

Image interpolation

Escuela de Ingeniería Informática de Oviedo

Content

- 1 Introduction
- 2 Nearest neighbor
- 3 Bilinear interpolation
- 4 Bicubic
- 5 Matlab

Outline

- 1 Introduction**
- 2 Nearest neighbor
- 3 Bilinear interpolation
- 4 Bicubic
- 5 Matlab

Image interpolation occurs in all digital images at some stage

- Resizing (resampling)
- Remapping (geometrical transformations- rotation, change of perspective,...)
- Inpainting (restauration of *holes*)
- Morphing, nonlinear transformations

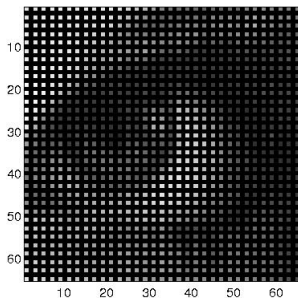
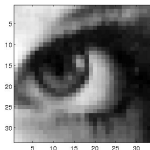


Image interpolation occurs in all digital images at some stage

- Resizing (resampling)
- Remapping (geometrical transformations- rotation, change of perspective,...)
- Inpainting (restauration of *holes*)
- Morphing, nonlinear transformations

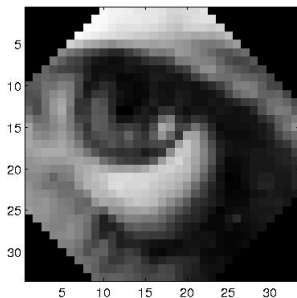
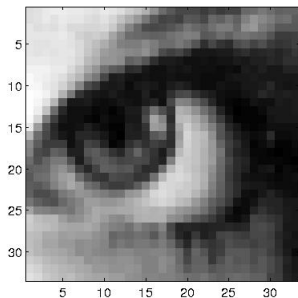


Image interpolation occurs in all digital images at some stage

- Resizing (resampling)
- Remapping (geometrical transformations- rotation, change of perspective,...)
- Inpainting (restauration of *holes*)
- Morphing, nonlinear transformations



Image interpolation occurs in all digital images at some stage

- Resizing (resampling)
- Remapping (geometrical transformations- rotation, change of perspective,...)
- Inpainting (restauration of *holes*)
- Morphing, nonlinear transformations



Outline

1 Introduction

2 Nearest neighbor

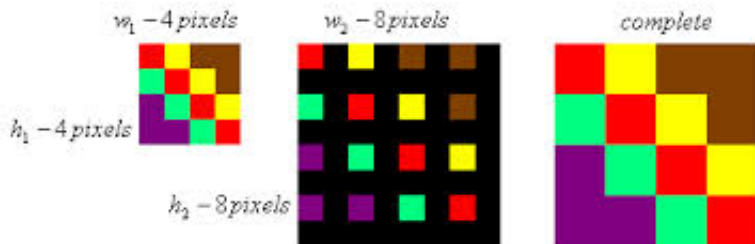
3 Bilinear interpolation

4 Bicubic

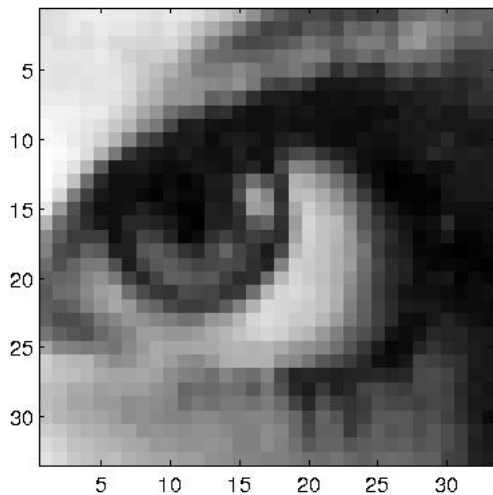
5 Matlab

Nearest neighbor

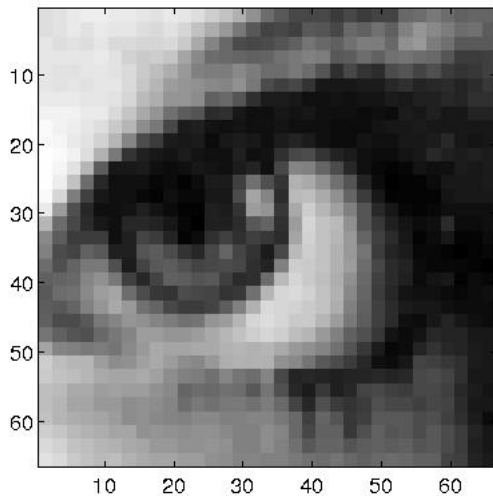
- Most basic method
- Requires the least processing time
- Only considers one pixel: the closest one to the interpolated point
- Has the effect of simply making each pixel bigger



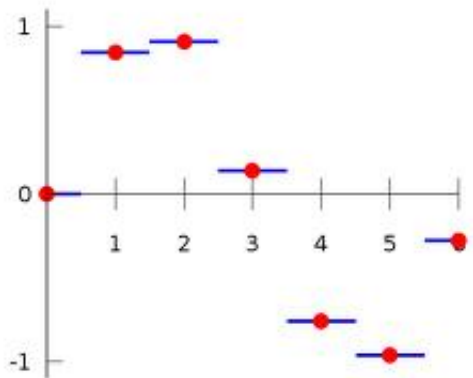
Nearest neighbor



Nearest neighbor



Relationship with 1D interpolation



Outline

- 1 Introduction
- 2 Nearest neighbor
- 3 Bilinear interpolation**
- 4 Bicubic
- 5 Matlab

Bilinear

- Considers the closest 2x2 neighborhood of known pixel values surrounding the unknown pixels
- Takes a weighted average of these 4 pixels to arrive at the final interpolated values
- Results in smoother looking images than nearest neighborhood
- Needs of more processing time

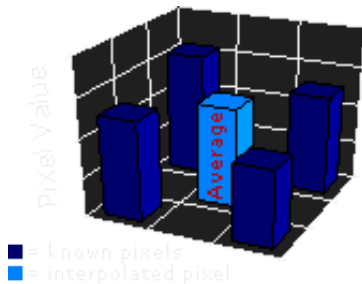
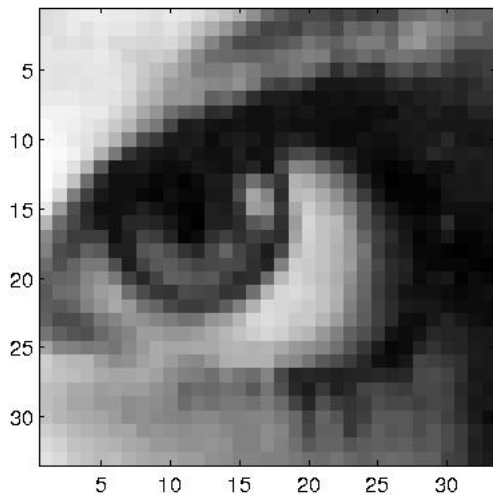
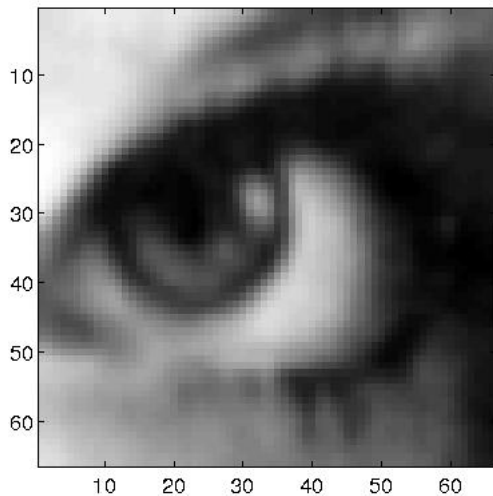


Figure: Case when all known pixel distances are equal. Interpolated value is simply their sum divided by four.

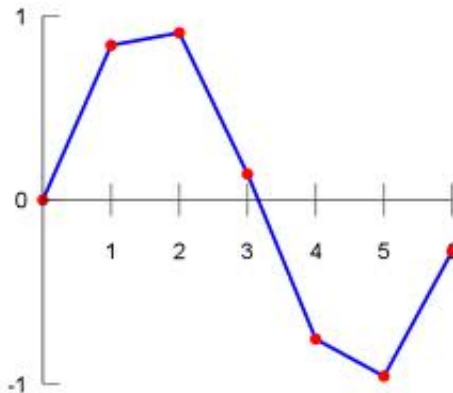
Bilinear



Bilinear



Relationship with 1D interpolation

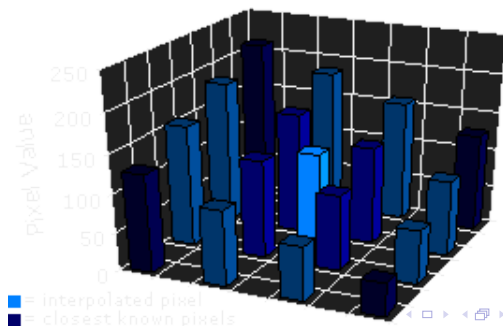


Outline

- 1 Introduction
- 2 Nearest neighbor
- 3 Bilinear interpolation
- 4 Bicubic**
- 5 Matlab

Bicubic

- One step beyond bilinear by considering the closest 4x4 neighborhood of known pixels, for a total of 16 pixels
- Since these are at various distances from the unknown pixel, closer pixels are given a higher weighting in the calculation
- Produces sharper images than the previous two methods.
- Good compromise between processing time and output quality
- Standard in many image editing programs, printer drivers and in-camera interpolation



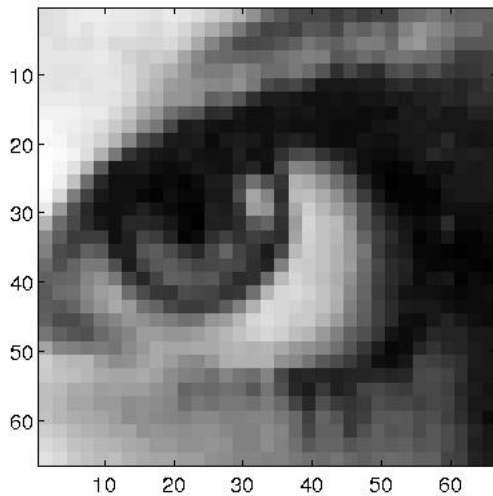


Figure: Nearest

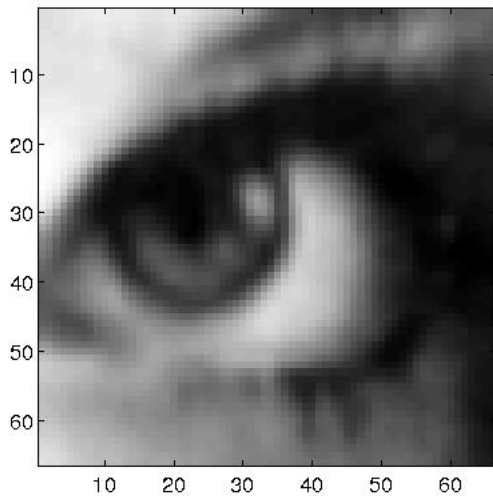


Figure: Bilinear

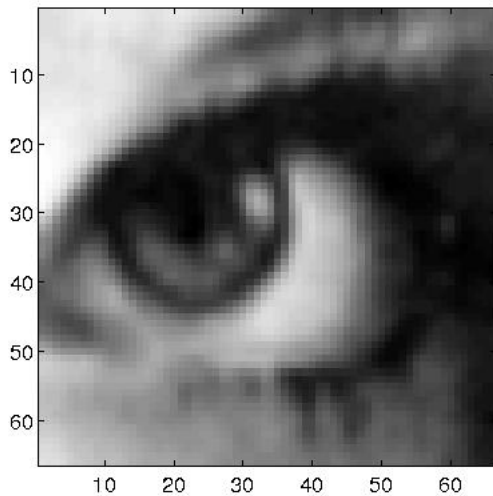
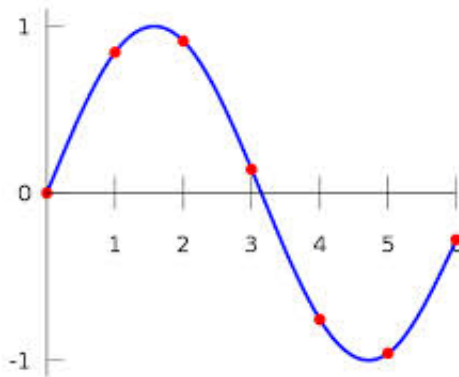


Figure: Bicubic

Relationship with 1D interpolation



Another example (wiki)

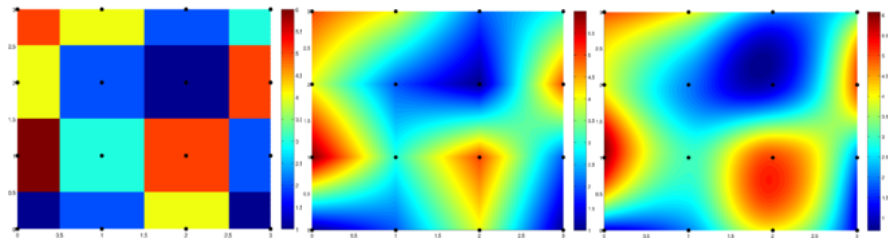


Figure: Nearest, bilinear and bicubic interpolations

Outline

- 1 Introduction
- 2 Nearest neighbor
- 3 Bilinear interpolation
- 4 Bicubic
- 5 Matlab**

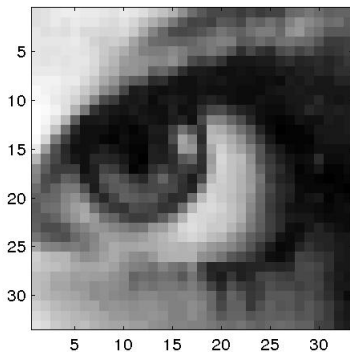
Generic interpolation

```

1 clear all
2 I=imread('lena_eye.png');
3 I=double(I);
4
5 [m n]= size(I);
6 [x,y] = meshgrid(1:n, 1:m);           % grid of input image
7
8 r=0.5;                                % scale factor
9 [p,q]=meshgrid(1:r:n, 1:r:m);        % grid for output image
10 I2=interp2(x,y,I,p,q,'nearest');    % interpolation
11                                     % 'nearest', ...
12                                     %     'bilinear','bicubic'
13
14 figure
15 subplot(1,2,1),imagesc(I),axis image
16 title('Original','FontSize',18)
17 subplot(1,2,2),imagesc(I2),axis image
18 title('NN interpolator','FontSize',18)
19 colormap(gray)
20 print -djpeg eye_ori_NN.jpg

```

Original



NN interpolator

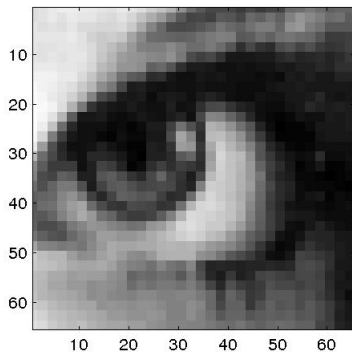
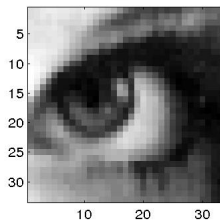


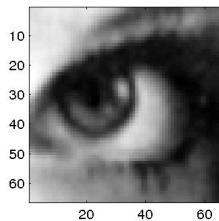
Image interpolation. Direct commands

```
1 clear all
2 I=imread('lena_eye.png');
3 I=double(I);
4
5 r=2;theta=45;
6 I2=imresize(I,r,'bicubic'); % resize by factor r
7 I3=imrotate(I2,theta,'bicubic'); % rotate theta degrees
8 I4=imrotate(I2,theta,'bicubic','crop'); % 'crop'-> original size
9
10 figure
11 subplot(2,2,1),imagesc(I),axis image
12 title('Original','FontSize',18)
13 subplot(2,2,2),imagesc(I2),axis image
14 title('Resize','FontSize',18)
15 subplot(2,2,3),imagesc(I3),axis image
16 title('Rotate resized','FontSize',18)
17 subplot(2,2,4),imagesc(I4),axis image
18 title('...and cropped','FontSize',18)
19 colormap(gray)
20
21 print -djpeg eye_several.jpg
```

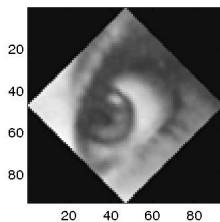
Original



Resize



Rotate resized



...and cropped

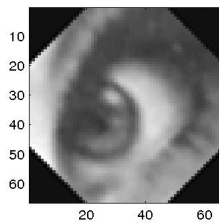


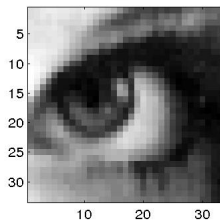
Image interpolation. Affine transformations

```

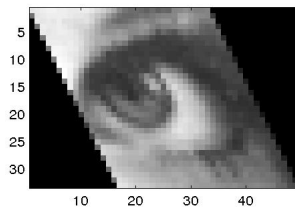
1 clear all
2 I=imread('lena_eye.png');
3 I=double(I);
4
5 tform2 = maketform('affine',[1 0 0; .5 1 0; 0 0 1]);           % ...
   shear
6 I2 = imtransform(I,tform2);
7
8 theta=pi/4;                                                   % ...
   rotation
9 A=[cos(theta) sin(theta) 0; -sin(theta) cos(theta) 0; 0 0 1];
10 tform3 = maketform('affine',A);
11 I3 = imtransform(I,tform3);
12
13 tform4 = maketform('composite',[tform2,tform3]);             % ...
   composition
14 I4 = imtransform(I,tform4);
15
16 figure
17 subplot(2,2,1),imagesc(I),axis image
18 title('Original','FontSize',18)
19 subplot(2,2,2),imagesc(I2),axis image

```

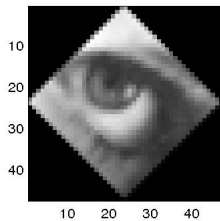
Original



Horizontal shear



Rotate



composition

