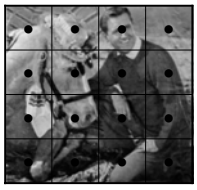
cs155 – z sweedyk

digital image processing

what is a digital image?

a digital image is a rectilinear array of samples of an analog image

sample: space & color



sampling

- space discretization
- color quantization

is it digital?

Analog, digital, analog, ...

Analog image formed by light striking lens of camera

↓

Digital image stored in computer file

↓

Analog image displayed on screen

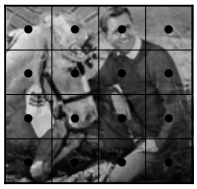
this is a digital image!!!

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

flavors of digital images

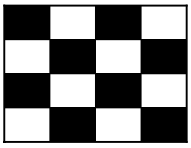
- space discretization:
  - width
  - height
- color quantization
  - number of channels
  - bits per channel

space discretization



nyquist criteria

sample at more than twice the highest frequency to avoid aliasing



aliasing in images

"jaggies"

color quantization

- number of channels
- bits per channel

channels

1 channel

3 channel

bits per channel

1 channel
1 bit per channel

1 channel
8 bits per channel

3 channels
8 bits per channel

3 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=maximum color

Quiz

- In an RGB 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
- create

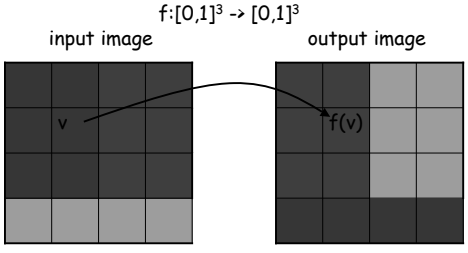
types of techniques

- 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



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

convert to gray

convert to gray

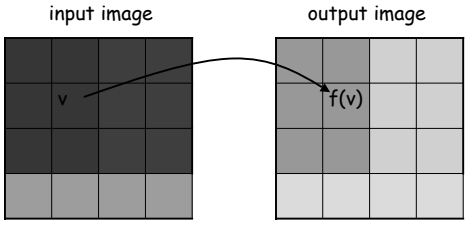
$$f(r,g,b) = c_r \bullet r + c_g \bullet g + c_b \bullet b$$

convert to gray

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

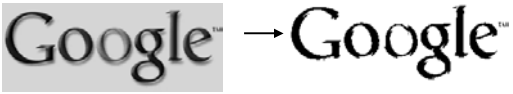
simple pixel modification

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



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

threshold



threshold

if v>t then f(v)=1

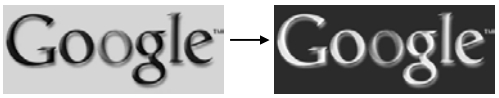
else f(v)=0

for 3-channel image:

a) convert to gray then threshold or

b) threshold each channel

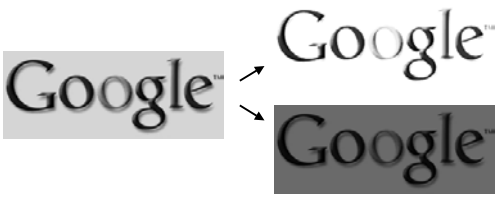
invert



invert

f(v) = 1-v

brighten/darken



brighten/darken

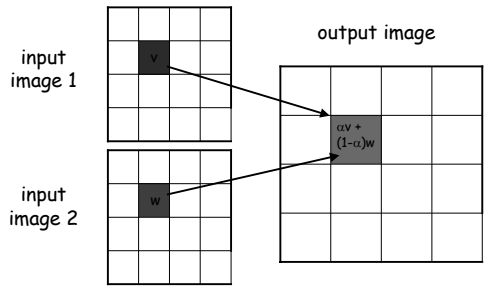
$f(v) = \alpha v$ for $\alpha \geq 0$

clamp to [0,1]

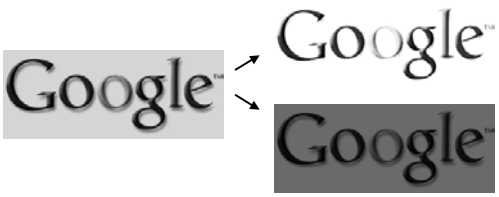
types of techniques

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

interpolation/extrapolation



interpolation/extrapolation



brighten/darken

interpolate/extrapolate image with

_____

question

what is the difference between:

- interpolating/extrapolating with white
- extrapolating/interpolating with black

invert

interpolate/extrapolate image with

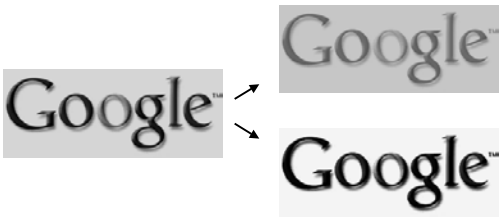
a little computation

- f(v)=1-v

- f(v) = (1- α) v + α x

What should α and x be?

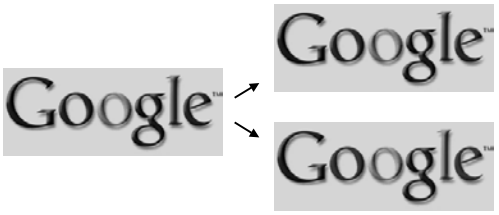
change contrast



change contrast

interpolate/extrapolate image with

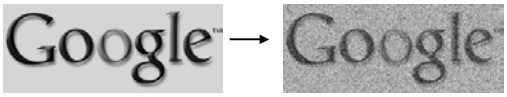
change saturation



change saturation

interpolate/extrapolate image with

noisify



noisify

interpolate/extrapolate with

type of techniques

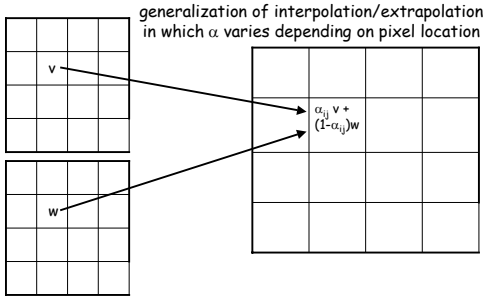
- simple pixel modification
- interpolation/extrapolation
- **compositing**
- convolution
- dithering
- warping
- morphing
- misc. effects

compositing

compositing

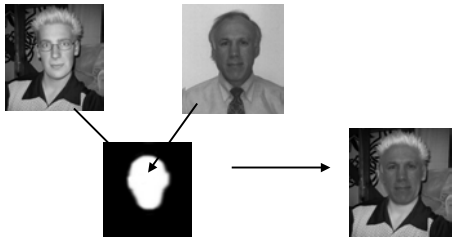
compositing

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



compositing

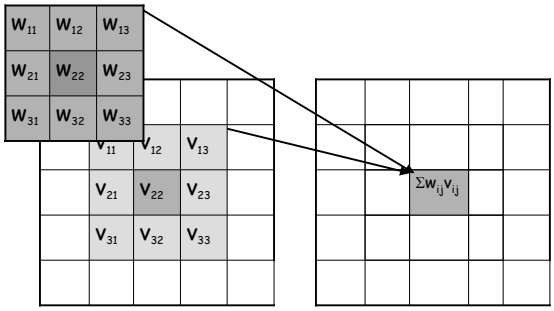
typically $\alpha \in [0,1]$ so the array of α values can represented by a single channel image called a *mask*



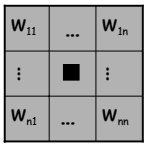
type of techniques

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

convolution

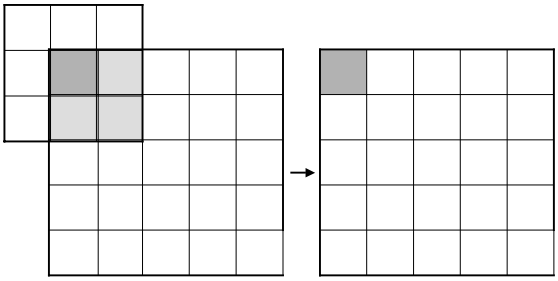


kernel

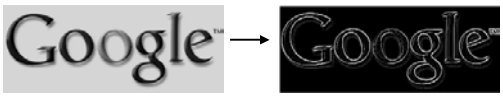


n odd

boundaries?



edge detect

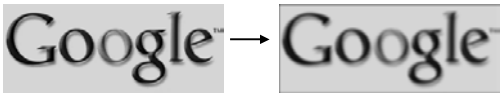


edge detect kernel

-1	-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	···	w
⋮	■	⋮
w	···	w

$w=1/n^2$

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