Lectures 1 & 2:
Basic Image Analysis

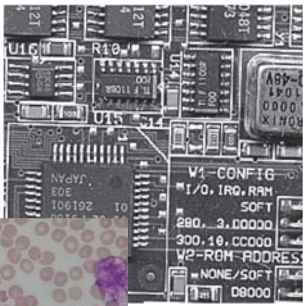
Dr Carole Twining

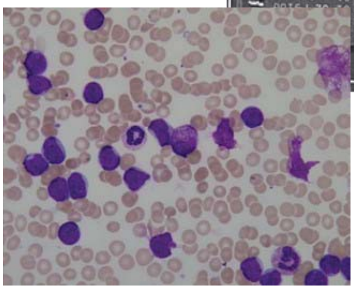
Tuesday 10th March 2020

11:15am – 13:15pm

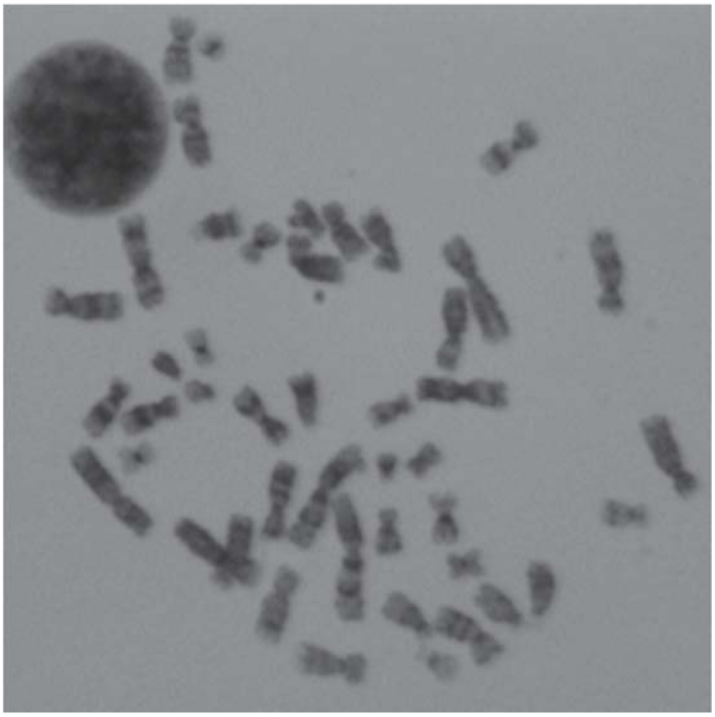
Basic Image Analysis

- **Limited to simple 2D scenes**
  - Adequately described as background and objects
- **Good contrast between objects and background**
  - staining or backlighting
- **Constrained applications**
  - microscopic materials analysis
  - biomedical microscopy
  - industrial inspection





Sample Problem:



- Stained preparation, light microscope
- Chromosomes, with bands
- Measure banding pattern

Solving the Problem

Plan

Distinguish between objects & background

↓

Locate each individual object

↓

Locate centerline

↓

Measure bands

Tasks

What is an image?

What is background?

From not-background to distinct objects

Shape of an object?

Measurements on object

Overview:

- Image Representation
  - What is an image?
- Grey-Level Processing
  - Improving the starting image
- Segmentation
  - Background pixels and object pixels
- Binary Image Processing
  - Improved background/object binary image
- Measurement
  - Object as connected region

Image Representation

Image Representation

- Isn't it totally obvious? We all know what an image is!
- Various ways of representing an image, depending on the task in hand
  - Image function
  - Landscape
  - Array of pixels
  - Image histogram
  - In another space entirely!

Image Representation

Image

Image Plane

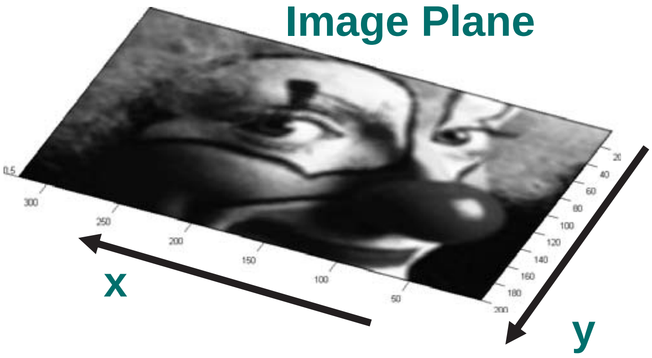


Image Function

$I(x, y)$

brightness becomes height

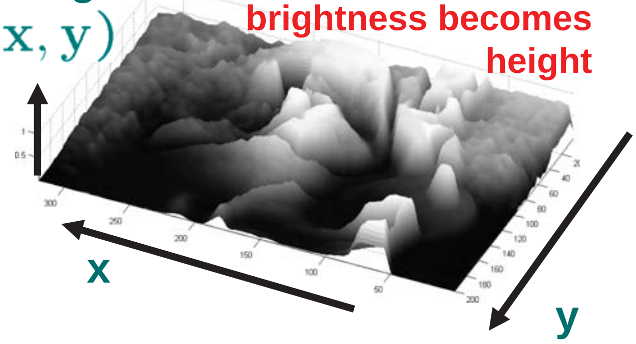


Image Landscape

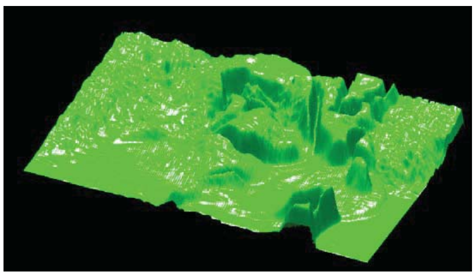
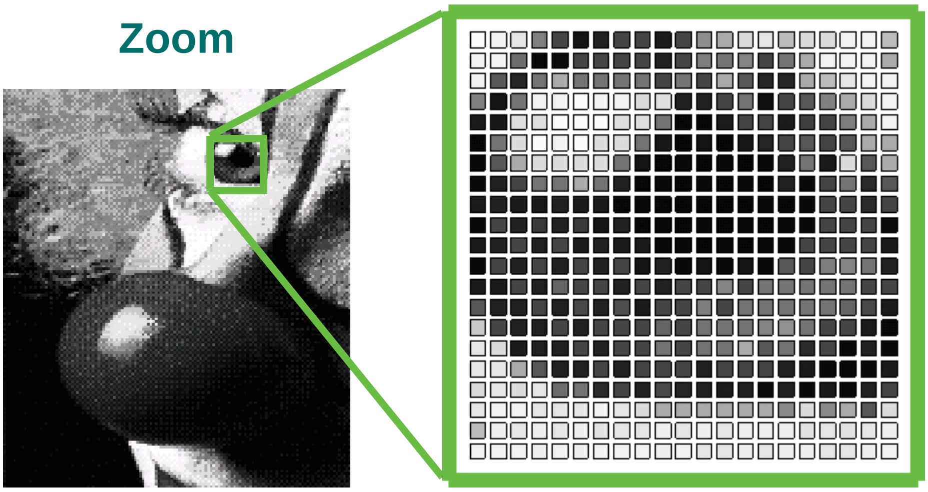


Image Representation

Zoom



Array of Pixels:

Values and spatial relationship

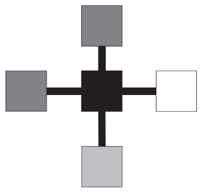
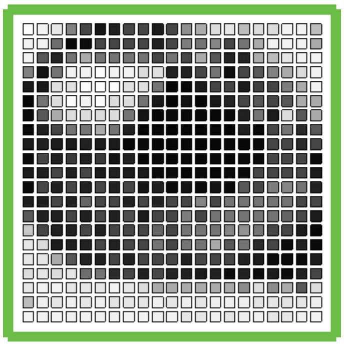


Image Representation



- Sort pixels by grayscale value/colour and stack them up

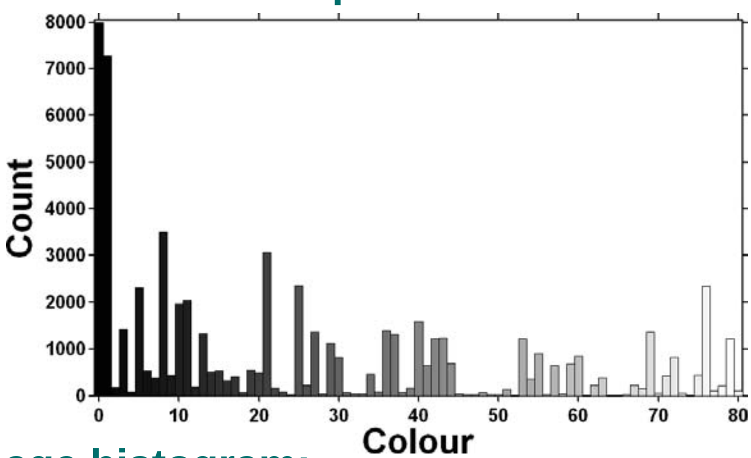
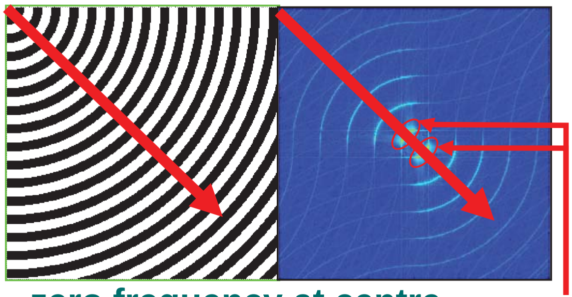


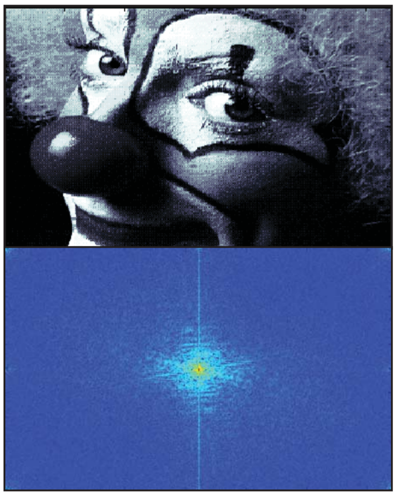
Image histogram:
Kept values but lost spatial information

Image Representation

- **Fourier Analysis:**
  - any signal can be decomposed into a sum of sinusoids (FFT)
  - low frequencies, general shape, high frequencies details



zero frequency at centre $f \propto \frac{1}{\text{spacing}}$



NOTE:
zero frequency removed by subtracting mean value across image from image before doing FFT

Image Representation

NOTE:
$e^{i\theta} \equiv \cos\theta + i\sin\theta$
$\Rightarrow \mathcal{F}_{\mathrm{I}}$ complex, $\mathbf{I}(\underline{r})$ real
so $\mathcal{F}_{\mathrm{I}}(-\underline{k}) \equiv \overline{\mathcal{F}}_{\mathrm{I}}(\underline{k})$

- **Frequency Space:**
  - Integrate over the image, weighted by complex exponentials

$$\mathcal{F}_I(u,v) \propto \iint I(x,y)\exp(iux+ivy)dxdy$$

  - Compact vector form:

$$\mathcal{F}_I(\underline{k}) \propto \iint I(\underline{r})\exp(i\underline{k}\cdot\underline{r})d\underline{r}$$

  - Inverse:

$$I(\underline{r}) \propto \iint \mathcal{F}_I(\underline{k})\exp(-i\underline{k}\cdot\underline{r})d\underline{k}$$

Grey-Level Processing

Grey-Level Processing

- **Restoration:**
  - **What is noise, what is signal?**
  - **Remove blurring**
- **Enhancement**
  - **Emphasize required features (e.g., linear features)**
  - **Emphasize change (e.g., surveillance)**

Grey-Level Processing: Overview

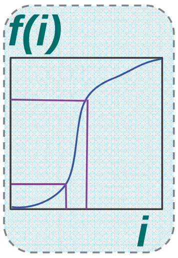
- Point processing
  - Transform global gray-level scale
- Neighbourhood Processing
  - Values and their context (local context & processing)
- Image Arithmetic
  - Using a sequence/pair of images
- Image Transforms
  - Images in a different space (frequency space)

Grey-Level Processing: Point Processing

Grey-Level Processing: Point Processing

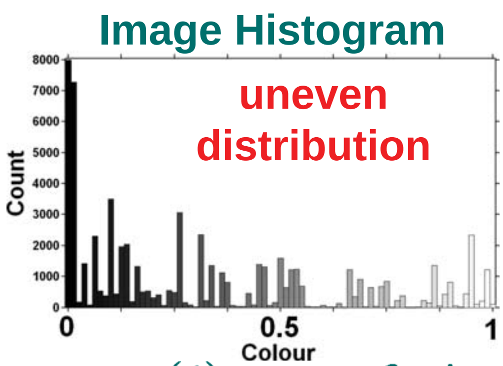
- Point = Pixel
- Transforms image based on single pixel value alone:

$$\underbrace{\tilde{I}(x,y)}_{\text{new pixel value}} = \underbrace{f}_{\text{function}}\Big(\underbrace{I(\overbrace{x,y}^{\text{position}})}_{\text{pixel value}}\Big)$$



- Various choices for monotonic function f(i)
  - Increase/decrease/stretch brightness and contrast
  - Gamma correction, power law : $f(i) = i^{\gamma}$
  - Histogram matching between images
  - Histogram equalization

Point Processing: Histogram Equalisation



- Re-assign colours, keep ordering (light to dark)
- Increase contrast

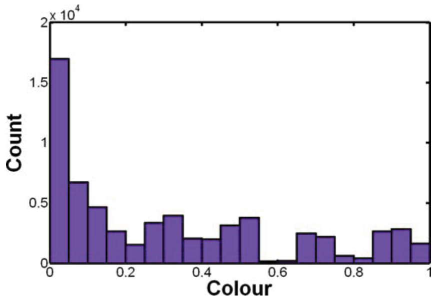
$n(i)$: no of pixels with colour i,

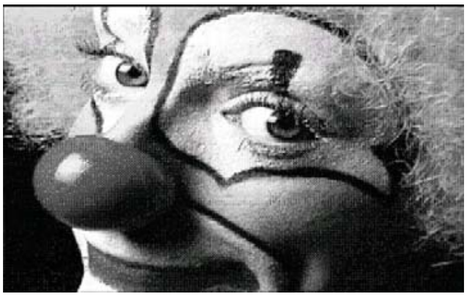
N : Total number

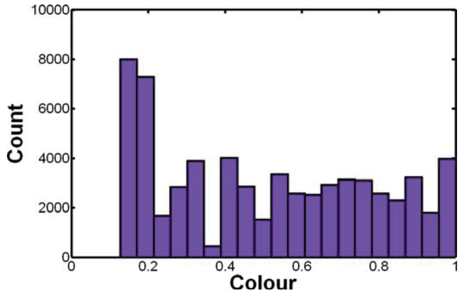
$$\boxed{\text{New Colour: } f(i) = \frac{1}{N}\sum_{j \leq i} n(j)}$$

$f(i) = 0.75$, 75% darker than this

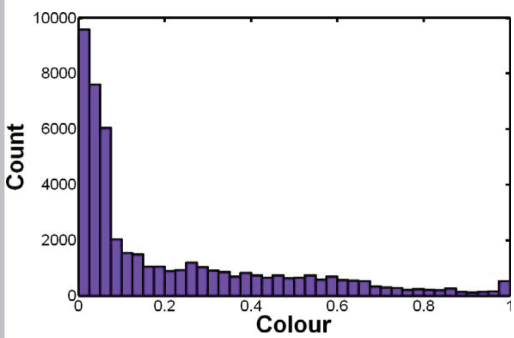
Point Processing: Histogram Equalisation

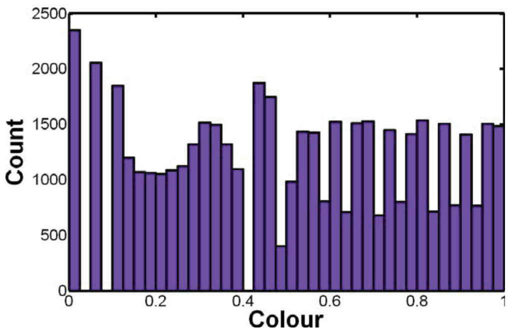






Point Processing: Histogram Equalisation





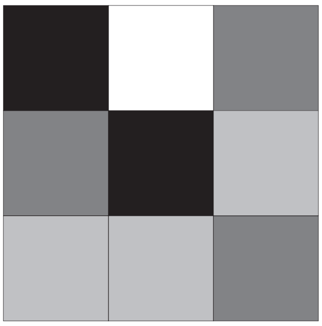
The University of Manchester

Grey-Level Processing: Neighbourhood Processing

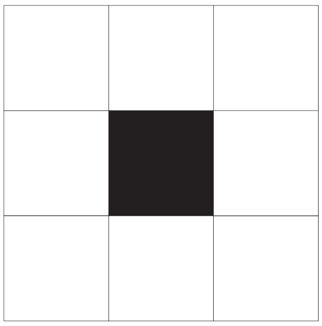
The University of Manchester

Neighbourhood Processing

single black pixel



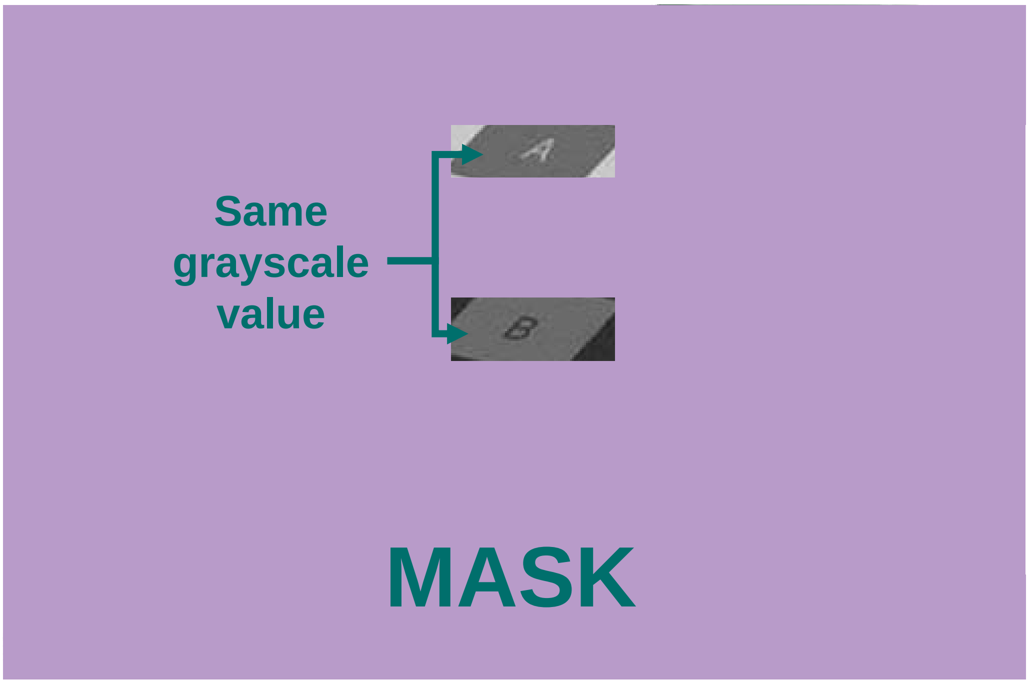
Noisy dark area



Just noise

- Consider a single pixel value in context of neighbours
- Neighbourhood (e.g. 3 x 3), structuring element (SE)
- Two methods:
  - Convolution
  - Rank Filtering

Aside: Context in Human Vision



Convolution: 1D Example

NOTE:
0 black to
1 white
0 to 255 8-bit
images

$g(a)$

1	2	1

$I(x)$

1	0	0.5	1	1	0.5	0.5	0.5	1	1

$\tilde{I}(x)$

1	0.37	0.5	0.9	0.9	0.6	0.5	0.6	0.9	1

- Weighted sum of neighbours

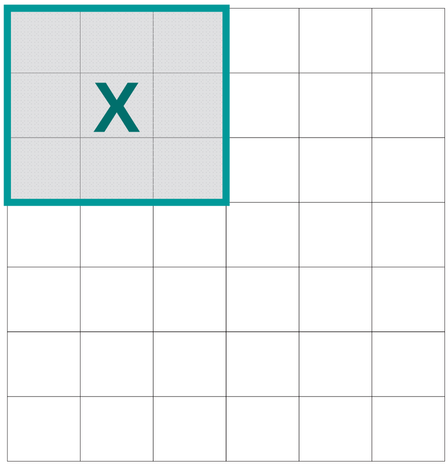
$$\tilde{I}(x) = \frac{\sum_a g(a) I(x+a)}{\sum_b g(b)}$$

Normalize the weights if averaging

Convolution: 2D

g(-1,1)	g(0,1)	g(1,1)
g(-1,0)	g(0,0)	g(1,0)
g(-1,-1)	g(0,-1)	g(1,-1)

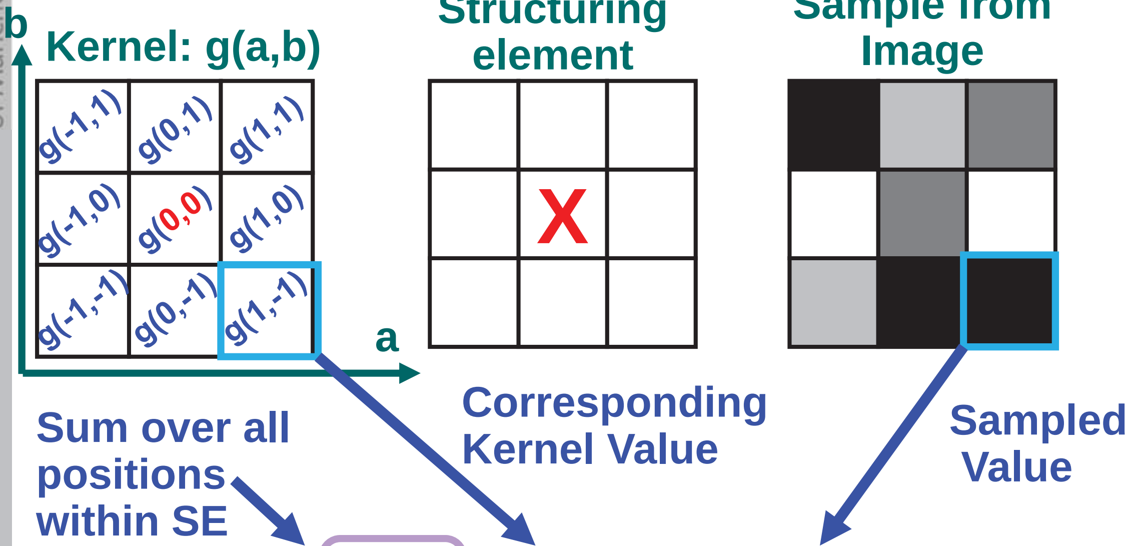
g(a,b)



$I(x, y)$

$$\tilde{I}(x, y) = \frac{\sum_a \sum_b g(a,b) I(x+a, y+b)}{\sum_c \sum_e g(c,e)}$$

Convolution: 2D



$$\tilde{I}(x, y) = \frac{\boxed{\sum_a \sum_b} g(a,b) I(x+a, y+b)}{\boxed{\sum_c \sum_e g(c,e)}}$$

Output

Normalize the weights if smoothing

Convolution: 2D

- Asterisk notation: $\tilde{I} = g * I$
- (but NOT in MATLAB!)
- Discrete form:

$$\tilde{I}(x,y) = \frac{\sum_a \sum_b g(a,b) I(x+a, y+b)}{\sum_c \sum_e g(c,e)}$$

- Integral form:

$$\tilde{I}(x,y) = \frac{\iint g(a,b) I(x+a, y+b)\, da\, db}{\iint g(c,e)\, dc\, de}$$

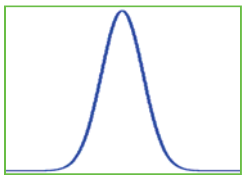
- Integral form (vector notation)

$$\underline{r} = (x,y), \quad \tilde{I}(\underline{r}) = \frac{\iint g(\underline{z}) I(\underline{r}+\underline{z})\, d\underline{z}}{\iint g(\underline{y})\, d\underline{y}}$$

MANCHESTER 1824 The University of Manchester COMP61342

Convolution: Common Kernels

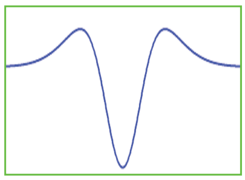
- Gaussian: $g(x,y) = A \exp(-(x^2+y^2)/2\sigma^2)$ σ width
  - Smoothing kernel
  - Any unimodal kernel smoothes the image



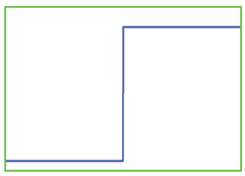
- Difference of Gaussian (DoG)

$$g(x,y) = A \exp(-(x^2+y^2)/2\sigma^2) - B \exp(-(x^2+y^2)/2\alpha^2)$$

- Laplacian (or Laplacian of Gaussian)
  - similar shape to DoG, second-derivative filter



- First-derivative edge filters
  - Give ridges/troughs at edge positions



Convolution Theorem

NOTE:
$e^{i\theta} \equiv \cos\theta + i\sin\theta$
$\Rightarrow \ \mathcal{F}_{\mathrm{I}}$ complex, $\mathrm{I}(\underline{r})$ real
so $\mathcal{F}_{\mathrm{I}}(-\underline{k}) \equiv \overline{\mathcal{F}}_{\mathrm{I}}(\underline{k})$

- **Frequency space (see Image Representation) :**

$$\mathcal{F}_I(\underline{k}) \propto \iint I(\underline{r}) \exp(i\underline{k}\cdot\underline{r})d\underline{r}$$

- **Look at it in frequency space or real space:**
  - convolution in real space $\Leftrightarrow$ multiplication in frequency space

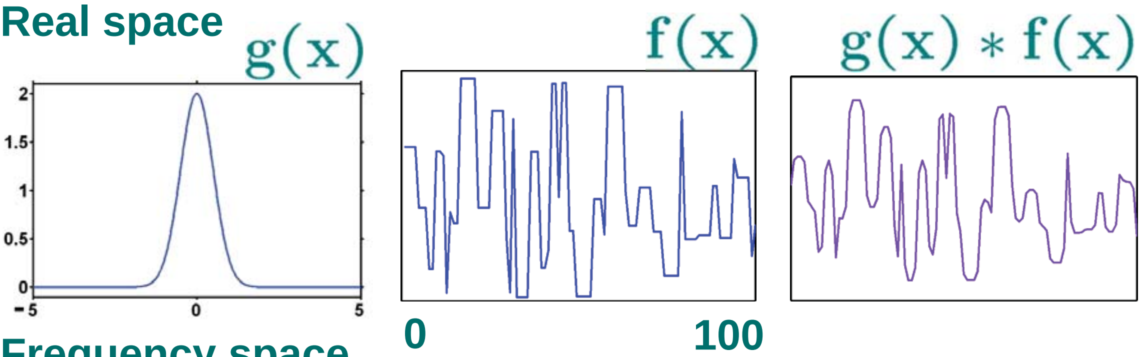
$$\mathrm{g} * \mathrm{I} \quad \Longleftrightarrow \quad \mathcal{F}_{\mathrm{g}} \times \mathcal{F}_{\mathrm{I}}, \quad \mathrm{g} * \mathrm{I} \equiv \mathcal{F}^{-1}\left(\mathcal{F}_{\mathrm{g}} \times \mathcal{F}_{\mathrm{I}}\right)$$

  - convolution in frequency space $\Leftrightarrow$ multiplication in real space

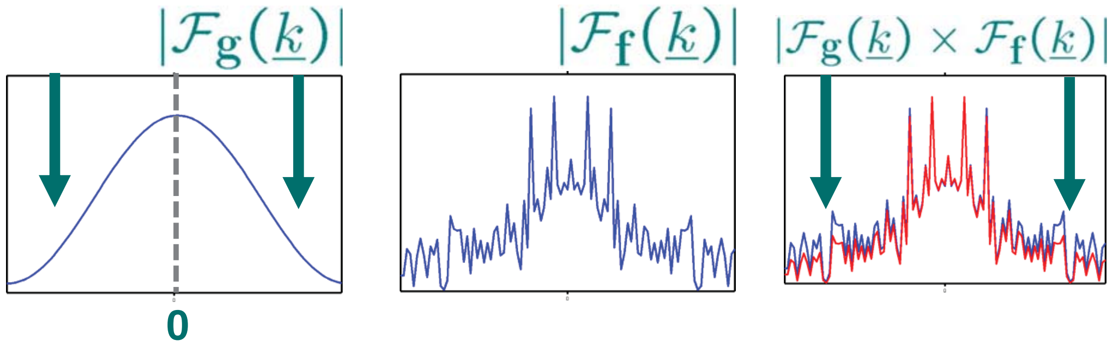
$$\mathcal{F}_{\mathrm{g}} * \mathcal{F}_{\mathrm{I}} \quad \Longleftrightarrow \quad \mathrm{g} \times \mathrm{I}, \quad \mathcal{F}_{\mathrm{g}} * \mathcal{F}_{\mathrm{I}} \equiv \mathcal{F}\left(\mathrm{g} \times \mathrm{I}\right)$$

Convolution Theorem: Gaussian

Real space

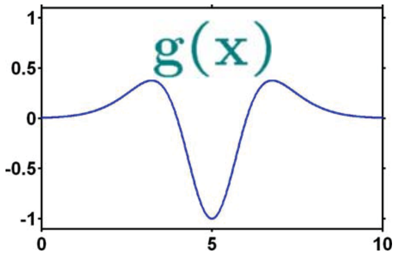


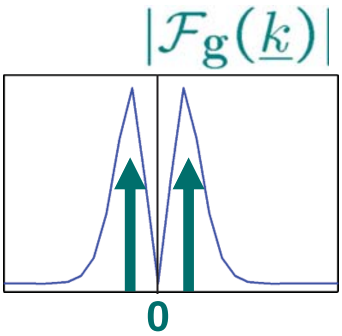
Frequency space



Convolution Theorem: Difference of Gaussians

$$g(x,y) = A\exp(-(x^2+y^2)/2\sigma^2) - B\exp(-(x^2+y^2)/2\alpha^2)$$





- band-pass filter, enhances edges
- Laplacian and LoG similar

signal at edges

Convolution Theorem: Laplacian of Gaussian & Difference of Gaussians

Gaussian and FT of Gaussian

$$\boxed{g(x) \propto e^{-\beta x^2},\ \mathcal{F}_g(k) \propto e^{-\alpha k^2}}$$

Convolution Theorem

$$\boxed{g * I \equiv \mathcal{F}^{-1}(\mathcal{F}_g \times \mathcal{F}_I)}$$

Laplacian of gaussian:

$$\underbrace{\frac{\partial^2}{\partial x^2}}_{\text{Laplacian}}\left(\int \underbrace{e^{-ikx}}_{\text{Inverse FT}}\underbrace{e^{-\alpha k^2}}_{\text{Gaussian}}\underbrace{\mathcal{F}_I(k)}_{\text{FT of Image}}dk\right)$$

- Do the derivative:

$$\int \boxed{-k^2 e^{-ikx}}\, \boxed{e^{-\alpha k^2}}\, \mathcal{F}_I(k)dk \rightarrow \frac{d}{d\alpha}\overbrace{\int e^{-ikx}e^{-\alpha k^2}\mathcal{F}_I(k)dk}^{\text{Convolution with Gaussian, parameter } \alpha}$$

- LoG: difference of infinitesimally-separated gaussians
- DoG: difference of finitely-separated gaussians

The University of Manchester
MANCHESTER 1824

Neighbourhood Processing: Rank Filtering

The University of Manchester
MANCHESTER 1824

Neighbourhood Processing: Rank Filtering

- Output is rank function of neighbourhood:
  - median (smoothes and preserves edges)
  - max and/or min (mathematical morphology)
  - rank number (seven of nine)
- Harder to analyse than convolution

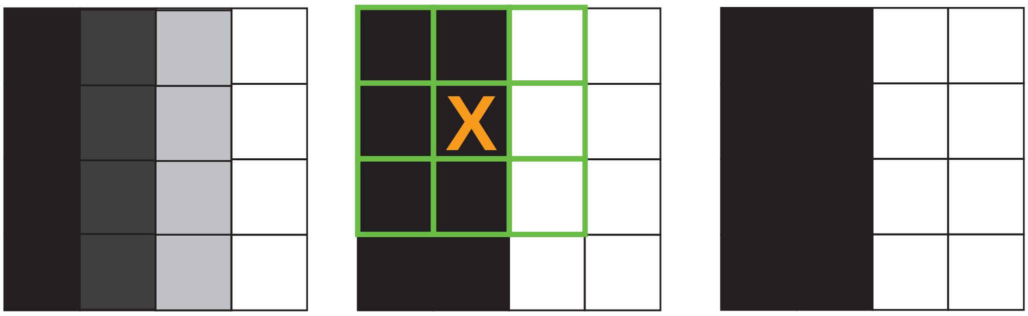
Noisy Image

3x3 mean

3x3 median

Rank Filtering & Edges: Example

- **Mean:**
- **2/3 ■ + 1/3 □ = ■**
- **1/3 ■ + 2/3 □ = □**
- **Median:**
- **6 ■ & 3 □ => ■**
- **6 □ & 3 ■ => □**

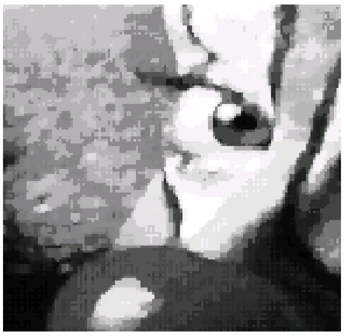


3x3 SE

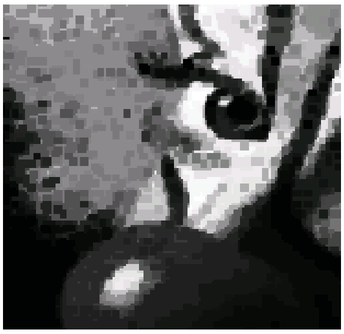
Neighbourhood Processing: Rank Filtering

- **Rank Number**
  - **3 x 3 structure element**

Original



maximum



7th of nine

blocky, impressionistic effect

The University of Manchester

Grey-Level Processing: Image Arithmetic

The University of Manchester

Image Arithmetic: Addition

- Take average over images in sequence
- Reduces noise

Original

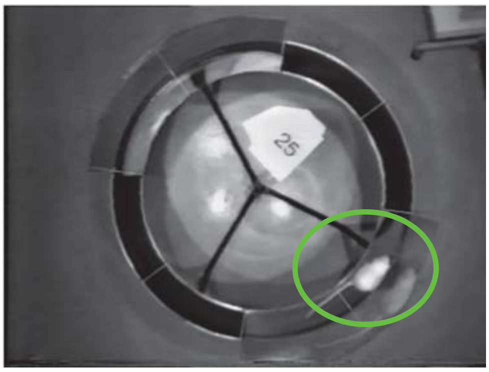
Addition

Noisy 1

Noisy 2

Image Arithmetic: Subtraction

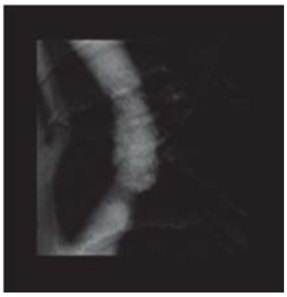
- Take difference:
  - Negative values? Shift and scale to get back to [0:255]
  - Or take absolute difference
- Static background, detects change
- Object, shadows & reflections in real-world scenes



MANCHESTER 1824
The University of Manchester
COMP61342

Image Arithmetic: Subtraction

- Digital subtraction angiography (DSA)
- Pre-study radiograph
- Contrast agent injection
- Post-contrast radiograph
- Difference



Introduction to Segmentation

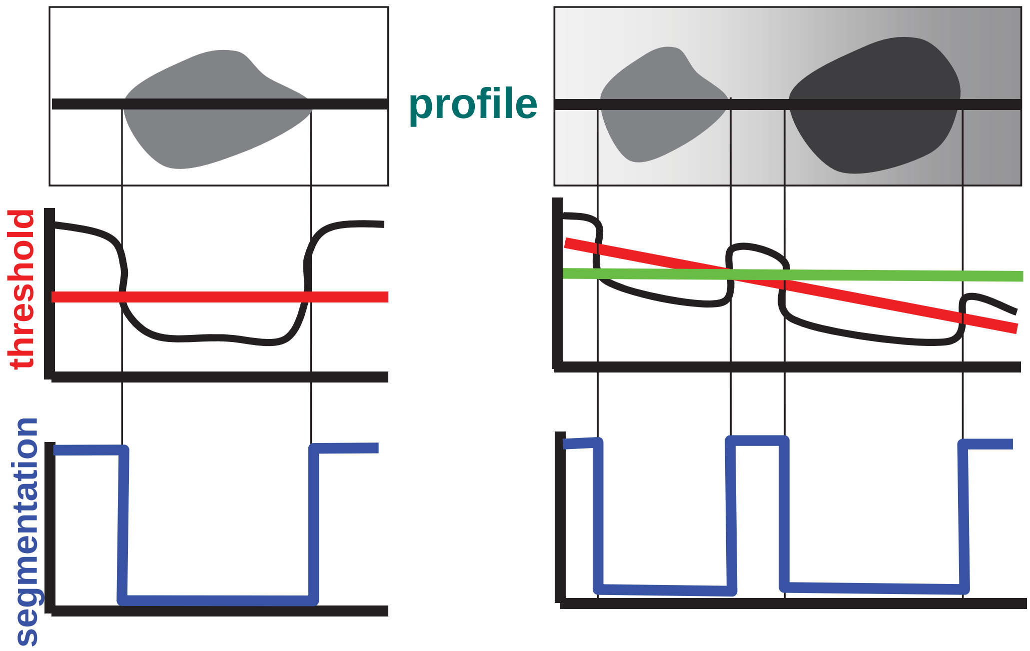
Segmentation:

Task: label each pixel as either object or background

- Grayscale image $\longrightarrow$ binary label image
- Thresholding
  - simple, high-contrast images
- Adaptive thresholding
  - simple images with shaded background
- Advanced Segmentation
  - open research problem

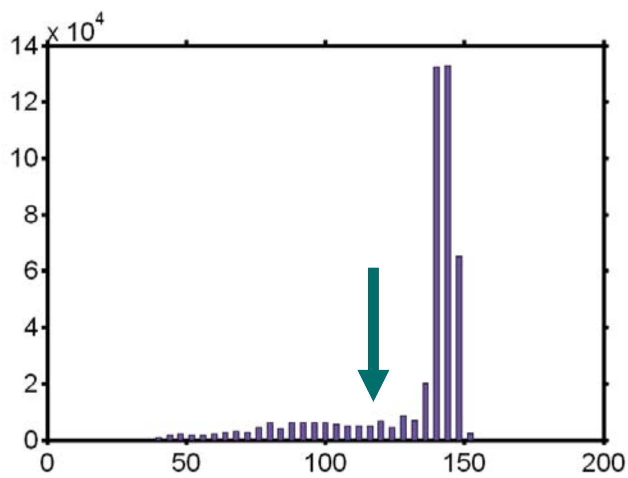
Segmentation: Thresholding

- Thresholding
- Adaptive Thresholding



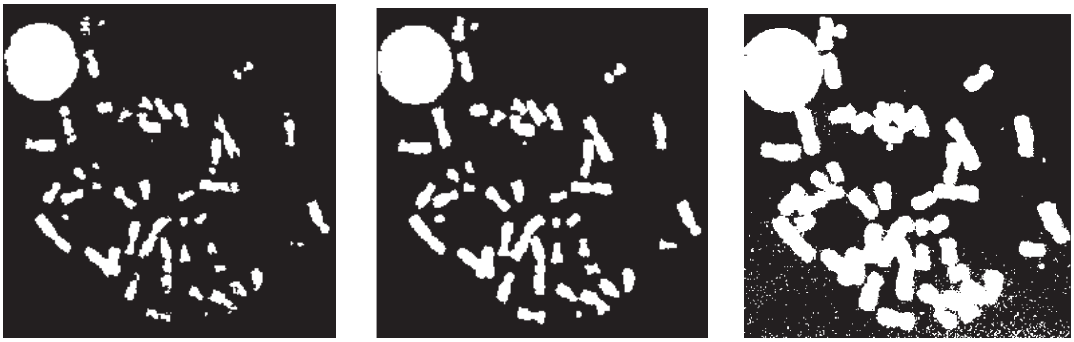
MANCHESTER 1824
The University of Manchester
COMP61342

Segmentation: Thresholding, Histogram



Segmentation: Thresholding

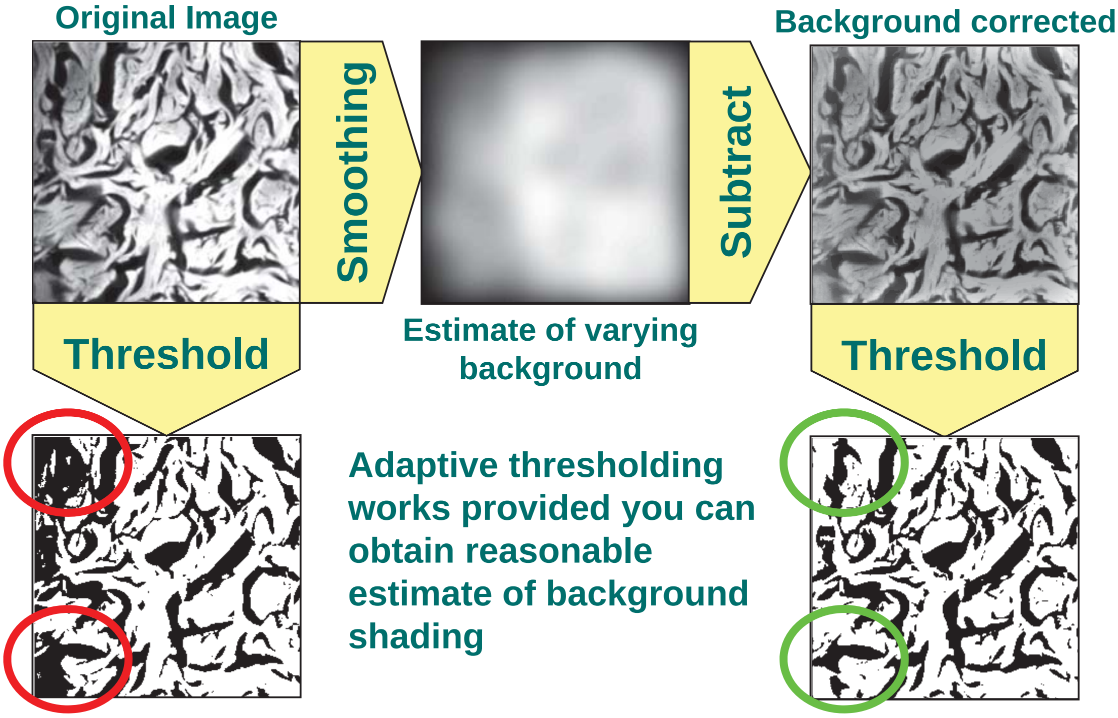
- Varying the Threshold



Threshold 100 Threshold 110 Threshold 140

- Need to choose threshold with care,
- How to improve the binary image

Segmentation: Adaptive Thresholding



Adaptive thresholding works provided you can obtain reasonable estimate of background shading

The University of Manchester
MANCHESTER 1824

Binary Processing

The University of Manchester
MANCHESTER 1824

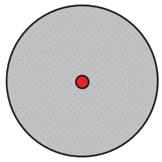
Binary Processing

Aim: Improved binary image

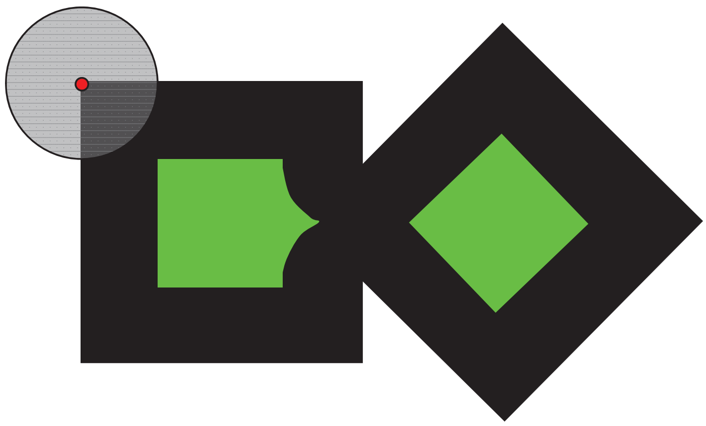
- Restoration or enhancement
- Neighbourhood Processing:
  - binary morphology (erosion & dilation)
  - skeletonization
- Image Logic:
  - combining binary images for more complicated processing

Binary Morphology: Erosion

- Structure element (example, centre marked):

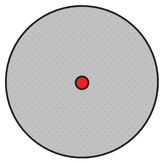


- Binary object:
- Sweep SE along boundary, and delete region covered

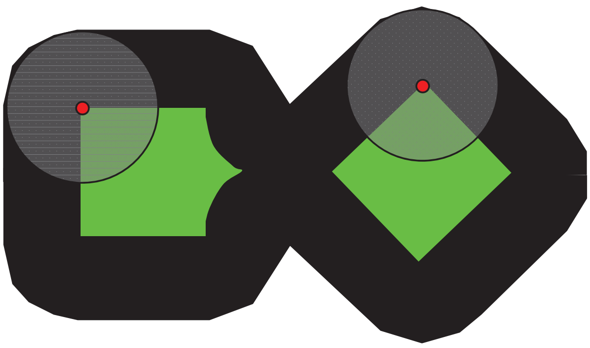


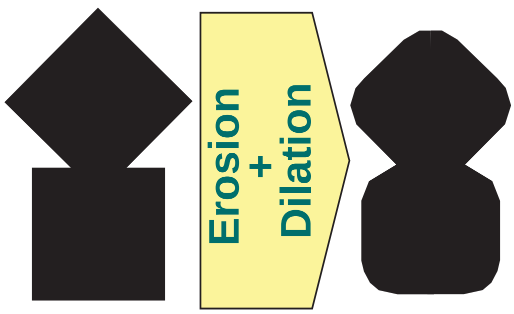
Binary Morphology: Dilation

- Structure element (centre marked):



- Binary object:
- Reverse of erosion
- Sweep SE along boundary, and **add** region covered

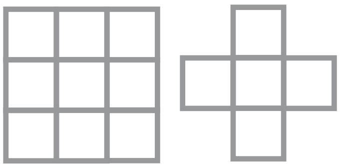


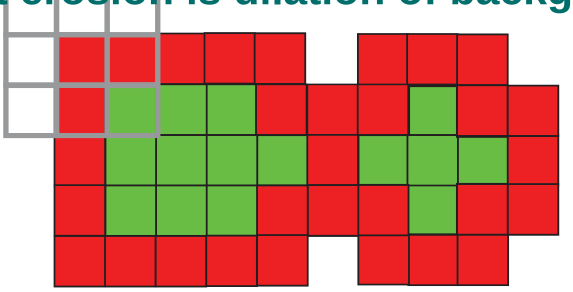


Rounded-off the corners

Binary Morphology: Dilation, Implementation via Neighbourhood Processing

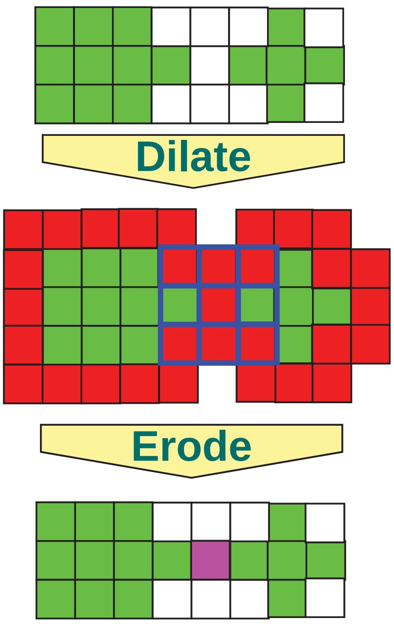
- Pixellated structuring element
- Pixellated image object
- Scan SE over image, and add pixel at defined centre if any object pixel lies within SE
- Object erosion is dilation of background, so similar





Binary Morphology: Closing & Opening

- Closing: reconnection



- Opening: disconnection

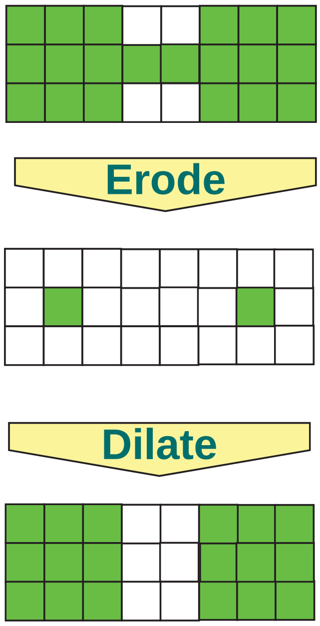
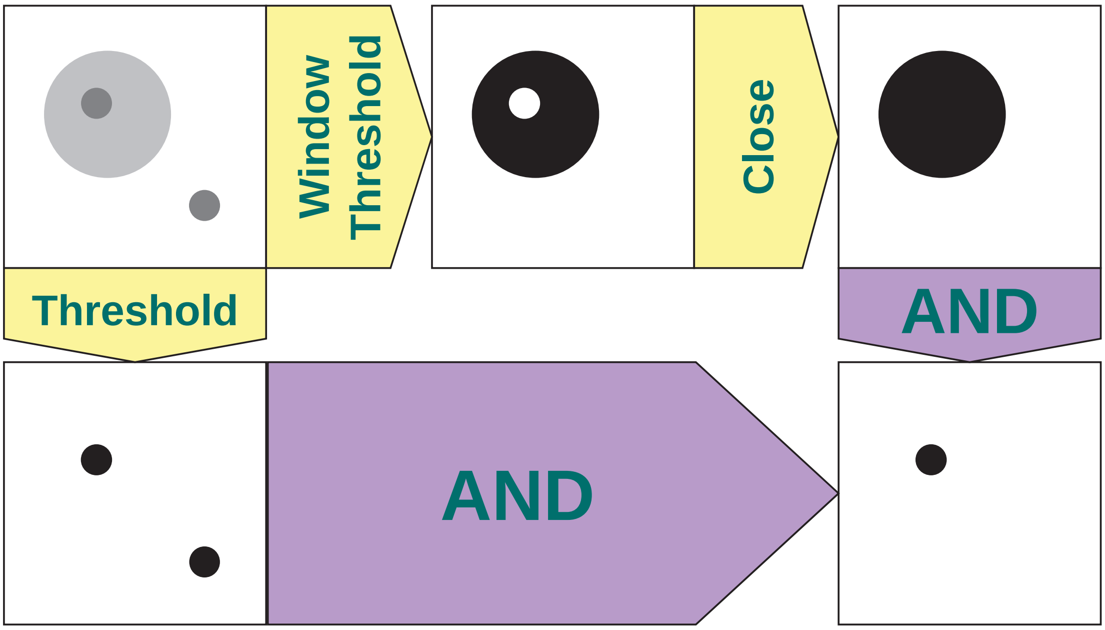


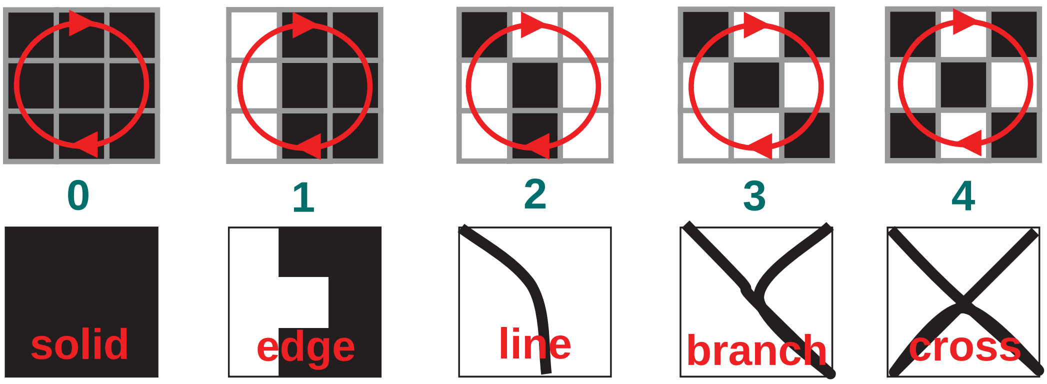
Image Logic:

Want dark object within lighter grey object



Binary Morphology: Skeletonisation

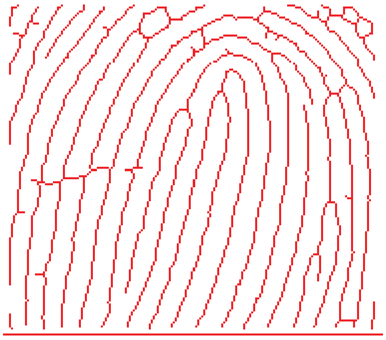
- Erosion that preserves connections
- Rutovitz Crossing Number: (3x3 SE)
  - loop and half the number of times value changes

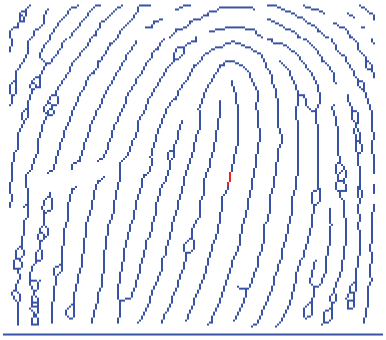


- Remove centre pixel if 1: nibble at edge, but leave crossings
- Repeat until no further change

Binary Morphology: Skeletonisation


fingerprint

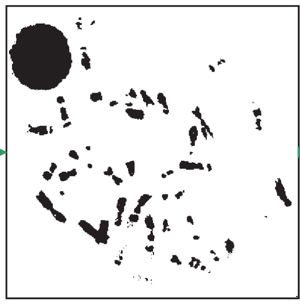

valleys

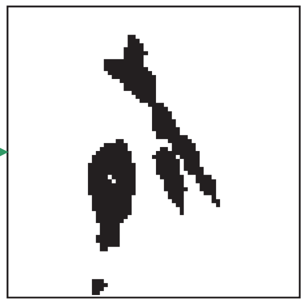

ridges

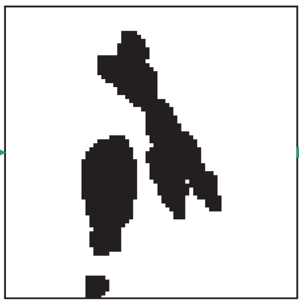
Feng Zhao and Xiaoou Tang
PREPROCESSING FOR SKELETON-BASED FINGERPRINT MINUTIAE EXTRACTION
CISST’02 International Conference

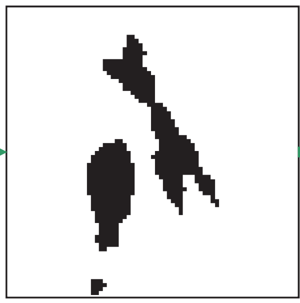
Chromosome Results:

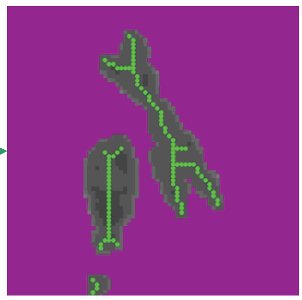

Original image


Thresholded


Detail


Dilated


Eroded


Skeleton

Measurement

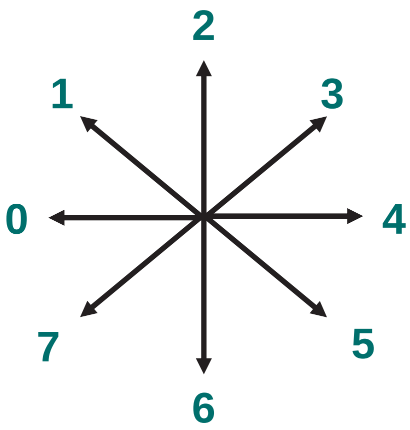
MANCHESTER 1824
The University of Manchester

Simple Measurements on Objects

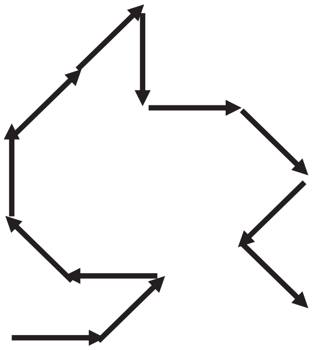
- Extracted objects as above
- Representing Objects:
  - Boundary representation
  - Area representation
- Simple geometric measurements
  - Area
  - Perimeter
  - Circularity

Representing Objects: Boundary

- Boundary Representation: chain code



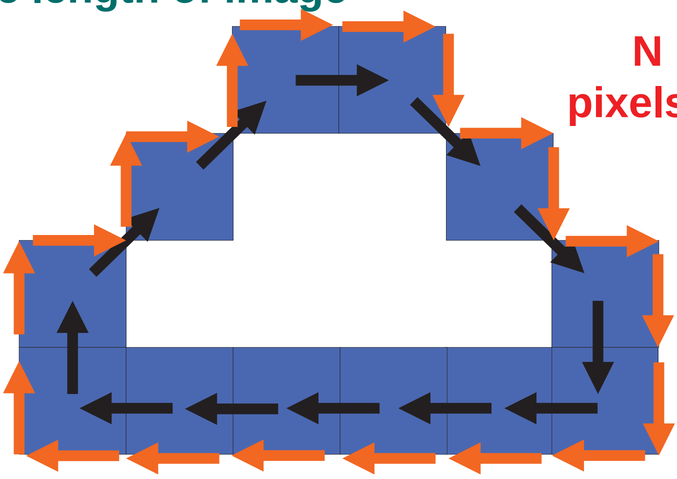
Pick a set of directions



chain code:
4 3 0 1 2 3 3 6 4 5 7 5

Representing Objects: Boundary

- Positions of boundary pixels: 2N times (one from L)
- L: side length of image

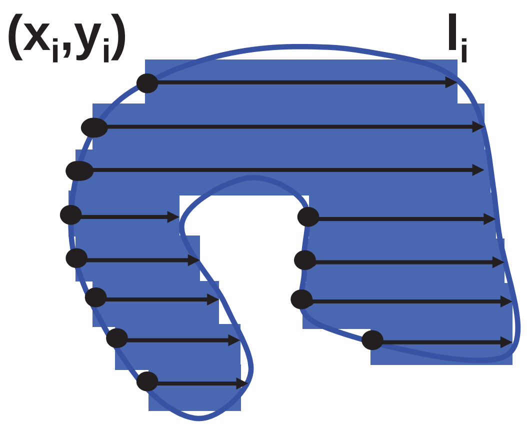


N Boundary pixels of an object

- Chain code: N times (one of eight)
- OR: ~1.5 N times (one of four)

Representing Objects: Area

- Area Representation: Chord List



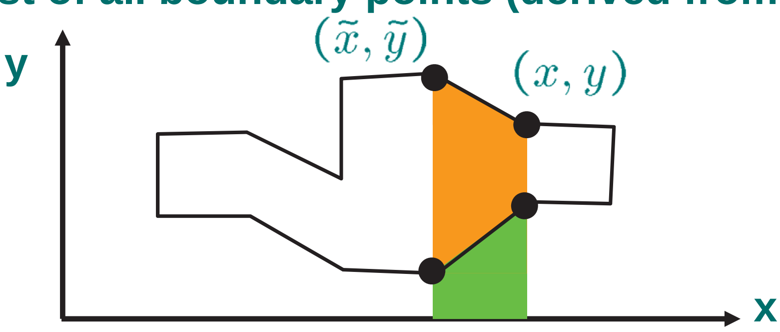
chord (x_i ,y_i ,l_i):
start position and length

Chord list represents the shape of the pixelated object

Much more efficient representation of data compared to storing position of every pixel within the region!

Measurement: Area

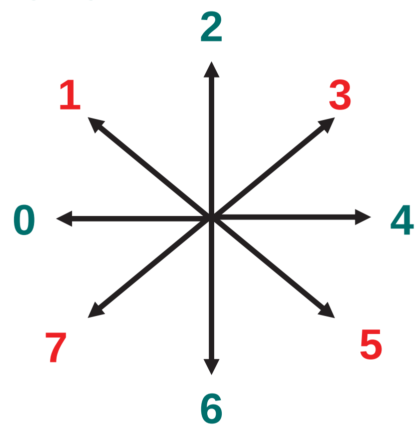
- List of all boundary points (derived from chord list)



- Trapezoidal rule $\text{Area} = \frac{(y+\tilde{y})(x-\tilde{x})}{2}$
- Take difference to find area of strip of shape

Measurement: Perimeter

- 8-piece Chain Code:
- Diagonals are longer!



$$P = N_{even} + \sqrt{2} N_{odd}$$

- 4-piece chain code: P = N, all equal length
- Circularity: $$C = \frac{4\pi \text{Area}}{P^2},$$
- C=1 for circle, C<1 for anything else

Summary

Basic Image Analysis:

- Mostly straightforward and fairly intuitive
- Can give good results on suitable images
- Have to grasp basics before can move on to more sophisticated methods