Digital image processing and analysis
12. Object properties: shape and texture

Professor Ela Claridge
School of Computer Science

Previous lecture:

- How to get coordinates of the object boundaries
- How to count objects in a segmented image
- How to measure objects
- How to measure object locations
- How to measure some object properties

In this lecture we shall find out about:

- Further object properties – descriptors and methods
- Shape
  - Outlines
  - Geometric features (e.g. curvature)
  - Fractal dimension
- Texture
  - Grey level statistics
  - Co-occurrence matrices
- From segmented images to segmented object properties

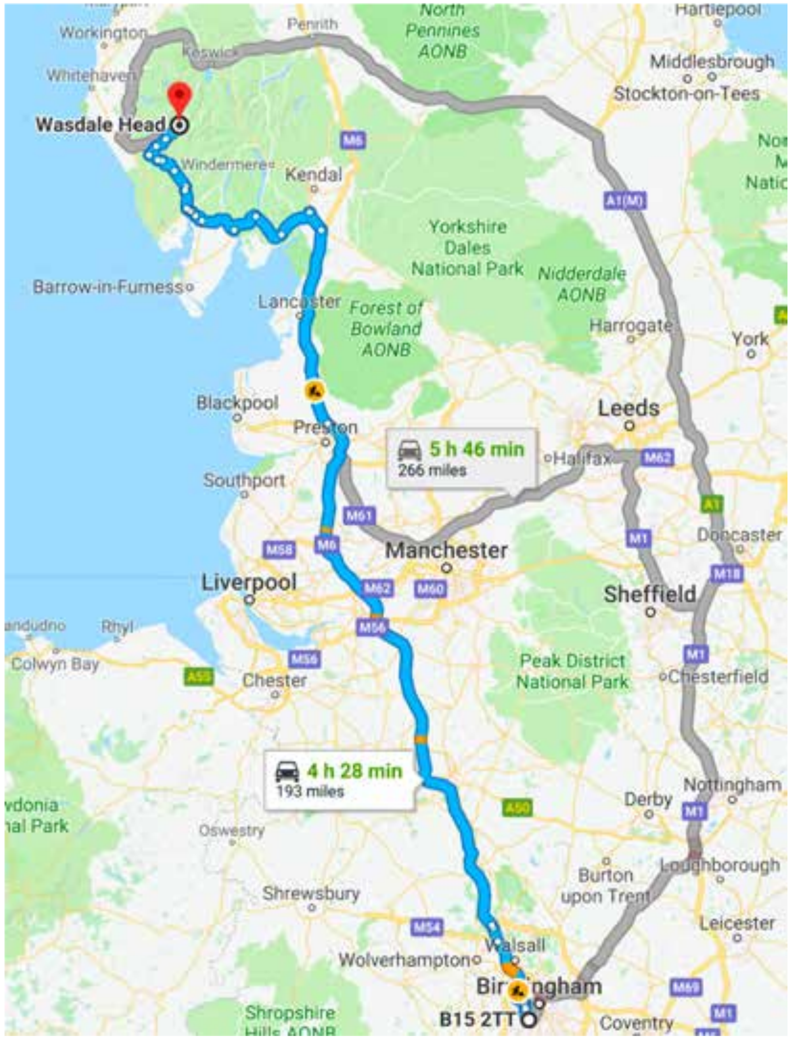
Shape
Polylines for open curves

Vectorisation

Objective:
Generate a compressed representation of a curved line.

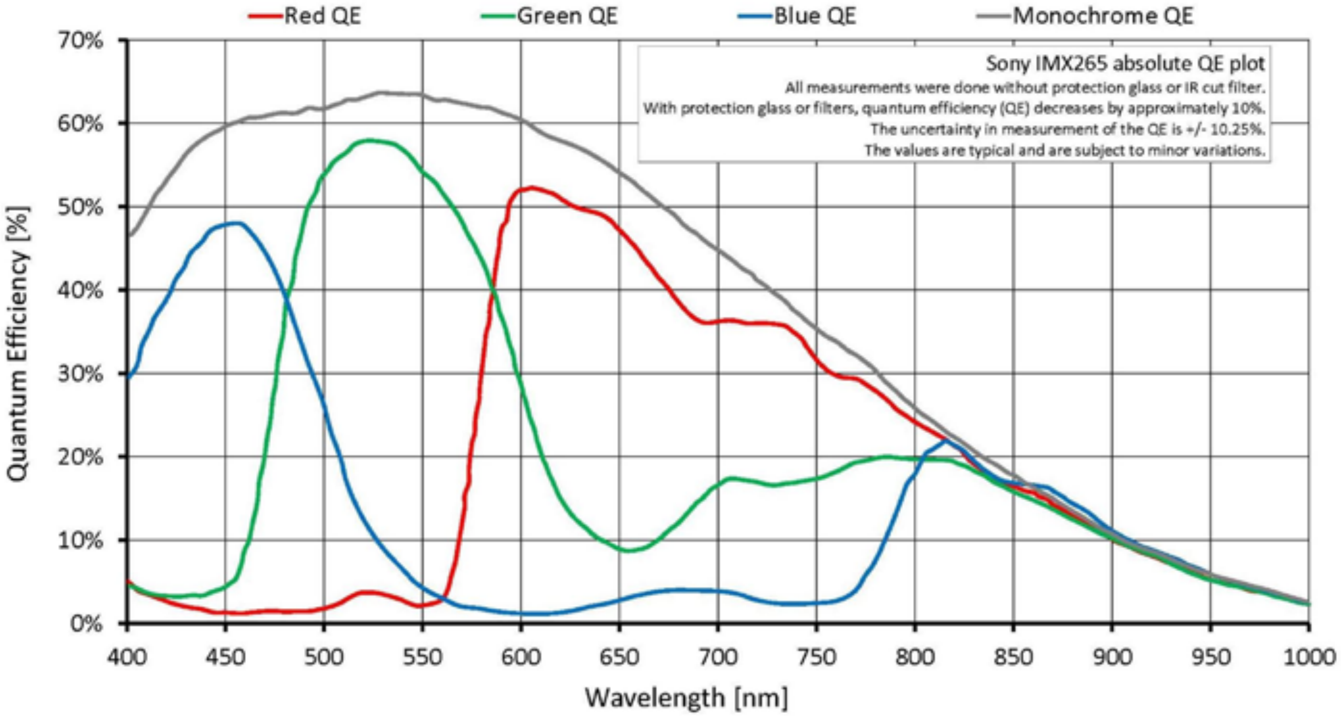
Method:
Given a set of (x,y) coordinates of every point on a line, fit a number of line segments which approximate the curved line.

Application:
Vectorised maps for SatNav



Shape
Polylines for open curves

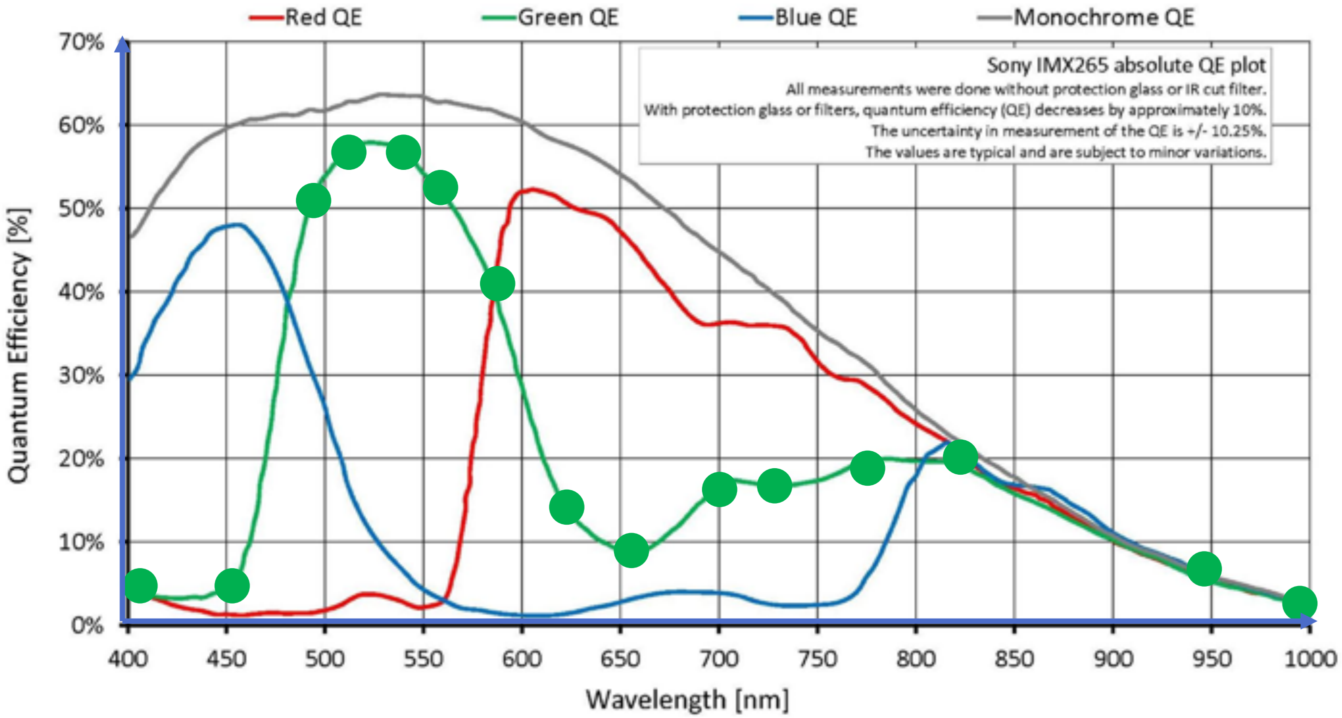
Vectorisation



Application:
Vectorised camera response curves

Shape
Polylines for open curves

- A curve is represented by a list of endpoints of line segments approximating nearly-straight runs of the original curve

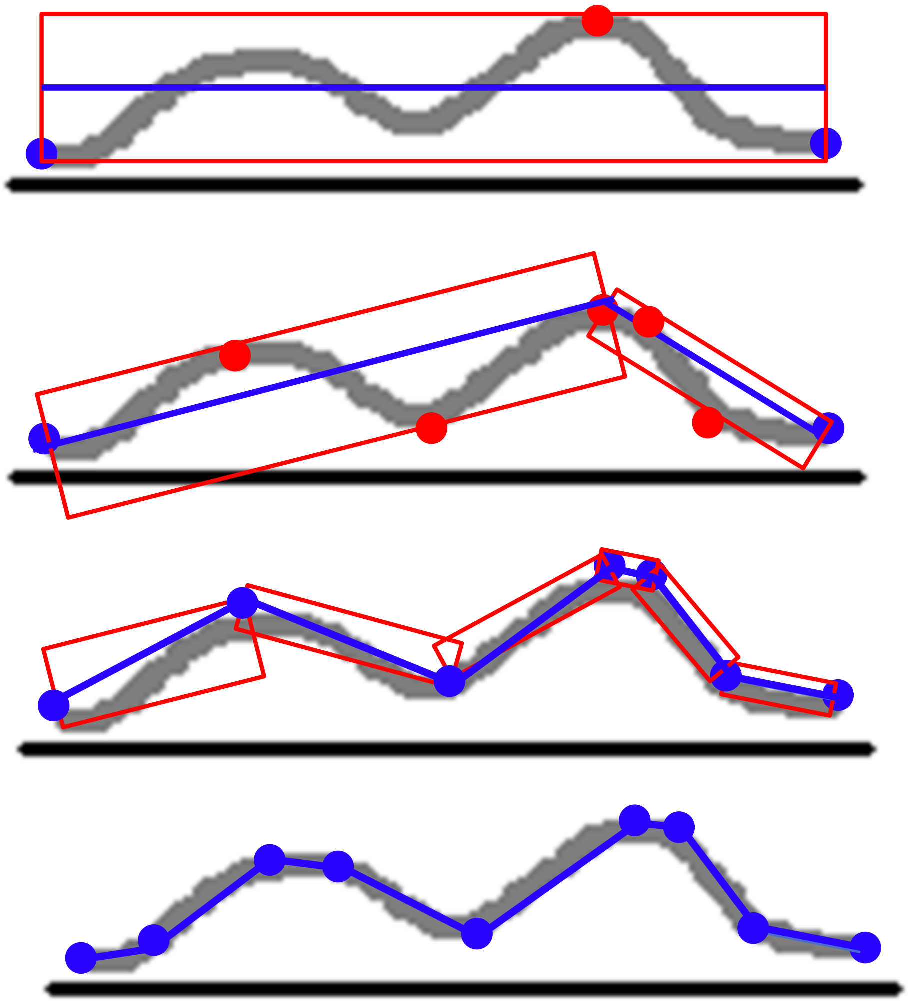


Reduction from 600 points to 15 points:

Factor of 40.

Shape
Polylines for open curves

Iterative (sequential) process



Shape
Polylines for open curves

Algorithm

- Initialise
  - Define the minimum width of the bounding box (the tolerance band)
  - Start from the entire curve being a single segment
- Iterate (repeat) for each segment
  - Find and save the curve endpoints
  - Find the oriented bounding box
  - If the width of the bounding box is smaller of equal than the tolerance band
    - Terminate
  - Otherwise
    - Find the point on the curve at the maximum distance from the base of the bounding box
    - Divide the curve into two segments at this point
    - Continue iterating

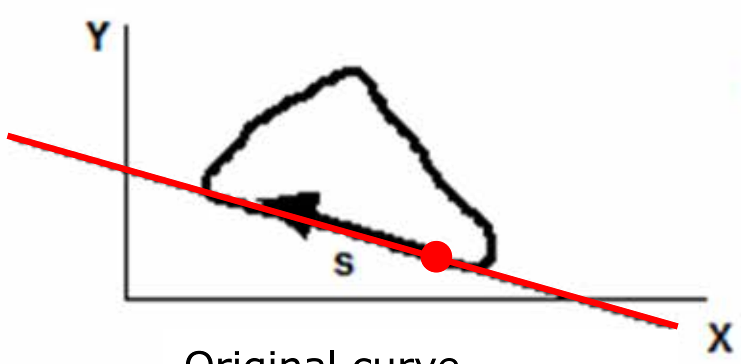
Shape
Ψ-s curves

How many cakes of each shape are there?

1. Triangular
2. Round
3. Square

Shape
s-Ψ curves

- s-Ψ curve describes the angle between a fixed line and a tangent to the boundary at each point of the original boundary
- Horizontal lines on s-Ψ curve correspond to straight lines in the original boundary
- Non-horizontal lines correspond to arcs of a circle (Ψ is changing at a constant rate)
- s-Ψ curve can be segmented to yield a description of the boundary in terms of line segments and arcs



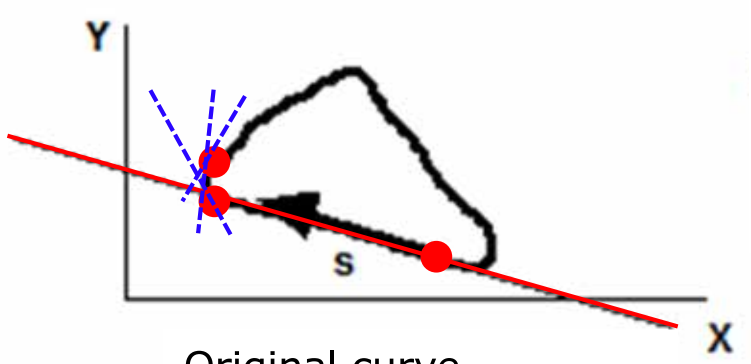
Original curve

s-Ψ curve

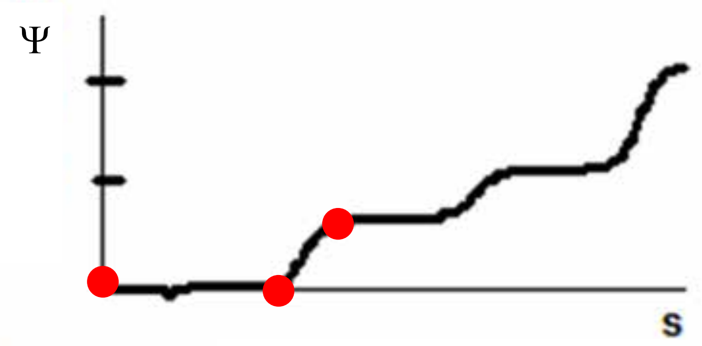
Shape

s-Ψ curves

- s-Ψ curve describes the angle between a fixed line and a tangent to the boundary at each point of the original boundary
- Horizontal lines on s-Ψ curve correspond to straight lines in the original boundary
- Non-horizontal lines correspond to arcs of a circle (Ψ is changing at a constant rate)
- s-Ψ curve can be segmented to yield a description of the boundary in terms of line segments and arcs



Original curve



Ψ -s curve

Shape

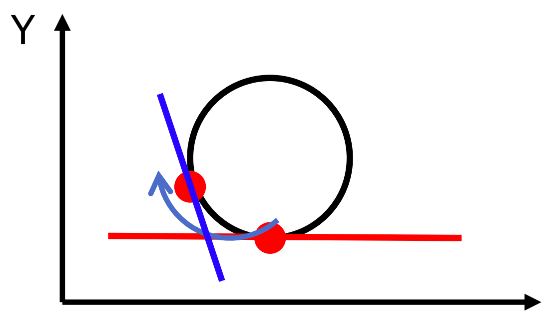
s-Ψ curves

Algorithm

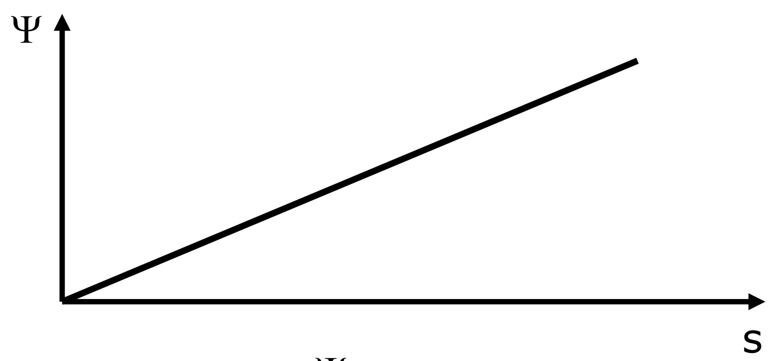
- Initialise
  - Find a starting point on the contour (any point), s_0
  - Find the line L_0 tangential to the contour at point s_0
- For every point on the contour
  - Move to the next point, s_i
  - Find the line L_i tangential to the contour at point s_i
  - Measure and save the angle Ψ_i between L_0 and L_i
  - Terminate when the current point equals the starting point

Shape
s-Ψ curves

- s-Ψ curve describes the angle between a fixed line and a tangent to the boundary at each point of the original boundary
- Horizontal lines on s-Ψ curve correspond to straight lines in the original boundary
- Non-horizontal lines correspond to arcs of a circle (Ψ is changing at a constant rate)
- s-Ψ curve can be segmented to yield a description of the boundary in terms of line segments and arcs



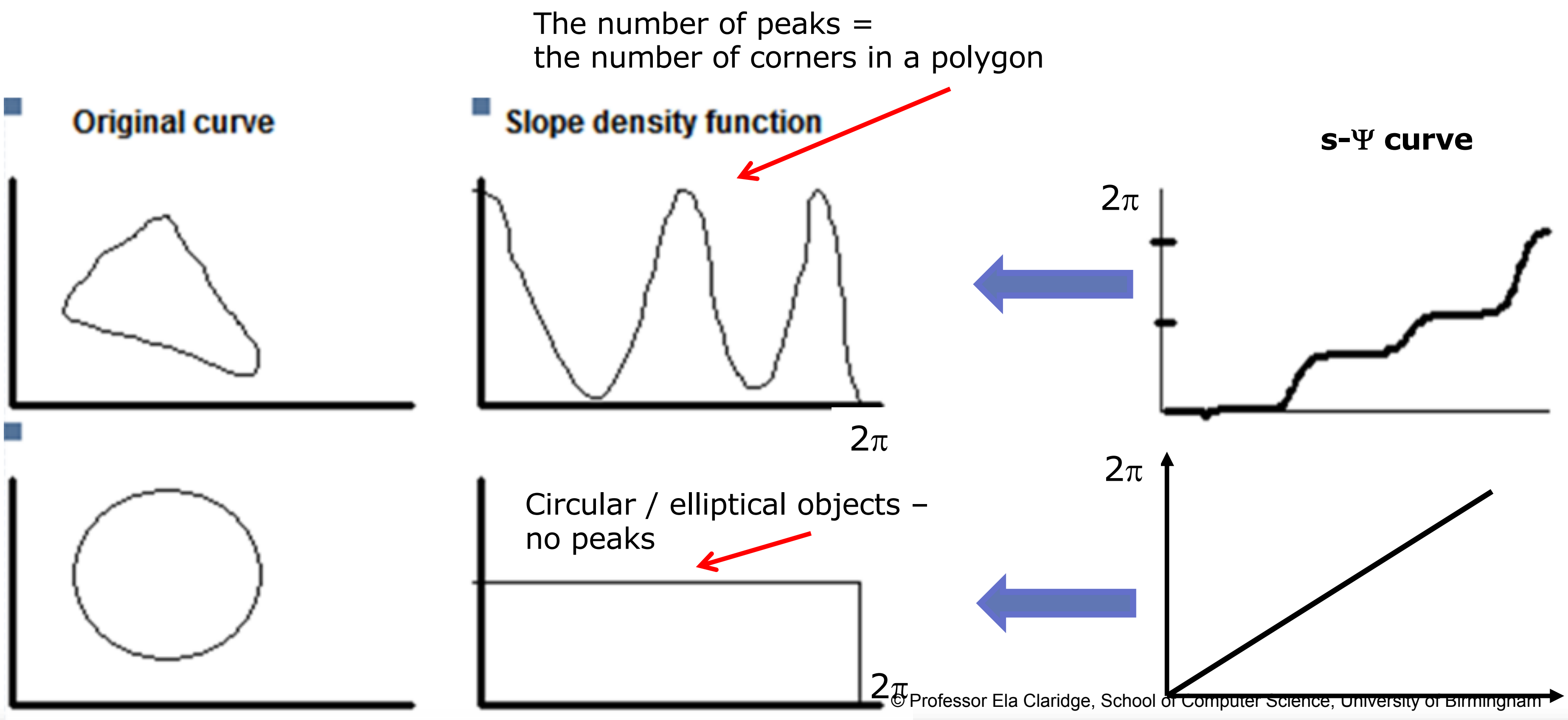
Original curve



s-Ψ curve

Shape
Slope density function

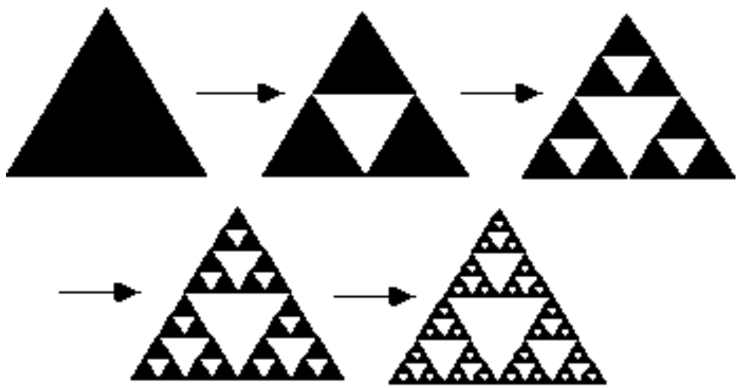
- Slope density function is a histogram of Ψ collected over the length of the boundary s.
- Derived from s-Ψ curve.



Shape
Fractal dimension

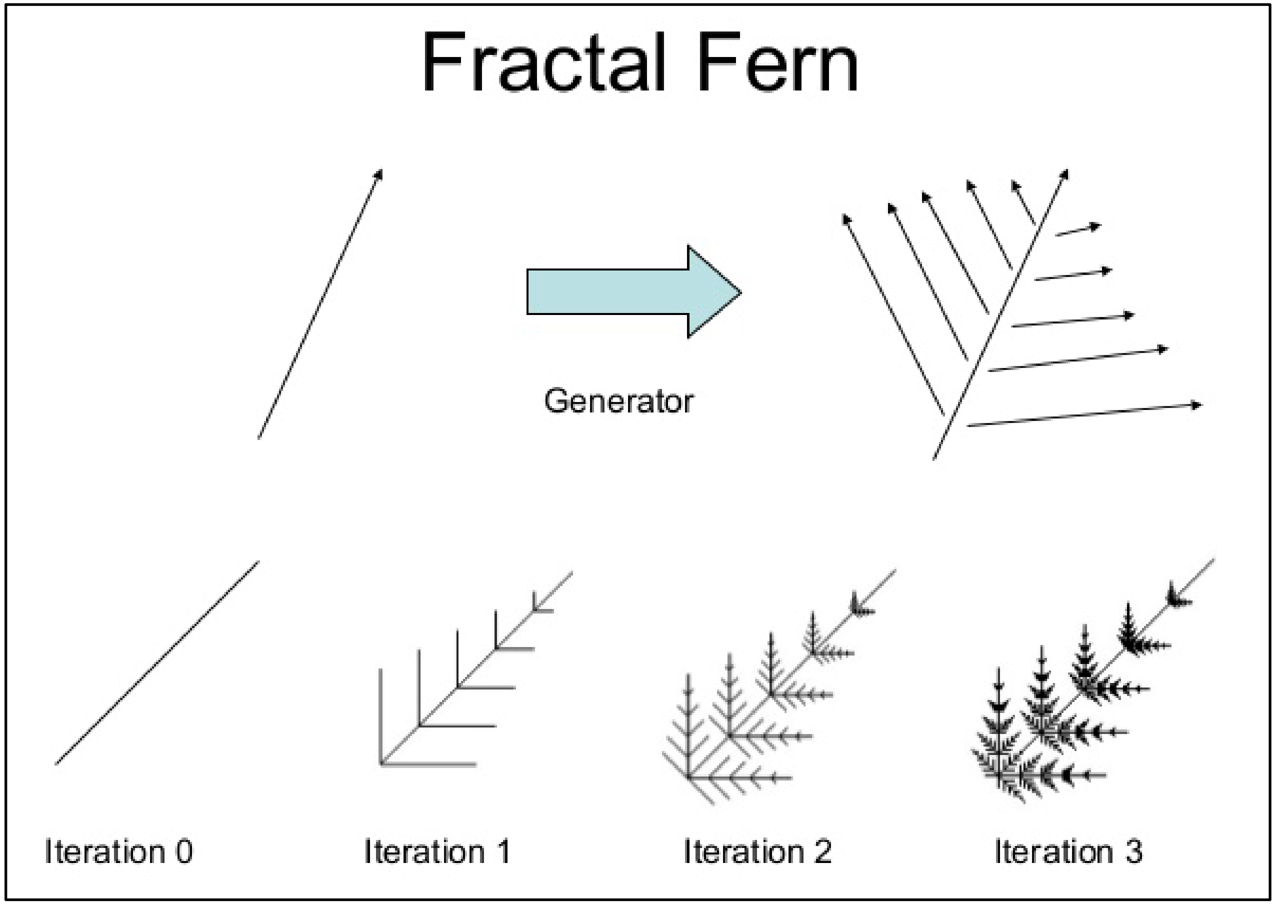
- A term coined by Mandelbrot
- Traditional view of geometry: curves are one-dimensional, plane figures are two-dimensional
- An insight: there are geometric structures which have dimensions between one and two (and three)
- In two-dimensional space fractal dimension characterises the plane-filling ability of a curve

Natural fractal object



Mathematical fractal object
Sierpinski triangle

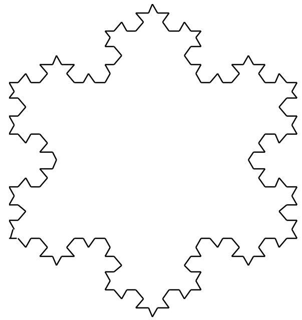
Shape
Fractal dimension



https://www.slideshare.net/YogeshjatinGupta/fractal-27497043

Shape
Fractal dimension

- Fractal dimension of a circle: 1
- Fractal dimension of a disk: 2
- Fractal dimension of a "Koch snowflake": 1.26



- Fractal dimension is derived by using different step lengths to measure the perimeter of the figure.
- The estimated perimeter increases as the step length decreases.

Shape
Fractal dimension

Question: How long is a coast of Great Britain?
Answer: It depends on the length of the ruler used for measurements

Unit = 200 km, length = 2400 km

Unit = 50 km, length = 3400 km

https://en.wikipedia.org/wiki/Fractal_dimension

Shape
Fractal dimension: measurement

THEORY – for interest only

- The estimated perimeter length L measured with the “ruler” of a given size:
- $$L(s) = s \cdot N(s)$$
  - s - length of the “ruler”
  - N(s) - the number of sides of length s of a polygon which approximates the perimeter
- Mandelbrot made an important observation that the number of polygon sides for a given perimeter is a function of the step length (s) and expressed this relationship as:

$$N(s) = \lambda \cdot s^{-D}$$

giving the final expression for the estimated perimeter length:

$$L(s) = \lambda \cdot s^{1-D}$$

Shape
Fractal dimension: measurement

THEORY – for interest only

- Estimated perimeter length:

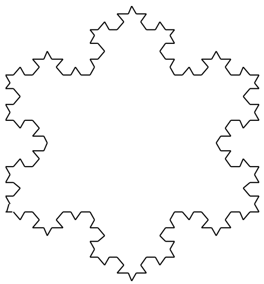
  $$L(s) = \lambda \cdot s^{1-D}$$

- By taking logarithm of both sides a line equation is obtained

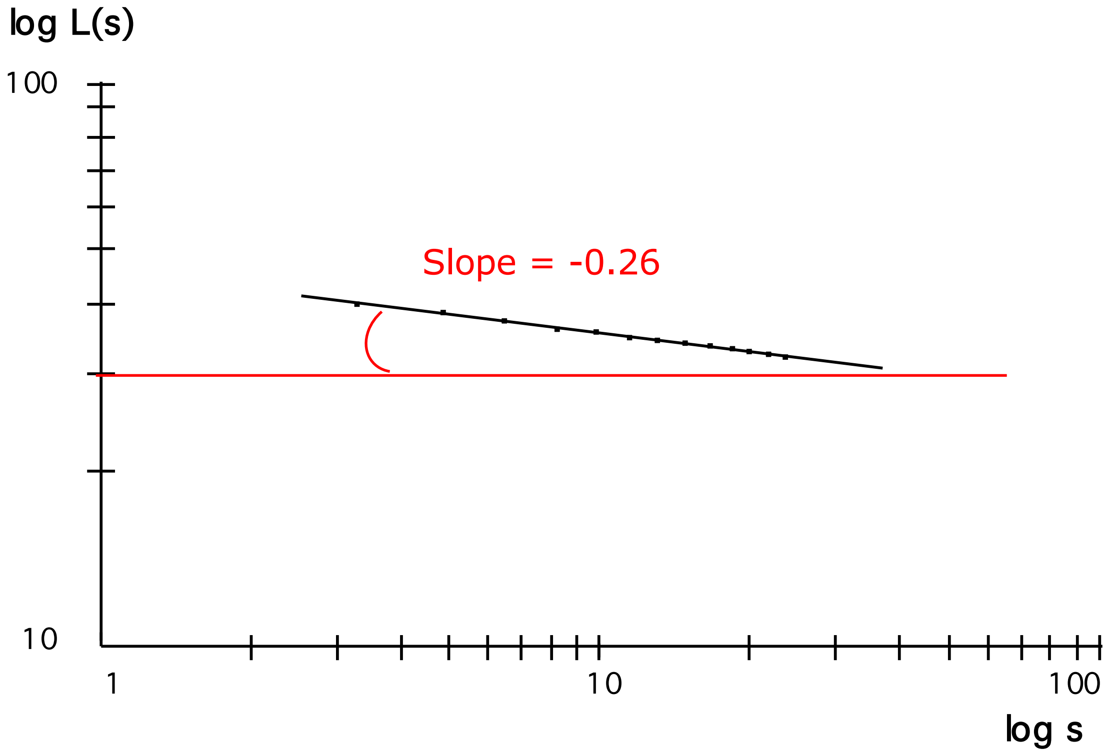
- $$\log L(s) = \log \lambda + (1 - D) \cdot \log s$$
-
  - $(1 - D)$ is the slope of the line
  - $\log \lambda$ is the intercept.
  - D is the fractal dimension

Shape
Fractal dimension: measurement

For interest only

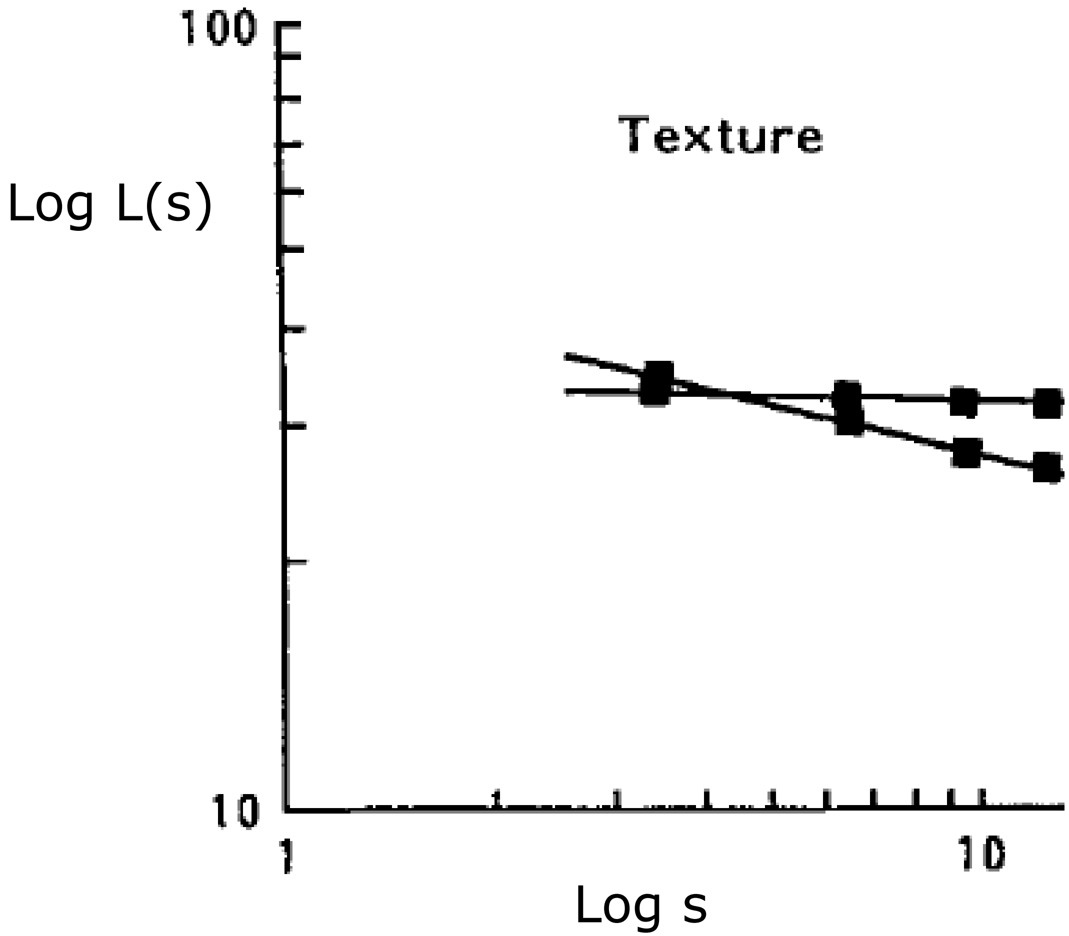


Fractal dimension =1.26



Shape
Fractal dimension: examples

For interest only



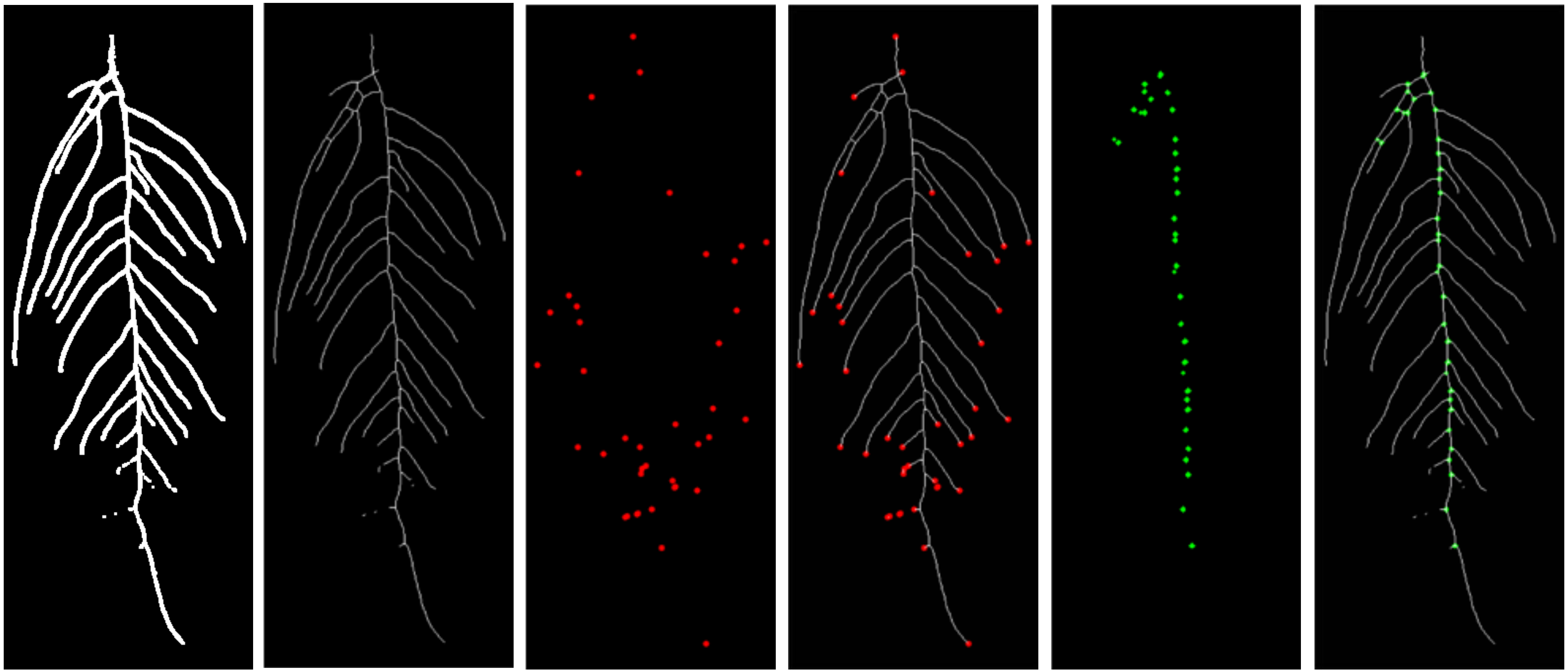
FD=1.025

FD=1.167

Shape
Endpoints and crossings

Analysis of relationships between linear features

Mathematical morphology: Hit-and-miss operations

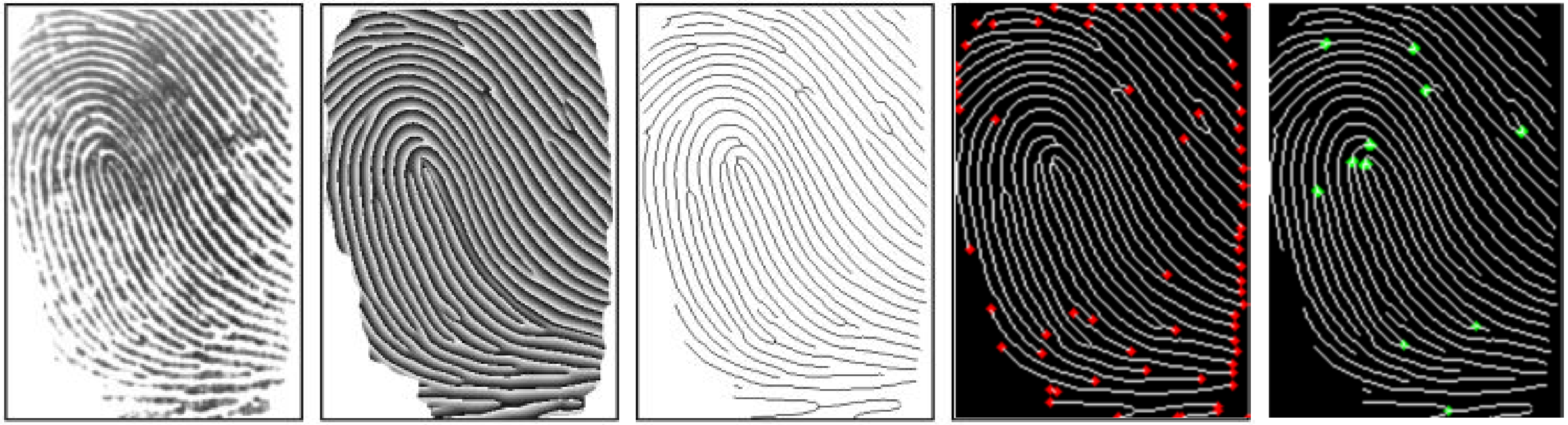


Root Skeleton Endpoints found Crossings found

Shape
Endpoints and crossings

Analysis of relationships between linear features

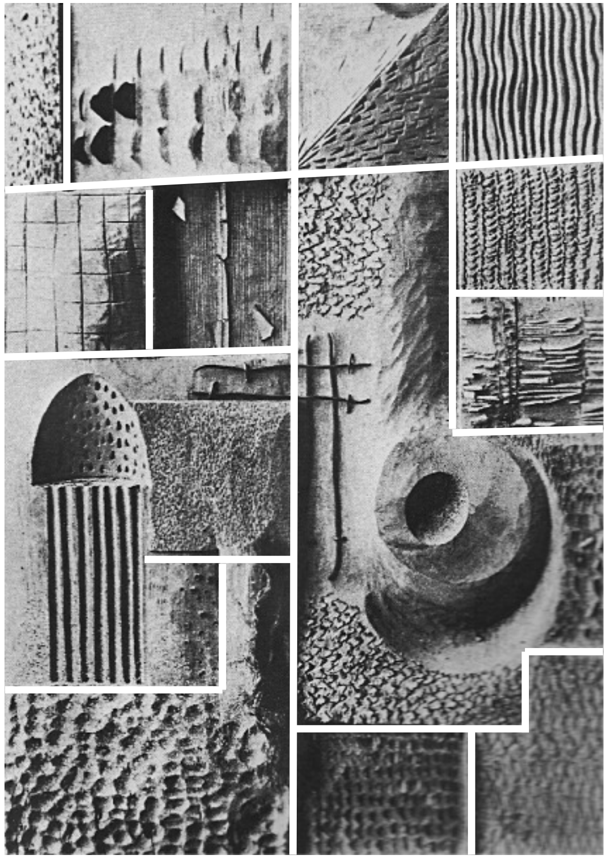
Mathematical morphology: Hit-and-miss operations



Fingerprint Segmentation Skeleton Endpoints found Crossings found

Application: detection of minutia for fingerprint recognition

Texture



Similar colours, different spatial arrangement of pixels

Texture

- Texture is characterised by the ***spatial inter-relations*** between pixel values.

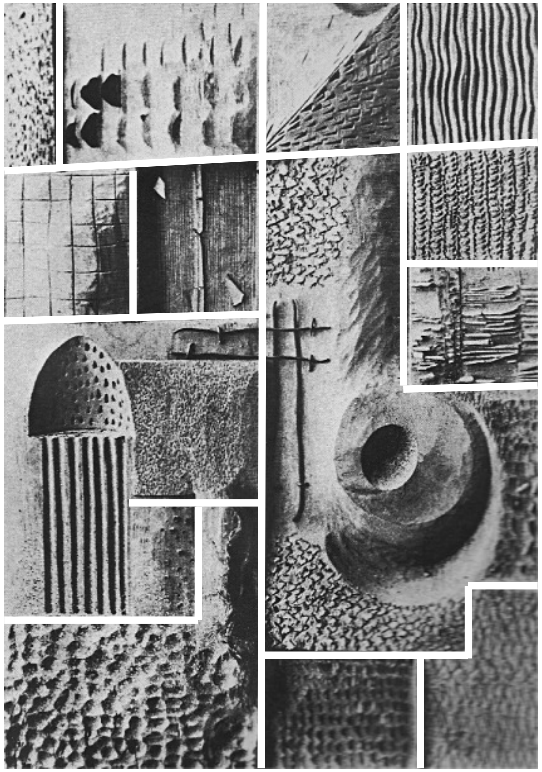
- Colour v/s texture

  - *Colour (including grey level)* properties depend only on pixel values and are independent of their spatial distribution within the area.
  - ***Texture*** properties relate to the way the pixels with various grey levels are spatially arranged within the area.
  - They depend both on grey level value and on the spatial distribution of grey levels.

Texture

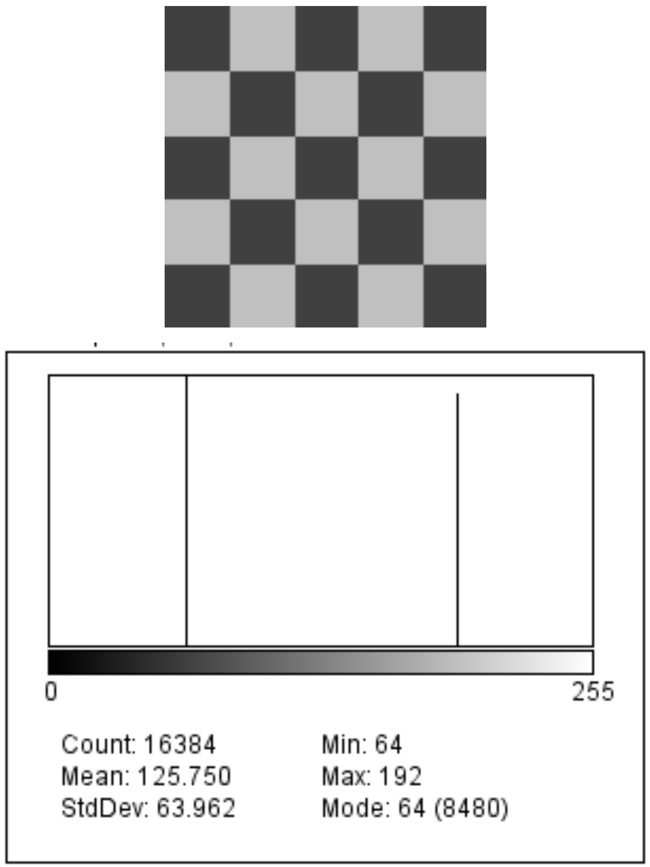
- Simple (first order) statistics (mean, standard deviation, etc.) are measures of *overall* variation in contrast and therefore tell nothing about spatial distribution.

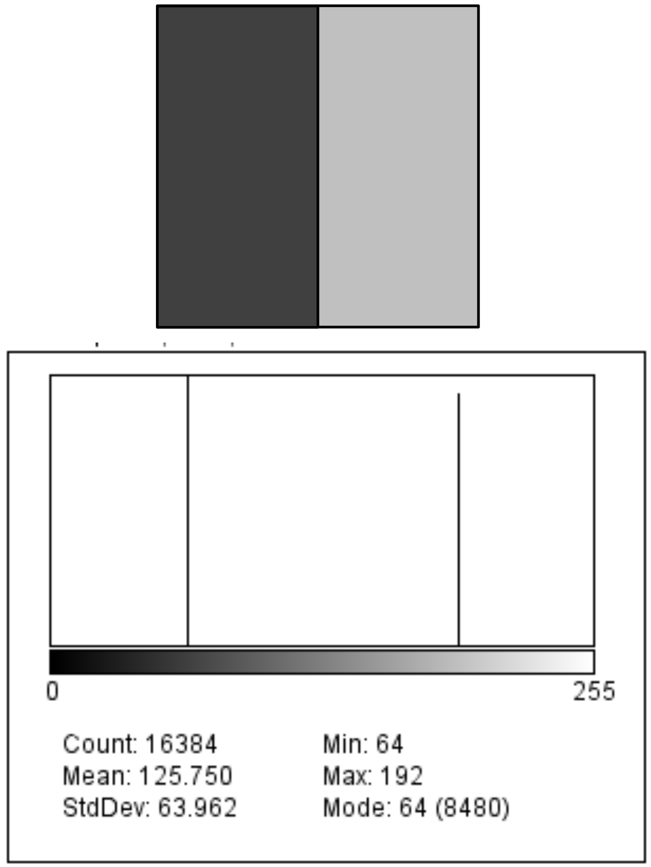
- **Second order statistics** consider more complex statistical properties involving sets of pixels in combination.
- The local properties characterise distribution and relationships among the pixels of various grey levels.
- Measures derived from these statistics describe various texture attributes, such as coarseness, homogeneity and contrast.



From "Image enhancement 1" lecture
Histogram

Two images with the same **statistical distribution** of pixel values but with different **spatial distribution** of pixel values



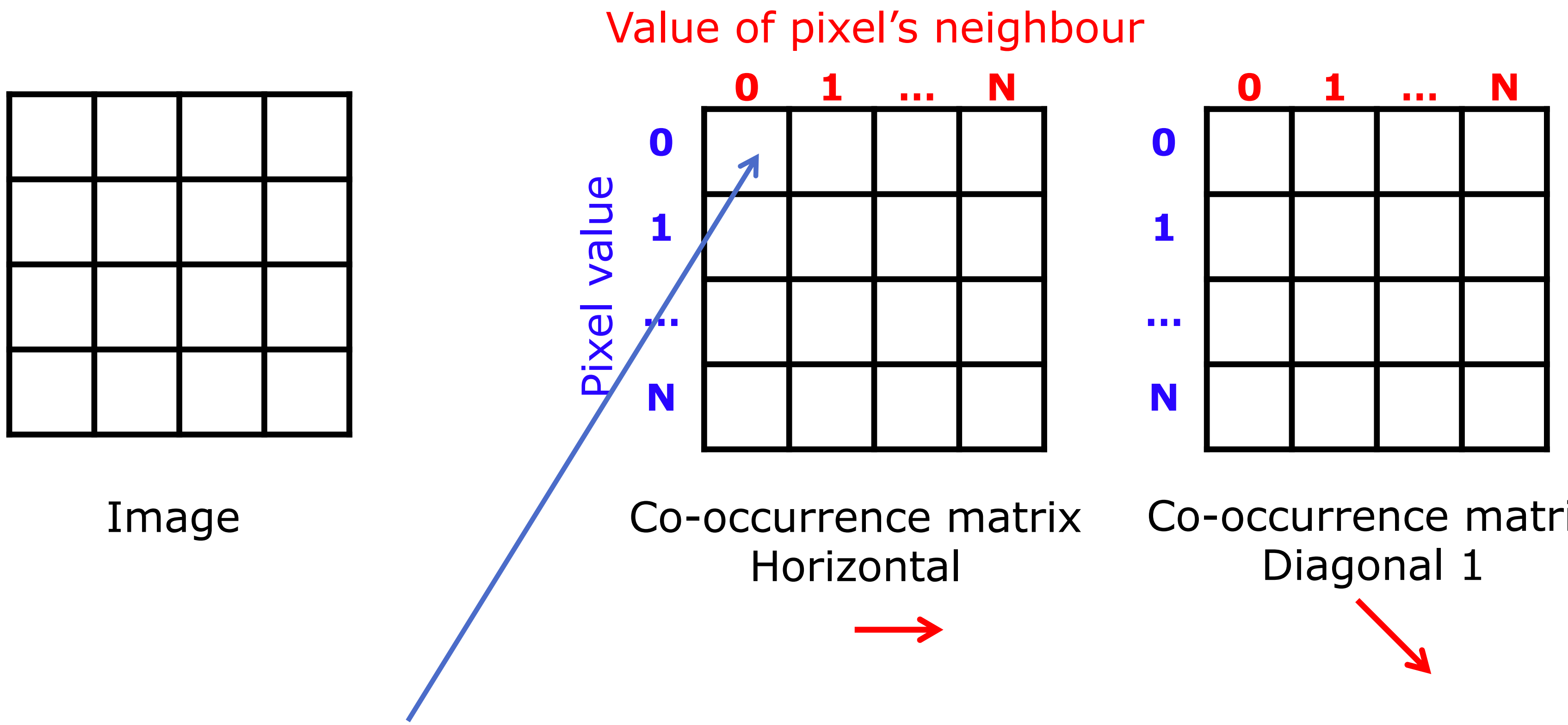


Identical histograms

Texture
Co-occurrence (spatial dependency) matrix

- A **co-occurrence matrix** is a two-dimensional histogram:
  - axis 1: grey level of a pixel
  - axis 2: grey level of the pixel's neighbour
- The position (*i,j*) in the histogram stores the number of times that a pixel with value *i* has as its neighbour a pixel with value *j* at a given distance and for a given angle.
- For each image a number of co-occurrence matrices can be generated
  - For different distances
  - For different directions
- Texture measures are then derived from these matrices.

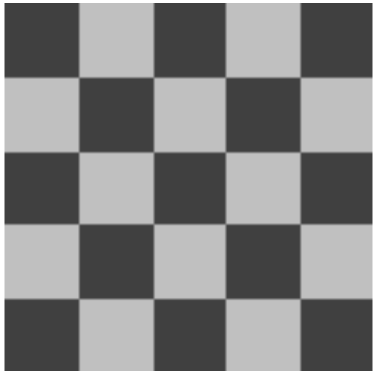
Texture
Co-occurrence matrix



Number of times pixel with value **i** has as its neighbour pixel with value **j**

Texture
Co-occurrence matrix

Assume image has 4 grey levels, 0-3



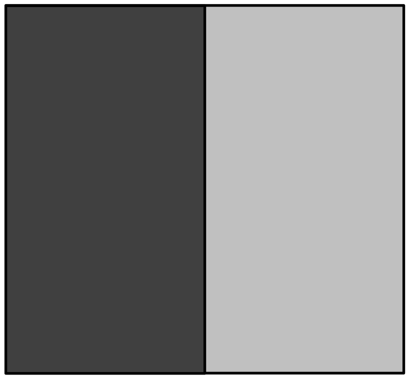
0	3	0	3
3	0	3	0
0	3	0	3
3	0	3	0

Co-occurrence matrix →

	0	1	2	3
0	0	0	0	6
1	0	0	0	0
2	0	0	0	0
3	6	0	0	0

Co-occurrence matrix ↘

	0	1	2	3
0	5	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	4



0	0	3	3
0	0	3	3
0	0	3	3
0	0	3	3

	0	1	2	3
0	4	0	0	4
1	0	0	0	0
2	0	0	0	0
3	0	0	0	4

	0	1	2	3
0	3	0	0	3
1	0	0	0	0
2	0	0	0	0
3	0	0	0	3

Image

Pixel values

Co-occurrence matrix Horizontal

Co-occurrence matrix Diagonal 1

Texture
Computing texture features

Co-occurrence matrix – **probabilistic representation**

Image

0	3	0	3
3	0	3	0
0	3	0	3
3	0	3	0

Co-occurrence matrix

	0	**1**	**2**	**3**
0	0	0	0	0.5
1	0	0	0	0
2	0	0	0	0
3	0.5	0	0	0

Co-occurrence matrix

	0	**1**	**2**	**3**
0	0.4	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0.3

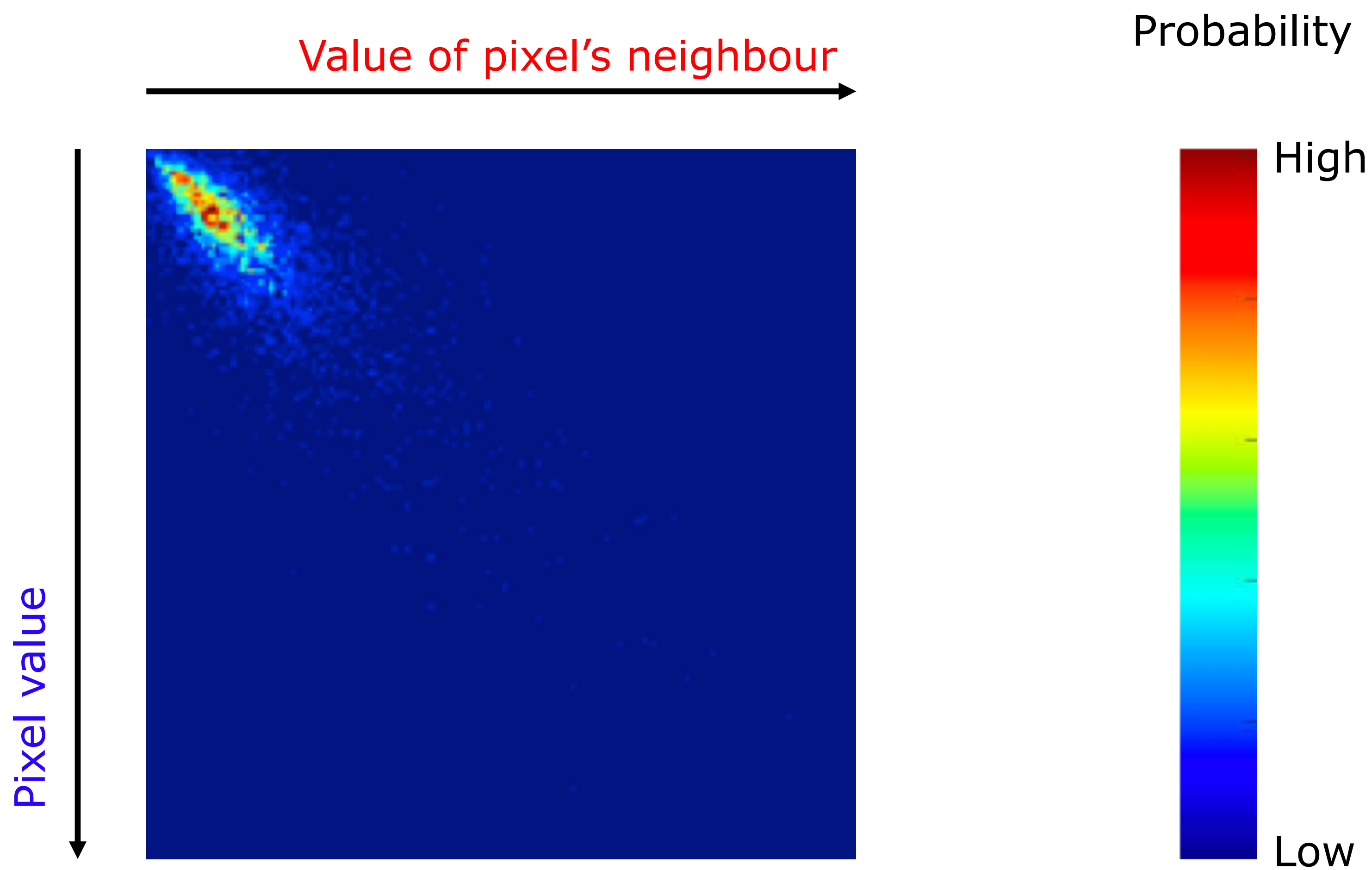
0	0	3	3
0	0	3	3
0	0	3	3
0	0	3	3

	0	**1**	**2**	**3**
0	0.3	0	0	0.3
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0.3

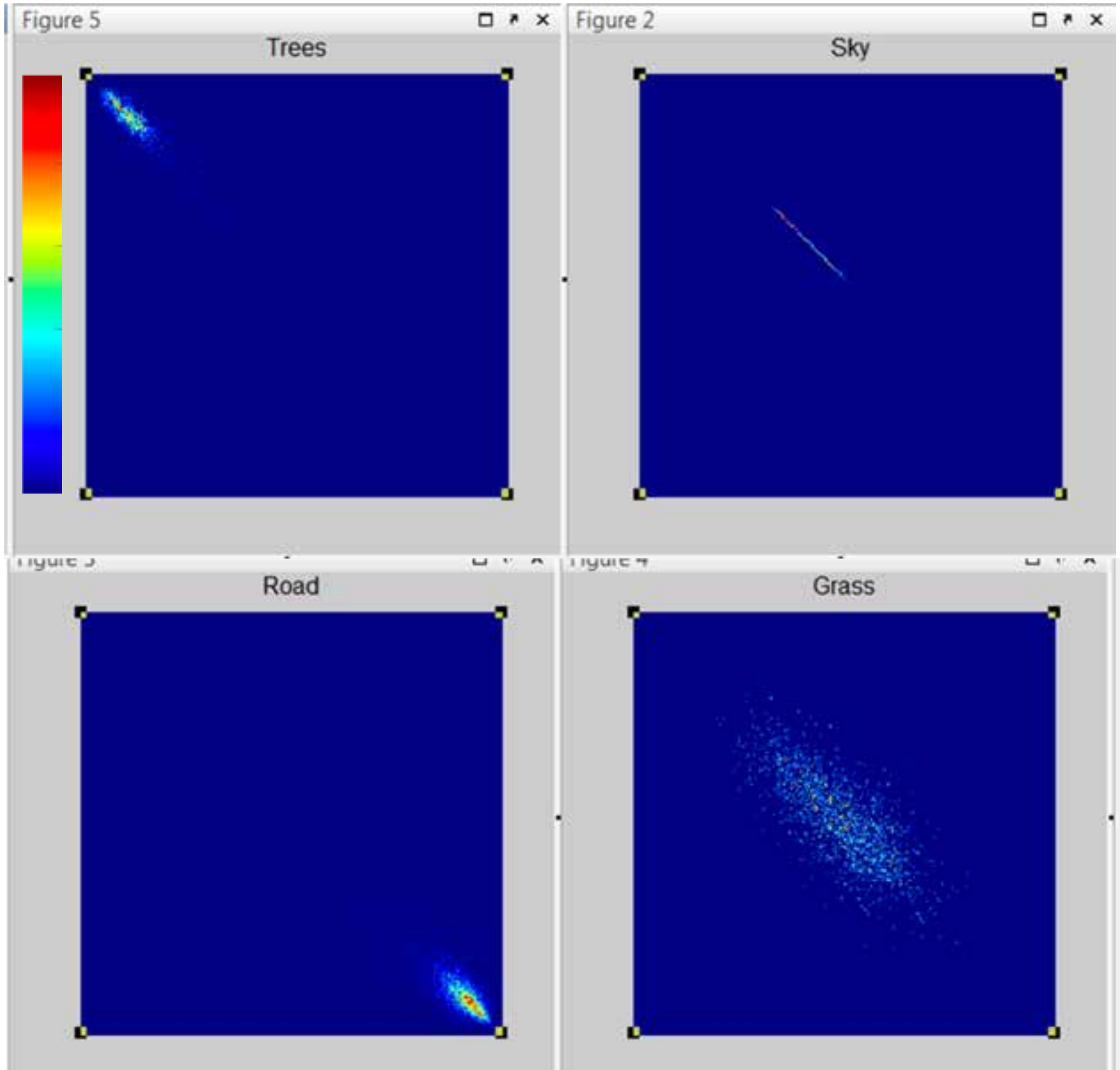
	0	**1**	**2**	**3**
0	0.25	0	0	0.25
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0.25

No. of co-occurrences for the range of pixels from 0 to 3: **6** (see previous slide)
Total number of co-occurrences in the image: **12** (4 rows * 3 columns)
Probability 6/12 = 0.5

Texture
Co-occurrence matrix



Texture
Co-occurrence matrix



Texture
Computing texture features

THEORY – for interest only

HOMOGENEITY
$$= \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\{P(i,j)\}^2$$

CONTRAST
$$= \sum_{n=0}^{G-1} n^2\{\sum_{i=1}^{G}\sum_{j=1}^{G}P(i,j)\}, \quad |\, i-j \,|= n$$

ENTROPY
$$= -\sum_{i=0}^{G-1}\sum_{j=0}^{G-1} P(i,j)\times log(P(i,j))$$

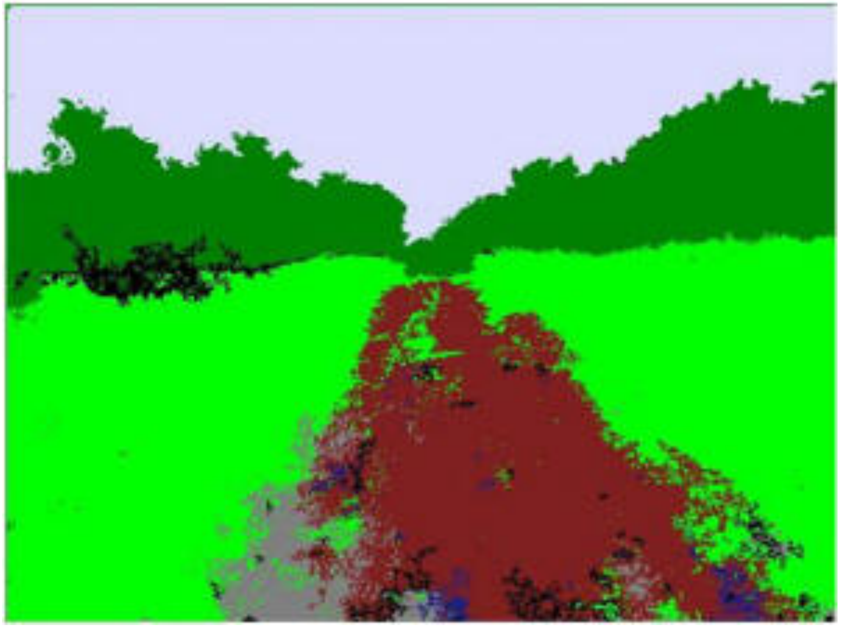
G – total number of grey levels
i,j – pixel greylevel value (i) and greylevel value of its neighbour (j)
P(I,j) – probability that pixel with greylevel value i has as its neighbour pixel with greylevel value j

Texture

Computing texture features

	Hom	Con	Ent
Sky	0.019	0.28	4.19
Trees	0.003	33.93	6.02
Grass	0.0007	286.8	6.94
Road	0.002	74.20	6.62

Texture
Computing texture features



Images can be segmented on the basis of their texture features

In this lecture we have covered:

- Object properties – methods and descriptors
- Shape
  - Outlines
  - Geometric features
- Texture
  - Grey level statistics
  - Co-occurrence matrices
- From segmented images to segmented object properties

Next lecture:

- Image registration: what it is and what it is for
- What we try to align
  - Image features
  - Image values
- How we carry out image transformations
  - Rigid and affine
  - Elastic
- How we measure the success of registration
- How we compute the final pixel values

Further reading and experimentation

- **Book chapters**:
- Gonzalez, R.C. & Woods, R.E. Digital Image Processing, Addison-Wesley (various editions), 7.4.2, 8.1 – 8.3.
- Sonka, M. Hlavac, V. Boyle, R. (various editions) Image Processing, Analysis and Machine Vision, Chapman & Hall Computing, 6.2-6.3 (shape), 13 (texture).
- Umbaugh, S.E. Computer vision and image processing : a practical approach using CVIPtools , Prentice Hall International (various editions), 2.6.

- **Skeleton: endpoints and crossings**
- https://crazybiocomputing.blogspot.co.uk/2013/02/hit-or-miss-in-imagej.html

- **Fractal dimension**
- https://en.wikipedia.org/wiki/Fractal_dimension

- **Texture**
- *haralick.org/journals/TexturalFeatures.pdf*
- *www.uio.no/studier/emner/matnat/ifi/INF4300/h08/undervisningsmateriale/glcm.pdf*