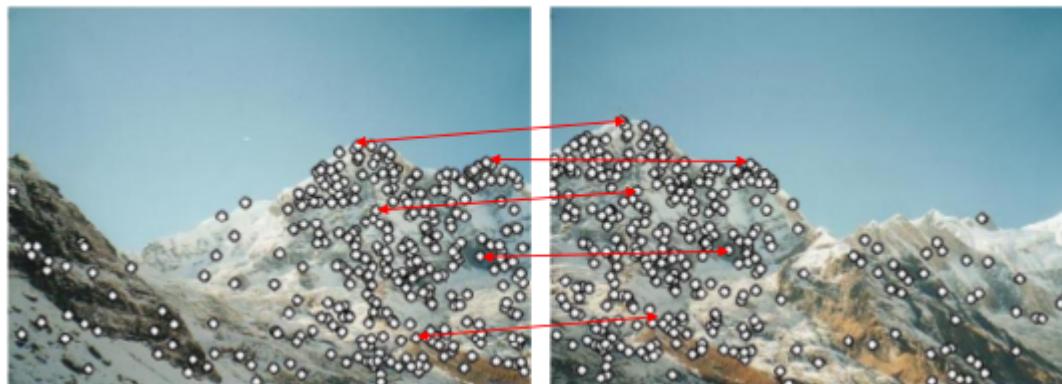


Ejemplo: panorama stitching

1) Detectar características en las imágenes



2) Establecer correspondencias entre pares de puntos



Ejemplo: panorama stitching

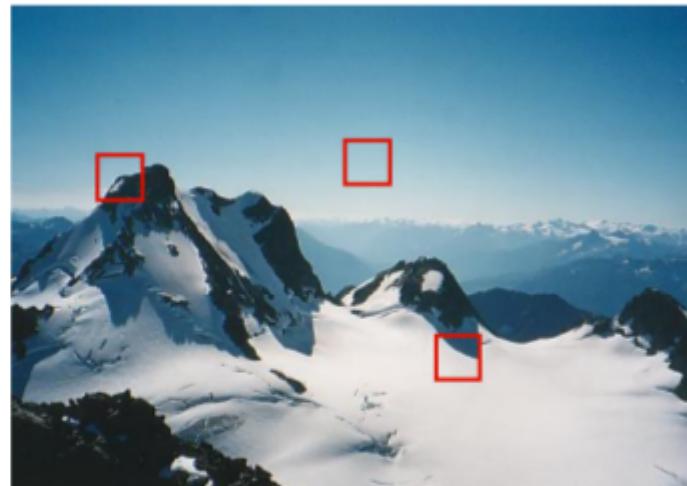
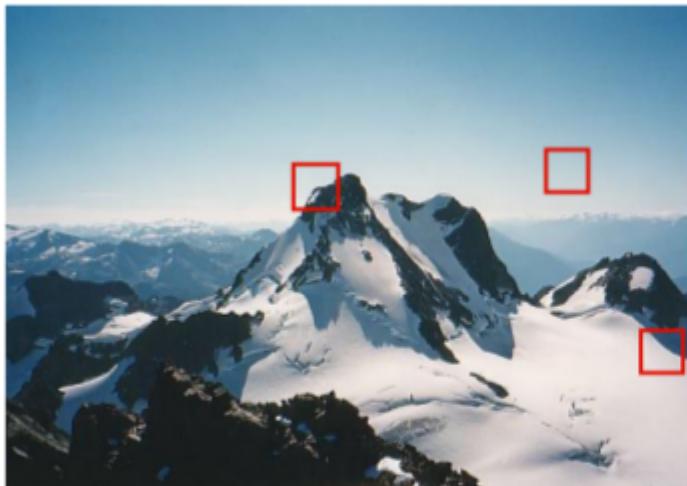
3) Estimar la transformación (homografía) que mejor las alinea



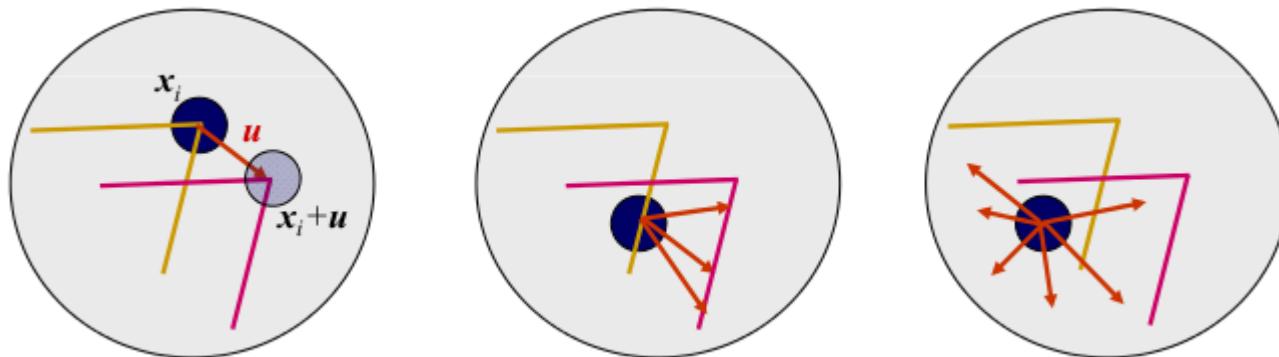
Problemas:

- Detectar los mismos puntos de manera independiente
 - Codificar la información en torno a un punto de forma efectiva
 - Establecer la noción de similitud entre patches
- ... de forma robusta ante cambios de iluminación, puntos de vista, etc.

Puntos y regiones



Detectores



- Las regiones uniformes son casi imposibles de localizar
- Los bordes son ambiguos (problema de apertura)
- Las **esquinas** son más fáciles de localizar → cambios de intensidad en ambas direcciones

Consideremos la siguiente función de comparación:

$$E_{\text{WSSD}}(\mathbf{u}) = \sum_i w(x_i) [I_1(x_i + \mathbf{u}) - I_0(x_i)]^2$$

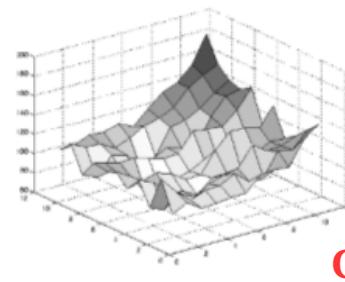
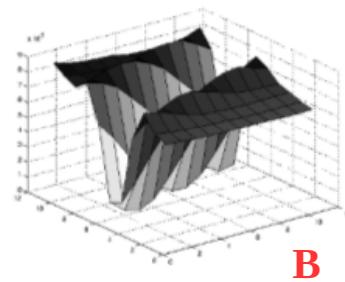
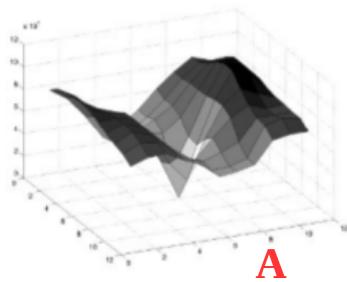
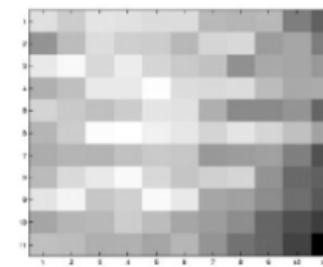
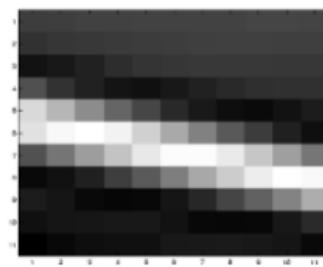
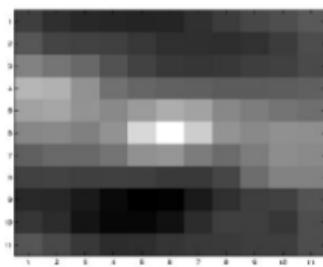
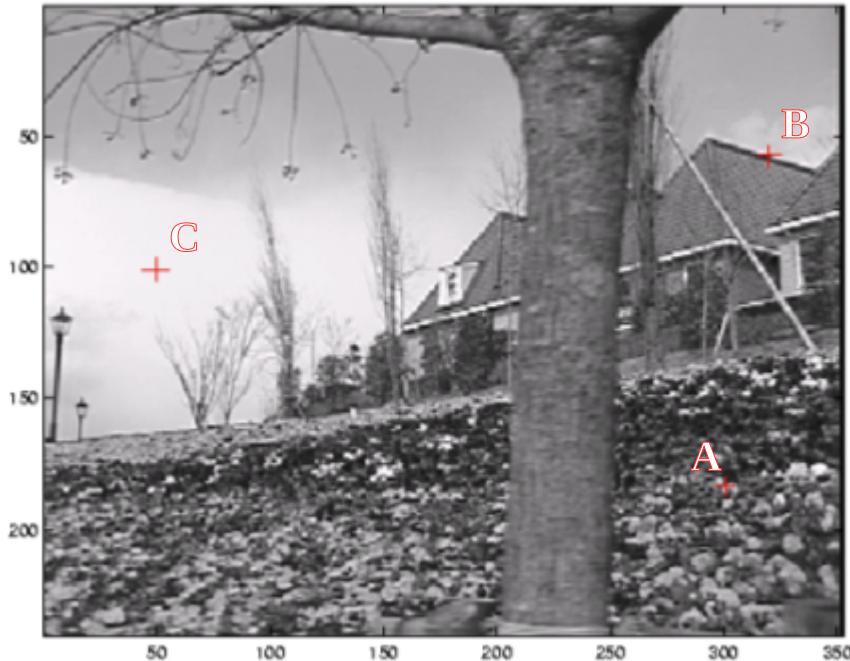
Podemos comparar un *patch* contra si mismo considerando pequeñas variaciones en posición $\Delta\mathbf{u}$:

$$\begin{aligned}
 E_{AC}(\Delta\mathbf{u}) &= \sum_i w(\mathbf{x}_i) [I_0(\mathbf{x}_i + \Delta\mathbf{u}) - I_0(\mathbf{x}_i)]^2 && \text{(función de autocorrelación)} \\
 &\approx \sum_i w(\mathbf{x}_i) [I_0(\mathbf{x}_i) + \nabla I_0(\mathbf{x}_i) \cdot \Delta\mathbf{u} - I_0(\mathbf{x}_i)]^2 \\
 &= \sum_i w(\mathbf{x}_i) [\nabla I_0(\mathbf{x}_i) \cdot \Delta\mathbf{u}]^2 \\
 &= \Delta\mathbf{u}^T \mathbf{A} \Delta\mathbf{u},
 \end{aligned}$$

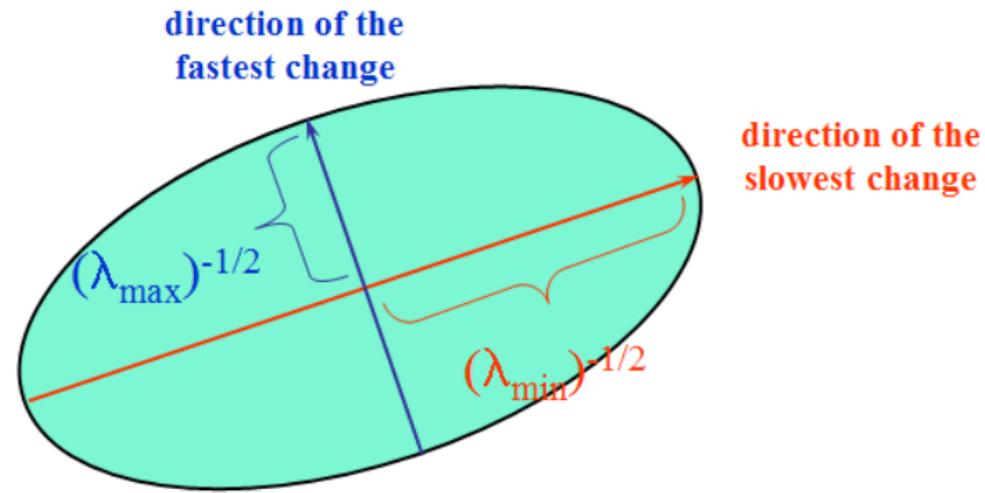
donde: $\nabla I_0(\mathbf{x}_i) = (\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y})(\mathbf{x}_i)$ (gradiente en xi)

$$\mathbf{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad \text{(tensor de estructura)}$$


 convolución

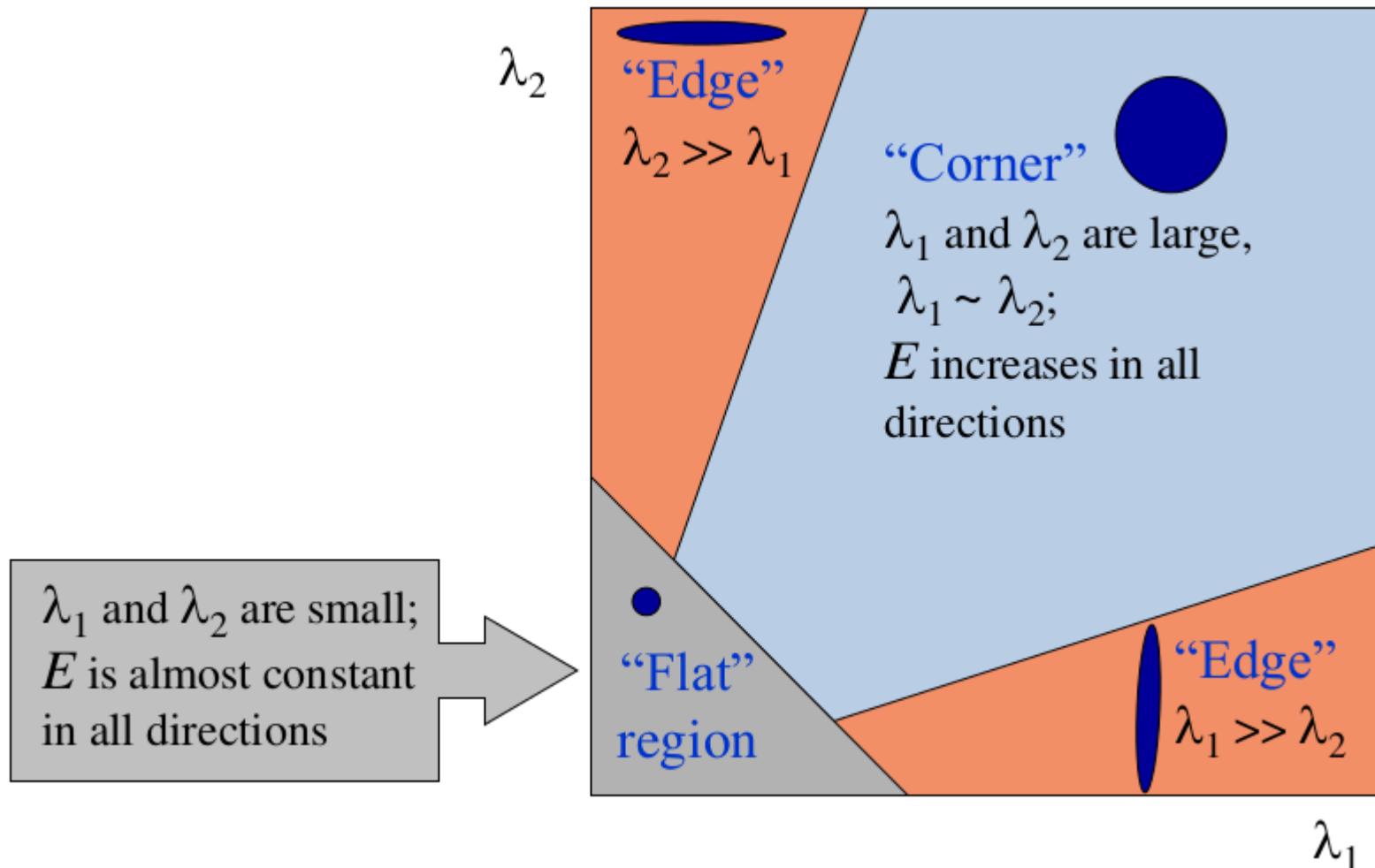


$$\mathbf{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



- Cambios de intensidad \rightarrow autovalores de \mathbf{A}
- Incertidumbre bajo ruido Gaussiano: $\Sigma \mathbf{u} = \sigma_n^2 \mathbf{A}^{-1}$
 $\Rightarrow \mathbf{A}^{-1}$ es una cota inferior a la incertidumbre de \mathbf{u}
- Una mayor incertidumbre depende de λ_{\min}
- buscar esquinas (*corners*) \rightarrow buscar máximos de λ_{\min}

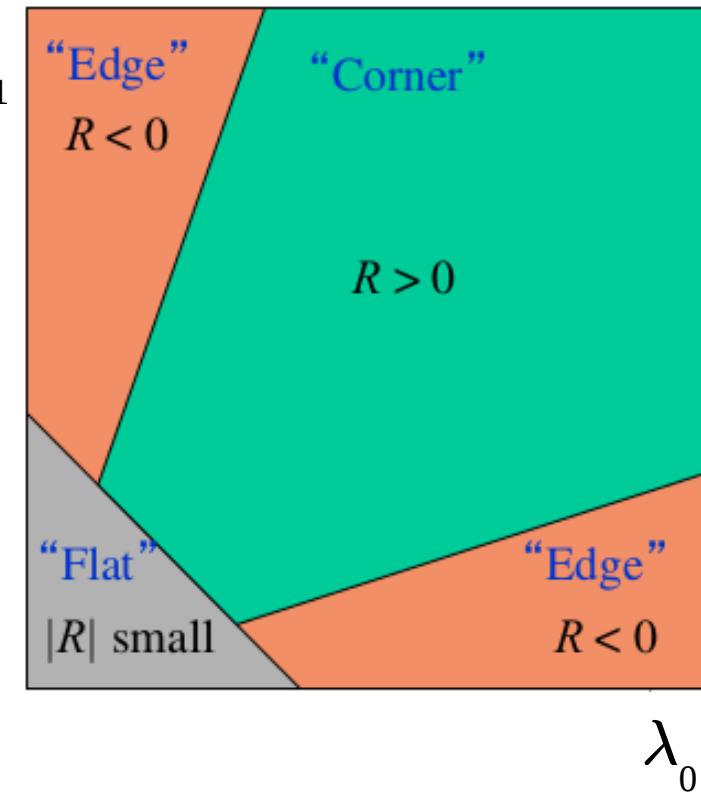
Clasificación de puntos usando los autovalores de A :



Detector de esquinas Harris & Stephens (1988):

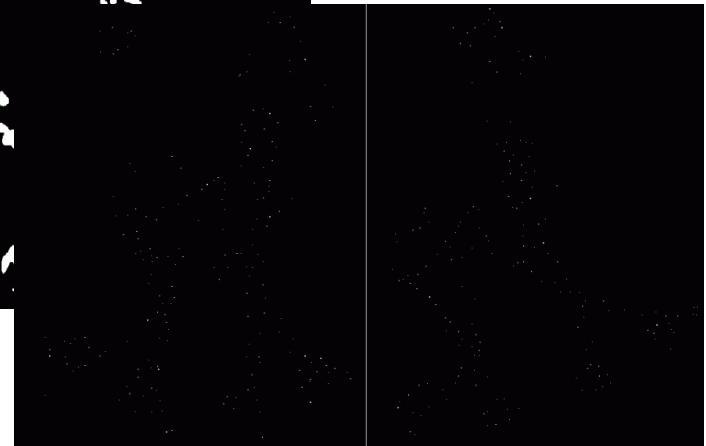
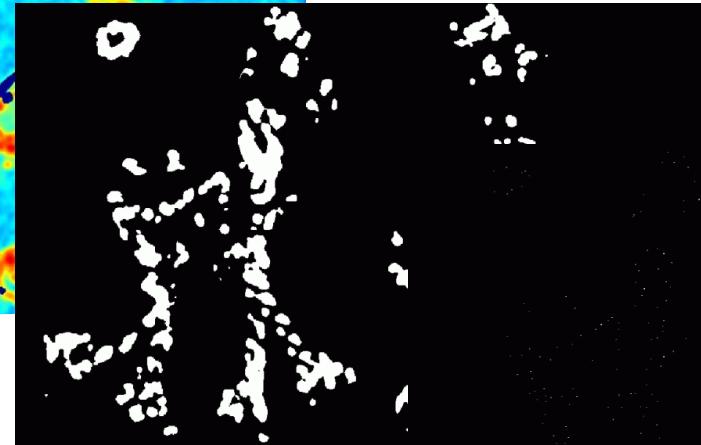
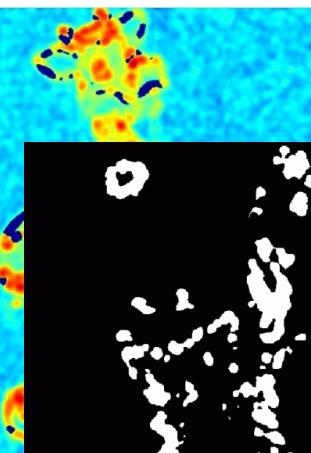
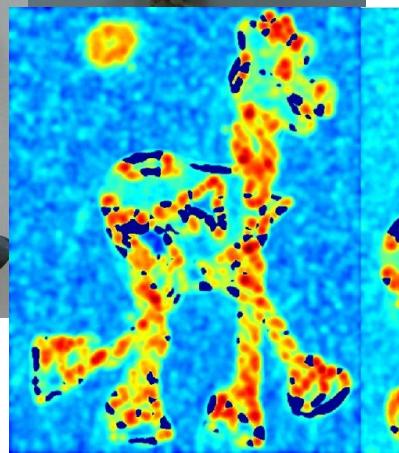
$$R = \det(\mathbf{A}) - \alpha \operatorname{trace}(\mathbf{A})^2 = \lambda_0 \lambda_1 - \alpha(\lambda_0 + \lambda_1)^2$$

- R depende solo de los autovalores de \mathbf{A}
- $R \gg 0$ en esquinas
- $R \ll 0$ en bordes
- $|R| \sim 0$ en regiones uniformes



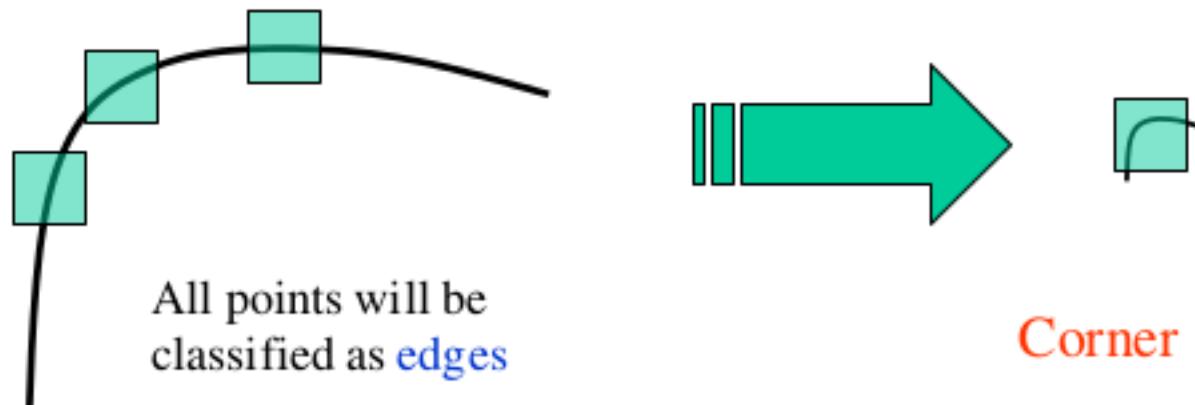
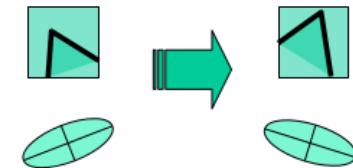
Algorítmico:

- Computar derivadas: I_x , I_y
- Computar términos de A : I_x^2 , $I_x I_y$ e I_y^2
- Suavizar términos de A con un kernel Gaussiano de ancho σ
- Computar R
- Seleccionar puntos con $R > \text{umbral}$
- Retener máximos locales (entorno 8n) de R



Análisis de la función R :

- Invariante ante traslaciones
- Invariante ante rotaciones
 - el elipse rota pero la forma se mantiene
- Invariante ante cambios uniformes de iluminación
 - se basa en las derivadas de I
 - tolerante a escalados de intensidad (se afecta el umbral)
- **No** es invariante ante cambios de *escala*



All points will be
classified as **edges**

Corner !