RECONOCIMIENTO DE PATRONES EN IMÁGENES TICS 585

FACULTAD DE INGENIERÍA Y CIENCIAS
UNIVERSIDAD ADOLFO IBÁÑEZ

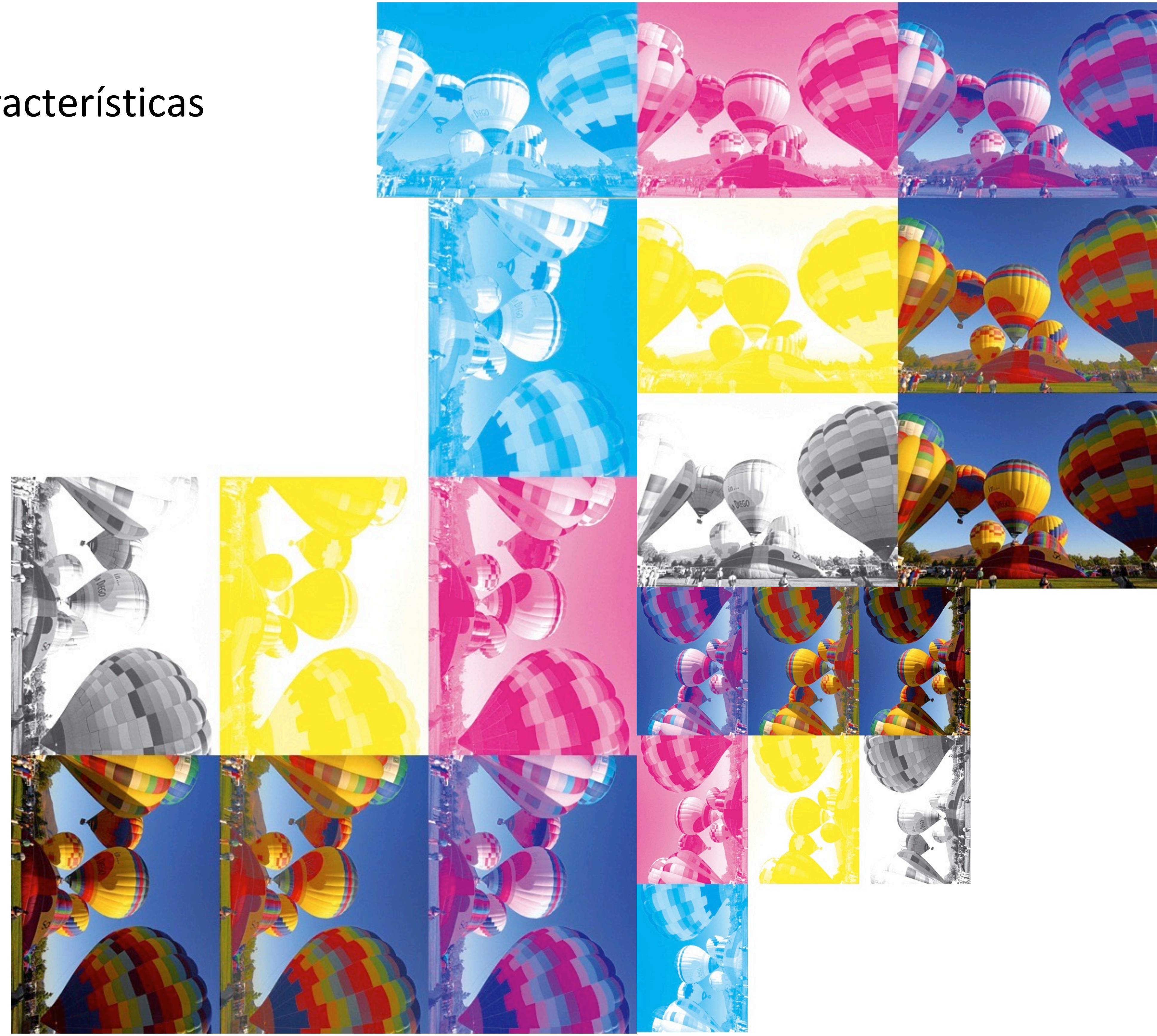
SEGUNDO SEMESTRE 2021

PROFESOR: MIGUEL CARRASCO

CARACTERÍSTICAS CROMÁTICAS

FACULTAD DE INGENIERÍA Y CIENCIAS

- Extracción de características
  - Geométricas
  - Cromáticas
  - Otras



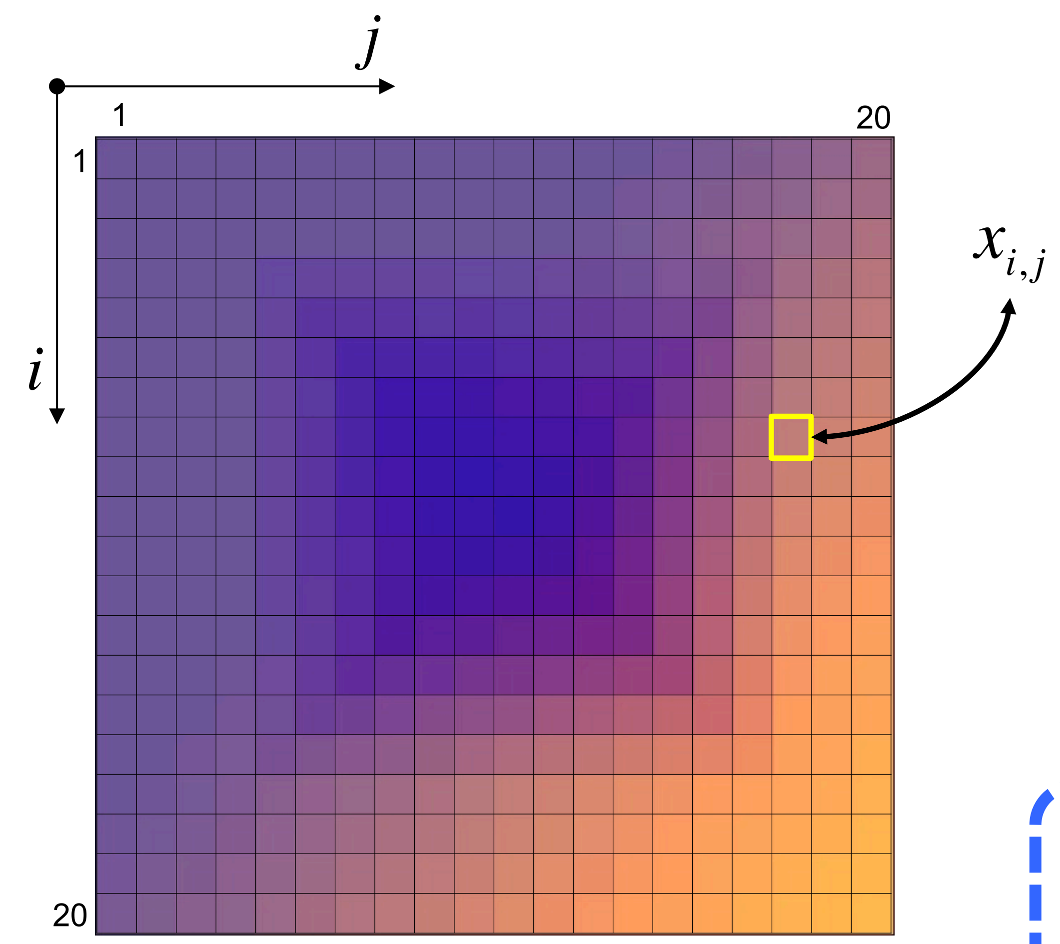
FACULTAD DE INGENIERÍA Y CIENCIAS
UNIVERSIDAD ADOLFO IBÁÑEZ

- Geométricas
  - Centro de masa
  - Tamaño
  - Perímetro
  - Redondez
  - Momentos binarios
  - Descriptores de Fourier
  - Elipes
  - Distancia al borde
- Cromáticas
  - Color promedio
  - Gradiente promedio
  - Promedio segunda derivada
  - Contraste
  - Momentos de color
  - Textura
- Otros descriptores
  - PHOG
  - SURF
  - SIFT
  - GPD

Características cromáticas

- **Color promedio**
- Gradiente promedio
- Promedio 2da derivada
- Contraste
- Momentos
- Textura

- El color promedio determina el promedio de una región según su componente de color. Recordemos que una imagen a color está compuesta por tres matrices R (rojo), G (verde) y B (azul).



Cada píxel de la imagen a color está compuesto por tres valores según su canal. Rojo, Verde o Azul. (RGB).

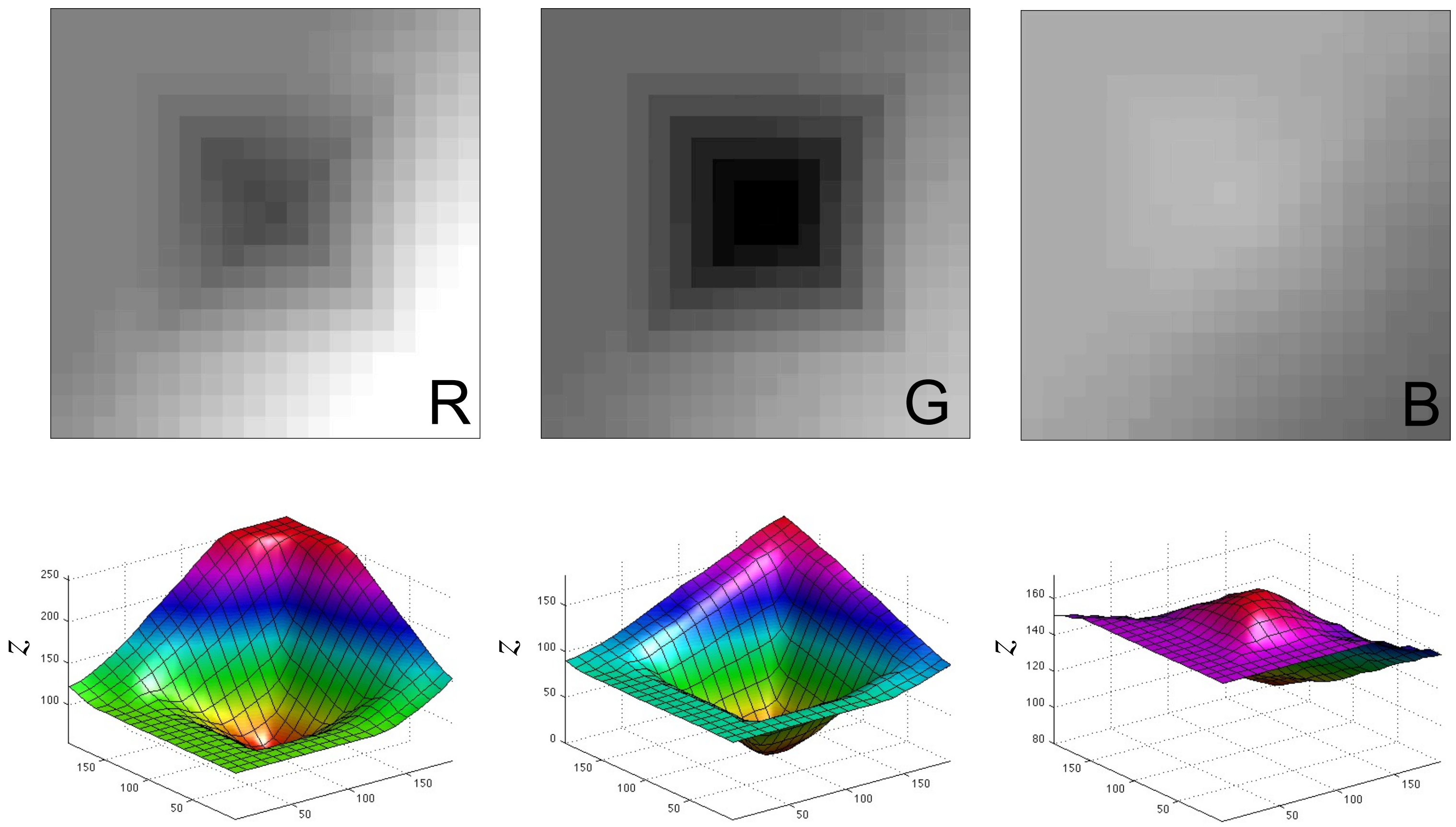
Recuerde que cada píxel tiene una posición $\{i,j\}$

Características cromáticas

- **Color promedio**
- Gradiente promedio
- Promedio 2da derivada
- Contraste
- Momentos
- Textura

- El color promedio determina el promedio de una región según su componente de color. Recordemos que una imagen a color está compuesta por tres matrices R (rojo), G (verde) y B (azul).

Componentes RGB de la imagen a color



Cada componente RGB posee un rango de valores entre 0 a 255 (8bits)

Visualización 3D de la escala de grises.
El eje Z corresponde al nivel de gris

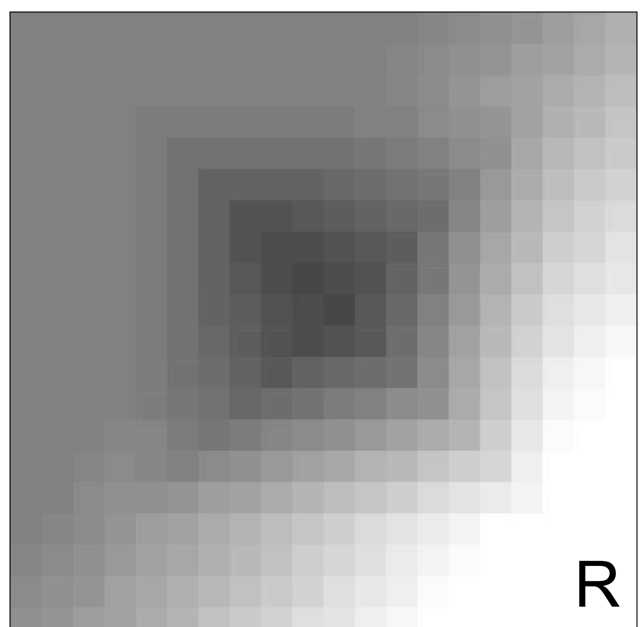
FACULTAD DE INGENIERÍA Y CIENCIAS

UAI UNIVERSIDAD ADOLFO IBÁÑEZ

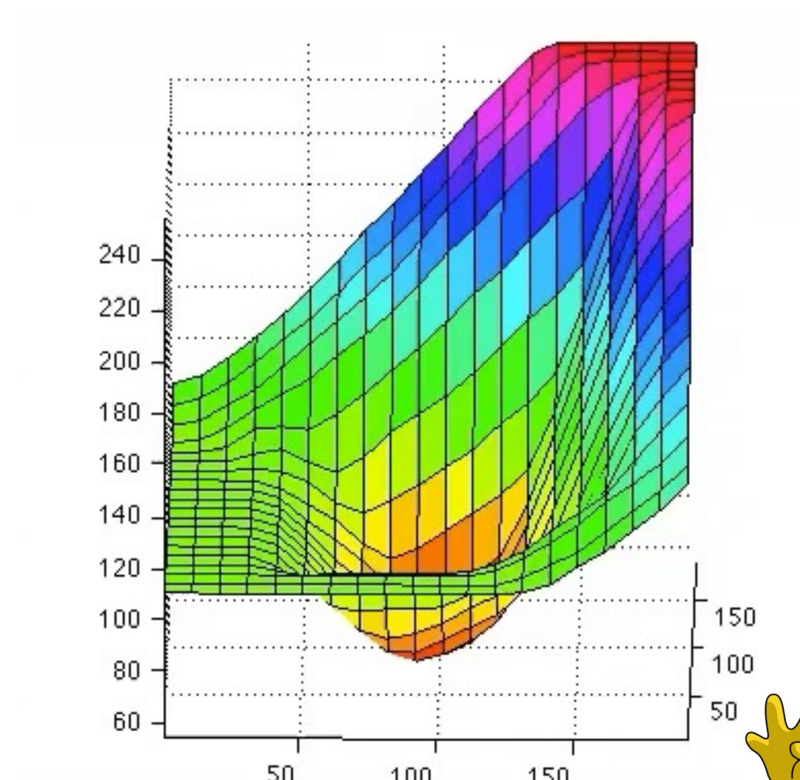
Características cromáticas

- **Color promedio**
- Gradiente promedio
- Promedio 2da derivada
- Contraste
- Momentos
- Textura

- El color promedio determina el promedio de una región según su componente de color. Recordemos que una imagen a color está compuesta por tres matrices R (rojo), G (verde) y B (azul).



Los valores más oscuros están más abajo



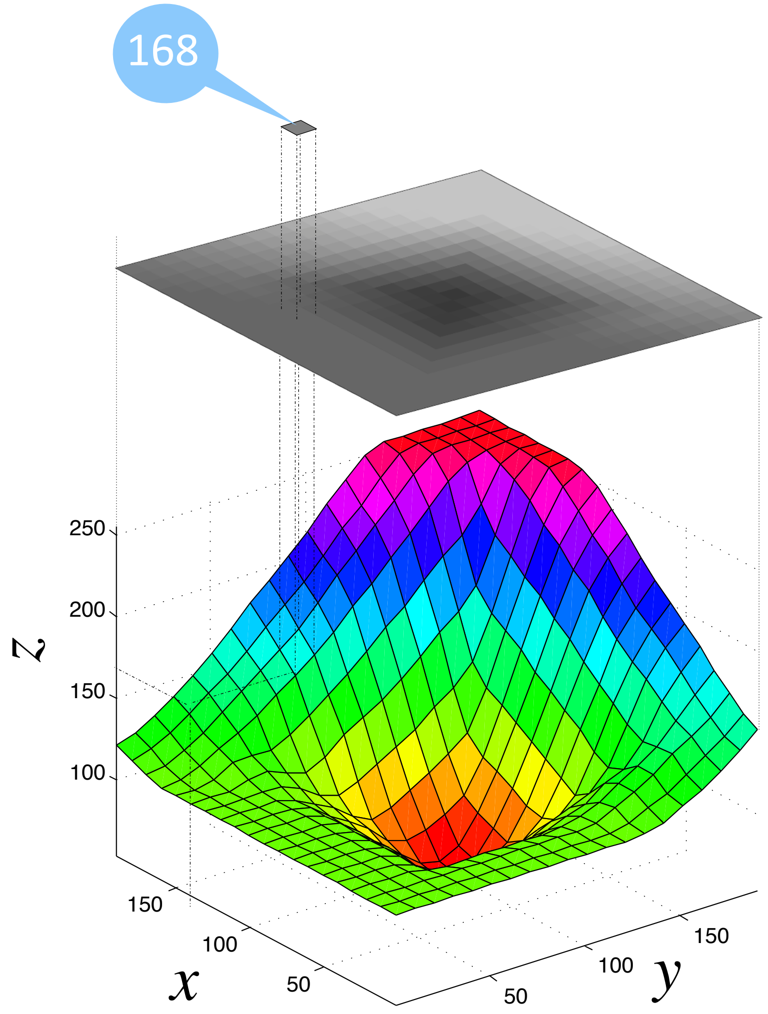
Animación 3D

FACULTAD DE INGENIERÍA Y CIENCIAS

Características cromáticas

- **Color promedio**
- Gradiente promedio
- Promedio 2da derivada
- Contraste
- Momentos
- Textura

- El color promedio determina el promedio de una región según su componente de color. Recordemos que una imagen a color está compuesta por tres matrices R (rojo), G (verde) y B (azul).



Definición

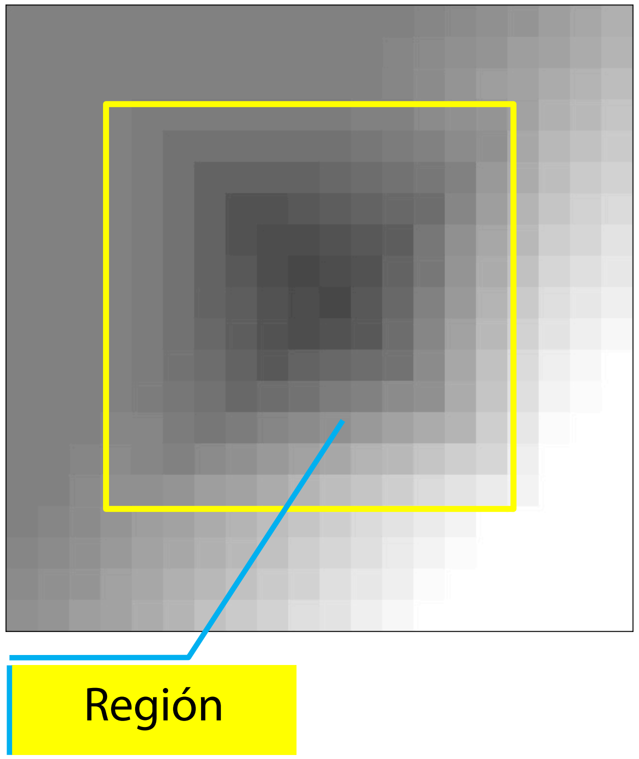
$$D = \frac{1}{A} \sum_{i,j \in R} x_{i,j}$$

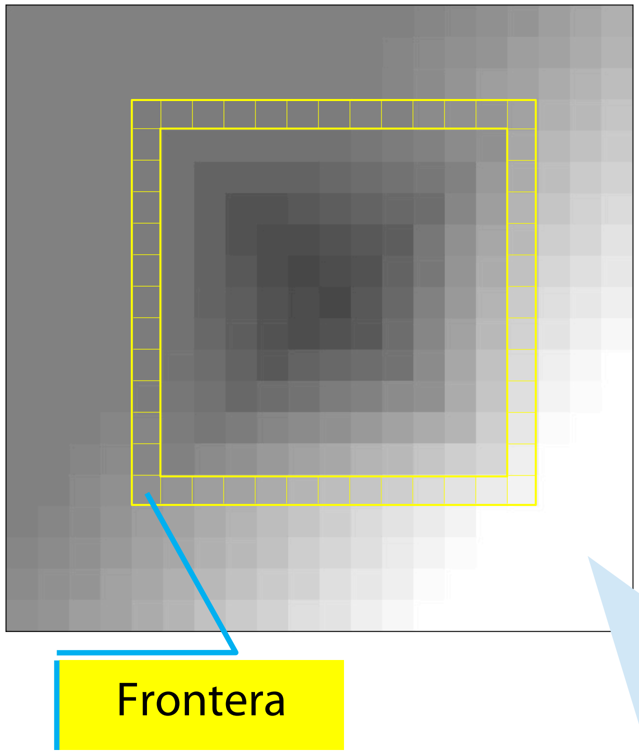
```
def color_promedio(im):

    b, g, r = cv2.split(im)
    area = r.shape[0]*r.shape[1]
    prom_r = np.sum(r)/area
    prom_g = np.sum(g)/area
    prom_b = np.sum(b)/area

    return   prom_r, prom_g, prom_b
```

- Características cromáticas

  - Color promedio
  - Gradiente promedio
  - Promedio 2da derivada
  - Contraste
  - Momentos
  - Textura

- El gradiente promedio es una característica que toma el gradiente de la variable de color en el frontera de la región. A través de esta característica podemos medir cuan abrupto es el cambio de color en la región





Este proceso lo realizamos por cada canal de la imagen a color.

Características cromáticas

- Color promedio
- **Gradiente promedio**
- Promedio 2da derivada
- Contraste
- Momentos
- Textura

- El gradiente promedio es una característica que toma el gradiente de la variable de color en el frontera de la región.

j →, i ↓

$i \backslash j$	1	2	3	4	5	6	7	8
1	2	1	2	3	2	1	4	
2	4	29	23	25	26	24	3	
3	3	28	23	19	13	27	3	1
4	0	28	18	12	17	18	25	4
5	1	27	22	45	45	23	29	2
6	0	1	32	31	21	12	25	3
7		2	4	31	28	29	30	4
8			5	1	1	3	3	6

$L(i,j)$

Supongamos la siguiente región y sus niveles de gris

Definición

$$L_x(i,j) = -\frac{1}{2}\cdot L(i-1,j) + 0\cdot L(i,j) + \frac{1}{2}\cdot L(i+1,j)$$

$$L_y(i,j) = -\frac{1}{2}\cdot L(i,j-1) + 0\cdot L(i,j) + \frac{1}{2}\cdot L(i,j+1)$$

Magnitud

$$\nabla L(i,j) = \sqrt{L_x(i,j)^2 + L_y(i,j)^2}$$

Descriptor

$$D = \frac{1}{N}\sum_{i,j\in L} \nabla L(i,j)$$

N : número de píxeles de la frontera L

- Características cromáticas

Color promedio

Gradiente promedio

Promedio 2da derivada

Contraste

Momentos

Textura

- A través de esta característica podemos medir cuan abrupto es el cambio de color en la región

j →, i ↓

	1	2	3	4	5	6	7	8
1	2	1	2	3	2	1	4	
2	4	29	23	25	26	24	3	
3	3	28	23	19	13	27	3	1
4	0	28	18	12	17	18	25	4
5	1	27	22	45	45	23	29	2
6	0	1	32	31	21	12	25	3
7		2	4	31	28	29	30	4
8			5	1	1	3	3	6

Supongamos la siguiente región y sus niveles de gris

Definiciones

$$L_x(i,j) = -\frac{1}{2}\cdot L(i-1,j) + 0\cdot L(i,j) + \frac{1}{2}\cdot L(i+1,j)$$

$$L_y(i,j) = -\frac{1}{2}\cdot L(i,j-1) + 0\cdot L(i,j) + \frac{1}{2}\cdot L(i,j+1)$$

$$\nabla L(i,j) = \sqrt{L_x(i,j)^2 + L_y(i,j)^2}$$

Ejemplo

$$L_x(2,2) = -\frac{1}{2}(1) + \frac{1}{2}\cdot(28) = 13.5$$

$$L_y(2,2) = -\frac{1}{2}(4) + \frac{1}{2}\cdot(23) = 9.5$$

$$\nabla L(2,2) = \sqrt{13.5^2 + 9.5^2} = 16.51$$

Características cromáticas

- Color promedio
- **Gradiente promedio**
- Promedio 2da derivada
- Contraste
- Momentos
- Textura

- A través de esta característica podemos medir cuan abrupto es el cambio de color en la región

i \ j	1	2	3	4	5	6	7	8
1	2	1	2	3	2	1	4	
2	4	29	23	25	26	24	3	
3	3	28	23	19	13	27	3	1
4	0	28	18	12	17	18	25	4
5	1	27	22	45	45	23	29	2
6	0	1	32	31	21	12	25	3
7		2	4	31	28	29	30	4
8			5	1	1	3	3	6

Recuerde que el gradiente sólo se determina en los píxeles de la región

Descriptor

$$D=\frac{1}{N}\sum_{i,j\in L}\nabla L(i,j)$$

Ejemplo

$$D=\frac{1}{N}\cdot\left(\nabla L(2,2)+\nabla L(2,3)+\cdots+\nabla L(3,2)\right)$$

$$D=\frac{1}{17}\cdot(16.51+10.69+\cdots+10.01)$$

$$D=11.675$$

El perímetro sólo tiene 17 pixeles.

Características cromáticas

- Color promedio
- **Gradiente promedio**
- Promedio 2da derivada
- Contraste
- Momentos
- Textura

- A través de esta característica podemos medir cuan abrupto es el cambio de color en la región

$j \rightarrow$, $i \downarrow$

i \ j	1	2	3	4	5	6	7	8
1	2	1	2	3	2	1	4	
2	4	29	23	25	26	24	3	
3	3	28	23	19	13	27	3	1
4	0	28	18	12	17	18	25	4
5	1	27	22	45	45	23	29	2
6	0	1	32	31	21	12	25	3
7		2	4	31	28	29	30	4
8			5	1	1	3	3	6

Recuerde que el gradiente sólo se determina en los píxeles de la región

Definiciones

$$L_x(i,j) = -\frac{1}{2}\cdot L(i-1,j) + 0\cdot L(i,j) + \frac{1}{2}\cdot L(i+1,j)$$

$$L_y(i,j) = -\frac{1}{2}\cdot L(i,j-1) + 0\cdot L(i,j) + \frac{1}{2}\cdot L(i,j+1)$$

$$\nabla L(i,j) = \sqrt{L_x(i,j)^2 + L_y(i,j)^2}$$

```python
def gradiente(i,j,L):

    Lx= -0.5*L[i-1,j]+0.5*L[i+1,j]
    Ly= -0.5*L[i,j-1]+0.5*L[i,j+1]
    out= sqrt(Lx**2+Ly**2)

    return out
```

- Características cromáticas

Color promedio

Gradiente promedio

Promedio 2da derivada

Contraste

Momentos

Textura

- A través de esta característica podemos medir cuan abrupto es el cambio de color en la región

	1	2	3	4	5	6	7	8
1	2	1	2	3	2	1	4	
2	4	29	23	25	26	24	3	
3	3	28	23	19	13	27	3	1
4	0	28	18	12	17	18	25	4
5	1	27	22	45	45	23	29	2
6	0	1	32	31	21	12	25	3
7		2	4	31	28	29	30	4
8			5	1	1	3	3	6

Recuerde que el gradiente sólo se determina en los píxeles de la región

Descriptor

$$D = \frac{1}{N} \sum_{i,j \in L} \nabla L(i,j)$$

```python
def gradiente_perimetro(I, J, im):
   #% Descriptor de gradiente perimetro
   #% promedio

    n = len(I)
    out = 0
    for i in range(0, n):
        out = out + gradiente(I[i],J[i],im)

    return float(out/n)
```

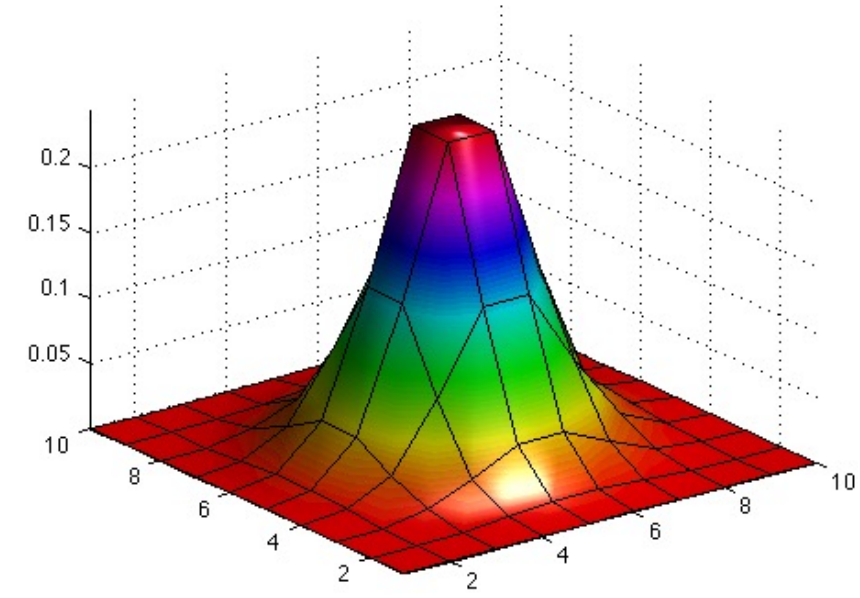
Características cromáticas

- Color promedio
- Gradiente promedio
- **Promedio 2da derivada**
- Contraste
- Momentos
- Textura

- El promedio de la segunda derivada nos permite medir si la región es más clara que su entorno.

Gaussiana bidimensional

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$




```
def gaussiana_3D(t, sigma):

    ventana = np.linspace(-t/2, t/2, t)
    u, v = np.meshgrid(ventana, ventana)

    G = (1/(sigma**2*2*pi))*np.exp(-(u**2+v**2)/(2*sigma**2))
    N = G/np.sum(G.flatten())  #normalizamos

    fig = plt.figure()
    ax  = plt.axes(projection='3d')
    ax.plot_surface(u, v, N, cmap='Spectral')
    plt.show()
```

El comando `meshgrid` crea una grilla completando todos los valores de X e Y

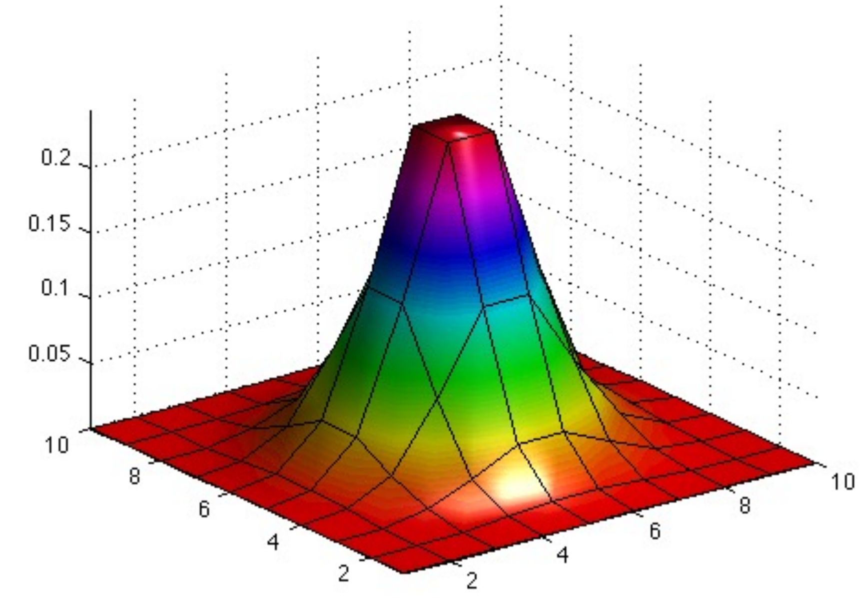
Características cromáticas

- Color promedio
- Gradiente promedio
- Promedio 2da derivada
- Contraste
- Momentos
- Textura

- Comencemos nuestro análisis con una gaussiana en dos dimensiones.

Gaussiana bidimensional

$$G_\sigma(x,y)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$




```
def gaussiana_3D(t, sigma):

    ventana = np.linspace(-t/2, t/2, t)
    u, v = np.meshgrid(ventana, ventana)

    G = (1/(sigma**2*2*pi))*np.exp(-(u**2+v**2)/(2*sigma**2))
    N = G/np.sum(G.flatten())  #normalizamos
```

Dado que tenemos todos los valores para x, e y podemos hacer el cálculo para todos los puntos

- Características cromáticas

Color promedio

Gradiente promedio

Promedio 2da derivada

Contraste

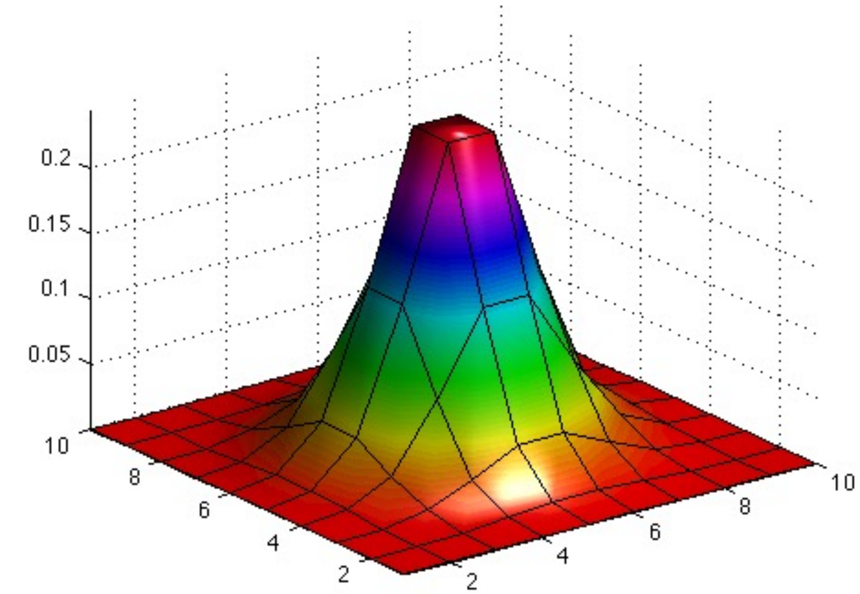
Momentos

Textura

- Comencemos nuestro análisis con una gaussiana en dos dimensiones.

Gaussiana bidimensional

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$




```
def gaussiana_3D(t, sigma):

    ventana = np.linspace(-t/2, t/2, t)
    u, v = np.meshgrid(ventana, ventana)

    G = (1/(sigma**2*2*pi))*np.exp(-(u**2+v**2)/(2*sigma**2))
    N = G/np.sum(G.flatten())  #normalizamos

    fig = plt.figure()
    ax  = plt.axes(projection='3d')
    ax.plot_surface(u, v, N, cmap='Spectral')
    plt.show()
```

Con estas opciones realizamos un ploteado "tuneado"

Características cromáticas

- Color promedio
- Gradiente promedio
- **Promedio 2da derivada**
- Contraste
- Momentos
- Textura

- Debido a que las imágenes digitales son muy sensibles al ruido, aplicamos un operador laplaciano a una gaussiana; filtro conocido como *LoG* (*Laplacian of Gaussian*)

Gaussiana bidimensional

$$G_\sigma(x,y)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

Si aplicamos el operador Laplaciano sobre una Gaussiana obtenemos el filtro LoG.

Operador Laplaciano (Formulación para dos variables)

$$L(x,y)=\nabla^2 G(x,y)=\frac{\partial^2 G(x,y)}{\partial^2 x}+\frac{\partial^2 G(x,y)}{\partial^2 y}$$

Operador Laplacian of Gausssian

$$LoG_\sigma(x,y)=-\frac{1}{\pi\sigma^4}\cdot\left(1-\frac{x^2+y^2}{2\sigma^2}\right)\cdot\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

Demostración
http://fourier.eng.hmc.edu/e161/lectures/gradient/node10.html

Características cromáticas

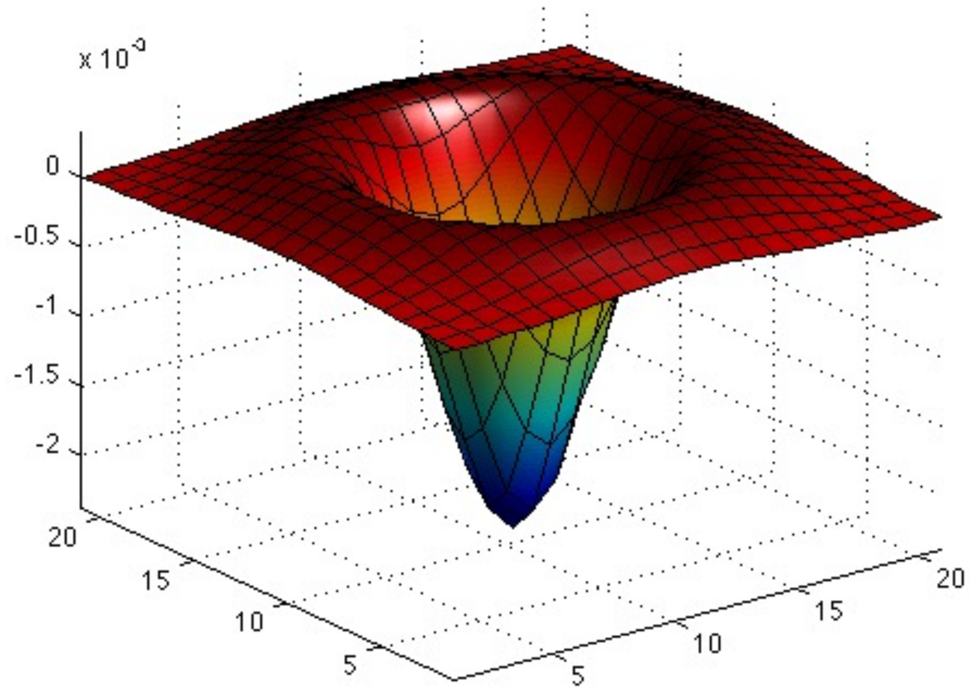
- Color promedio
- Gradiente promedio
- Promedio 2da derivada
- Contraste
- Momentos
- Textura

- Debido a que las imágenes digitales son muy sensibles al ruido, aplicamos un operador laplaciano a una gaussiana; filtro conocido como *LoG* (*Laplacian of Gaussian*)

Operador Laplacian of Gausssian

$$LoG_{\sigma}(x,y) = -\frac{1}{\pi\sigma^4}\cdot\left(1-\frac{x^2+y^2}{2\sigma^2}\right)\cdot\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

También es conocido como sobrero mexicano por sus semejanza



FACULTAD DE INGENIERÍA Y CIENCIAS

UNIVERSIDAD ADOLFO IBÁÑEZ

- Características cromáticas

Color promedio

Gradiente promedio

Promedio 2da derivada

Contraste

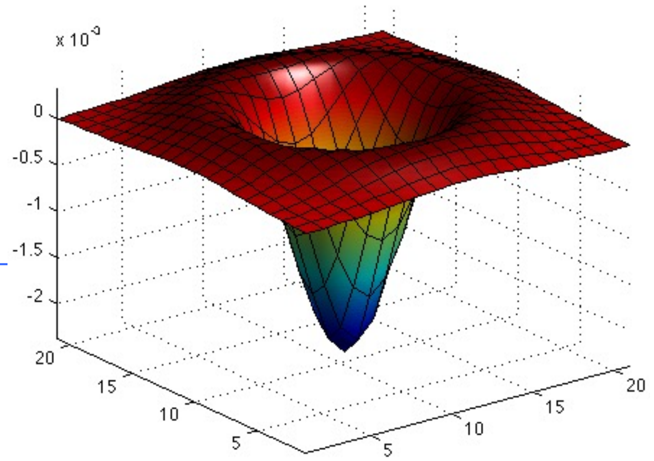
Momentos

Textura

- Debido a que las imágenes digitales son muy sensibles al ruido, aplicamos un operador laplaciano a una gaussiana; filtro conocido como *LoG* (*Laplacian of Gaussian*)

Operador Laplacian of Gausssian

$$LoG_\sigma(x,y) = -\frac{1}{\pi\sigma^4}\cdot\left(1-\frac{x^2+y^2}{2\sigma^2}\right)\cdot\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$




```
def laplaciano(t, sigma):

    ventana = np.linspace(-t/2, t/2, t)
    u,v = np.meshgrid(ventana, ventana)

    mat = (u**2 + v**2) /(2*sigma**2)
    L = (-1/(pi*sigma**4)) * (1-mat)*np.exp(-mat)

    fig = plt.figure()
    ax  = plt.axes(projection='3d')
    ax.plot_surface(u, v, L, cmap='Spectral')
    plt.show()
```

Características cromáticas

- Color promedio
- Gradiente promedio
- **Promedio 2da derivada**
- Contraste
- Momentos
- Textura

- El descriptor promedio de la segunda derivada aplicado en la región. Este proceso requiere una máscara que recorre la región píxel por píxel.

Centro de la máscara

j →, i ↓

i \ j	1	2	3	4	5	6	7	8
1	2	1	2	3	2	1	4	
2	4	29	23	25	26	24	3	
3	3	28	23	19	13	27	3	1
4	0	28	18	12	17	18	25	4
5	1	27	22	45	45	23	29	2
6	0	1	32	31	21	12	25	3
7		2	4	31	28	29	30	4
8			5	1	1	3	3	6

Ejemplo para la primera máscara

subregión

2	1	2
4	29	23
3	28	23

LoG (3x3)

0.27	0.68	0.27
0.68	-5.09	0.68
0.27	0.68	0.27

Multiplicamos punto por punto

$$I \bullet LoG = 2 \times 0.27 + \cdots + 29 \times -5.09 + \cdots + 23 \times 0.27$$

$$I \bullet LoG = -100.72$$

Cada resultado es almacenado en una nueva matriz sobre la cual calcularemos nuestro descriptor

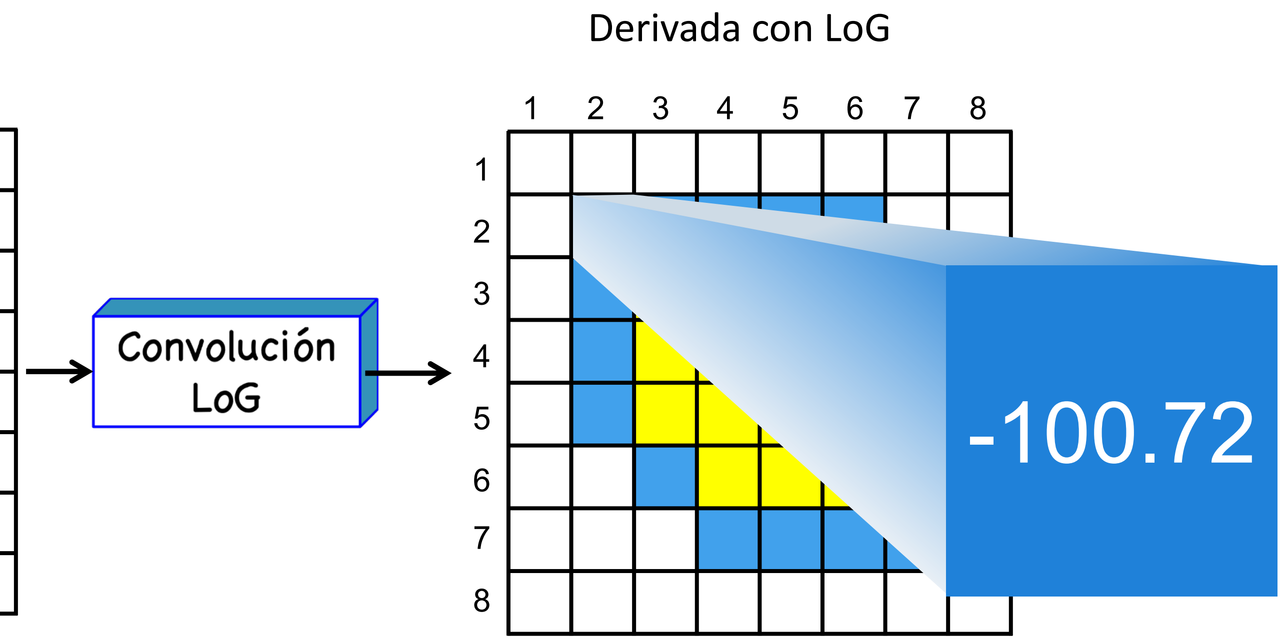
Características cromáticas

- Color promedio
- Gradiente promedio
- **Promedio 2da derivada**
- Contraste
- Momentos
- Textura

- Por cada máscara que empleamos generamos un nuevo resultado. Dicho valor corresponde a la derivada de segundo orden. En este caso, a Laplaciano de la Gaussiana.

Imagen original

	1	2	3	4	5	6	7	8
1	2	1	2	3	2	1	4	
2	4	29	23	25	26	24	3	
3	3	28	23	19	13	27	3	1
4	0	28	18	12	17	18	25	4
5	1	27	22	45	45	23	29	2
6	0	1	32	31	21	12	25	3
7		2	4	31	28	29	30	4
8			5	1	1	3	3	6



subregión

2	1	2
4	29	23
3	28	23

LoG (3x3)

0.27	0.68	0.27
0.68	-5.09	0.68
0.27	0.68	0.27

$$I \bullet LoG = 2 \times 0.27 + \cdots + 29 \times -5.09 + \cdots + 23 \times 0.27$$
$$I \bullet LoG = -100.72$$

Características cromáticas

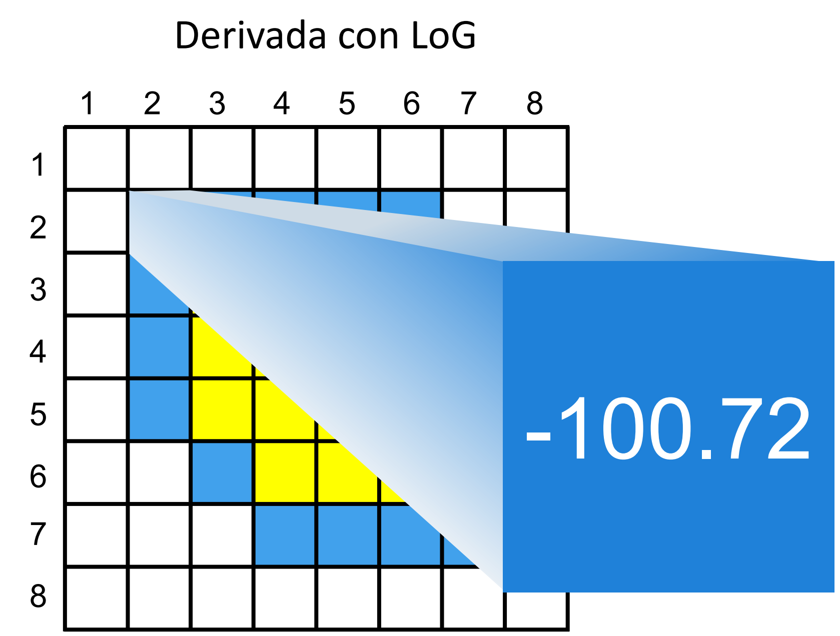
- Color promedio
- Gradiente promedio
- **Promedio 2da derivada**
- Contraste
- Momentos
- Textura

- Finalmente el descriptor calcula el promedio segunda derivada en cada píxel de la región

Descriptor

$$D = \frac{1}{A}\sum_{i,j\in R} x''(i,j) = \frac{1}{A}\sum_{i,j\in R} \underbrace{LoG(\sigma,N)}\ast x(i,j)$$

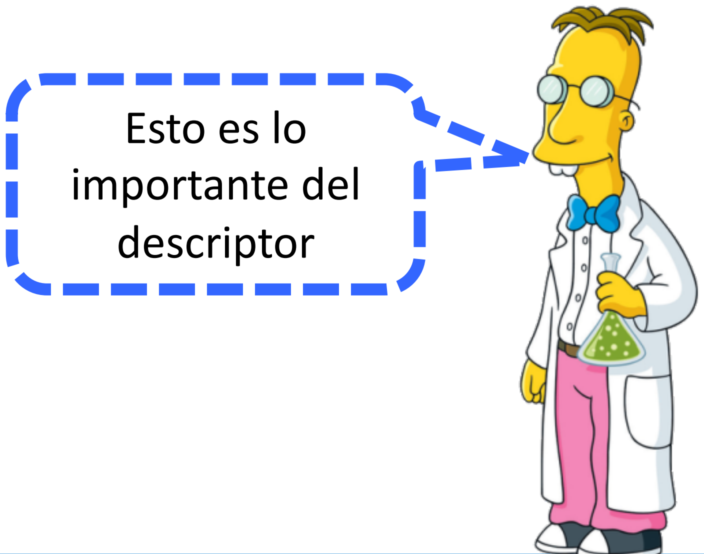
La máscara LoG es el filtro que se convoluciona con la imagen original



Resultados

si $D < 0$: La región es más oscura que su entorno

si $D > 0$: La región es más clara que su entorno



Características cromáticas

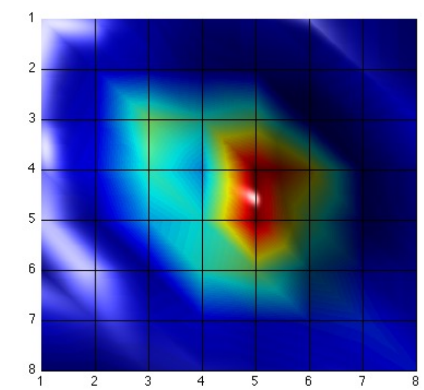
- Color promedio
- Gradiente promedio
- Promedio 2da derivada
- **Contraste**
- Momentos
- Textura

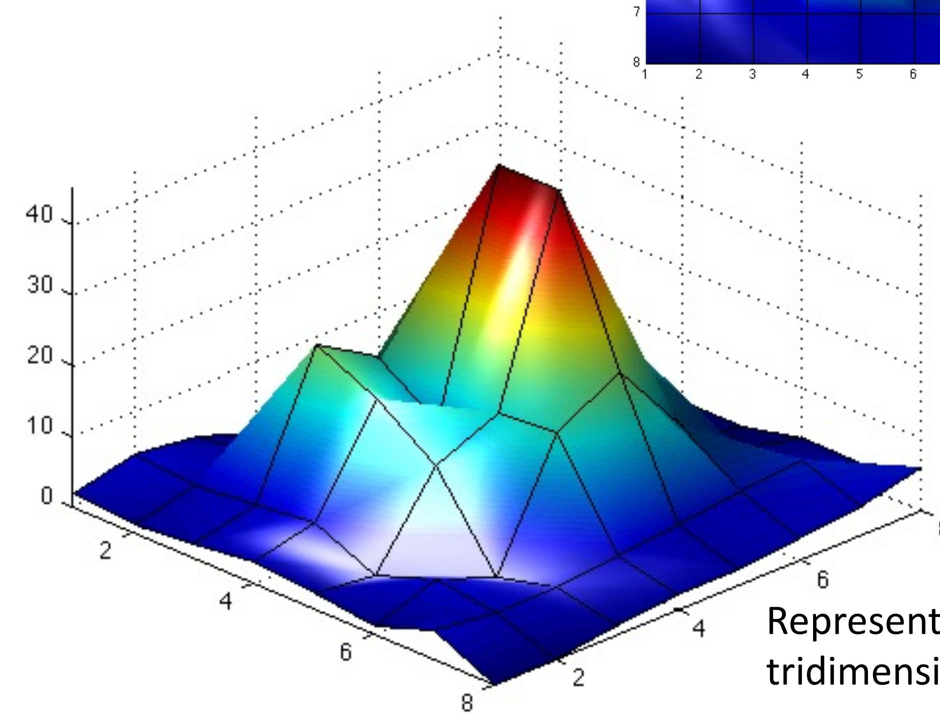
- El contraste mide la diferencia de color (o brillo) entre una región y su entorno. La región y el contorno no tienen píxeles en común. En general, se define que el entorno corresponde al doble en altura y ancho de la región.

i \ j	1	2	3	4	5	6	7	8
1	2	1	2	3	2	1	4	0
2	4	3	4	5	1	4	3	0
3	3	4	23	19	13	1	3	1
4	0	2	18	12	17	18	4	4
5	1	1	22	45	45	23	5	2
6	0	1	1	31	21	12	6	3
7	2	2	4	2	3	5	7	4
8	0	0	5	1	1	3	3	6

h

w





Representación tridimensional

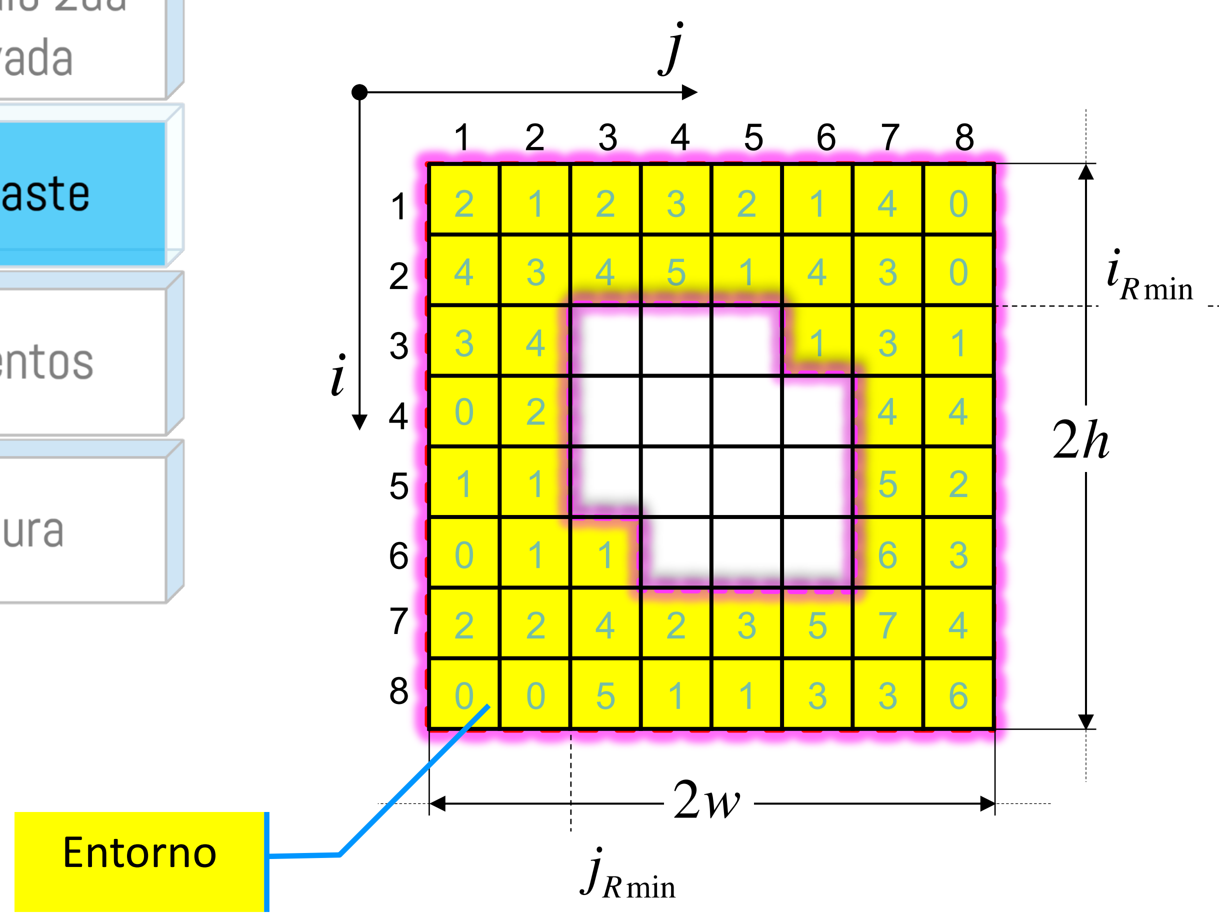
FACULTAD DE INGENIERÍA Y CIENCIAS

UAI UNIVERSIDAD ADOLFO IBÁÑEZ

Características cromáticas

- Color promedio
- Gradiente promedio
- Promedio 2da derivada
- **Contraste**
- Momentos
- Textura

- El contraste mide la diferencia de color (o brillo) entre una región y su entorno. La región y el contorno no tienen píxeles en común. En general, se define que el entorno corresponde al doble en altura y ancho de la región.



Fronteras del entorno

$$i_{E\min} = \left[i_{R\min} - \frac{h}{2} \right] \qquad j_{E\min} = \left[j_{R\min} - \frac{h}{2} \right]$$

$$i_{E\max} = i_{E\min} + 2h - 1 \qquad j_{E\max} = j_{E\min} + 2w - 1$$

E: Entorno R: Región

Ejemplo

$$i_{E\min} = \left[3 - \frac{4}{2} \right] = 1 \qquad j_{E\min} = \left[3 - \frac{4}{2} \right] = 1$$

$$i_{E\max} = 1 + 8 - 1 = 8 \qquad j_{E\max} = 1 + 8 - 1 = 8$$

Características cromáticas

- Color promedio
- Gradiente promedio
- Promedio 2da derivada
- **Contraste**
- Momentos
- Textura

- El contraste mide la diferencia de color (o brillo) entre una región y su entorno. La región y el contorno no tienen píxeles en común. En general, se define que el entorno corresponde al doble en altura y ancho de la región.

$i \backslash j$	1	2	3	4	5	6	7	8
1	2	1	2	3	2	1	4	0
2	4	3	4	5	1	4	3	0
3	3	4	23	19	13	1	3	1
4	0	2	18	12	17	18	4	4
5	1	1	22	45	45	23	5	2
6	0	1	1	31	21	12	6	3
7	2	2	4	2	3	5	7	4
8	0	0	5	1	1	3	3	6

Definición

$$G_E = \frac{1}{A_E} \sum_{i,j \in E} x_{i,j}$$

Promedio del entorno

$$G_R = \frac{1}{A_R} \sum_{i,j \in R} x_{i,j}$$

Promedio de la región

Formulaciones de contraste

$$K_1 = \frac{G_R - G_E}{G_E} \qquad K_2 = \frac{G_R - G_E}{G_R + G_E}$$

$$K_3 = \ln\left(\frac{G_R}{G_E}\right)$$

Características cromáticas

- Color promedio
- Gradiente promedio
- Promedio 2da derivada
- **Contraste**
- Momentos
- Textura

- El contraste mide la diferencia de color (o brillo) entre una región y su entorno. La región y el contorno no tienen píxeles en común. En general, se define que el entorno corresponde al doble en altura y ancho de la región.

i \ j	1	2	3	4	5	6	7	8
1	2	1	2	3	2	1	4	0
2	4	3	4	5	1	4	3	0
3	3	4	23	19	13	1	3	1
4	0	2	18	12	17	18	4	4
5	1	1	22	45	45	23	5	2
6	0	1	1	31	21	12	6	3
7	2	2	4	2	3	5	7	4
8	0	0	5	1	1	3	3	6

Resultados del ejemplo

$$G_E = 2.54$$
$$G_R = 22.78$$

$$K_1 = \frac{G_R - G_E}{G_E} \longrightarrow K_1 = 7.9708$$

$$K_2 = \frac{G_R - G_E}{G_R + G_E} \longrightarrow K_2 = 0.7994$$

$$K_3 = \ln\left(\frac{G_R}{G_E}\right) \longrightarrow K_3 = 2.1940$$

Características cromáticas

Color promedio

Gradiente promedio

Promedio 2da derivada

Contraste

Momentos

Textura

- Los momentos estadísticos se pueden aplicar en imágenes en escala de grises. Para ello la ecuación se modifica para incorporar el nivel de gris en cada posición de la imagen.

j →, i ↓

i \ j	1	2	3	4	5	6
1	4	0	0	2	2	3
2	2	1	12	15	5	4
3	1	11	25	34	56	2
4	3	16	34	3	4	1
5	2	0	4	33	65	3
6	4	1	6	3	4	2

Región R

$x(i,j)$

Momento estadísticos en escala de grises

$$m'_{rs} = \sum_{i,j \in \mathbf{R}} i^r \cdot j^s \cdot x(i,j) \qquad \text{para} \quad r,s \in N$$

Momento centrales

$$\mu_{rs} = \sum_{i,j \in \mathbf{R}} \left(i-\bar{i}\right)^r \left(j-\bar{j}\right)^s \cdot x(i,j)$$

Sobre este ecuación empleamos los momentos de HU.

IMPORTANTE: Dividir cada píxel por 255, de esta forma los valores serán más estables y estarán entre cero y uno.

Características cromáticas

- Color promedio
- Gradiente promedio
- Promedio 2da derivada
- Contraste
- **Momentos**
- Textura

- Analicemos el primer caso, calculando el centro de masa de la región

$i \backslash j$	1	2	3	4	5	6
1	4	0	0	2	2	3
2	2	1	12	15	5	4
3	1	11	25	34	56	2
4	3	16	34	3	4	1
5	2	0	4	33	65	3
6	4	1	6	3	4	2

Momento centrales

$$m'_{rs} = \sum_{i,j \in \mathbf{R}} i^r \cdot j^s \cdot x(i,j)$$

Centro de masa

$$(\bar{i}, \bar{j}) = \left(\frac{m'_{10}}{m'_{00}}, \frac{m'_{01}}{m'_{00}} \right)$$

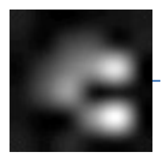
Ejemplo

$$\left.\begin{aligned}
m'_{00} &= 3^0 \cdot 2^0 \cdot \frac{11}{255} + \cdots + 5^0 \cdot 5^0 \cdot \frac{65}{255} = 1.18 \\
m'_{10} &= 3^1 \cdot 2^0 \cdot \frac{11}{255} + \cdots + 5^1 \cdot 5^0 \cdot \frac{65}{255} = 4.40 \\
m'_{01} &= 3^0 \cdot 2^1 \cdot \frac{11}{255} + \cdots + 5^0 \cdot 5^1 \cdot \frac{65}{255} = 4.71
\end{aligned}\right\} (\bar{i}, \bar{j}) = \left(\frac{4.4}{1.18}, \frac{4.71}{1.18} \right) = (3.72, 3.99)$$

Características cromáticas

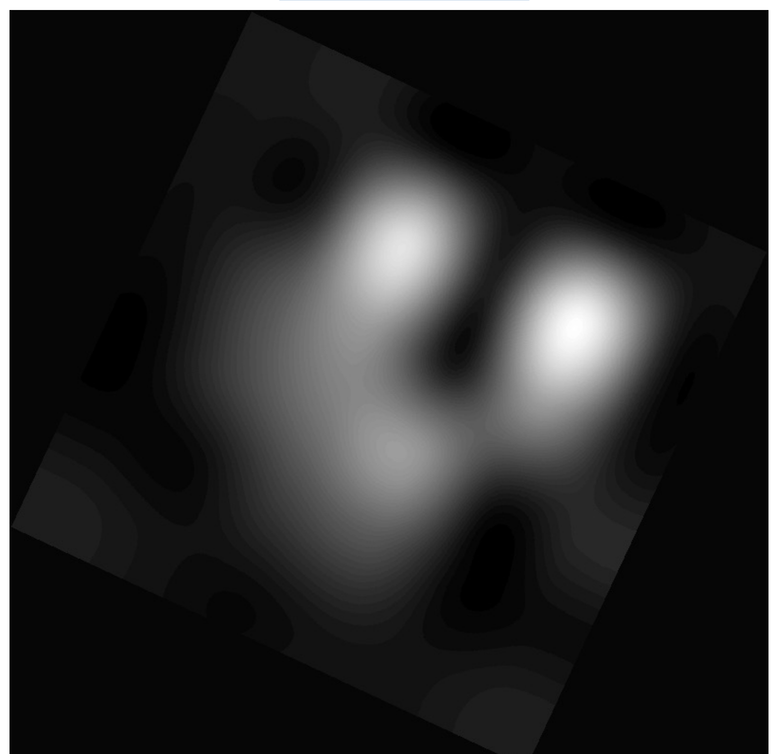
- Color promedio
- Gradiente promedio
- Promedio 2da derivada
- Contraste
- **Momentos**
- Textura

- Veamos que sucede cuando modificamos la escala y rotación y aplicamos los primeros tres momentos de Hu.



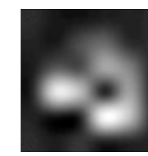
60 × 60

$\phi_1 = 1.7480$
$\phi_2 = 0.1462$
$\phi_3 = 0.9282$



800 × 800.
Rotado 65°

$\phi_1 = 1.7440$
$\phi_2 = 0.1441$
$\phi_3 = 0.9091$



60 × 60

$\phi_1 = 1.5171$
$\phi_2 = 0.1050$
$\phi_3 = 0.6502$

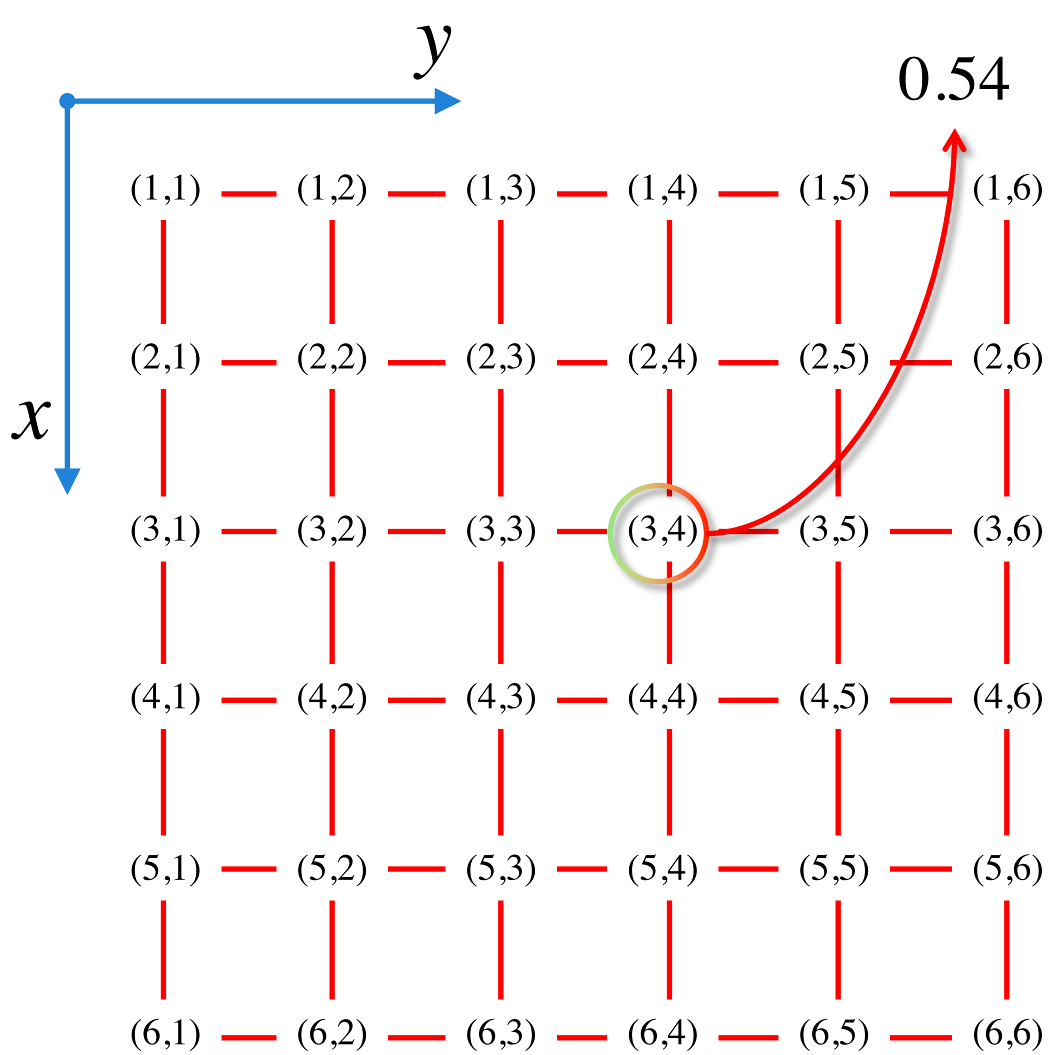
669 × 669.
Rotado 83°

$\phi_1 = 1.5181$
$\phi_2 = 0.1067$
$\phi_3 = 0.6486$

Anexo

Comando Meshgrid

- `X,Y = np.meshgrid(x,y)`
  Es un comando de numpy que permite crear la malla completa de puntos x, y que pueden ser evaluados en otra función.



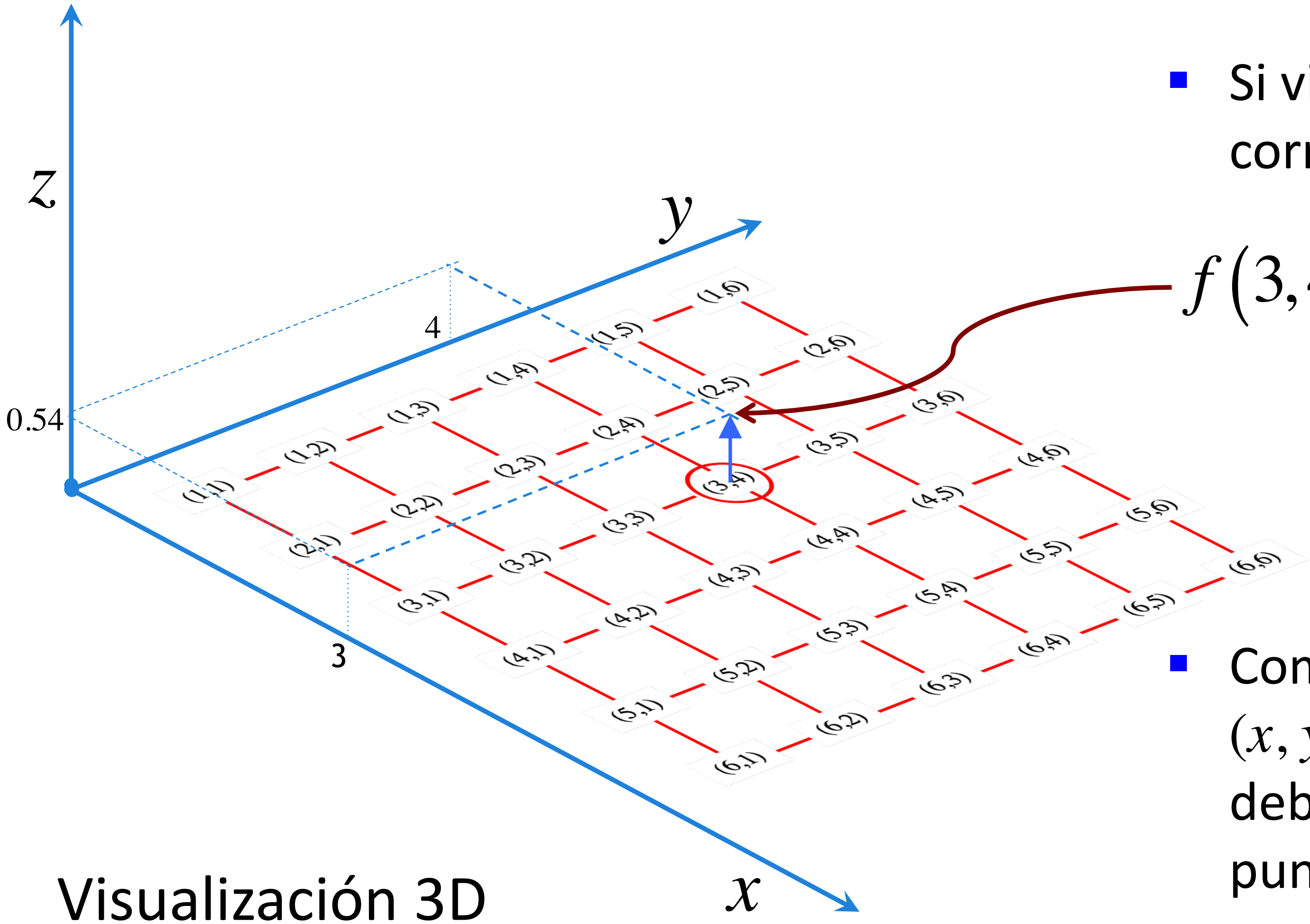
- Suponga que usted desea evaluar la siguiente función $f(x,y)$ entre 1 y 6 para x e y :

$$f(x,y) = \cos(x-y) \qquad \forall x = [1,\cdots,6] \quad \forall y = [1,\cdots,6]$$

- Para cada punto de la matriz debe evaluar la ecuación, por ejemplo:

$$f(3,4) = \cos(3-4) = 0.54$$

- `X,Y = np.meshgrid(x,y)`
  Es un comando de numpy que permite crear la malla completa de puntos x, y que pueden ser evaluados en otra función.



Visualización 3D

- Si visualizamos en el eje Z el resultado, este corresponde a 0.54

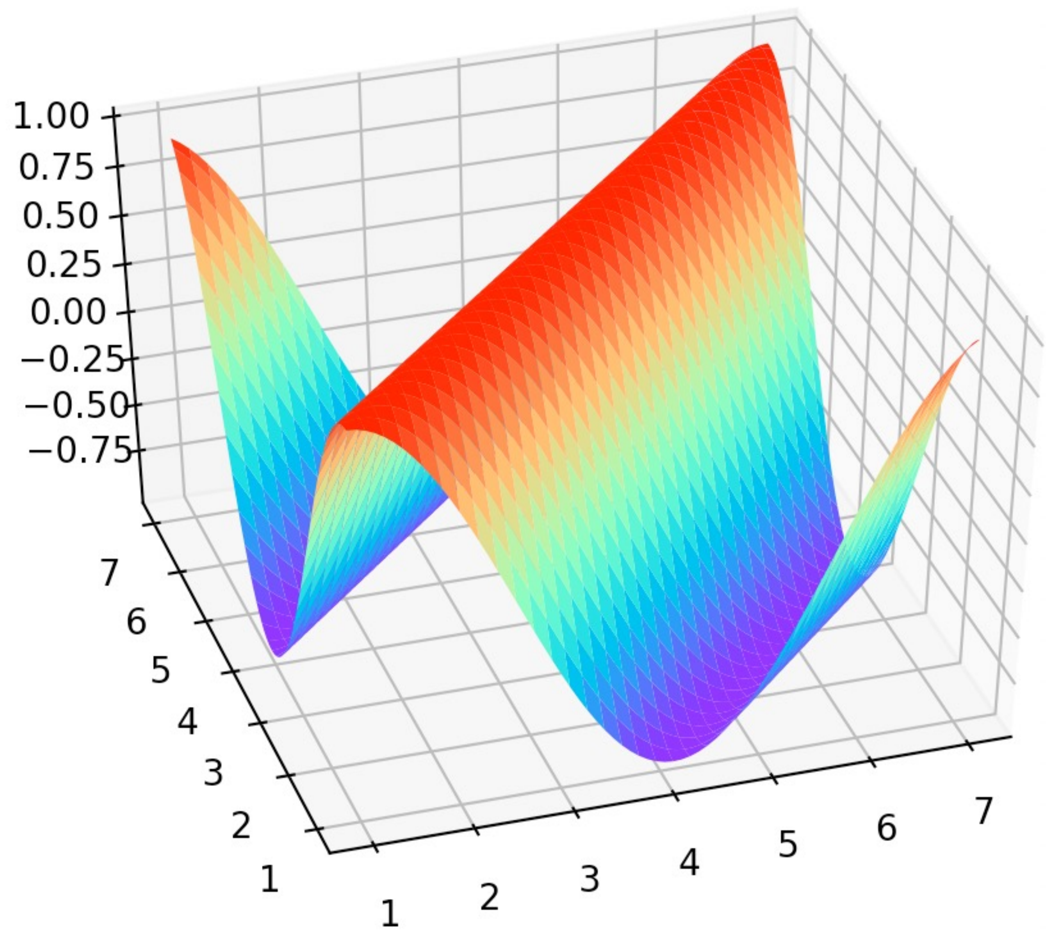
$$f(3,4) = \cos(3-4) = 0.54$$

- Como observamos, cada par ordenado (x, y) genera un resultado. Por lo tanto debemos calcular para cada uno de los puntos de la matriz un valor.

FACULTAD DE INGENIERÍA Y CIENCIAS

UNIVERSIDAD ADOLFO IBÁÑEZ

- `X,Y = np.meshgrid(x,y)`
  Es un comando de numpy que permite crear la malla completa de puntos x, y que pueden ser evaluados en otra función.



- Lo anterior puede ser calculado con doble ciclo `for`. De esta forma en el ciclo interno vamos generando cada coordenada que será evaluada en nuestra función

$$f(x,y) = \cos(x-y) \qquad \forall x = [1,\cdots,6]$$
$$\forall y = [1,\cdots,6]$$

En Python (versión simple y **lenta**)

```
f = np.zeros((6,6))

for x in range(6):
    for y in range(6):
        f[x,y] = np.cos(x-y)
    end
end
```

Meshgrid

- `X,Y = np.meshgrid(x,y)`
  Es un comando de numpy que permite crear la malla completa de puntos x, y que pueden ser evaluados en otra función.

En Python (con meshgrid)

```
import numpy as np
x   = np.linspace(1,6,6)
y   = np.linspace(1,6,6)
X,Y = np.meshgrid(x,y)
```

- Python provee una función que determina todos los puntos x, y que requerimos para evaluar la función.

```
X =

    1  2  3  4  5  6
    1  2  3  4  5  6
    1  2  3  4  5  6
    1  2  3  4  5  6
    1  2  3  4  5  6
    1  2  3  4  5  6
```

```
Y =

    1  1  1  1  1  1
    2  2  2  2  2  2
    3  3  3  3  3  3
    4  4  4  4  4  4
    5  5  5  5  5  5
    6  6  6  6  6  6
```

Los índices de estas matrices serán ocupados en nuestra función. Si observan cada punto corresponde a un par ordenado

FACULTAD DE INGENIERÍA Y CIENCIAS

UAI UNIVERSIDAD ADOLFO IBÁÑEZ

- `X,Y = np.meshgrid(x,y)`
  Es un comando de numpy que permite crear la malla completa de puntos x, y que pueden ser evaluados en otra función.

En Python (con meshgrid)

```
import numpy as np
x   = np.linspace(1,6,6)
y   = np.linspace(1,6,6)
X,Y = np.meshgrid(x,y)
P   = X-Y
```

- Dado que `X` e `Y` son matrices, podemos operar con ellas (sumar, restar multiplicar, dividir, etc.)

```
X =

   1  2  3  4  5  6
   1  2  3  4  5  6
   1  2  3  4  5  6
   1  2  3  4  5  6
   1  2  3  4  5  6
   1  2  3  4  5  6
```

```
Y =

   1  1  1  1  1  1
   2  2  2  2  2  2
   3  3  3  3  3  3
   4  4  4  4  4  4
   5  5  5  5  5  5
   6  6  6  6  6  6
```

```
P=

[[ 0.  1.  2.  3.  4.  5.  6.]
 [-1.  0.  1.  2.  3.  4.  5.]
 [-2. -1.  0.  1.  2.  3.  4.]
 [-3. -2. -1.  0.  1.  2.  3.]
 [-4. -3. -2. -1.  0.  1.  2.]
 [-5. -4. -3. -2. -1.  0.  1.]
 [-6. -5. -4. -3. -2. -1.  0.]]
```

- `X,Y = np.meshgrid(x,y)`
  Es un comando de numpy que permite crear la malla completa de puntos x, y que pueden ser evaluados en otra función.

En Python (con meshgrid)

```
import numpy as np
x   = np.linspace(1,6,6)
y   = np.linspace(1,6,6)
X,Y = np.meshgrid(x,y)
P   = X-Y
f   = np.cos(P)
```

- También podemos aplicar una función sobre una matriz, por ejemplo, coseno, seno, exponencial, tangente, etc. Esto se aplica a cada valor de la matriz

```
P=

[[ 0.   1.   2.   3.   4.   5.   6.]
 [-1.   0.   1.   2.   3.   4.   5.]
 [-2.  -1.   0.   1.   2.   3.   4.]
 [-3.  -2.  -1.   0.   1.   2.   3.]
 [-4.  -3.  -2.  -1.   0.   1.   2.]
 [-5.  -4.  -3.  -2.  -1.   0.   1.]
 [-6.  -5.  -4.  -3.  -2.  -1.   0.]]
```

coseno →

```
f =

[[ 1.     0.54  -0.42  -0.99  -0.65   0.28   0.96]
 [ 0.54  1.      0.54  -0.42  -0.99  -0.65   0.28]
 [-0.42   0.54   1.      0.540 -0.42  -0.99  -0.65]
 [-0.99  -0.42   0.54   1.      0.54  -0.42  -0.99]
 [-0.65  -0.99  -0.42   0.54   1.      0.54  -0.42]
 [ 0.28  -0.65  -0.99  -0.42   0.54   1.      0.54]
 [ 0.96   0.28  -0.65  -0.99  -0.42   0.54   1.  ]]
```

- El comando `surf` permite plotear tridimensionalmente cada valor de la matriz. En el eje Z queda se muestra el valor de cada punto de la matriz

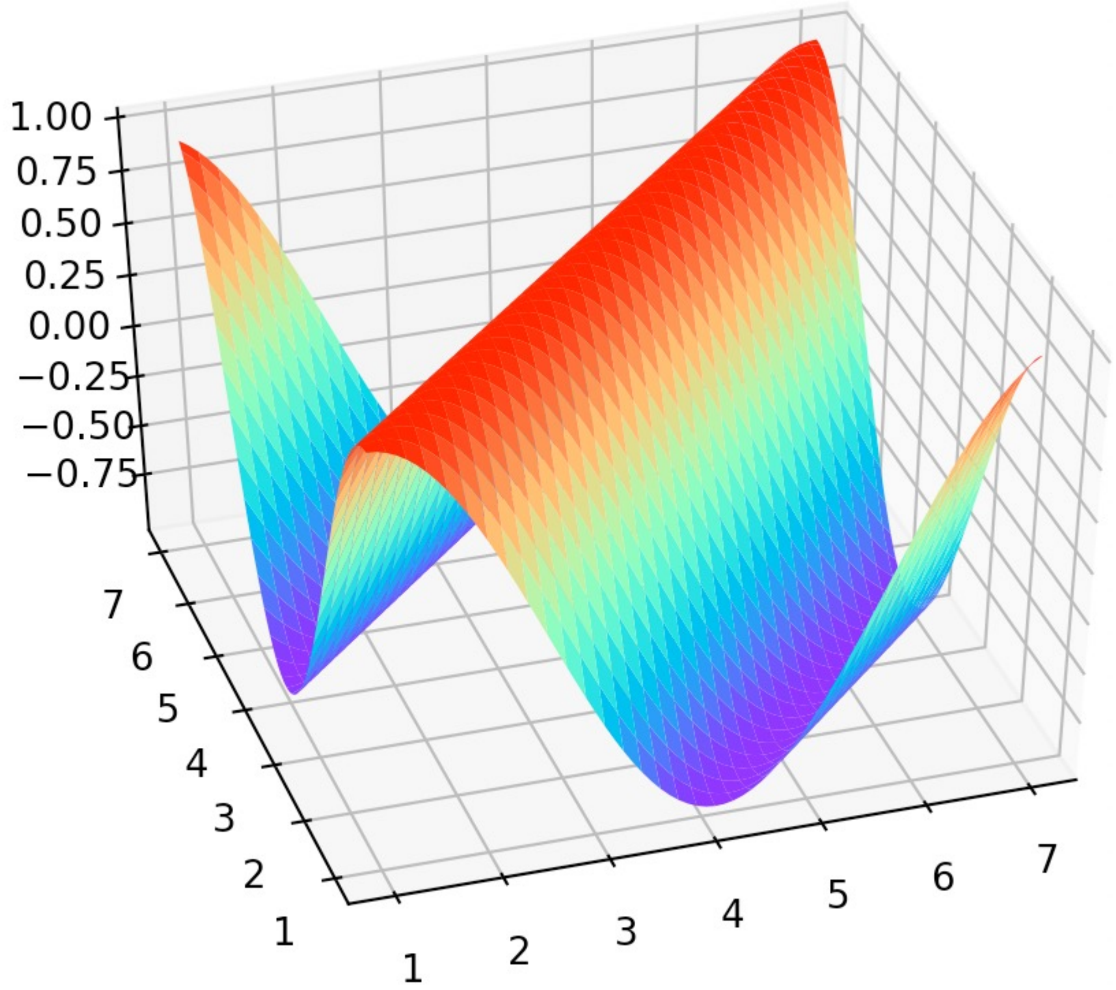
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
```

evaluación de la función

```
x  = np.linspace(1,7,70)
y  = np.linspace(1,7,70)

X,Y = np.meshgrid(x,y)
P   = X-Y
f   = np.cos(P)
```

gráfica de la matriz f

```
fig = plt.figure()
ax  = fig.gca(projection='3d')

surf = ax.plot_surface(X, Y, f,
                cmap=cm.rainbow,
                linewidth=0,
                antialiased=True)

plt.show()
```



- `X,Y = np.meshgrid(x,y)`
  Es un comando de numpy que permite crear la malla completa de puntos x, y que pueden ser evaluados en otra función.

En Python (MUY LENTO)

```
for x in range(1,7):
    for y in range(1,7):
        f[x,y]=np.cos(x-y)
    end
end
```

=

En Python (EFICIENTE)

```
import numpy as np
x   = np.linspace(1,7,7)
y   = np.linspace(1,7,7)
X,Y = np.meshgrid(x,y)
P   = X-Y
f   = np.cos(P)
```

Obtenemos el mismo resultado, pero la versión con for es lenta

Evite ocupar ciclos `for` en Python ya que son muy lentos. Ocupe operaciones matriciales que son eficientes y rápidas