



II. Características y descriptores

- Objetivo: Extraer y etiquetar información pertinente.

2.1 Extracción de características

- Bordes (Sobel, Canny...)
- Formas geométricas (líneas, círculos)
- Esquinas (Harris)
- Espacio-escala (pirámides, SIFT)

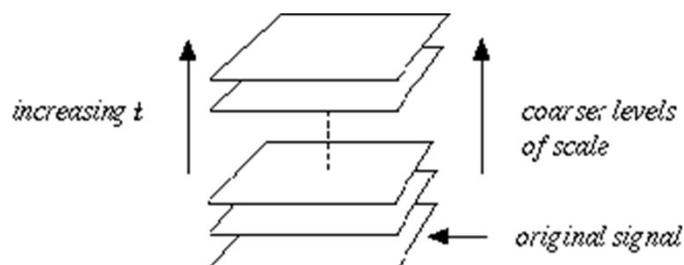
Análisis de escala

- Hemos visto que la dimensión del filtro de suavizado determina el tamaño de los detalles detectados.
- Los detectores vistos hasta ahora consideran un solo nivel de detalle (*escala*).
- Los objetos de interés pueden tener diferentes tamaños o estar a diferentes distancias de la cámara.
- No es posible conocer *a priori* la(s) escala(s) útil(es) de una imagen.
- Es indispensable analizar todas las escalas posibles para obtener la información completa.

3

Teoría espacio-escala

- La representación espacio-escala de una imagen consiste en una familia de imágenes suavizadas a diferentes niveles de detalle (definido por un parámetro de escala t).



4

Teoría espacio-escala

- Una imagen $I(x,y)$ genera la familia:

$$L(x,y;t) = g(x,y;t) * I(x,y)$$

donde $g(x,y;t)$ es el núcleo de suavizado en la escala t .

- Una restricción razonable es que la estructura a bajo nivel de detalle debe resultar de la simplificación de la estructura a alto nivel de detalle, es decir: no deben aparecer nuevos elementos cuando subimos la escala.
- Para eso, se puede mostrar que $g(x,y;t)$ debe ser una Gaussiana.

5

Teoría espacio-escala

- La familia $L(x,y;t)$ es la **representación espacio-escala** de la imagen $I(x,y)$.

$$g(x,y;t) = \frac{1}{2\pi t} e^{-\frac{x^2+y^2}{2t}}$$

- Para $t = 0$, $g(x,y;0)$ es una función impulso $\delta(x,y)$, de manera que $L(x,y;0) = I(x,y)$ es la imagen original.
- La desviación estándar de $g(x,y;t)$ es \sqrt{t} , de manera que $L(x,y;t)$ no contiene detalles menores a \sqrt{t} .

6

Teoría espacio-escala



$t = 0$



$t = 1$



$t = 4$



$t = 16$



$t = 64$



$t = 256$

7

Teoría espacio-escala

- Se puede mostrar que la familia de funciones $L(x,y;t)$ es solución de la ecuación de difusión del calor:

$$\frac{\partial L}{\partial t} = \frac{1}{2} \left(\frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2} \right)$$

- Esto significa que la representación espacio-escala puede ser vista como la difusión al largo del tiempo de la distribución de temperatura representada por la imagen inicial $I(x,y)$.

8

Pirámides

- Ya que los detalles pequeños desaparecen cuando la escala aumenta, se puede pensar en reducir la resolución a escalas más altas.
- Así se obtiene una **pirámide de imágenes**, es decir una colección de imágenes con resolución decreciente arregladas en forma de pirámide.

Pirámides

La imagen de la base de la pirámide contiene la más alta resolución, mientras que la punta de la pirámide contiene la más baja resolución. Conforme nos movemos hacia arriba de la pirámide, tanto el tamaño como la resolución de las imágenes disminuyen.

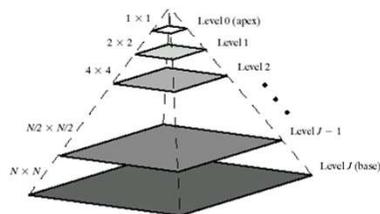
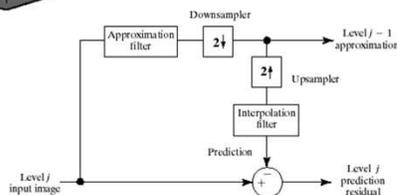


FIGURE 7.2 (a) A pyramidal image structure and (b) system block diagram for creating it.

Tamaño : $2^j \times 2^j$,
donde $0 \leq j \leq J$



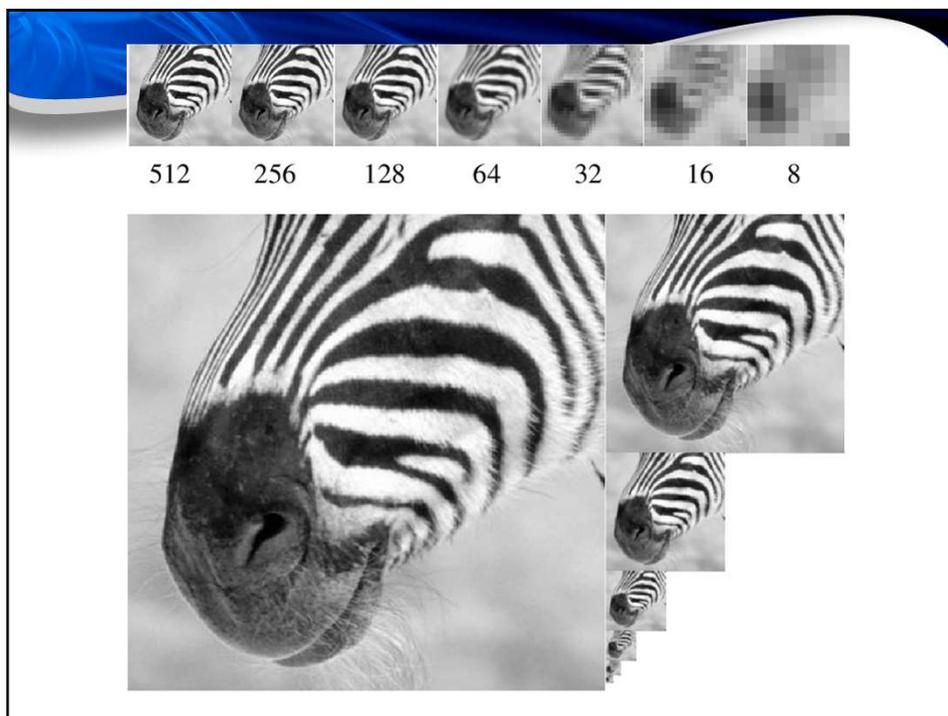
Pirámides Gaussianas

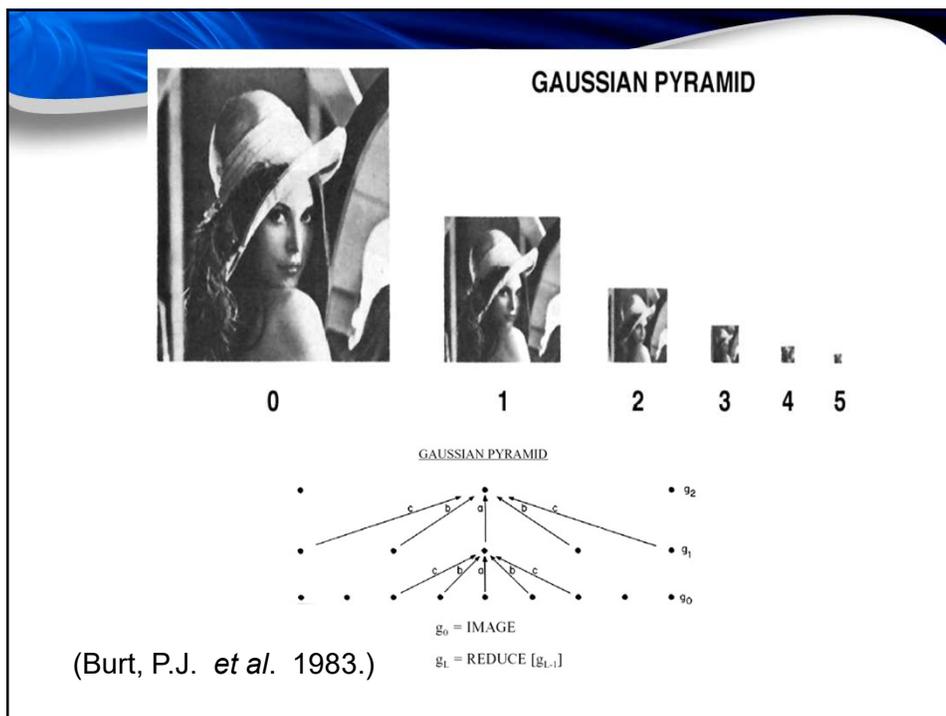
En una pirámide Gaussiana cada nivel de la pirámide es *suavizado* por un kernel Gaussiano simétrico y *submuestreado* para obtener la siguiente capa.

Notación: S^\downarrow submuestra la imagen I .

$$P_{Gaussian}(I)_{J-1} = S^\downarrow(G_\sigma \otimes P_{Gaussian}(I)_J)$$

donde, $P_{Gaussian}(I)_J = I$; es la original





Pirámides Gaussianas

Aplicaciones:

- **Búsqueda de escala:** Muchos objetos pueden ser representados con pequeños patrones. Por ejemplo las caras tiene patrones muy bien definidos: dos ojos hundidos en fondos oscuros, debajo de dos rayas oscuras (cejas), separados de una luz especular (nariz) y sobre una barra oscura (boca). En una pirámide podemos buscar caras grandes medianas y pequeñas a lo largo de sus diferentes capas
- **Búsqueda espacial:** En visión estereoscópica, o análisis de movimiento, se buscan pares de puntos que casen en una escala gruesa y luego se va buscando en escalas más finas, con más detalle pero en zonas de búsqueda significativamente menores.

Pirámides Gaussianas

Aplicaciones:

- **Seguimiento de características (tracking):** la mayoría de las características encontradas en escalas gruesas están asociadas con cambios bruscos de contraste. Típicamente, encontrar objetos en escalas gruesas subestima tanto el tamaño como la localización de los objetos. Por ejemplo un error de un sólo pixel a escala gruesa representa un error de múltiples pixeles en escala fina.

En escalas finas existen muchos eventos asociados con cambios pequeños de contrastes bajos. Una estrategia para mejorar la caracterización de objetos en escalas finas es hacer el seguimiento de la característica de escalas finas a gruesas y quedarse con aquellas que tiene sus respectivos padres a lo largo de la pirámide (por ejemplo, diferencia entre ruido y estructuras de ciertas texturas).

Pirámides Laplacianas

Las pirámides Laplacianas hacen uso del hecho de que una capa gruesa de la pirámide Gaussiana predice la apariencia de la siguiente capa fina.

Si utilizamos un operador de *sobremuestreo* que pueda producir una versión de capa gruesa del mismo tamaño que la siguiente capa fina, entonces sólo necesitamos almacenar la diferencia entre estas dos predicciones y la siguiente capa fina.

Es decir, es una secuencia de imágenes de error, diferencias de dos capas de la pirámide Gaussiana. Cada una de las capas finas de la pirámide Laplaciana es la diferencia entre una capa de pirámide Gaussiana y una predicción obtenida *sobremuestreando* la siguiente capa Gaussiana de la pirámide.

$$P_{Laplaciana}(I)_k = P_{Gaussiana}(I)_k - S^\uparrow(P_{Gaussiana}(I)_{k+1})$$

Pirámides Laplacianas

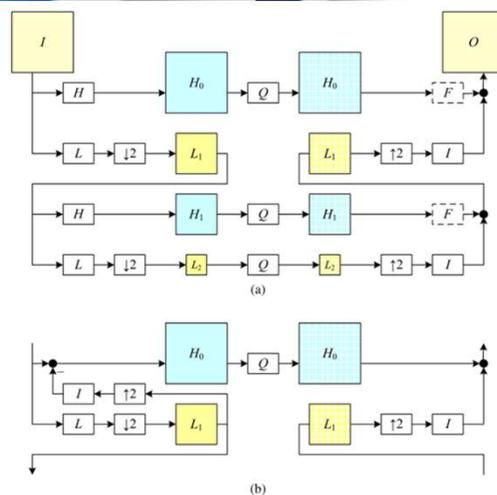
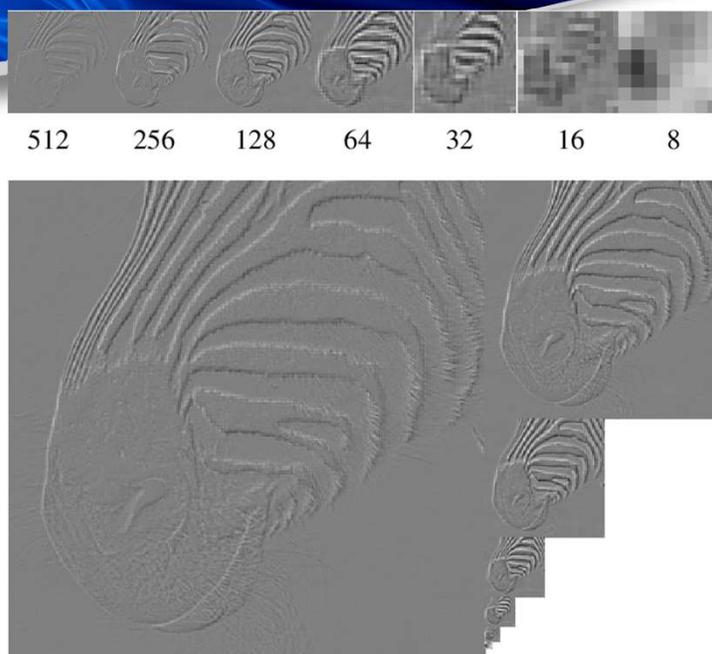
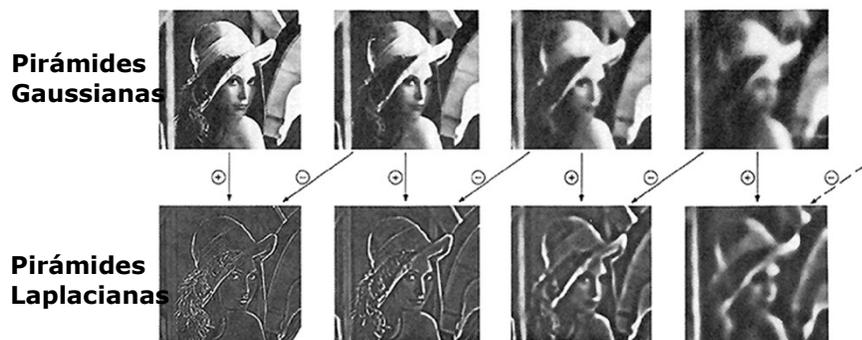


Figure 3.34 The Laplacian pyramid: (a) The conceptual flow of images through processing stages: images are high-pass and low-pass filtered, and the low-pass filtered images are processed in the next stage of the pyramid. During reconstruction, the interpolated image and the (optionally filtered) high-pass image are added back together. The Q box indicates quantization or some other pyramid processing, e.g., noise removal by *coring* (setting small wavelet values to 0). (b) The actual computation of the high-pass filter involves first interpolating the downsampled low-pass image and then subtracting it. This results in perfect reconstruction when Q is the identity. The high-pass (or band-pass) images are typically called *Laplacian* images, while the low-pass images are called *Gaussian* images.



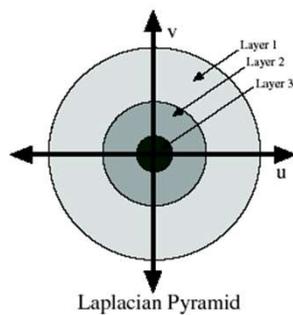
Pirámides Laplacianas



(Burt, P.J. *et al.* 1983.)

Pirámides Laplacianas

Cada capa de la pirámide Laplaciana puede verse como la respuesta de un *filtro paso-banda*, y representa diferentes frecuencias espaciales (aplicación compresión).



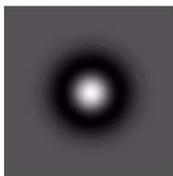
Detector SIFT

- En 1999, David Lowe propuso un detector invariante a escala y rotaciones, el **Scale-Invariant Feature Transform (SIFT)**.
- También es parcialmente robusto a cambios de iluminación y punto de vista.
- El detector se calcula rápido y es muy distintivo.
- Combina varias de las ideas descritas hasta ahora:
 - Filtro LoG
 - Espacio-escala
 - Distinción entre bordes y esquinas

Detector SIFT

El regreso del LoG

- Vimos que los cruces por cero del LoG permiten detectar bordes.
- Por otro lado, los **extremos** del LoG permiten detectar **manchas**:



- El tamaño de las manchas detectadas depende de la escala del LoG.

Detector SIFT

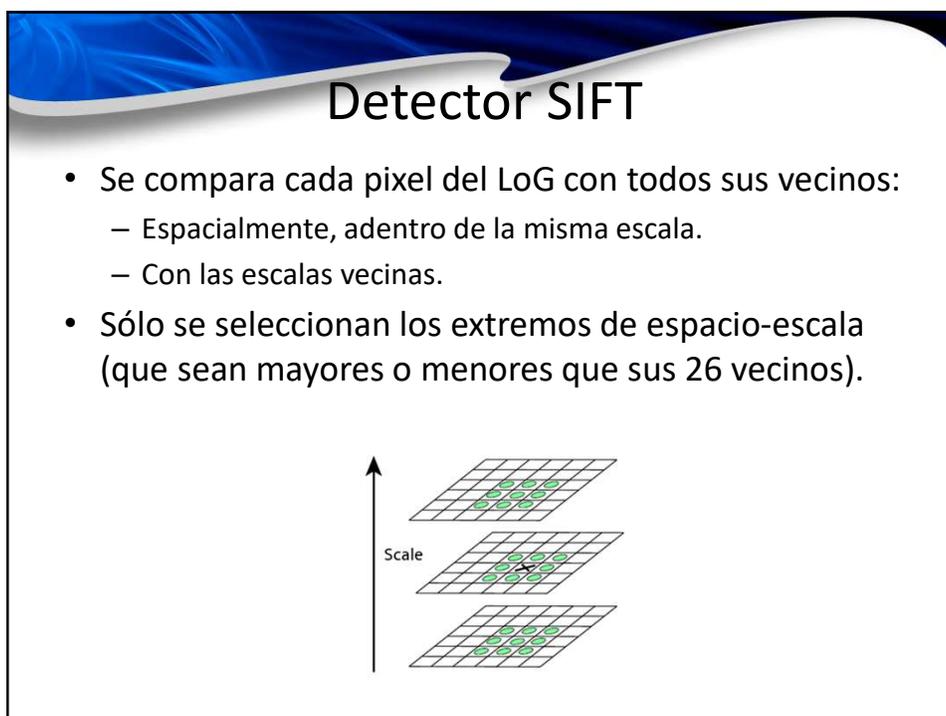
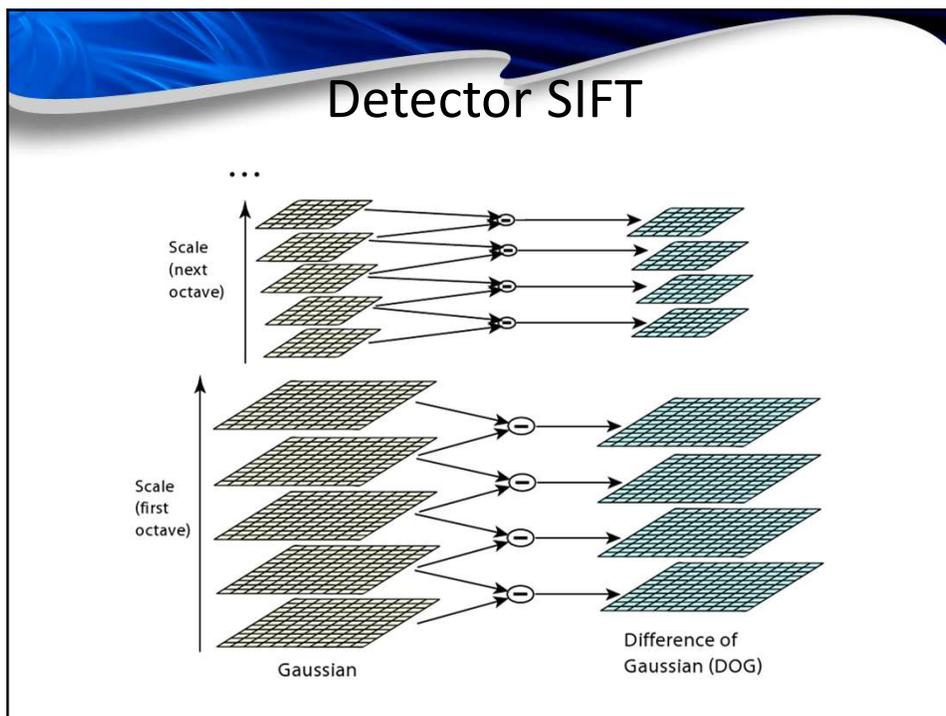
- Vimos que el LoG se puede aproximar con el DoG.

$$\begin{aligned}
 \text{LoG}(x, y; t) * I(x, y) &\approx \text{DoG}(x, y; t) * I(x, y) \\
 &= [g(x, y; kt) - g(x, y; t)] * I(x, y) && k > \sim 1 \\
 &= g(x, y; kt) * I(x, y) - g(x, y; t) * I(x, y) \\
 &= L(x, y; kt) - L(x, y; t)
 \end{aligned}$$

- El resultado del LoG se obtiene como **la diferencia entre dos escalas vecinas**.

Detector SIFT

- Se combinan la representación espacio-escala y la pirámide Gaussiana:
 - Se calculan varias escalas entre t y $2t$ (una octava), todas con la misma resolución.
 - A partir de la escala $2t$, se divide a la mitad la resolución y se usa esta resolución reducida para la siguiente octava.
 - Etc.
- Una vez obtenidas todas las escalas, se obtienen los LoG por sustracción entre dos escalas consecutivas.



Detector SIFT

- Etapas adicionales:
 - Localización sub-píxel de los extremos por interpolación.
 - Eliminación de los extremos poco contrastados donde $|DoG(x,y;t)| < 0.03$
 - Eliminación de los puntos sobre bordes (localización inestable). Esto se hace usando las curvaturas de $DoG(x,y;t)$ dadas por las segundas derivadas, y aplicando un criterio similar al de Harris para esquinas: la razón entre las dos curvaturas principales debe ser menor a 10.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad R = \frac{\text{Tr}(H)^2}{\text{Det}(H)}$$

Referencias

- Gonzalez , R.C. Y Woods, R.E. *Digital Image Processing* Prentice Hall, 2002. **Capítulo 3 y 10.**
- Forsyth, D.A. y Ponce, J. *Computer Vision: A modern approach*. Prentice-Hall, 2003. **Capítulos 7 y 9.**
- Lindeberg, T., *Scale-Space Theory in Computer Vision*, Kluwer Academic Publishers, 1994.
- Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints”, *International Journal of Computer Vision*, 60, 2, pp. 91-110, 2004.

II. Características y descriptores

2.2 Descriptores

- Parches de imágenes
- Histogramas
- SIFT, SIFT-PCA, GLOH

29

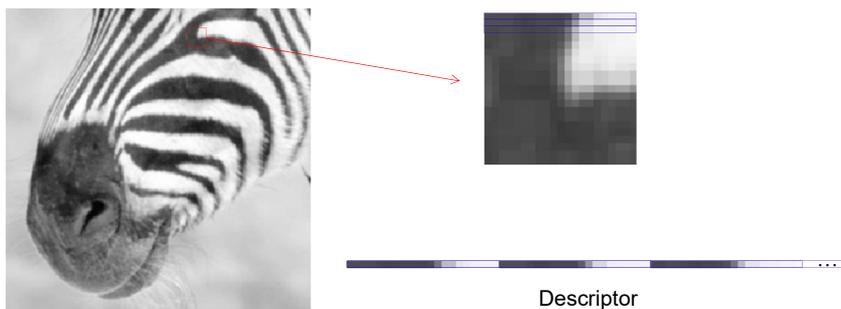
Descriptores locales

- Con los detectores, hemos encontrado puntos de interés en la imagen. Pero ¿cómo diferenciar un punto de otro?
- Definir un vector (**descriptor**) que caracteriza la región alrededor de un punto de interés.
- Para comparar imágenes, el descriptor debe ser invariante a cambios de condiciones (punto de vista, iluminación...).

30

Parche de imagen

- La idea más simple consiste en crear un vector que contiene los valores de los píxeles de una ventana alrededor del punto de interés.



31

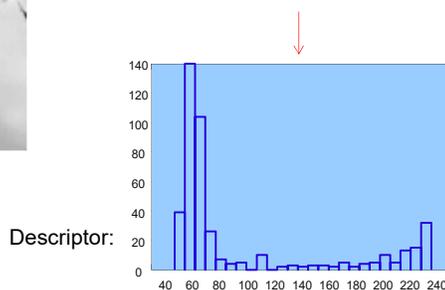
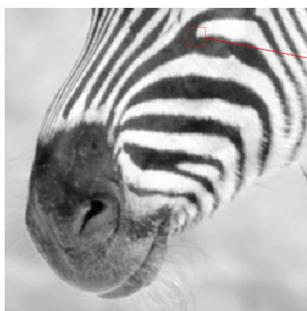
Parche de imagen

- Desventajas:
 - Dimensión alta para una descripción correcta.
 - Muy sensible a cambios de iluminación, de punto de vista, etc.
- Sólo utilizable para imágenes tomadas en condiciones similares, por ejemplo para estereoscopia.

32

Histograma local

- El descriptor es el histograma de la ventana alrededor del punto de interés.



33

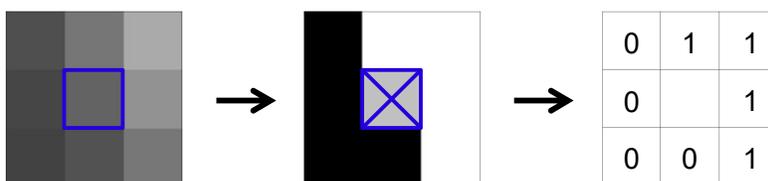
Histograma local

- **Ventaja:**
 - La dimensionalidad es reducida.
 - Un poco más robusto a ciertos cambios como rotaciones.
- **Desventajas:**
 - Descriptor muy general/poco discriminante.
 - Sigue sensible a cambios de iluminación.

34

LBP

- Binarizar los pixeles de la ventana usando el valor del pixel central como umbral.
- El descriptor es la secuencia de bits obtenida por lectura secuencial de todos los pixeles.
- Más robusto que los valores de intensidad.
- Más detallado que el histograma.



35

Momentos generalizados

- Para una ventana Ω de la imagen I , se pueden calcular los momentos generalizados:

$$M_{pq}^a = \iint_{\Omega} x^p y^q [I(x, y)]^a dx dy$$

de orden $p+q$ y de grado a .

- Para $a = 1$, se obtienen los momentos espaciales:
 - M_{10}^1 y M_{01}^1 : coordenadas del centroide de la ventana
 - M_{20}^1 , M_{11}^1 y M_{02}^1 : covarianza (extensión espacial)
- Para $p = q = 0$, se obtienen los momentos de intensidad

36

Momentos generalizados

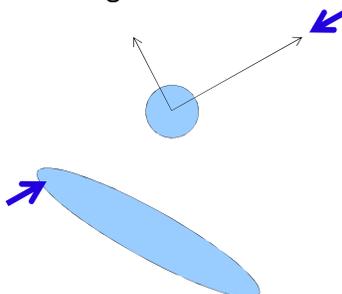
- Ventajas:
 - Fáciles de calcular.
 - Un número suficiente de momentos puede ser calculado con valores bajos de orden y grado.
- Desventajas:
 - Se necesitan muchos momentos para obtener una descripción suficientemente discriminante.
 - Los momentos de orden y/o grado alto son muy sensibles a cambios geométricos y de intensidad.

37

Regiones afín-invariantes y imágenes spin

Normalización afín

- Detectores como Harris o DoG nos dan una razón entre eigenvalores, que indica la diferencia de extensión entre los dos ejes principales.
- Se puede normalizar la región aplicando una transformación afín que iguale los dos eigenvalores.



38

Referencias

- R. Zabih and J. Woodfill, Non-Parametric Local Transforms for Computing Visual Correspondance, Proc. ECCV, pp. 151-158, 1994.
- S. Belongie, J. Malik, J. Puzicha, Shape Matching and Object Recognition Using Shape Contexts, IEEE Trans. PAMI, Vol. 24, No. 24, pp. 509-522, 2002.
- S. Lazebnik, C. Schmid, and J. Ponce, Sparse Texture Representation Using Affine-Invariant Neighborhoods, Proc. CVPR, pp. 319-324, 2003.
- K. Mikolajczyk and C. Schmid, A Performance Evaluation of Local Descriptors, IEEE Trans. PAMI, Vol. 27, No. 10, pp. 1615-1630, 2005.