Robert Collins
CSE486

Lecture 10:
Pyramids and Scale Space

Recall

- Cascaded Gaussians
  - Repeated convolution by a smaller Gaussian to simulate effects of a larger one.
- G*(G*f) = (G*G)*f [associativity]

$$\boxed{G_{\sigma_1} * G_{\sigma_2} = G_{\sigma} \quad \sigma^2 = \sigma_1^{\,2} + \sigma_2^{\,2}}$$

Example: Cascaded Convolutions

[1 1] * [1 1] --> [1 2 1]

[1 1] * [1 2 1] --> [1 3 3 1]

[1 1] * [1 3 3 1] --> [1 4 6 4 1]

..and so on…

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

and so on…

Pascal's Triangle

Aside: Binomial Approximation

1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
and so on…

Pascal's Triangle

$$a_{nr} \equiv \frac{n!}{r!(n-r)!} \equiv \binom{n}{r}$$

n = number of elements in the 1D filter minus 1
r = position of element in the filter kernel (0, 1, 2…)

Binomial Coefficients

Aside: Binomial Approximation

Look at odd-length rows of Pascal's triangle:

1 1

$\boxed{1\ 2\ 1}$

1 3 3 1

$\boxed{1\ 4\ 6\ 4\ 1}$

1 5 10 10 5 1

and so on…

[1 2 1]/4 - approximates
Gaussian with sigma=1/sqrt(2)

[1 4 6 4 1]/16 - approximates
Gaussian with sigma=1

An easy way to generate integer-coefficient Gaussian approximations.

From Homework 2

Row 1	1 1	$\sigma = 1/2$
Row 2	1 2 1	$\sigma = 1/\sqrt{2}$
Row 3	1 3 3 1	$\sigma =$ ____
Row 4	1 4 6 4 1	$\sigma =$ ____
Row 5	_ _ _ _ _ _	$\sigma = \sqrt{5}/2$
Row 6	_ _ _ _ _ _ _	$\sigma = \sqrt{3/2}$

$$\boxed{G_{\sigma_1} * G_{\sigma_2} = G_{\sigma} \quad \sigma^2 = \sigma_1^2 + \sigma_2^2}$$

More about Cascaded Convolutions

(for the mathematically inclined)

Fun facts:

The distribution of the sum of two random variables X + Y is the convolution of their two distributions

Given N i.i.d. random variables, X1 … XN, the distribution of their sum approaches a Gaussian distribution (aka the central limit theorem)

Therefore:

The repeated convolution of a (nonnegative) filter with itself takes on a Gaussian shape.

recall

Gaussian Smoothing at Different Scales

original

sigma = 1

Gaussian Smoothing at Different Scales

original

sigma = 3

Gaussian Smoothing at Different Scales

original

sigma = 10

Idea for Today:

Form a Multi-Resolution Representation

original

sigma = 1

sigma = 3

sigma = 10

Pyramid Representations

Because a large amount of smoothing limits the frequency of features in the image, we do not need to keep all the pixels around!

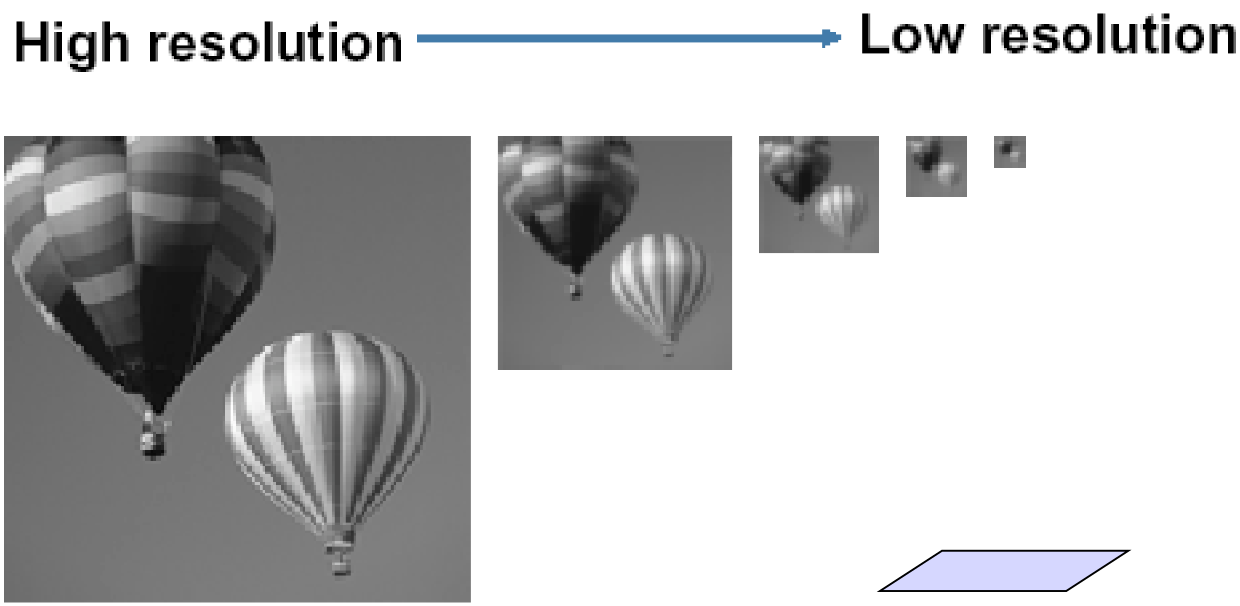
Strategy: progressively reduce the number of pixels as we smooth more and more.

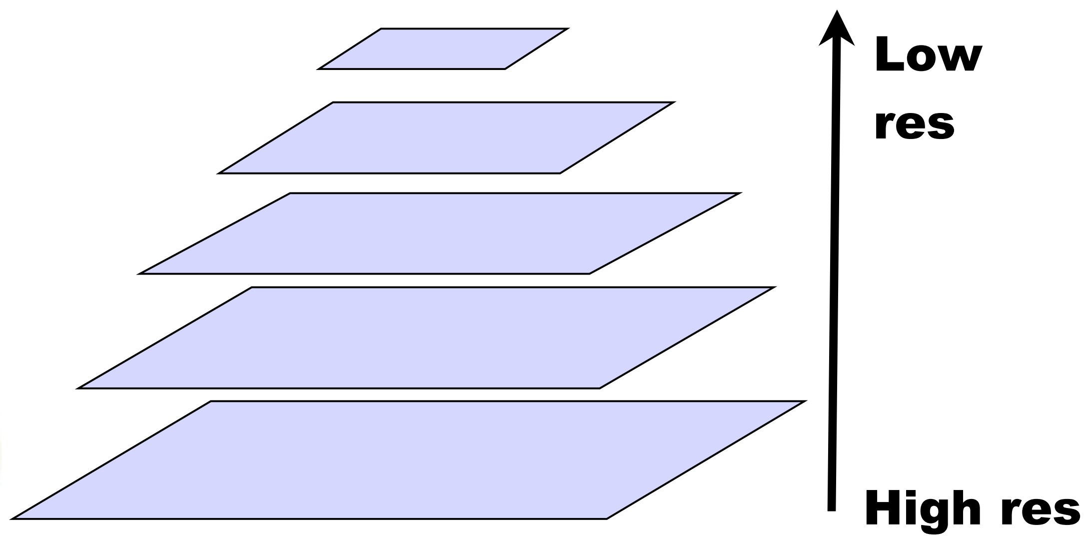
Leads to a “pyramid” representation if we subsample at each level.

Gaussian Pyramid

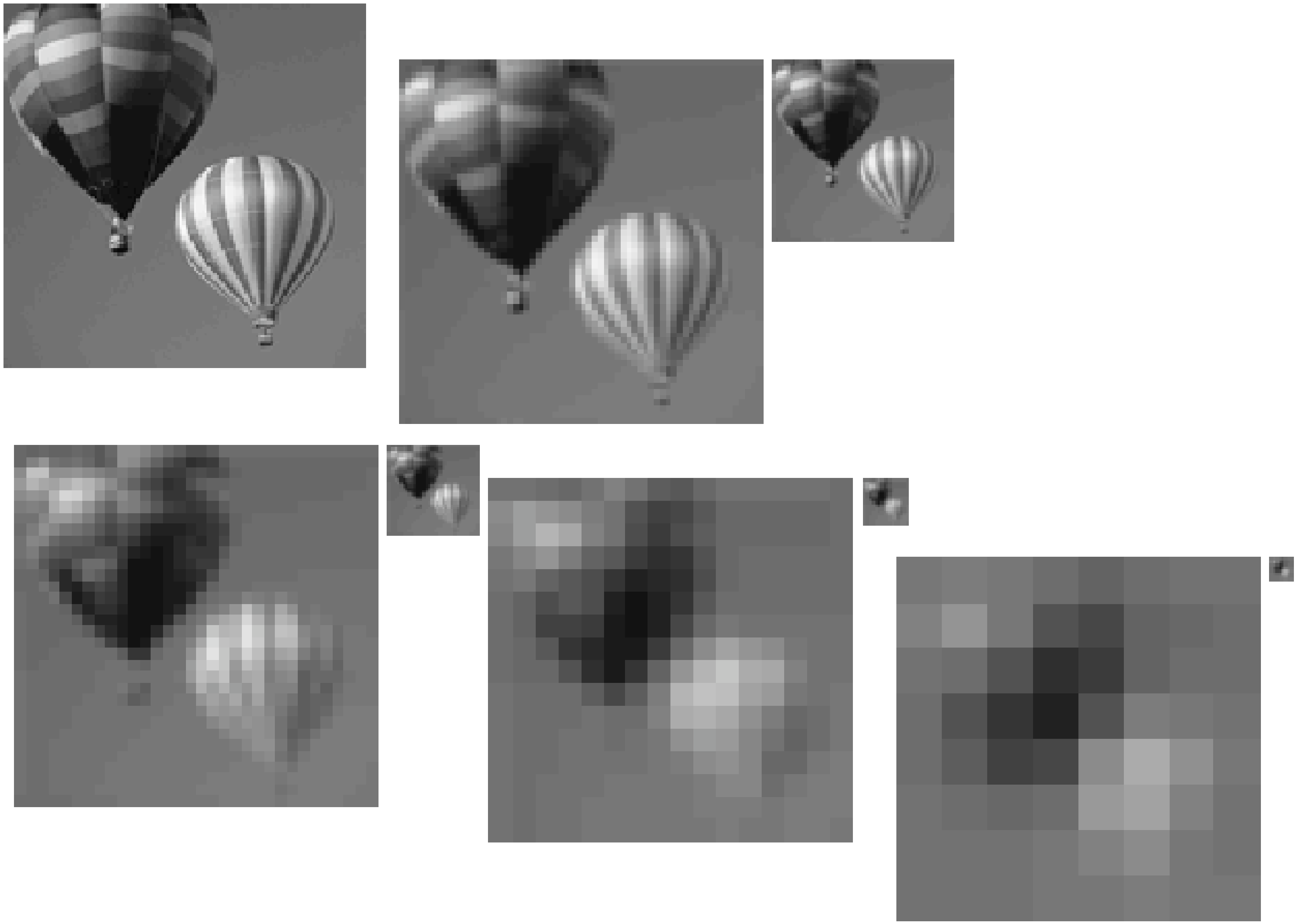
- Synthesis: Smooth image with a Gaussian and downsample. Repeat.
- Gaussian is used because it is self-reproducing (enables incremental smoothing).
- Top levels come "for free". Processing cost typically dominated by two lowest levels (highest resolution).

Gaussian Pyramid





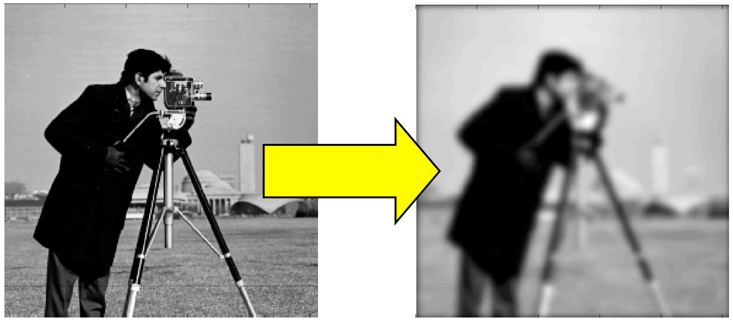
Emphasis: Smaller Images have Lower Resolution



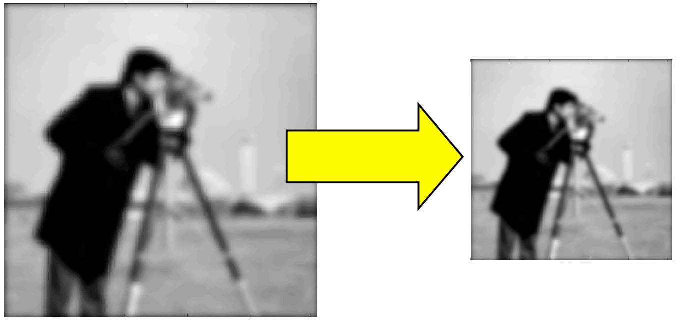
Generating a Gaussian Pyramid

Basic Functions:

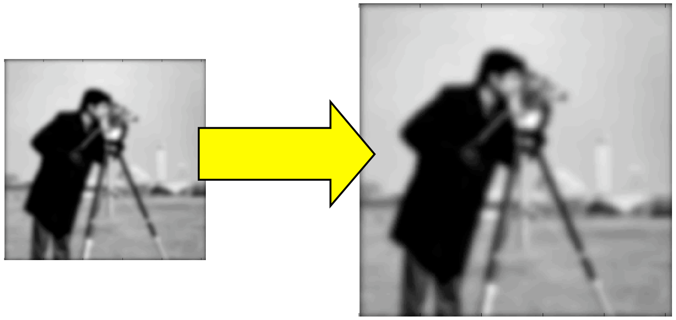
Blur (convolve with Gaussian to smooth image)



DownSample (reduce image size by half)



Upsample (double image size)

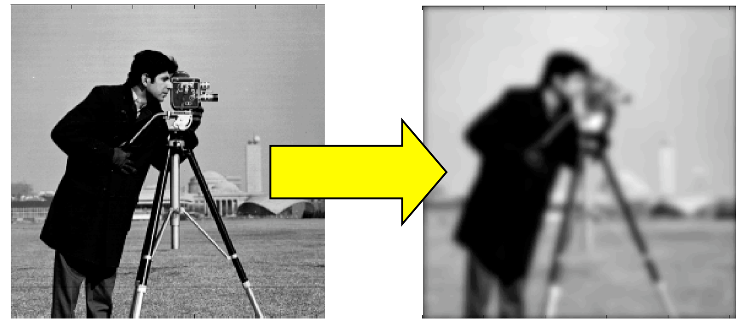


Generating a Gaussian Pyramid

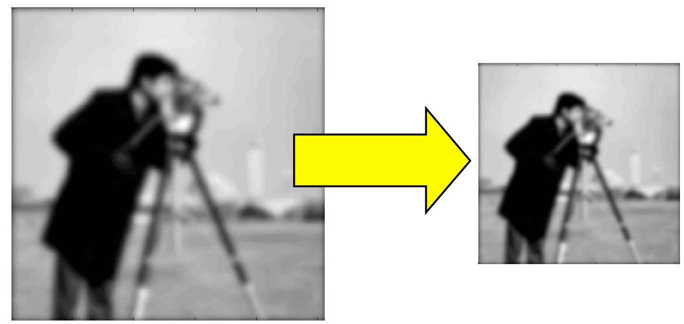
Basic Functions:

Blur (convolve with Gaussian to smooth image)

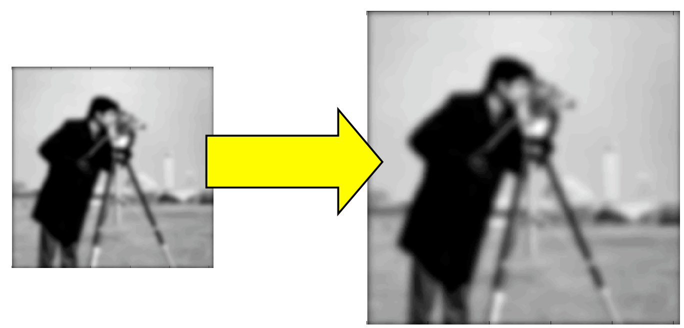
We've talked about blur already



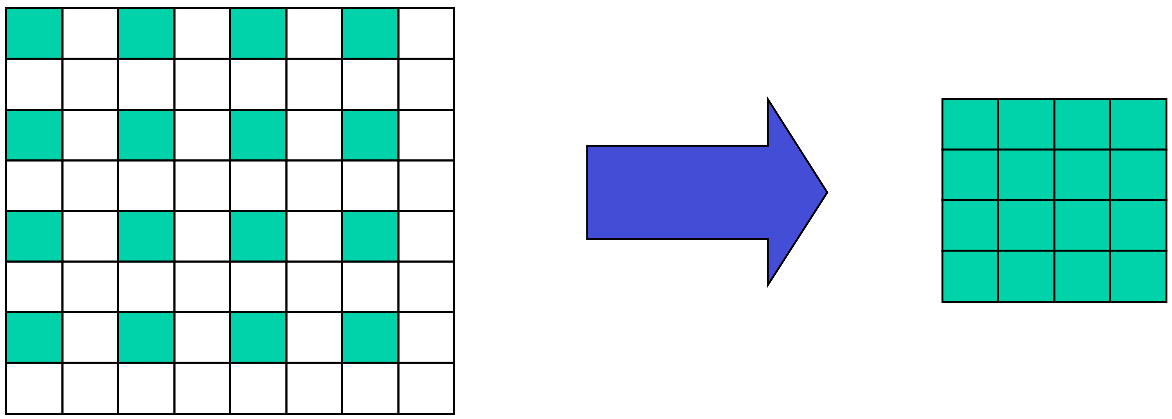
DownSample (reduce image size by half)



Upsample (double image size)



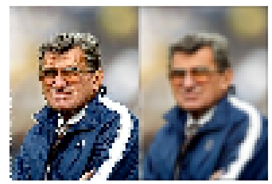
Downsample



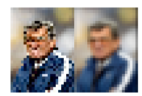
By the way: Subsampling is a bad idea unless you have previously blurred/smoothed the image! (because it leads to aliasing)

To Elaborate: Thumbnails

131x97



65x48



32x24

original image
262x195

downsampled (left)
vs. smoothed then
downsampled (right)

To Elaborate: Thumbnails

original image
262x195

downsampled (left)
vs. smoothed then
downsampled (right)

131x97

To Elaborate: Thumbnails

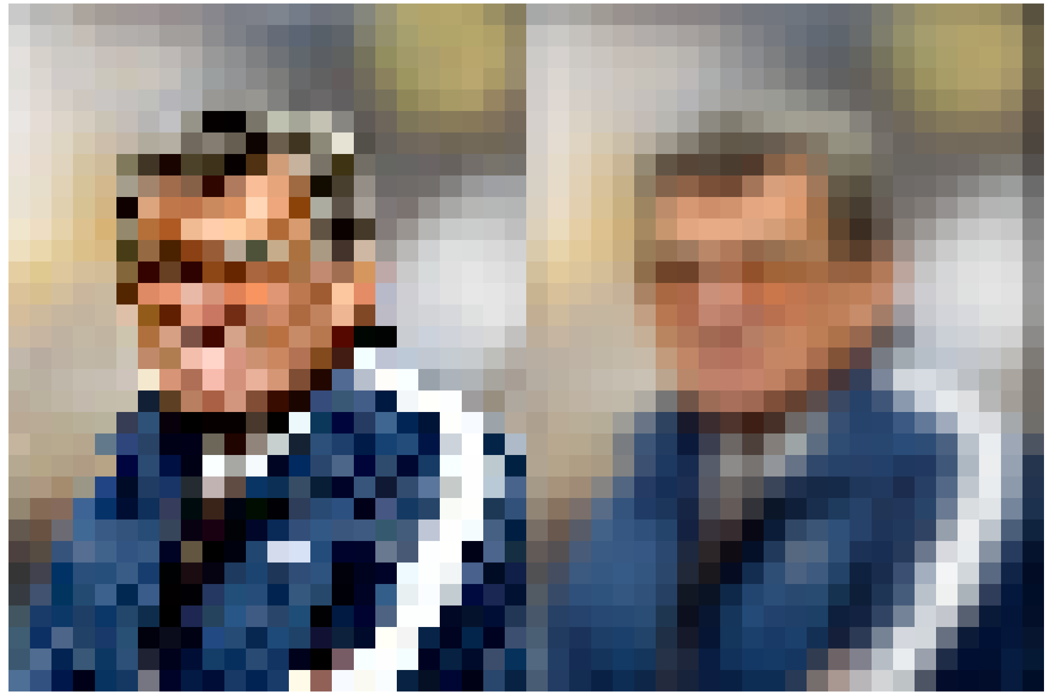
original image
262x195

downsampled (left)
vs. smoothed then
downsampled (right)

65x48

To Elaborate: Thumbnails

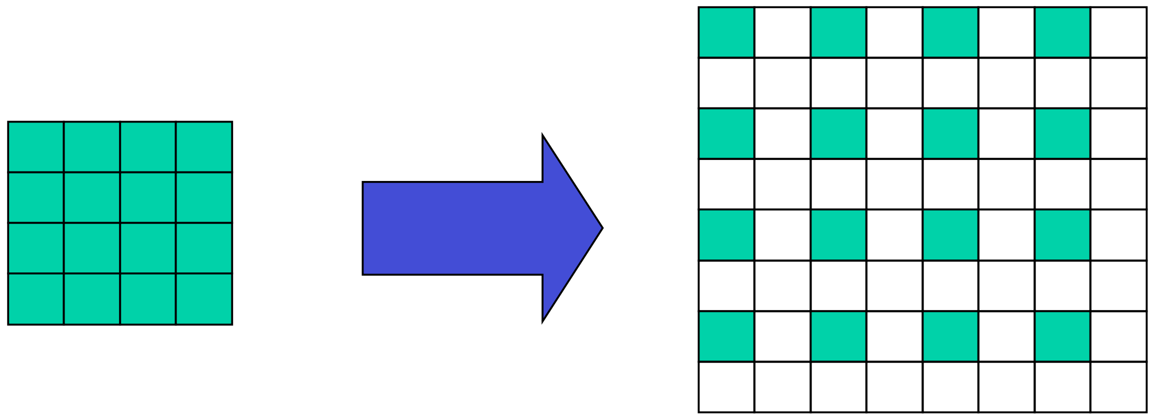
original image 262x195



downsampled (left) vs. smoothed then downsampled (right)

32x24

Upsample



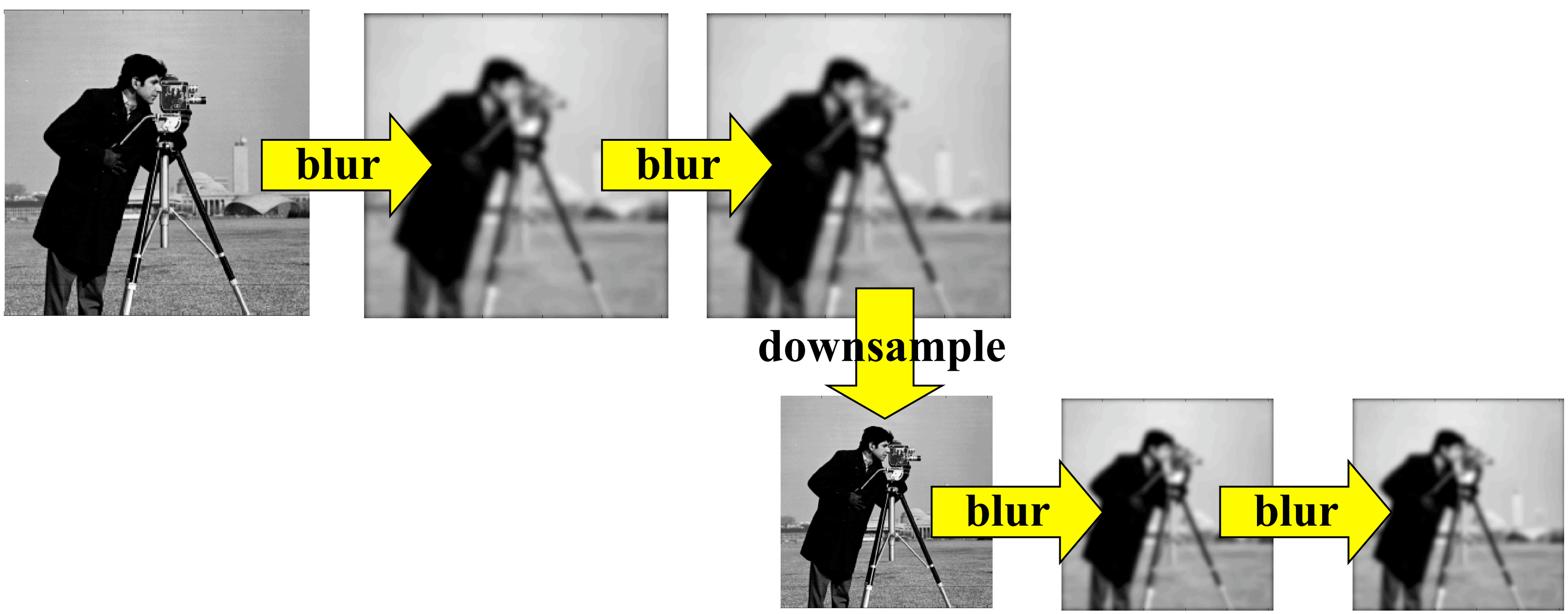
How to fill in the empty values?
Interpolation:

- initially set empty pixels to zero
- convolve upsampled image with Gaussian filter!
  e.g. 5x5 kernel with sigma = 1.
- **Must also multiply by 4. Explain why.**

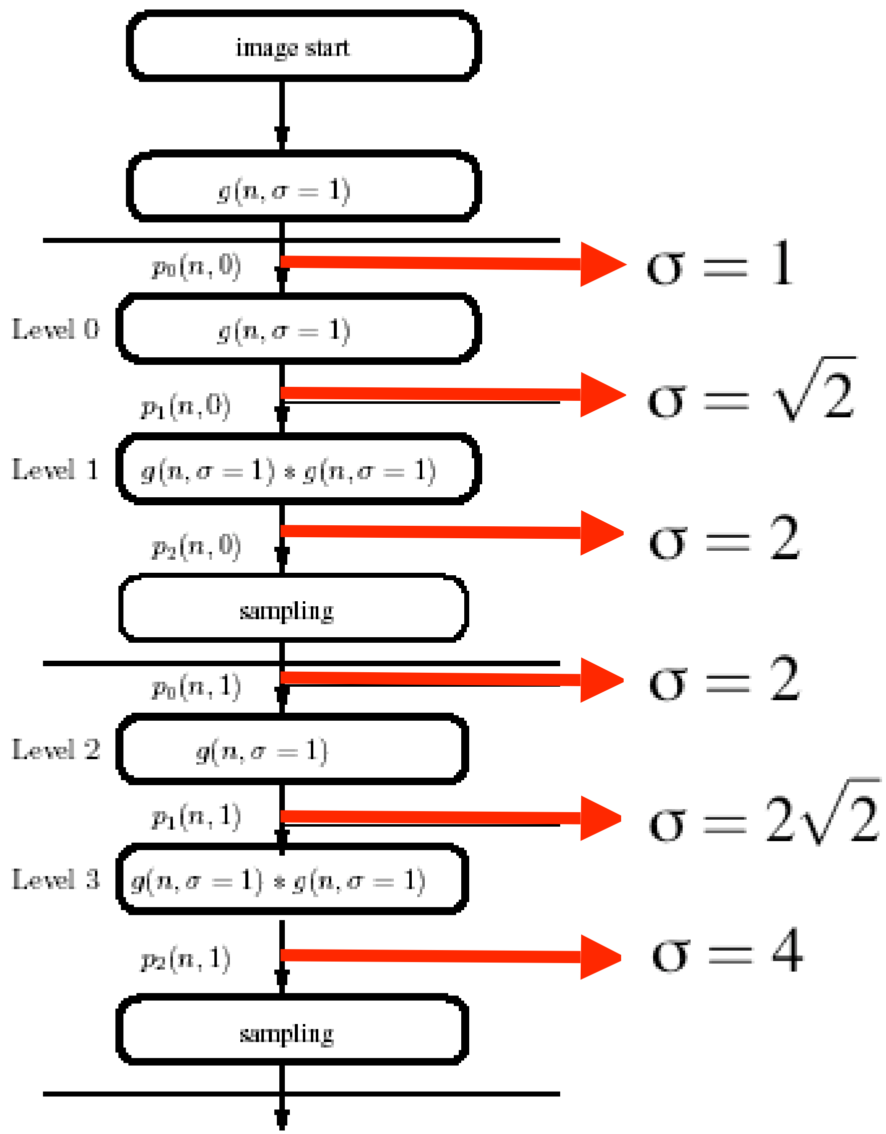
Specific Example

From Crowley et.al., "Fast Computation of Characteristic Scale using a Half-Octave Pyramid." *Proc International Workshop on Cognitive Vision (CogVis),* Zurich, Switzerland, 2002.

General idea: cascaded filtering using [1 4 6 4 1] kernel to generate a pyramid with two images per octave (power of 2 change in resolution). When we reach a full octave, downsample the image.



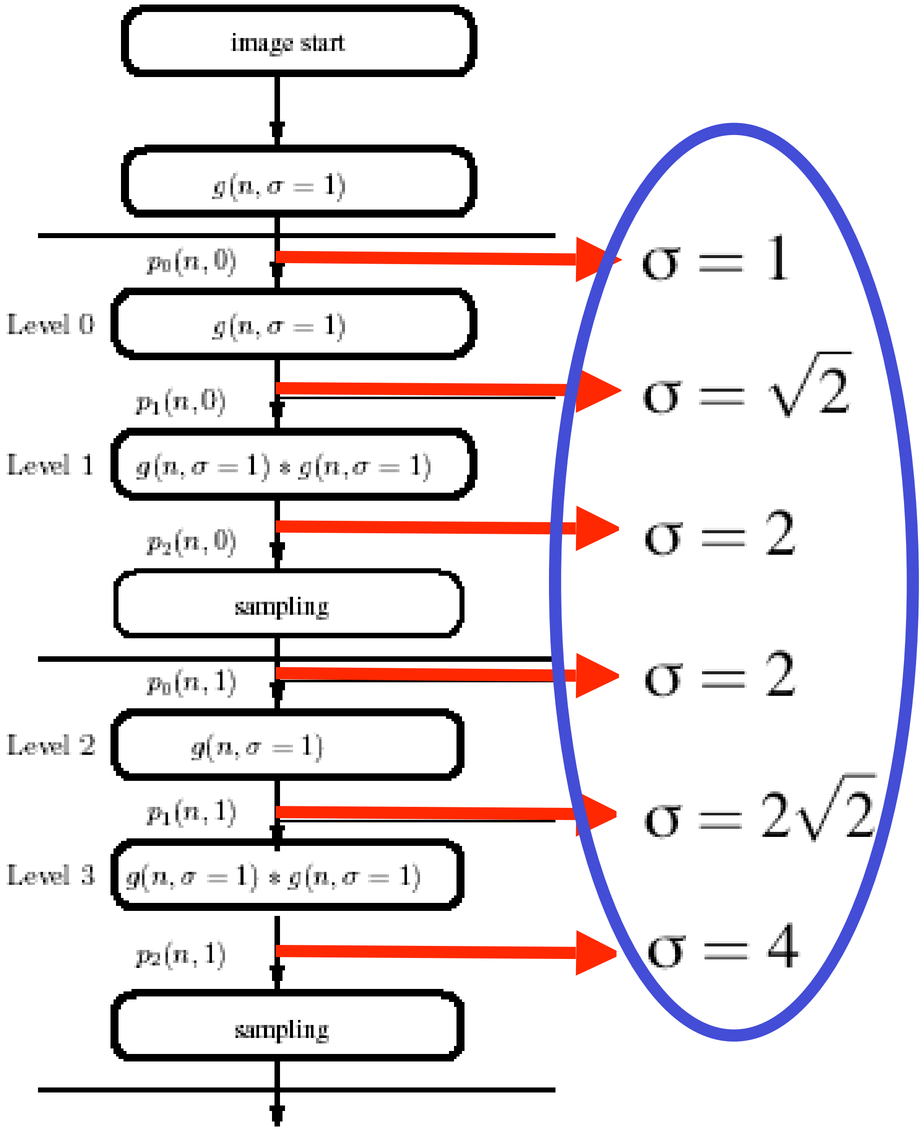
Effective Sigma at Each Level



Crowley etal.

Robert Collins
CSE486

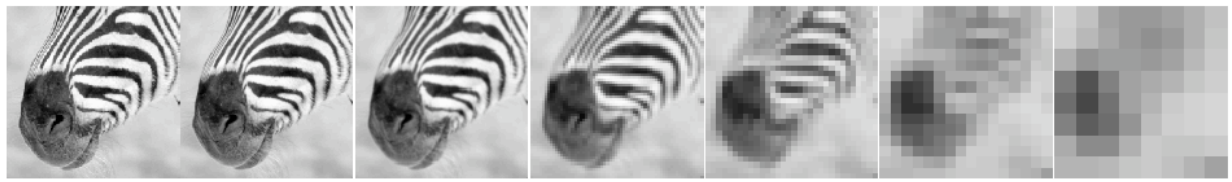
Effective Sigma at Each Level



CAN YOU EXPLAIN HOW THESE VALUES ARISE?

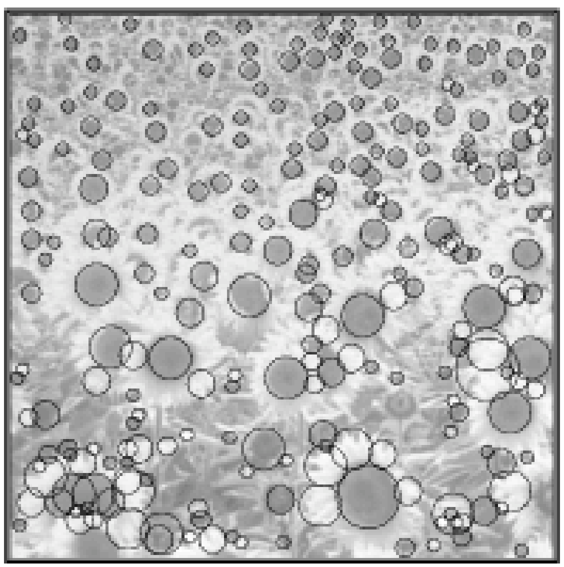
Concept: Scale Space

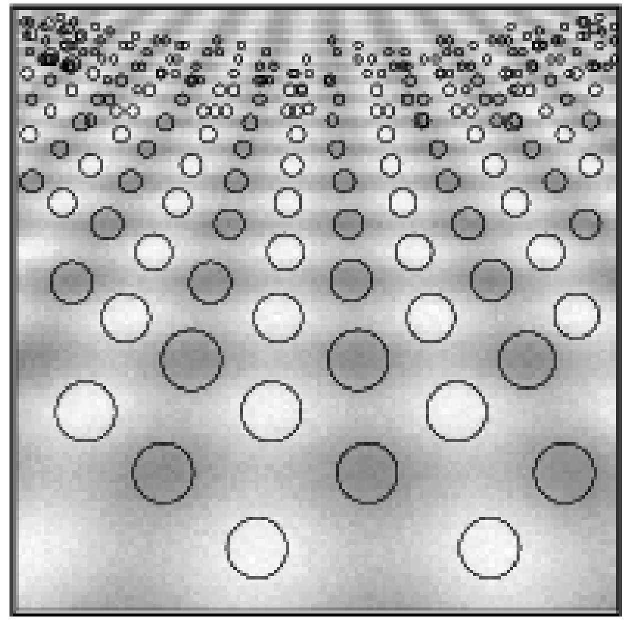
Basic idea: different scales are appropriate for describing different objects in the image, and we may not know the correct scale/size ahead of time.



512 256 128 64 32 16 8

Example: Detecting "Blobs" at Different Scales.





But first, we have to talk about detecting blobs at one scale...