Imperial College London

Digital Image Procesing

Spatial Filters in Image Processing

DR TANIA STATHAKI

READER (ASSOCIATE PROFFESOR) IN SIGNAL PROCESSING

IMPERIAL COLLEGE LONDON

Image averaging in the case of many realizations
White noise

- We have L different images which represent the same image $f(x,y)$ corrupted by **white noise** as follows:

$$g_i(x,y) = f(x,y) + n_i(x,y), i = 1, \dots, L$$

- Noise realizations are zero mean and have the same variance.
  - In discrete time/space, white noise is a discrete signal whose samples are regarded as a sequence of **uncorrelated** random variables with **zero mean** and **finite variance**.
  - Depending on the context, one may also require that the samples be **independent** and have **identical probability distribution** (in other words independent and identically distributed **(iid)** random variables are the simplest representation of white noise).
  - In particular, if each sample has a **normal distribution** with zero mean, the signal is said to be additive **white Gaussian noise**.

Image averaging in the case of many realizations

Statistics

- The following statistical relationships hold:
  - The **mean** values of all noise realizations are zero.

$$E\{n_i(x,y)\} = 0$$

  - The **covariance** between any two noise realizations is zero.

$$\mathrm{Cov}(n_i, n_i) = E\left\{\left(n_i(x,y) - m_{n_i}(x,y)\right)\left(n_j(x,y) - m_{n_j}(x,y)\right)\right\}$$
$$= E\{(n_i(x,y) - 0)(n_j(x,y) - 0)\} = E\{n_i(x,y)n_j(x,y)\} \Rightarrow$$
$$\mathrm{Cov}(n_i,\ n_i) = \begin{cases} \sigma_{n_i}^2 = \sigma_n^2 & i = j \\ 0 & i \neq j \end{cases}$$

  - The **autocorrelation** function of the noise is non-zero only at $(0,0)$ lag.

$$R_{n_i}[k,l] = E\{n_i(x,y)n_i(x+k,y+l)\} = \begin{cases} \sigma_n^2 & k = l = 0 \\ 0 & k \neq l \end{cases} = \sigma_n^2 \delta[k,l]$$

  - Note that the noise process is **stationary**, i.e., the function $E\{n_i(x,y)n_i(x+k,y+l)\}$ depends only on the pair (k,l).
  - Furthermore, the noise process is **ergodic**, i.e., the statistics along different realizations are the same as the statistics along one realization.

Imperial College London

Image averaging in the case of many realizations

- We define a new image which is the average

$$\bar{g}(x,y) = \frac{1}{L}\sum_{i=1}^{L} g_i(x,y) = f(x,y) + \frac{1}{L}\sum_{i=1}^{L} n_i(x,y)$$

- By doing this, it is like we create a new noisy version, where the noise is the average of all the original noise realizations $n(x,y) = \frac{1}{L}\sum_{i=1}^{L} n_i(x,y)$.
- Notice that the average is calculated across realizations.
- **Problem:** Find the mean and variance of the new noise realization $n(x,y)$.
  - $\boldsymbol{E\{n(x,y)\}} = E\left\{\frac{1}{L}\sum_{i=1}^{L} n_i(x,y)\right\} = \frac{1}{L} E\{\sum_{i=1}^{L} n_i(x,y)\} = \frac{1}{L}\ \sum_{i=1}^{L} E\{n_i(x,y)\} = \mathbf{0}$
  - $\mathbf{Var}\big(\boldsymbol{n(x,y)}\big) = E\{n^2(x,y)\} - E\{n(x,y)\}^2 = E\{n^2(x,y)\} - 0 = E\{n^2(x,y)\} =$

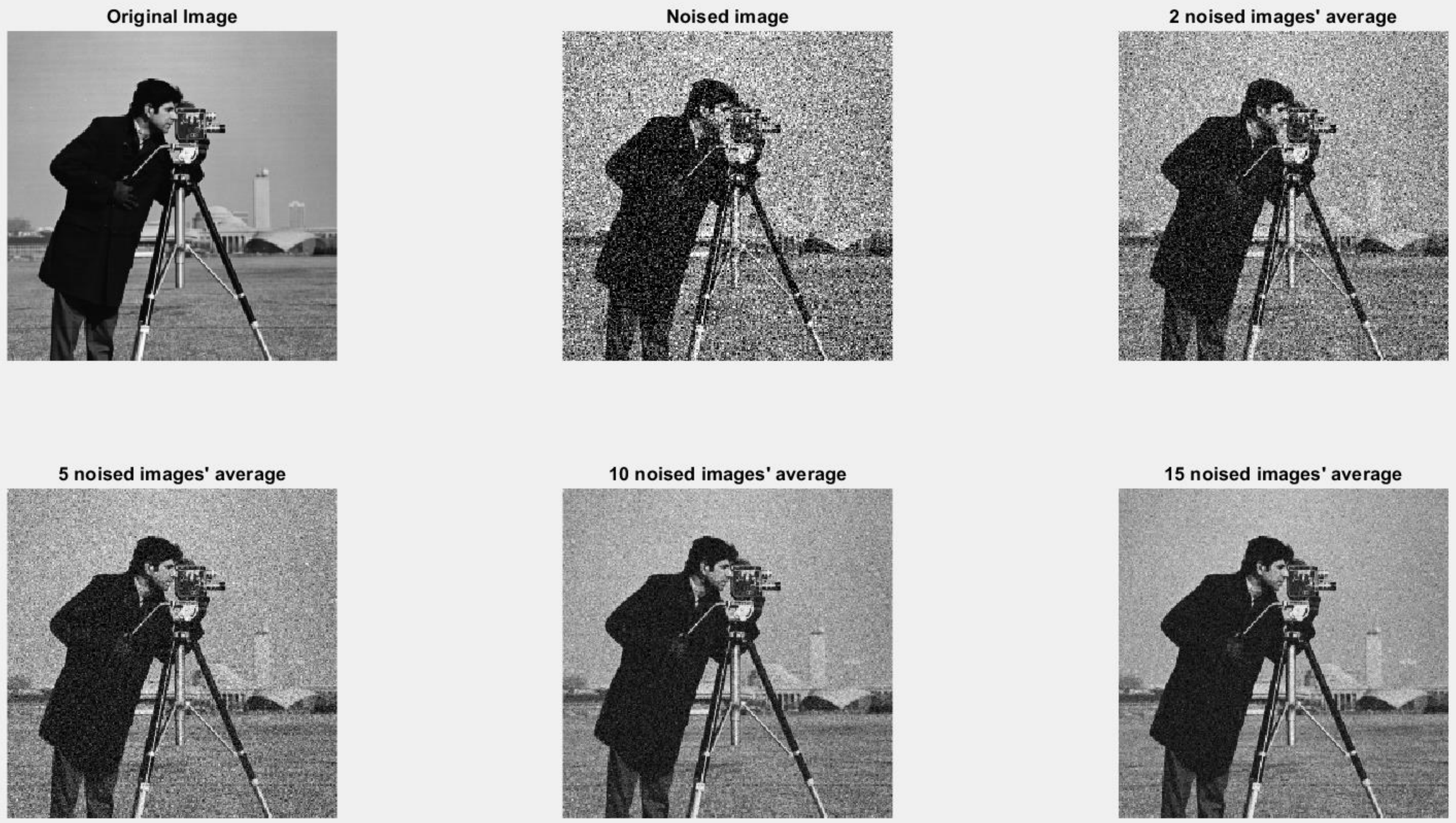
$$= E\left\{\left(\frac{1}{L}\sum_{i=1}^{L} n_i(x,y)\right)^2\right\} = \frac{1}{L^2} E\left\{\sum_{i=1}^{L} n_i^2(x,y) + \sum_{\substack{i=1 \\ i\neq j}}^{L}\sum_{j=1}^{L} n_i(x,y)\, n_j(x,y)\right\} =$$

$$\frac{1}{L^2}\sum_{i=1}^{L} E\{n_i^2(x,y)\} + \sum_{\substack{i=1 \\ i\neq j}}^{L}\sum_{j=1}^{L} E\{n_i(x,y)n_j(x,y)\} = \frac{1}{L^2}\sum_{i=1}^{L} E\{n_i^2(x,y)\} + 0 = \frac{1}{L^2} L\sigma_n^2 = \boldsymbol{\frac{1}{L}\ \sigma_n^2}$$

- We observe that for large L the variance of $n(x,y)$ becomes very small.

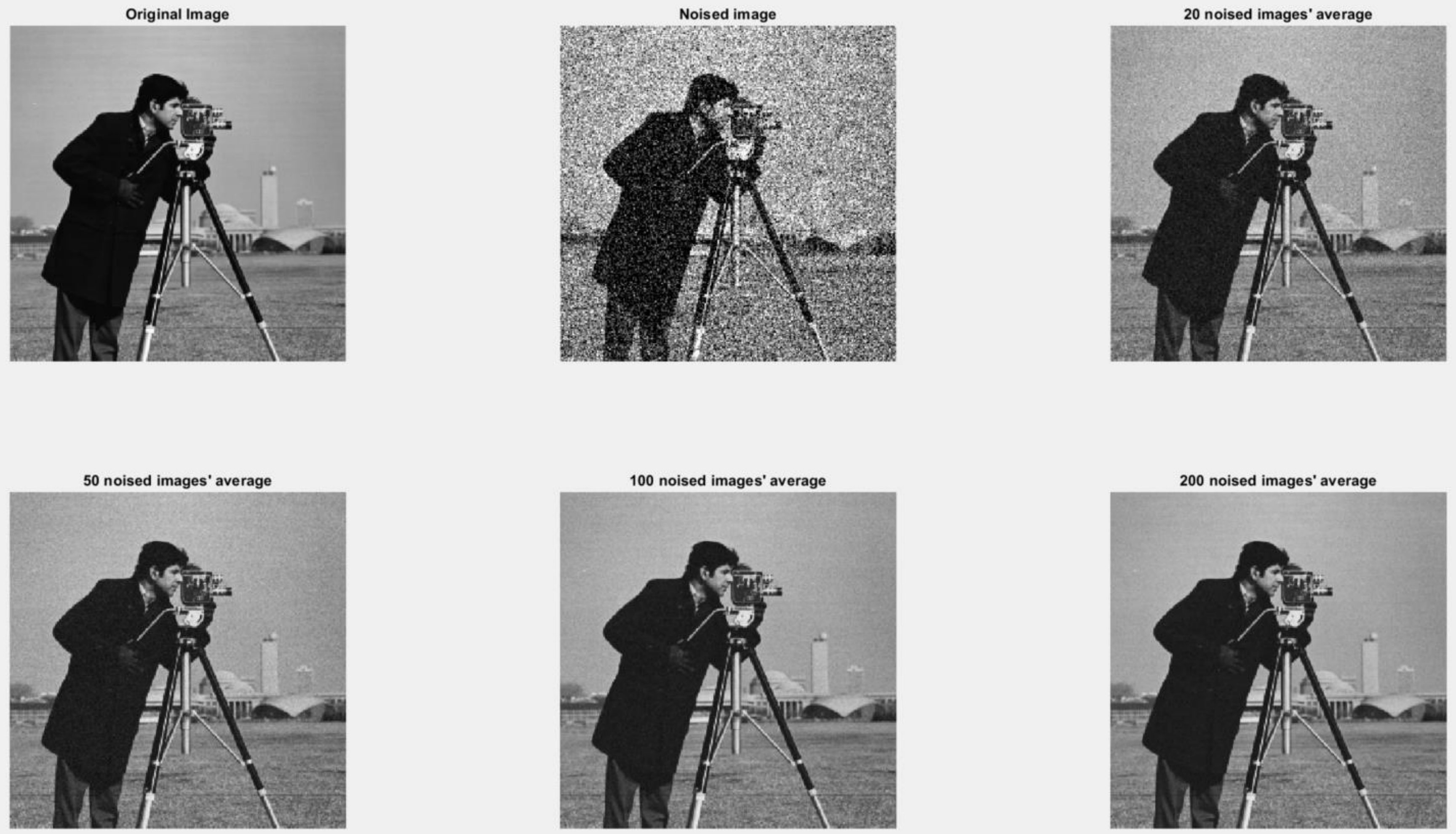
Example: Image averaging experiment

Junfeng Cheng 2020



Example: Image averaging experiment

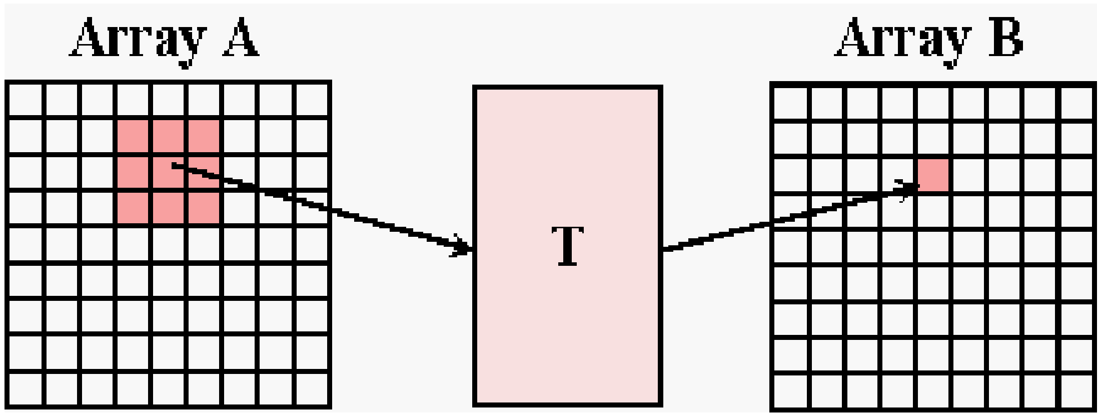
Junfeng Cheng 2020

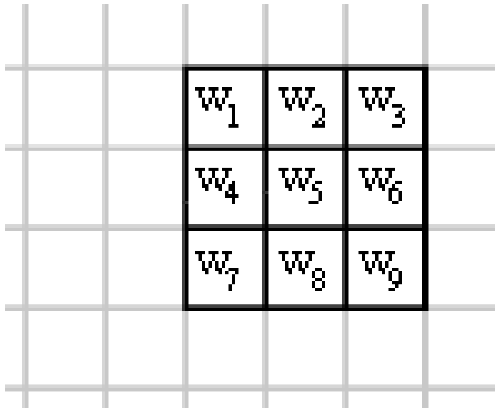

Original Image
Noised image
20 noised images' average
50 noised images' average
100 noised images' average
200 noised images' average

Imperial College London

Spatial filters: Definition of filter kernels

- Linear spatial filtering consists of convolving an image with a **filter kernel**.
- The figure below illustrates the mechanics of linear spatial filtering using a 3×3 kernel $w(x, y)$.
- At any point (x, y) in the image, the response, $g(x, y)$, of the filter is the sum of products of the kernel coefficients $w(k, l)$ and the image pixels $f(x, y)$ encompassed by the kernel $\Rightarrow g(x, y) = \sum_{k=-K}^{K} \sum_{l=-L}^{L} w(k, l) f(x + k, y + l)$
- As coordinates x and y are varied, the centre of the kernel moves from pixel to pixel generating the filtered image in the process.
- Observe that the centre coefficient of the kernel, $w(0,0)$, aligns with the pixel at location (x, y).

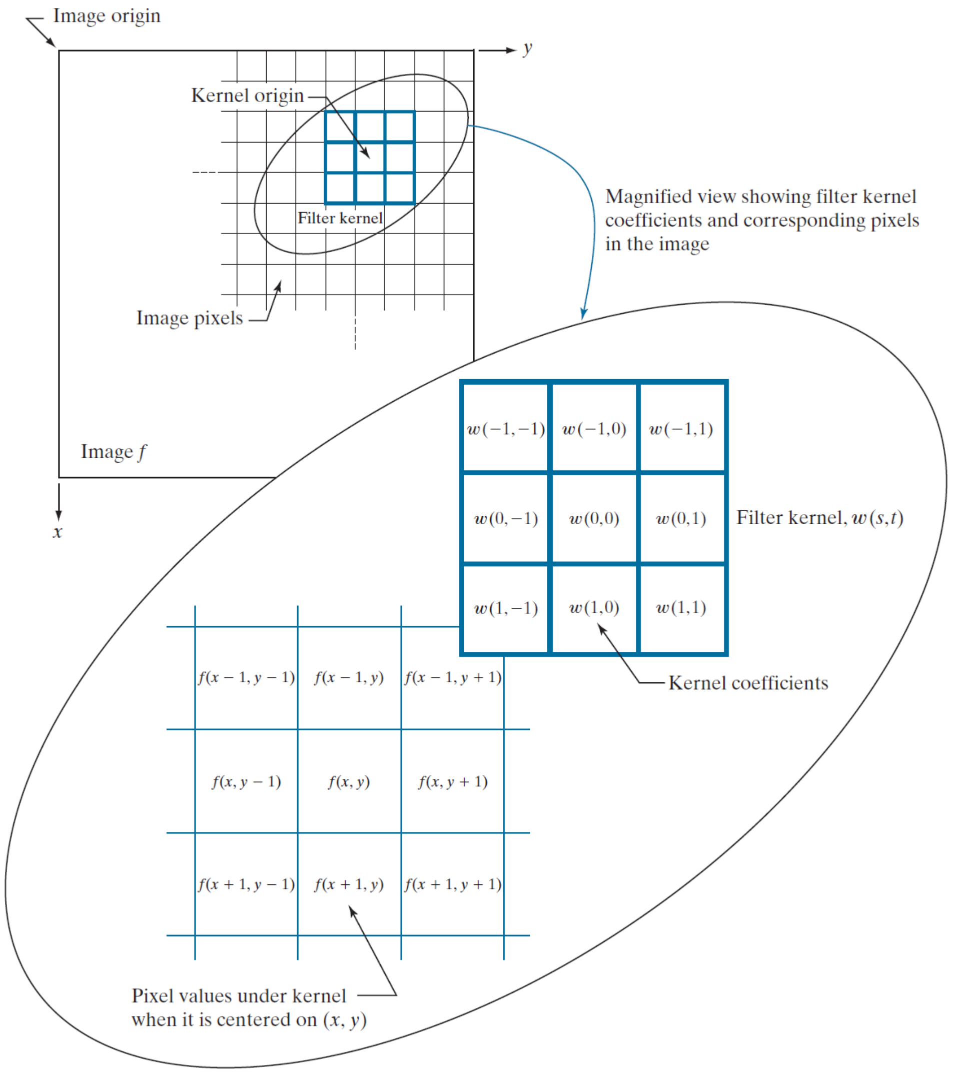




Imperial College London

Spatial filters: Illustration

FIGURE The mechanics of linear spatial filtering using a 3×3 kernel. The pixels are shown as squares to simplify the graphics. Note that the origin of the image is at the top left, but the origin of the kernel is at its center. Placing the origin at the center of spatially symmetric kernels simplifies writing expressions for linear filtering.



Smoothing (lowpass) spatial filters

- Convolving a **smoothing kernel** with an image blurs the image, with the degree of blurring being determined by the size of the kernel and the values of its coefficients.
- Smoothing (also called **averaging**) spatial filters are used to reduce sharp transitions in intensity.
- Because random noise typically consists of sharp transitions in intensity, an obvious application of smoothing is **noise reduction**.
- Smoothing is used to **reduce irrelevant detail** in an image, where "irrelevant" refers to pixel regions that are small with respect to the size of the filter kernel.
- Another application is **smoothing false contours** that result from using an insufficient number of intensity levels in an image.
- Smoothing filters can be used in combination with other techniques for image enhancement, such as the histogram processing techniques.

Imperial College London

Local averaging spatial masks for image de-noising (smoothing)
Box filter kernels

- The simplest, separable lowpass filter kernel is the **box kernel**, whose coefficients have the same value (typically 1). The name "box kernel" comes from a constant kernel resembling a box when viewed in 3-D.
- An $m \times n$ box filter is an $m \times n$ array of 1's, with a normalizing constant in front, whose value is 1 divided by the sum of the values of the coefficients (i.e., $1/mn$ when all the coefficients are 1).
- This normalization, which we apply to all lowpass kernels, has two purposes.
  - First, the average value of an area of constant intensity would equal that intensity in the filtered image, as it should.
  - Second, normalizing the kernel in this way prevents introducing a **bias** during filtering; that is, the sum of the pixels in the original and filtered images will be the same.
- The box kernel is **separable**.
- **Padding** extends the boundaries of an image to avoid undefined operations when parts of a kernel lie outside the border of the image during filtering. When zero (black) padding is used, the net result of smoothing at or near the border is a dark gray border that arises from including black pixels in the averaging process.

Definition of separable kernels and illustration of box kernels

- A two dimensional spatial kernel is separable if it can be expressed as the product of two one-dimensional spatial kernels.
- The box kernel is separable since

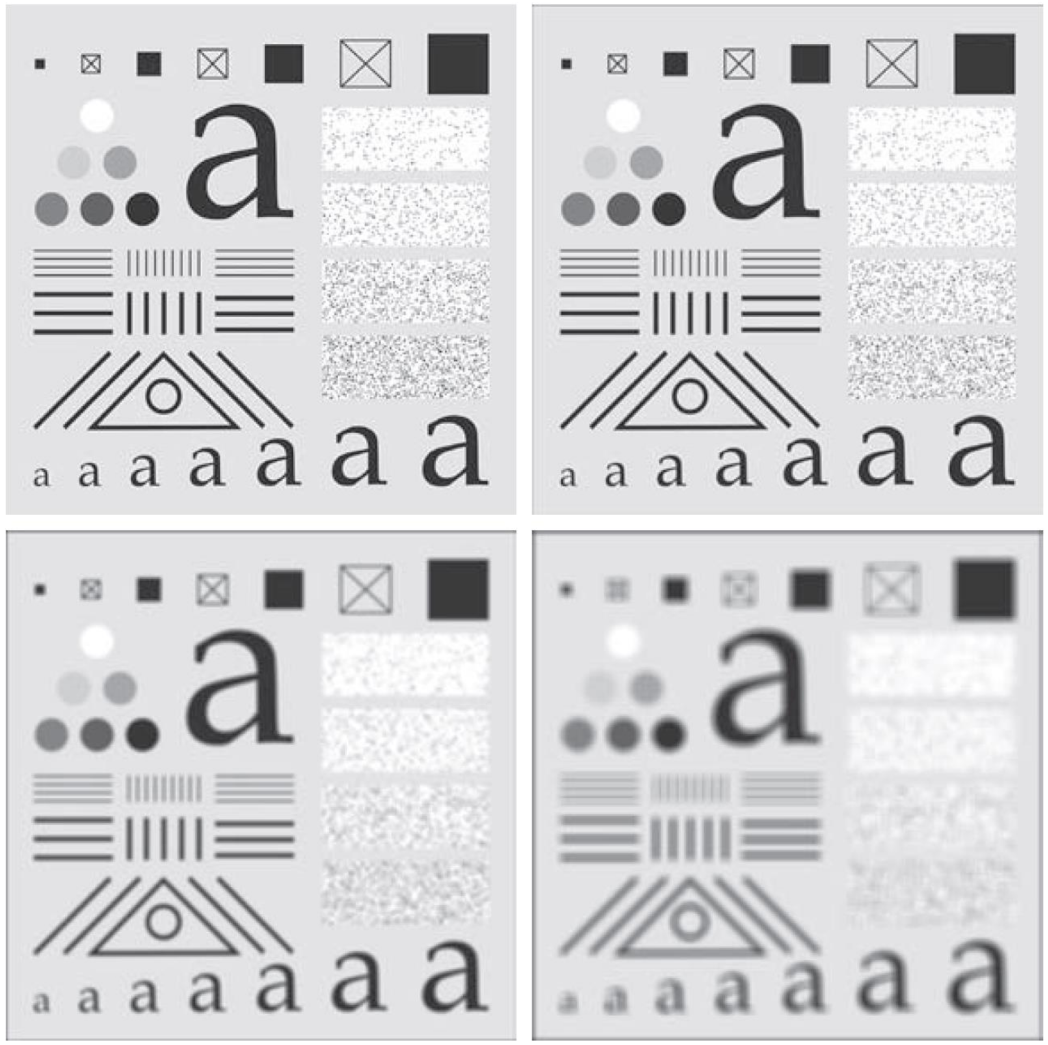
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} = \frac{1}{3}\begin{bmatrix}1\\ 1\\ 1\end{bmatrix}\cdot\frac{1}{3}\begin{bmatrix}1 & 1 & 1\end{bmatrix}$$

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1&1&1\\ \hline 1&1&1\\ \hline 1&1&1\\ \hline\end{array}\qquad \frac{1}{25}\times\begin{array}{|c|c|c|c|c|}\hline 1&1&1&1&1\\ \hline 1&1&1&1&1\\ \hline 1&1&1&1&1\\ \hline 1&1&1&1&1\\ \hline 1&1&1&1&1\\ \hline\end{array}\qquad \frac{1}{49}\times\begin{array}{|c|c|c|c|c|c|c|}\hline 1&1&1&1&1&1&1\\ \hline 1&1&1&1&1&1&1\\ \hline 1&1&1&1&1&1&1\\ \hline 1&1&1&1&1&1&1\\ \hline 1&1&1&1&1&1&1\\ \hline 1&1&1&1&1&1&1\\ \hline 1&1&1&1&1&1&1\\ \hline\end{array}$$

Imperial College London

Local averaging spatial masks for image de-noising (smoothing)

Example: Box filter kernels



a b
c d

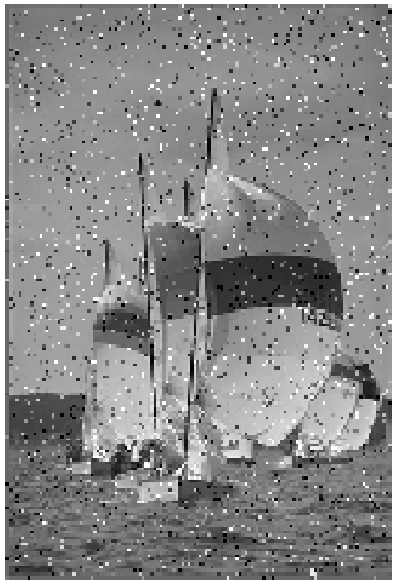
FIGURE
(a) Test pattern of size 1024×1024 pixels.
(b)-(d) Results of lowpass filtering with box kernels of sizes 3×3, 11×11, and 21×21, respectively.

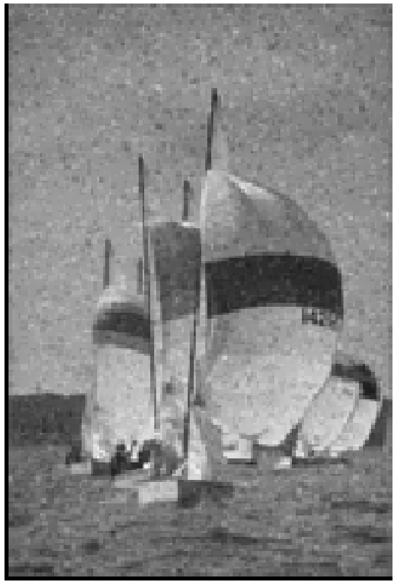
Imperial College London

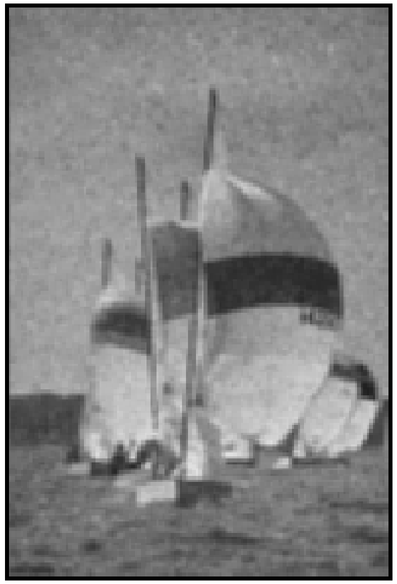
Local averaging spatial masks for image de-noising (smoothing)

Example: Box filter kernels

Original Image


Noisy Image

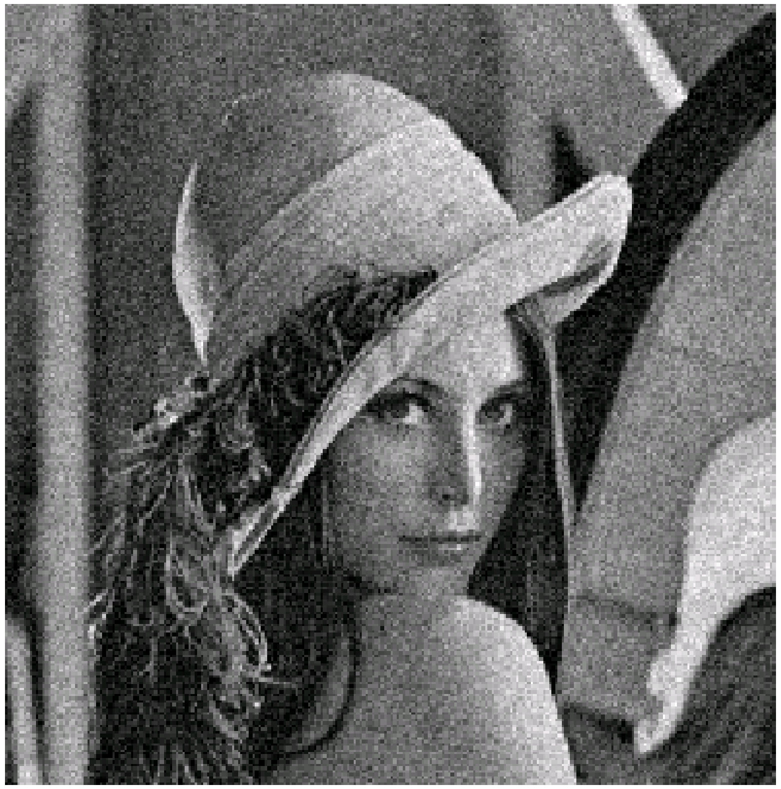

3×3 smoothing


5×5 smoothing

Imperial College London

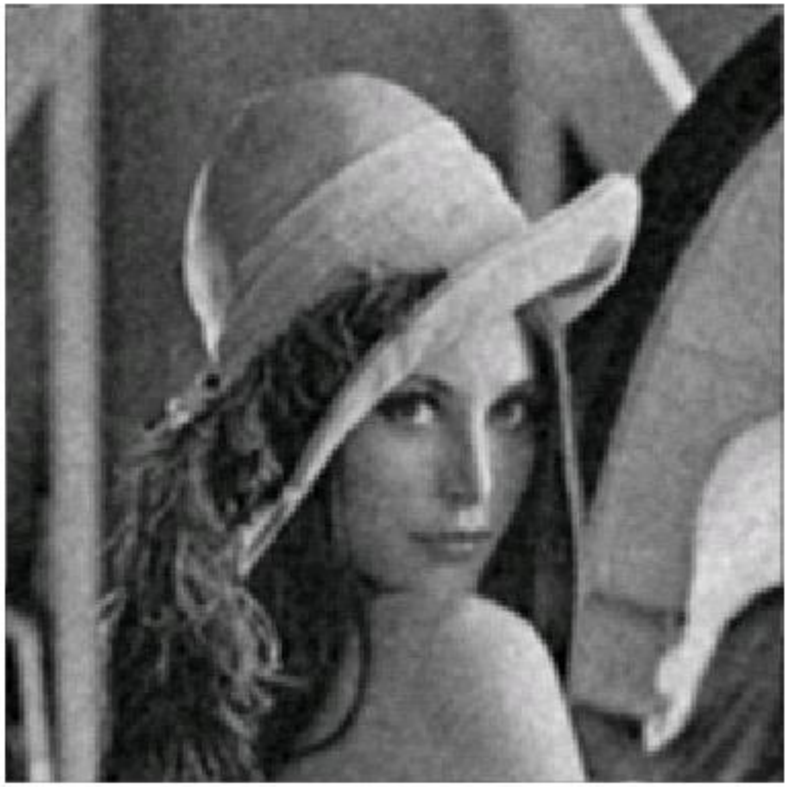
A noiseless image and its noisy version

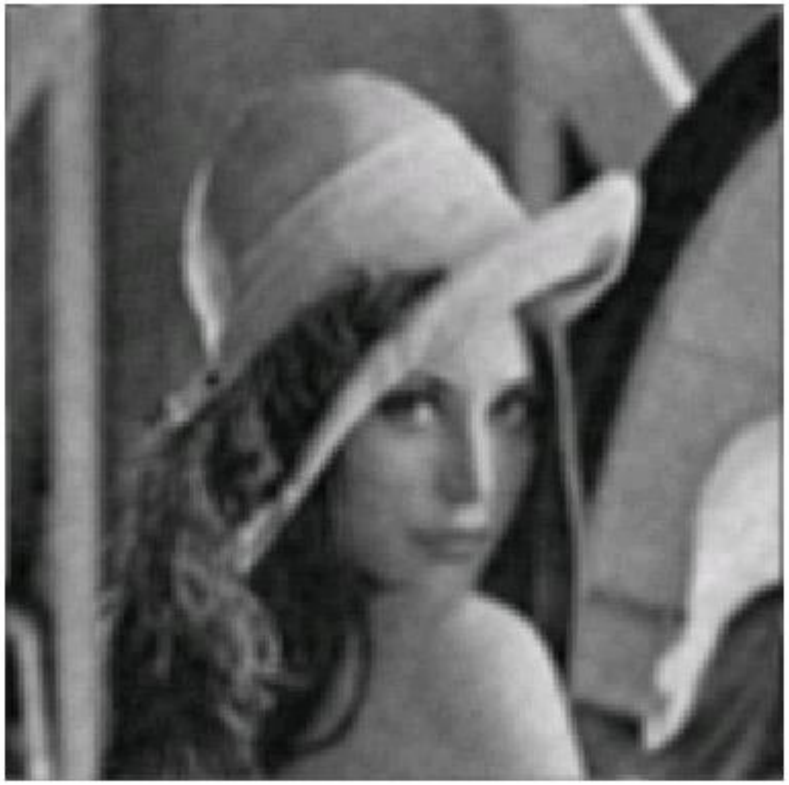
Original Lena image


Lena image with 10db noise

Imperial College London

A noiseless image and its noisy version: example


3×3 averaging mask


5×5 averaging mask

Imperial College London

Comments on previous example

- For $m = 3$, we note a slight overall blurring of the image.
- The image features whose sizes are comparable to the size of the kernel are affected significantly more. Such features include:
  - The **thinner** lines in the image.
  - The **noise pixels** contained in the boxes on the right side of the image.
- The filtered image has a thin gray border, the result of zero-padding the image prior to filtering.
- As indicated earlier, padding extends the boundaries of an image to avoid undefined operations when parts of a kernel lie outside the border of the image during filtering. When zero (black) padding is used, the net result of smoothing at or near the border is a dark gray border that arises from including black pixels in the averaging process.

Comments on previous example

- Using the 11×11 kernel resulted in more pronounced blurring throughout the image, including a more prominent dark border.
- The result with the 21×21 kernel shows significant blurring of all components of the image, including the loss of the characteristic shape of some components, as for example, the small square on the top left and the small character on the bottom left.
- The dark border resulting from zero padding is proportionally thicker than before.
- There are more intelligent methods for padding but these are out of the scope of this course.

Imperial College London

Local averaging spatial masks for image de-noising (smoothing)
Gaussian filter kernels

- Box filters have limitations that make them poor choices in many applications. For example, a defocused lens is often modelled as a lowpass filter, but box filters are poor approximations to the blurring characteristics of lenses.
- The **Gaussian kernel** is a circularly symmetric and separable kernel, defined as

$$w(x,y) = Ke^{-\frac{x^2+y^2}{2\sigma^2}}$$

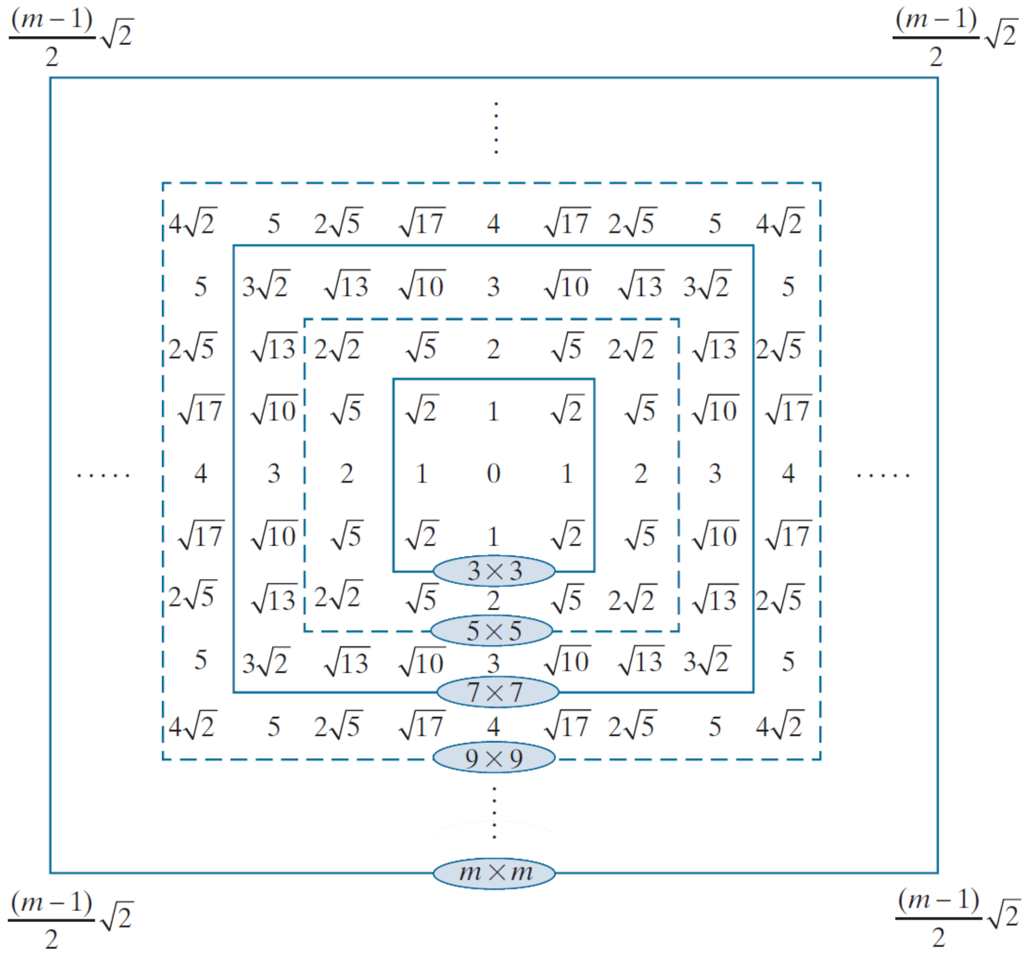
  - By letting $r = \sqrt{x^2 + y^2}$ we can replace $w(x,y)$ with $w(r) = Ke^{-\frac{r^2}{2\sigma^2}}$.

Imperial College London

Local averaging spatial masks for image de-noising (smoothing)

Gaussian filter kernels

- What you see below is the distances from the centre for various sizes of square Gaussian kernels.



$\frac{(m-1)}{2}\sqrt{2}$
$\frac{(m-1)}{2}\sqrt{2}$
$4\sqrt{2}$ 5 $2\sqrt{5}$ $\sqrt{17}$ 4 $\sqrt{17}$ $2\sqrt{5}$ 5 $4\sqrt{2}$
5 $3\sqrt{2}$ $\sqrt{13}$ $\sqrt{10}$ 3 $\sqrt{10}$ $\sqrt{13}$ $3\sqrt{2}$ 5
$2\sqrt{5}$ $\sqrt{13}$ $2\sqrt{2}$ $\sqrt{5}$ 2 $\sqrt{5}$ $2\sqrt{2}$ $\sqrt{13}$ $2\sqrt{5}$
$\sqrt{17}$ $\sqrt{10}$ $\sqrt{5}$ $\sqrt{2}$ 1 $\sqrt{2}$ $\sqrt{5}$ $\sqrt{10}$ $\sqrt{17}$
4 3 2 1 0 1 2 3 4
$\sqrt{17}$ $\sqrt{10}$ $\sqrt{5}$ $\sqrt{2}$ 1 $\sqrt{2}$ $\sqrt{5}$ $\sqrt{10}$ $\sqrt{17}$
$2\sqrt{5}$ $\sqrt{13}$ $2\sqrt{2}$ $\sqrt{5}$ 2 $\sqrt{5}$ $2\sqrt{2}$ $\sqrt{13}$ $2\sqrt{5}$
5 $3\sqrt{2}$ $\sqrt{13}$ $\sqrt{10}$ 3 $\sqrt{10}$ $\sqrt{13}$ $3\sqrt{2}$ 5
$4\sqrt{2}$ 5 $2\sqrt{5}$ $\sqrt{17}$ 4 $\sqrt{17}$ $2\sqrt{5}$ 5 $4\sqrt{2}$
3×3
5×5
7×7
9×9
$m\times m$
$\frac{(m-1)}{2}\sqrt{2}$
$\frac{(m-1)}{2}\sqrt{2}$

Local averaging spatial masks for image de-noising (smoothing)

Gaussian filter kernels

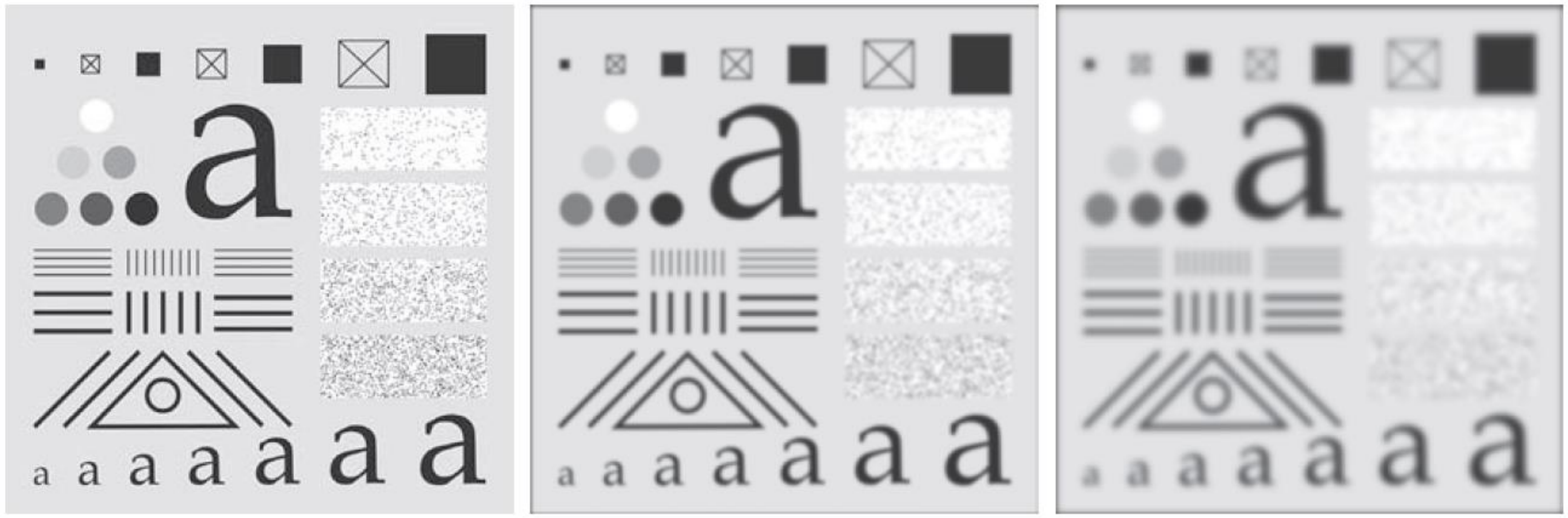
- 3×3 average (box) filter and Gaussian filter with $K = \sigma = 1$.

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{4.8976} \times \begin{array}{|c|c|c|} \hline 0.3679 & 0.6065 & 0.3679 \\ \hline 0.6065 & 1.0000 & 0.6065 \\ \hline 0.3679 & 0.6065 & 0.3679 \\ \hline \end{array}$$

Imperial College London

Local averaging spatial masks for image de-noising (smoothing)

Gaussian filter kernels: example



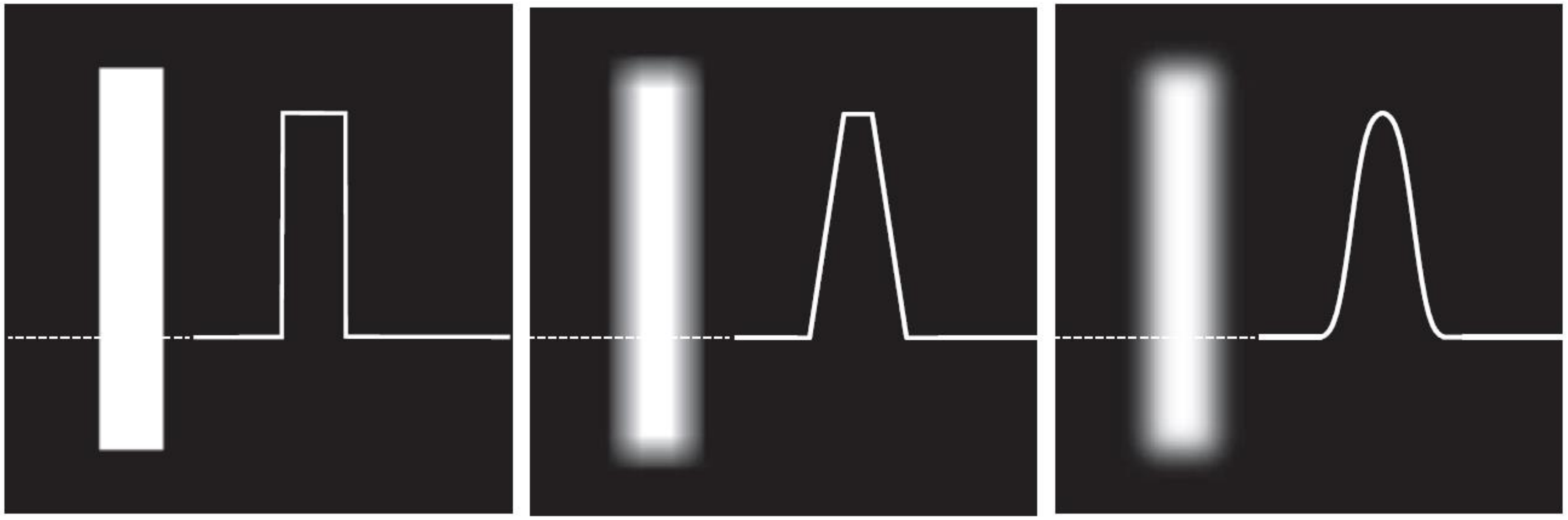
a b c

(a) A test image of size 1024×1024. (b) Result of lowpass filtering with a Gaussian kernel of size 21×21, with standard deviation $\sigma = 3.5$. (c) Result of using a kernel of size 43×43, with $\sigma = 7$. We used $K = 1$ in all cases.

Imperial College London

Local averaging spatial masks for image de-noising (smoothing)

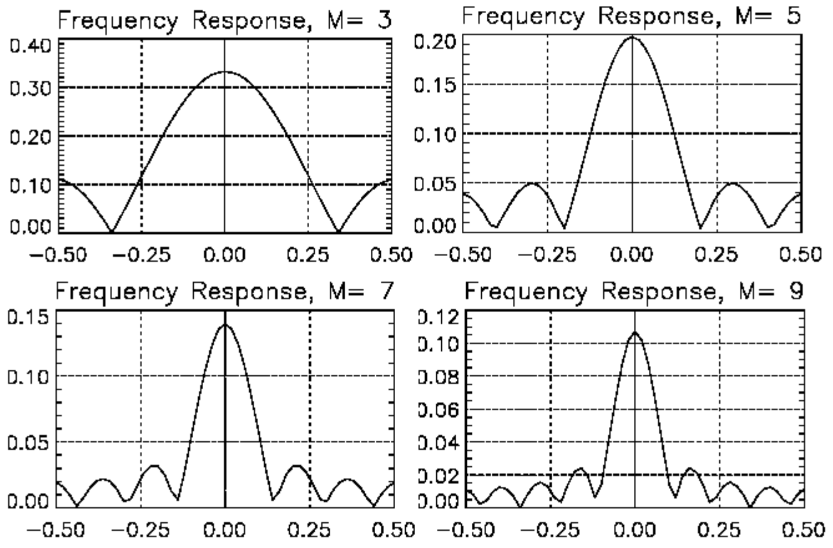
Gaussian filter kernels: example

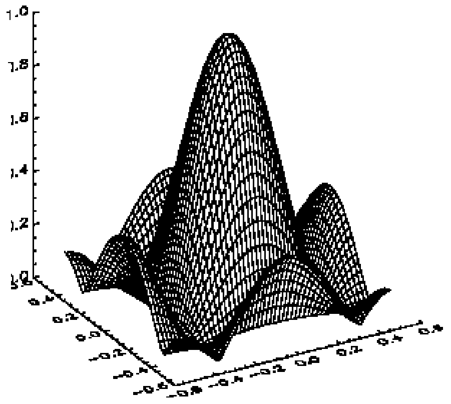

a b c

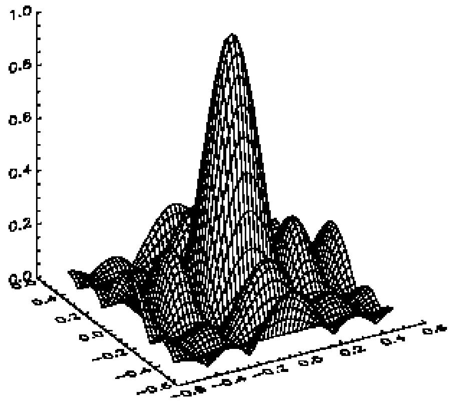
(a) Image of a white rectangle on a black background, and a horizontal intensity profile along the dotted scan line. (b) Result of smoothing this image with a box kernel of size 71×71, and corresponding intensity profile. (c) Result of smoothing the image using a Gaussian kernel of size 151×151, with $K = 1$ and $\sigma = 25$. Note the smoothness of the profile in (c) compared to (b). The image and rectangle are of sizes 1024×1024 and 768×128, respectively.

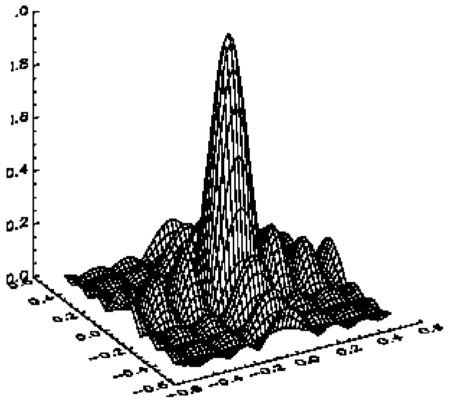
Imperial College London

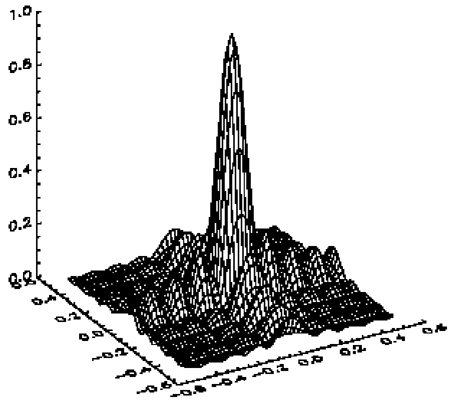
Response of 1D and 2D local averaging spatial masks




Frequency Response for M=3


Frequency Response for M=5


Frequency Response for M=7


Frequency Response for M=9

Imperial College London

Order-statistic (nonlinear) filters -Median filtering

- **Order-statistic filters** are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the region encompassed by the filter.
- Smoothing is achieved by replacing the value of the centre pixel with the value determined by the ranking result.
- The best-known filter in this category is the **median filter**, which, as its name implies, replaces the value of the centre pixel by the median of the intensity values in the neighbourhood of that pixel (the value of the centre pixel is included in computing the median).
- Median filters provide excellent noise reduction capabilities for certain types of random noise, with considerably less blurring than linear smoothing filters of similar size.
- Median filters are particularly effective in the presence of **impulse noise** (sometimes called **salt-and-pepper noise**, when it manifests itself as white and black dots superimposed on an image).

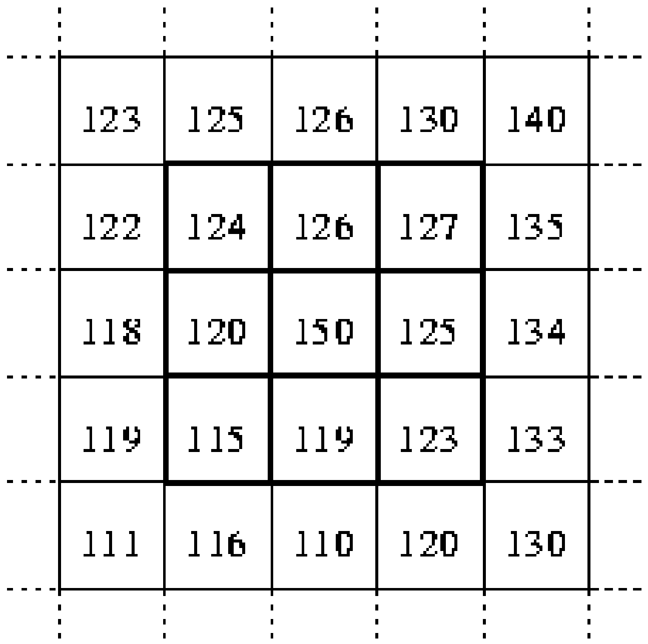
Imperial College London

Order-statistic (nonlinear) filters -Median filtering

- ❑ The median, m, of a set of values is such that half the values in the set are less than or equal to m and half are greater than or equal to m.
- ❑ In order to perform median filtering at a point in an image, we first sort the values of the pixels in a pre-specified neighbourhood, determine their median, and assign that value to the pixel in the filtered image corresponding to the centre of the neighbourhood.
- ❑ For example, in a 3×3 neighborhood the median is the 5th largest value, in a 5×5 neighbourhood it is the 13th largest value, and so on.
- ❑ When several values in a neighbourhood are the same, all equal values are grouped. For example, suppose that a 3×3 neighbourhood has values $(10, 20, 20\ 15, 20, 20, 25, 20, 100)$. These values are sorted as $(\mathbf{10, 15, 20, 20, 20, 20, 20, 25, 100})$ which results in a median of 20.
- ❑ The principal function of median filters is to force points to be like their neighbours.
- ❑ Isolated clusters of pixels that are light or dark with respect to their neighbours, and whose area is less than half of the pixels in the pre-specified neighbourhood, are forced to have the value of the median intensity of the pixels in that neighbourhood.

Imperial College London

Median filtering: example



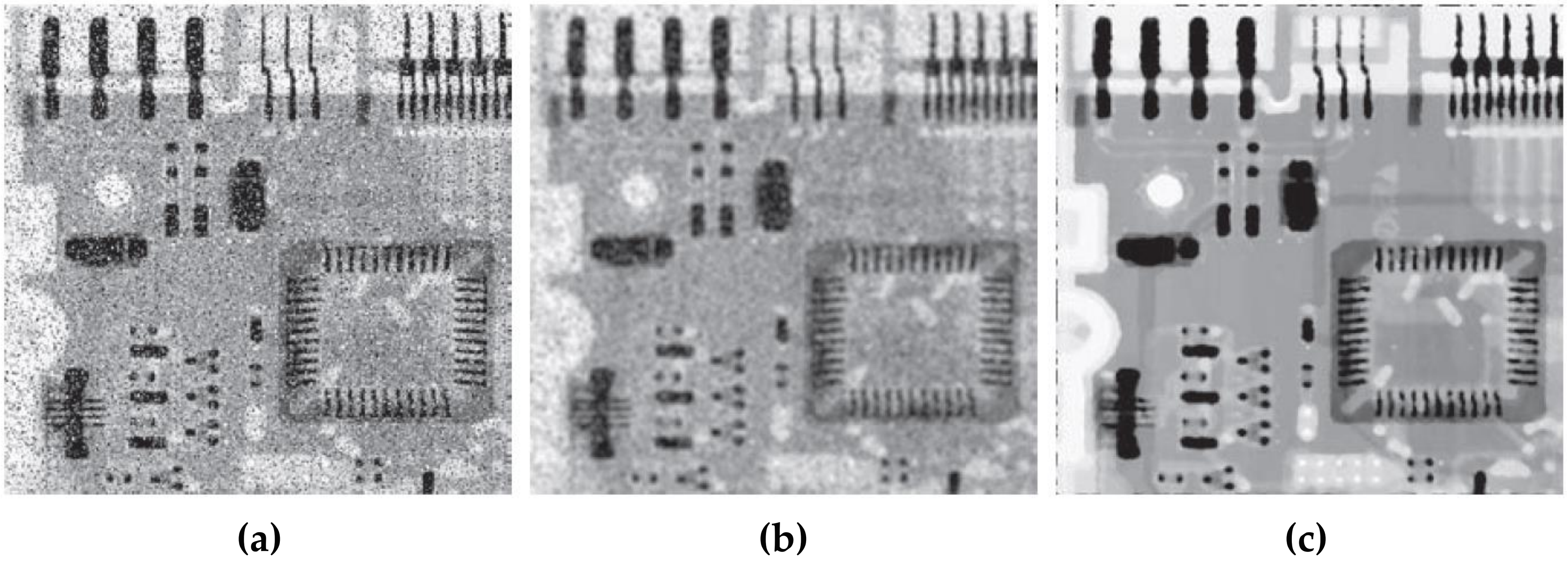
Neighbourhood values:

115, 119, 120, 123, 124,
125, 126, 127, 150

Median value: 124

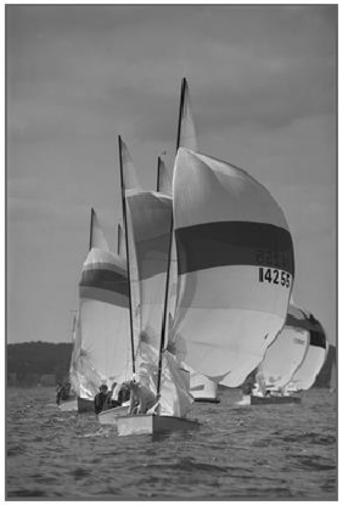
Imperial College London

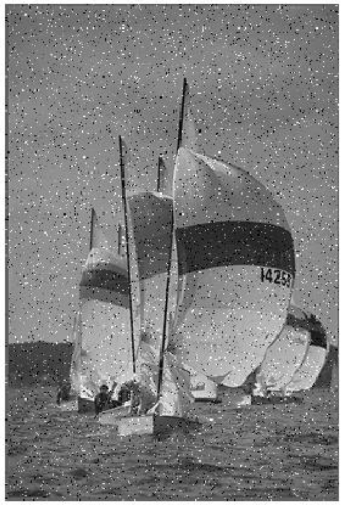
Median filtering: example

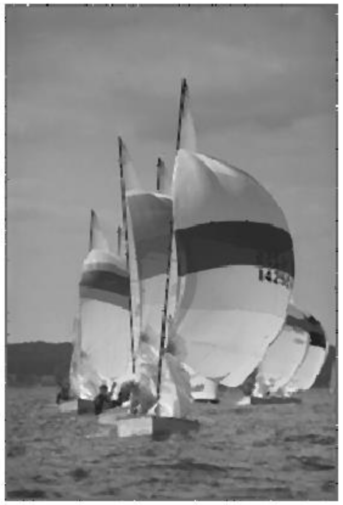

(a) (b) (c)

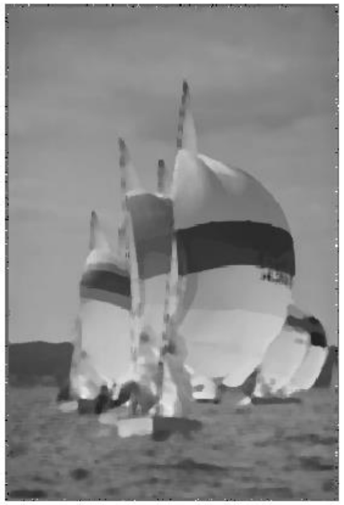
X-ray image of a circuit board, corrupted by salt-and-pepper noise. (b) Noise reduction using a 19×19 Gaussian lowpass filter kernel with $\sigma = 3$. (c) Noise reduction using a 7×7 median filter.

Median filtering: example


Original

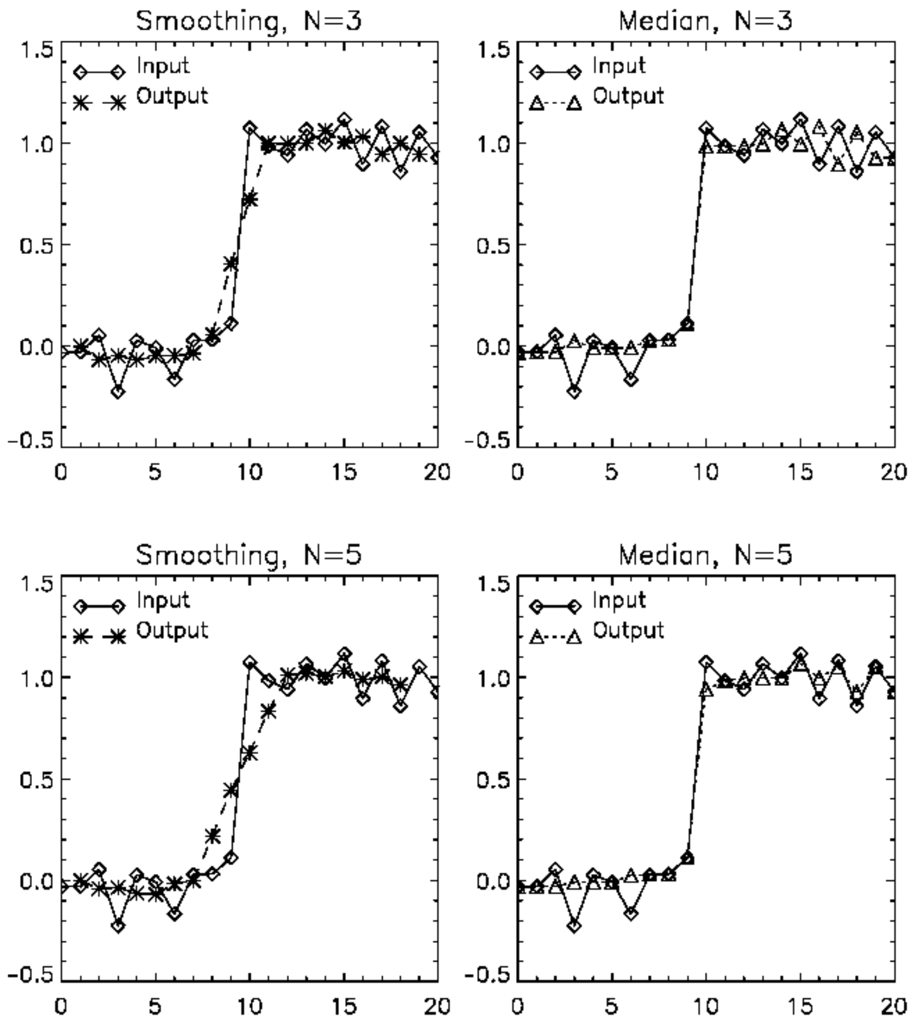

Original with Noise


After N=3 Median Filter


After N=5 Median Filter

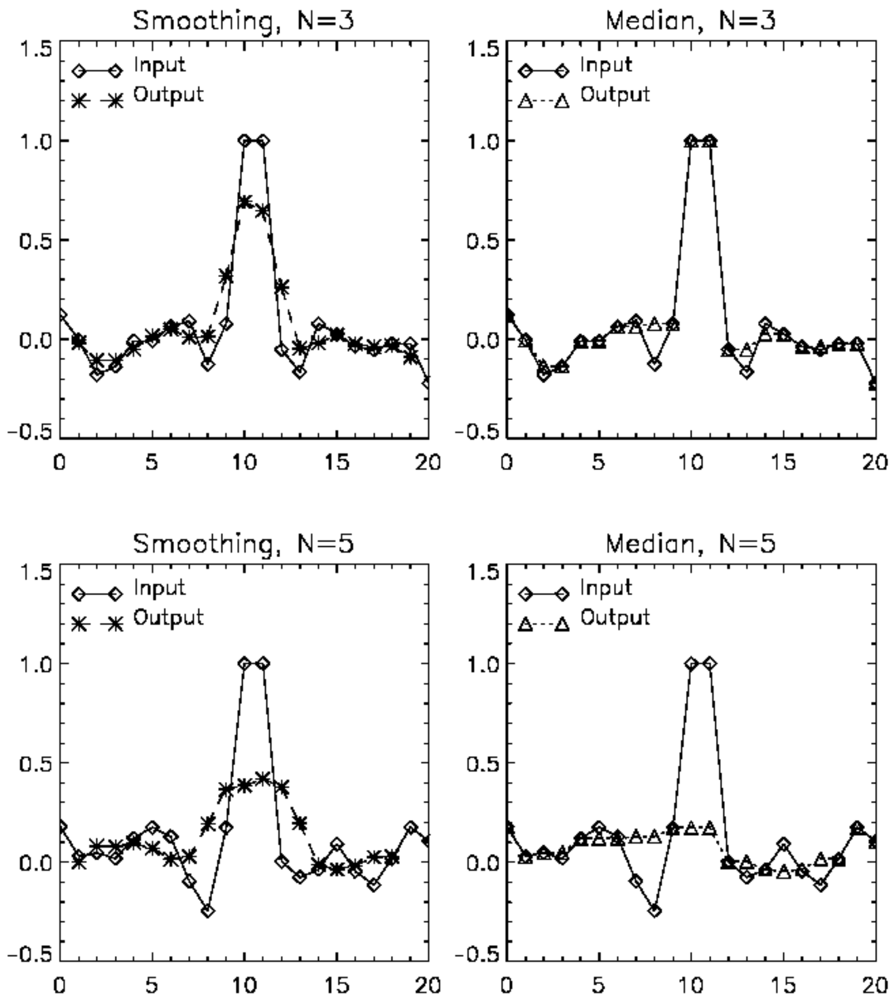
Imperial College London

Median versus local averaging filter response around local step edge



Imperial College London

Median versus local averaging filter response around local ridge edge



Imperial College
London

High pass filtering

- We use masks with positive coefficients around the centre of the mask and negative coefficients in the periphery or the other way round.
- In the mask shown below the central coefficient is $+8$ and its 8 nearest neighbours are -1.
- The reversed signs are equally valid, i.e., -8 in the middle and $+1$ for the rest of the coefficients.
- The sum of the mask coefficients must be zero. This leads to the response of the mask within a constant (or slowly varying) intensity area being zero (or very small).

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & -8 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Imperial College London

Example of high pass filtering

original image

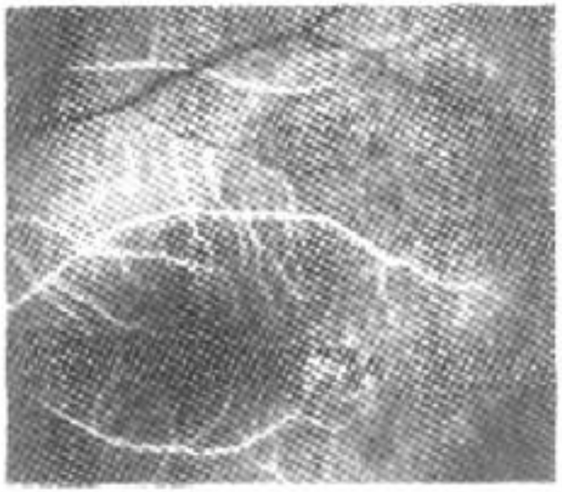
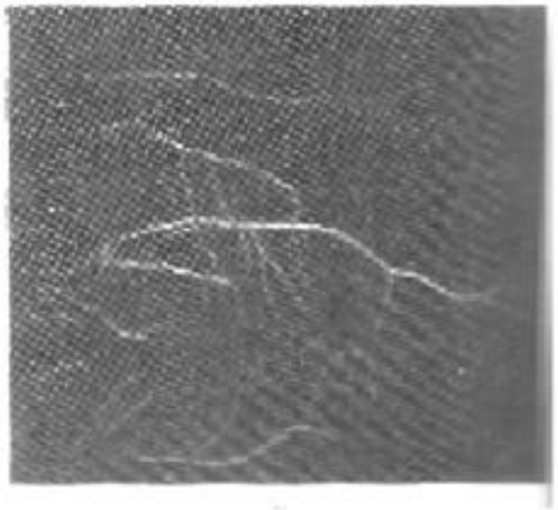
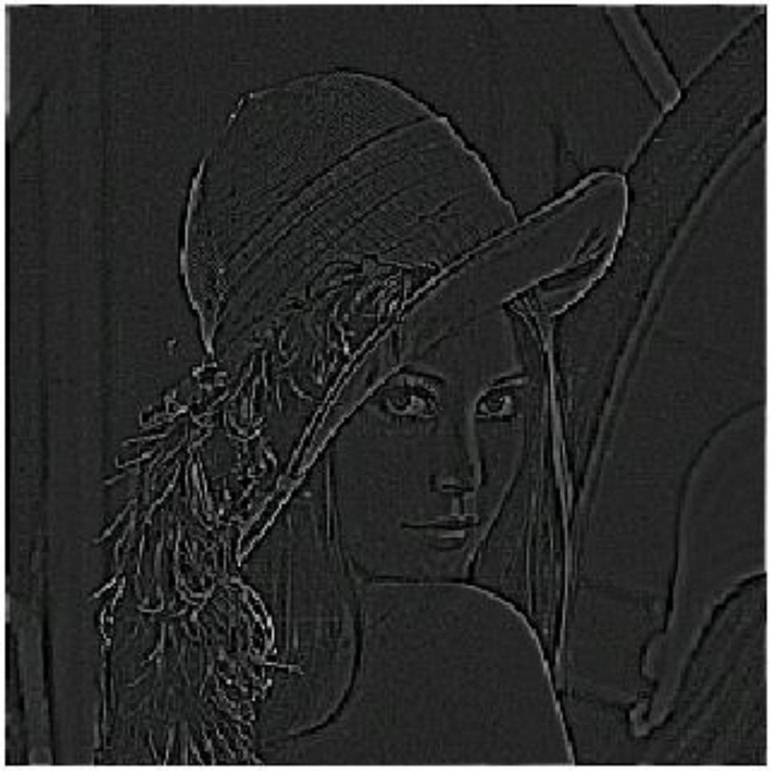
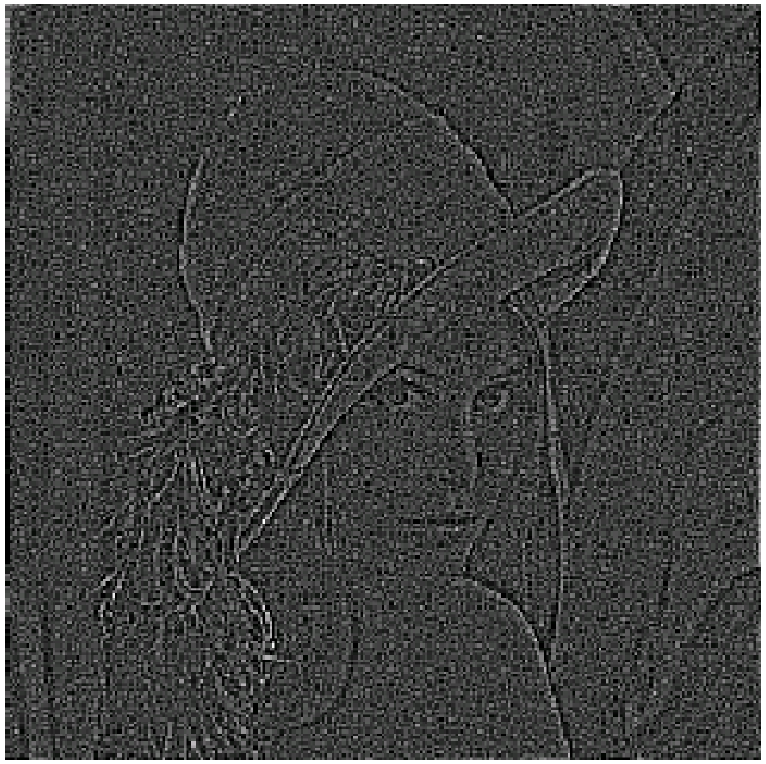


image filtered using the 3x3 mask: highlighting of edges, but lower contrast



Example of high pass filtering


High pass filtered image


High pass filtered image with $10dB$ noise

High boost filtering

- The goal of high boost filtering is to enhance the high frequency information without completely eliminating the background of the image.
- We know that:

  (High-pass filtered image)=(Original image)-(Low-pass filtered image)
- We define:

  (High boost filtered image)=A ×(Original image)-(Low-pass filtered image)

  (High boost)=$(A-1)$ ×(Original)+(Original)-(Low-pass)

  (High boost)=$(A-1) \times$ (Original)+(High-pass)
- As you can see, when $A > 1$, part of the original is added back to the high-pass filtered version of the image in order to partly restore the low frequency components that would have been eliminated with standard high-pass filtering.
- Typical values for A are values slightly higher than 1, as for example 1.15, 1.2, etc.

Imperial College London

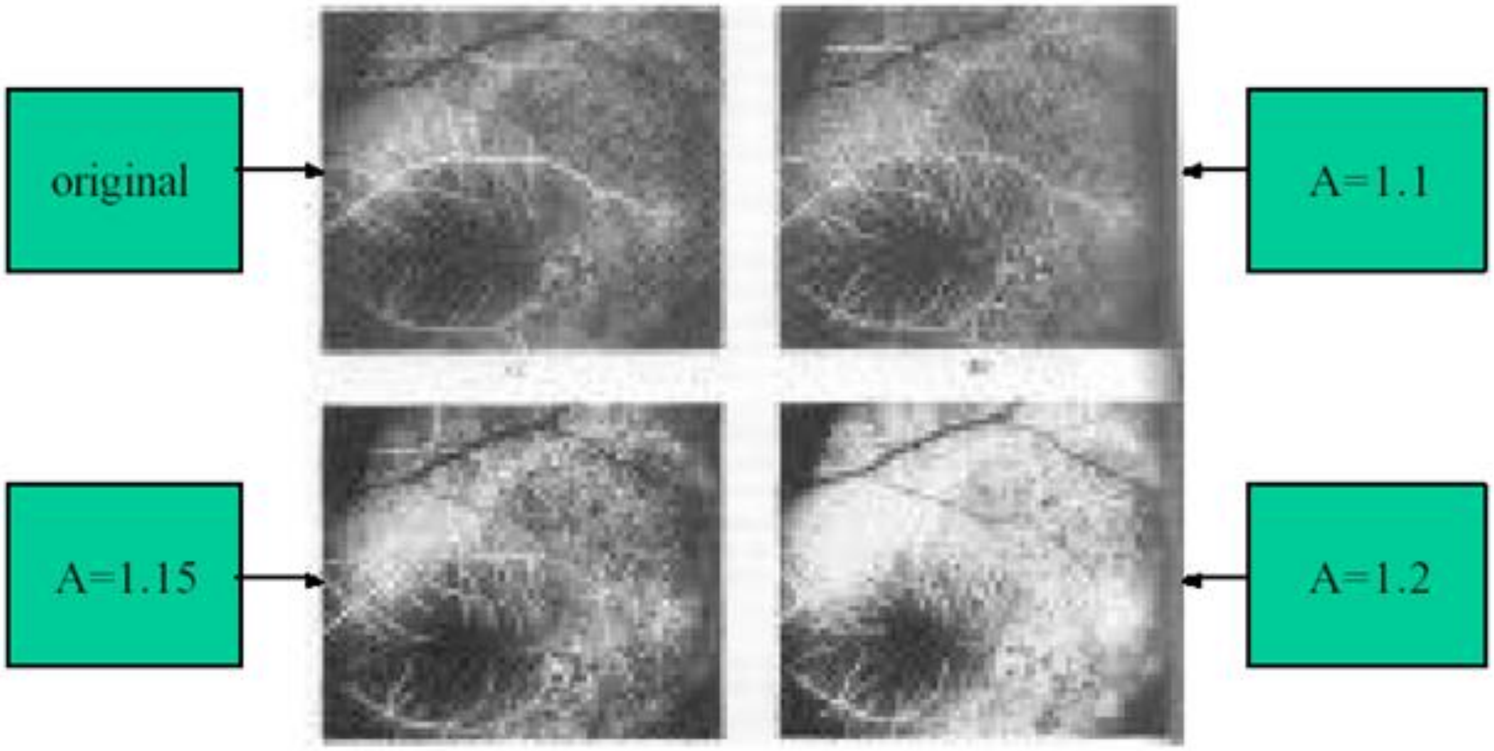
High boost filtering

- The resulting image looks similar to the original image with some edge enhancement.
- The spatial mask that implements the high boost filtering algorithm is shown below.
- The resulting image depends on the choice of A.

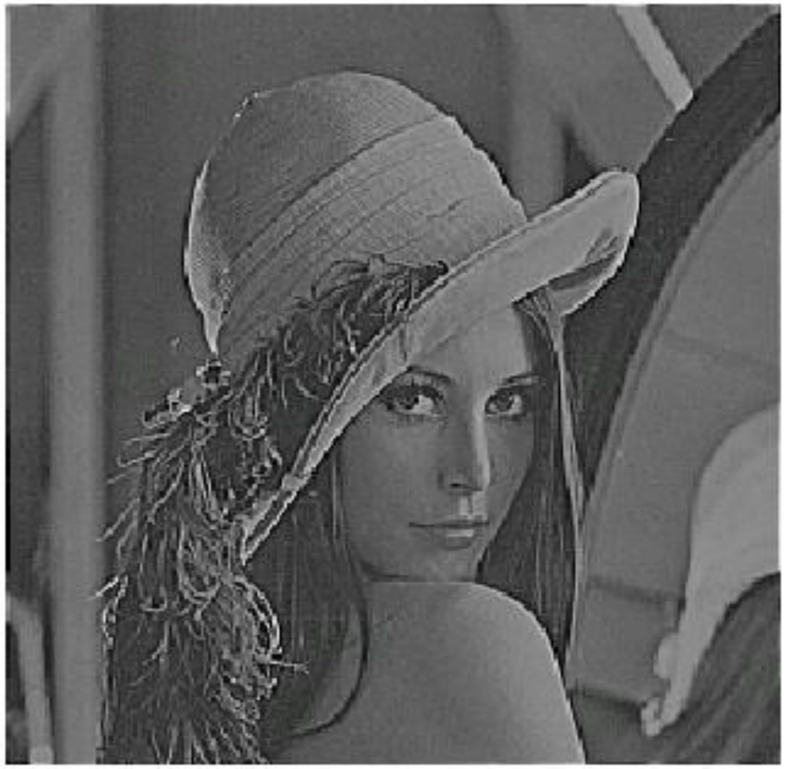
$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & w = 9A - 1 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

- High boost filtering is used in printing and publishing industry.

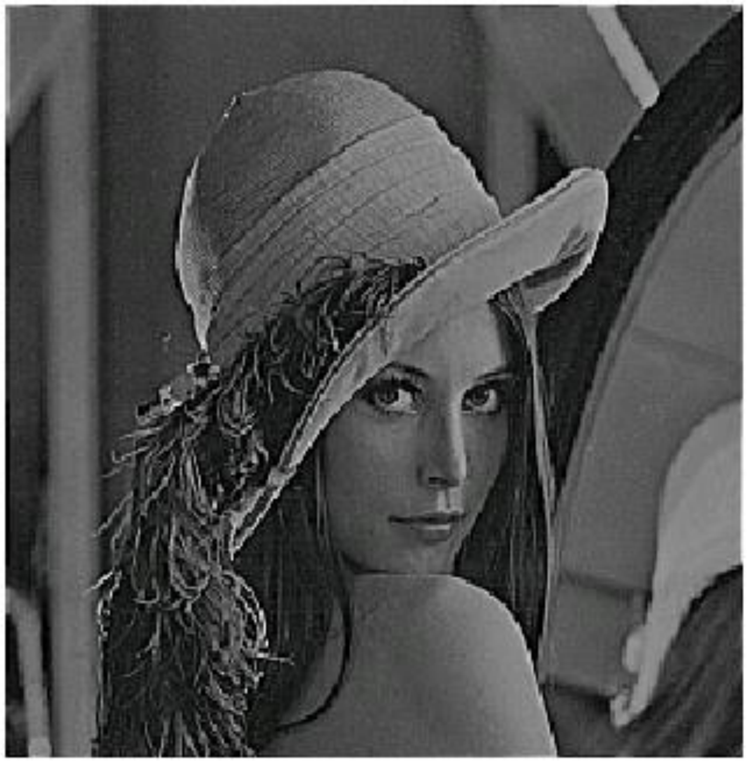
Example of high boost filtering



Example of high boost filtering



$A = 1.15$



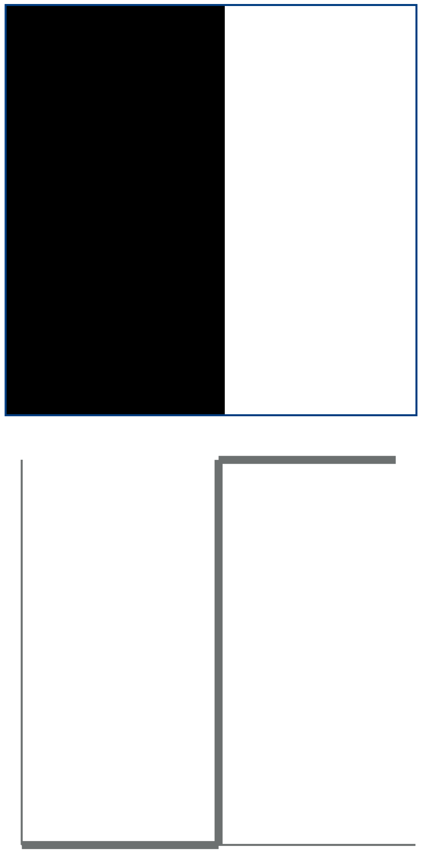
$A = 1.2$

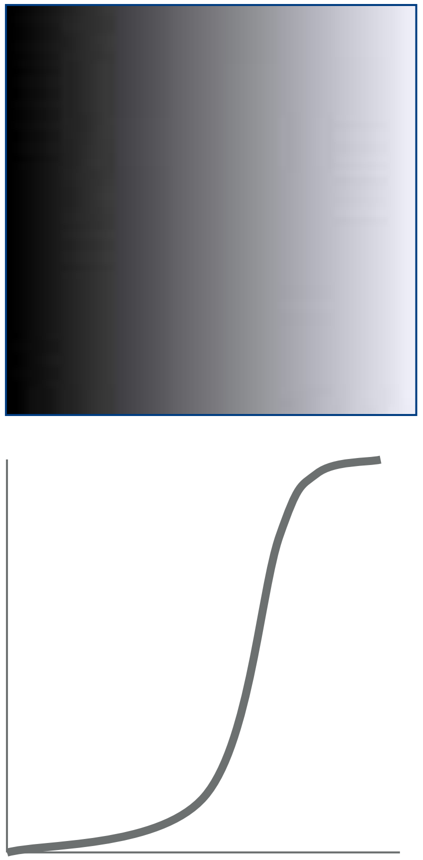
Edge detection

- Edges are abrupt local changes in the image value (for example grey level).
- Edges are of crucial important because they are related to the boundaries of the various objects present in the image.
- Object boundaries are key feature for object detection and identification.
- Edge information in an image is found by looking at the relationship a pixel has with its neighborhoods.
- If a pixel's grey-level value is similar to those around it, there is probably <u>not</u> an edge at that point.
- If a pixel's has neighbors with widely varying grey levels, it may present an edge point.

Imperial College London

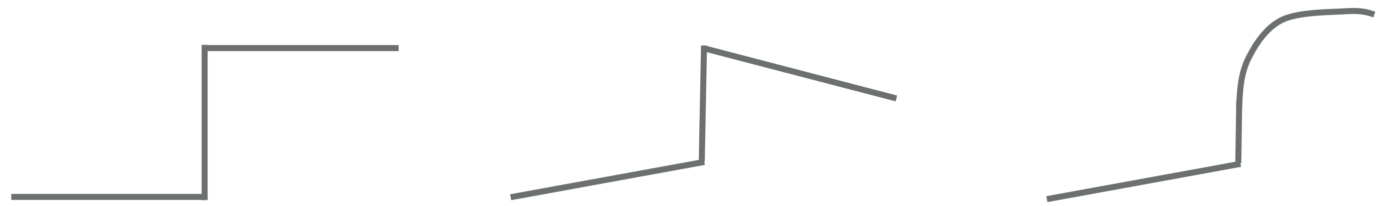
Abrupt versus gradual change in image intensity



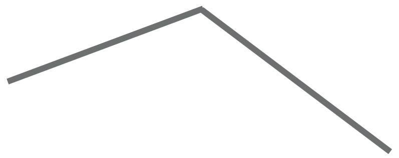


Imperial College London

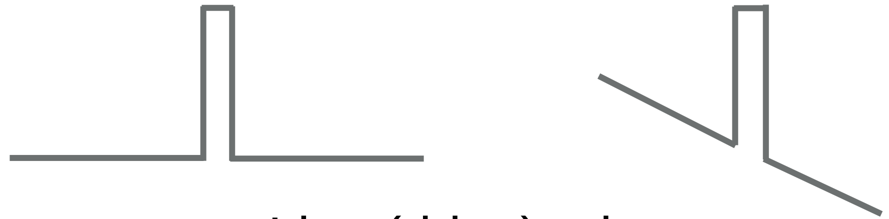
Type of edges



Step edges

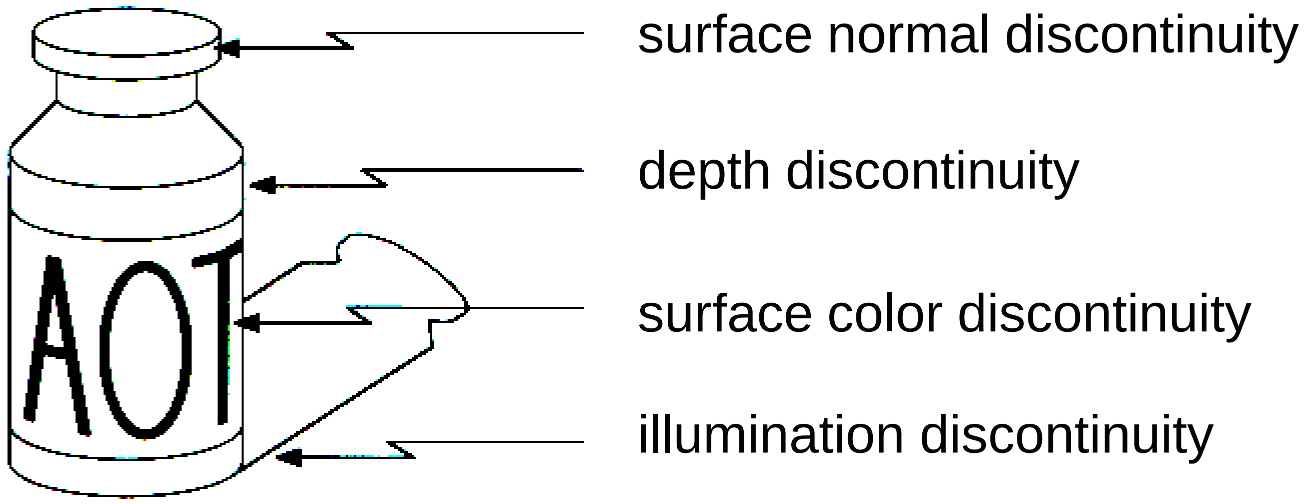


Roof edge



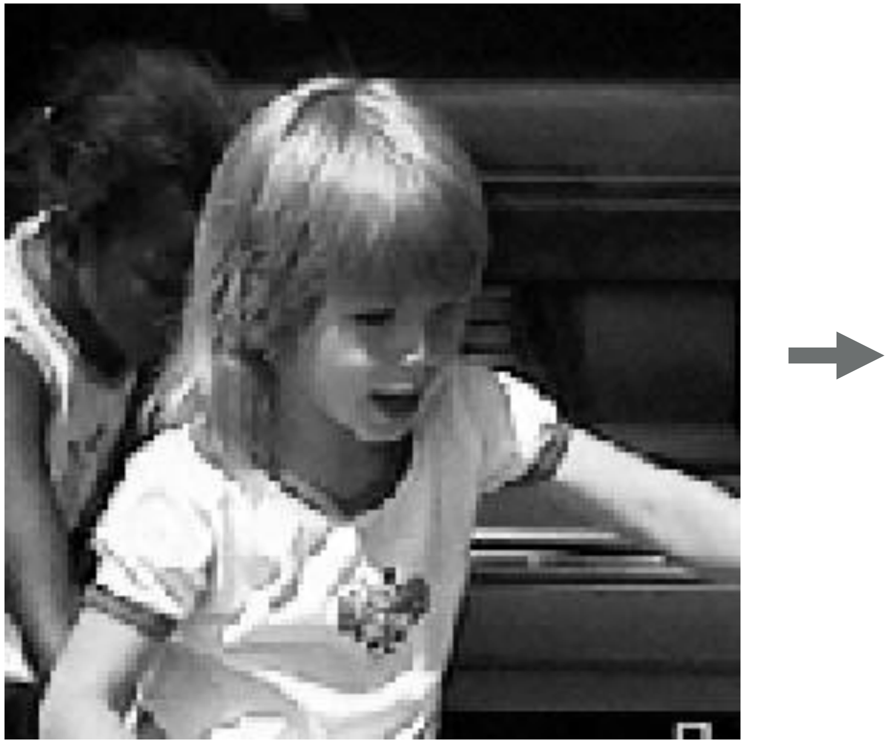
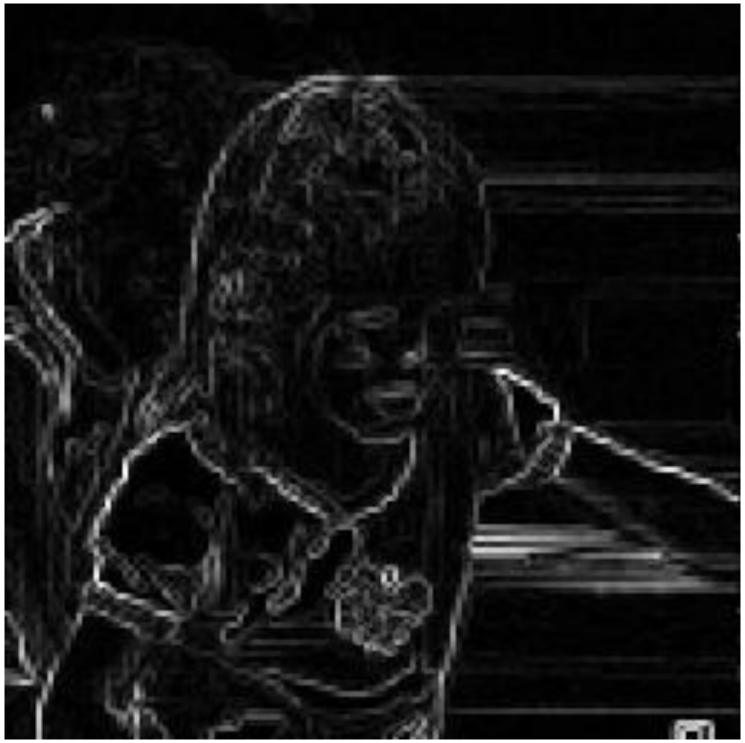
Line (ridge) edges

Edges are caused by a variety of factots


AOT
surface normal discontinuity
depth discontinuity
surface color discontinuity
illumination discontinuity

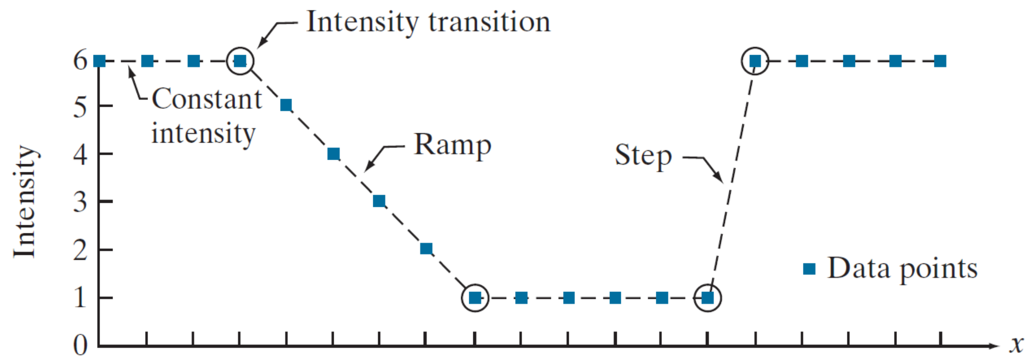
Imperial College London

From a grey level image to an edge image (or edge map)

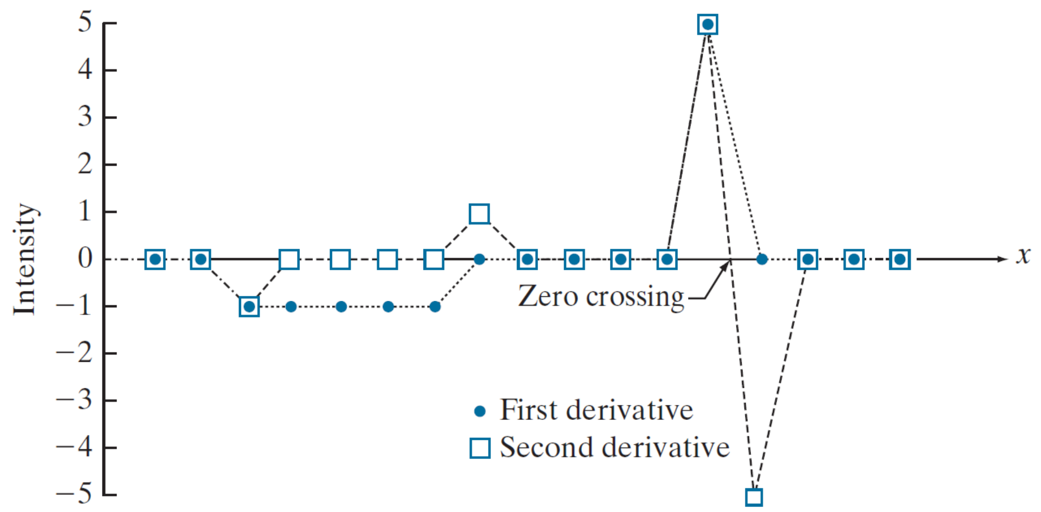



Imperial College London

From a grey level image to an edge image (or edge map)



Values of scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
1st derivative		0	0	−1	−1	−1	−1	−1	0	0	0	0	0	5	0	0	0	0	
2nd derivative		0	0	−1	0	0	0	0	1	0	0	0	0	5	−5	0	0	0	



a
b
c

FIGURE
(a) A section of a horizontal scan line from an image, showing ramp and step edges, as well as constant segments.
(b) Values of the scan line and its derivatives.
(c) Plot of the derivatives, showing a zero crossing. In (a) and (c) points were joined by dashed lines as a visual aid.

Imperial College London

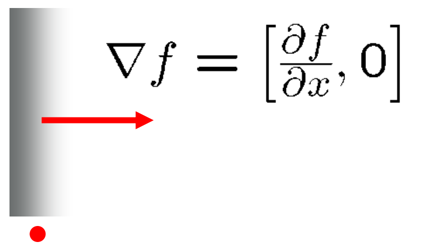
How do we detect the presence of a local edge?

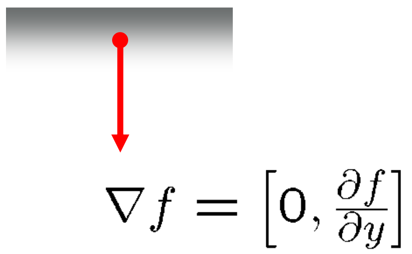
- The gradient (first derivative) of the image around a pixel might give information about how detailed is the area around a pixel and whether there are abrupt intensity changes.
- The image is a 2-D signal and therefore, the gradient at a location (x, y) is a 2-D vector which contains the two partial derivatives of the image with respect to the coordinates x, y.

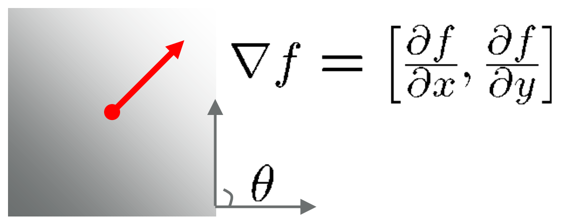
$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- Edge strength is given by $||\nabla f|| = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{\frac{1}{2}}$ or $||\nabla f|| \cong \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$.
- Edge orientation is given by $\theta = tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$.

Examples of edge orientations






$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$
θ

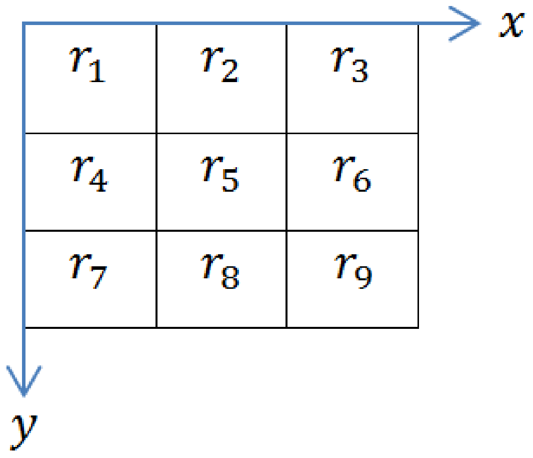
- Edge strength is given by $\|\nabla f\| = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{\frac{1}{2}}$ or $\|\nabla f\| \cong \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$.

- Edge orientation is given by $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

Imperial College London

Spatial masks for edge detection

- Consider an image region of size 3×3 pixels. The coordinates x, y are shown.
- The quantity r denotes grey level values.
- The magnitude $\|\nabla f\| = \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$ of the gradient at pixel with intensity r_5 can be approximated by:
  - ➢ differences along x and y
    $\|\nabla f\| = |r_5 - r_6| + |r_5 - r_8|$
  - ➢ cross-differences (along the two main diagonals)
  - ➢ $\|\nabla f\| = |r_6 - r_8| + |r_5 - r_9|$



Spatial masks for edge detection – Roberts operator

- The magnitude $\|\nabla f\| = \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$ of the gradient at pixel with intensity r_5 can be approximated by:
  - differences along x and y
    $\|\nabla f\| = |r_5 - r_6| + |r_5 - r_8|$
  - cross-differences (along the two main diagonals)
    $\|\nabla f\| = |r_6 - r_8| + |r_5 - r_9|$
- Each of the above approximation is the sum of the magnitudes of the responses of two masks. These sets of masks are the so called **Roberts operator**.

$$\text{abs}\left\{\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & -1 & 0 \\ \hline\end{array}\right\} + \text{abs}\left\{\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & -1 \\ \hline 0 & 0 & 0 \\ \hline\end{array}\right\} \qquad \text{abs}\left\{\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline 0 & -1 & 0 \\ \hline\end{array}\right\} + \text{abs}\left\{\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & -1 \\ \hline\end{array}\right\}$$

Imperial College London

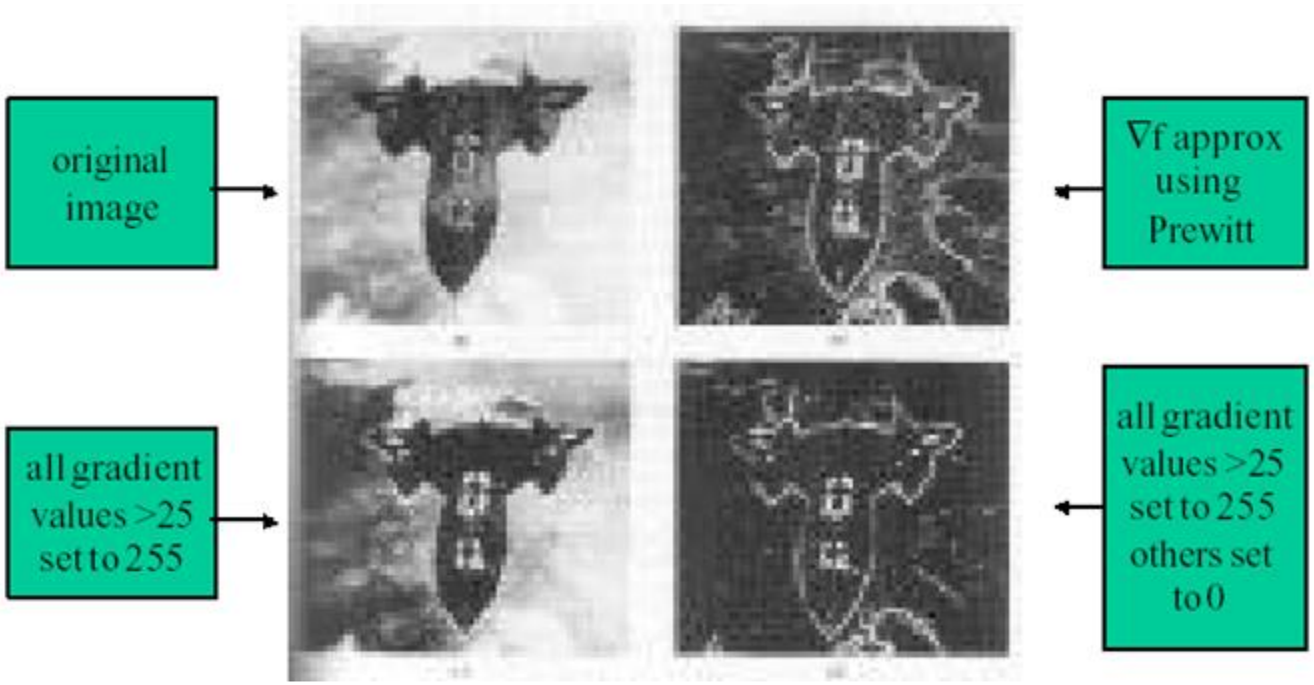
Spatial masks for edge detection – Prewitt and Sobel operators

- The magnitude $\|\nabla f\| = \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$ of the gradient at pixel with intensity r_5 can be also approximated by involving more pixels:
  - ➢ Roberts operator

    $\|\nabla f\| = |(r_3+r_6 + r_9) - (r_1 + r_4 + r_7)| + |(r_7+r_8 + r_9) - (r_1 + r_2 + r_3)|$
  - ➢ Sobel operator
  - ➢ $\|\nabla f\| = |(r_3+2r_6 + r_9) - (r_1 + 2r_4 + r_7)| + |(r_7+2r_8 + r_9) - (r_1 + 2r_2 + r_3)|$
- Each of the above approximation is the sum of the magnitudes of the responses of two masks.

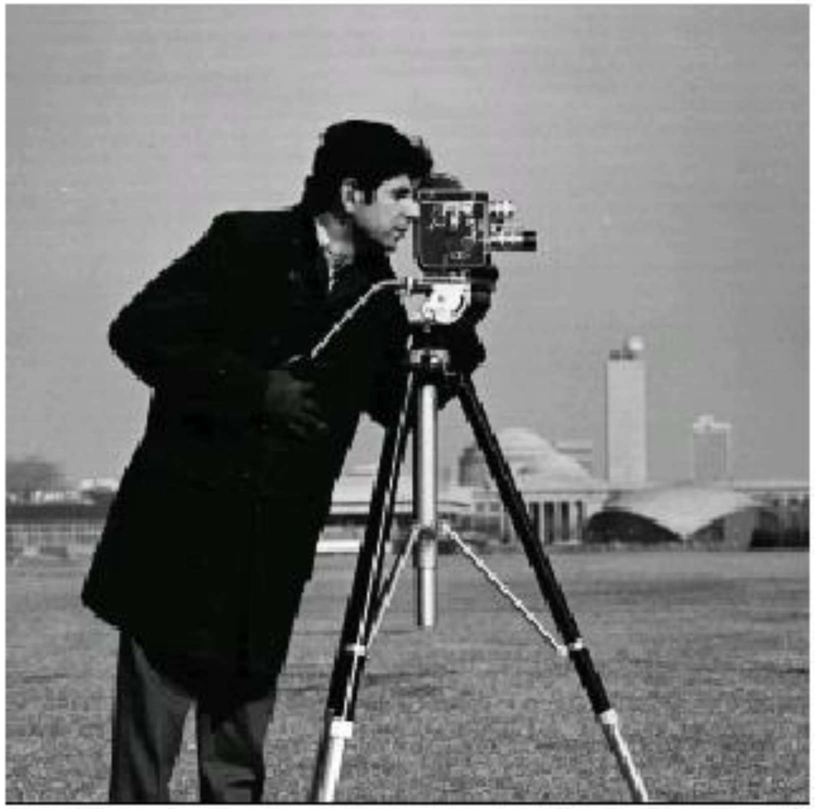
$$\text{abs}\{\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}\}+\text{abs}\{\begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array}\} \quad \text{abs}\{\begin{array}{|c|c|c|}\hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}\}+\text{abs}\{\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}\}$$

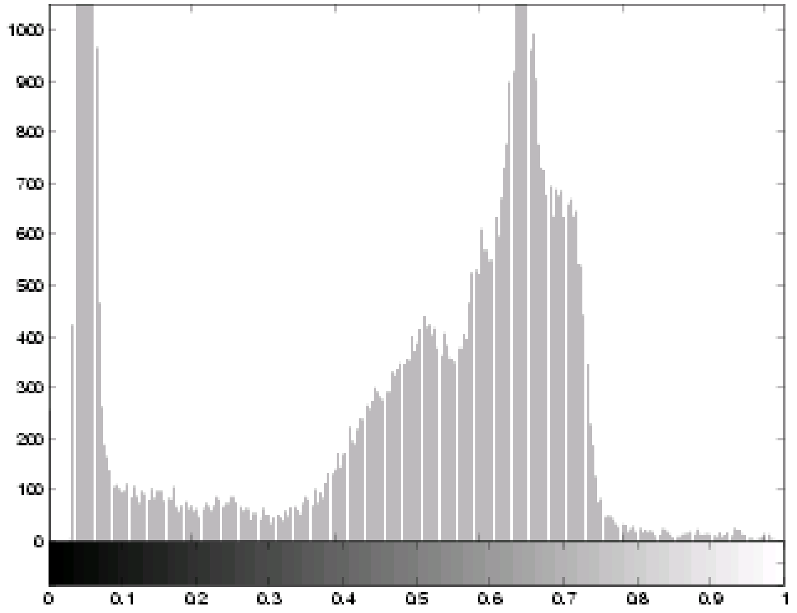
Example of edge detection using Prewitt


original image
all gradient values >25 set to 255
∇f approx using Prewitt
all gradient values >25 set to 255 others set to 0

Imperial College
London

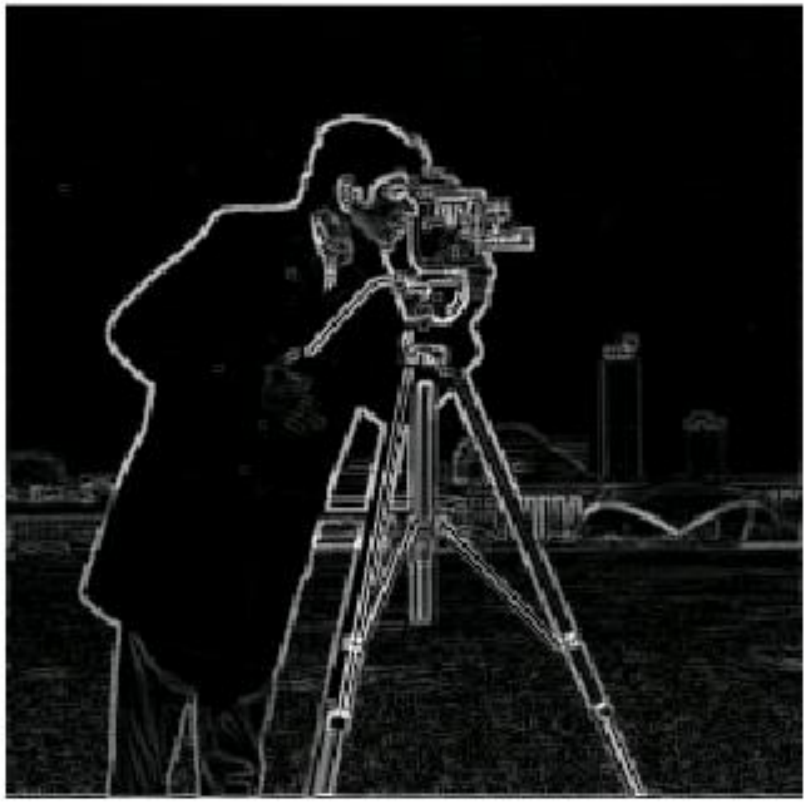
Original image and its histogram

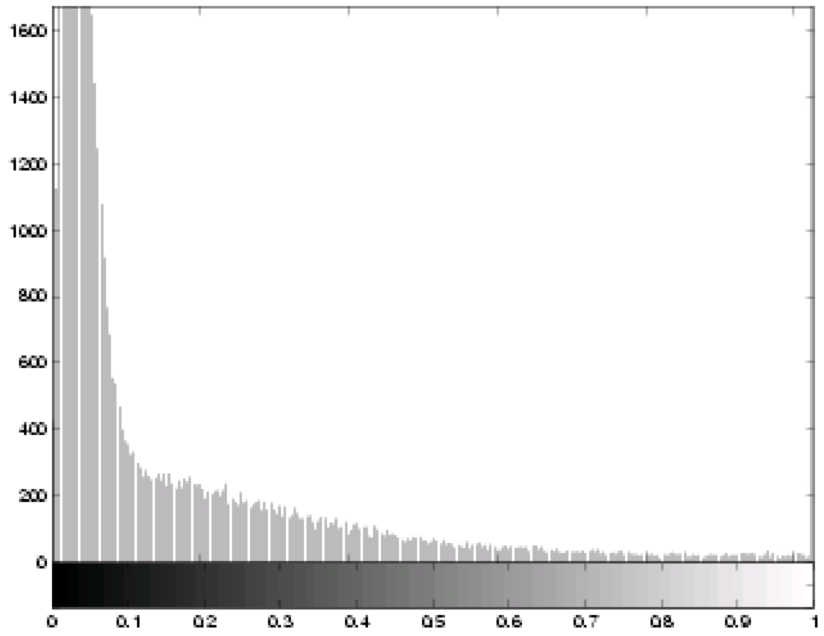




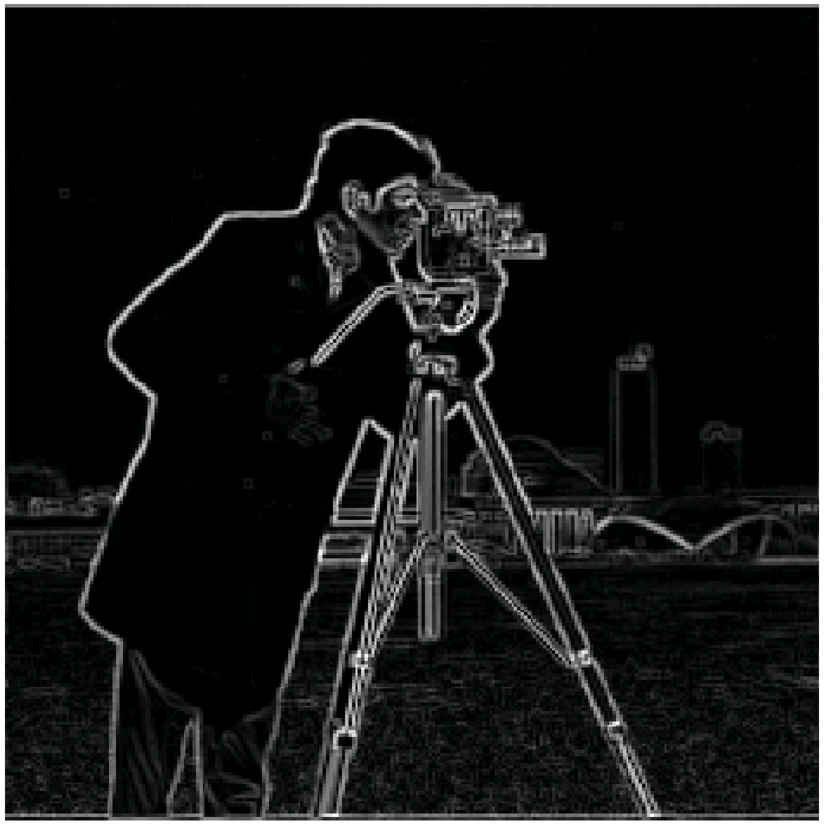
Imperial College London

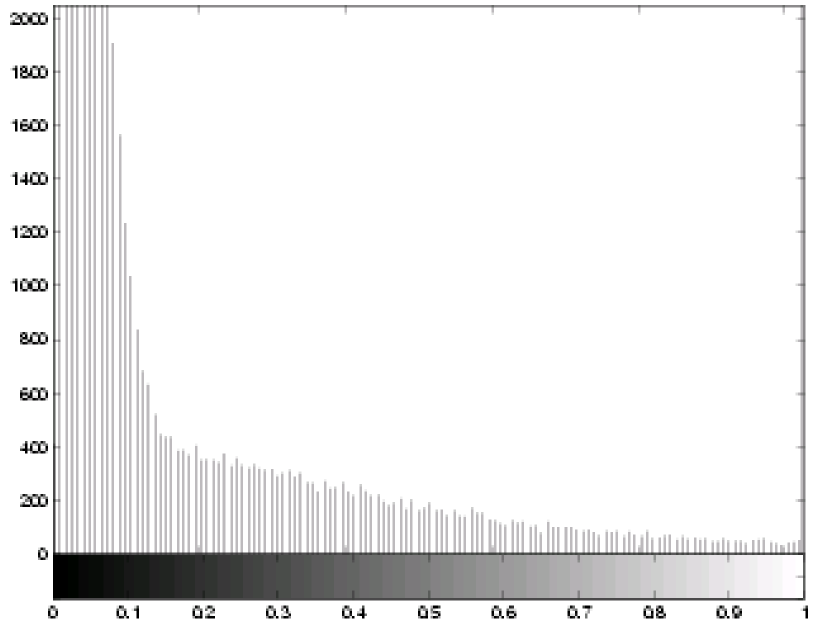
Example of edge detection using Prewitt





Example of edge detection using Sobel





Edge detection using second order gradient: Laplacian mask

- The second order gradient gives also information about how detailed is the area around a pixel. It is defined as:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- The most popular mask that can approximate the above function is the the so-called Laplacian. It has one of the two forms shown below.
- If we consider the mask on the right we see that if the input to this mask is $f(x,y)$ the output is
- $g(x,y) = 4f(x,y) - f(x,y-1) - f(x,y+1) - f(x-1,y) - f(x+1,y)$.

  We define $r(x,y) = f(x,y) - f(x,y-1)$ and $z(x,y) = f(x,y) - f(x-1,y)$.

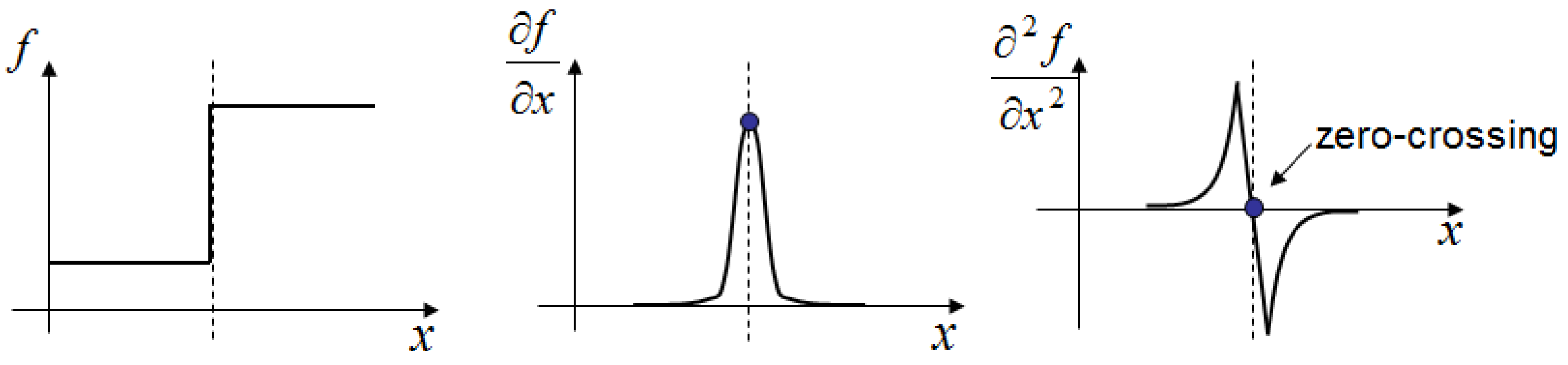
0	1	0
1	−4	1
0	1	0

0	−1	0
−1	4	−1
0	−1	0

Imperial College London

Edge detection using second order gradient: Laplacian mask cont.

- $g(x,y) = 4f(x,y) - f(x,y-1) - f(x,y+1) - f(x-1,y) - f(x+1,y) \Rightarrow$
  $= \big(f(x,y) - f(x,y-1)\big) - \big(f(x,y+1) - f(x,y)\big) + \big(f(x,y) - f(x-1,y)\big)$
  $- \big(f(x+1,y) - f(x,y)\big) = r(x,y) - r(x,y+1) + z(x,y) - z(x+1,y)$
  - $r(x,y) = f(x,y) - f(x,y-1)$: the 1st derivative of $f(x,y)$ with respect to y
  - $z(x,y) = f(x,y) - f(x-1,y)$: the 1st derivative of $f(x,y)$ with respect to x
  - $r(x,y) - r(x,y+1)$: the 1st derivative of $r(x,y)$ with respect to $y \Rightarrow$ the 2nd derivative of of $f(x,y)$ with respect to y
  - $z(x,y) - z(x+1,y)$: the 1st derivative of $z(x,y)$ with respect to $x \Rightarrow$ the 2nd derivative of of $f(x,y)$ with respect to x


f
x
$\frac{\partial f}{\partial x}$
x
$\frac{\partial^2 f}{\partial x^2}$
zero-crossing
x