Lecture 18:Feature Extraction

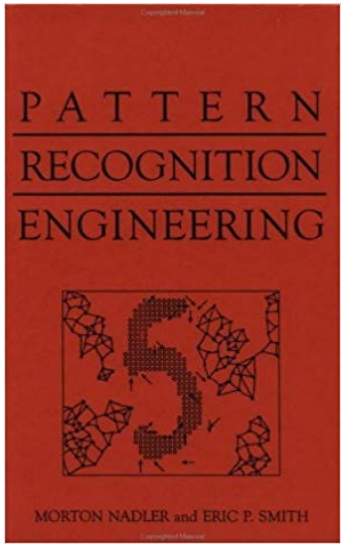
Andreas Wichert
Department of Computer Science and Engineering
Técnico Lisboa

Deep Learning Approach

- Deep Learning machines usually work better than traditional ML tools because they also **learn the feature** extraction part
- Deep learning schemes also **optimize the features** that are extracted which largely explains why they perform good
- We can extract the **wrong features**, deep learning extracts the correct one from the sample!

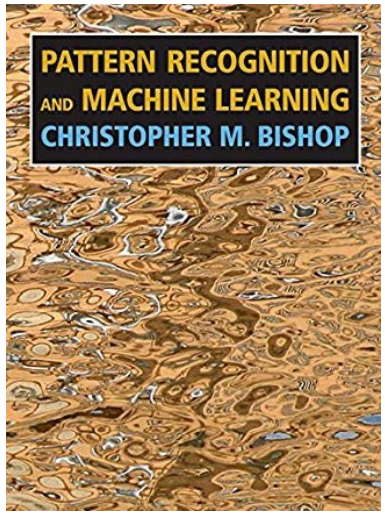
- But: We need **huge labeled data** set (huge sample)
  - Sample, the bigger, the better!
  - **Where do we get it/them?**

Times changed…



1993

A lot about feature extraction and preprocessing



2006

Feature extraction and preprocessing is mentioned only on page 2,3 and 606 shortly

... the United States' Apollo 11 was the first manned mission to land on the Moon, on 20 July 1969

Why Feature Extraction?

- Reduce the dimension of the training patterns
- Reduce **the size** required training set

- Good features are **linear separable**
- Good features allow **unsupervised learning**
  - Do you know why?

- But: We do not know a lot about feature extraction..
  - It is both difficult and expensive
  - That is why we ignore it

Time Signals: Noise reduction

- It is difficult to identify the frequency components by looking at the original signal
- Converting to the frequency domain
- If dimension reduction, store only a fraction of frequencies (with high amplitude)

- If noise reduction
  - (remove high frequencies, fast change, smoothing)
  - (remove low frequencies, slow change, remove global trends)
  - Inverse discrete Fourier transform

Discrete Fourier Transform DFT is a Linear Transform

- Operates on discrete complex-valued function
  - Given a function *a* :

  - The discrete Fourier transform produces a function *A* :

$$a : [0,1,\ldots,N-1] \rightarrow C$$

$$A : [0,1,\ldots,N-1] \rightarrow C$$

$$A(x) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} a(k) \cdot e^{2\pi i \cdot \frac{kx}{N}}$$

- DFT can be seen as a linear transform talking the column vector ***a*** to a column vector ***A***

$$\begin{pmatrix} A(0) \\ A(1) \\ \vdots \\ A(N-1) \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{N}} \cdot e^{2\pi i \frac{0 \cdot 0}{N}} & \frac{1}{\sqrt{N}} \cdot e^{2\pi i \frac{0 \cdot 1}{N}} & \cdots & \frac{1}{\sqrt{N}} \cdot e^{2\pi i \frac{0 \cdot N-1}{N}} \\ \frac{1}{\sqrt{N}} \cdot e^{2\pi i \frac{0 \cdot 1}{N}} & \frac{1}{\sqrt{N}} \cdot e^{2\pi i \frac{1 \cdot 1}{N}} & \cdots & \frac{1}{\sqrt{N}} \cdot e^{2\pi i \frac{(N-1) \cdot 1}{N}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\sqrt{N}} \cdot e^{2\pi i \frac{0 \cdot (N-1)}{N}} & \frac{1}{\sqrt{N}} \cdot e^{2\pi i \frac{1 \cdot (N-1)}{N}} & \cdots & \frac{1}{\sqrt{N}} \cdot e^{2\pi i \frac{(N-1) \cdot (N-1)}{N}} \end{pmatrix} \cdot \begin{pmatrix} a(0) \\ a(1) \\ \vdots \\ a(N-1) \end{pmatrix}$$

- Simplification

$$\begin{pmatrix} A(0) \\ A(1) \\ \vdots \\ A(N-1) \end{pmatrix} = \frac{1}{\sqrt{N}} \cdot \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{2\pi i \frac{1 \cdot 1}{N}} & \cdots & e^{2\pi i \frac{(N-1) \cdot 1}{N}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{2\pi i \frac{1 \cdot (N-1)}{N}} & \cdots & e^{2\pi i \frac{(N-1) \cdot (N-1)}{N}} \end{pmatrix} \cdot \begin{pmatrix} a(0) \\ a(1) \\ \vdots \\ a(N-1) \end{pmatrix}$$

- Example, N=4

$$\begin{pmatrix} A(0) \\ A(1) \\ \vdots \\ A(N-1) \end{pmatrix} = \frac{1}{2} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \cdot \begin{pmatrix} a(0) \\ a(1) \\ \vdots \\ a(N-1) \end{pmatrix}$$

- Let $a : [0,1,\ldots,N-1] \rightarrow C$ be a **periodic function**

$$a(x) = e^{-2\pi i \frac{ux}{N}}$$

$$a(x) = \cos(2\pi \frac{ux}{N}) + i \cdot \sin(2\pi \frac{ux}{N})$$

$$e^{iu} = \cos(u) + i \cdot \sin(u)$$

- A complex root of unity is a complex number
- There are exactly *n*th roots of unity: $\omega^N = 1$

- We define $e^{2\pi i \frac{k}{N}} \quad for \quad k = 0,1,\ldots,N-1$

$$\omega_N = e^{2\pi i \frac{1}{N}}$$

$$e^{iu} = \cos(u) + i \cdot \sin(u)$$

$$\begin{pmatrix} A(0) \\ A(1) \\ \vdots \\ A(N-1) \end{pmatrix} = \frac{1}{\sqrt{N}} \cdot \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{2\pi i \frac{1 \cdot 1}{N}} & \cdots & e^{2\pi i \frac{(N-1) \cdot 1}{N}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{2\pi i \frac{1 \cdot (N-1)}{N}} & \cdots & e^{2\pi i \frac{(N-1) \cdot (N-1)}{N}} \end{pmatrix} \cdot \begin{pmatrix} a(0) \\ a(1) \\ \vdots \\ a(N-1) \end{pmatrix}$$

$$\begin{pmatrix} A(0) \\ A(1) \\ \vdots \\ A(N-1) \end{pmatrix} = \frac{1}{\sqrt{N}} \cdot \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega_N^{1 \cdot 1} & \cdots & \omega_N^{(N-1) \cdot 1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{1(N-1)} & \cdots & \omega_N^{(N-1)(N-1)} \end{pmatrix} \cdot \begin{pmatrix} a(0) \\ a(1) \\ \vdots \\ a(N-1) \end{pmatrix}$$

Remarks

$$\begin{pmatrix} A(0) \\ A(1) \\ \vdots \\ A(N-1) \end{pmatrix} = \frac{1}{\sqrt{N}} \cdot \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega_N^{1\cdot 1} & \cdots & \omega_N^{(N-1)\cdot 1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{1(N-1)} & \cdots & \omega_N^{(N-1)(N-1)} \end{pmatrix} \cdot \begin{pmatrix} a(0) \\ a(1) \\ \vdots \\ a(N-1) \end{pmatrix}$$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{pmatrix} = \frac{1}{\sqrt{N}} \cdot \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega_N^{1\cdot 1} & \cdots & \omega_N^{(N-1)\cdot 1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{1(N-1)} & \cdots & \omega_N^{(N-1)(N-1)} \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix}$$

- Input vector of complex numbers of length *N*

$$x_0, x_2, \ldots\ldots x_{N-1}$$

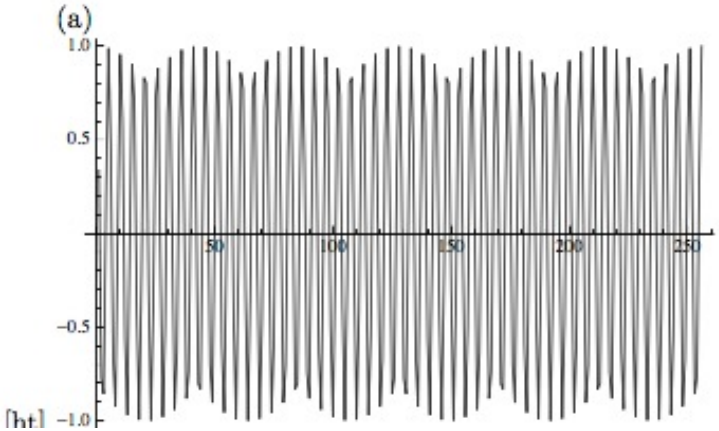
$$y_0, y_2, \ldots\ldots y_{N-1}$$

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{-\frac{2\pi i}{N} kj} \qquad k \in \{0,1,\ldots,N-1\}$$

$$inverse:$$

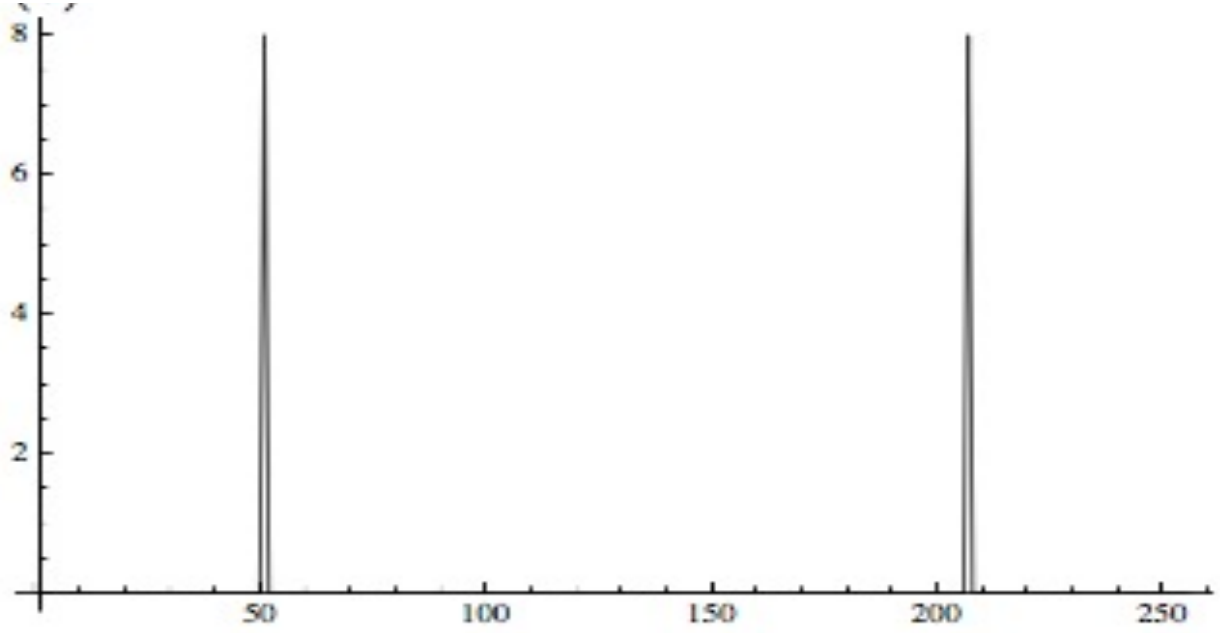
$$x_k = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k e^{\frac{2\pi i}{N} kj} \qquad j \in \{0,1,\ldots,N-1\}$$

Example



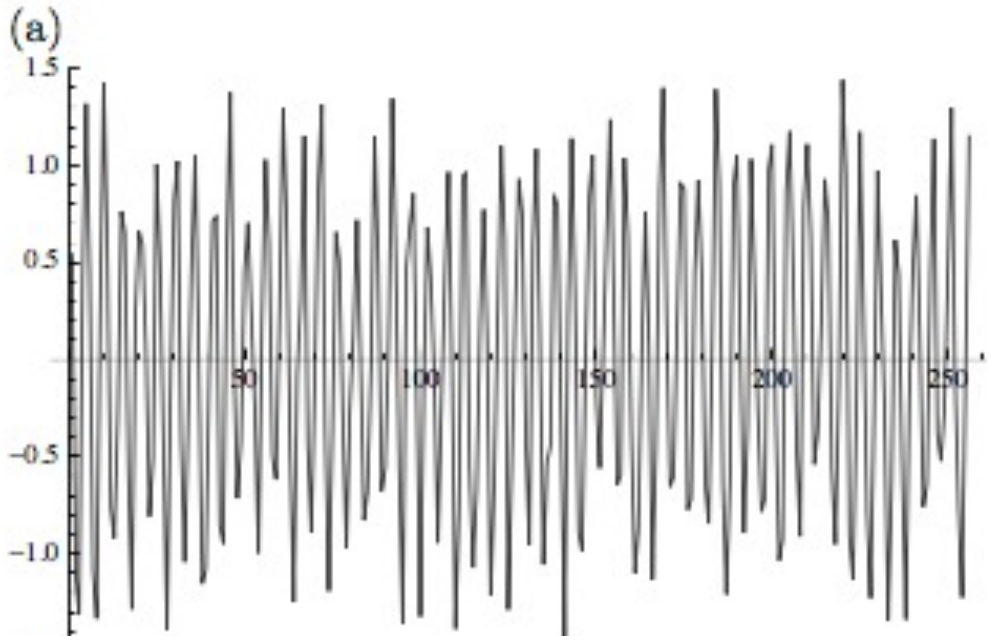
We generate a list with $256 = 2^8$ elements containing a periodic signal α_t

$$\alpha_t = cos\left(\frac{50 \cdot t \cdot 2 \cdot \cdot \pi}{256}\right),$$



The discrete Fourier transform ω_f of the real valued signal αt is symmetric. It shows a strong peak at *50 + 1* and a symmetric peak at *256 − 50 + 1* representing the frequency component of the signal

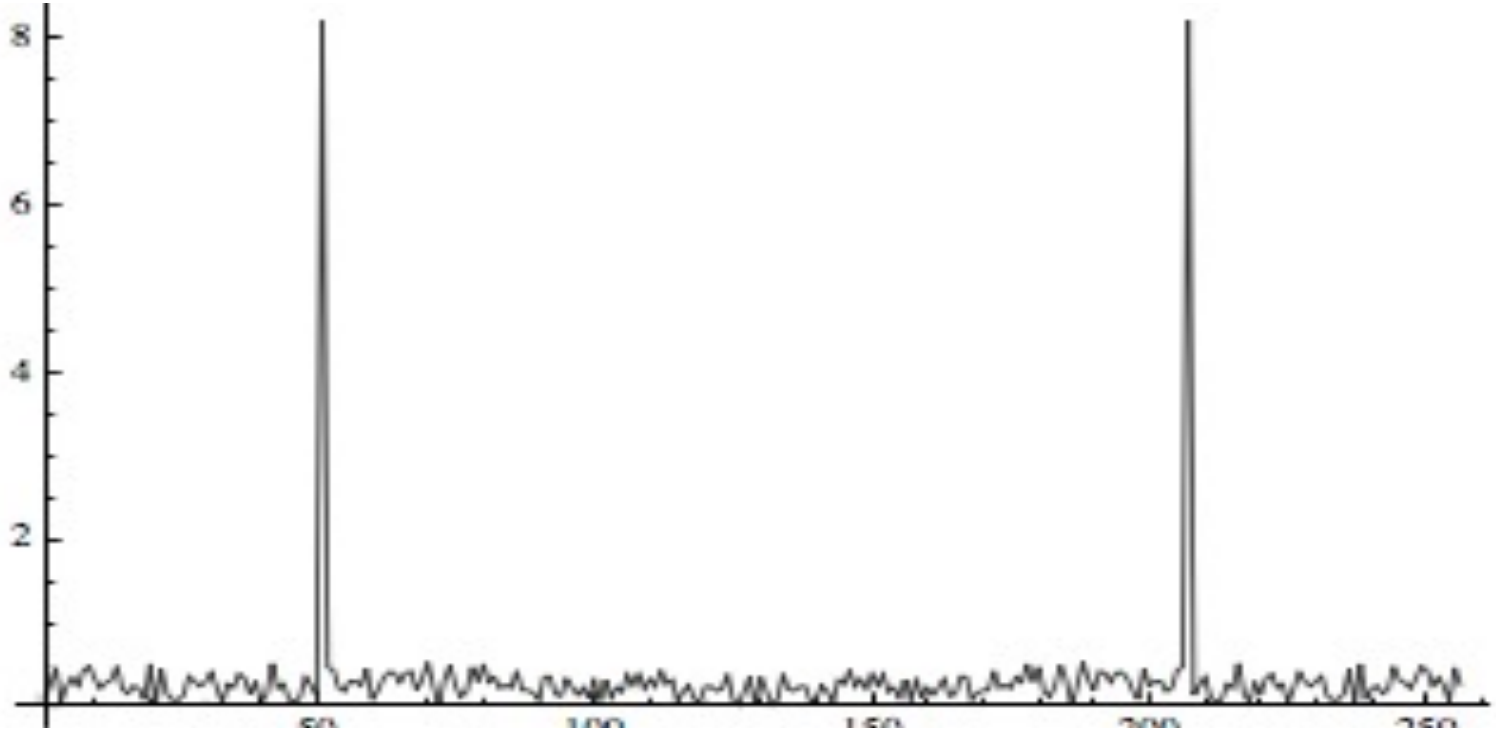
- The frequency spectrum of a real valued signal is always symmetric. The top plot illustrates this point
- However, since the symmetric part is exactly a mirror image of the first part
- This symmetric second part is usually not shown



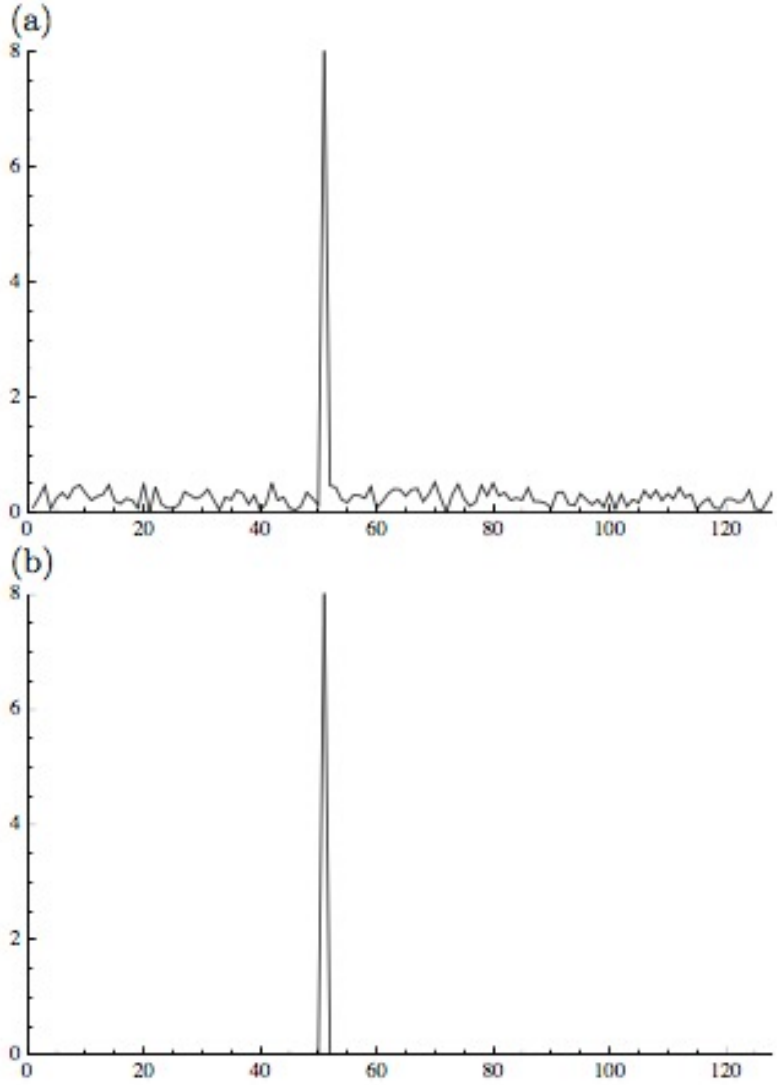
$$\alpha_t^* = cos\left(\frac{50 \cdot t \cdot 2 \cdot \cdot \pi}{256}\right) + noise.$$

- We add to the periodic signal αt Gaussian random noise from the interval [*−0.5, 0.5*].
- The represented data looks random

The frequency component

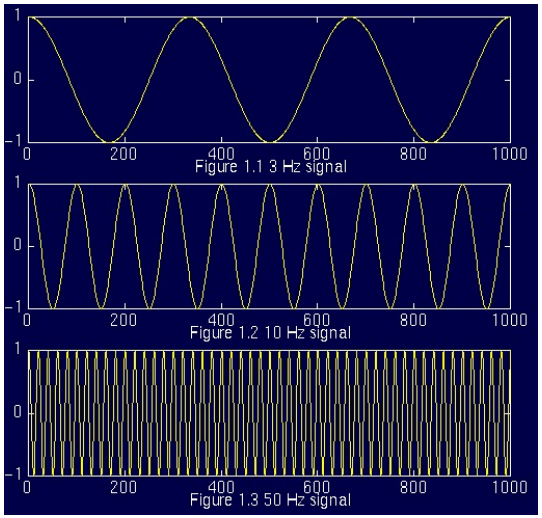


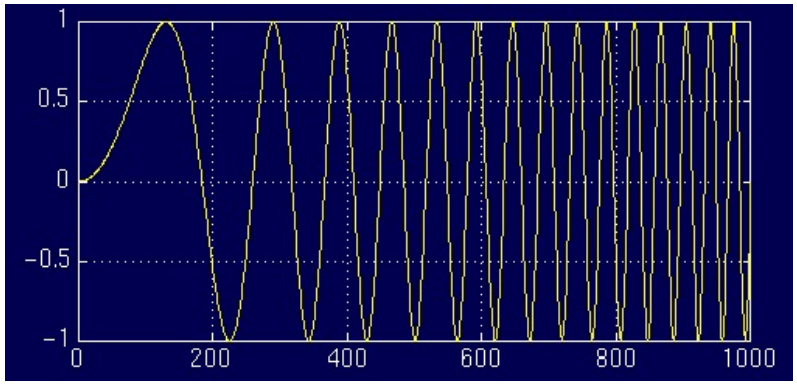
- A filter that reduces Gaussian noise based on DFT removes frequencies with low amplitude of ω_f and performs the inverse discrete Fourier transform
- For dimension reduction of the signal, only a fraction of frequencies with high amplitude are represented.


(a)
8
6
4
2
0
0
20
40
60
80
100
120
(b)
8
6
4
2
0
0
20
40
60
80
100
120

Stationary Signal

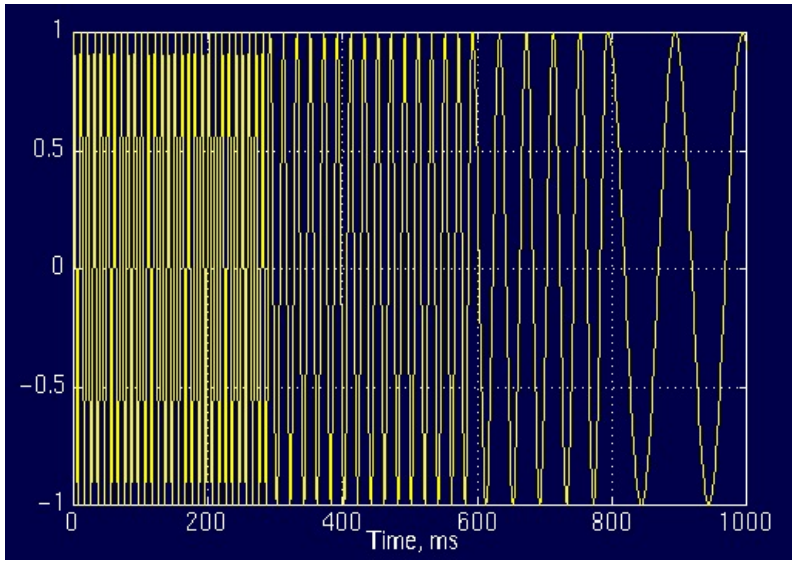
- Signals whose frequency content do not change in time are called stationary signals
- Non stationary signal, frequency content does change over time

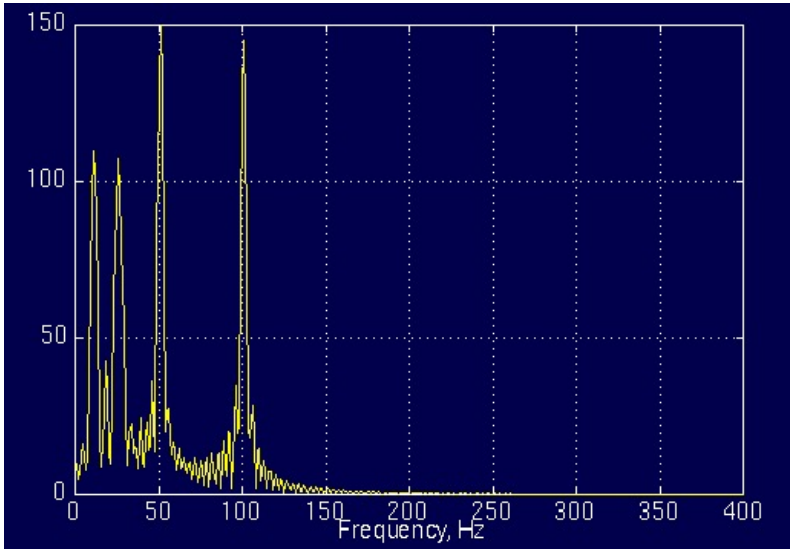




- At what times (or time intervals), do these frequency components occur?

- FT gives the spectral content of the signal, but it gives no information regarding where in time those spectral components appear!



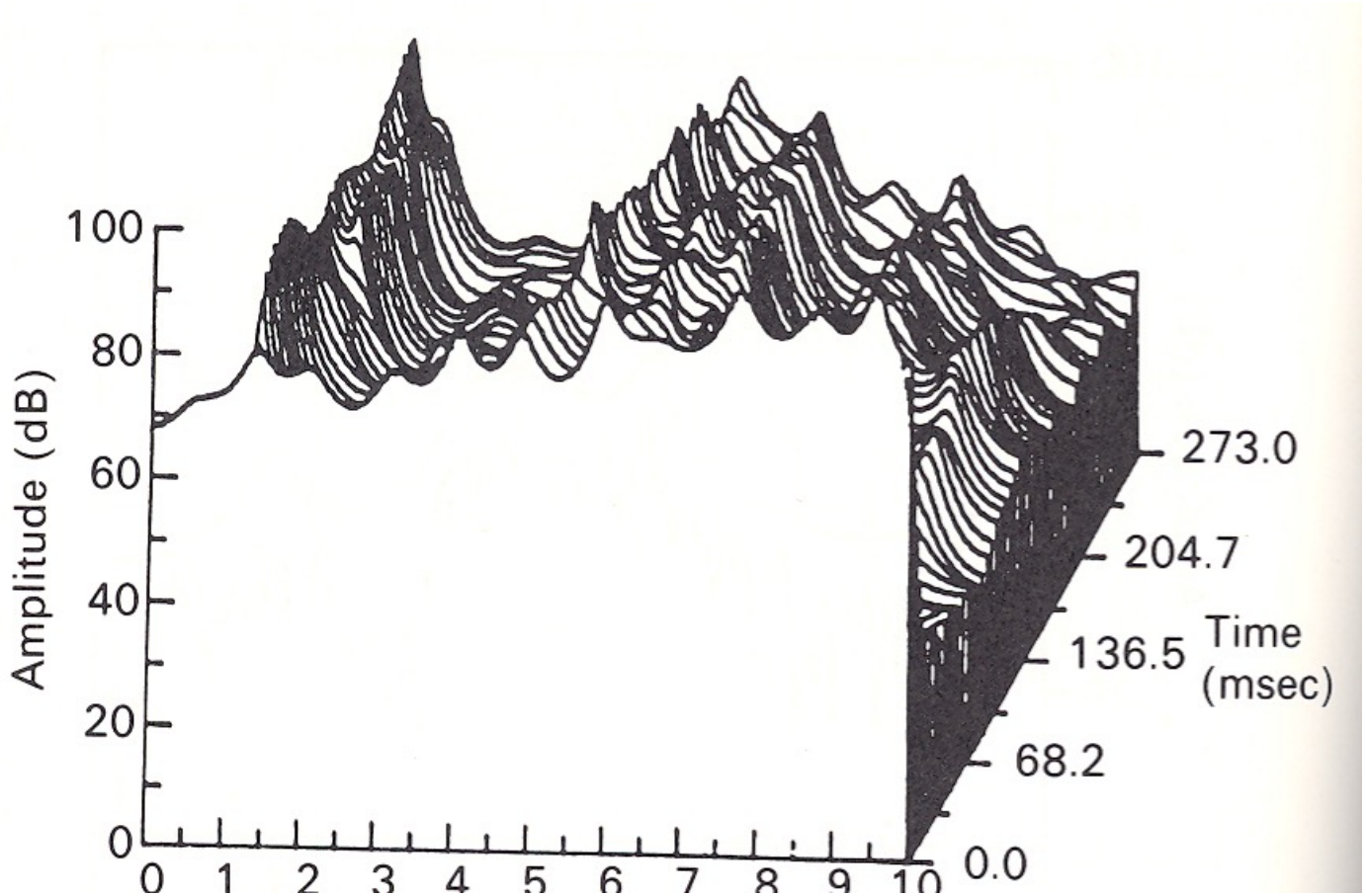


Shot Term Fourier Transform (STFT)

- If this region where the signal can be assumed to be stationary small...
  - we look at that signal from narrow windows, narrow enough that the portion of the signal seen from these windows are indeed stationary
  - This approach of researchers ended up with a revised version of the Fourier transform, so-called : The Short Time Fourier Transform (STFT)

- There is only a minor difference between STFT and FT
- In STFT, the signal is divided into small enough segments, where these segments (portions) of the signal can be assumed to be stationary
- For this purpose, a window function "*w*" is chosen
- The width of this window must be equal to the segment of the signal where its stationarity is valid...

$$w(t) = e^{-a \cdot t^2 / 2}$$


Amplitude (dB)
100
80
60
40
20
0
0 1 2 3 4 5 6 7 8 9 10
273.0
204.7
136.5
68.2
0.0
Time
(msec)

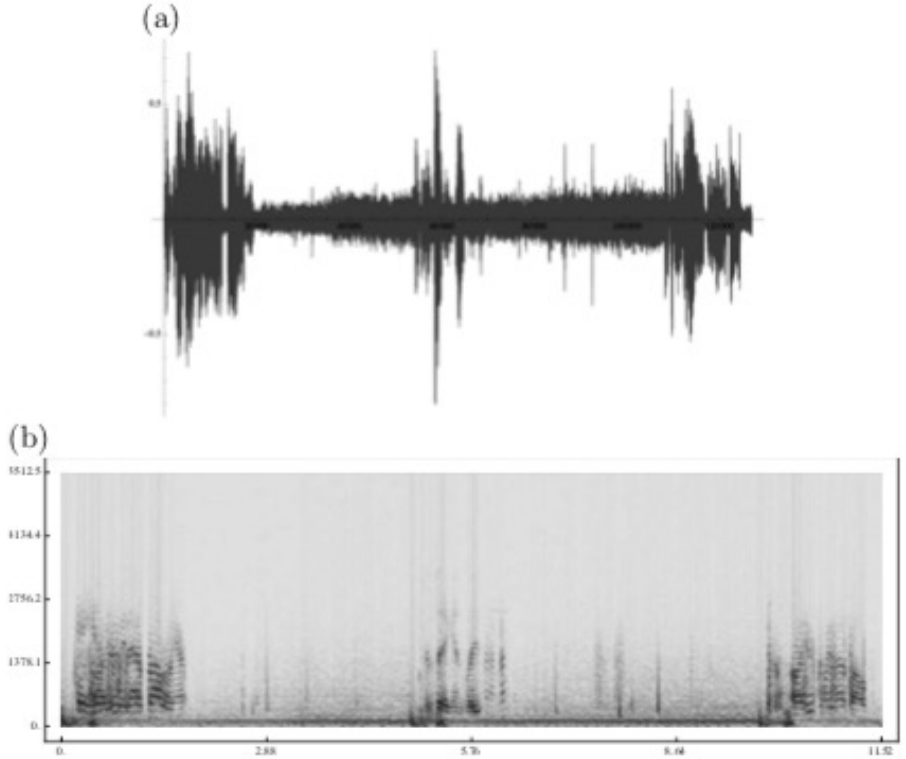
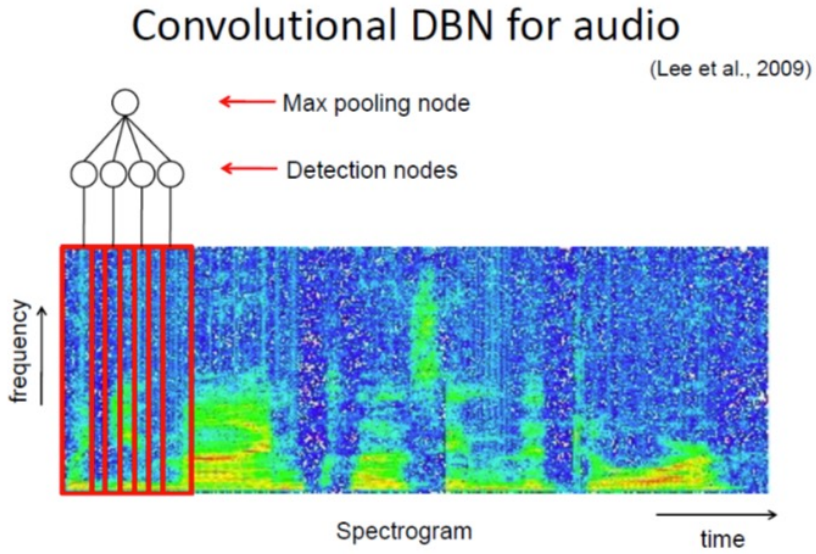


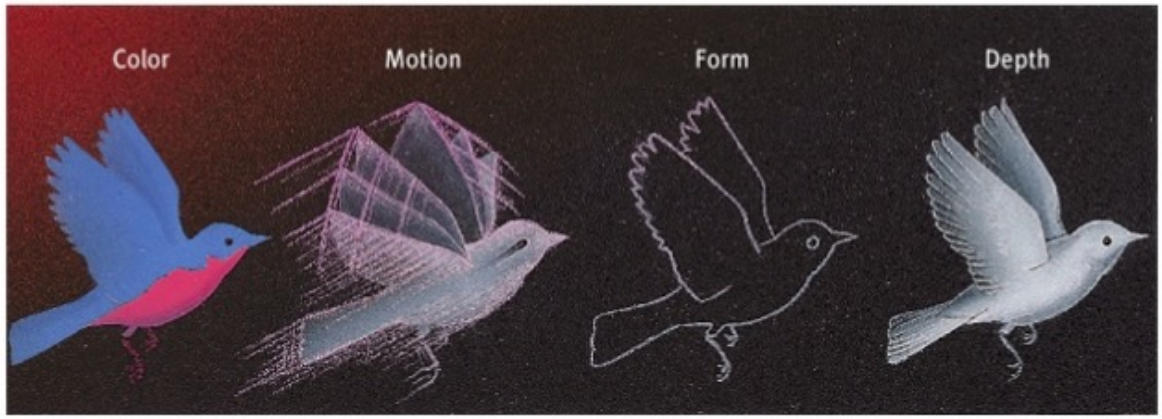
Fig. 3.15 (a) The sound signal "Houston, we have a problem", spoken during the Apollo 13 mission. (b) Corresponding spectral representation.



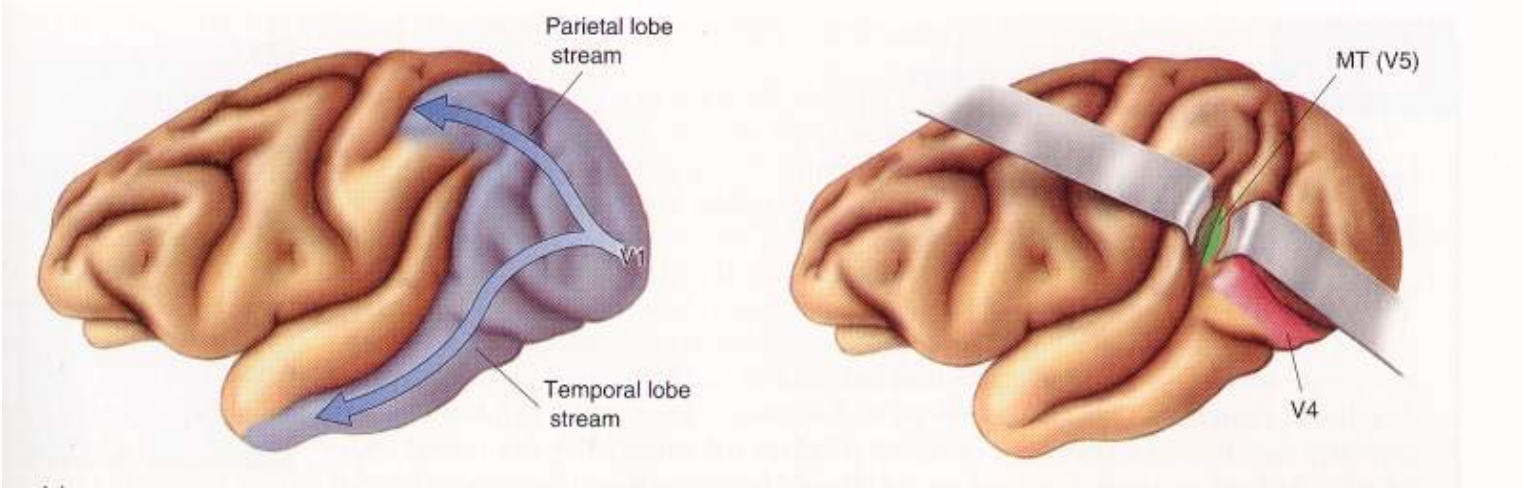
What Does the Brain Tell Us?

- Mirrors the way the biological sense organs describe the world

- In Vision independent processing of



Dorsal (“Where”) and Ventral (“What”) Visual Streams in Monkey

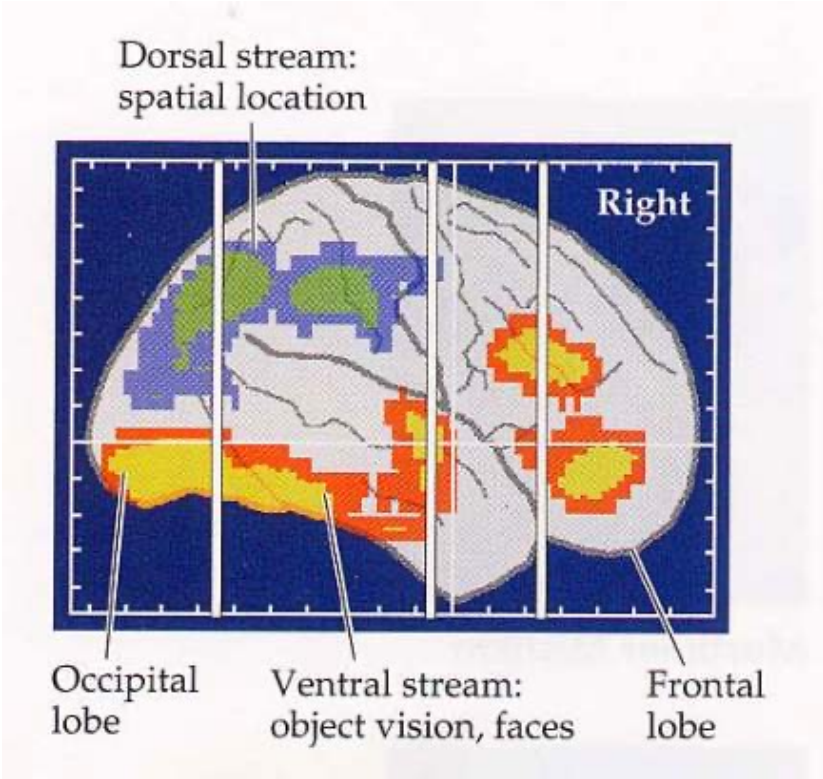


Parietal (Dorsal) and Temporal (Ventral) Processing Streams

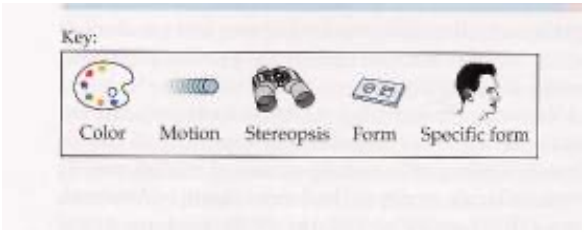
Areas MT and V4 in the Macaque Brain

Dorsal (“Where”) and Ventral (“What”) Visual Streams in Human (PET)

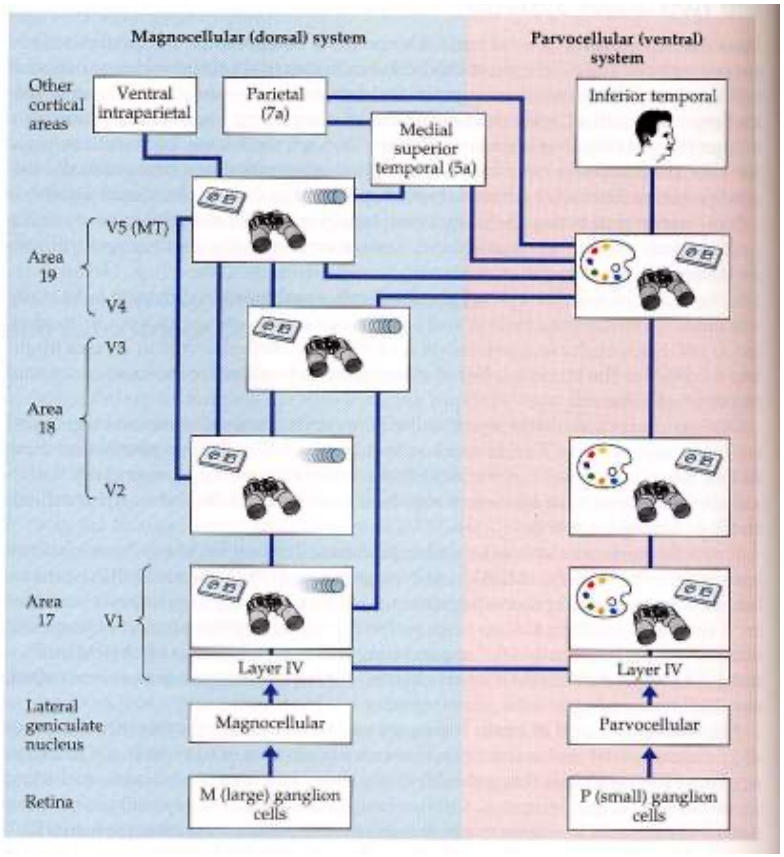
Dorsal (***where***) pathway shown in **green** and **blue** and Ventral (***what***) pathway shown in **yellow** and **red** serve different functions. (Courtesy of Leslie Ungerleider).



Retinal and Thalamic Precursors of the Dorsal and Ventral Visual Pathways

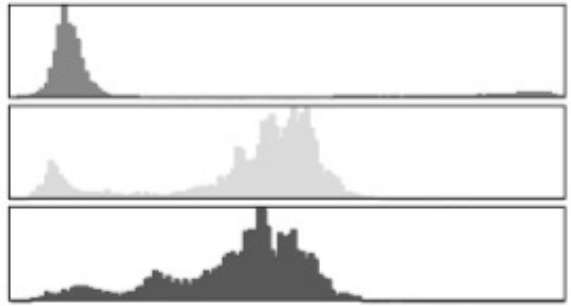


Magnocellular (dorsal) and parvocellular (ventral) pathways from the retina to the higher levels of the visual cortex are separate at the lower levels of the visual system. At higher levels they show increasing overlap.



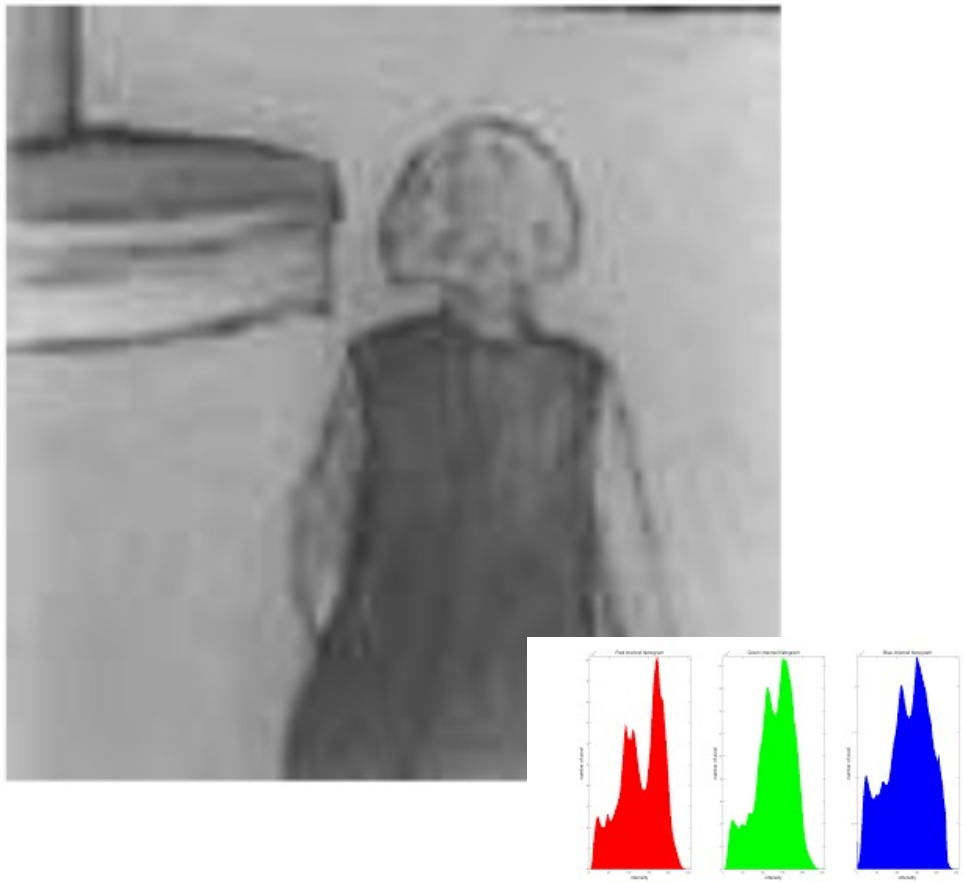
Convert the in different low dimensional modalities

Edge Detection



Three histograms that represent the RGB level distribution of the digital color RGB image

Discrete Wavelet Transform and Images



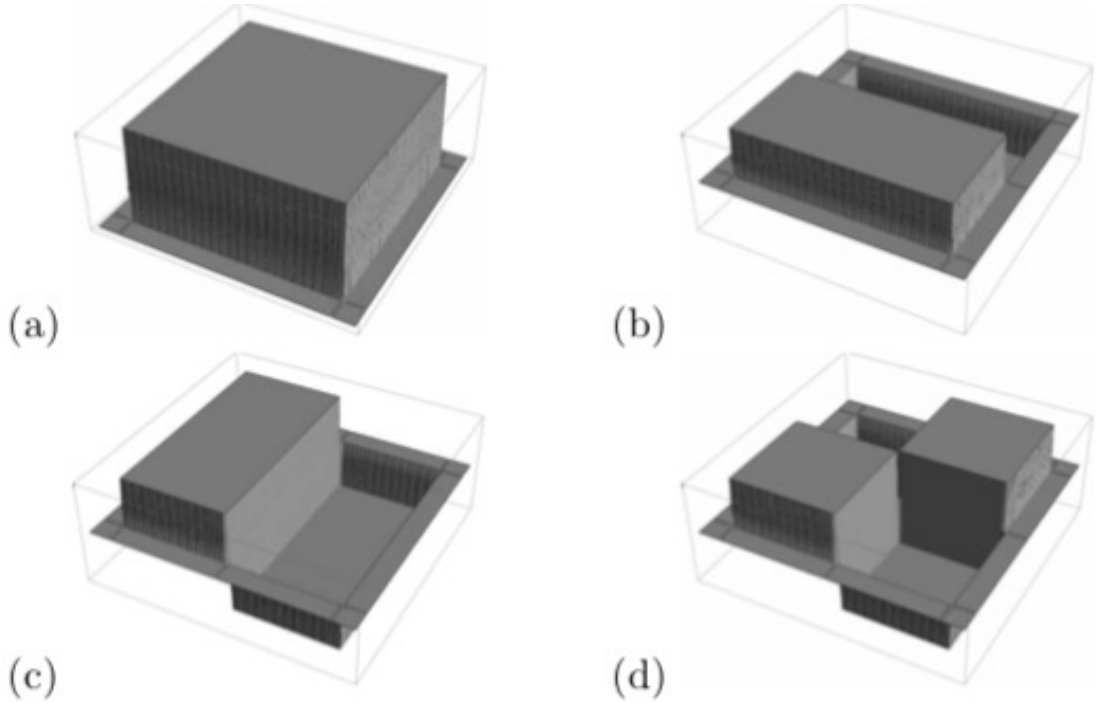


Fig. 3.25 (a) (a) The Haar scaling function $\phi(x, y)$. (b) The horizontal sensitive wavelet $\psi^H(x, y)$. (c) The vertical sensitive wavelet $\psi^V(x, y)$. (d) The diagonal sensitive wavelet $\psi^D(x, y)$.

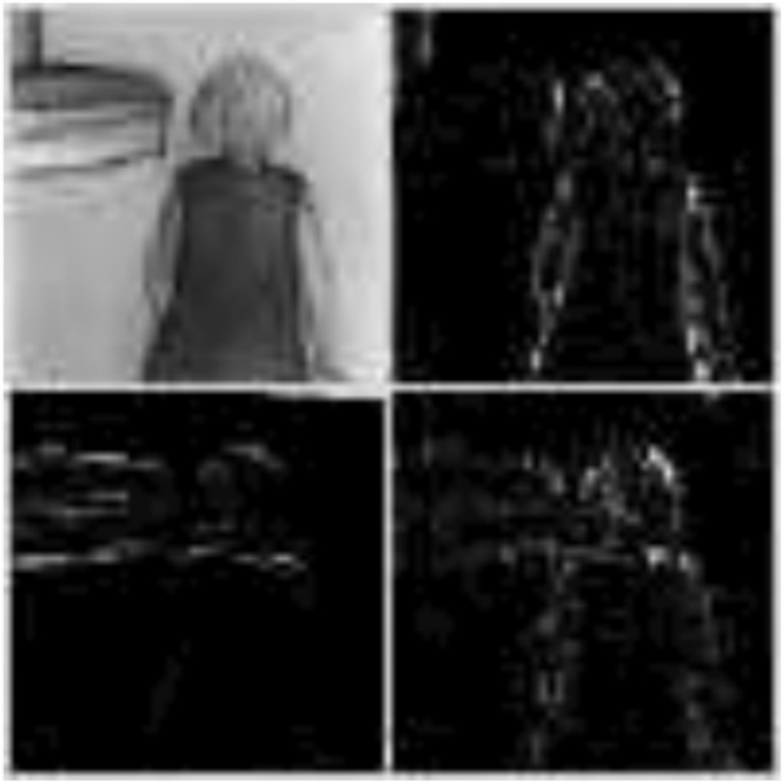
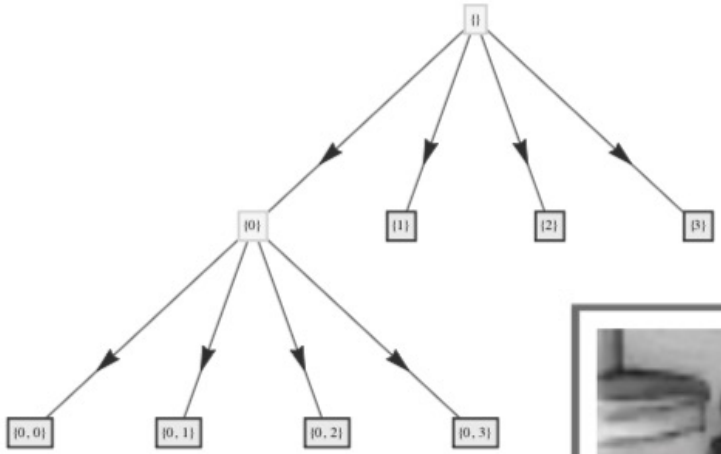
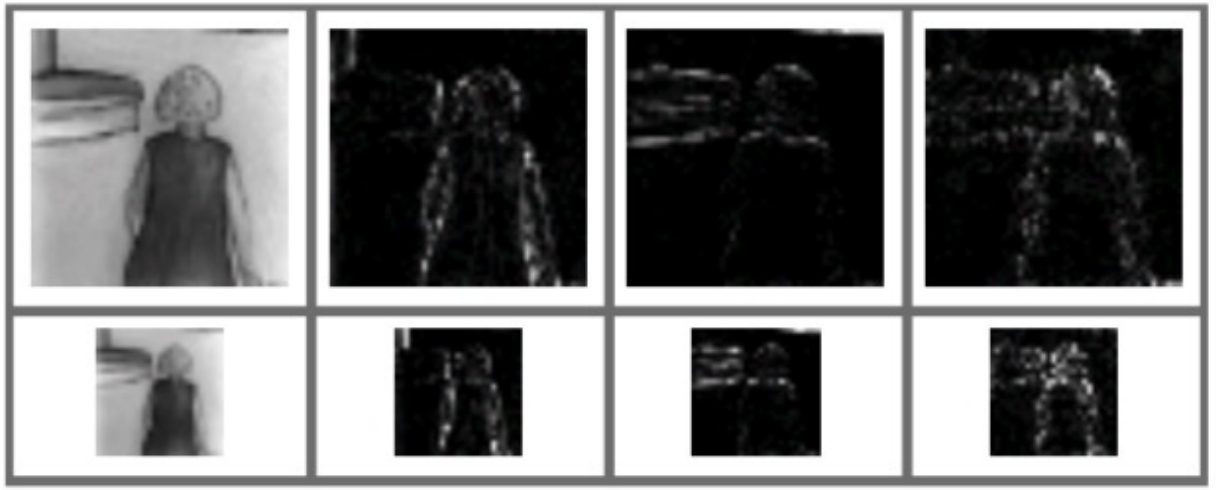


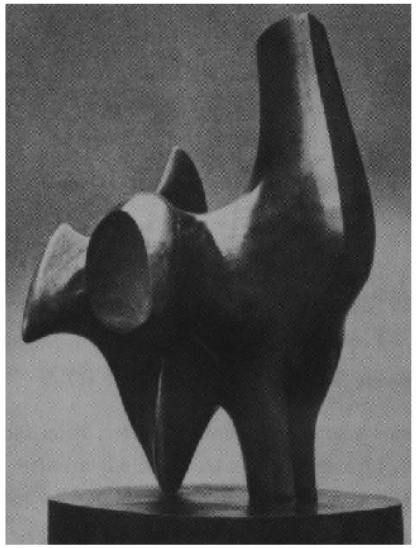
Fig. 3.27 Convolution of four filters and subsampling by 2. The filters are defined by he Haar scaling function $\phi(x, y)$ (low pass filter), the horizontal sensitive wavelet $\psi^H(x, y)$ (high pass filter), the vertical sensitive wavelet $\psi^V(x, y)$ and the diagonal sensitive wavelet $\psi^D(x, y)$.

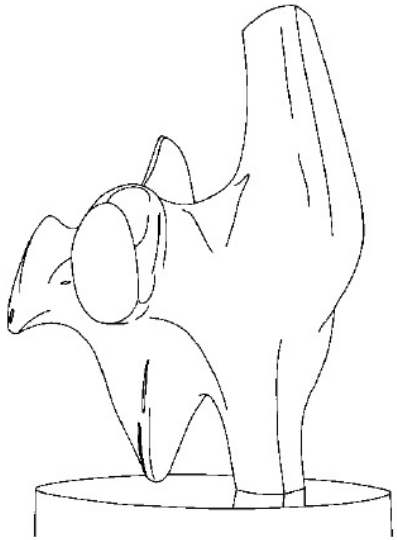
Filter bank decomposition (different resolutions)





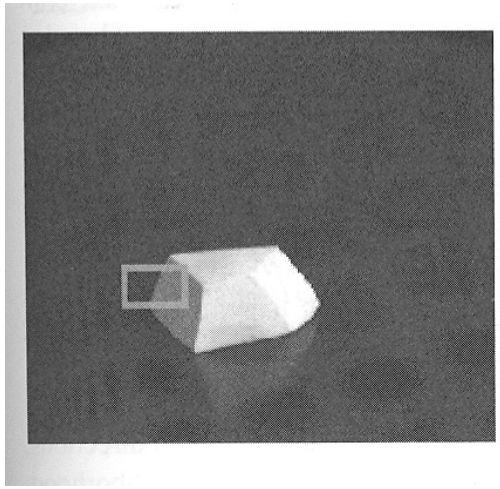
Edge detection

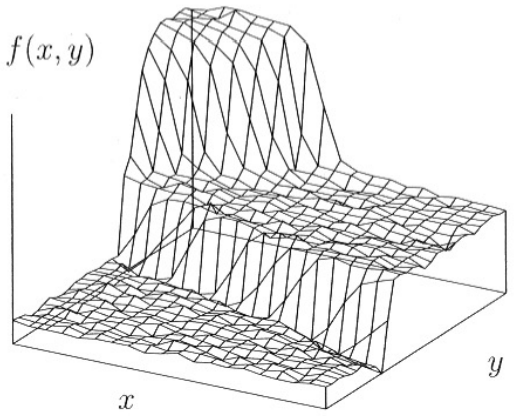




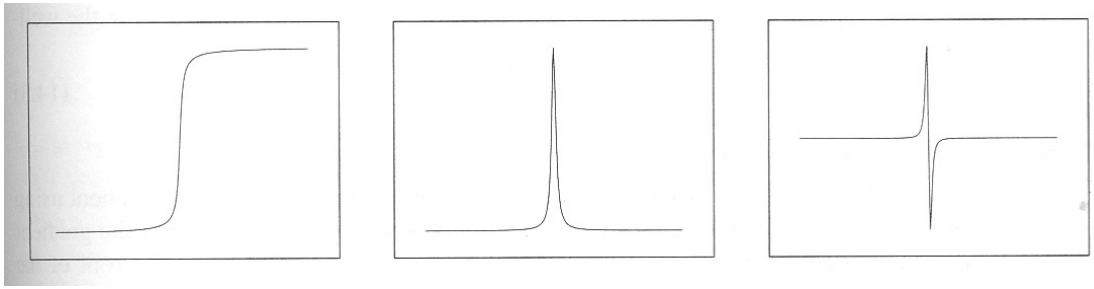
- Convert a 2D image into a set of curves
  - Extracts salient features of the scene
  - More compact than pixels

- The basic idea behind edge detection is to localize discontinuities of the intensity function in the image





- Many approaches to the edge detection are based on the idea that rapid changes and discontinuities in the gray level
- Function can be detected using maxima in the first derivative or zero-crossing in the second derivative



Edge is Where Change Occurs

- Change is measured by derivative in 1D
- Biggest change, derivative has maximum magnitude
- Or 2nd derivative is zero.

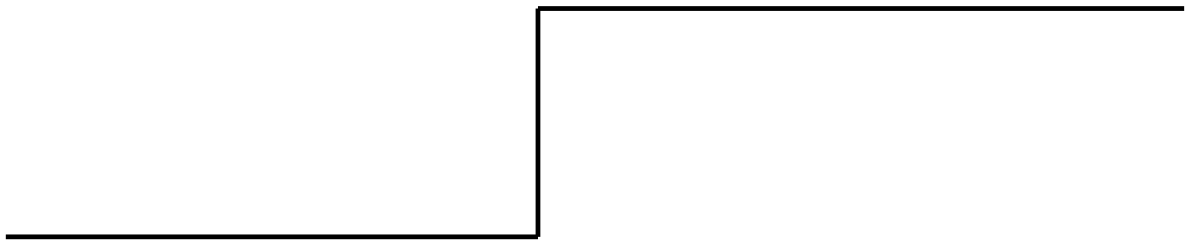
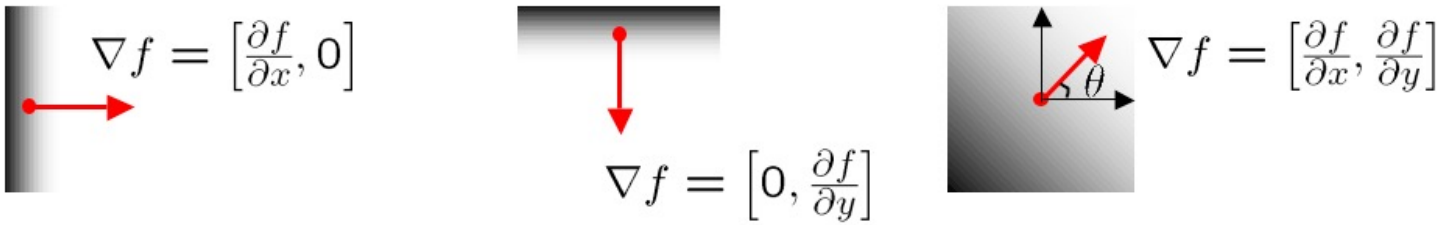


Image gradient

- The gradient of an continuous function *f(x,y)* representing an image:

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

- The gradient points in the direction of most rapid change in intensity



- The magnitude of the gradient defines the edge direction and edge strength
- Edge direction is usually defined to be orthogonal to the gradient
    - This is not always true

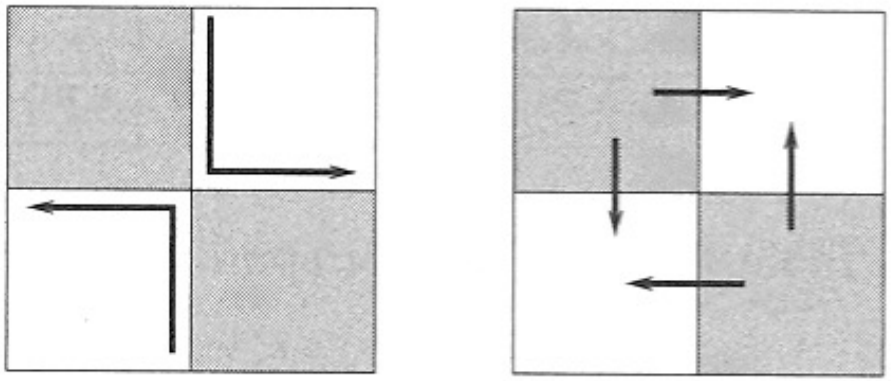
- The gradient direction is given by:

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

- The *edge strength* is given by the gradient magnitude

$$\| \nabla f \| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Edge orientation



- Definition of the edge orientation (left) and the gradient (right)

- How can we differentiate a *digital* image f[x,y]?
  - Option 1: reconstruct a continuous image, then take gradient
  - Option 2: take discrete derivative (finite difference)

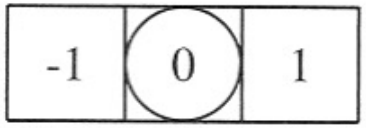
$$\frac{\partial f}{\partial x}[x, y] \approx f[x+1, y] - f[x, y]$$

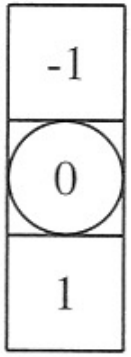
Mask operators

$$\frac{\partial f}{\partial x}[x,y] \approx f[x+1,y] - f[x,y]$$

$$\frac{\partial f}{\partial y}[x,y] \approx f[x,y+1] - f[x,y]$$

Interpretation of the equations as a symetric mask





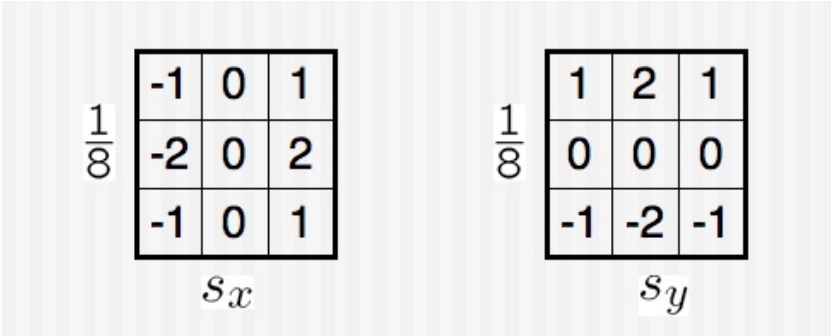
For image *f* and mask *h* the derative is simply computed as *g=f*h*

Mask operators

- Since the distance from the central point where the derivative is estimated is one pixel to the right..
- …computation yields only half of the derivative (sensitive to noise)
- Improvement, take more pixels into account

The Sobel operator

- Better approximations of the derivatives exist
  - The *Sobel* operators below are very commonly used



The standard defn. of the Sobel operator omits the 1/8 term

- doesn't make a difference for edge detection
- the 1/8 term **is** needed to get the right gradient value

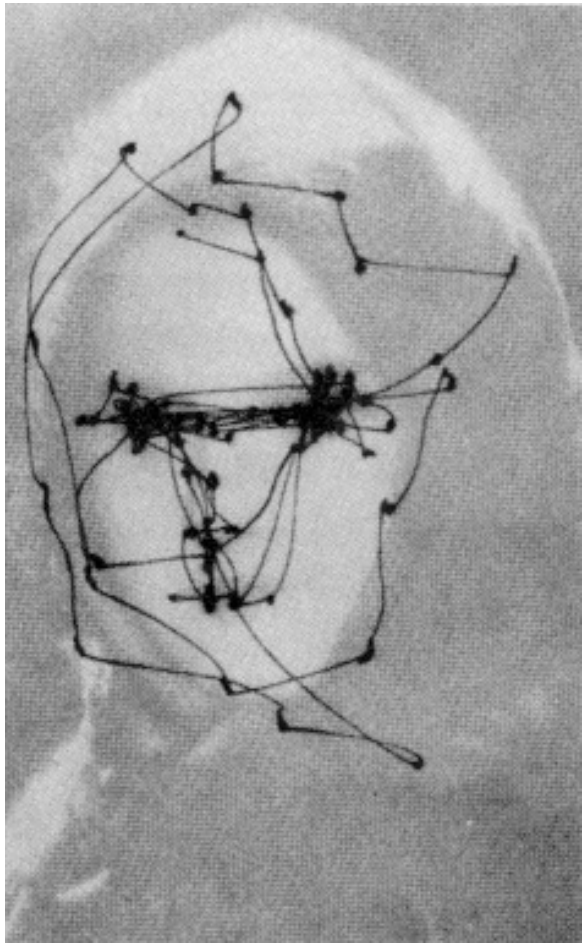
- An image is described by several vectors of fixed dimension 128. Each vector represents an invariant key feature descriptor of the image.
- In the first step of the SIFT algorithm, keypoints are detected. Key- points are locations that are invariant to scale change
- In the next step, the local maxima and minima are determined by the SIFT algorithm. They correspond to the local extrema with respect to both space and scale.

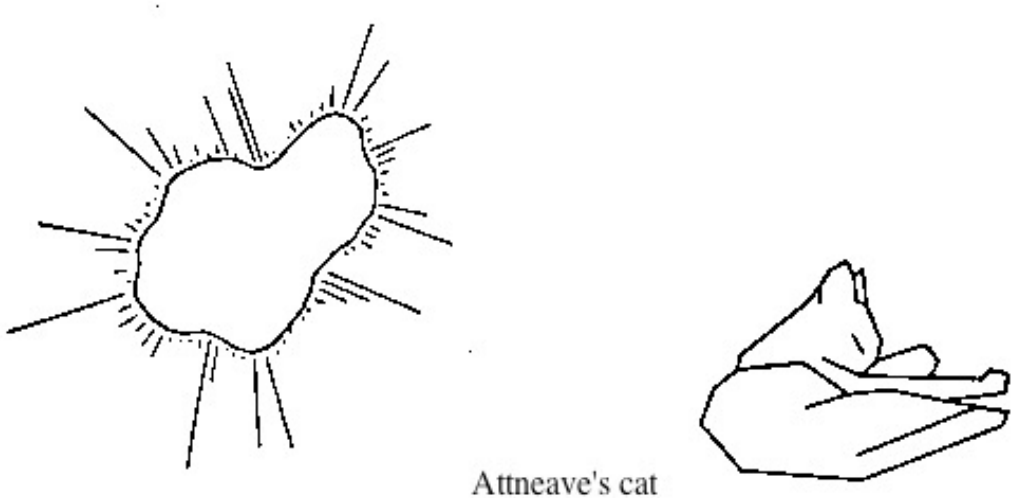
SIFT

A scale-invariant feature transform (SIFT) generates local features of an image that are invariant to image scaling and rotation and illumination

Eye Movements

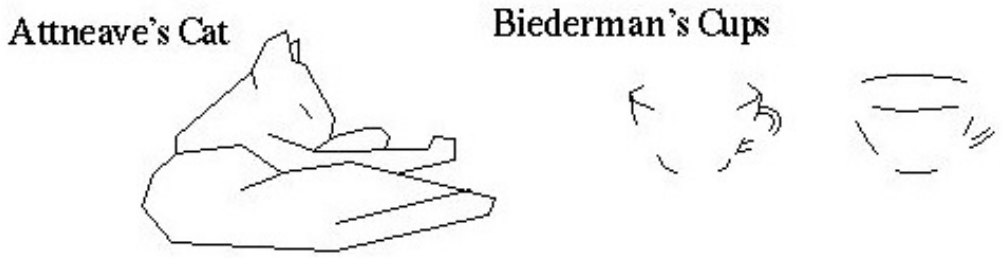
- Saccadic Movement
  - fixation point to fixation point
  - dwell period: 200-600 msec
  - saccade: 20-100 msec
- Smooth Pursuit Movement
  - tracking moving objects in visual field
- Convergent Movement
  - tracking objects moving away or toward us





Attneave's cat

- For example Attneave showed that a picture of a cat can be simplified by replacing all lines of low curvature with straight lines, as shown below, without adversely affecting the recognizability of the cat
- In other words, the lines of low curvature represent redundant information, which therefore need not be explicitly stored in memory



- Biederman gives further evidence for this notion by showing that a cup remains recognizable after removal of its lines of low curvature, whereas it becomes unrecognizable after removal of its points of high curvature and line intersections.

Information Content of Contours

- Information associated with a contour is not uniformed distributed
- Experiment
  - Ask subject to place a number of fixed points on a blank sheet of paper so that they provide best approximation of a curve
  - People tend to place the points in the same relation
  - Histogram, number of subjects that placed points into these segments

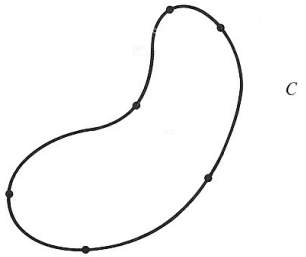


Figure 5.6.1 A simple closed curve.

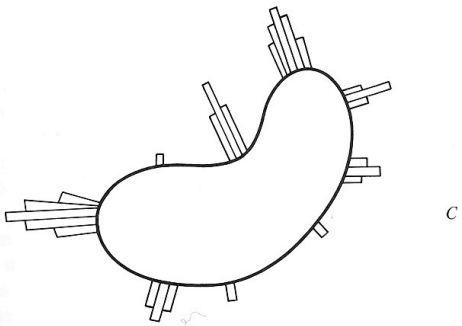
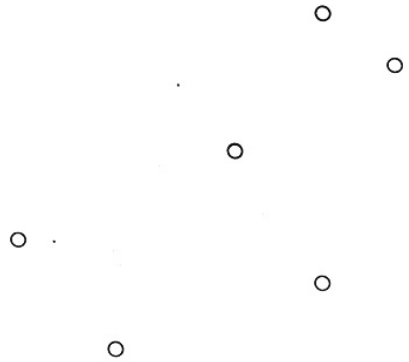


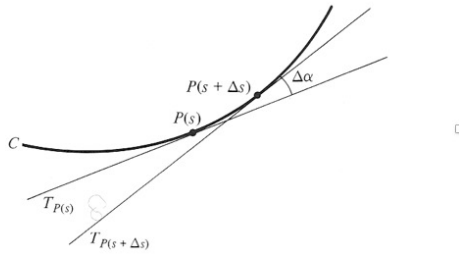
Figure 5.6.2 Histogram for approximations to *C*.

- If only six points were available, a person would most likely place them as shown (approximate the curve best)



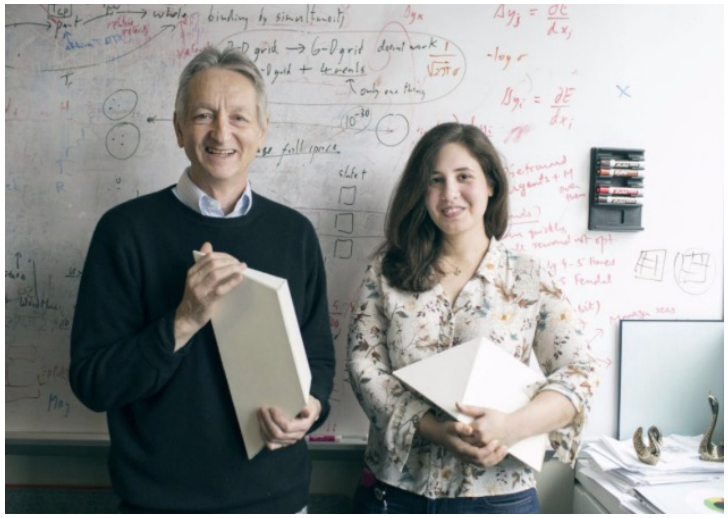
- Illusion of the cat remains if large portions of the line segments are erased
- The information is concentrated in the neighborhood of the points of extreme curvature

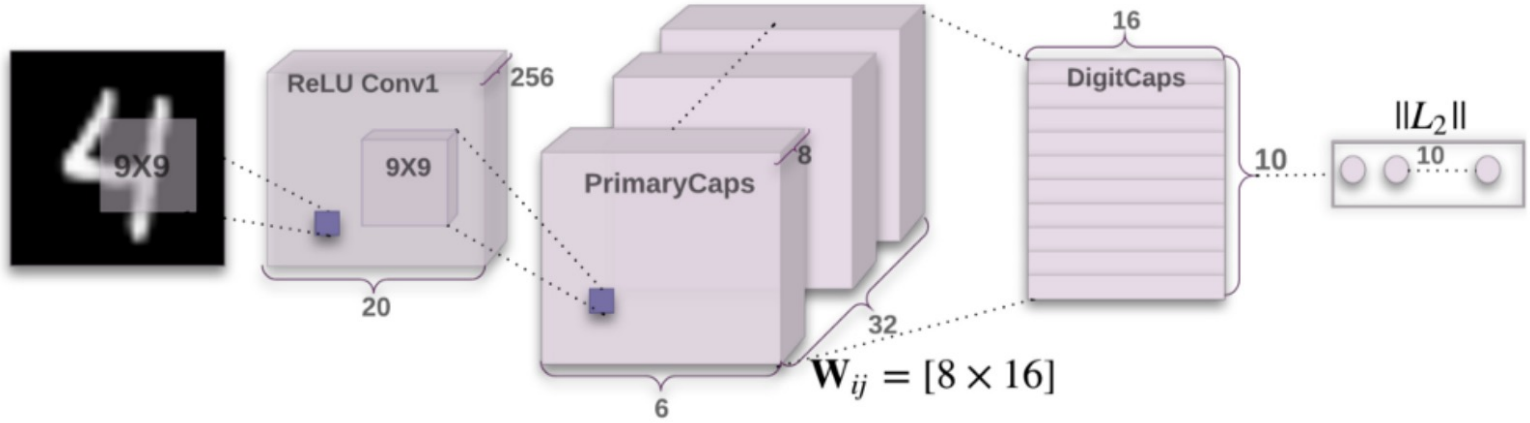
- The right angle is the only place where the contour is curved, changes its direction
- At the point of extreme curvature, the information is concentrated
- Corners yield the greatest information
- More strongly curved points yield more information
- Information content of a contour is concentrated in the neighborhood of points where the absolute value of the curvature is a local maximum



- Geoffrey Hinton:

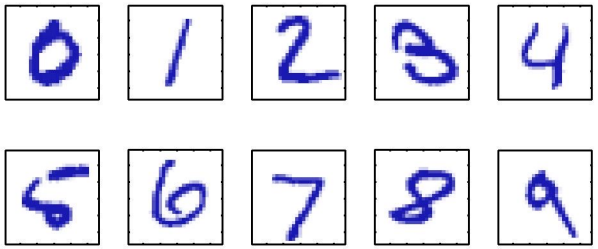
- *The pooling operation used in convolutional neural networks is a big mistake, and the fact that it works so well is a disaster!”*
- The pooling layer was destroying information and making it impossible for networks to learn higher-level concepts.
- A new architecture that didn’t rely so heavily on this operation.





The new model CapsuleNet proposed by Sara Sabour (and Geoffry Hinton) claims to deliver state of the art results on MNIST

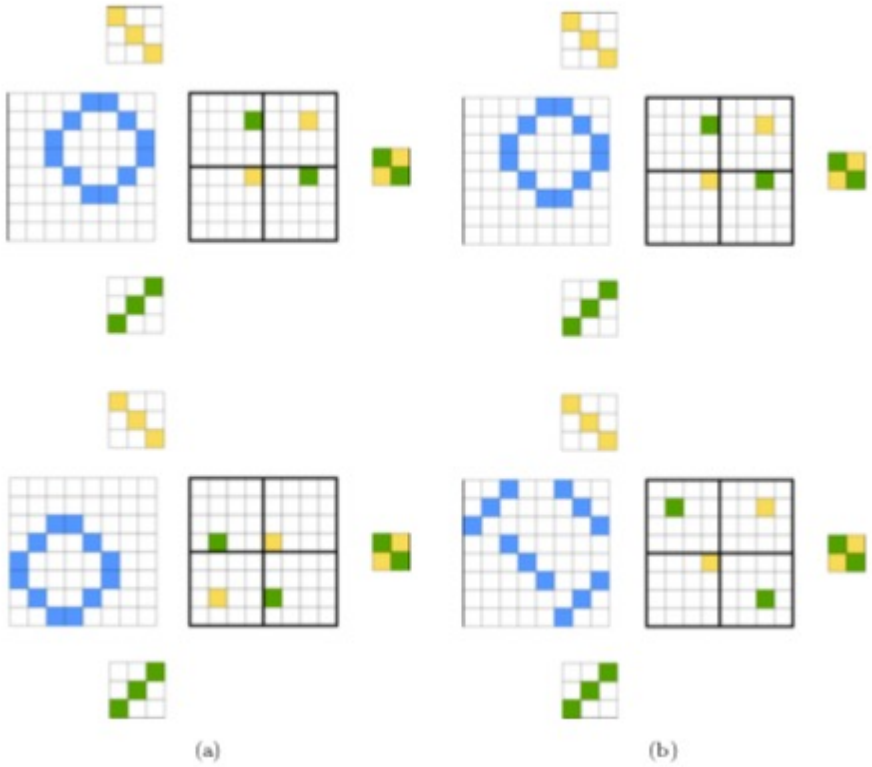
Attention Inspired Network



Handwritten Digit Recognition

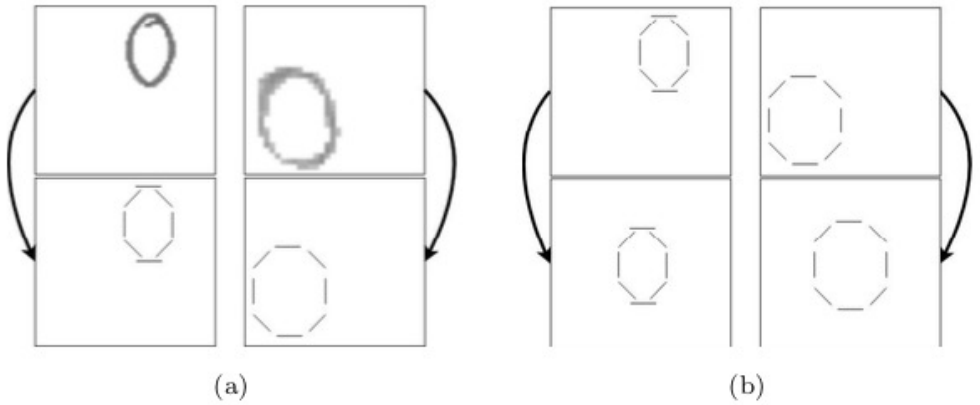
- Luis Sá-Couto, Attention Network: steep learning curve in an invariant pattern recognition model, MSc in Computer Science and Engineering,
- Luis Sá-Couto and Andreas Wichert, Attention Inspired Network: steep learning curve in an invariant pattern recognition model, **Neural Networks**, in Press, doi:10.1016/j.neunet.2019.01.01

Convolutional Networks



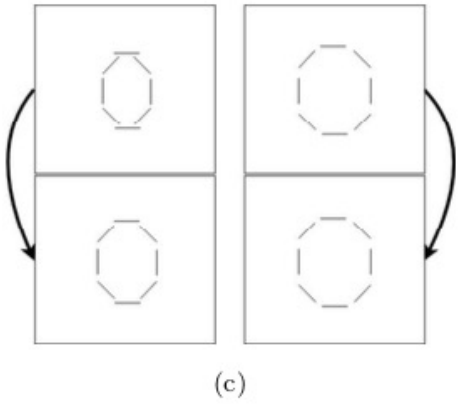
Large Subsampling Window

Two variants of the same category.

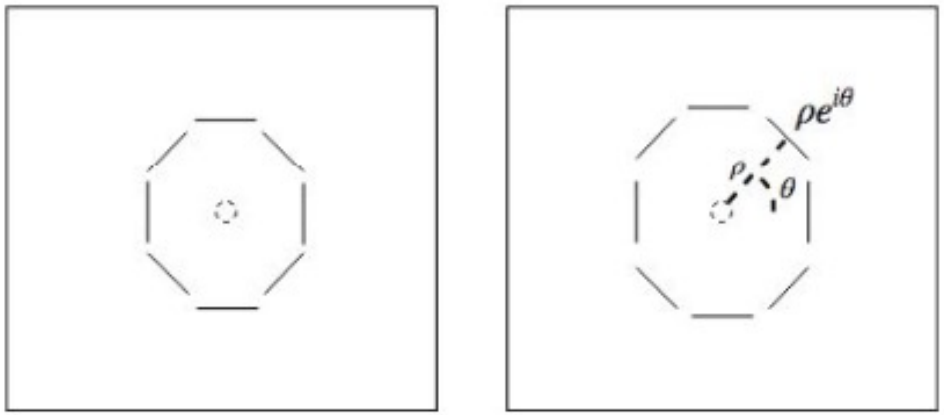


(a) Two patterns are mapped into features in the retinotopic system

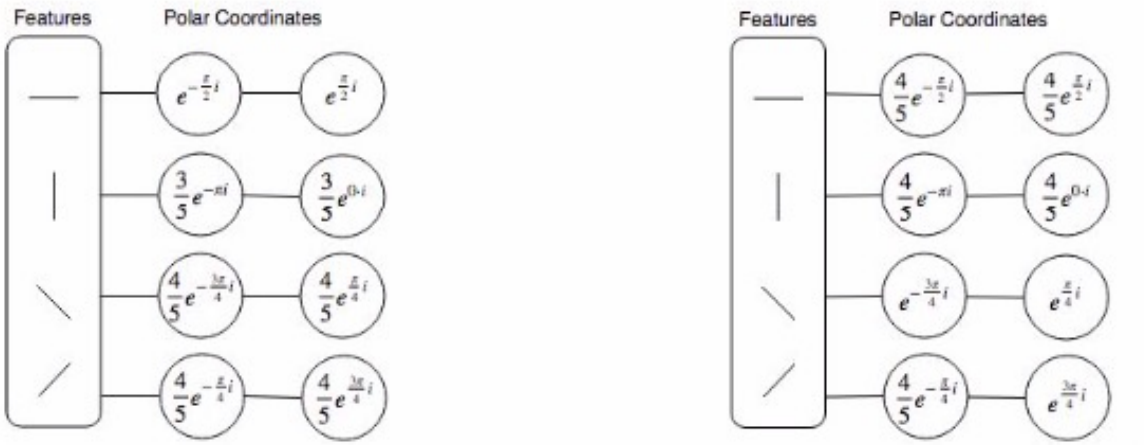
(b) Object-centered coordinate system



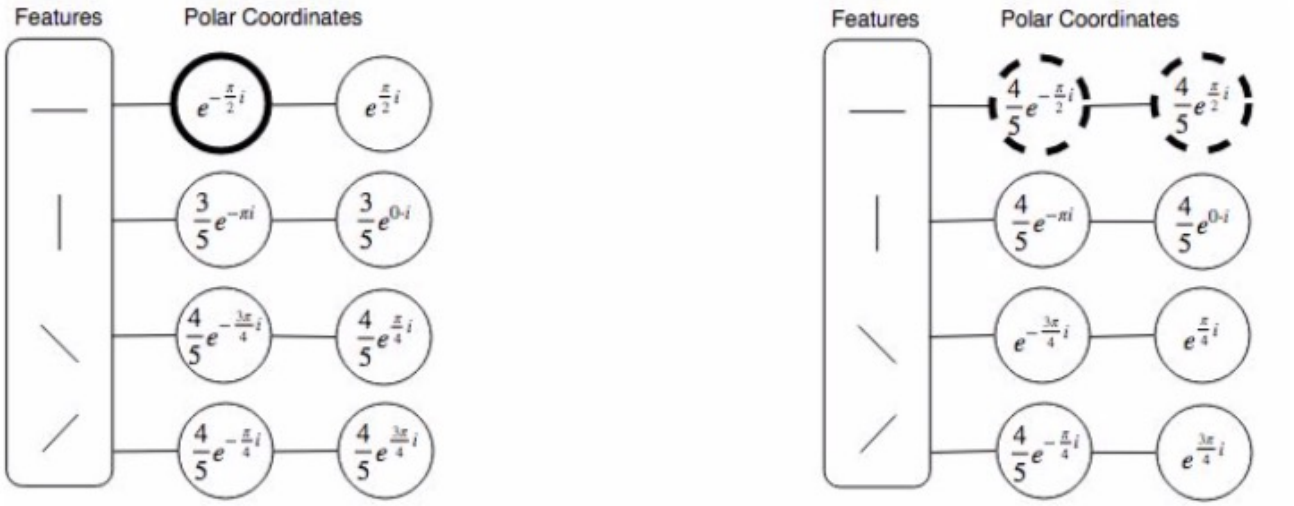
(c) Normalization to the unitary radius circle



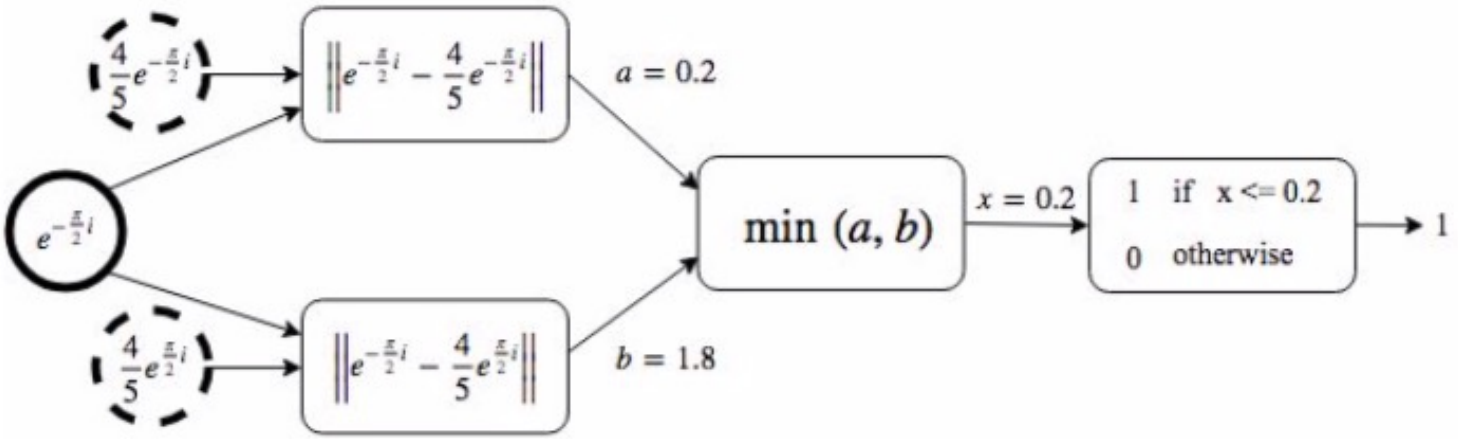
- Each feature's position is coded as a complex number
  - (Polar Coordinate)



Two patterns coded as sets where each feature is coded by its type and its position



A feature is selected for analysis and is compared with previously learned features of the same kind.



A sketch of the analysis operation for a given feature

Table 1: Results of comparable biologically inspired models on the MNIST test set.

Model	Train Set	Test Set	Accuracy
CNN (LeCun et al., 1998)	60000	10000	98.30%
MTC (Cardoso & Wichert, 2013)	60000	10000	99.30%
Ensemble of CNNs (Ciregan et al., 2012)	60000	10000	99.70%
Capsule Networks (Sabour et al., 2017)	60000	10000	99.75%
Attention Inspired Network	7000	10000	100.00%
Attention Inspired Network (No Reg)	60000	10000	98.36%
Attention Inspired Network (Reg)	60000	10000	100.00%

Table 2: Results of comparable biologically inspired models on ETL-1.

Model	Train Set	Test Set	Accuracy
MTC (Cardoso & Wichert, 2010)	200	3000	93.33%
Attention Inspired Network	200	3000	98.47%
Neocognitron (Fukushima, 2003)	3000	3000	98.6%
Attention Inspired Network	3000	3000	99.88%

Object Recognition

- What is Object Recognition?
  - Segmentation/Figure-Ground Separation:
  - Labeling an object [The focus of most studies]
  - Extracting a parametric description as well

- Object Recognition versus Scene Analysis
  - An object may be part of a scene or
  - Itself be recognized as a “scene”

Shape perception and scene analysis

- Shape-selective neurons in cortex
  - Coding: one neuron per object
  - or population codes?
- Biologically-inspired algorithms for shape perception

- Visual memory: how much do we remember of what we have seen?
- The world as an outside memory and our eyes as a lookup tool

Four stages of representation (Marr, 1982)

- 1) pixel-based (light intensity)
- 2) primal sketch (discontinuities in intensity)
- 3) 2 ½ D sketch (oriented surfaces, relative depth between surfaces)
- 4) 3D model (shapes, spatial relationships, volumes)

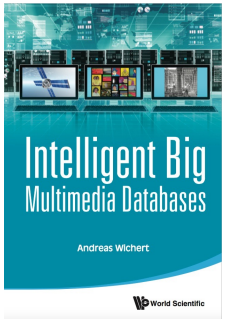
Artificial intelligence pioneer (Geoffrey Hinton) says we need to start over

We need to invent **new** machine learning algorithm /approaches

How about you?

- Back-propagation still has a core role in AI's future.
- Entirely new methods will probably have to be invented
- "I don't think it's how the brain works," he said. "We clearly don't need all the labeled data.

Literature



- Intelligent Big Multimedia Databases, A. Wichert, World Scientific, 2015
  - Chapter 3, 5