cs155 - z sweedyk

digital image processing

types of techniques

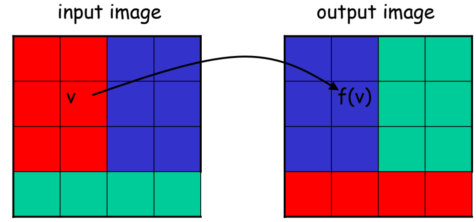
- simple pixel modification
- interpolation/extrapolation
- compositing
- convolution
- dithering
- warping
- morphing
- misc. effects

types of techniques

- **simple pixel modification**
- interpolation/extrapolation
- compositing
- convolution
- dithering
- warping
- morphing
- misc. effects

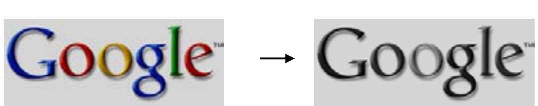
simple pixel modification

$f:[0,1]^3 \rightarrow [0,1]^3$



apply function **f** to each pixel of input image

convert to gray



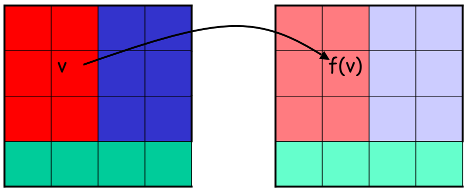
convert to gray

$f(r,g,b) = .3r + .59g + .11b$

simple pixel modification

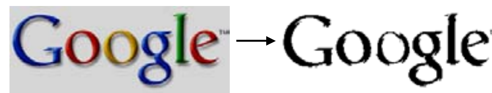
f:[0,1] -> [0,1]

input image output image



apply function **f** to each channel of each pixel of input image

threshold



threshold

if $v>t$ then $f(v)=1$

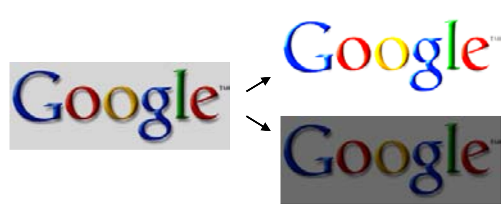
else $f(v)=0$

invert

invert

$f(v) = 1-v$

brighten/darken



brighten/darken

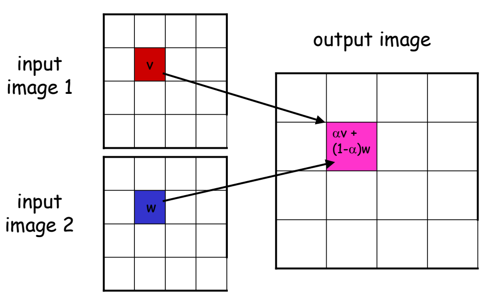
$f(v) = \alpha v$ for $\alpha \geq 0$

clamp to [0,1]

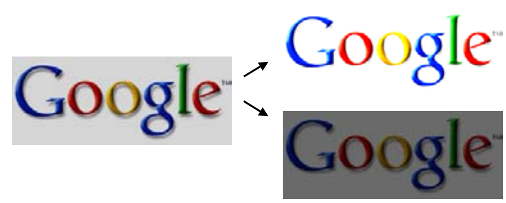
types of techniques

- simple pixel modification
- **interpolation/extrapolation**
- compositing
- convolution
- dithering
- warping
- morphing
- non-photo-realistic effects

interpolation/extrapolation



interpolation/extrapolation



brighten/darken

interpolate/extrapolate image with

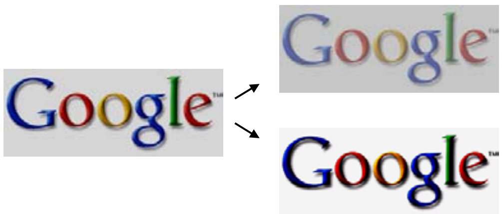
____

invert

interpolate/extrapolate image with

____

change contrast

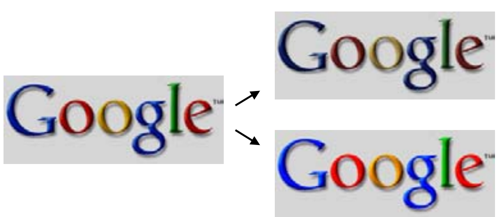


change contrast

interpolate/extrapolate image with

____

change saturation

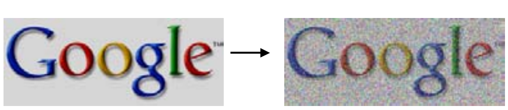


change saturation

interpolate/extrapolate image with

____

noisify



noisify

interpolate/extrapolate with

____

type of techniques

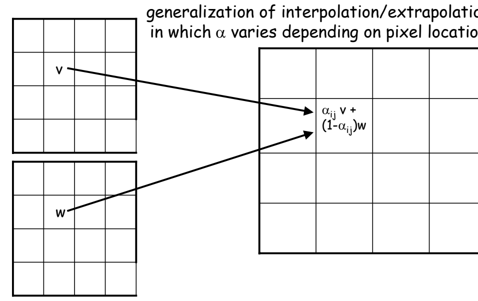
- simple pixel modification
- interpolation/extrapolation
- **compositing**
- convolution
- dithering
- warping
- morphing
- misc. effects

compositing

compositing

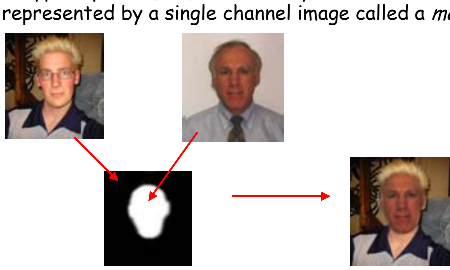
compositing

generalization of interpolation/extrapolation in which α varies depending on pixel location



compositing

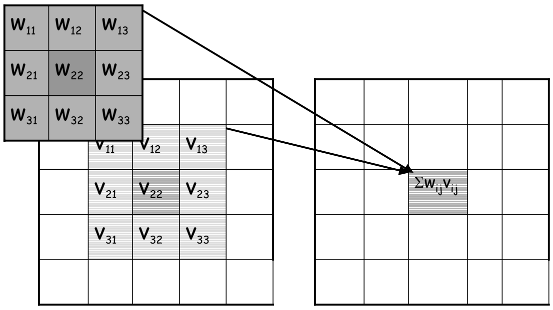
typically $\alpha \in [0,1]$ so the array of α values can represented by a single channel image called a *mask*



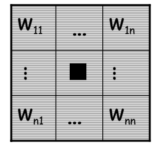
type of techniques

- simple pixel modification
- interpolation/extrapolation
- compositing
- **convolution**
- dithering
- warping
- morphing
- misc. effects

convolution

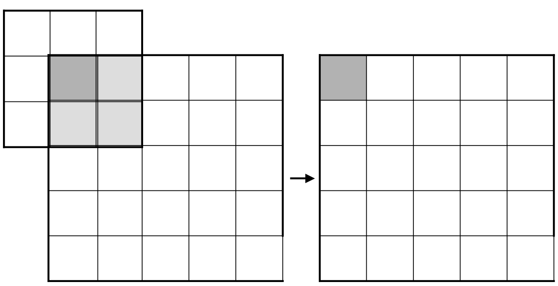


kernel

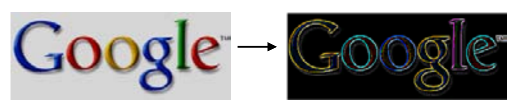


n odd

boundaries?



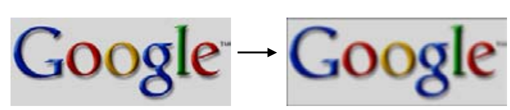
edge detect



edge detect kernel

-1/8	-1/8	-1/8
-1/8	1	-1/8
-1/8	-1/8	-1/8

blur



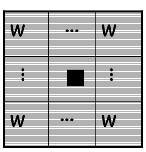
why blur?

anti-aliasing

3x3 box blur

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

nXn box blur



$w=1/n^2$

why is it important that the sum of the weights is 1?

separability

a kernel is separable if $W_{ij}=w_i w_j$

is the box filter separable?

box blur vs. triangle blur

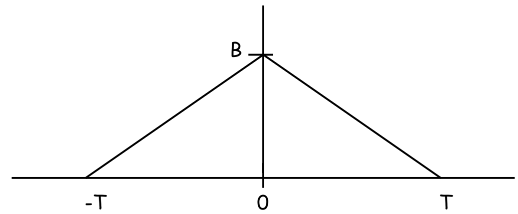
3x3 triangle blur

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

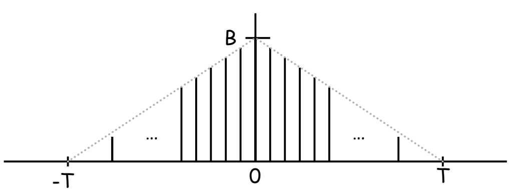
separability

1/4	1/16	1/8	1/16
1/2	1/8	1/4	1/8
1/4	1/16	1/8	1/16
	1/4	1/2	1/4

triangle function

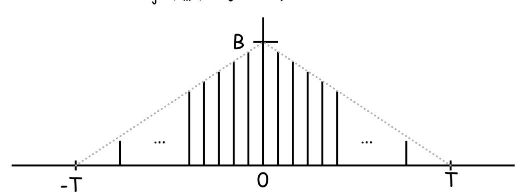


discrete triangle

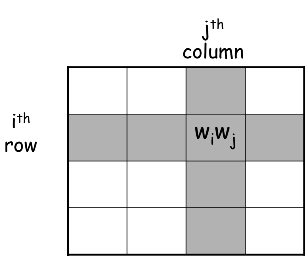


normalized, discrete triangle

1. T= (n+1)/2 gives n non-zero samples
2. $\Sigma_{j=-T \dots T}$ f(j) = 1 provided B=2/(n+1)



triangle blur filter

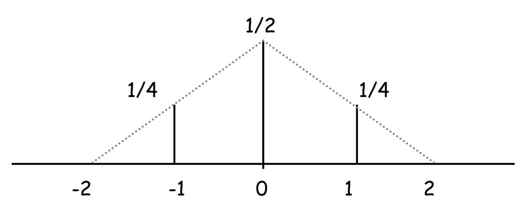


triangle samples: w_1, w_2, … , w_n

example: n=3

T=(n+1)/2 = 2

B = 2/(n+1) = 1/2



3x3 triangle blur filter

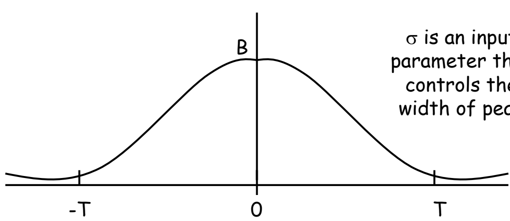
1/4	1/16	1/8	1/16
1/2	1/8	1/4	1/8
1/4	1/16	1/8	1/16
	1/4	1/2	1/4

box, triangle and guassian blurs

gaussian function

$f(x) = Be^{-x^2/\sigma^2}$

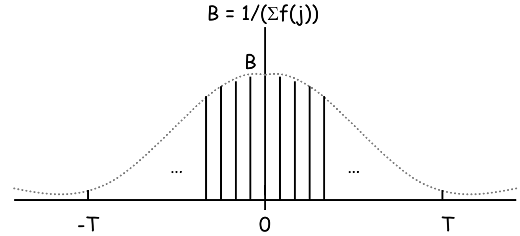
σ is an input parameter that controls the width of peak



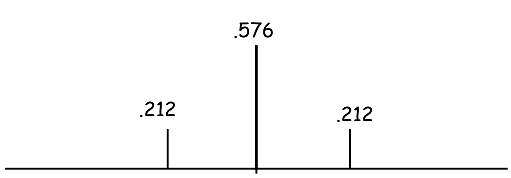
normalized, discrete version

T= (n+1)/2 gives n samples

$B = 1/(\Sigma f(j))$



example: n=3, σ=1



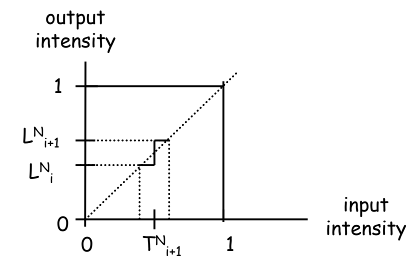
3x3 gaussian blur, σ = 1

.212	.045	.122	.045
.576	.122	.332	.122
.212	.045	.122	.045
	.212	.576	.212

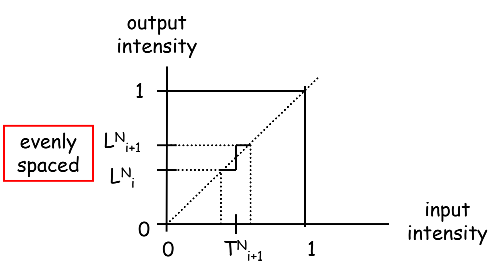
type of techniques

- simple pixel modification
- interpolation/extrapolation
- compositing
- convolution
- **dithering**
- warping
- morphing
- misc. effects

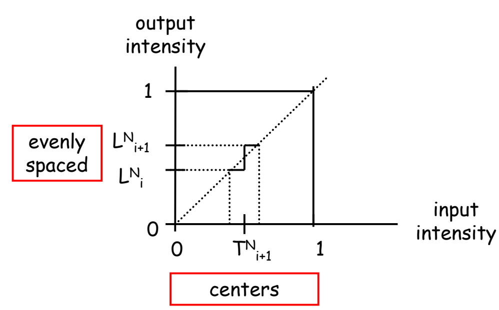
N level uniform quantization



N level uniform quantization



N level uniform quantization



quantization

8 bits per channel per pixel

1 bits per channel per pixel

random dither

add noise to camouflage quantization artifacts

8 bits per channel per pixel

1 bits per channel per pixel

1 bits per channel per pixel noisy

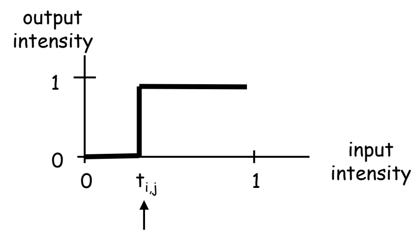
ordered dither

8 bits per channel per pixel

1 bits per channel per pixel

1 bits per channel per pixel dithered

ordered dither: 2 level output



quantization threshold depends on pixel location

ordered dither

to simulate $M=m^2+1$ ($m\geq 1$) intensity levels:

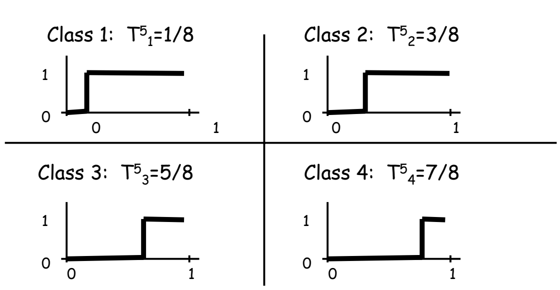
- assign pixel locations to m^2 classes
- use T^M_i as threshold to quantize pixel in i^{th} class

ordered dither: m=2

apply 2x2 mask to classify pixel locations

4	**2**	4	2	4	2	4
1	**3**	1	3	1	3	1
4	2	4	2	4	2	4
1	3	1	3	1	3	1
4	2	4	2	4	2	4

ordered dither: m=2



ordered dither: example

$\frac{1}{2}$	$\frac{1}{2}$
$\frac{1}{2}$	$\frac{1}{2}$

→

0	1
1	0

ordered dither: another view

add $\frac{1}{2}$ - $T^M{}_i$ "noise" to pixel in class i then quantize

ordered dither

To simulate $M=m^2+1$ intensity levels:

- assign pixel locations to m^2 classes
- use $T^M{}_i$ as threshold to quantize pixel in i^{th} class

how? answer: carefully

bayer's 2x2 ordered dither

4	2
1	3

bayer's 4x4 ordered dither matrix

15	8	14	6
4	12	2	10
13	5	0	7
1	9	3	11

error-diffusion dither


8 bits per channel per pixel


1 bits per channel per pixel

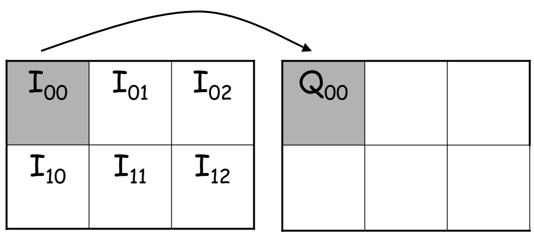

1 bits per channel per pixel dithered

error-diffusion dither

intuition

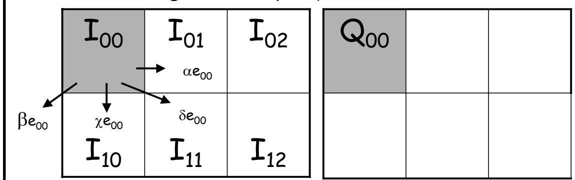
error diffusion dither

quantize I_{00} using uniform quantization



error diffusion dither

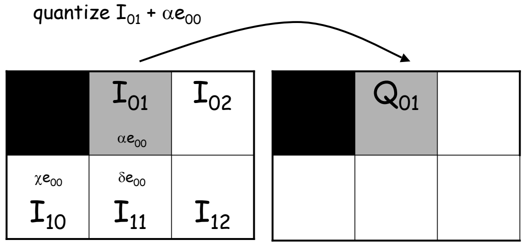
distribute error $e_{00}=I_{00} - Q_{00}$ to neighbors not yet quantized



$\alpha+\beta+\chi+\delta=1$

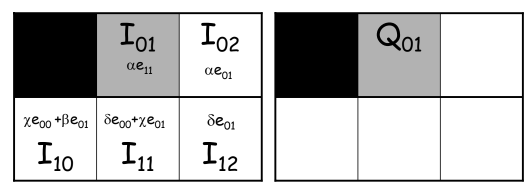
error diffusion dither

quantize $I_{01} + \alpha e_{00}$



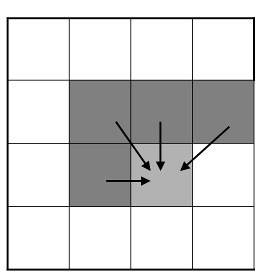
error diffusion dither

distribute error: $e_{01} = I_{01}+\alpha e_{00} - Q_{01}$



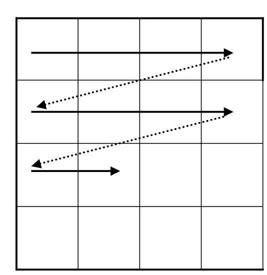
error diffusion dither

error contributions by upper & left neighbors



error diffusion dither

order of quantization is important



floyd-steinberg

$\alpha = 7/16$
$\beta = 3/16$
$\chi = 5/16$
$\delta = 1/16$

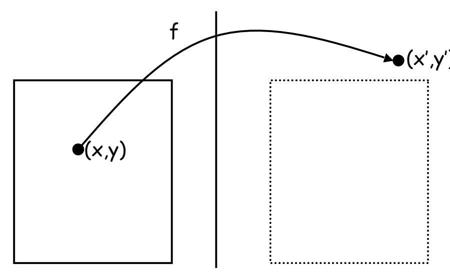
floyd-steinberg: example

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

types of techniques

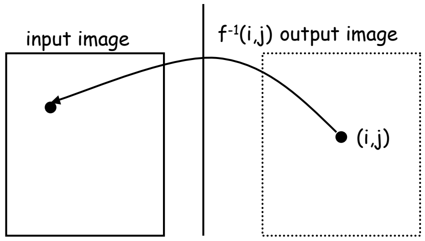
- simple pixel modification
- interpolation/extrapolation
- compositing
- convolution
- dithering
- **warping**
- morphing
- misc. effects

forward warp



f maps points in input image to the plane

forward warp



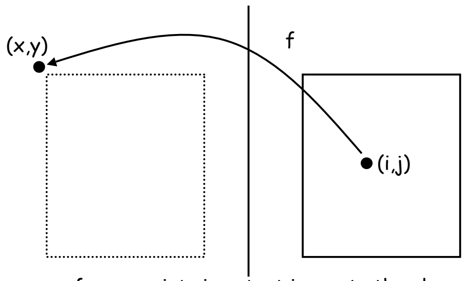
pixel at (i,j) in output image is assigned the value at location $f^{-1}(i,j)$ in input image

forward warp: problems

if f is not bijective

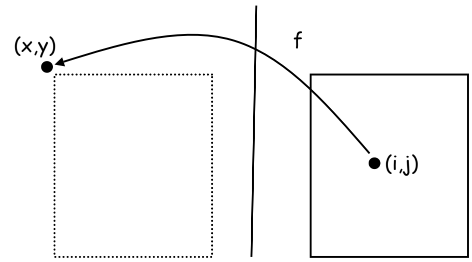
1. $f^{-1}(i,j)$ may not be defined
2. $f^{-1}(i,j)$ may not be unique

backward warp



f maps points in output image to the plane

backward warp

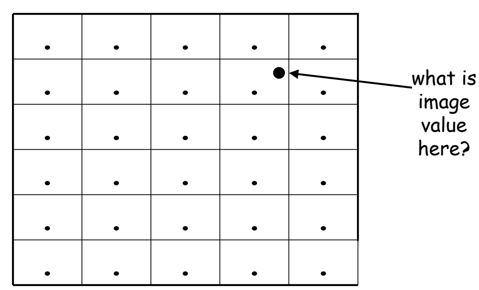


pixel (i,j) in output image is assigned value of input image at location f(i,j)

backward warp: problems

1. f(i,j) may lie outside the input image area
   solution: give image an infinite, black (or other default) border
2. f(i,j) may not lie on a sample of the input image
   solution: resample input

re-sample

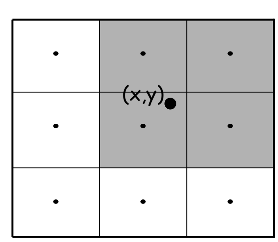


re-sample

interpolate based on nearby samples

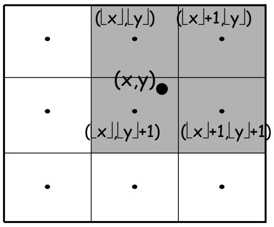
- nearest
- bilinear
- bicubic
- gaussian

which way is up?



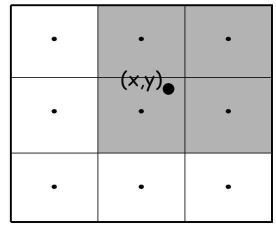
what are the coordinates of the pixels surrounding (x,y)?

which way is up?



what are the coordinates of the pixels surrounding (x,y)?

nearest

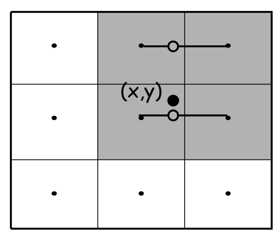


- compute distance between x,y and the locations of the neighboring samples
- set value at x,y to the value of the closest neighbor

re-sample

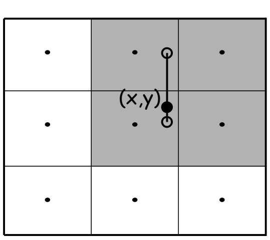
interpolate based on nearby samples
- nearest
- **bilinear**
- bicubic
- gaussian

bilinear interpolation



1. interpolate to find values at (x,⌊y⌋) and (x,⌊y⌋+1)

bilinear interpolation

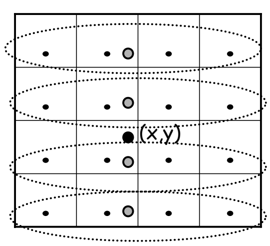


1. interpolate to find values at(x,⌊y⌋) and (x,⌊y⌋+1)
2. interpolate to find value at x,y

re-sample

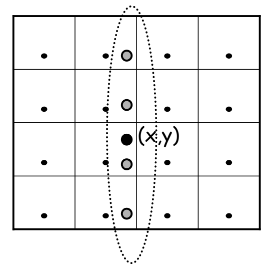
interpolate based on nearby samples
- nearest
- bilinear
- **bicubic**
- gaussian

bicubic



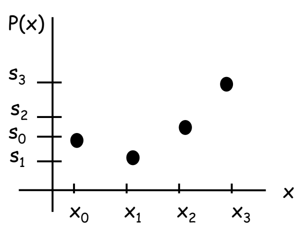
1. interpolate to find values at $(x, \lfloor y \rfloor + i)$

bicubic



1. interpolate to find values at $(x, \lfloor y \rfloor + i)$
2. interpolate to find value at (x,y)

bicubic: lagrangian



there is a unique cubic polynomial through any four distinct sample point

lagrange cubic polynomial

$$P(x) = \Sigma_{i=0,1,2,3}\, s_i\, \Pi_{j=0,1,2,3,j\neq i}\, (x-x_j)/(x_i-x_j)$$

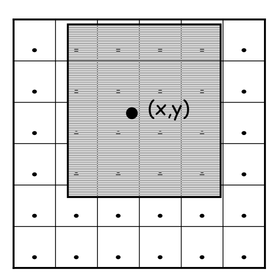
exercise: what is the value of $P(x_i)$ i=0,1,2,3

re-sample

interpolate based on nearby samples

- nearest
- bilinear
- bicubic
- **gaussian**

gaussian



interpolate nearby samples using normalized gaussian weights

unnormalized weight at (i,j) in window is

$\exp[-((x-i)^2+(y-j)^2)/\sigma^2]$

types of techniques

- simple pixel modification
- interpolation/extrapolation
- compositing
- convolution
- dithering
- warping
- **morphing**
- misc. effects

morphing how to

types of techniques

- simple pixel modification
- interpolation/extrapolation
- compositing
- convolution
- dithering
- warping
- morphing
- **non-photo-realistic effects**

non-photo-realistic effects

- emboss
- cubism
- mosaic
- etc.

See photoshop, gimp, or our own ip for more examples.