CMPSCI 370: Intro to Computer Vision

Image processing
[linear filtering]

University of Massachusetts, Amherst
February 11, 2016

Instructor: Subhransu Maji

Slides credit: L. Lazebnik and others

Administrivia

- Homework 2 will be posted today
  - Will be due Tue., Feb. 23 before class
  - Questions on
    - Linearity of light
    - Color constancy
    - Hybrid images — (today)
- Get started early

Enhancing images

- What can we do to “enhance” an image after it has already been digitized?
  - We can make the information that is there easier to visualize.
  - We can guess at data that is not there, but we cannot be sure, in general.

contrast enhancement

deblurring

Contrast stretching

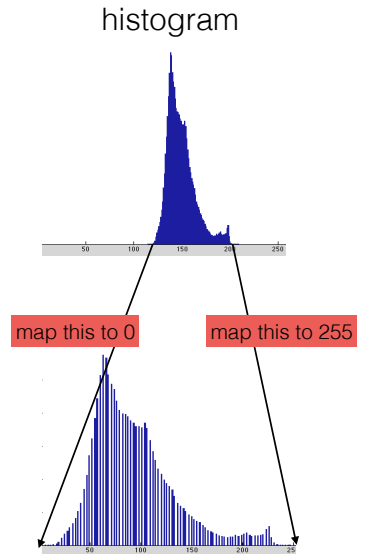
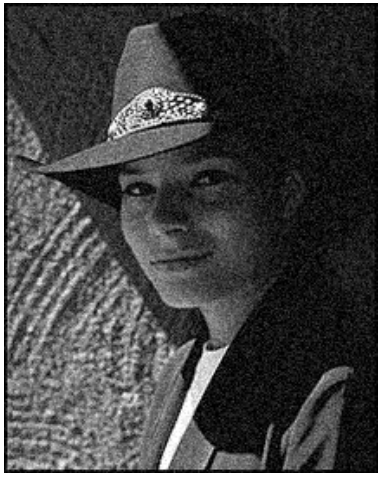


image source: wikipedia

Motivation: Image de-noising

- How can we reduce noise in a photograph?



Moving average

- Let's replace each pixel with a ***weighted*** average of its neighborhood
- The weights are called the ***filter***
- What are the weights for the average of a 3x3 neighborhood?

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

"box filter"

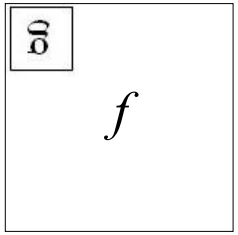
Source: D. Lowe

Convolution

- Let ***f*** be the image and ***g*** be the kernel. The output of convolving ***f*** with ***g*** is denoted ***f*** * ***g***.

$$(f * g)[m,n] = \sum_{k,l} f[m-k, n-l]\, g[k,l]$$

Convention:
kernel is "flipped"



- MATLAB functions: conv2, filter2, imfilter

Source: F. Durand

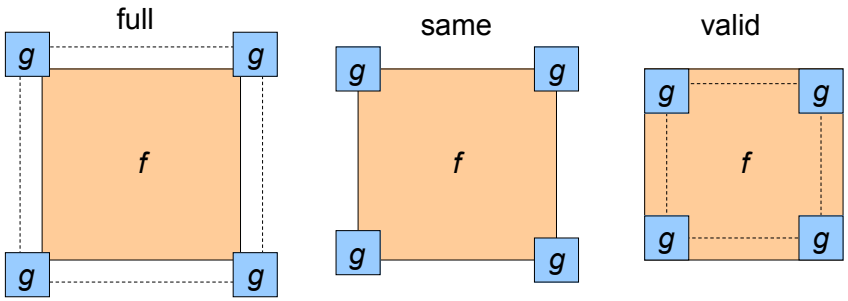
Some properties

- **Linearity:** filter(f_1 + f_2) = filter(f_1) + filter(f_2)
- **Scalars factor out:** filter(k f_1) = k filter(f_1)

Annoying details

What is the size of the output?

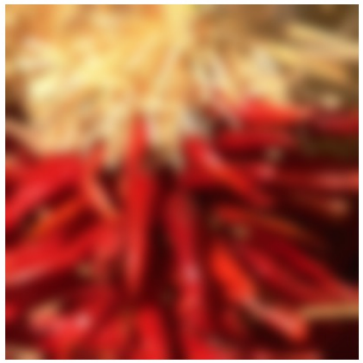
- MATLAB: filter2(g, f, ***shape***) or conv2(g, f, ***shape***)
  - ***shape*** = 'full': output size is sum of sizes of f and g
  - ***shape*** = 'same': output size is same as f
  - ***shape*** = 'valid': output size is difference of sizes of f and g



Annoying details

What about near the edge?

- the filter window falls off the edge of the image
- need to extrapolate
- methods:
  - clip filter (black)
  - wrap around
  - copy edge
  - reflect across edge

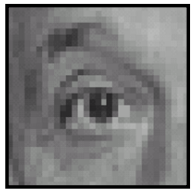


Annoying details

What about near the edge?

- the filter window falls off the edge of the image
- need to extrapolate
- methods (MATLAB):
  - clip filter (black): imfilter(f, g, 0)
  - wrap around: imfilter(f, g, 'circular')
  - copy edge: imfilter(f, g, 'replicate')
  - reflect across edge: imfilter(f, g, 'symmetric')

Practice with linear filters

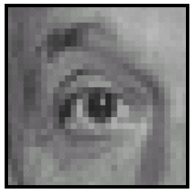


Original

0	0	0
0	1	0
0	0	0

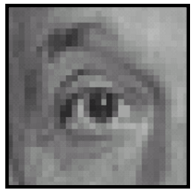
?

Practice with linear filters



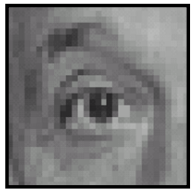
Original

0	0	0
0	1	0
0	0	0



Filtered
(no change)

Practice with linear filters

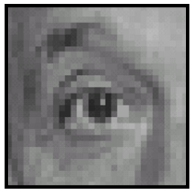


Original

0	0	0
0	0	1
0	0	0

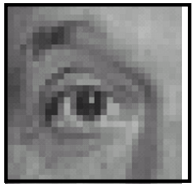
?

Practice with linear filters



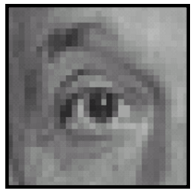
Original

0	0	0
0	0	1
0	0	0



Shifted *left*
By 1 pixel

Practice with linear filters



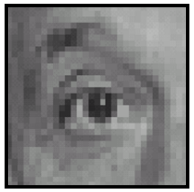
Original

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

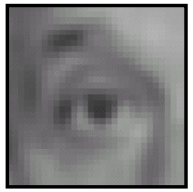
?

Practice with linear filters



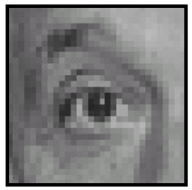
Original

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$



Blur (with a box filter)

Practice with linear filters



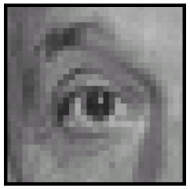
Original

$$\begin{bmatrix}0 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 0\end{bmatrix} - \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

?

(Note that filter sums to 1)

Practice with linear filters



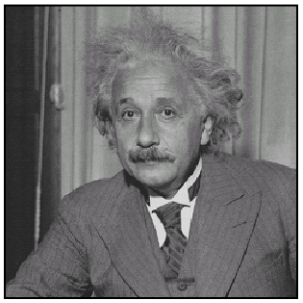
Original

$$\begin{bmatrix}0 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 0\end{bmatrix} - \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

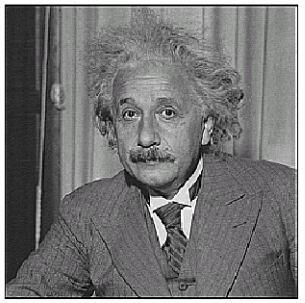
Sharpening filter

- Accentuates differences with local average

Sharpening



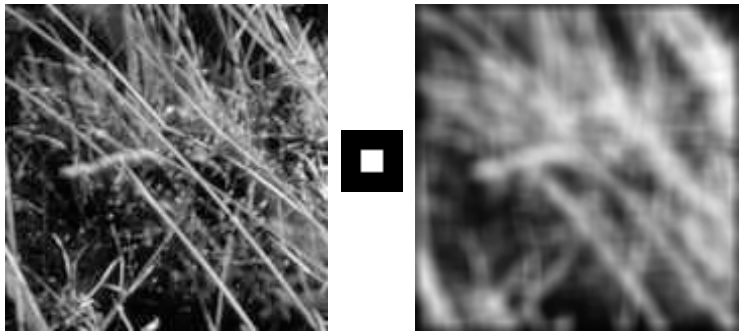
before



after

Smoothing with box filter revisited

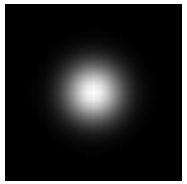
- What's wrong with this picture?
- What's the solution?



Source: D. Forsyth

Smoothing with box filter revisited

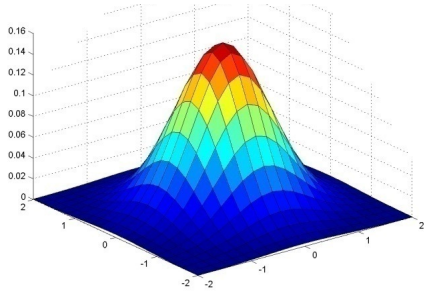
- What's wrong with this picture?
- What's the solution?
  - To eliminate edge effects, weight contribution of neighborhood pixels according to their closeness to the center



"fuzzy blob"

Gaussian Kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

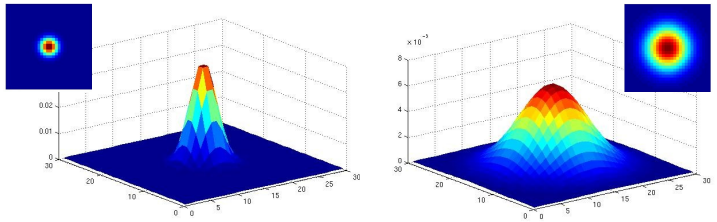


- Constant factor at front makes volume sum to 1 (can be ignored when computing the filter values, as we should renormalize weights to sum to 1 in any case)

Source: C. Rasmussen

Gaussian Kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



σ = 2 with 30 x 30 kernel

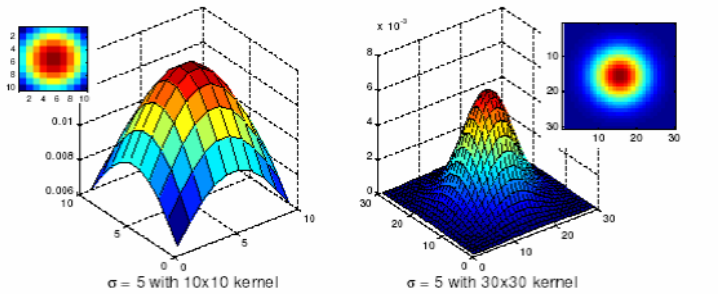
σ = 5 with 30 x 30 kernel

- Standard deviation σ: determines extent of smoothing

Source: K. Grauman

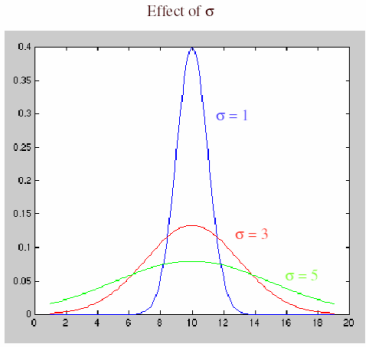
Choosing kernel width

- The Gaussian function has infinite support, but discrete filters use finite kernels



Choosing kernel width

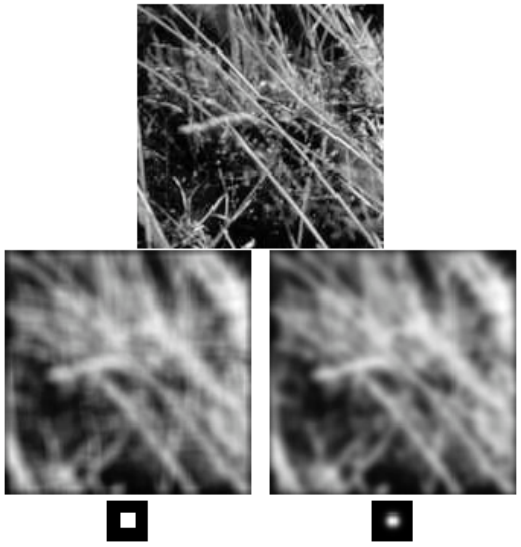
- Rule of thumb: set filter half-width to about 3σ



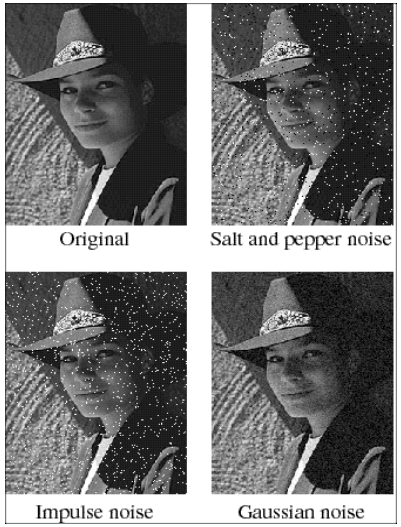
Matlab command
fspecial('gaussian', hsize, sigma)

```
>> fspecial('gaussian', 5, 1)

ans =

    0.0030    0.0133    0.0219    0.0133    0.0030
    0.0133    0.0596    0.0983    0.0596    0.0133
    0.0219    0.0983    0.1621    0.0983    0.0219
    0.0133    0.0596    0.0983    0.0596    0.0133
    0.0030    0.0133    0.0219    0.0133    0.0030
```


Gaussian vs. box filtering



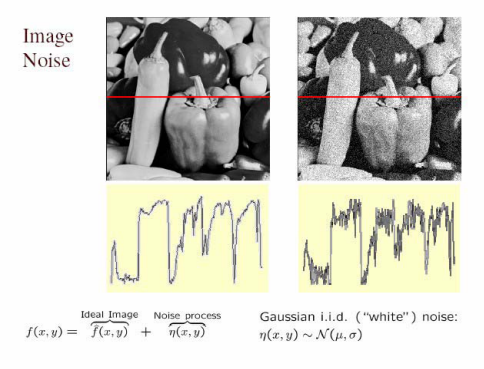
Noise



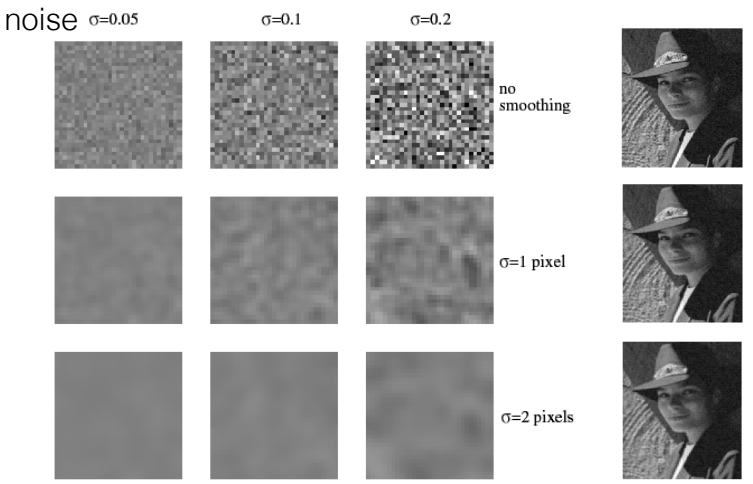
- **Salt and pepper noise**: contains random occurrences of black and white pixels
- **Impulse noise:** contains random occurrences of white pixels
- **Gaussian noise**: variations in intensity drawn from a Gaussian normal distribution

Gaussian noise

- Mathematical model: sum of many independent factors
- Good for small standard deviations
- Assumption: independent, zero-mean noise

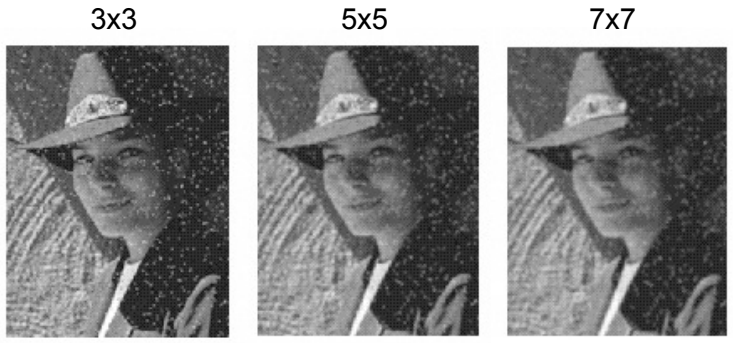


Reducing Gaussian noise



Smoothing with larger standard deviations suppresses noise, but also blurs the image

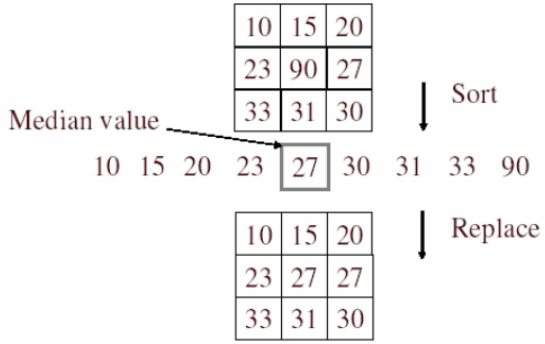
Reducing salt-and-pepper noise



What's wrong with the results?

Alternative idea: Median filtering

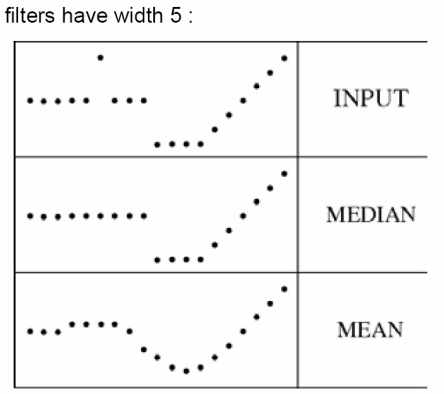
- A **median filter** operates over a window by selecting the median intensity in the window



- Is median filtering linear?

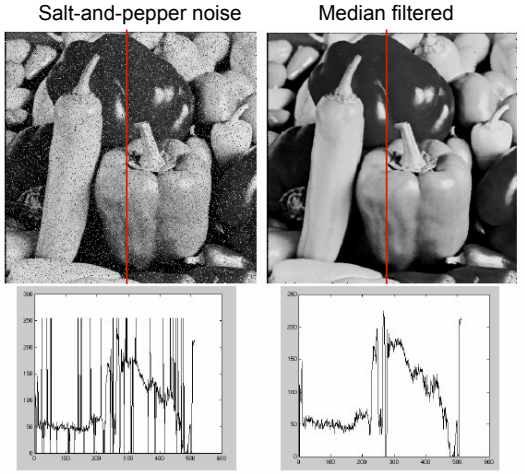
Median filter

- What advantage does median filtering have over Gaussian filtering?
  - Robustness to outliers



Source: K. Grauman 33

Median filter

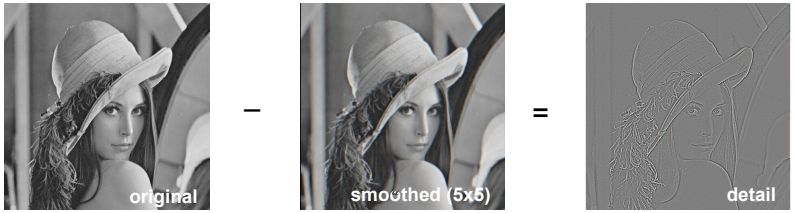


MATLAB: medfilt2(image, [h w])

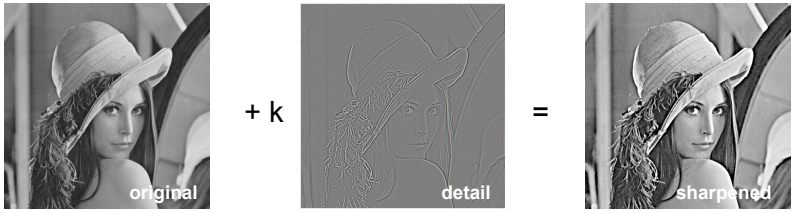
Source: M. Hebert 34

Sharpening revisited

What does blurring take away?



Let's add it back:

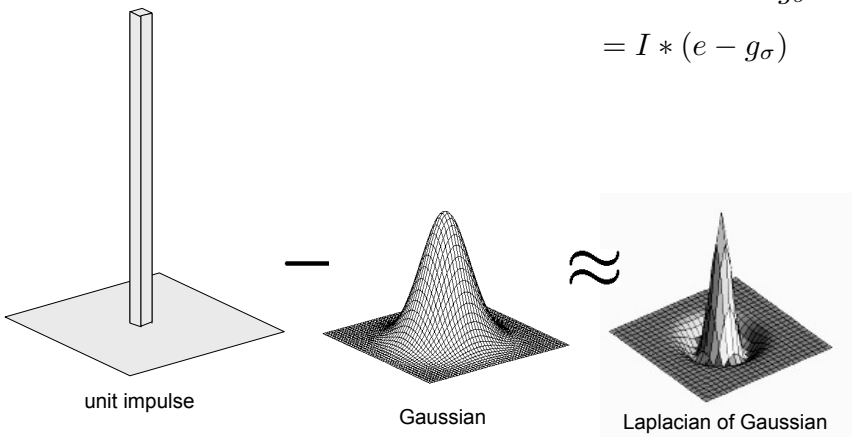


35

Sharpening filter

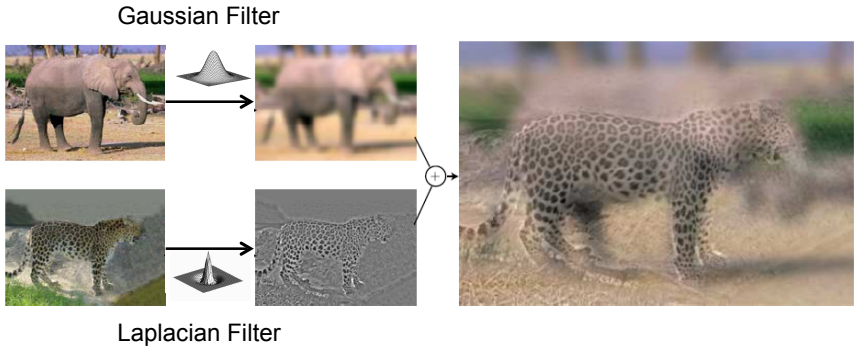
$$I = blurry(I) + sharp(I)$$

$$sharp(I) = I - blurry(I)$$
$$= I * e - I * g_\sigma$$
$$= I * (e - g_\sigma)$$

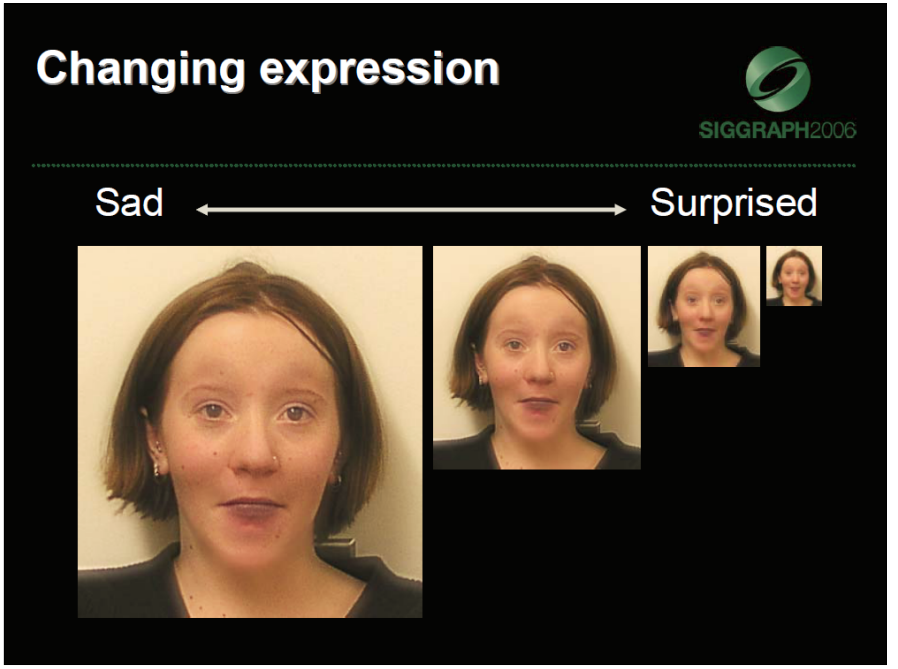


36

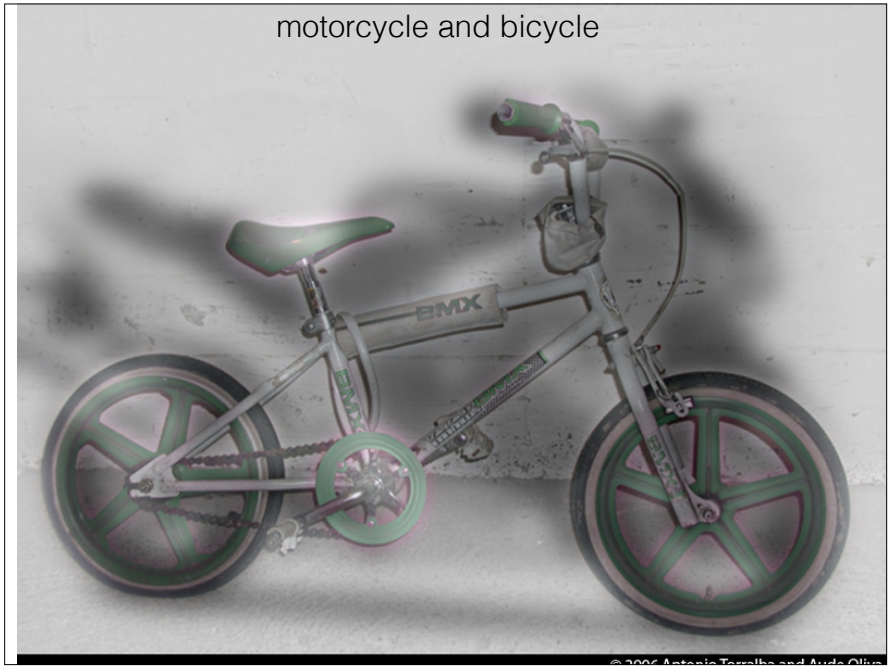
Application: Hybrid Images



A. Oliva, A. Torralba, P.G. Schyns,
"Hybrid Images," SIGGRAPH 2006



motorcycle and bicycle



dolphin and car