

Visual Computing Summary

September 27, 2020

Chapter 1

The Digital Image

1.1 What is an image:

Signal: A function depending on some variable with physical meaning

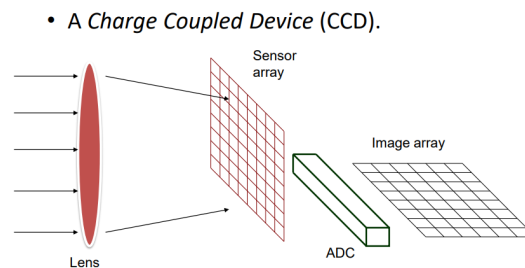
Image: A continuous function. The value can take on physical values. e.g Brightness, temperature, pressure, depth, etc.
We distinguish images with:

- **2 variables:** xy- coordinates
- **3 variables:** xy + time (video)

Hence an image is a picture or pattern of a value varying in space and/or time.

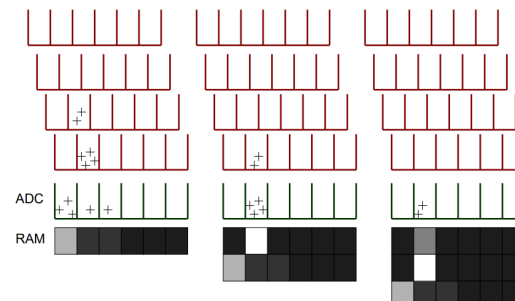
$$f : \mathbb{R}^n \rightarrow S$$

1.2 The digital camera



The sensor array: An array of photosites. Each photosite is a bucket of electrical charge and contain a charge proportional to the incident light intensity during exposure.

Analog to Digital Conversion: The ADC measures the charge and digitizes the result. The conversion happens line by line. The charges in each photosite move down through the sensor array.

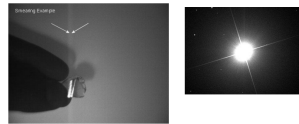


Various errors:

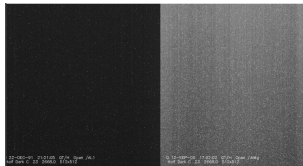
- **Blooming:** The buckets have finite capacity, saturation causes blooming. It happens when a large amount of light gets focused to a single point on your cameras image sensor. This can create so much charge that it actually bleeds from pixel to pixel until it eventually spreads out.



- **Bleeding or Smearing:** During transit buckets still accumulate some charges. The amount is influenced by the time "in transit" versus the integration time



- **Dark Current:** A relatively small electric current that flows through photosensitive devices even when no photons are entering the device.

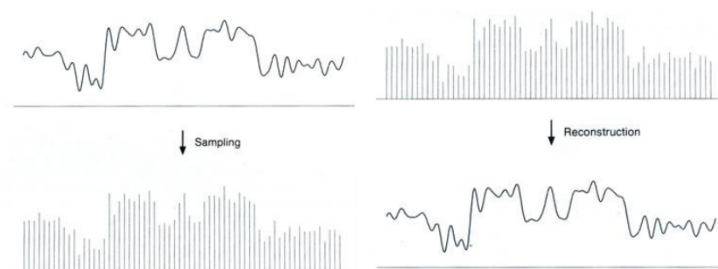


CMOS: Contains the same sensor elements as CCD but each photo sensor has its own amplifier. the benefits:

- Recent technology
- Standard IC technology
- Cheap
- Low Power
- Less sensitive
- Per pixel amplification
- Random pixel access
- Smart pixels
- On chip integration with other components

1.3 Sampling 1D:

Sampling in 1D takes a function, and returns a vector whose elements are values of that function at the sample points.

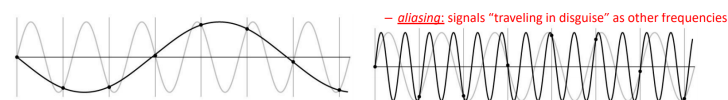


1.3.1 Reconstruction:

Making samples back into a continuous function. This amounts to guessing what the function did in between the individual samples

1.3.2 Undersampling:

Occurs if not enough sampling points are available. It results in a loss of information and can be indistinguishable from other samples e.g undersampling a sin wave:

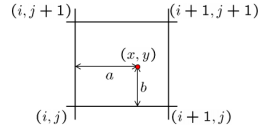


1.4 Sampling 2D:

Sampling in 2D takes a function and returns an array. The array can have infinite dimensions and have negative as well as positive indices.

1.4.1 Reconstruction continuous signals:

- e.g. Bilinear interpolation



$$f(x, y) = (1-a)(1-b) f[i, j] + a(1-b) f[i+1, j] + ab f[i+1, j+1] + (1-a)b f[i, j+1]$$

■

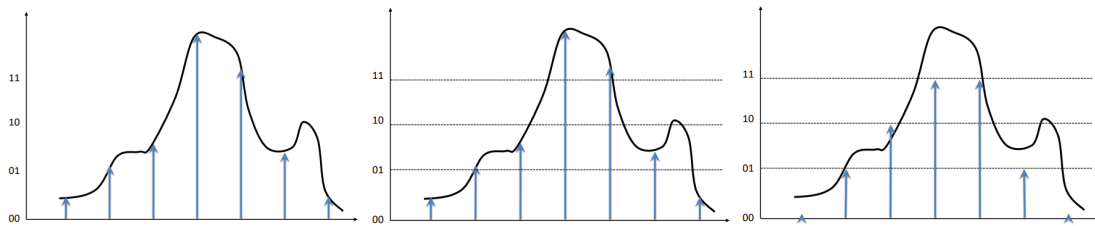
1.4.2 Nyquist Frequency:

Half the sampling frequency of a discrete signal processing system. The signals max frequency (bandwidth) must be smaller than this.

1.4.3 Quantization:

Real valued function gets digital values (integer values). Quantization is lossy i.e the original signal cannot be reconstructed anymore. Simple quantization uses equally spaced levels with k intervals:

$$k = 2^b$$



1.4.4 Image Properties:

Geometric Resolution: How many pixels per area

Radiometric Resolution: How many bits per pixel

1.5 Image Noise

The different types of common Noise models:

- **Gaussian noise:**

$$I(x, y) = f(x, y) + c$$

where $c \sim N(0, \sigma^2)$ so that

$$p(c) = (2\pi\sigma^2)^{-1} e^{-\frac{c^2}{2\sigma^2}}$$

- **Poisson noise:**

$$p(k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

- **Rician noise:**

$$p(I) = \frac{I}{\sigma^2} e^{-\frac{(I^2 + f^2)}{2\sigma^2}} I_0\left(\frac{If}{\sigma^2}\right)$$

- **Multiplicative noise:**

$$I = f + fc$$

Signal to noise ration (SNR) An index of image quality:

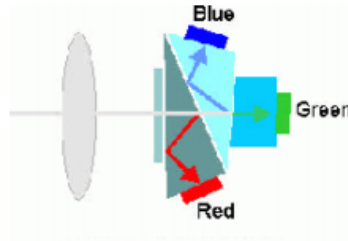
$$s = \frac{F}{\sigma} \text{ where } F = \frac{1}{XY} \sum_{x=1}^X \sum_{y=1}^Y Y f(x, y)$$

Peak Signal to Noise Ration (PSNR)

$$s_{peak} = \frac{F_{max}}{\sigma}$$

1.6 Color Cameras:

Prism Color Camera: Separates the light into 3 beams using a dichroic prism



Filter mosaic: Filter is coated directly on sensor

Filter wheel: Rotate multiple filters in front of lens. This is only suitable for static scenes.

Prism vs. mosaic vs. wheel

approach	Prism	Mosaic	Wheel
# sensors	3	1	1
Separation	High	Average	Good
Cost	High	Low	Average
Framerate	High	High	Low
Artefacts	Low	Aliasing	Motion
Bands	3	3	3 or more
	High-end cameras	Low-end cameras	Scientific applications

Chapter 2

Image Segmentation:

Image segmentation is the concept of partitioning an image into regions of interest.

Complete Segmentation: A finite set of regions R_1, \dots, R_N , such that

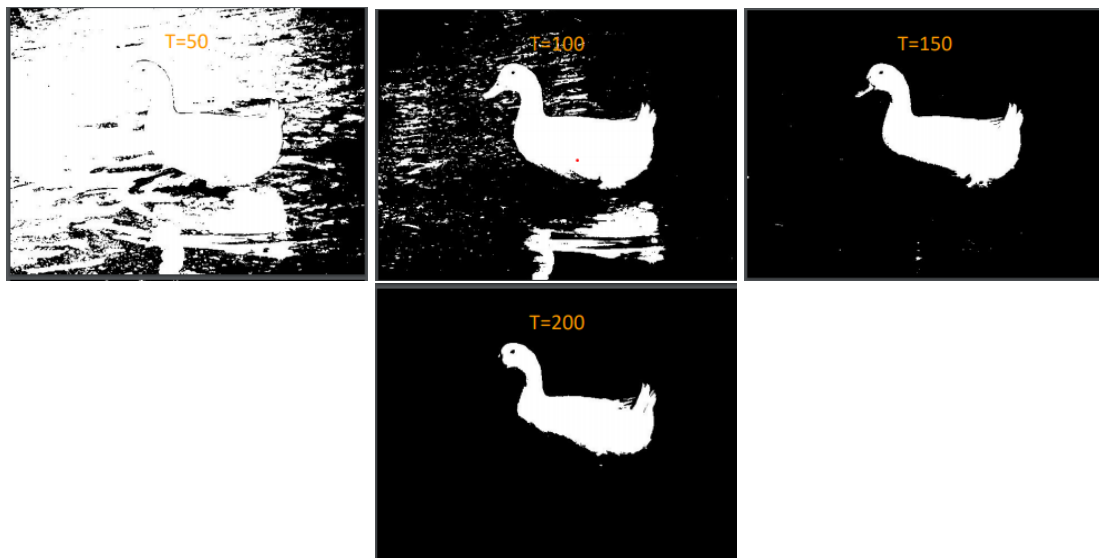
$$I = \bigcup_{i=1}^N R_i \text{ and } R_i \cap R_j = \phi \quad \forall i \neq j$$

Where I is an image.

2.1 Thresholding

Thresholding is a simple segmentation process which produces a binary image B. It labels each pixel "in" or "out" of the region of interest by comparison of the greylevel with a threshold T:

$$B(x, y) = \begin{cases} 1 & \text{if } I(x, y) \geq T \\ 0 & \text{if } I(x, y) < T \end{cases}$$



2.2 ROC Analysis

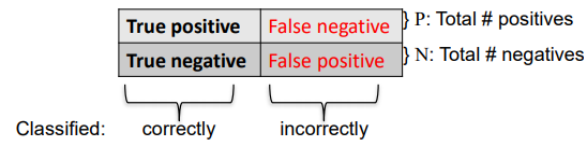
ROC = Receiver Operating Characteristic

An ROC curve characterizes the performance of a binary classifier. A binary classifier distinguishes between two different types of things e.g:

- Healthy/afflicted patients
- Pregnancy tests
- Object detection
- Foreground/background image pixels

2.2.1 Classification Error

Binary classifiers make errors. There are two inputs to a binary classifier i.e positives and negatives but there are four possible outcomes in any test:



2.2.2 ROC Curve

Characterizes the error trade-off in binary classification tasks. It plots the TP (True-positive) against the FP (False-positive) fraction

$$\text{TP fraction (sensitivity): } \frac{\text{True positive count}}{P}$$

$$\text{FP fraction (1-specificity): } \frac{\text{False positive count}}{N}$$

An ROC curve always passes through (0,0) and (1,1)

We choose an operating point by assigning a relative costs and values to each outcome:

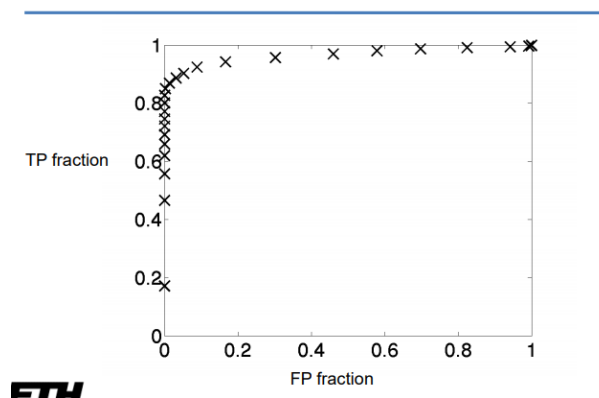
- V_{TN} Value of true negative
- V_{TP} Value of a true positive
- C_{FN} Cost of a false negative
- C_{FP} Cost of a false positive

We choose the point on the ROC curve with gradient:

$$\beta = \frac{N}{P} \frac{V_{TN} + C_{FP}}{V_{TP} + C_{FN}}$$

For simplicity we often set $V_{TN} = V_{TP} = 0$

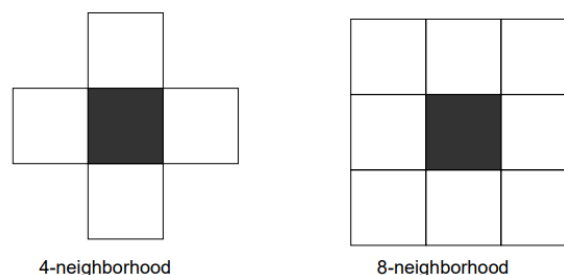
ROC curve



In reality we use 2-3 separate sets of test data:

1. **Training set:** For tuning the algorithm
2. **Validation set:** For tuning the performance score
3. **Test set:** To get a final performance score on the tuned algorithm

2.3 Pixel Connectivity:



Pixel Neighbourhood:

Pixel Paths:

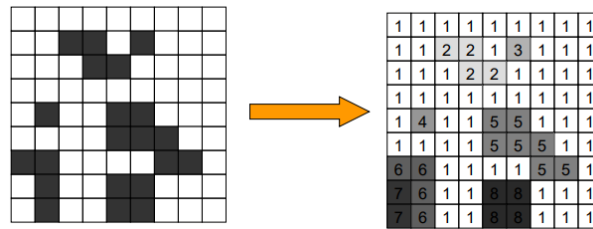
- A 4-connected path between pixels p_1 and p_n is a set of pixels $\{p_1, p_2, \dots, p_n\}$ such that p_i is a 4-neighbour of p_{i+1} $i=1, \dots, n-1$
- A 8-connected path, p_i is an 8-neighbour of p_{i+1}

Connected Regions:

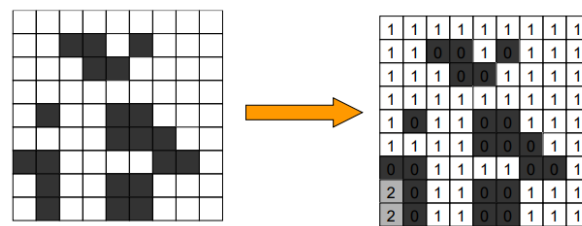
- A region is 4-connected if it contains a 4-connected path between any two of its pixels
- A region is 8-connected if it contains an 8-connected path between any two of its pixels

2.4 Image Labelling

Connected components Labelling: Labels each connected component of a binary image with a separate number



Foreground Labelling: Only extract the connected components of the foreground



Region Growing: Start from a seed point or region and add neighbouring pixels that satisfy the criteria defining a region, we repeat until we can include no more pixels. There are 3 meaningful variations when considering region Growing:

- Seed Selection i.e from where do we start the region growing
- Inclusion criteria
- Boundary constraints and snakes i.e how can the boundary evolve.

Seed selection: We can select a seed point or region by hand or automatically e.g from a conservative thresholding. Multiple seeds can be chosen as well.

Inclusion Criteria: Until now we have looked at Greylevel thresholding. Another possibility would be to use a Greylevel distribution model. We use a mean μ and a standard deviation σ in the seed region. We include the pixel if:

$$(I(x, y) - \mu)^2 < (n\sigma)^2$$

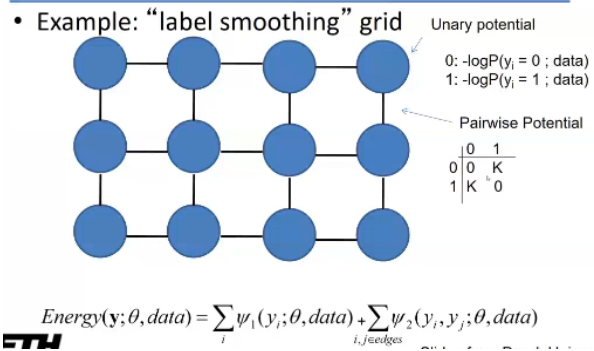
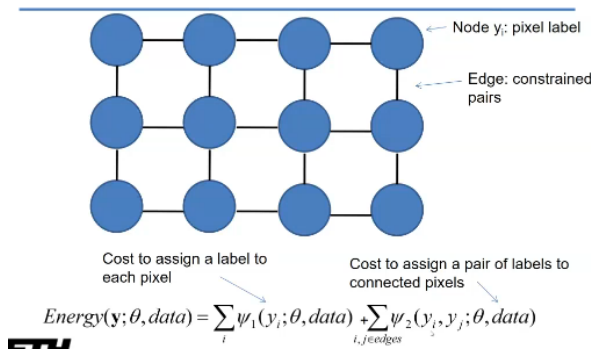
We can update the mean and standard deviation after every iteration.

snakes A snake is an active contour. It is a polygon i.e an ordered set of points joined up by lines. Each point on the contour moves away from the seed while its image neighborhood satisfies an inclusion criterion. contours usually have smoothness constraints. To avoid snakes the algorithm iteratively minimizes an energy function:

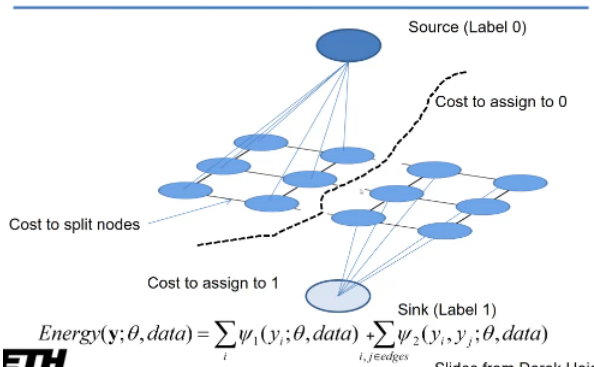
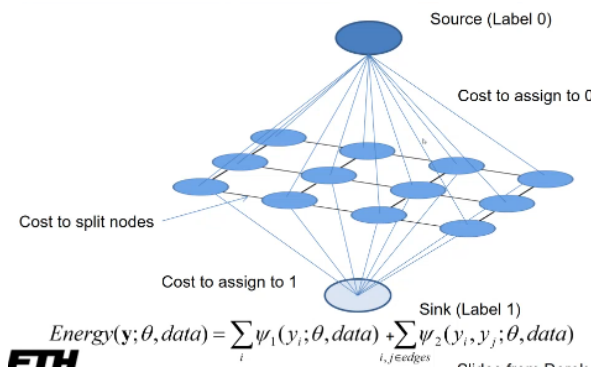
$$E = E_{tension} + E_{stiffness} + E_{image}$$

Stiffness and tension are used to keep the region from growing too much and grow in a regular way.

Markov Random Fields



Markov Random Fields: The Energy Function consists of two terms. The first term is an unary term i.e only considering the current pixel e.g whether or not the pixel is foreground (F) or background (BG) and the second term is a pairwise term which has 4 possible states: F F, F BG, BG F, BG BG. Usually we favor pixels agreeing with each other, hence these pixels will have a lower energy. Our goal is to minimize this energy function. One way to solve this is using graph cuts i.e Min-Cut Max Flow



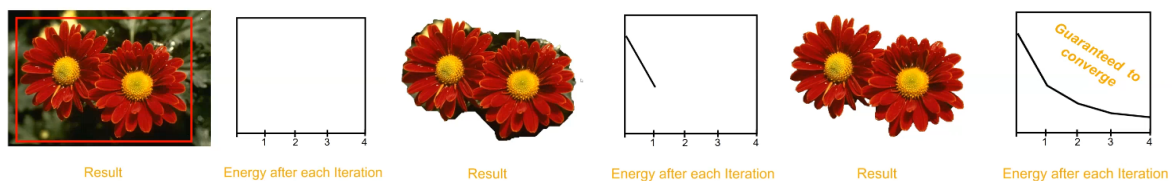
We want to find a cut such that all the foreground pixels are connected to one side and the background pixels connected to the other side while minimizing the separation of the two. The cost in the picture on the right is the pairwise cost of cutting the three boundary edges (3K) and the data cost for labelling each pixel to the source or the sink.

Terminology:

- **Cut:** Separates source from sink
- **Energy:** Collection of edges
- **Min Cut:** Global minimal energy

The Min-Cut can be found in polynomial time.

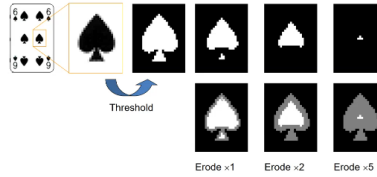
Iterated Graph Cut: Given the picture the user picks a rough boundary which contains all of the pixels which belong to the foreground. We look at the 2 distribution of colours inside and outside of the chosen boundary. We then apply **K-means clustering** which is a method in which we define K clusters and assign each pixel to one of the clusters. After assigning the clusters we check whether or not the cluster fits better to the distribution of the foreground or background and assign it to the corresponding match. The background model can then retrain itself with the newly assigned background pixels to achieve an even tighter foreground model. Using this method it is guaranteed that we saturate at a certain energy level and don't oscillate between values. The saturated energy value is not necessarily a local or global minimum.



2.5 Morphological Operations

Operations which are purely spatial. Hence we won't use data driven operations. They are local pixel transformations for processing region shapes. Most often used on binary images. Logical transformations based on comparison of pixel neighbourhoods with a pattern. Some examples:

- **8-neighbour erode (Minkowsky Subtraction)** if the current pixel has an 8-neighbour which belongs to the background add it to the background.



- **8-neighbour dilate (Minkowsky Addition)** Paint any background pixel that has one eight-connected neighbour that is foreground

Morphological Operation are used to smooth region boundaries for shape analysis and can be used to remove noise and artefacts from an imperfect segmentation.

Structuring Elements The structuring element is a binary array indicating pixels of interest in a region. A structuring element has an origin. Morphological operations take two arguments

- A binary image
- A structuring element

We compare the structuring element to the neighbourhood of each pixel, this determines the output of the morphological operation.

We define 3 operations (S is a structuring element) :

- **Fitting:** S fits I at \underline{x} if

$$\{y : y = \underline{x} + \underline{s}, \underline{s} \in S\} \subset I$$

- **Hitting:** S hits I at \underline{x} if

$$\{\underline{y} : \underline{y} = \underline{x} - \underline{s}, \underline{s} \in S\} \cap I \neq \emptyset$$

- **Missing:** S misses I at \underline{x} if

$$\{\underline{y} : \underline{y} = \underline{x} - \underline{s}, \underline{s} \in S\} \cap I = \emptyset$$

Image										Structuring element		
0	1	0	1	1	1	1	1	1		0	1	0
1	0	1	1	1	1	1	1	1		1	1	1
0	1	1	1	0	0	0	0	0		0	1	0
0	1	1	1	0	0	0	0	0		0	1	0
0	1	1	1	1	1	0	0	0				
0	1	0	1	0	1	0	0					
0	1	0	0	0	0	0	1	0				
0	1	0	0	0	0	0	0	0				

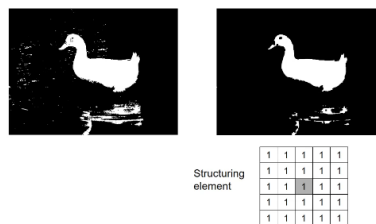
The above picture there are 3 regions of interest: upper left, upper right, down center:

- For the upper left region the structuring element origin placed in the center all the ones coincide with the ones in the region
- The upper right region demonstrates hits: We place the structuring element origin at all the squares with one e.g the middle square which is 0 is covered by a one in the structuring element and is hence a hit.
- The down center image demonstrates a miss. We place the structuring element origin at all the squares with a one but the middle 0 does not get covered with a one and hence its a miss.

Erosion: The image $E = I \ominus S$ is the erosion of image I by structuring element S:

$$E(\underline{x}) = \begin{cases} 1 & \text{if S fits I at } \underline{x} \\ 0 & \text{otherwise} \end{cases}$$

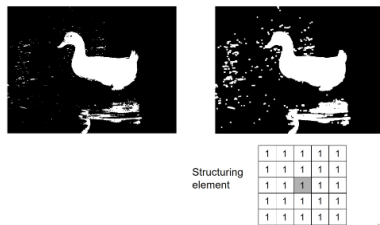
$$E = \{x : x + s \in I \text{ for every } s \in S\}$$



Dilation: The image $D = I \oplus S$ is the dilation of image I by structuring element S :

$$D(\underline{x}) = \begin{cases} 1 & \text{if } S \text{ hits } I \text{ at } \underline{x} \\ 0 & \text{otherwise} \end{cases}$$

$$D = \{\underline{x} : \underline{x} - \underline{s}, \underline{y} \in I \text{ and } \underline{s} \in S\}$$

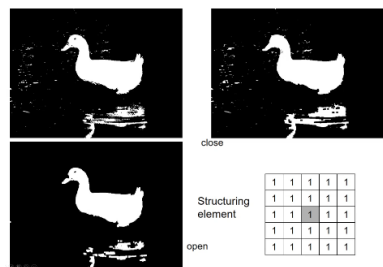


Opening The opening of I by S is

$$I \circ S = (I \ominus S) \oplus S$$

Closing The closing of I by S is

$$I \bullet S = (I \oplus S) \ominus S$$



Morphological filtering uses both opening and closing to remove holes in the foreground and islands in the background. The size and shape of the structuring element determine which features survive. In the absence of knowledge about the shape of features to remove, use a circular structuring element.

Granulometry: Provides a size distribution of distinct regions or "granules" in the image. We open the image with increasing structuring element size and count the number of regions after each operation. This creates a "morphological sieve". (e.g counting red blood cells)

Hit-and-miss transform Searches for an exact match of the structuring element. $H = I \otimes S$ is the hit-and-miss transform of image I by structuring element S . It is a simple form of template matching

$$\begin{array}{ccc} \begin{array}{|c|c|c|c|c|} \hline 1 & 0 & 1 & 1 & 1 \\ \hline 1 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 \\ \hline \end{array} & \otimes & \begin{array}{|c|c|c|} \hline 1 & 0 & 1 \\ \hline \end{array} \\ \hline \begin{array}{|c|c|c|c|c|} \hline 1 & 0 & 1 & 1 & 1 \\ \hline 1 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 \\ \hline \end{array} & \otimes & \begin{array}{|c|c|c|} \hline 1 & 0 & 1 \\ \hline * & 1 & \\ \hline \end{array} \end{array} = \begin{array}{ccc} \begin{array}{|c|c|c|c|c|} \hline 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 \\ \hline \end{array} & & \begin{array}{|c|c|c|c|c|} \hline 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \end{array}$$

The $*$ indicates a dont care hence can either be zero or one

Thinning and Thickening Defined in terms of the hit-and-miss transform:

- The thinning of I by S is:

$$I \oslash S = I \setminus (I \otimes S)$$

- The thickening of I by S is:

$$I \odot S = I \cup (I \otimes S)$$

We have the following equality:

$$I \odot S)^C = I^C \oslash S$$

In contrast to Erosion and Dilation, Thinning and thickening will stop at some point. Erosion and Dilation will continue until the whole image is either background or foreground only.

Skeletonization and the Medial Axis Transform (MAT): The MAT are stick-figure representations of a region $X \subset \mathbb{R}^2$. Essentially what it is, is starting a grassfire at the boundary of the region. The skeleton is the set of points at which two fire fronts meet

