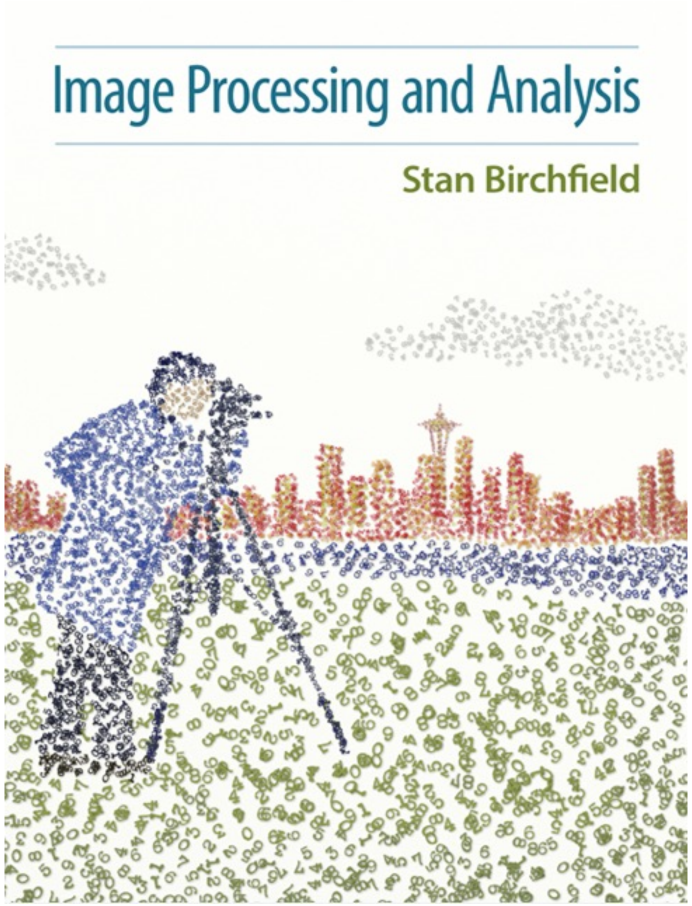
University of Stavanger

Prof. Kjersti Engan

ELE510 Image processing and computer vision

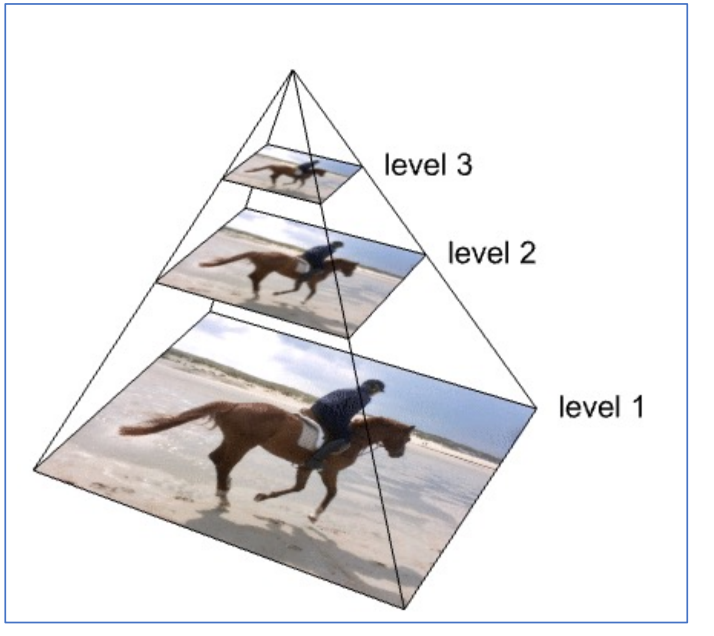
Edges and Features , (chap 7 Birchfield) 2020



Parts in course presentations is material from Cengage learning. It can be used for teaching, and it can be share it with students on access controlled web-sites (Canvas) for use in THIS course. **But it should not be copied or distributed further in any way** (by you or anybody).

Edges and features - Image scale and pyramids, Canny ++

- Three points from the topic:



1. Image pyramids and multiresolution – what do we mean by that?
2. How can multiresolution be useful in computer vision?
3. Some details on the Canny edge detection algorithm

(7.1) Image scale and pyramids

- Images look quite different at *different* ***scales:***
- A face that appears to occupy a 40 X 40 region in the original image will occupy only a 20 X 20 region in the downsampled image, a 10 X 10 region in the twice downsampled image, and so forth.
  - Because each successive image is smaller than its predecessor, stacking the images on top of one another yields the shape of a pyramid.
  - The sequence of images is known as an **image pyramid**.

Image pyramids – what?

- Lowpass pyramids, **Gaussian pyramids**: Smoothing the image with an appropriate smoothing filter (Gaussian) and then subsampling the smoothed image, usually by a factor of 2 along each coordinate direction.

- Bandpass pyramids, **Laplacian pyramids**: Difference between image and smoothed image (DoG) is approximation to LoG. Take smoothed image, downsample and repeat.

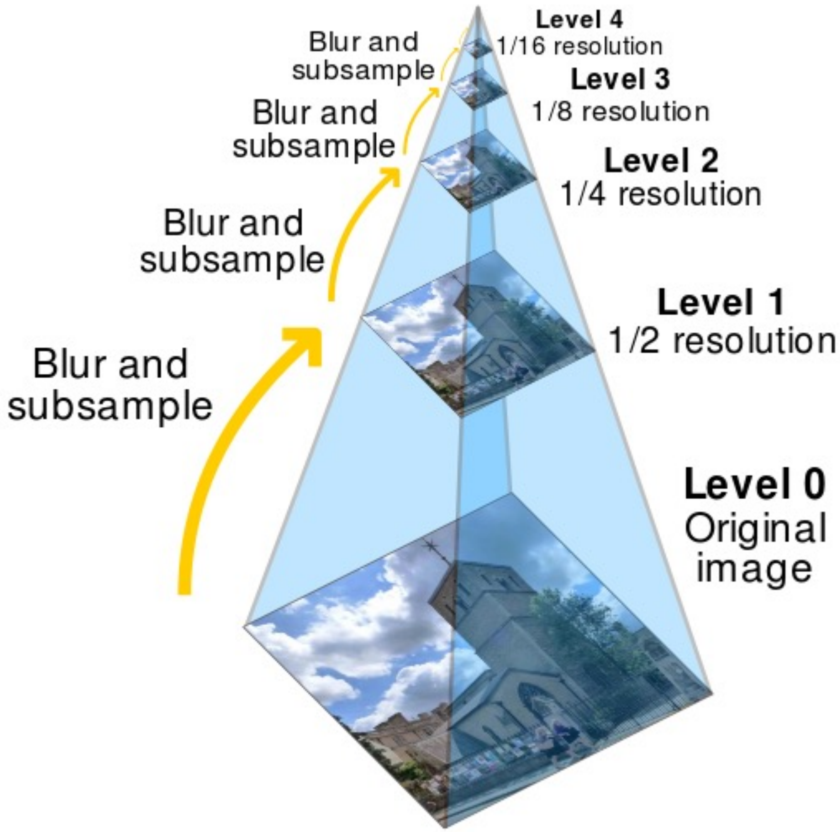


Image scale and pyramids – why?

- decompose images into information at multiple scales, to extract features/structures of interest and edges at different scales etc.
- Search over Scale,
- Spatial Search – coarse-to-fine matching,
- Feature Tracking.
- …

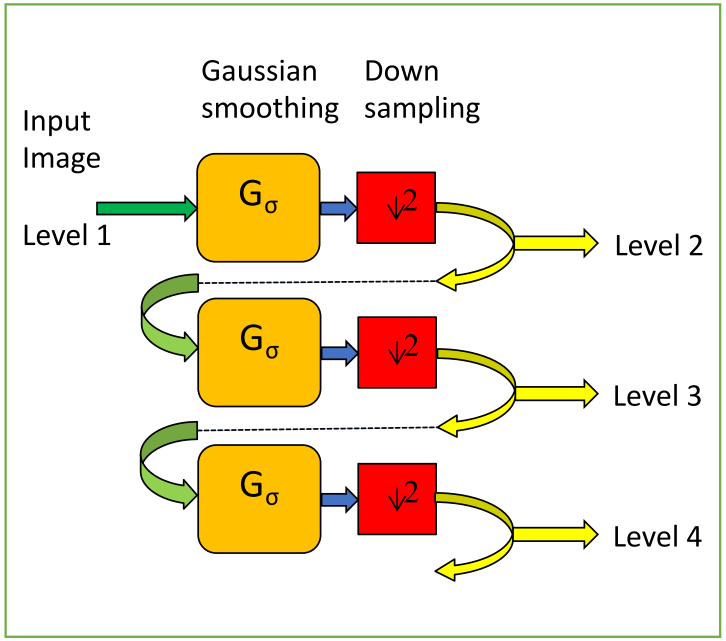
Gaussian Pyramide

- I(x,y) ↓ 2 means downsampling an image I(x,y) by a factor 2 in horizontal and vertical direction. Remember from Chap 3: I´(x,y)=I(2x,2y) downsample by 2, every other row and column are discarded. We have to smooth first to avoid aliasing!
- We do that by Gaussian filtering.

$$I^0(x,y) \equiv I(x,y)$$

$$I^{(i+1)}(x,y) = (I^{(i)}(x,y) * G_{auss_{\sigma^2}}(x,y)) \downarrow 2$$

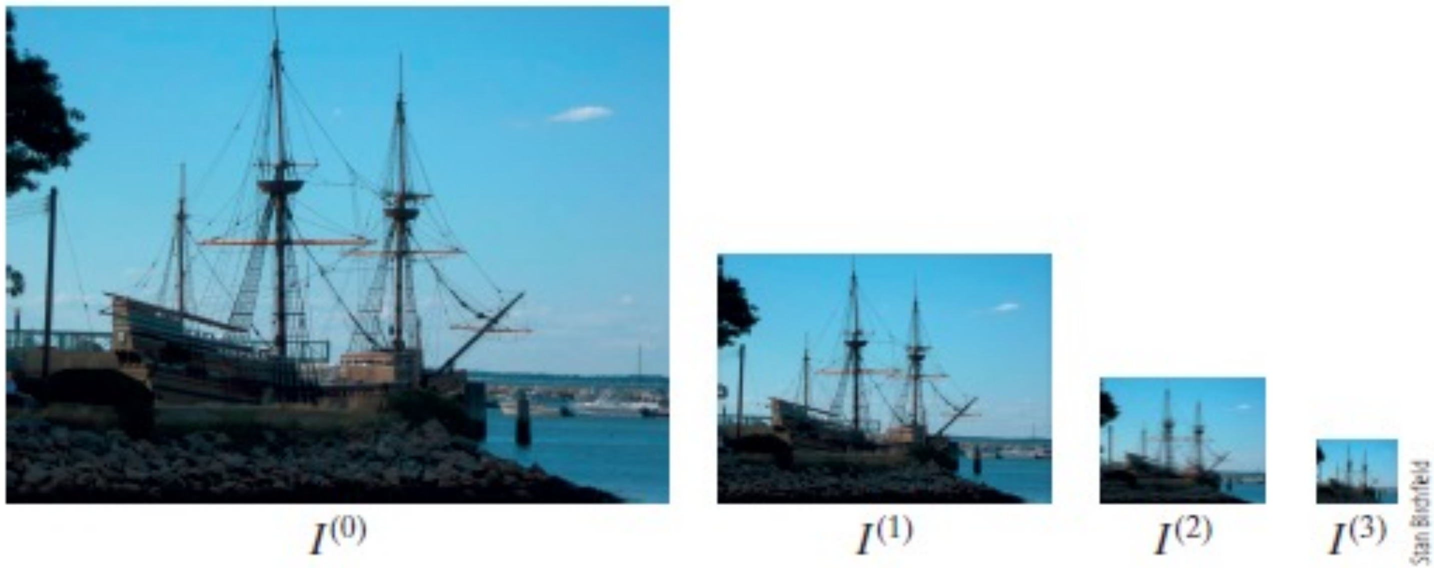
Downsampling by 2: every other row and column are discarded.



A 3x3 gaussian kernel works fine
σ^2 = 0.5 -> kernel: (1/4)*[1 2 1] is a good choice.

It satisfies the equal contribution property (each pixel in image contributes equal amount to the downsampled version)

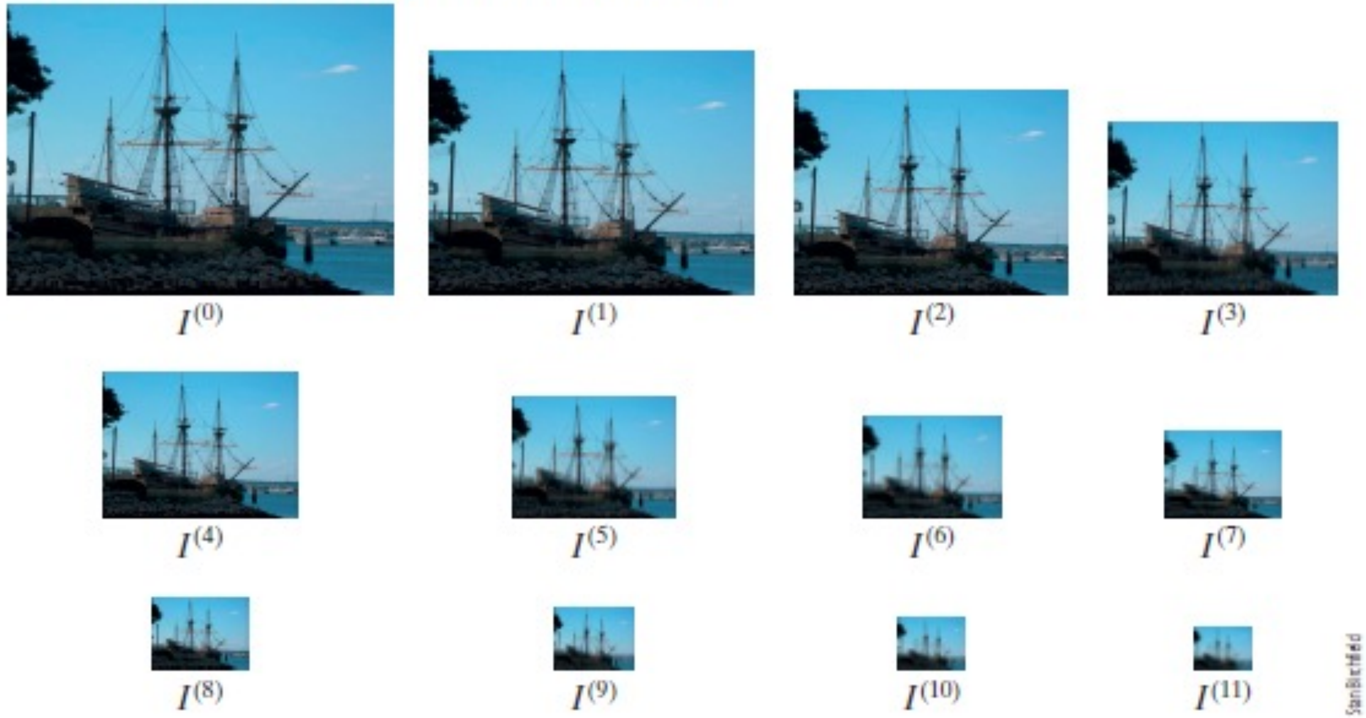
Figure 7.1 Four levels of a Gaussian pyramid, obtained with $\sigma^2 = 0.5$ and a downsampling factor of 2.



Each reduction by a factor of 2 is known as an octave

$\downarrow 2^{\frac{1}{n}} \implies$ n images pr. octave. Here n=1

Figure 7.2 Twelve levels of a Gaussian pyramid, obtained with $\sigma'^2 = \frac{1}{4}(0.5) = 0.125$ and a downsampling factor of $\sqrt[4]{2}$. Note that $I^{(4)}$ is half as large as $I^{(0)}$ in each direction, and that $I^{(8)}$ is half as large as $I^{(4)}$.



$\downarrow 2^{\frac{1}{n}} \Rightarrow$

Here n=4. i-e- 4 images pr. octave

We dont have to downsample by 2, but usually we do not want to do the interpolation necessary if downsampled by other than 2 in each direction, like here.
One way to avoid that is to have multiple layers of smoothing between each downsampling

Laplacian Pyramid

- Bandpass pyramids might be preferable for extracting features or interest points.
- Since the Laplacian is a bandpass operator, convolving the image with a Laplacian of Gaussian (LoG) kernel with increasing variance yields the **Laplacian pyramid**.

$$L^{(i+1)}(x, y) \equiv (I^{(0)}(x, y) \circledast LoG_{(i+1)\sigma^2}(x, y)) \downarrow (i+1)d$$

d= amount of downsampling

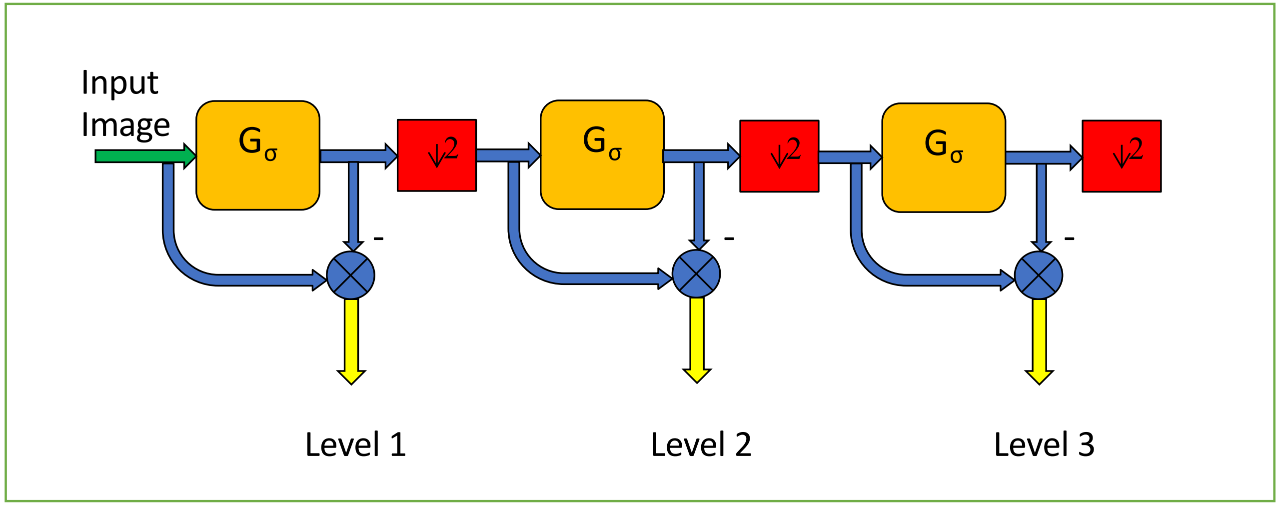
- This is inconvenient because it is defined by the original sized image at all levels!
- Remember LoG ≈ DoG . Laplacian pyramid usually implemented by the DoG (difference of Gaussian)

Laplacian Pyramid

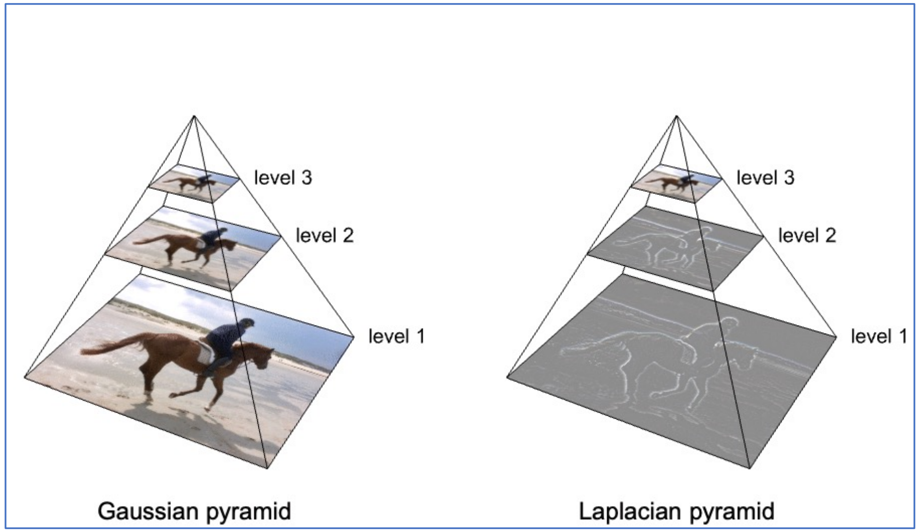
$$I^{(i+1)}_{temp}(x,y) = I^{(i)}(x,y) * G_{auss_{\sigma_i^2}}(x,y)$$

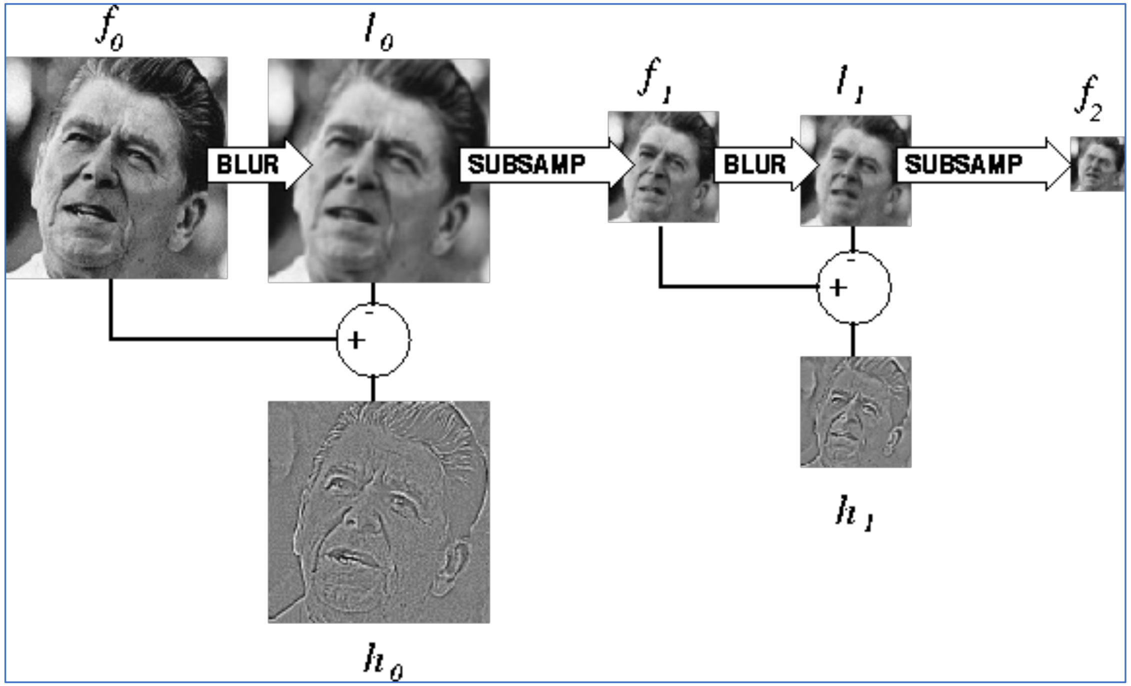
$$L^{(i+1)}(x,y) = I^{(i+1)}_{temp}(x,y) - I^{(i)}(x,y)$$

$$I^{(i+1)}(x,y) = (I^{(i+1)}_{temp}(x,y)) \downarrow d$$



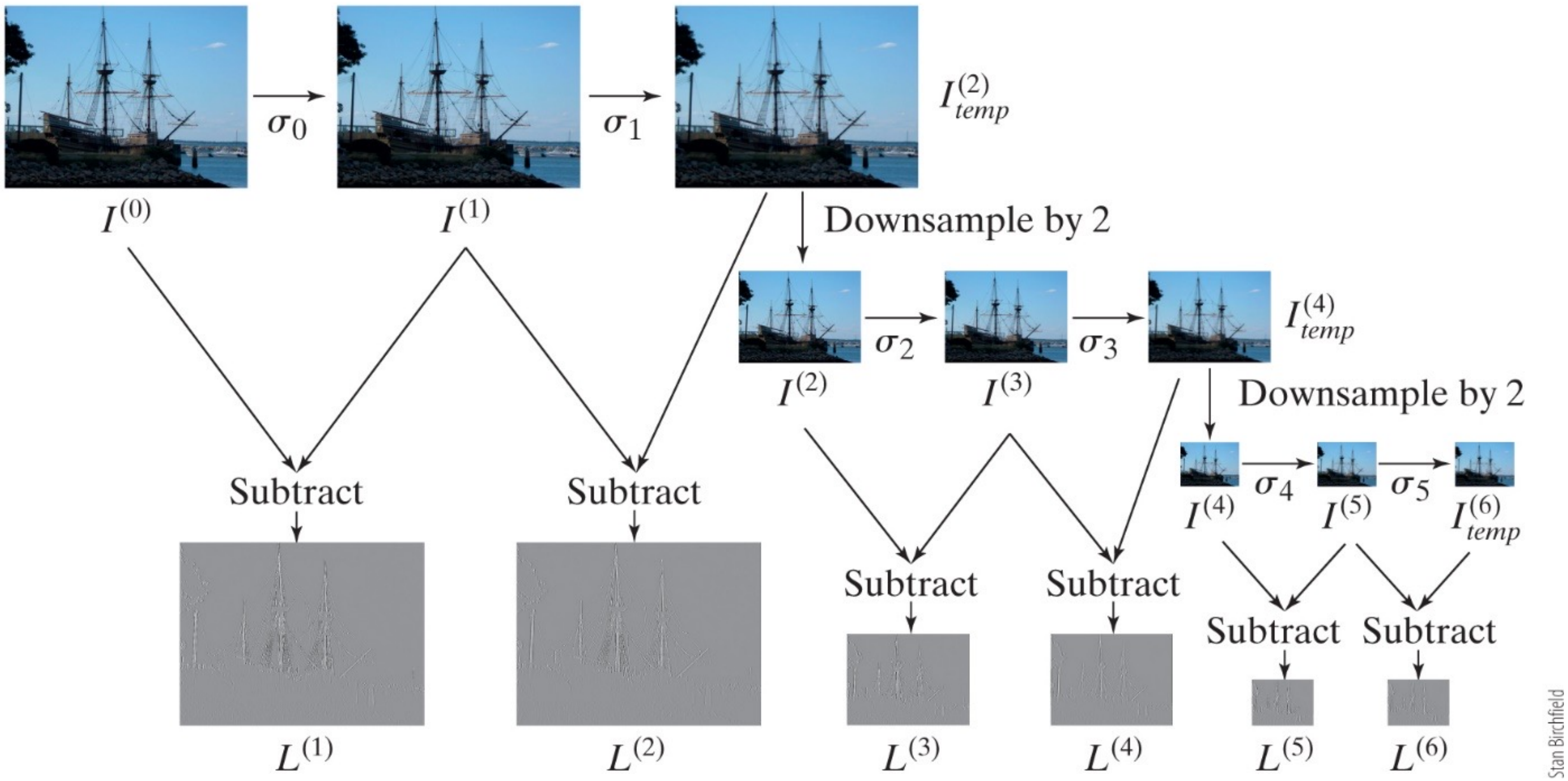
Laplacian pyramid (bandpass)

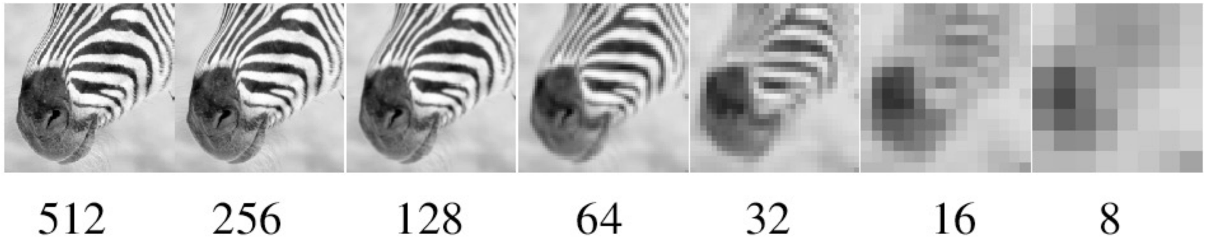




We have to low pass filter (smooth) before subsampling, if not -> introduces aliasing (sampling theorem)

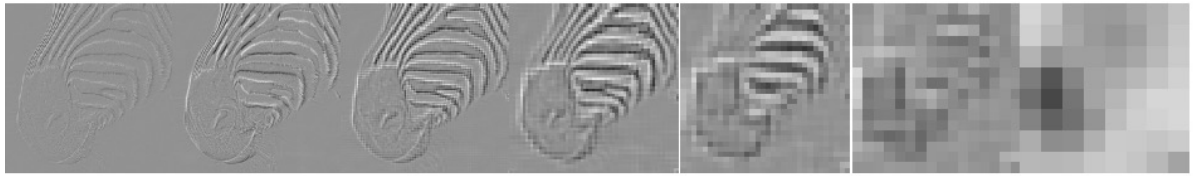
Figure 7.3 Laplacian pyramid with $n = 2$ images per octave. The images are successively convolved with a Gaussian, then downsampled at the end of each octave to produce something that closely resembles a Gaussian pyramid. Differences between successive Gaussian-smoothed images yield DoGs, which approximate LoGs. The initial variance is $\frac{1}{2}(0.5) = 0.25$, and the ratio between successive standard deviations is $\rho = \sqrt{2}$.



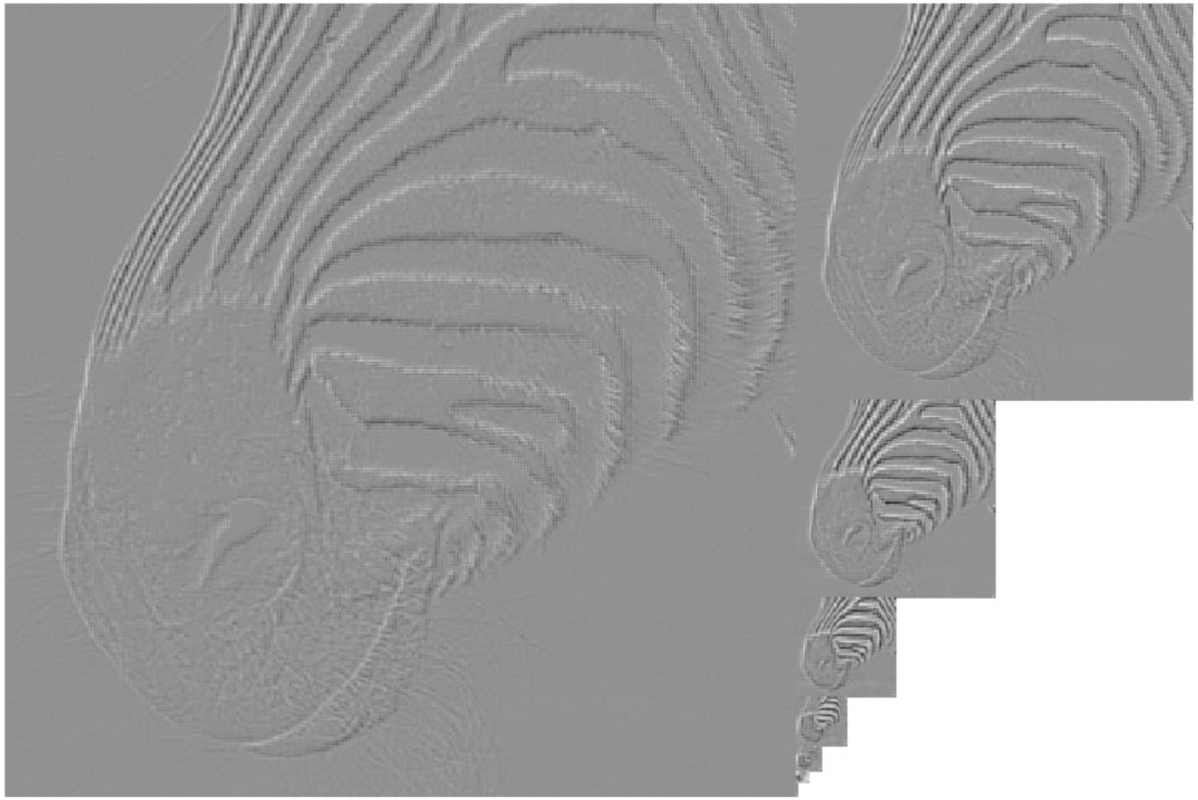


Example:

Gaussian Pyramid



512 256 128 64 32 16 8



Example:

Laplacian Pyramid

Scale Space

- Consider the family of images obtained by convolving the original image *I*(*x,y*) with Gaussian kernels having *continuously increasing* variance:

$$I(x, y, t) \equiv I(x, y) \circledast Gauss_t(x, y)$$

- Treating *x* and *y* as continuous values as well, the resulting 3D continuous volume is known as the **scale space** of the image, and $t = \sigma^2$ is the **scale-space parameter**.

Gaussian Scale Space

- For the scale space to be a meaningful representation of the image, it is important that several basic properties, called the **scale-space axioms**, should be satisfied.

- Among these axioms is the **causality criterion**, which ensures that the number of local extrema does not increase as we proceed to coarser levels of scale. I.e. maxima are flattened, minima are raised.

- The scale space is constructed using the Gaussian kernel and is known as the **Gaussian scale space**. The Gaussian kernel is the only kernel that guaranties the causality criterion.

Figure 7.4 The Gaussian scale space of an image consists of a continuous 3D volume in which each slice is an increasingly blurred version of the original image. Shown here are ten sample images from the scale space.

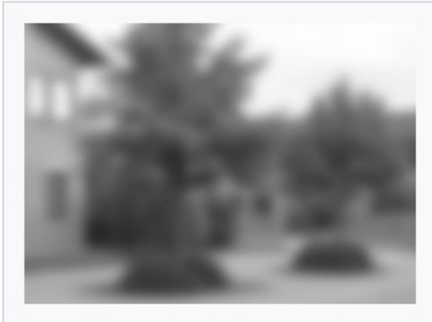
Example (source wikipedia)

The scale parameter, t, is the variance of the 2D Gaussian filter

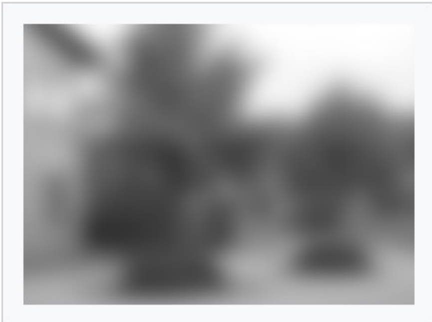
Scale-space representation $L(x, y; t)$ at scale $t = 0$, corresponding to the original image f

Scale-space representation $L(x, y; t)$ at scale $t = 1$

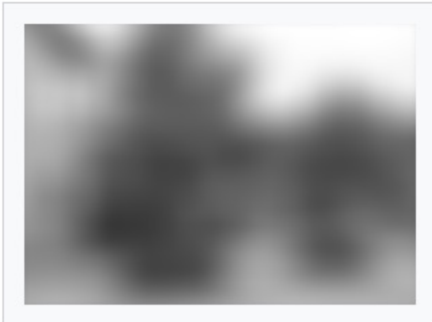
Scale-space representation $L(x, y; t)$ at scale $t = 4$



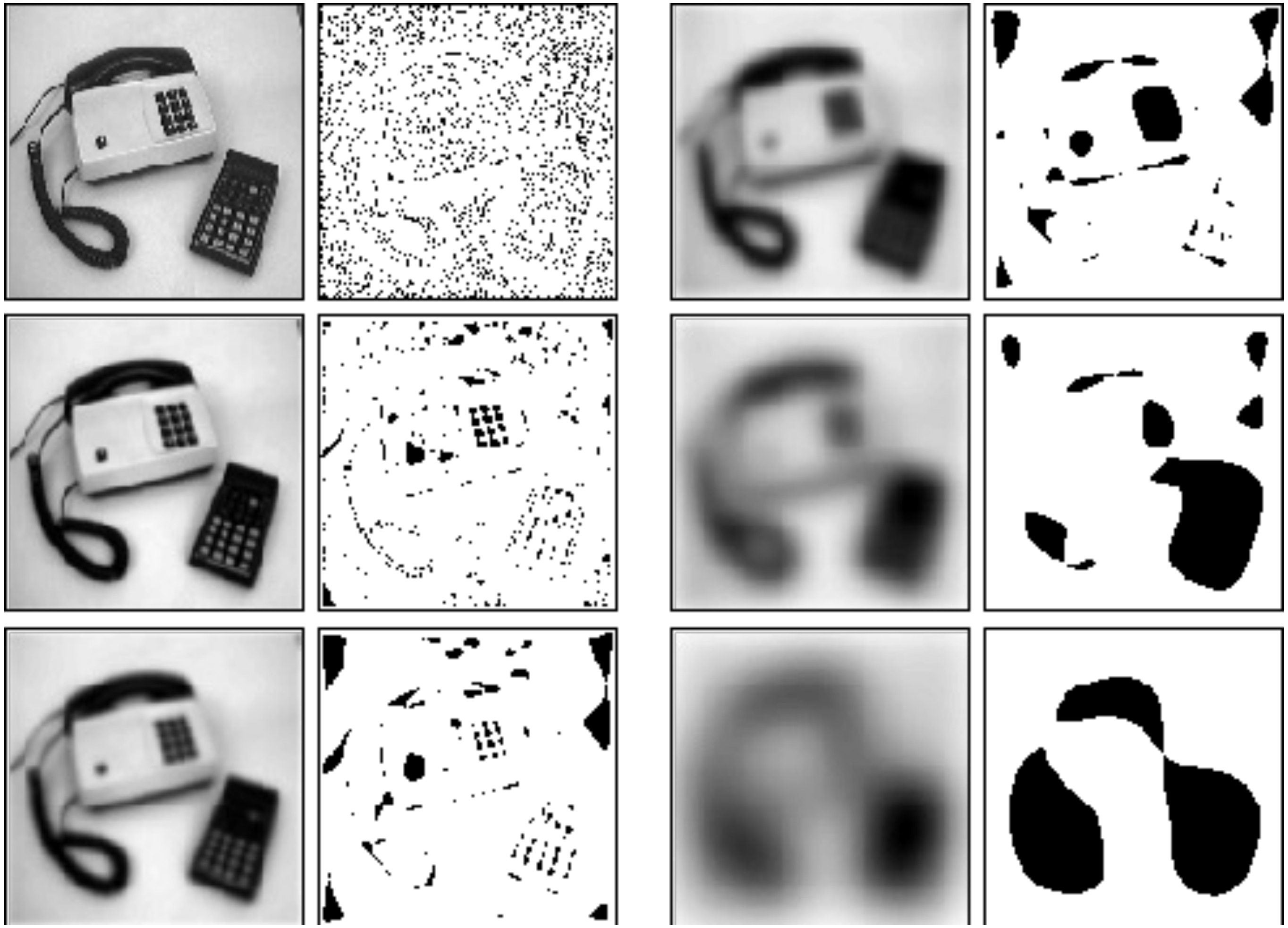
Scale-space representation $L(x, y; t)$ at scale $t = 16$



Scale-space representation $L(x, y; t)$ at scale $t = 64$

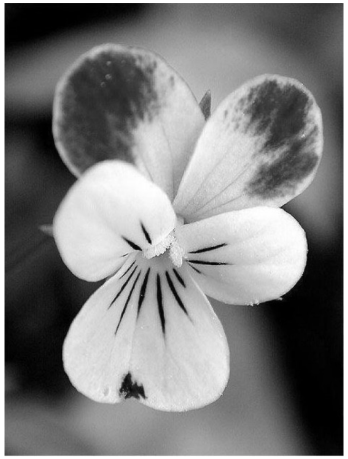


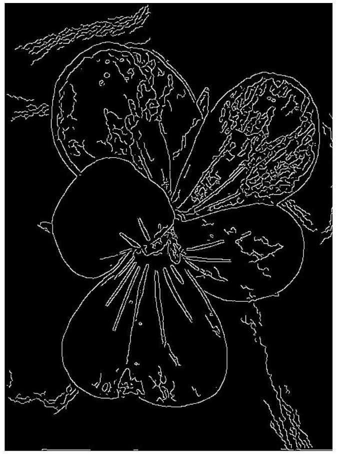
Scale-space representation $L(x, y; t)$ at scale $t = 256$

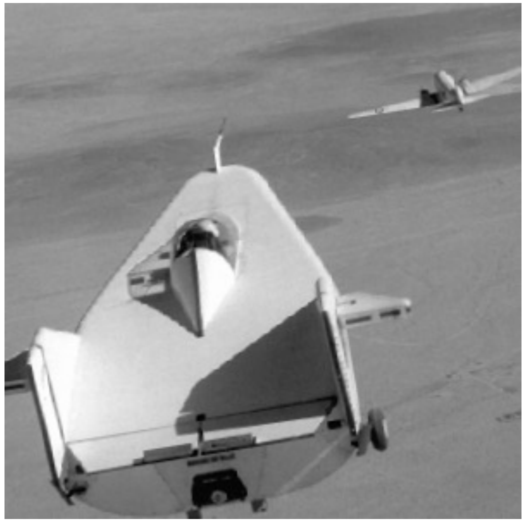


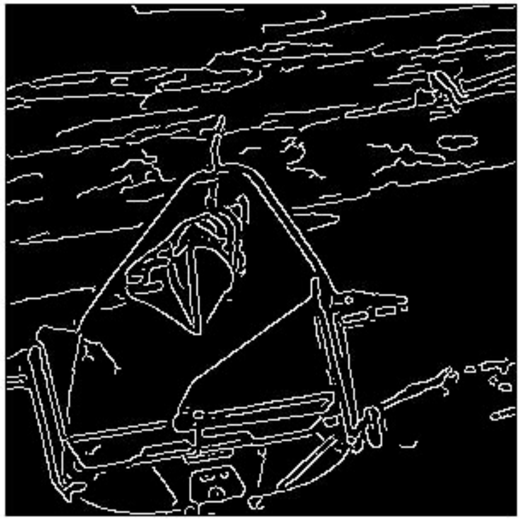
Different levels in the scale-space representation of a two-dimensional image at scale levels $t = 0$, 2, 8, 32, 128 and 512 together with grey-level blobs indicating local minima at each scale.

(7.2) Edges and edge detection - revisited









We have already looked at derivatives and gradients.
Defining Sobel, Prewitt and LoG filters -> finds high frequencies and gradients.

How to define the edge? (we looked at Canny and zero crossing of LoG)

Edge Detection

- **Intensity edges** are pixels in the image where the intensity (or graylevel) function changes rapidly.
- **Step edge**: occurs when a light region is adjacent to a dark region.
- **Line edge:** occurs when a thin light (or dark) object, such as a wire, is in front of a dark (or light) background.
- **Roof edge**: the change is not in the lightness itself but rather the derivative of the lightness.
- **Ramp edge:** occurs when the lightness changes slowly across a region.

Figure 7.6 Intensity edges capture a rich representation of the scene. The scenes and objects in these line drawings are, with little difficulty, recognizable by the average human viewer. From Walther et al. [2011]. For the original images, turn to Figure 7.7.

D. B. Walther, B. Chai, E. Caddigan, D. M. Beck, and L. Fei-Fei, "Simple line drawings suffice for functional MRI decoding of natural scene categories," Proceedings of the National Academy of Sciences (PNAS), 108(23):9661-9666, 2011.

Canny Edge detector (revisited)

- The **Canny edge detector** is a classic algorithm for detecting intensity edges in a grayscale image that relies on the gradient magnitude.

- The algorithm involves three steps:
  - First, the gradient of the image is computed, including the magnitude and phase. Can be found by different methods (Sobel ..)
  - Second, in **non-maximum suppression**, any pixel is set to zero whose gradient magnitude is not a local maximum in the direction of the gradient (ridge tracking).

    Ridge pixel > T2 -> strong edge pixel

    T1<Ridge pixel<T2 -> weak edge pixel
  - Finally, edge linking is performed to discard pixels without much support. include weak edge pixels that are 8- connected to strong edge pixels

- The result is a binary image whose edge pixels along one-pixel-thick boundaries are ON, while all other pixels are OFF.

Example: used in Canny – edge detection

1. Image is smoothed by Gaussian filter to reduce noise
2. Local gradient and edge direction are found (by sobel / prewitt)
3. The ridges of the gradient image is tracked, set to zero all pixels not on the ridge top -> thin line   (non –maximum supression)
4. Thereafter, hysteresis threshold

   Ridge pixel > T2 -> strong edge pixel

   T1<Ridge pixel<T2 -> weak edge pixel
5. Edge linking -> include weak edge pixels that are 8- connected to strong edge pixels.

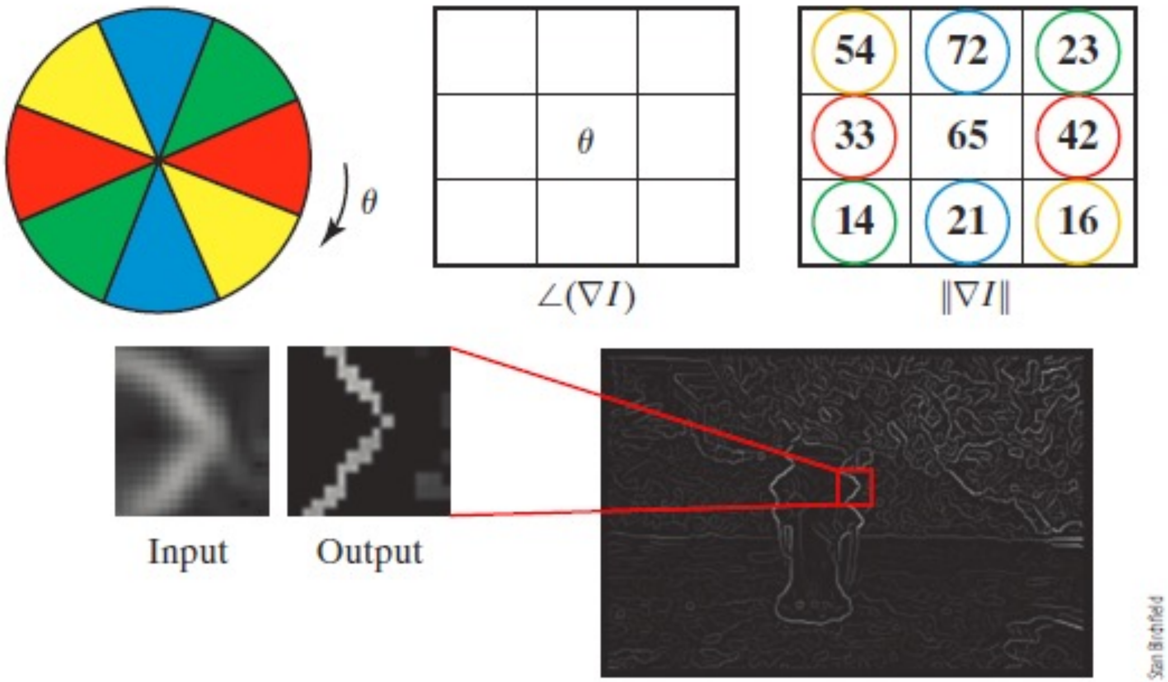


Figure 7.10 Non-maximum suppression. The gradient direction (or phase) θ is quantized into one of four values, shown by the colored wedges of the circle. The quantized phase governs which of the two neighbors to compare with the pixel. If the gradient magnitude of the pixel is not at least as great as both neighbors, then it is set to zero. This has the effect of thinning the edges, as shown in the inset.

How to select the filter to compute the gradient? If Gaussian derivative is used, sigma is the parameter. A large sigma yields a better signal-to-noise ratio (SNR), but a smaller sigma yields a more accurate location for the edge. This dilemma is known as the **localization-detection tradeoff.**

To derive the optimal step detector, two criteria are specified:

The detector should yield low false positive and false negative rates.

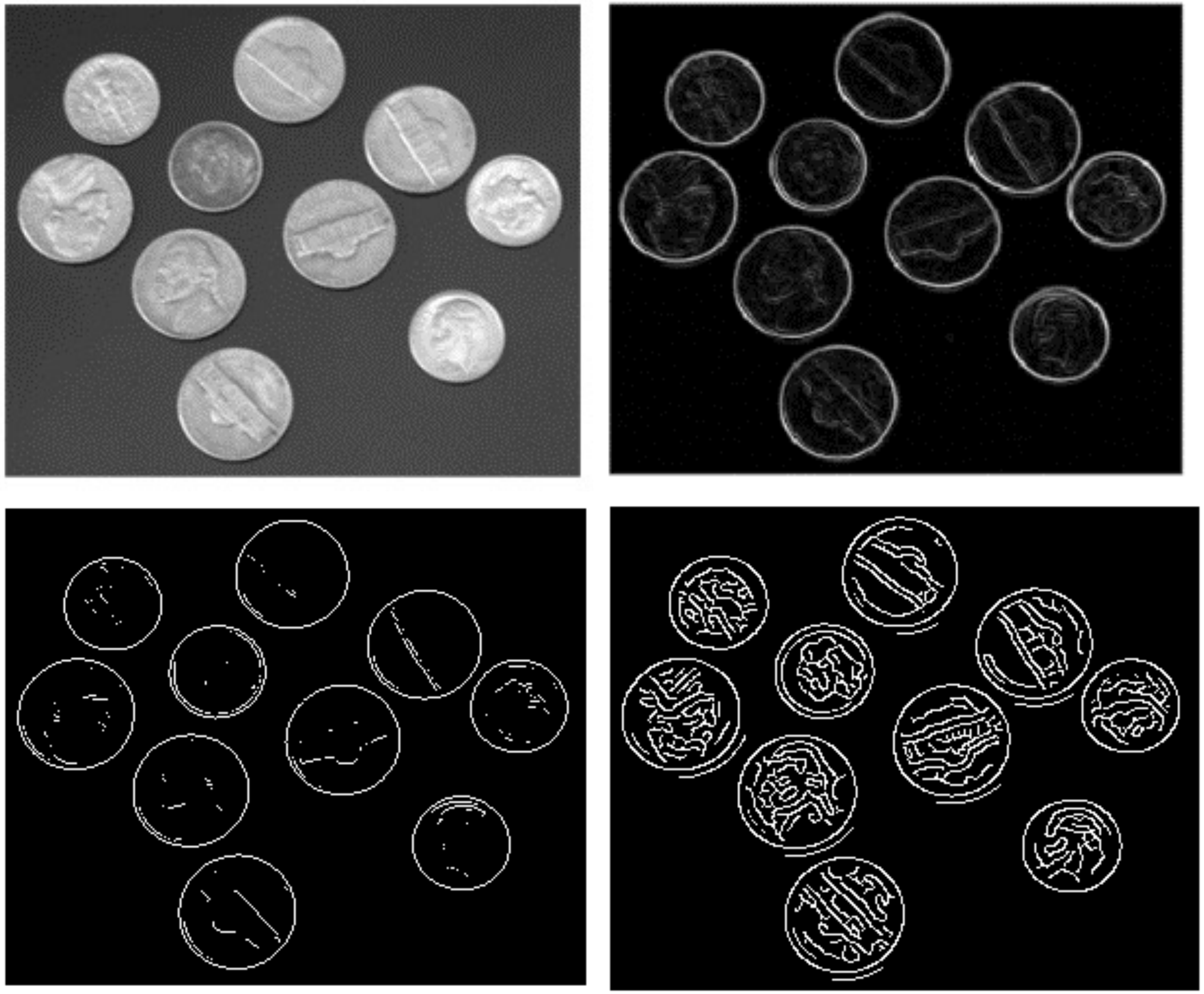
The detected edge should be close to the true edge (that is, good *localization*).

Original image,

Sobel gradient

sobel edge map,

Canny edge map



Edges and features - Image pyramids, Canny ++

- Three points from the topic:

1. Image pyramids and multiresolution – what do we mean by that?
   - ✓ Smoothed and downsampled repeatedly gives gaussian pyramids. DoG before downsampling makes Laplacian pyramids.
2. How can multiresolution be useful in computer vision?
   - ✓ For example find features /points shapes that should be invariant to scaling
3. Some details on the Canny edge detection algorithm
   - ✓ Non-maximum supression for thinning edges, edge linking, gives good edges.