博学慎思
明辨笃行

# Chapter 3
# Image Pre-processing

# Outline

- 3.0 Overview
- 3.1 Pixel brightness transformations
- 3.2 Geometric transformations
- 3.3 Local pre-processing
- 3.4 Image restoration

# 3.0 Overview

- Pre-processing is a common name for operations with images at the lowest level of abstraction -- both input and output are intensity images

- The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing.

# Why we need to know "digital image fundamentals" ?

Video of "cow tracking "

Image capture → Preprocessing → Segmentation → Model fitting → Model prediction → Conclusion (understanding)

A Video = … + … Many frames

track.mpg

A frame = an image!

Image processing + Image understanding !

Fig. Flowchart of "cow tracking"

# Why we need to learn “image Pre-processing” ?

Video of “cow tracking ”

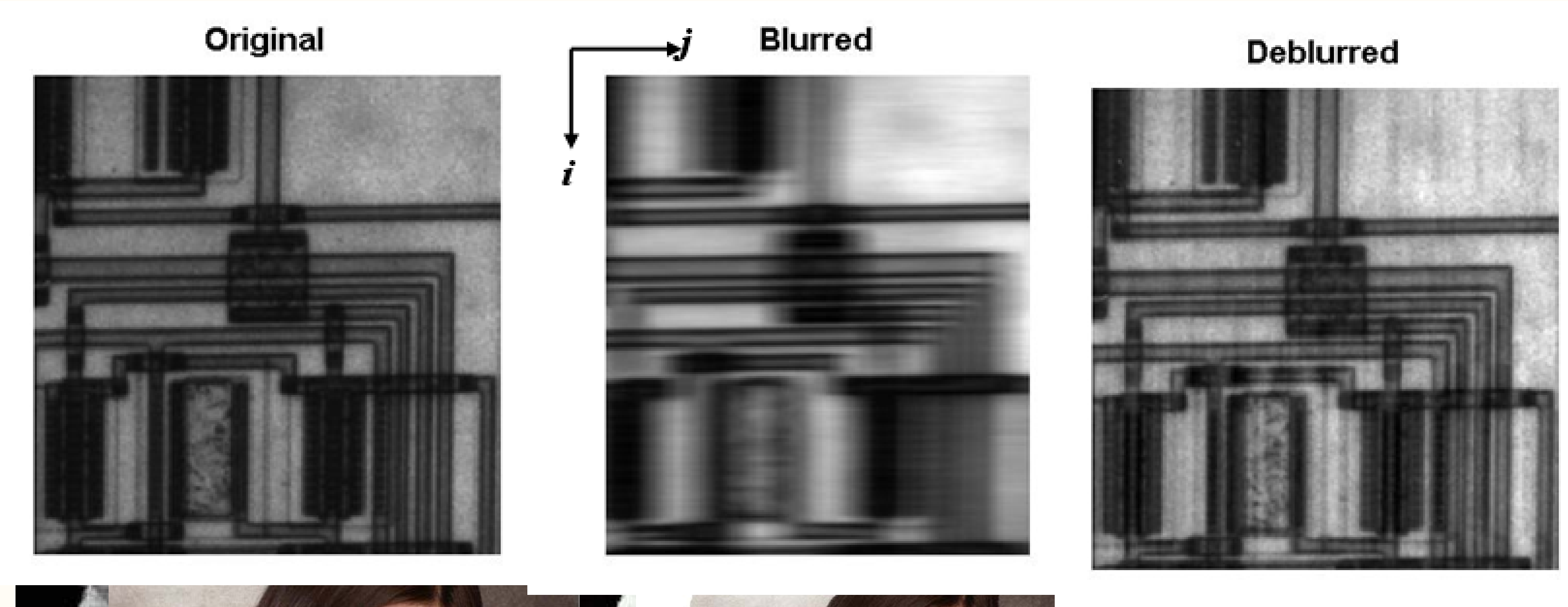
Image capture
↓
Preprocessing
↓
Segmentation
↓
Model fitting
↓
Model prediction
↓
Conclusion (understanding)

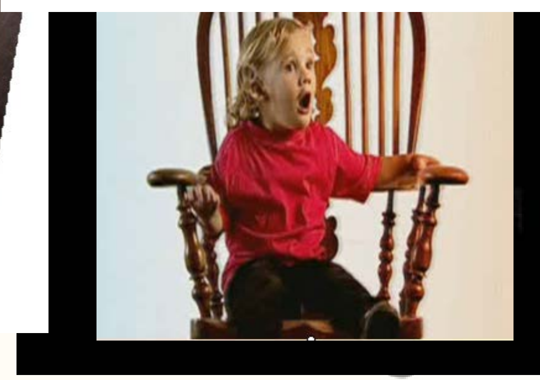
Fig. Flowchart of “cow tracking”

→ Image enhancement
Image restoration

→
1) pixel brightness transformations
2) geometric transformations
3) pre-processing methods
4) image restoration

# 3.0 Overview



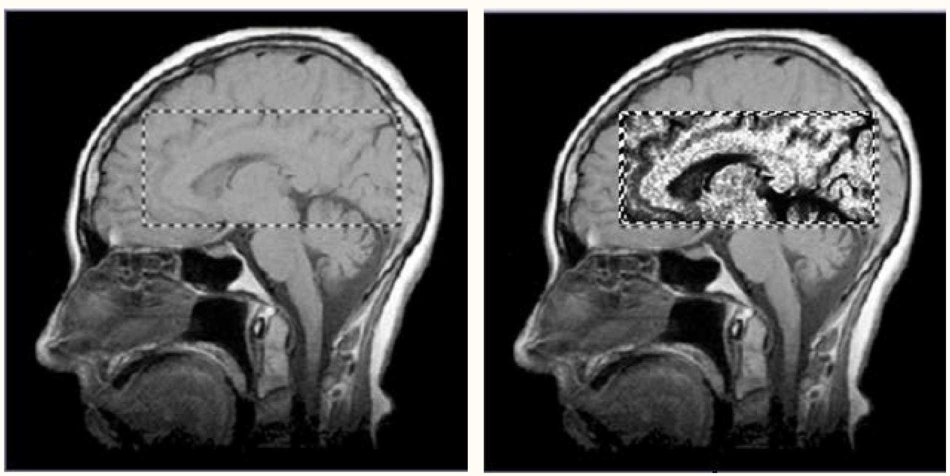



# 3.0 Overview

- Other classifications of image pre-processing methods exist.

  -- image enhancement,

  -- image restoration.



(a) (b)

# 3.0 Overview

**Why “distorted pixel can be restored as an average value of neighboring pixels”?**

- Image pre-processing methods use the considerable redundancy in images.
- Neighboring pixels corresponding to one object in real images have essentially the same or similar brightness value.
- Thus, distorted pixel can often be restored as an average value of neighboring pixels.

# 3.0 Overview

- If pre-processing aims to correct some degradation in the image, the nature of a priori information is important:

  knowledge about the nature of the degradation;

  knowledge about the properties of the image acquisition device, and conditions under which the image was obtained.

  knowledge about objects that are searched for in the image, which may simplify the pre-processing very considerably.

If knowledge about objects is not available in advance it can be estimated during the processing!

# 3.1 Pixel brightness transformations

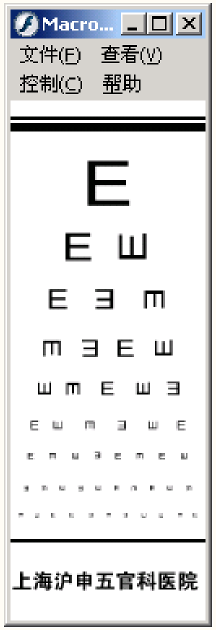
- Brightness transformations modify pixel brightness

  -- the transformation depends on the properties of a pixel itself.

- 1) Brightness correction
  - considers original brightness
  - pixel position in the image.
- 2) Gray scale transformations
  - change brightness without regard to position in the image.

# 3.1.1 Position dependent brightness correction

- Ideally, the sensitivity of image acquisition and digitization devices should not depend on position in the image, but this assumption is not valid in many practical cases.

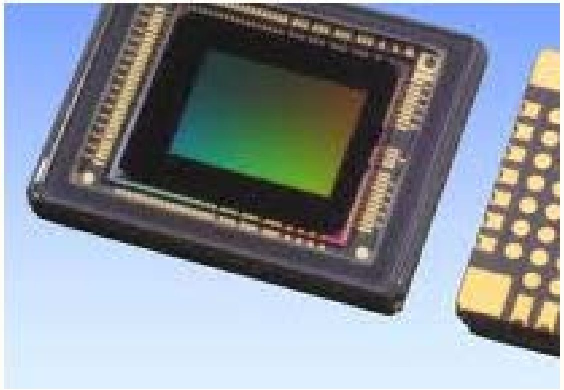



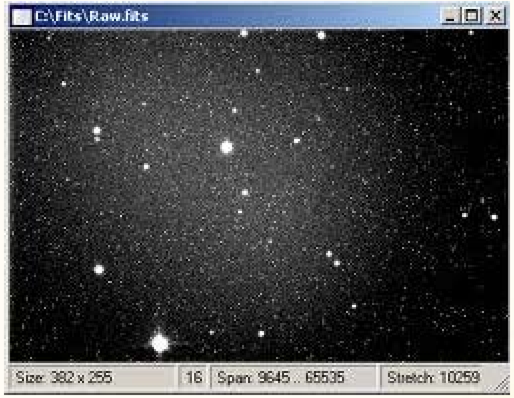
# 3.1.1 Position dependent brightness correction

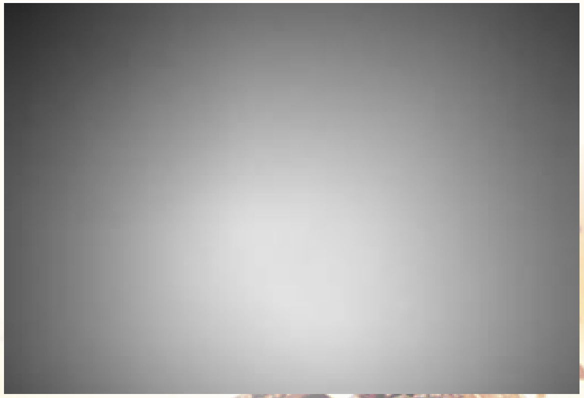
- Sources of degradation.

  Uneven sensitivity of light sensors

  Uneven object illumination

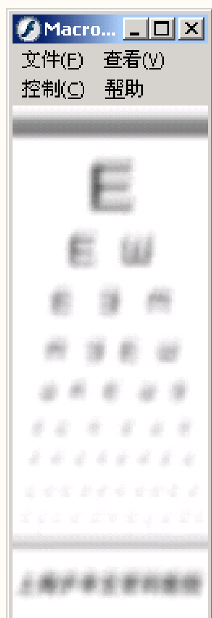


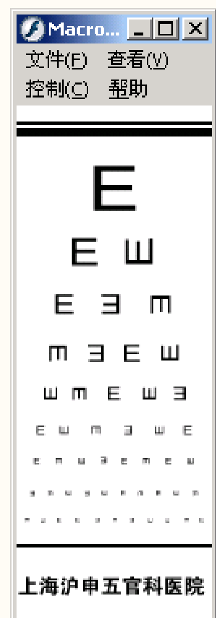


# 3.1.1 Position dependent brightness correction

- Systematic degradation can be suppressed by brightness correction.



# 3.1.1 Position dependent brightness correction

- Let a multiplicative error coefficient e(i,j) describe the change from the ideal identity transfer function;

  g(i,j) is the original undegraded image (or desired image)

  f(i,j) is the image containing degradation.

$$f(i,j) = e(i,j)\, g(i,j)$$

# 3.1.1 Position dependent brightness correction

- If a reference image g(i,j) is known (e.g., constant brightness c) then

  the degraded result is $f_c(i, j)$

  systematic brightness errors can be suppressed:

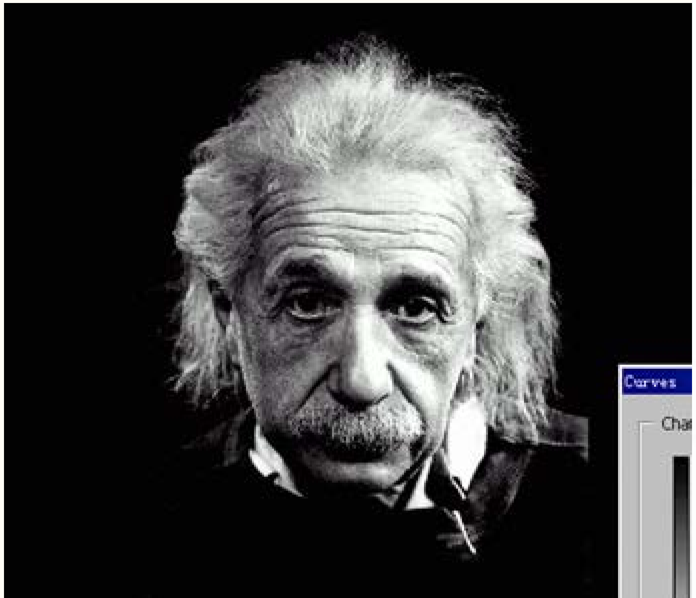
$$g(i,j) = \frac{f(i,j)}{e(i,j)} = \frac{c\, f(i,j)}{f_c(i,j)}$$

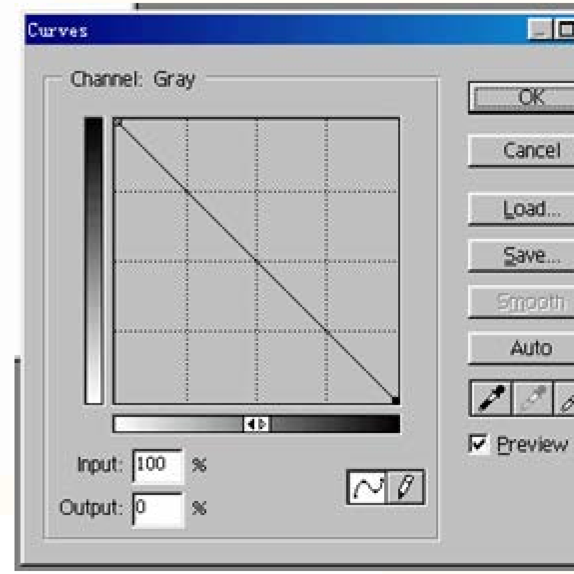
# 3.1.1 Position dependent brightness correction

- Image degradation process must be stable,

  the device should be calibrated time to time (find error coefficients e(i,j))

# mation



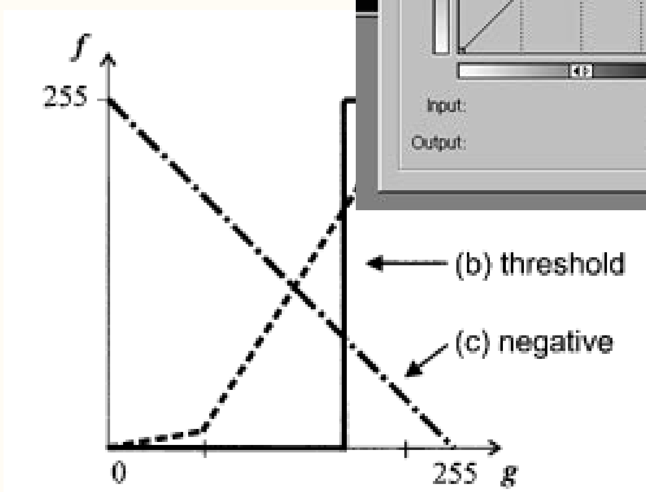
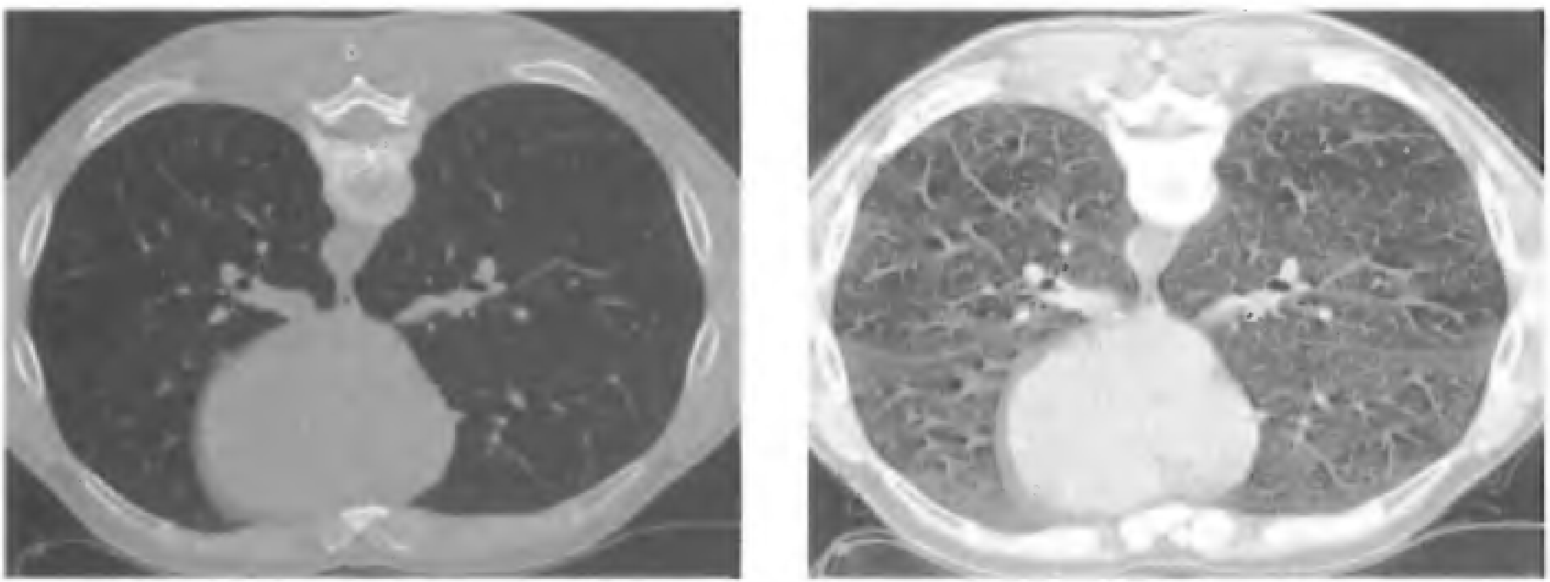




**Figure 5.1**: Perspective projection geometry examples.

Grey scale transformations are mostly used if the result is viewed by a human!

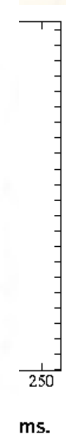
# 3.1.2 Grey scale transformation

- Typical grey level transform ... histogram equalization is usually found automatically
- The aim - image with equally distributed brightness levels over the whole brightness scale



(a) (b)

**Figure 5.3:** Histogram equalization. (a) Original image. (b) Equalized image.



# 3.1.2 Grey scale transformation

- The histogram can be treated as a discrete probability density function, look in the book for details.
- The monotonic property of the transform T implies

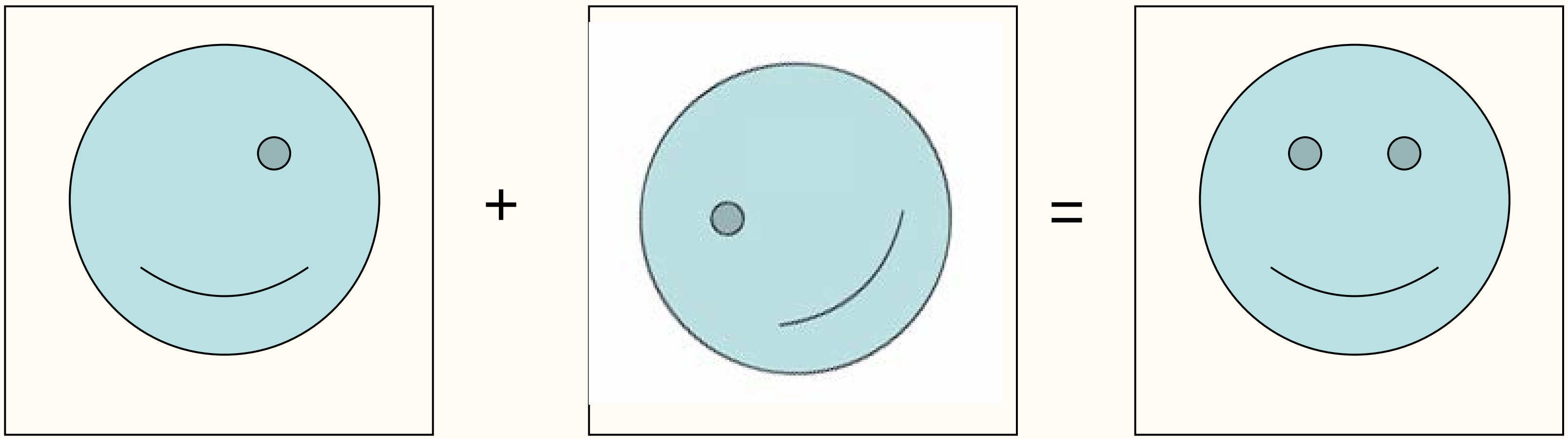
$$\sum_{i=0}^{k} G(q_i) = \sum_{i=0}^{k} H(p_i)$$

# 3.2 Geometric transformations

- Geometric transforms permit the elimination of geometric distortion that occurs when an image is captured.

# 3.2 Geometric transformations

- Geometric transforms permit the elimination of geometric distortion that occurs when an image is captured.




参考图像R　　浮动图像F　　参考图像R

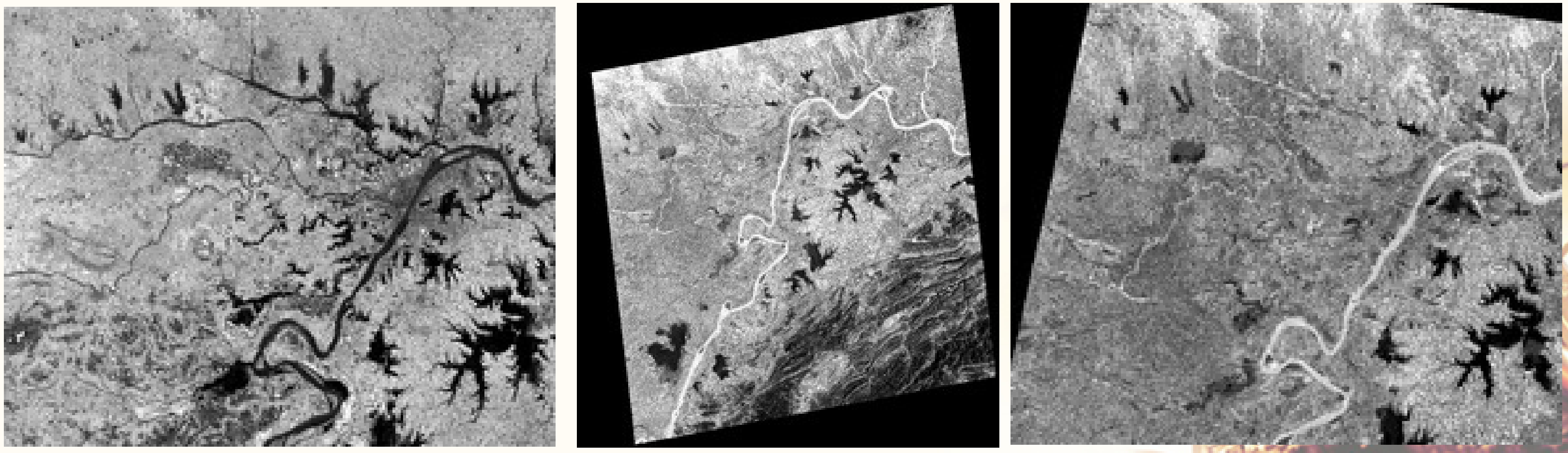
Image fusion

Image registration and fusion!

Image match!

# 3.2 Geometric transformations

- An example is an attempt to match remotely sensed images of the same area taken after one year, when the more recent image was probably not taken from precisely the same position.
- To inspect changes over the year, it is necessary first to execute a geometric transformation, and then subtract one image from the other.



# 3.2 Geometric transformations

- A geometric transform is a vector function T that maps the pixel (x,y) to a new position (x',y').

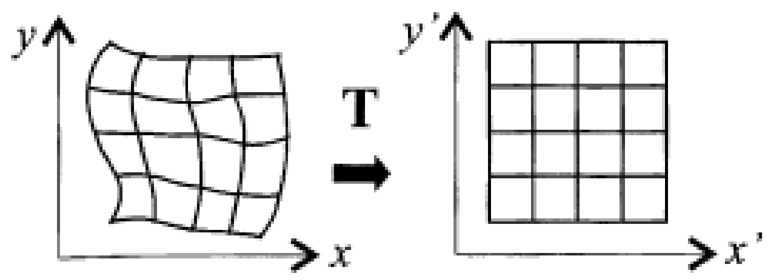



**Figure 5.5:** Geometric transform on a plane.

# 3.2 Geometric transformations

- The transformation equations are either known in advance or can be determined from known original and transformed images.
- Several pixels in both images with known correspondence are used to derive the unknown transformation.

$$x' = T_x(x, y), \qquad y' = T_y(x, y)$$

# 3.2 Geometric transformations

- A geometric transform consists of two basic steps ...
- 1. determining the pixel co-ordinate transformation

mapping of the co-ordinates of the input image pixel to the point in the output image.

the output point co-ordinates should be computed as continuous values (real numbers) as the position does not necessarily match the digital grid after the transform.

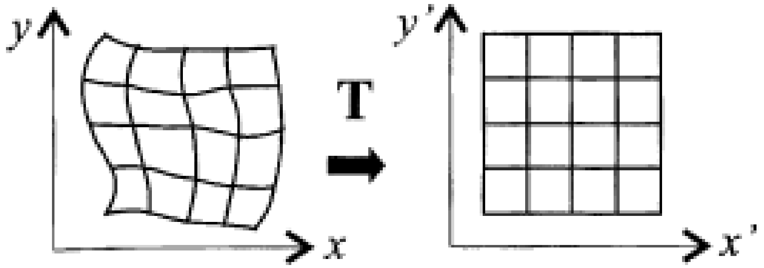



Figure 5.5: Geometric transform on a plane.

# 3.2 Geometric transformations

- A geometric transform consists of two basic steps ...
- 2. finding the point in the digital raster which matches the transformed point and determining its brightness.

  brightness is usually computed as an interpolation of the brightnesses of several points in the neighborhood.

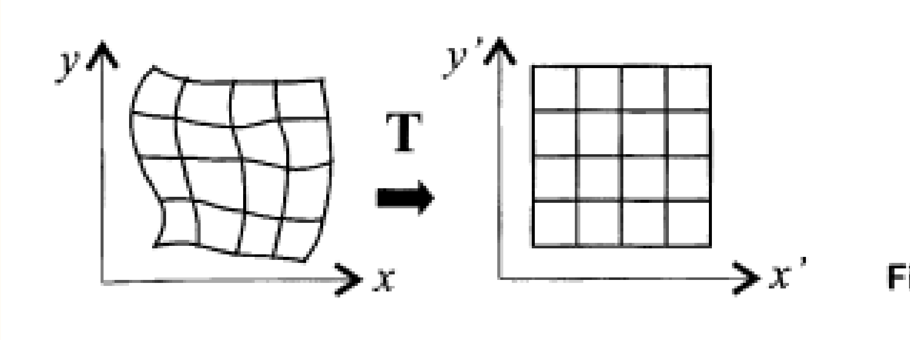



**Figure 5.5:** Geometric transform on a plane.

# 3.2.1 Pixel co-ordinate transformations

- General case of finding the co-ordinates of a point in the output image after a geometric transform.

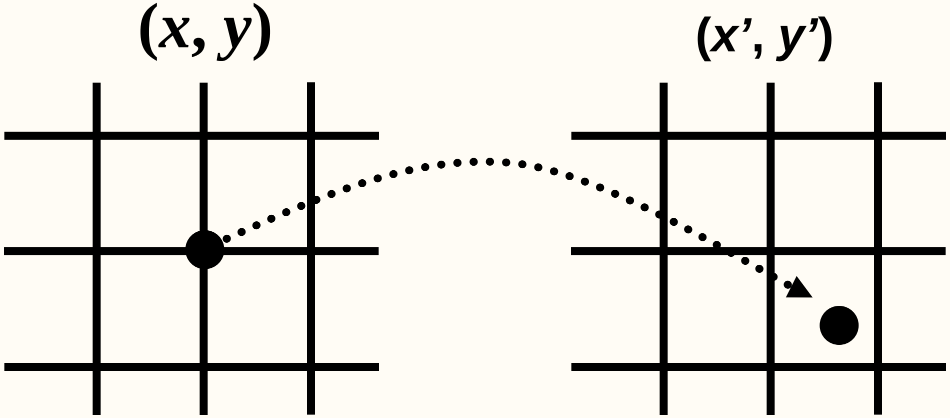
  usually approximated by a polynomial equation

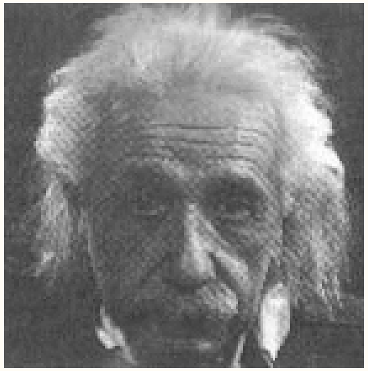
$$x' = \sum_{r=0}^{m} \sum_{k=0}^{m-r} a_{rk} \, x^r \, y^k \qquad y' = \sum_{r=0}^{m} \sum_{k=0}^{m-r} b_{rk} \, x^r \, y^k$$

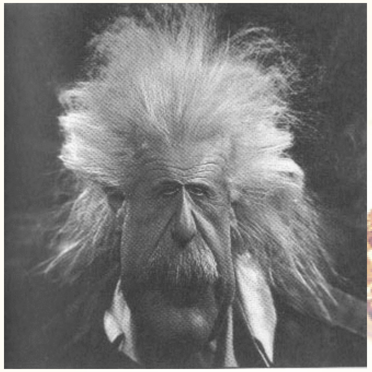
- This transform is linear with respect to the coefficients $a_{rk}$, $b_{rk}$

# 3.2.1 Pixel co-ordinate transformations

- If pairs of corresponding points (x,y), (x',y') in both images are known, it is possible to determine $a_{rk}$, $b_{rk}$ by solving a set of linear equations.





# 3.2.1 Pixel co-ordinate transformations

- More points than coefficients are usually used to get robustness.
- The higher the degree of the approximating polynomial, the more sensitive to the distribution of the pairs of corresponding points the geometric transform.

The computational cost increases!

# 3.2.1 Pixel co-ordinate transformations

- In practice, the geometric transform is often approximated by the bilinear transformation

  4 pairs of corresponding points are sufficient to find transformation coefficients

$$x' = a_0 + a_1 x + a_2 y + a_3 xy$$
$$y' = b_0 + b_1 x + b_2 y + b_3 xy$$

# 3.2.1 Pixel co-ordinate transformations

- Even simpler is the **affine transformation** for which three pairs of corresponding points are sufficient to find the coefficients.

$$x' = a_0 + a_1 x + a_2 y$$
$$y' = b_0 + b_1 x + b_2 y$$

# 3.2.1 Pixel co-ordinate transformations

- The affine transformation includes typical geometric transformations such as

  rotation, translation, scaling and skewing.

# 3.2.1 Pixel co-ordinate transformations

- Important geometric transformations
- Rotation - by the angle phi about the origin

$$\begin{aligned} x' &= x \cos\phi + y \sin\phi \\ y' &= -x \sin\phi + y \cos\phi \\ J &= 1 \end{aligned}$$

- Change of scale - a in the x axis and b in the y axis

$$\begin{aligned} x' &= ax \\ y' &= bx \\ J &= ab \end{aligned}$$

# 3.2.1 Pixel co-ordinate transformations

- Skewing by the angle $\phi$

$$\begin{aligned} x' &= x + y \tan\phi \\ y' &= y \\ J &= 1 \end{aligned}$$

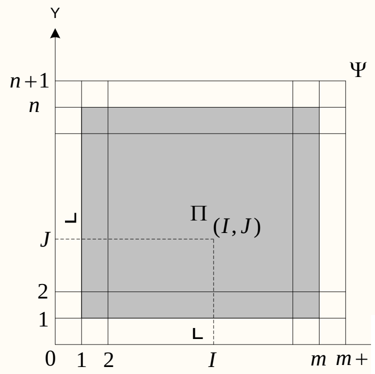
# 3.2.1 Pixel co-ordinate transformations

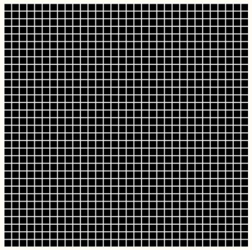
- Complex geometric transformations (distortion)

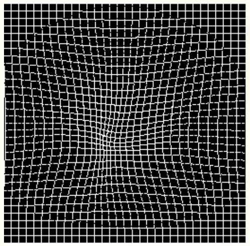
  approximation by partitioning an image into smaller rectangular subimages;

  for each subimage, a simple geometric transformation, such as the affine, is estimated using pairs of corresponding pixels.

  geometric transformation (distortion) is then performed separately in each subimage.



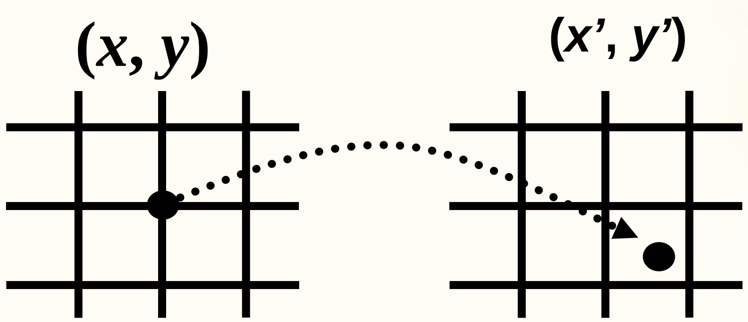





Think about ”Free-Form Deformation ” !

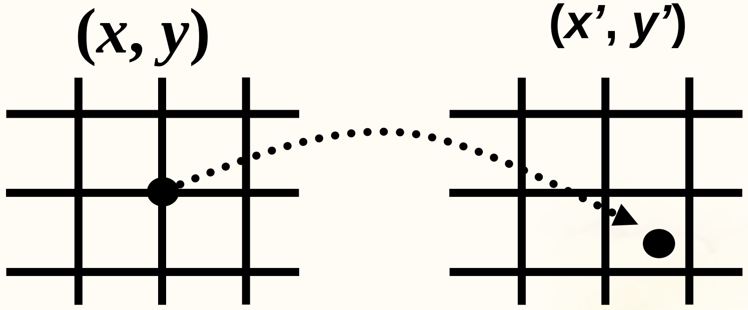
# 3.2.2 Brightness interpolation

- Assume that the planar transformation has been accomplished, and new point co-ordinates (x',y') were obtained.
- The position of the point does not in general fit the discrete raster of the output image.
- Values on the integer grid are needed.
- Each pixel value in the output image raster can be obtained by **brightness interpolation** of some neighboring noninteger samples.



# 3.2.2 Brightness interpolation

- The brightness interpolation problem is usually expressed in a dual way (by determining the brightness of the original point in the input image that corresponds to the point in the output image lying on the discrete raster).

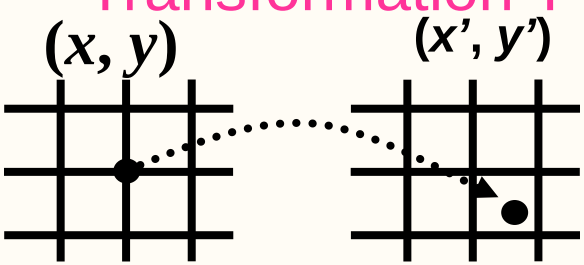


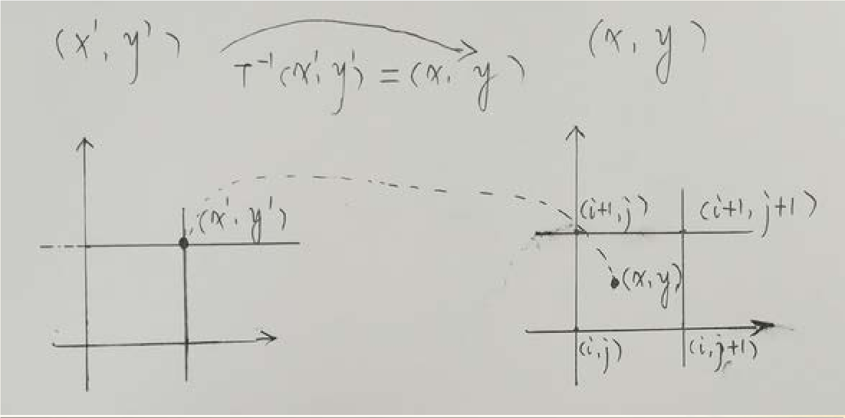
# 3.2.2 Brightness interpolation

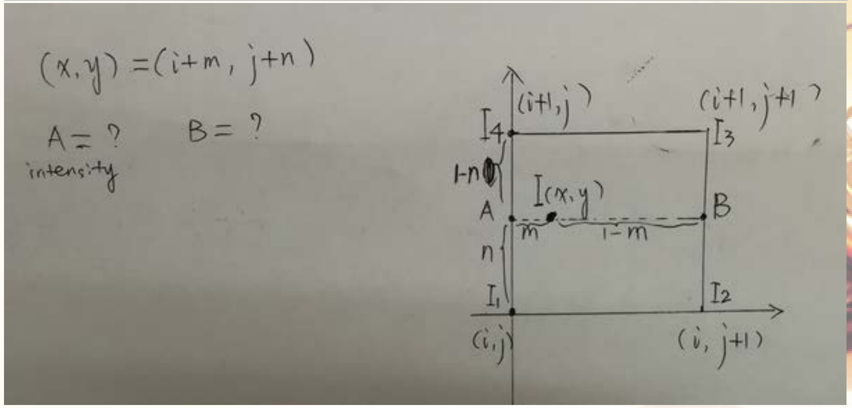
- Computing the brightness value of the pixel (x',y') in the output image where x' and y' lie on the discrete raster.

$$(x, y) = \mathbf{T}^{-1}(x', y')$$

Transformation T !



# 3.2.2 Brightness interpolation

- In general the real co-ordinates after inverse transformation (dashed lines in Figures) do not fit the input image discrete raster (solid lines), and so brightness is not known.

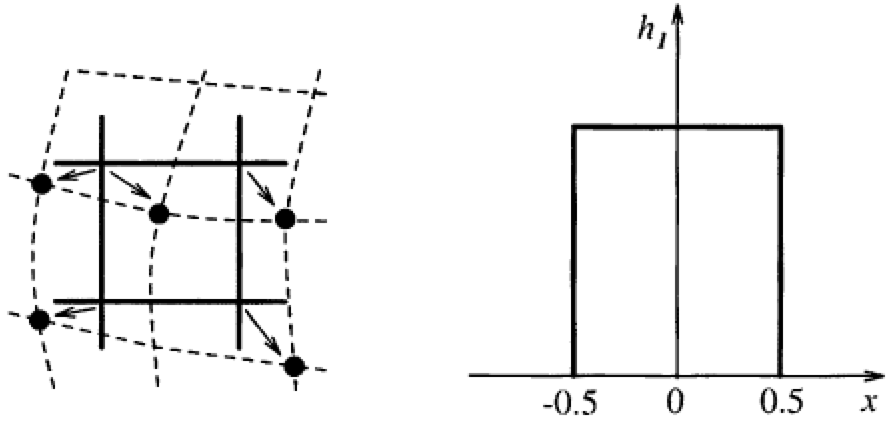



**Figure 5.7**: Nearest-neighborhood interpolation. The discrete raster of the original image is depicted by the solid line.

Not integer !

# 3.2.2 Brightness interpolation

- To get the brightness value of the point (x,y) the input image is resampled.

$$f_n(x,y) = \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} g_s(l\,\Delta x,\ k\,\Delta y)\, h_n(x - l\,\Delta x,\ y - k\,\Delta y)$$

- $f_n(x,y)$ ... result of interpolation
- $h_n$ is the **interpolation kernel**
  – Usually, a small neighborhood is used, outside which $h_n$ is zero.

That is why “we need to study the knowledge – ‘interpolation’ ”!

# 3.2.2 Neighbor interpolation methods

- -Nearest neighbor interpolation
- -Linear interpolation
- -Bicubic interpolation

# 3.2.2 -Nearest neighbor interpolation

- assigns to the point (x,y) the brightness value of the nearest point g in the discrete raster

$$f_1(x,y) = g_s(round\,(x),\, round\,(y))$$

# 3.2.2 -Nearest neighbor interpolation

- The right side of Figure shows how the new brightness is assigned.
- Dashed lines show how the inverse planar transformation maps the raster of the output image into the input image - full lines show the raster of the input image.

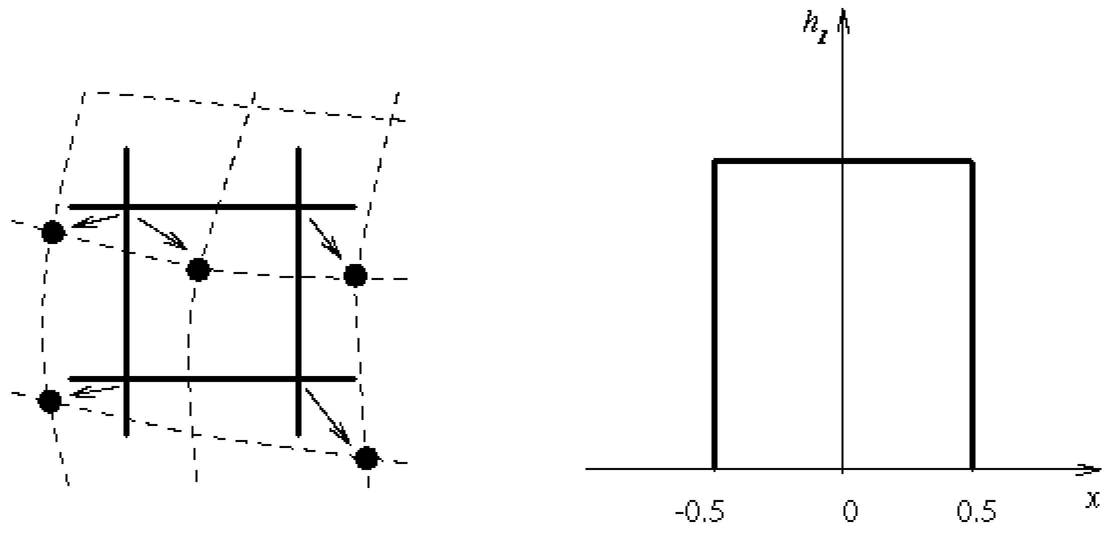



Figure 4.7 Nearest neighbourhood interpolation.

# 3.2.2 -Nearest neighbor interpolation

- The position error of the nearest neighborhood interpolation is at most half a pixel.
- This error is perceptible on objects with straight line boundaries that may appear step-like after the transformation.

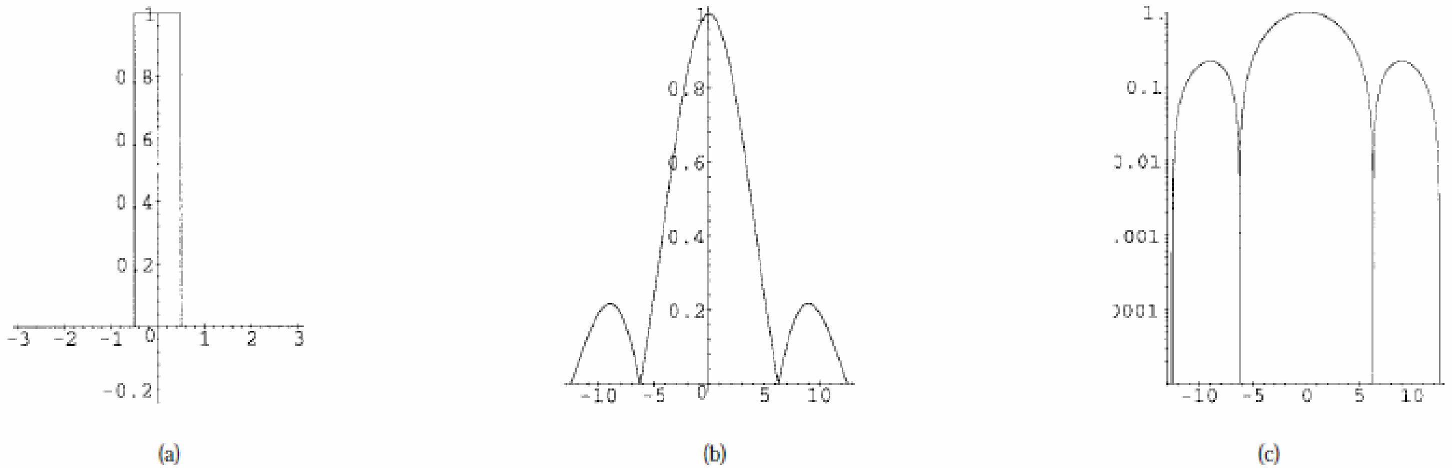



Fig. 9. Nearest neighbor interpolation. (a) Kernel. (b) Magnitude of Fourier transform. (c) Logarithmic plot of magnitude.

# 3.2.2 -Linear interpolation

- Explores four points neighboring the point (x,y), and assumes that the brightness function is linear in this neighborhood.

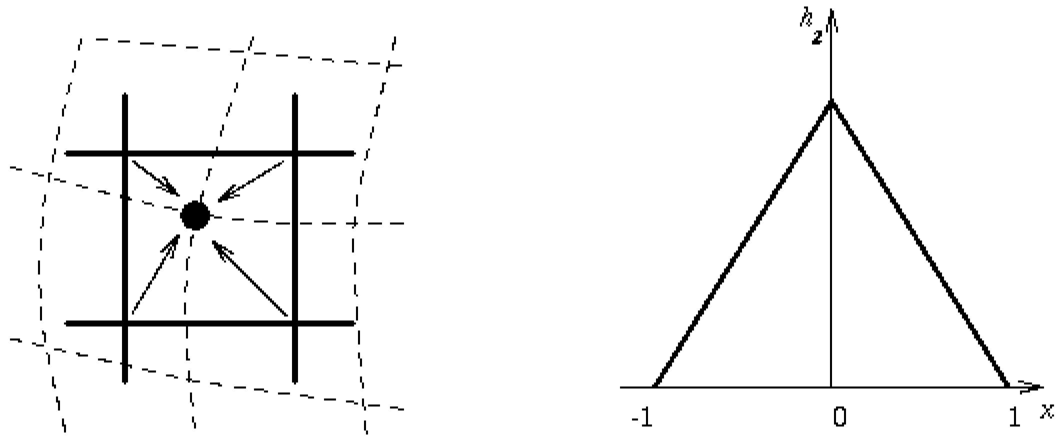



Figure 4.8 Linear interpolation.

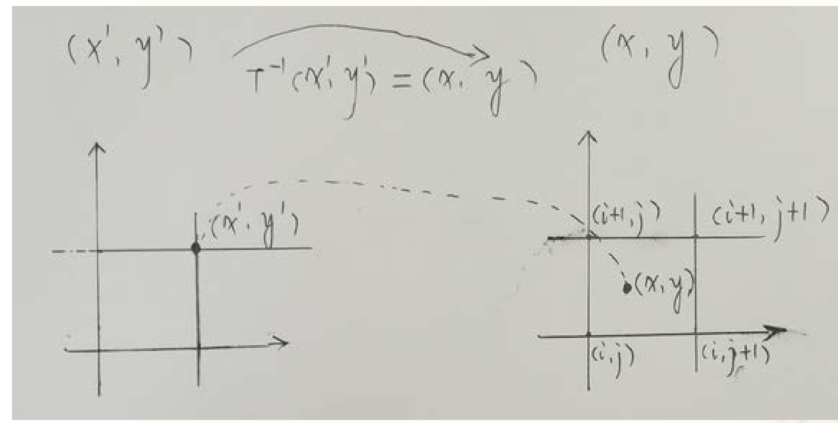
# 3.2.2 -Linear interpolation

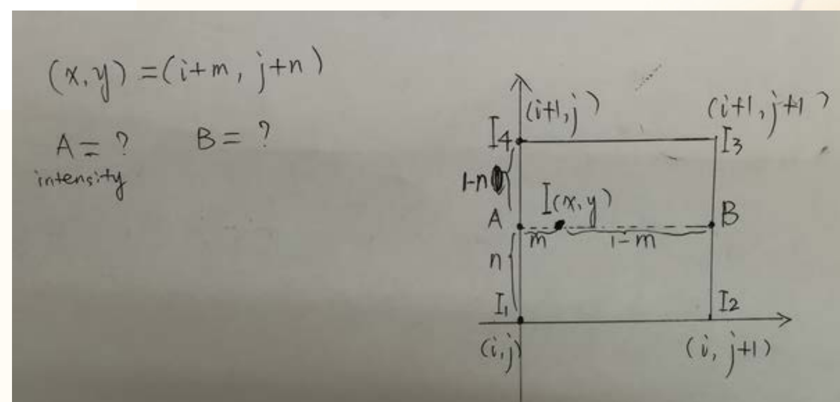
- Linear interpolation is given by the equation.

$$
\begin{aligned}
f_2(x, y) = \;& (1-a)(1-b)\, g_s(l, k) \\
& + a(1-b)\, g_s(l+1, k) \\
& + b(1-a)\, g_s(l, k+1) \\
& + ab\, g_s(l+1, k+1)
\end{aligned}
$$

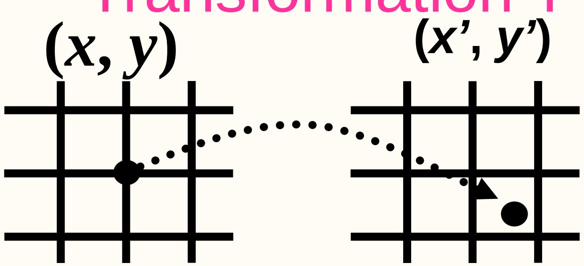
$$
l = floor\,(x), \qquad a = x - l
$$

$$
k = floor\,(y), \qquad b = y - k
$$




Transformation T !



# 3.2.2 -Linear interpolation

- Linear interpolation can cause a small decrease in resolution and blurring due to its averaging nature.
- The problem of step like straight boundaries with the nearest neighborhood interpolation is reduced.

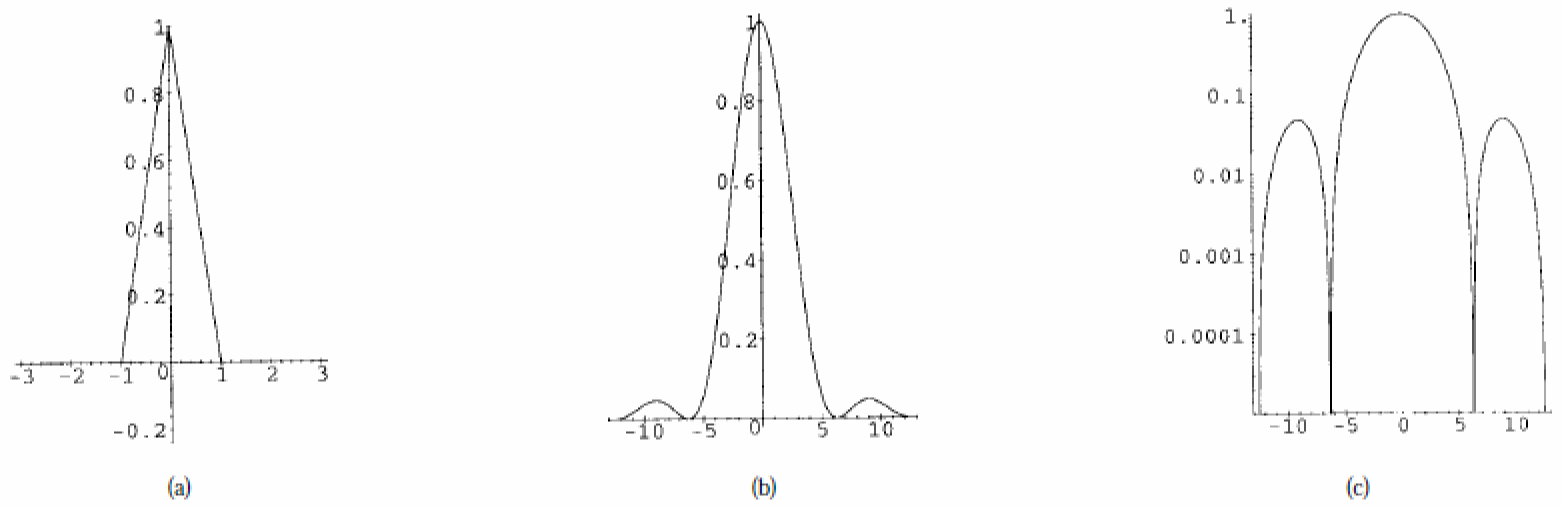


Fig. 10. Linear interpolation. (a) Kernel. (b) Magnitude of Fourier transform. (c) Logarithmic plot of magnitude.

# 3.2.2 -Bicubic interpolation

- improves the model of the brightness function by approximating it locally by a bicubic polynomial surface; sixteen neighboring points are used for interpolation.
- interpolation kernel (`Mexican hat') is given by

$$h_3 = \begin{cases} 1 - 2|x|^2 + |x|^3 & \text{for } 0 < |x| < 1 \\ 4 - 8|x| + 5|x|^2 - |x|^3 & \text{for } 1 < |x| < 2 \\ 0 & \text{otherwise} \end{cases}$$

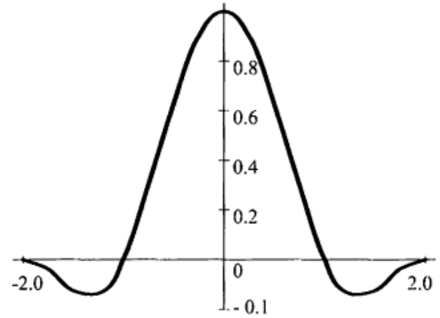



**Figure 5.9**: Bi-cubic interpolation kernel.

# 3.2.2 -Bicubic interpolation

- Bicubic interpolation does not suffer from the step-like boundary problem of nearest neighborhood interpolation, and copes with linear interpolation blurring as well.

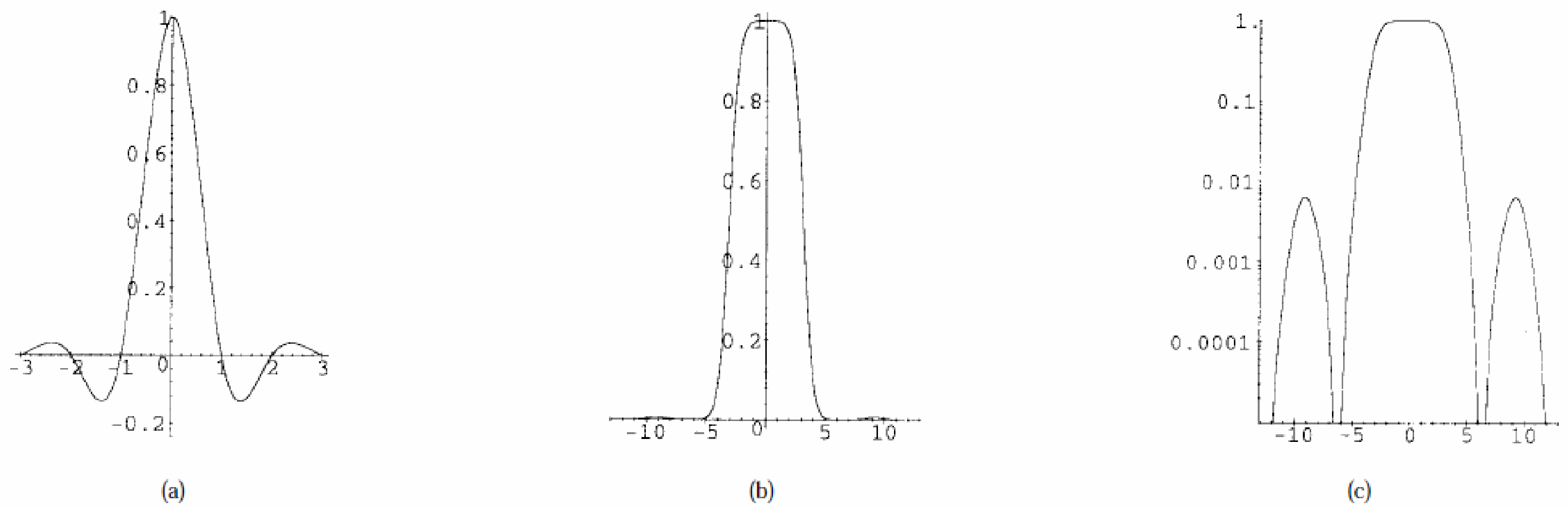
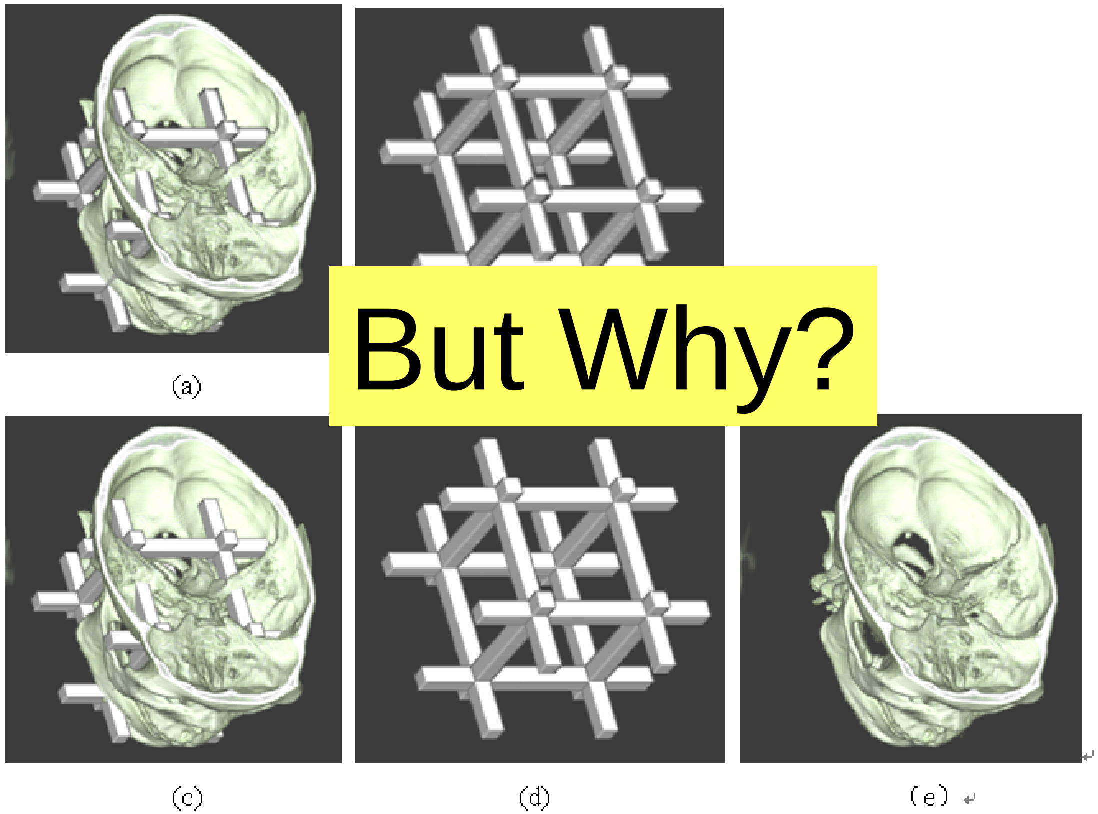



Fig. 14. Cubic B-spline interpolation. (a) Kernel plotted for $|x| < 3$. (b) Magnitude of Fourier transform. (c) Logarithmic plot of magnitude.

# 3.2.2 -Bicubic interpolation

- Bicubi … r displays that e … bitrary point … d were used, … increase.
- Bicubi … in the image




But Why?

图 6 关于不同体数据的混合体绘制的图像

Fig.6 The reconstructed image by hybrid volume rendering method combining different kinds of volume data

PLS read the paper" Survey: Interpolation Methods in Medical Image Processing". IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. 18, NO. 11, NOVEMBER 1999

**Review!**

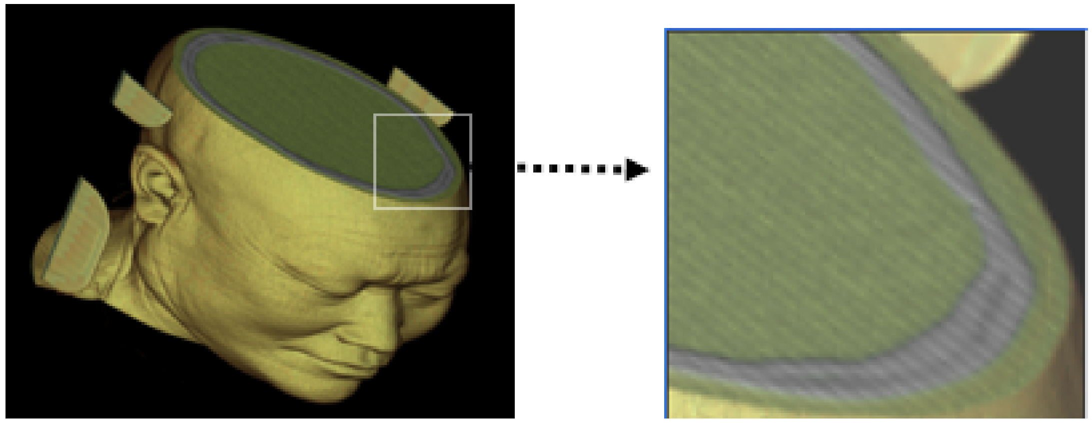
# 2.1.2 -Sampling

- A continuous image function f(x,y) can be sampled using a discrete grid of sampling points in the plane.
- The image is sampled at points x = j ( Delta_x), y = k (Delta_y)
- Two neighboring sampling points are separated by distance Delta_x along the x axis and Delta_y along the y axis. Distances Delta_x and Delta_y are called the **sampling interval** and the matrix of samples constitutes the discrete image

Review!

# 2.2.2 -Sampling

- Periodic repetition of the Fourier transform result F(u,v) may under certain conditions cause distortion of the image which is called **aliasing**; this happens when individual digitized components F(u,v) overlap.

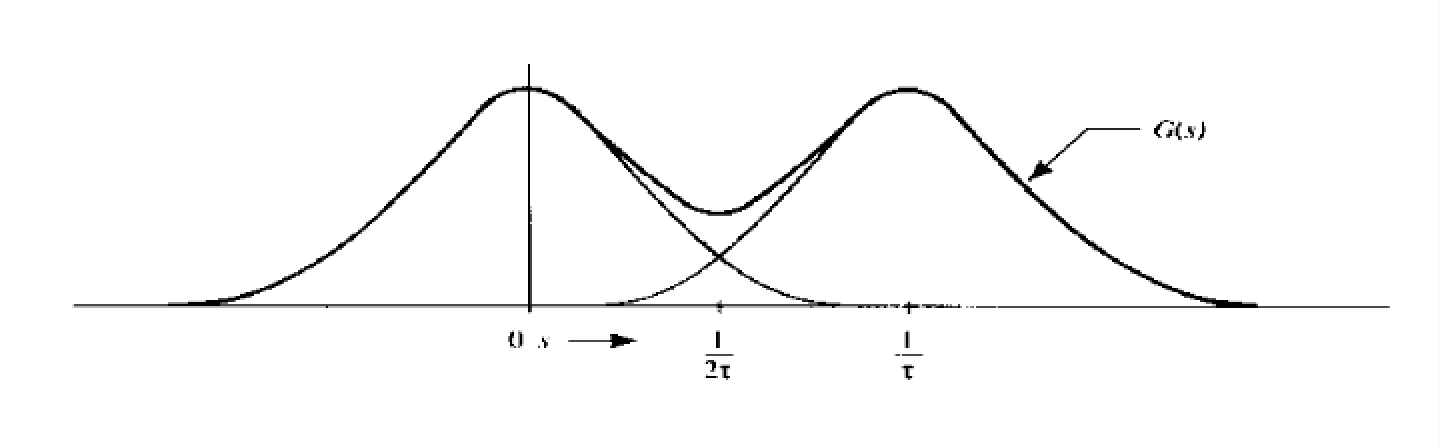


(a)整体图
(a) Whole image

(b)局部放大图
(b)Local zooming of Fig.4-9a

Streak
Aliasing
artifact

# 2.2.2 -Sampling




Overlapped!

**PLS review the knowledge points**
**In the course “digital signal processing”!**

Review!

# 2.2.2 -Sampling

- There is no aliasing if the image function f(x,y) has a band limited spectrum ... its Fourier transform F(u,v)=0 outside a certain interval of frequencies |u| > U; |v| > V.

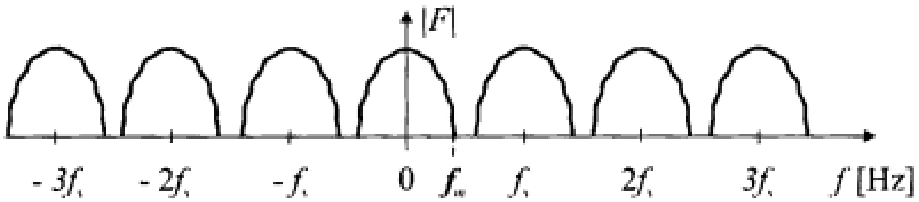



**Figure 3.10**: Repeated spectra of the 1D signal due to sampling. Non-overlapped case when $f_s \geq 2f_m$.

**PLS review the knowledge points**
**In the course "digital signal processing"!**

**Review!**

# 2.2.2 -Sampling

- As you know from general sampling theory, overlapping of the periodically repeated results of the Fourier transform F(u,v) of an image with band limited spectrum can be prevented if the sampling interval is chosen according to Eq. 3.43, pp63

$$\Delta x < \frac{1}{2U}, \Delta y < \frac{1}{2V}$$

Review!

# 2.2.2 -Sampling

- This is the **Shannon sampling theorem** that has a simple physical interpretation in image analysis: The sampling interval should be chosen in size such that it is less than or equal to half of the smallest interesting detail in the image.

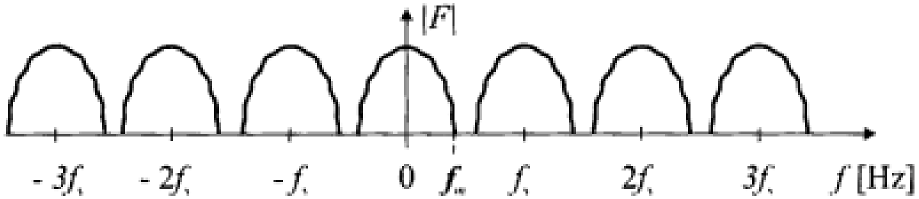
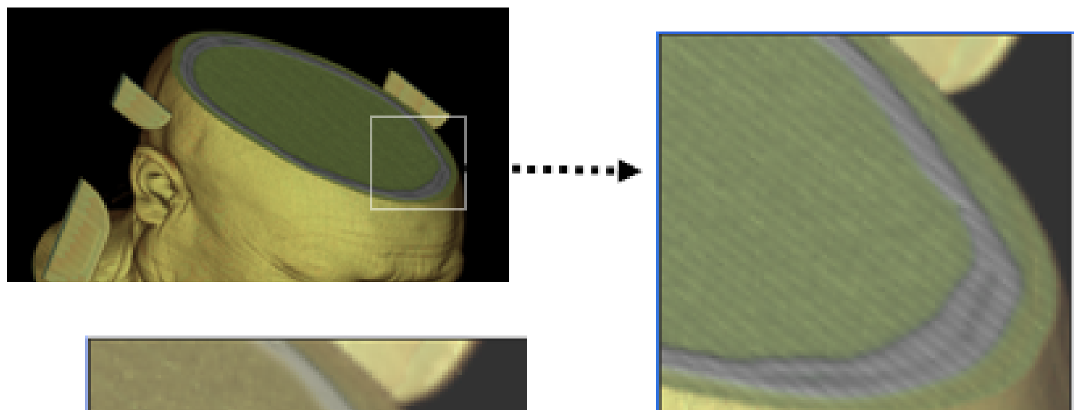



**Figure 3.10**: Repeated spectra of the 1D signal due to sampling. Non-overlapped case when $f_s \geq 2f_m$.

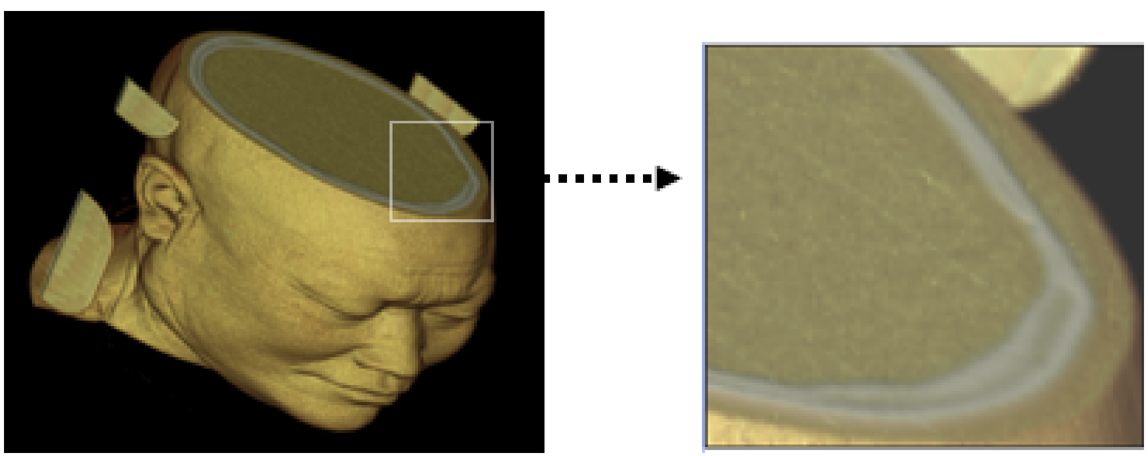
Review!

# 2.2.2 -Sampling

- Practical examples of digitization help to understand the reality of sampling.



(b)局部放大图
(b)Local zooming of Fig.4-9a



(a)整体图
(a)Whole image

(b)局部放大图
(b)Local zooming of Fig.4-10a

Streak
Aliasing
artifact

# Images are intrinsically singals!

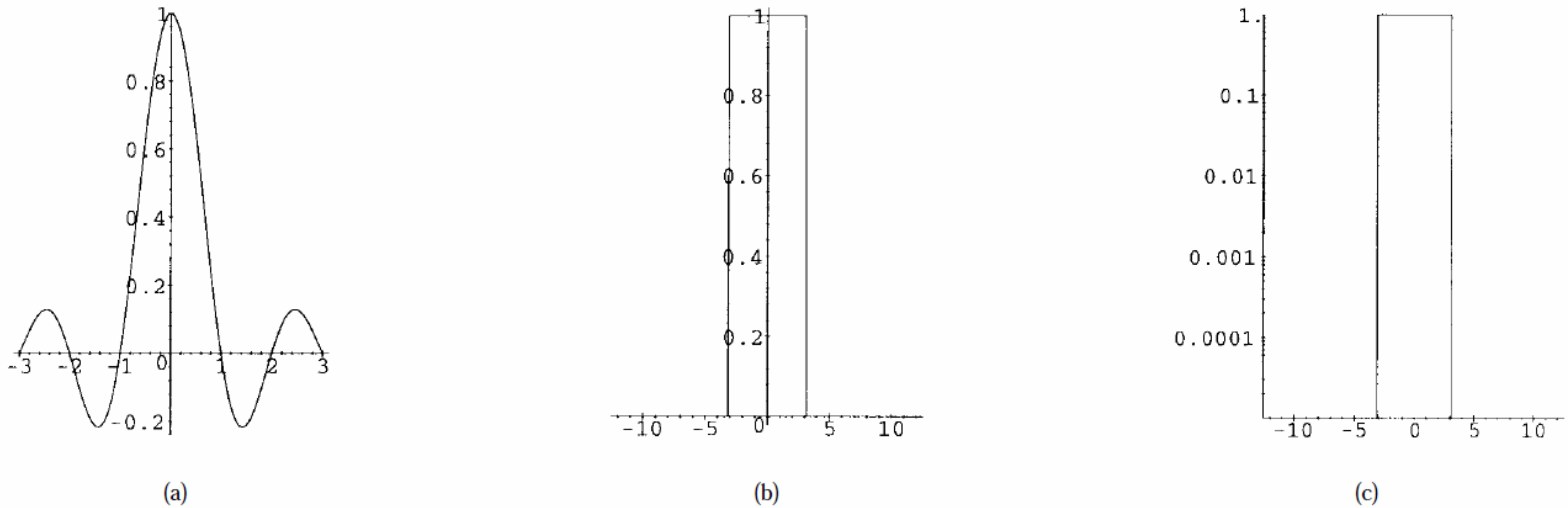



Fig. 5. Ideal interpolation. (a) Kernel plotted for $|x| < 3$. (b) Magnitude of Fourier transform. (c) Logarithmic plot of magnitude.

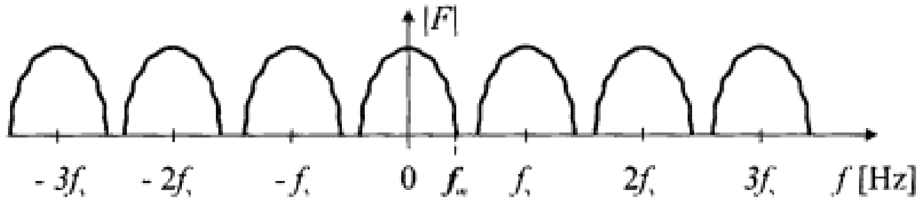



**Figure 3.10**: Repeated spectra of the 1D signal due to sampling. Non-overlapped case when $f_s \geq 2f_m$.

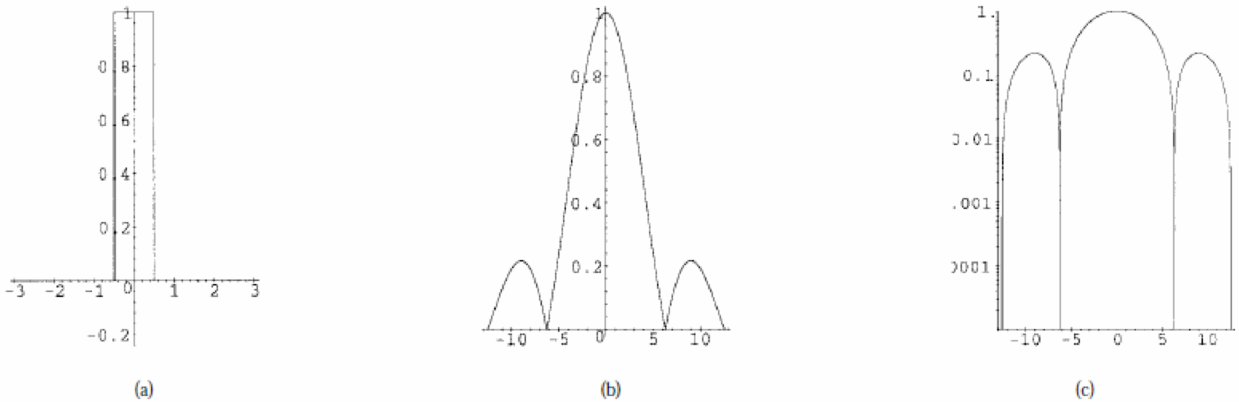
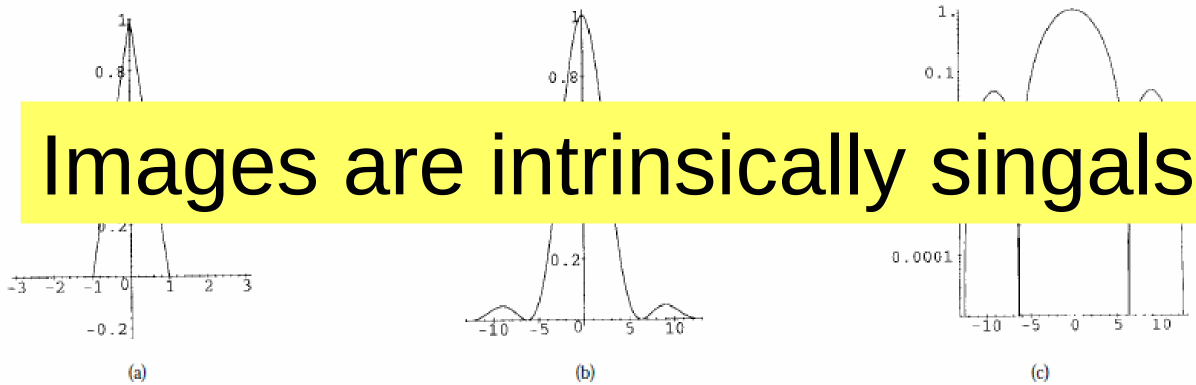



Fig. 9. Nearest neighbor interpolation. (a) Kernel. (b) Magnitude of Fourier transform. (c) Logarithmic plot of magnitude.




**Images are intrinsically singals!**

Fig. 10. Linear interpolation. (a) Kernel. (b) Magnitude of Fourier transform. (c) Logarithmic plot of magnitude.

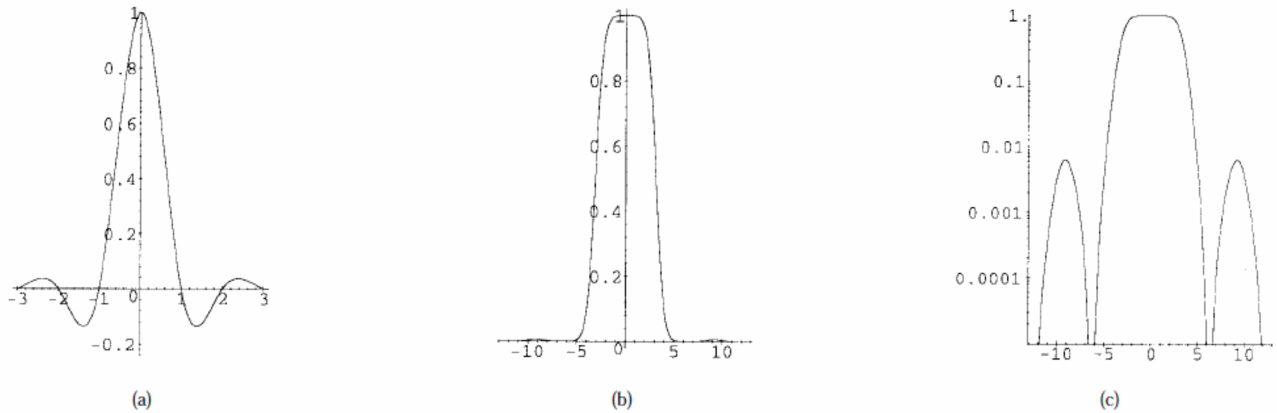
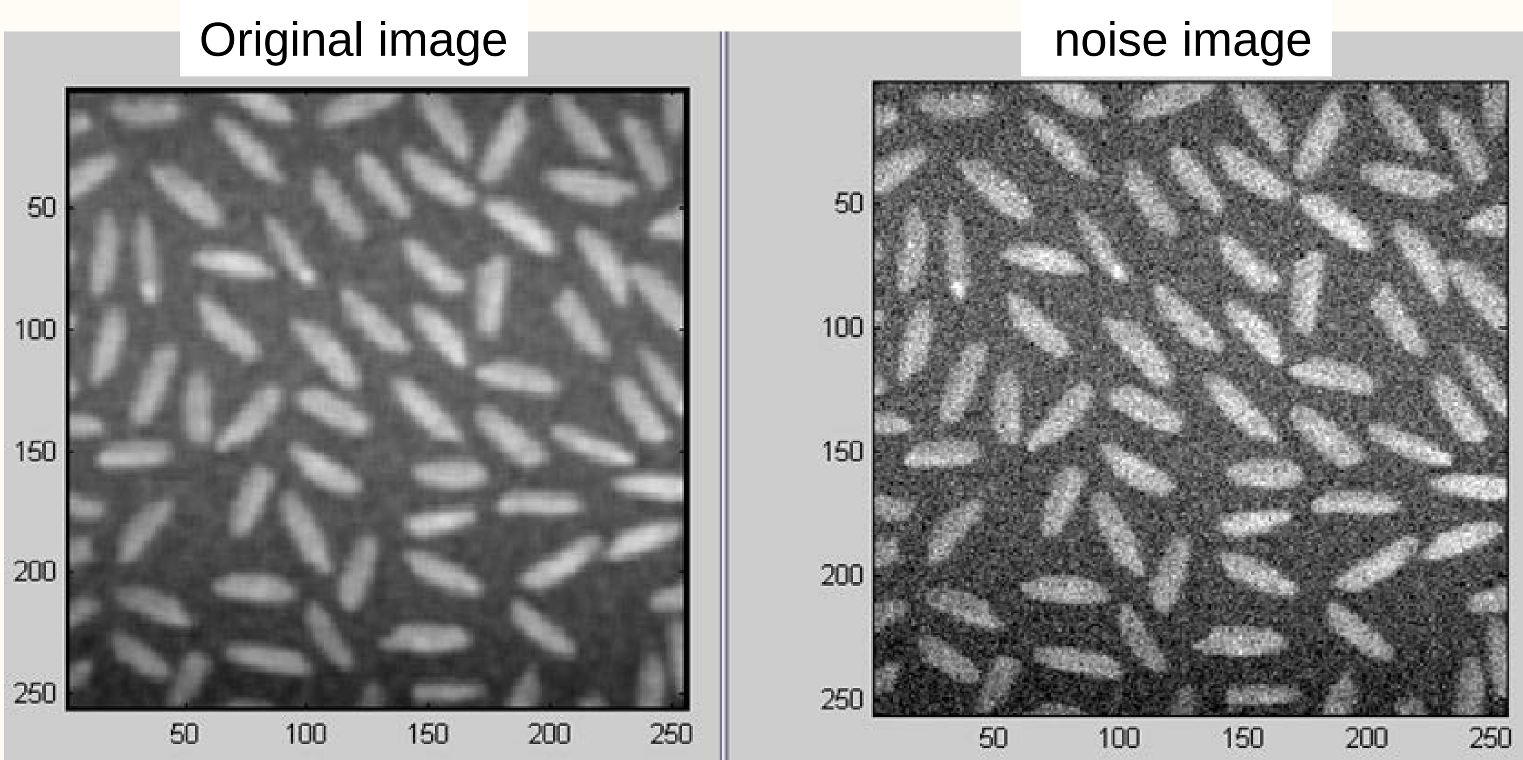



Fig. 14. Cubic B-spline interpolation. (a) Kernel plotted for $|x| < 3$. (b) Magnitude of Fourier transform. (c) Logarithmic plot of magnitude.

# 3.3 Local pre-processing

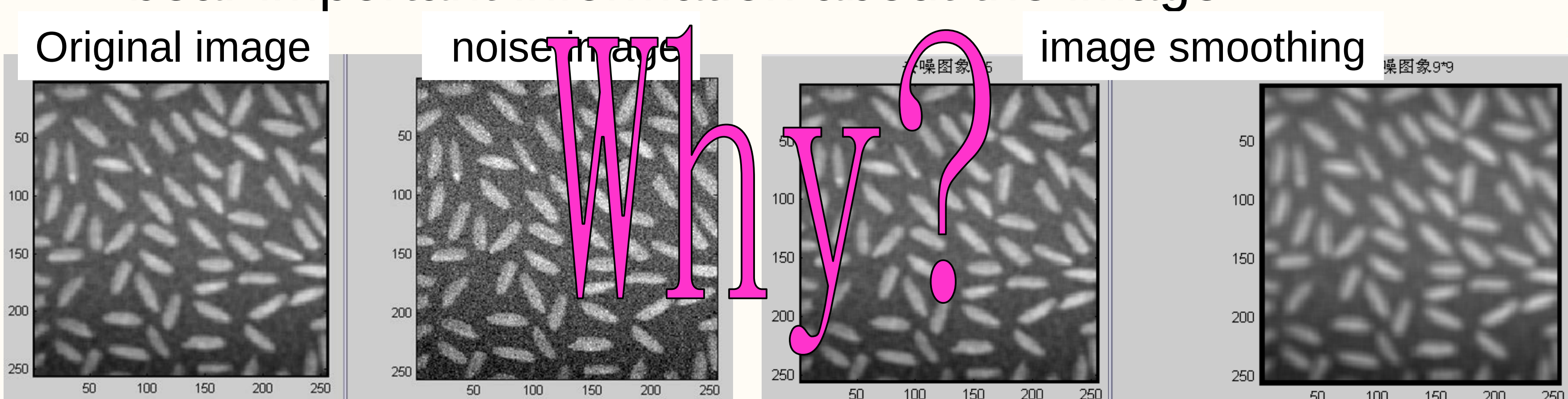
- Pre-processing methods use a small neighborhood of a pixel in an input image to get a new brightness value in the output image.
- Such pre-processing operations are called also **filtering**.

明辨笃行
博学慎思




the
:

- Unfortunately, smoothing also blurs all sharp edges that bear important information about the image.



# 3.3 Local pre-processing

- **Gradient operators** are based on local derivatives of the image function.
- Derivatives are bigger at locations of the image where the image function undergoes rapid changes. The aim of gradient operators is to indicate such locations in the image.

Sharpening !

# 3.3 Local pre-processing

- Gradient operators have a similar effect as suppressing low frequencies in the frequency domain.
- Noise is often high frequency in nature; unfortunately, if a gradient operator is applied to an image the noise level increases simultaneously.
- Clearly, smoothing and gradient operators have conflicting aims.

Why?

Sharpening !

# 3.3 Local pre-processing

- The contribution of the pixels in the neighborhood is weighted by coefficients h

$$f(i,j) = \sum_{(m,n)} \sum_{\in \mathcal{O}} h(i-m,\, j-n)\, g(m,n)$$

- The above equation is equivalent to discrete convolution with the kernel h, that is called a **convolution mask**.

# About the Rectangular neighborhoods (mask/window)!

- Rectangular neighborhoods O are often used with an odd number of pixels in rows and columns, enabling the specification of the central pixel of the neighborhood.
- Local pre-processing methods typically use very little a priori knowledge about the image contents. It is very difficult to infer this knowledge while an image is processed as the known neighborhood O of the processed pixel is small.
- The choice of the local transformation, size, and shape of the neighborhood O depends strongly on the size of objects in the processed image.

# 3.3.1 Image smoothing

- Image smoothing is the set of local pre-processing methods which have the aim of suppressing image noise - it uses redundancy in the image data.
- Calculation of the new value is based on averaging of brightness values in some neighborhood O.

# 3.3.1 Image smoothing

- Smoothing poses the problem of blurring sharp edges in the image, and so we shall concentrate on smoothing methods which are **edge preserving**. They are based on the general idea that the average is computed only from those points in the neighborhood which have similar properties to the processed point.
- Local image smoothing can effectively eliminate impulsive noise or degradations appearing as thin stripes, but does not work if degradations are large blobs or thick stripes.

# 3.3.1 -Averaging

- Assume that the noise value at each pixel is an independent random variable with zero mean and standard deviation
- We can obtain such an image by capturing the same static scene several times.

# 3.3.1 -Averaging

- The result of smoothing is an average of the same n points in these images g1,...,g{n} with noise values {1},..., {n}

$$\frac{g_1 + \ldots + g_n}{n} + \frac{\nu_1 + \ldots + \nu_n}{n}$$

- The second term here describes the effect of the noise ... a random value with zero mean.

# 3.3.1 -Averaging

- In many cases only one image with noise is available, and averaging is then realized in a local neighborhood.
- Results are acceptable if the noise is smaller in size than the smallest objects of interest in the image, but blurring of edges is a serious disadvantage.

# 3.3.1 -Averaging

- Averaging is a special case of discrete convolution. For a 3 x 3 neighborhood the convolution mask h is

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

- Larger convolution masks for averaging are created analogously.

# 3.3.1 -Averaging

- The significance of the central pixel may be increased to better reflect properties of Gaussian noise.

$$h = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad h = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

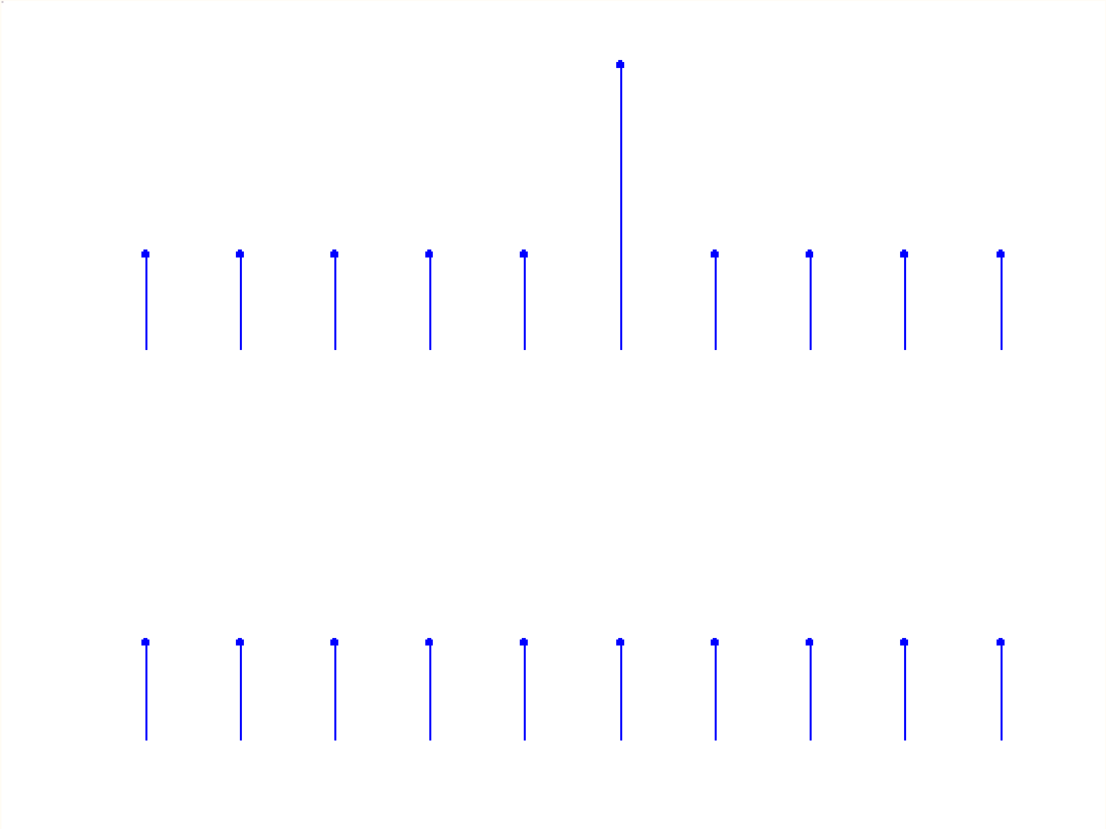
# 3.3.1 -Median smoothing

- In a set of ordered values, the median is the central value.
- The idea is to replace the current point in the image by the median of the brightness in its neighborhood.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| L | 80 | 90 | 200 | 110 | 120 | L | L 80 |
| L | 80 | 90 | 200 | 110 | 120 | L | L 80 90 |
| L | 80 | 90 | 200 | 110 | 120 | L | L 80 90 110 |
| L | 80 | 90 | 200 | 110 | 120 | L | L 80 90 110 120 |
| L | 80 | 90 | 200 | 110 | 120 | L | L 80 90 110 120 120 L |

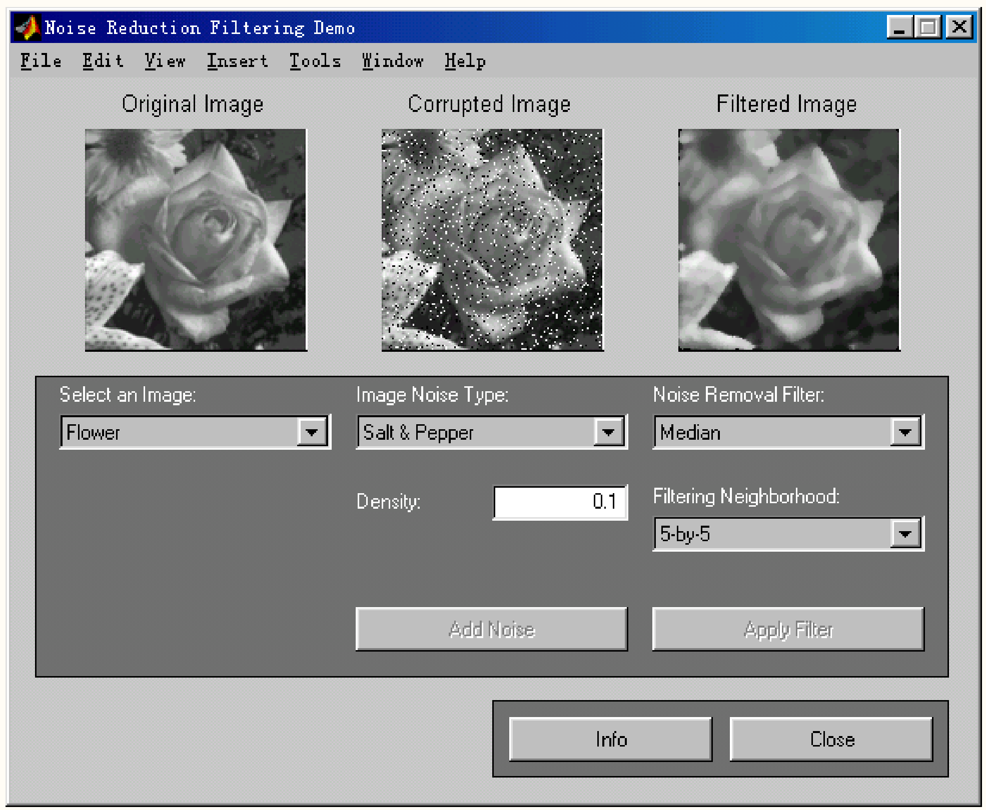
# 3.3.1 -Median smoothing

- not affected by individual noise spikes
- eliminates impulsive noise quite well



# 3.3.1 -Median smoothing

- Median filtering reduces blurring of edges.
- does not blur edges much and can be applied iteratively.



# 3.3.1 -Median smoothing

- The main disadvantage of median filtering in a rectangular neighborhood is its damaging of thin lines and sharp corners in the image -- this can be avoided if another shape of neighborhood is used.

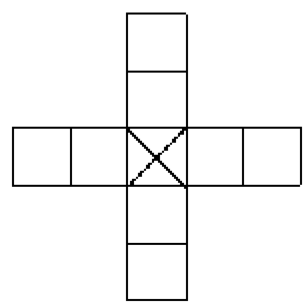


**Figure 4.14** *Horizontal/vertical line-preserving neighbourhood for median filtering.*

# 3.3.2 Edge detectors

- locate sharp changes in the intensity function
- edges are pixels where brightness changes abruptly.
- Calculus describes changes of continuous functions using derivatives; an image function depends on two variables - partial derivatives.
- A change of the image function can be described by a gradient that points in the direction of the largest growth of the image function.

# 3.3.2 Edge detectors

- An edge is a property attached to an individual pixel and is calculated from the image function behavior in a neighborhood of the pixel.

- It is a **vector variable**
  - **magnitude** of the gradient
  - **direction**

# 3.3.2 Edge detectors

- The gradient direction gives the direction of maximal growth of the function, e.g., from black (f(i,j)=0) to white (f(i,j)=255).
- This is illustrated below; closed lines are lines of the same brightness.

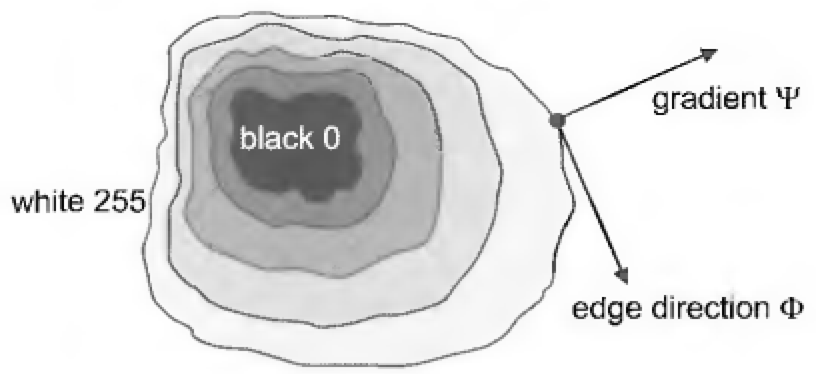



**Figure 5.18**: Gradient direction and edge direction.

# 3.3.2 Edge detectors

- Boundary and its parts (edges) are perpendicular to the direction of the gradient.

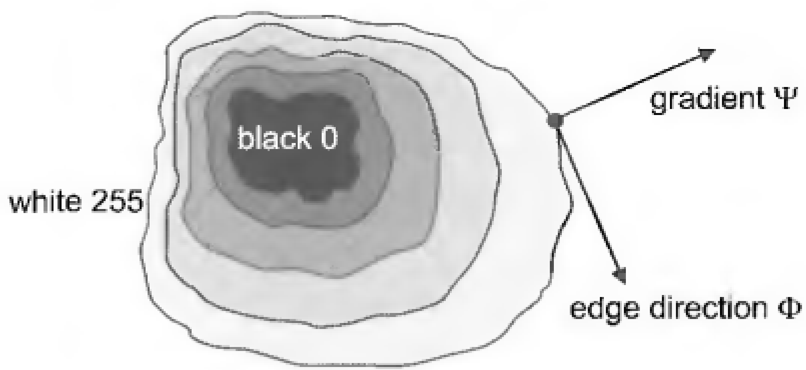



**Figure 5.18**: Gradient direction and edge direction.

# 3.3.2 Edge detectors

- Edges are often used in image analysis for finding region boundaries.

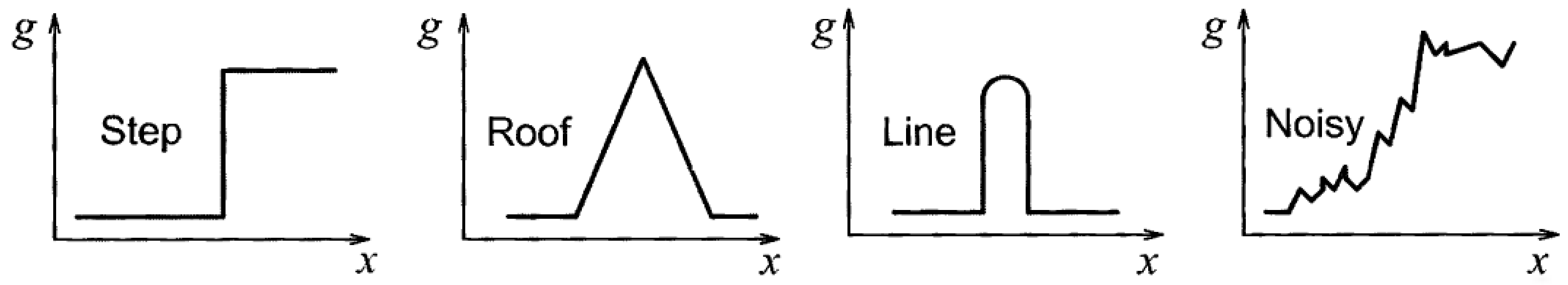



**Figure 5.19**: Typical edge profiles.

# 3.3.2 Edge detectors

- The gradient magnitude and gradient direction are continuous image functions where arg(x,y) is the angle (in radians) from the x-axis to the point (x,y).

$$|grad\ g(x,y)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2}$$

$$\psi = arg(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y})$$

How to compute the gradient magnitude?

# 3.3.2 Edge detectors

- Sometimes we are interested only in edge magnitudes without regard to their orientations.
- The **Laplacian** may be used.
- The Laplacian has the same properties in all directions and is therefore invariant to rotation in the image.

$$\nabla^2 g(x,y) = \frac{\partial^2 g(x,y)}{\partial x^2} + \frac{\partial^2 g(x,y)}{\partial y^2} \,. \tag{5.34}$$

明辨笃行 慎思

# 3.3.2 Edge detectors

- Image sharpening makes edges steeper -- the sharpened image is intended to be observed by a human.
- C is a positive coefficient which gives the strength of sharpening and S(i,j) is a measure of the image function sheerness that is calculated using a gradient operator.
- The Laplacian is very often used to estimate S(i,j).

$$f(i,j) = g(i,j) - C\,S(i,j)\,, \tag{5.35}$$

# 3.3.2 -Laplace operator

- The Laplace operator is a very popular operator approximating the second derivative which gives the gradient magnitude only.
- The Laplacian is approximated in digital images by a convolution sum.
- A 3 x 3 mask for 4-neighborhoods and 8-neighborhood

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \qquad h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{5.40}$$

# 3.3.2 -Laplace operator

- A Laplacian operator with stressed significance of the central pixel or its neighborhood is sometimes used. In this approximation it loses invariance to rotation

$$h = \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix}, \qquad h = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}. \tag{5.41}$$

- The Laplacian operator has a disadvantage -- it responds doubly to some edges in the image.

# 3.3.2 Edge detectors

- Image sharpening / edge detection can be interpreted in the frequency domain as well.
- The result of the Fourier transform is a combination of harmonic functions.
- The derivative of the harmonic function sin(nx) is ncos(nx); thus the higher the frequency, the higher the magnitude of its derivative.
- This is another explanation of why gradient operators enhance edges.

# 3.3.2 Edge detectors

- Gradient operators can be divided into **three** categories
- 1. Operators approximating derivatives of the image function using differences.
  - rotationally invariant (e.g., Laplacian) need one convolution mask only.
  - approximating first derivatives use several masks ... the orientation is estimated on the basis of the best matching of several simple patterns.
- 2. Operators based on the zero crossings of the image function second derivative (e.g., Marr-Hildreth or Canny edge detector).
- 3. Operators which attempt to match an image function to a parametric model of edges.

# 3.3.2 Edge detectors

- Parametric models describe edges more precisely than simple edge magnitude and direction and are much more computationally intensive.
- Individual gradient operators that examine small local neighborhoods are in fact convolutions and can be expressed by convolution masks.
- Operators which are able to detect edge direction as well are represented by a collection of masks, each corresponding to a certain direction.

# 3.3.2 -other operator

- Roberts operator pp.135
- Prewitt operator pp.136
- Sobel operator pp.136
- Robinson operator pp.137
- Kirsch operator pp.138

# 3.3.3 Marr-Hildreth Edge Detection: Zero crossings of the second derivative

- Edge detection techniques like the Kirsch, Sobel, Prewitt operators are based on convolution in very small neighborhoods and work well for specific images only.
- The main disadvantage of these edge detectors is their dependence on the size of objects and sensitivity to noise.

# 3.3.3 Marr-Hildreth Edge Detection: Zero crossings of the second derivative

- The Marr-Hildreth edge detection technique, based on the **zero crossings** of the second derivative explores the fact that a step edge corresponds to an abrupt change in the image function.
- The first derivative of the image function should have an extreme at the position corresponding to the edge in the image, and so the second derivative should be zero at the same position.

# 3.3.3 Marr-Hildreth Edge Detection: Zero crossings of the second derivative

- It is much easier and more precise to find a zero crossing position than an extreme

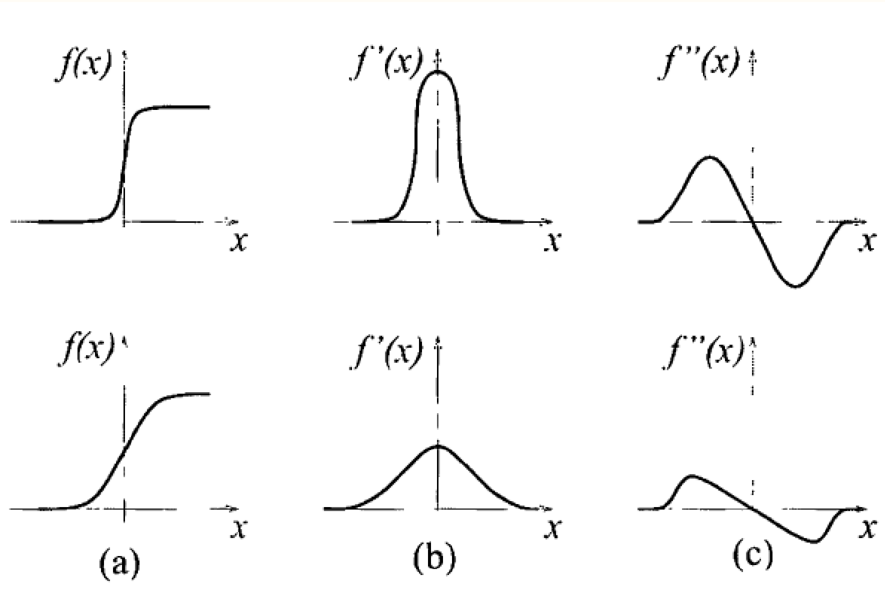



**Figure 5.22**: 1D edge profile of the zero-crossing.

# 3.3.3 Marr-Hildreth Edge Detection: Zero crossings of the second derivative

- Robust calculation of the 2nd derivative:
  - smooth an image first (to reduce noise) and then compute second derivatives.
  - The 2D Gaussian smoothing operator G(x,y)

$$G(x,y) = e^{-(x^2+y^2)/2\sigma^2} \;, \tag{5.47}$$

- where x, y are the image co-ordinates and sigma is a standard deviation of the associated probability distribution.

# 3.3.4 -Local pre-processing in the frequency domain

- Many filters used in image pre-processing were presented—the convolution masks in most cases were used for image filtering or image gradient computation. It is natural to think about processing these (and many other) convolutions in the frequency domain. Such operations are usually called spatial frequency filtering.

- Spatial filtering are linear low-pass, high-pass, and bandpass frequency filters.

# 3.3.4 -Local pre-processing in the frequency domain

- To explain the working of frequency filters, we shall again consider regions of low and high spatial frequency.
- ency regions of an image manifest
  relatively large areas of uniform gray,
  ch gray does not change rapidly
  tly

Regions of low spatial frequency?

Regions of high spatial frequency?

- gh frequency are characterised by rapid changes in gray. Edges!

明辨笃行

博学慎思

# 3.3.4 -Local pre-processing in the frequency domain

Assume that $f$ is an input image and $F$ is its Fourier transform. A convolution filter $h$ can be represented by its Fourier transform $H$; $h$ may be called the unit pulse response of the filter and $H$ the frequency transfer function, and either of the representations $h$ or $H$ can be used to describe the filter. The Fourier transform of the filter output after an image $f$ has been convolved with the filter $h$ can be computed in the frequency domain

$$G(u, v) = F(u, v) \mathbin{.*} H(u, v)\,, \tag{5.64}$$

where operation $.*$ represents a element-by-element multiplication of matrices $F$ and $H$, not a matrix multiplication. Matrices $F$ and $H$ must be of the same size. The filtered image $g$ can be obtained by applying the inverse Fourier transform to $\mathbf{G}$ —equation (3.28).

low-pass, high-pass, and bandpass frequency filters!

# 3.3.4 -Local pre-processing in the frequency domain

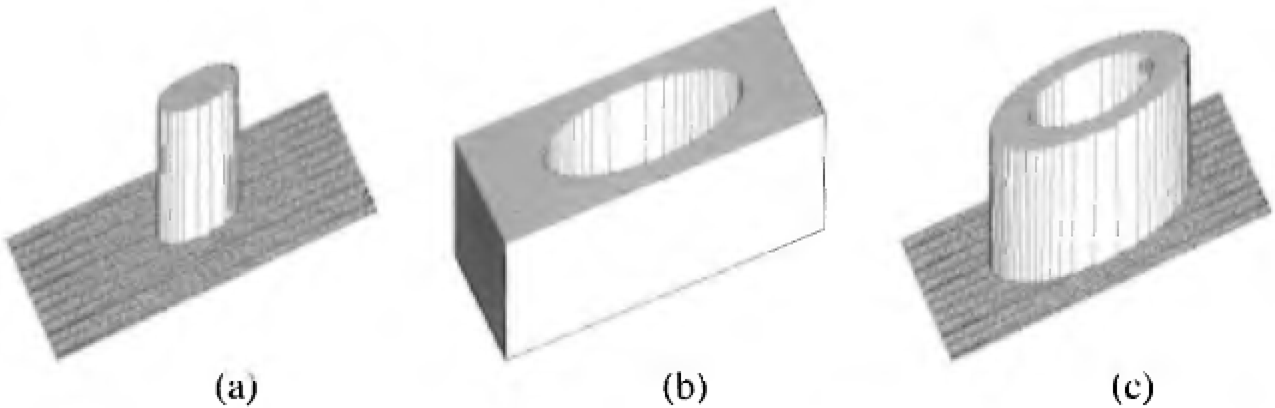



**Figure 5.26**: Frequency filters displayed in 3D. (a) Low-pass filter. (b) High-pass filter. (c) Band-pass filter.

# 3.3 Local pre-processing (other contents)

- Scale in image processing
- Parametric edge models
- Edges in multi-spectral images
- Local pre-processing in the frequency domain
- Line detection by local pre-processing operators
- The task can be formulated as an optimization
- Detection of corners (interest points)

# 3.4 Image restoration

- **Image restoration** - suppressing image degradation using knowledge about its nature
- Most image restoration methods are based on convolution applied globally to the whole image.
- The objective of image restoration is to reconstruct the original image from its degraded version.

## 3.4.1 **Image restoration as inverse convolution of the whole image**

- Image restoration techniques - two groups:
- **Deterministic** methods - applicable to images with little noise and a known degradation function.
  - The original image is obtained from the degraded one by a transformation inverse to the degradation.
- **Stochastic** techniques - the best restoration is sought according to some stochastic criterion, e.g., a least squares method.
  - In some cases the degradation transformation must be estimated first.

# 3.4.1 **Image restoration as inverse convolution of the whole image**

- It is advantageous to know the degradation function explicitly.
- The better this knowledge is, the better are the results of the restoration.

# 3.4.1 **Image restoration as inverse convolution of the whole image**

- Degradation causes:
  - – defects of optical lenses,
  - – nonlinearity of the electro-optical sensor,
  - – graininess of the film material,
  - – relative motion between an object and camera
  - – wrong focus,
  - – atmospheric turbulence in remote sensing or astronomy,
  - – etc.

# 3.4.1 **Image restoration as inverse convolution of the whole image**

- There are three typical degradations with a simple function:
  - Relative constant speed movement of the object with respect to the camera,
  - wrong lens focus,
  - and atmospheric turbulence.
- In most practical cases, there is insufficient knowledge about the degradation, and it must be estimated and modeled.

# 3.4.1 **Image restoration as inverse convolution of the whole image**

- <u>Methods:</u>
  - **A priori knowledge** about degradation - either known in advance or obtained before restoration.
  - If it is clear in advance that the image was degraded by relative motion of an object with respect to the sensor then the modeling only determines the speed and direction of the motion.

# 3.4.1 **Image restoration as inverse convolution of the whole image**

- <u>Methods:</u>
  - **A posteriori knowledge** is obtained by analyzing the degraded image.
  - A typical example is to find some interest points in the image (e.g. corners, straight lines) and guess how they looked before degradation.
- Another possibility is to use spectral characteristics of the regions in the image that are relatively homogeneous.

## 3.4.1 **Image restoration as inverse convolution of the whole image**

- A degraded image g can arise from the original image f by a process which can be expressed as

$$g(i,j) = s\left(\int\int_{(a,b)\in\mathcal{O}} f(a,b)\, h(a,b,i,j)\,\mathrm{d}a\,\mathrm{d}b\right) + \nu(i,j)\,, \qquad (5.77)$$

where s is some nonlinear function and $\nu$ describes the noise.

# 3.4.1 **Image restoration as inverse convolution of the whole image**

- The degradation is very often simplified by
  - neglecting the nonlinearity
  - assuming that the function h is invariant with respect to position in the image.
- Degradation can be then expressed as convolution

$$g(i,j) = (f * h)(i,j) + \nu(i,j)\,. \tag{5.78}$$

# 3.4.1 **Image restoration as inverse convolution of the whole image**

- If the degradation is given by equation (5.78) and the noise is not significant then image restoration equates to **inverse convolution** (also called **deconvolution**). !
- If noise is not negligible then the inverse convolution is solved as an overdetermined system of linear equations.

$$g(i,j) = (f * h)(i,j) + \nu(i,j)\,. \tag{5.78}$$

# 3.4.2 **Degradations that are easy to restore**

- Some degradations can be easily expressed mathematically (convolution) and also restored simply in images.
- The Fourier transform H of the convolution function is used.

In the absence of noise, the relationship between the Fourier representations $F, G, H$ of the undegraded image $f$, the degraded image $g$, and the degradation convolution kernel $h$, respectively, is

$$G = H F . \tag{5.79}$$

# 3.4.2 -**Relative motion of the camera and object**

- Assume an image is acquired with a camera with a mechanical shutter.
- Relative motion of the camera and the photographed object during the shutter open time T causes smoothing of the object in the image.
- Suppose V is the constant speed in the direction of the x axis; the Fourier transform H(u,v) of the degradation caused in time T is given by

$$H(u,v) = \frac{\sin(\pi V T u)}{\pi V u} \,. \tag{5.80}$$

# 3.4.2 -**Wrong lens focus**

- Image smoothing caused by imperfect focus of a thin lens can be described by the following function

$$H(u, v) = \frac{J_1(a\,r)}{a\,r}\,, \qquad (5.81)$$

where $J_1$ is the Bessel function of the first order, r^2 = u^2 + v^2, and a is the displacement.

# 3.4.2 -**Atmospheric turbulence**

- needs to be restored in remote sensing and astronomy
- caused by temperature non-homogeneity in the atmosphere that deviates passing light rays.
- The mathematical model

$$H(u,v) = e^{-c(u^2+v^2)^{5/6}}, \tag{5.82}$$

where c is a constant that depends on the type of turbulence which is usually found experimentally.

- The power 5/6 is sometimes replaced by 1.

# 3.4.3 **Inverse filtration**

- based on the assumption that degradation was caused by a linear function h(i,j)
- the additive noise nu is another source of degradation.
- It is further assumed that nu is independent of the signal.
- Applying the Fourier transform

$$G(u,v) = F(u,v)\,H(u,v) + N(u,v)\,. \tag{5.83}$$

- The degradation can be eliminated if the restoration filter has a transfer function that is inverse to the degradation h ... or in Fourier transform H^-1(u,v).

# 3.4.3 **Inverse filtration**

- The undegraded image F is derived from its degraded version G.

$$F(u,v) = G(u,v)\,H^{-1}(u,v) - N(u,v)\,H^{-1}(u,v)\;. \tag{5.84}$$

- This equation shows that inverse filtration works well for images that are not corrupted by noise.

# 3.4.3 **Inverse filtration**

- If noise is present and additive error occurs, its influence is significant for frequencies where H(u,v) has small magnitude. These usually correspond to high frequencies u,v and thus fine details are blurred in the image.
- The changing level of noise may cause problems as well because small magnitudes of H(u,v) can cause large changes in the result.
- Inverse filter values should not be of zero value so as to avoid zero divides in equation (5.84).

$$F(u,v) = G(u,v)\,H^{-1}(u,v) - N(u,v)\,H^{-1}(u,v)\,. \qquad (5.84)$$

# 3.4.4 **Wiener filtration**

- It is no surprise that inverse filtration gives poor results in pixels suffering from noise since the noise is not taken into account.
- Wiener filtration explores a priori knowledge about the noise.
- Restoration by the Wiener filter gives an estimate of the original uncorrupted image f with minimal mean square error e^2

$$e^2 = \mathcal{E}\left\{\left(f(i,j) - \hat{f}(i,j)\right)^2\right\}\,, \tag{5.85}$$

# 3.4.4 **Wiener filtration**

- No constraints applied to equation (5.85)
  - optimal estimate $\hat{f}$ is the conditional mean value of the ideal image f under the condition g
  - complicated from the computational point of view
  - also, the conditional probability density between the optimal image f and the corrupted image *g* is not usually known.
  - the optimal estimate is in general a nonlinear function of the image g.

$$e^2 = \mathcal{E}\left\{\left(f(i,j) - \hat{f}(i,j)\right)^2\right\} , \qquad (5.85)$$

# 3.4.4 **Wiener filtration**

- Minimization of equation (5.85) is easy if estimate $\hat{f}$ is a linear combination of the values in the image g -close-optimal solution

$$e^2 = \mathcal{E}\left\{\left(f(i,j) - \hat{f}(i,j)\right)^2\right\} , \tag{5.85}$$

# 3.4.4 **Wiener filtration**

- $H_W$ ... Fourier transform of the Wiener filter - considers noise - see Eq. (5.87) below
- G ... Fourier transform of the degraded image

$$\hat{F}(u,v) = H_W(u,v)\, G(u,v)\,. \tag{5.86}$$

$$H_W(u,v) = \frac{H^*(u,v)}{\left|H(u,v)\right|^2 + \left[S_{\nu\nu}(u,v)/S_{ff}(u,v)\right]}\,, \tag{5.87}$$

- where H is the transform function of the degradation, $*$ denotes complex conjugate
  $S_{\nu\nu}$ is the spectral density of the noise
  $S_{ff}$ is the spectral density of the degraded image

# 3.4.4 **Wiener filtration**

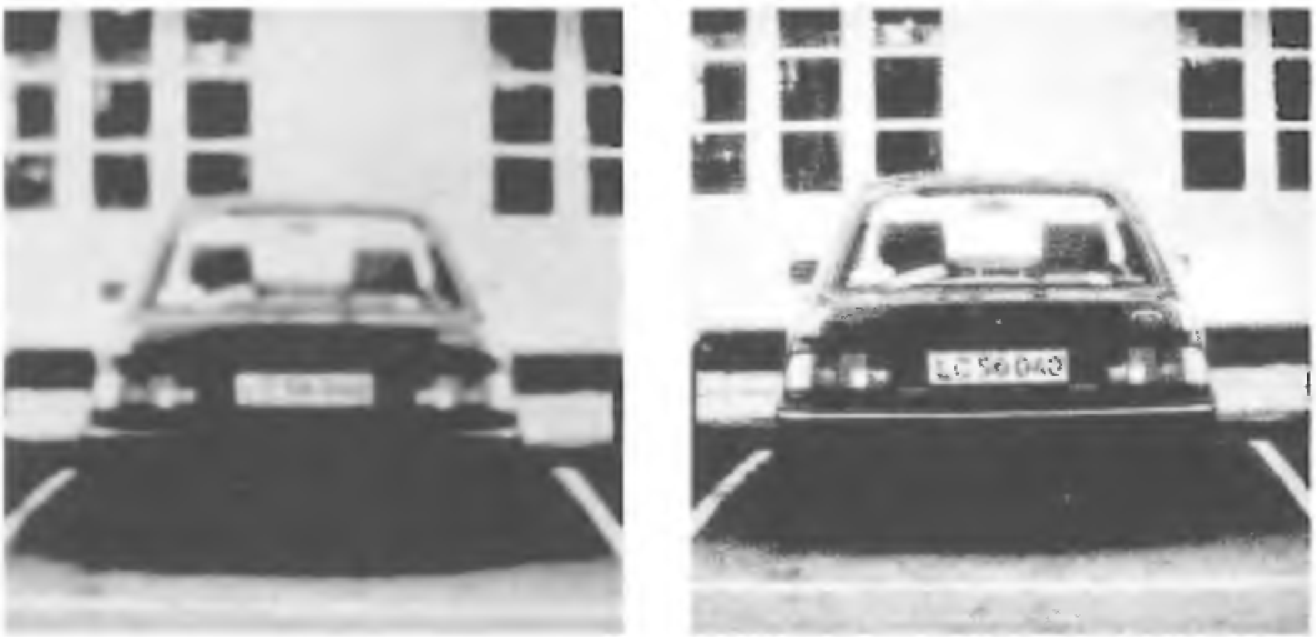
- If Wiener filtration is used, the nature of degradation H and statistical parameters of the noise need to be known.
- Wiener filtration theory solves the problem of a posteriori linear mean square estimate -- all statistics (e.g., power spectrum) should be available in advance.
- Note that the inverse filter discussed earlier is a special case of the Wiener filter where noise is absent i.e. $S_{\nu\nu}$ = 0.

# 3.4.4 **Wiener filtration**

(a) (b)

**Figure 5.39**: Restoration of motion blur using Wiener filtration. *Courtesy of P. Kohout, Criminalistic Institute, Prague.*

# 3.4.4 **Wiener filtration**



(a) (b)

**Figure 5.40**: Restoration of wrong focus blur using Wiener filtration. *Courtesy of P. Kohout, Criminalistic Institute, Prague.*

# Knowledge points

- Pixel brightness transformations
- Geometric transformations

    Pixel co-ordinate transformations

    Interpolation
- Local pre-processing

    Image Smoothing !

    Image Sharpening !
- Image restoration

明辨笃行

博学慎思

# Questions and Practices

- See “Practice 3  Image pre-processing”