

# Interpolación de Imágenes

Escuela de Ingeniería Informática de Oviedo

# Contenidos

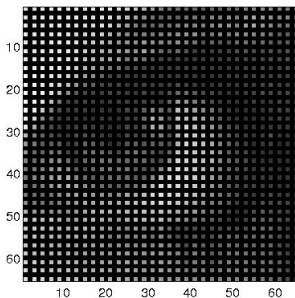
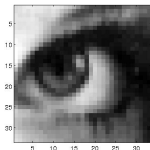
- 1 **Introducción**
- 2 **Vecino más cercano (Nearest neighbor)**
- 3 **Bilineal**
- 4 **Bicúbica**
- 5 **Matlab**

# Esquema

- 1 **Introducción**
- 2 Vecino más cercano (Nearest neighbor)
- 3 Bilineal
- 4 Bicúbica
- 5 Matlab

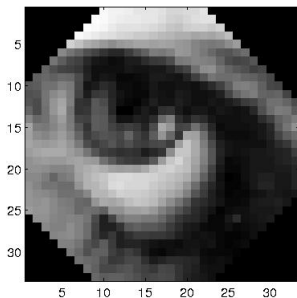
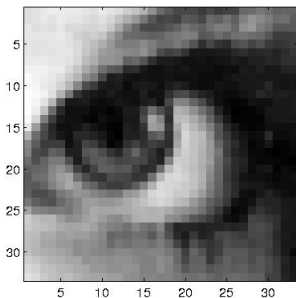
La interpolación de imágenes se utiliza siempre en el procesamiento de imágenes digitales

- Escalado (remuestreo).
- Reasignación (transformaciones geométricas - rotación, cambio de perspectiva,...).
- Relleno (restauración de *agujeros*).
- Deformación, transformaciones no lineales.



La interpolación de imágenes se utiliza siempre en el procesamiento de imágenes digitales

- Escalado (remuestreo).
- Reasignación (transformaciones geométricas - rotación, cambio de perspectiva,...).
- Relleno (restauración de *agujeros*).
- Deformación, transformaciones no lineales.



La interpolación de imágenes se utiliza siempre en el procesado de imágenes digitales

- Escalado (remuestreo).
- Reasignación (transformaciones geométricas - rotación, cambio de perspectiva,...).
- Relleno (restauración de *agujeros*).
- Deformación, transformaciones no lineales.



La interpolación de imágenes se utiliza siempre en el procesado de imágenes digitales

- Escalado (remuestreo).
- Reasignación (transformaciones geométricas - rotación, cambio de perspectiva,...).
- Relleno (restauración de *agujeros*).
- Deformación, transformaciones no lineales.



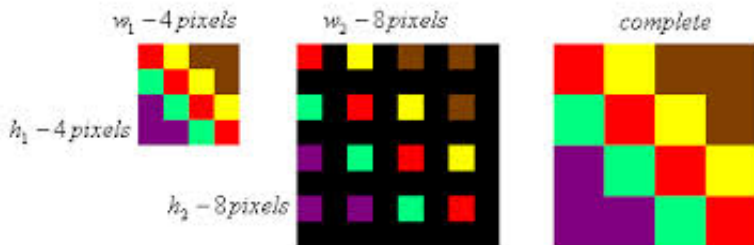
# Esquema

- 1 Introducción
- 2 Vecino más cercano (Nearest neighbor)**
- 3 Bilineal
- 4 Bicúbica
- 5 Matlab

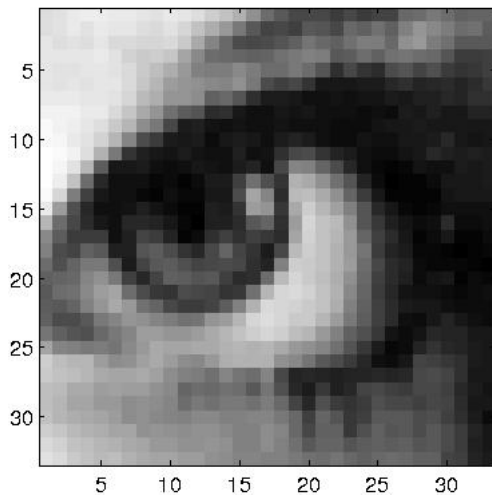


# Vecino más cercano (Nearest neighbor)

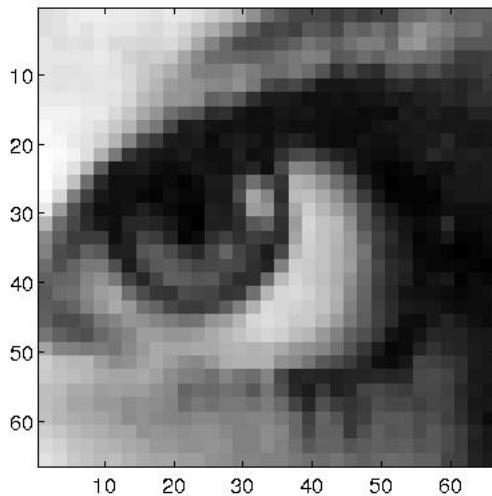
- Es un método básico.
- Requiere un tiempo de procesamiento bajo.
- Solo tiene en cuenta un píxel: el más cercano al punto interpolado.
- Simplemente aumenta el tamaño de cada píxel.



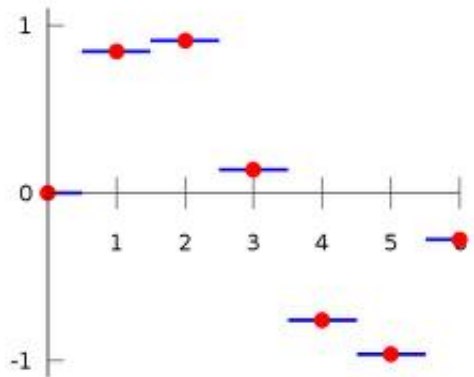
# Vecino más cercano (Nearest neighbor)



# Vecino más cercano (Nearest neighbor)



# Relaciones con la interpolación 1D

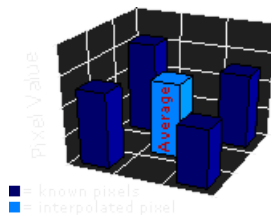


# Esquema

- 1 Introducción
- 2 Vecino más cercano (Nearest neighbor)
- 3 Bilineal**
- 4 Bicúbica
- 5 Matlab

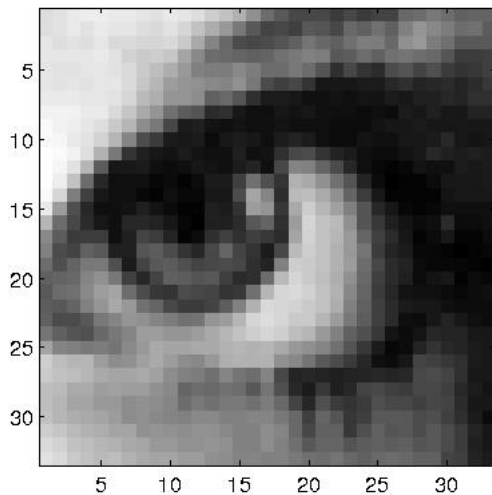
# Bilineal

- Tiene en cuenta los valores en los píxeles conocidos que rodean a uno dado en una vecindad de los 2x2 píxeles más cercanos.
- Se toma el promedio ponderado de estos 4 píxeles y se calcula el valor interpolado.
- El resultado está más suavizado que las imágenes obtenidas utilizando el método del píxel más cercano.
- Necesita más tiempo de procesamiento.

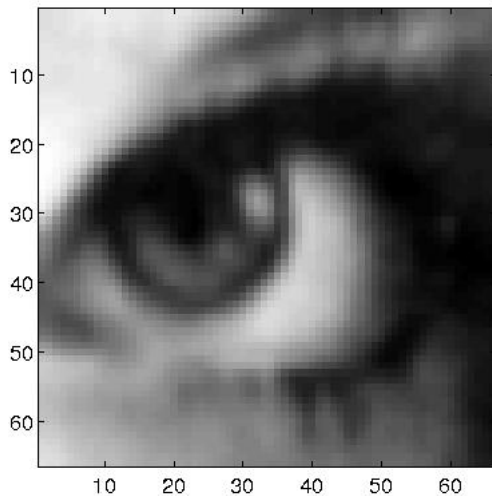


**Figura :** Ejemplo en el que todos los píxeles conocidos están a la misma distancia. El valor interpolado es entonces la suma de sus valores dividida por cuatro.

# Bilineal

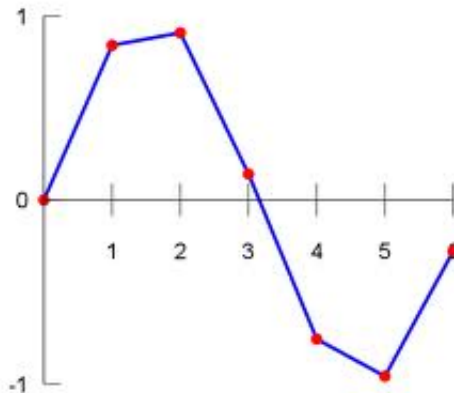


# Bilineal





# Relación con la interpolación 1D

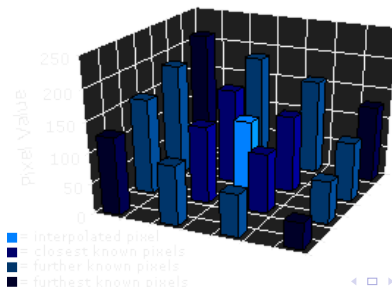


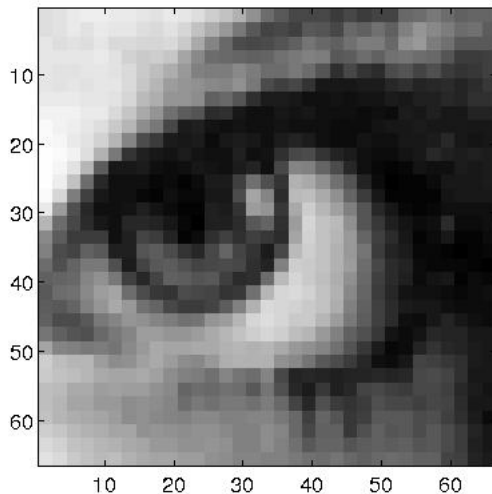
# Esquema

- 1 Introducción
- 2 Vecino más cercano (Nearest neighbor)
- 3 Bilineal
- 4 Bicúbica**
- 5 Matlab

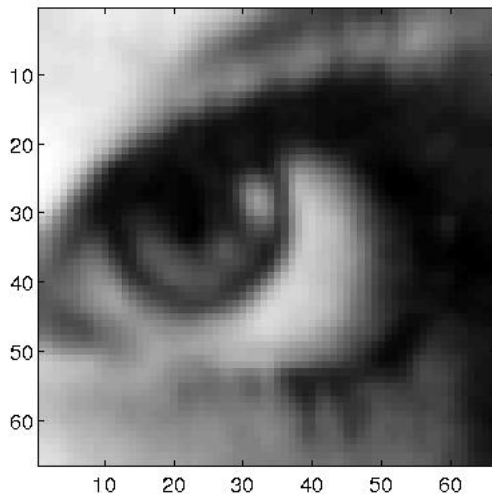
# Bicúbica

- Un paso más allá del caso bilineal es considerar la vecindad de los 4x4 píxeles conocidos más cercanos, es decir, un total de 16 píxeles.
- Como están situados a distancias distintas del píxel de valor desconocido, se da mayor peso en el cálculo a los más cercanos.
- Produce imágenes más nítidas que los dos métodos anteriores.
- Es un buen compromiso entre tiempo de procesamiento y calidad de resultado.
- Es un procedimiento estandar en programas de edición de imágenes, drivers de impresoras e interpolación en cámaras.

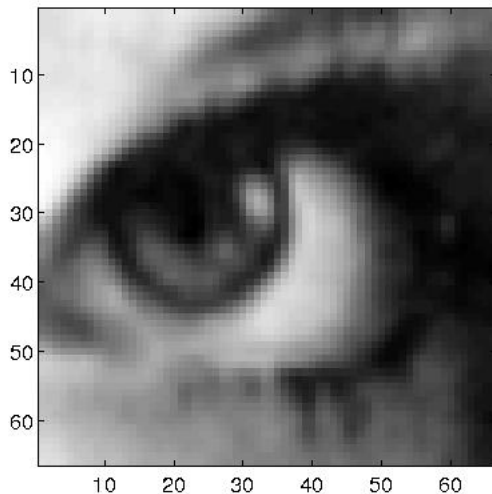




**Figura :** Nearest

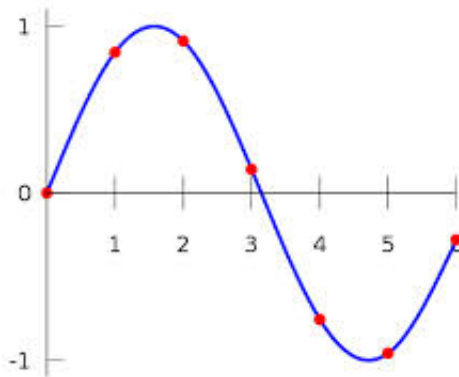


**Figura : Bilineal**

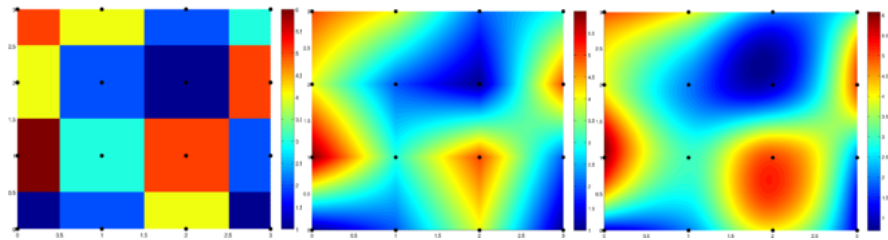


**Figura :** Bicúbica

# Relación con interpolación 1D



# Otro ejemplo (wiki)



**Figura :** Interpolación mediante el vecino más cercano, bilineal y bicúbica



# Esquema

- 1 Introducción
- 2 Vecino más cercano (Nearest neighbor)
- 3 Bilineal
- 4 Bicúbica
- 5 Matlab**

# Interpolación genérica

```
clear all
I=imread('lena_eye.png');
I=double(I);

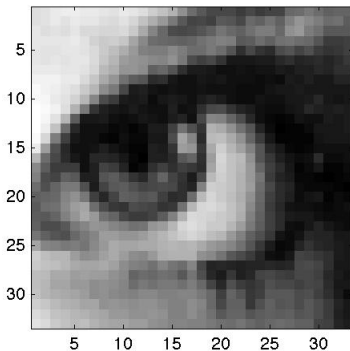
[m n]= size(I);
[x,y] = meshgrid(1:n, 1:m);           % rejilla imagen inicial

r=0.5;                                % factor de escala
[p,q]=meshgrid(1:r:n, 1:r:m);        % rejilla imagen final
I2=interp2(x,y,I,p,q,'nearest');     % interpolación
                                     % 'nearest', ...
                                     % 'bilinear','bicubic'

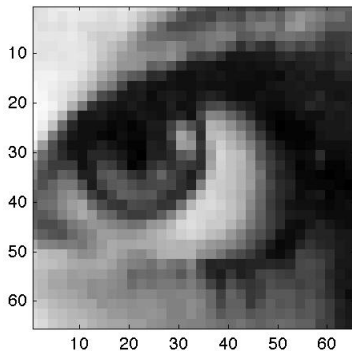
figure
subplot(1,2,1),imagesc(I),axis image
title('Original','FontSize',18)
subplot(1,2,2),imagesc(I2),axis image
title('Interpolador NN ','FontSize',18)
colormap(gray)

print -djpeg eye_ori_NN.jpg
```

Original



Interpolador NN



# Interpolación. Órdenes directas

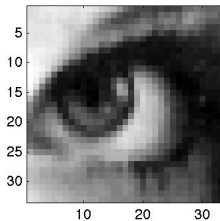
```
clear all
I=imread('lena_eye.png');
I=double(I);

r=2;theta=45;
I2=imresize(I,r,'bicubic');           % escalado con factor r
I3=imrotate(I2,theta,'bicubic');      % rota theta grados
I4=imrotate(I2,theta,'bicubic','crop'); % 'crop' -> tamaño original

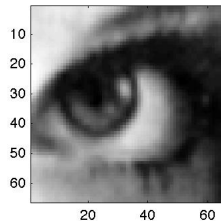
figure
subplot(2,2,1),imagesc(I),axis image
title('Original','FontSize',18)
subplot(2,2,2),imagesc(I2),axis image
title('Escalado','FontSize',18)
subplot(2,2,3),imagesc(I3),axis image
title('Rotado escalado','FontSize',18)
subplot(2,2,4),imagesc(I4),axis image
title('...y recortado','FontSize',18)
colormap(gray)

print -djpeg eye_several.jpg
```

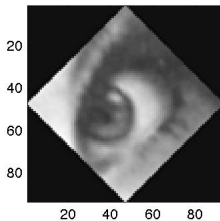
Original



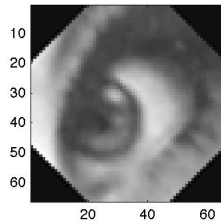
Escalado



Rotado, escalado



...y recortado



# Interpolación de imágenes. Transformaciones afines

```
clear all
I=imread('lena_eye.png');
I=double(I);

% cizallamiento
tform2 = maketform('affine',[1 0 0; .5 1 0; 0 0 1]);
I2 = imtransform(I,tform2);

% rotación
theta=pi/4;
A=[cos(theta) sin(theta) 0; -sin(theta) cos(theta) 0; 0 0 1];
tform3 = maketform('affine',A);
I3 = imtransform(I,tform3);

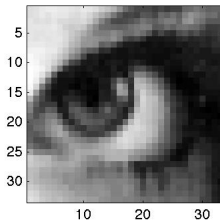
% composición
tform4 = maketform('composite',[tform2,tform3]);
I4 = imtransform(I,tform4);
```

# Interpolación de imágenes. Transformaciones afines

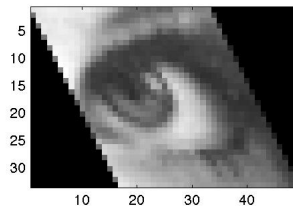
```
figure
subplot(2,2,1),imagesc(I),axis image
title('Original','FontSize',18)
subplot(2,2,2),imagesc(I2),axis image
title('Cizallado horizontal','FontSize',18)
subplot(2,2,3),imagesc(I3),axis image
title('Rotación','FontSize',18)
subplot(2,2,4),imagesc(I4),axis image
title('Composición','FontSize',18)
colormap(gray)

print -djpeg eye_custom.jpg
```

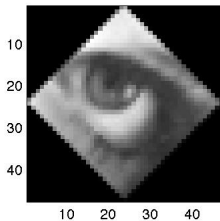
Original



Cizallado horizontal



Rotación



Composición

