

# Texturas y *features*

Procesamiento Digital de Imágenes

Departamento de Computación

Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

1er Cuatrimestre 2021



# *Features* en imágenes

## Qué es un *feature*?

- Atributo distintivo o descripción de ‘algo’ que queremos etiquetar o diferenciar.

# *Features* en imágenes

## Qué es un *feature*?

Atributo distintivo o descripción de *algo* que queremos etiquetar o diferenciar.

- *algo*: objetos individuales en imágenes, o imágenes enteras.

# *Features* en imágenes

## Qué es un *feature*?

Atributo distintivo o descripción de *algo* que queremos *etiquetar* o *diferenciar*.

- *algo*: objetos individuales en imágenes, o imágenes enteras.
- *etiquetar*: asignar una clase a un objeto en la imagen.

# *Features* en imágenes

## Qué es un *feature*?

Atributo distintivo o descripción de *algo* que queremos *etiquetar* o *diferenciar*.

- *algo*: objetos individuales en imágenes, o imágenes enteras.
- *etiquetar*: asignar una clase a un objeto en la imagen.
- *diferenciar*: queremos que sea de valor para distinguir esos objetos.

# *Features* en imágenes

## Extracción de features

- Detección: encontrar los features en una imagen, región o límite de la misma.
- Descripción: asignar atributos cuantitativos (o cualitativos) a los features.
- Matching: cómo podemos compararlos entre sí?

# Features en imágenes

## Extracción de features

- Detección: encontrar los features en una imagen, región o límite de la misma.
- Descripción: asignar atributos cuantitativos (o cualitativos) a los features.
- Matching: cómo podemos compararlos entre sí?

## Ejemplo de feature: esquinas

- Detección: encontrar las esquinas con el método de Harris.
- Descripción: ubicación espacial de las esquinas, determinar su orientación, etc.
- Matching: introducir una noción de distancia

# *Features* en imágenes

## Características

### *Features* invariantes vs. covariantes

- Invariantes: un feature es invariante a un conjunto de transformaciones si la aplicación de las transformaciones sobre la imagen no modifica la descripción del feature.
- Covariantes: la aplicación de las transformaciones afecta o modifica la descripción del feature.

# *Features* en imágenes

## Características

### *Features* invariantes vs. covariantes

- Invariantes: un feature es invariante a un conjunto de transformaciones si la aplicación de las transformaciones sobre la imagen no modifica la descripción del feature.
- Covariantes: la aplicación de las transformaciones afecta o modifica la descripción del feature.

### Características deseables

- Features invariantes a traslaciones, rotaciones, escala, cambios de iluminación, cambios a los puntos de vista del sensor de captura, etc.

# Qué es una textura?

# Qué es una textura?

## Definición 1

"La textura es la variación de los datos a escalas más pequeñas que las escalas de interés"

Petrou, Sevilla, 'Image Processing - Dealing with Texture', 2006



- Por ejemplo, si estamos interesados en identificar a la persona en la figura, el patrón en la camisa se considera como textura.
- Si estamos interesados en identificar una flor en la camisa, cada flor del patrón es un objeto sin textura en la escala de esta imagen, ya que apenas podemos ver ningún detalle dentro de ella.

# Qué es una textura?

## Definicion 2

“Una región en una imagen tiene una textura constante si un conjunto de estadísticas locales u otras propiedades locales de la función imagen son constantes, varían lentamente o son aproximadamente periódicas.”

Sklansky, 'Image Segmentation and Feature Extraction,' IEEE Trans. on Systems, Man, and Cybernetics, 1978.

## Definicion 3

“La noción de textura parece depender de tres ingredientes: (i) se repite algún ‘orden’ local en una región que es grande en comparación con el tamaño del orden, (ii) el orden consiste en la disposición no aleatoria de partes elementales, y (iii) las partes son entidades aproximadamente uniformes que tienen aproximadamente las mismas dimensiones en todas partes dentro de la región texturizada.”

Hawkins, 'Textural Properties for Pattern Recognition,' In Picture Processing and Psychopictorics, 1969.

# Taxonomía

## Categorías de descriptores de texturas

- **Estructural:** conjunto de texturas primitivas con una relación regular.
- **Estadístico:** medida cuantitativa del arreglo de intensidades en una imagen con lo cual se generan vectores de patrones.

Haralick, 'Statistical and structural approaches to texture'. (1979)

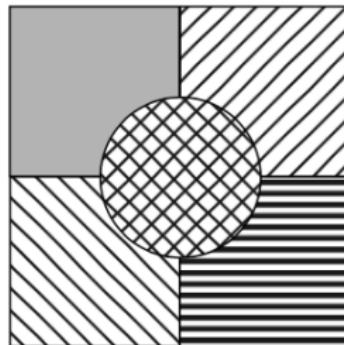
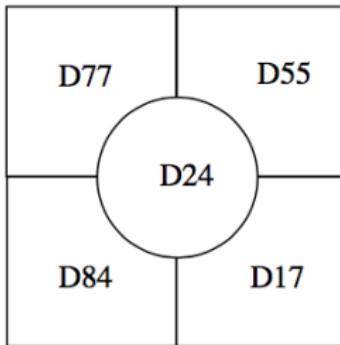
## Otras categorías

- Métodos dependencia espacial de niveles de gris.
- Features basados en frecuencias espaciales.
- Features basados en modelos estocásticos.
- Métodos de filtrado.
- Enfoques heurísticos.

Wu, Chen, 'Statistical feature matrix for texture analysis' (1992)

# Qué podemos hacer analizando texturas?

- **Segmentación:** identificar las regiones de textura similares en una imagen.
- **Reconocimiento:** clasificar una región de textura entre un conjunto de clases de texturas conocidas.
- **Síntesis:** generación automática de texturas a partir de muestras.



# Matrices de Co-ocurrencia

## Definiciones

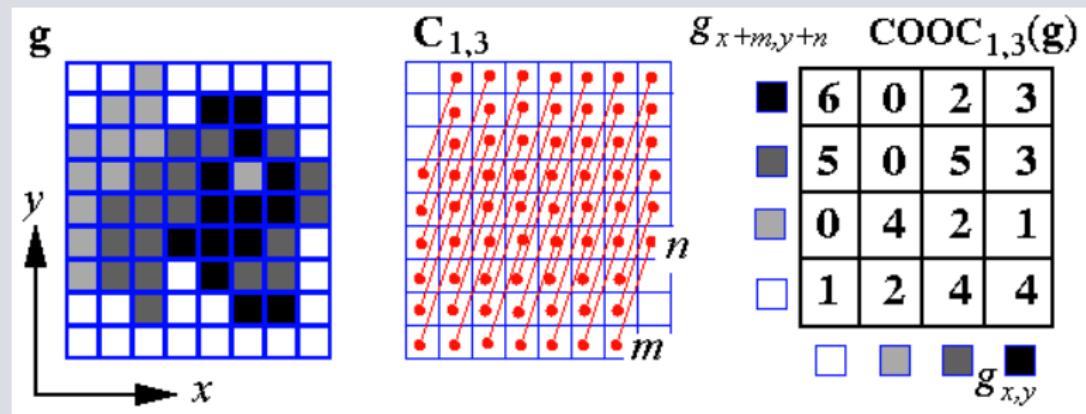
- Las imágenes de 256 escalas de gris tienen matrices de co-ocurrencia de  $256 \times 256$
- Sea  $(\Delta x, \Delta y)$  un desplazamiento en la dirección  $x$  e  $y$ .
- La posición  $(i, j)$  de la matriz de co-ocurrencia  $C_{\Delta x, \Delta y}$  indica la cantidad de veces que el pixel de gris  $i$  se encuentra desplazado  $(\Delta x, \Delta y)$  respecto de un pixel de gris  $j$ :

$$C_{\Delta x, \Delta y}(i, j) = \sum_{x=1}^n \sum_{y=1}^m \begin{cases} 1 & I(x, y) = i \wedge I(x + \Delta x, y + \Delta y) = j \\ 0 & \text{en otro caso} \end{cases}$$

# Matrices de Co-ocurrencia

## Ejemplo

- $(\Delta x, \Delta y) = (1, 3)$



# Matrices de Co-ocurrencia

## Estadísticos sobre la matriz de co-ocurrencia

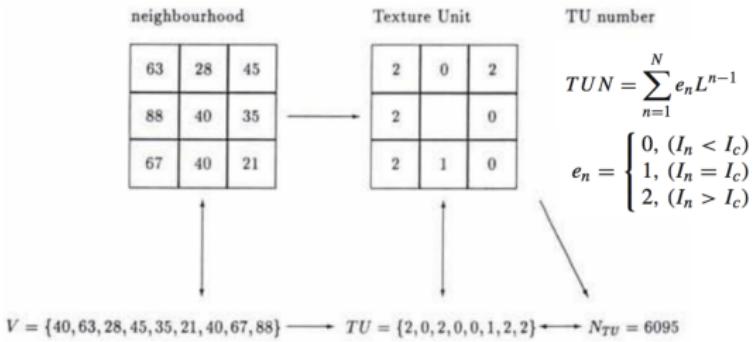
- Homogeneidad  $\sum_{i,j} \frac{C(i,j)}{1+|i-j|}$
- Energía  $\sum_{i,j} C(i,j)^2$
- Contraste  $\sum_{i,j} (i - j)^2 C(i,j)$
- Correlación  $\sum_{i,j} (ijC(i,j) - \mu_x\mu_y)/(\sigma_x\sigma_y)$
- Entropía  $-\sum_{i,j} C(i,j) \log C(i,j)$
- Haralick et al. (SMC 1973) definió 14 características estadísticas que capturan comportamiento de textura (homogeneidad, contraste, estructura organizada, complejidad, etc)
- Los estadísticos se agrupan en *feature vectors* para caracterizar diferentes texturas.

# Locally Binary Pattern

- LBP es un operador de textura, simple y eficiente, que etiqueta los pixels de una imagen mediante el umbral en la vecindad de cada pixel y su resultado es un número binario.
- LBP es un método que unifica los modelos estadísticos y estructurales tradicionalmente divergentes del análisis de textura.
- LBP es invariante a variaciones de iluminación. Esta es una de las propiedades más importantes en aplicaciones a imágenes del mundo real.
- Su simplicidad computacional permite analizar imágenes en entornos desafiantes en tiempo real.

## Local Binary Patterns

- Introducido en 'Texture Classification using Texture Spectrum', Wang, 1990.
- **Texture units:** unidades pequeñas esenciales en las que se descompone una textura.
- Un Texture Unit se representa por 8 elementos.
- Cada elemento tiene 3 posibles valores (0, 1 o 2) según la comparación del elemento central con sus vecinos.



## Local Binary Pattern

- Los Texture Units captan aspectos locales de la textura.
- Total:  $3^8 = 6561$  Texture Units
- Texture Spectrum: distribución de los Texture Units

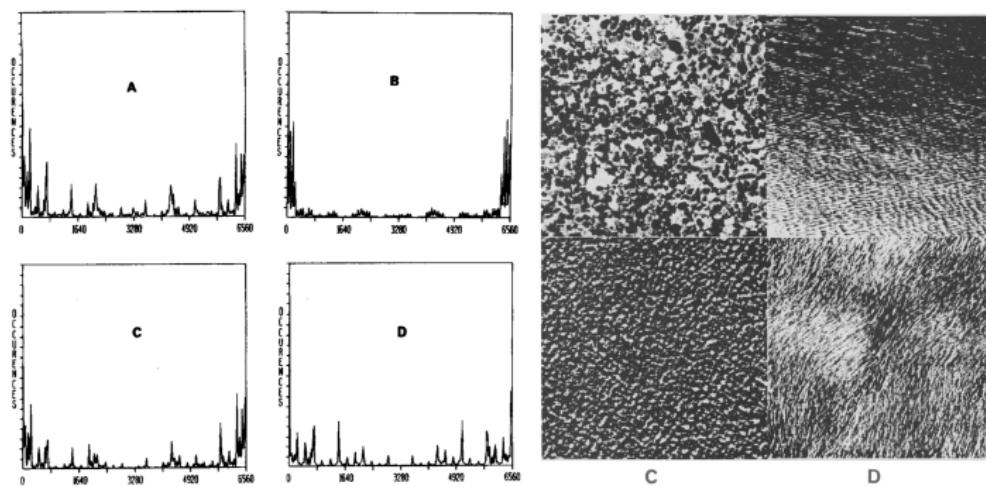


Fig. 2. Example of Texture Spectra calculated from the images of Fig. 3.

Fig. 3. Four of Brodatz' texture images: (A) beach sand, (B) water, (C) pressed cork and (D) for hide of unborn calf.

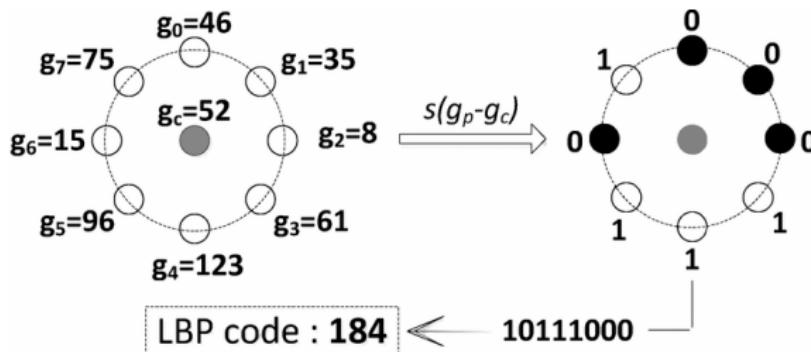
Texturas

## Local Binary Pattern

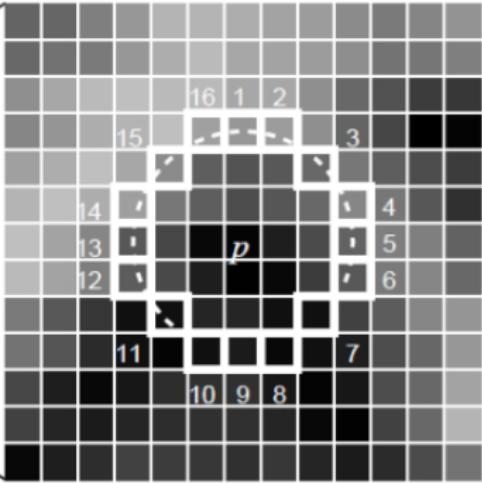
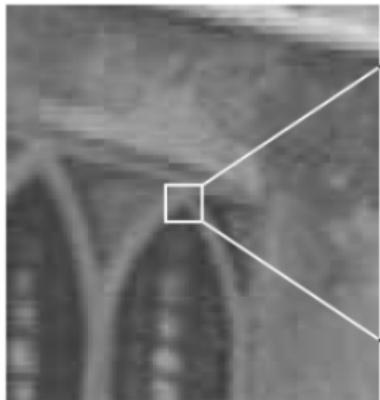
$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p$$

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

- $g_c$ : pixel central
- $g_p$ : valor de su  $p$ -ésimo pixel vecino
- $p = 0, \dots, P - 1$
- $P$ : número total de píxeles vecinos
- $R$ : radio de la vecindad



Ojala, Pietikainen, Harwood, "A comparative study of texture measures with classification based on feature distributions". Pattern Recognition, 1996



### Texturas

## Ejemplo del cálculo de un LBP

The original LBP

6	5	2
7	6	1
9	8	7

example

thresholded

1	0	0
1		0
1	1	1

weights

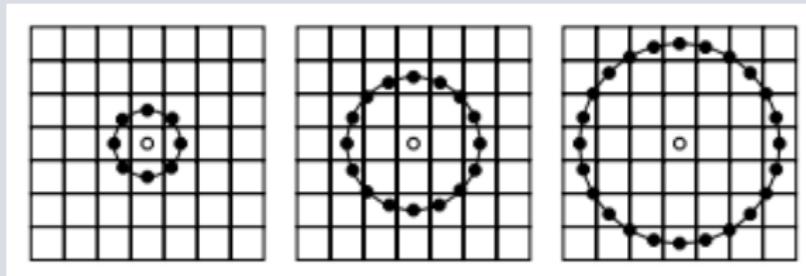
1	2	4
128		8
64	32	16

$$\text{Pattern} = 11110001$$

$$\begin{aligned}\text{LBP} &= 1 + 16 + 32 + 64 + 128 = 241 \\ \text{C} &= (6+7+9+8+7)/5 - (5+2+1)/3 = 4.7\end{aligned}$$

En un vecindario de  $3 \times 3$ , se utiliza el valor central como umbral.  
El vecindario consta de 8 pixels, se pueden obtener un total de  $2^8 = 256$  patrones diferentes.

## Diferentes vecindarios circulares



- 8-vecinos con radio  $R = 1$
- 16-vecinos con radio  $R = 2$
- 24-vecinos con radio  $R = 3$

## Definición de LBP

Dada una imagen  $I$ , el LBP para el pixel  $I(x, y)$  viene dado por:

$$LBP_{P,R}(x, y) = \sum_{n=0}^{P-1} S(I_n - I(x, y))2^n$$

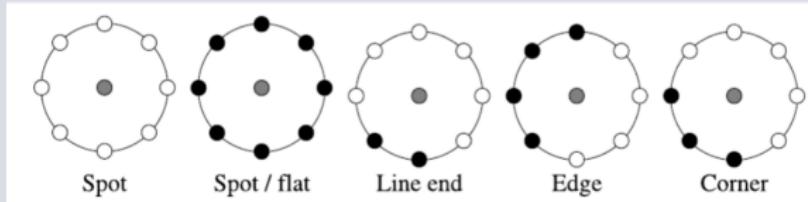
donde:

$R$  es el radio,  $I_n$ ,  $n = 0, \dots, P - 1$ , son los valores de los vecinos de  $I(x, y)$  y la función umbral  $S(z)$  está definida por:

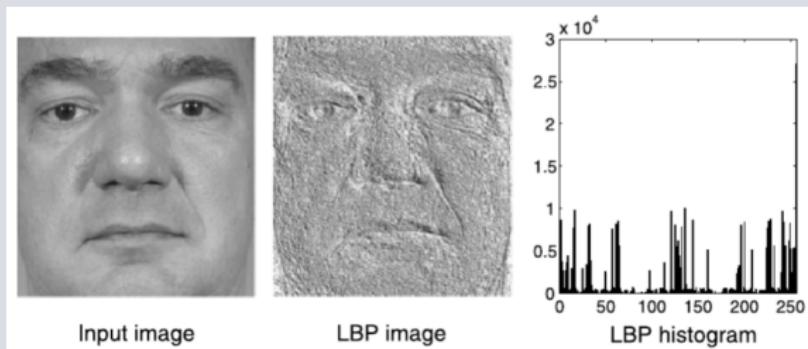
$$S(z) = \begin{cases} 1 & \text{si } z \geq 0 \\ 0 & \text{si } z < 0 \end{cases}$$

# LBP

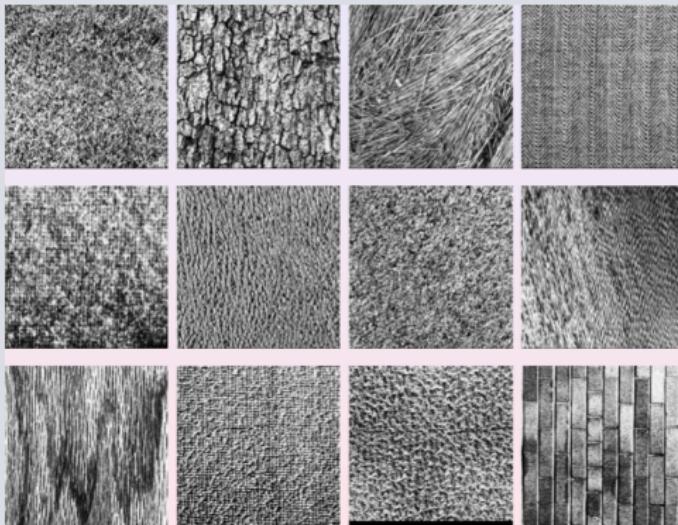
## Primitivas detectadas con LBP



## Ejemplo en una imagen

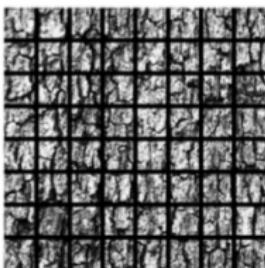
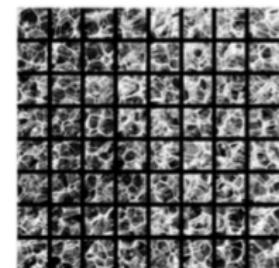
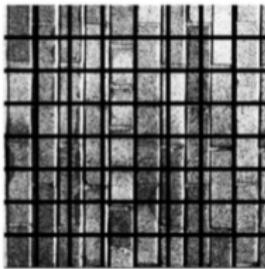
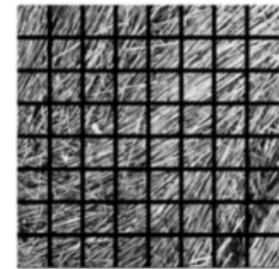


## Cómo discriminar texturas usando LBP?



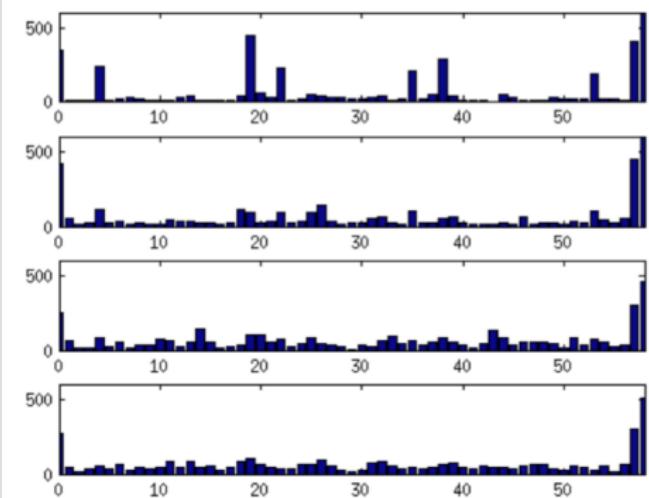
# LBP

## Muestras de 4 texturas diferentes



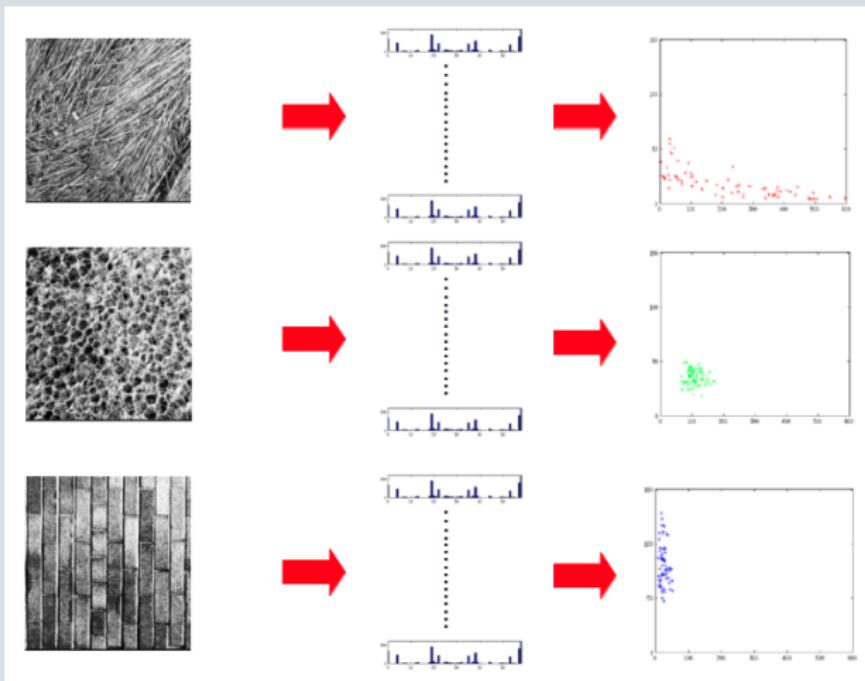
Texturas

## Histogramas de LBP de cada textura



Texturas

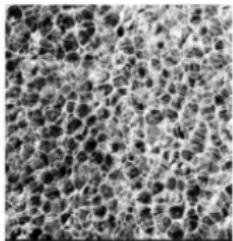
## 3 texturas diferentes, los LBPs y sus scatter plots



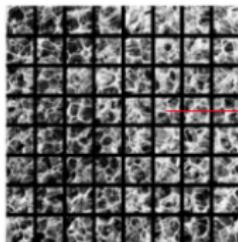
Texturas

# LBP

## Generación de histogramas de LBP para una textura



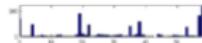
texture  $t_i$



$n$  samples  $S_i$

$s_{ij}$

$s_{i,f}$

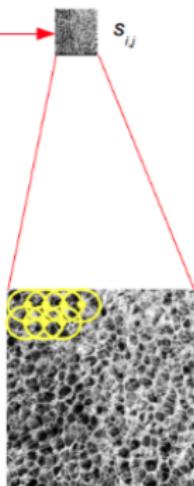


doing the same  
for all the samples  $s_{ij}$

$s_{i,n}$



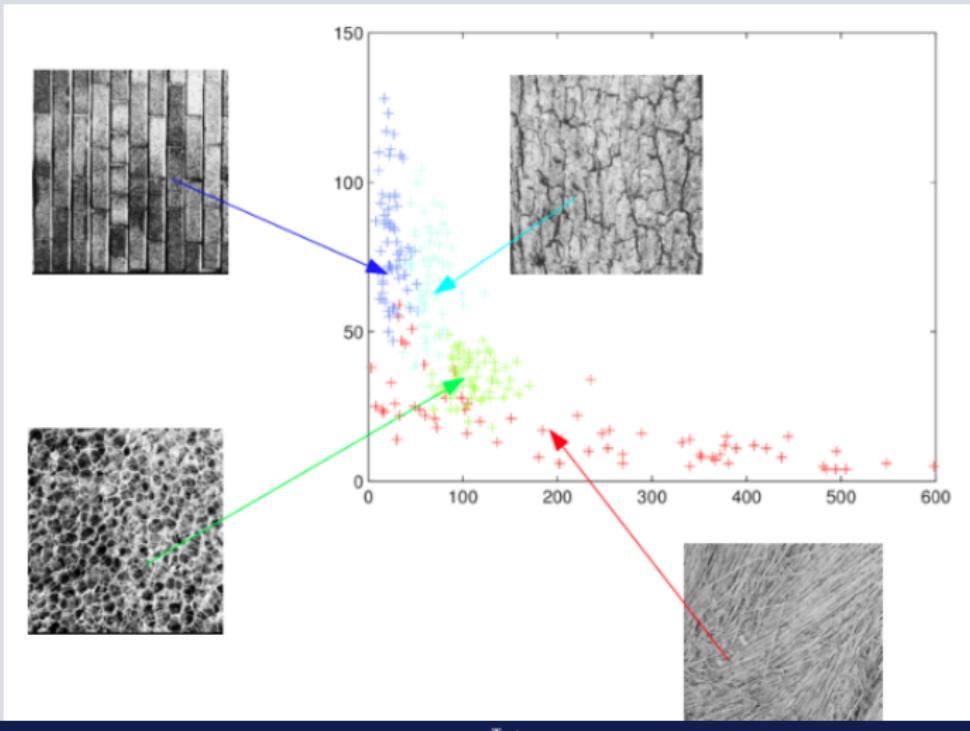
LBP pattern histogram  $lbp_{ij}$



LBP pattern calculation  
for each pixel  $p$  in  $s_{ij}$

Texturas

## Plot de histogramas de los LBPs correspondientes a 4 texturas



## Ejercicio LBP

Programar en cualquier lenguaje:

- para cada textura  $t_i \in \{t_1, \dots, t_k\}$ :
  - tomar muestras  $S_i = \{s_{i,1}, \dots, s_{i,n}\}$ ,  $i \in \{1, \dots, k\}$
  - para cada muestra  $s_{i,j} \in S_i$ ,  $j \in \{1, \dots, n\}$ 
    - calcular el LBP para cada pixel  $p \in s_{i,j}$
    - calcular el histograma LBP  $lbp_{i,j}$
- clasificar las texturas

## Invariancia a Rotaciones

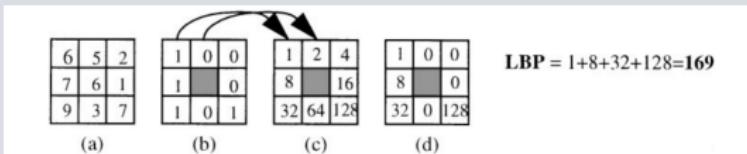


Fig. 2. Computation of local binary pattern (LBP).

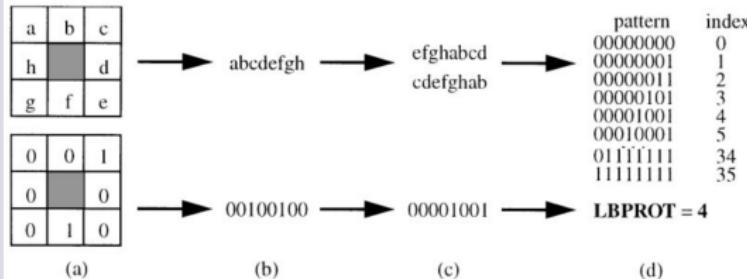
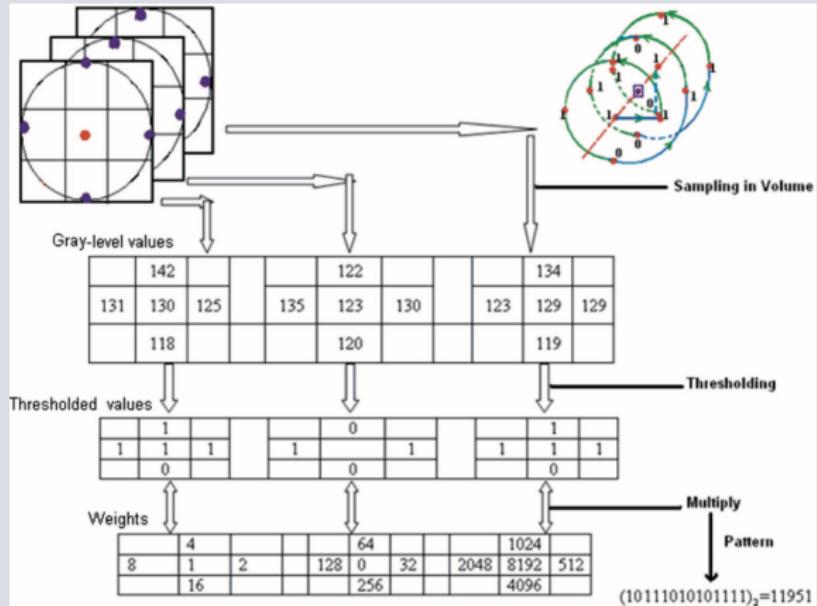


Fig. 3. Computation of LBPROT, rotation-invariant version of LBP.

M Pietikainen, T Ojala, Z Xu, "Rotation-invariant texture classification using feature distributions", In Pattern Recognition, 2000.

# VLBP

## Volume LBP



Zhao, Pietikainen, "Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions", IEEE Trans. of Pattern Analysis and Machine Intelligence, 2007.

## Local Binary Patterns from Three Orthogonal Planes

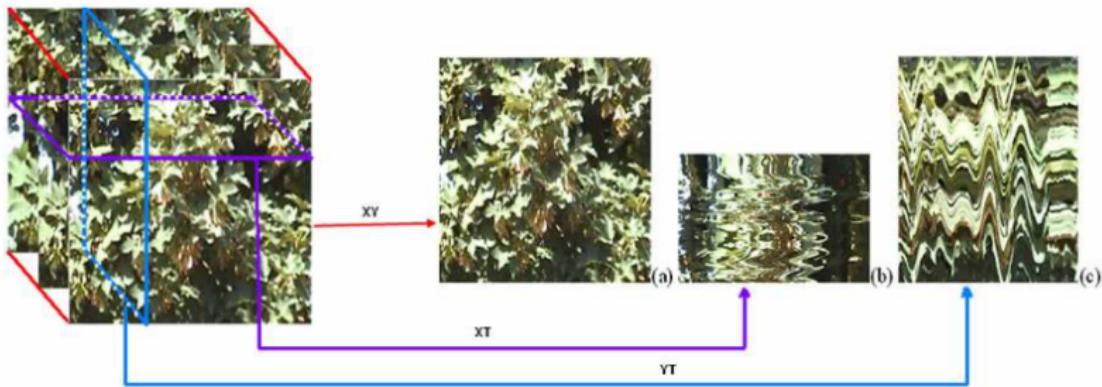


Fig. 4. (a) Image in XY plane ( $400 \times 300$ ) (b) Image in XT plane ( $400 \times 250$ ) in  $y = 120$  (last row is pixels of  $y = 120$  in first image) (c) Image in TY plane ( $250 \times 300$ ) in  $x = 120$  (first column is the pixels of  $x = 120$  in first frame).

# LBP-TOP

## Local Binary Patterns from Three Orthogonal Planes

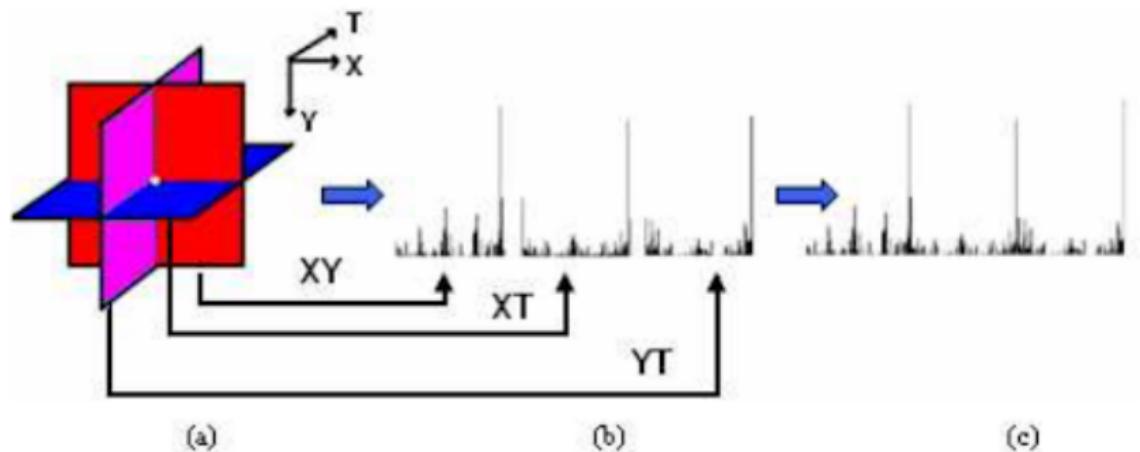


Fig. 5. (a) Three planes in dynamic texture (b) LBP histogram from each plane (c) Concatenated feature histogram.

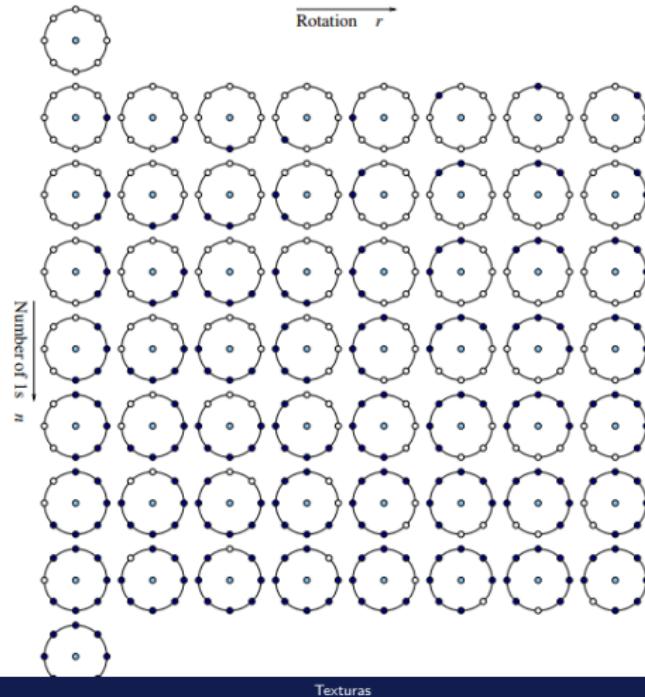
## Otras variantes y descriptores binarios

- BRIEF (Binary Robust Independent Elementary Features)
- FAST (Features from Accelerated Segment Test)
- ORB (Oriented FAST and Rotated BRIEF)
- BRISK (Binary Robust and Invariant Scalable Keypoints)
- FREAK (Fast Retina Keypoint)
- FRIF (Fast Robust Invariant Feature)

# Ejemplo de aplicación

## LBP y Descriptores de Fourier

## 58 LBPs uniformes para una 8-vecindad ( $U_P(n, r)$ )



# Definiciones

## Definición de LBP uniformes

Una medida de uniformidad para los LBPs está dada por el número de transiciones de  $0 \rightarrow 1$  o de  $1 \rightarrow 0$  que posee.

$$Unif(LBP_{P,R}(x,y)) = |S(I_{P-1} - I(x,y)) - S(I_0 - I(x,y))| + \\ \sum_{n=1}^{P-1} |S(I_n - I(x,y)) - S(I_{n-1} - I(x,y))|$$

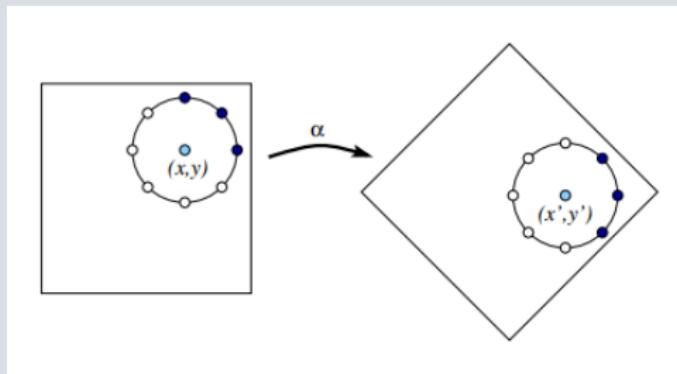
Si el número es menor o igual que dos, estos patrones son considerados uniformes.

## Patrones uniformes

Consideramos  $U_P(n, r)$  los patrones uniformes

- $n$  es el número de bits en el patrón, correspondiente a las filas.
- $r$  es la rotación del patrón, correspondiente a las columnas.
- $0 \leq n \leq P + 1$  para  $P$  elementos del vecindario.
- Si  $1 \leq n \leq P - 1$ , entonces la rotación  $r$  está en el rango  $0 \leq r \leq P - 1$ .
- Si  $n = P + 1$ , se corresponde a la clase de los patrones no uniformes.

# LBP y descriptores de Fourier. LBP-HF



- Patrón de la vecindad de un pixel  $(x, y)$ , el patrón rota junto con el pixel.
- $I^{\alpha^\circ}(x, y)$  es la rotación de la imagen  $I(x, y)$  en  $\alpha$  grados.
- $\alpha^\circ = a \frac{360^\circ}{P}$ ,  $a = 0, \dots, P - 1$
- $U_P(n, r)$  es reemplazado por el patrón uniforme  $U_P(n, r + a \bmod P)$  en el punto  $(x, y)$  de la imagen rotada.

## Histogramas LBP

- $h_I(U_P(n, r))$ : histograma LBP uniforme.
- El valor de  $h_I$  en  $U_P(n, r)$  es el número de ocurrencias del patrón uniforme  $U_P(n, r)$  en la imagen  $I$ .
- La rotación causa un desplazamiento circular en el histograma a lo largo de cada fila:

$$h_{I^\alpha} (U_P(n, r + a)) = h_I(U_P(n, r))$$

## DFT

- Las rotaciones inducen un desplazamiento en la representación polar  $(P, R)$  de la vecindad.
- Consideramos la clase de features que son invariantes a rotaciones de la imagen, y se calculan en los histogramas de cada fila, que son invariantes a desplazamiento.
- Usamos la Transformada Discreta de Fourier (DFT) para construir estos features.
- $H_n(\cdot)$  es la DFT de la  $n$ -ésima fila del histograma  $h_I(U_P(n, r))$

$$H_n(u) = \sum_{r=0}^{P-1} h_I(U_P(n, r)) e^{-2\pi i u r / P}$$

## DFT

- El desplazamiento en el vector causa un desplazamiento de fase en la DFT.
- Si  $h'_I(U_P(n, r)) = h_I(U_P(n, r - a))$  entonces,

$$H'_n(u) = H_n(u)e^{-2\pi iua/P}$$

- 

$$\begin{aligned} H'_{n_1}(u)\overline{H'_{n_2}(u)} &= H_{n_1}(u)e^{-2\pi iua/P} \overline{H_{n_2}(u)}e^{2\pi iua/P} \\ &= H_{n_1}(u) \overline{H_{n_2}(u)} \end{aligned}$$

donde  $1 \leq n_1, n_2 \leq P - 1$  y  $\overline{H_{n_2}(u)}$  es el complejo conjugado de  $H_{n_2}(u)$ .

- Para  $1 \leq n_1, n_2 \leq P - 1$  y  $0 \leq u \leq P - 1$  los features:

$$LBP^{u^2}HF(n_1, n_2, u) = H_{n_1}(u) \overline{H_{n_2}(u)}$$

son invariantes a desplazamientos circulares de las filas de  $h_I(U_P(n, r))$  y a rotaciones de la imagen  $I(x, y)$

- La magnitud

$$|H_n(u)| = \sqrt{H_n(u) \overline{H_n(u)}}$$

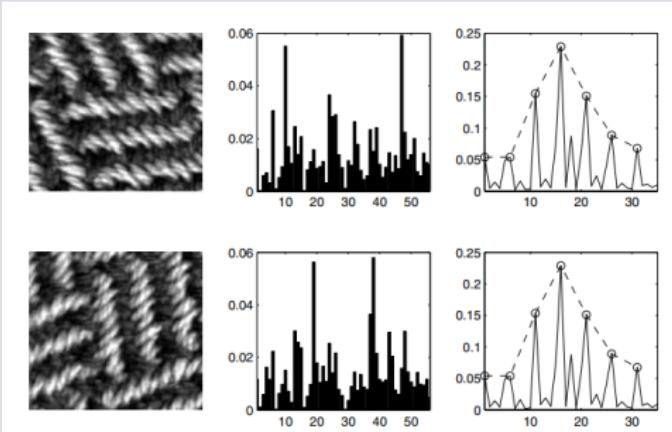
se considera como un caso especial de estos features

# LBP-HF

Feature Vector de tamaño  $1 \times ((P - 1)(P/2 + 1) + 3)$

$$\begin{aligned}fv_{LBP-HF} = [ & |H(1, 0)|, \dots, |H(1, P/2)|, \\ & \dots, \\ & |H(P - 1, 0)|, \dots, |H(P - 1, P/2)|, \\ & h(U_P(0, 0)), h(U_P(P, 0)), h(U_P(P + 1, 0)) ]\end{aligned}$$

# LBP-HF



Los histogramas de LBP de features de Fourier son invariantes a rotaciones, y se basan en los histogramas de LBPs uniformes.

## Bibliografía

- [AH09] T Ahonen and J. Matas and C. He y M. Pietikainen, Rotation Invariant Image Description with Local Binary Pattern Histogram Fourier Features, Scandinavian Conference on Image Analysis (SCIA), 2009.
- [OJ96] T. Ojala and M. Pietikainen and D. Harwood, A comparative study of texture measures with classification based on feature distributions, *Pattern Recognition*, 29, 1, 51-59, 1996.
- [AH06] T. Ahonen and A. Hadid and M. Pietikainen, Face Description with Local Binary Patterns: Application to Face Recognition, *IEEE Transactions on pattern Analysis and Machine Intelligence*, 28, 12, 2006.
- [PI11] M. Pietikainen and A. Hadid and G. Zhao and T. Ahonen, *Computer Vision Using Local Binary Patterns*, Springer, 2011.
- [HE09] M. Heikkila and M. Pietikainen and C. Schmid, Description of Interest Regions with Local Binary Patterns, *Pattern Recognition*, Vol. 42, 3, pp. 425?436?, 2009.