Neural Networks models

- Feedforward: perceptrons, MLP/BP
- ANN

Neural Networks models

- Linear classifiers
- Single layers – the XOR problem
- Multi layers

Neural Networks models

Association network

Basic operation: $r_i = \sum_j m_{ij} s_j + f_i$

Update rule: $\frac{d}{dt} m_{ij} = \lambda f_i s_j$

Neural Networks models

In vector form:

$$r = Ms + f$$

$$\frac{d}{dt} M = \lambda f \; s^T$$

Neural Networks models

Initially $m_{ij} = 0$

After learning first pair $M(t) = \lambda t\, f^{1} s^{1}$

Finally $$M = \sum_{i=1}^{p} f^{(i)} s^{(i)T}$$

If the S vectors are orthogonal: association (auto/hetero)

Neural Networks models

- Perceptrons: single level feedforward
- Changing the synaptic weights:

$$w_i(t+1) = w_i(t) + \lambda r_i(t)$$

Hebbian $$r_i(t) = x_i(t)\, y(t)$$

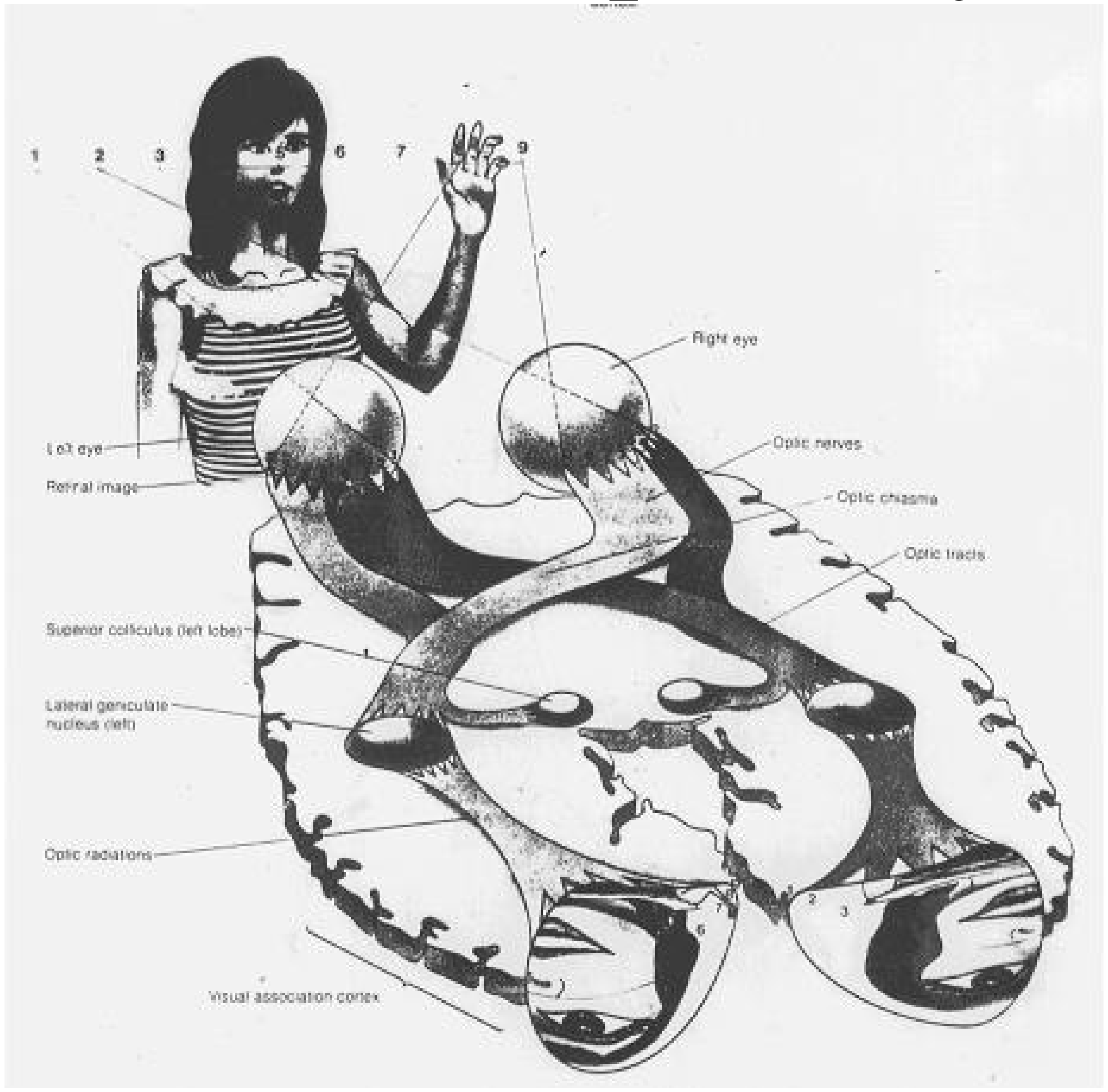
or

Reinforcement $$r_i(t) = [z(t) - \sum w_j(t) x_j(t)] x_i(t)$$

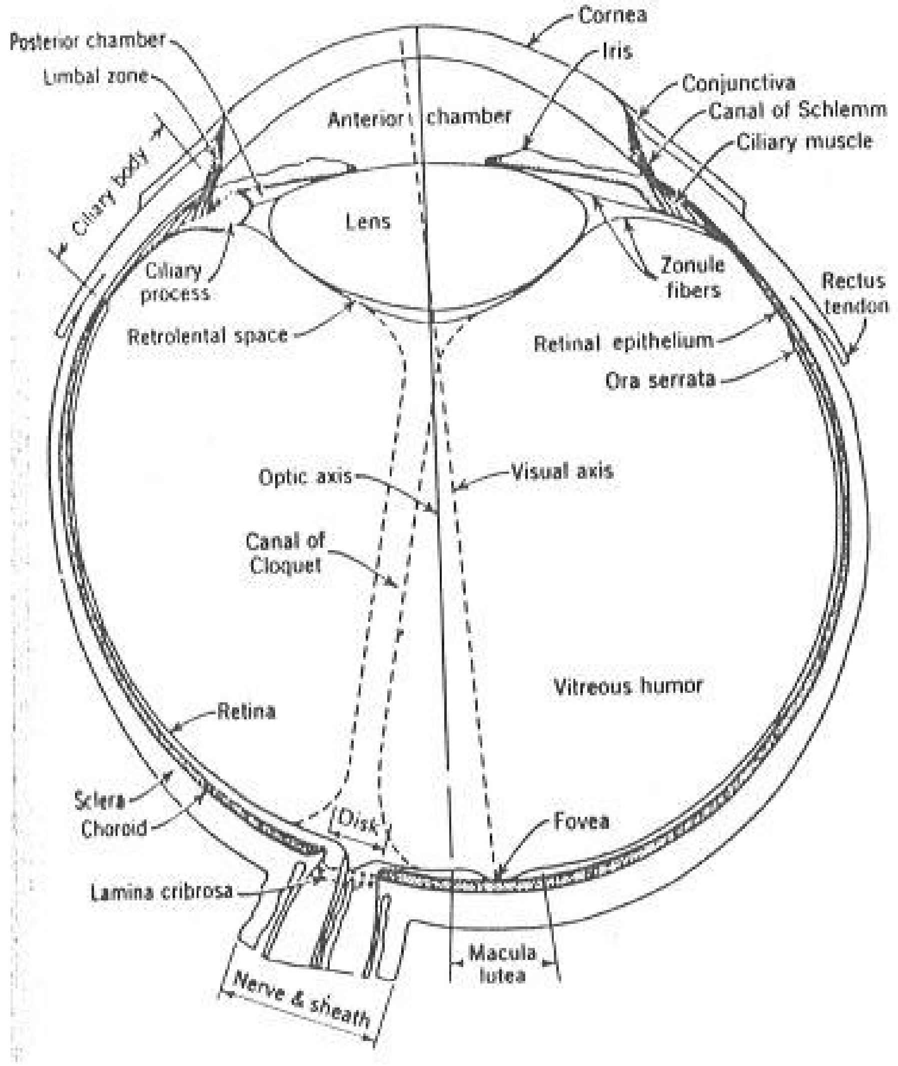
Neural Networks models

- ANN: $S(t) = F(S(t-1))$

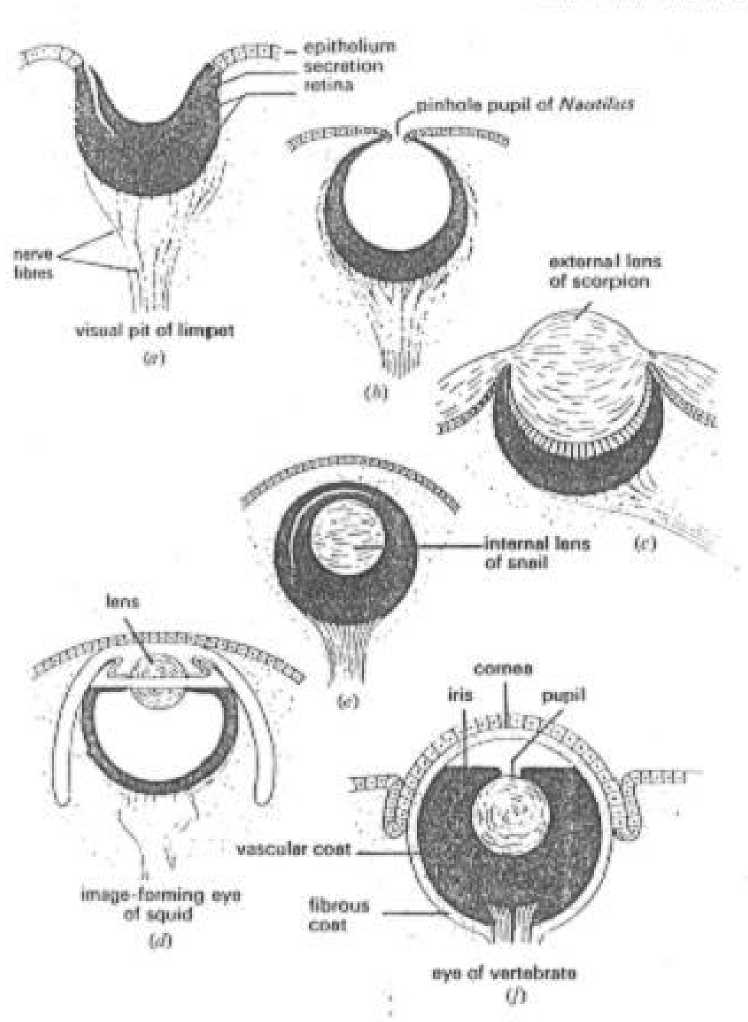
The visual pathway


1
2
3
6
7
9
Right eye
Left eye
Retinal image
Optic nerves
Optic chiasma
Optic tracts
Superior colliculus (left lobe)
Lateral geniculate nucleus (left)
Optic radiations
Visual association cortex

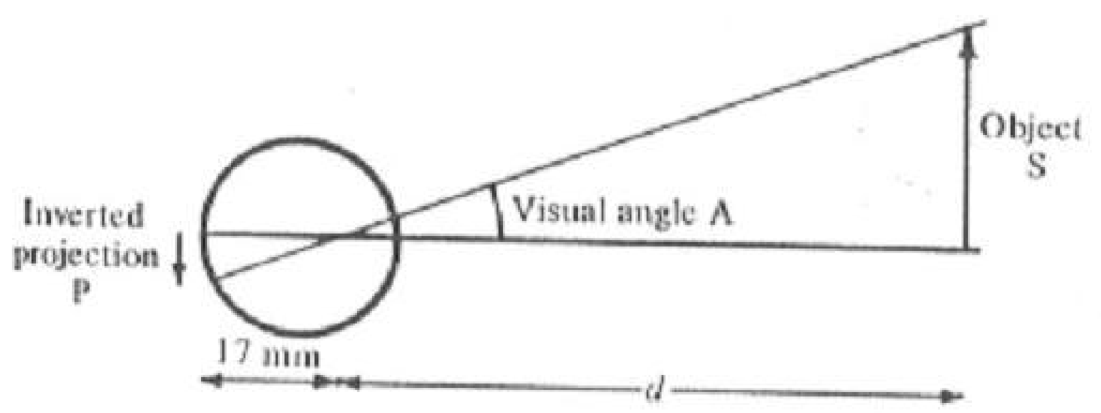
The eye


Posterior chamber
Limbal zone
Ciliary body
Cornea
Iris
Conjunctiva
Canal of Schlemm
Ciliary muscle
Anterior chamber
Lens
Ciliary process
Zonule fibers
Rectus tendon
Retrolental space
Retinal epithelium
Ora serrata
Optic axis
Visual axis
Canal of Cloquet
Vitreous humor
Retina
Sclera
Choroid
Disk
Fovea
Lamina cribrosa
Macula lutea
Nerve & sheath

Eye types


epithelium
secretion
retina
nerve fibres
visual pit of limpet
(a)
pinhole pupil of *Nautilus*
(b)
external lens of scorpion
(c)
internal lens of snail
(e)
lens
image-forming eye of squid
(d)
cornea
iris
pupil
vascular coat
fibrous coat
eye of vertebrate
(f)

Eye geometry

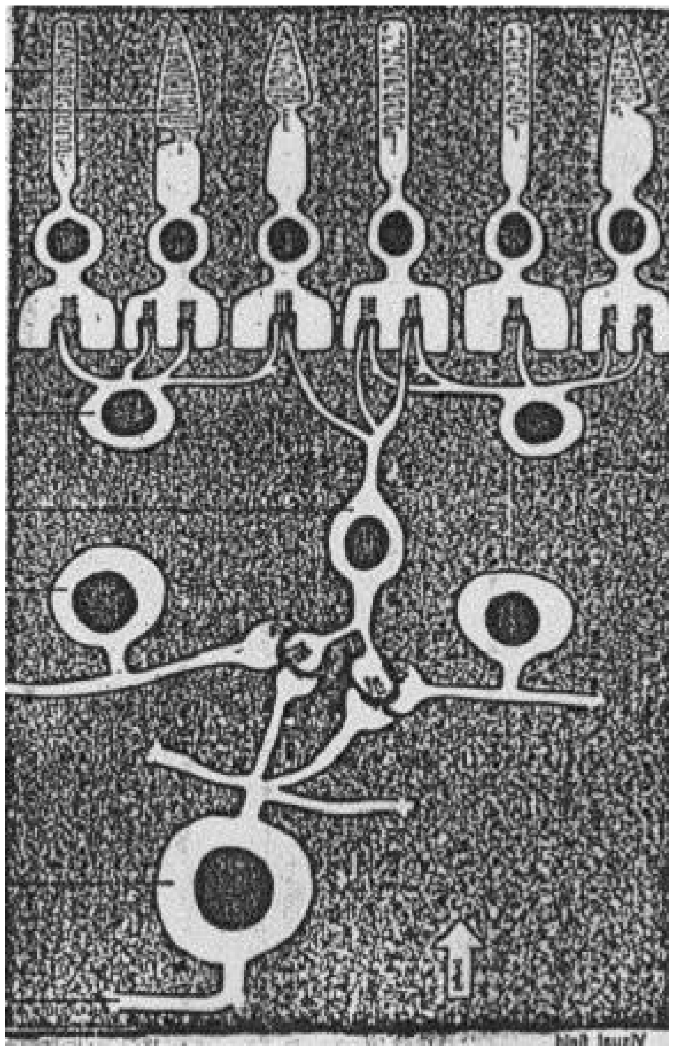


$$P = \frac{17S}{d} \qquad \text{mm}$$

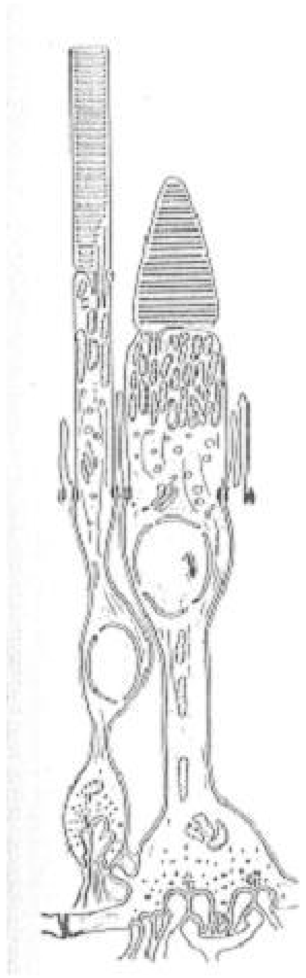
and the visual angle A is given by

$$A = \tan^{-1}\left(\frac{S}{d}\right) \qquad \text{degrees}$$

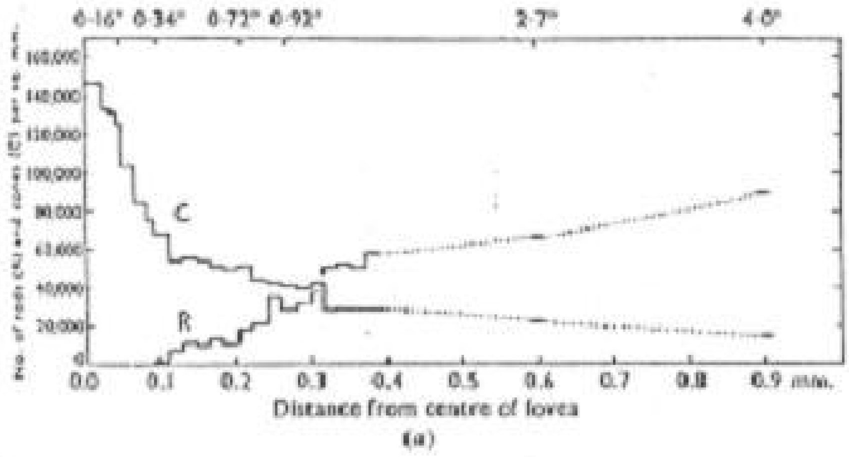
The Retina



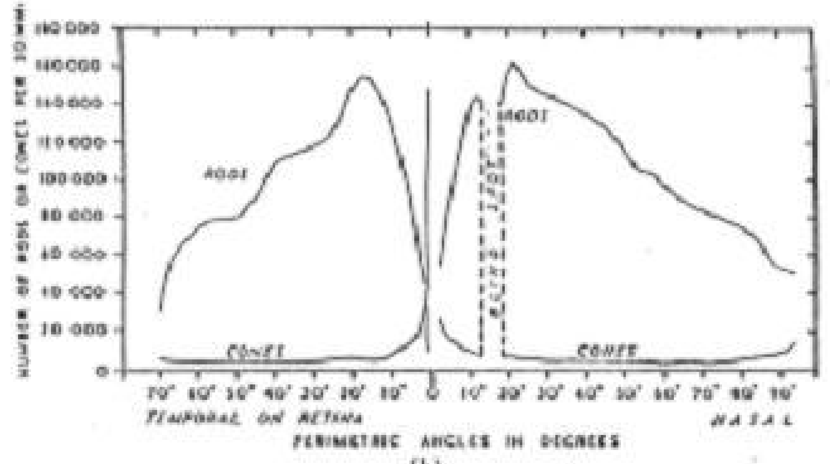
Rods and Cones



Rods and Cones

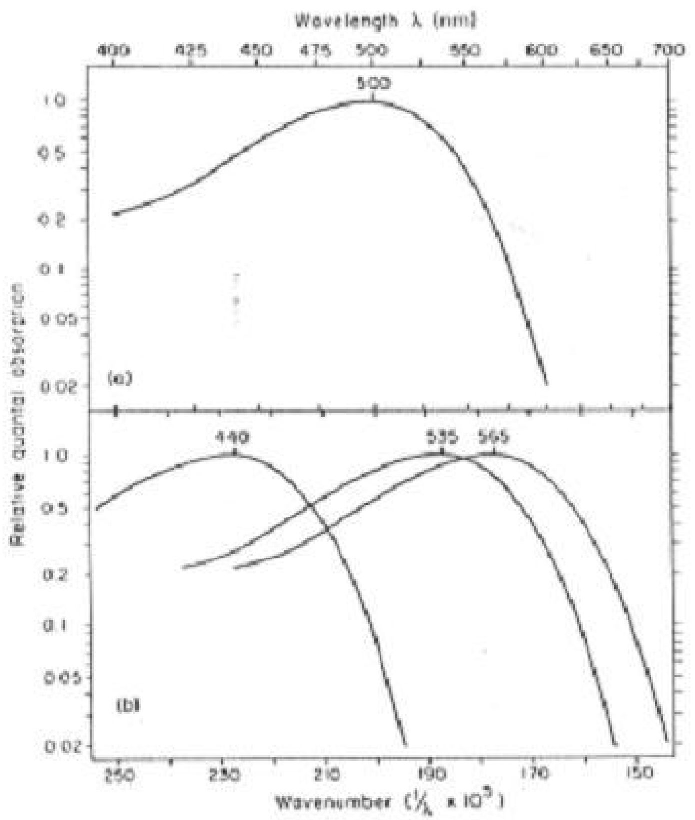


(a)

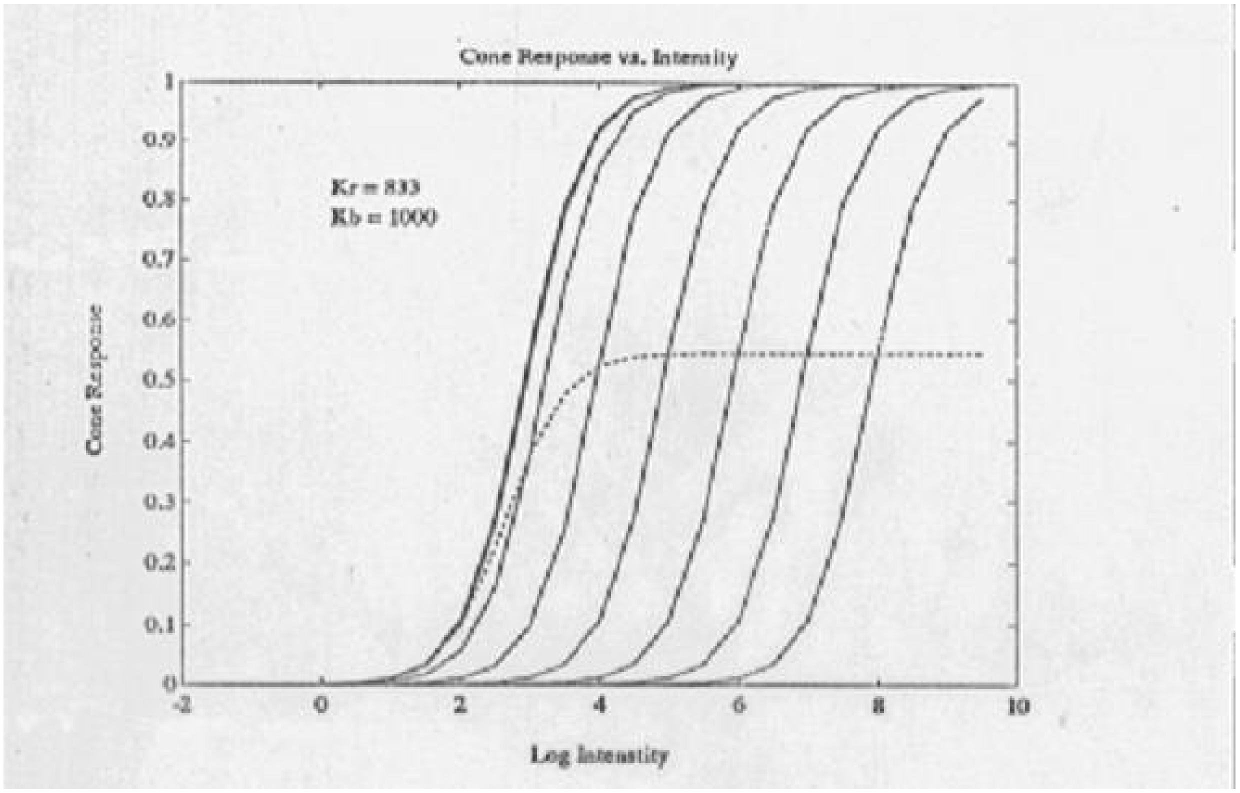


(b)

Rods and Cones



Dynamic range



Retina scheme

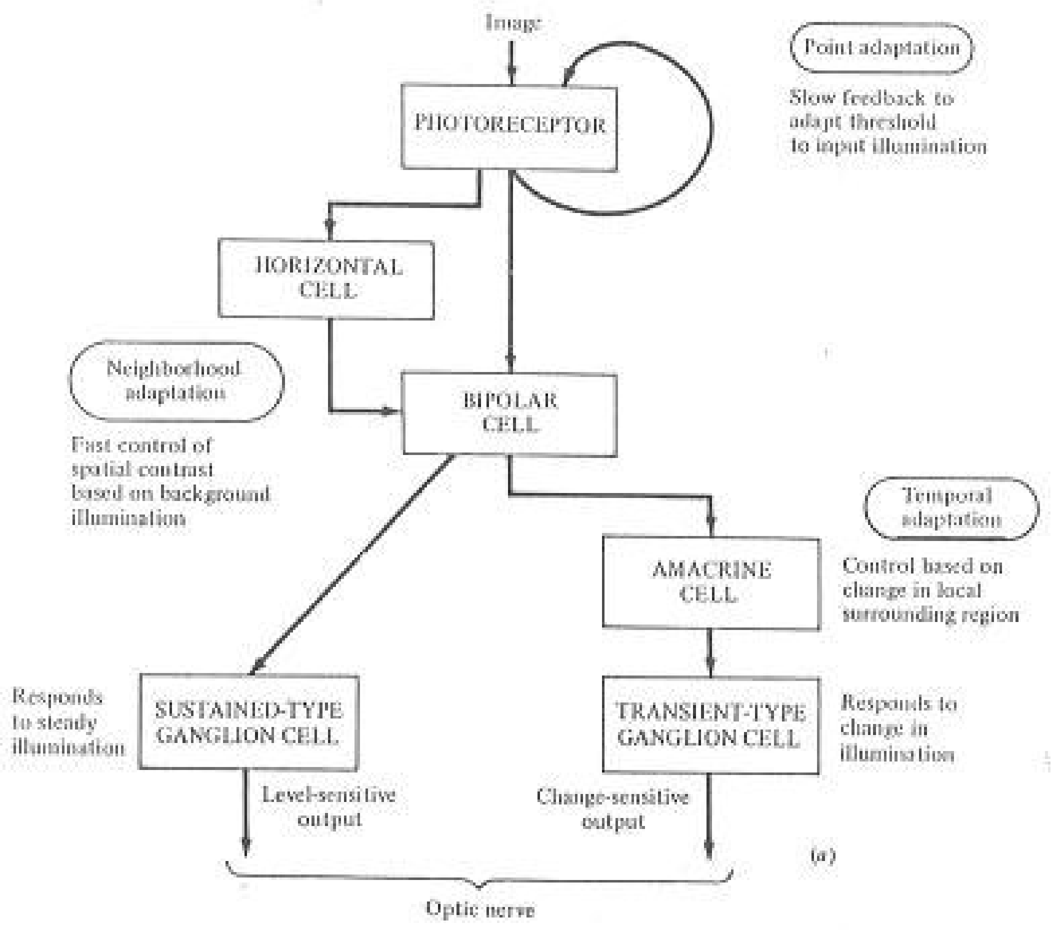
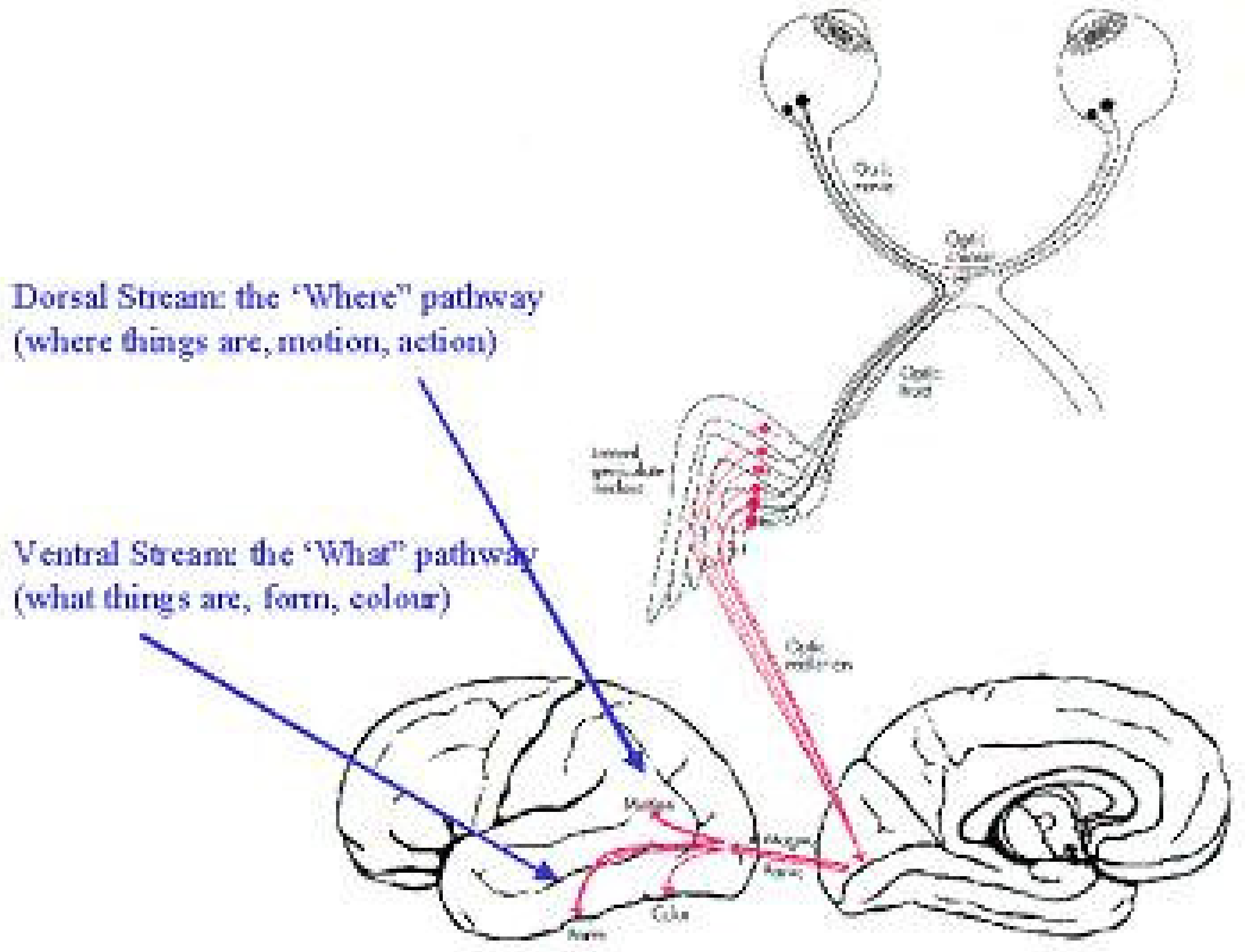
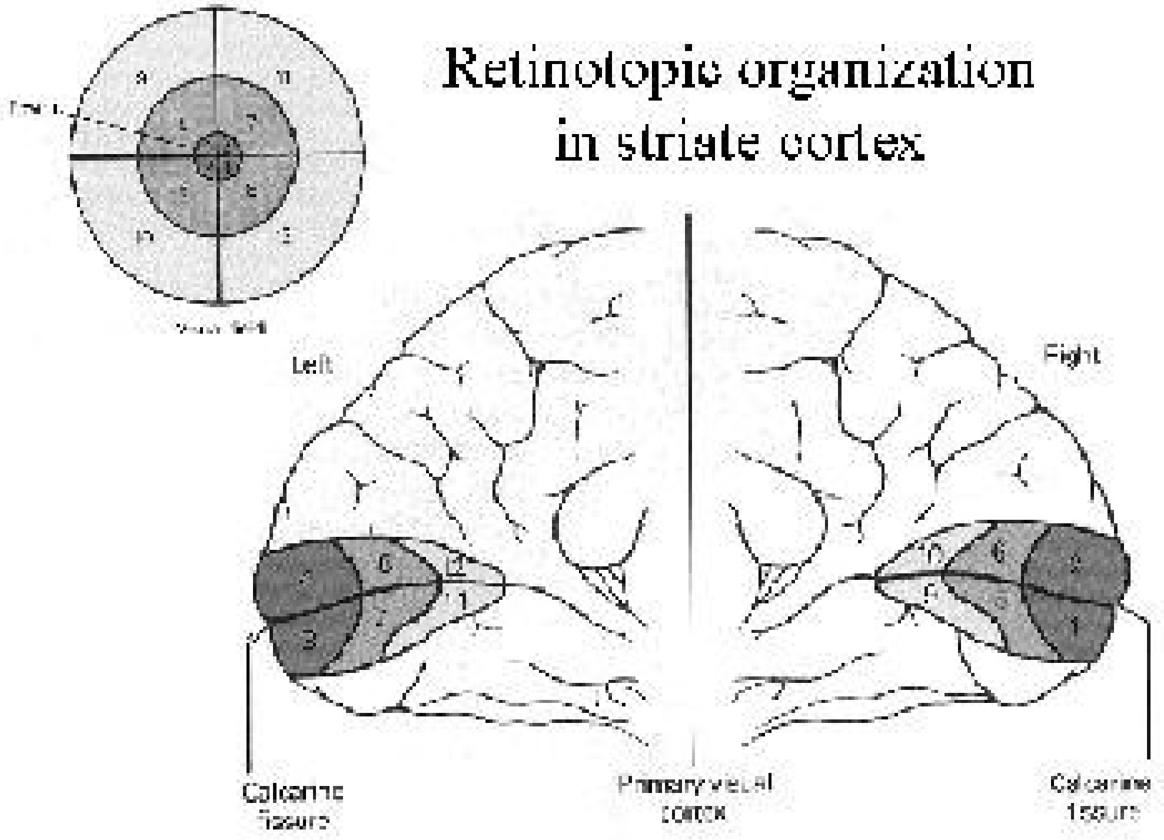


Image
PHOTORECEPTOR
Point adaptation
Slow feedback to adapt threshold to input illumination
HORIZONTAL CELL
Neighborhood adaptation
Fast control of spatial contrast based on background illumination
BIPOLAR CELL
Temporal adaptation
AMACRINE CELL
Control based on change in local surrounding region
Responds to steady illumination
SUSTAINED-TYPE GANGLION CELL
TRANSIENT-TYPE GANGLION CELL
Responds to change in illumination
Level-sensitive output
Change-sensitive output
(a)
Optic nerve

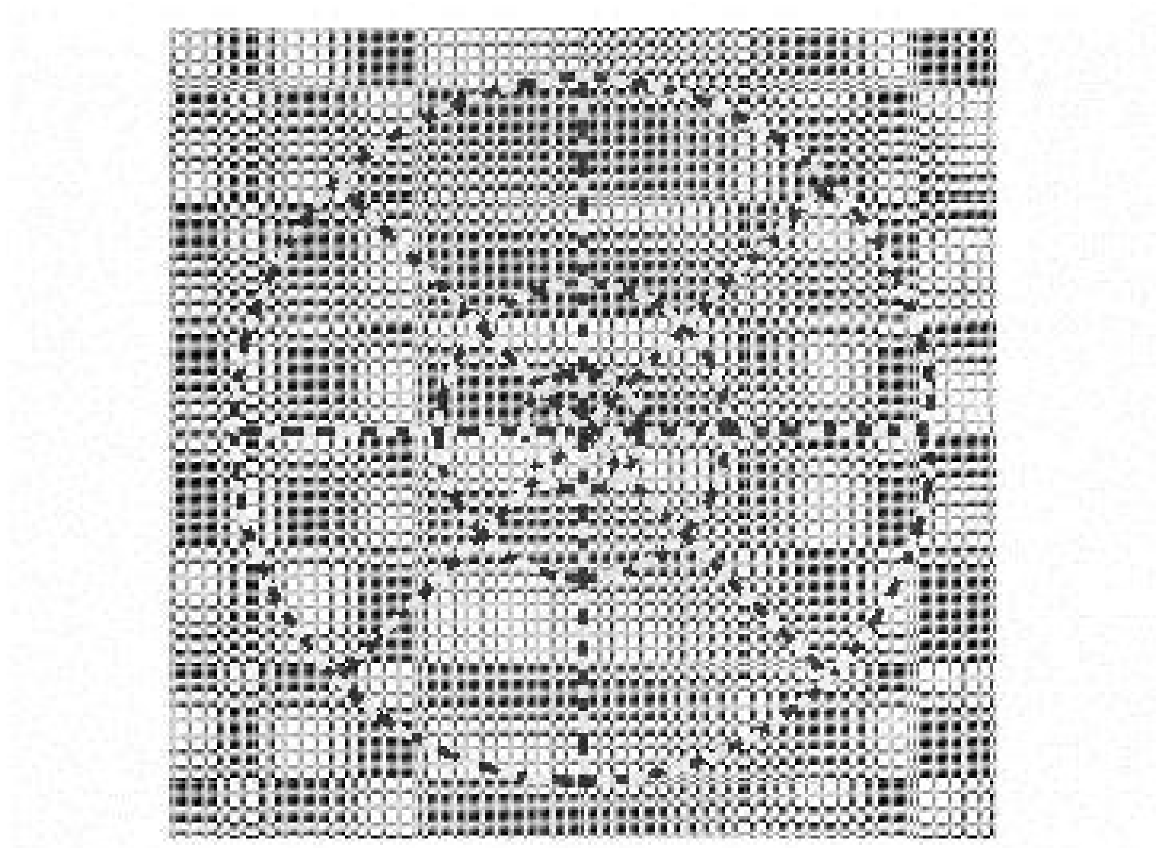
What & Where


Dorsal Stream: the 'Where" pathway
(where things are, motion, action)
Ventral Stream: the 'What" pathway
(what things are, form, colour)

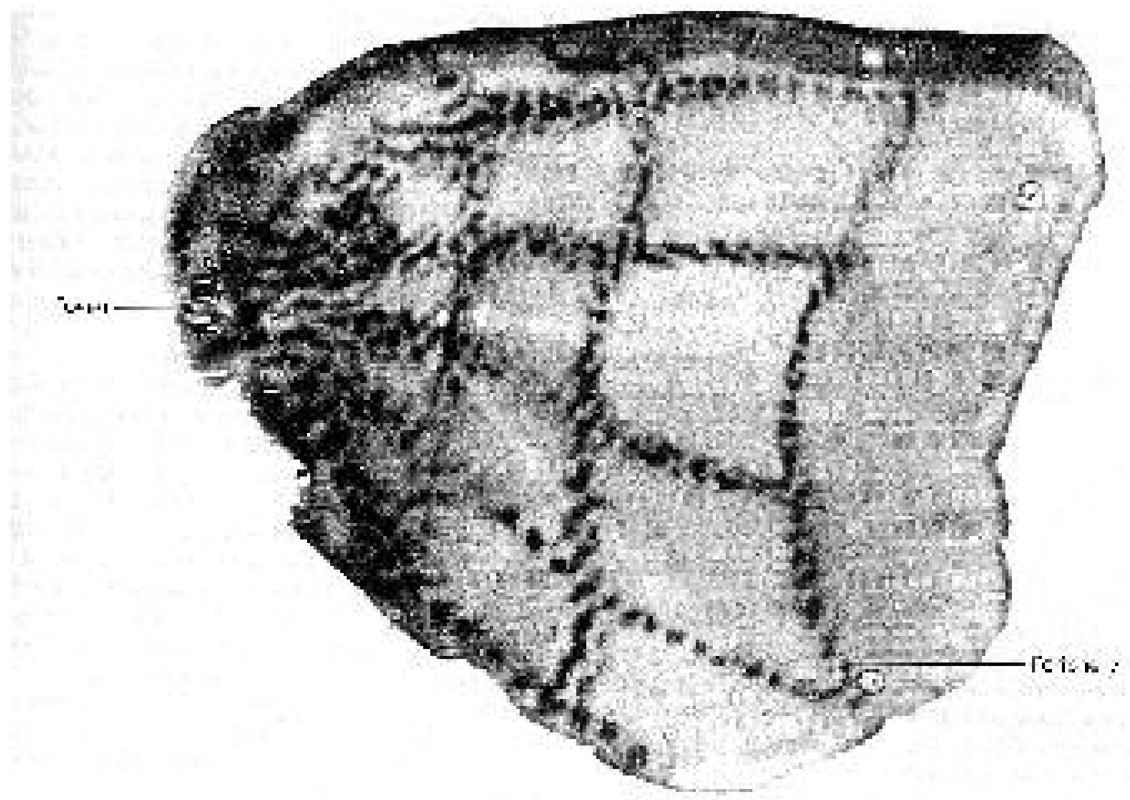
Retinotopic projections



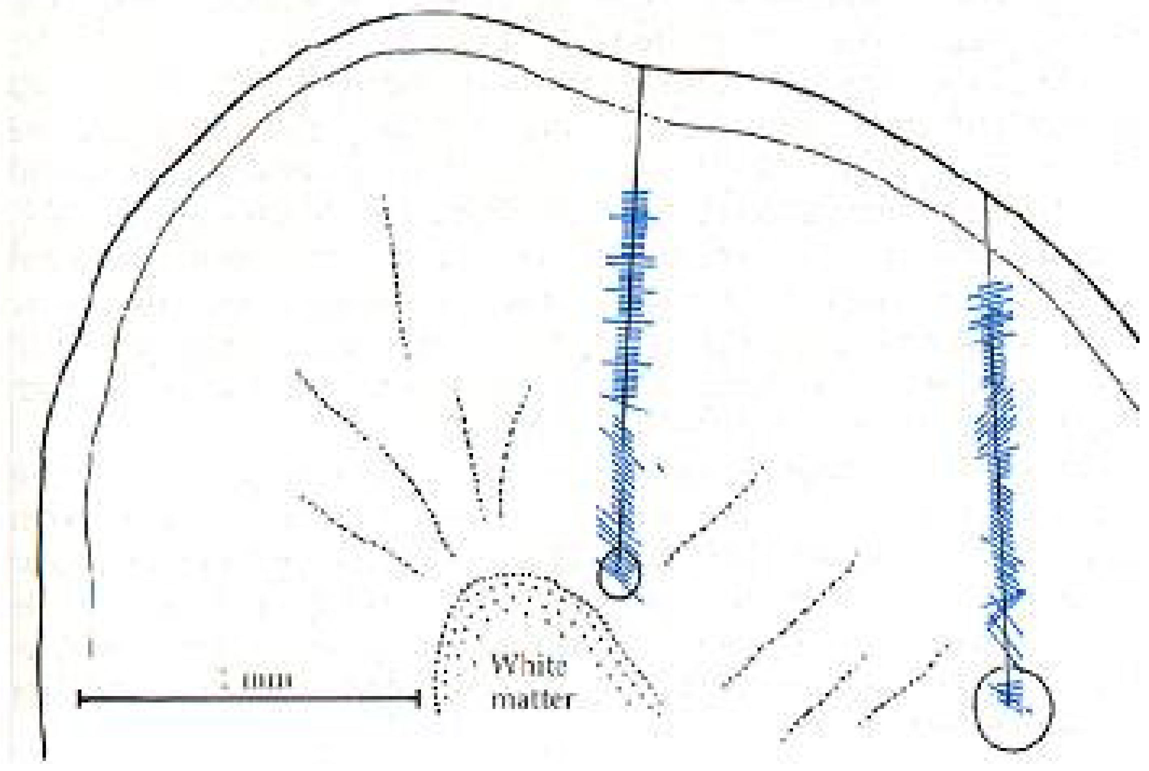
Log polar



Log polar



Orientation columns



V1 structure

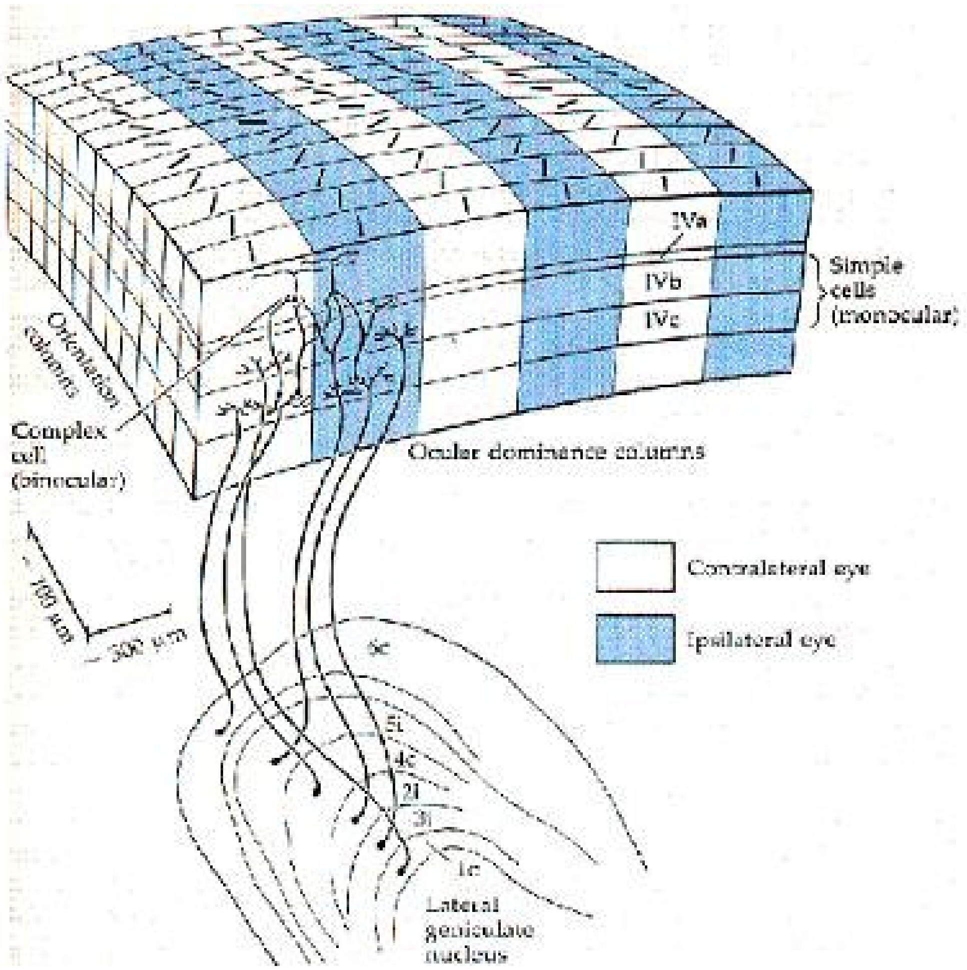
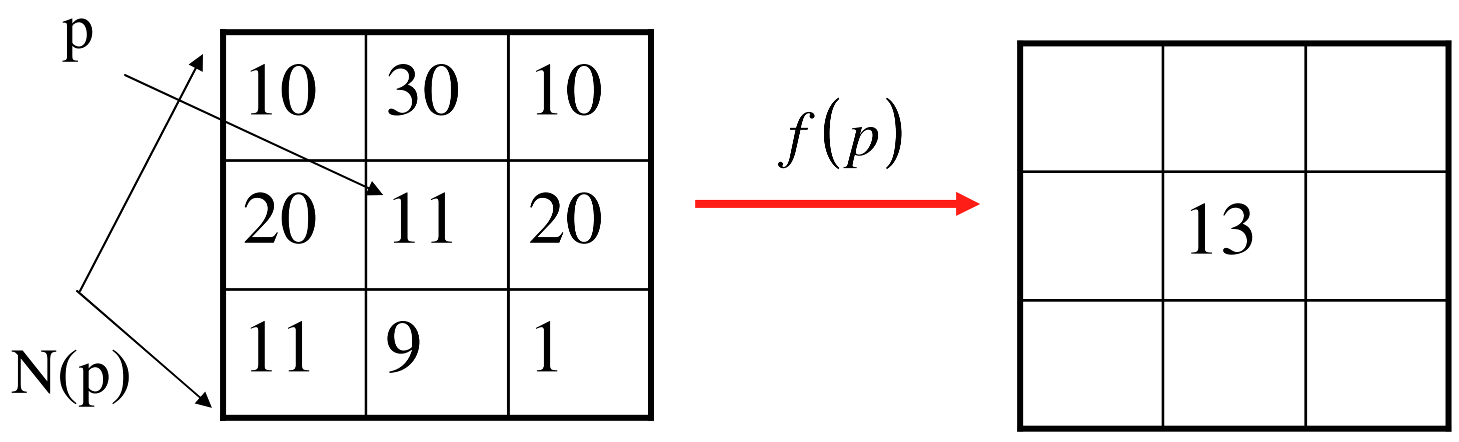


Image Processing

Image Filtering

- Modifying the pixels in an image based on some function of a local neighborhood of the pixels



Linear Filtering

- The output is the linear combination of the neighborhood pixels

$$f(p) = \sum_{q_i \in N(p)} a_i q_i$$

- The coefficients of this linear combination combine to form the "filter-kernel"

$$\begin{array}{|c|c|c|} \hline 1 & 3 & 0 \\ \hline 2 & 10 & 2 \\ \hline 4 & 1 & 1 \\ \hline \end{array} \otimes \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0.1 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline & & \\ \hline & 5 & \\ \hline & & \\ \hline \end{array}$$

Image   Kernel   Filter Output

Convolution

$$f(i,j) = I * H = \sum_{k}\sum_{l} I(k,l)\,H(i-k,\,j-l)$$

$I = \text{Image}$

$H = \text{Kernel}$

$$\begin{tikzcd}
 & \begin{array}{|c|c|c|}\hline H_7 & H_8 & H_9 \\ \hline H_4 & H_5 & H_6 \\ \hline H_1 & H_2 & H_3 \\ \hline \end{array} \arrow[d, "Y-flip"] & \overset{H}{\begin{array}{|c|c|c|}\hline H_1 & H_2 & H_3 \\ \hline H_4 & H_5 & H_6 \\ \hline H_7 & H_8 & H_9 \\ \hline \end{array}} \arrow[l, "X-flip"'] \\
\overset{I}{\begin{array}{|c|c|c|}\hline I_1 & I_2 & I_3 \\ \hline I_4 & I_5 & I_6 \\ \hline I_7 & I_8 & I_9 \\ \hline \end{array}} \arrow[r, phantom, "\otimes"] & \begin{array}{|c|c|c|}\hline H_9 & H_8 & H_7 \\ \hline H_6 & H_5 & H_4 \\ \hline H_3 & H_2 & H_1 \\ \hline \end{array} \arrow[r] & \begin{array}{l} I * H = I_1H_9 + I_2H_8 + I_3H_7 \\ \quad + I_4H_6 + I_5H_5 + I_6H_4 \\ \quad + I_7H_3 + I_8H_2 + I_9H_1 \end{array}
\end{tikzcd}$$

Separable kernels

$$f(i,j) = I * H = \sum_k \sum_l I(k,l) H(i-k, j-l)$$

Height*width multiplications for each pixel

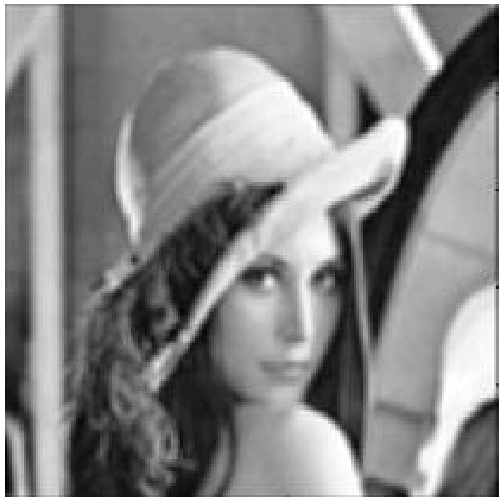
$$H = H_1 * H_2$$

$$I * H = I * H_1 * H_2$$

Height+width multiplications for each pixel

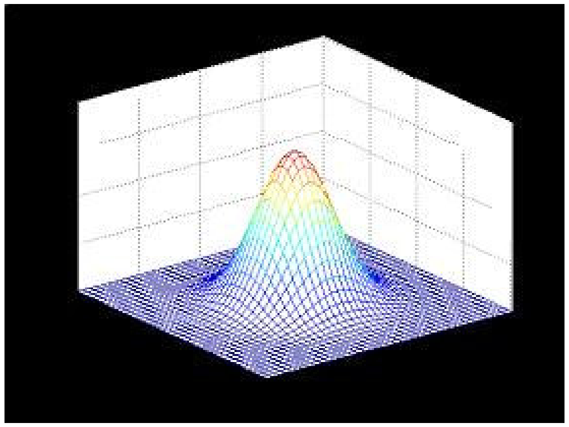
Linear Filtering

$$* \frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} =$$



Gaussian Filter

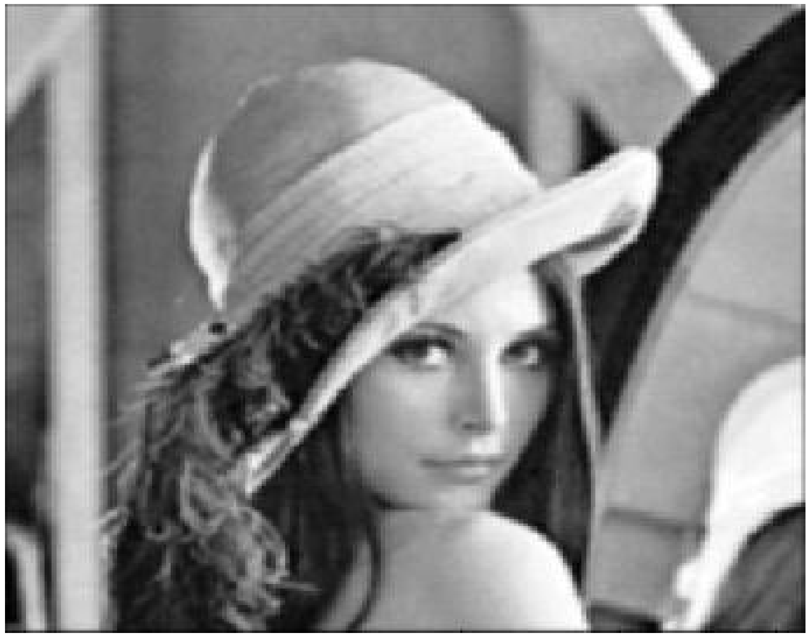
$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{\left(x^2 + y^2\right)}{2\sigma^2}\right)$$



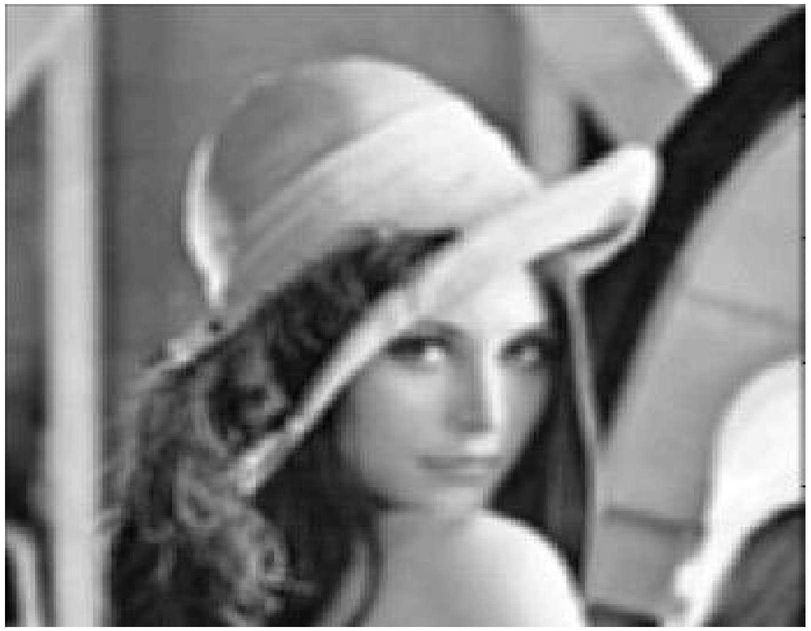
$$H(i, j) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{\left((i-k-1)^2 + (j-k-1)^2\right)}{2\sigma^2}\right)$$

where $H(i, j)$ is $(2k+1)\times(2k+1)$ array

Gaussian Vs Average



Gaussian Smoothing



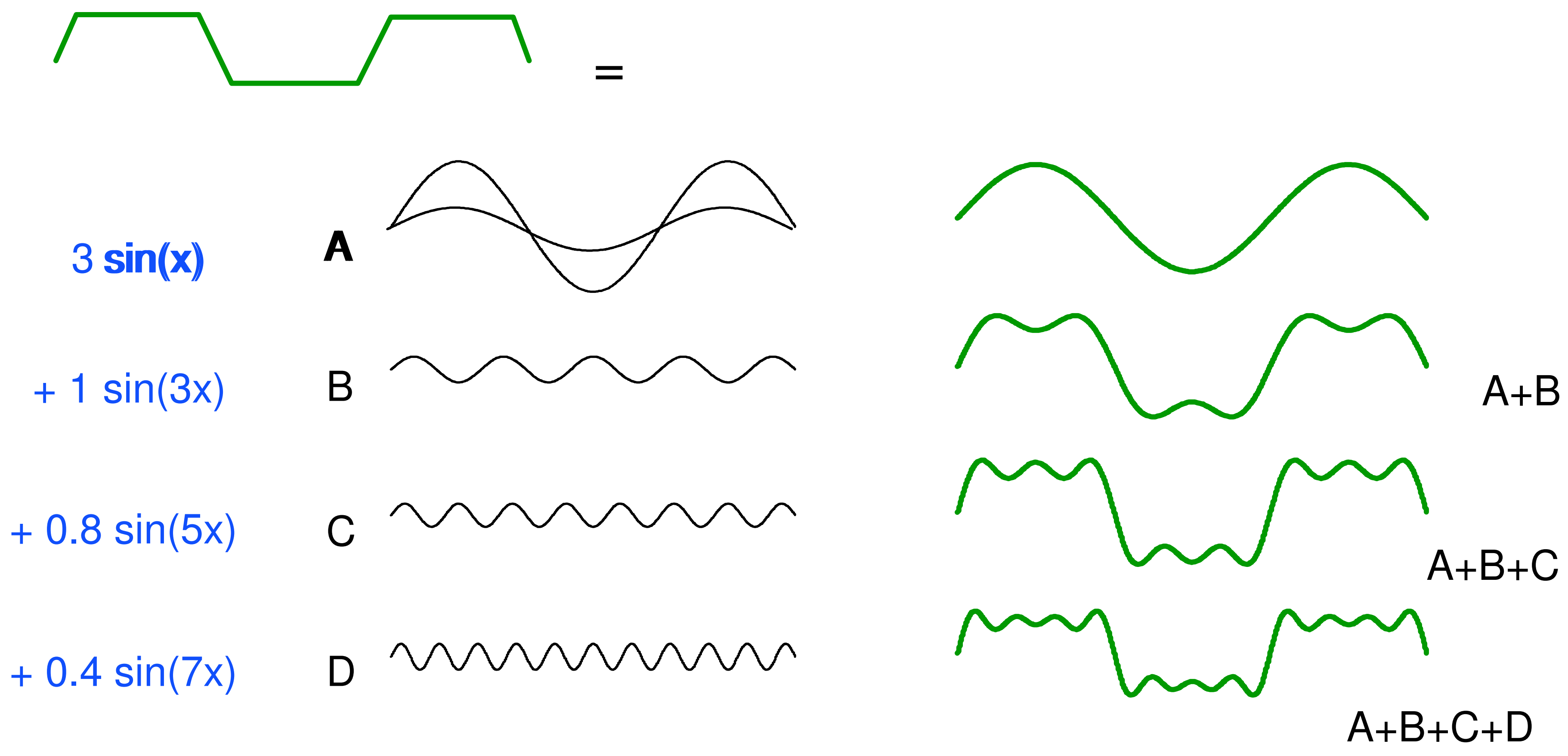
Smoothing by Averaging

Fourier Transform

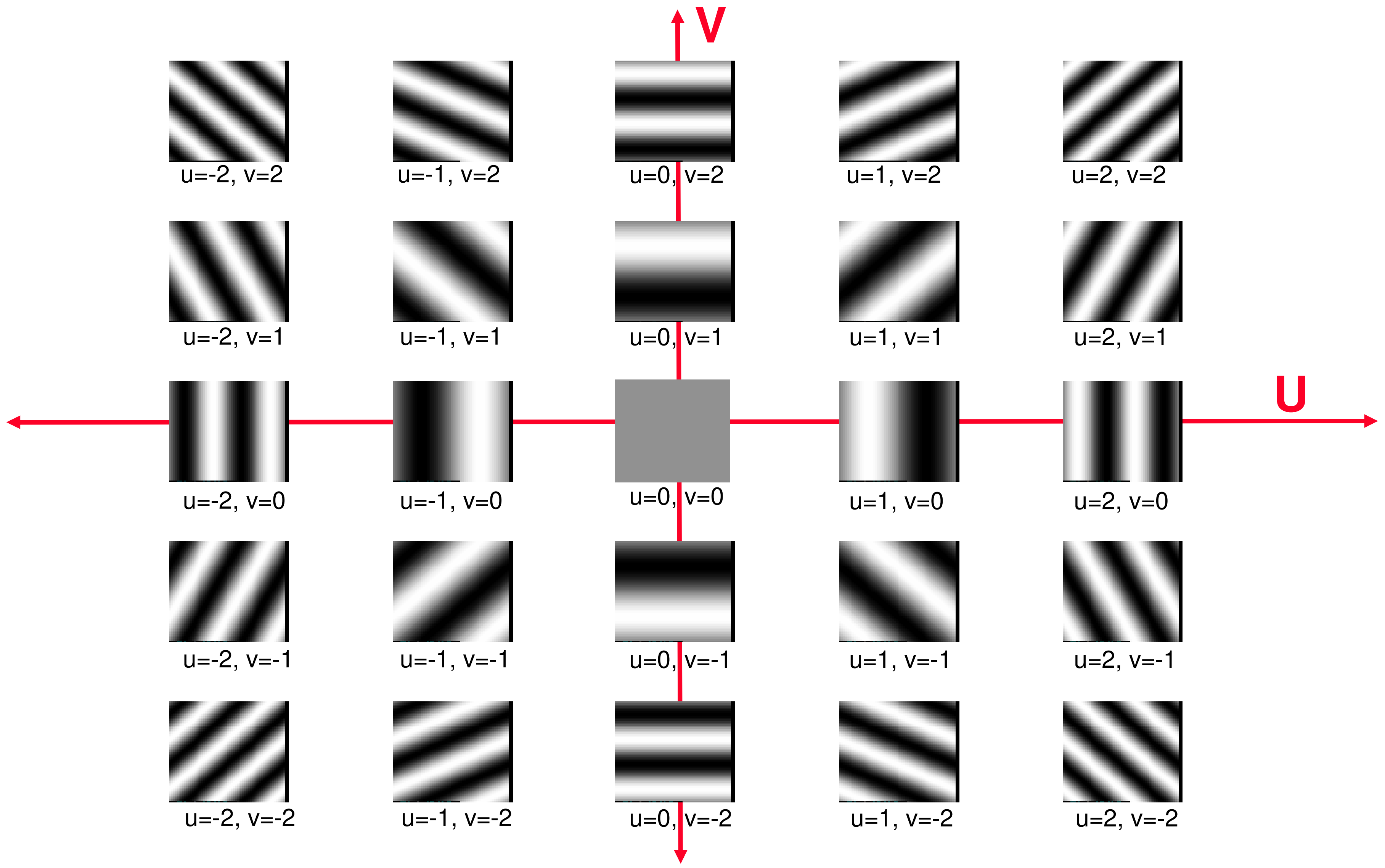
Continuous : $F\left(g\left(x,y\right)\right)\left(u,v\right)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}g\left(x,y\right)e^{-i2\pi\left(ux+vy\right)}dxdy$

Discrete $$F[m,n]=\sum_{k=0}^{M-1}\sum_{l=0}^{N-1}f[k,l]e^{-\pi i\left(\frac{km}{M}+\frac{\ln}{N}\right)}$$

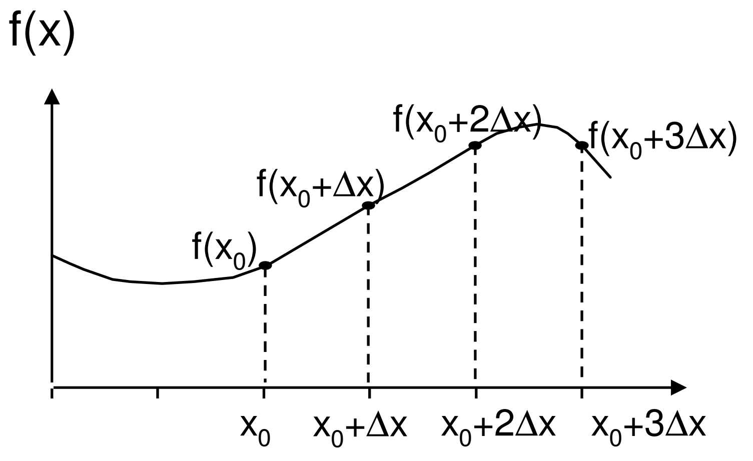
A sum of sines and cosines

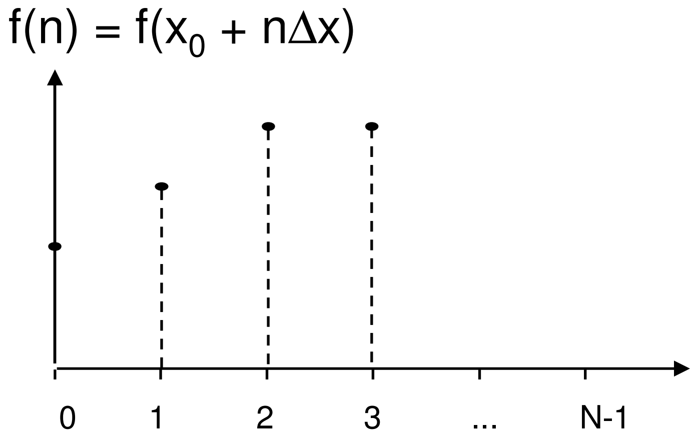


The 2D Basis Functions $e^{2\pi i(ux+vy)}$



Discrete Functions



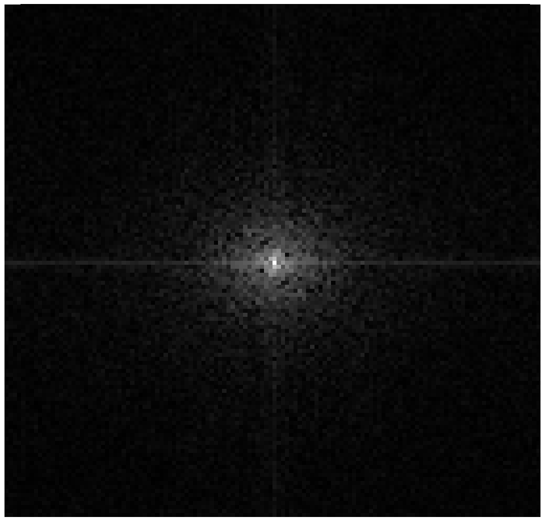


The discrete function f:
{ f(0), f(1), f(2), … , f(N-1) }

The Fourier Image

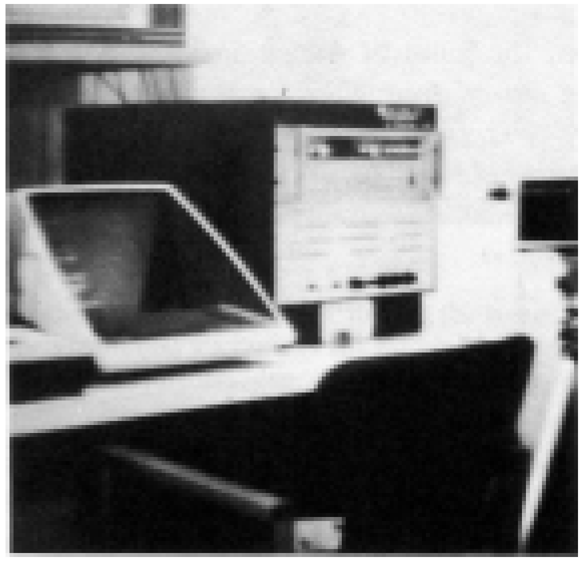
Image f

Fourier spectrum |F(u,v)| log(1 + |F(u,v)|)

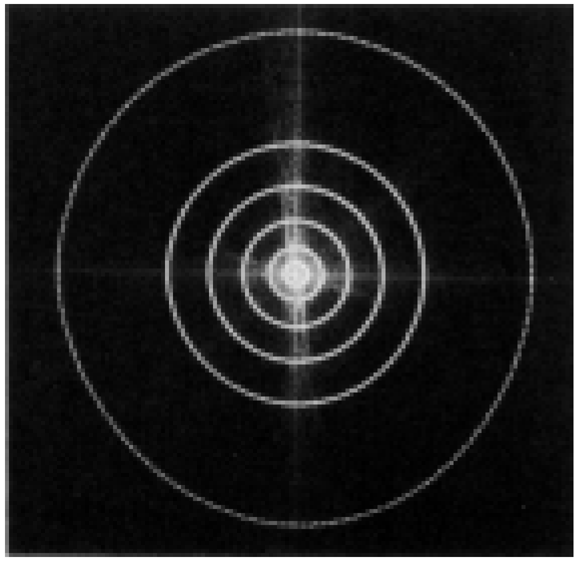


Frequency Bands

Image



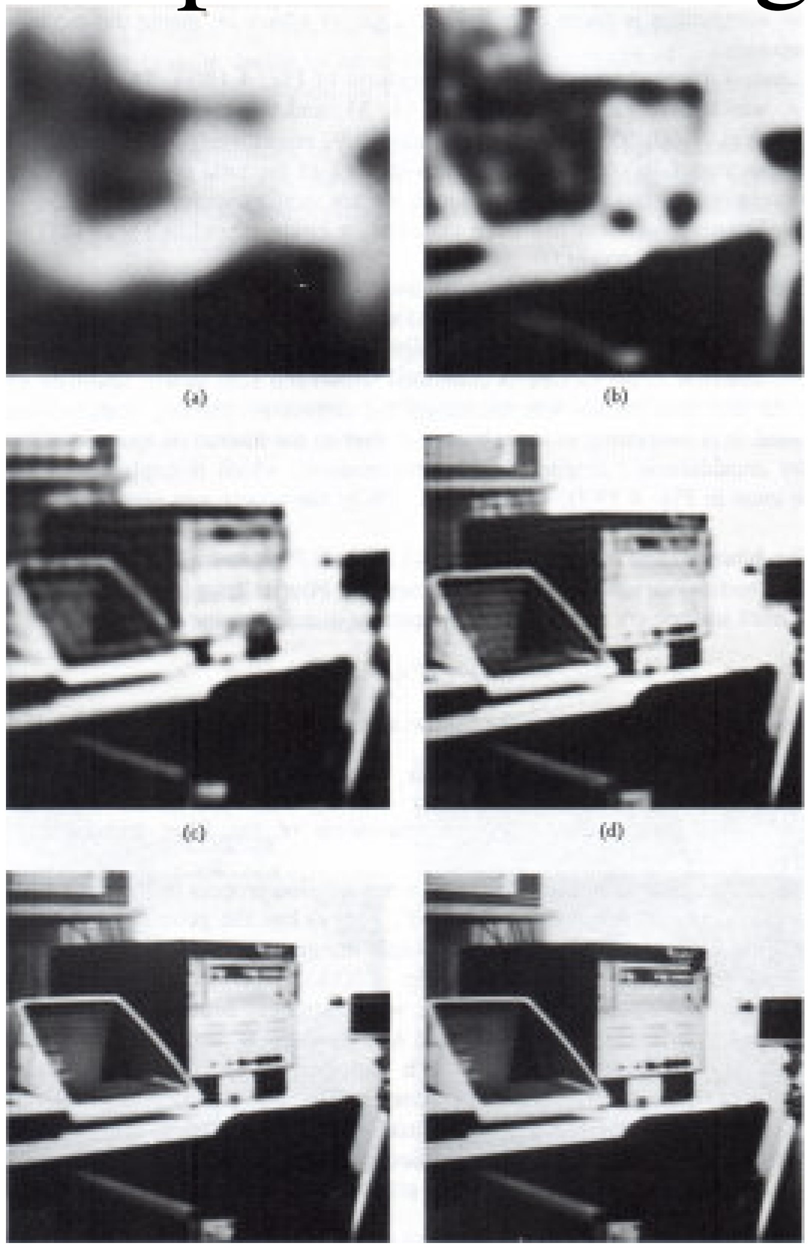
Fourier Spectrum



Percentage of image power enclosed in circles (small to large) :

90%, 95%, 98%, 99%, 99.5%, 99.9%

Low pass Filtering



90%

95%

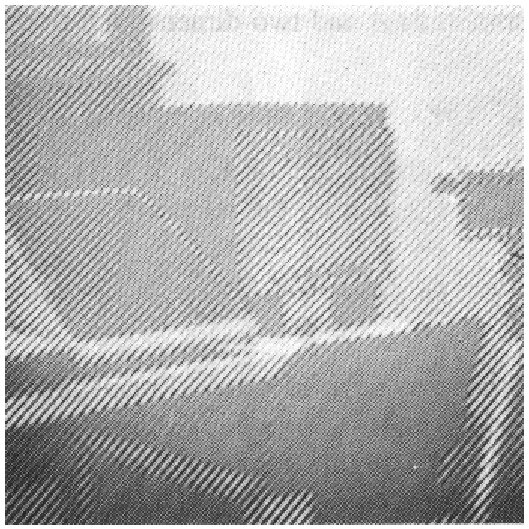
98%

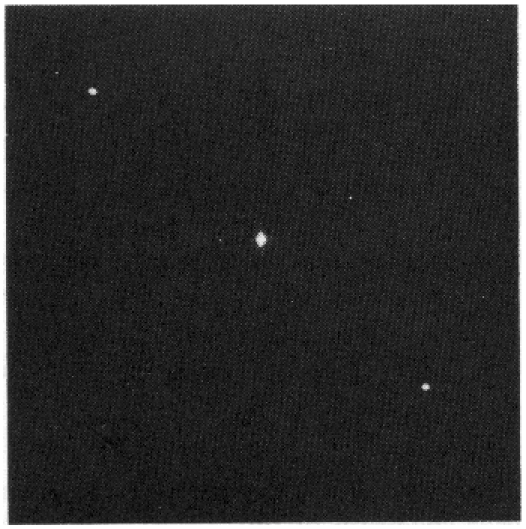
99%

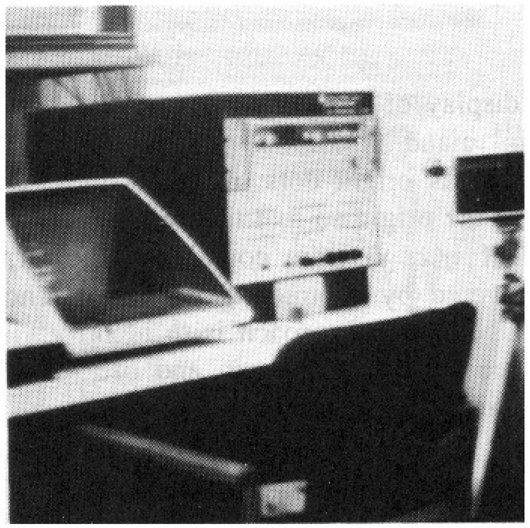
99.5%

99.9%

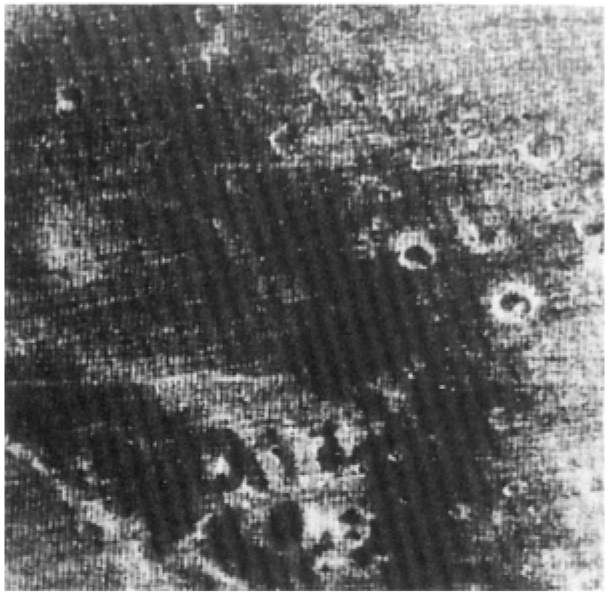
Noise Removal

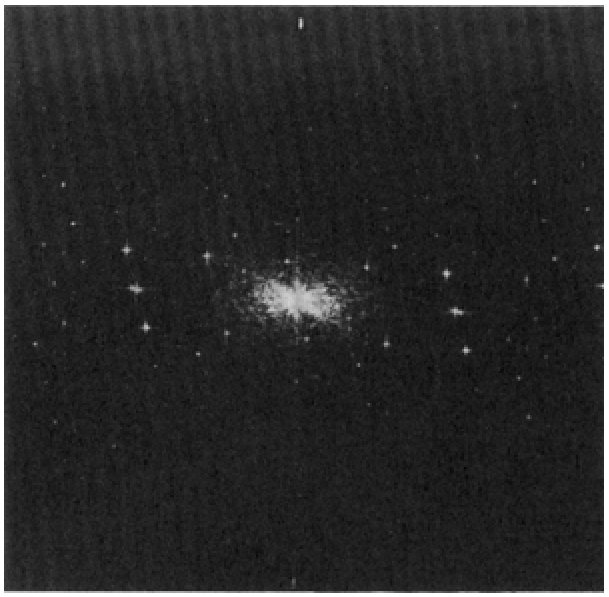

Noisy image

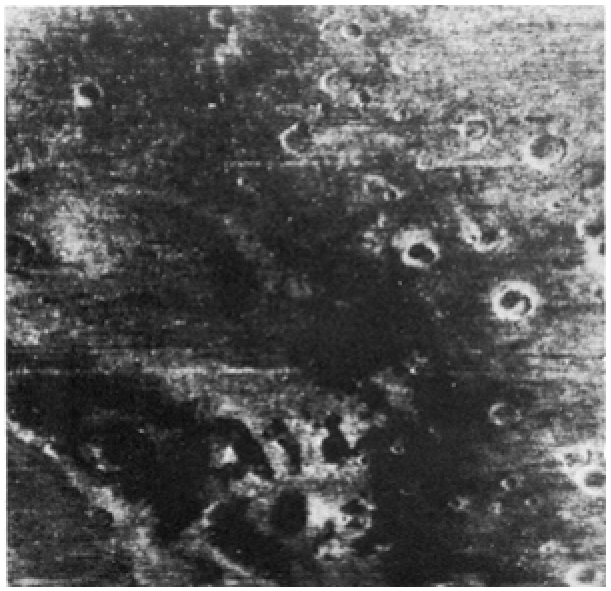

Fourier Spectrum


Noise-cleaned image

Noise Removal

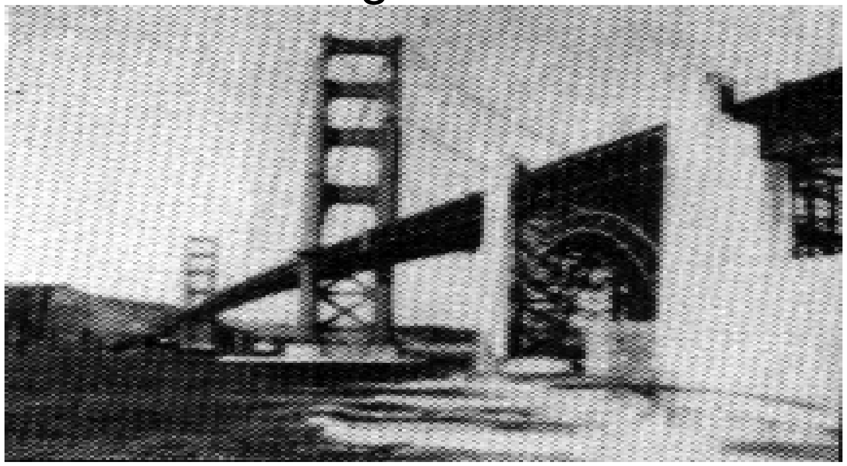

Noisy image


Fourier Spectrum

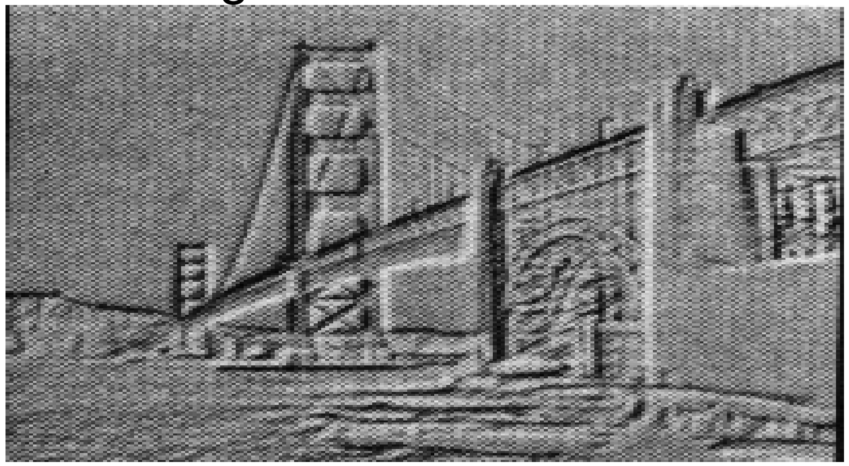

Noise-cleaned image

High Pass Filtering

Original

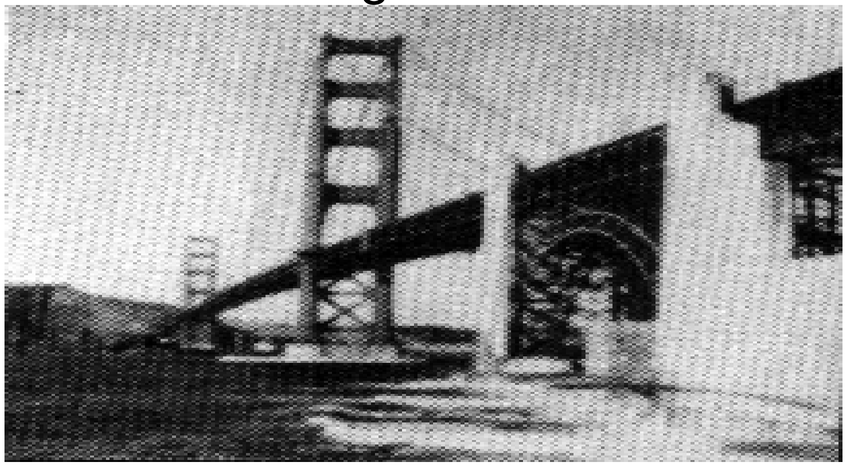


High Pass Filtered



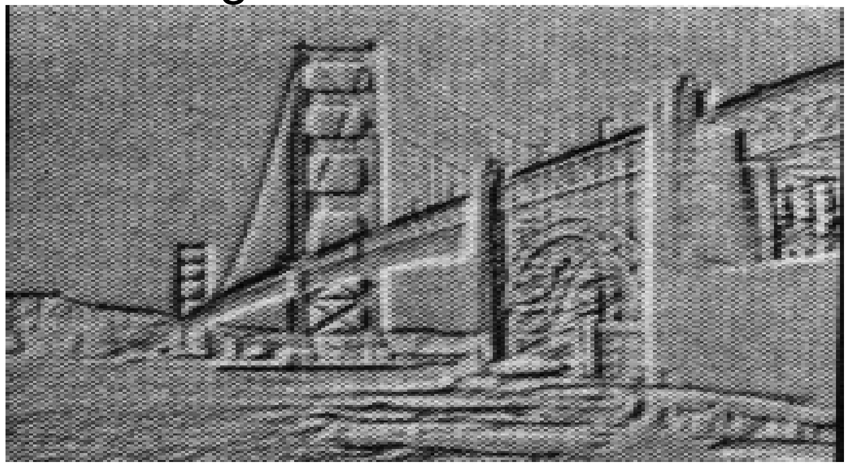
High Frequency Emphasis

Original



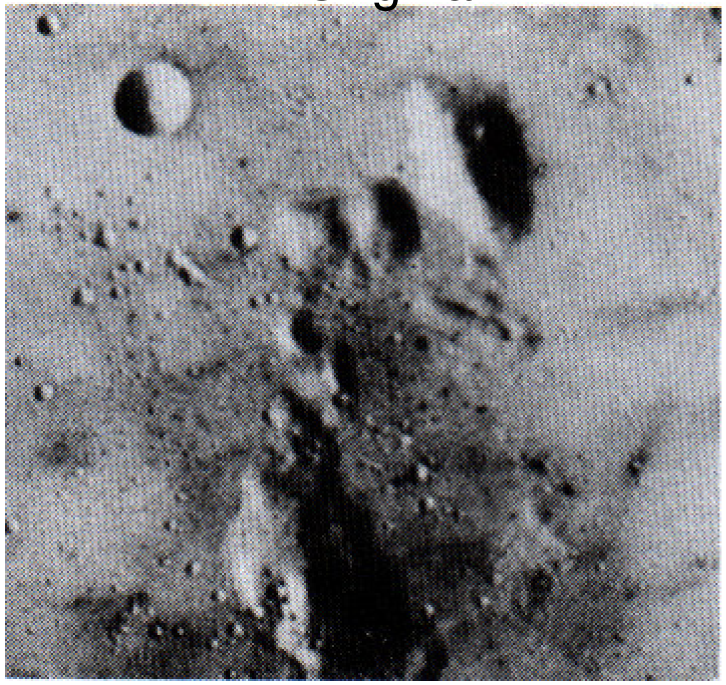
+

High Pass Filtered

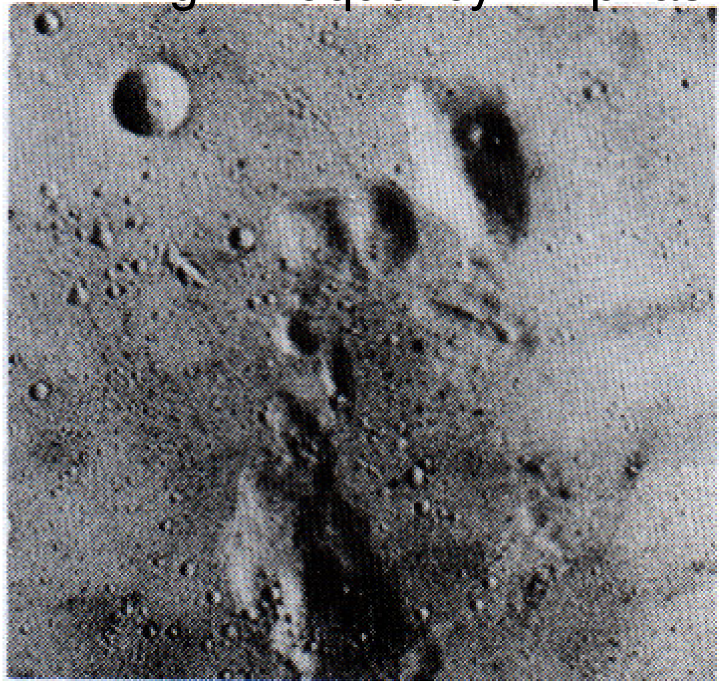


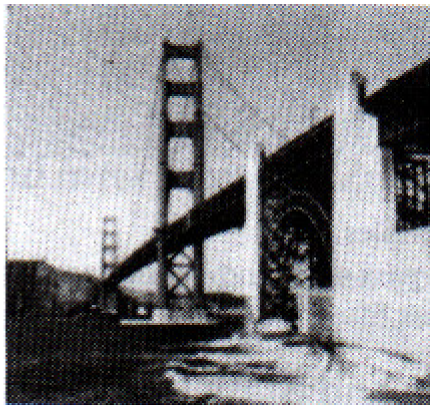
High Frequency Emphasis

Original

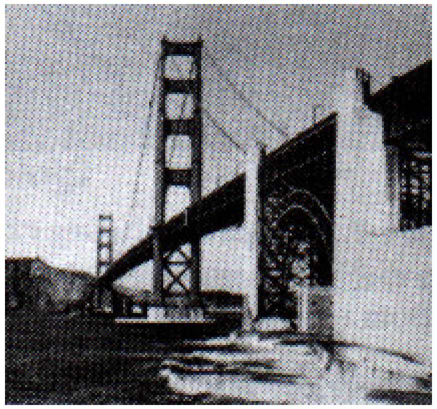


High Frequency Emphasis





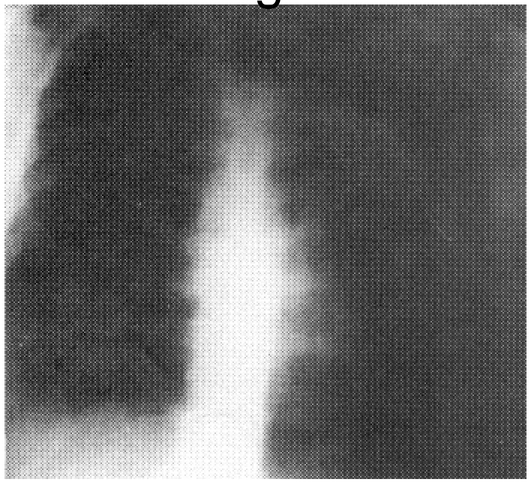
Original



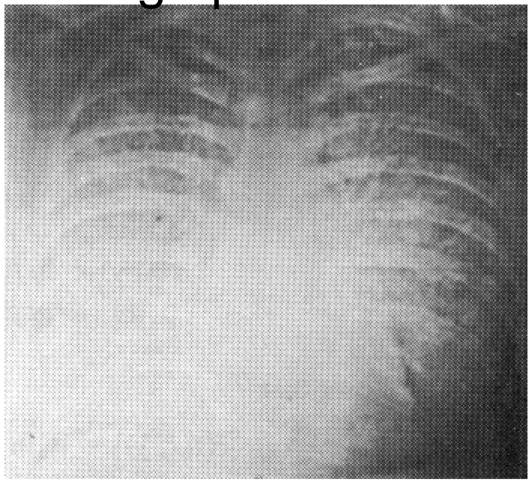
High Frequency Emphasis

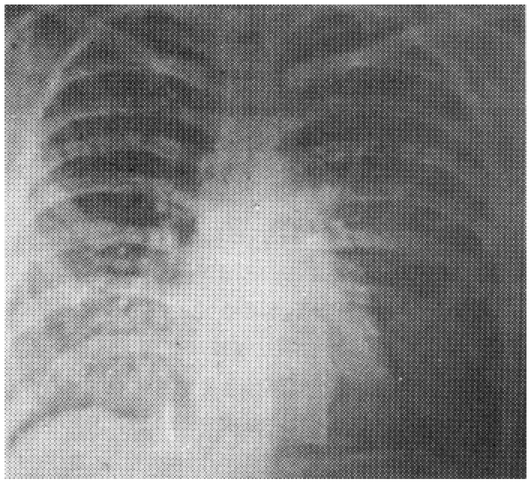
High Frequency Emphasis

Original

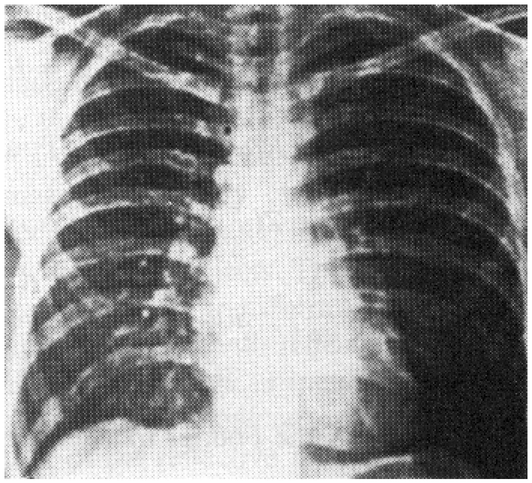


High pass Filter





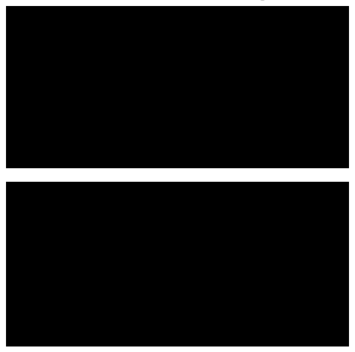
High Frequency Emphasis



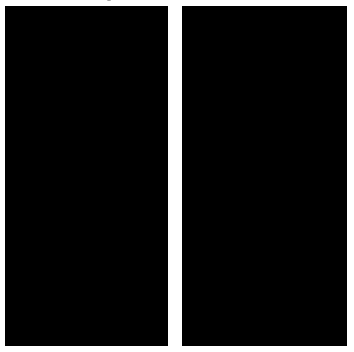
High Frequency Emphasis
+
Histogram Equalization

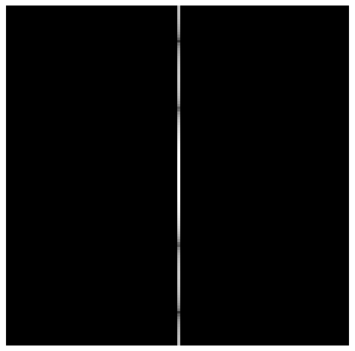
Rotation

2D Image

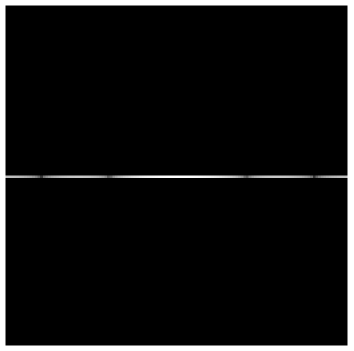


2D Image - Rotated





Fourier Spectrum

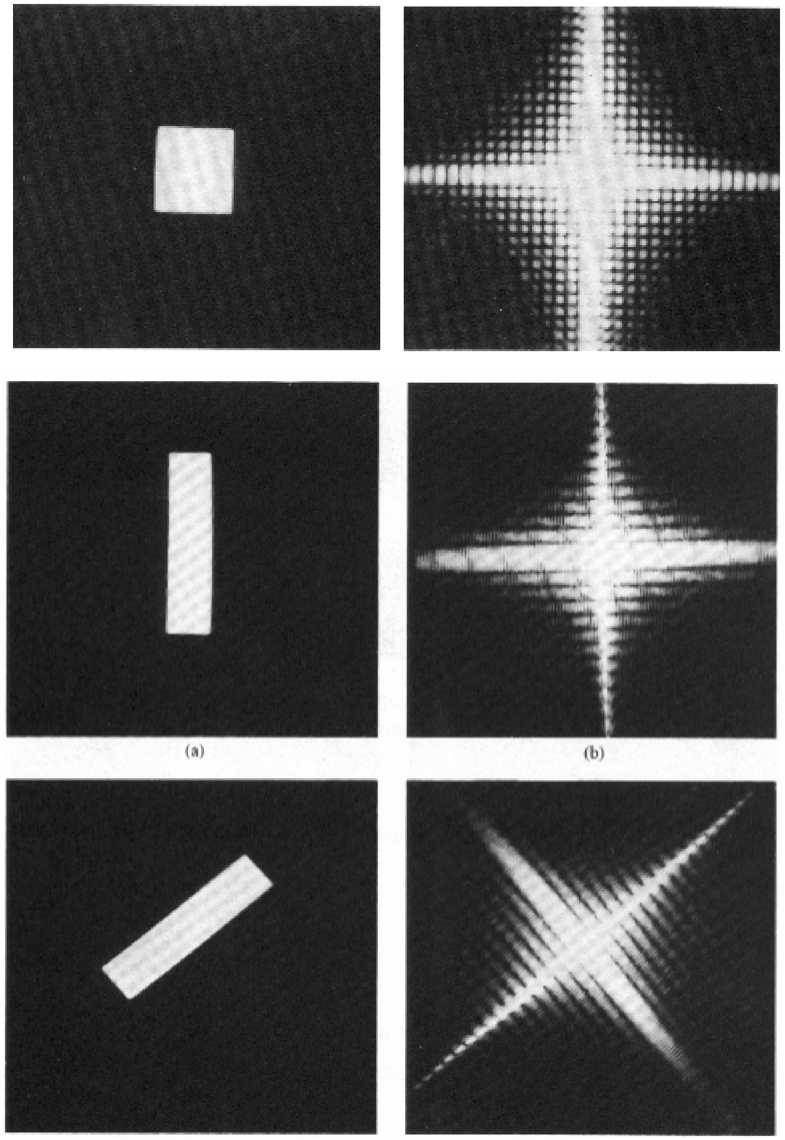


Fourier Spectrum

Fourier Transform -- Examples

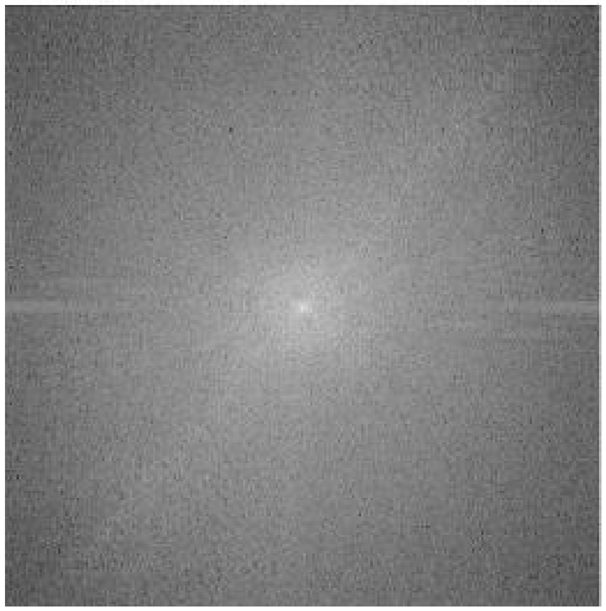
Image Domain

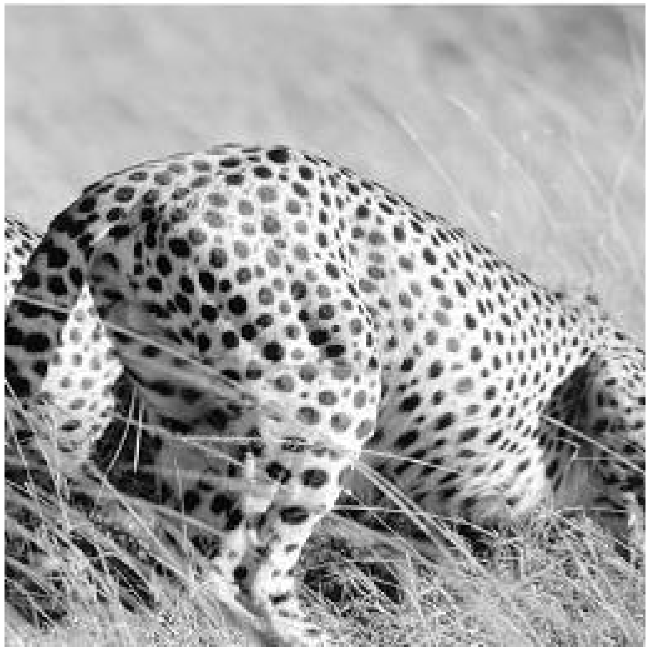
Frequency Domain

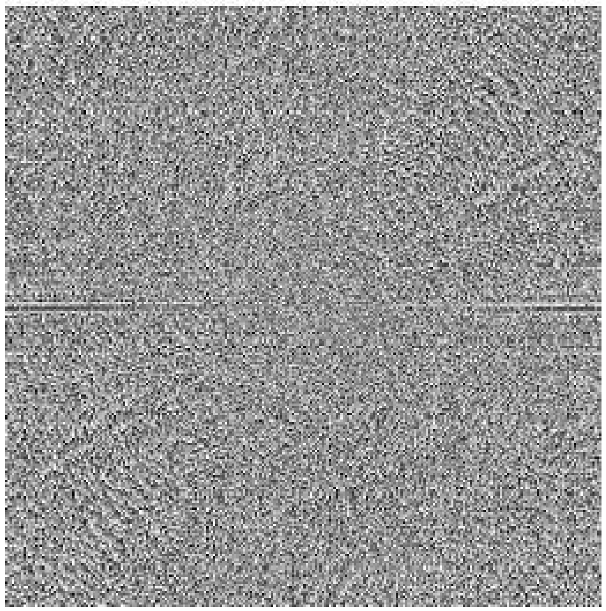


Phase and Magnitude

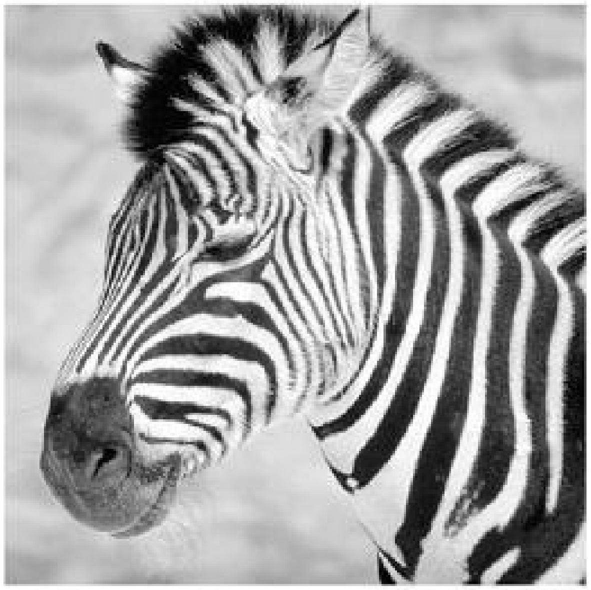
- Fourier transform of a real function is complex
  - difficult to plot, visualize
  - instead, we can think of the phase and magnitude of the transform
- Phase is the phase of the complex transform
- Magnitude is the magnitude of the complex transform
- Curious fact
  - all natural images have about the same magnitude transform
  - hence, phase seems to matter, but magnitude largely doesn't
- Demonstration
  - Take two pictures, swap the phase transforms, compute the inverse - what does the result look like?

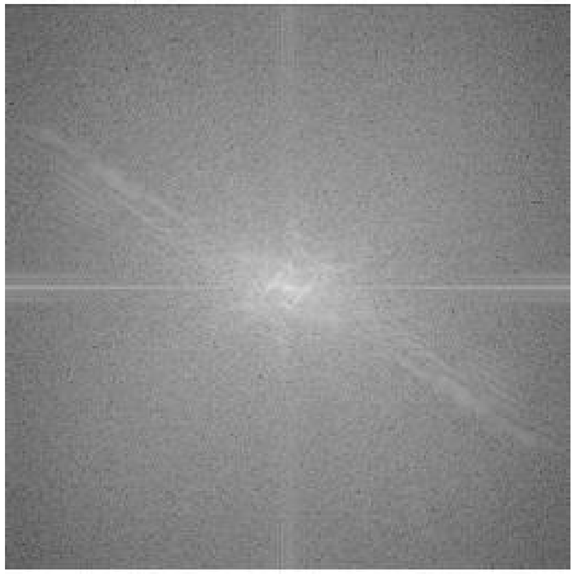


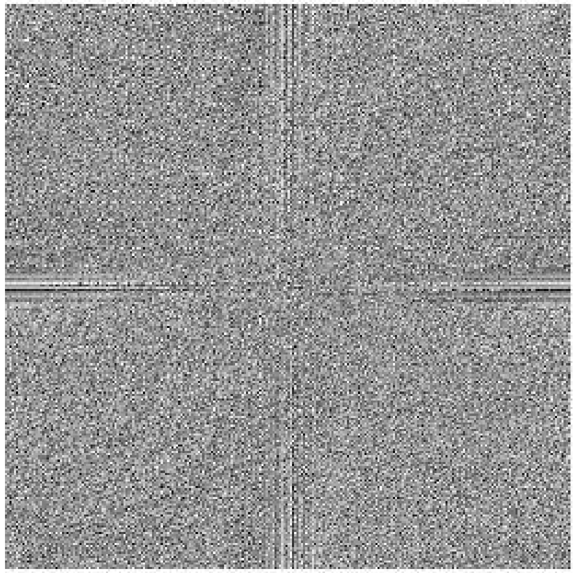




Cheetah Image
Fourier Magnitude (above)
Fourier Phase (below)

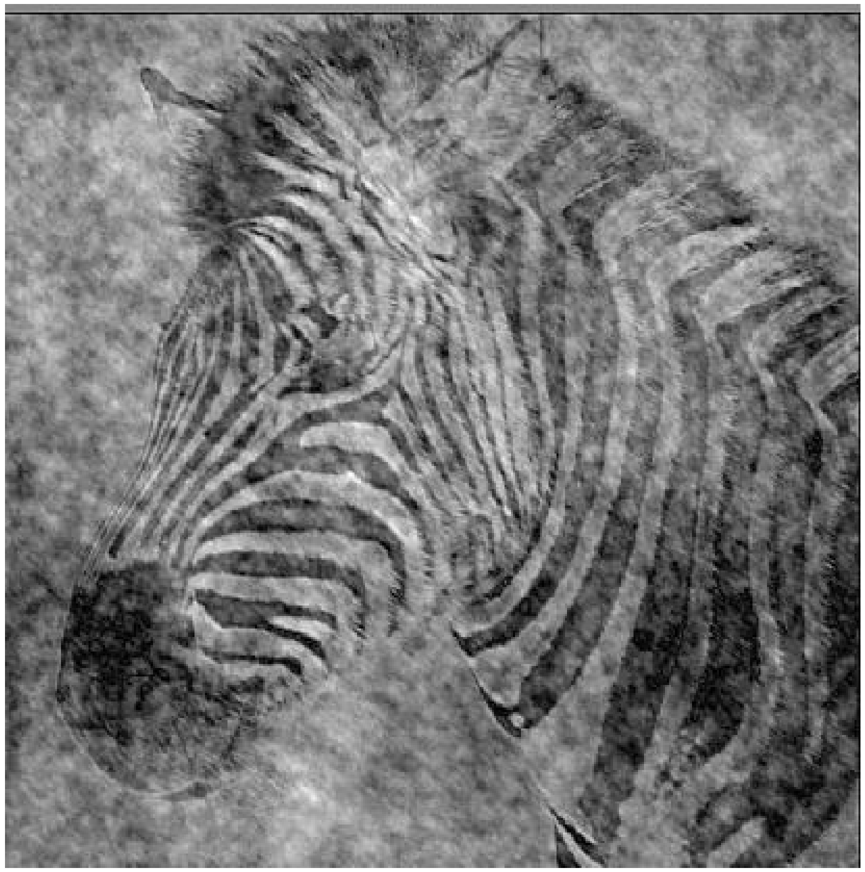




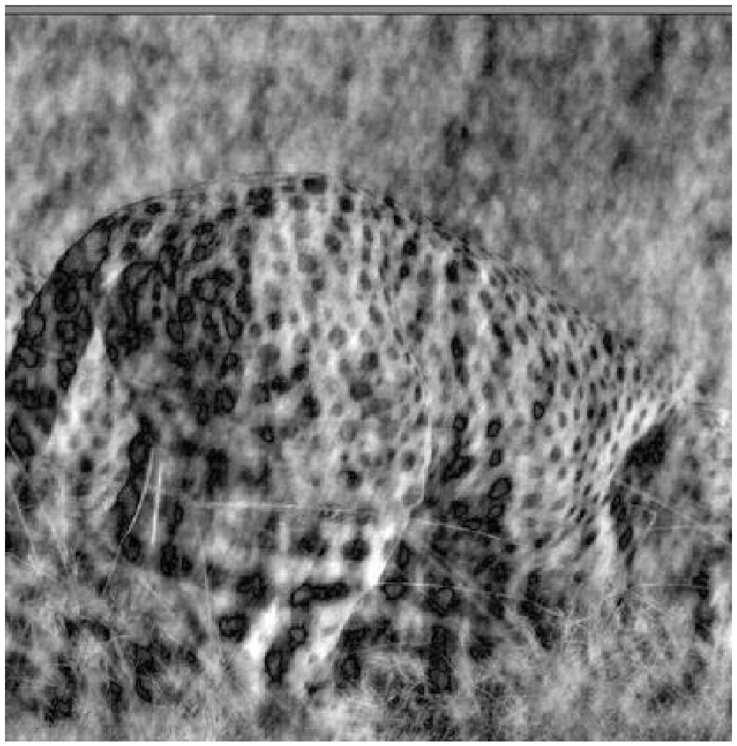


Zebra Image
Fourier Magnitude (above)
Fourier Phase (below)

Reconstruction with
Zebra phase,
Cheetah Magnitude

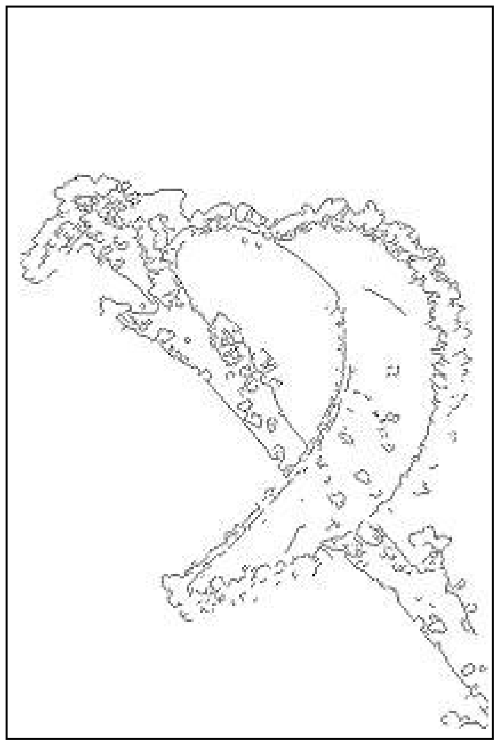


Reconstruction with
Cheetah phase,
Zebra Magnitude



Edge Detection in Images

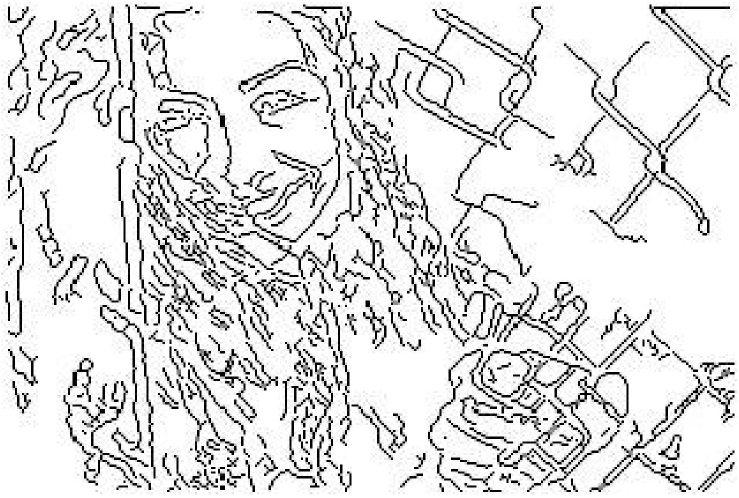
- Finding the contour of objects in a scene



Edge Detection in Images

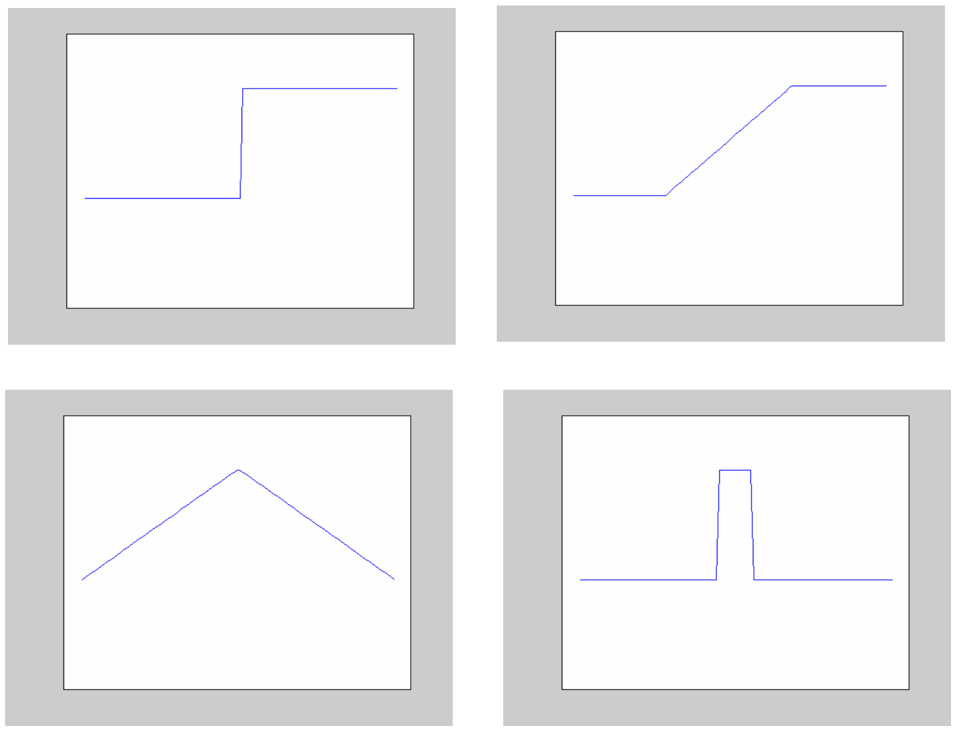
- What is an object?

  It is one of the goals of computer vision to identify objects in scenes.



What is an Edge

- Lets define an edge to be a discontinuity in image intensity function.
- Edge Models
  - Step Edge
  - Ramp Edge
  - Roof Edge
  - Spike Edge



Detecting Discontinuities

- Discontinuities in signal can be detected by computing the derivative of the signal.

Finite Difference in 2D

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\varepsilon \to 0}\left(\frac{f(x+\varepsilon, y) - f(x,y)}{\varepsilon}\right)$$

$$\frac{\partial f(x,y)}{\partial y} = \lim_{\varepsilon \to 0}\left(\frac{f(x, y+\varepsilon) - f(x,y)}{\varepsilon}\right)$$

Definition

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x_{n+1}, y_m) - f(x_n, y_m)}{\Delta x}$$

$$\frac{\partial f(x,y)}{\partial y} \approx \frac{f(x_n, y_{m+1}) - f(x_n, y_m)}{\Delta x}$$

Discrete Approximation

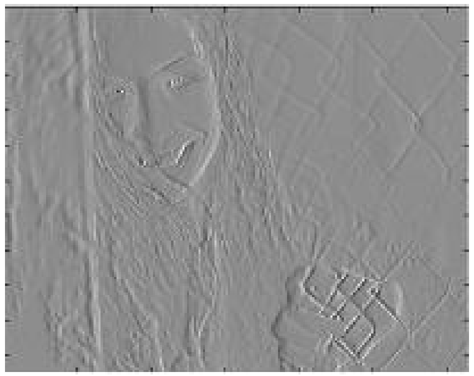
$$\begin{bmatrix} 1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

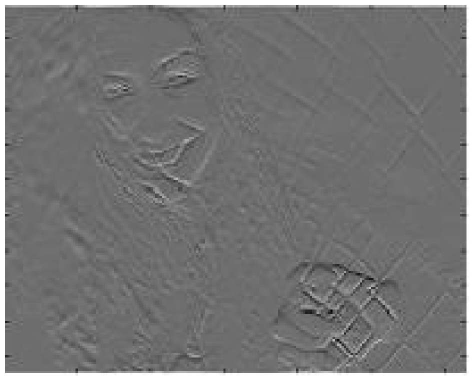
Convolution Kernels

Finite differences

I



$I_x = I * \begin{bmatrix} 1 & -1 \end{bmatrix}$



$I_y = I * \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Classical Operators

Prewitt's Operator

$$\longrightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & -1 \end{bmatrix}$$

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

Smooth

Differentiate

$$\longrightarrow \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

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

Classical Operators

Sobel's Operator

$$\longrightarrow \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 1 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & -1 \end{bmatrix}$$

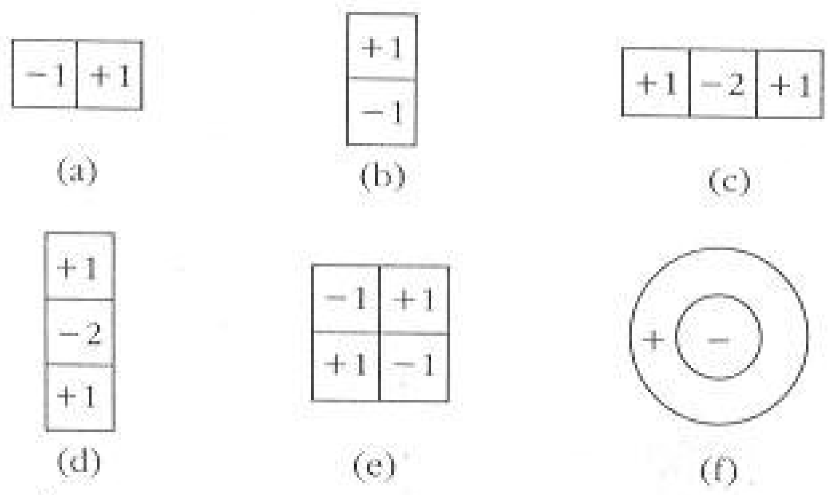
$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Smooth Differentiate

$$\longrightarrow \begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

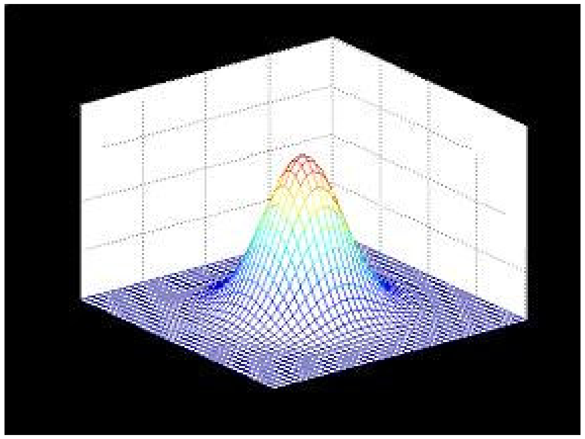
$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The visual pathway



Gaussian Filter

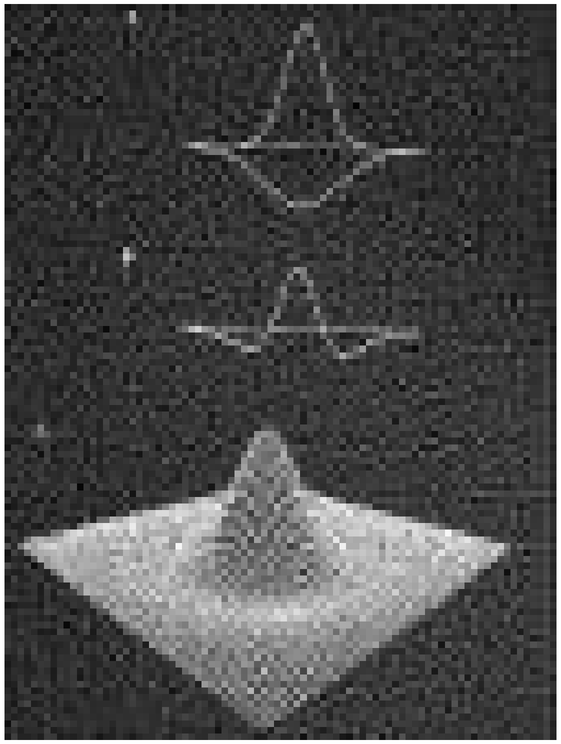
$$G_\sigma(x, y) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{\left(x^2 + y^2\right)}{2\sigma^2} \right)$$



$$H(i, j) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{\left((i-k-1)^2 + (j-k-1)^2\right)}{2\sigma^2} \right)$$

where $H(i, j)$ is $(2k+1)\times(2k+1)$ array

Difference of Gausians

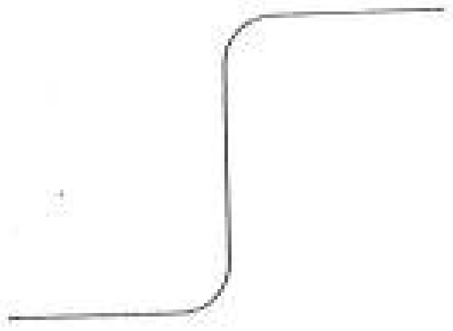


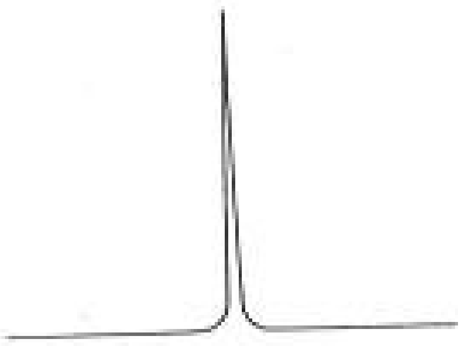
$$f(x,y) = c_1 e^{\left(\frac{-(x^2+y^2)}{\sigma_1^2}\right)} - c_2 e^{\left(\frac{-(x^2+y^2)}{\sigma_2^2}\right)}$$

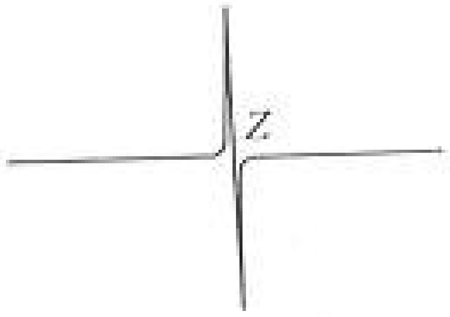
Zero Crossing

[illegible]

$$G(x,y) = e^{-\frac{x^2+y^2}{2\pi\sigma^2}}$$


(a)


(b)



(c)

LoG

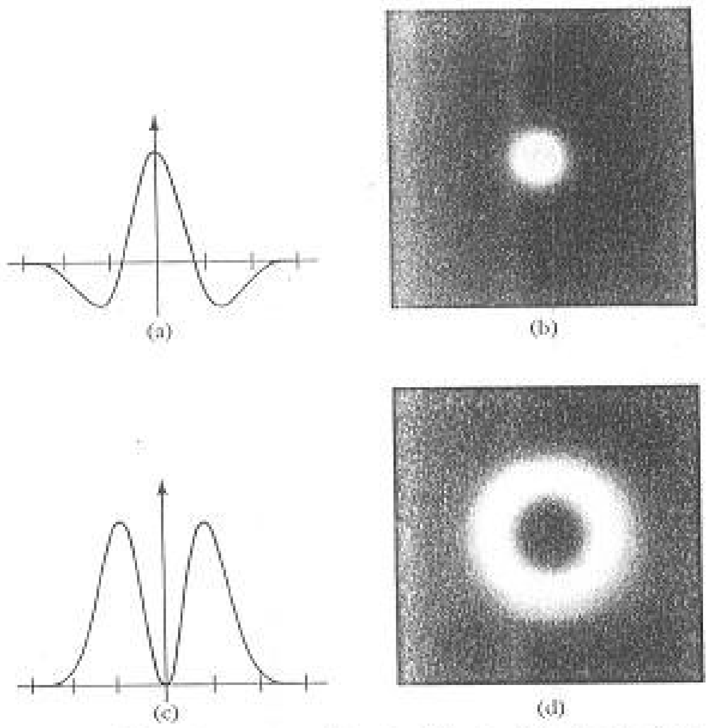
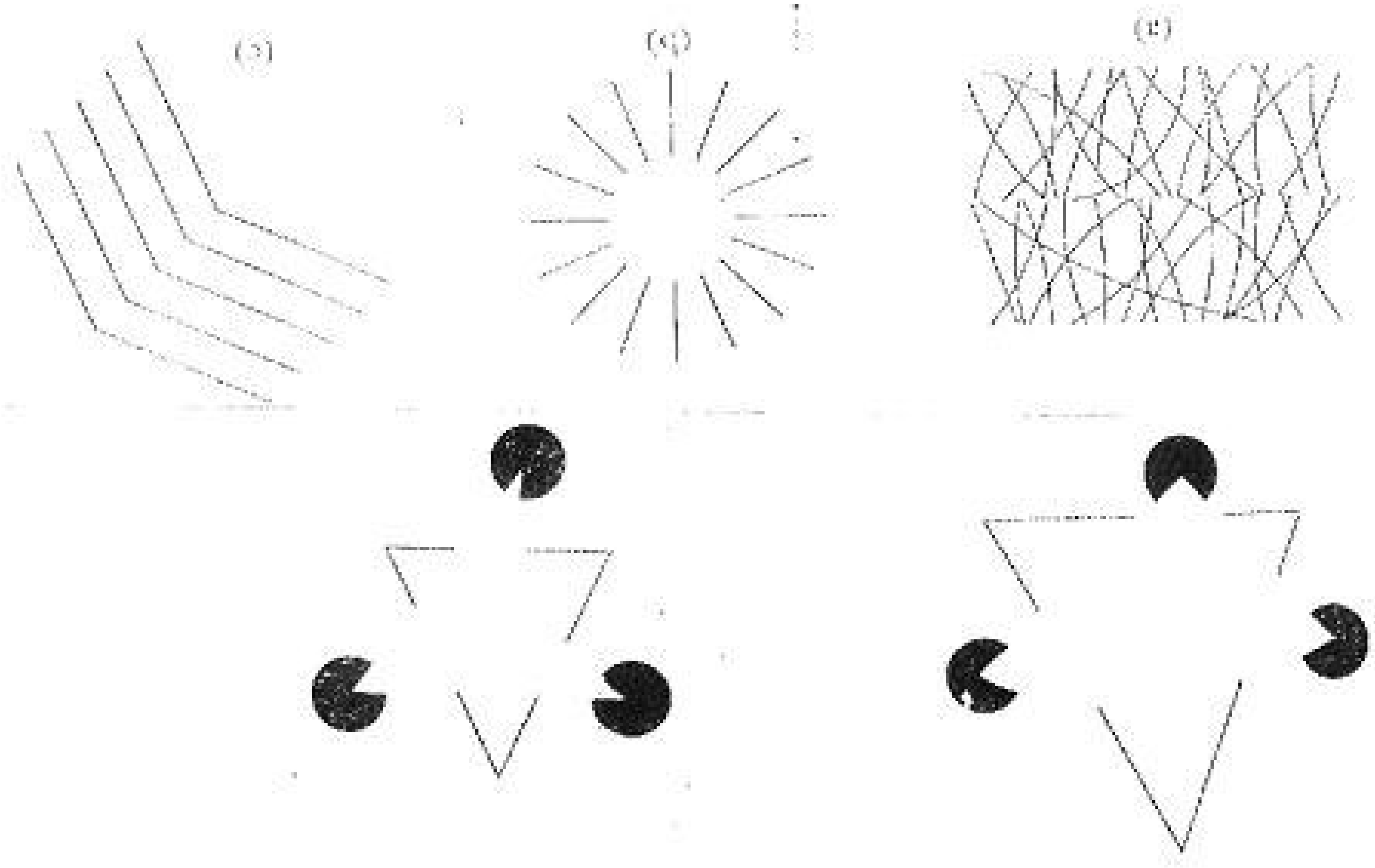


Figure 2–9. $\nabla^2 G$ is shown as a one-dimensional function (a) and in two-dimensions (b) using intensity to indicate the value of the function at each point. (c) and (d) show the Fourier transforms for the one- and two-dimensional cases respectively. (Reprinted by permission from D. Marr and E. Hildreth, "Theory of edge detection," *Proc. R. Soc. Lond. B 207*, pp. 187–217.)

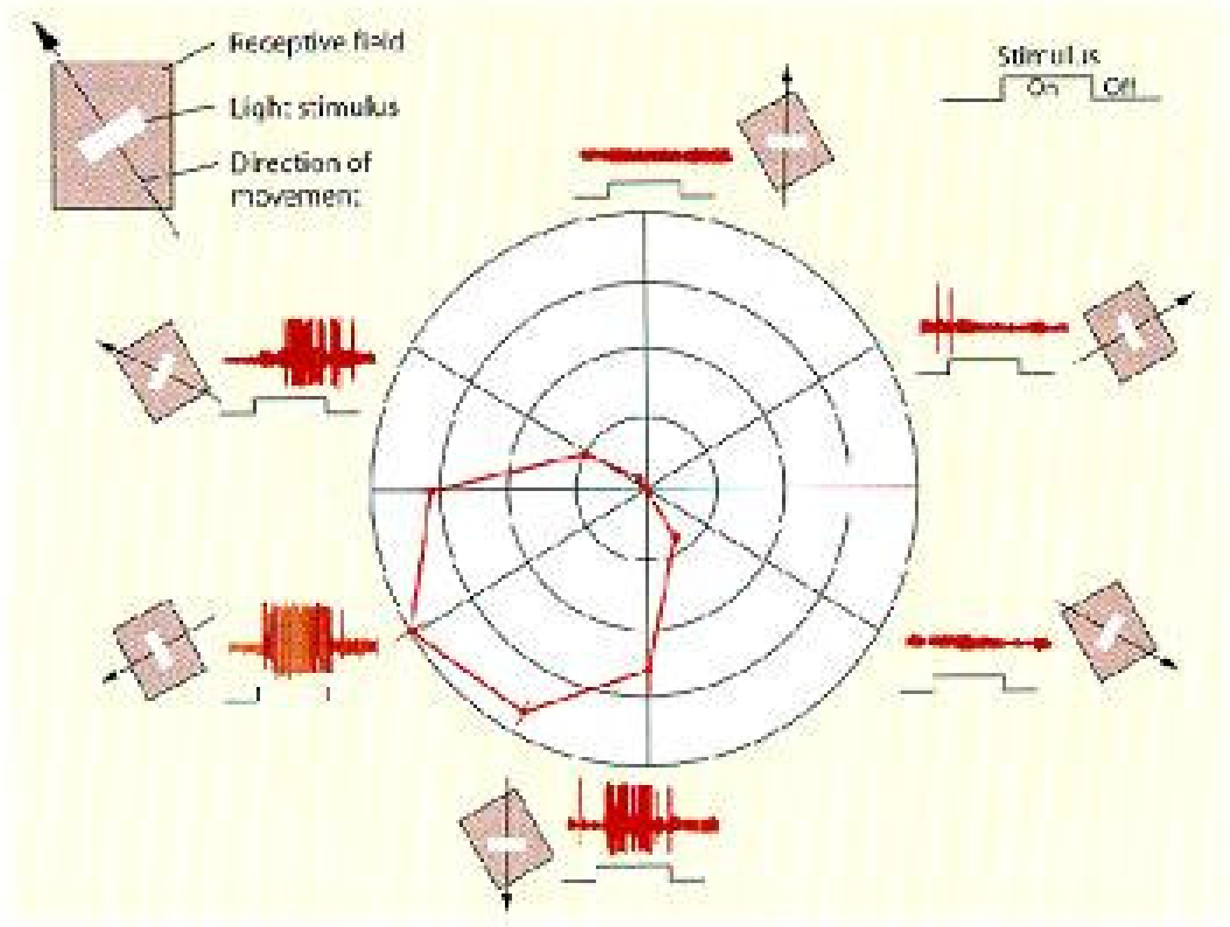
which has standard deviation σ. $\nabla^2 G$ is a circularly symmetric Mexican-hat-shaped operator whose distribution in two dimensions may be expressed in terms of the radial distance r from the origin by the formula

$$\nabla^2 G(r) = \frac{-1}{\pi\sigma^4}\left(1 - \frac{r^2}{2\sigma^2}\right)e^{\frac{-r^2}{2\sigma^2}}$$

Subjective contours

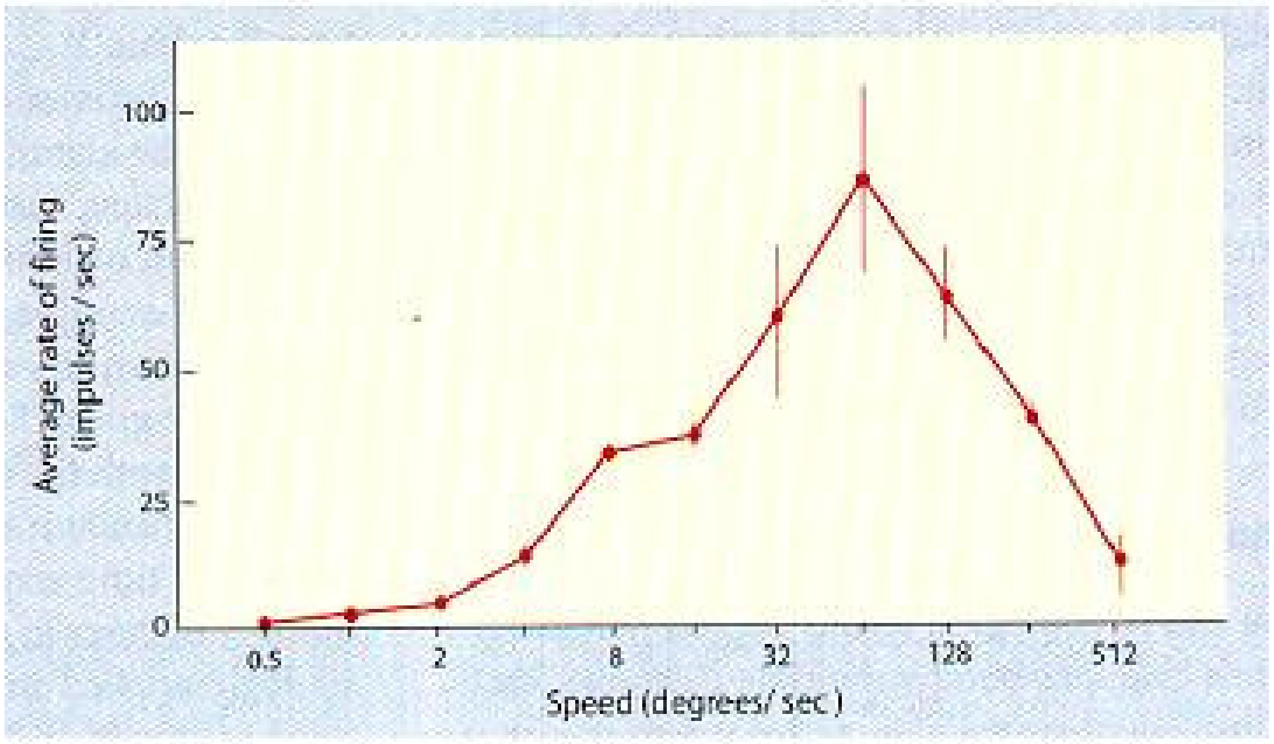


Motion selectivity

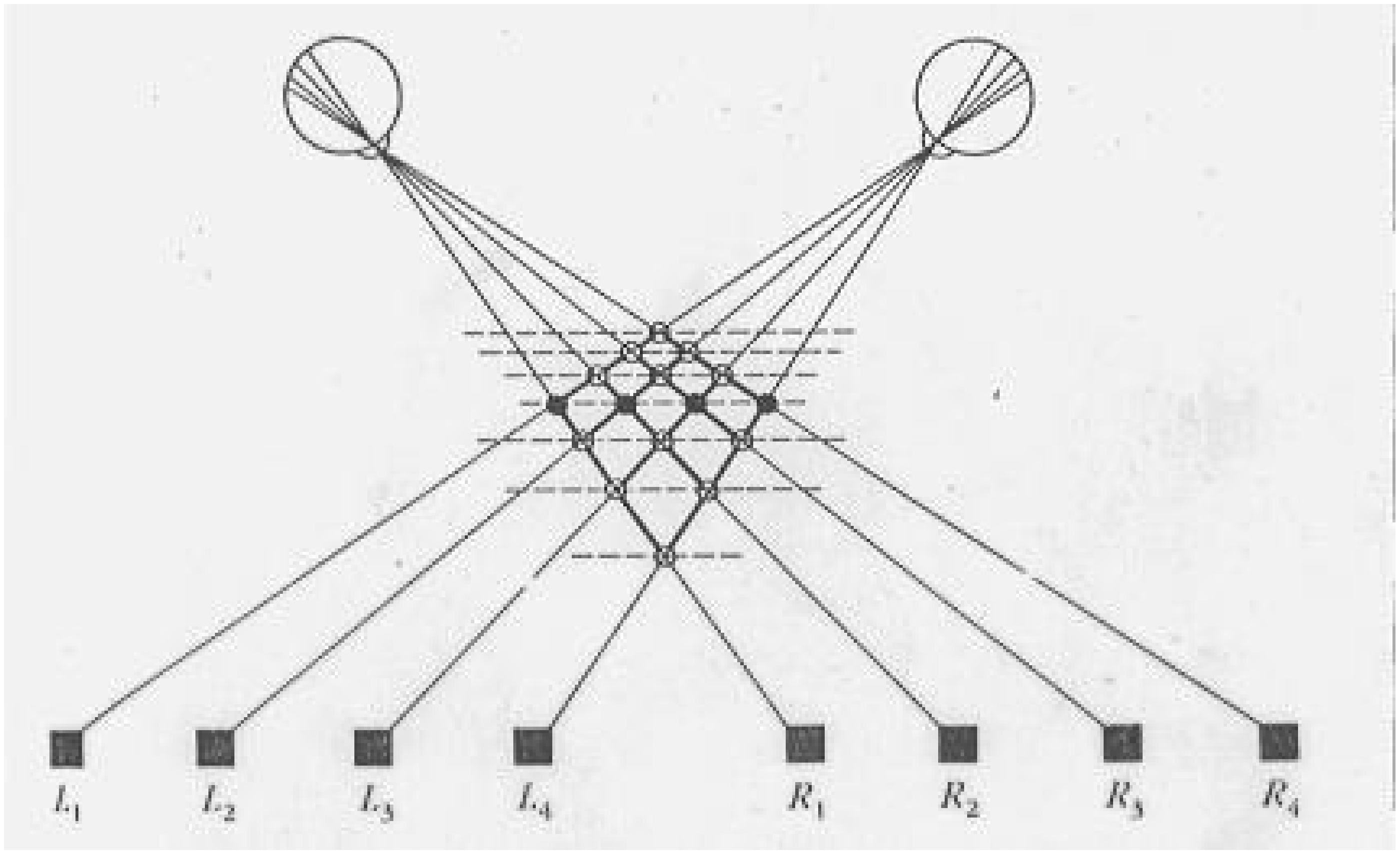


Motion selectivity

Tuning curve for preference for speed; cell in MT



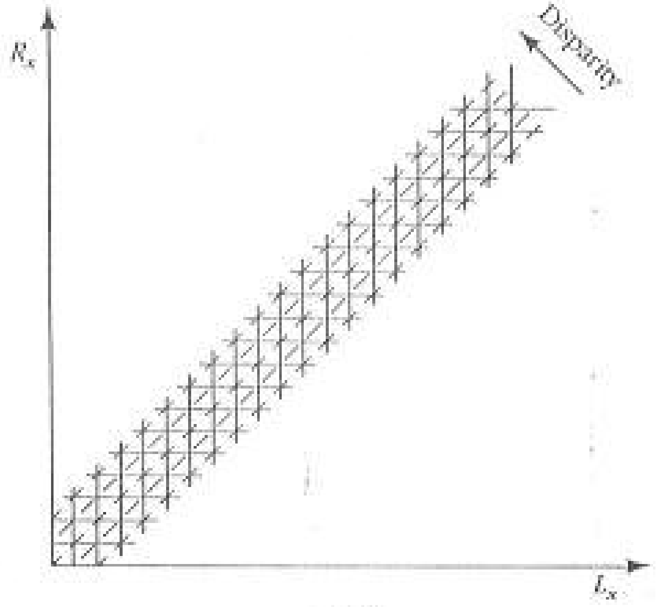
Stereo vision



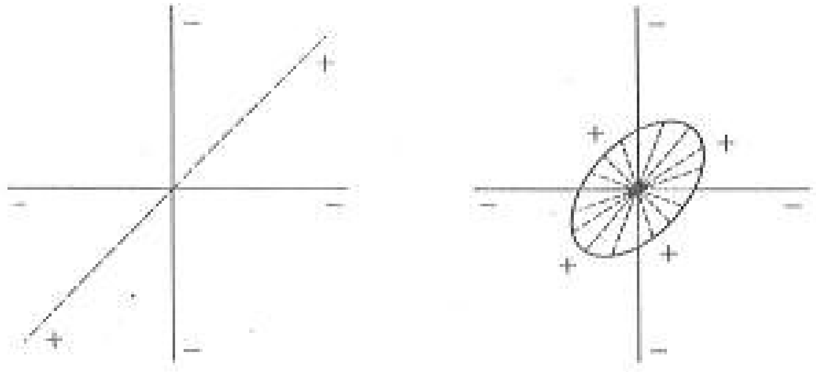
Stereo – Marr Poggio

$$C^{t+1}_{x,y;d} = \sigma\left\{ \sum_{x',y';d' \in S(x,y;d)} C^{t}_{x',y';d'} - \varepsilon \sum_{x',y',d' \in O(x,y;d)} C^{t}_{x',y';d'} + C^{0}_{x,y;d} \right\}$$

Stereo – Marr Poggio



(a)



(b) (c)