DEALING WITH IRREGULARLY SAMPLED DATA

Maria Petrou

Imperial College London

- Scientists in many other disciplines had to deal with irregularly sampled data for many years

- Several techniques have been developed for that

- Only one method was specifically developed for computer vision

How can we do image processing when the data are not regularly sampled?

• Interpolate the data so that they appear on a regular grid: **griding**
Involves an extra step;
May introduce errors;
• Perform the operations directly on the available data
How?
☐ Some techniques of IP readily applicable to irregularly placed data, eg
Co-occurrence matrices
Mathematical morphology
☐ BUT: linear IP is largely based on the use of a regular grid
convolution based
transform based

INTERPOLATION OF IRREGULARLY SAMPLED DATA

- **Kriging**
- **Iterative error correction**
- **Normalised convolution**

Kriging

In a nutshell:
Give to the missing points values that are weighted linear combinations of the values of the points you have.
Choose the weights so that the covariances of the data are preserved.

BLUE: Best Linear Unbiased Estimator

$P_1, P_2, \ldots, P_n$: irregularly placed data points
m: their mean
σ^2: their variance
$V(P_i)$: random variable defined at points $P_1, P_2, \ldots, P_n$.

Problem: Estimate the value of this variable at point P_0.
Kriging estimation:

$$\hat{V}(P_0) = \sum_{i=1}^{n} w_i(P_0) V(P_i) \tag{1}$$

Residual error:

$$R(P_0) \equiv \hat{V}(P_0) - V(P_0) \tag{2}$$

Choose weights w_i so that the variance of error $R(P_0)$ is minimised.

Variance of the error function:

$$\tilde{\sigma}_R^2 \;=\; \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \tilde{C}_{ij} - 2\sum_{i=1}^{n} w_i \tilde{C}_{i0} + \tilde{\sigma}^2 \tag{3}$$

where

$\tilde{C}_{ij}$: covariance between random variables $V(P_i)$ and $V(P_j)$

$\tilde{C}_{i0}$: covariance between random variables V_i and V_0.

From the given data work out the covariance matrix $\tilde{C}$

Choose weights so that the variance of the error is minimal, subject to the condition

$$\sum_{i=1}^{n} w_i = 1 \tag{4}$$

Use **Lagrange Parameter** optimisation.

Minimise:

$$\tilde{\sigma}_R^2 \;=\; \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \tilde{C}_{ij} - 2\sum_{i=1}^{n} w_i \tilde{C}_{i0} + \tilde{\sigma}^2 + \underbrace{2\mu\left(\sum_{i=1}^{n} w_i - 1\right)}_{0} \qquad (5)$$

where
μ is the Lagrange multiplier.
We have $(n+1)$ unknowns now.
Differentiate (5), with respect to the $(n+1)$ unknowns and set these first partial derivatives to zero:

$$CW = D \qquad (6)$$

Solve for the weights:

$$W = C^{-1}D \qquad (7)$$

where

$$C = \begin{pmatrix} \tilde{C}_{11} & \dots & \tilde{C}_{1n} & 1 \\ \dots & \dots & \dots & \dots \\ \tilde{C}_{n1} & \dots & \tilde{C}_{nn} & 1 \\ 1 & \dots & 1 & 0 \end{pmatrix}_{(n+1)\times(n+1)} \tag{8}$$

$$W = \begin{pmatrix} w_1 \\ \dots \\ w_n \\ \mu \end{pmatrix}_{(n+1)\times 1} \tag{9}$$

$$D = \begin{pmatrix} \tilde{C}_{10} \\ \dots \\ \tilde{C}_{n0} \\ 1 \end{pmatrix}_{(n+1)\times 1} \tag{10}$$

Estimation of the covariance matrix of the data

This is done via the Variogram (or semi-variogram)

To reduce the effect of noise, we use a parametric model to fit the variogram

Variogram definition

Semi-variogram: half the expected squared difference between two data points separated by a distance vector h:

$$\gamma(h) \equiv \frac{1}{2} E\{[V(P_i) - V(P_i + h)]^2\} \tag{11}$$

where
E: the expectation operator.
$N(h)$: the total number of distinct pairs of data points V_i and V_j, the positions of which are at a distance h from each other.
Then:

$$\gamma(h) = \frac{1}{2|N(h)|} \sum_{(i,j)|d_{ij}=h} (V_i - V_j)^2 \tag{12}$$

Relationship between the variogram and its corresponding covariance

$$\gamma(h) = \tilde{\sigma}^2 - \tilde{C}(h) \tag{13}$$

where
$\tilde{\sigma}^2$: the variance of the random variables.

Three important parameters of the semi-variogram

•**Nugget**: For a very small value of the distance, it is expected that the value of the semi-variogram will reach zero but if the value of the variogram does not approach zero due to sampling error or some other factors, then this nonzero value is known as the nugget effect.

•**Range**:As the distance between the data points increases the value of the semi-variogram also increases. After a certain point, the increase in the distance does not have any effect on the value of the variogram, i.e. after this particular distance the value of the variogram becomes constant. This particular distance is known as the Range.

•**Sill**: The maximum vertical height attained by the semi-variogram at the range is known as the Sill.

Variogram models

- **Fractal model**

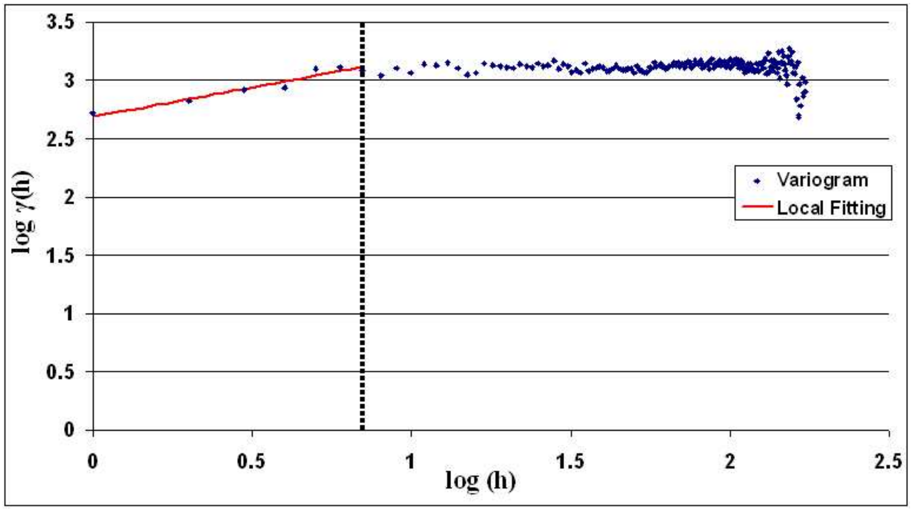
$$\tilde{\gamma}(h) = \gamma_0 h^{2H} \Rightarrow \log \tilde{\gamma}(h) = \log \gamma_0 + 2H \log h \tag{14}$$

The corresponding fractal covariance function

$$\tilde{C}(h) = \begin{cases} C_0 + C_1 & \text{if } |h| = 0 \\ C_0 + C_1 - \gamma_0 h^{2H} & \text{if } |h| > 0 \end{cases} \tag{15}$$

where C_0 is the nugget effect, $C_0 + C_1$ is the sill.

Example of variogram fitting with the fractal model.
Vertical dotted line: the distance up to which the model fits.



- **Exponential model**

$$\tilde{\gamma}(h) = \begin{cases} 0 & \text{if } |h| = 0 \\ C_0 + C_1 \left(1 - \exp\left(\frac{-|h|}{a}\right)\right) & \text{if } |h| > 0 \end{cases} \tag{16}$$

The corresponding covariance function:

$$\tilde{C}(h) = \begin{cases} C_0 + C_1 & \text{if } |h| = 0 \\ C_1 \exp\left(\frac{-|h|}{a}\right) & \text{if } |h| > 0 \end{cases} \tag{17}$$

where C_0 is the nugget effect, $C_0 + C_1$ is the sill and a is the range.

- **Spherical model**

$$\tilde{\gamma}(h) = \begin{cases} C_0 + C_1 & \text{if } |h| \geq a \\ C_0 + C_1 \left(1.5\frac{h}{a} - 0.5(\frac{h}{a})^3\right) & \text{if } |h| < a \end{cases} \tag{18}$$

The corresponding covariance function:

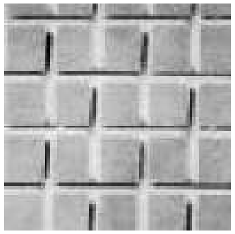
$$\tilde{C}(h) = \begin{cases} 0 & \text{if } |h| \geq a \\ C_1 \left(1 - 1.5\frac{h}{a} + 0.5(\frac{h}{a})^3\right) & \text{if } |h| < a \end{cases} \tag{19}$$

- **Gaussian model**

$$\tilde{\gamma}(h) = C_0 + C_1 - C_1 \exp\left(\frac{-|h|^2}{a^2}\right) \tag{20}$$

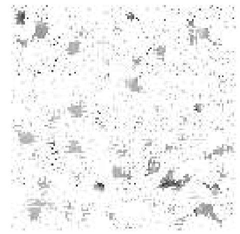
- **Linear model**

$$\tilde{\gamma}(h) = C_0 + C_1 \frac{h}{a} \quad \text{if } |h| > 0 \tag{21}$$

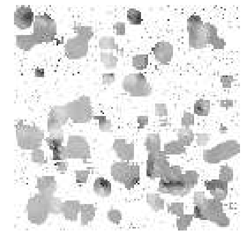


(a) Original

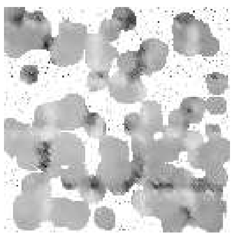
(b) Given data



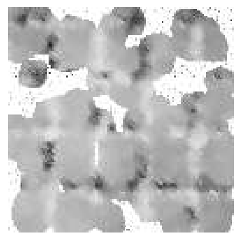
(c): Iter 1



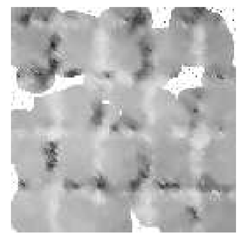
(d): Iter 2



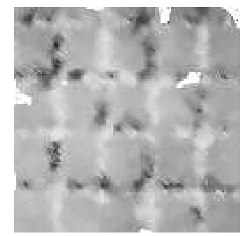
(e): Iter 3



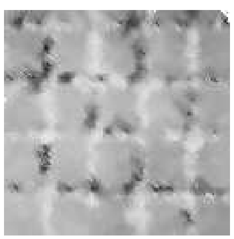
(f): Iter 4



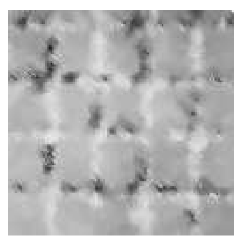
(g): Iter 5



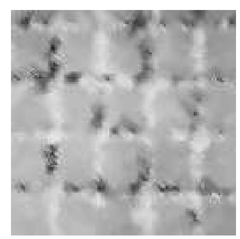
(h): Iter 6



(i): Iter 7



(j): Iter 8



(k) : Iter 9

Iterative error correction

Problem statement: Given a sequence of sampled values of some unknown signal, design functions called **frames** such that the internal product, indicated by $< \cdot >$, of the unknown signal with the frame functions is equal to the values of the signal at the sampling points.

Theory of frames: under certain conditions it is possible to determine frame functions from which a **Frame Operator** used to reconstruct the signal from its samples in an iterative way.

Difference from using **basis functions** of the function space to cover the whole space and express any signal as a linear combination of them:

1) The coefs of the expansion in terms of basis functions are unique; not so the expansion in terms of frames.

2) The number of frames is larger than the number of basis functions (frames do not constitute an orthogonal basis).

In practice:

f: the function we wish to reconstruct;

A: a frame operator.

Set

$$f_0 = Af \tag{22}$$

and

$$f_{n+1} = f_n + A(f - f_n). \tag{23}$$

For an appropriate A it can be shown that

$$\lim_{n \to \infty} f_n = f, \tag{24}$$

with the estimation error after n iterations being

$$||f - f_n|| \leq \gamma^{n+1} ||f||, \tag{25}$$

where $\gamma < 1$ is a constant and $|| \quad ||$ is an appropriate norm

Different methods according to the type of initial guess and operator A:

- **Wiley/Marvasti (WILMAR) method:** initial guess is a trivial interpolation, where all the unknown points are set equal to zero.
- **Adaptive Weights (ADW) method:** the trivial interpolation is weighted by some factors that reflect the distance of the irregular samples from the neighbouring sampling points.
- **Voronoi (VOR) method:** uses the nearest neighbour interpolation.
- **Piecewise-linear method:** interpolation with a linear continuous function defined between successive sampling points.
- **Projection onto Convex Sets (POCS) method:** operator A not linear: the iterations are obtained by application of successive *projections*. A projection is equivalent to low pass filtering in the Fourier domain. A projection, therefore, limits the band of the signal.

An example

$f(x)$: the continuous signal;
$f(n)$: set of N irregular samples, for $n = 1, \ldots, N$.

Step 1: Use the Voronoi method to interpolate the samples on a regular grid.
Step 2: Calculate the Fourier transform of the regularly re-sampled signal.
Step 3: Discard the high frequencies.
Step 4: Calculate the inverse FT, to produce estimate f_1.
Step 5: Calculate the error committed at the irregularly sampled coordinates $f(n) - f_1(n)$, for $n = 1, \ldots, N$.
Step 6: Interpolate the error using the Voronoi method
Step 7: Use the estimated error at each point to produce improved estimate $f_2(n)$.
Go to step 2 and repeat.

Convergence to the continuous signal is guaranteed if the maximal gap between two irregular samples is smaller than the Nyquist limit.

Typically 20 iterations

Fastest algorithm: ADW

Normalised Convolution

$f(x, y)$: an image
$g(x, y)$: a smoothing filter
(x_s, y_s), for $s = 1, \ldots, S$: random positions for which the image values are known.
$c(x, y)$: the sampling mask:

$$c(x, y) = \begin{cases} 1 & \text{if } (x, y) = (x_s, y_s) \text{ for some } s \in [1, S] \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

Step 1: Convolve $f(x,y)c(x,y)$ with $g(x,y)$:

$$C(x,y) \equiv \Big(f(x,y)c(x,y)\Big) * g(x,y). \tag{27}$$

Step 2: Convolve $c(x,y)$ with $g(x,y)$:

$$NC(x,y) \equiv c(x,y) * g(x,y). \tag{28}$$

Step 3: Divide the two results point by point:

$$\tilde{f}(x,y) = \frac{C(x,y)}{NC(x,y)}. \tag{29}$$

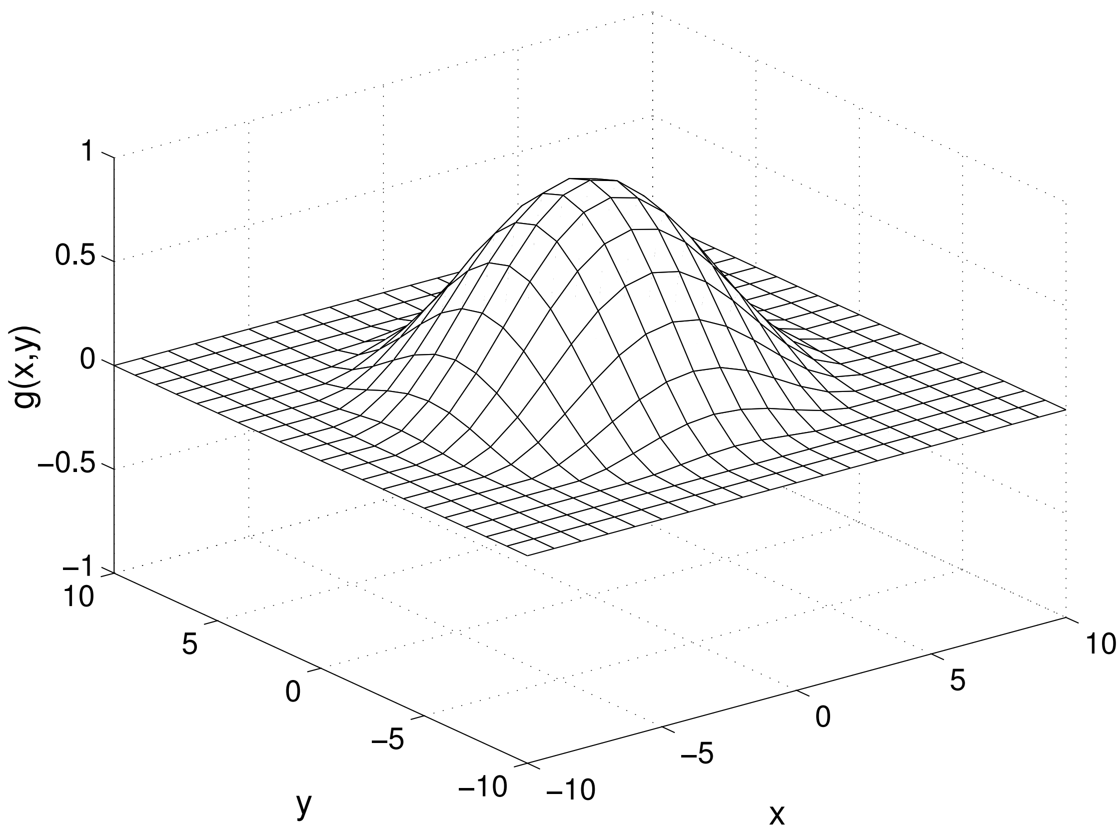
Use for $g(x, y)$ a low pass filter, eg the integral of the Canny filter.

Proposed filter by Westin and Knutsson:

$$g(x,y) = \begin{cases} r^{-\alpha} \cos^{\beta}(\frac{\pi r}{2r_{max}}) & \text{if } r < r_{max} \\ 0 & \text{otherwise} \end{cases}, \tag{30}$$

where:
$r \equiv \sqrt{x^2 + y^2}$
α and β: some positive integers.

Filter $g(x, y)$ for $\alpha = 0$, $\beta = 2$ and $r_{max} = 8$:



10% of the original number of pixels.

An example

$$f(t) = \left[x_1, 0, 0, x_4, x_5, 0, x_7, 0\right], \tag{31}$$

x_i: known samples, while the missing samples are zeros.
Sampling sequence:

$$c(t) = \left[1, 0, 0, 1, 1, 0, 1, 0\right] \tag{32}$$

Filter:

$$g(t) = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]. \tag{33}$$

Step 1:

$$f(t) * g(t) = \left[\frac{x_1}{3}, \frac{x_1}{3}, \frac{x_4}{3}, \frac{x_4 + x_5}{3}, \frac{x_4 + x_5}{3}, \frac{x_5 + x_7}{3}, \frac{x_7}{3}, \frac{x_7 + x_1}{3}\right]. \tag{34}$$

Step 2:

$$c(t) * g(t) = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \frac{1}{3}, \frac{2}{3}\right] \tag{35}$$

Step 3:

$$\tilde{f}(t) = \frac{f(t) * g(t)}{c(t) * g(t)} = \left[x_1, x_1, x_4, \frac{x_4 + x_5}{2}, \frac{x_4 + x_5}{2}, \frac{x_5 + x_7}{2}, x_7, \frac{x_7 + x_1}{2}\right]. \tag{36}$$

A numerical example:

The perfect signal:

$$f(t) = [1, 2, 3, 4, 4, 1, 1, 1, 2, 4, 5, 4, 3, 2, 1]. \tag{37}$$

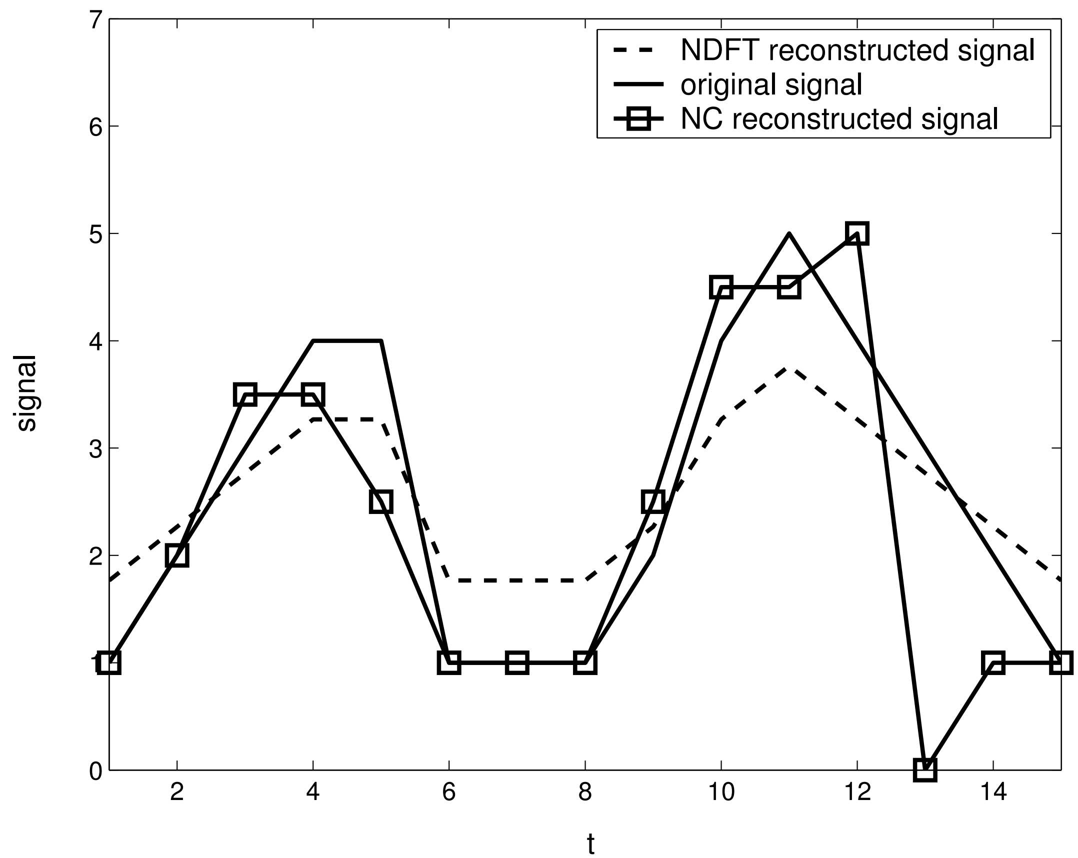
An irregular sampling of this signal:

$$f_i(t) = [1, 0, 3, 4, 0, 1, 0, 1, 0, 4, 5, 0, 0, 0, 1], \tag{38}$$

$$c(t) = [1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1]. \tag{39}$$

Filter:

$$g(t) = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right] \tag{40}$$



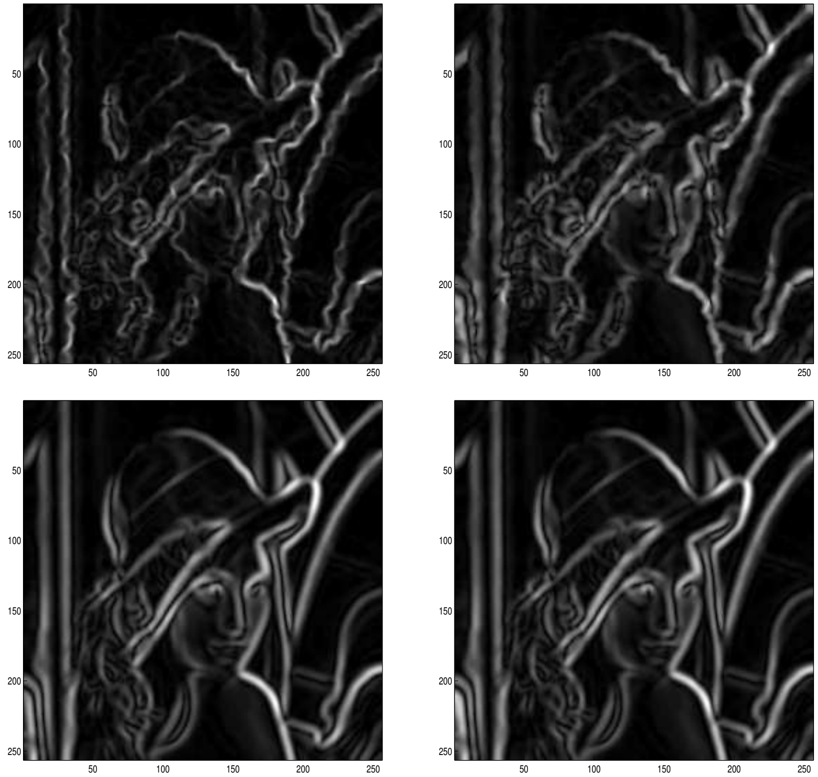
NDFT reconstructed signal
original signal
NC reconstructed signal
signal
t
0
1
2
3
4
5
6
7
2
4
6
8
10
12
14

- Error iterative correction produces better results than Normalised Convolution.
BUT!

- Normalised convolution can cope with lower rates of subsampling (bigger gaps).

- If I know how to do convolution with irregularly sampled images, I know how to do linear Image processing!!!

- Gradient detection: straightforward!



Gradient magnitude from 10% of pixels and from full image using Canny filter

This leads to the problem of estimating features **directly** from the irregular samples!

Can we compute the Fourier transform from irregular samples?

DFT OF IRREGULARLY SAMPLED DATA

Continuous FT:

$$P(\omega) = \int_{-\infty}^{+\infty} p(x) e^{-j\omega x} dx. \tag{41}$$

Inverse Continuous FT:

$$p(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} P(\omega) e^{j\omega x} d\omega \tag{42}$$

From Continuous FT to DFT:

Function $p(x)$ is sampled at N regular intervals equal to T_S, to produce samples p_n, so that:
$p_n \equiv p(x_n)$ where $x_n = nT_S$, for $n = 0, \ldots, N-1$
Total signal duration: $T = NT_S$
The Fourier transform of p_n is defined only at certain regularly spaced frequencies:
function $P(\omega)$ is only for certain values ω_m.
Samples $P(\omega_m)$ are regularly spaced as well;
they are multiples of a dominant frequency $\frac{1}{T}$:
$\omega_m \equiv m\left(\frac{2\pi}{T}\right)$, for $m = 0, \ldots, N-1$.

$$P(\omega_m) \equiv \sum_{n=0}^{N-1} p(x_n) e^{-j\omega_m x_n} \tag{43}$$

Or:

$$P(\omega_m) = \sum_{n=0}^{N-1} p(x_n) e^{-j\left(m\frac{2\pi}{T}\right)\left(nT_S\right)} = \sum_{n=0}^{N-1} p(x_n) e^{-j\left(m\frac{2\pi}{NT_S}\right)\left(nT_S\right)} \tag{44}$$

Finally DFT:

$$P(m) = \sum_{n=0}^{N-1} p(n) e^{-j\frac{2\pi}{N}mn} \tag{45}$$

Inverse DFT:

$$p(n) = \frac{1}{N} \sum_{m=0}^{N-1} P(m) e^{j\frac{2\pi}{N}mn} \tag{46}$$

From DFT to Non-uniform Discrete Fourier Transform (NDFT)

• Usually, we want to know the frequency content at regularly placed frequency samples:

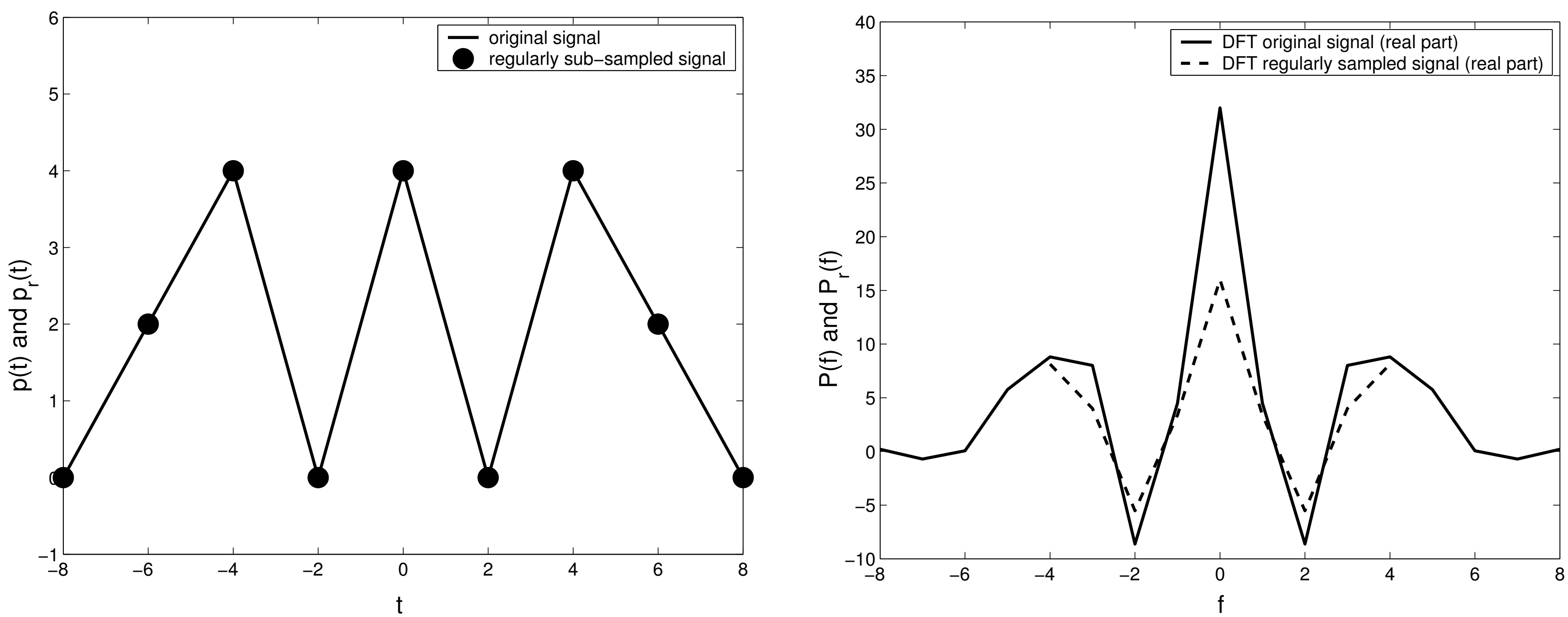
$$P(m) = \sum_{n=0}^{N-1} p(n) e^{-jm\Delta k x_n} \tag{47}$$

Set $\Delta k = \frac{2\pi}{T}$ where T is the range of extension of samples x_n. NDFT similar to DFT except for the presence of the spatial coordinates x_n instead of index n. So:

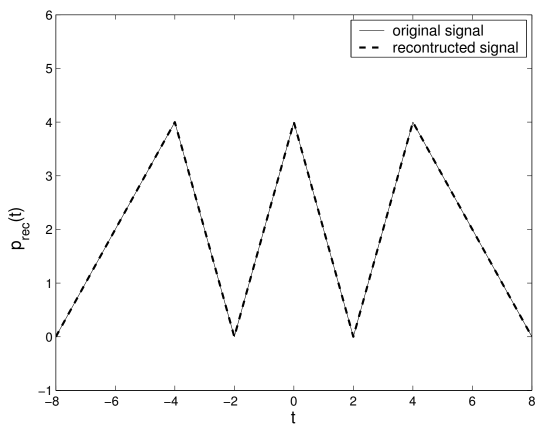
$$P(m) = \sum_{n=0}^{N-1} p(n) e^{-j\frac{2\pi}{T} m x_n} \tag{48}$$

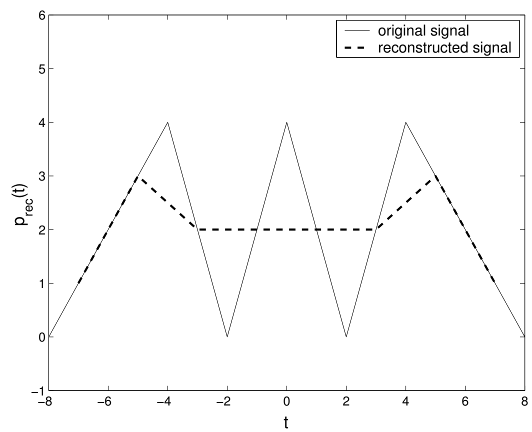
An example

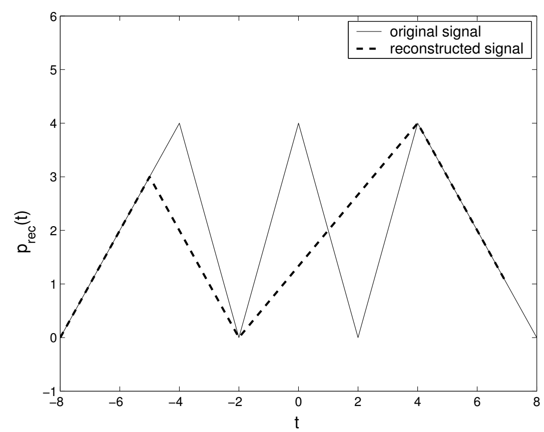
A 1D continuous signal $p(t)$, sampled at integer values of t: $[0, 1, 2, 3, 4, 2, 0, 2, 4, 2, 0, 2, 4, 3, 2, 1, 0]$ and the real part of its DFT.



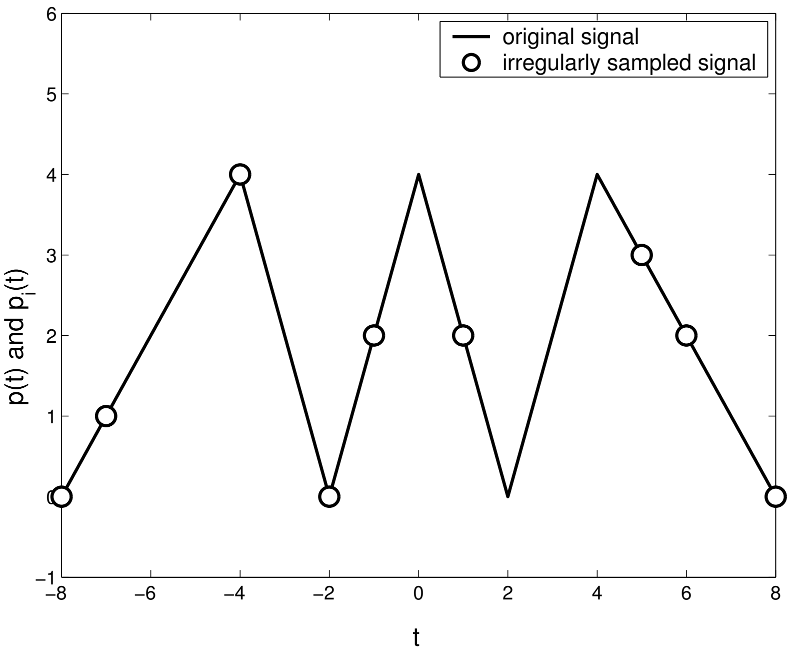
Reconstruction of the signal using $N = 9$, $N = 8$ and $N = 6$ regularly spaced samples:

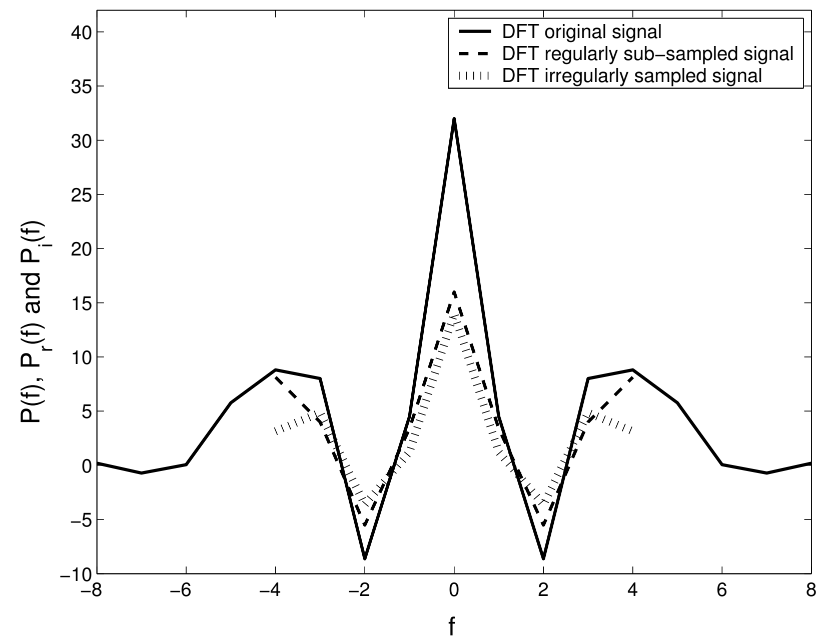




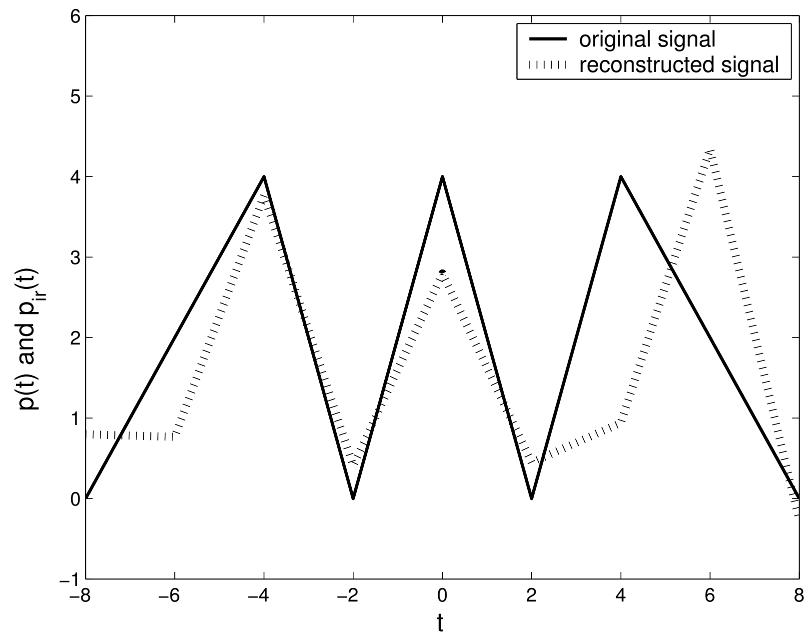


Use an irregular sampling pattern $t_i = [-8, -7, -4, -2, -1, 1, 5, 6, 8]$ resulting in the sequence: $p_i(t) = [0, 1, 4, 0, 2, 2, 3, 2, 0]$.





Reconstruction of the signal using the irregularly spaced samples:



How can we do general Image Processing and Pattern Recognition with irregularly sampled data?

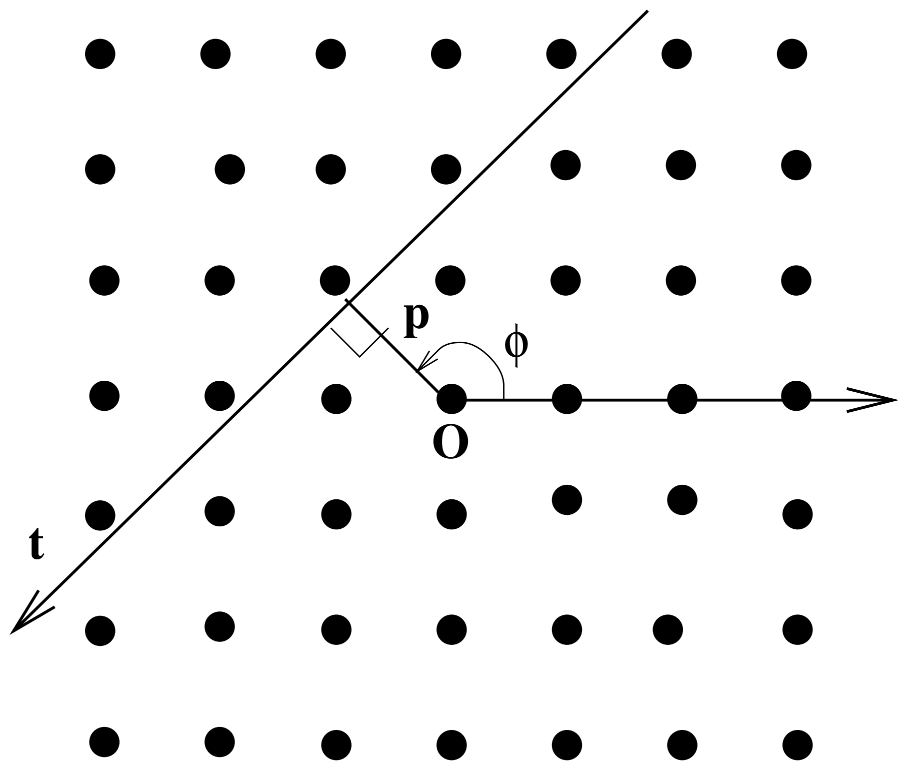
There are alternative representations of the image information that do not rely on the use of regular grids.

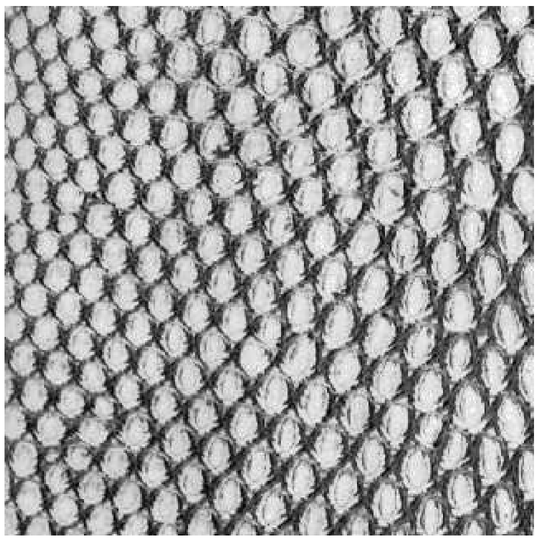
We do not need to use features that make perceptual sense! They may make mathematical sense, or pattern recognition sense instead!

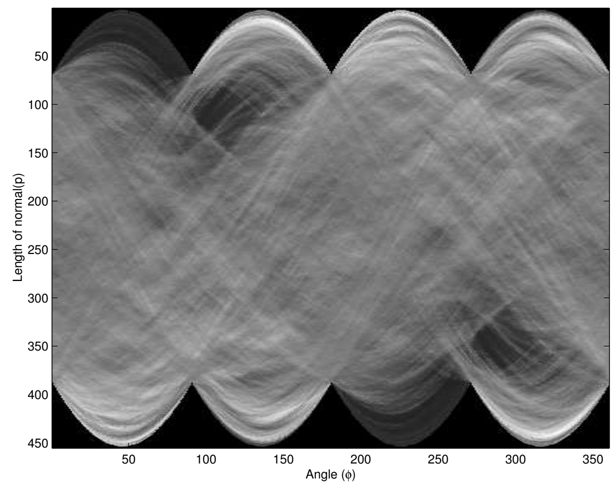
We may use features that can do the job, just like the human brain does it in the subconscious level.

Sub-conscious Image processing!

The trace transform







Length of normal(p)
50
100
150
200
250
300
350
400
450
50
100
150
200
250
300
350
Angle (ϕ)

Features from the Trace transform

• Compute from the image a functional T along tracing lines (ϕ, p).
Trace transform: $Result_T(\phi, p)$.

• Compute on $Result_T(\phi, p)$ a functional P for all values of p
Circus function: $Result_P(\phi)$.

• Compute on $Result_P(\phi)$ a functional Φ for all values of ϕ
Triple feature: A number that depends on the combination $TP\Phi$.

	T Functional
1	$\sum_{i=0}^{N} x_i$
2	$\sum_{i=0}^{N} i x_i$
3	2nd_Central_Moment/Sum_of_all_values
4	$\sqrt{\sum_{i=0}^{N} x_i^2}$
5	$Max_{i=0}^{N} x_i$
6	$\sum_{i=0}^{N-1} \lvert x_{i+1} - x_i \rvert$
7	$\sum_{i=0}^{N-1} \lvert x_{i+1} - x_i \rvert^2$
8	$\sum_{i=3}^{N-3} \lvert x_{i-3} + x_{i-2} + x_{i-1} - x_{i+1} - x_{i+2} - x_{i+3} \rvert$
9	$\sum_{i=0}^{N-2} \lvert x_{i-2} + x_{i-1} - x_{i+1} - x_{i+2} \rvert$
10	$\sum_{i=4}^{N-4} \lvert x_{i-4} + x_{i-3} + ... + x_{i-1} - x_{i+1} - ... - x_{i+3} - x_{i+4} \rvert$
11	$\sum_{i=5}^{N-5} \lvert x_{i-5} + x_{i-4} + ... + x_{i-1} - x_{i+1} - ... - x_{i+4} - x_{i+5} \rvert$
12	$\sum_{i=6}^{N-6} \lvert x_{i-6} + x_{i-5} + ... + x_{i-1} - x_{i+1} - ... - x_{i+5} - x_{i+6} \rvert$
13	$\sum_{i=7}^{N-7} \lvert x_{i-7} + x_{i-6} + ... + x_{i-1} - x_{i+1} - ... - x_{i+6} - x_{i+7} \rvert$
14	$\sum_{i=4}^{N-4} \sum_{k=0}^{4} \lvert x_{i-k} - x_{i+k} \rvert$
15	$\sum_{i=5}^{N-5} \sum_{k=0}^{5} \lvert x_{i-k} - x_{i+k} \rvert$
16	$\sum_{i=6}^{N-6} \sum_{k=0}^{6} \lvert x_{i-k} - x_{i+k} \rvert$
17	$\sum_{i=7}^{N-7} \sum_{k=0}^{7} \lvert x_{i-k} - x_{i+k} \rvert$
18	$\sum_{i=10}^{N-10} \sum_{k=0}^{10} \lvert x_{i-k} - x_{i+k} \rvert$
19	$\sum_{i=15}^{N-15} \sum_{k=0}^{15} \lvert x_{i-k} - x_{i+k} \rvert$
20	$\sum_{i=20}^{N-20} \sum_{k=0}^{20} \lvert x_{i-k} - x_{i+k} \rvert$

	P Functional
1	$Max_{i=0}^{N} x_i$
2	$Min_{i=0}^{N} x_i$
3	$\sqrt{\sum_{i=0}^{N} x_i^2}$
4	$\frac{\sum_{i=0}^{N} i x_i}{\sum_{i=0}^{N} x_i}$
5	$\sum_{i=0}^{N} i x_i$
6	$\frac{1}{N} \sum_{i=0}^{N} (x_i - x\hat{)}^2$
7	c so that: $\sum_{i=0}^{c} x_i = \sum_{i=c}^{N} x_i$
8	$\sum_{i=0}^{N-1} \lvert x_{i+1} - x_i \rvert$
9	c so that: $\sum_{i=0}^{c} \lvert x_{i+1} - x_i \rvert = \sum_{i=c}^{N-1} \lvert x_{i+1} - x_i \rvert$
10	$\sum_{i=0}^{N-4} \lvert x_i - 4x_{i+1} + 6x_{i+2} - 4x_{i+3} + x_{i+4} \rvert$

	Φ Functional
1	$\sum_{i=0}^{N-1} \lvert x_{i+1} - x_i \rvert^2$
2	$\sum_{i=0}^{N-1} \lvert x_{i+1} - x_i \rvert$
3	$\sqrt{\sum_{i=0}^{N} x_i^2}$
4	$\sum_{i=0}^{N} x_i$
5	$Max_{i=0}^{N} x_i$
6	$Max_{i=0}^{N} x_i$- $Min_{i=0}^{N} x_i$
7	i so that $x_i = Min_{i=0}^{N} x_i$
8	i so that $x_i = Max_{i=0}^{N} x_i$
9	i so that $x_i = Min_{i=0}^{N} x_i$ without first harmonic
10	i so that $x_i = Max_{i=0}^{N} x_i$ without first harmonic
11	Amplitude of the first harmonic
12	Phase of the first harmonic
13	Amplitude of the second harmonic
14	Phase of the second harmonic
15	Amplitude of the third harmonic
16	Phase of the third harmonic
17	Amplitude of the fourth harmonic
18	Phase of the fourth harmonic

- Use many functional combinations to produce thousands of features

- Select the features that do the job you want

- Method has been demonstrated on texture recognition, monitoring the level of use of a car park, face recognition, etc

Advantages:

• A tracing line can be made up from points that do not come from a regular grid!

• Working with tracing lines allows us to escape from the need of regular grids!

• In practice: We may use the Hough transform to identify which sampling points form lines and use them in the trace transform.

• We may use the process to reverse engineer the human vision system!

Which features do the humans use to group textures in perceptual classes?

Make sure you do not bias the answer!!!

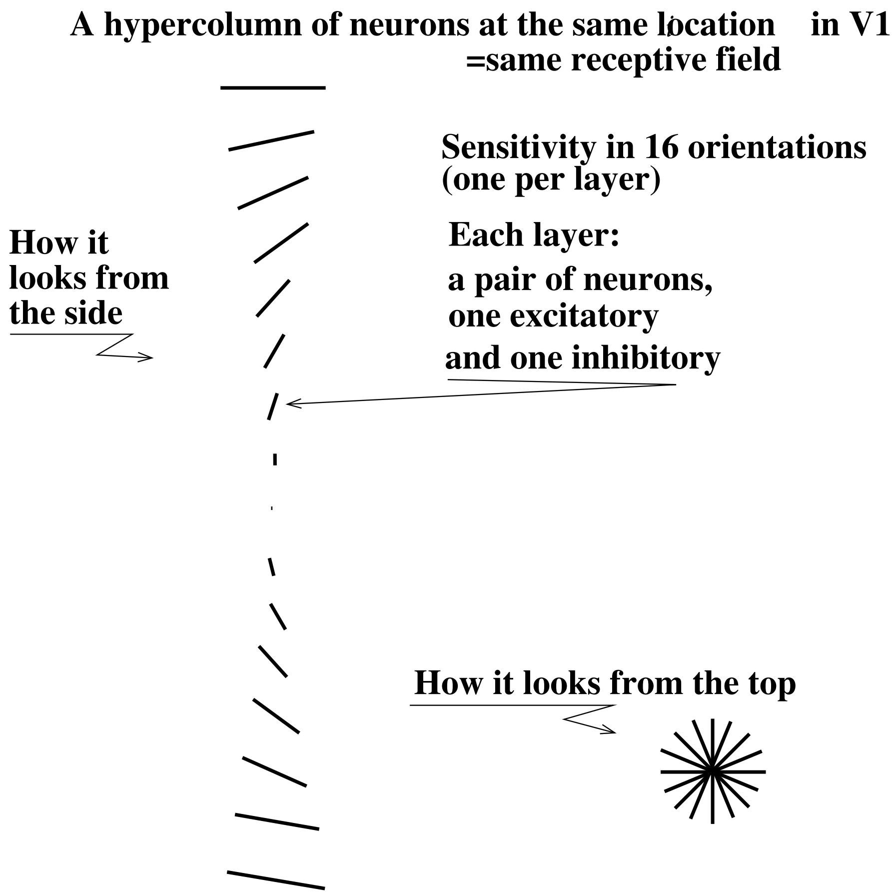
- Produce thousand of features
- Select those that rank textures the same way as human do
- Check whether these features will rank a totally new and different set of textures the same way as the humans.
- Thus form an opinion on which features the human brain might use

Combinations of functionals from the tables that produce the same ranking of textures as humans.

T Functional	P Functional	Φ Functional
$\sum_{i=0}^{N-1} \lvert x_{i+1} - x_i \rvert$	$Max_{i=0}^{N} x_i$	$\sum_{i=0}^{N-1} \lvert x_{i+1} - x_i \rvert^2$
$\sum_{i=0}^{N-1} \lvert x_{i+1} - x_i \rvert$	$Min_{i=0}^{N} x_i$	$\sum_{i=0}^{N-1} \lvert x_{i+1} - x_i \rvert^2$
$\sum_{i=0}^{N-1} \lvert x_{i+1} - x_i \rvert$	$Min_{i=0}^{N} x_i$	Amplitude of the fourth harmonic
$\sum_{i=0}^{N-1} \lvert x_{i+1} - x_i \rvert$	$\sum_{i=0}^{N} i x_i$	$\sum_{i=0}^{N-1} \lvert x_{i+1} - x_i \rvert^2$
$\sum_{i=4}^{N-4} \sum_{k=0}^{4} \lvert x_{i-k} - x_{i+k} \rvert$	$\sum_{i=0}^{N-4} \lvert x_i - 4x_{i+1} + 6x_{i+2} - 4x_{i+3} + x_{i+4} \rvert$	Amplitude of the second harmonic

SALIENCY

A model for V1



$$
\begin{aligned}
\frac{dx_{i\theta}}{dt} &= -\alpha_x x_{i\theta} - \sum_{\Delta\theta} \psi(\Delta\theta) g_y(y_{i,\theta+\Delta\theta}) \\
&\quad + J_0 g_x(x_{i\theta}) + \sum_{j\neq i,\theta'} J_{i\theta,j\theta'} g_x(x_{j\theta'}) + I_{i\theta} + I_0 \qquad (49)
\end{aligned}
$$

$$
\frac{dy_{i\theta}}{dt} = -\alpha_y y_{i\theta} + g_x(x_{i\theta}) + \sum_{j\neq i,\theta'} W_{i\theta,j\theta'} g_x(x_{j\theta'}) + I_c \qquad (50)
$$

where

$x_{i\theta}$: the membrane potential of the excitatory cell;

$y_{i\theta}$: the membrane potential of the inhibitory cell;

α_x: the time constant of decay of the excitatory cell;

α_y: the time constant of decay of the inhibitory cell;

$J_{i\theta,j\theta'}$: excitatory synaptic strengths or **horizontal** cortical connections;

$W_{i\theta,j\theta'}$: inhibitory synaptic strengths or **horizontal** cortical connections;

$g_x(x_{i\theta})$: sigmoid-type analogue activation function modelling the firing rate of the excitatory cell;

$g_y(y_{i\theta})$: sigmoid-type analogue activation function modelling the firing rate of the inhibitory cell;

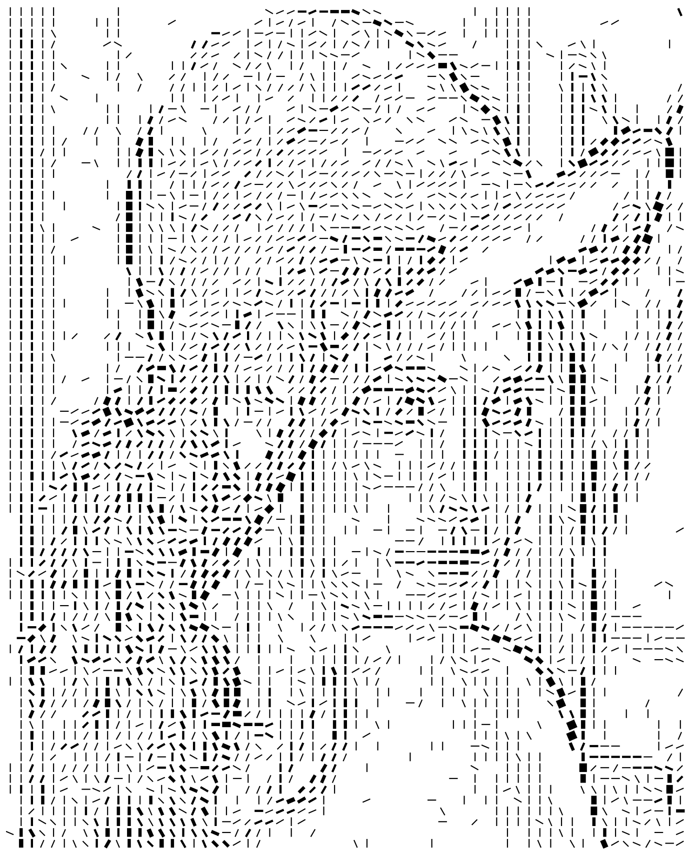
$\psi(\Delta\theta)$: an even function modelling inhibition **within the hypercolumn**;

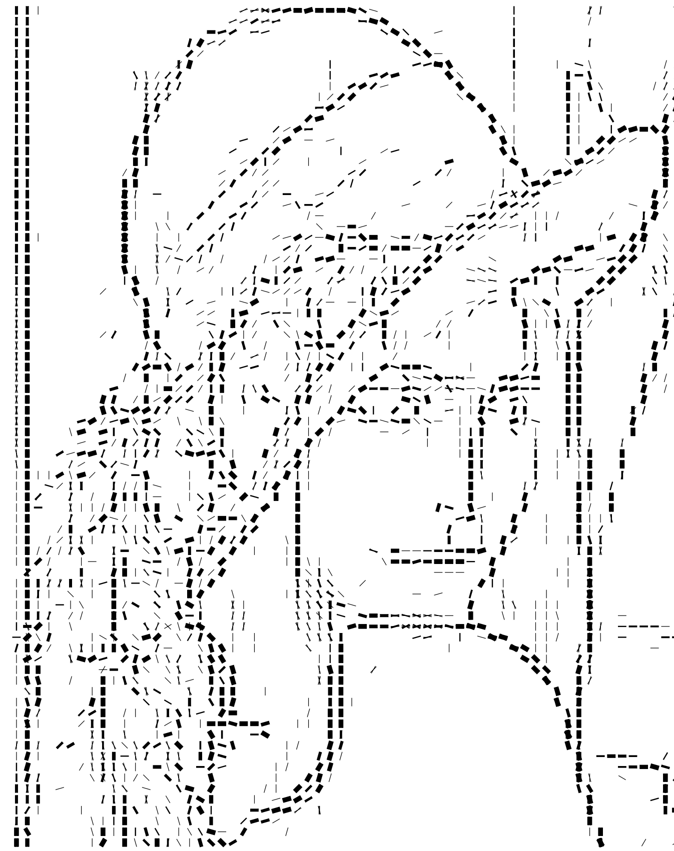
J_0: self-excitatory constant;
I_0: background input to the excitatory cell;
I_c: background input to the inhibitory cell;
$I_{i\theta}$: the external input to the excitatory cell;

$$I_{i\theta} = \hat{I}_{i\beta}\phi(\theta - \beta) \tag{51}$$

where
$\hat{I}_{i\beta}$: the magnitude of the edgel (gradient) detected at i with orientation β;
$\phi(\theta - \beta)$: orientation tuning function between the orientation to which the neuron is sensitive and the orientation of the edgel:

$$\phi(\theta - \beta) = e^{-\frac{|\theta-\beta|8}{\pi}} \tag{52}$$





From V1 to computational economics

A Digital Business Ecosystem(DBE) is a closed or semi-closed system of Small and Medium Enterprises (SMEs), which come together in cyberspace the same way companies gather in a business park in the physical world. These companies will interact with each other through buyer-seller relationships.

How can we model and study the dynamics of such a system?

Health of a company $\Longleftrightarrow$ membrane potential of a neuron

In lack of any external stimulus, it decays like:

$$\frac{dy}{dt} = -\tau y \Rightarrow y = y_0 e^{-\tau t} \tag{53}$$

where

τ: the time constant of the system

y_0: the value of y for the boundary condition $t = 0$.

C_i: company i
y_i: measures how well the company does

- The stronger a company is, the more strong it is likely to become:
This is a self-excitation term, of the form $J_0 g_y(y_i)$.

- Effects in real life are only linear over a certain scale. They saturate and the benefit we receive by changing the independent variable y_i levels off. On the other hand, before this positive feedback in the strength is triggered, a so called "critical mass" of strength y_i has to be reached.
Use sigmoid function $g_y(y_i)$ to model this

A company gets excitatory signals from other companies if they want to buy what it sells

Excitatory signal: Model it according to the number of products E_{ij} company C_i sells that C_j wants to buy:

$$J_{ij} \equiv 1 - e^{-E_{ij}} \tag{54}$$

How much C_j will stimulate C_i depends on how strong C_j is. Include also a weight W_{ij} to model mutual trust:

$$\sum_{j \in C, j \neq i} W_{ij} J_{ij} g_y(y_j) \tag{55}$$

A company gets inhibitory signals from other companies if they sell the same products

Inhibitory signal: Model it according to the number of similar products F_{ij} companies C_i and C_j sell:

$$K_{ij} \equiv 1 - e^{-F_{ij}} \tag{56}$$

How much C_j will inhibit C_i depends on how strong C_j is.

$$-\sum_{j \in C, j \neq i} K_{ij} g_y(y_j) \tag{57}$$

I_i: external input to the company, like total volume of transactions originating outside the DBE

I_0: expresses the background input, eg the general economic climate

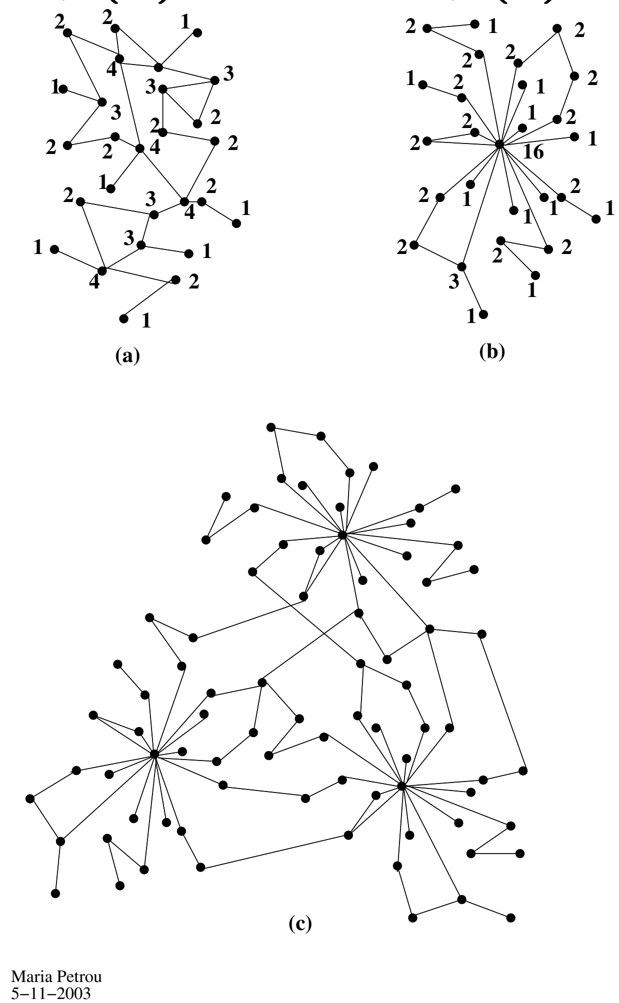
Putting them all together:

$$\frac{dy_i}{dt} = -\tau_y y_i + J_0 g_y(y_i) + \sum_{j \in C, j \neq i} W_{ij} J_{ij} g_y(y_j) - \sum_{j \in C, j \neq i} K_{ij} g_y(y_j) + I_i + I_0 \tag{58}$$

A set of coupled differential equations concerning all companies in the ecosystem.

NETWORKS OF IDEAS

(a)Random; (b)Scale-free; (c)Hierarchical



Characterising a network

Shortest path: the shortest path connecting any two nodes;

mean path the average of the shortest paths calculated between all pairs of nodes in the graph, between which a path exists

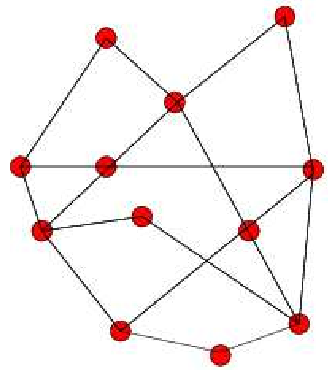
k: the **degree** (or connectivity) of a node;
how many links the node has with other nodes.

$P(k)$: The **degree distribution**;
the probability that a selected node has k links
Networks of different topologies have different degree distributions:

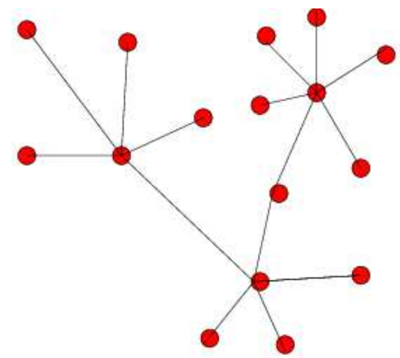
For **scale-free networks**:

$$P(k) \sim k^{-c} \tag{59}$$

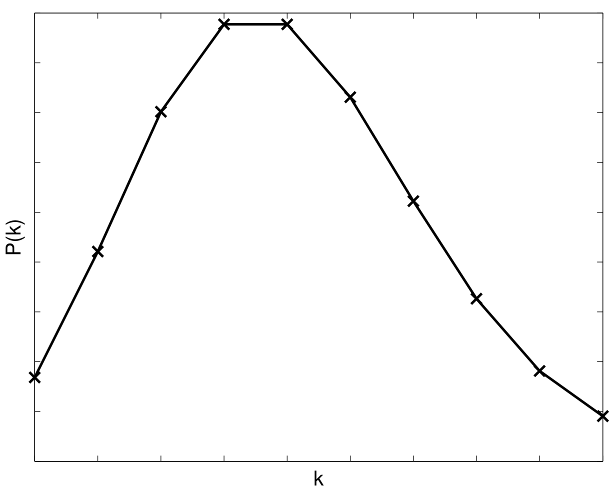
c: characterises the behaviour of the network
$c > 3$: the hubs are not relevant and the scale-free network behaves like a random network;
$2 \leq c \leq 3$: there is a hierarchy of hubs;
$c < 2$: only the most connected hub has a role.



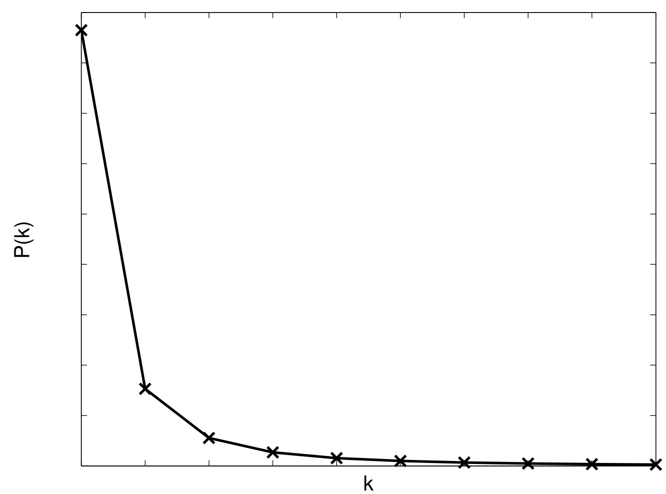
(a) Random network



(b) Scale-free network



(c) Poisson distribution



(d) Power law distribution

Most biological networks, the Internet and social networks have $c \leq 3$, which gives them robustness and a particular resilience to failure.

What is the topology of the network of ideas in our brain?

We designed an experiment to assess the structure of the network of ideas in the human brain

The ideas were poked either visually or verbally

The same ideas in both experiments

The same people took part in both experiments

Conclusions:

- Low mean path between nodes: a small world behaviour of the networks obtained both by visual and verbal cues
- Same number of hubs in both networks
- the two networks were found to be statistically equivalent in topology
- the two networks were consistent with being scale-free with $c \sim 1.7$
- the two networks were appeared to be **distinct** as different concepts acted as hubs in them!

BIBLIOGRAPHY

For irregular data

1)R Piroddi and M Petrou, 2004: "Analysis of irregularly sampled data: a review". Advances in Imaging and Electron Physics, Vol 132, pp 109–165.

2)H Knutsson and C-F Westin, 1993. "Normalized and differential convolution: Methods for Interpolation and Filtering of incomplete and uncertain data", IEEE Conference on Computer Vision and Pattern Recognition, pp 515–523

3)J C Davis, 1973. "Statistics and data Analysis in Geology", John Wiley (for Kriging)

4)T C Haas, 1990. "Kriging and automated variogram modelling within a moving window", Atmospheric Environment, Part A, Vol 24, pp 1759–1769.

5)M Petrou, R Piroddi and A Telebpour, 2006. “Texture recognition from sparsely and irregularly sampled data”. Computer Vision and Image Understanding, Vol 102, pp 95–104.
6)S Chandra, M Petrou and R Piroddi, 2005. “Texture interpolation using ordinary Krigging”. Pattern Recognition and Image Analysis, Second Iberian Conference, IbPRIA2005, Estoril, Portugal, June 7–9, J S Marques, N Perez de la Blanca and P Pina (eds), Springer LNCS 3523, Vol II, pp 183–190.

The Trace transform

1)A Kadyrov and M Petrou, 2001. “The Trace transform and its applications”. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, Vol 23, pp 811–828.
2)M Petrou and A Kadyrov, 2004. “Affine invariant features from the Trace transform”. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-26, pp 30–44.
3)S Srisuk, M Petrou, W Kurutach and A Kadyrov, 2005. “A face authentication system using the trace transform”. Pattern Analysis and Applications, Vol 8, pp 50–61.
4)M Petrou, A Talebpour and A Kadyrov, 2007. “Reverse Engineering the way humans run textures”, Pattern Analysis and Applications, Vol 10 (2) pp 101–114.

The V1 model

1)Z Li, 1998. "A neural model of contour integration in the primary visual cortex", Neural Computation, Vol 10, pp 903–940.
2)Z Li, 1999. "Visual segmentation by contextual influences via intra-cortical interactions in the primary visual cortex", Networks:Computation in Neural Systems Vol 10 pp 187–212
3)Z Li, 2001. "Computational design and nonlinear dynamics of a recurrent network model of the primary visual cortex", Neural Computation, Vol 13(8), pp 1749–1780.
4)M Petrou, S Gautam and K N Giannoutakis, 2006. "Simulating a Digital Business Ecosystem", Computational Finance and its Applications II, WIT Press, M Costantino and C A Brebbia (eds), ISBN 1-84564-1744, pp 277–287.

Networks

1) L Barabasi, 2002. "Linked: The New Science of Networks", Perseus Publishing.

2)M Petrou and R Piroddi, 2006. "On the structure of the mind", Proceedings of AISB'06: Adaptation in Artificial and Biological Systems, T Kovacs and J Marshall (eds), Vol 2, pp 60–63.