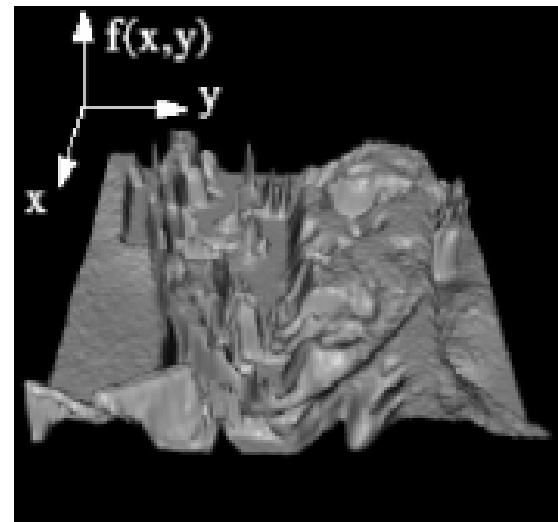


From 1D to 2D Signals

Lecture 9

What is an image?

- Ideally, we think of an **image** as a **2-dimensional light intensity function**, $f(x,y)$, where x and y are spatial coordinates, and f at (x,y) is related to the brightness or color of the image at that point.
- In practice, most images are defined over a rectangle.
- Continuous in amplitude („continuous-tone“)
- Continuous in space: no pixels!

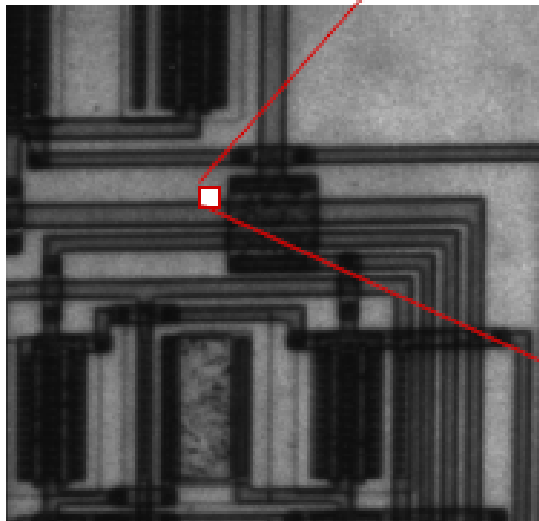


Digital Images and Pixels

- A **digital image** is the representation of a continuous image $f(x,y)$ by a 2-d array of discrete samples. The amplitude of each sample is quantized to be represented by a finite number of bits.
- Each element of the 2-d array of samples is called a **pixel** or **pel** (from “**picture element**”)
- Pixels are point samples, without extent.
- A pixel is not:
 - Round, square, or rectangular
 - An element of an image sensor
 - m An element of a display

A Digital Image is Represented by Numbers

272 pixels



280 pixels

128	125	107	105	110	118	116	114	110
121	122	115	108	106	107	116	116	107
110	114	112	107	105	103	106	106	100
100	96	100	99	94	94	101	101	89
85	82	81	80	76	75	80	82	72
58	58	56	54	53	52	51	49	45
41	41	41	39	39	38	36	35	33
43	43	42	43	41	41	41	43	40
60	60	59	59	60	59	59	58	56

- Pixel = "picture element"
- Represents brightness at one point

An image can be represented as a matrix

- The pixel values $f(x,y)$ are sorted into the matrix in “natural” order, with x corresponding to the column and y to the row index.
- Matlab, instead, uses matrix convention. This results in $f(x,y) = f_{yx}$, where f_{yx} denotes an individual element in common matrix notation.
- For a color image, \mathbf{f} might be one of the components.

$$\mathbf{f} = \begin{matrix} & \xrightarrow{x} & & \\ \begin{bmatrix} f(0,0) & f(1,0) & \cdots & f(N-1,0) \\ f(0,1) & f(1,1) & \cdots & f(N-1,1) \\ \vdots & \vdots & & \vdots \\ f(0,L-1) & f(1,L-1) & \cdots & f(N-1,L-1) \end{bmatrix} & & \downarrow y & \end{matrix}$$

Image Size and Resolution



200x200



100x100



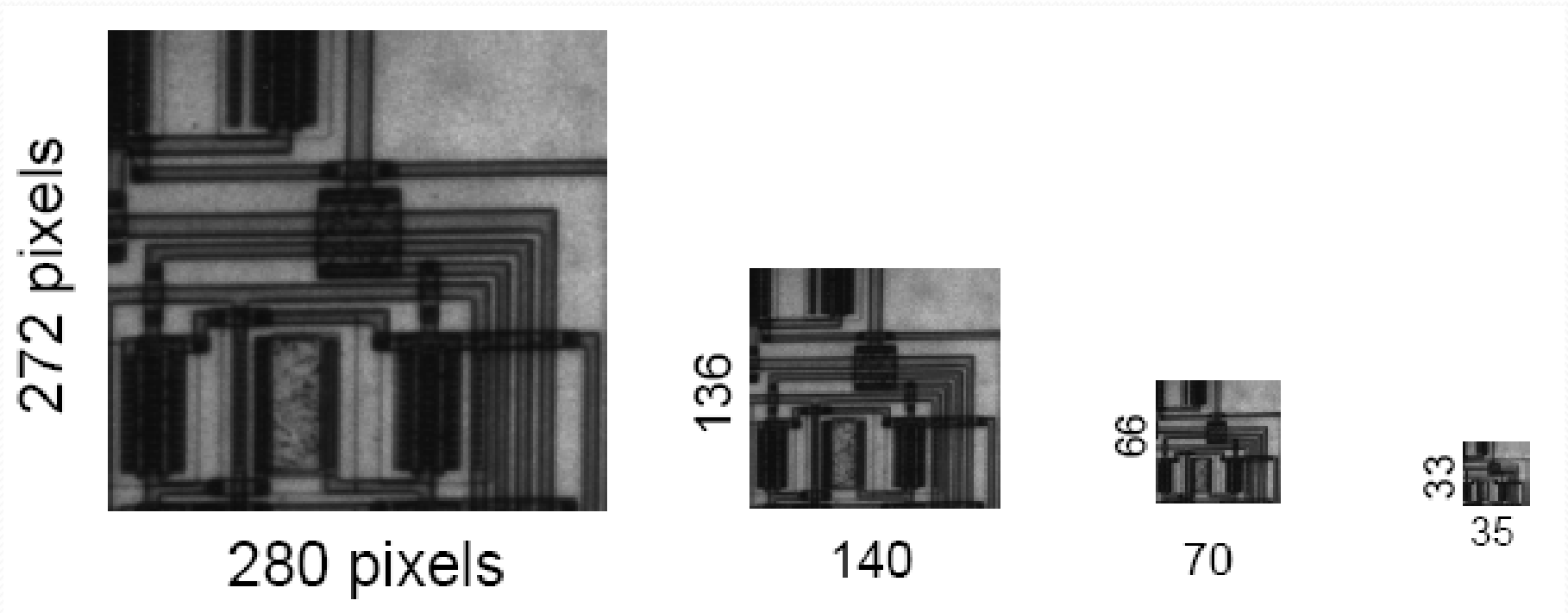
50x50



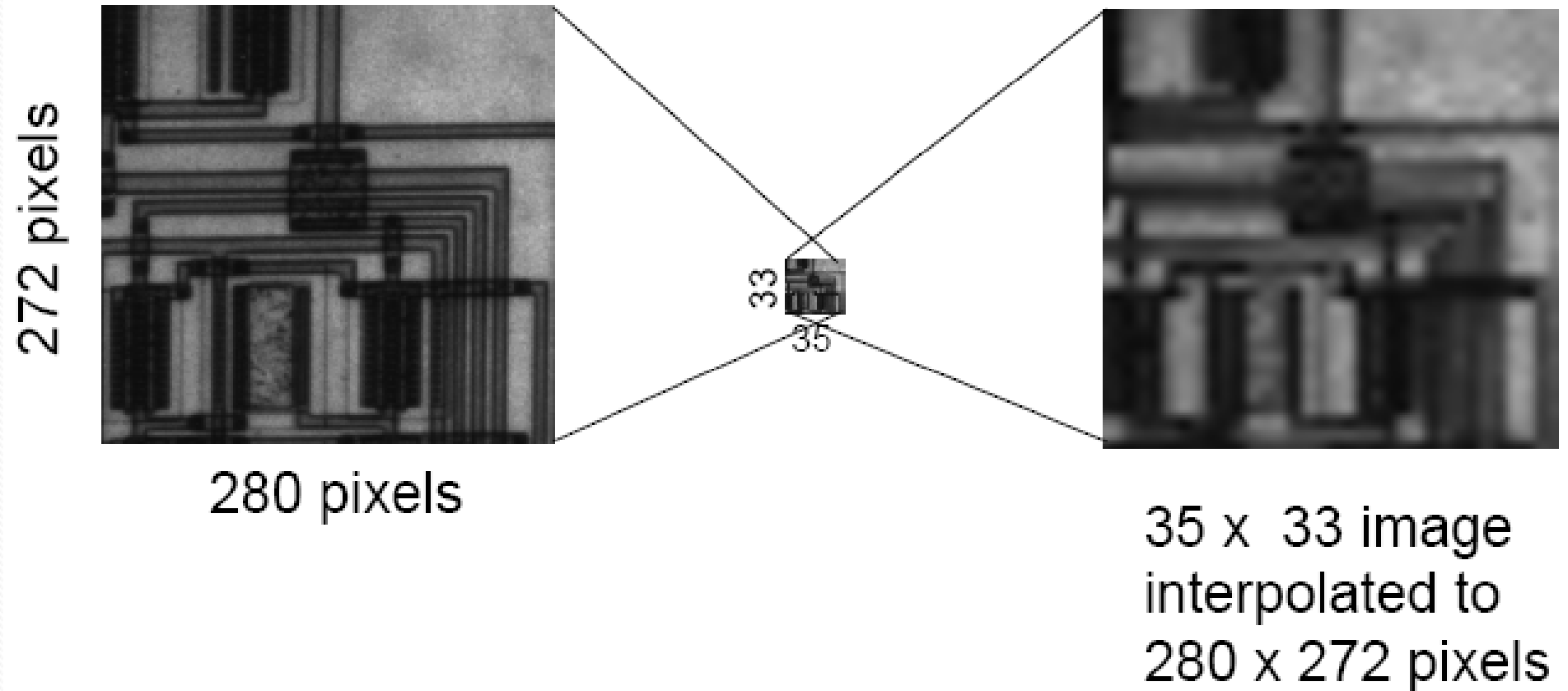
25x25

- These images were produced by simply picking every n -th sample horizontally and vertically and replicating that value $n \times n$ times.
- We can do better
 - *prefiltering before subsampling to avoid aliasing*
 - *Smooth interpolation*

Image of different size



Fewer Pixels Mean Lower Spatial Resolution



Color Components

Monochrome image



$$R(x,y) = G(x,y) = B(x,y)$$



Red $R(x,y)$

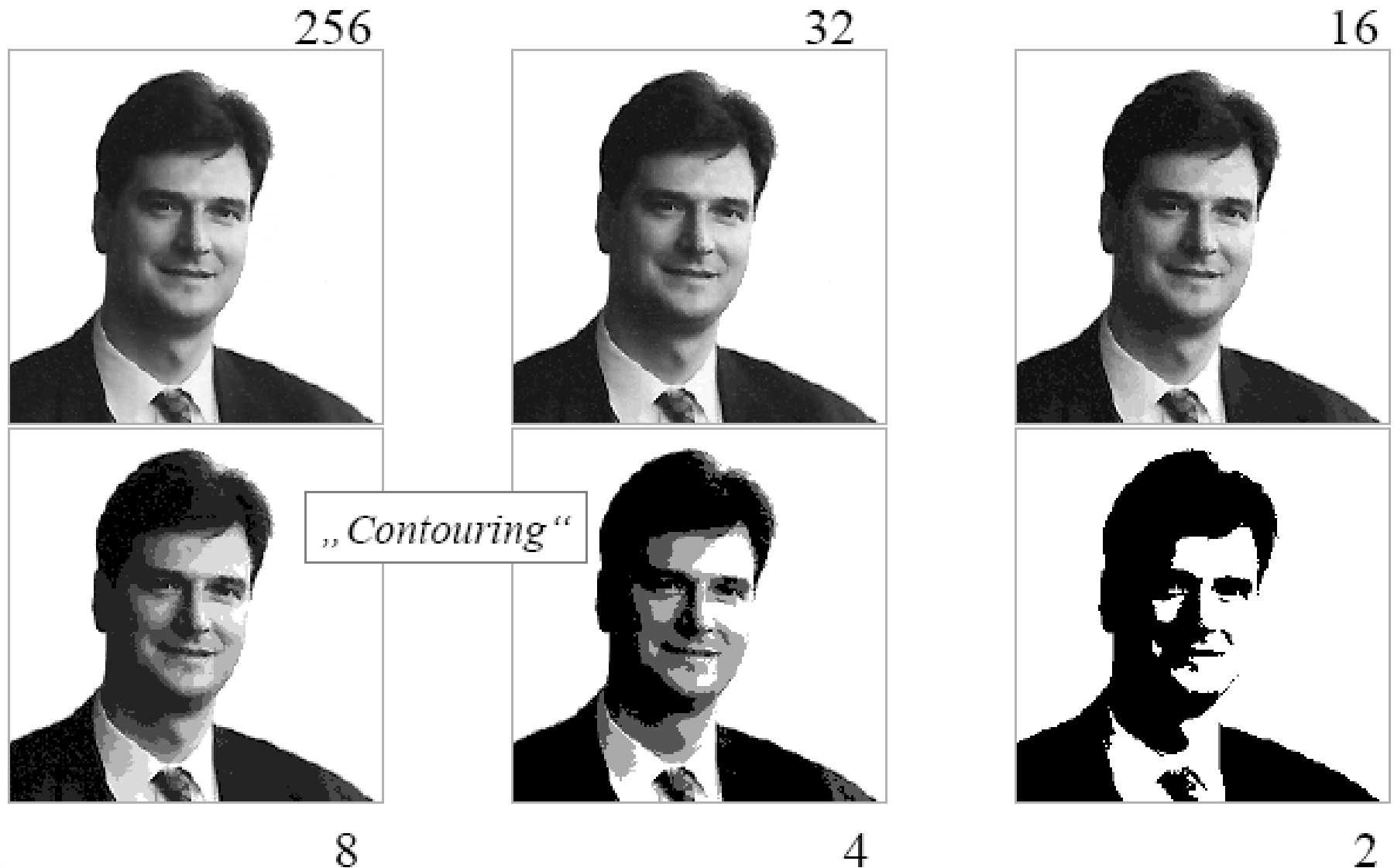


Green $G(x,y)$



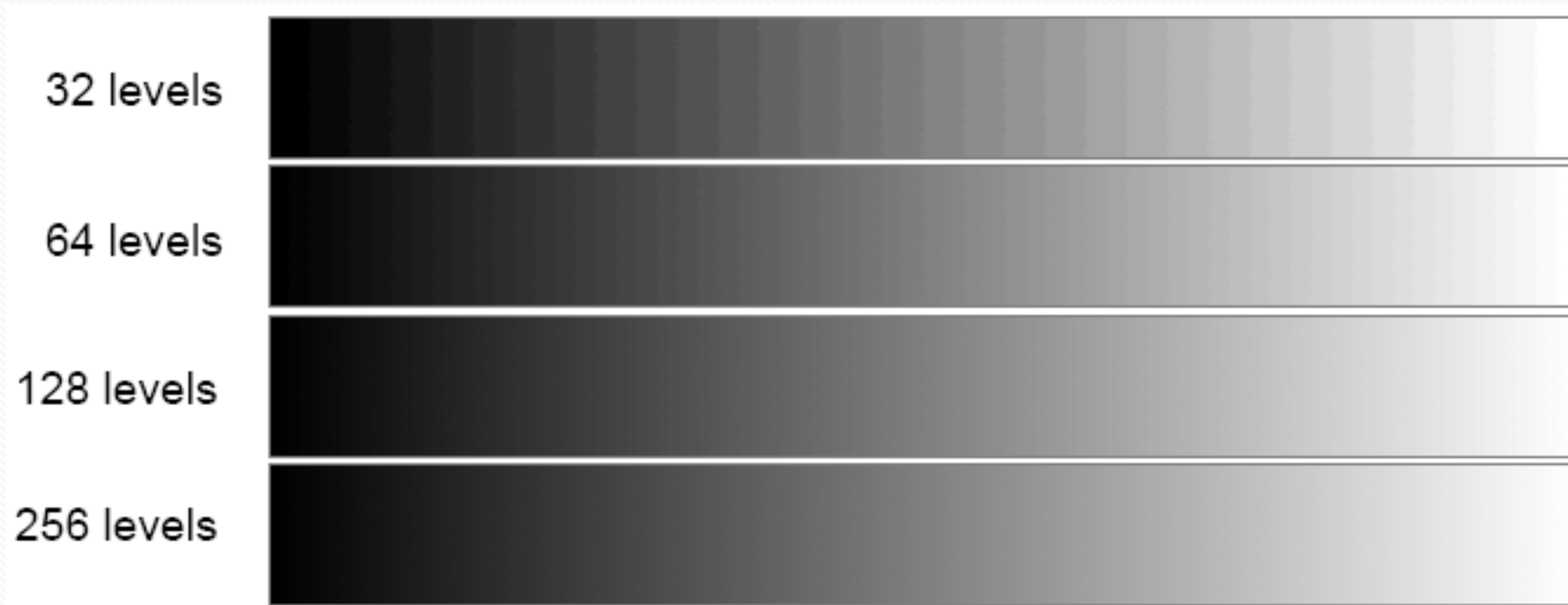
Blue $B(x,y)$

Different numbers of gray levels



How many gray levels are required?

- How many gray levels are required?

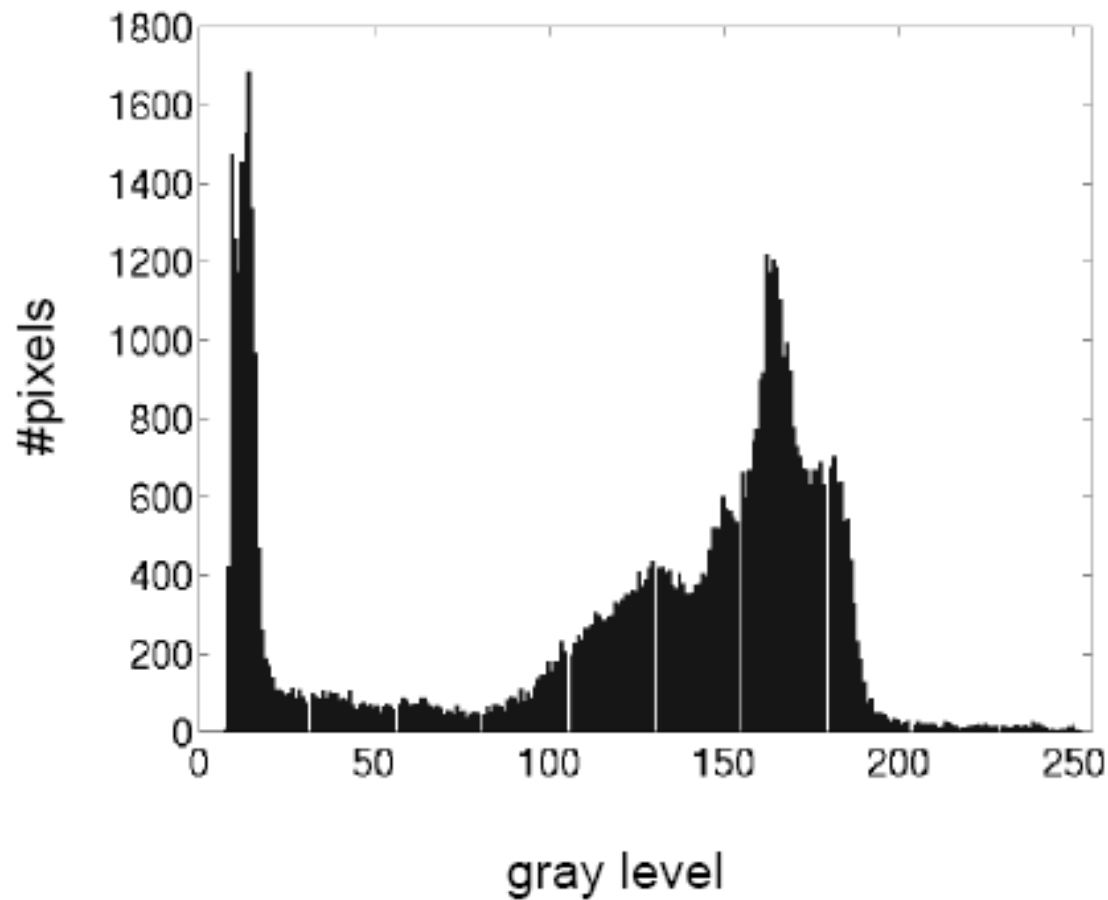


- Digital images typically are quantized to 256 gray levels.

Histograms

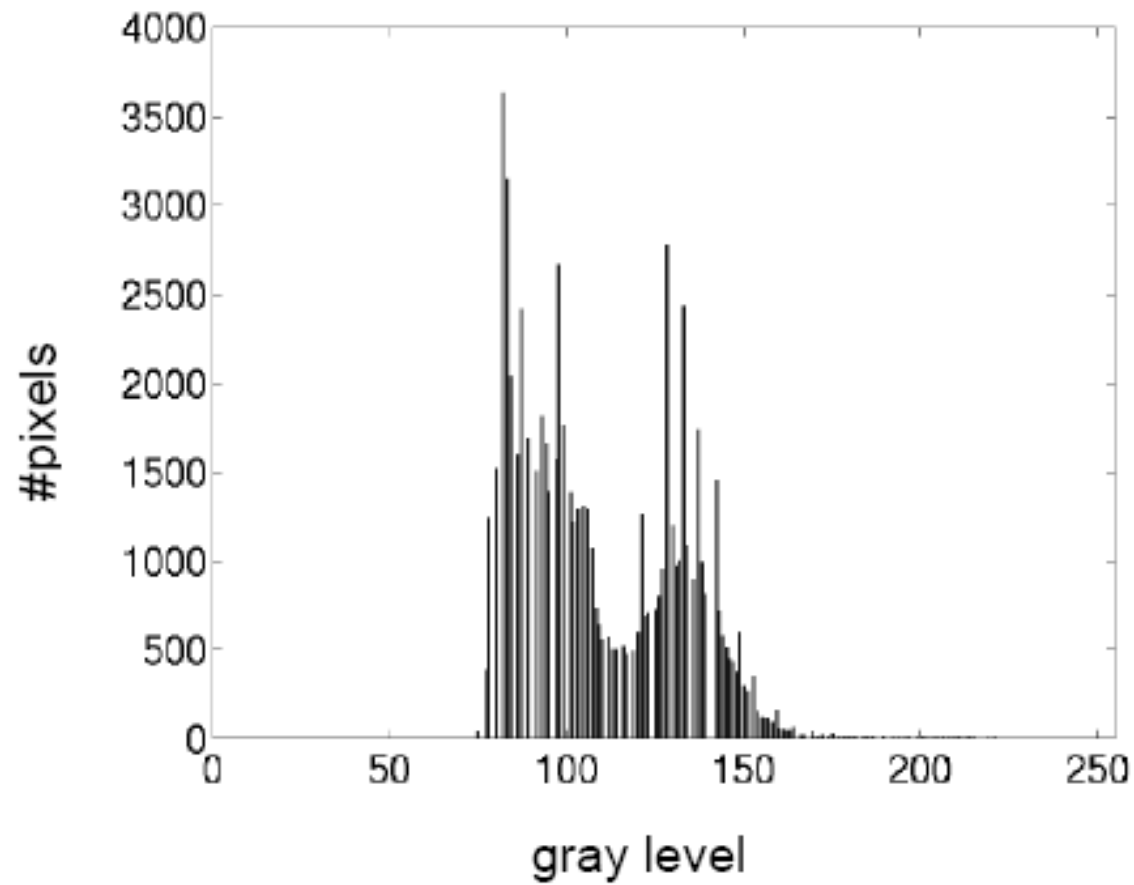
- Distribution of gray-levels can be judged by measuring a histogram:
 - For B-bit image, initialize 2^B counters with 0
 - Loop over all pixels x,y
- When encountering gray level $f(x,y)=i$, *increment counter #i*
- Histogram can be interpreted as an estimate of the probability density function (*pdf*) of an underlying random process.
- You can also use fewer, larger bins to trade off amplitude resolution against sample size.

Example for the histogram



Cameraman
image

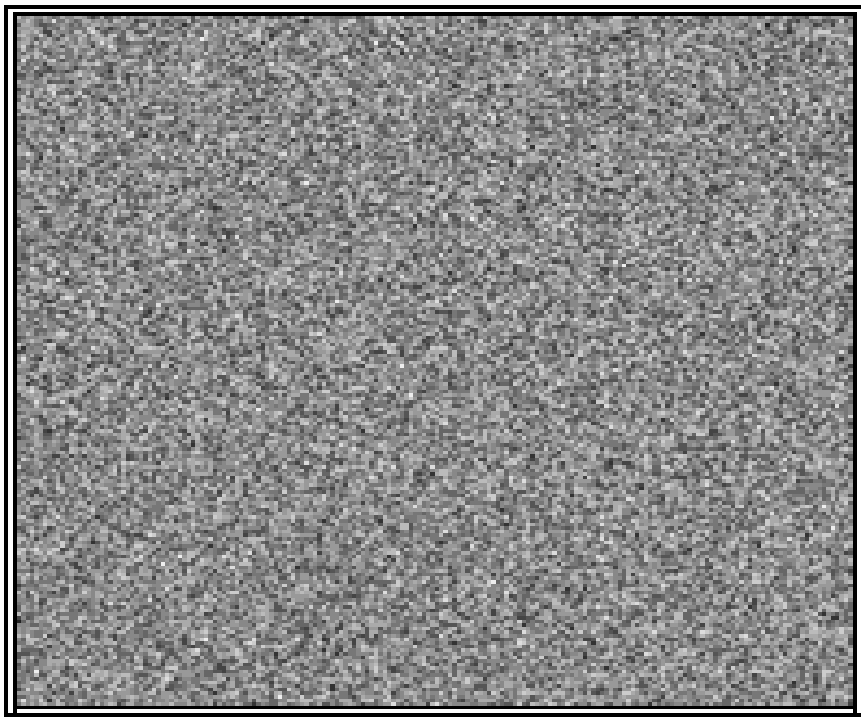
Histogram Example



Pout
image

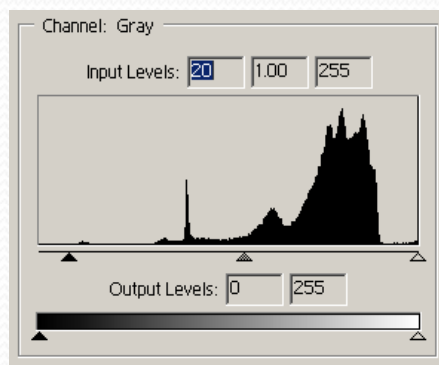
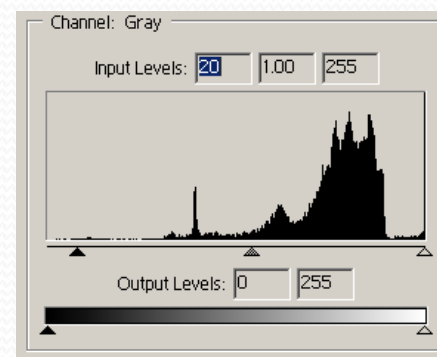
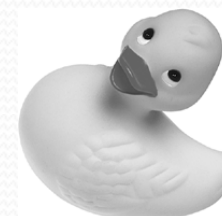
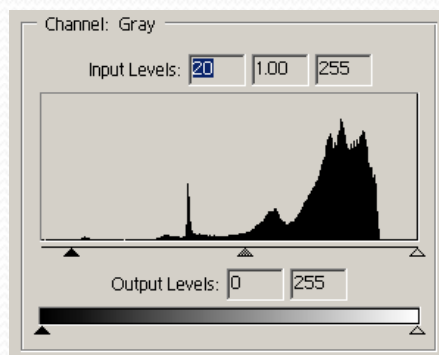
Histogram comparison

- Both these images present the same Histogram



Histogram comparison

- Histogram as an invariant feature



Histogram comparison

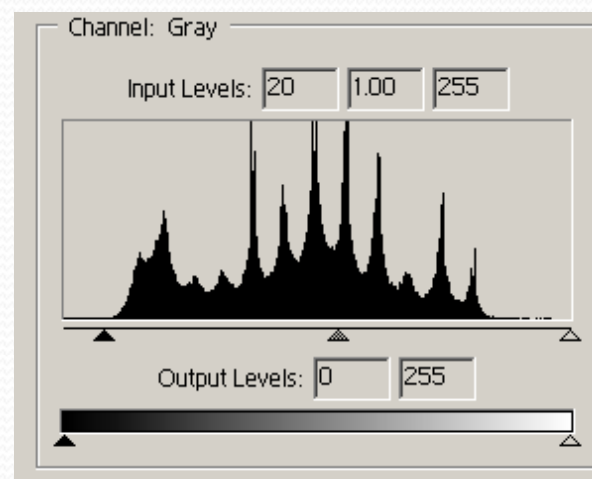
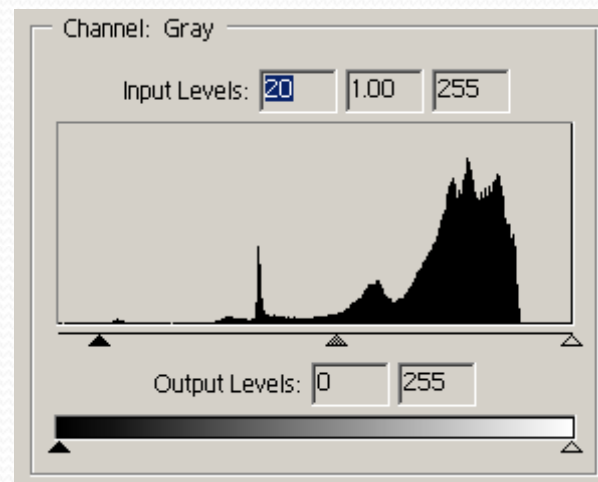
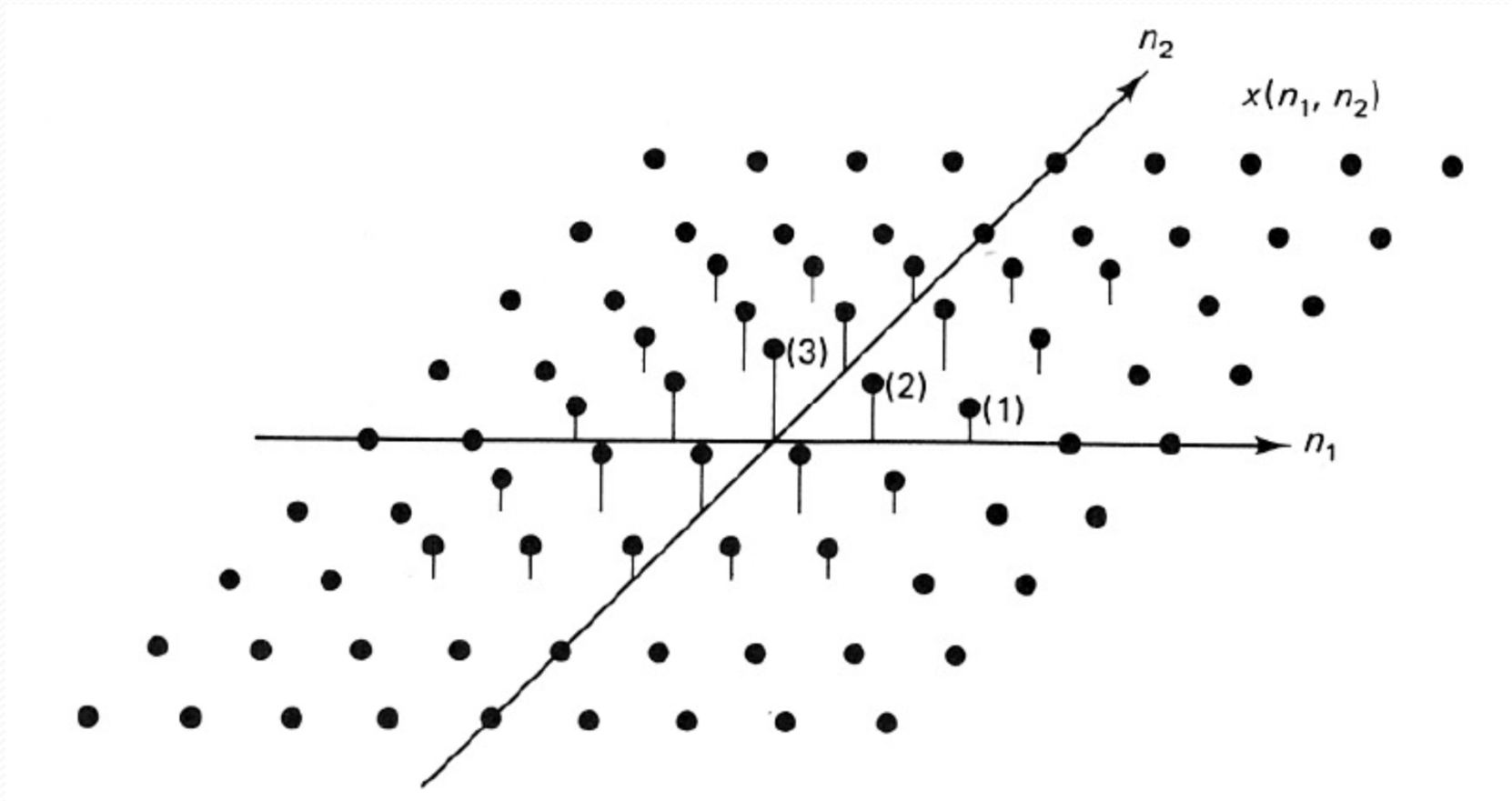


Image as a 2D sampling

- A digital image can be considered as a 2D discrete signal

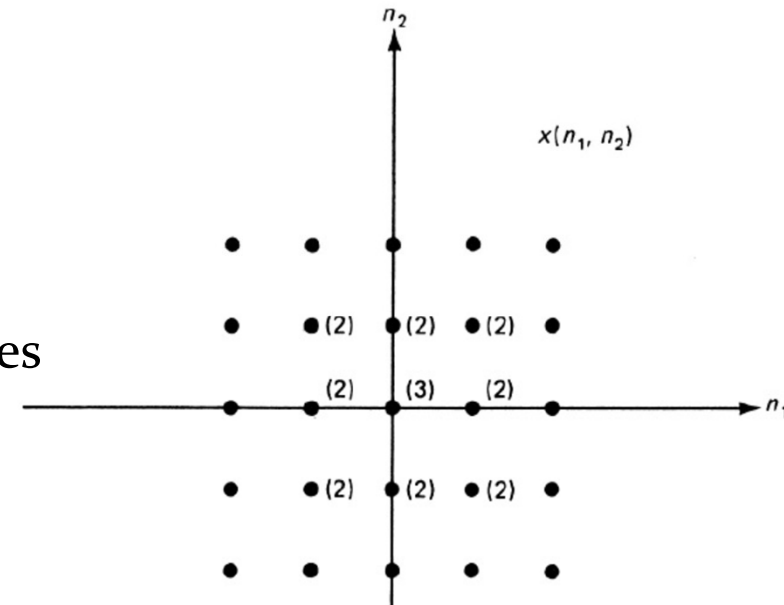
$$x(n_1, n_2)$$



Impulse definition

$$\delta(n_1, n_2) = \begin{cases} 1, & n_1 = n_2 = 0 \\ 0, & \text{otherwise.} \end{cases}$$

Each sequence can be considered as a sequence of impulses



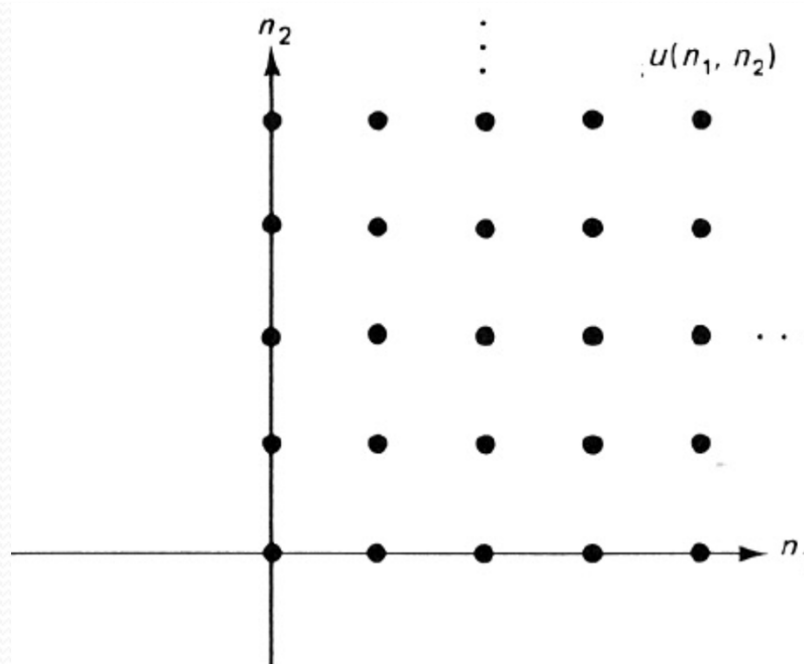
$$\begin{aligned} x(n_1, n_2) &= \cdots + x(-1, -1)\delta(n_1 + 1, n_2 + 1) + x(0, -1)\delta(n_1, n_2 + 1) \\ &\quad + x(1, -1)\delta(n_1 - 1, n_2 + 1) + \cdots + x(-1, 0)\delta(n_1 + 1, n_2) \\ &\quad + x(0, 0)\delta(n_1, n_2) + x(1, 0)\delta(n_1 - 1, n_2) \\ &\quad + \cdots + x(-1, 1)\delta(n_1 + 1, n_2 - 1) \\ &\quad + x(0, 1)\delta(n_1, n_2 - 1) + x(1, 1)\delta(n_1 - 1, n_2 - 1) + \cdots \\ &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2)\delta(n_1 - k_1, n_2 - k_2). \end{aligned}$$

The step function

- The step function a combination of impulses.

$$u(n_1, n_2) = \sum_{k_1=-\infty}^{n_1} \sum_{k_2=-\infty}^{n_2} \delta(k_1, k_2)$$

$$\delta(n_1, n_2) = u(n_1, n_2) - u(n_1 - 1, n_2) - u(n_1, n_2 - 1) + u(n_1 - 1, n_2 - 1).$$



Separable sequences

- A separable 2D sequence can be written as:

$$x(n_1, n_2) = f(n_1)g(n_2)$$

- The impulse and step function are separable functions

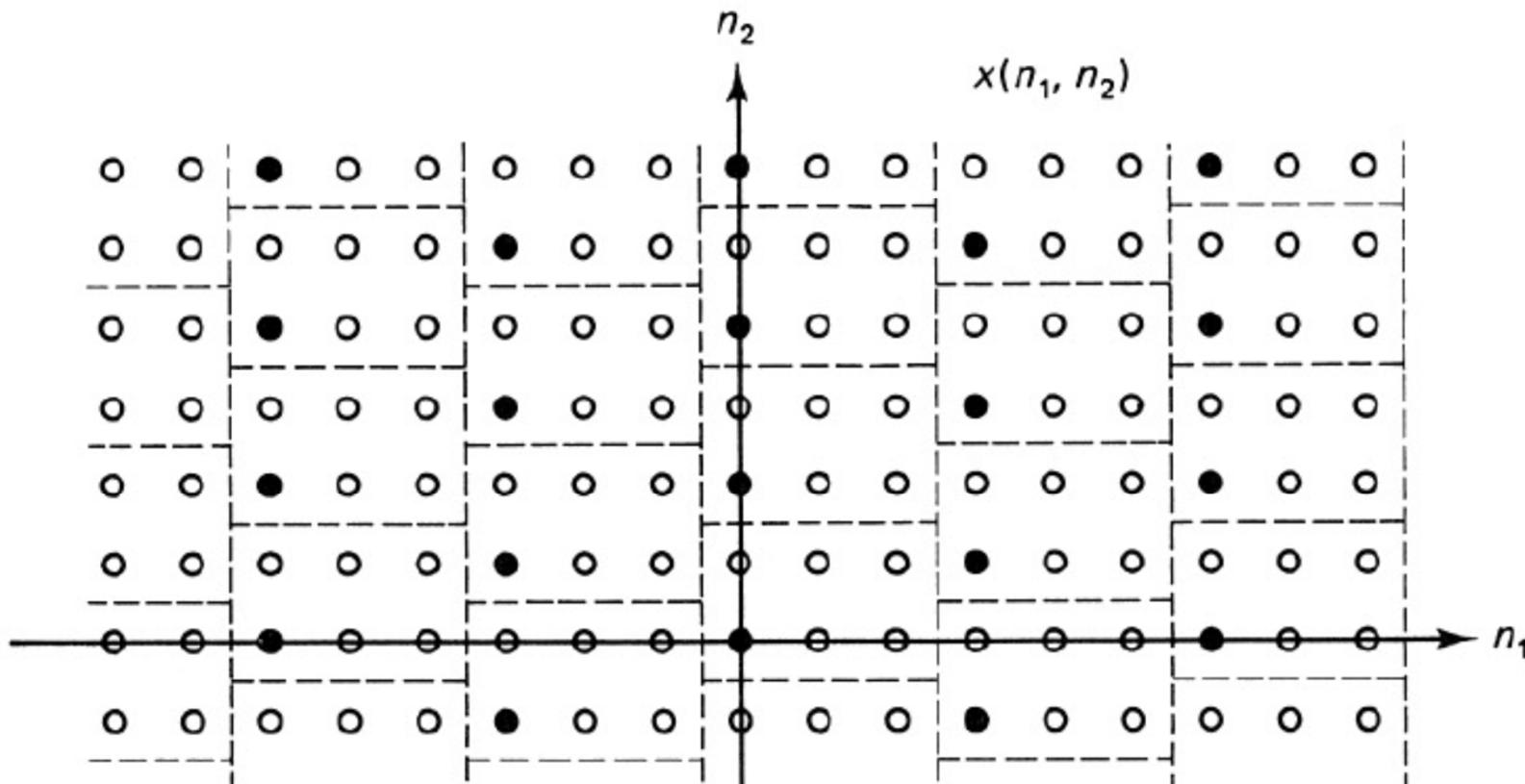
$$\delta(n_1, n_2) = \delta(n_1) \delta(n_2)$$

$$u(n_1, n_2) = u(n_1)u(n_2)$$

Periodic sequences

- A sequence $x(n_1, n_2)$ is periodic of period $N_1 \times N_2$ if:

$$x(n_1, n_2) = x(n_1 + N_1, n_2) = x(n_1, n_2 + N_2) \text{ for all } (n_1, n_2)$$



LTI Systems

- Linearity

$$T[ax_1(n_1, n_2) + bx_2(n_1, n_2)] = ay_1(n_1, n_2) + by_2(n_1, n_2)$$

- Spatial invariance

$$T[x(n_1 - m_1, n_2 - m_2)] = y(n_1 - m_1, n_2 - m_2)$$

- The impulse response

$$\begin{aligned} y(n_1, n_2) &= T[x(n_1, n_2)] = T\left[\sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) \delta(n_1 - k_1, n_2 - k_2)\right] \\ &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) T[\delta(n_1 - k_1, n_2 - k_2)]. \end{aligned}$$

Convolution

- Defined the impulse response

$$h(n_1, n_2) = T[\delta(n_1, n_2)].$$

- The Input/Output relation is given by:

$$y(n_1, n_2) = T[x(n_1, n_2)] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2)h(n_1 - k_1, n_2 - k_2).$$

$$y(n_1, n_2) = x(n_1, n_2) * h(n_1, n_2)$$

$$= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2)h(n_1 - k_1, n_2 - k_2).$$

Convolution properties

Commutativity

$$x(n_1, n_2) * y(n_1, n_2) = y(n_1, n_2) * x(n_1, n_2)$$

Associativity

$$(x(n_1, n_2) * y(n_1, n_2)) * z(n_1, n_2) = x(n_1, n_2) * (y(n_1, n_2) * z(n_1, n_2))$$

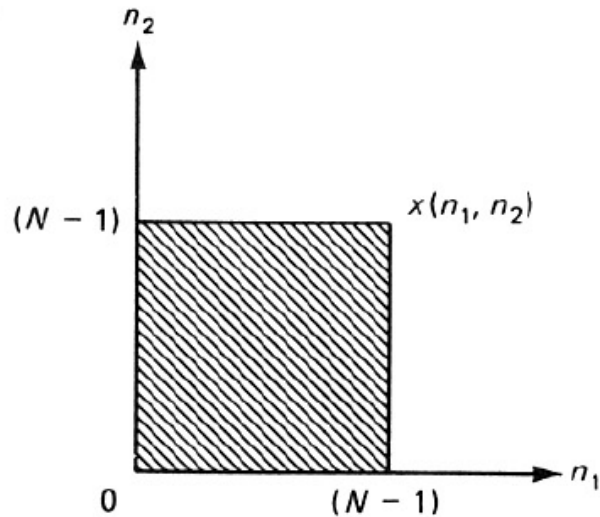
Distributivity

$$\begin{aligned} x(n_1, n_2) * (y(n_1, n_2) + z(n_1, n_2)) \\ = (x(n_1, n_2) * y(n_1, n_2)) + (x(n_1, n_2) * z(n_1, n_2)) \end{aligned}$$

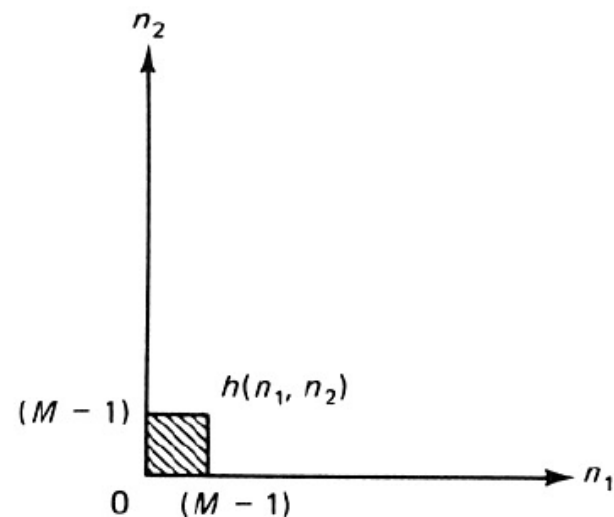
Convolution with Shifted Impulse

$$x(n_1, n_2) * \delta(n_1 - m_1, n_2 - m_2) = x(n_1 - m_1, n_2 - m_2)$$

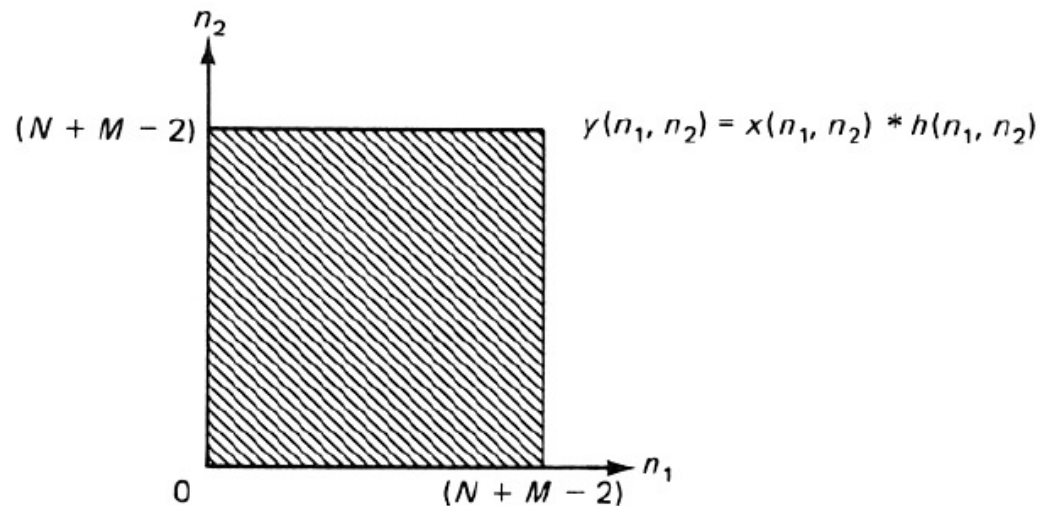
Convolution examples



(a)



(b)



The 2D Fourier Transform

- The analysis and synthesis formulas for the 2D continuous Fourier transform are as follows:
- Analysis

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy$$

- Synthesis

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} du dv$$

Separability of 2D Fourier Transform

- The 2D analysis formula can be written as a 1D analysis in the x direction followed by a 1D analysis in the y direction:

$$F(u, v) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(x, y) e^{-j2\pi ux} dx \right] e^{-j2\pi vy} dy$$

- The 2D synthesis formula can be written as a 1D synthesis in the x direction followed by a 1D synthesis in y direction:

$$f(x, y) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} F(u, v) e^{j2\pi ux} du \right] e^{j2\pi vy} dv$$

Separability Theorem

$$f(x, y) = f(x)g(y) \xrightarrow{\mathcal{F}} F(u, v) = F(u)G(v)$$

Proof:

$$F(u, v)$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy$$

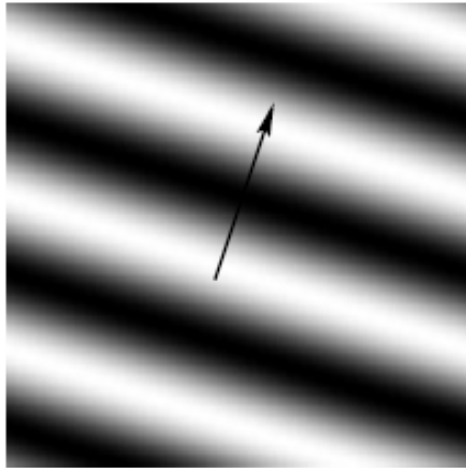
$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x)g(y) e^{-j2\pi ux} e^{-j2\pi vy} dx dy$$

$$= \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx \int_{-\infty}^{\infty} g(y) e^{-j2\pi vy} dy$$

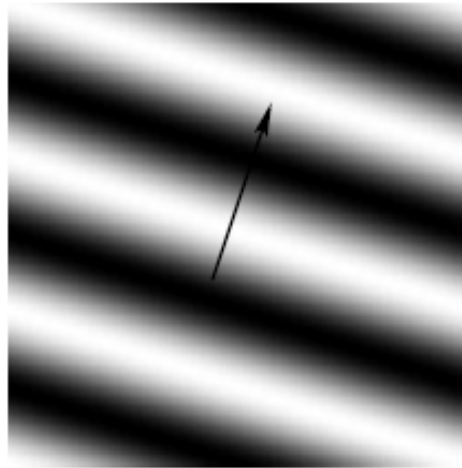
$$= F(u) G(v)$$

2D Fourier Basis Functions

Grating for $(k,l) = (1,-3)$

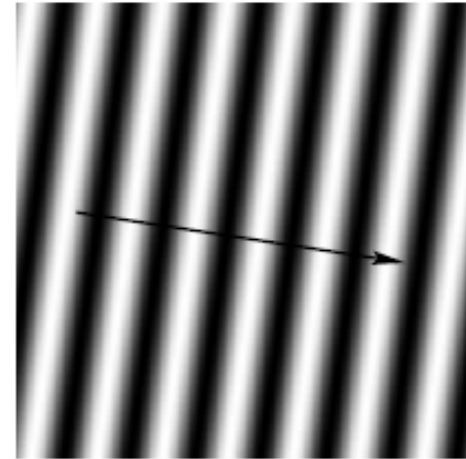


Real

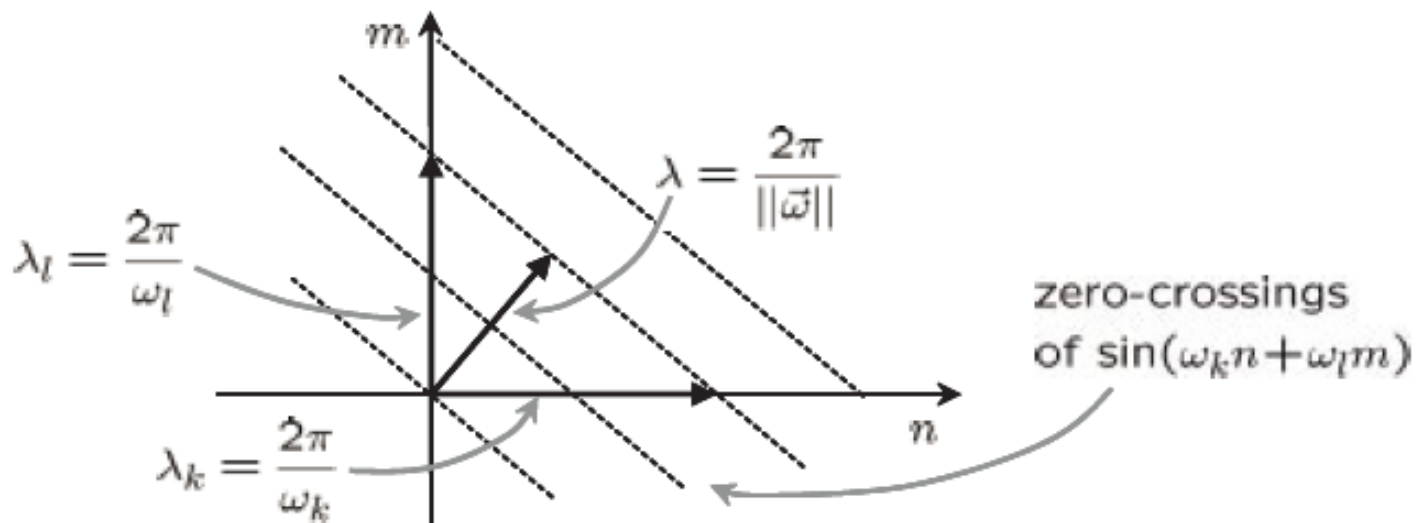


Imag

Grating for $(k,l) = (7,1)$



Real



The 2D Discrete Fourier Transform for periodic signals

- The analysis and synthesis formulas for the 2D discrete Fourier transform are as follows:

$$\hat{F}(k, \ell) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F(m, n) e^{-j2\pi(k\frac{m}{M} + \ell\frac{n}{N})}$$

$$F(m, n) = \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1} \sum_{\ell=0}^{N-1} \hat{F}(k, \ell) e^{j2\pi(k\frac{m}{M} + \ell\frac{n}{N})}$$

Separability of 2D DFT

$$\hat{F}(k, \ell) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \left[\frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} F(m, n) e^{-j2\pi(k\frac{m}{M})} \right] e^{-j2\pi(\ell\frac{n}{N})}$$

- The 2D forward DFT can be written in matrix notation:

$$\hat{\mathbf{F}} = (\mathbf{W}^* \mathbf{F}) \mathbf{W}^*$$

- Where
$$W_{rc}^* = \frac{1}{\sqrt{C}} e^{-j2\pi r \frac{c}{C}}$$

Separability of 2D DFT

- And

$$F(m, n) = \frac{1}{\sqrt{N}} \sum_{\ell=0}^{N-1} \left[\frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} \hat{F}(k, \ell) e^{j2\pi(k\frac{m}{M})} \right] e^{j2\pi(\ell\frac{n}{N})}.$$

- The 2D inverse DFT can be written in matrix notation:

$$\mathbf{F} = (\mathbf{W}\hat{\mathbf{F}}) \mathbf{W}$$

- where

$$W_{rc} = \frac{1}{\sqrt{C}} e^{j2\pi r\frac{c}{C}}$$

The 2D Discrete Time Fourier Transform

Discrete-Space Fourier Transform Pair

$$X(\omega_1, \omega_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2) e^{-j\omega_1 n_1} e^{-j\omega_2 n_2}$$

$$x(n_1, n_2) = \frac{1}{(2\pi)^2} \int_{\omega_1=-\pi}^{\pi} \int_{\omega_2=-\pi}^{\pi} X(\omega_1, \omega_2) e^{j\omega_1 n_1} e^{j\omega_2 n_2} d\omega_1 d\omega_2$$

La trasformata di un segnale può essere espresso come componente reale e immaginaria o come ampiezza e fase.

$$X(\omega_1, \omega_2) = |X(\omega_1, \omega_2)| e^{j\theta_x(\omega_1, \omega_2)} = X_R(\omega_1, \omega_2) + jX_I(\omega_1, \omega_2).$$

Fourier Transform properties

$$\begin{aligned}x(n_1, n_2) &\longleftrightarrow X(\omega_1, \omega_2) \\ y(n_1, n_2) &\longleftrightarrow Y(\omega_1, \omega_2)\end{aligned}$$

Property 1. Linearity

$$ax(n_1, n_2) + by(n_1, n_2) \longleftrightarrow aX(\omega_1, \omega_2) + bY(\omega_1, \omega_2)$$

Property 2. Convolution

$$x(n_1, n_2) * y(n_1, n_2) \longleftrightarrow X(\omega_1, \omega_2)Y(\omega_1, \omega_2)$$

Property 3. Multiplication

$$x(n_1, n_2)y(n_1, n_2) \longleftrightarrow X(\omega_1, \omega_2) \odot Y(\omega_1, \omega_2)$$

$$= \frac{1}{(2\pi)^2} \int_{\theta_1 = -\pi}^{\pi} \int_{\theta_2 = -\pi}^{\pi} X(\theta_1, \theta_2)Y(\omega_1 - \theta_1, \omega_2 - \theta_2) d\theta_1 d\theta_2$$

Property 4. Separable Sequence

$$x(n_1, n_2) = x_1(n_1)x_2(n_2) \longleftrightarrow X(\omega_1, \omega_2) = X_1(\omega_1)X_2(\omega_2)$$

Property 5. Shift of a Sequence and a Fourier Transform

$$(a) \ x(n_1 - m_1, n_2 - m_2) \longleftrightarrow X(\omega_1, \omega_2)e^{-j\omega_1 m_1}e^{-j\omega_2 m_2}$$

$$(b) \ e^{jv_1 m_1}e^{jv_2 m_2}x(n_1, n_2) \longleftrightarrow X(\omega_1 - v_1, \omega_2 - v_2)$$

Fourier Transform properties

Property 6. Differentiation

$$(a) -jn_1x(n_1, n_2) \longleftrightarrow \frac{\partial X(\omega_1, \omega_2)}{\partial \omega_1}$$

$$(b) -jn_2x(n_1, n_2) \longleftrightarrow \frac{\partial X(\omega_1, \omega_2)}{\partial \omega_2}$$

Property 7. Initial Value and DC Value Theorem

$$(a) x(0, 0) = \frac{1}{(2\pi)^2} \int_{\omega_1=-\pi}^{\pi} \int_{\omega_2=-\pi}^{\pi} X(\omega_1, \omega_2) d\omega_1 d\omega_2$$

$$(b) X(0, 0) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2)$$

Property 8. Parseval's Theorem

$$(a) \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2)y^*(n_1, n_2) = \frac{1}{(2\pi)^2} \int_{\omega_1=-\pi}^{\pi} \int_{\omega_2=-\pi}^{\pi} X(\omega_1, \omega_2)Y^*(\omega_1, \omega_2) d\omega_1 d\omega_2$$

$$(b) \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |x(n_1, n_2)|^2 = \frac{1}{(2\pi)^2} \int_{\omega_1=-\pi}^{\pi} \int_{\omega_2=-\pi}^{\pi} |X(\omega_1, \omega_2)|^2 d\omega_1 d\omega_2$$

Fourier Transform properties

Property 9. Symmetry Properties

(a) $x(-n_1, n_2) \longleftrightarrow X(-\omega_1, \omega_2)$

(b) $x(n_1, -n_2) \longleftrightarrow X(\omega_1, -\omega_2)$

(c) $x(-n_1, -n_2) \longleftrightarrow X(-\omega_1, -\omega_2)$

(d) $x^*(n_1, n_2) \longleftrightarrow X^*(-\omega_1, -\omega_2)$

(e) $x(n_1, n_2): \text{real} \longleftrightarrow X(\omega_1, \omega_2) = X^*(-\omega_1, -\omega_2)$

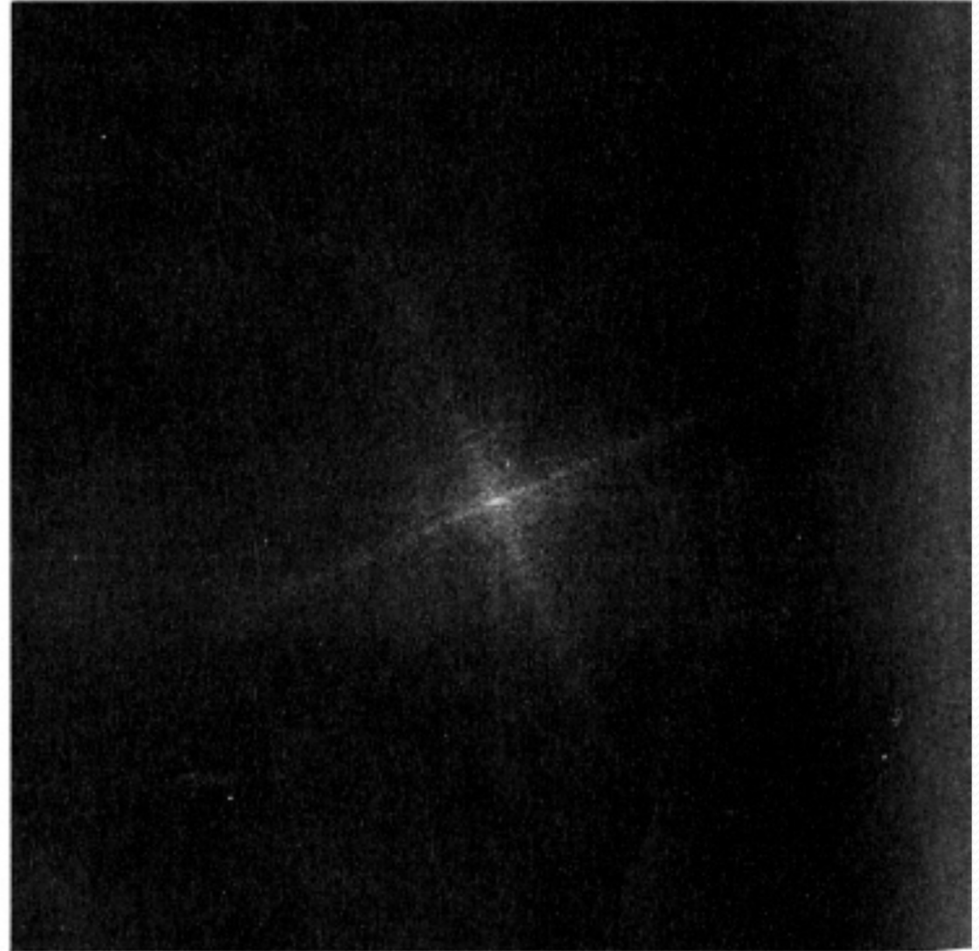
$X_R(\omega_1, \omega_2), |X(\omega_1, \omega_2)|$: even (symmetric with respect to the origin)

$X_I(\omega_1, \omega_2), \theta_x(\omega_1, \omega_2)$: odd (antisymmetric with respect to the origin)

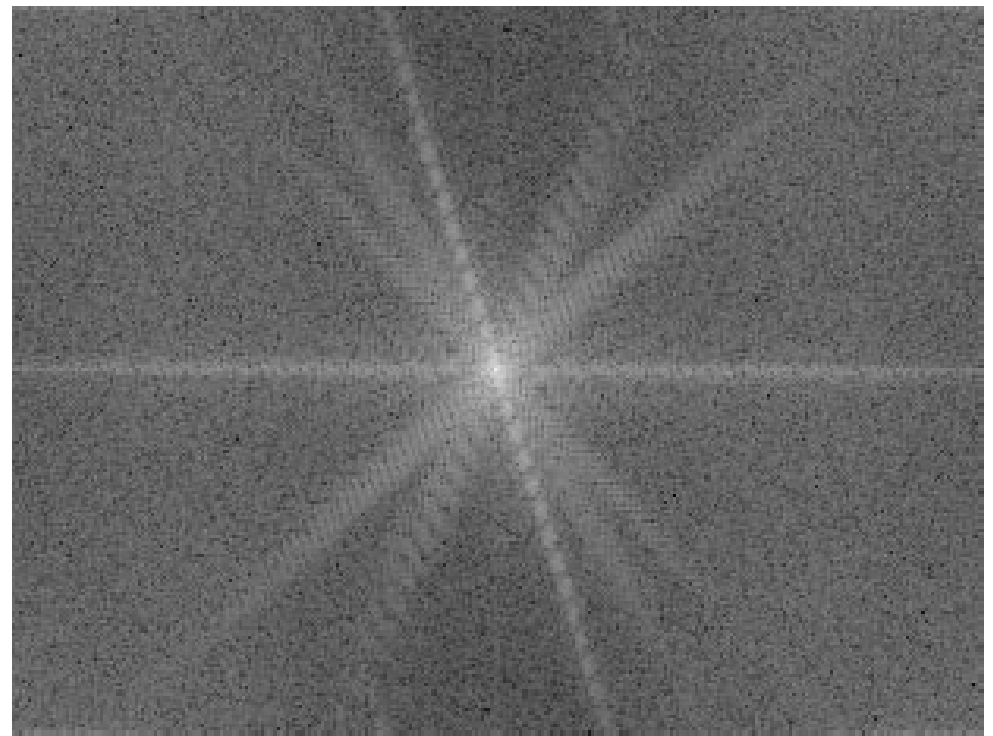
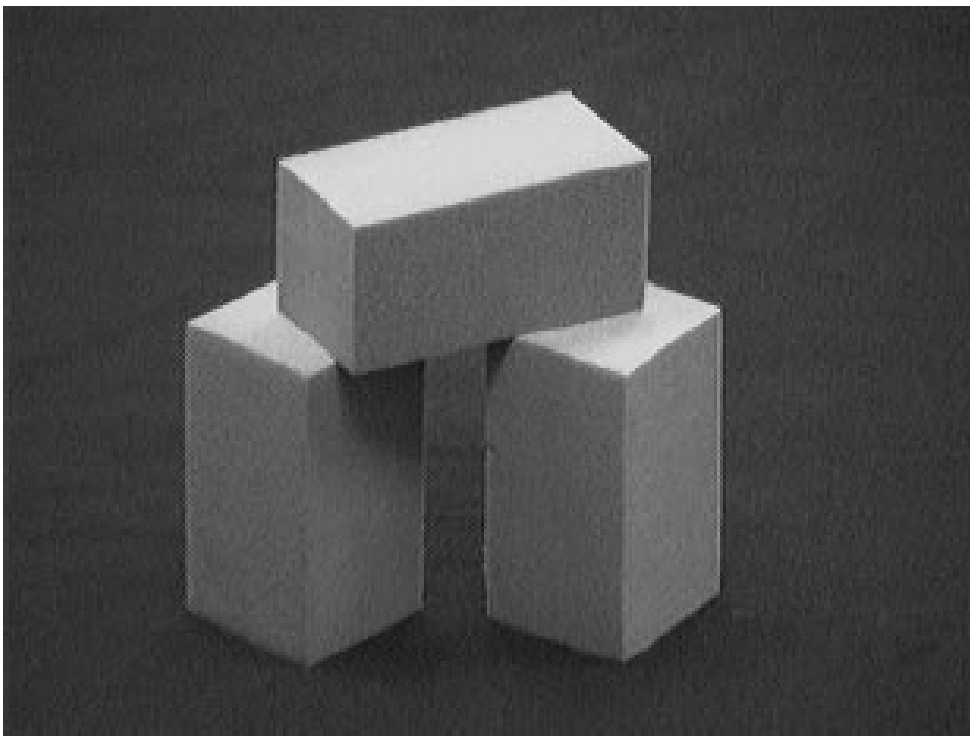
(f) $x(n_1, n_2): \text{real and even} \longleftrightarrow X(\omega_1, \omega_2): \text{real and even}$

(g) $x(n_1, n_2): \text{real and odd} \longleftrightarrow X(\omega_1, \omega_2): \text{pure imaginary and odd}$

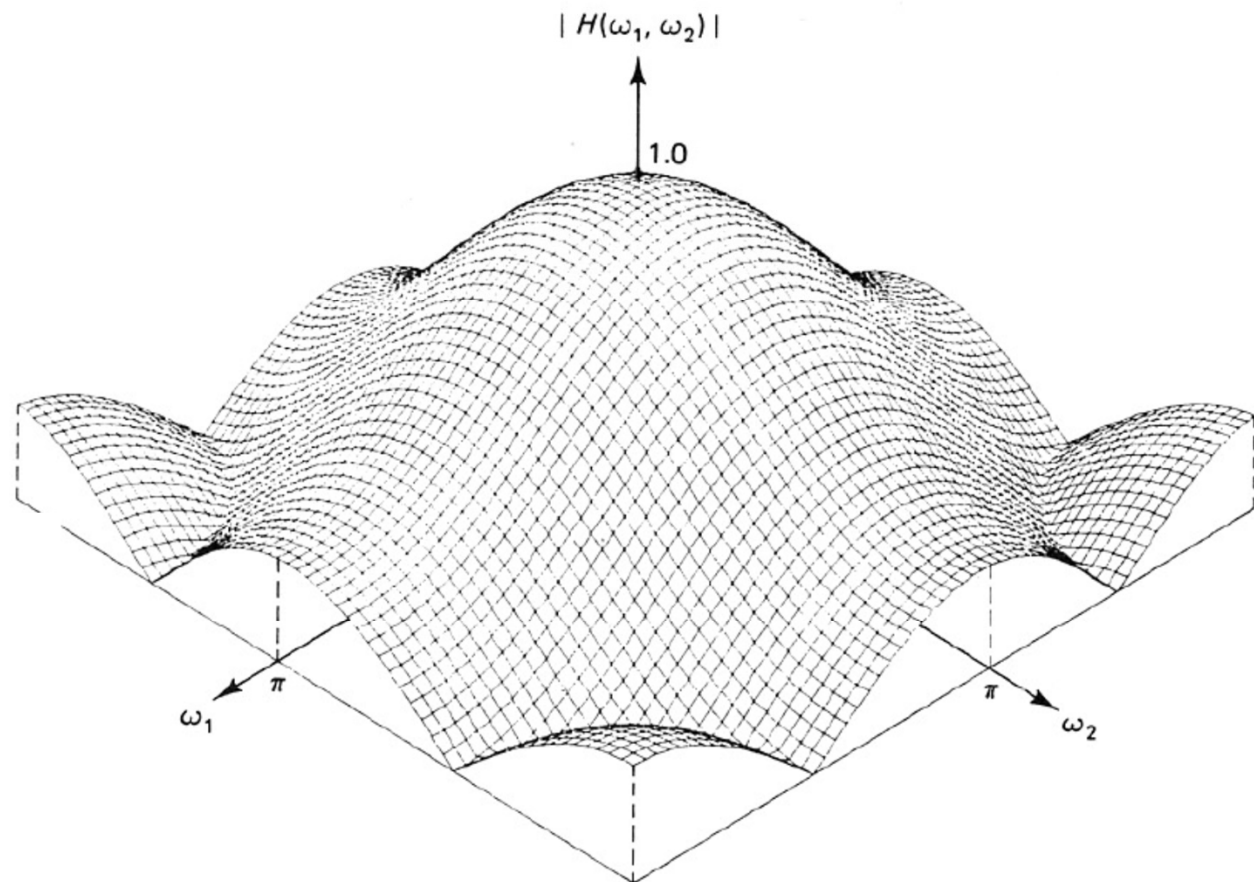
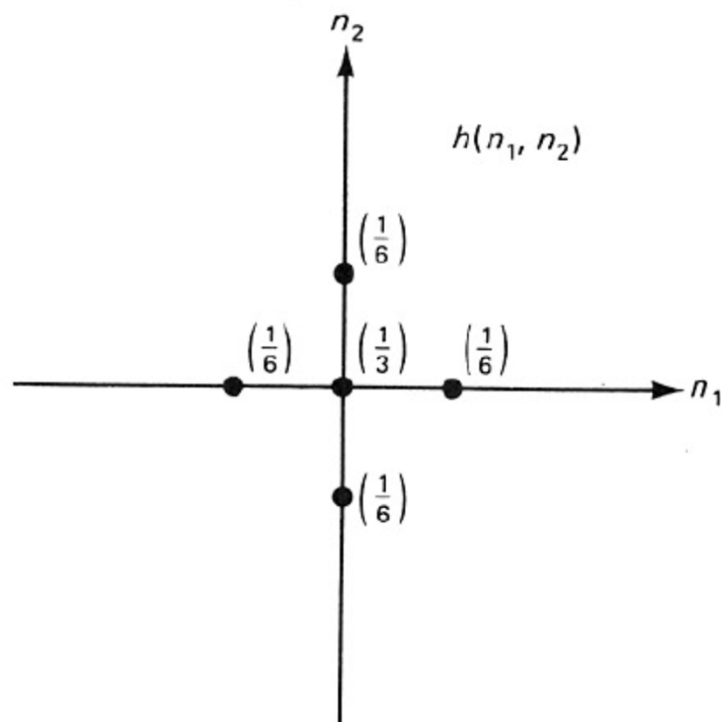
Transform examples



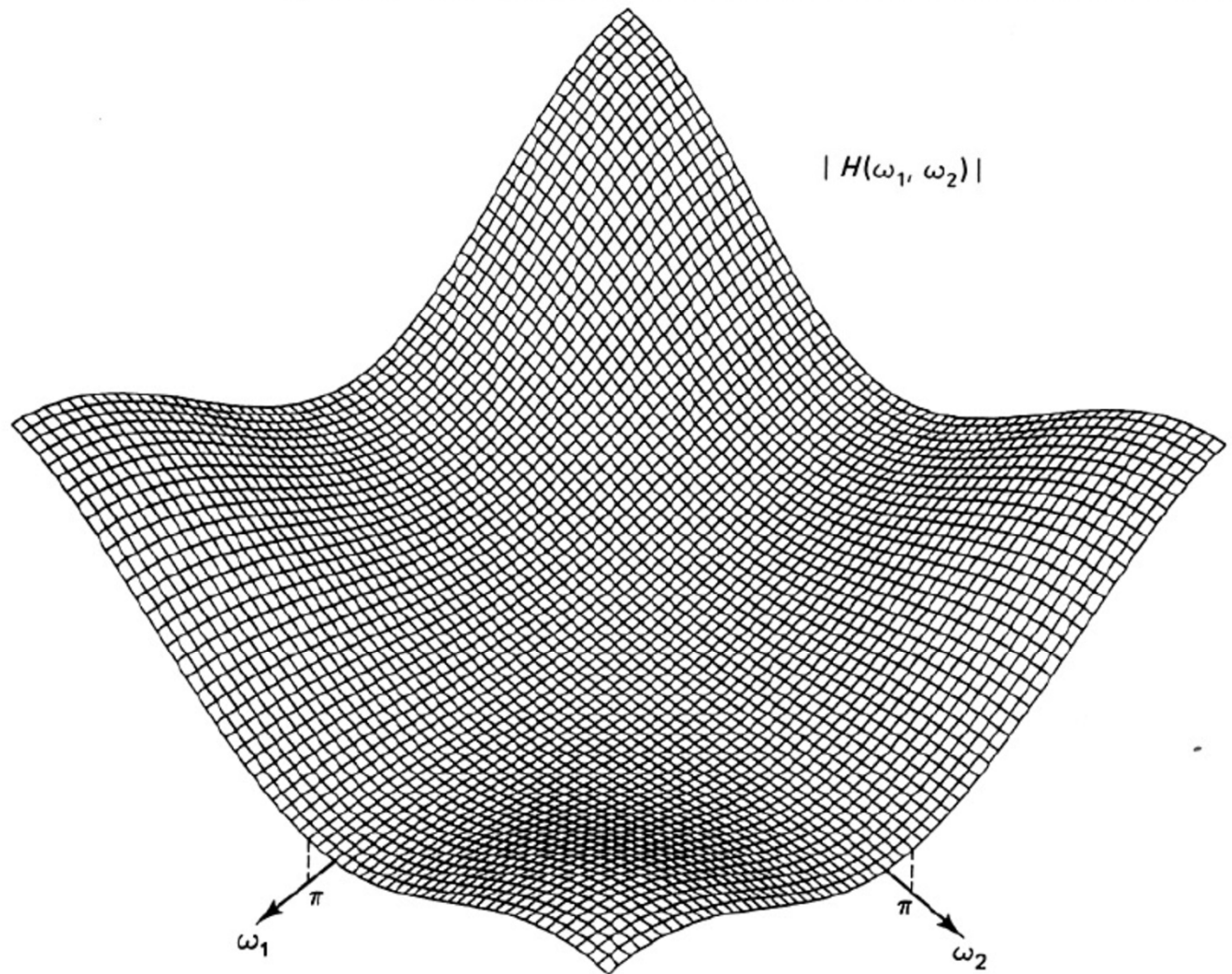
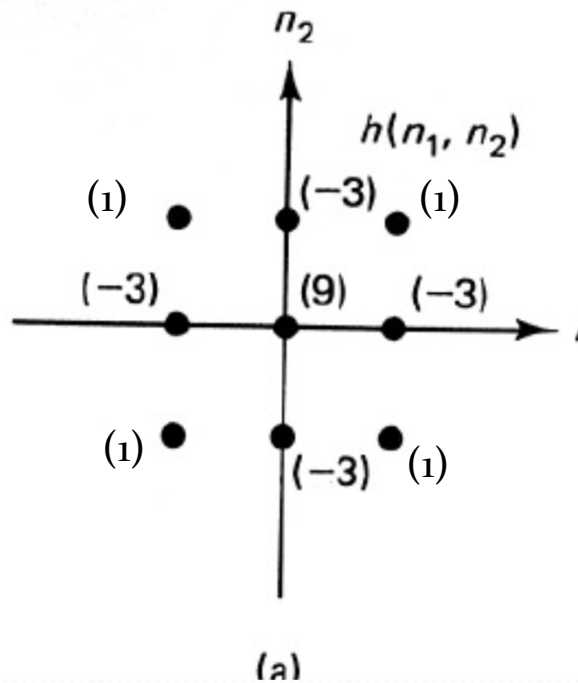
2D DFT Example



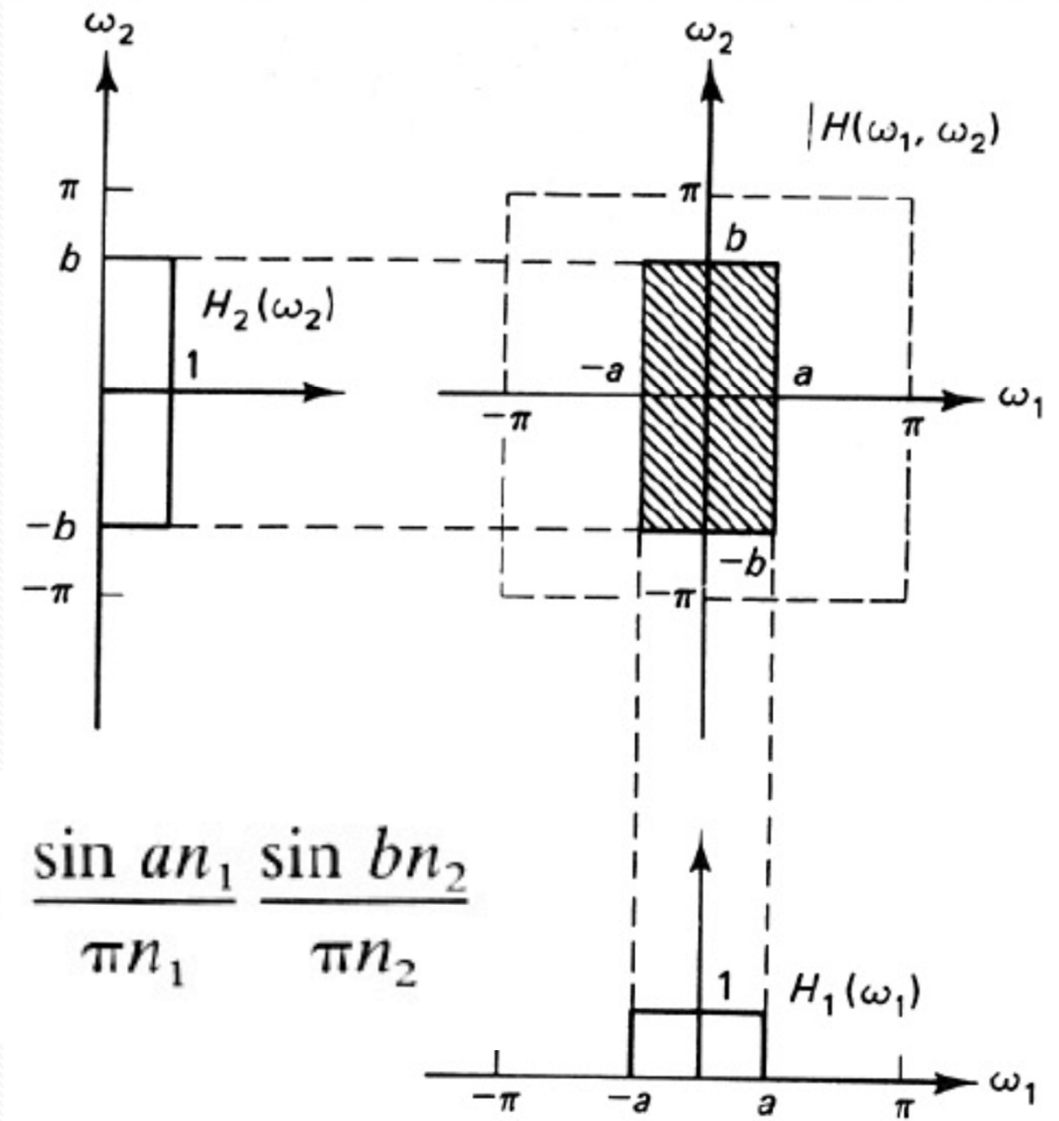
DTFT exam



Hi-Pass Filter example



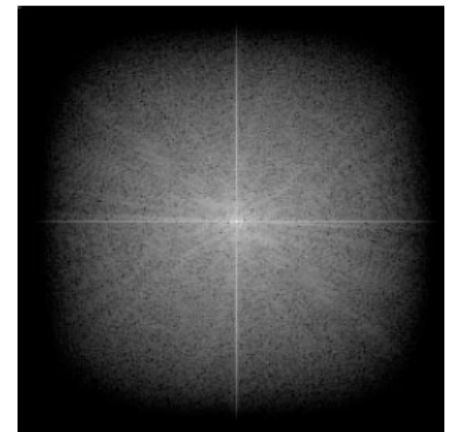
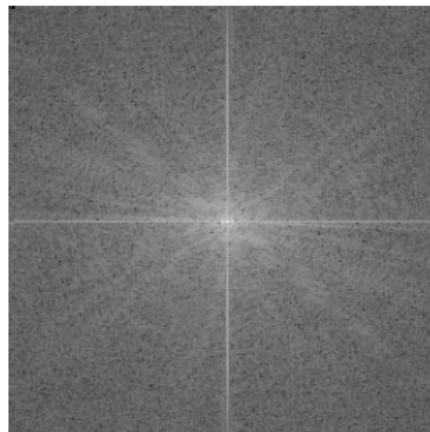
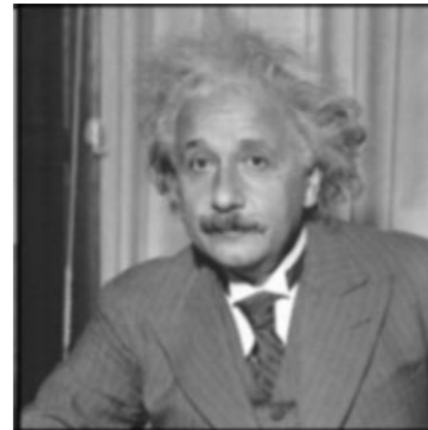
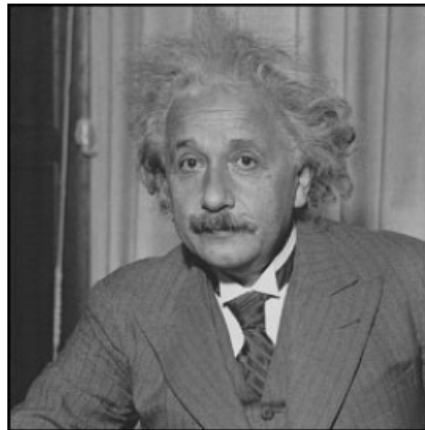
Separable Low-pass Filter example



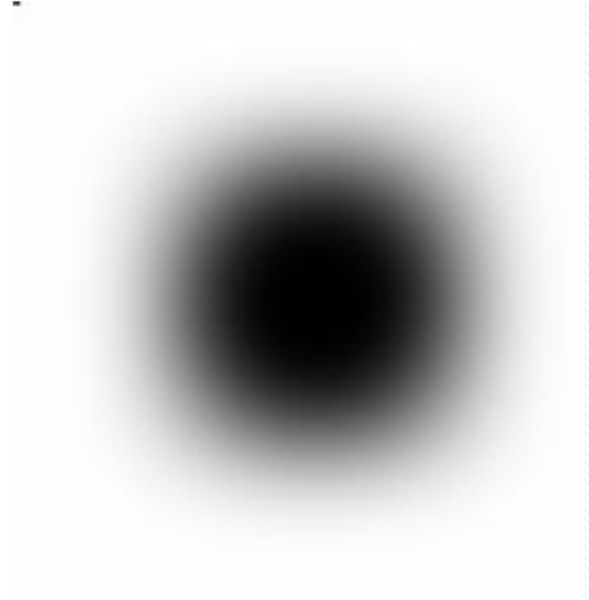
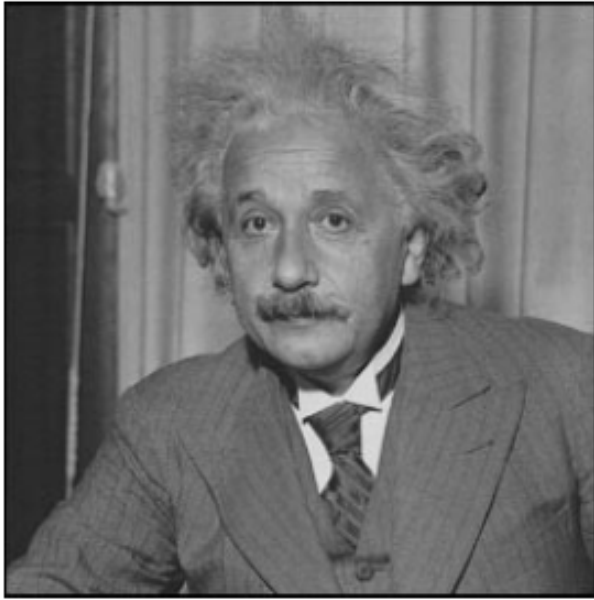
$$h(n_1, n_2) = h_1(n_1)h_2(n_2) = \frac{\sin an_1}{\pi n_1} \frac{\sin bn_2}{\pi n_2}$$

Image of Albert and a low-pass (blurred) version of it

- $h(n) = \frac{1}{16} (1, 4, 6, 4, 1)$ (1D impulse response in both dimensions)

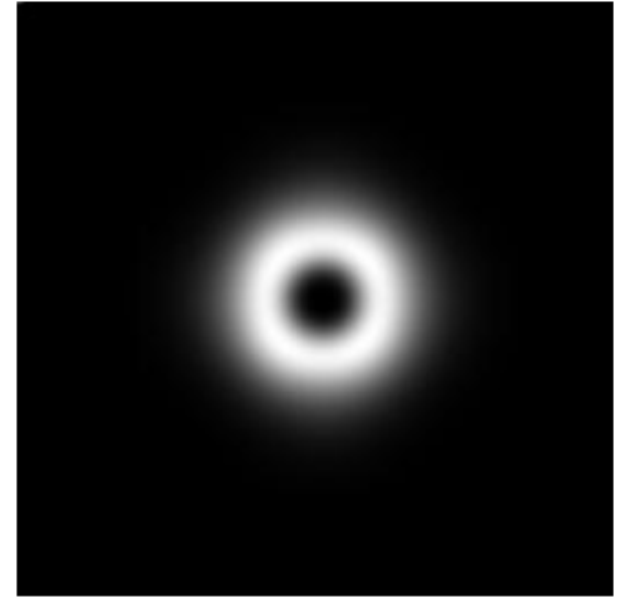
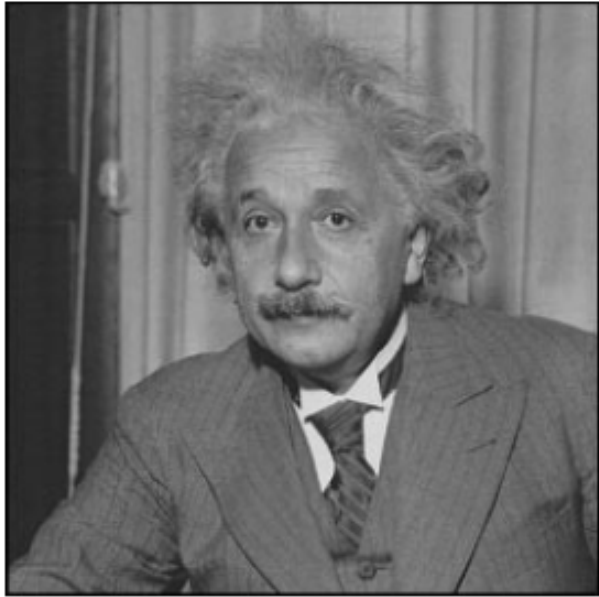


Hi-Pass Filter



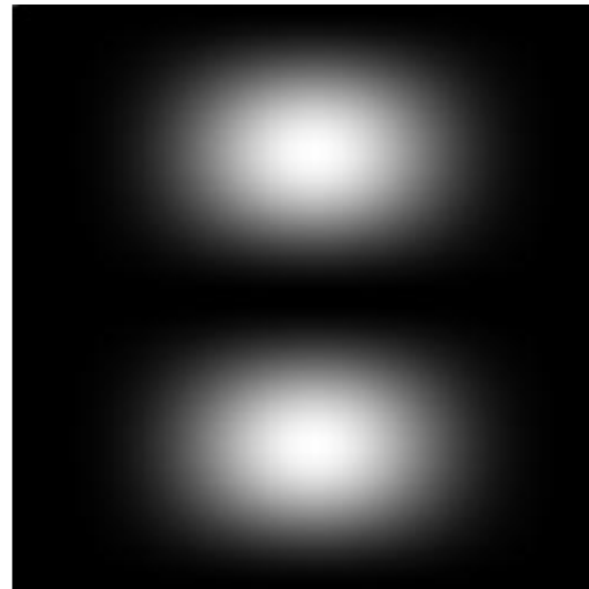
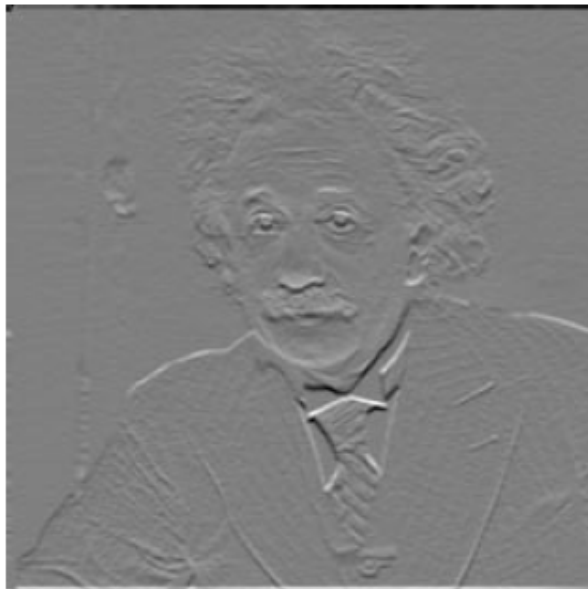
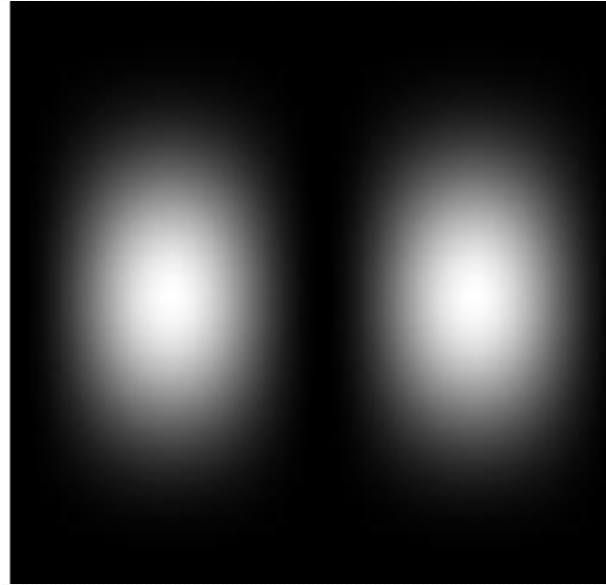
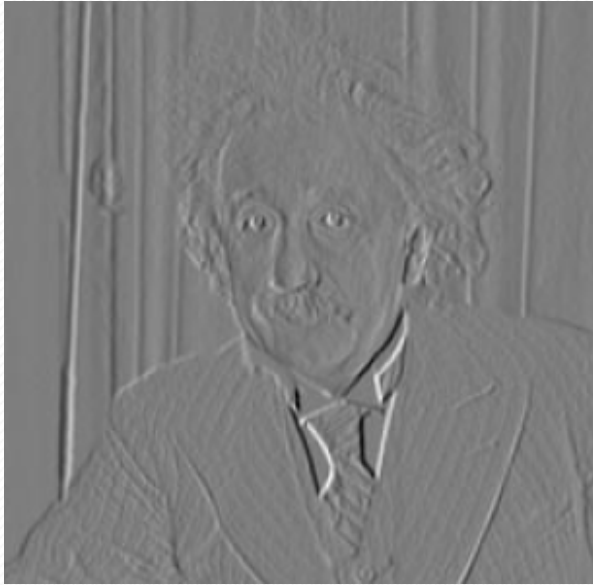
- a high-pass filtered version of Albert, and the amplitude spectrum of the filter. This impulse response is defined by $\delta(n) - h(n, m)$ where $h(n, m)$ is the separable blurring kernel used in the previous figure

Band Pass Filter

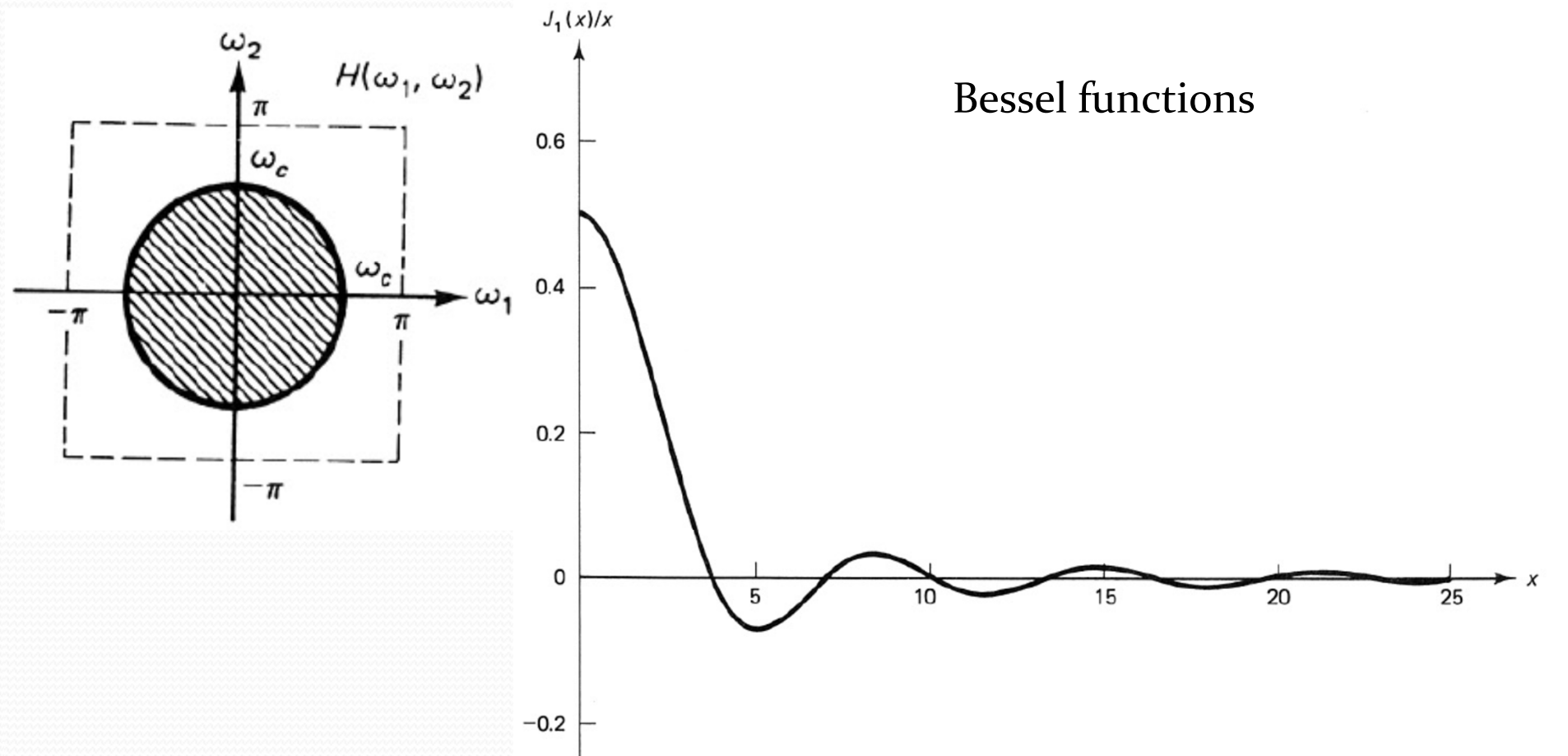


- a band-pass filtered version of Albert, and the amplitude spectrum of the filter.
- This impulse response is defined by the difference of two low-pass filters.

Directional filters



Circular filter example



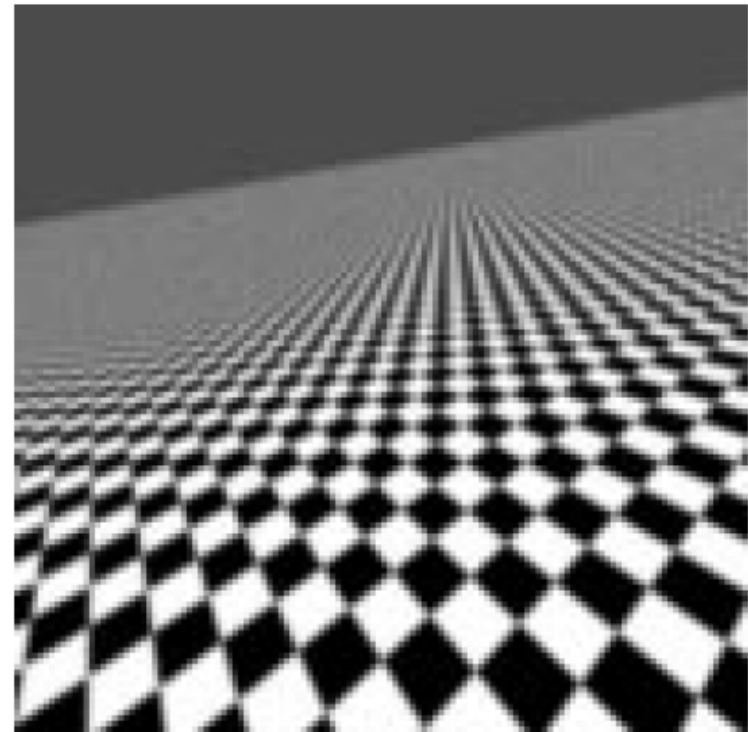
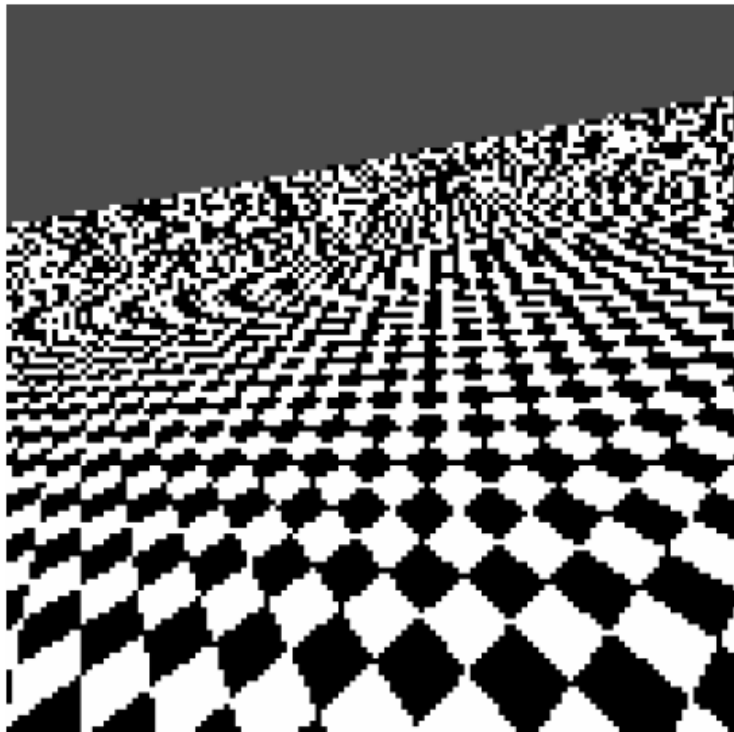
$$h(n_1, n_2) = \frac{\omega_c}{2\pi\sqrt{n_1^2 + n_2^2}} J_1(\omega_c \sqrt{n_1^2 + n_2^2})$$

Nyquist Sampling Theorem and Aliasing

- Consider a perspective image of an infinite checkerboard.
- The signal is dominated by high frequencies in the image near the horizon.
- Properly designed cameras blur the signal before sampling using:
 - The point spread function due to diffraction
 - Imperfect focus
 - Averaging the signal over each CCD element.

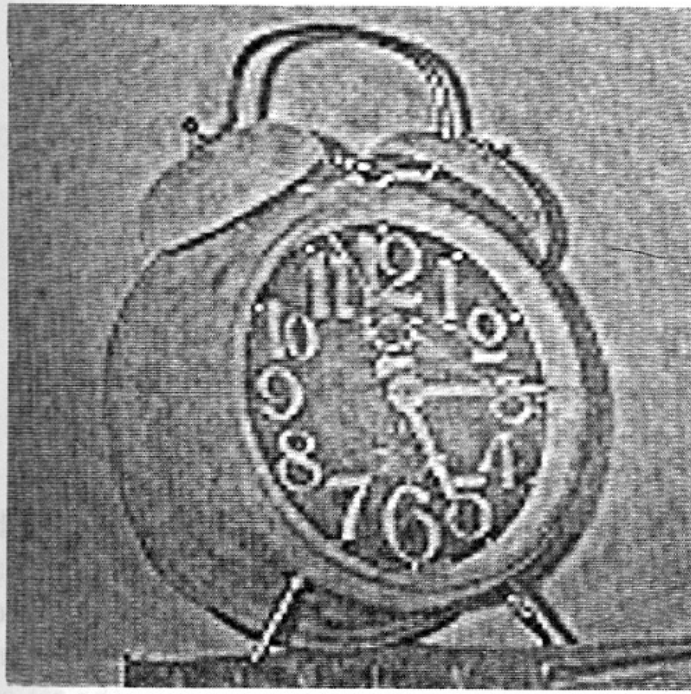
Nyquist Sampling Theorem and Aliasing

- These operations attenuate high frequency components in the signal.
- Without this (physical) preprocessing, the sampled image can be severely aliased (corrupted):



Reconstruction using just phase or intensity

Only phase



Only amplitude

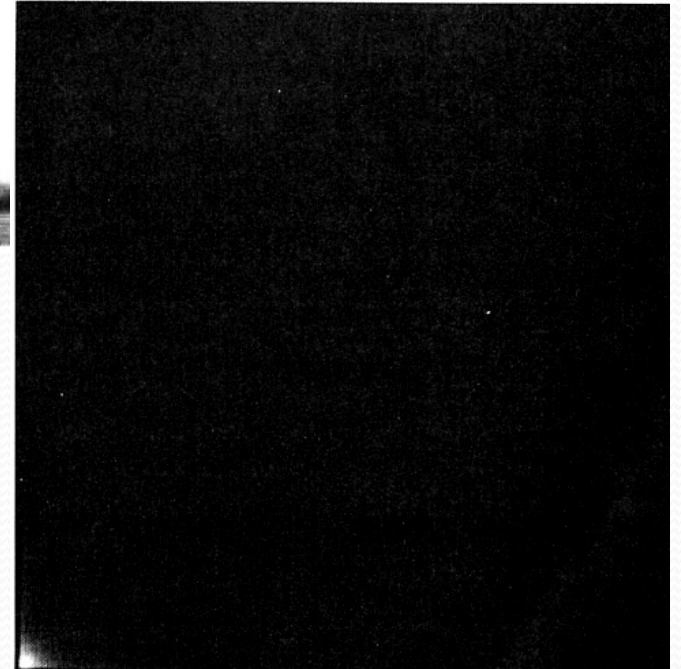
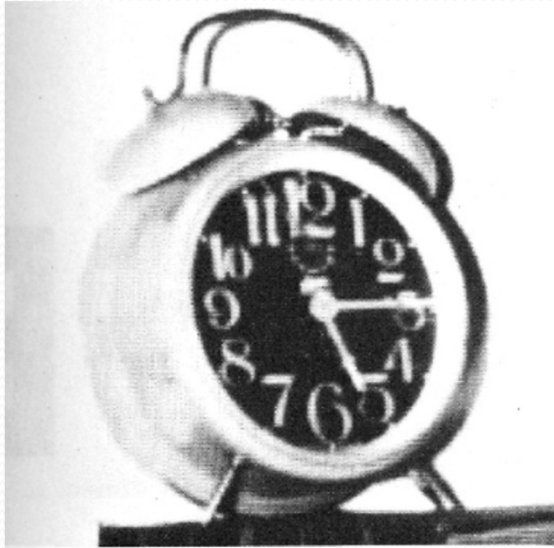
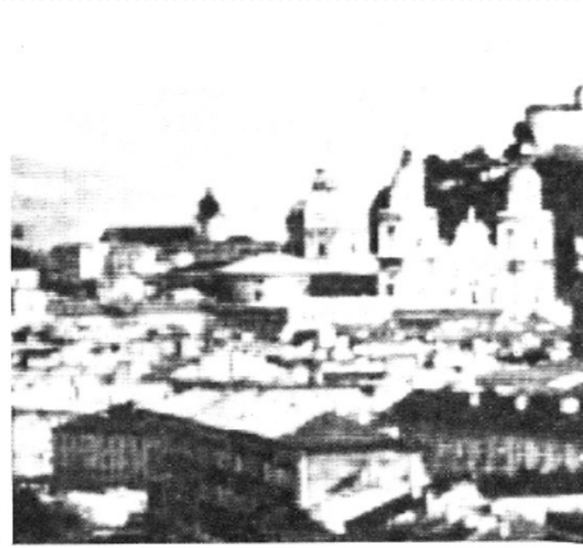


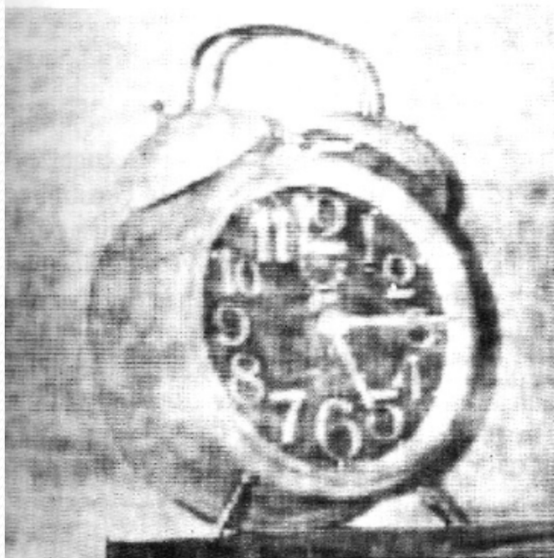
Image superposition combining phases and intensities



(a)



(b)



Filtering examples



Original
Cameraman

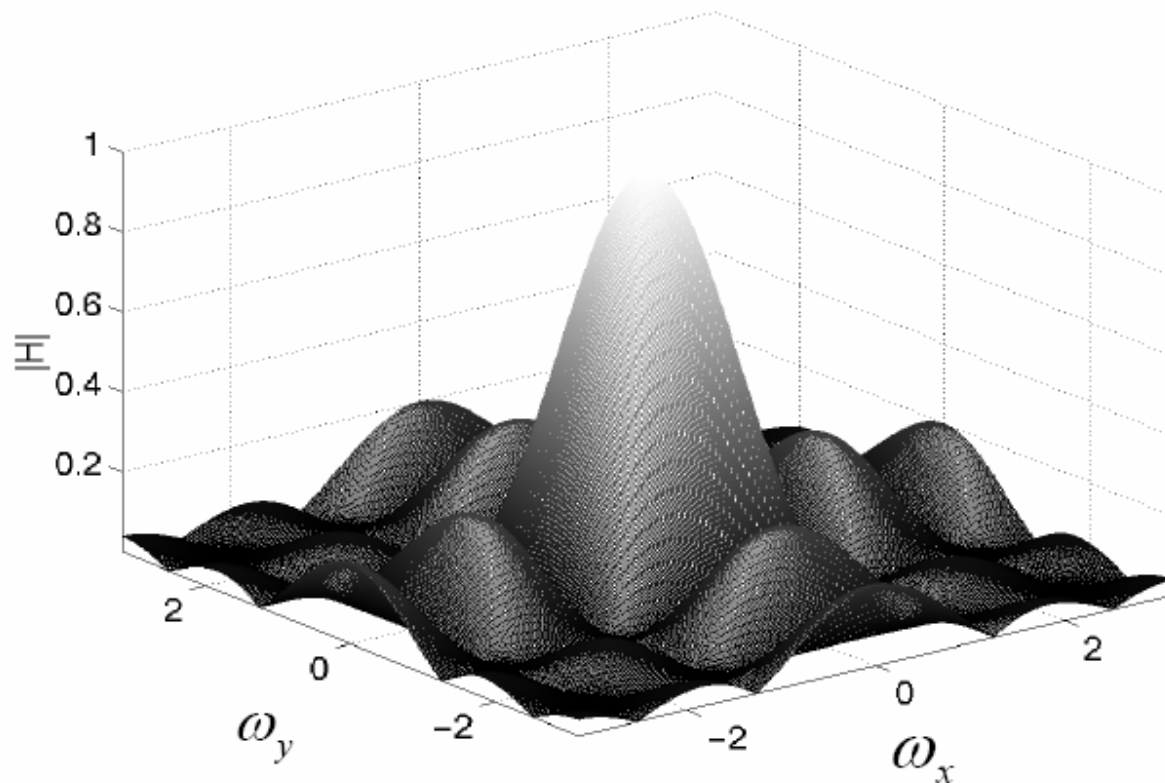


Cameraman blurred by convolution
Filter impulse response

$$\frac{1}{25} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & [1] & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Fourier interpretation

$$\begin{aligned} H(e^{j\omega_x}, e^{j\omega_y}) &= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h[m, n] e^{-j\omega_x m - j\omega_y n} \\ &= \frac{1}{25} \sum_{m=-2}^2 \sum_{n=-2}^2 e^{-j\omega_x m - j\omega_y n} = \frac{1}{25} \sum_{m=-2}^2 e^{-j\omega_x m} \sum_{n=-2}^2 e^{-j\omega_y n} \\ &= \frac{1}{25} (1 + 2 \cos \omega_x + 2 \cos(2\omega_x)) (1 + 2 \cos \omega_y + 2 \cos(2\omega_y)) \end{aligned}$$



Filtering Examples



Original
Cameraman



Cameraman blurred horizontally
Filter impulse response

$$\frac{1}{5} \begin{pmatrix} 1 & 1 & [1] & 1 & 1 \end{pmatrix}$$

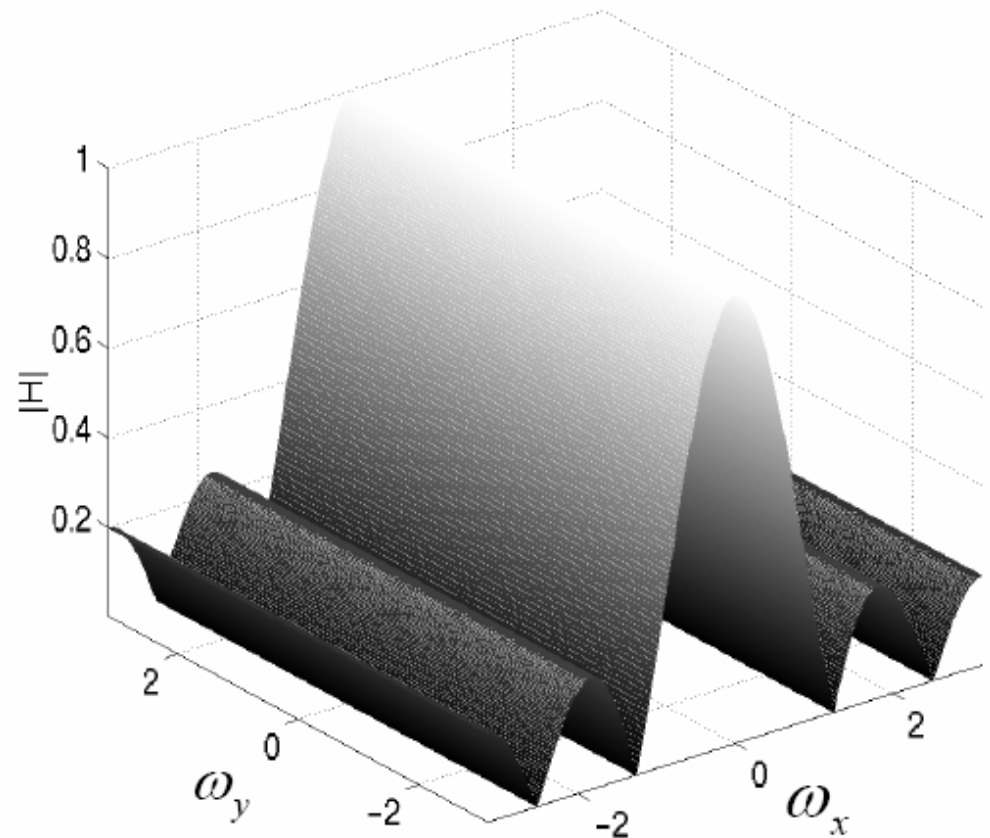
Fourier interpretation



Cameraman blurred horizontally

Filter impulse response

$$\frac{1}{5} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$



Filtering examples



Original
Cameraman



Cameraman blurred vertically
Filter impulse response

$$\frac{1}{5} \begin{pmatrix} 1 \\ 1 \\ [1] \\ 1 \\ 1 \end{pmatrix}$$

Filtering examples



Original
Cameraman



Cameraman sharpened
Filter impulse response

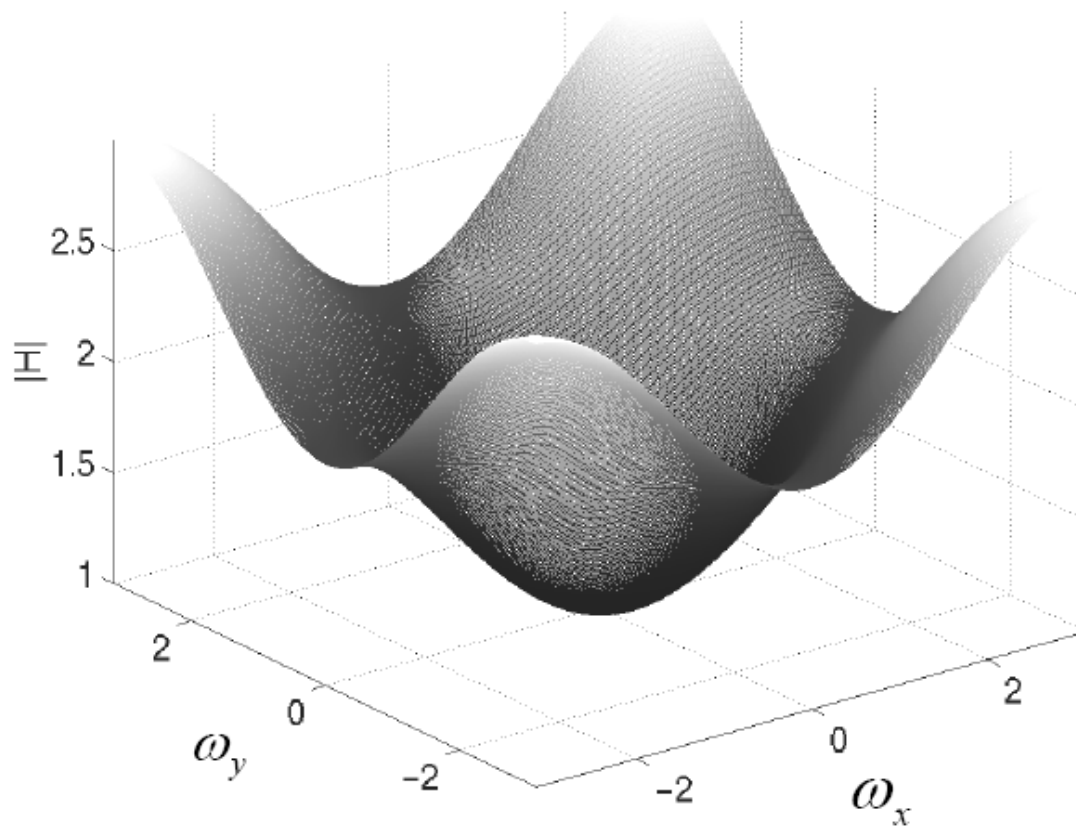
$$\frac{1}{4} \begin{pmatrix} 0 & -1 & 0 \\ -1 & [8] & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Fourier interpretation

$$H(e^{j\omega_x}, e^{j\omega_y}) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h[m, n] e^{-j\omega_x m - j\omega_y n}$$

$$= \frac{1}{4} (8 - e^{-j\omega_x} - e^{j\omega_x} - e^{-j\omega_y} - e^{j\omega_y})$$

$$= 2 - \frac{1}{2} \cos \omega_x - \frac{1}{2} \cos \omega_y$$



Filtering examples



Original
Cameraman



Cameraman sharpened
Filter impulse response

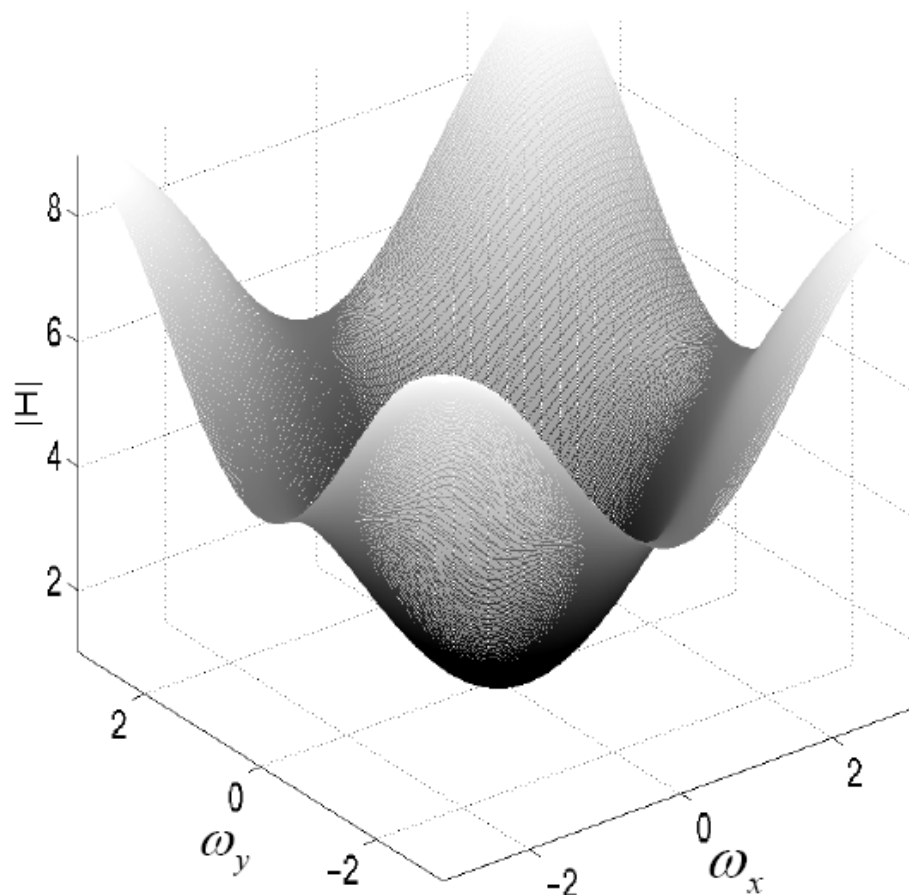
$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & [5] & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Fourier interpretation



Cameraman sharpened
Filter impulse response

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & [5] & -1 \\ 0 & -1 & 0 \end{pmatrix}$$



Linear and non linear operations



10	13	9
12	8	9
15	11	6

Replace center pixel by:

Median Filter: (6, 8, 9, 9, **10**, 11, 12, 13, 15) = 10

Minimum = 6; Maximum: 15

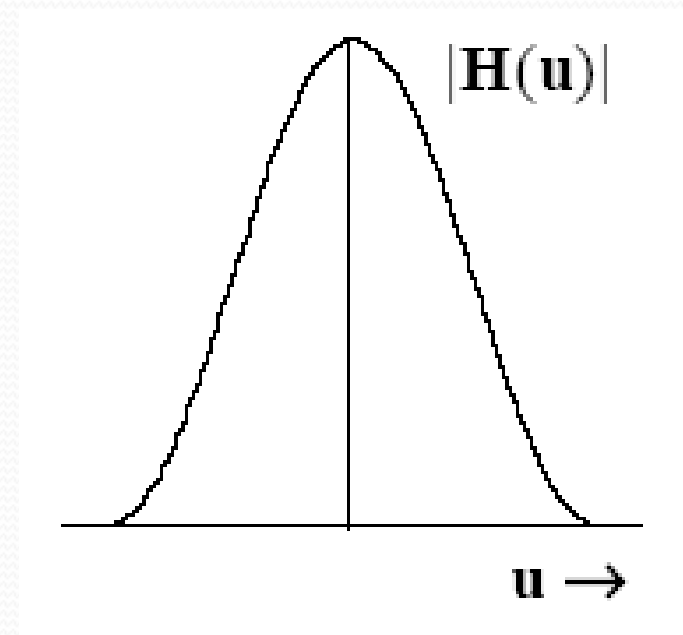
Average of nearest neighbours:

$$(10+13+9+12+8+9+15+11+6)/9 = 10.33 \rightarrow \mathbf{10}$$

Low pass gaussian filter

$$\frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} = \frac{1}{4} (1 \quad 2 \quad 1) * \frac{1}{4} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

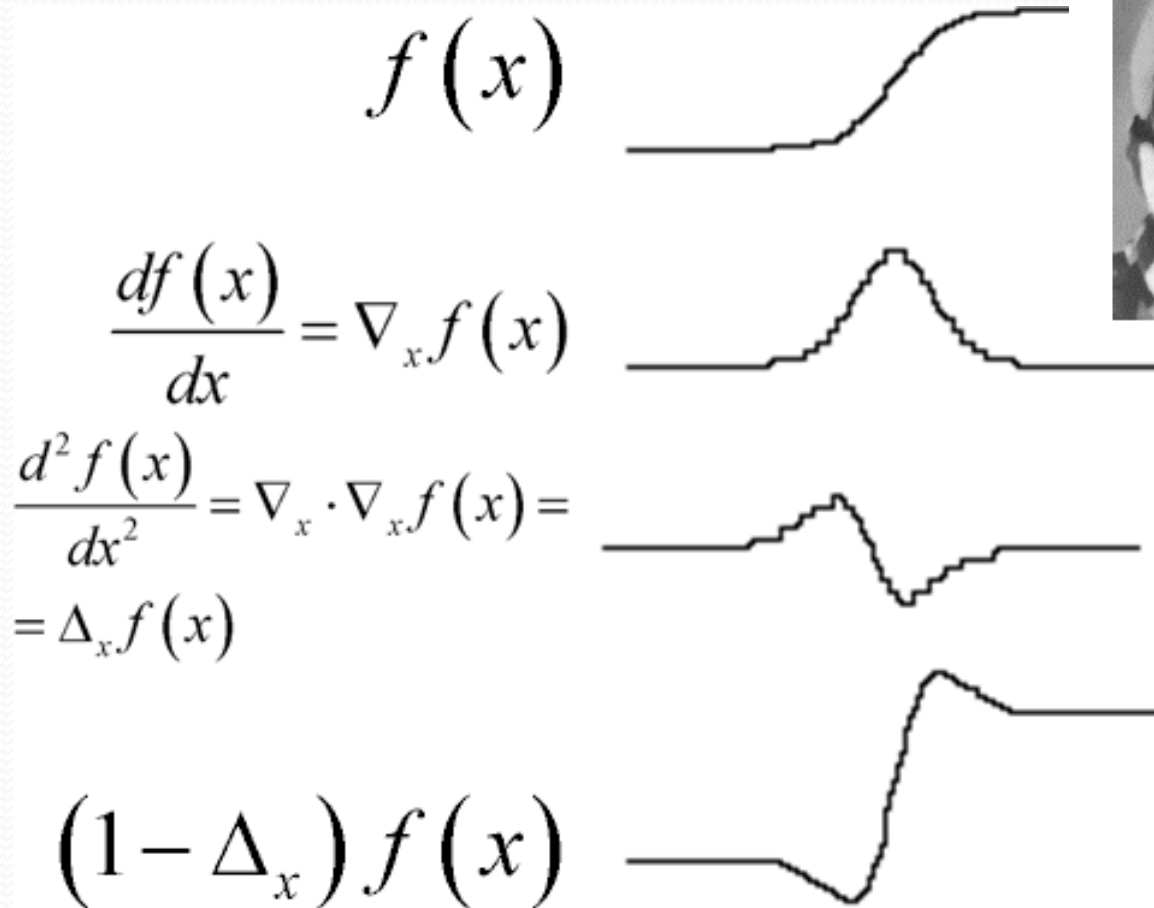
$$\frac{1}{441} \begin{pmatrix} 4 & 10 & 14 & 10 & 4 \\ 10 & 25 & 35 & 25 & 10 \\ 14 & 35 & 49 & 35 & 14 \\ 10 & 25 & 35 & 25 & 10 \\ 4 & 10 & 14 & 10 & 4 \end{pmatrix}$$



Median filtering



Hi pass filtering for high frequencies



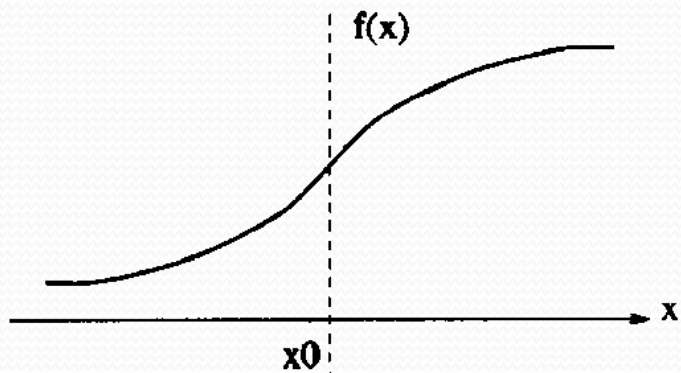
- **2D case:**

$$\begin{bmatrix} 0 & +1 & 0 \\ +1 & -4 & +1 \\ 0 & +1 & 0 \end{bmatrix}$$

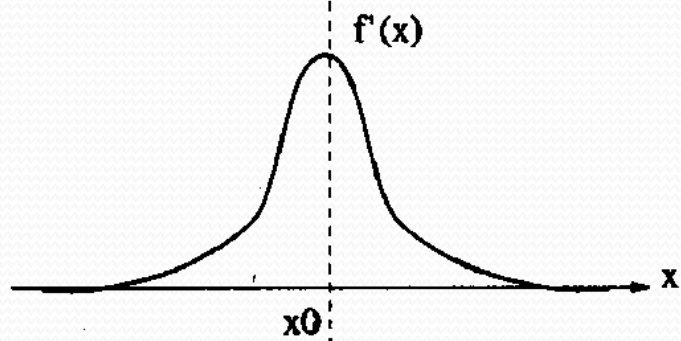
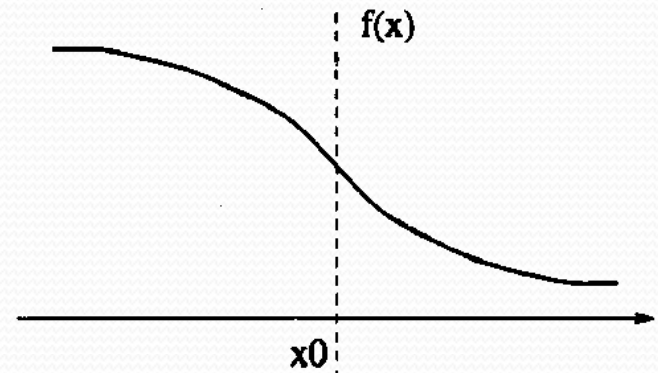
Example



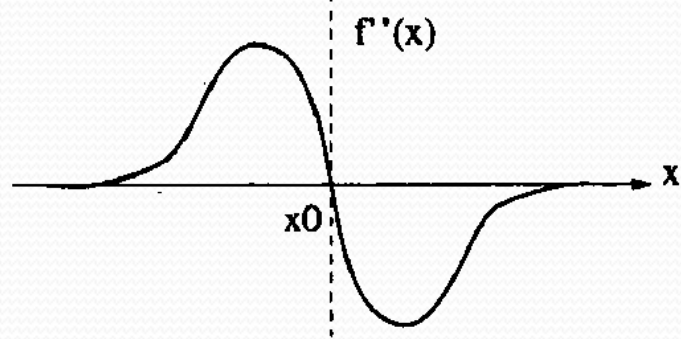
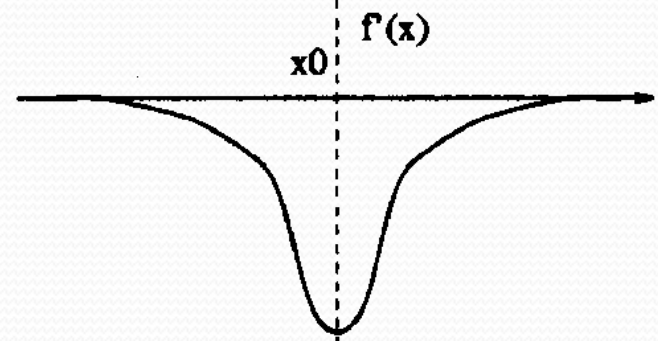
1D derivatives



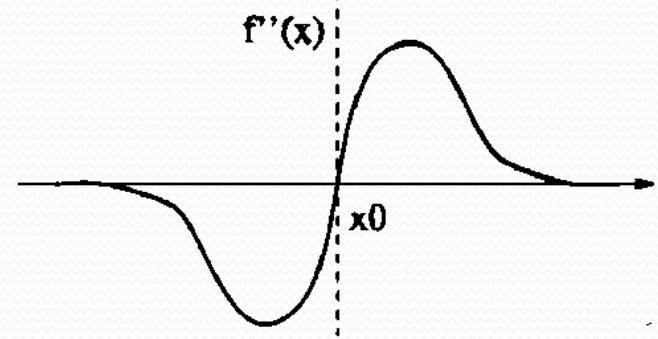
(a)



(b)

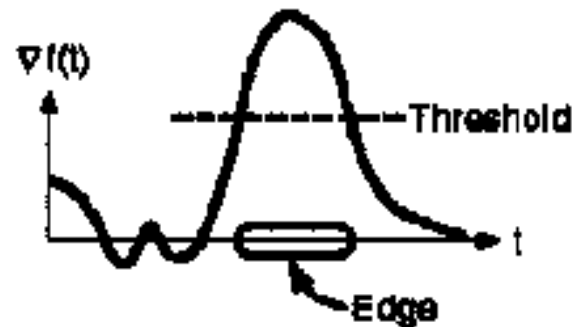
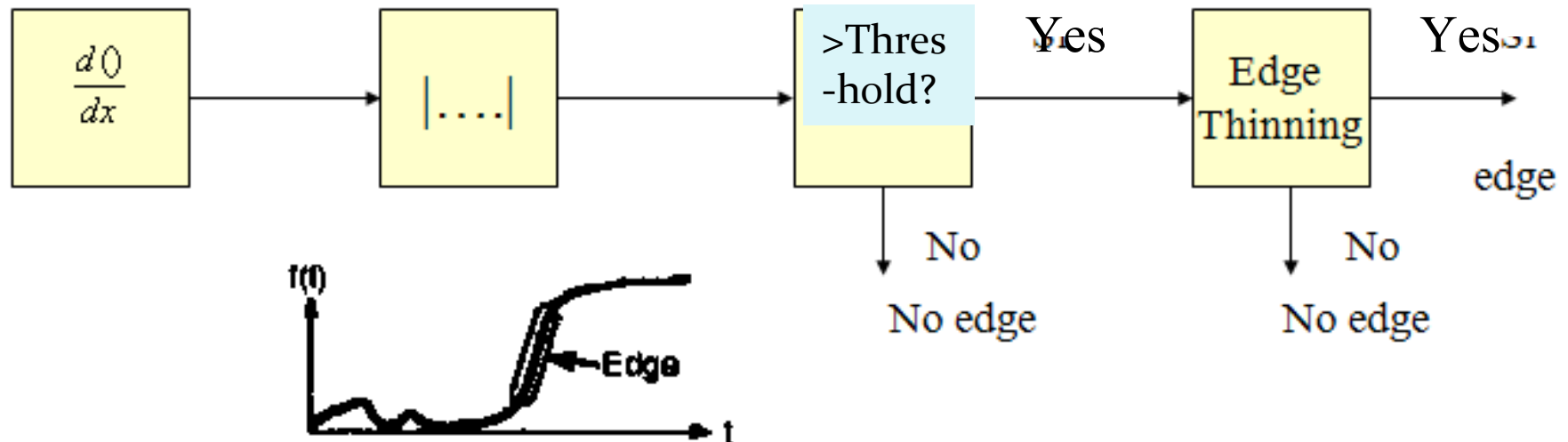


(c)



Gradient method

- 1D example



2D case

- The first derivative is substituted by the gradient

$$\nabla f(x, y) = \frac{\partial f(x, y)}{\partial x} \hat{i}_x + \frac{\partial f(x, y)}{\partial y} \hat{i}_y$$

- Omnidirectional detector

- Based on $|\nabla f(x, y)|$: isotropic behaviour

- Directional detector

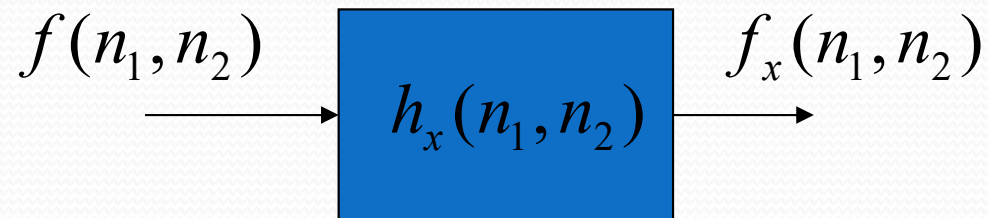
- Based on an oriented derivative:
 - ex.: a possible horizontal edge detector is $|\partial f / \partial y|$

Finite Impulse Response Model

Discrete operators for derivative estimation can be estimated as FIR filters.

$$f_y(n_1, n_2) = f(n_1, n_2) * h_y(n_1, n_2)$$

$$f_x(n_1, n_2) = f(n_1, n_2) * h_x(n_1, n_2)$$



Discrete differential operators

- Pixel difference: luminance difference between to neighbour pixels along orthogonal directions.

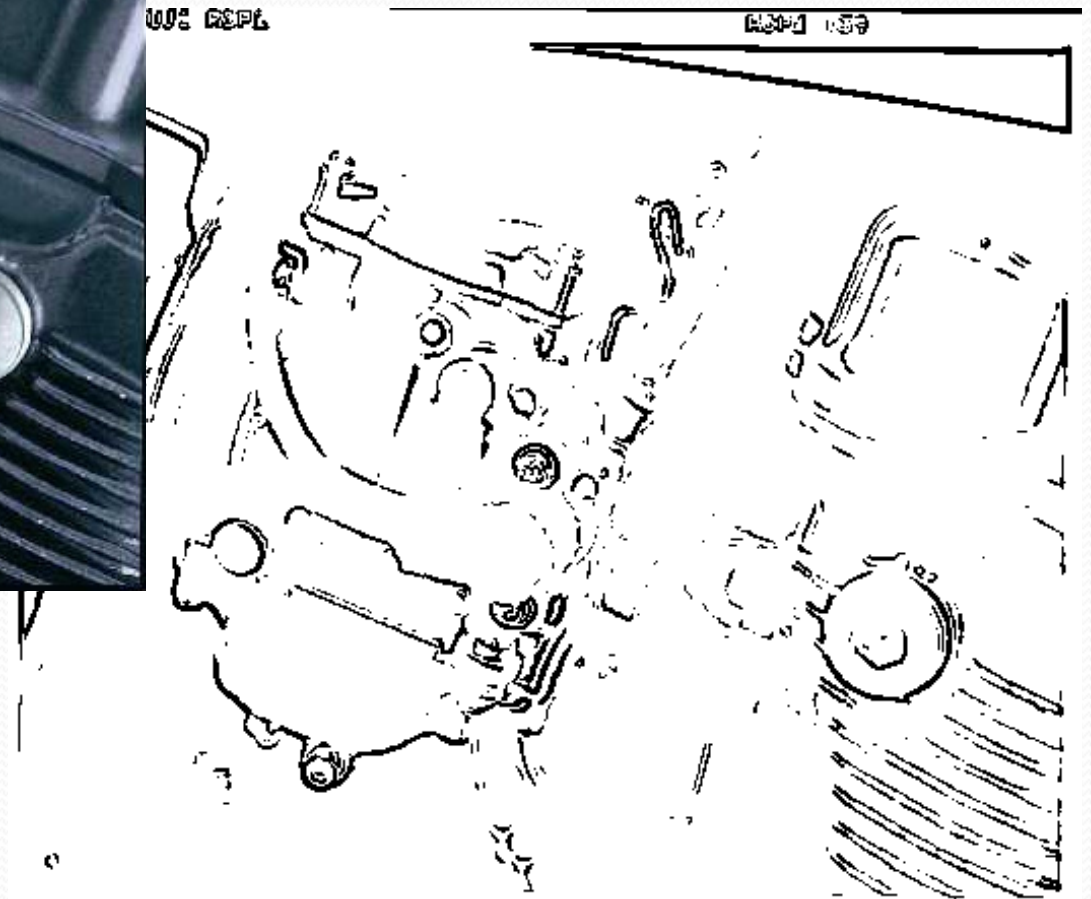
$$f_x(j, k) = f(j, k) - f(j, k - 1)$$

$$f_y(j, k) = f(j, k) - f(j + 1, k)$$

- Separable filters

$$h_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad h_y = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Example: pixel difference



Separated pixel difference

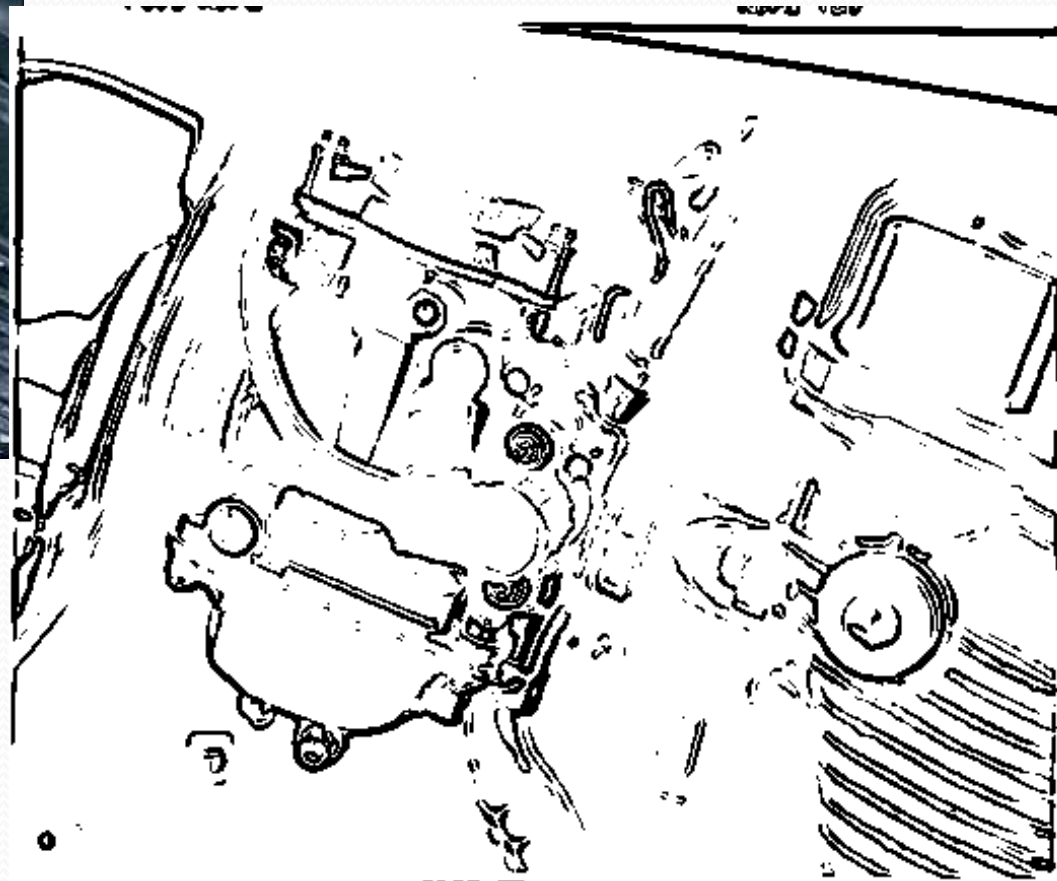
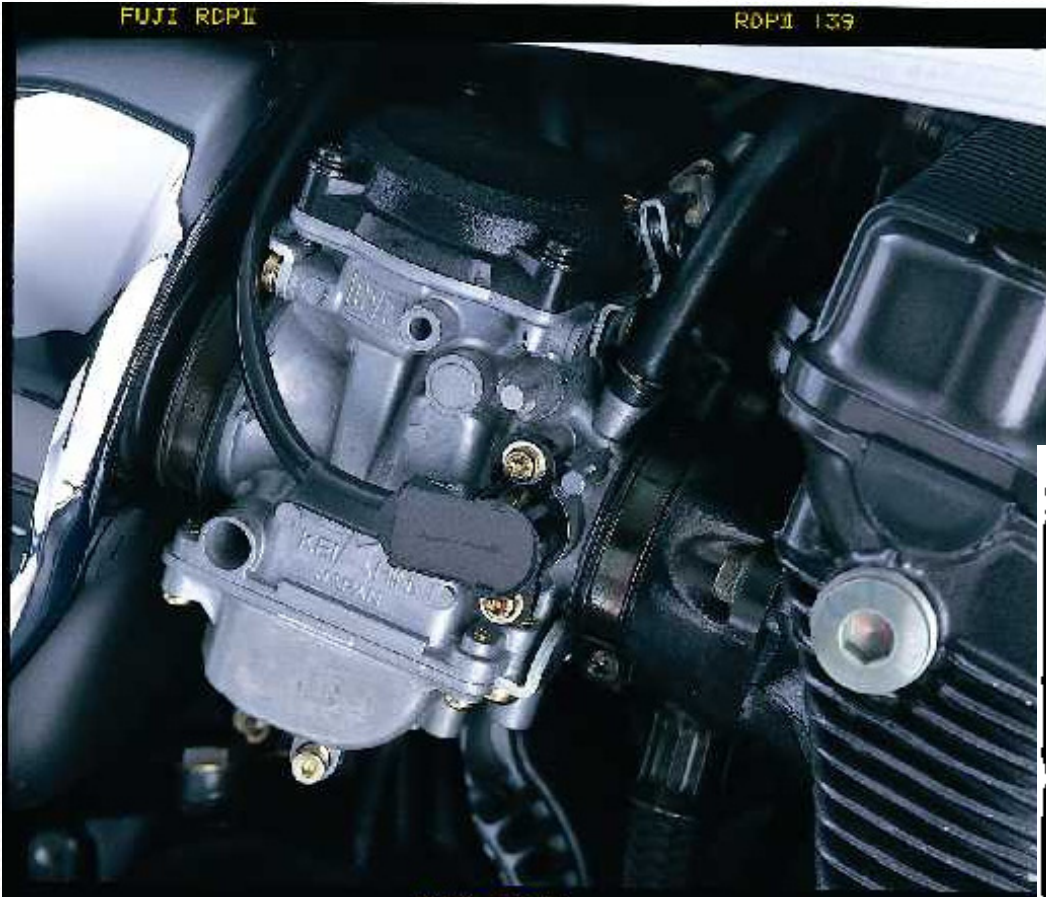
- If farther pixels are chosen there is a higher noise rejection, and there is no phase translation in edge definition.

$$f_x(j, k) = f(j, k + 1) - f(j, k - 1)$$

$$f_y(j, k) = f(j - 1, k) - f(j + 1, k)$$

$$h_x = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad h_y = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Ex.: separated pixel difference



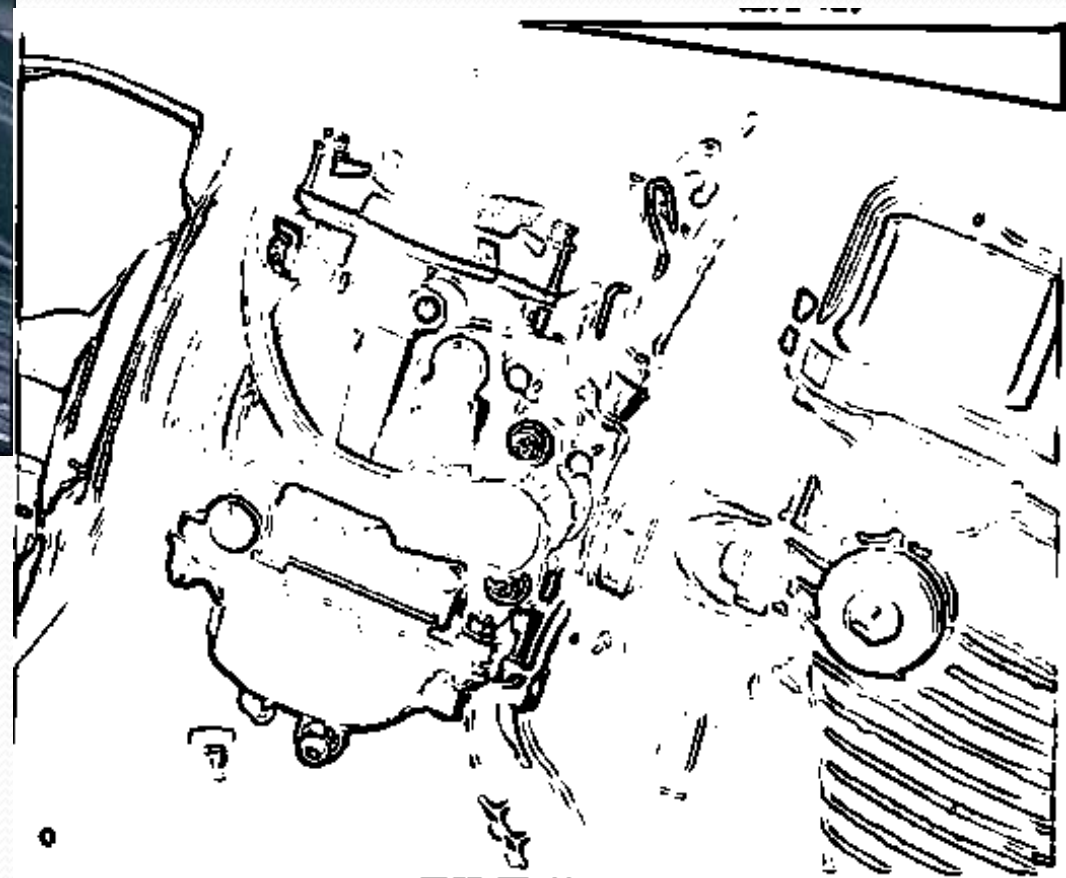
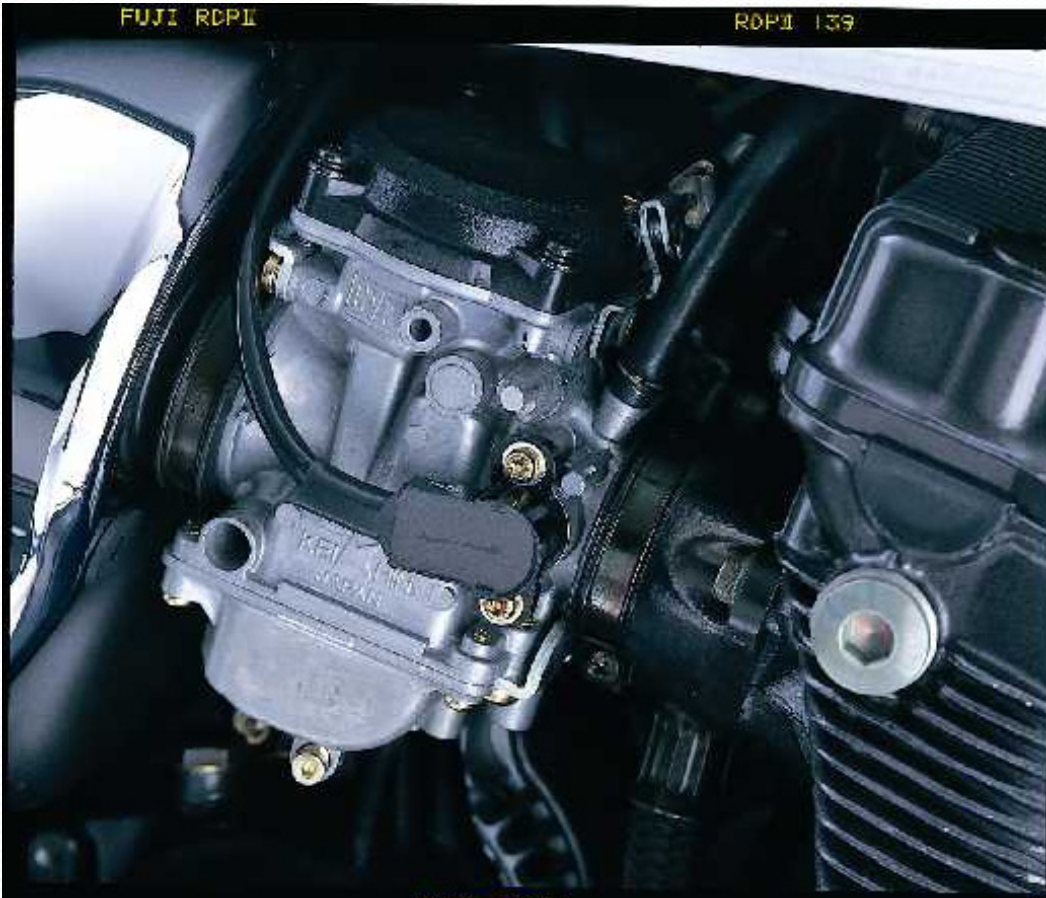
Roberts edge extraction

$$f_x(j, k) = f(j, k) - f(j + 1, k + 1)$$

$$f_y(j, k) = f(j, k + 1) - f(j + 1, k)$$

$$h_x = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad h_y = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Ex.: Roberts method



Prewitt method

Estimation can be improved involving more samples for the gradient operator 3x3

$$K * \left(\frac{\partial f}{\partial x} \right) \cong [f(n_1 + 1, n_2 + 1) - f(n_1 - 1, n_2 + 1)] + \\ + [f(n_1 + 1, n_2) - f(n_1 - 1, n_2)] + [f(n_1 + 1, n_2 - 1) - f(n_1 - 1, n_2 - 1)]$$

= vertical low pass* horizontal high pass

$$K * \left(\frac{\partial f}{\partial y} \right) \cong [f(n_1 + 1, n_2 + 1) - f(n_1 + 1, n_2 - 1)] + \\ + [f(n_1, n_2 + 1) - f(n_1, n_2 - 1)] + [f(n_1 - 1, n_2 + 1) - f(n_1 - 1, n_2 - 1)]$$

= vertical high pass* horizontal low pass

Gradient estimation

- Gradient modulus = the value of the higher directional derivative
- Gradient phase = orientation of the higher directional derivative

Squared lattice = eight possible directions

$$h(n_1, n_2) = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

→ E

$$h(n_1, n_2) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

↑ N

$$h(n_1, n_2) = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

↗ NE

$$h(n_1, n_2) = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$$

↖ NW

Gradient estimation

$$h(n_1, n_2) = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

← W

$$h(n_1, n_2) = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

↙ SW

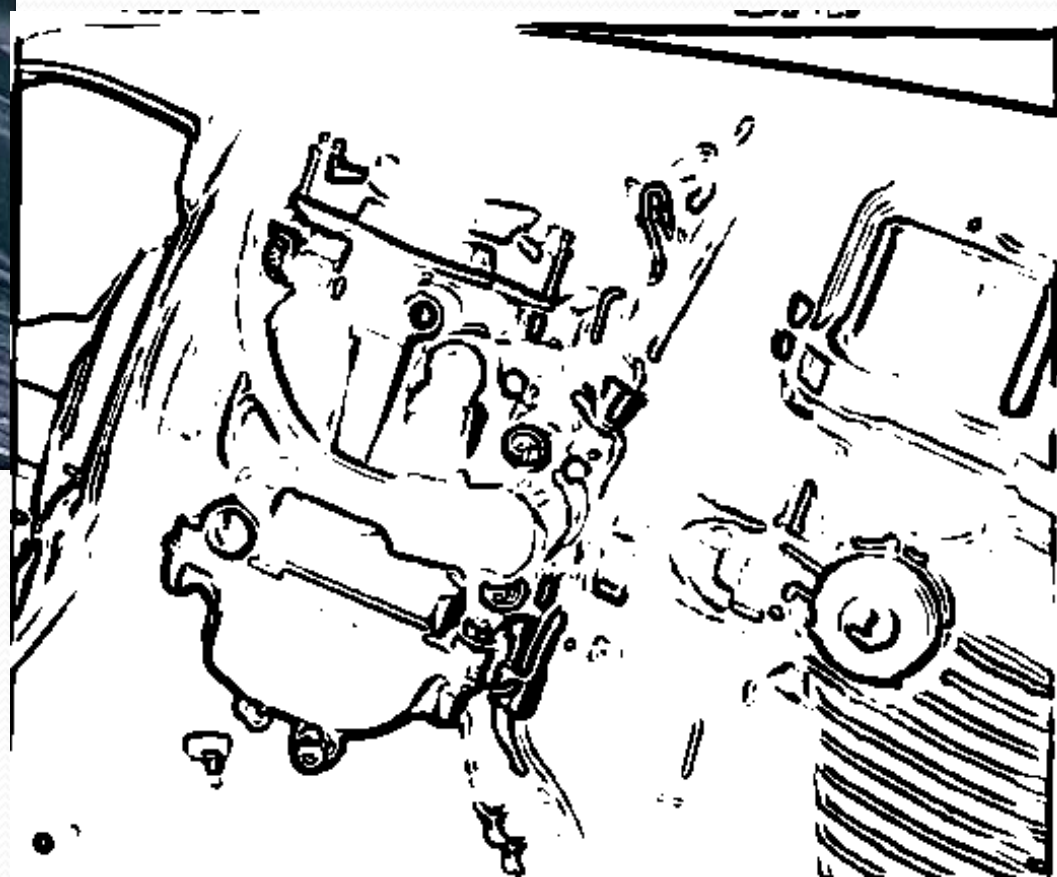
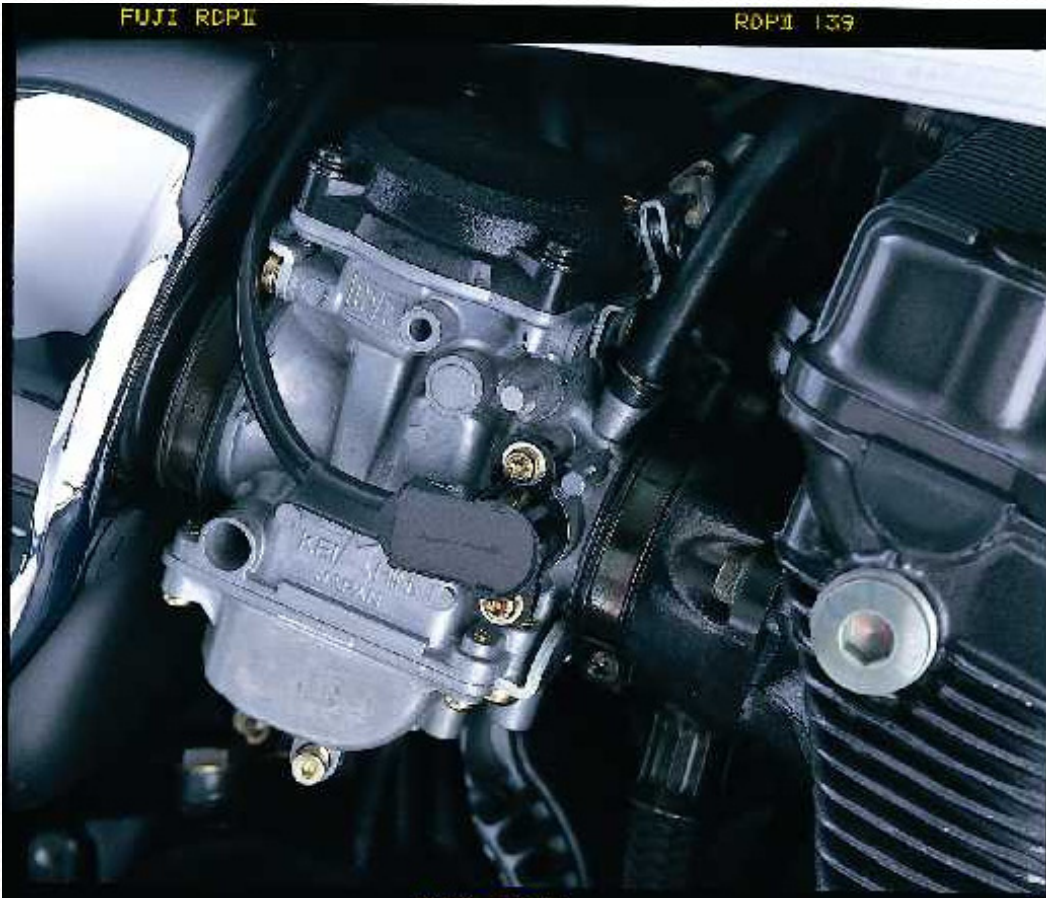
$$h(n_1, n_2) = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

↓ S

$$h(n_1, n_2) = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

↘ SE

Ex.: Prewitt method 3x3



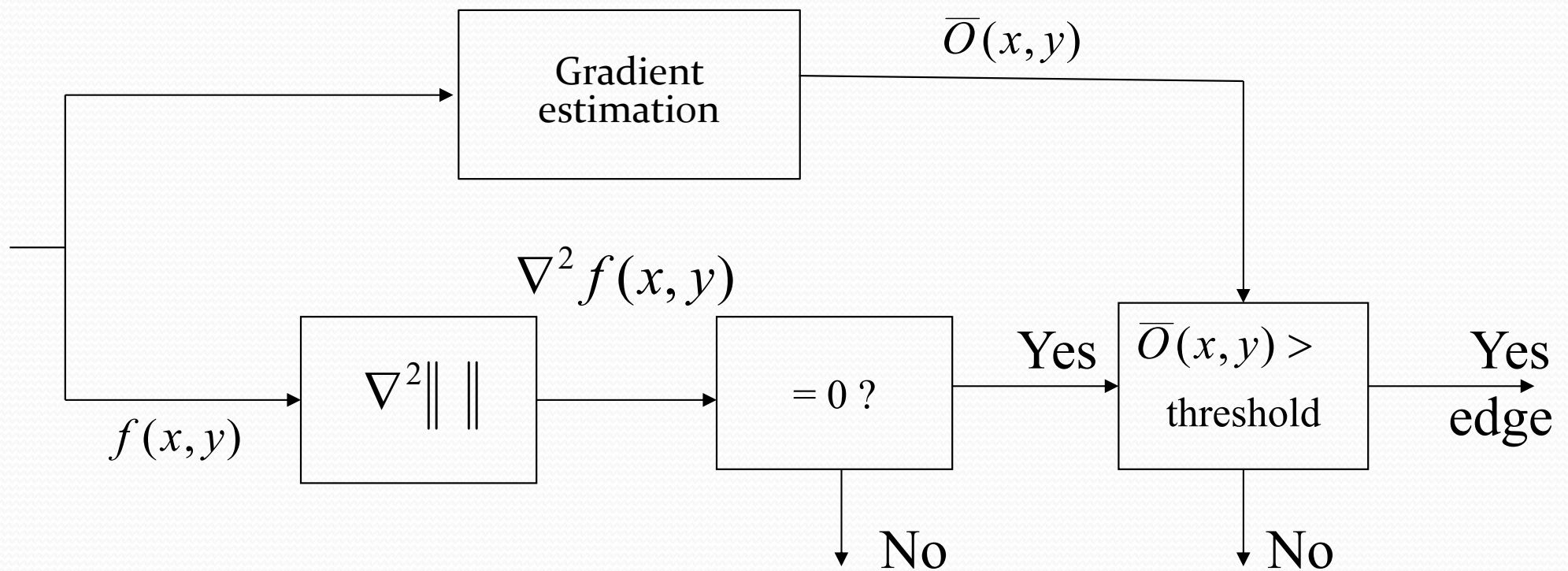
Laplacian

- For the 2D case the 2nd order differential operator is the Laplacian

$$\nabla^2 f(x, y) = \nabla \cdot (\nabla f(x, y)) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$$

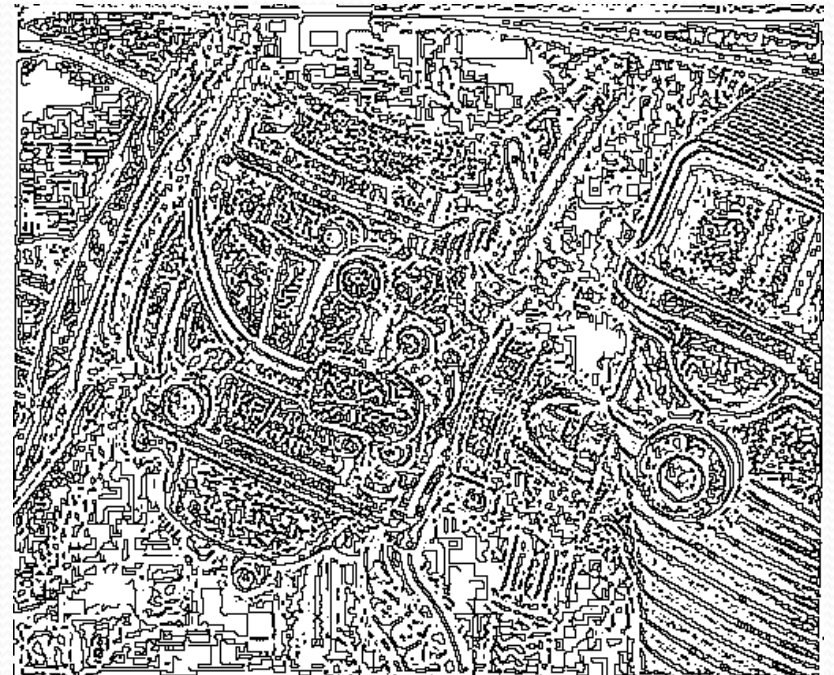
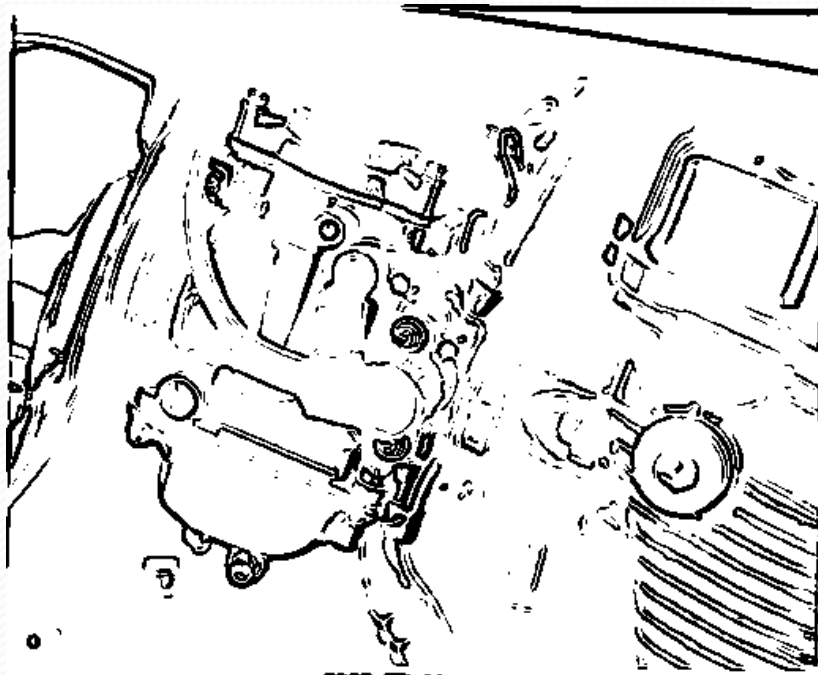
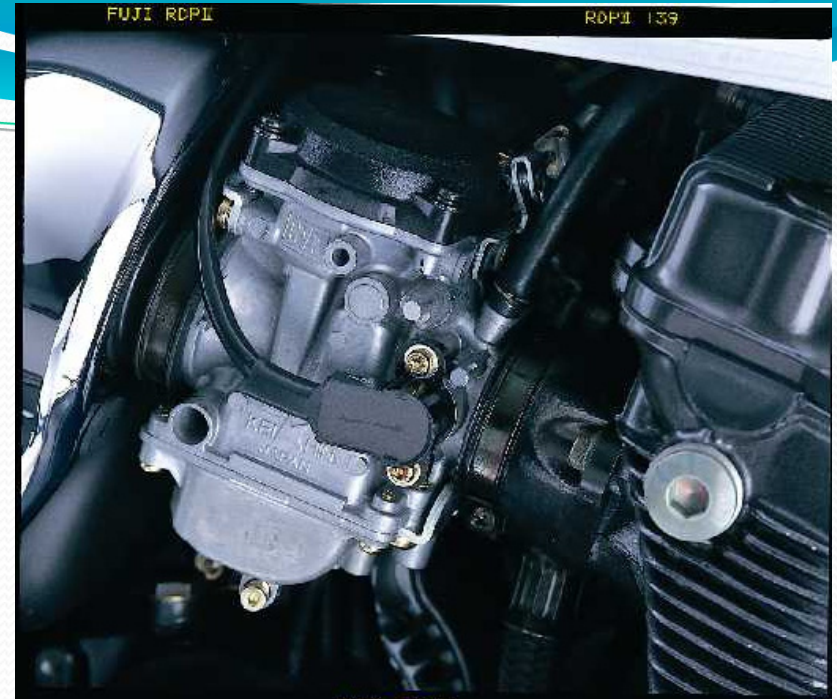
- Isotropic operator
- More sensible to noise with respect to gradient
 - False edges can be generated due to noise.
- Thinner edges are produced.

Gradient + Laplacian



Zero-crossing without threshold

Just Laplacian



Laplacian Discretization

$$\nabla^2 f(i, j) = f(i+1, j) + f(i-1, j) + f(i, j+1) + f(i, j-1) - 4f(i, j)$$

Can be seen as the convolution of $f(n_1, n_2)$ with the impulse response $h(n_1, n_2)$ of a linear system.

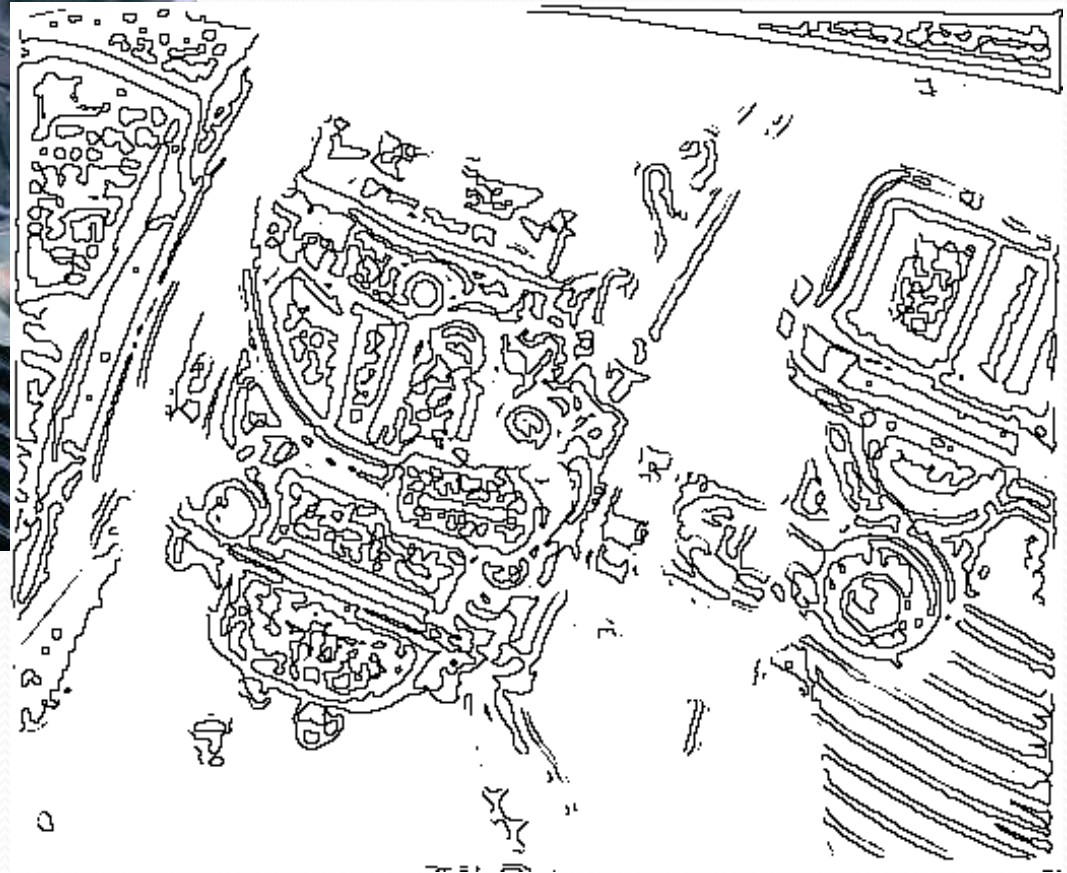
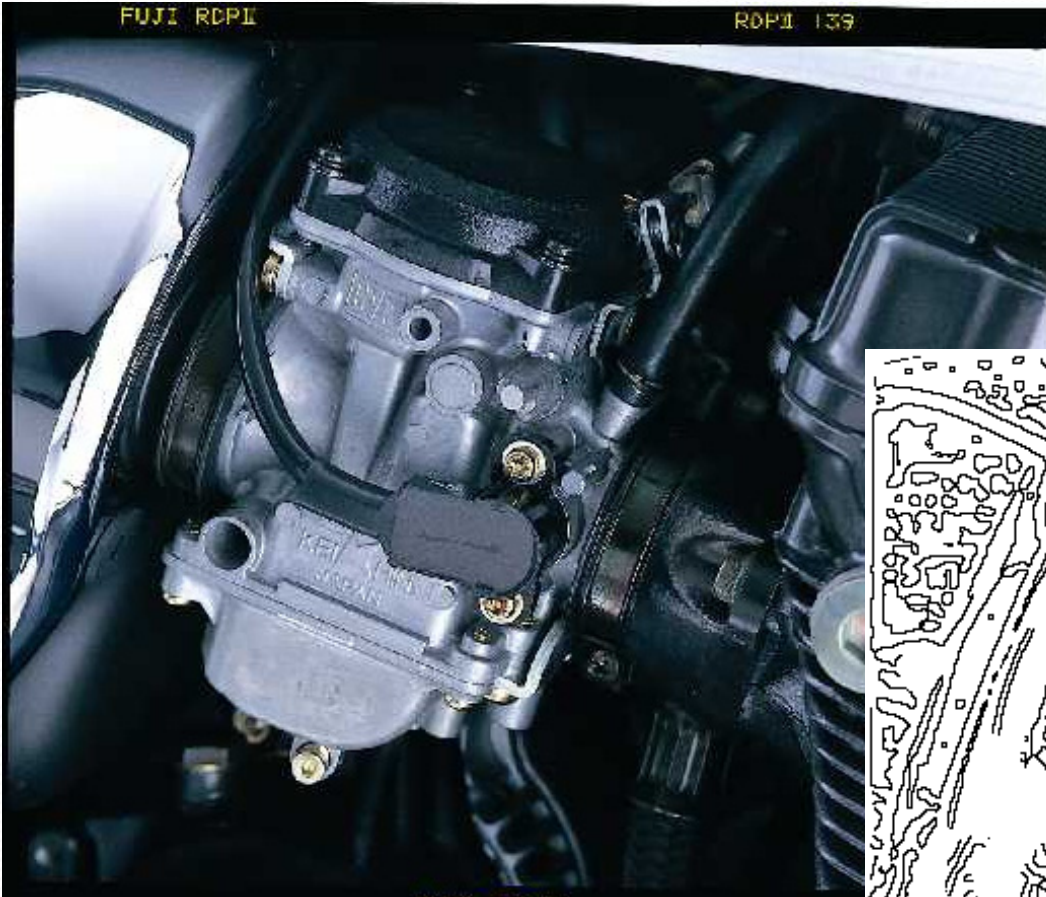
$$\nabla^2 f(n_1, n_2) = f(n_1, n_2) * h(n_1, n_2)$$

4 neighbours method

- Separable normalized filter
 - Unit gain for continuous component
 - The sign of $\mathbf{h}(\mathbf{n}_1, \mathbf{n}_2)$ can be changed without changes in the final result (since we are looking for zeros of laplacian)

$$\begin{aligned} h(n_1, n_2) &= \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} + \frac{1}{4} \begin{bmatrix} 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix} \\ &= \frac{1}{4} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \end{aligned}$$

Ex.: 4 neighbours method



Laplacian Discretization

- The laplacian can be approximated with finite differences

$$\frac{\partial f(x, y)}{\partial x} \Rightarrow f_x(x, y) = f(j+1, k) - f(j, k)$$

$$\begin{aligned} \frac{\partial^2 f(x, y)}{\partial x^2} &\Rightarrow f_{xx}(j, k) = f_x(j, k) - f_x(j-1, k) = \\ &= f(j+1, k) - 2f(j, k) + f(j-1, k) \end{aligned}$$

Discretization examples

Prewitt method

- Not separable filter

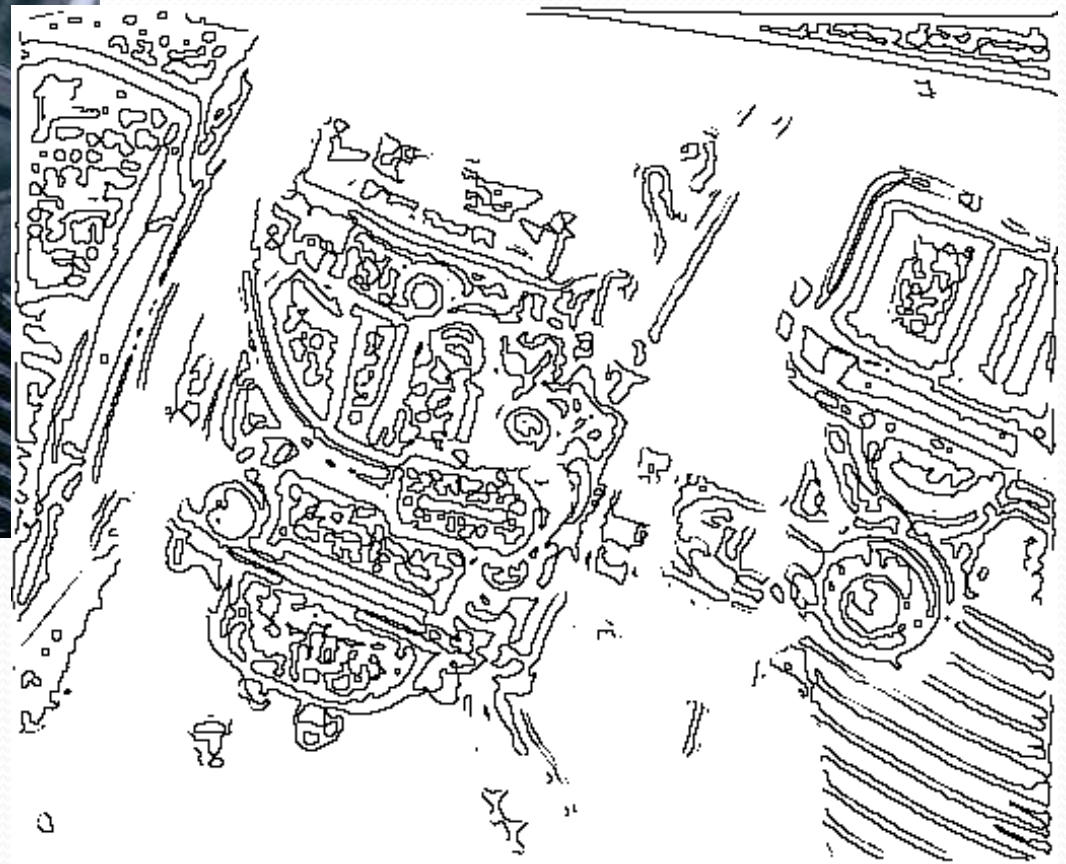
$$h(n_1, n_2) = \frac{1}{8} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

8 Neighbours method

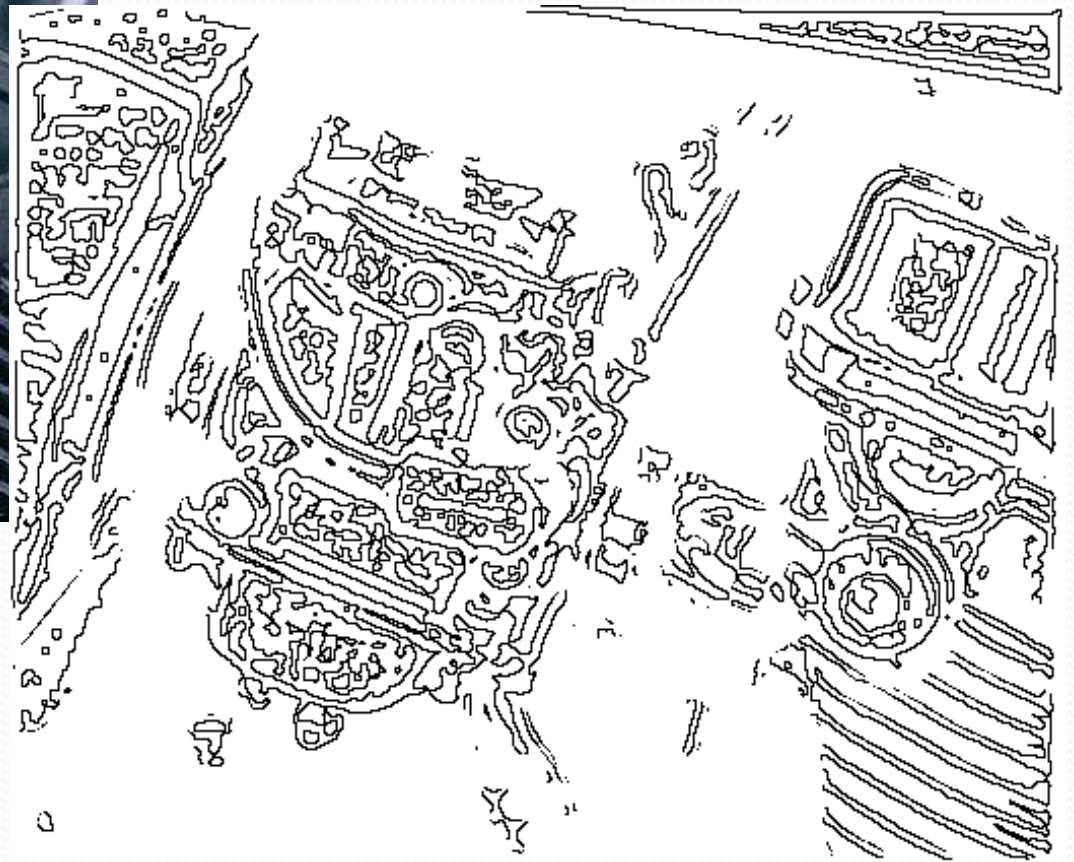
- Similar to Prewitt but with a separable formulation

$$h(n_1, n_2) = \frac{1}{8} \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} + \frac{1}{8} \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

Ex.: 8 neighbours method



Ex.: Prewitt not separable



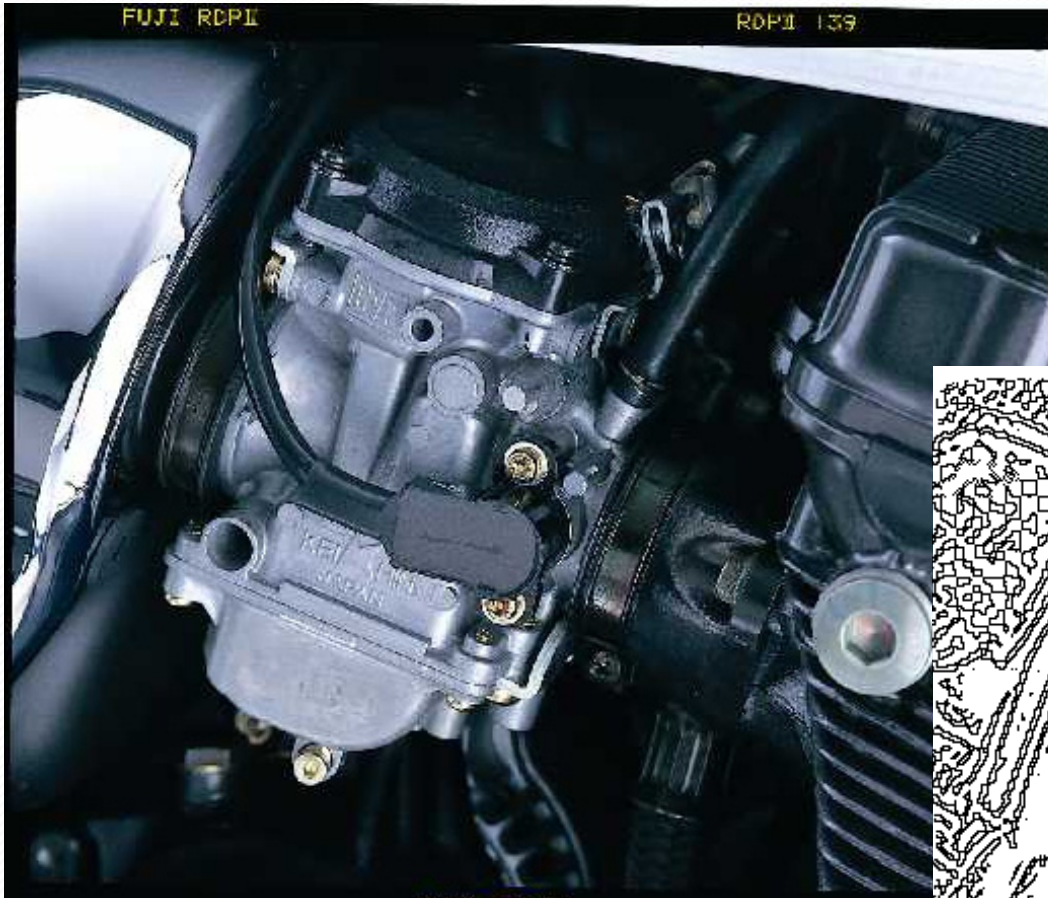
Noise presence

- When noise is significant these filters could not be accurate for diagonal edges. The Prewitt filter can work even in regions with high density of edges.

$$h(n_1, n_2) = \frac{1}{8} \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

- Since edges are directional and noise can generate luminance variations, zero-crossing for laplacian could find non-correct edges.

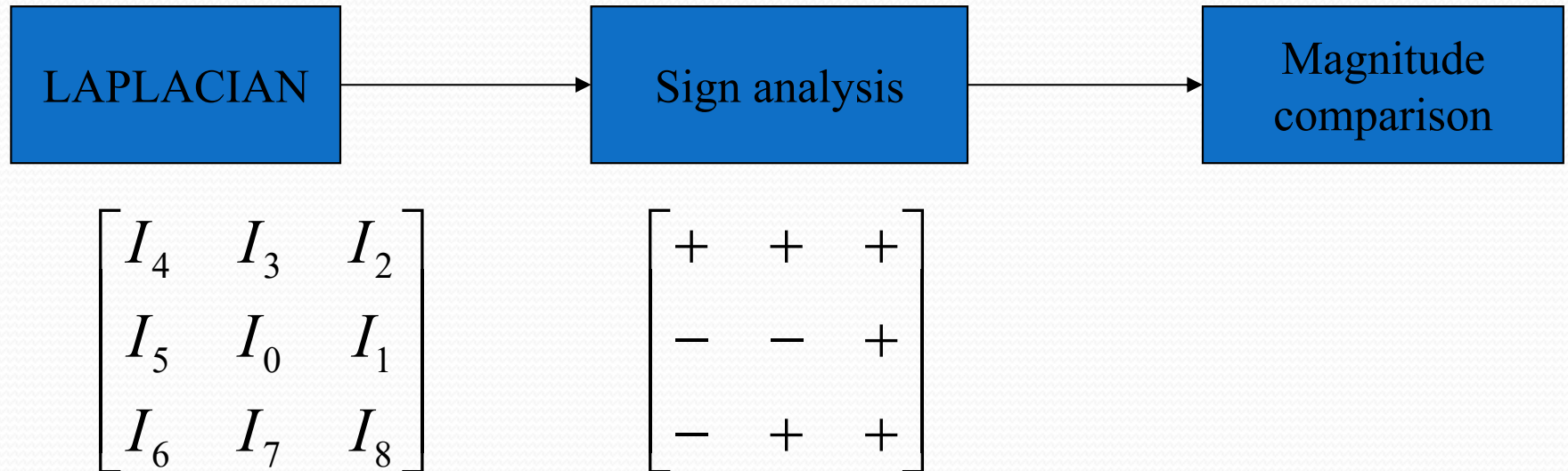
Ex.: Laplacian for diagonal edges



Super-resolution (Laplacian)

First method.

- Given two neighbour pixels, mark as possible edge point the intrapixels points if the laplacian values in the two pixels have different signs.
- Assume as effective edge the point, among them, with the largest gradient.
- Apply this analysis to all the pixels couples.



superresolution

Second method: analytical approach

- Approximate the continuous form of function $f(n_1, n_2)$ with a 2D polynomial in order to describe the laplacian in an analytical way.
- Polynomial example:

$$\hat{F}(r, c) = K_1 + K_2 r + K_3 c + K_4 r^2 + K_5 r c + K_6 c^2 + K_7 r^2 c + K_8 r c^2 + K_9 r^2 c^2$$

- where K_i are the weights obtained from the discrete image.

$$\frac{-(W-1)}{2} \leq r, c \leq \frac{(W-1)}{2}$$

- r and c then become continuous variables associated to a discrete image matrix.
- Polynomial formulation can be found with small efforts.

Gaussian filtering

$$s(x, y) = f(x, y) * h(x, y)$$

Why we should use a gaussian function?

- Since the Fourier transform of a Gaussian is still a gaussian,
- The cut-off frequency can be expressed as a function of the width of the impulse response
- It has a low aliasing
- The filter is separable and isotropic at the same time

$$h(x, y) = h_{\sigma}(x)h_{\sigma}(y)$$

$$h_{\sigma}(x) = \frac{e^{-x^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}}$$

The noise sensitivity (numerous zero crossing) decrease as the filter strength (width) increase.

“LOG” operator

- The low pass gaussian filter has a variable cut-off frequency.

$$h(x, y) = \exp\left[-\frac{x^2 + y^2}{2\pi\sigma^2}\right]$$

$$H(\omega_x, \omega_y) = 2\pi^2\sigma^2 \exp\left[-\frac{\pi\sigma^2(\omega_x^2 + \omega_y^2)}{2}\right]$$

- It follows that the standard deviation σ is inversely proportional to the filter width.

“LOG” operator

$$g(x, y) = \nabla^2 (f(x, y) * h(x, y))$$

- Laplacian and filtering are interchangeable since both of them are linear

$$g(x, y) = f(x, y) * [\nabla^2 h(x, y)]$$

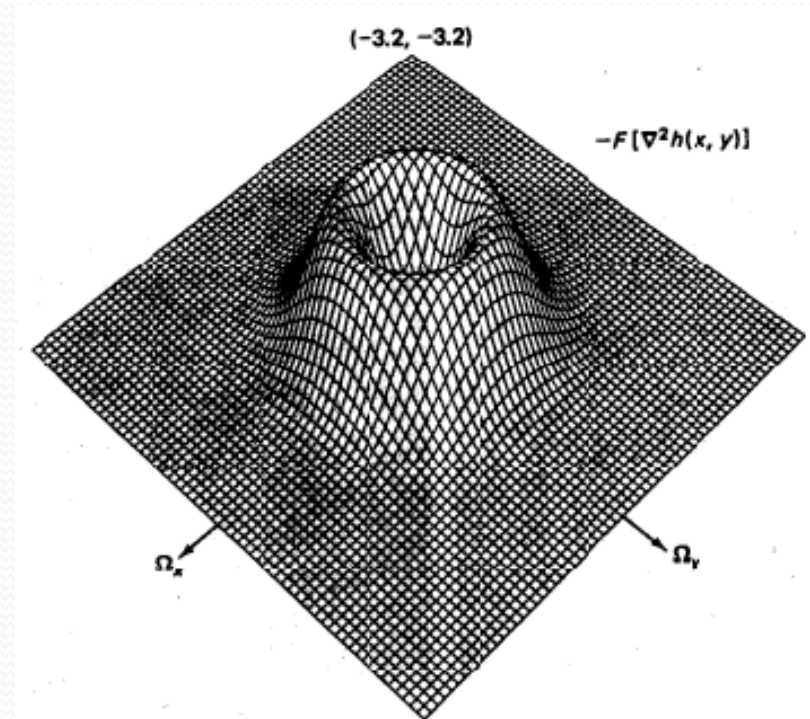
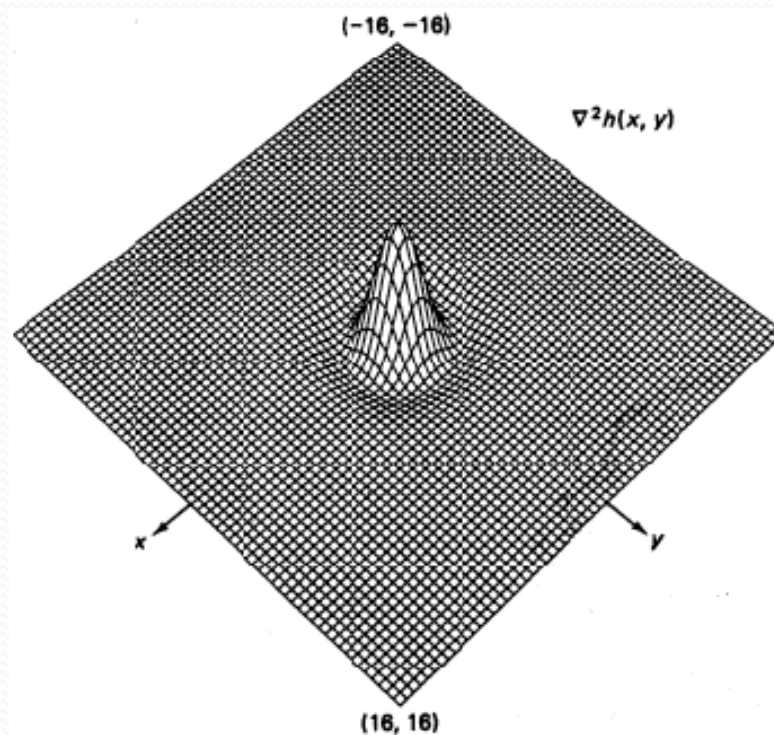
$$\nabla^2 h(x, y) = \frac{x^2 + y^2 - 2\pi\sigma^2}{\pi^2\sigma^4} \exp\left[-\frac{x^2 + y^2}{2\sigma^2\pi}\right]$$

$$\mathfrak{F}[\nabla^2 h(x, y)] = 2\pi\sigma^2 \exp\left[-\pi\sigma^2 \frac{\omega_x^2 + \omega_y^2}{2}\right] (\omega_x^2 + \omega_y^2)$$

Difference of gaussians

The LOG, Laplacian of a Gaussian corresponds to the derivative of a gaussian with respect to $2\sigma^2$

The laplacian can be approximated with the difference of two gaussian filters with different σ .



DOG application

