



Algoritmos de Visión Artificial con Matlab

Sesión 2. Preprocesamiento

Ignacio Alvarez García

Rafael C. González de los Reyes



Indice

- ❑ Estructura del curso
- ❑ Preprocesamiento de imágenes
- ❑ Mejora de contraste y elim. de fondo
- ❑ Filtrado de ruidos
- ❑ Transformaciones geométricas
- ❑ Ejemplos con Matlab



Indice

- ❑ **Estructura del curso**

- ❑ Preprocesamiento de imágenes

- ❑ Mejora de contraste y elim. de fondo

- ❑ Filtrado de ruidos

- ❑ Transformaciones geométricas

- ❑ Ejemplos con Matlab



Estructura del curso

Sesión 1 (3h)

- Sesión 1.1
 - Introducción a la visión por computador.
 - Elementos de un sistema de visión por computador.
 - Etapas del procesamiento de imágenes.
 - Formatos de almacenamiento de imágenes en memoria y disco.
 - Funcionalidades básicas de Matlab para la manipulación de imágenes.

- Sesión 1.2
 - Preprocesamiento de imágenes: introducción.
 - Mejora de contraste.
 - Zoom e interpolación.
 - Filtrado de ruidos.
 - Ejemplos con Matlab.

- Sesión 1.3
 - Resaltado de bordes.
 - Binarización y segmentación.
 - Operaciones con imágenes binarizadas.
 - Ejemplos con Matlab.

Sesión 2 (2h)

- Sesión 2.1
 - Búsqueda y ajuste de rectas.
 - Obtención de regiones.
 - Descriptores de regiones.
 - Uso de los descriptores.
 - Ejemplos con Matlab.

- Sesión 2.2
 - Calibración de cámaras.
 - Obtención de información 3D

- Sesión 2.3
 - Programación C/C++ con OpenCV
 - Inteligencia Artificial
 - Conclusiones



Indice

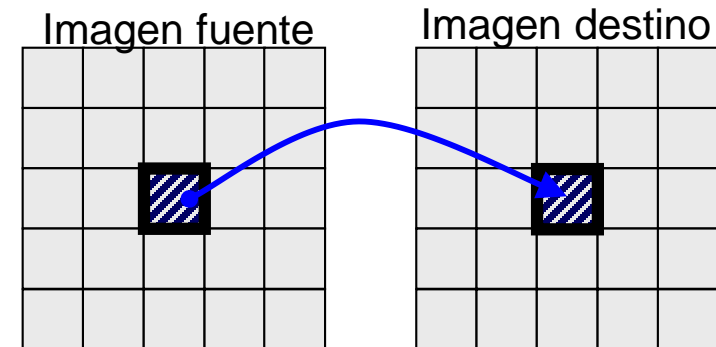
- ❑ Estructura del curso
- ❑ **Preprocesamiento de imágenes**
- ❑ Mejora de contraste y elim. de fondo
- ❑ Filtrado de ruidos
- ❑ Transformaciones geométricas
- ❑ Ejemplos con Matlab

Preprocesamiento de imágenes

- ❑ **Preprocesamiento:** primera etapa del procesamiento de una imagen.
- ❑ **Entrada:** imagen capturada.
- ❑ **Salida:** imagen mejorada sobre la que aplicar algoritmos de segmentación, reconocimiento, ...
- ❑ **Objetivos:**
 - Resaltar las características 'interesantes' de la imagen.
 - Reducir la influencia del ruido y el fondo (partes no 'interesantes').
 - Rectificar distorsiones.
 - Mejorar la calidad de la imagen.
- ❑ Las técnicas de preprocesamiento se pueden clasificar en 3 categorías:
 - Transformaciones de brillo del pixel (operaciones puntuales)
 - Transformaciones de vecindad del pixel.
 - Transformaciones geométricas.

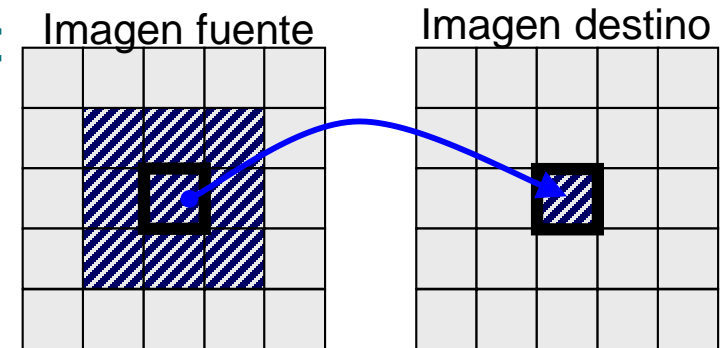
Preprocesamiento de imágenes

- Operaciones puntuales:
 - Sólo utilizan el valor del mismo pixel.
 - Suelen utilizarse para mejorar la visualización.
- Operaciones puntuales típicas:
 - Transformaciones del nivel de gris:
 - Transformaciones lineales del nivel de gris
 - Mejora del contraste
 - Ecuación de histograma
 - Mejora local del contraste
 - Aritmética de Imágenes:
 - Suma y promediado
 - Resta
 - Mínimo y máximo de dos imágenes
 - Corrección de la iluminación



Preprocesamiento de imágenes

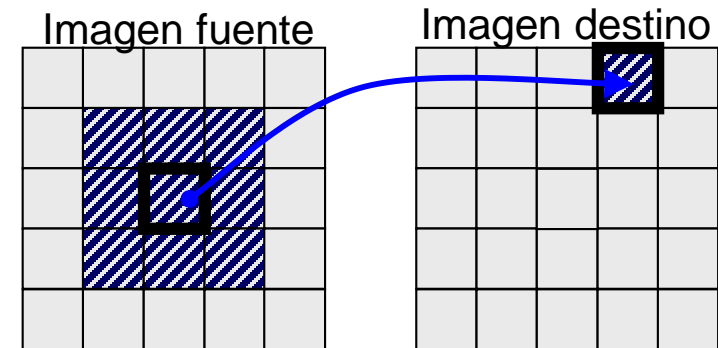
- Transformaciones de vecindad:
 - Utilizan el valor de un pixel y de los adyacentes.
 - Suelen utilizarse para reducir ruido y destacar características.
- Operaciones de vecindad típicas:
 - Filtrado (reducción de ruidos):
 - Lineal: media, gaussiana, etc.
 - No lineal: mediana, tophat, bottomhat, etc.
 - Resaltado de contornos:
 - Filtros detectores de contornos
 - Filtros detectores de máximos/mínimos



Preprocesamiento de imágenes

- Transformaciones geométricas:
 - Cambian la posición de los pixels.
 - Suelen utilizarse para eliminar distorsiones.
 - Precisan interpolación.

- Transformaciones geométricas típicas:
 - Zoom
 - Cambio de perspectiva
 - Corrección de distorsiones





Indice

- ❑ Estructura del curso
- ❑ Preprocesamiento de imágenes
- ❑ **Mejora de contraste y elim. de fondo**
- ❑ Filtrado de ruidos
- ❑ Transformaciones geométricas
- ❑ Ejemplos con Matlab

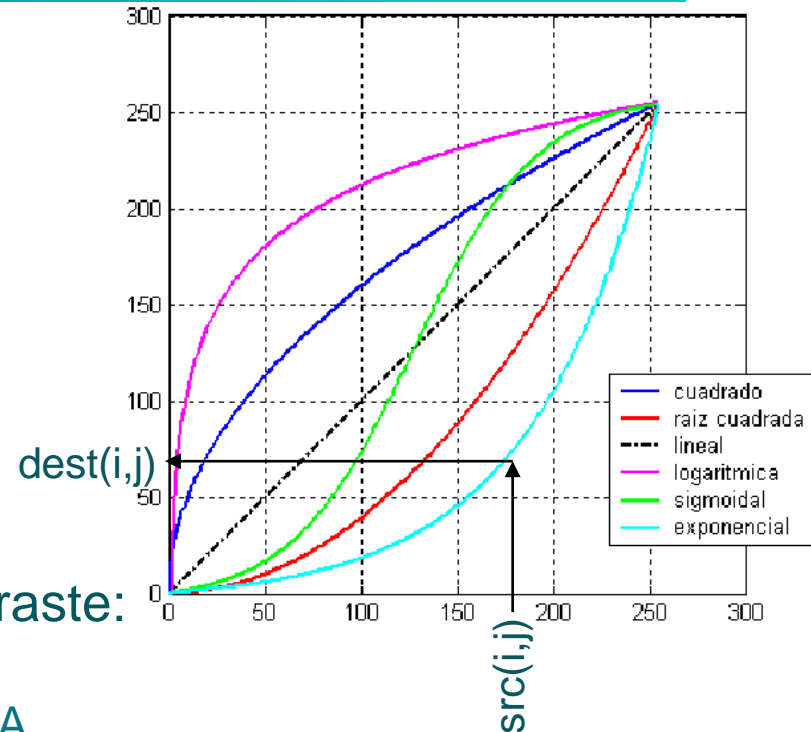
Mejora de contraste y elim. de fondo

- ❑ Cambian el valor de un píxel en función sólo de su valor anterior: $\text{dest}(i,j)=F_n(\text{src}(i,j))$
- ❑ Posibilidades para Mejora de contraste y elim. de fondo:
 - Aplicación de funciones:
 - Aumento/reducción de brillo
 - Aumento/reducción de contraste
 - Corrección gamma
 - ...
 - Aplicación de funciones mediante tablas look-up (LUT)
- ❑ Algunas técnicas aplican una LUT que es función de la propia imagen:
 - Ecuación de histograma
- ❑ Algunas técnicas utilizan múltiples imágenes:
 - Promediado, diferencia, corrección de iluminación de fondo

Mejora de contraste y elim. de fondo

□ Aplicación de funciones:

- $dest(i,j)=Fn(src(i,j)),$
 $i=1..nfilas$
 $j=1..ncols$



□ Funciones lineales para brillo y contraste:

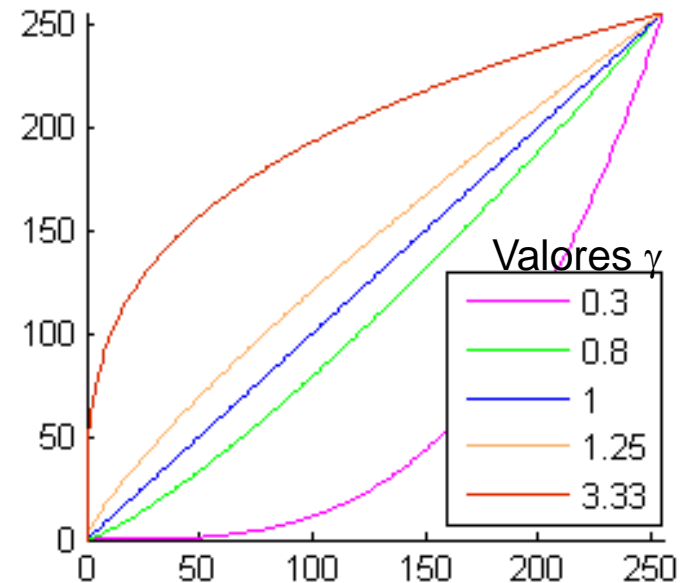
- $dest(i,j) = c2 * (c1 + src(i,j))$
- Si $C1 > 0$ el BRILLO se INCREMENTA
- Si $C1 < 0$ el BRILLO se DECREMENTA
- Si $|C2| > 1$ el CONTRASTE se INCREMENTA
- Si $|C2| < 1$ el CONTRASTE se DECREMENTA
- Mejora de contraste: $c2 = 255 / (\max_{src} - \min_{src})$ $c1 = -\min_{src}$
- OJO: salidas de rango

Mejora de contraste y elim. de fondo

❑ Funciones no lineales:

- Ajuste gamma:

$$\text{dest}(i, j) = 255 \cdot \left[\frac{\text{src}(i, j)}{255} \right]^{1/\gamma}, \quad \gamma > 0$$



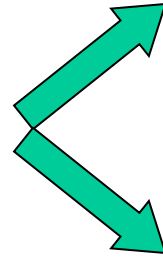
❑ Aplicación de LUTs:

- Para evitar cálculos largos o aplicar funciones más complejas:
 - Se hace una tabla con los valores la función para los valores de entrada posibles (0 – 255).
 - Se busca en la tabla según el índice.

Mejora de contraste y elim. de fondo

□ Ejemplos:

Imagen Original



Aumento de contraste



Aumento de brillo



Mejora de contraste y elim. de fondo

- Ecuación de histograma:
 - Permite calcular una LUT de forma que la imagen destino utilice todo el rango de valores de luminosidad.
- Histograma de una imagen:
 - Tabla con tantos datos como niveles de luminosidad (ej. 0 a 255 = 256 datos).
 - Cada elemento de la tabla contiene la cuenta del nº de pixels de la imagen con esa luminosidad.
 - La gráfica del histograma informa sobre la utilización en la imagen de los diferentes niveles de luminosidad.
 - La ecualización del histograma permite optimizar la utilización de los niveles de luminosidad

Mejora de contraste y elim. de fondo

□ Ejemplos:

Imagen Original

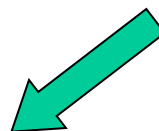
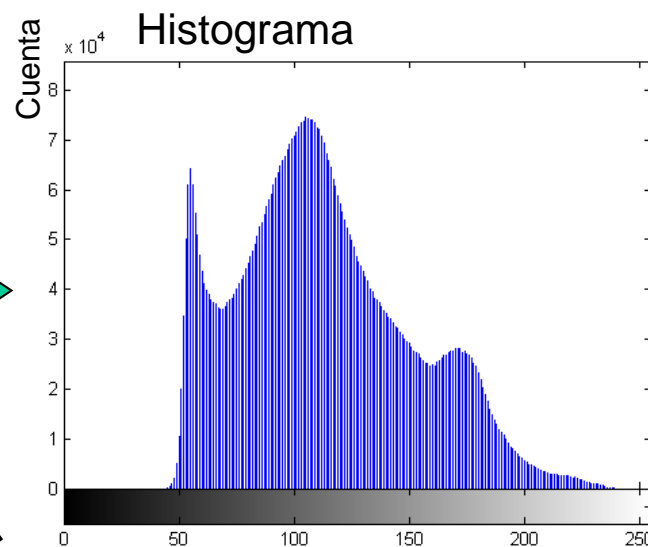
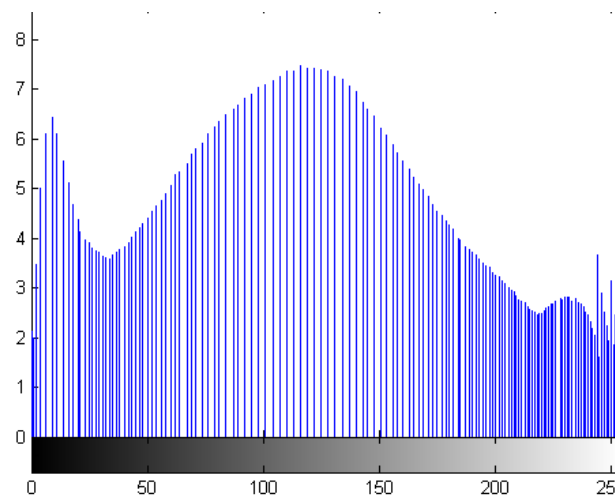


Imagen Ecualizada



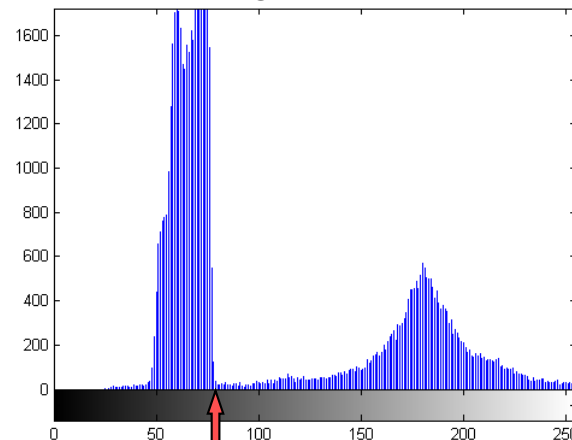
Mejora de contraste y elim. de fondo

- Uso de histograma. Eliminación del fondo:
 - En el histograma se suelen apreciar dos zonas, una correspondientes al (los) objeto(s), y otra(s) al fondo.
 - Se puede utilizar esta información para eliminar el efecto del fondo.

Imagen original



Histograma



Corte objetos/fondo

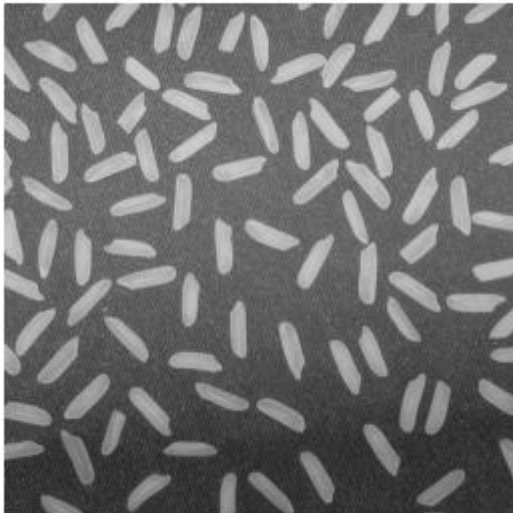
Original sin pixels < 78



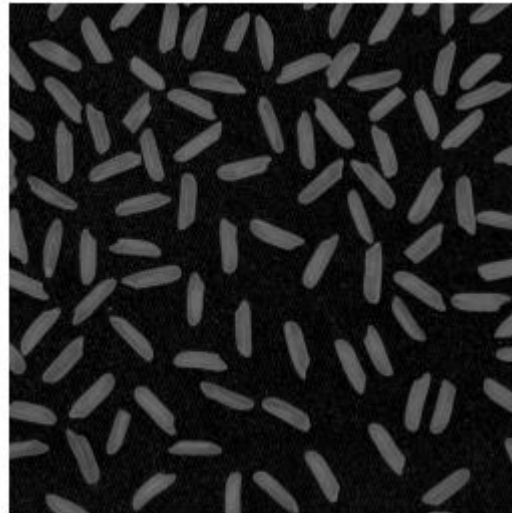
Mejora de contraste y elim. de fondo

- ❑ Otras técnicas para eliminación de fondo:
 - Filtro tophat (objeto claro, fondo oscuro):
 - Se le resta a cada pixel el mínimo de una región que lo incluye.
 - Filtro bottomhat (objeto oscuro, fondo claro).

Imagen original



Filtro top-hat



Filtro + ajuste contraste



Mejora de contraste y elim. de fondo

- ❑ Operaciones con múltiples imágenes:
 - Suma o promedio de imágenes iguales: mejora el rango dinámico.
 - Resta de imágenes similares: resalta cambios de una imagen a otra.

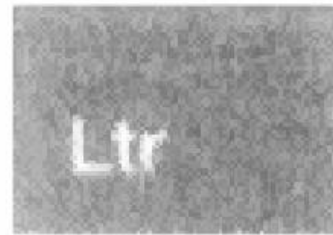
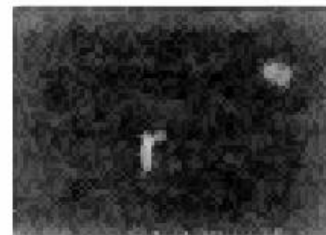


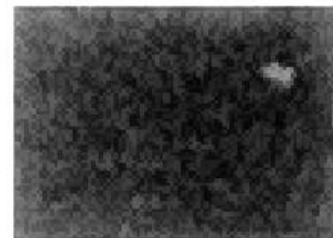
Imagen de referencia



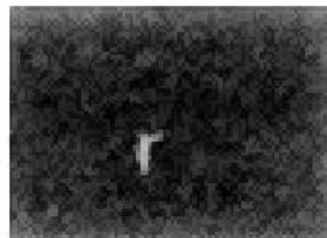
Pieza errónea



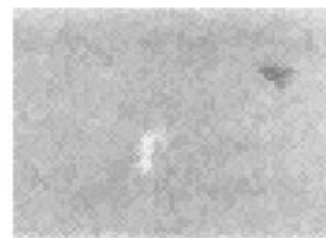
Diferencias Absolutas



Selección de positivos



Selección de Negativos



Escalado Lineal



Indice

- ❑ Estructura del curso
- ❑ Preprocesamiento de imágenes
- ❑ Mejora de contraste y elim. de fondo
- ❑ **Filtrado de ruidos**
- ❑ Transformaciones geométricas
- ❑ Ejemplos con Matlab

Filtrado de ruidos

- ❑ Las imágenes suelen contener ruido superpuesto a la información deseada.
- ❑ El procesamiento de una imagen con ruido suele dar lugar a una detección incorrecta.
- ❑ Tipos de ruido principales:
 - Gaussiano (normalmente ruido de amplificación)
 - Salt & pepper (aparición de outliers)
- ❑ Filtrado de ruido:
 - Requiere siempre operaciones de vecindad

Filtrado de ruidos

Operaciones de vecindad:

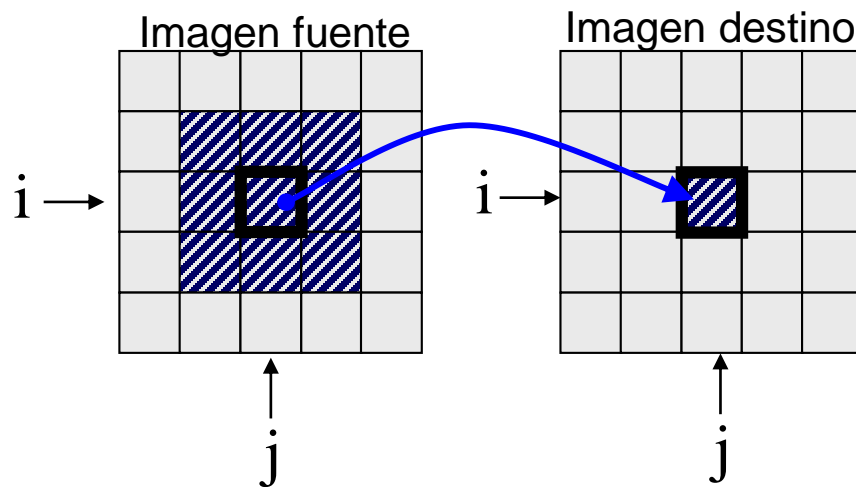
- $dest(i,j)=Fn(src(i-n_k : i+n_k, j-m_k : j+m_k))$

Posibilidades:

- $F_n()$ lineal (convolución)
- $F_n()$ no lineal

Problemática:

- ¿Qué hacer para los pixels que no tienen suficientes vecinos?

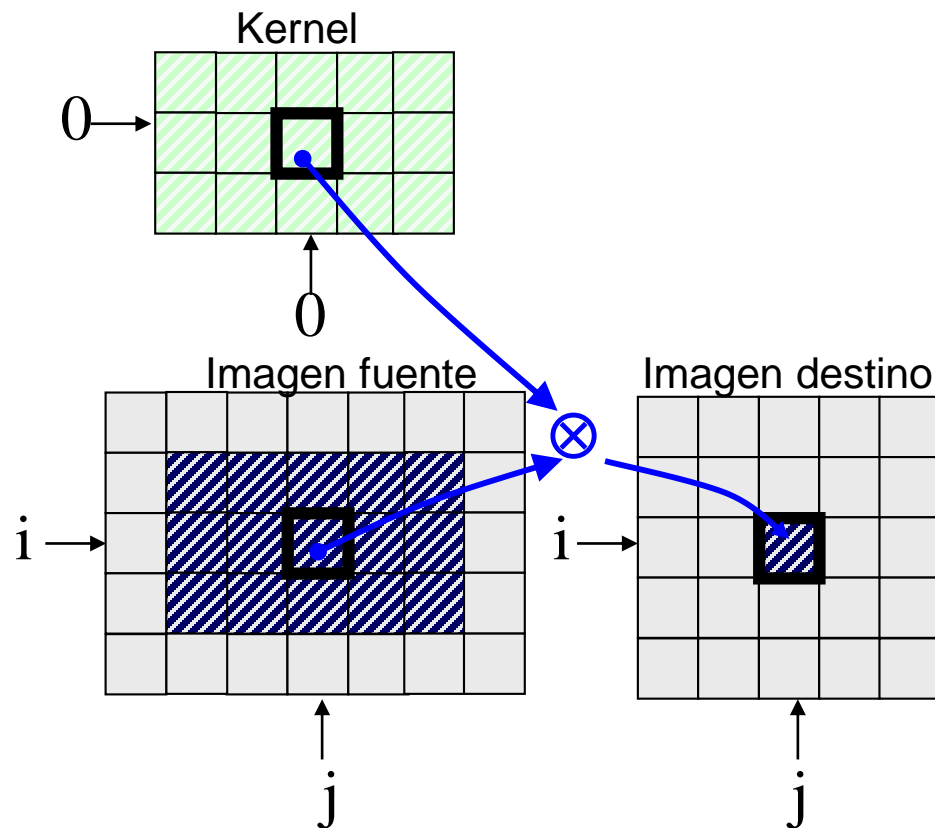


Filtrado de ruidos

□ Filtros lineales (convolución):

$$\text{dest}(i, j) = \sum_{n=-n_k}^{n=n_k} \sum_{m=-m_k}^{m=m_k} \text{kernel}(n, m) \cdot \text{src}(i + n, j + m)$$

- Kernel o máscara: matriz de tamaño $(2n_k+1) \times (2m_k+1)$
- Por simplicidad de notación, consideramos que el índice 0 del kernel corresponde con el pixel a tratar.
- La operación se conoce como convolución o filtrado:
 - $\text{dest} = \text{src} \otimes \text{kernel}$
 - Convolución y filtrado sólo se distinguen por la organización del kernel

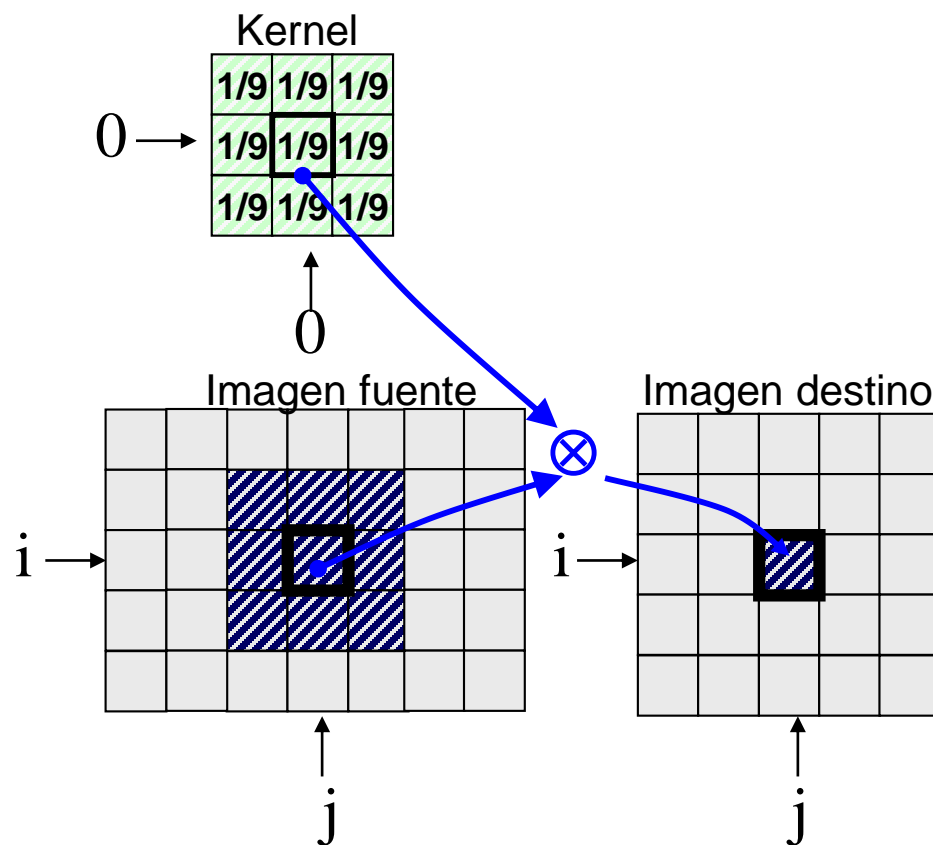


Filtrado de ruidos

- Ejemplo de filtro lineal: promediado de 3x3

$$\text{kernel} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Cada pixel de la imagen destino es la media de los que rodean al pixel fuente → efecto de suavizado



Filtrado de ruidos

□ Kernels para suavizado de imagen:

▪ Filtros de media:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

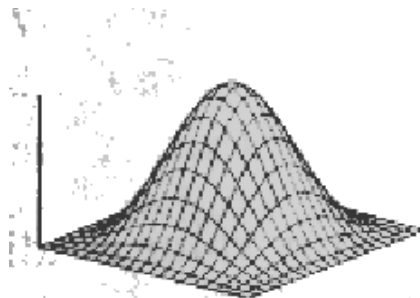
$$\frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

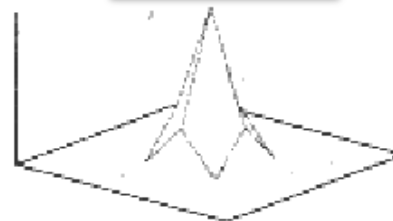
▪ Filtros de gaussiana:

$$\frac{1}{28} \begin{pmatrix} 1 & 4 & 6 \\ 4 & 12 & 4 \\ 6 & 4 & 1 \end{pmatrix}$$

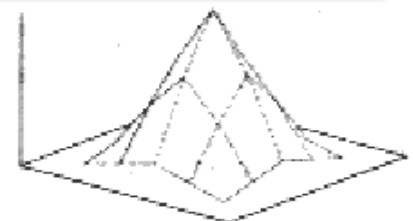
$$\frac{1}{121} \begin{pmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 7 & 11 & 7 & 2 \\ 3 & 11 & 17 & 11 & 3 \\ 2 & 7 & 11 & 7 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{pmatrix}$$



Gaussian operator



3 × 3 filter kernel



5 × 5 filter kernel

Filtrado de ruidos

□ Kernels para suavizado de imagen:

- Se produce efecto de suavizado siempre que:
 - Todos los pesos del kernel sean positivos o nulos
Y
 - La suma de todos los pesos sea igual a 1
- En estos casos, el kernel se comporta como un filtro paso-bajo:
 - Elimina las altas frecuencias (variaciones bruscas y ruido).
 - Deja pasar las bajas frecuencias (variaciones suaves).
- ¡ OJO ! Esto afecta tanto al ruido (indeseado) como a otras características deseadas (ej. los bordes).

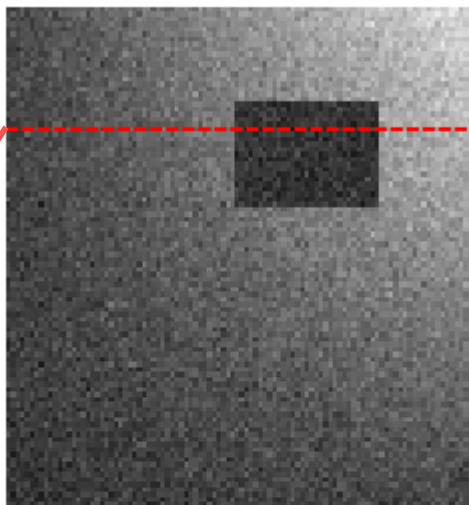
Filtrado de ruidos

- Usos de los kernels de suavizado:
 - Evitar el efecto de pequeños ruidos aleatorios.
 - Eliminar pequeños detalles para resaltar los más grandes
- Un tamaño de ventana mayor:
 - Suaviza más
 - Elimina rasgos de mayor tamaño
 - Tiene una frecuencia de corte más baja

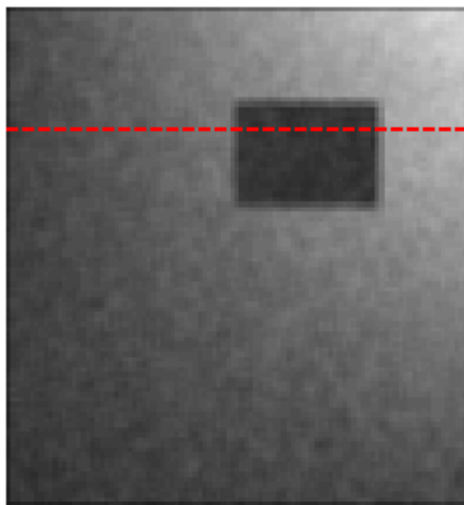
Filtrado de ruidos

- Ejemplos de suavizado de ruido:

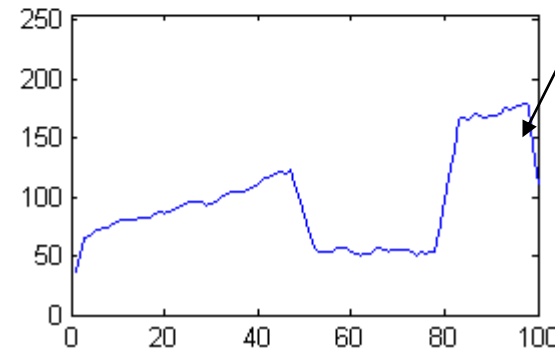
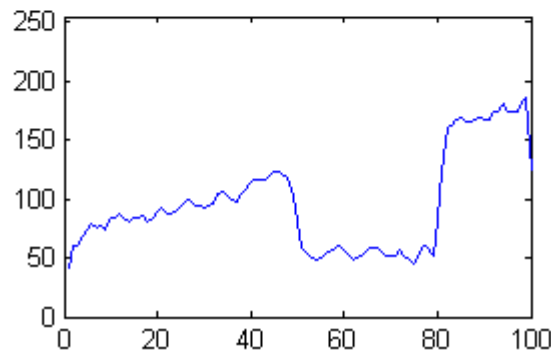
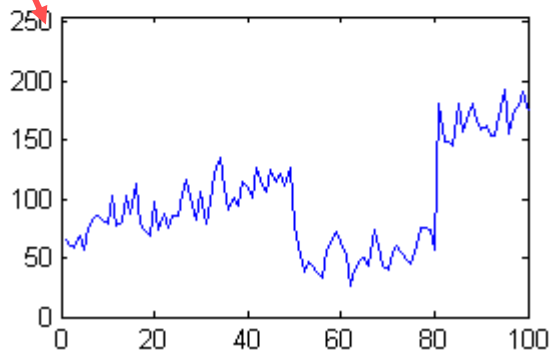
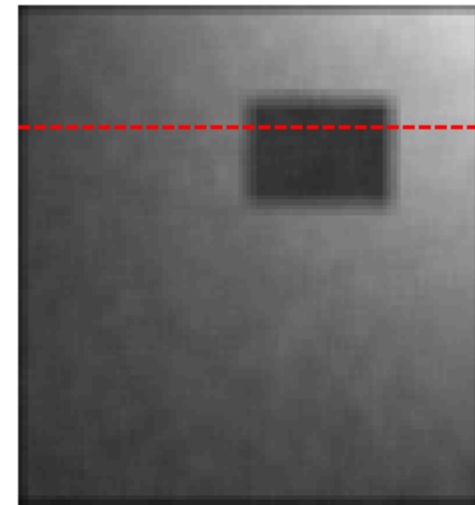
Imagen ruidosa



Filtro media 3x3



Filtro media 5x5



efecto borde

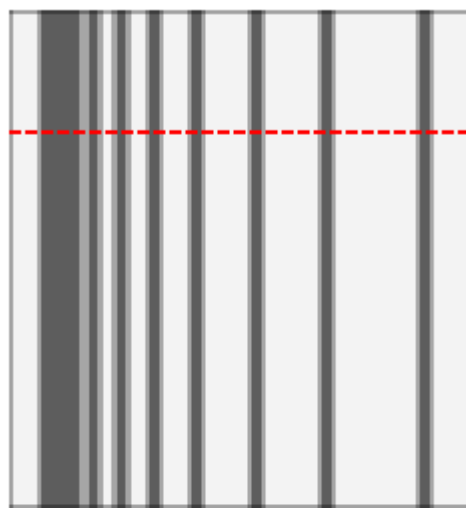
Filtrado de ruidos

- Ejemplos de suavizado de detalles:

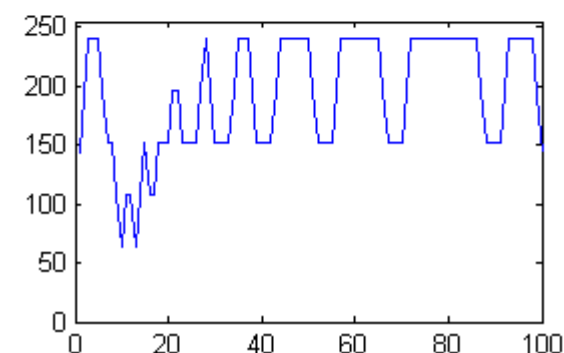
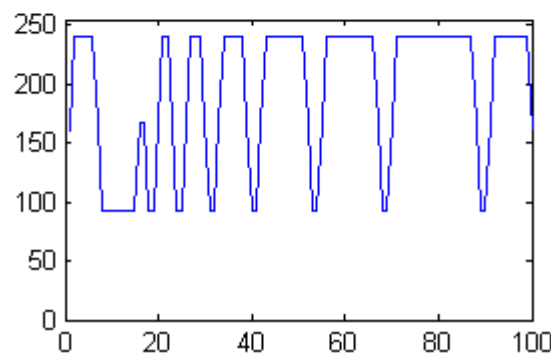
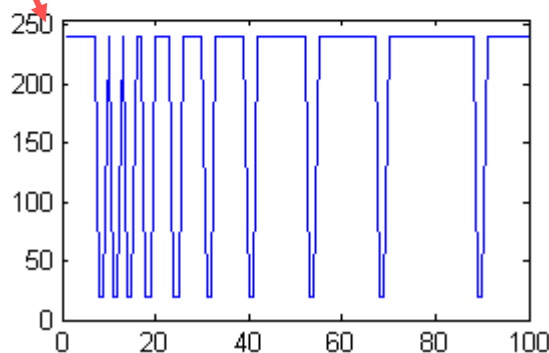
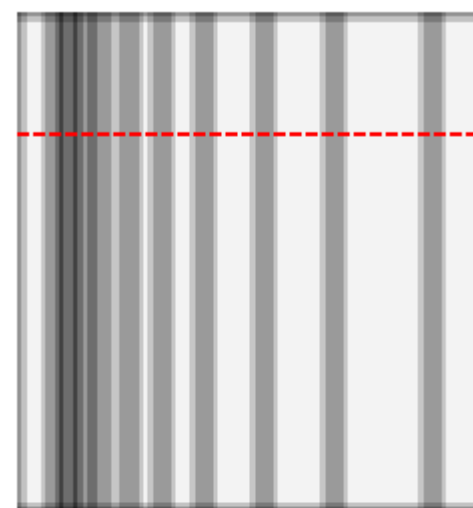
Imagen original



Suavizado media 3x3



Suavizado media 5x5



Filtrado de ruidos

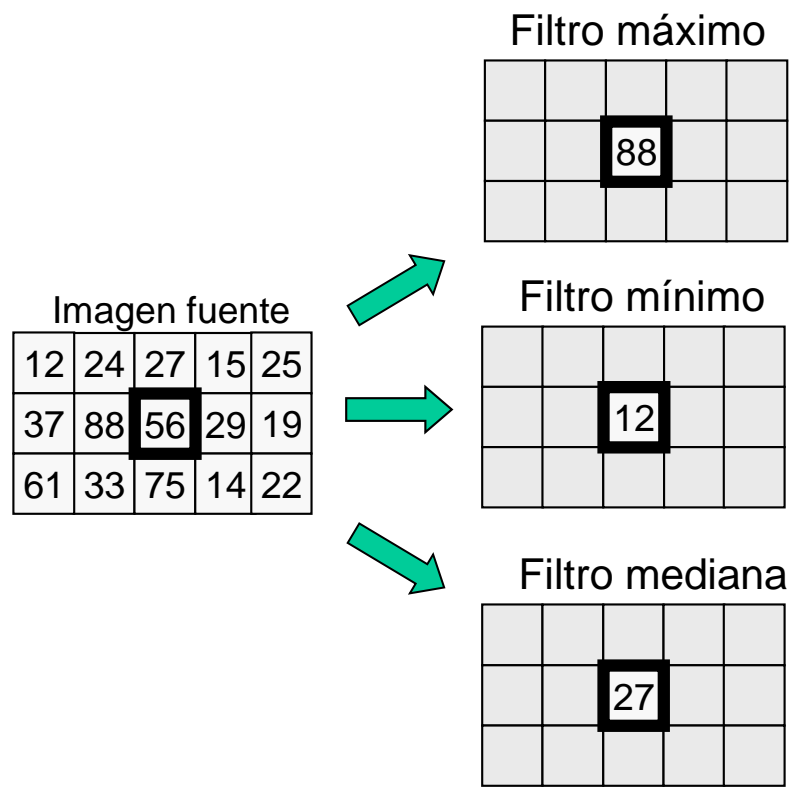
❑ Filtros no lineales:

$$\text{dest}(i, j) = F_n \left(\text{src}(i + n, j + m) \right)$$

$$\begin{matrix} n = -n_k \text{ a } n_k \\ m = -m_k \text{ a } m_k \end{matrix}$$

❑ Filtros no lineales típicos:

- Máximo (o dilatación)
- Mínimo (o erosión)
- Mediana (valor intermedio del entorno)



Filtrado de ruidos

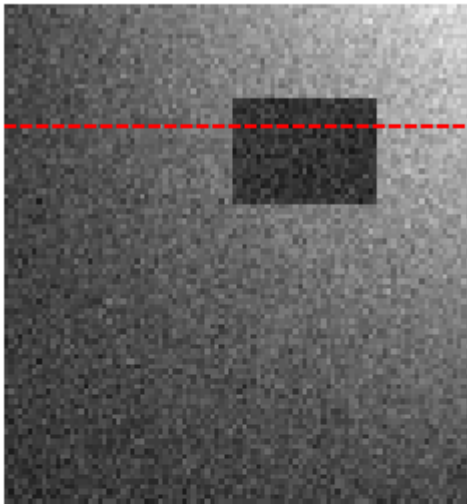
- Usos típicos de los filtros no lineales:
 - Dilatación:
 - Obtener una imagen de fondo (objetos oscuros, fondo claro).
 - Erosión:
 - Obtener una imagen de fondo (objetos claros, fondo oscuro).
 - Mediana:
 - Filtrado con menor pérdida de detalles (bordes).
 - Filtrado de ruido salt & pepper.

- OJO: los filtros no lineales requieren operaciones tipo if, mucho más lentas de ejecución.

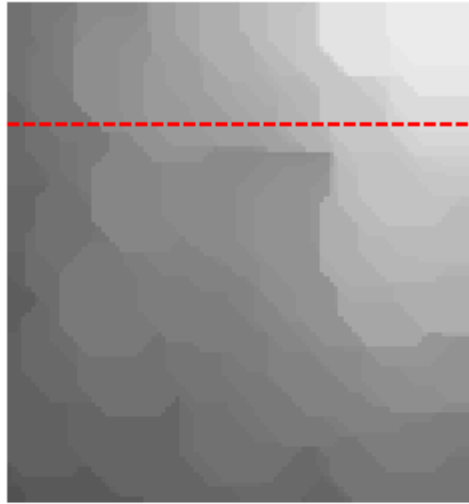
Filtrado de ruidos

- Ejemplos de usos de los filtros no lineales:

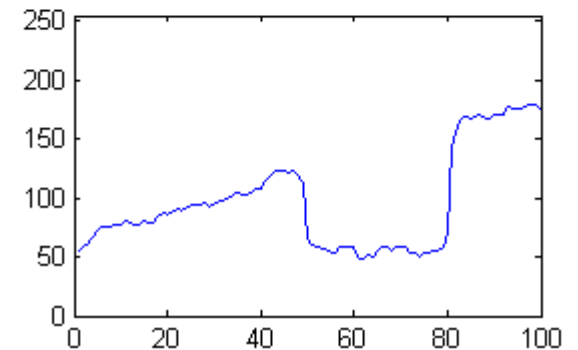
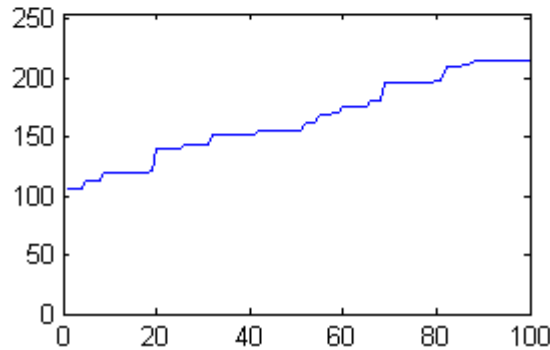
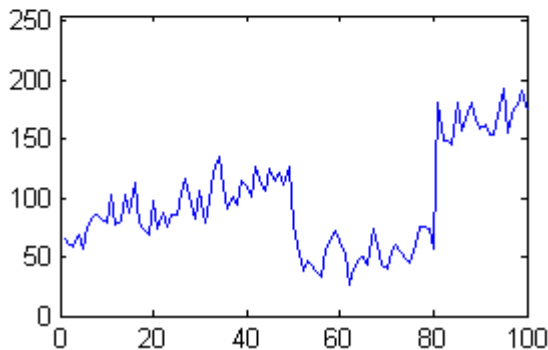
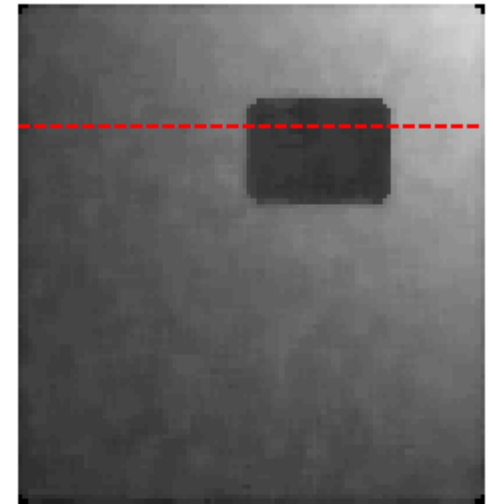
Imagen original



Dilatación disco r=12



Filtro mediana 5x5



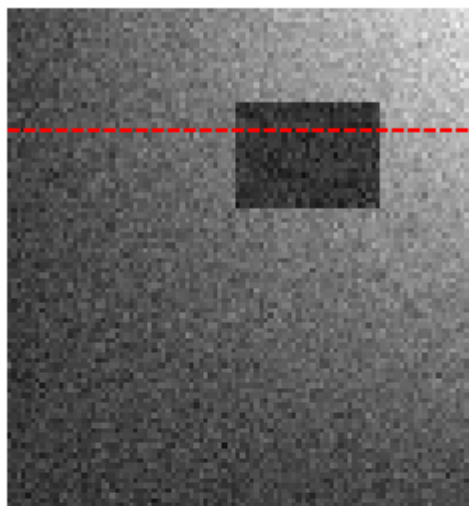
Filtrado de ruidos

- ❑ Problemática de todos los métodos de vecindad: el efecto de borde (boundary).
 - ¿Qué hacer para los pixels que no tienen suficientes vecinos? (efecto de borde)
- ❑ Soluciones:
 - Recortar la imagen destino, eliminando los pixels que sufren el efecto borde.
 - Ampliar la imagen fuente, utilizando un valor determinado para el borde ampliado:
 - El mismo para toda la imagen.
 - El del pixel más cercano (mejor).
 - La imagen simétrica de la zona cercana (mejor).

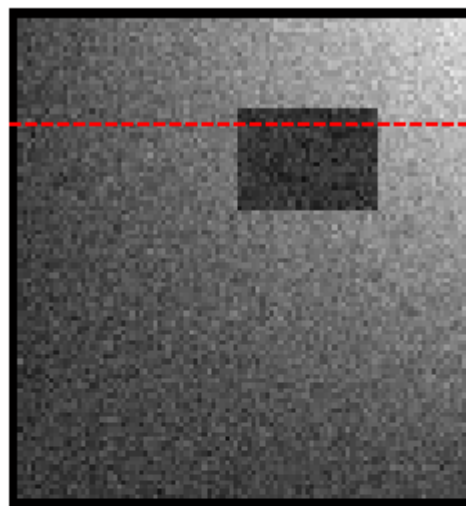
Filtrado de ruidos

□ Ejemplos de selección de borde ampliado:

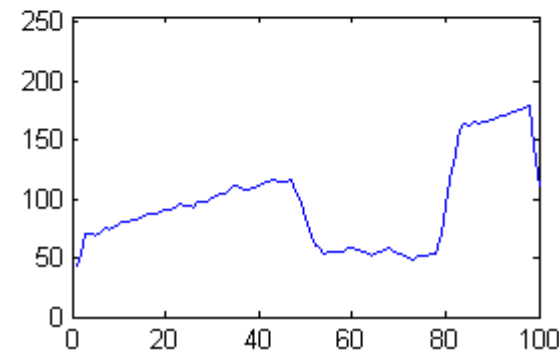
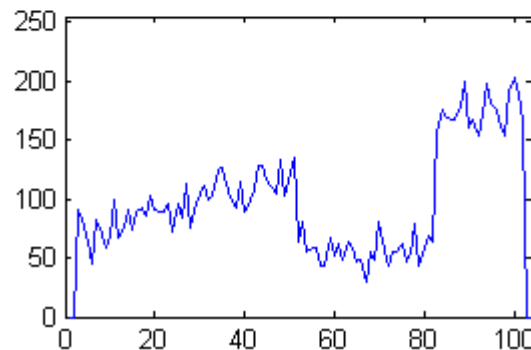
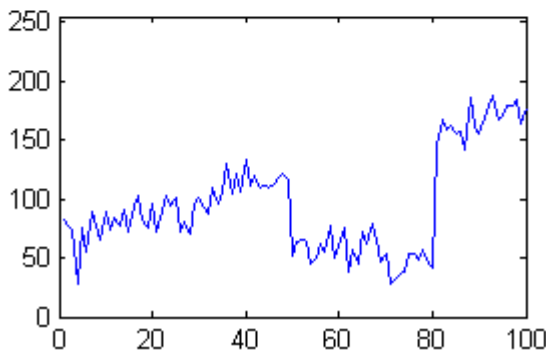
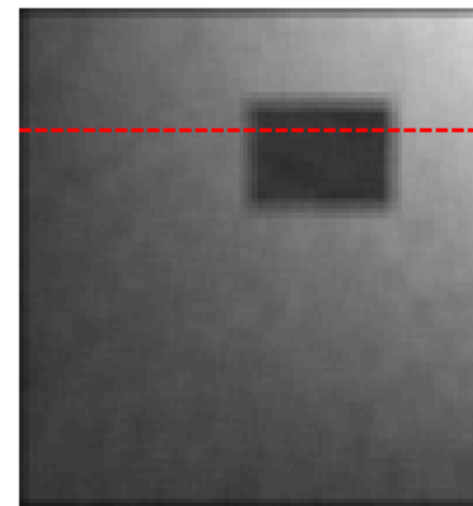
Imagen original



Borde ampliado cte



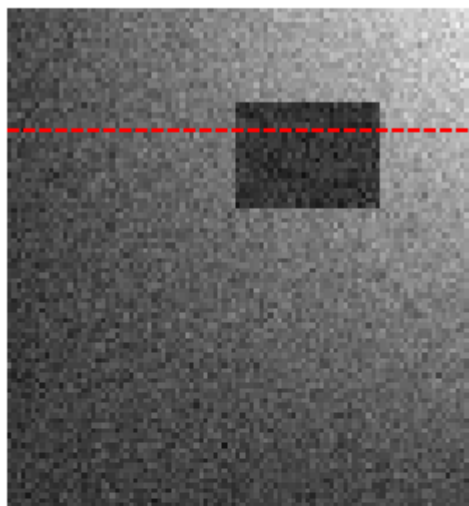
Filtrado media 5x5



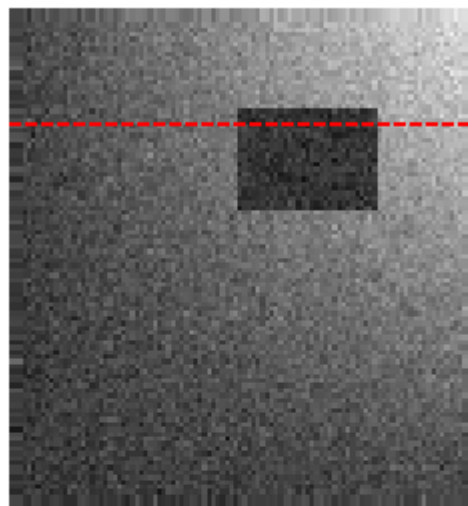
Filtrado de ruidos

□ Ejemplos de selección de borde ampliado:

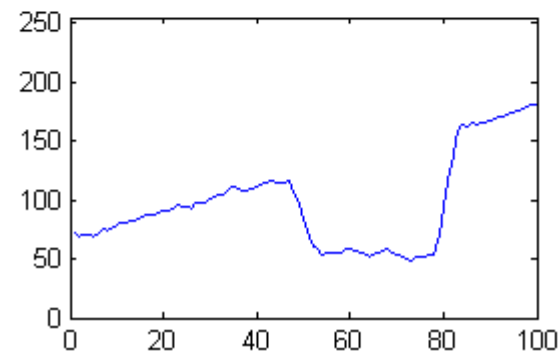
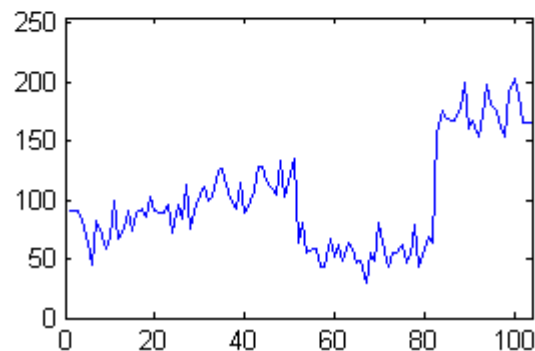
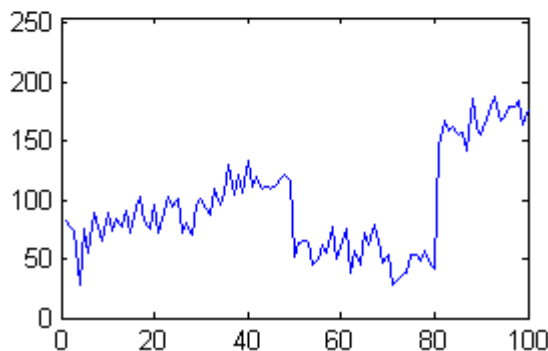
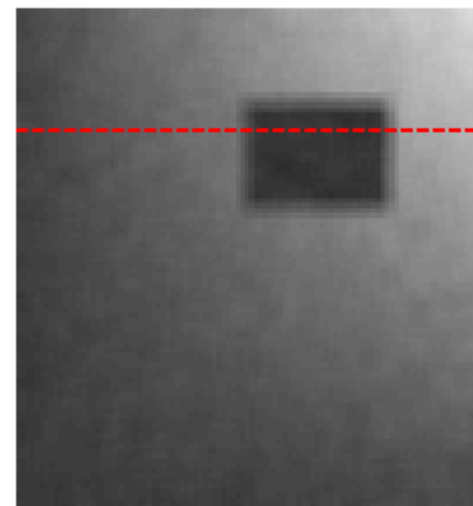
Imagen original



Borde ampliado=replicate



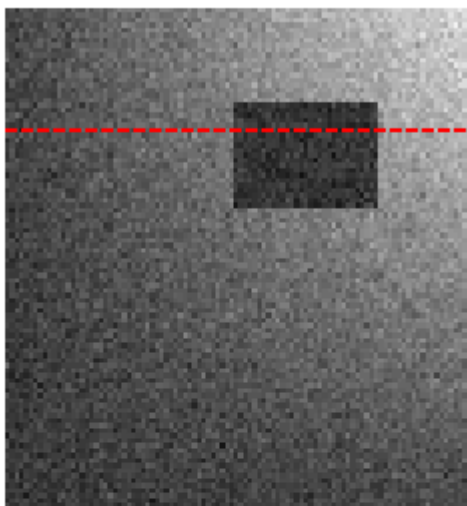
Filtrado media 5x5



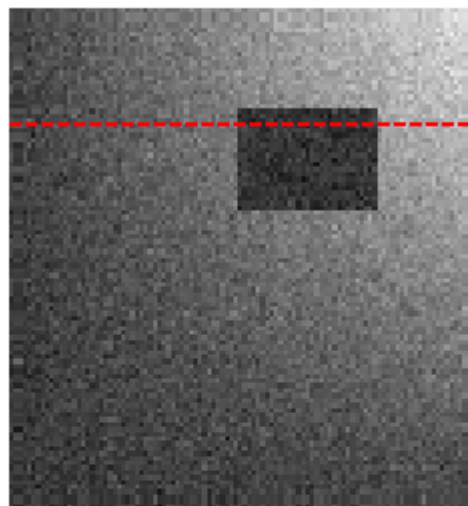
Filtrado de ruidos

□ Ejemplos de selección de borde ampliado:

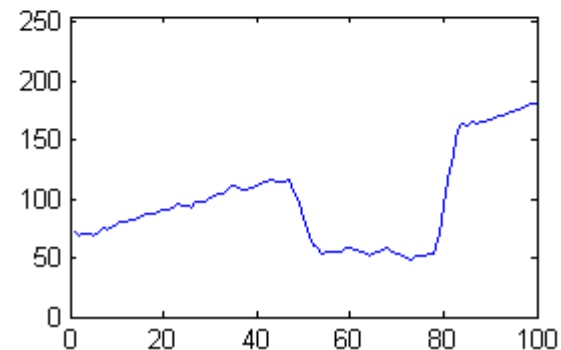
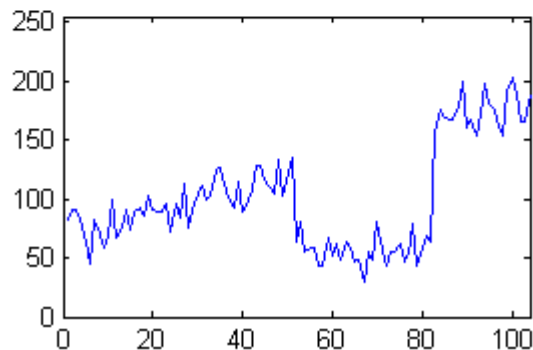
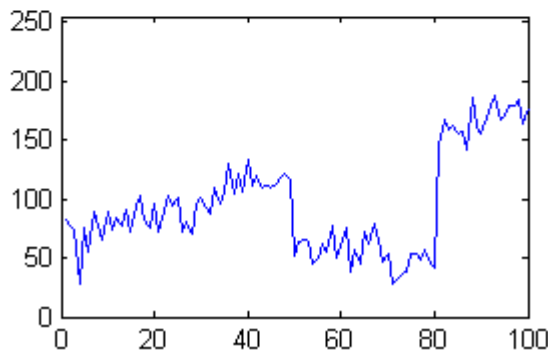
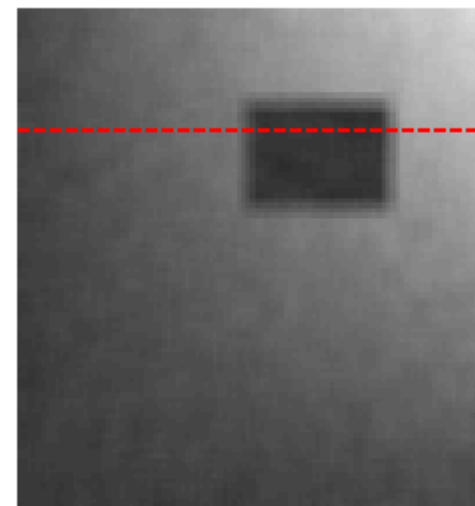
Imagen original



Borde ampliado=symmetric



Filtrado media 5x5



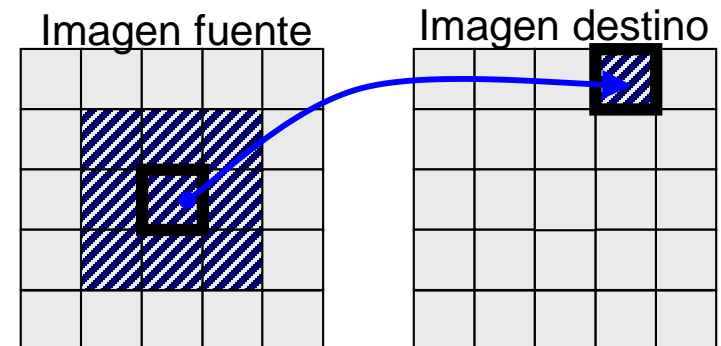


Indice

- ❑ Estructura del curso
- ❑ Preprocesamiento de imágenes
- ❑ Mejora de contraste y elim. de fondo
- ❑ Filtrado de ruidos
- ❑ **Transformaciones geométricas**
- ❑ Ejemplos con Matlab

Transformaciones geométricas:

- Transformaciones geométricas:
 - Cambian la posición de los pixels.
 - Suelen utilizarse para eliminar distorsiones.
 - Precisan interpolación.
- Transformaciones geométricas típicas:
 - Zoom
 - Cambio de perspectiva
 - Corrección de distorsiones



Transformaciones geométricas:

- Transformaciones geométricas típicas:
 - Afín (affine): escala, traslación, rotación.
 - Proyectiva (projective): id + perspectiva.

- Funciones de interpolación típicas:
 - Punto más cercano (nearest)
 - Bi-lineal (bilinear)
 - Bi-cúbica (bicubic)

Transformaciones geométricas:

- Transformación afín:

$$\mathbf{M}_a = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} i_{dest} \\ j_{dest} \\ 0 \end{bmatrix} = \mathbf{M}_a \cdot \begin{bmatrix} i_{src} \\ j_{src} \\ 0 \end{bmatrix}$$

- Ejemplo: zoom 1/2

$$\mathbf{M}_a = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



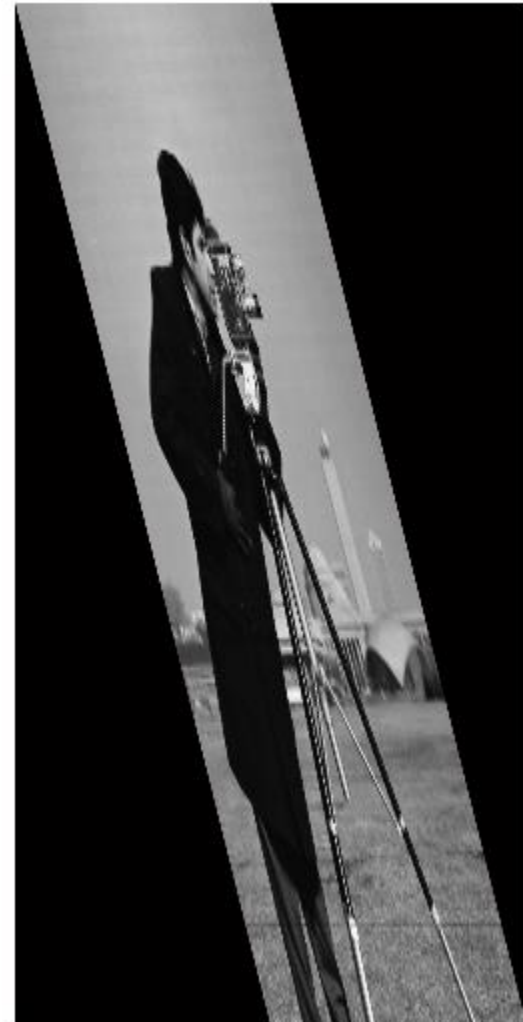
Transformaciones geométricas:

- Transformación afín:

$$M_a = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{31} & 1 \end{bmatrix} \quad \begin{bmatrix} i_{dest} \\ j_{dest} \\ 1 \end{bmatrix} = M_a \cdot \begin{bmatrix} i_{src} \\ j_{src} \\ 1 \end{bmatrix}$$

- Ejemplo: zoom y shear

$$M_a = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Transformaciones geométricas:

- Transformación afín:

$$M_a = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{31} & 1 \end{bmatrix} \quad \begin{bmatrix} i_{dest} \\ j_{dest} \\ 1 \end{bmatrix} = M_a \cdot \begin{bmatrix} i_{src} \\ j_{src} \\ 1 \end{bmatrix}$$

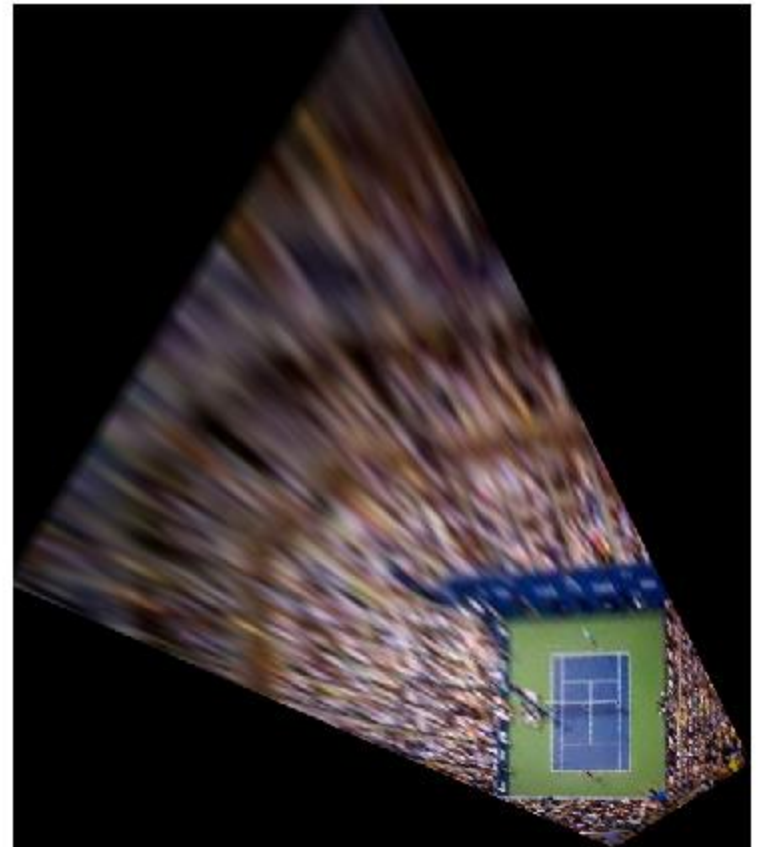
- Ejemplo: rotación 45°

$$M_a = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Transformaciones geométricas:

- ❑ Transformación proyectiva:
 - Transformación más compleja (rotación, traslación, escala y perspectiva). Ver ayuda Matlab para funciones de generación de la matriz.
- ❑ Ejemplo: transformación de perspectiva



Transformaciones geométricas:

- Interpolación: ejemplo zoom x4

Nearest



Bilinear



Bicubic





Indice

- ❑ Estructura del curso
- ❑ Preprocesamiento de imágenes
- ❑ Mejora de contraste y elim. de fondo
- ❑ Filtrado de ruidos
- ❑ Transformaciones geométricas
- ❑ **Ejemplos con Matlab**

Ejemplos con Matlab

□ Funciones de Matlab para transformaciones puntuales:

■ Carga de imagen y conversión a escala de gris:

```
>> img=imread(' ../imagenes/176.jpg');
```

```
>> whos img
```

Name	Size	Bytes	Class	Attributes
img	245x246x3	180810	uint8	

```
>> imgs=rgb2gray(img);
```

```
>> whos imgs
```

Name	Size	Bytes	Class	Attributes
imgs	245x246	60270	uint8	

■ Aritmética sencilla:

```
>> imgd=5*imgs-40;
```

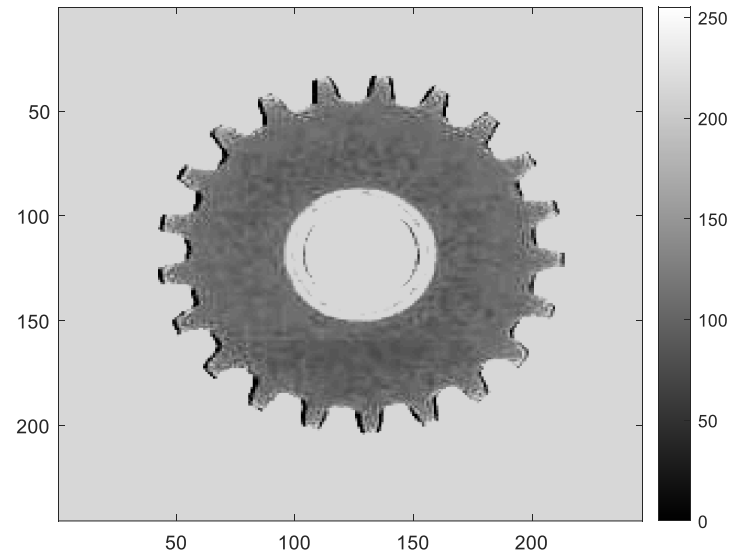
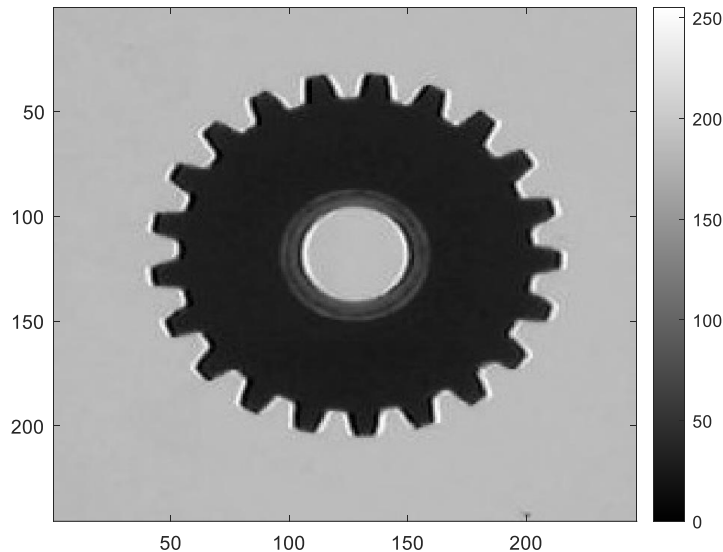
```
0 bien
```

```
>> imgd=imadd(imultiply(imgs,5),-40);
```

Ejemplos con Matlab

□ Funciones de Matlab para transformaciones puntuales:

```
>> figure; imagesc(imgs); caxis([0 255]); colormap gray; colorbar;  
>> figure; imagesc(imgd); caxis([0 255]); colormap gray; colorbar
```



Ejemplos con Matlab

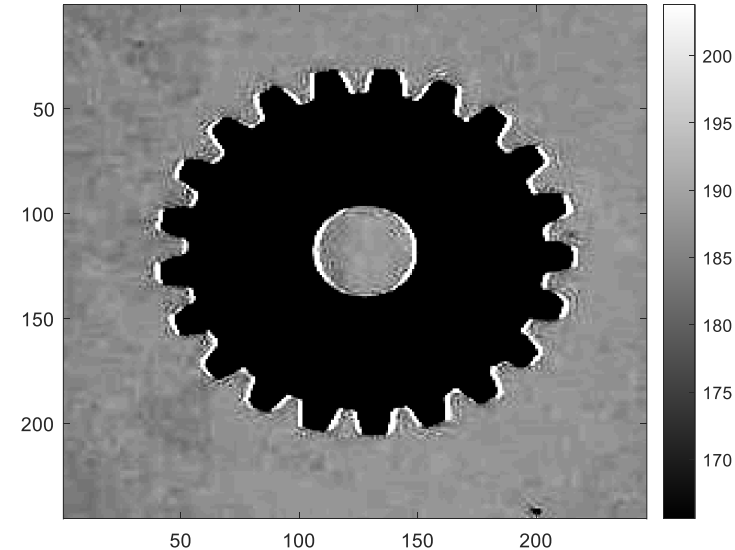
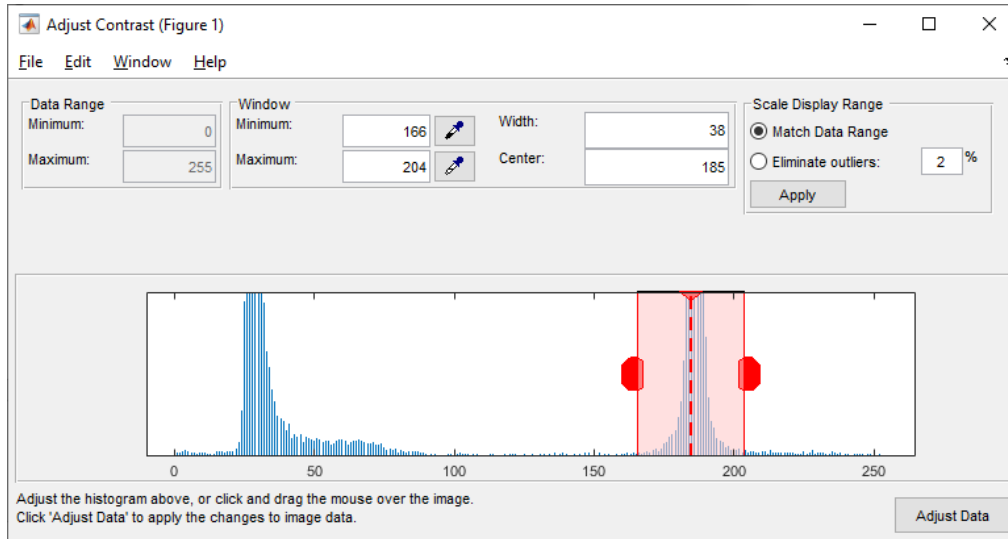
❑ Funciones de Matlab para transformaciones puntuales:

- Mejora de contraste: `imadjust()`, `imcontrast()`

```
>> imga=imadjust(imgs);
```

```
>> figure; imagesc(imga); caxis([0 255]); colormap gray; colorbar;
(Seleccionar ventana que contiene imgs)
```

```
>> imcontrast
```

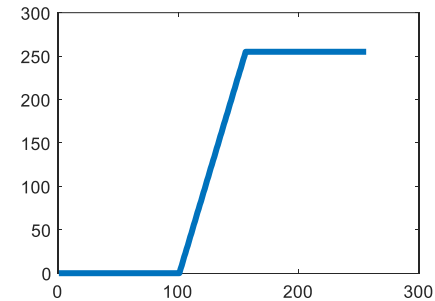


Ejemplos con Matlab

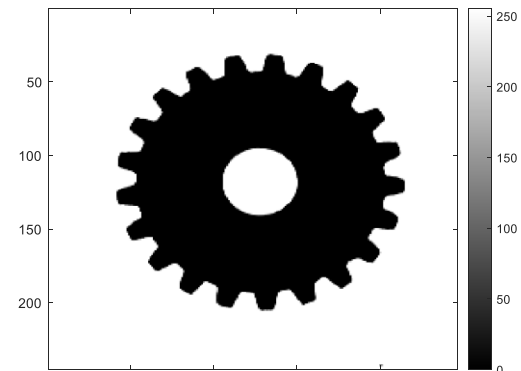
□ Funciones de Matlab para transformaciones puntuales:

■ Manejo de tablas look-up: **intlut()**

```
>> lut=uint8([zeros(1,100),linspace(0,255,56),255+zeros(1,100)]);  
>> figure; plot(lut,'linewidth',3);  
>> img1=intlut(imgs,lut);
```



```
>> figure; imagesc(img1); caxis([0 255]); colormap gray; colorbar;
```



Ejemplos con Matlab

□ Funciones de Matlab para transformaciones puntuales:

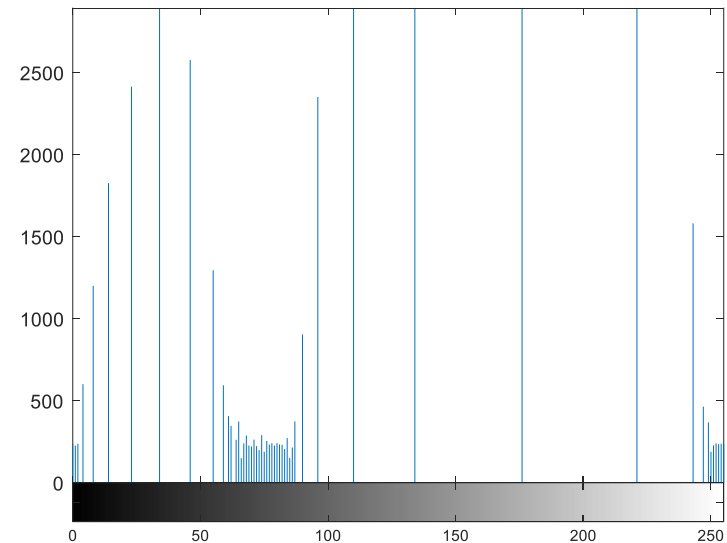
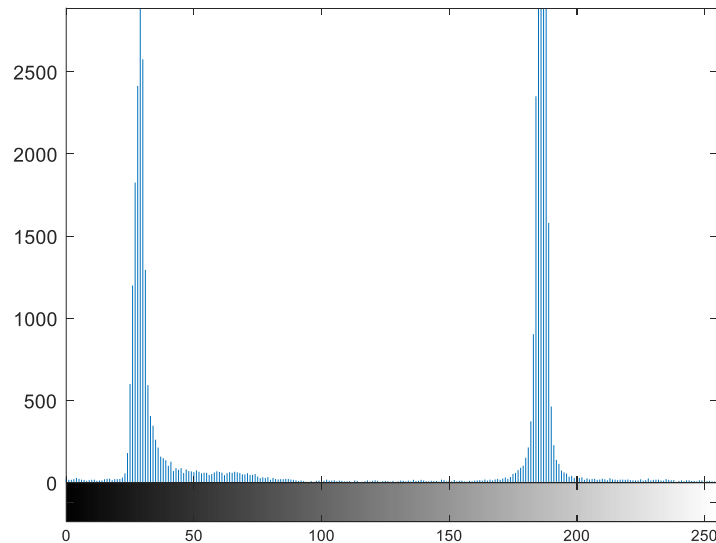
■ Manejo de histogramas: **imhist()**, **histeq()**

```
>> figure; imhist(imgs,256);
```

```
>> imge=histeq(imgs,256);
```

```
>> figure; imagesc(imge); caxis([0 255]); colormap gray; colorbar;
```

```
>> figure; imhist(imge,256);
```



Ejemplos con Matlab

- Funciones de Matlab para transformaciones de vecindad:
 - Transformaciones lineales: **imfilter()**
 - Filtro de media 3 x 3:

```
>> imgf=imfilter(imgs,1/9*ones(3,3));
```
 - Se pueden seleccionar como parámetros:
 - Acción en el borde (valor, 'symmetric','replicate','circular')
 - Tamaño de la imagen de salida ('same','full')
 - Correlación o convolución ('corr','conv')
 - Creación de filtros: **fspecial()**
 - Filtro gaussiano 5x5, std=10:

```
>> hfilt=fspecial('gaussian',[5 5],10);  
>> imgf=imfilter(imgs,hfilt);
```
 - Adición de ruido para pruebas de filtros: **imnoise()**

```
>> imgn = imnoise(imgs,'salt & pepper', 0.02);
```

Ejemplos con Matlab

- Funciones de Matlab para transformaciones de vecindad:
 - Transformaciones no lineales:
 - Filtro de mediana: **medfilt2()**

```
>> imgd=medfilt2(imgs,[5 7]);
```
 - Transformaciones no lineales con elementos estructurales:
 - Creación de elemento estructural: **strel()**
 - Opciones: 'disk', 'square', 'diamond', 'line', 'rectangle', 'arbitrary'
 - Filtro de máximo: **imdilate()**

```
>> s=strel('disk',12);  
>> imgd=imdilate(imgs,s);
```
 - Filtro de mínimo: **imerode()**
 - Eliminación de fondo: **imtophat()**, **imbothat()**

Ejemplos con Matlab

- Funciones de Matlab para transformaciones geométricas:
 - **imrotate()**: realiza un giro
 - **imresize()**: realiza zoom
 - **imcrop()**: selección interactiva de zona

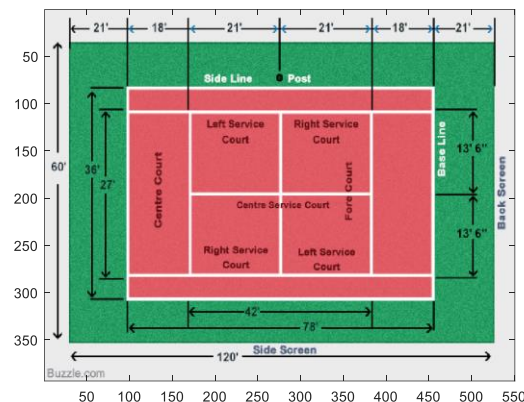
Ejemplos con Matlab

- Funciones de Matlab para transformaciones geométricas:
 - **maketform()**: crea matriz de transformación
 - **imtransform()**: realiza una transformación
 - **cpselect()**: selección interactiva de parejas de puntos de transformación
 - **cp2tform()**: convierte un conjunto de parejas de puntos de transformación en una matriz de transformación

Ejemplos con Matlab

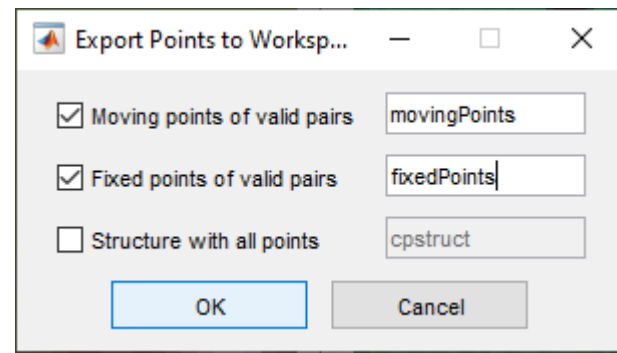
□ Ejemplo:

```
>> img1=imread(' ../imagenes/tenis.jpg');
>> img2=imclipboard('paste'); // Image of tennis court copied
>> figure; fig1=imagesc(img1); figure; fig2=imagesc(img2);
```



Selección interactiva de parejas de puntos correspondientes

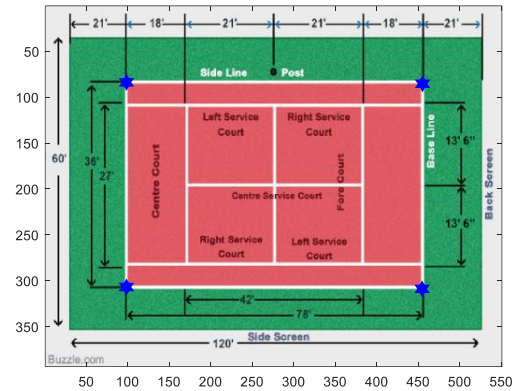
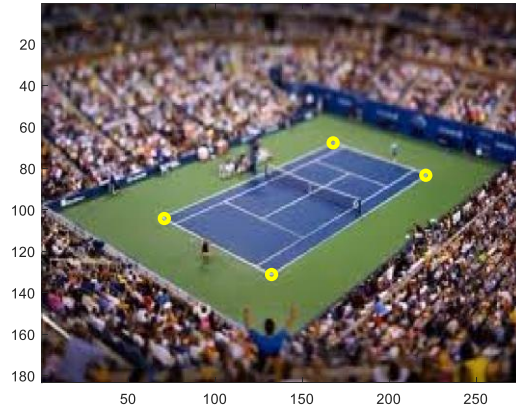
```
>> cpselect(img1,img2);
(terminar con File → Export points to workspace)
```



Ejemplos con Matlab

□ Ejemplo:

```
>> figure(fig1); hold on; plot(movingPoints(:,1),movingPoints(:,2),'oy')
>> figure(fig2); hold on; plot(fixedPoints(:,1),fixedPoints(:,2),'hb')
```



```
>> tform=cp2tform(movingPoints,fixedPoints,'projective');
>> figure; imagesc(imtransform(img1,tform));
```

