P1

Imagen y Vídeo por Computador

. . .

Práctica 1: Imágenes en Matlab

Vicente Morell
vicente.morell@ua.es

Ingeniería Multimedia

Universitat d'Alacant
Universidad de Alicante

HUMAN ROBOTICS

ÍNDICE

Vicente Morell
vicente.morell@ua.es

Ingeniería Multimedia

Universitat d'Alacant
Universidad de Alicante

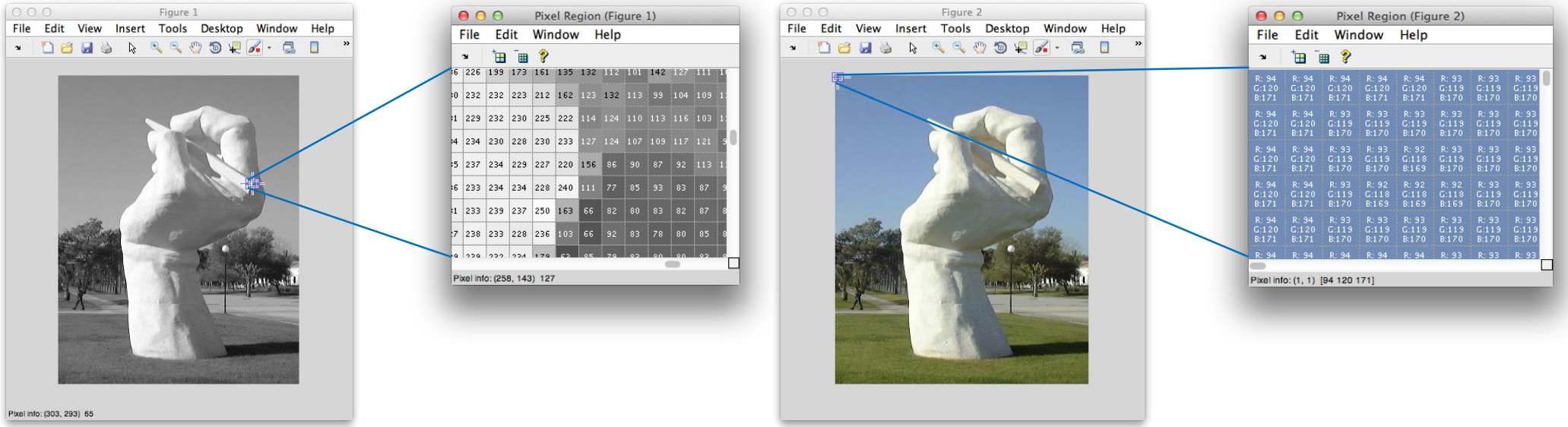
HUMAN ROBOTICS

S1: Índice de contenidos

Vicente Morell vicente.morell@ua.es

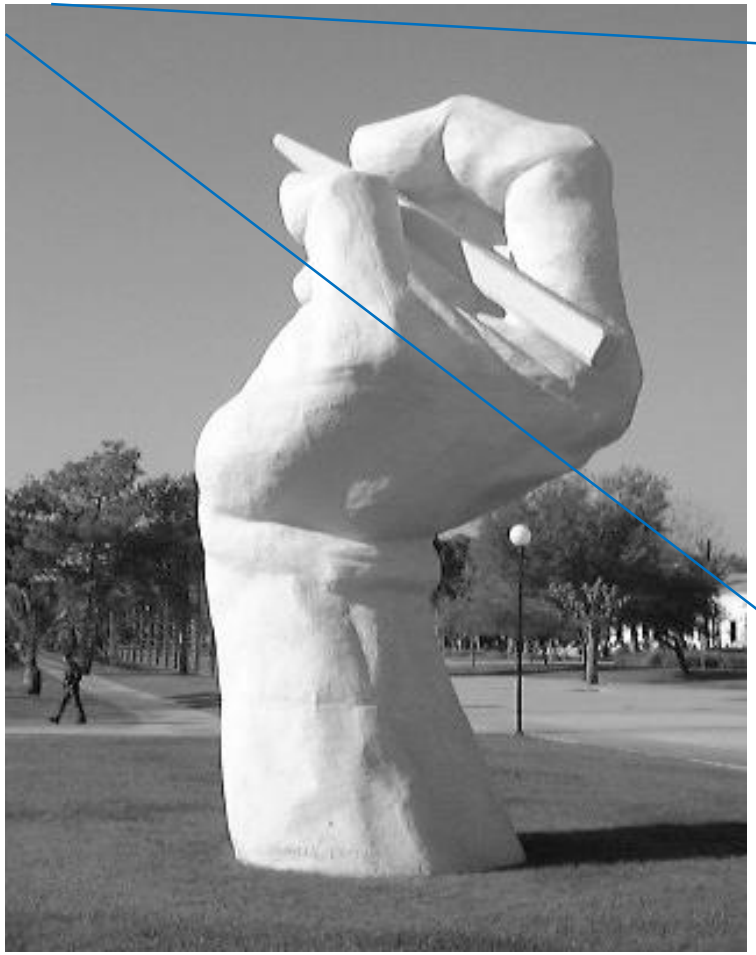
Introducción

- Una imagen de tamaño NxM píxeles se representa en Matlab como una matriz de NxM.
- Cada elemento de la matriz representa el tono de gris para imágenes en escala de grises.
- Si la imagen es en color, existen 3 matrices, cada una representa el valor de un tono de color.
- Los índices de la matriz son (f,c), donde f representa la fila y c la columna.



Vicente Morell vicente.morell@ua.es

Introducción



	1	2	3	4	5	6	7	8	9	10	11	12
1	118	118	118	118	118	117	117	117	119	118	117	116
2	118	118	117	117	117	117	117	117	117	117	117	117
3	118	118	117	117	116	117	117	117	116	116	117	118
4	118	118	117	116	116	116	117	117	117	117	117	118
5	118	118	117	117	117	117	117	117	119	118	117	117
6	118	118	118	118	118	117	117	117	119	118	117	117
7	117	118	119	119	119	118	117	117	118	118	117	118
8	117	118	119	120	120	119	118	117	116	117	117	118

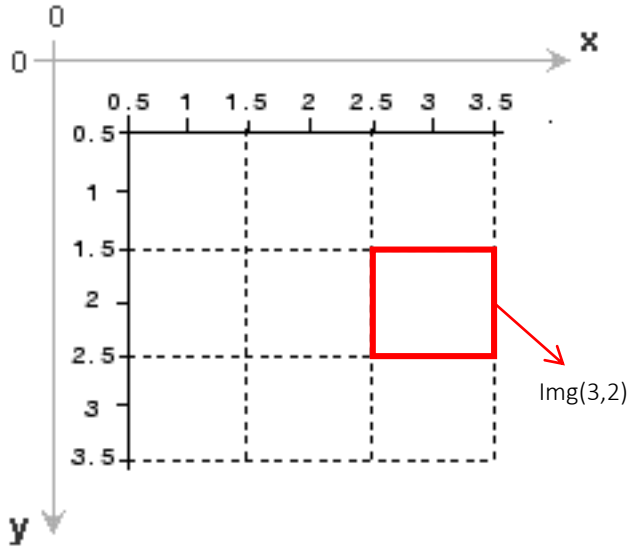
img(8,6)

Imagen y Vídeo por Computador

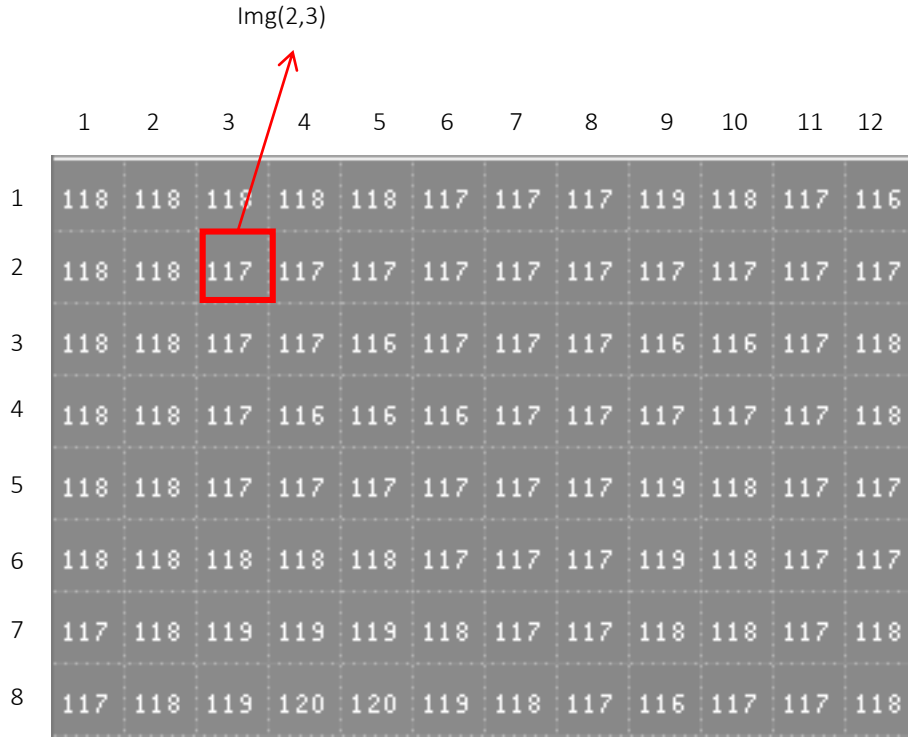
Vicente Morell vicente.morell@ua.es

Introducción

- Sistema de coordenadas intrínseco:
  - Valores **continuos**.
  - El plano imagen empieza en **(0.5,0.5)** y acaba en **(numcol+0.5,numfil+0.5)**.
  - El **centro** de cada píxel coincide con números enteros.
  - Existe una **relación** con el sistema de indexación de píxel:
    - **Im(x,y)=im(c,f)**

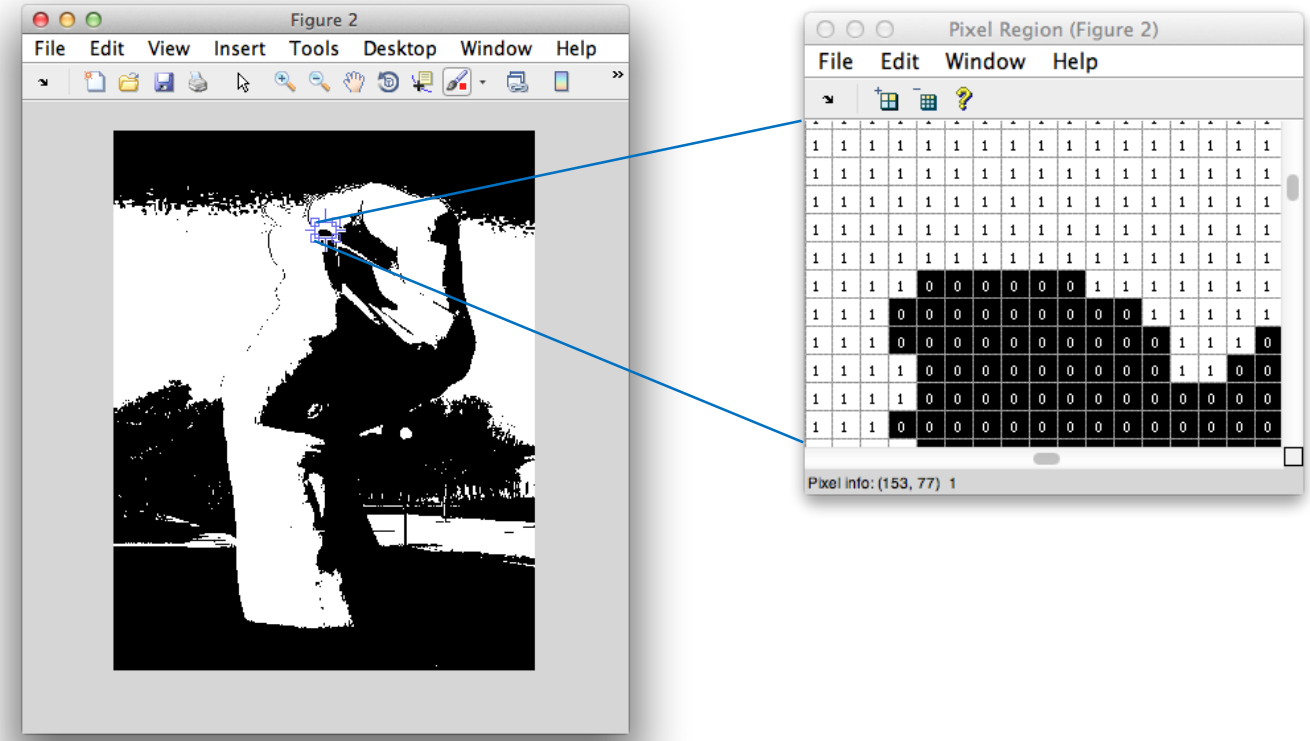


- Sistema de indexación de píxel:
  - Sistema por **defecto** para acceder al valor de un píxel.
  - Valores **discretos**.



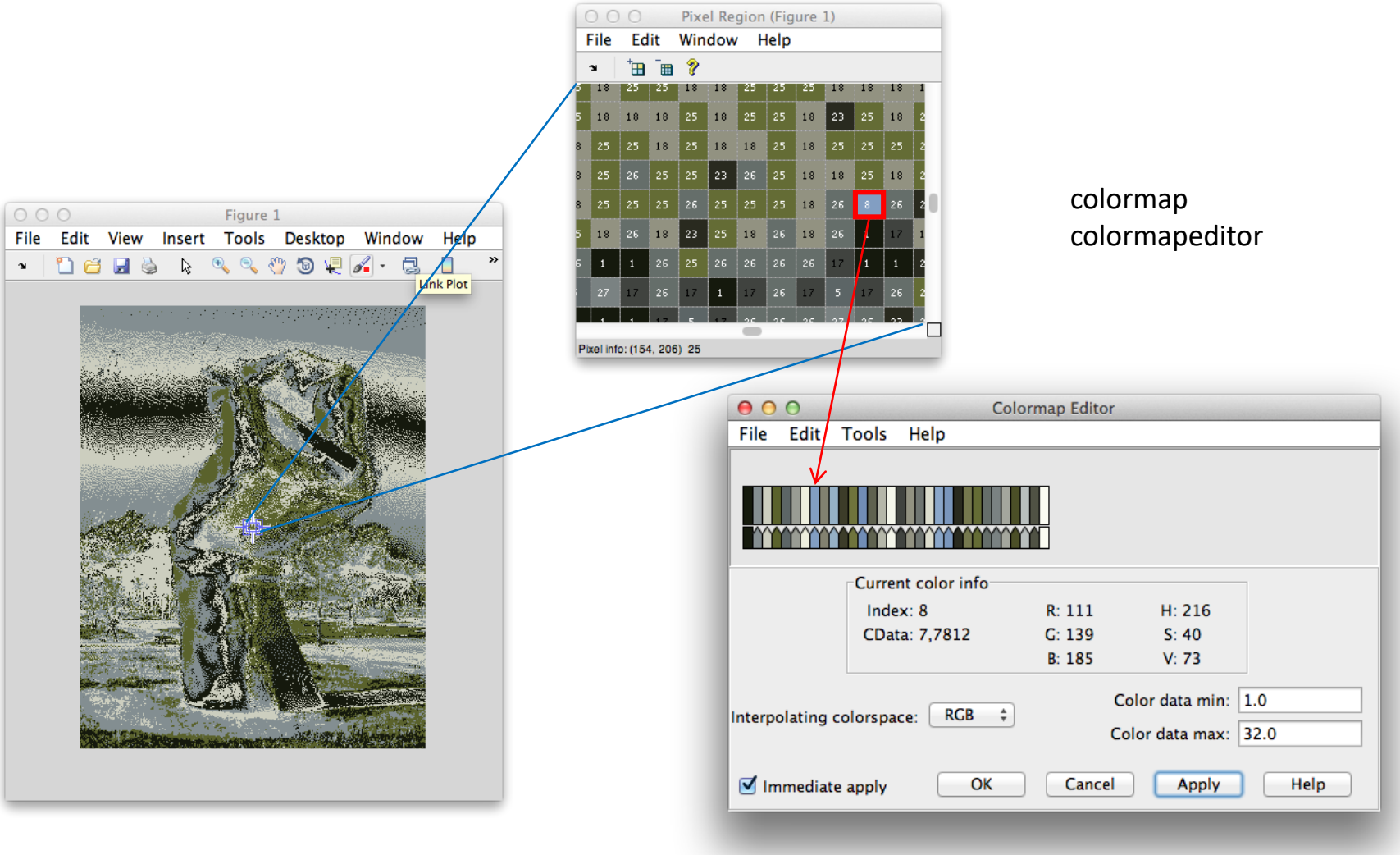
Vicente Morell vicente.morell@ua.es

Tipos: Imagen binaria



Vicente Morell vicente.morell@ua.es

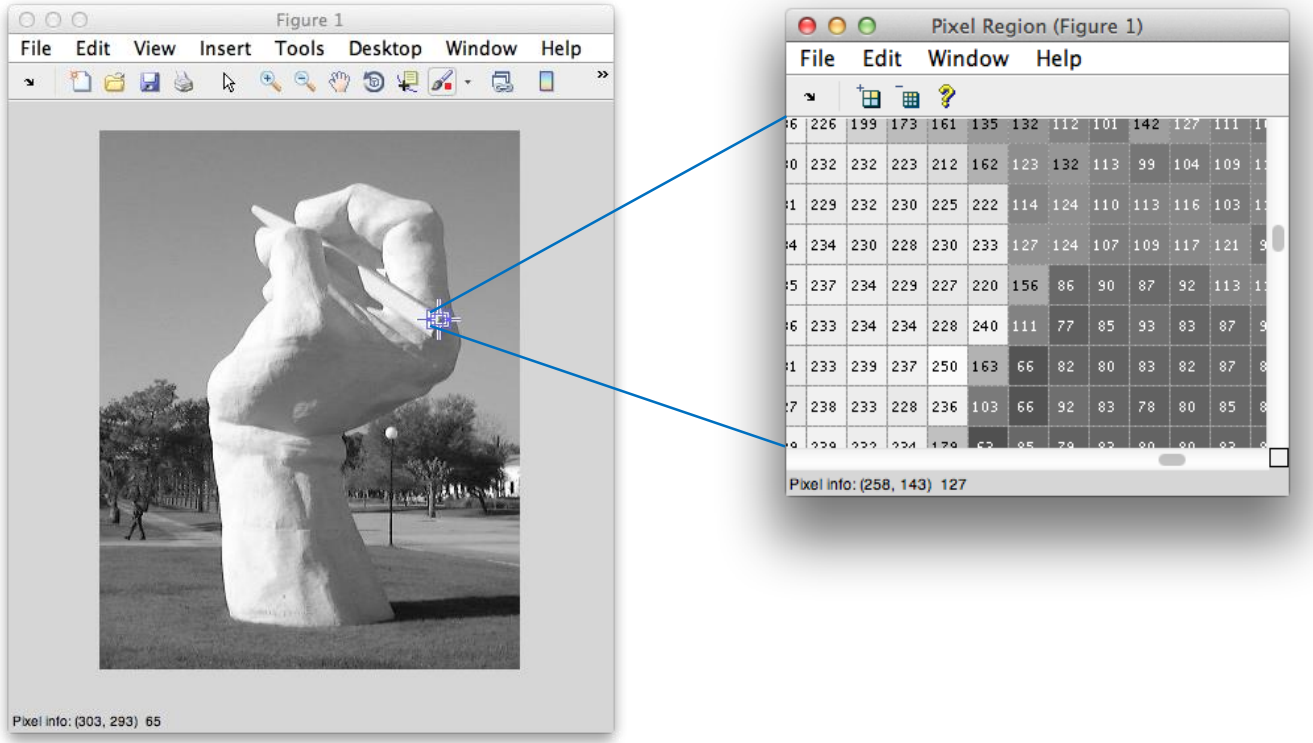
Tipos: Imagen indexada



colormap
colormapeditor

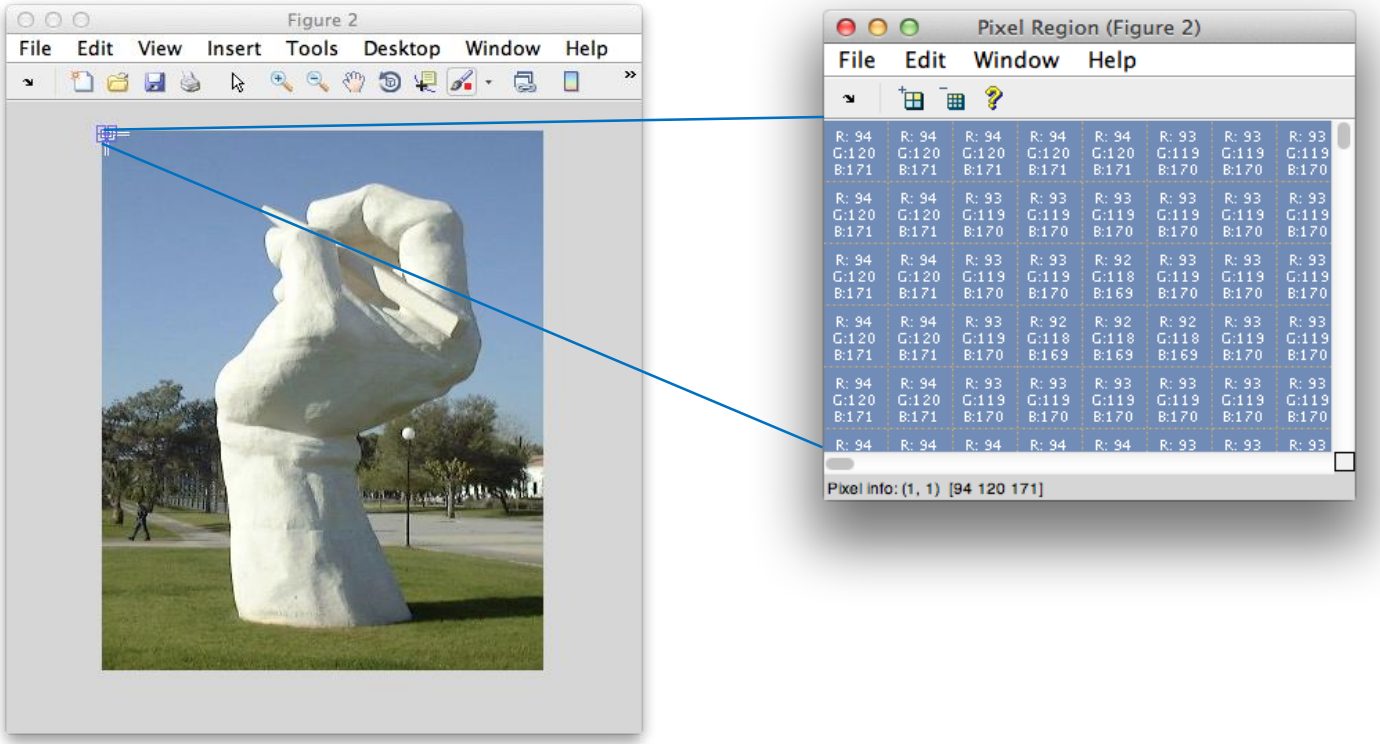
Vicente Morell vicente.morell@ua.es

Tipos: Imagen en escala de grises



Vicente Morell vicente.morell@ua.es

Tipos: Imagen en color verdadero RGB



Vicente Morell vicente.morell@ua.es

Tipo de datos del valor de un píxel

- **double** Doble precisión, números en punto flotante que varían en un rango aproximado de -10^{308} a 10^{308} (8 bytes por elemento)
- **uint8** Enteros de 8 bits en el rango de [0,255] (1 byte por elemento)
- **uint16** Enteros de 16 bits en el rango de [0, 65535] (2 bytes por elemento)
- **uint32** Enteros de 32 bits en el rango de [0, 4294967295] (4 bytes por elemento)
- **int8** Enteros de 8 bits en el rango de [-128, 127] (1 byte por elemento)
- **int16** Enteros de 16 bits en el rango de [-32768, 32767] (2 bytes por elemento)
- **int32** Enteros de 32 bits en el rango de [-2147483648,2147483647] (4 bytes por elemento)
- **single** Número en punto flotante de precisión simple, con valores aproximadamente en el rango de -10^{38} a 10^{38} (4 bytes por elemento)
- **Caracteres** (2 byte por elemento)
- **logical** Los valores son 0 ó 1 (1 byte por elemento)

Vicente Morell vicente.morell@ua.es

Abrir / Guardar imagen en Matlab


```
>> info = imfinfo(filename) ;
>> help imfinfo;
```

Información del archivo imagen

```
>> img=imread('mano_ua_gris.jpg');
>> help imread;
```

Leer el archivo imagen

```
>> imwrite(img,'mano_ua_gris.jpg');
>> help imwrite;
```

Guardar la imagen en un archivo

Vicente Morell vicente.morell@ua.es

Conversión entre tipos de imágenes

- **gray2ind** Convierte imágenes en escala de grises o binarias en imágenes indexadas.
- **ind2gray** Convierte imágenes indexadas en imágenes en escalas de grises.
- **mat2gray** Convierte matrices en imágenes en escalas de grises.
- **rgb2gray** Convierte imágenes RGB mapas de color en imágenes en escalas de grises.
- **ind2rgb** Convierte imágenes indexadas en imágenes RGB.
- **im2bw** Convierte imágenes en imágenes binarias, basándose en un umbral.
- **grayslice** Convierte imágenes en escala de grises en imágenes indexadas utilizando umbrales multinivel.

- **im2double** Convierte imágenes a precisión double.
- **im2int16** Convierte imágenes a enteros de 16-bit con signo.
- **im2single** Convierte imágenes a precisión single.
- **im2uint16** Convierte imágenes a enteros de 16-bit sin signo.
- **im2uint8** Convierte imágenes a enteros de 8-bit sin signo.

Vicente Morell vicente.morell@ua.es

Mostrar una imagen en Matlab

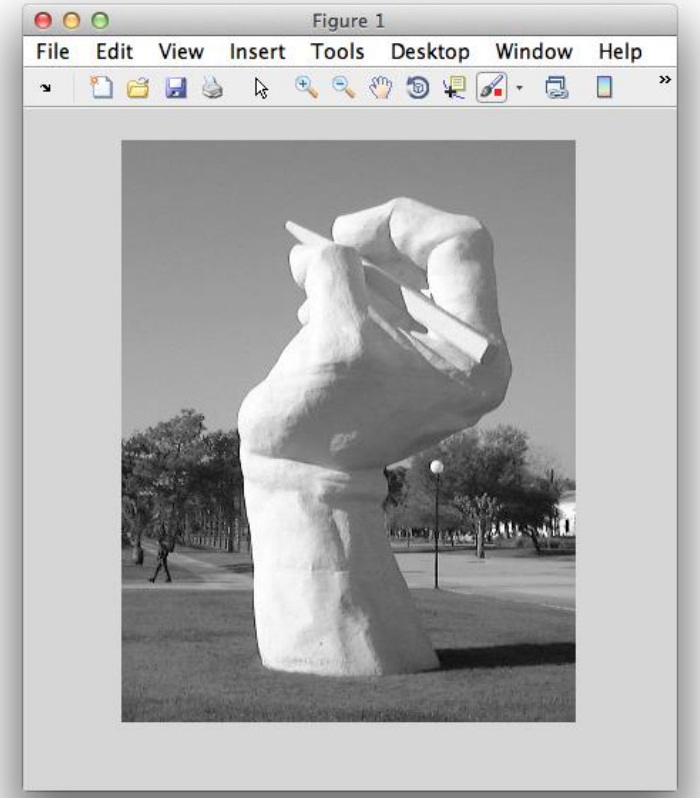
```
>> moon = imread('moon.tif');
>> imshow(moon);
```

```
>> help imshow;
```

- **imtool** Herramienta de imagen
- **imageinfo** Herramienta de información de imagen
- **imcontrast** Herramienta de ajuste de contraste
- **imdisplayrange** Heramienta de display de rango
- **imdistline** Herramienta de distancia
- **impixelinfo** Herramienta de información de píxel
- **impixelinfoval** Herramienta de información de píxel sin etiqueta de texto
- **impixelregion** Herramienta de región de píxel
- **immagbox** Ventana de zoom para el panel de scroll
- **imoverview** Herramienta de vista para la imagen mostrada en el panel de scroll

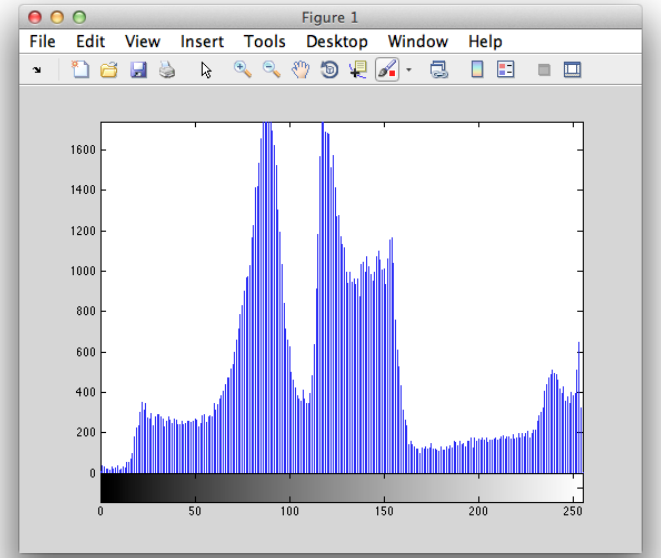
Vicente Morell vicente.morell@ua.es

Histograma




```
>> i=imread('mano_ua.jpg');
>> img=rgb2gray(i);
>> figure, imhist(img)
```

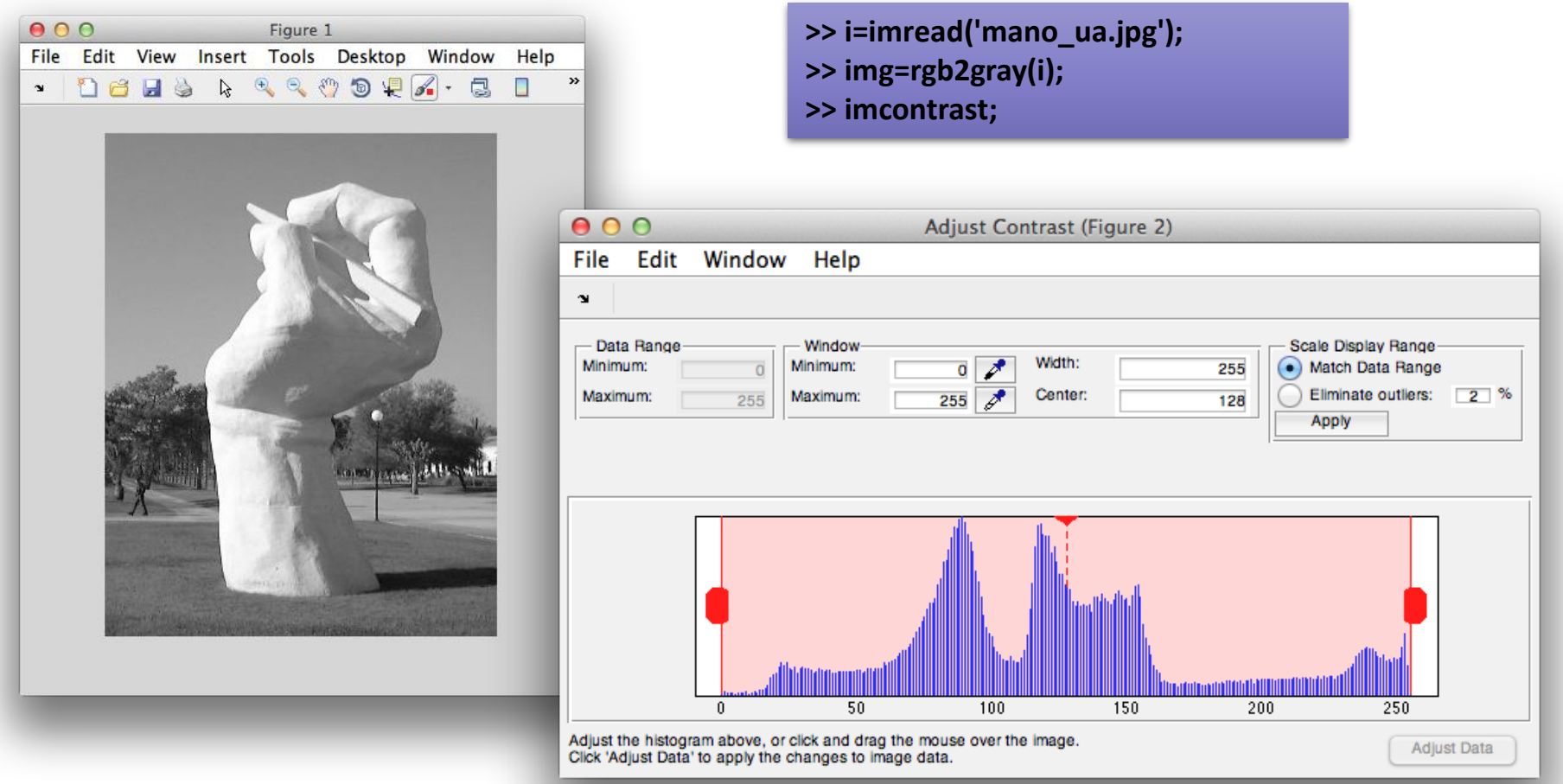
```
>> help imhist;
```



Vicente Morell vicente.morell@ua.es

Histograma: Ajustar contraste y brillo con el histograma

```
>> i=imread('mano_ua.jpg');
>> img=rgb2gray(i);
>> imcontrast;
```



Vicente Morell vicente.morell@ua.es

Histograma: Ajustar contraste y brillo con el histograma

- **imadjust** — Ajuste de los valores de intensidad de la imagen o del mapa de color
- **histeq** — Mejora el contraste usando ecualización del histograma
- **adapthisteq** — Contrast-limited adaptive histogram equalization (CLAHE)
- **imhistmatch** — Ajuste del histograma de la imagen para coincidir con el histograma N-contenedores de la imagen de referencia
- **stretchlim** — Encuentra los límites del histograma que permiten ajustar el contraste

Vicente Morell vicente.morell@ua.es

ÍNDICE

1. Sesión 1: Imágenes en Matlab
2. **Sesión 2: Transformaciones geométricas y en entorno de vecindad**
3. Sesión 3: Procesamientos morfológicos

Vicente Morell
vicente.morell@ua.es

Ingeniería Multimedia

Universitat d'Alacant
Universidad de Alicante

HUMAN ROBOTICS

S2: Índice de contenidos

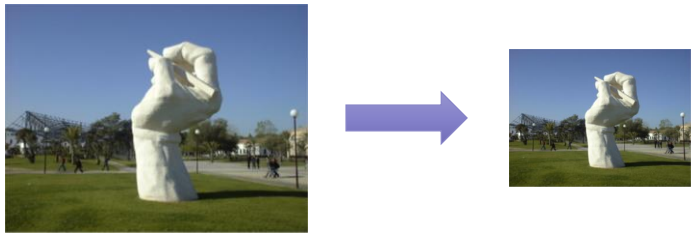
1. Modificar el tamaño de una imagen
2. Rotación de una imagen
3. Recorte de una imagen
4. Transformaciones geométricas 2D
5. Ruido y transformaciones en entorno de vecindad

Vicente Morell vicente.morell@ua.es

Modificar el tamaño de una imagen

```
>> I = imread('mano_ua.jpg');
>> J = imresize(I, 0.5);
```

```
>> help imresize;
```



- Para **reducir** el tamaño de la imagen, el segundo parámetro de imresize debe estar comprendido entre 0 y 1.
- Para **aumentar** el tamaño, el segundo parámetro debe ser mayor que 1.
- Se puede definir el nuevo **alto** y **ancho** en vez del factor de escala.
- También es posible definir un factor de escala a cada eje (por ejemplo con [0.5 2])
- Se puede indicar el **método de interpolación** utilizado para modificar el tamaño, los métodos disponibles son:
  - Interpolación por **vecino más cercano**: 'nearest'
  - Interpolación **Bilineal**: 'bilinear'
  - Interpolación **Bicúbica** (por defecto): 'bicubic'
- Es posible definir una serie de **parámetros** como: 'Antialising', 'Colormap' o 'Dither'
- Otra forma de escalar una imagen es utilizando la función **imtransform** (se verá en la Sección 4).

Vicente Morell vicente.morell@ua.es

Rotación de una imagen

```
>> I = imread('mano_ua.jpg');
>> J = imrotate(I,35,'bilinear');
```

```
>> help imrotate;
```

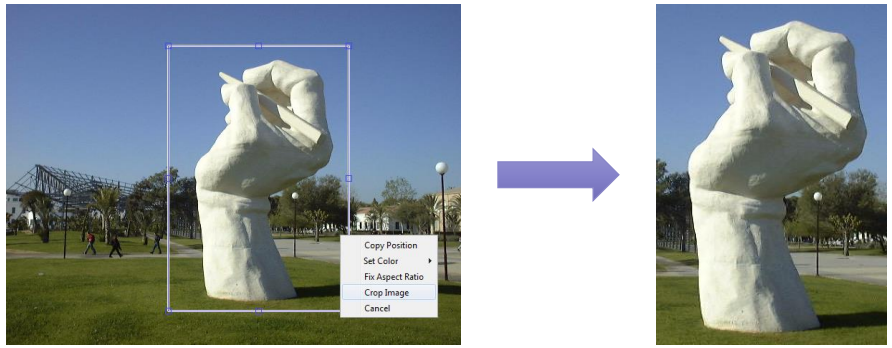
- Para rotar según el **sentido de las agujas del reloj** usar como parámetro un escalar negativo.
- Para rotar según el **sentido opuesto a las agujas del reloj** usar como parámetro un escalar positivo.
- Crea una imagen de tamaño mayor que la imagen original, rellenando los píxeles que están fuera de la imagen original con valor 0 (negro).
- Se puede obligar a que la nueva imagen tenga el mismo alto y ancho que la original con el parámetro 'crop'.
- Se puede indicar el **método de interpolación** utilizado para realizar la rotación, los métodos disponibles son:
  - Interpolación por **vecino más cercano** (por defecto): 'nearest'
  - Interpolación **Bilineal**: 'bilinear'
  - Interpolación **Bicúbica** (por defecto): 'bicubic'
- Otra forma de rotar una imagen es utilizando la función **imtransform** (se verá en la Sección 4).

Vicente Morell vicente.morell@ua.es

Recorte de una imagen

```
>> I = imread('Universidad.jpg');
>> J = imcrop(I,[229.5 62.5 246 367]);
```

```
>> help imcrop;
```

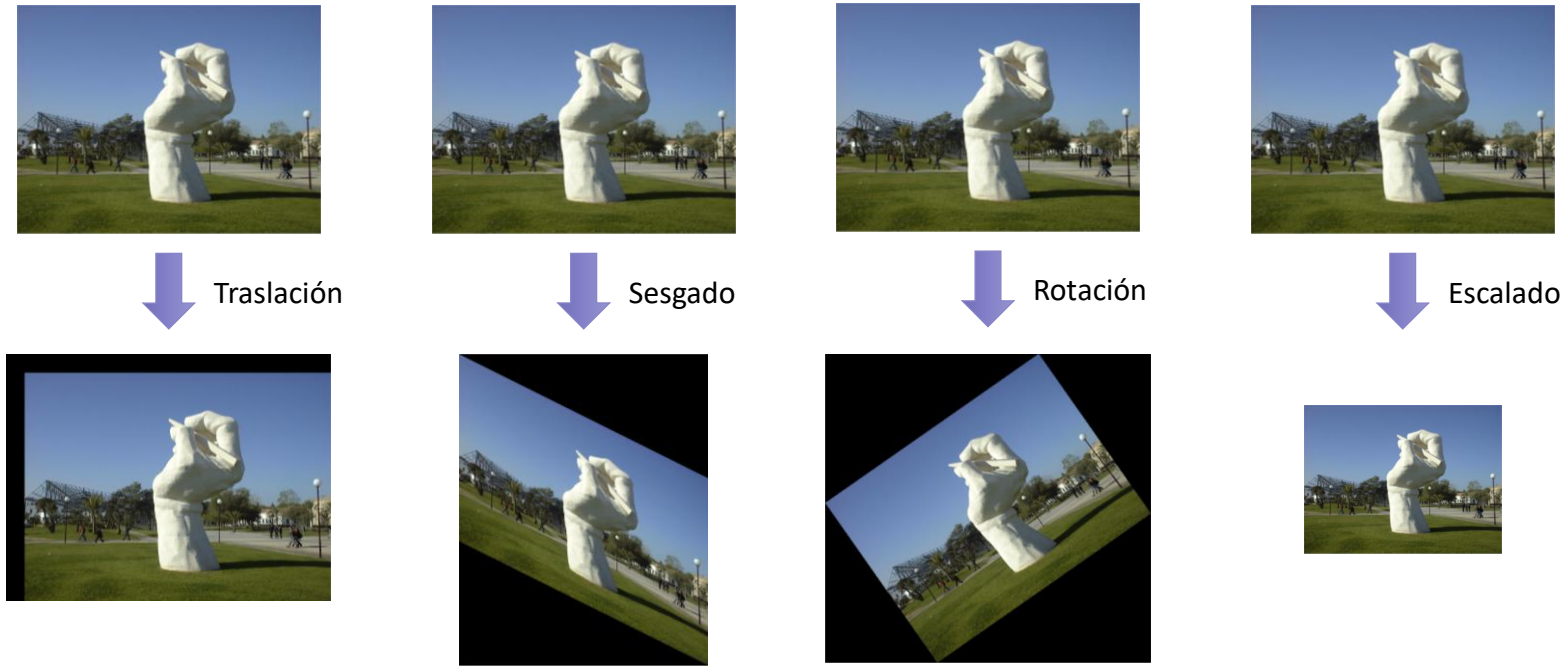


- La llamada a imcrop **sin parámetros** (o sólo con la imagen original como parámetro) muestra la imagen original y permite definir mediante el ratón la zona de recorte. Una vez definida, con el botón derecho se accede a distintas opciones.
- Se puede indicar en la llamada a imcrop el rectángulo a recortar. Para ello, se indica mediante el segundo parámetro un vector rect con la forma:
  - rect = **[x y ancho alto]**
  - **Cuidado** con el sistema de coordenadas utilizado. Por defecto, x indica las columnas, y las filas, y por lo tanto, el ancho se mide en columnas que se quiere recortar y el alto en filas que se quieren recortar.
- Otra forma de recortar una imagen es seleccionando las filas y columnas que se quieran recortar directamente de la matriz que almacena los datos de la imagen.

Vicente Morell vicente.morell@ua.es

Transformaciones geométricas 2D

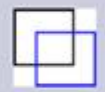
- Proceso general en tres pasos:
  1. Definir los **parámetros** de la transformación espacial.
  2. Crear una **estructura de transformación**, denominada TFORM, que define el tipo de transformación deseada.
  3. Realizar la transformación con la función **imtransform**.



Vicente Morell vicente.morell@ua.es

Trans. Geo: Definir parámetros de la transformación

- Se proporcionan dos formas distintas para definir los parámetros de la transformación:
  1. Usando una **matriz de transformación**.

Transformación Afín	Ejemplo	Matriz de transformación	
Traslación		$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	t_x especifica el desplazamiento sobre el eje X t_y especifica el desplazamiento sobre el eje Y
Escalado		$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	s_x especifica el factor de escala sobre el eje X s_y especifica el factor de escala sobre el eje Y
Sesgado		$\begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	sh_x especifica el factor de sesgado sobre el eje X sh_y especifica el factor de sesgado sobre el eje Y
Rotación		$\begin{bmatrix} \cos(q) & \sin(q) & 0 \\ -\sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	q especifica el ángulo de rotación

  2. Usando un **conjunto de puntos**.
     - Se elige un conjunto de puntos en la imagen original y cómo queremos que queden en la imagen destino. Para transformaciones afines se deben indicar tres puntos no coplanares.

Vicente Morell vicente.morell@ua.es

Trans. Geo: Crear una estructura de transformación

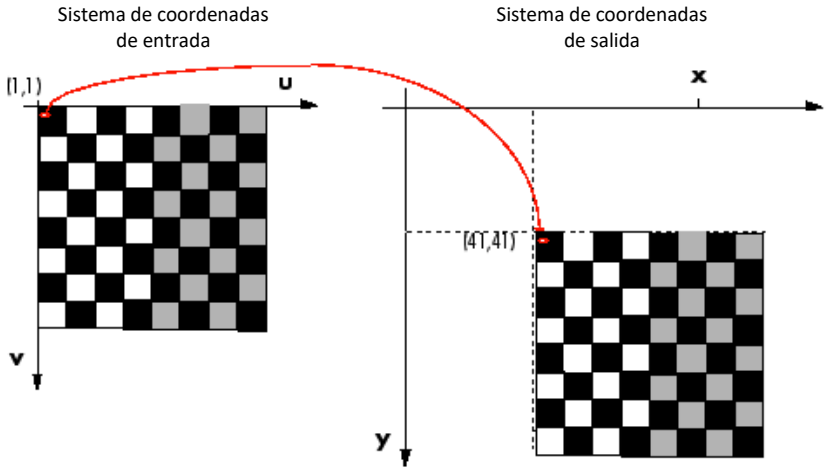
- Para crear la estructura de transformación **TFORM** se utiliza la función **maketform**.
- Se define el tipo de transformación de entre los siguientes:
  - '**affine**': transformación afín. Incluye traslación, rotación, escalado y sesgado. Las líneas rectas se mantienen rectas y las paralelas mantienen su paralelismo, pero los rectángulos pueden transformarse a paralelogramos.
  - '**projective**': transformación proyectiva. Las líneas rectas se mantienen rectas, pero las paralelas convergen hacia puntos de fuga. Los puntos de fuga pueden situarse dentro o fuera de la imagen, incluso en el infinito.
  - '**box**': caso especial de la transformación afín, donde cada dimensión se escala de manera independiente.
  - '**custom**': transformación definida por el usuario.
  - '**composite**': composición de dos o más transformaciones.

Vicente Morell vicente.morell@ua.es

Trans. Geo: Realizar la transformación

```
>> I = imread('mano_ua.jpg');
>> tform = maketform('affine',[1 0 0; .5 1 0; 0 0 1]);
>> J = imtransform(I,tform);
```

```
>> help imtransform;
```



- Para realizar la transformación se utiliza la función **imtransform**.
- Es posible definir el tipo de interpolación de entre los siguientes:
  - '**nearest**'.
  - '**blinear**' (por defecto).
  - '**bicubic**'.
- También se pueden añadir otros parámetros a la función, como '**FillValues**' que permite seleccionar un color en los casos en que la transformación geométrica requiera rellenar píxeles que no existían en la imagen original.

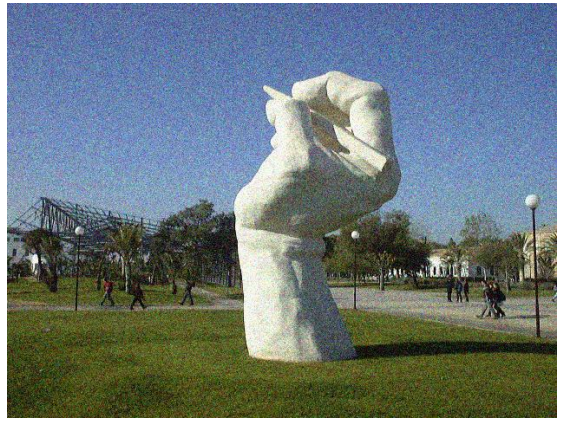
Vicente Morell vicente.morell@ua.es

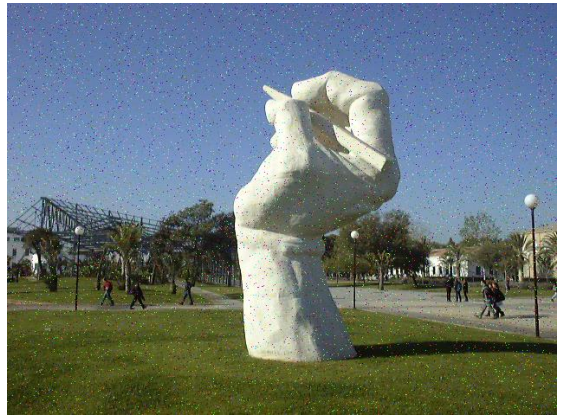
Transformaciones: Añadir ruido a una imagen

```
>> I = imread('mano_ua.jpg');
>> J = imnoise(I,'salt & pepper',0.02);
```

```
>> help imnoise;
```

- Tipos de ruido disponibles en imnoise:
  - "**gaussian**", con los parámetros:
    - Media.
    - Varianza.
  - "**localvar**", con parámetro:
    - Varianza local.
  - "**poisson**", sin parámetros.
  - "**salt & pepper**", con parámetro:
    - Densidad de ruido.
  - "**speckle**", con parámetro:
    - Varianza.

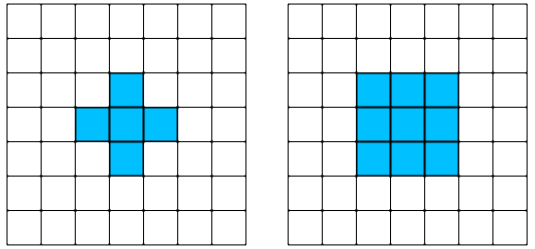




Vicente Morell vicente.morell@ua.es

Transformaciones: Eliminar ruido

- Transformaciones en **entorno de vecindad**:
  - Uso de máscaras de convolución para operar sobre un entorno de vecindad de cada píxel de la imagen.



  - Filtro lineal:
    - **fspecial** para definir el tipo de filtro que se va a realizar, así como el tamaño de la máscara de convolución (por defecto de 3x3):
      - Filtro de media: '**average**'.
      - Filtro Gaussiano: '**gaussian**'.
    - **imfilter**: realiza la convolución con los filtros definidos mediante fspecial.
  - Filtro de mediana: **medfilt2**.
    - Define el tamaño de la máscara de convolución mediante el parámetro [m n]. Por defecto realiza el filtro de mediana con una máscara de 3x3.

Vicente Morell vicente.morell@ua.es

ÍNDICE

1. Sesión 1: Imágenes en Matlab
2. Sesión 2: Transformaciones geométricas y en entorno de vecindad
3. **Sesión 3: Procesamientos morfológicos**

Vicente Morell
vicente.morell@ua.es

Ingeniería Multimedia

Universitat d'Alacant
Universidad de Alicante

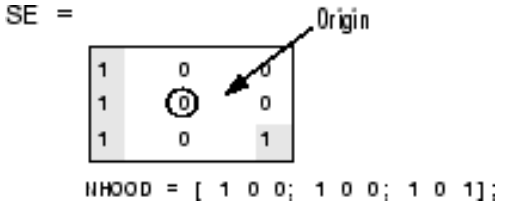
HUMAN ROBOTICS

S3: Índice de contenidos

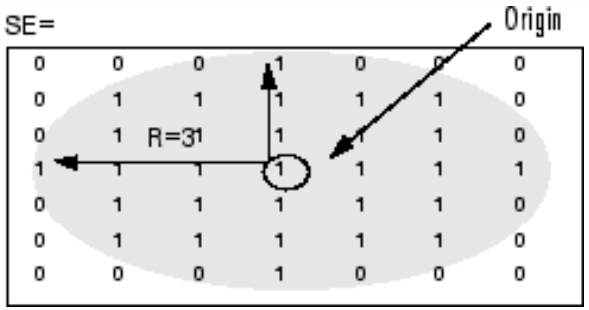
Vicente Morell vicente.morell@ua.es

Elemento estructurante

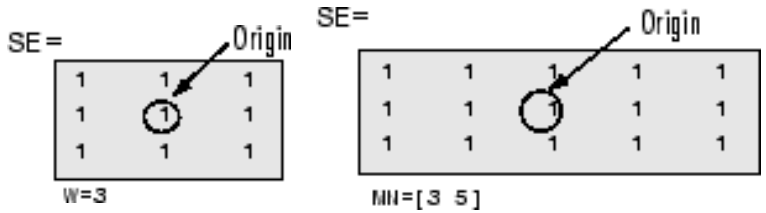
>> SE = strel('arbitrary', NHOOD);



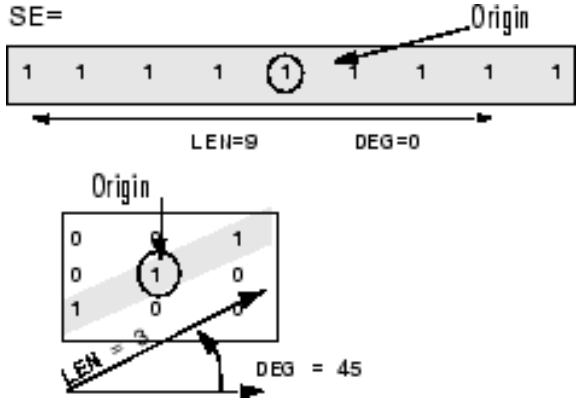
>> SE = strel('disk', radio);



>> SE = strel('square', lado);
>> SE = strel('rectangle', [filas,columnas]);



>> SE = strel('line', longitude, ángulo);



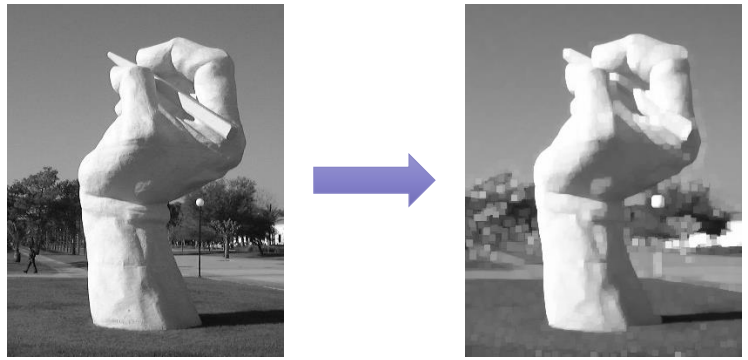
Vicente Morell vicente.morell@ua.es

Dilatación

```
>> I = imread('mano_ua_gris.jpg');
>> SE = strel('square',7);
>> BW2 = imdilate(I,SE);
```

```
>> help imdilate;
```

- Para **dilatar** una imagen, se usa la función **imdilate**. Esta función acepta dos parámetros inicialmente:
  - La **imagen** de origen que se desea procesar (en escala de grises o binario).
  - Un objeto que representa un **elemento estructurante**, devuelto por la función **strel**, o una matriz binaria que defina el vecindario de un elemento estructurante.



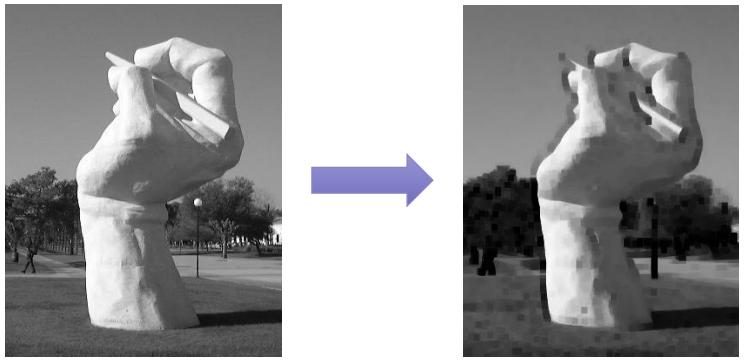
Vicente Morell vicente.morell@ua.es

Dilatación

```
>> I = imread('mano_ua_gris.jpg');
>> SE = strel('square',7);
>> BW3 = imerode(I,SE);
```

```
>> help imerode;
```

- Para **erosionar** una imagen, se usa la función **imerode**. Esta función acepta dos parámetros inicialmente:
  - La **imagen** de origen que se desea procesar (en escala de grises o binario).
  - Un objeto que representa un **elemento estructurante**, devuelto por la función **strel**, o una matriz binaria que defina el vecindario de un elemento estructurante.



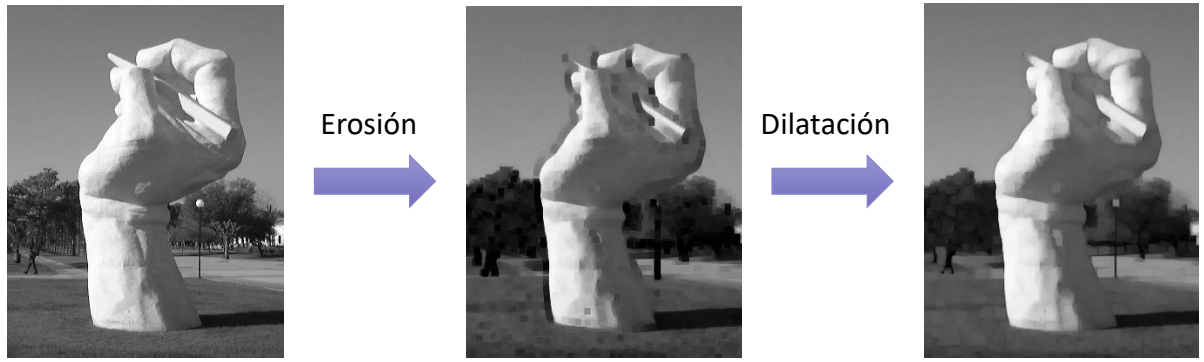
Vicente Morell vicente.morell@ua.es

Apertura

```
>> I = imread('mano_ua_gris.jpg');
>> SE = strel('square',7);
>> J = imopen(I,SE);
```

```
>> help imopen;
```

- Se puede realizar una apertura con la función **imopen**. Esta apertura es una erosión seguida de una dilatación utilizando el **mismo elemento estructurante**.
- También se puede realizar una apertura aplicando sobre la imagen original una erosión con el elemento estructurante que se quiera, y posteriormente, sobre la imagen erosionada, aplicar una dilatación también con el elemento estructurante deseado.



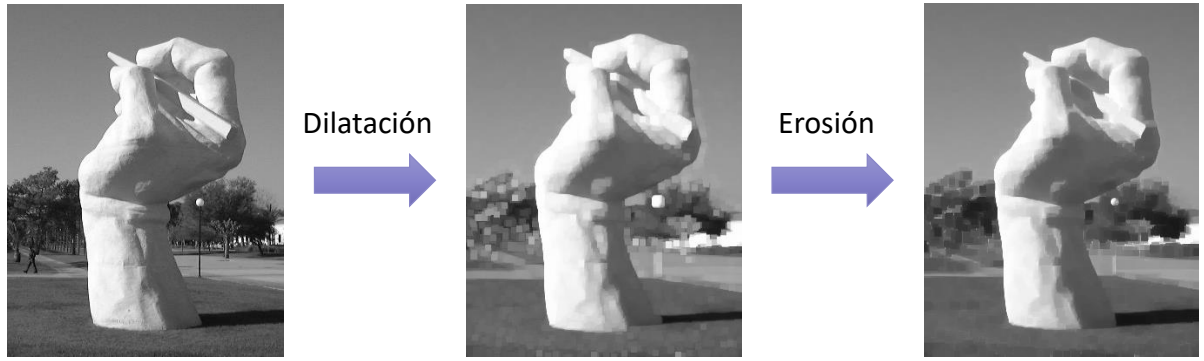
Vicente Morell vicente.morell@ua.es

Cierre

```
>> I = imread('mano_ua_gris.jpg');
>> SE = strel('square',7);
>> J = imclose(I,SE);
```

```
>> help imclose;
```

- Se puede realizar un cierre con la función **imclose**. Este cierre es una dilatación seguida de una erosión utilizando el **mismo elemento estructurante**.
- También se puede realizar un cierre aplicando sobre la imagen original una dilatación con el elemento estructurante que se quiera, y posteriormente, sobre la imagen dilatada, aplicar una erosión también con el elemento estructurante deseado.



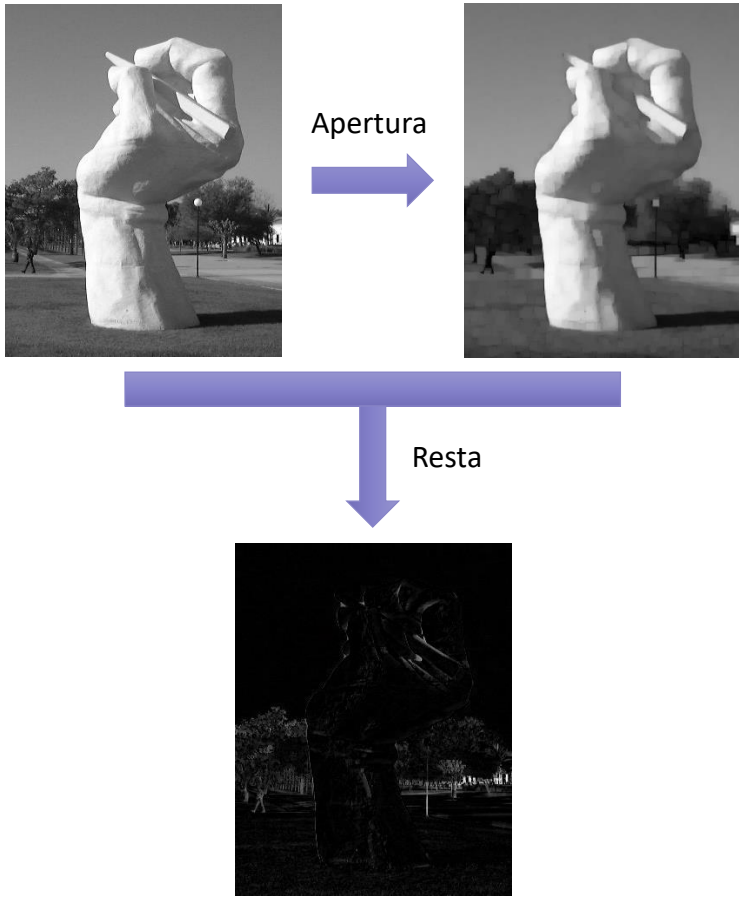
Vicente Morell vicente.morell@ua.es

Transformación top-hat

```
>> I = imread('mano_ua_gris.jpg');
>> SE = strel('square',7);
>> J = imtophat(I,SE);
```

```
>> help imtophat;
```

- Consiste en descubrir aquellas estructuras de la imagen que han sido eliminadas en el filtrado de apertura.
- El top-hat es el residuo entre la imagen original y una apertura.



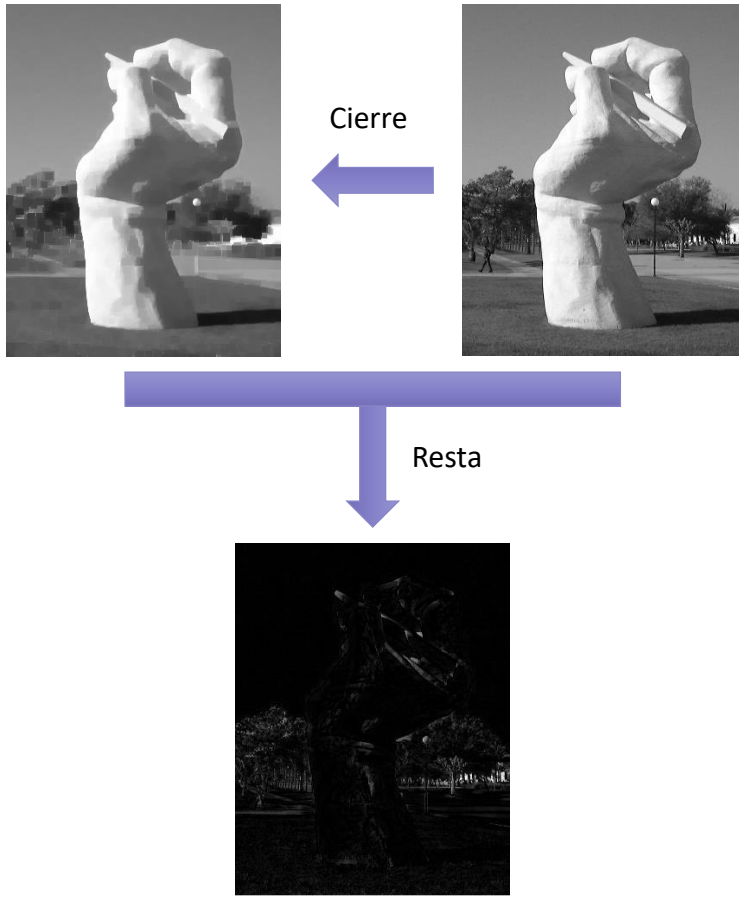
Vicente Morell vicente.morell@ua.es

Transformación top-hat dual

```
>> I = imread('mano_ua_gris.jpg');
>> SE = strel('square',7);
>> J = imbothat(I,SE);
```

```
>> help imbothat;
```

- Consiste en descubrir aquellas estructuras de la imagen que han sido eliminadas en el filtrado de cierre.
- El top-hat dual (o bottom-hat como es llamado en Matlab) es el residuo entre el cierre y la imagen original.

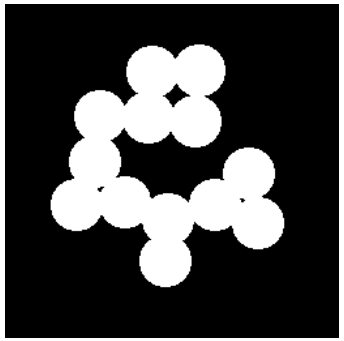
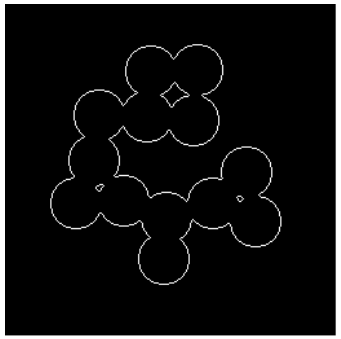
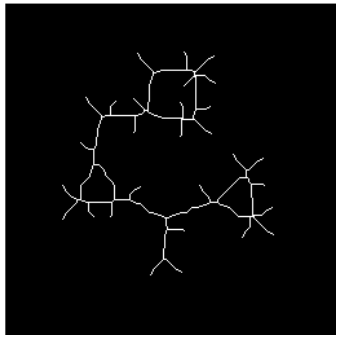


Vicente Morell vicente.morell@ua.es

Operaciones morfológicas en imágenes binarias

- Se utiliza la función bwmorph:
  - Permite realizar todas las operaciones vistas hasta ahora y muchas más

'bothat'	'erode'	'shrink'
'bridge'	'fill'	'skel'
'clean'	'hbreak'	'spur'
'close'	'majority'	'thicken'
'diag'	'open'	'thin'
'dilate'	'remove'	'tophat'

Vicente Morell vicente.morell@ua.es

Otras funciones

- bwareaopen
- bwhitmiss
- imfill
- imclearborder
- imcomplement
- imregionalmax
- imregionalmin
- imextendedmax
- imextendedmin
- imhmax
- Imhmin

- http://www.mathworks.es/es/help/images/morphological-filtering.html

Vicente Morell vicente.morell@ua.es

P1

Imagen y Vídeo por Computador

. . .

Práctica 1: Imágenes en Matlab

Vicente Morell
vicente.morell@ua.es

Ingeniería Multimedia

Universitat d'Alacant
Universidad de Alicante

HUMAN ROBOTICS