PROCESAMIENTO DE IMÁGENES

Francisco J. Hernández López

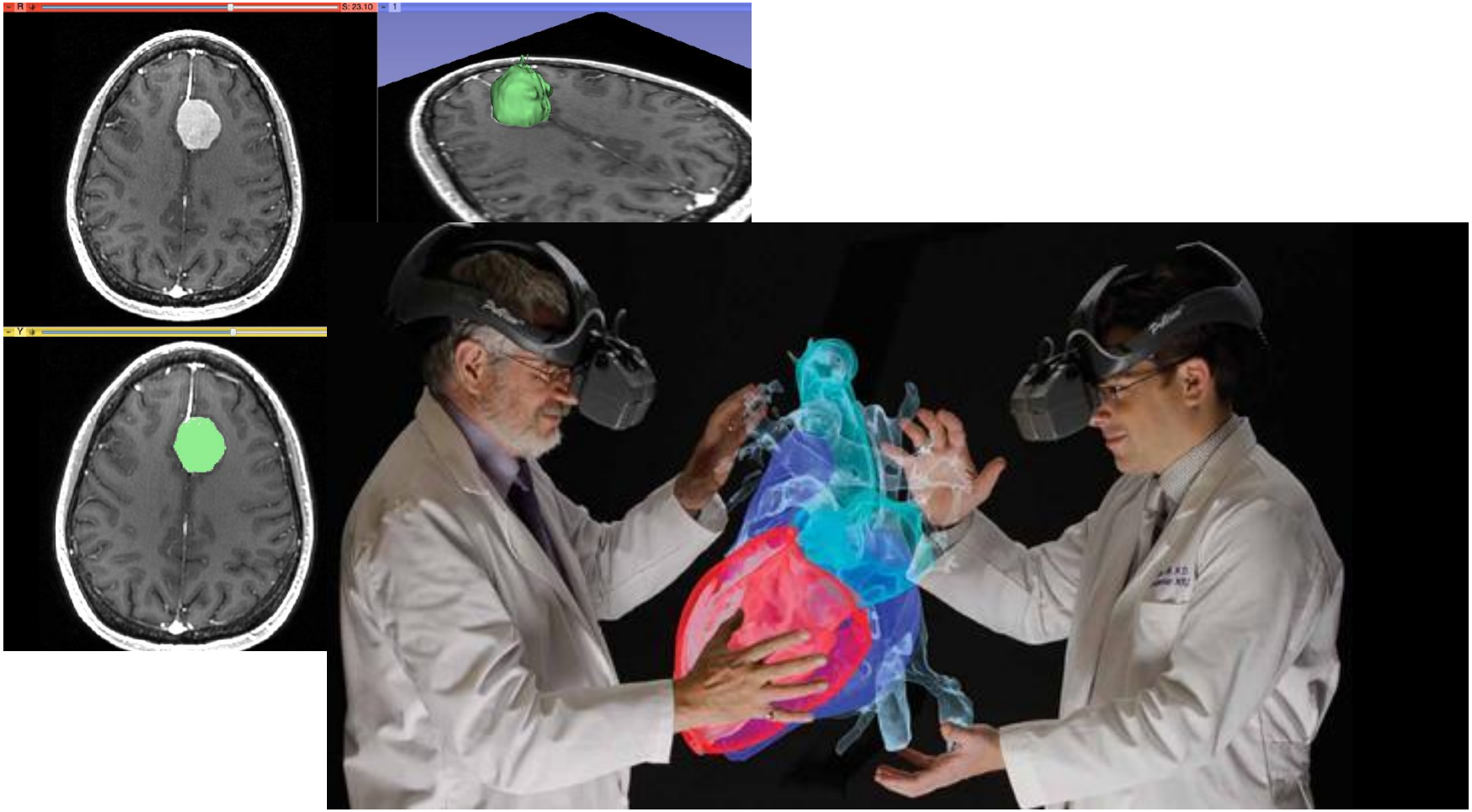
fcoj23@cimat.mx

PROCESAMIENTO DE IMÁGENES

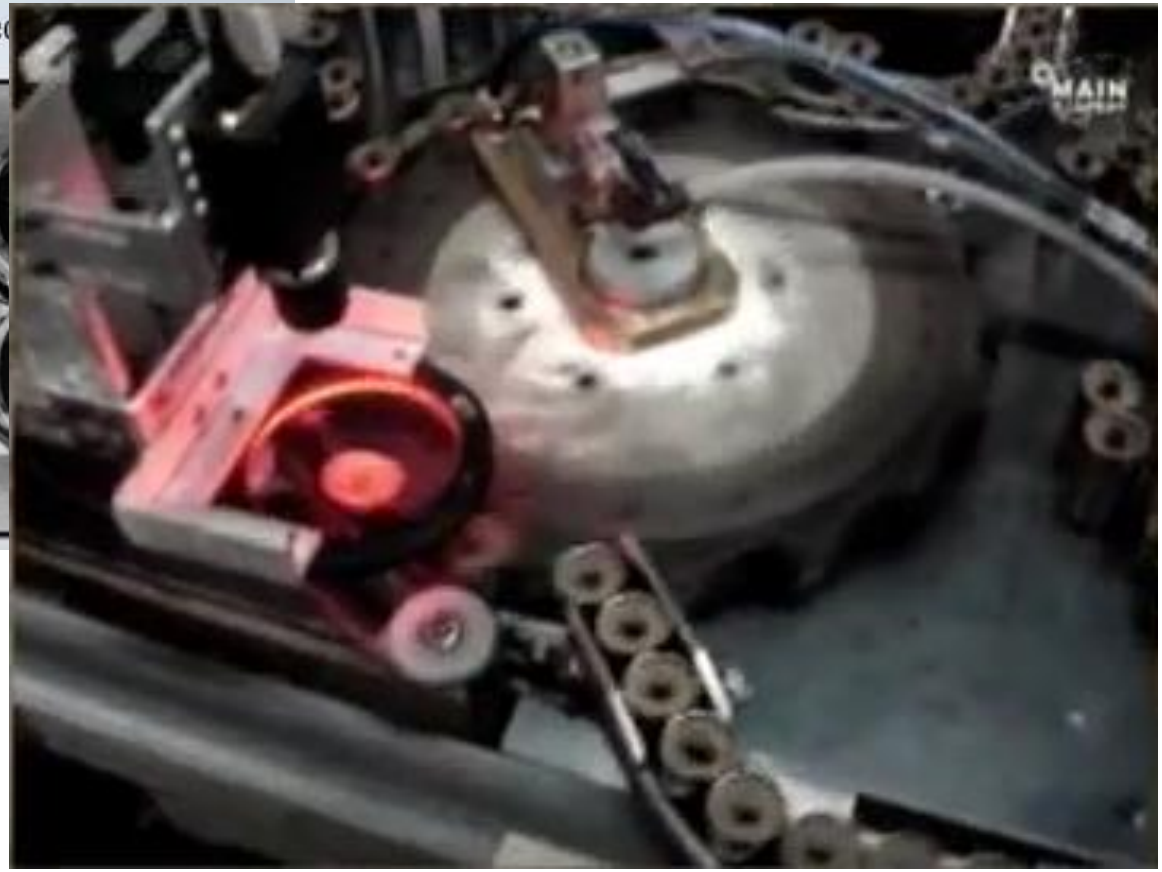
- Proceso computacional que transforma una o más imágenes de entrada en una imagen de salida.
- Se utiliza para analizar e interpretar la imagen, por medio de algoritmos que permiten resaltar sus principales características:
  - Bordes
  - Contraste
  - Puntos de interés
  - Etc.

Corke, Peter. *Robotics, vision and control: fundamental algorithms in MATLAB*. Vol. 73. Springer Science & Business Media, 2011.

APLICACIONES MÉDICAS



APLICACIONES EN LA INDUSTRIA

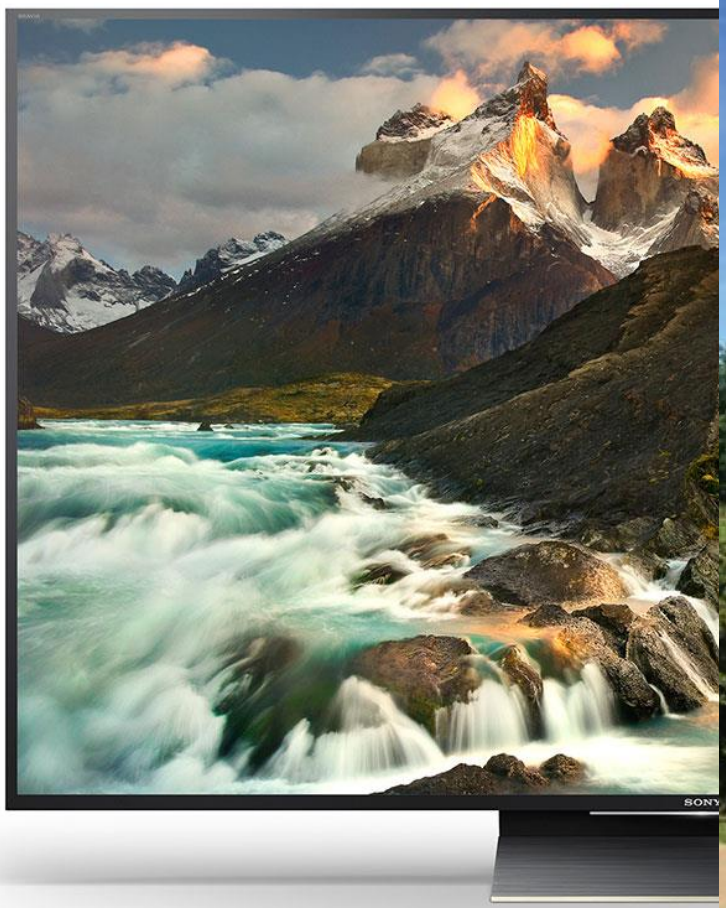


Detección de pilas defectuosas

https://www.youtube.com/watch?v=y6wzRThO7vM

APLICACIONES DE SEGURIDAD-VIGILANCIA

ENTRETENIMIENTO

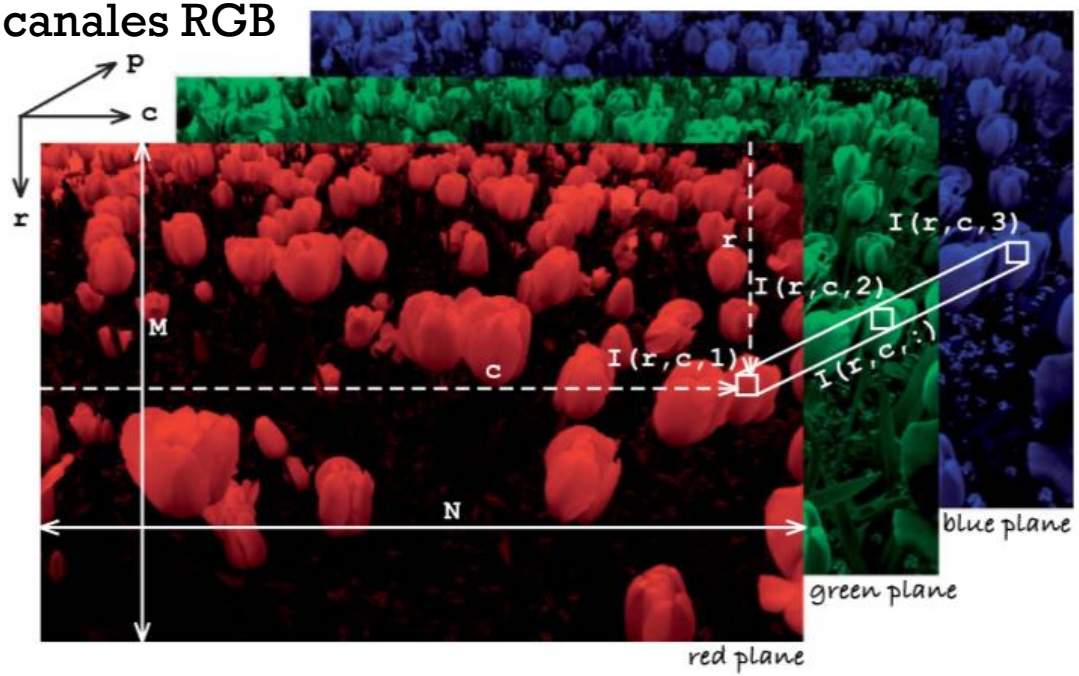


IMAGEN

- Arreglo rectangular de elementos (pixeles)
- En escala de gris es una función de dos variables:
  - l para columnas (“c”)
  - l para renglones (“r”)

IMAGEN RGB

- Es una función de tres variables:
  - 1 para columnas (“c”)
  - 1 para renglones (“r”)
  - 1 para los canales RGB



Estructura de una imagen de 3 dimensiones: fila, columna y color. Peter Corke. 2011.

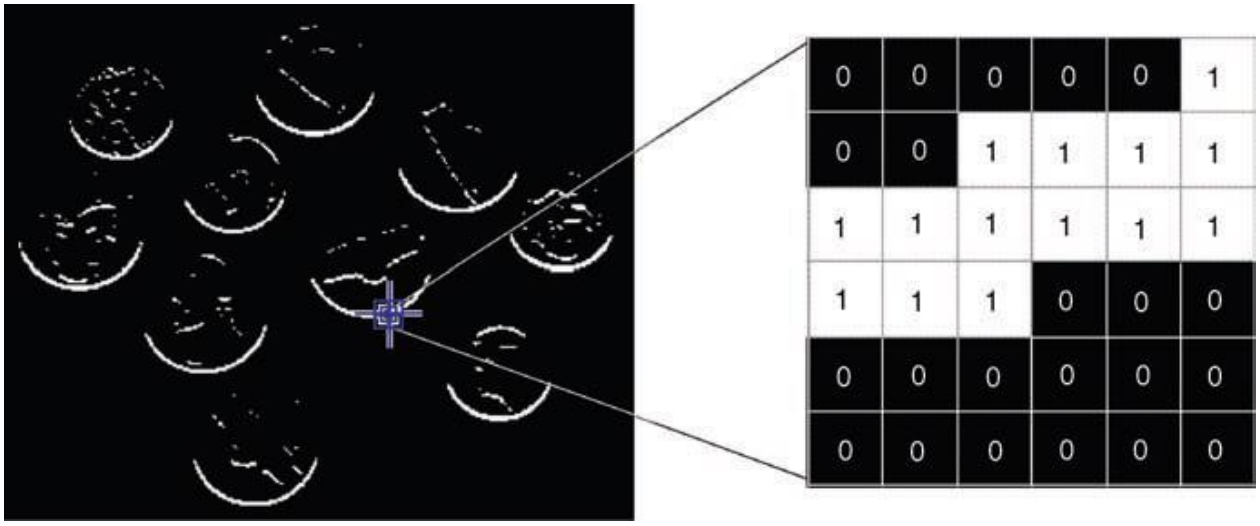


Imagen Binaria

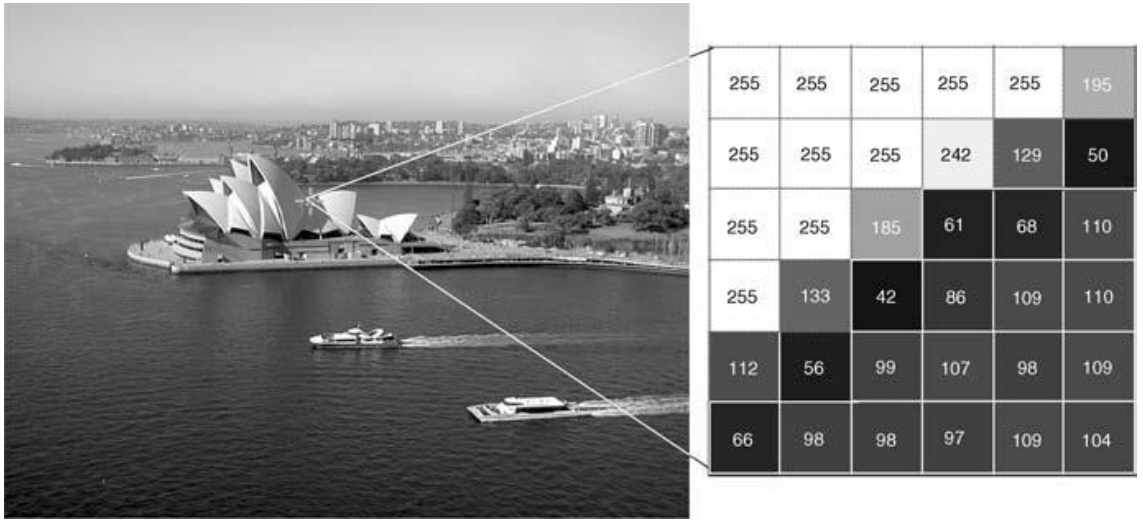


Imagen en escala de grises

Marques, O. (2011). *Practical image and video processing using MATLAB*. Hoboken, NJ: Wiley-IEEE Press.

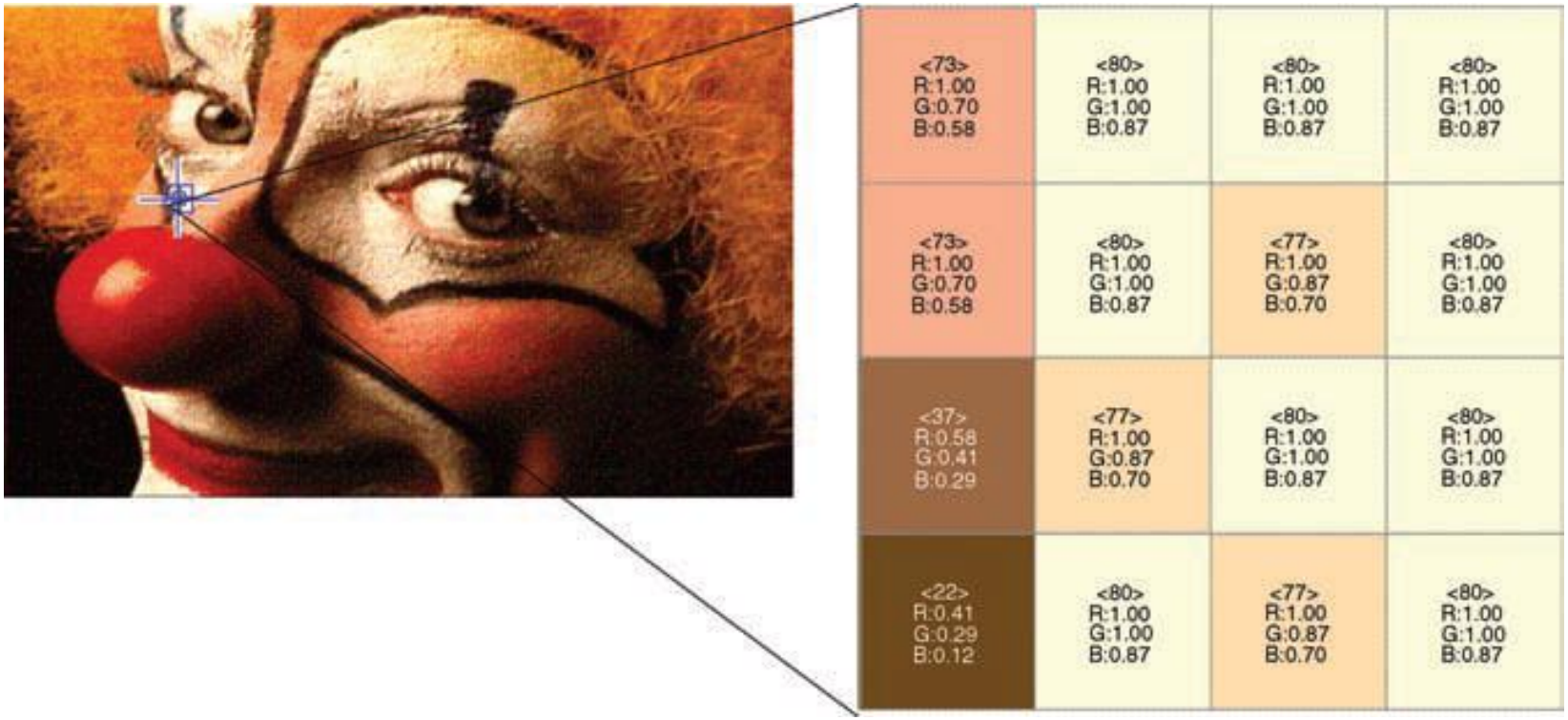


Imagen a color RGB

Marques, O. (2011). *Practical image and video processing using MATLAB*. Hoboken, NJ: Wiley-IEEE Press.

(a) Imagen en RGB. (b) Canal R. (c) Canal G. (d) Canal B

Marques, O. (2011). *Practical image and video processing using MATLAB*. Hoboken, NJ: Wiley-IEEE Press.

OBTENER UNA IMAGEN

- A partir de archivos
  - MatLab:
    - I1=imread('flowers8.png');
  - OpenCV:
    - Image = cv::imread("flowers8.png");

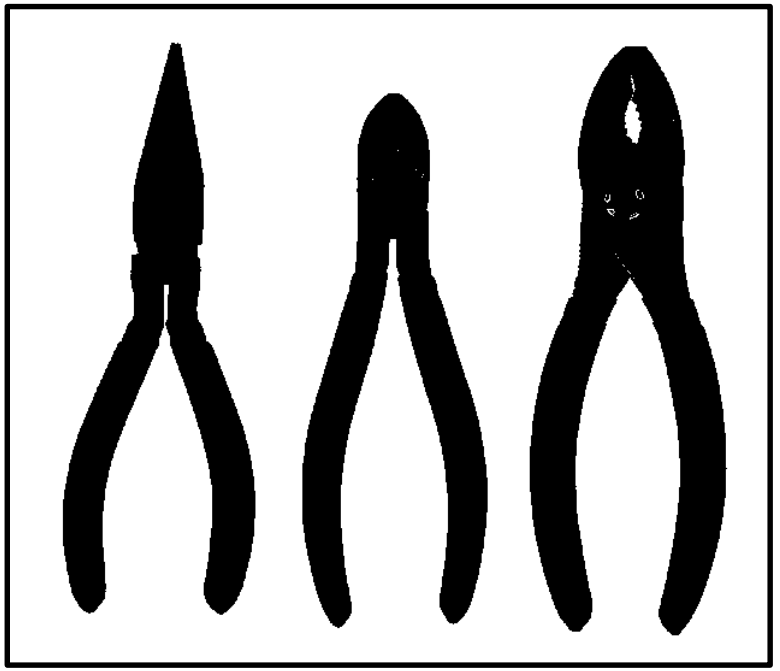
- A partir de una cámara
  - MatLab:
    - obj=videoinput('winvideo',1);
    - frame = getsnapshot(obj);
  - OpenCV:
    - cv::VideoCapture capture;
    - capture.open(0);

OBTENER UNA IMAGEN (C1)

- A partir de un archivo de video
  - MatLab:
    - video=VideoReader('highway.avi');
    - frame=read(video,frame_number);
  - OpenCV:
    - cv::VideoCapture capture;
    - capture.open("highway.avi");

UMBRAL

I



O

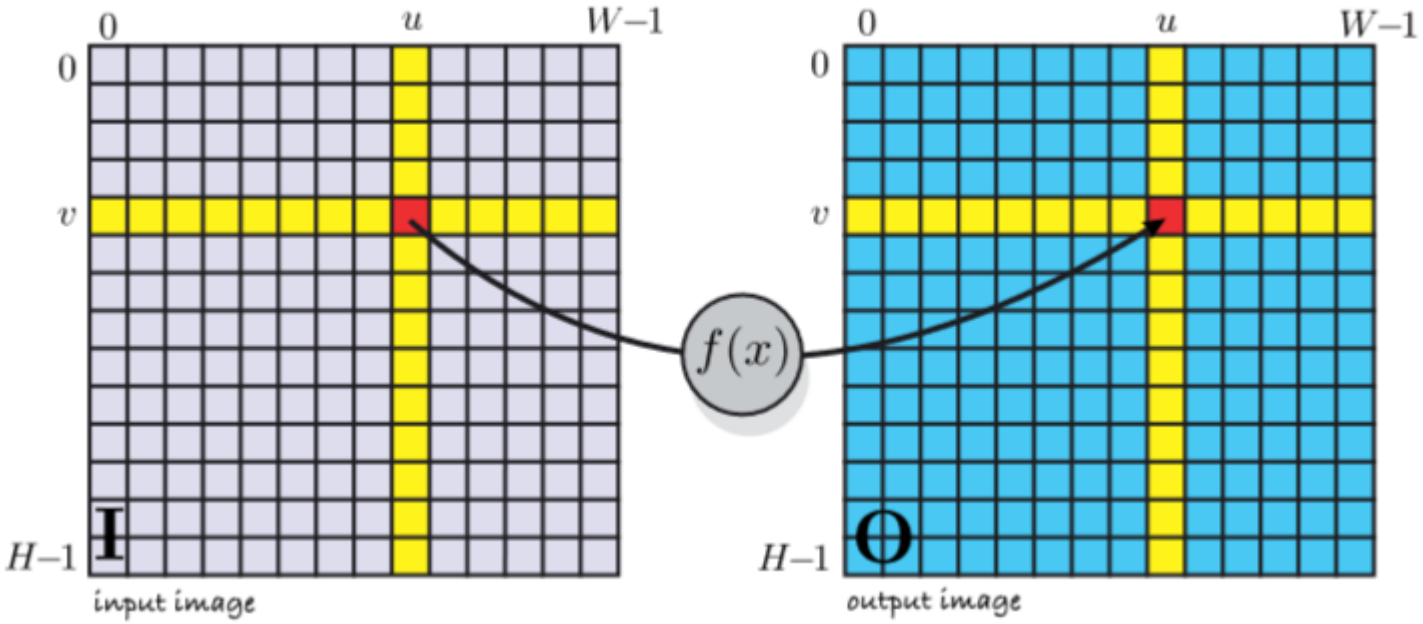
$$O(x,y) = \begin{cases} 1 & si\ I(x,y) > T \\ 0 & si\ I(x,y) \leq T \end{cases}$$

Con T un valor dentro del rango dinámico de I.

OPERACIONES MONÁDICAS

- El resultado es una imagen del mismo tamaño que la imagen de entrada
- Cada pixel de salida es una función del correspondiente pixel de entrada:

$$O[u,v] = f(I[u,v]), \forall (u,v) \in I$$



Ejemplos:

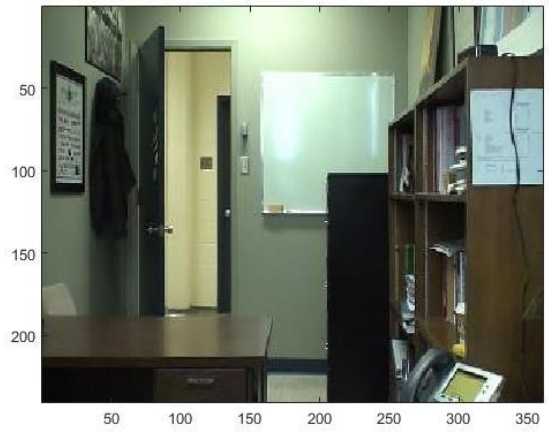
- $I * escalar$
- $abs(I)$
- $sqrt(I)$
- Etc.

Operaciones monádicas. Peter Corke. 2011.

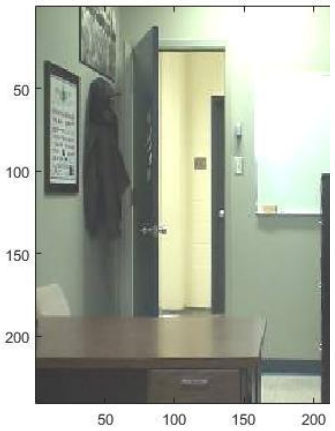
CAMBIAR EL BRILLO A UNA IMAGEN

$I + escalar$

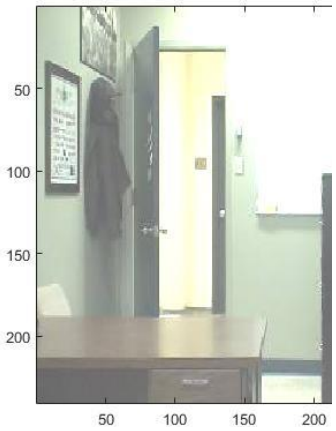
I



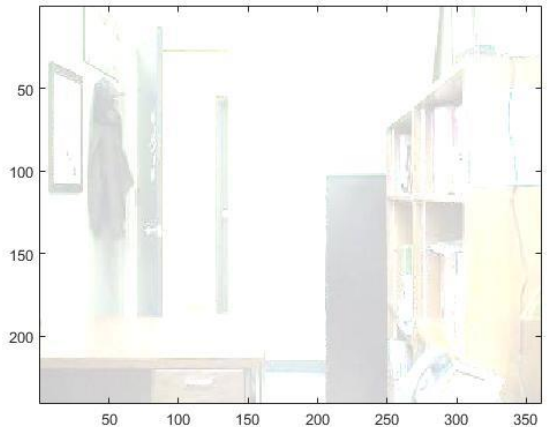
$I + 50$

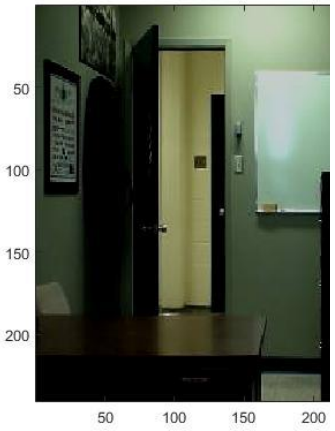


$I + 100$



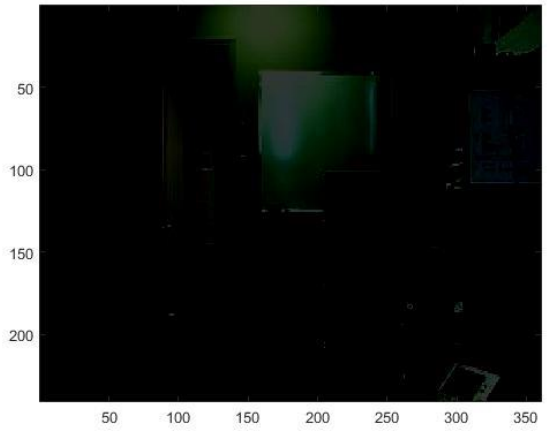
$I + 200$





$I - 50$

$I - 100$

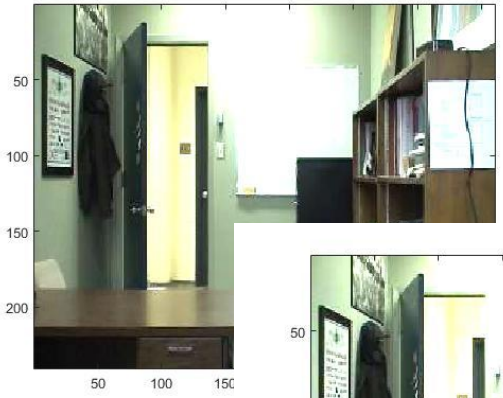


$I - 200$

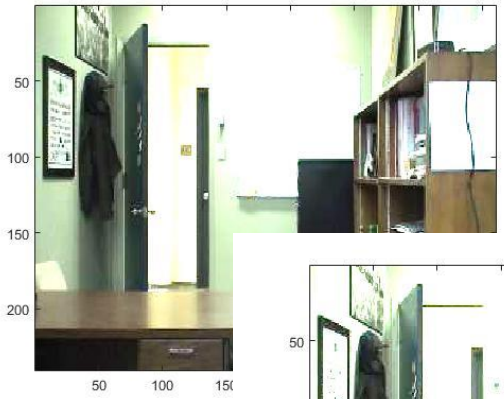
CAMBIAR EL CONTRASTE A UNA IMAGEN

$I * escalar$

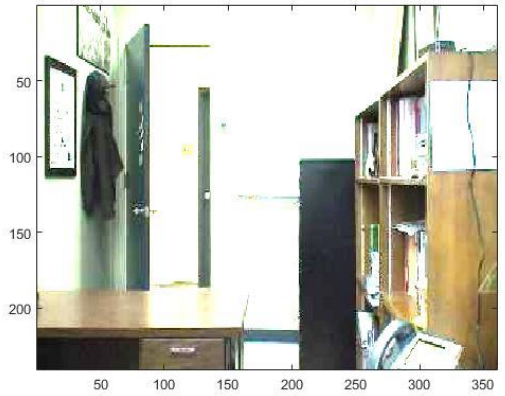
I



$I * 1.5$



$I * 2.0$

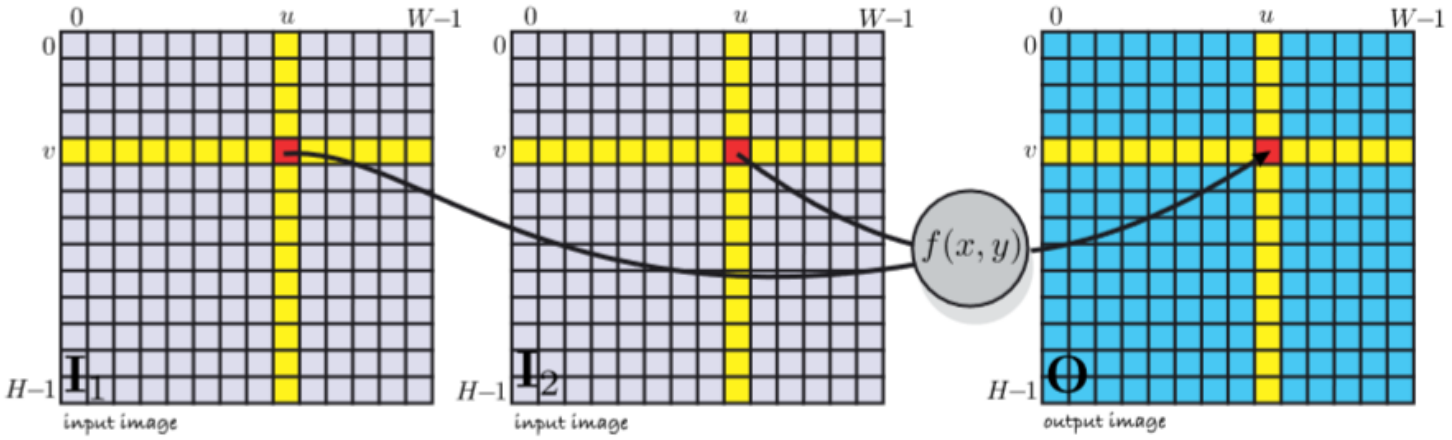


$I * 3.0$

OPERACIONES DIÁDICAS

- Cada pixel de salida es una función de los correspondientes pixeles en las dos imágenes de entrada:

$$O[u,v] = f\big(I_1[u,v], I_2(u,v)\big), \forall (u,v) \in I_1$$



Ejemplos:

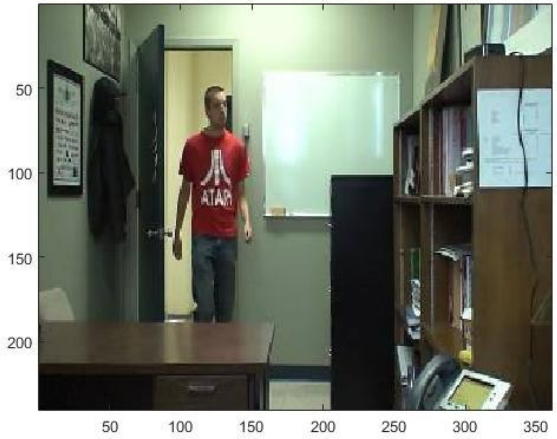
- $I_1 + I_2$
- $I_1 - I_2$
- $I_1 .* I_2$
- Etc.

Operaciones diádicas. Peter Corke. 2011.

SUMAR DOS IMÁGENES $I_3 = I_1 + I_2$

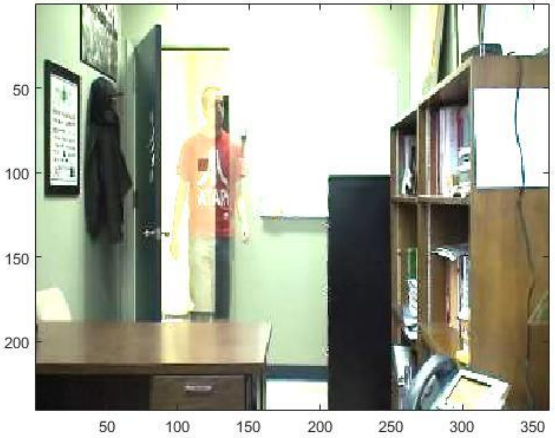
I_1

+

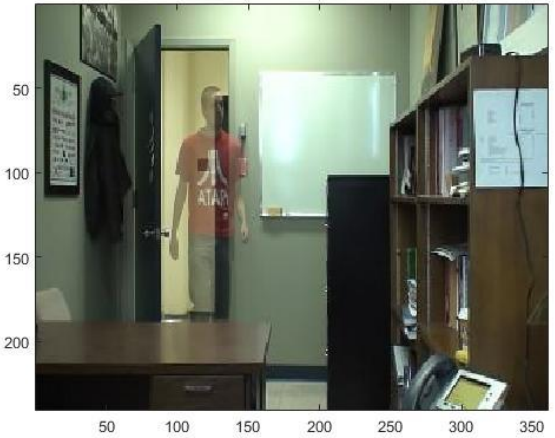


I_2

=



I_3 con uint8

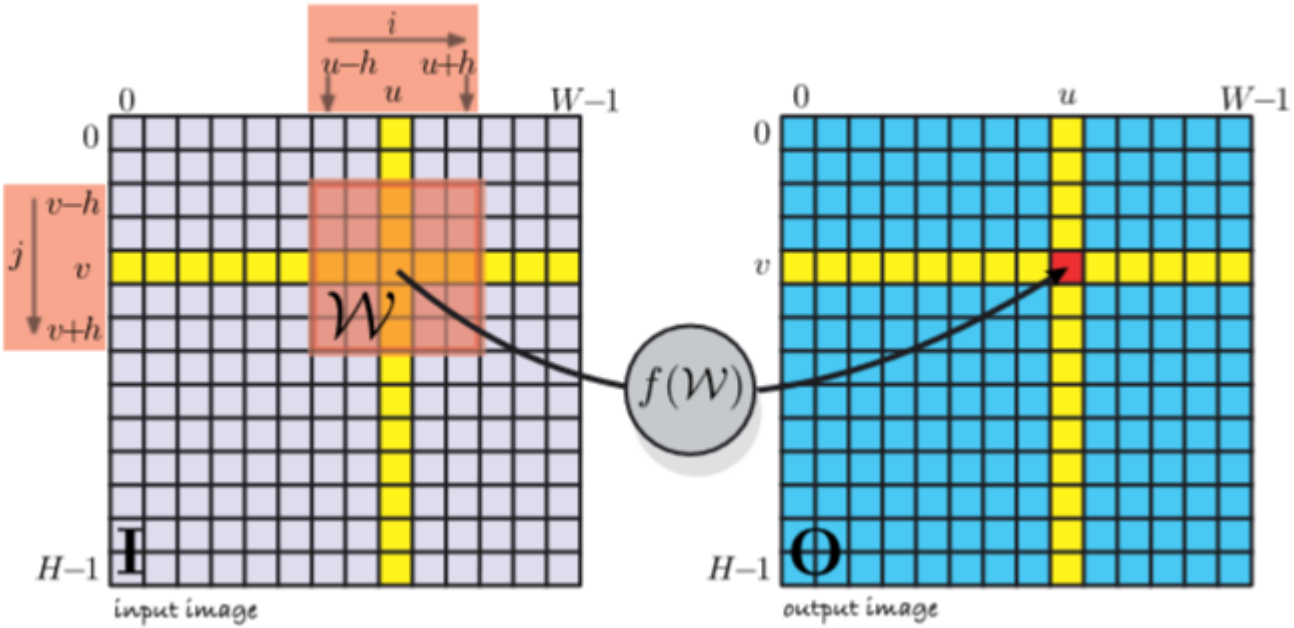


I_3 con double

OPERACIONES DE VENTANA

- Cada pixel de salida es una función de los pixeles de una cierta ventana o región:

$$O[u,v] = f(I[u+i, v+j]), \forall (i,j) \in \mathcal{W}, \forall (u,v) \in I,\ \ i,j \in [-h,h]$$



Lineales:

- Suavizadores
- Detección de bordes, etc.

No lineales:

- Filtro de rango
- Emparejamiento
- Etc.

Operaciones espaciales. Peter Corke. 2011.

CORRELACIÓN

- Operador espacial lineal:

$$O[u,v] = \sum_{(i,j)\in\mathcal{W}} I[u+i, v+j]K[i,j],\ \forall(u,v) \in I, i,j \in [-h,h]$$

donde $K \in \mathbb{R}^{w\times w}$ es un kernel.

- Para una imagen de tamaño $N \times N$, se requiere w^2N^2 multiplicaciones y sumas

CONVOLUCIÓN

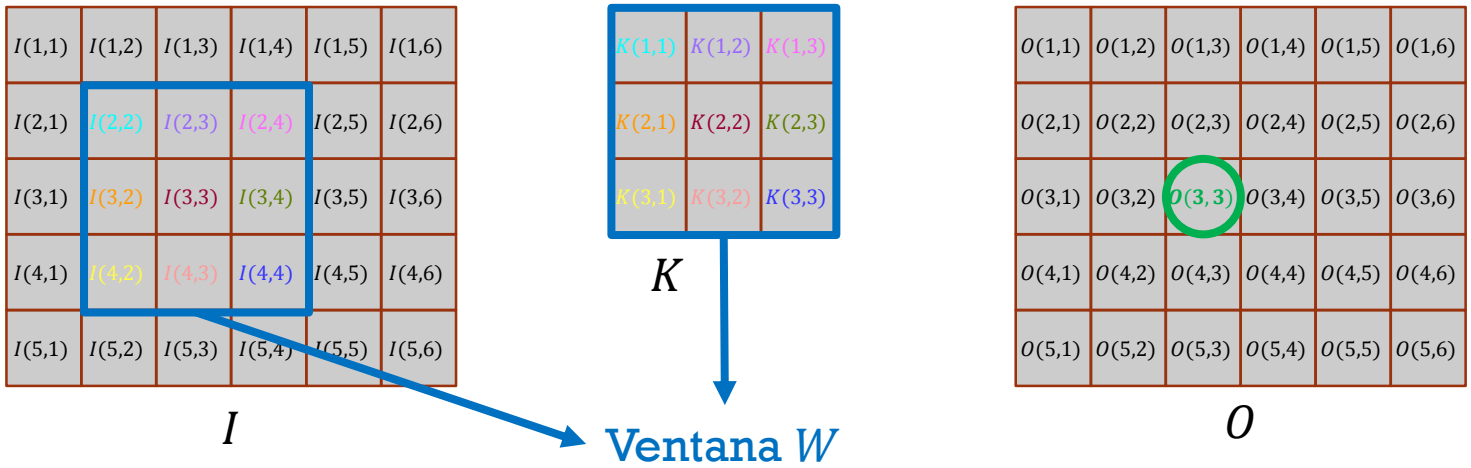
- Operador espacial lineal $O = I \otimes K$:

$$O[u,v] = \sum_{(i,j)\in\mathcal{W}} I[u-i, v-j]K[i,j],\ \forall(u,v) \in I, i,j \in [-h,h]$$

donde $K \in \mathbb{R}^{w\times w}$ es un kernel de convolución

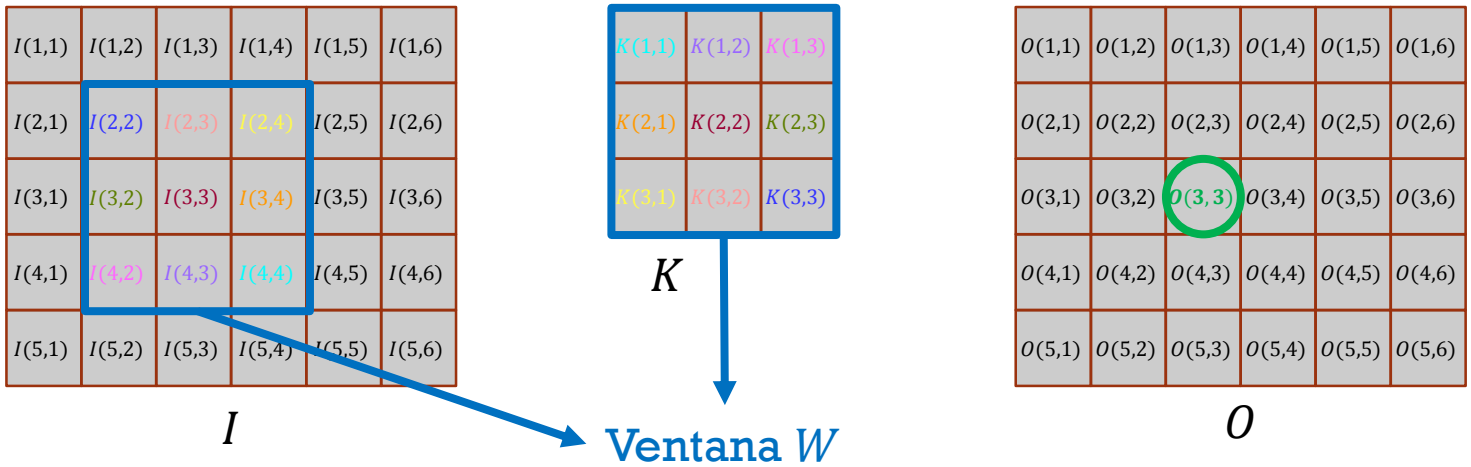
- Si el kernel es simétrico entonces la convolución es igual a la correlación
- Propiedades:
  - Conmutativo $A \otimes B = B \otimes A$
  - Asociativo $A \otimes B \otimes C = (A \otimes B) \otimes C = A \otimes (B \otimes C)$
  - Distributivo $A \otimes (\alpha B) = \alpha(A \otimes B)$
  - Lineal $A \otimes (B + C) = A \otimes B + A \otimes C$

CORRELACIÓN



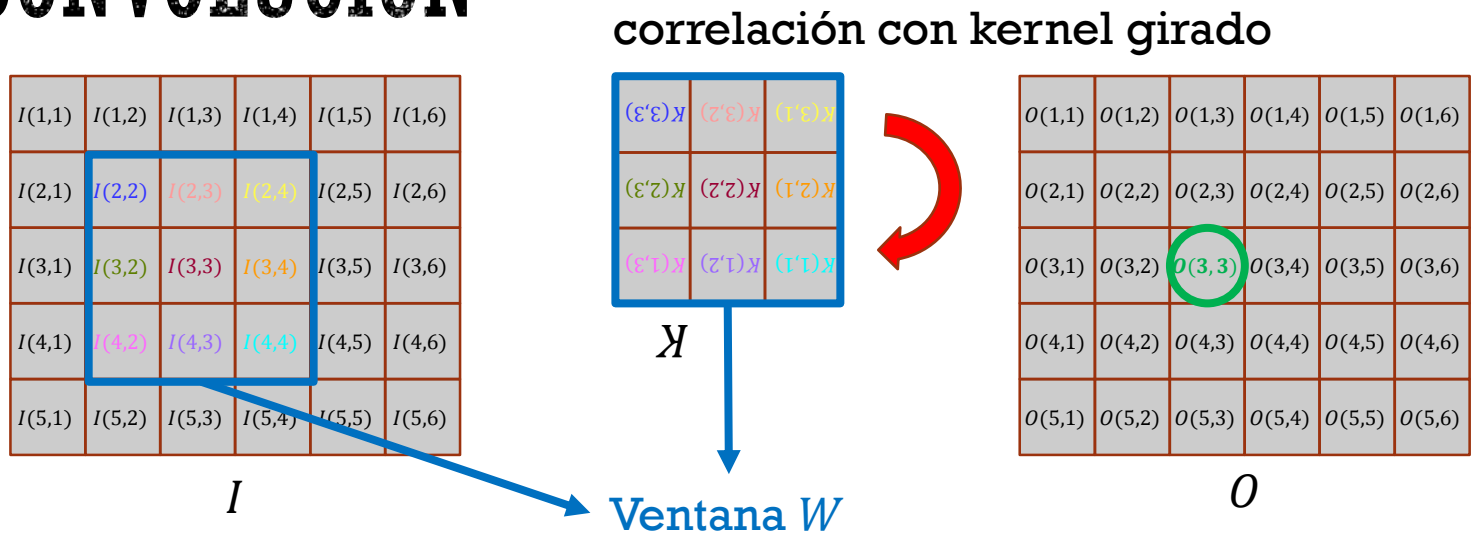
$$O(3,3) = I(2,2) * K(1,1) + I(2,3) * K(1,2) + I(2,4) * K(1,3) + I(3,2) * K(2,1) + I(3,3) * K(2,2) + I(3,4) * K(2,3) + I(4,2) * K(3,1) + I(4,3) * K(3,2) + I(4,4) * K(3,3)$$

CONVOLUCIÓN



$$O(3,3) = I(4,4) * K(1,1) + I(4,3) * K(1,2) + I(4,2) * K(1,3) + I(3,4) * K(2,1) + I(3,3) * K(2,2) + I(3,2) * K(2,3) + I(2,4) * K(3,1) + I(2,3) * K(3,2) + I(2,2) * K(3,3)$$

CONVOLUCIÓN



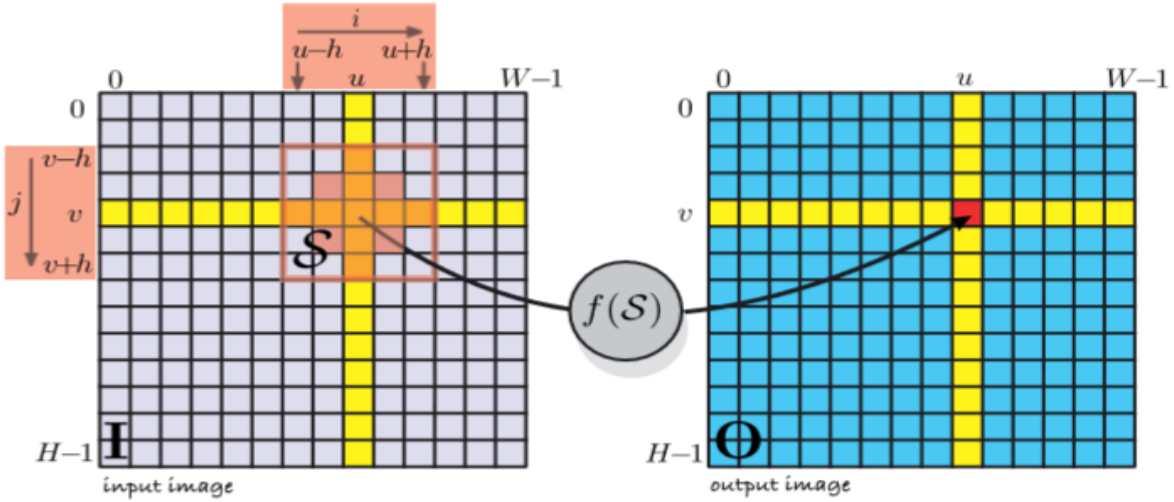
$$O(3,3) = I(4,4) * K(1,1) + I(4,3) * K(1,2) + I(4,2) * K(1,3) + I(3,4) * K(2,1) + I(3,3) * K(2,2) + I(3,2) * K(2,3) + I(2,4) * K(3,1) + I(2,3) * K(3,2) + I(2,2) * K(3,3)$$

MORFOLOGÍA MATEMÁTICA

- Cada pixel de salida es una función de un sub-conjunto de pixeles en una región circundante al pixel correspondiente en la imagen de entrada:

$$O[u,v] = f(I[u+i, v+j]), \forall (i,j) \in \mathcal{S}, \forall (u,v) \in I$$

donde $\mathcal{S}$ es una ventana de estructura, típicamente de $w \times w$ con longitud impar $w = 2h + 1$, con $h \in \mathbb{Z}^+$ la mitad de la longitud.



Operaciones morfológicas. Peter Corke. 2011.

Ejemplos:

- Erosión
- Dilatación
- Etc.

MORFOLOGÍA MATEMÁTICA (C1)

- Erosión

$$O = I \ominus S$$

$$O[u,v] = f(I[u+i, v+j]), \forall (i,j) \in S, \forall (u,v) \in I$$

con $\quad f(\cdot) = \min(\cdot)$

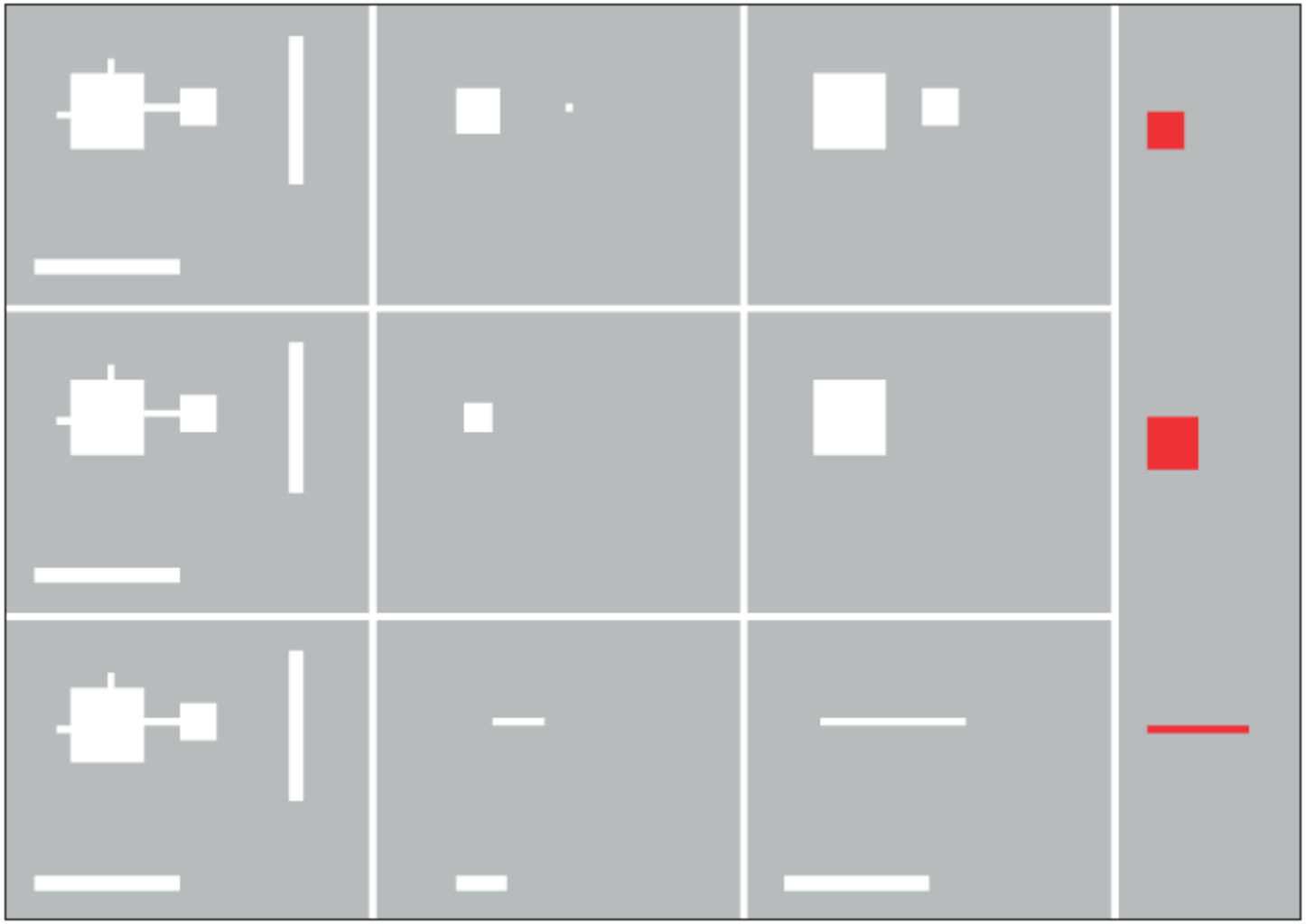
- Dilatación

$$O = I \oplus S$$

$$O[u,v] = f(I[u+i, v+j]), \forall (i,j) \in S, \forall (u,v) \in I$$

con $\quad f(\cdot) = max(\cdot)$

EROSIÓN Y DILATACIÓN



I $imerode(I, se)$ $imdilate(I, se)$ se

CORTAR IMAGEN

- Image=imread('tenis.bmp');
- subImage=Image(50:210,200:270,:);

Image

subImage

CAMBIAR DE TAMAÑO (ESCALAR)

- Image_new = imresize(Image,2.0);

Image

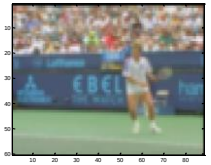
Image_new

PIRÁMIDE DE UNA IMAGEN

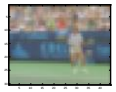
- I2 = impyramid(Image, 'reduce');
- I3 = impyramid(I2, 'reduce');
- I4 = impyramid(I3, 'reduce');

Image

I2



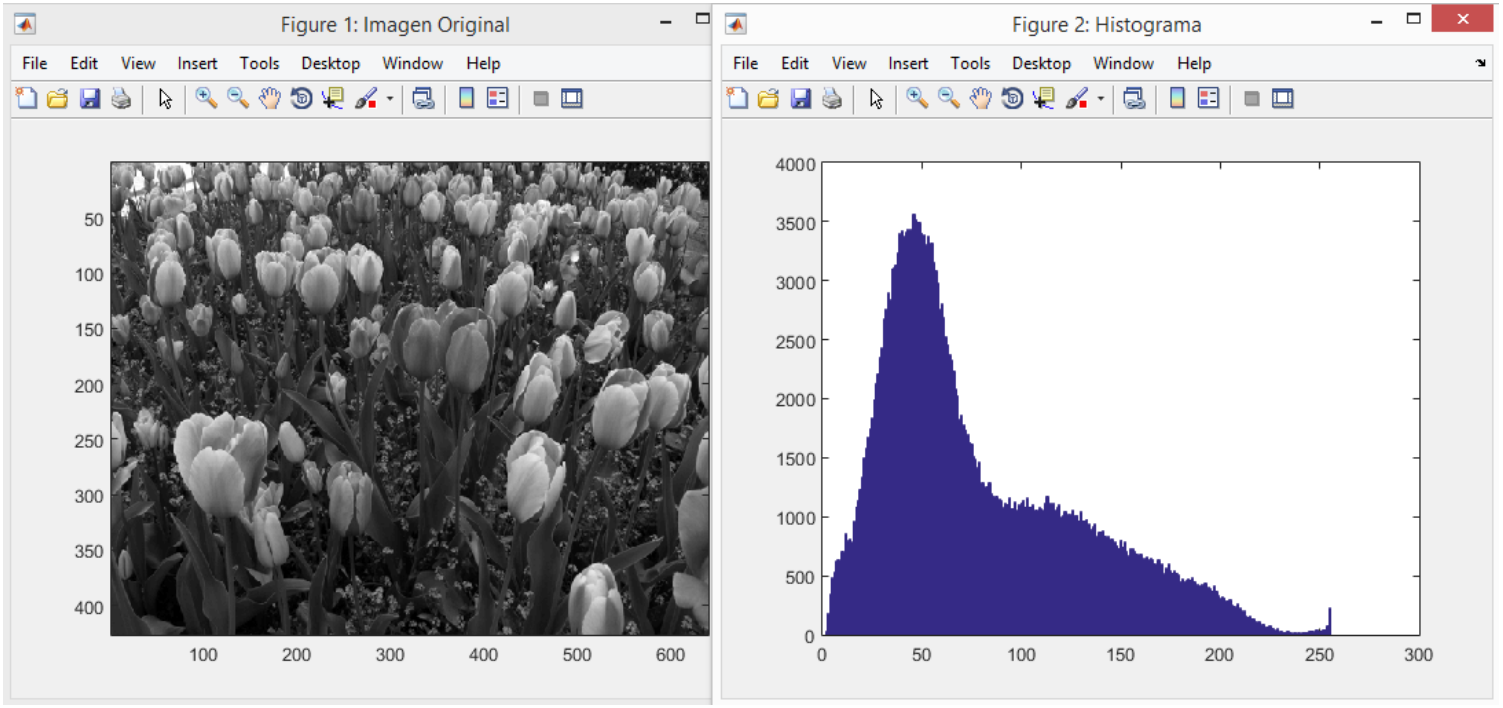
I3



I4

HISTOGRAMA DE UNA IMAGEN

- Inicializar con ceros el vector h_I de tamaño 256.
- Para todos los pixeles $\vec{x}$ de la imagen I
  - $idx = I(\vec{x})$ → en C/C++ o $idx = I(\vec{x}) + 1$ → en MatLab
  - $h_I(idx) = h(idx) + 1$



Sonka, Milan, Vaclav Hlavac, and Roger Boyle. Image processing, analysis, and machine vision. Cengage Learning, 2014.