

Tratamiento Digital de Imágenes

Tema 6: Extracción de características geométricas

...puntos, esquinas, rectas, contornos, regiones...

Miguel Ángel García García
José María Martínez Sánchez



Escuela Politécnica Superior



Universidad Autónoma de Madrid
E28049 Madrid (SPAIN)



Video Processing and Understanding Lab
Grupo de Tratamiento e Interpretación de Vídeo



Índice



- Introducción
- Extracción y caracterización de puntos/esquinas
- Extracción de bordes
- Extracción de regiones

Objetivo

- Extracción automática de características geométricas básicas en imágenes digitales. Etapa previa a la aplicación de algoritmos de análisis y reconocimiento.

Tipos de características geométricas

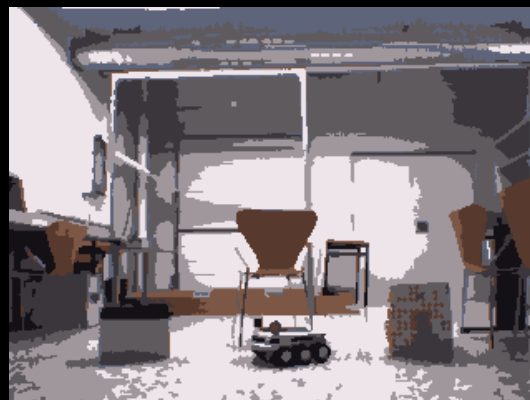
- Puntos, esquinas
- Bordes (rectas, círculos, elipses, curvas)
- Regiones homogéneas (color, textura,...)



Puntos/Esquinas



Rectas/Bordes



Regiones homogéneas

• Aplicaciones

- Extracción de objetos de interés (caras, personas, vehículos, obstáculos, ...)
- Extracción de elementos básicos (suelo, techo, paredes, ...)
- Extracción de características para indexación de imágenes y vídeos en BBDD
- Extracción de delimitadores viales (arcenes, carriles, aceras, ...)
- Reconocimiento óptico de caracteres
- Determinación de movimientos de cámara
- Reconstrucción 3D de escenas a partir de secuencias de imágenes
- etc...

- Introducción
- Extracción y caracterización de puntos/esquinas
 - Detector de Harris & Stephens
 - Detector SUSAN
 - Detector SIFT
- Extracción de bordes
- Extracción de regiones

- Características adimensionales de una imagen que son detectables sin ambigüedad.
- Son posiciones en cuyo vecindario confluyen dos o más bordes con distintas direcciones.
- Los detectores de puntos suelen satisfacer ciertas propiedades de invariancia que permiten la detección de los mismos puntos ante pequeñas transformaciones de las imágenes: translaciones, rotaciones, cambios de escala, variaciones de iluminación, etc.
- Algunos detectores determinan un vector de características para cada punto extraído, teniendo en cuenta la apariencia de su vecindario. Cada vector identifica a su punto respectivo y permite reconocerlo en caso que ese punto aparezca en otras imágenes.
- Detectores de puntos / esquinas más populares:
 - Detector de Harris & Stephens (1988)
 - Detector SUSAN (1997)
 - Detector y caracterizador SIFT (1999, 2004)
 - Detector y caracterizador SURF (2006)

C. Harris and M.J. Stephens. A combined corner and edge detector. *Alvey Vision Conf.*, pp.147–152, 1988

• **Detector con invariancia fuerte a rotaciones, cambios de escala, variaciones de iluminación y ruido en la imagen.**

• **Derivación del detector:**

- Dada una imagen de niveles de gris ψ , se define una función, $E_{x,y}(\Delta x, \Delta y)$, que determina la similitud entre una ventana centrada en el píxel $[x, y]$ y la misma ventana centrada en el píxel $[x + \Delta x, y + \Delta y]$, siendo $(\Delta x, \Delta y)$ un desplazamiento dado. Se consideran ventanas cuadradas de tamaño $2W + 1$. La similitud se estima mediante Suma de Diferencias Cuadráticas:

$$E_{x,y}(\Delta x, \Delta y) = \sum_{x_i} \sum_{y_j} (\psi[x_i + \Delta x, y_j + \Delta y] - \psi[x_i, y_j])^2, \quad \forall x_i \in [x - W, x + W], \quad \forall y_j \in [y - W, y + W]$$

- Considerando desplazamientos $(\Delta x, \Delta y)$ pequeños:
 - Si un píxel $[x, y]$ pertenece a una región uniforme, la función $E_{x,y}$ retornará valores próximos a cero para cualquier desplazamiento.
 - Si pertenece a un borde, retornará valores pequeños para aquellos desplazamientos en la dirección del borde y grandes para el resto.
 - Si pertenece a una esquina, $E_{x,y}$ retornará valores grandes para cualquier desplazamiento (en este principio se basa el detector de esquinas previamente propuesto por H. Moravec en 1980).

- Para que el comportamiento del detector sea más robusto a ruido, se aplica una ventana circular suave (Gaussiana), multiplicando cada término del doble sumatorio por un factor de peso w_{x_i, y_i} :

$$E_{x,y}(\Delta x, \Delta y) = \sum_{x_i} \sum_{y_i} w_{x_i, y_i} \left(\psi[x_i + \Delta x, y_i + \Delta y] - \psi[x_i, y_i] \right)^2, \quad w_{x_i, y_i} = e^{-((x_i-x)^2 + (y_i-y)^2) / 2\sigma^2}$$

- Desarrollando por serie de Taylor de primer orden:

$$\psi[x_i + \Delta x, y_i + \Delta y] = \psi[x_i, y_i] + \frac{\partial \psi}{\partial x}[x_i, y_i] \Delta x + \frac{\partial \psi}{\partial y}[x_i, y_i] \Delta y$$

- Las derivadas parciales de la imagen se aproximan mediante convolución unidimensional:

$$\frac{\partial \psi}{\partial x}[x_i, y_i] \approx \psi[x_i, y_i] * [-1, 0, 1] \quad \frac{\partial \psi}{\partial y}[x_i, y_i] \approx \psi[x_i, y_i] * [-1, 0, 1]^T$$

- La función de similitud queda así expresada como:

$$\begin{aligned} E_{x,y}(\Delta x, \Delta y) &= \sum_{x_i} \sum_{y_i} w_{x_i, y_i} \left(\frac{\partial \psi}{\partial x}[x_i, y_i] \Delta x + \frac{\partial \psi}{\partial y}[x_i, y_i] \Delta y \right)^2 \\ &= \sum_{x_i} \sum_{y_i} w_{x_i, y_i} \left(\frac{\partial \psi^2}{\partial x}[x_i, y_i] \Delta x^2 + \frac{\partial \psi^2}{\partial y}[x_i, y_i] \Delta y^2 + 2\Delta x \Delta y \frac{\partial \psi}{\partial x}[x_i, y_i] \frac{\partial \psi}{\partial y}[x_i, y_i] \right) \\ &= \Delta x^2 \sum_{x_i} \sum_{y_i} w_{x_i, y_i} \frac{\partial \psi^2}{\partial x}[x_i, y_i] + \Delta y^2 \sum_{x_i} \sum_{y_i} w_{x_i, y_i} \frac{\partial \psi^2}{\partial y}[x_i, y_i] + 2\Delta x \Delta y \sum_{x_i} \sum_{y_i} w_{x_i, y_i} \frac{\partial \psi}{\partial x}[x_i, y_i] \frac{\partial \psi}{\partial y}[x_i, y_i] \end{aligned}$$

- Los cuadrados de las derivadas parciales de la imagen se obtienen elevando al cuadrado cada uno de sus elementos. Los productos de derivadas parciales de la imagen se obtienen multiplicando sus elementos homólogos elemento a elemento.

- Los sumatorios anteriores son equivalentes a convolucionar con una Gaussiana los cuadrados de las derivadas parciales, así como el producto de derivadas parciales (se les aplica un filtro de la Gaussiana):

$$\sum_{x_i} \sum_{y_i} w_{x_i, y_i} \frac{\partial \psi^2}{\partial x}[x_i, y_i] = \left(\frac{\partial \psi^2}{\partial x} * w_{x,y} \right) [x, y] \equiv \overline{\frac{\partial \psi^2}{\partial x}} [x, y]$$

$$\sum_{x_i} \sum_{y_i} w_{x_i, y_i} \frac{\partial \psi^2}{\partial y}[x_i, y_i] = \left(\frac{\partial \psi^2}{\partial y} * w_{x,y} \right) [x, y] \equiv \overline{\frac{\partial \psi^2}{\partial y}} [x, y]$$

$$\sum_{x_i} \sum_{y_i} w_{x_i, y_i} \frac{\partial \psi}{\partial x}[x_i, y_i] \frac{\partial \psi}{\partial y}[x_i, y_i] = \left(\frac{\partial \psi}{\partial x} \frac{\partial \psi}{\partial y} * w_{x,y} \right) [x, y] \equiv \overline{\frac{\partial \psi \partial \psi}{\partial x \partial y}} [x, y]$$

- La función de similitud queda así expresada como:

$$E_{x,y}(\Delta x, \Delta y) = \Delta x^2 \overline{\frac{\partial \psi^2}{\partial x}} [x, y] + \Delta y^2 \overline{\frac{\partial \psi^2}{\partial y}} [x, y] + 2\Delta x \Delta y \overline{\frac{\partial \psi \partial \psi}{\partial x \partial y}} [x, y]$$

$$E_{x,y}(\Delta x, \Delta y) = (\Delta x, \Delta y) \begin{pmatrix} \overline{\frac{\partial \psi^2}{\partial x}} [x, y] & \overline{\frac{\partial \psi \partial \psi}{\partial x \partial y}} [x, y] \\ \overline{\frac{\partial \psi \partial \psi}{\partial x \partial y}} [x, y] & \overline{\frac{\partial \psi^2}{\partial y}} [x, y] \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = (\Delta x, \Delta y) \mathbf{M}_{x,y} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

- Según el principio de Moravec, el valor de $E_{x,y}$ para un píxel que forme parte de una esquina ha de ser elevado para cualquier desplazamiento. Esto ocurrirá siempre que los dos valores propios de la matriz $\mathbf{M}_{x,y}$, λ_1 y λ_2 , sean elevados.

- Se define la función de respuesta de esquina en un píxel $[x, y]$ como:

$$R[x, y] = \lambda_1 \lambda_2 - k (\lambda_1 + \lambda_2)^2$$

- con la constante k definida empíricamente entre 0.04 (valor sugerido por Harris) y 0.15.
- El cálculo de los valores propios se puede evitar substituyendo el producto de valores propios por el determinante de la matriz y la suma de valores propios por la traza:

$$R[x, y] = \det(\mathbf{M}_{x,y}) - k \text{Tr}^2(\mathbf{M}_{x,y})$$

- Los píxeles candidatos de ser esquinas han de tener un valor de $R[x, y]$ elevado.

Procedimiento resumido:

- Obtener las dos imágenes correspondientes a la derivada parcial horizontal y vertical de la imagen dada ($\frac{\partial \psi}{\partial x}$, $\frac{\partial \psi}{\partial y}$) mediante sendas convoluciones con vectores 1D: $[-1, 0, 1]$ y $[-1, 0, 1]^t$
- Obtener las tres imágenes producto de derivadas parciales ($\frac{\partial \psi^2}{\partial x}$, $\frac{\partial \psi^2}{\partial y}$, $\frac{\partial \psi}{\partial x} \frac{\partial \psi}{\partial y}$) mediante productos elemento a elemento de las imágenes correspondientes.
- Aplicar filtrado Gaussiano a las tres imágenes producto anteriores.
- Para cada píxel de la imagen dada, calcular la matriz $\mathbf{M}_{x,y}$ a partir de sus píxeles correspondientes en las tres imágenes filtradas anteriores y determinar la función de respuesta de esquina de ese píxel.
- Detección de bordes: Píxeles con un valor propio pequeño y el otro grande.

Detector de Shi & Tomasi:

- Variante del detector de Harris & Stephens en que la función de respuesta de esquina de un píxel $[x, y]$ se define como el mínimo de los dos valores propios de $\mathbf{M}_{x,y}$: $R[x, y] = \min(\lambda_1, \lambda_2)$

• Inconvenientes del detector de esquinas de Harris & Stephens:

- Necesidad de definir empíricamente un valor umbral para la función de respuesta de esquinas $R[x, y]$
- Los píxeles cuyas funciones de respuesta de esquina superan el umbral forman regiones alrededor de la esquina: no se identifican puntos sino regiones.



Resaltados en negro los píxeles que superan el umbral de la función de respuesta de esquina del detector de Harris & Stephens.

El umbral en cada imagen es diez veces el de la imagen anterior.

Todo el proceso se aplica sobre la imagen inicial en niveles de gris.



• Solución a la detección de regiones en lugar de puntos:

- Seleccionar como esquinas sólo aquellos píxeles cuyo valor de la función de respuesta de esquina sea un máximo local (el valor máximo en un vecindario de por ejemplo 3x3 píxeles).



- Además se pueden descartar aquellos píxeles que cumplan el criterio anterior pero estén cercanos a otro píxel previamente seleccionado como esquina. La distancia mínima admisible es un valor escogido empíricamente.

S.M. Smith and J.M. Brady, *SUSAN - a new approach to low level image processing*, *International Journal of Computer Vision*, 23(1):45-78, Mayo 1997

• **SUSAN (Smallest Univalve Segment Assimilating Nucleus)**

- *Detector con invariancia parcial a rotaciones, cambios de escala, variaciones de iluminación y ruido en la imagen.*
- *Robustez al ruido por no requerir el cálculo de derivadas parciales de la imagen dada.*
- *Eficiencia elevada por no requerir cálculo de convoluciones.*

Derivación del detector:

- Para cada píxel $[x, y]$ de una imagen de niveles de gris ψ , se considera una ventana circular de radio ρ centrada en ese píxel:

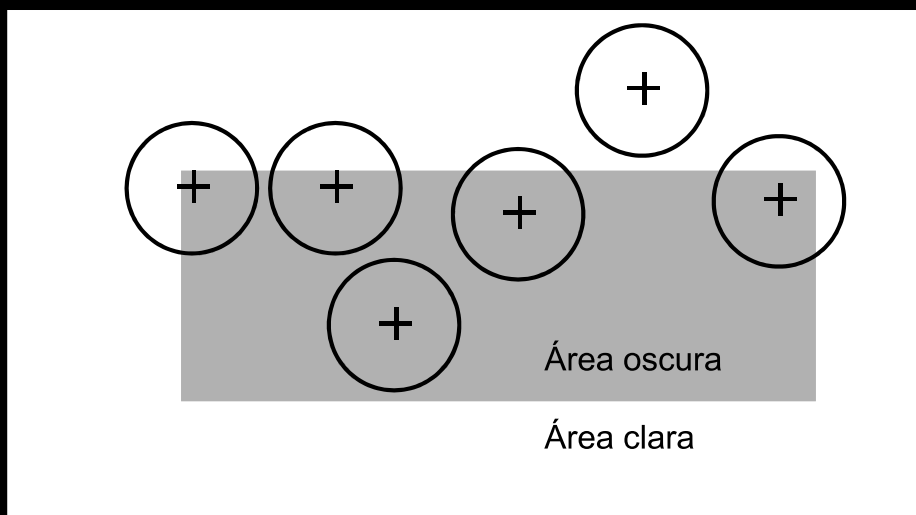
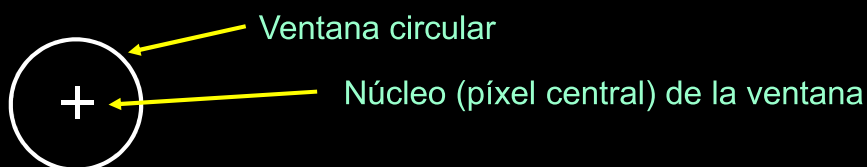
$$W_{x,y} = \left\{ (x_i, y_i) \mid \sqrt{(x_i - x)^2 + (y_i - y)^2} \leq \rho \right\}$$

- Se define una función $n_\psi(x, y)$ que determina cuántos píxeles de la ventana tienen un nivel de gris cuya diferencia con el nivel de gris del píxel central no supera un umbral τ

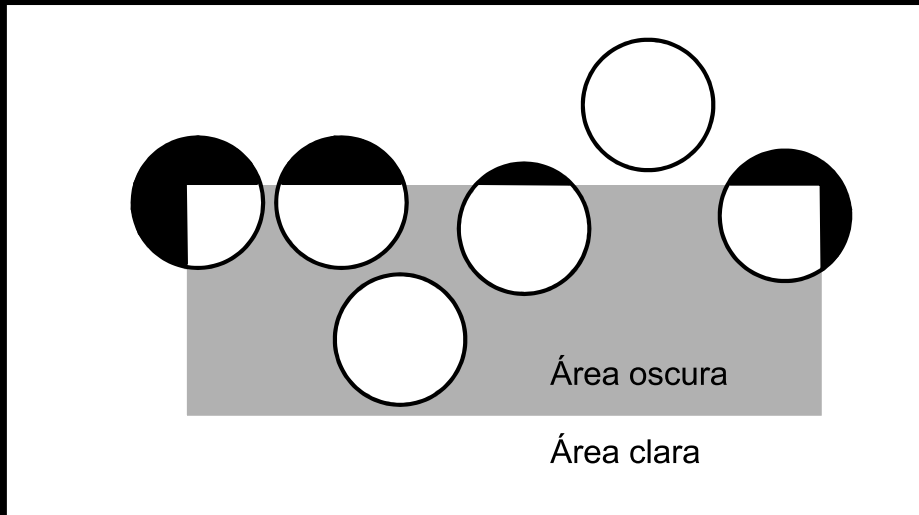
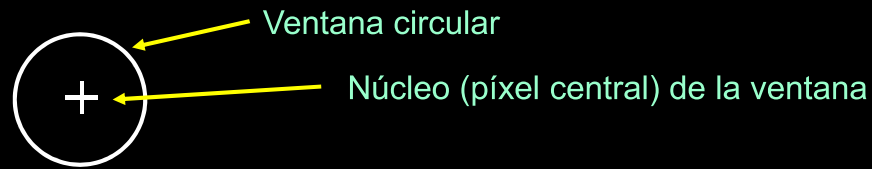
$$n_\psi(x, y) = \sum_{W_{x,y}} c_{\psi,x,y}(x_i, y_i)$$

$$c_{\psi,x,y}(x_i, y_i) = \begin{cases} 1 & \|\psi[x_i, y_i] - \psi[x, y]\| \leq \tau \\ 0 & \|\psi[x_i, y_i] - \psi[x, y]\| > \tau \end{cases}$$

- El "segmento" de la ventana circular cuyos píxeles satisfacen el criterio booleano de la función $c_{\psi,x,y}(x_i, y_i)$ componen el USAN (Univalue Segment Assimilating Nucleus). La función $n_\psi(x, y)$ determina el área del segmento USAN (número de píxeles con un nivel de gris parecido al del píxel central).



Se centra una ventana circular en cada uno de los píxeles de la imagen dada



Segmentos USAN (*Univalue Segment Assimilating Nucleus*) en blanco.
El área mínima de los segmentos USAN se da en las esquinas de los objetos de la imagen.

- El detector tiene un comportamiento más estable si se suavizan los flancos de la función $c_{\psi,x,y}(x_i, y_i)$ definiéndola alternativamente como:

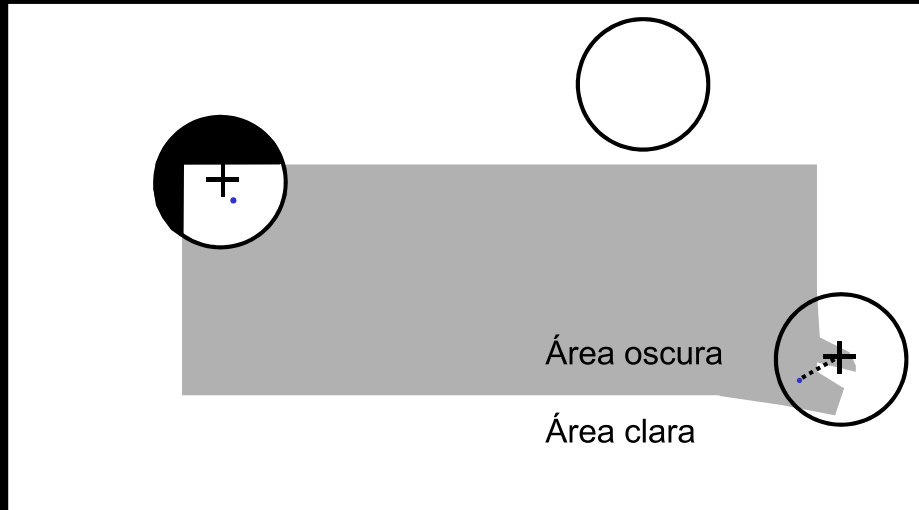
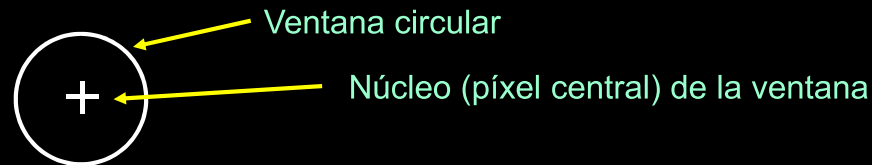
$$c_{\psi,x,y}(x_i, y_i) = e^{-\left(\frac{\psi[x_i, y_i] - \psi[x, y]}{\tau}\right)^6}$$

- A partir del área USAN, se define una función de respuesta de esquina $R_{\psi}(x, y)$ que descarta aquellos segmentos USAN cuyas áreas superan la mitad del área total de la ventana circular $W_{x,y}$

$$R_{\psi}(x, y) = \begin{cases} \gamma - n_{\psi}(x, y) & n_{\psi}(x, y) < \gamma \\ 0 & n_{\psi}(x, y) \geq \gamma \end{cases} \quad \gamma = |W_{x,y}| / 2$$

- Para disminuir el riesgo de detección de falsas esquinas en imágenes ruidosas o poco contrastadas, también se anula la función de respuesta correspondiente a:
 - Los segmentos USAN cuyos centros de masas están próximos al centro de sus ventanas circulares correspondientes.
 - Las ventanas circulares tales que algunos de los píxeles que pertenecen a la recta que une el centro de la ventana con el centro de masas del segmento USAN asociado no están contenidos en el segmento USAN.
- Finalmente se seleccionan como esquinas aquellos píxeles cuya función de respuesta es un máximo global en un vecindario de 5x5 píxeles:

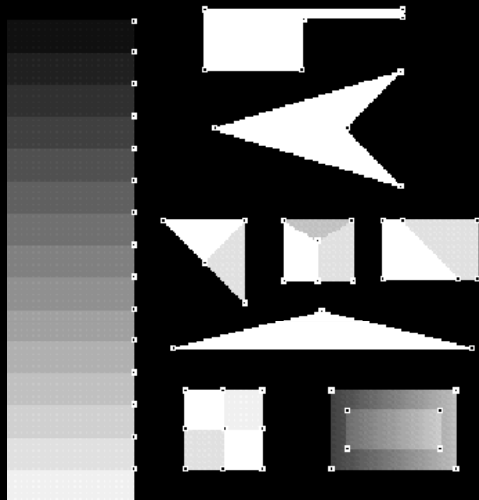
$$\text{esquina}(\psi[x, y]) \Leftrightarrow R_{\psi}(x, y) > R_{\psi}(x_i, y_i), \forall x_i \in [x-2, x+2], \forall y_i \in [y-2, y+2], x_i \neq x, y_i \neq y$$



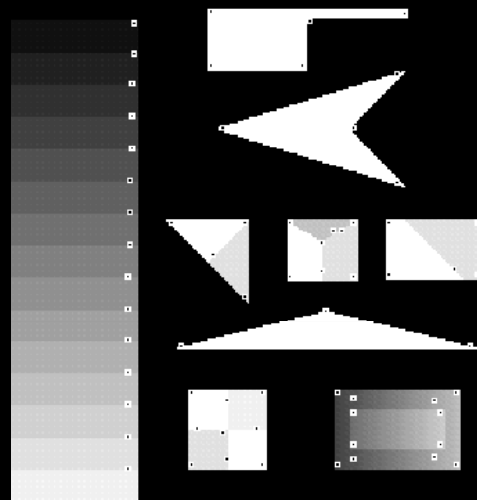
Segmentos USAN (*Univalue Segment Assimilating Nucleus*) rechazados

• Procedimiento resumido:

- Se centra una ventana circular en cada píxel $[x, y]$ y se evalúa la función $n_w(x, y)$, que indica cuántos píxeles dentro de esa ventana tienen un nivel de gris cuya diferencia con el nivel de gris del píxel central es inferior a τ unidades.
- Este umbral determina el contraste mínimo de las esquinas detectadas y la cantidad máxima de ruido que será ignorado, controlando así la densidad de esquinas generadas. Los píxeles que cumplen ese criterio constituyen el segmento USAN.
- Los autores recomiendan un radio de ventana, ρ de 3.4 píxeles (la ventana contiene así 37 píxeles) y un umbral τ de 25 para la mayoría de imágenes, que puede ser reducido hasta un valor de 7 para imágenes poco contrastadas.
- Se descartan aquellas ventanas con un valor de la función $n_w(x, y)$ superior a $37/2$.
- De las ventanas que superan el criterio anterior, se descartan aquellas en las que el centro de masas de su segmento USAN está próximo al centro de la ventana, o en las que alguno de los píxeles situados en la recta que une ese centro de masas y el centro de la ventana no pertenece al segmento USAN.
- Se escogen como esquinas aquellos píxeles cuyo valor de $n_w(x, y)$ es un mínimo absoluto en una ventana de 5×5 píxeles centrada en ellos. De cara al cálculo del mínimo, se consideran los píxeles cuyas ventanas han sido descartadas a partir de los criterios previos.

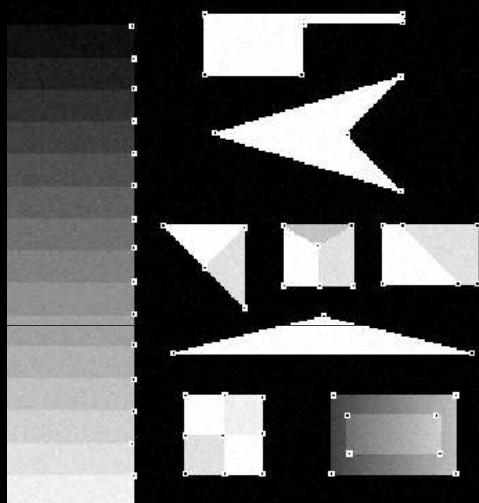


Ejemplo detector SUSAN
(imagen sintética)

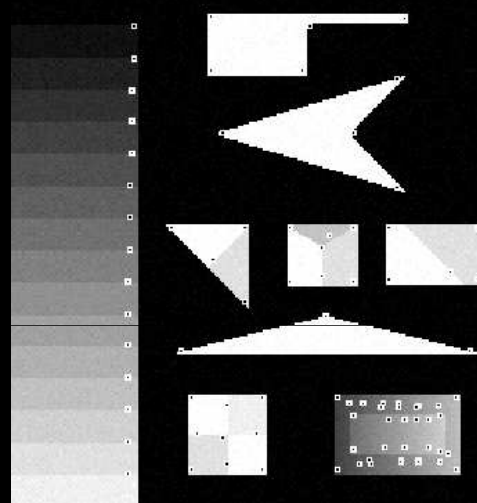


Ejemplo detector Harris & Stephens
(imagen sintética)

Resultados publicados en: S.M. Smith and J.M. Brady, SUSAN - a new approach to low level image processing. International Journal of Computer Vision, 23(1):45-78, Mayo 1997

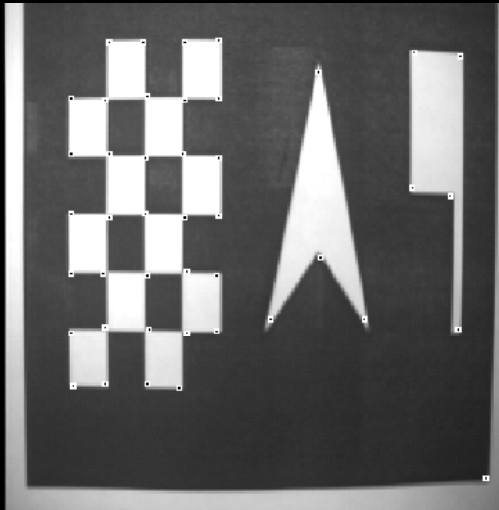


Ejemplo detector SUSAN
(imagen sintética con ruido)



Ejemplo detector Harris & Stephens
(imagen sintética con ruido)

Resultados publicados en: S.M. Smith and J.M. Brady, SUSAN - a new approach to low level image processing. International Journal of Computer Vision, 23(1):45-78, Mayo 1997



Ejemplos detector SUSAN (imágenes reales)

Resultados publicados en: S.M. Smith and J.M. Brady, SUSAN - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45-78, Mayo 1997

• Detección de bordes:

- Mismos pasos que el detector de esquinas, pero sólo se descartan ventanas con $n_w(x, y) > 3 |W_{x,y}| / 2$. Se obtiene la dirección del borde a partir de los momentos de los segmentos USAN.

David G. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, 60, 2 (2004), pp. 91-110.

• SIFT (Scale Invariant Feature Transform)

- Detector con invariancia a rotaciones y cambios de escala, e invariancia parcial (robustez) ante transformaciones afines, ruido, cambios de punto de vista y de iluminación.
- Además de detectar esquinas, obtiene una descripción de las mismas (vector de características) que permite identificarlas y reconocerlas en aquellas imágenes donde aparezcan.

• Derivación del detector:

- Dada una imagen de niveles de gris ψ , se le aplica un filtrado paso bajo mediante convolución con una función Gaussiana, obteniendo una nueva imagen suavizada L_σ :

$$L_\sigma[x, y] = \psi[x, y] * G(x, y, \sigma)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

- Variando la desviación estándar, se obtiene una secuencia de imágenes filtradas cada vez más suavizadas (con menos detalles). El valor $k\sigma$ constituye la escala asociada a cada imagen:

$${}^1\Lambda_\sigma = \left\{ L_{k\sigma}[x, y] \mid k = 2^i, \forall i \in [0, s+2] \right\}$$

- Por ejemplo, si $s = 2$, se obtiene una secuencia de 5 imágenes Gaussianas con escalas $\{\sigma, 2^{1/2}\sigma, 2\sigma, 2^{3/2}\sigma, 2^2\sigma\}$

$${}^1\Lambda_\sigma = \{ L_\sigma[x, y], L_{2^{1/2}\sigma}[x, y], L_{2\sigma}[x, y], L_{2^{3/2}\sigma}[x, y], L_{2^2\sigma}[x, y] \}$$

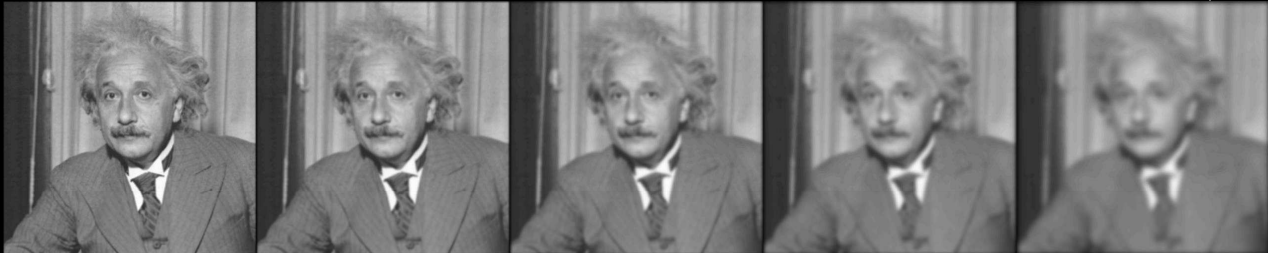
$$L_{\sigma}[x, y]$$

$$L_{2^{1/2}\sigma}[x, y]$$

$$L_{2\sigma}[x, y]$$

$$L_{2^{3/2}\sigma}[x, y]$$

$$L_{2^2\sigma}[x, y]$$



Ejemplo de secuencia de imágenes Gaussianas, ${}^1\Lambda_{\sigma}$, generada por el detector SIFT ($s = 2$)

Adaptación del material del curso "Foundations of Computer Vision", Prof. Allan Jepson, Universidad de Toronto

- A continuación se genera una nueva imagen Gaussiana submuestreando por dos cada dimensión de la imagen $L_{2\sigma}$:

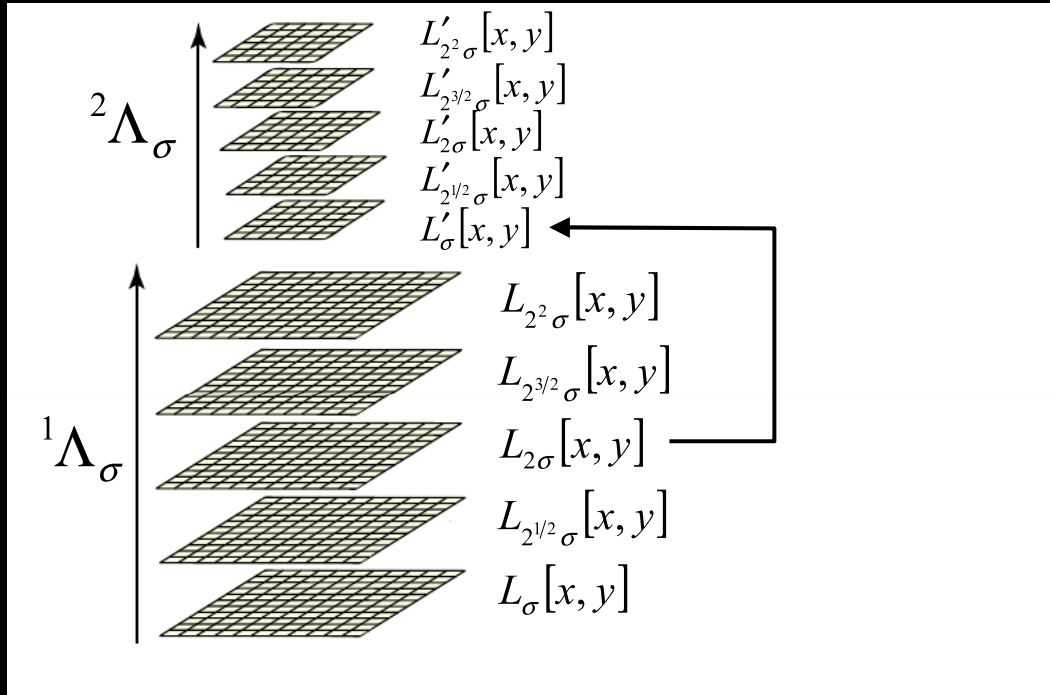
$$L'_{\sigma}[x, y] = L_{2\sigma}[2x+1, 2y+1]$$

- Variando la desviación estándar se genera una nueva secuencia de imágenes Gaussianas:

$${}^2\Lambda_{\sigma} = \left\{ L'_{k\sigma}[x, y] \mid k = 2^i, \forall i \in [0, s+2] \right\}$$

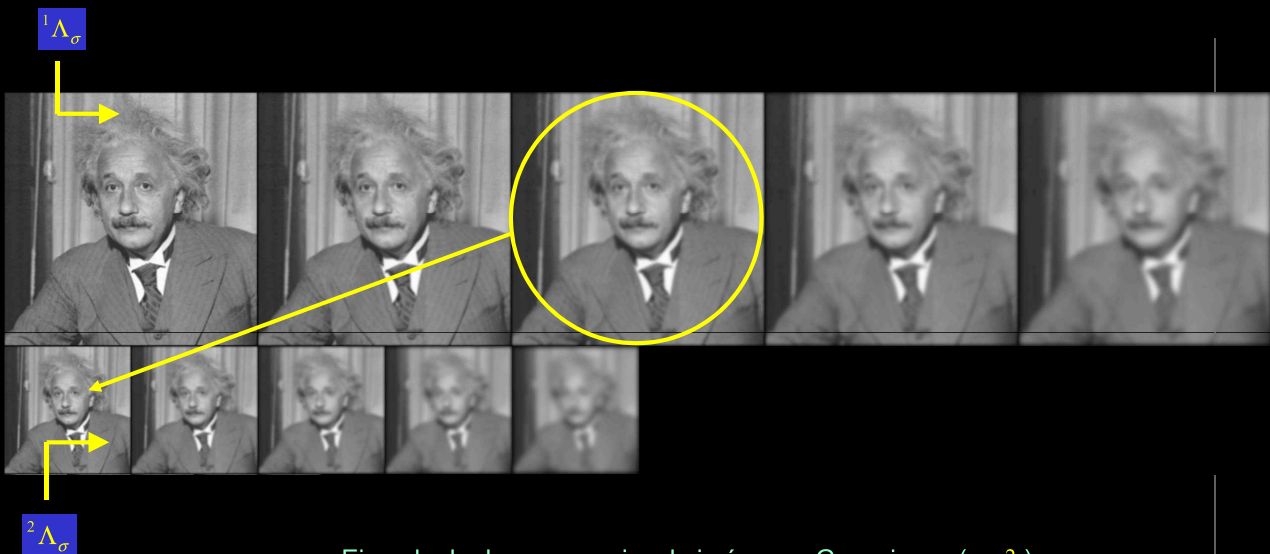
- Si $s = 2$, se obtiene una nueva secuencia de 5 imágenes Gaussianas con escalas $\{\sigma, 2^{1/2}\sigma, 2^{2/2}\sigma, 2^{3/2}\sigma, 2^{4/2}\sigma\}$:

$${}^2\Lambda_{\sigma} = \left\{ L'_{\sigma}[x, y], L'_{2^{1/2}\sigma}[x, y], L'_{2\sigma}[x, y], L'_{2^{3/2}\sigma}[x, y], L'_{2^2\sigma}[x, y] \right\}$$



Ejemplo de dos secuencias de imágenes Gaussianas ($s = 2$)

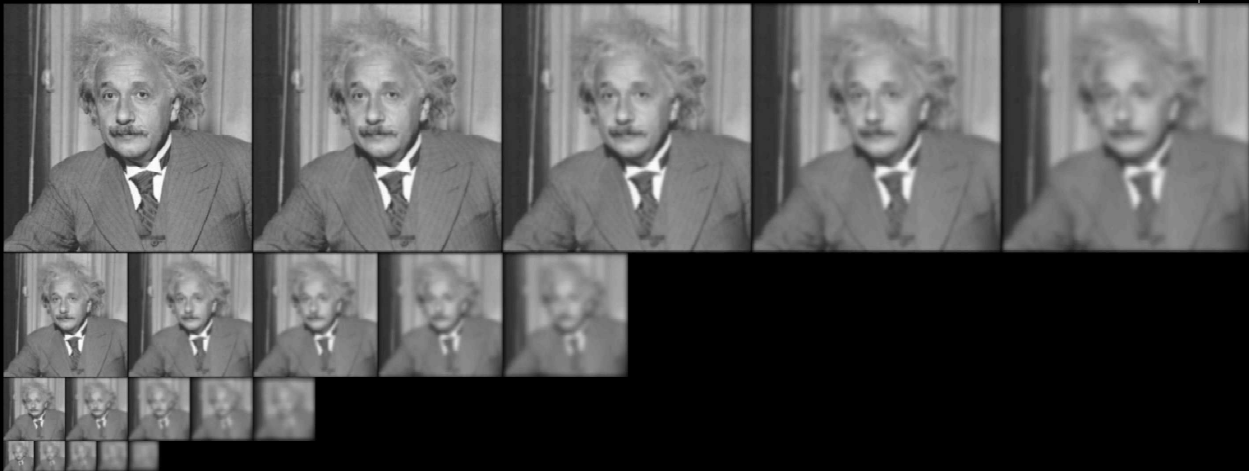
Adaptación del material del curso "Foundations of Computer Vision", Prof. Allan Jepson, Universidad de Toronto



Ejemplo de dos secuencias de imágenes Gaussianas ($s = 2$)

Adaptación del material del curso "Foundations of Computer Vision", Prof. Allan Jepson, Universidad de Toronto

- Así se generan sucesivas secuencias de imágenes Gaussianas hasta que se llega a un tamaño mínimo de las imágenes muestreadas.



Ejemplo de cuatro secuencias de imágenes Gaussianas ($s = 2$)

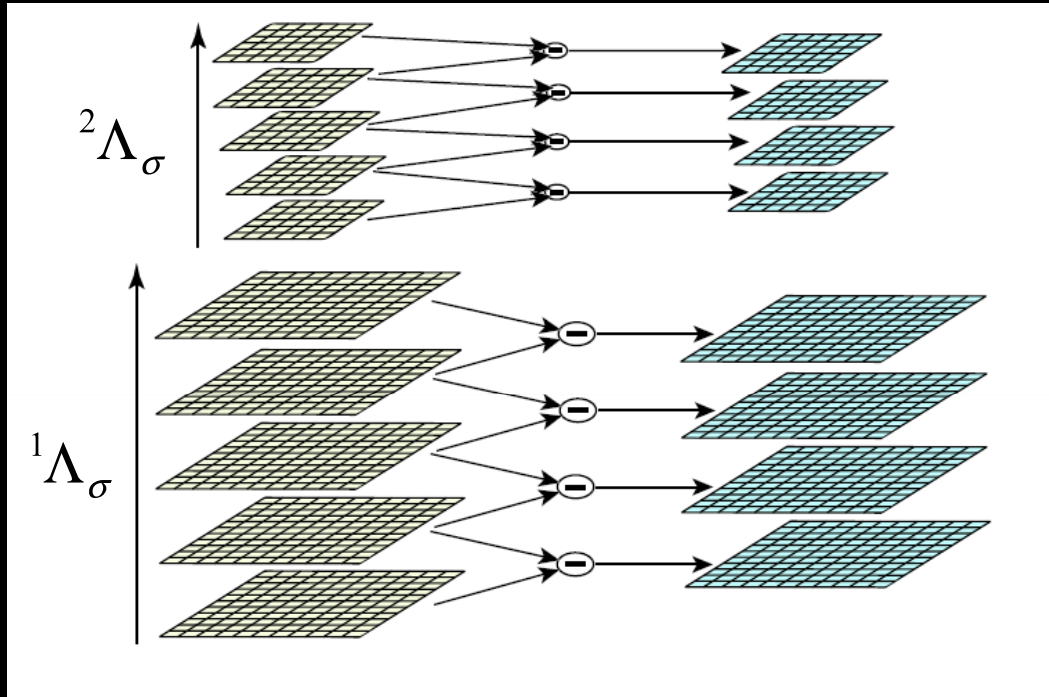
Material del curso "Foundations of Computer Vision", Prof. Allan Jepson, Universidad de Toronto

- Por cada pareja consecutiva de imágenes Gaussianas pertenecientes a la misma secuencia $J \Lambda_\sigma$, y para todas las secuencias, se calcula una nueva imagen, diferencia de ambas:

$$D_{k,\sigma} [x, y] = L_{k,\sigma} [x, y] - L_{k+1,\sigma} [x, y]$$

- Por ejemplo, si $s = 2$ se obtienen las siguientes diferencias de Gaussianas para la secuencia $^1 \Lambda_\sigma$:

$$\{ D_\sigma [x, y], D_{2^{1/2}\sigma} [x, y], D_{2\sigma} [x, y], D_{2^{3/2}\sigma} [x, y] \}$$



- $D_{2^{3/2}\sigma}[x, y]$
- $D_{2\sigma}[x, y]$
- $D_{2^{1/2}\sigma}[x, y]$
- $D_{\sigma}[x, y]$

Ejemplo de dos secuencias de diferencias entre imágenes Gaussianas ($s = 2$)

Adaptación del material del curso "Foundations of Computer Vision", Prof. Allan Jepson, Universidad de Toronto

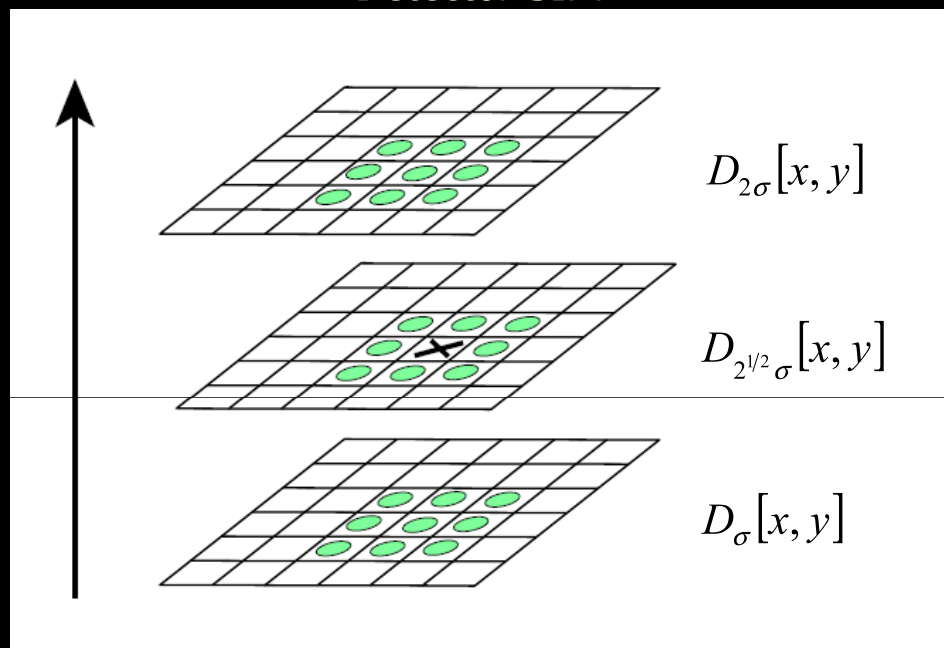
- $D_{\sigma}[x, y]$
- $D_{2^{1/2}\sigma}[x, y]$
- $D_{2\sigma}[x, y]$
- $D_{2^{3/2}\sigma}[x, y]$



Ejemplo de cuatro secuencias de imágenes de diferencias de Gaussianas ($s = 2$)

Material del curso "Foundations of Computer Vision", Prof. Allan Jepson, Universidad de Toronto

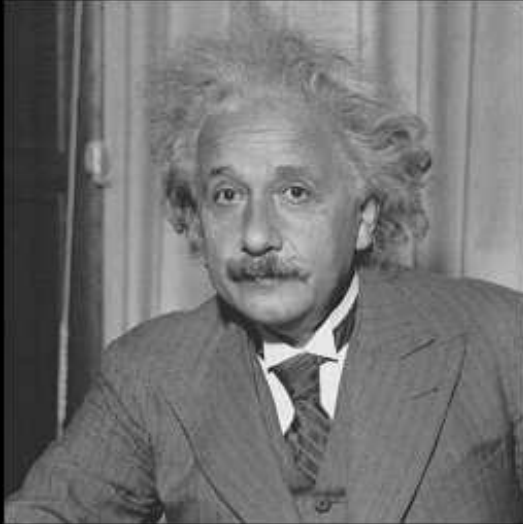
- Cada píxel en una imagen de diferencia de Gaussianas es comparado con 26 vecinos: sus 8 vecinos de la misma imagen y los 9 vecinos correspondientes a las dos imágenes adyacentes (en total 9x2 píxeles vecinos) anterior y posterior de la misma secuencia.
- Por ejemplo, para los píxeles de la imagen $D_{2^{1/2}\sigma}[x, y]$ se consideran también las imágenes adyacentes $D_{\sigma}[x, y]$ y $D_{2\sigma}[x, y]$.



Cada píxel se compara con sus vecinos en la misma imagen de diferencias Gaussianas y en sus adyacentes

Adaptación del material del curso "Foundations of Computer Vision", Prof. Allan Jepson, Universidad de Toronto

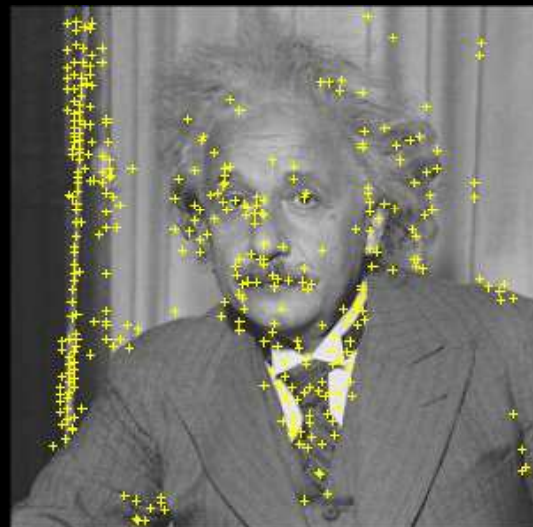
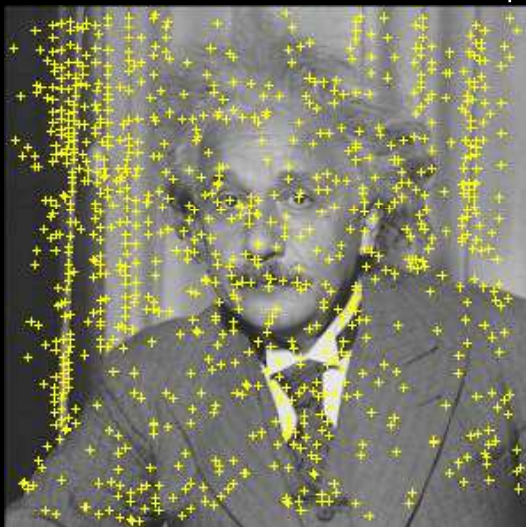
- Aquellos píxeles cuyo valor es un máximo o mínimo local en las imágenes de diferencias de Gaussianas son considerados candidatos a puntos de interés (esquinas). La escala asociada al punto es el valor del múltiplo de σ correspondiente a la imagen diferencia a la que pertenece el punto.



Píxeles candidatos de ser puntos de interés (máximos o mínimos locales en las imágenes de diferencias de Gaussianas)

Material del curso "Foundations of Computer Vision", Prof. Allan Jepson, Universidad de Toronto

- Se eliminan aquellos candidatos con contraste por debajo de un umbral. El contraste viene definido por el valor del propio píxel en la imagen de diferencia de Gaussianas a la que pertenece.



Se eliminan píxeles candidatos poco contrastados

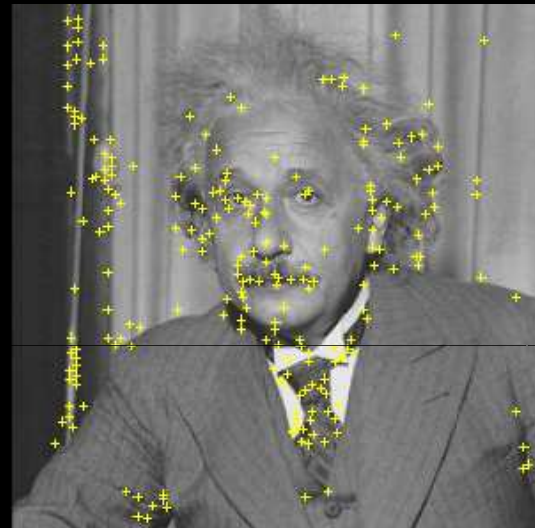
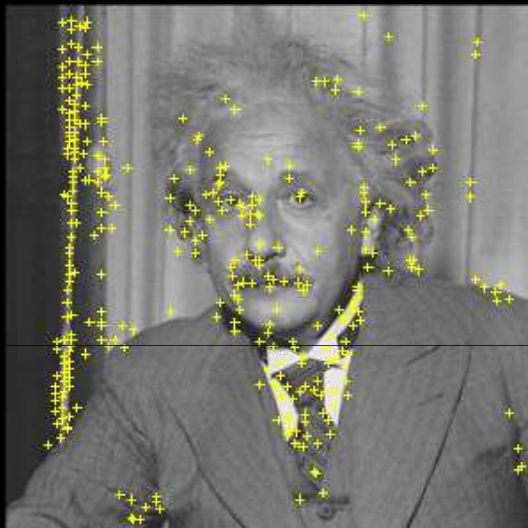
Material del curso "Foundations of Computer Vision", Prof. Allan Jepson, Universidad de Toronto

- Se eliminan los candidatos que forman parte de rectas. Para ello se calcula la matriz Hessiana de la imagen de diferencia de Gaussianas en el píxel candidato. Las derivadas parciales se aproximan mediante diferencias de píxeles vecinos:

$$H = \begin{pmatrix} \frac{\partial^2 D}{\partial x^2} [x, y] & \frac{\partial^2 D}{\partial x \partial y} [x, y] \\ \frac{\partial^2 D}{\partial x \partial y} [x, y] & \frac{\partial^2 D}{\partial y^2} [x, y] \end{pmatrix} \equiv \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix}$$

- Se eliminan los candidatos que cumplen (el cociente de curvaturas principales es elevado):

$$\frac{(D_{xx} + D_{yy})^2}{D_{xx} D_{yy} - D_{xy}^2} \geq \frac{(r+1)^2}{r} \quad (r = 10)$$

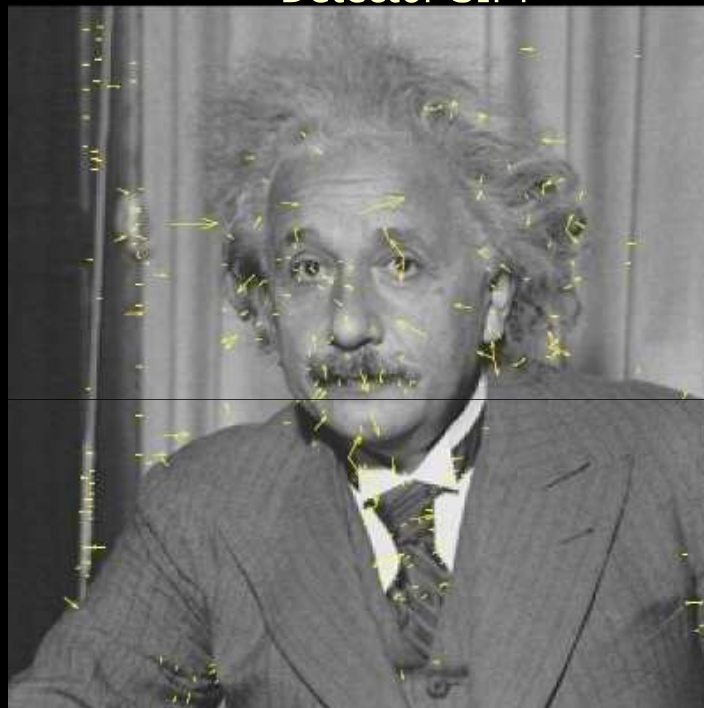


Se eliminan píxeles candidatos que forman parte de rectas

Material del curso "Foundations of Computer Vision", Prof. Allan Jepson,
Universidad de Toronto

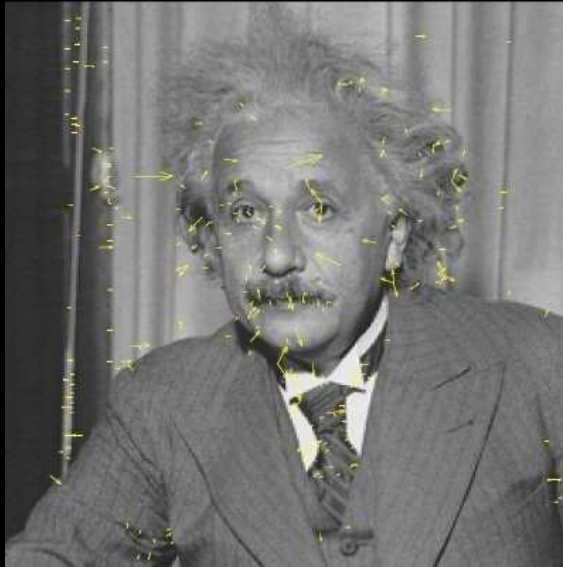
- Para cada uno de los puntos de interés que quedan tras las etapas de filtrado previas, se calcula un histograma de orientaciones de las direcciones del gradiente en sus imágenes Gaussianas correspondientes $L_{k\sigma}[x, y]$, considerando todos los píxeles contenidos en una ventana de 16×16 centrada en el píxel en cuestión.

Cada histograma tiene 36 bins, con cada bin representando 10 grados. De ese histograma se escoge el bin con el valor máximo y aquellos bins con valores por encima del 80% del máximo. Esos bins determinan las orientaciones asociadas al punto de interés.



Puntos de interés con sus orientaciones de gradiente asociadas. El tamaño de la flecha indica la escala asociada al punto.

- Los puntos de interés y sus orientaciones y escalas se preservan en gran medida ante transformaciones de las imágenes.

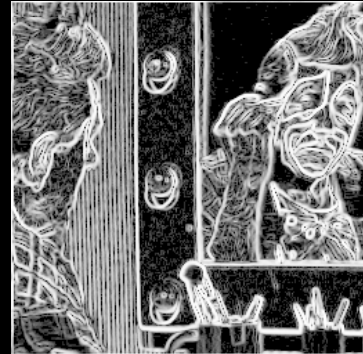
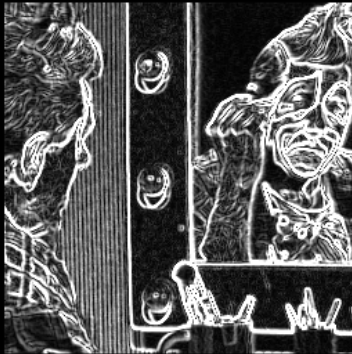


Robustez de las características SIFT ante transformaciones de las imágenes
*Material del curso "Foundations of Computer Vision", Prof. Allan Jepson,
Universidad de Toronto*

- Cada punto de interés tiene también asociado un vector de características que actúa como descriptor de su vecindario y que permite identificar el mismo punto en otras imágenes.
 - Este vector se obtiene a partir de los histogramas de orientaciones del gradiente en un vecindario del píxel en cuestión en la imagen Gaussiana.
- ❁ **Detector PCA-SIFT:**
- PCA-SIFT: A More Distinctive Representation for Local Image Descriptors, Y. Ke and R. Sukthankar, Computer Vision and Pattern Recognition, 2004.
 - Variante del detector SIFT en que el vector de características de cada punto de interés se obtiene aplicando análisis de componentes principales a los vectores gradientes en un vecindario del píxel.
 - Estos descriptores son así más distintivos, más robustos a transformaciones de la imagen y más compactos que la representación original del detector SIFT

- **Introducción**
- **Extracción y caracterización de puntos/esquinas**
 - Detector de Harris & Stephens
 - Detector SUSAN
 - Detector SIFT
- **Extracción de bordes**
 - Transformada de Hough
 - Transformada de Hough generalizada
 - Seguimiento de bordes
 - Contornos dinámicos
- **Extracción de regiones**

- **Características unidimensionales (rectas o curvas) de una imagen que son detectables sin ambigüedad.**
- **Son secuencias contiguas de píxeles en cuyo vecindario confluyen dos o más regiones homogéneas.**
- **Detectores de bordes:**
 - La mayoría (ej.: Roberts, Sobel o Prewitt) generan una imagen en la que los bordes de la imagen original aparecen resaltados con niveles de gris proporcionales a su grado de brusquedad / definición.
 - El detector de Canny genera una imagen binaria donde los bordes de la imagen original aparecen con grosor de un píxel.



Resultado de detectores de Sobel, Roberts y Prewitt.

Material del curso interactivo HIPR2 (R.Fisher, S.Perkins, A.Walker, E.Wolfart, 2004).



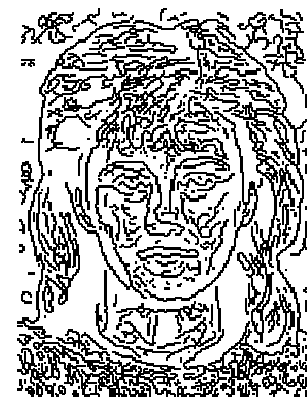
Imagen girl



$m=1, m=9, l=0.3, h=0.8$



$m=0.5, m=7, l=0.3, h=0.8$



$m=1, m=9, l=0.1, h=0.3$

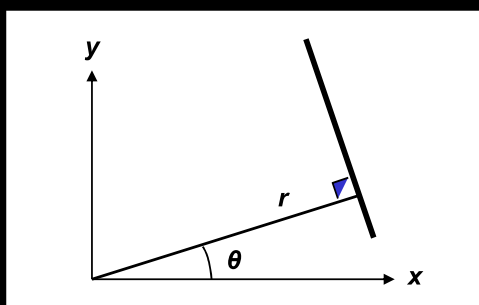
Aplicación del detector de Canny con diferentes parámetros.

Material extraído de CVonline: "The Canny Edge Detector", Andrew Wallace y Sarah Price.

- Los detectores de bordes por sí mismos no identifican / extraen los distintos segmentos que constituyen los bordes de la imagen.
- Los extractores de bordes identifican los segmentos (secuencias de píxeles) que constituyen los bordes de una imagen a partir del resultado generado previamente por un detector de bordes.
- Extractores de bordes más populares:
 - Transformada de Hough
 - Transformada de Hough Generalizada
 - Seguimiento de contornos
 - Contornos activos (snakes)

R.O. Duda and P.E. Hart. Use of the Hough Transform to detect lines and curves. Communications of the ACM, Vol 15, No. pp. 11-15, Enero 1972

- Permite identificar bordes con forma regular, tales como: líneas, circunferencias y elipses.
- Extracción de rectas:
 - Se reparametriza la ecuación explícita de una recta a partir de las coordenadas polares de la perpendicular a ella que pasa por el origen de coordenadas:



$$y = mx + n$$

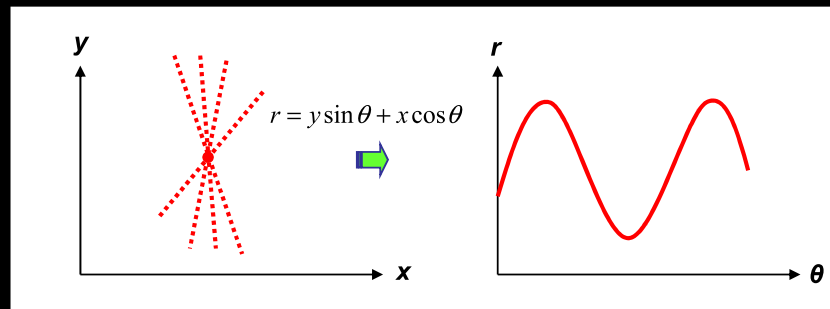
$$m = -\frac{\cos \theta}{\sin \theta}$$

$$n = y - mx = r \sin \theta - \left(-\frac{\cos \theta}{\sin \theta}\right) r \cos \theta = \frac{r(\sin^2 \theta + \cos^2 \theta)}{\sin \theta} = \frac{r}{\sin \theta}$$

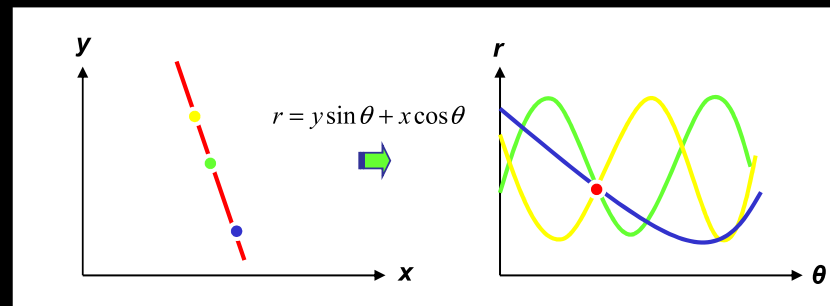
$$y = mx + n = -\frac{\cos \theta}{\sin \theta} x + \frac{r}{\sin \theta} \Rightarrow$$

$$r = y \sin \theta + x \cos \theta$$

- Cada punto (x, y) en el espacio de imagen corresponde a una curva sinusoidal en el espacio transformado de Hough, (θ, r) :

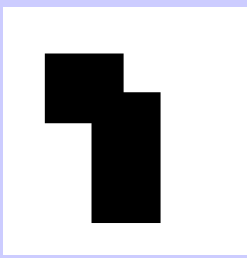


- Una recta en el espacio de imagen corresponde a un punto (intersección de sinusoides) en el espacio transformado de Hough:

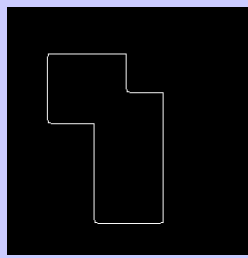


- El espacio 2D de Hough, (θ, r) , se representa discretamente mediante una matriz 2D:
 $H[\theta, r], \theta \in [0, 2\pi), r \in [0, R]$
- Los elementos de esta matriz están inicialmente a cero.
- Sobre la imagen original se aplica un detector de bordes cualquiera. Para cada píxel detectado como borde se 'pinta' en $H[\theta, r]$ la sinusoides correspondiente a dicho píxel, incrementando en uno los valores de la matriz correspondientes a la sinusoides.
- De este modo, cada elemento de la matriz es un contador (acumulador) del número de veces que una curva sinusoidal ha pasado por la posición (θ, r) correspondiente a dicho elemento, es decir, del número de píxeles de la imagen de bordes alineados en una misma recta.
- Los contadores que al final del proceso superan un umbral predefinido identifican los parámetros (θ, r) de los bordes rectos más destacados de la imagen original. Inconveniente: Contadores vecinos suelen superar el umbral, generándose así haces de rectas.
- El proceso de extracción es robusto a la presencia de ruido y a discontinuidades en los bordes.

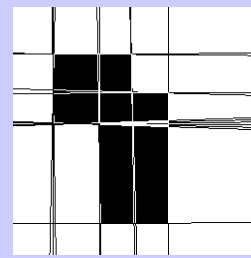
Original



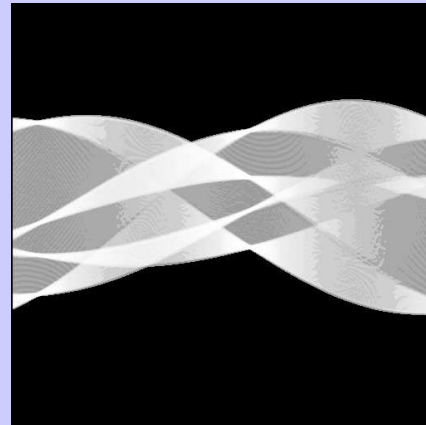
Canny



Rectas detectadas



Espacio de Hough



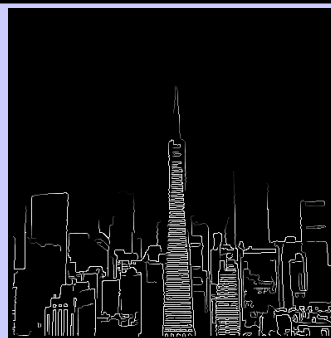
Espacio de Hough (ecualizado)

Material del curso interactivo HIPR2 (R.Fisher, S.Perkins, A.Walker, E.Wolfart, 2004).

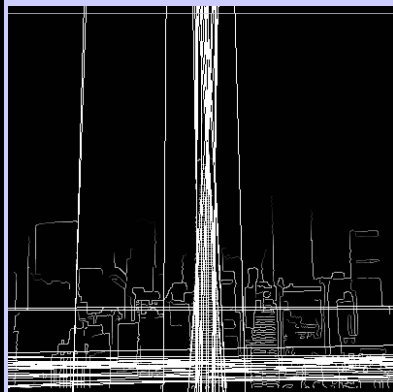
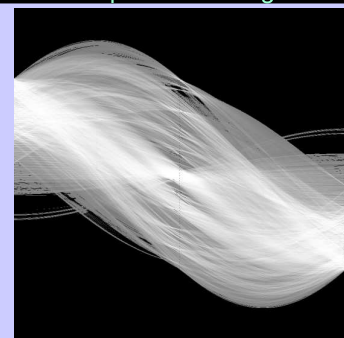
Original



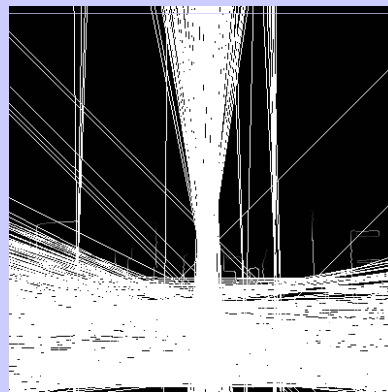
Canny



Espacio de Hough



Rectas detectadas



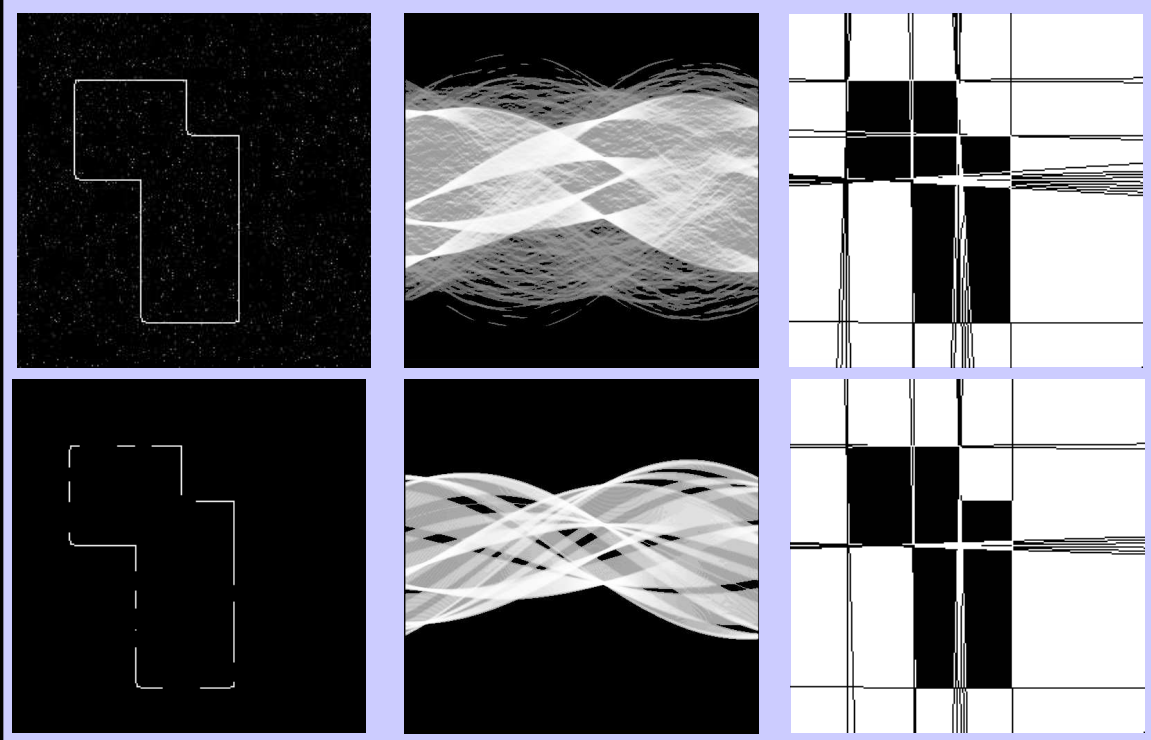
Rectas detectadas con umbral inferior

Material del curso interactivo HIPR2 (R.Fisher, S.Perkins, A.Walker, E.Wolfart, 2004).

Canny (imagen ruidosa)

Espacio de Hough

Rectas detectadas



Canny (imagen con discontinuidades)

Espacio de Hough

Rectas detectadas

Material del curso interactivo HIPR2 (R.Fisher, S.Perkins, A.Walker, E.Wolfart, 2004).

Extracción de circunferencias:

- Ecuación implícita de la circunferencia de centro (a, b) y radio r :

$$(x-a)^2 + (y-b)^2 = r^2$$

- Espacio de Hough de 3 dimensiones:

$$H[a, b, r], \quad a \in [0, A], \quad b \in [0, B], \quad r \in [0, R]$$

- Para cada píxel (x, y) detectado como borde en la imagen original, se itera sobre todas las combinaciones posibles de (a, b) y se calcula para cada una de ellas la r correspondiente aplicando la ecuación implícita. Los contadores en las distintas posiciones seleccionadas, $H[a, b, r]$, son incrementados.
- Los contadores que al final del proceso superan un umbral predefinido identifican los parámetros (a, b, r) de las circunferencias más destacadas de la imagen original.

Imagen original

Imagen binarizada



Imagen de bordes

Circunferencias detectadas

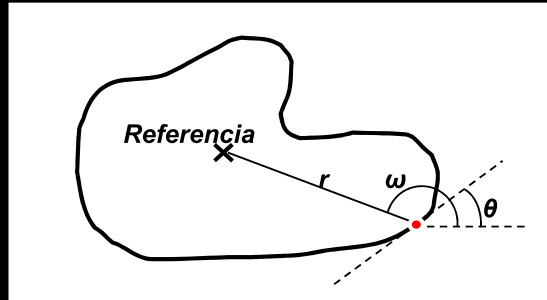
Material del curso "Introduction to Digital Image Processing", H. Rhody, Rochester Institute of Technology, 2005.

D.H. Ballard, *Generalizing the Hough Transform to Detect Arbitrary Shapes*, *Pattern Recognition*, Vol.13, No.2, p.111-122, 1981

- Permite localizar en una imagen bordes de forma arbitraria (no describibles mediante una ecuación).
- Requiere una fase de aprendizaje del modelo de cada borde y una fase de extracción en las imágenes de entrada:
 - Extracción de bordes sujetos a variaciones de posición
 - Extracción de bordes sujetos a variaciones de posición y orientación
 - Extracción de bordes sujetos a variaciones de posición, orientación y escala

• Fase de aprendizaje del modelo del borde:

- Se define un punto de referencia arbitrario. En el caso de bordes cerrados acostumbra a ser el centroide.
- Para cada punto (píxel) del borde (o subconjunto de puntos muestreados del borde) se calcula la orientación de la tangente al borde en ese punto, θ , y las coordenadas polares (ω, r) del punto de referencia respecto del punto de borde analizado:



- Este cálculo se aplica a cada punto considerado del borde, generando una tabla (Tabla R) que almacena todas las tuplas (ω, r) que han sido obtenidas para cada valor distinto de θ

$$({}^1\omega_1, {}^1r_1), \dots, ({}^1\omega_n, {}^1r_n) \quad \dots \quad ({}^m\omega_1, {}^m r_1), \dots, ({}^m\omega_n, {}^m r_n)$$

• Fase de extracción de bordes sujetos a variaciones de posición:

- Dada una imagen en la que se desean localizar instancias del modelo de borde entrenado previamente, se aplica un detector de bordes con grosor de un píxel (ej.: Canny).
- Se considera como espacio discretizado de Hough una matriz con el mismo tamaño que la imagen dada, en la que cada elemento $H[x, y]$ actúa como un contador (acumulador).
- Para cada píxel $\psi[x_i, y_i]$ perteneciente a los bordes detectados en el primer paso de la fase de extracción, se estima la orientación de su recta tangente. Esta tangente es perpendicular a la dirección del gradiente del borde en ese píxel. El gradiente en cada píxel se suele conocer como parte del propio proceso de detección de bordes.
- Se extraen de la Tabla R todas las tuplas correspondientes a $\theta_i : ({}^i\omega_1, {}^i r_1), \dots, ({}^i\omega_n, {}^i r_n)$

- Se calculan las posiciones candidatas del punto de referencia aplicando las coordenadas polares de cada tupla a la posición (x_i, y_i) del píxel analizado.

$$\begin{matrix} {}^1x_i = x_i + {}^1r_i \cos({}^1\omega_i) & \dots & {}^n x_i = x_i + {}^n r_i \cos({}^n \omega_i) \\ {}^1y_i = y_i + {}^1r_i \sin({}^1\omega_i) & & {}^n y_i = y_i + {}^n r_i \sin({}^n \omega_i) \end{matrix}$$

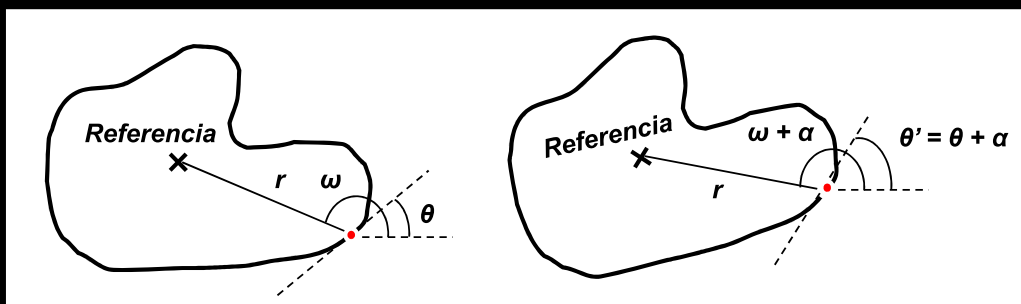
- Se incrementan los contadores del espacio de Hough en las posiciones candidatas de los puntos de referencia:

$$H[{}^1x_i, {}^1y_i] = H[{}^1x_i, {}^1y_i] + 1 \quad \dots \quad H[{}^n x_i, {}^n y_i] = H[{}^n x_i, {}^n y_i] + 1$$

- Cada contador que al final del proceso supera un umbral predefinido identifica las coordenadas (x, y) en la imagen original donde está situada la posición de referencia del borde buscado y, por tanto, la localización del borde en si.

• Fase de extracción de bordes sujetos a variaciones de posición y orientación:

- El proceso descrito anteriormente sólo permite localizar bordes con el mismo tamaño y orientación que el modelo (sólo permite translaciones del modelo en la imagen).
- El borde del modelo puede aparecer en la imagen sujeto a una rotación arbitraria α :



- El espacio de Hough discreto es una matriz 3D en la que cada elemento es un contador (acumulador). Las dos primeras dimensiones representan la posición del centro de referencia y la tercera el ángulo del modelo en la imagen:

$$H[x, y, \alpha], \quad x \in [0, x_{MAX}], \quad y \in [0, y_{MAX}], \quad \alpha \in [0, \alpha_{MAX}]$$

- Para cada píxel $\psi[x, y_i]$ perteneciente a los bordes detectados en el primer paso de la fase de extracción, se estima la orientación θ_i de su recta tangente.
- Para cada una de las posibles orientaciones del modelo en la imagen, $\alpha \in [0, \alpha_{MAX}]$, se extraen de la Tabla R las tuplas correspondientes al ángulo $\theta_i - \alpha$: $({}^1\omega_1, {}^1r_1), \dots, ({}^n\omega_n, {}^nr_n)$
- Se calculan las posiciones candidatas del punto de referencia aplicando las coordenadas polares de cada tupla a la posición (x_i, y_i) del píxel analizado, considerando que el modelo está rotado un ángulo α .

$$\begin{matrix} {}^1x_i = x_i + {}^1r_1 \cos({}^1\omega_1 + \alpha) & \dots & {}^nx_i = x_i + {}^nr_n \cos({}^n\omega_n + \alpha) \\ {}^1y_i = y_i + {}^1r_1 \sin({}^1\omega_1 + \alpha) & & {}^ny_i = y_i + {}^nr_n \sin({}^n\omega_n + \alpha) \end{matrix}$$

- Se incrementan los contadores del espacio de Hough en las posiciones candidatas de los puntos de referencia y el ángulo considerado:

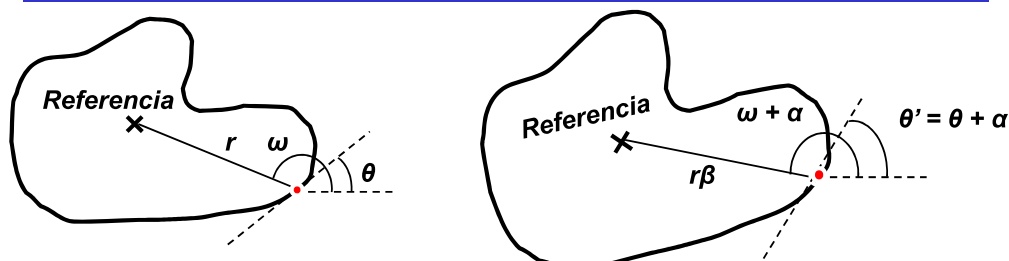
$$H[{}^1x_i, {}^1y_i, \alpha] = H[{}^1x_i, {}^1y_i, \alpha] + 1 \quad \dots \quad H[{}^nx_i, {}^ny_i, \alpha] = H[{}^nx_i, {}^ny_i, \alpha] + 1$$

- Cada contador $H[x, y, \alpha]$ que al final del proceso supera un umbral predefinido identifica las coordenadas (x, y) en la imagen original de la posición de referencia del borde buscado, así como su orientación α .

• Fase de extracción de bordes sujetos a variaciones de posición, orientación y escala:

- El borde del modelo puede aparecer en la imagen sujeto a una rotación α y escala β arbitrarias.
- Se considera como espacio de Hough discreto una matriz 4D en la que cada elemento actúa como un contador (acumulador). Las dos primeras dimensiones representan la posición del centro de referencia, la tercera el ángulo del modelo en la imagen y la cuarta su factor de escala:

$$H[x, y, \alpha, \beta], \quad x \in [0, x_{MAX}], \quad y \in [0, y_{MAX}], \quad \alpha \in [0, \alpha_{MAX}], \quad \beta \in [\beta_{MIN}, \beta_{MAX}]$$

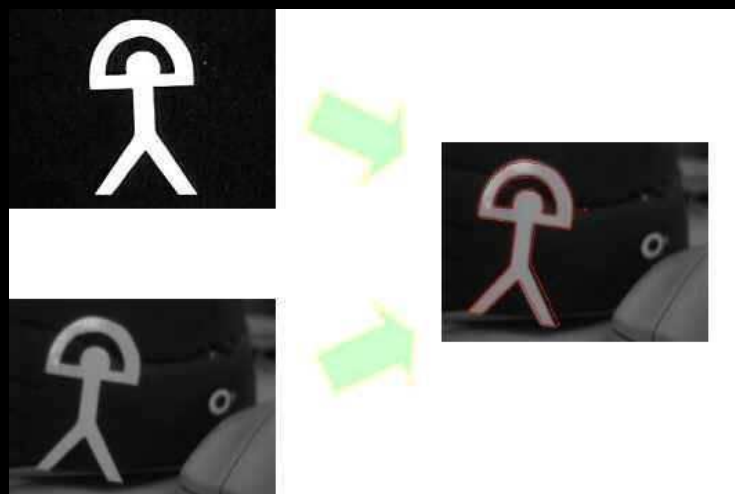


- Para cada píxel $\psi[x_i, y_i]$ perteneciente a los bordes detectados en el primer paso de la fase de extracción, se estima la orientación θ_i de su recta tangente.
- Para cada una de las posibles orientaciones del modelo en la imagen, $\alpha \in [0, \alpha_{MAX}]$, se extraen de la Tabla R las tuplas correspondientes al ángulo $\theta_i - \alpha : ({}^i\omega_1, {}^i r_1), \dots, ({}^i\omega_n, {}^i r_n)$.
- Para cada uno de los posibles factores de escala, $\beta \in [\beta_{MIN}, \beta_{MAX}]$, se calculan las posiciones candidatas del punto de referencia aplicando las coordenadas polares de cada tupla a la posición (x_i, y_i) del píxel analizado, considerando que el modelo está rotado un ángulo α y escalado un factor β .

$$\begin{aligned} {}^1x_i &= x_i + {}^1r_1 \beta \cos({}^1\omega_1 + \alpha) & \dots & & {}^n x_i &= x_i + {}^n r_n \beta \cos({}^n \omega_n + \alpha) \\ {}^1y_i &= y_i + {}^1r_1 \beta \sin({}^1\omega_1 + \alpha) & & & {}^n y_i &= y_i + {}^n r_n \beta \sin({}^n \omega_n + \alpha) \end{aligned}$$

- Se incrementan los contadores del espacio de Hough en las posiciones candidatas de los puntos de referencia y en el ángulo y escala considerados:

$$H[{}^1x_i, {}^1y_i, \alpha, \beta] = H[{}^1x_i, {}^1y_i, \alpha, \beta] + 1 \quad \dots \quad H[{}^n x_i, {}^n y_i, \alpha, \beta] = H[{}^n x_i, {}^n y_i, \alpha, \beta] + 1$$
- Cada contador $H[x, y, \alpha, \beta]$ que al final del proceso supera un umbral predefinido identifica las coordenadas (x, y) en la imagen original de la posición de referencia del borde buscado, así como su orientación α y escala β .



Ejemplo de aplicación de la Transformada de Hough Generalizada con variaciones de posición, orientación y escala



Ejemplos de aplicación de la Transformada de Hough Generalizada con variaciones de posición, orientación y escala

Imágenes generadas por el Grupo de Procesamiento de Vídeo e Imágenes del Departamento de Arquitectura de Computadores de la Universidad de Málaga

❁ Inconveniente:

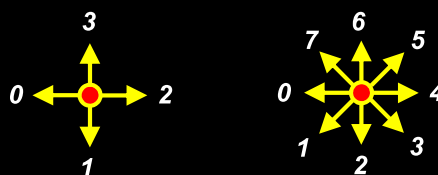
- La transformada de Hough generalizada es muy costosa computacionalmente cuando el número de dimensiones del espacio de Hough es mayor que dos y la resolución de la discretización es elevada.
- Sin embargo, el algoritmo es altamente paralelizable y diversos autores han desarrollado implementaciones por hardware que permiten la extracción de contornos de forma arbitraria en tiempo real.

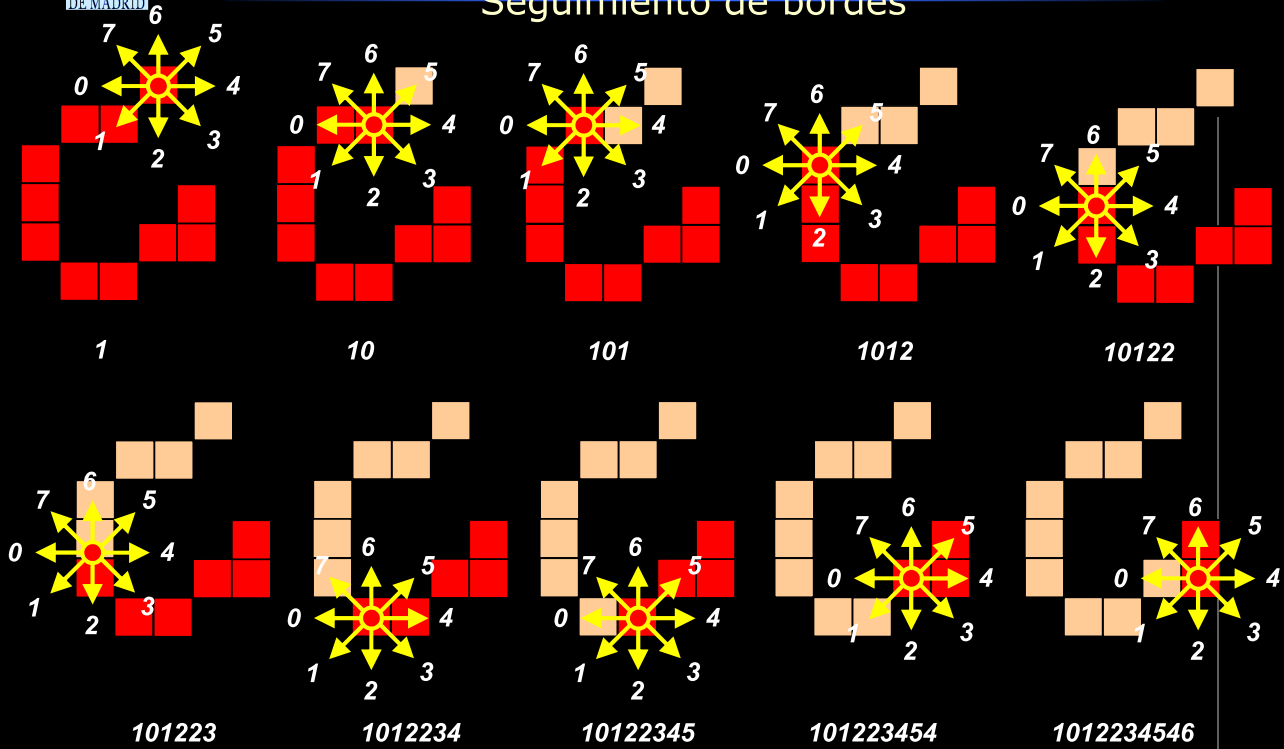
❁ Ventaja:

- El proceso de votación basado en información local de los contornos permite el reconocimiento de bordes en presencia de ruido y oclusiones.

- Permite extraer las secuencias de píxeles de los diferentes bordes presentes en el resultado de un detector de bordes de grosor de un píxel (ej.: Canny), e identificar formas elementales, por ejemplo rectas.
- Algoritmo de extracción de bordes:
 1. Seleccionar el primer píxel detectado como borde que aparezca al recorrer las filas de la imagen, de izquierda a derecha, desde la primera.
 2. Terminar la extracción si no aparece ningún píxel de borde.
 3. Reseguir el borde desde el píxel seleccionado codificando la secuencia de píxeles visitados como un código cadena y suprimiendo dichos píxeles de la imagen de bordes para que no vuelvan a ser visitados. El seguimiento del borde finaliza cuando se llega a una interrupción del mismo.
 4. Continuar desde el paso 1.

- Seguimiento de un borde: Generación del código cadena de Freeman
 - Genera un valor numérico que codifica la forma de la cadena de píxeles que conforman una curva abierta o cerrada de un píxel de grosor.
 - Se consideran 4 u 8 direcciones de desplazamiento y se codifica cada una con un dígito decimal, dando valores crecientes a las direcciones en sentido antihorario.

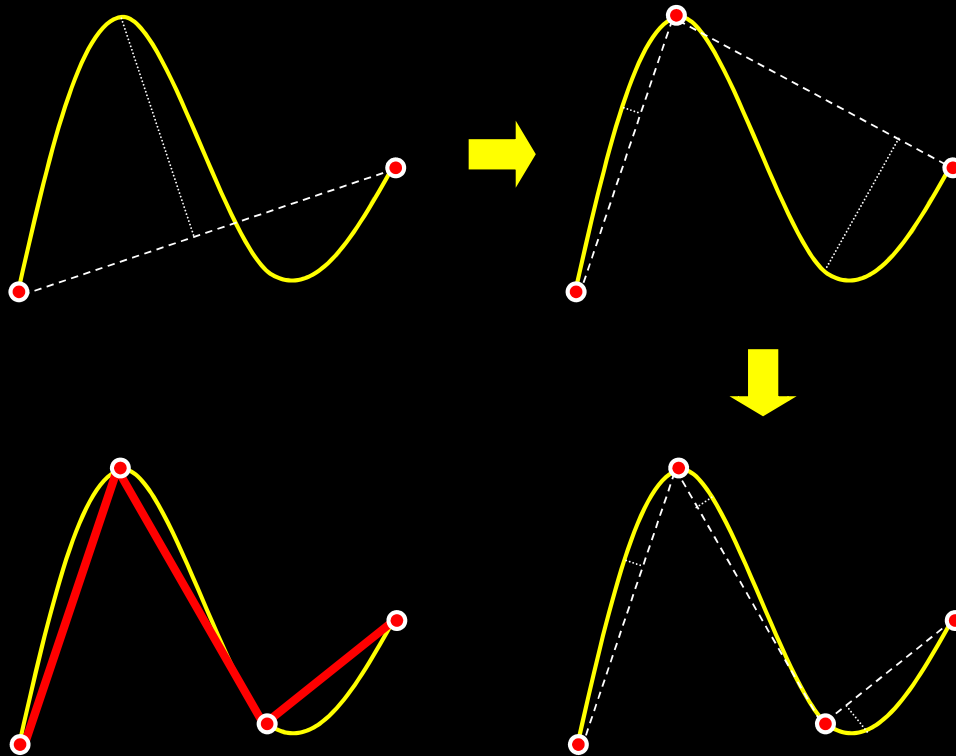




Ejemplo de generación de **código cadena de Freeman** con conectividad 8.
El siguiente píxel es el vecino que está en la primera dirección a partir de la 0.

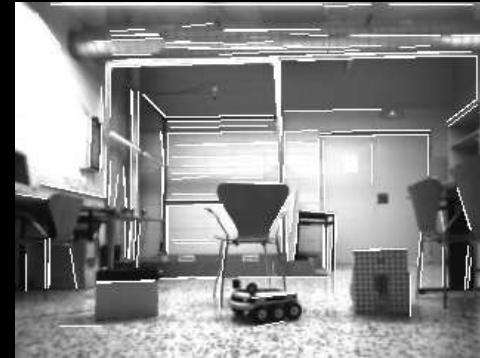
• Aproximación de bordes

- Un código cadena representa de forma numérica la curva definida por todos los píxeles que constituyen un borde de la imagen.
- Una curva de forma arbitraria puede aproximarse por una secuencia de tramos rectos.
- Se considera el segmento recto que une dos puntos no adyacentes de la curva, por ejemplo sus extremos.
- La curva se parte en dos secciones por el punto que está a mayor distancia del segmento.
- Este procedimiento se aplica recursivamente a cada una de las secciones hasta que no se produce ninguna nueva partición por ser la máxima distancia al segmento menor que un umbral predefinido que determina el error de aproximación.



Ejemplo de aproximación de una curva por subdivisiones sucesivas.

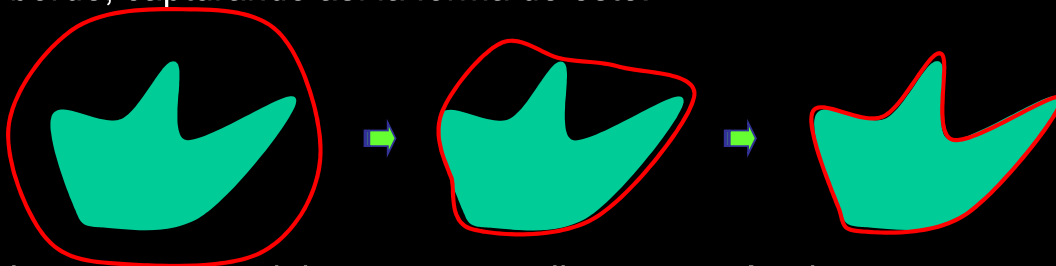
- Las secciones rectas obtenidas tras el procedimiento de aproximación anterior pueden interpretarse como los bordes rectos de la imagen.
- Este procedimiento es una alternativa eficiente a la Transformada de Hough para rectas.



Ejemplos de bordes rectos extraídos mediante seguimiento y aproximación de contornos.

M. Kass, A. Witkin, D. Terzopoulos, *Snakes: Active Contour Models*, *International Journal of Computer Vision*, 321-331, 1988.

- Contornos (curvas) que se adaptan automáticamente a bordes de la imagen.
- Se parte de una curva cerrada de forma simple que envuelve el borde que se desea delimitar y se deja evolucionar iterativamente la forma de la curva hasta que se estabiliza ajustándose a dicho borde, capturando así la forma de éste:



- La curva se modela como una spline que está sujeta a una energía interna que tiende a que la curva se contraiga y suavice, y una energía externa que tiende a atraer la curva hacia los bordes u otras características que se desean extraer de la imagen.
- La curva (snake) se estabiliza cuando ambas energías se anulan entre sí.

Derivación del extractor:

- La spline se modela como una curva 2D parametrizada: $\mathbf{v}(s) = (x(s), y(s)) \in \mathcal{R}^2, s \in [0, 1] \subset \mathcal{R}$

- Si la curva es cerrada, su punto inicial y final coinciden: $\mathbf{v}(0) = \mathbf{v}(1)$

- Se define la energía de la snake como la suma de una energía interna y una externa:

$$E_{snake}(\mathbf{v}(s)) = E_{int}(\mathbf{v}(s)) + E_{ext}(\mathbf{v}(s))$$

- La energía interna es una media ponderada entre un primer término proporcional a la deformación de la curva respecto de una recta, y un segundo término proporcional a la curvatura de la curva:

$$E_{int}(\mathbf{v}(s)) = \frac{1}{2} \left(\alpha(s) \left| \frac{d\mathbf{v}}{ds} \right|^2 + \beta(s) \left| \frac{d^2\mathbf{v}}{ds^2} \right|^2 \right), \quad \alpha(s) + \beta(s) = 1$$

- La energía externa viene dada por una función que mide lo lejos que está la snake de la característica de la imagen que se desea extraer: $E_{ext}(\mathbf{v}(s)) = f(\mathbf{v}(s))$

- Si el objetivo es extraer el contorno de un objeto contenido en una imagen $\psi[x, y]$, la función de energía externa se define como: $f(\mathbf{v}(s)) = -(G_\sigma * \nabla \psi)[x(s), y(s)]$

- La imagen $\nabla \psi[x, y]$ es la imagen de gradiente, que puede obtenerse en la práctica aplicando un detector de bordes (ej.: Sobel).

- La imagen $(G_\sigma * \nabla \psi)[x, y]$ es la imagen de gradiente suavizada con un filtro Gaussiano de desviación σ . Este filtrado tiene por objetivo que la snake sea atraída por bordes significativos de la imagen, siendo así más inmune al ruido presente en la imagen.

- El signo negativo en la formulación de $f(\mathbf{v}(s))$ hace que la energía externa disminuya cuanto más se aproxima la curva de la snake a los gradientes altos de la imagen.

- La snake se deforma iterativamente con el objeto de **minimizar la suma de su energía interna y externa a lo largo de la curva:**

$$\min \int_0^1 (E_{int}(\mathbf{v}(s)) + E_{ext}(\mathbf{v}(s))) ds$$

- Como la suma de energías depende de los valores que presenta una imagen en las posiciones de la curva, la minimización no puede realizarse analíticamente.

- Para minimizar la suma de energías mediante un método numérico hay que discretizar la curva de la snake, convirtiéndola en una secuencia de puntos equiespaciados una distancia h :

$$V = \{ \mathbf{v}_i = (x_i, y_i) \in \mathcal{R}^2, \forall i \in [1, n] \}, \quad \mathbf{v}_1 = \mathbf{v}_n$$

- La energía interna en cada punto se expresa en forma discreta aproximando las derivadas de la curva mediante diferencias finitas entre puntos vecinos:

$$E_{int}(i) = \frac{1}{2} \left(\alpha_i \left| \frac{\mathbf{v}_i - \mathbf{v}_{i-1}}{h} \right|^2 + \beta_i \left| \frac{\mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1}}{h^2} \right|^2 \right), \quad \alpha_i + \beta_i = 1$$

- La energía externa en cada punto se obtiene evaluando la función de imagen en dicho punto. En el caso de que la snake sea atraída por los bordes cercanos de una imagen, la energía externa se define como:

$$E_{ext}(i) = f(\mathbf{v}_i) = -(G_\sigma * \nabla \psi)[x_i, y_i]$$

- La energía total de la snake es: $E_{snake} = \sum_{i=1}^n E_{int}(i) + E_{ext}(i)$

- Para que la snake minimice su energía total, debe satisfacer la siguiente ecuación de Euler-Lagrange:

$$\alpha_i (\mathbf{v}_i - \mathbf{v}_{i-1}) - \alpha_{i+1} (\mathbf{v}_{i+1} - \mathbf{v}_i) + \beta_{i-1} (\mathbf{v}_i - 2\mathbf{v}_{i-1} + \mathbf{v}_{i-2}) - 2\beta_i (\mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1}) + \beta_{i+1} (\mathbf{v}_{i+2} - 2\mathbf{v}_{i+1} + \mathbf{v}_i) + (f_x(\mathbf{v}_i), f_y(\mathbf{v}_i)) = 0$$

$$f_x(\mathbf{v}_i) = \left. \frac{\partial f}{\partial x} \right|_{\mathbf{v}_i} \quad f_y(\mathbf{v}_i) = \left. \frac{\partial f}{\partial y} \right|_{\mathbf{v}_i}$$

- Las derivadas de la energía externa, $f_x(\mathbf{v}_i)$ y $f_y(\mathbf{v}_i)$ se pueden aproximar mediante diferencias finitas entre píxeles vecinos:

$$f_x(\mathbf{v}_i) \approx -(G_\sigma * \nabla \psi)[x_i + 1, y_i] + (G_\sigma * \nabla \psi)[x_i, y_i]$$

$$f_y(\mathbf{v}_i) \approx -(G_\sigma * \nabla \psi)[x_i, y_i + 1] + (G_\sigma * \nabla \psi)[x_i, y_i]$$

- La ecuación de Euler-Lagrange anterior puede expresarse en forma matricial como:

$$\begin{array}{l} \mathbf{Ax} + \mathbf{f}_x(\mathbf{x}, \mathbf{y}) = 0 \quad \mathbf{x} = (x_1, \dots, x_n)^T \quad \mathbf{f}_x(\mathbf{x}, \mathbf{y}) = (f_x(x_1, y_1), \dots, f_x(x_n, y_n))^T \\ \mathbf{Ay} + \mathbf{f}_y(\mathbf{x}, \mathbf{y}) = 0 \quad \mathbf{y} = (y_1, \dots, y_n)^T \quad \mathbf{f}_y(\mathbf{x}, \mathbf{y}) = (f_y(x_1, y_1), \dots, f_y(x_n, y_n))^T \end{array}$$

- La matriz \mathbf{A} es pentadiagonal y está formada por combinaciones lineales de los coeficientes α_i y β_i
- Las ecuaciones anteriores no pueden resolverse analíticamente por depender de una imagen: han de resolverse iterativamente.
- La posición de la snake en un instante de tiempo dado es $\mathbf{v}_t = (\mathbf{x}_t, \mathbf{y}_t)$
- Las ecuaciones de Euler-Lagrange anteriores representan las fuerzas que actúan sobre la snake en un instante dado:

$$\mathbf{Ax}_t + \mathbf{f}_x(\mathbf{x}_t, \mathbf{y}_t) = \mathbf{F}_x(t) \quad \mathbf{Ay}_t + \mathbf{f}_y(\mathbf{x}_t, \mathbf{y}_t) = \mathbf{F}_y(t)$$

- Cuando la snake se estabiliza, su variación de posición en un diferencial de tiempo ha de ser tal que compense las fuerzas que actúan sobre ella:

$$\frac{\mathbf{x}_t - \mathbf{x}_{t-1}}{dt} = -\mathbf{F}_x(t) = -\mathbf{Ax}_t - \mathbf{f}_x(\mathbf{x}_t, \mathbf{y}_t) \quad \frac{\mathbf{y}_t - \mathbf{y}_{t-1}}{dt} = -\mathbf{F}_y(t) = -\mathbf{Ay}_t - \mathbf{f}_y(\mathbf{x}_t, \mathbf{y}_t)$$

- El objetivo es determinar las nuevas posiciones $\mathbf{v}_t = (\mathbf{x}_t, \mathbf{y}_t)$ a partir de las anteriores.
- Haciendo la aproximación de que la energía externa en la nueva posición será similar a la de la posición anterior, que ya se tiene calculada: $\mathbf{f}_x(\mathbf{x}_t, \mathbf{y}_t) \cong \mathbf{f}_x(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})$, queda:

$$\frac{\mathbf{x}_t - \mathbf{x}_{t-1}}{dt} = -\mathbf{Ax}_t - \mathbf{f}_x(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) \quad \frac{\mathbf{y}_t - \mathbf{y}_{t-1}}{dt} = -\mathbf{Ay}_t - \mathbf{f}_y(\mathbf{x}_{t-1}, \mathbf{y}_{t-1})$$

- Despejando se obtienen las nuevas posiciones de la snake a partir de las anteriores:

$$\mathbf{x}_t = \left(\mathbf{A} + \frac{1}{dt} \mathbf{I} \right)^{-1} \left(\frac{1}{dt} \mathbf{x}_{t-1} - \mathbf{f}_x(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) \right) \quad \mathbf{y}_t = \left(\mathbf{A} + \frac{1}{dt} \mathbf{I} \right)^{-1} \left(\frac{1}{dt} \mathbf{y}_{t-1} - \mathbf{f}_y(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) \right)$$

• Procedimiento resumido:

- Se define una curva poligonal con n puntos equiespaciados, $V = \{(x_1, y_1), \dots, (x_n, y_n)\}$, situada próxima al borde que se desea extraer en la imagen dada $\psi[x, y]$.
- Si la curva es cerrada se fuerza $(x_1, y_1) = (x_n, y_n)$
- Se obtiene la imagen de gradiente de la imagen dada aplicando un detector de bordes (ej.: Sobel, Roberts, Prewitt): $\nabla\psi[x, y]$.
- Se suaviza la imagen de gradiente anterior convolucionándola con un filtro Gaussiano de desviación σ : $(G_\sigma * \nabla\psi)[x, y]$
- Se define la energía externa de la snake como la función $f(x, y) = -(G_\sigma * \nabla\psi)[x, y]$
- Se estiman las derivadas parciales de la función de energía externa:

$$f_x(x, y) \approx -(G_\sigma * \nabla\psi)[x+1, y] + (G_\sigma * \nabla\psi)[x, y] \quad f_y(x, y) \approx -(G_\sigma * \nabla\psi)[x, y+1] + (G_\sigma * \nabla\psi)[x, y]$$

- Para cada vértice de la snake, (x_i, y_i) , se definen dos porcentajes constantes, α_i y β_i . El primero determina el grado de elasticidad que tendrá la snake en ese vértice, y el segundo el grado de rigidez. Variando la relación entre esos coeficientes se modifica la manera en que la snake se ajusta al borde. Si la rigidez es muy baja, la snake puede ajustarse a esquinas (discontinuidades de la segunda derivada). Sin embargo, en ese caso también le afectará más el posible ruido del borde.
- A partir de los coeficientes anteriores se define la matriz **A**
- En tiempo 0, las posiciones de la snake son las iniciales: $\mathbf{x}_0 = (x_1, \dots, x_n)^T$, $\mathbf{y}_0 = (y_1, \dots, y_n)^T$
- Se evalúan las funciones de energía externa a partir de las posiciones actuales:

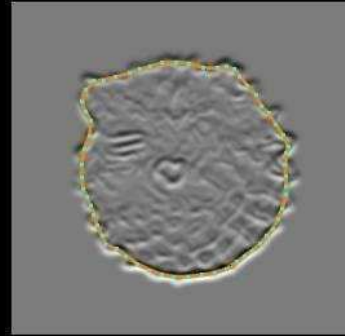
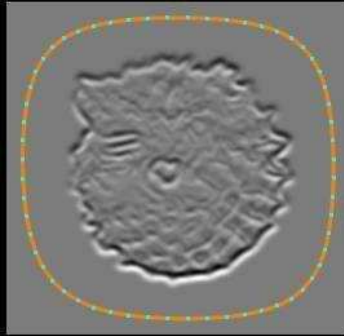
$$\mathbf{f}_x(\mathbf{x}_0, \mathbf{y}_0) = (f_x(x_1, y_1), \dots, f_x(x_n, y_n))^T \quad \mathbf{f}_y(\mathbf{x}_0, \mathbf{y}_0) = (f_y(x_1, y_1), \dots, f_y(x_n, y_n))^T$$
- Se calculan las nuevas posiciones, \mathbf{x}_1 e \mathbf{y}_1 , a partir de: **A, $\mathbf{x}_0, \mathbf{y}_0, \mathbf{f}_x(\mathbf{x}_0, \mathbf{y}_0), \mathbf{f}_y(\mathbf{x}_0, \mathbf{y}_0)$**
- Si las nuevas posiciones varían significativamente respecto de las anteriores, se repite el proceso desde el paso de evaluación de las funciones de energía, substituyendo $(\mathbf{x}_0, \mathbf{y}_0)$ por $(\mathbf{x}_1, \mathbf{y}_1)$ y así sucesivamente hasta que la snake se estabilice.



59 vértices



57 iteraciones



Ejemplo de extracción de bordes complejos con snakes.

Material del curso "Introduction to Scientific Computing", Svetlana Lazebnik, Universidad de Illinois.



59 vértices



97 iteraciones

Ejemplo de extracción de bordes complejos con snakes.

Material del curso "Introduction to Scientific Computing", Svetlana Lazebnik, Universidad de Illinois.



59 vértices



292 iteraciones

Ejemplo de extracción de bordes complejos con snakes.

Material del curso "Introduction to Scientific Computing", Svetlana Lazebnik, Universidad de Illinois.

- Introducción
- Extracción y caracterización de puntos/esquinas
 - Detector de Harris & Stephens
 - Detector SUSAN
 - Detector SIFT
- Extracción de bordes
 - Transformada de Hough
 - Transformada de Hough generalizada
 - Seguimiento de bordes
 - Contornos dinámicos
- Extracción de regiones
 - Umbralización
 - Etiquetado de componente conexas
 - Crecimiento/partición de regiones

- Características bidimensionales (áreas cerradas) de una imagen que son detectables sin ambigüedad.
- Son subconjuntos conexos de píxeles que presentan uniformidad respecto de alguna característica visual (luminosidad, color, textura, ...).
- Suelen corresponder a los diferentes objetos presentes en una imagen o a sus partes constituyentes.
- Extractores de regiones más populares:
 - Umbralización
 - Etiquetado de componentes conexas
 - Crecimiento / partición de regiones

- Generación de imágenes binarias (binarización) a partir de imágenes de nivel de gris, de forma que todos los píxeles cuyos valores están por debajo de un umbral pasan a ser 0 y los que están por encima pasan a 1.
- El objetivo es que los píxeles a 1 representen los objetos presentes en la imagen original, mientras que los que valen 0 representen el fondo de la escena. Esto sólo se consigue en imágenes relativamente sencillas.



Imagen original

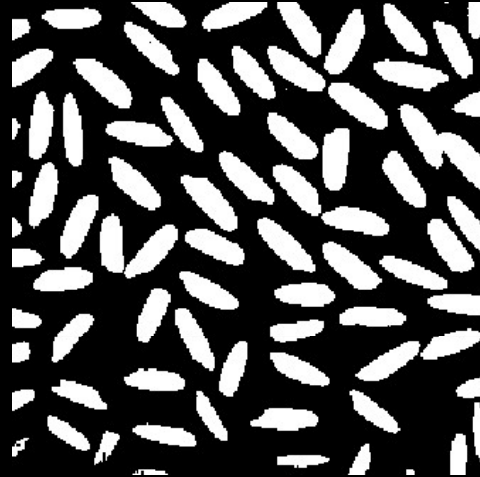


Imagen binarizada con umbral global

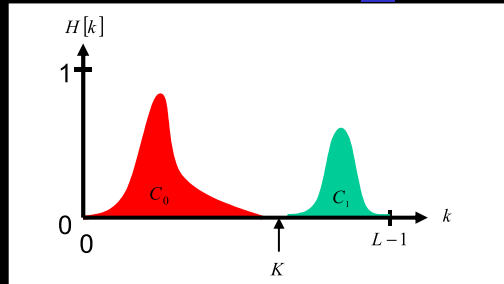
Material del curso "Visión Artificial", Universidad Nacional de Quilmes (Argentina), 2005.

- Los diversos métodos de umbralización se diferencian por la manera de calcular el valor del umbral:
 - Umbralización global: Un único umbral para toda la imagen.
 - Umbralización variable: Distintos umbrales para distintas zonas de la imagen.
 - **Umbralización local o regional:** Umbralización variable en que el umbral aplicado a cada píxel se calcula a partir de los valores de los píxeles en un vecindario del primero.
 - **Umbralización dinámica o adaptativa:** Umbralización variable en que el valor del umbral depende de las propias coordenadas de los píxeles.
- Estas técnicas pueden generalizarse a umbralización múltiple con el objetivo de calcular varios umbrales y, con ellos, diferentes clases de objetos.
- El proceso de umbralización tiene enorme aplicación como etapa previa en algoritmos de reconocimiento e inspección, especialmente en el ámbito industrial.

Umbralización global óptima: Método de Otsu

N. Otsu, "A threshold selection method from gray level histograms", IEEE Trans. Systems, Man and Cybernetics, vol.9, 62-66, 1979

- Se asume que la imagen de entrada, $\psi[x, y]$, es de L niveles de gris y tiene un histograma bimodal $H[k]$, $k \in [0, L-1]$.
- El objetivo del método es calcular el umbral óptimo K que maximice la separación entre las clases correspondientes a los píxeles de fondo, C_0 , y a los de objeto, C_1 .



- La entrada k del histograma, $H[k]$, indica la frecuencia relativa de píxeles con nivel de gris k en la imagen.
- Fijado un umbral arbitrario k , las probabilidades de que un píxel pertenezca respectivamente a la clase del fondo o a la de objetos son:

$$P_0(k) = \sum_{i=0}^k H[i] \quad P_1(k) = \sum_{i=k+1}^{L-1} H[i]$$

- Los valores medios de intensidad de los píxeles pertenecientes a la clase de fondo y de objetos son respectivamente:

$$m_0(k) = \frac{1}{P_0(k)} \sum_{i=0}^k i H[i] \quad m_1(k) = \frac{1}{P_1(k)} \sum_{i=k+1}^{L-1} i H[i]$$

- El valor medio de intensidad de toda la imagen es:

$$m_G = \sum_{i=0}^{L-1} i H[i] = P_0(k) m_0(k) + P_1(k) m_1(k)$$

- La varianza inter-clase (separabilidad de las clases) se define como:

$$\sigma_{01}^2(k) = P_0(k) (m_0(k) - m_G)^2 + P_1(k) (m_1(k) - m_G)^2$$

- Y aplicando la definición de m_G :

$$\sigma_{01}^2(k) = 2P_0(k)P_1(k)(m_0(k) - m_1(k))^2$$

- Por tanto, la separabilidad entre las clases depende de la separación entre sus medias de intensidad respectivas.

- El umbral óptimo K es el nivel de gris que maximiza la varianza inter-clase:

$$K = \arg \max_{\forall k \in [0, L-1]} \sigma_{01}^2(k)$$

- Para obtenerlo se calcula $\sigma_{01}^2(k)$ para todos los niveles de gris, y se escoge aquel nivel para el que dicha varianza es máxima.

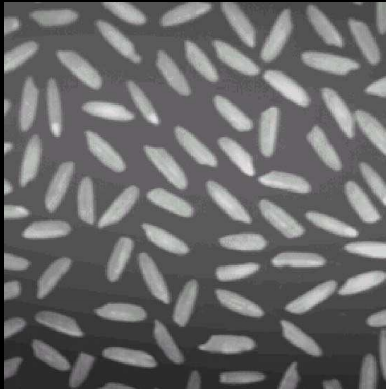
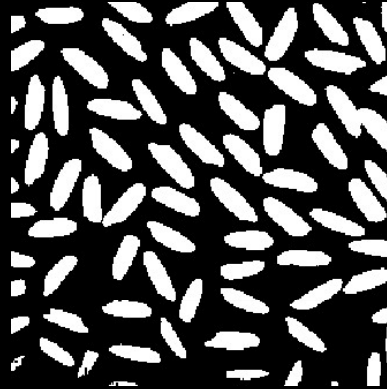


Imagen original



Umbral global
arbitrario



Umbral global
óptimo (Otsu)

Material del curso "Visión Artificial", Universidad Nacional de Quilmes (Argentina), 2005.

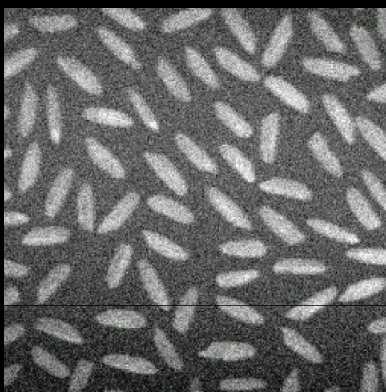
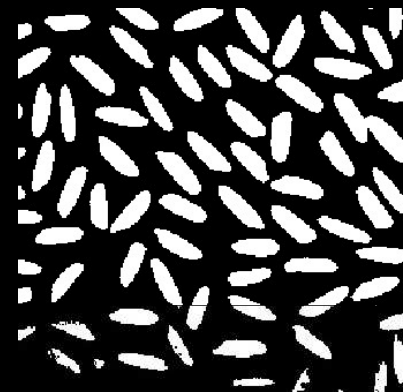


Imagen original
con ruido Gaussiano



Mismo umbral
global arbitrario



Umbral global
óptimo (Otsu)

Material del curso "Visión Artificial", Universidad Nacional de Quilmes (Argentina), 2005.

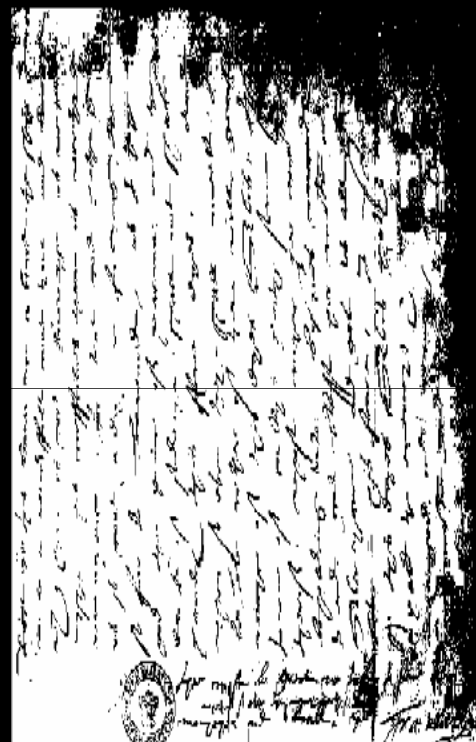
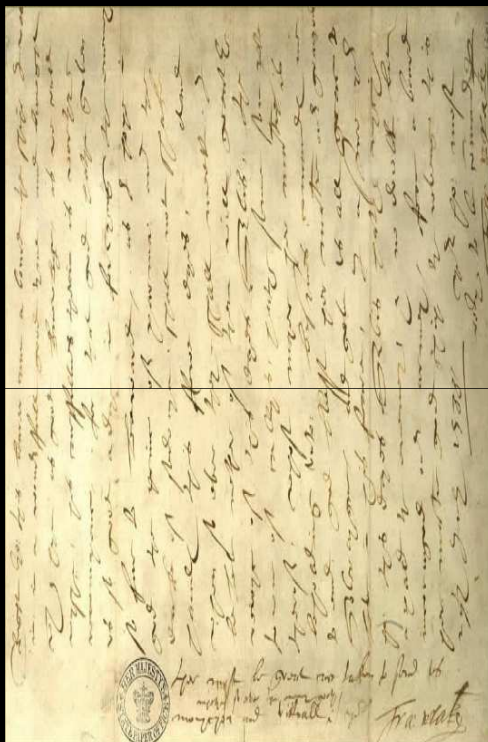
❁ Inconvenientes de la umbralización global:

- Si el número de píxeles del fondo de la imagen es mucho mayor que el número de píxeles de los objetos, $P_0(k) \gg P_1(k)$, se produce un sesgo indeseado del umbral global hacia la zona de fondo.

Solución: Calcular el histograma de la imagen teniendo sólo en cuenta los píxeles que pertenecen a zonas en las que hay variaciones significativas del nivel de gris, es decir, los píxeles resaltados por un detector de bordes (ej.: Sobel, Roberts, Prewitt).

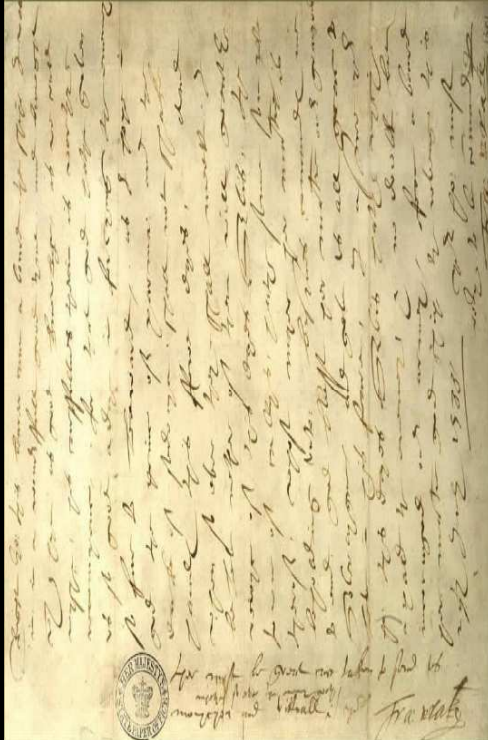
- Variaciones de iluminación en la imagen afectan negativamente.

Solución: Aplicar la técnica global por separado a ventanas alrededor de cada píxel (umbralización local) o a distintas regiones de la imagen (umbralización dinámica).



Efecto de variaciones de luz con umbralización global

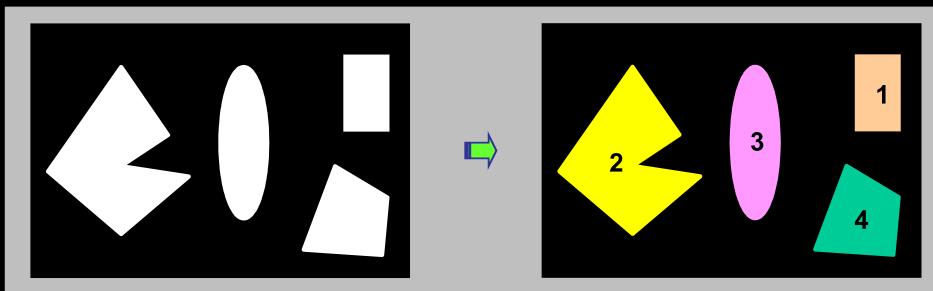
Material del curso "Digital Image Analysis", Lars Aurdal, Norwegian Computer Center, 2006.



Efecto de variaciones de luz con umbralización variable

Material del curso "Digital Image Analysis", Lars Aurdal, Norwegian Computer Center, 2006.

- Permite extraer las diferentes regiones conexas que aparecen en una imagen generalmente binarizada mediante umbralización.
- Se itera sobre todos los píxeles de la imagen dada, detectando los píxeles conexas que tienen el mismo valor de intensidad o, en el caso de imágenes de nivel de gris, todos los píxeles conexas que tienen niveles de gris comprendidos en un intervalo reducido.
- Todos los píxeles que cumplen dichas condiciones reciben una etiqueta que los identifica como miembros de una misma clase.



- Aplicado sobre imágenes binarias, permite contar y extraer los objetos presentes en ellas para poderles aplicar posteriormente y a nivel individual funciones de extracción de características básicas (área, perímetro, momentos de inercia, relación de aspecto, etc...), que permitan, por ejemplo, su identificación.

- Algoritmo de etiquetado de componentes conexas de 2 pasadas. Permite el etiquetado de las componentes conexas de una imagen en dos pasadas sobre ésta.
- Primera Fase: Etiquetado inicial y construcción de la tabla de equivalencias.
 - Sea N el número máximo de clases (componentes conexas) consideradas. Se define una tabla de equivalencias entre clases, $C = \{C[i], \forall i \in [0, N]\}$, tal que la clase i -ésima es equivalente a la clase $C[i]$. Inicialmente $C[i] = i, \forall i$.
 - Se define una matriz L de etiquetas de clase del mismo tamaño que la imagen dada, de forma que $L[x, y]$ indica la etiqueta (clase) correspondiente al píxel $\psi[x, y]$. Inicialmente $L[x, y] = 0, \forall x, y$.
 - Se define una variable, Γ , que contiene el valor de la siguiente etiqueta que será asignada. Inicialmente $\Gamma = 1$.

- Se realiza una primera pasada sobre la imagen para etiquetar todos sus píxeles que cumplen $\psi[x, y] > 0$, recorriendo todas las filas de arriba abajo e izquierda a derecha:

- Sea $\lambda(x, y)$ la etiqueta menor de los vecinos de $\psi[x, y]$ ya etiquetados (no =0):

$$\lambda(x, y) = \min \{L(x-1, y-1), L(x, y-1), L(x+1, y-1), L(x-1, y)\}$$

$$L(x, y) = \begin{cases} 0 & x < 0 \vee y < 0 \vee x \geq x_{MAX} \vee y \geq y_{MAX} \vee L[x, y] = 0 \\ L[x, y] & x \geq 0 \wedge y \geq 0 \wedge x < x_{MAX} \wedge y < y_{MAX} \end{cases}$$

- Si $\lambda(x, y) = 0$ se asigna una nueva etiqueta al píxel, se actualiza la tabla de equivalencias y se incrementa la variable de siguiente etiqueta:

$$L[x, y] = \Gamma, C[\Gamma] = \Gamma, \Gamma = \Gamma + 1$$

- Si $\lambda(x, y) \neq 0$ se asigna esa etiqueta al píxel y se actualiza la tabla de equivalencias:

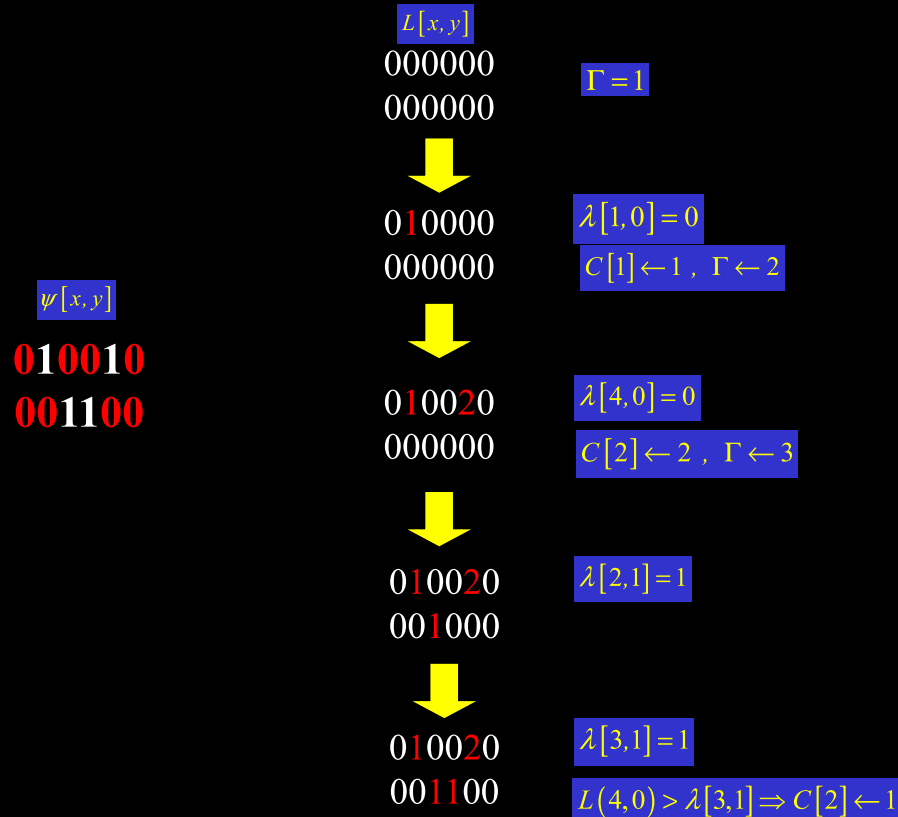
$$L[x, y] = \lambda(x, y)$$

$$L(x-1, y-1) > \lambda[x, y] \Rightarrow C[L(x-1, y-1)] = \lambda[x, y]$$

$$L(x+1, y-1) > \lambda[x, y] \Rightarrow C[L(x+1, y-1)] = \lambda[x, y]$$

$$L(x, y-1) > \lambda[x, y] \Rightarrow C[L(x, y-1)] = \lambda[x, y]$$

$$L(x-1, y) > \lambda[x, y] \Rightarrow C[L(x-1, y)] = \lambda[x, y]$$



• Segunda Fase: Compactación de la tabla de equivalencias.

- La tabla de equivalencias entre clases (etiquetas) es la representación de un árbol con puntero al padre, de manera que cada clase apunta a su clase equivalente o a ella misma.
- Se substituye cada clase que no sea equivalente a si misma por su clase equivalente de valor más pequeño:

```

Para  $i=1$  hasta  $N$  hacer
   $j \leftarrow i$ 
  hacer  $j \leftarrow C[j]$ 
  mientras  $j \neq C[j]$  fmientras
  {  $j$  indica la clase equivalente mínima }
   $C[i] \leftarrow j$ 
fpara
    
```

- La tabla de equivalencias puede volverse a compactar de modo que todas las etiquetas finales sean consecutivas sin huecos intermedios.

• Tercera Fase: Etiquetado final.

- Se efectúa una segunda pasada sobre la imagen para cambiar las etiquetas de los píxeles por sus equivalencias:

$$L[x, y] = C[L[x, y]], \quad \forall x, y$$

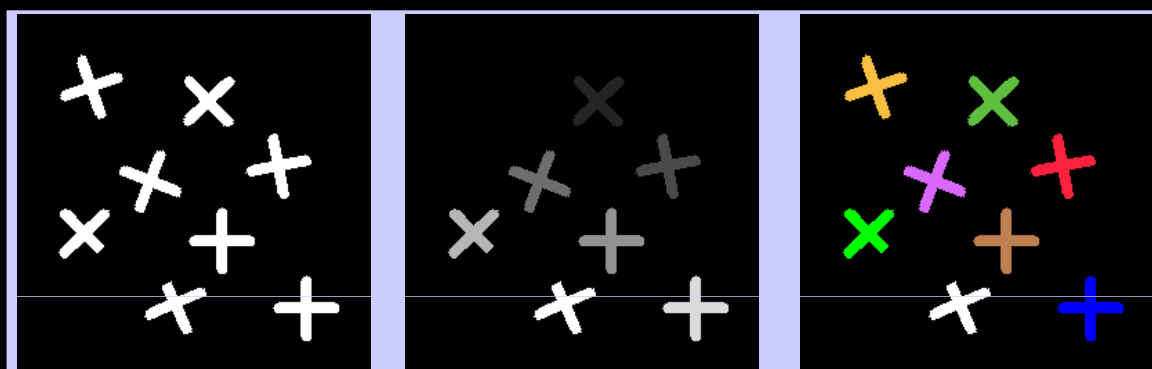


Imagen binaria

Imagen etiquetada
(eualizada)

Imagen etiquetada
(pseudocolor)

Material del curso interactivo HIPR2 (R.Fisher, S.Perkins, A.Walker, E.Wolfart, 2004).

Imagen original

Imagen binarizada

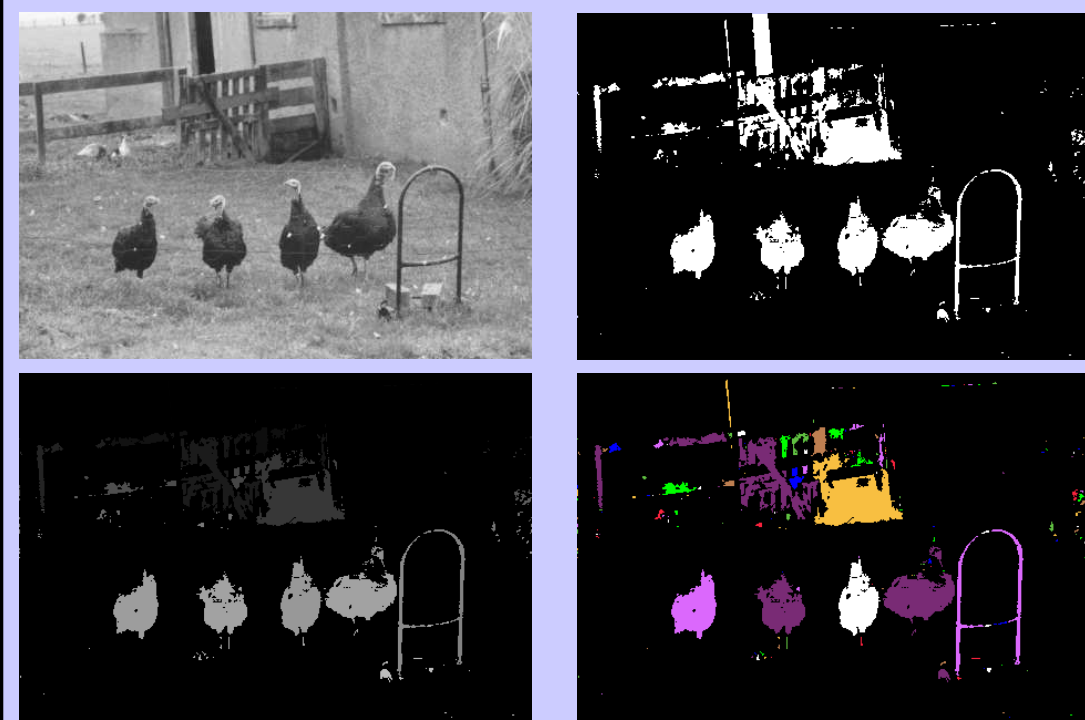


Imagen etiquetada

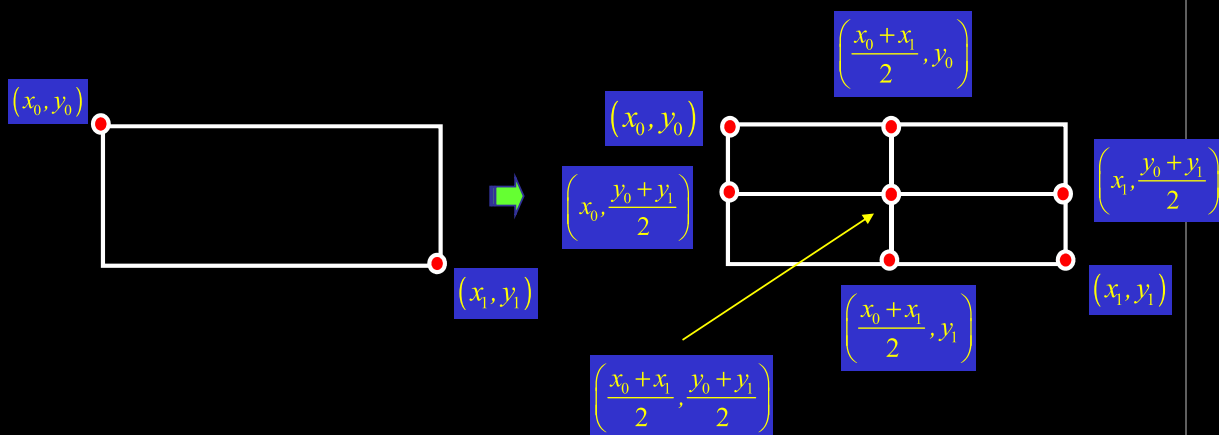
Imagen etiquetada (pseudocolor)

Material del curso interactivo HIPR2 (R.Fisher, S.Perkins, A.Walker, E.Wolfart, 2004).

- Permite extraer las diferentes regiones uniformes que aparecen en una imagen.
- Se define una función de heterogeneidad H que, aplicada a una región usualmente rectangular, $\{(x_0, y_0), (x_1, y_1)\}$, de una imagen ψ , indica el grado de heterogeneidad de dicha región con respecto a alguna característica visual (intensidad, color, textura, ...): $H(\psi, (x_0, y_0), (x_1, y_1))$
- La función de heterogeneidad puede definirse de diversas formas según la característica visual que se esté considerando:
 - Varianza de los niveles de gris de los píxeles contenidos en la región.
 - Diferencia entre el nivel de gris máximo y el mínimo (criterio Max-Min) de los píxeles contenidos en la región. Más eficiente computacionalmente.
- Criterio de **partición** de regiones: Una región cuya función de heterogeneidad supere un umbral predefinido se puede partir en dos o más regiones por considerarse no homogénea.
- Criterio de **crecimiento** de regiones: Dos regiones adyacentes cuya función de heterogeneidad conjunta esté por debajo de un umbral predefinido se pueden fusionar por considerarse homogéneas.

Segmentación mediante partición de regiones: **Subdivisión Quadtree**

- Sea una imagen $\psi[x, y], x \in [0, x_{MAX}), y \in [0, x_{MAX})$, una función de heterogeneidad H , un umbral de heterogeneidad η y un tamaño mínimo de región Δ .
- La imagen dada se subdivide iterativamente en regiones homogéneas más pequeñas hasta que todas las regiones así generadas son homogéneas o todas las regiones heterogéneas que quedan tienen un tamaño menor que Δ .
- Siempre que una región no homogénea se subdivide, se parte en cuatro regiones del mismo tamaño: **subdivisión quadtree**.



Ejemplo de subdivisión quadtree

- Algoritmo recursivo de segmentación mediante partición quadtree:

```

procedimiento Segmentar (  $\psi, x_0, y_0, x_1, y_1$  ) es
{ Si la región presenta tamaño inferior al mínimo finaliza la recursividad y se añade al resultado de la segmentación }
si  $x_1 - x_0 + 1 < \Delta$  o  $y_1 - y_0 + 1 < \Delta$  entonces Nuevo_Segmento (  $x_0, y_0, x_1, y_1$  );
si_no
    { Si la región es homogénea finaliza la recursividad y se añade al resultado de la segmentación }
    si  $H(\psi, (x_0, y_0), (x_1, y_1)) < \eta$  entonces Nuevo_Segmento (  $x_0, y_0, x_1, y_1$  );
    si_no
        { La región es heterogénea: se subdivide en cuatro }
        Segmentar (  $\psi, x_0, y_0, (x_0 + x_1)/2, (y_0 + y_1)/2$  );
        Segmentar (  $\psi, (x_0 + x_1)/2, y_0, x_1, (y_0 + y_1)/2$  );
        Segmentar (  $\psi, x_0, (y_0 + y_1)/2, (x_0 + x_1)/2, y_1$  );
        Segmentar (  $\psi, (x_0 + x_1)/2, (y_0 + y_1)/2, x_1, y_1$  );
    fsi
    fsi
fprocedimiento
    
```

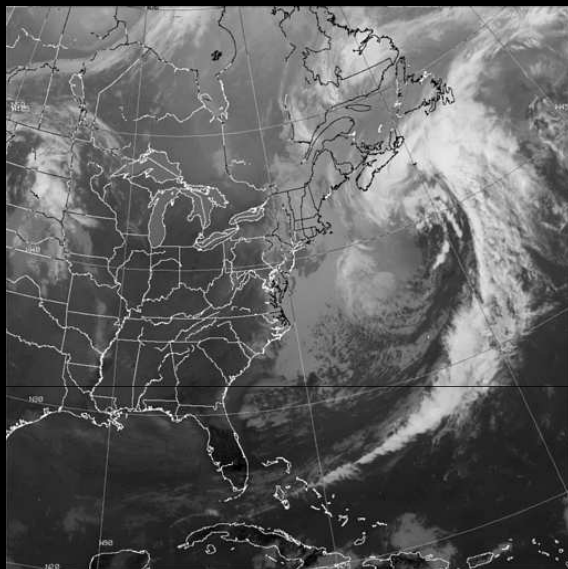
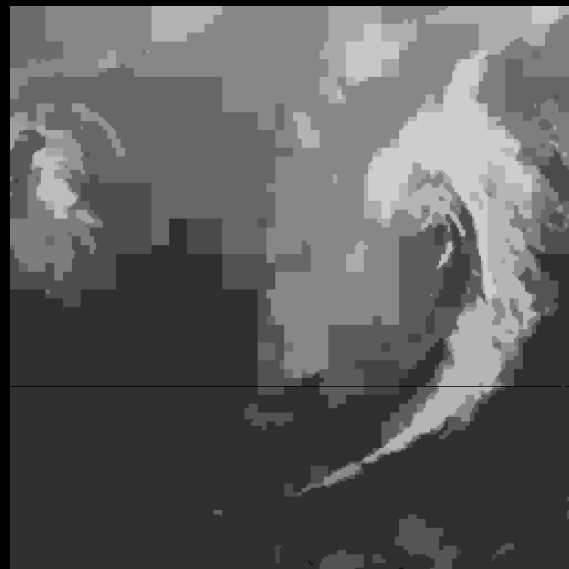


Imagen de satélite en banda de infrarrojos



Segmentación en regiones de nivel de gris uniforme mediante partición quadtree (permite detectar las zonas de nubes)

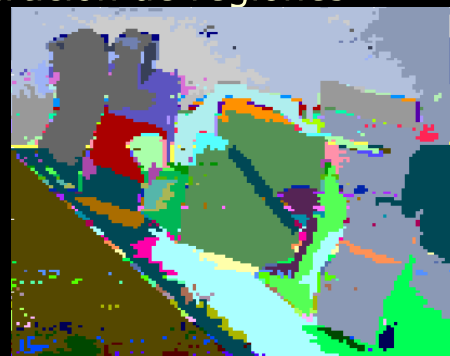
Canadian Meteorological and Oceanographic Society, 2005.

• Segmentación mediante crecimiento de regiones:

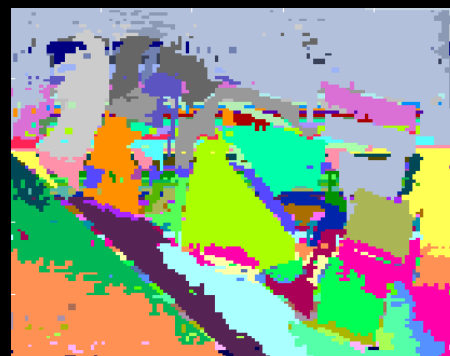
- La imagen dada se parte en bloques disjuntos pequeños de tamaño fijo, por ejemplo de 2x2 píxeles usando una subdivisión quadtree.
- Siempre que dos bloques adyacentes tienen una función de heterogeneidad por debajo de un umbral predefinido, los bloques se fusionan en un solo por considerarse homogéneos.
- Un criterio más simple que la función de heterogeneidad en este caso consiste en calcular la **diferencia entre las medias de los niveles de gris** de cada bloque. Si la diferencia de medias está por debajo de un umbral, es que son bloques homogéneos y, por tanto, se fusionan.
- El procedimiento se repite hasta que no se puede fusionar ningún bloque más.



Imagen original



Crecimiento de regiones con diferencia de medias < 8



Crecimiento de regiones con Max-Min < 8

Material del curso "Digital Image Processing",
J.Nearing, R.Pickel, A.Maliyekkel, Rice University.

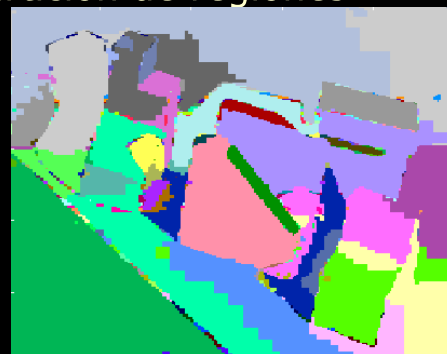
Segmentación mediante partición y crecimiento de regiones (split & merge):

(S.L. Horowitz, T. Pavlidis, "Picture Segmentation by a Tree Traversal Algorithm", Journal of the ACM, vol.23, no.2, 368-388, Abril 1976).

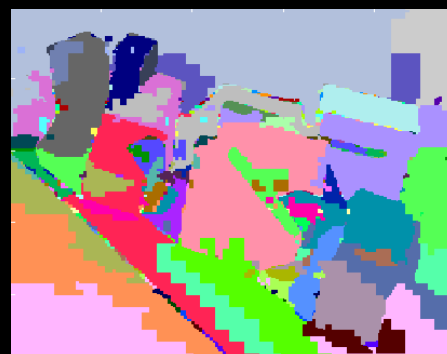
- Es una combinación de ambos métodos: Obtiene la solución de manera más rápida.
- La imagen dada se parte en bloques disjuntos de tamaño fijo, por ejemplo de 16x16 píxeles (se escoge un tamaño de bloque mayor que con crecimiento puro) usando una subdivisión quadtree.
- Se aplica una etapa de partición recursiva sobre los bloques iniciales.
- Finalmente se aplica una etapa de crecimiento de regiones sobre los bloques resultantes.



Imagen original



Split & Merge con diferencia de medias < 8



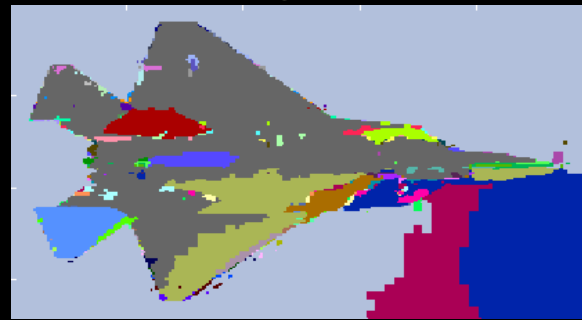
Split & Merge con Max-Min < 8

Material del curso "Digital Image Processing",
J.Nearing, R.Pickel, A.Maliyekkel, Rice University.



Imagen original

*Material del curso "Digital Image Processing",
J.Nearing, R.Pickel, A.Maliyekkel, Rice University.*

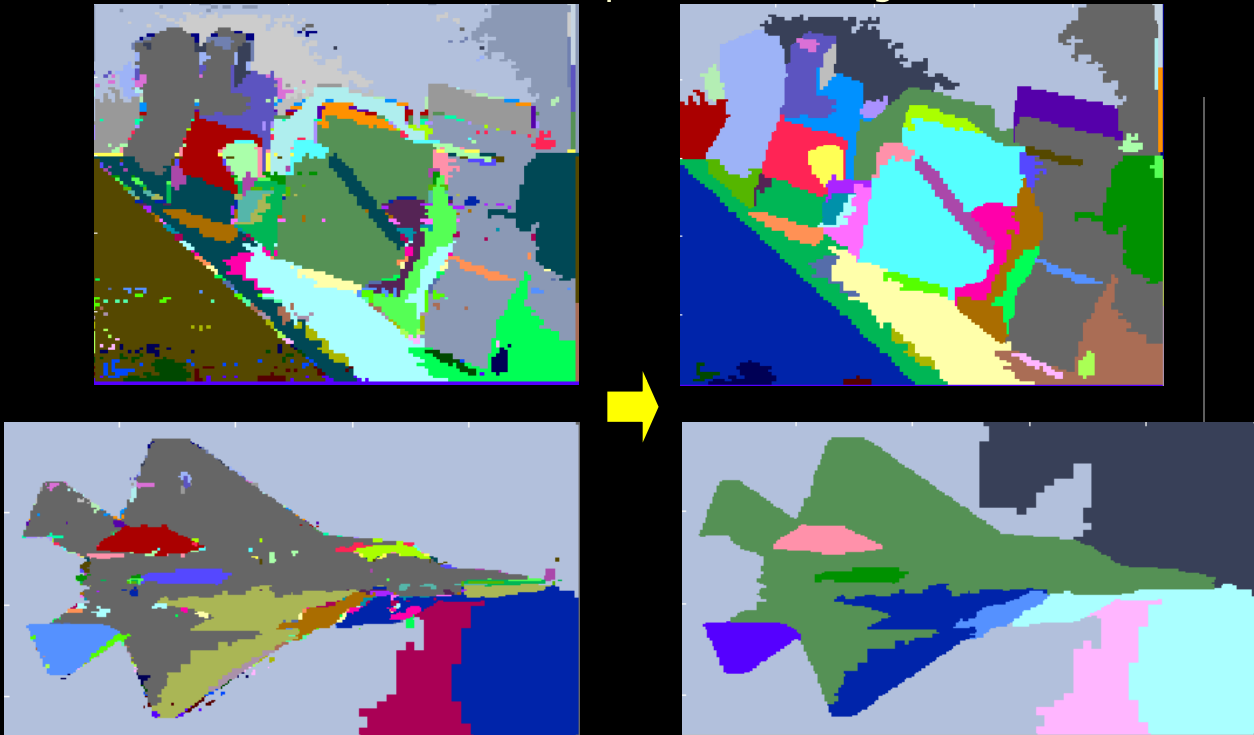


Split & Merge con diferencia de medias < 16



Split & Merge con Max-Min < 16

- Estas técnicas de segmentación admiten la aplicación de un postproceso de filtrado que agrega las regiones resultantes de área muy pequeña en regiones adyacentes mayores.



Filtrado de regiones no significativas

● Introducción

- Tipos de características geométricas
- Aplicaciones

● Extracción y caracterización de puntos/esquinas

- Detector de Harris & Stephens
- Detector SUSAN
- Detector SIFT

● Extracción de bordes

- Transformada de Hough
- Transformada de Hough generalizada
- Seguimiento de bordes
- Contornos dinámicos

● Extracción de regiones

- Umbralización
- Etiquetado de componente conexas
- Crecimiento/partición de regiones