cs155 – z sweedyk

digital image processing

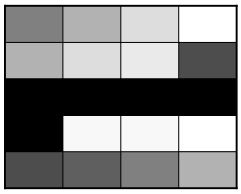
what is a digital image?

a digital image is a grid of sample color values

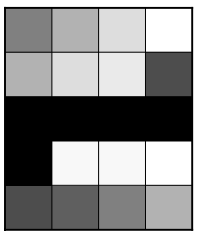
digital image example

.38	.22	.1	.009
.28	.17	.18	.58
.99	.99	.99	.98
.97	.07	.07	.009
.5	.48	.38	.21

conventional representation



digital image creation



synthesize

sample an analog image

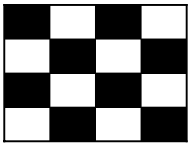
fidelity

- **discrete samples**
- quantized colors

discrete samples

nyquist criteria

sample at more than twice the highest frequency to avoid aliasing



fidelity

- discrete samples
- **quantized colors**

color quantization

- number of channels
- bits per channel

channels

1 channel

3 channel

three channel color models

- RGB: red, green, blue
- HSV: hue, saturation, value

RGB model

- In a *b*-bit image, each channel is represented by
  - an integer in the range [0 ,2 $^{b-1}$]
  - one of 2^{b-1} rational numbers in the range [0,1]
- We'll use the latter method with 0=no color and 1=full color

bits per channel

1 channel,
1 bit per channel

1 channel,
8 bits per channel

Quiz

- In a 3-channel image what color is a pixel with value
  - (1,0,0)
  - (0,1,0)
  - (1,0,1)
  - (1,1,1)
  - (0,0,0)
  - (0.5,0.5,0.5)

digital image processing

- avoid/correct errors
- restore
- enhance
- analyze
- synthesize

types of techniques

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

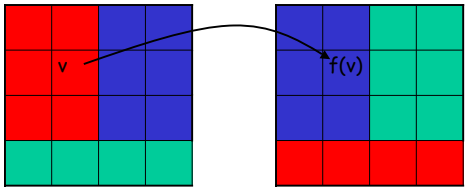
types of techniques

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

simple pixel modification

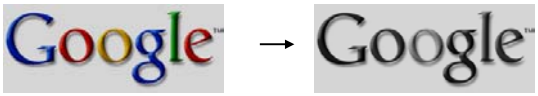
$f:[0,1]^3 \rightarrow [0,1]^3$

input image output image



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

convert to gray



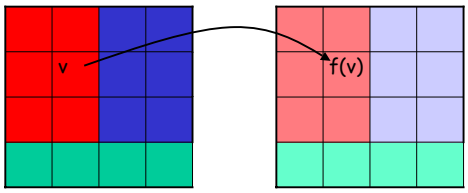
convert to gray

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

simple pixel modification

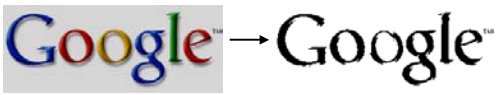
$f:[0,1] \rightarrow [0,1]$

input image output image



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

threshold



threshold

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

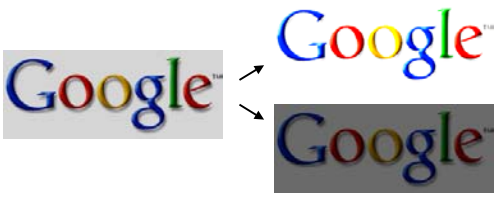
else $f(v)=0$

invert

invert

$f(v) = 1-v$

brighten/darken



brighten/darken

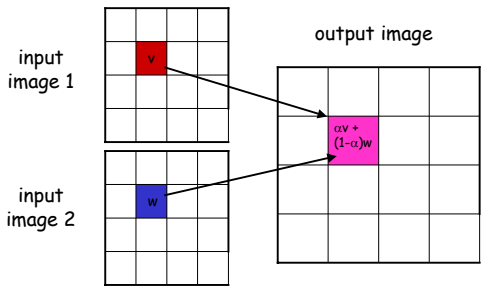
$f(v) = \alpha v$ for $\alpha \geq 0$

clamp to [0,1]

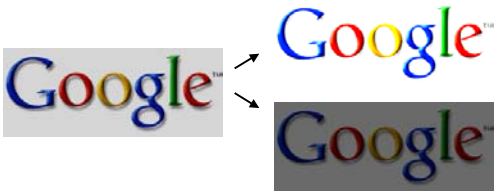
types of techniques

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

interpolation/extrapolation



interpolation/extrapolation



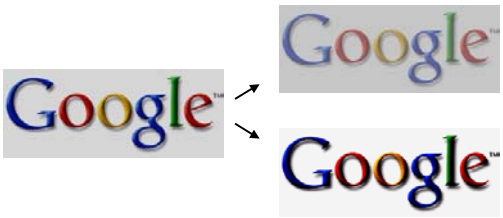
brighten/darken

interpolate/extrapolate image with

invert

interpolate/extrapolate image with

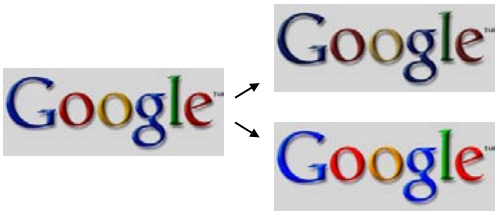
change contrast



change contrast

interpolate/extrapolate image with

change saturation

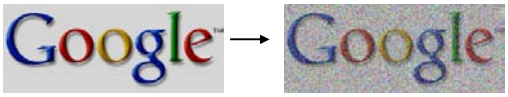


change saturation

interpolate/extrapolate image with

_____

noisify



noisify

interpolate/extrapolate with

_____

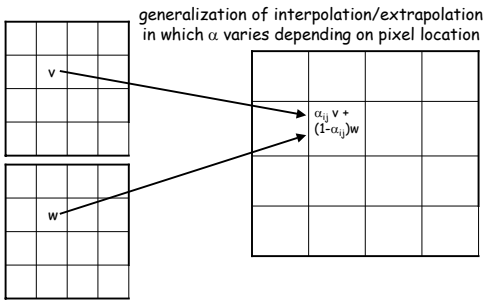
type of techniques

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

compositing

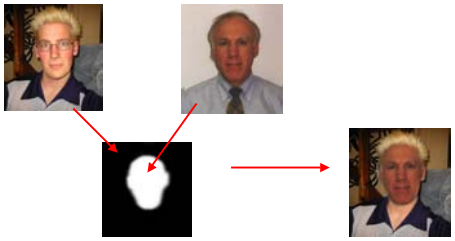
compositing

compositing



compositing

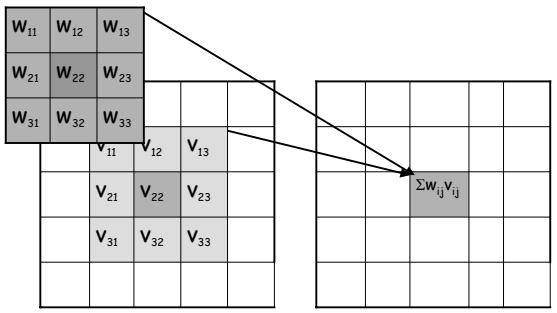
typically $\alpha \in [0,1]$ so the array of α values can represented by a single channel image called a *mask*



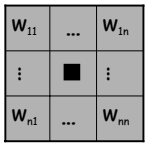
type of techniques

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

convolution

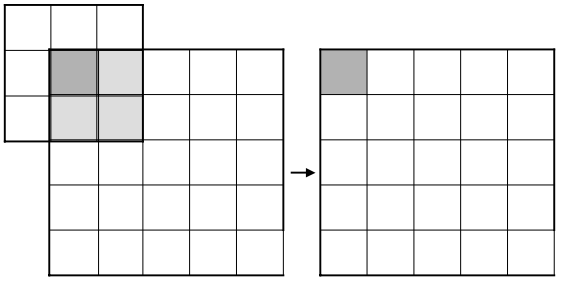


kernel

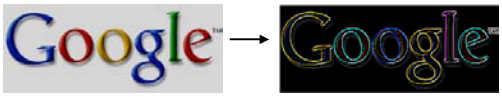


n odd

boundaries?



edge detect

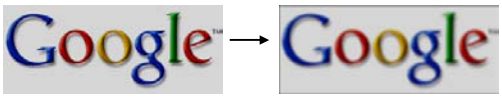


edge detect kernel

-1	-1	-1
-1	8	-1
-1	-1	-1

need to clamp values to [0,1]

blur



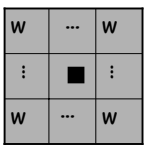
why blur?

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?

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

separability

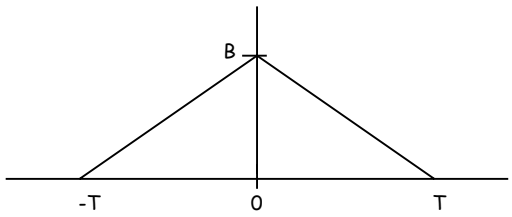
a kernel is separable if $W_{ij}=w_i w_j$

is the box filter separable?

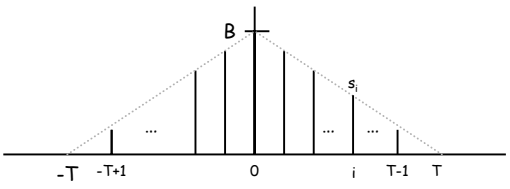
n x n triangle blur kernel

- Compute sample values for 1D triangle function: $s_{-T+1}, \ldots, s_0, \ldots s_{T-1}$
- Compute kernel $W_{ij} = s_{i-T+1} s_{j-T+1}$

triangle function

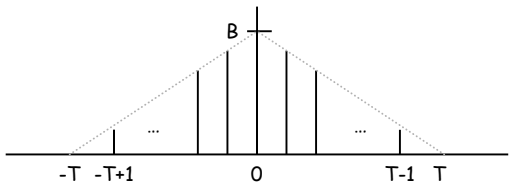


discrete triangle



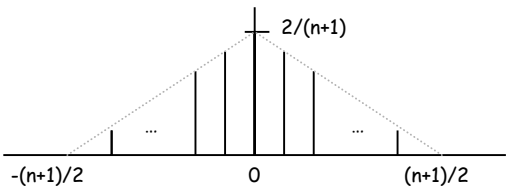
normalized, discrete triangle

1. T= (n+1)/2 chosen to give n non-zero samples
2. B=2/(n+1) chosen so that $\Sigma_{i=-T+1 \ldots T-1} s_i = 1$

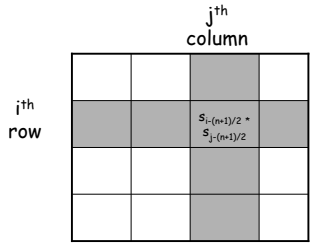


normalized, discrete triangle

N is odd so samples are taken at integer values



triangle blur filter

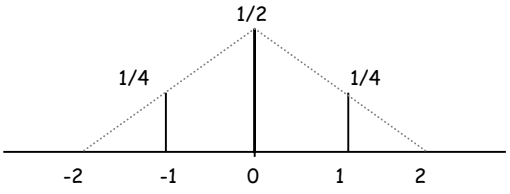


for row i=1, … , n and column j=1, … , n

example: n=3

T=(n+1)/2 = 2

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



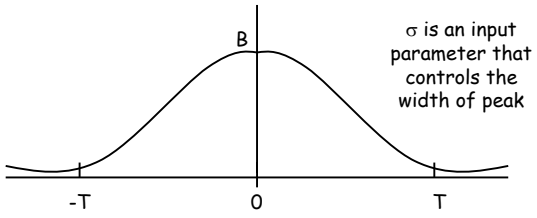
3x3 triangle blur filter

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

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

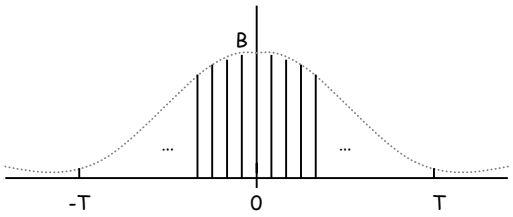


σ is an input parameter that controls the width of peak

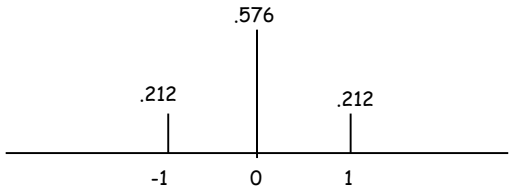
normalized, discrete version

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

B = $1/(\Sigma_{i=-T+1 \ldots T-1} e^{i^2/\sigma^2})$ is the normalizing constant



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