Computer Graphics and Image Processing

Part 3 – Image Processing

Image Segmentation – Non contextual segmentation

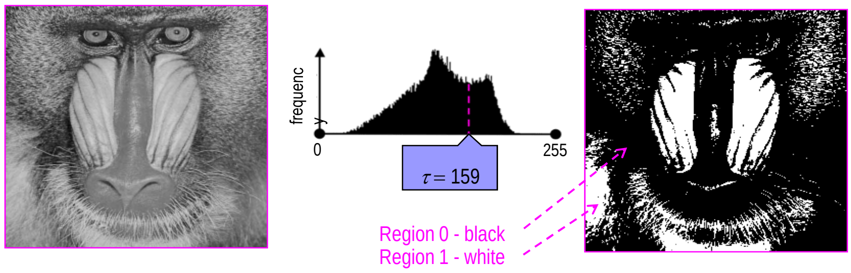
Image Segmentation Problem

- **Segmentation** partitions an image into distinct regions of pixels with similar attributes
  - Meaningful regions relate to objects or features of interest
    - Meaningful segmentation is the first step from low-level image processing transforming an image into one or more other images to high-level image description in terms of features, objects, and scenes
  - Success of image analysis depends on reliable segmentation
    - Accurate image partitioning is generally a very challenging problem!
  - Types of segmentation:
    - **Non-contextual**: grouping pixels with similar global features
    - **Contextual:** grouping pixels with similar features and in close locations

Non-contextual Thresholding

- **Thresholding** is the simplest non-contextual technique
- **Single threshold** τ: an image $f \rightarrow$ a binary region map g

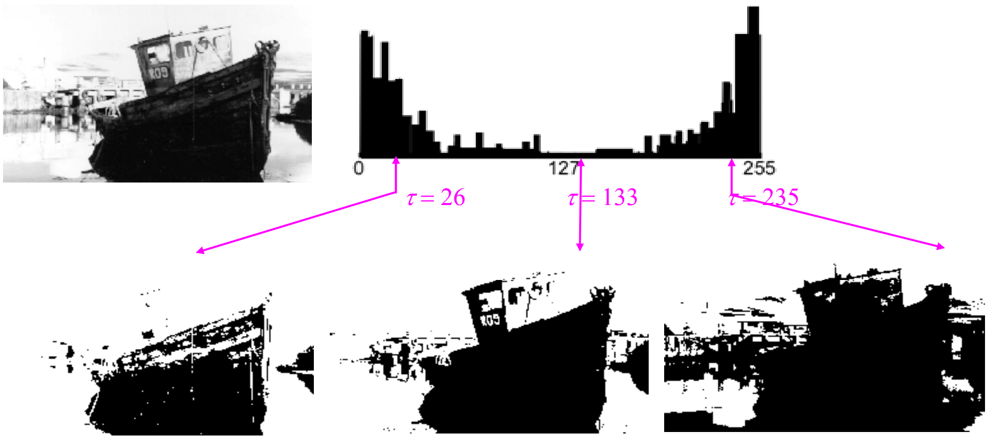
$$g(x,y) = \begin{cases} 0 & \text{if} \quad f(x,y) < \tau \\ 1 & \text{if} \quad f(x,y) \geq \tau \end{cases}$$



Simple Thresholding

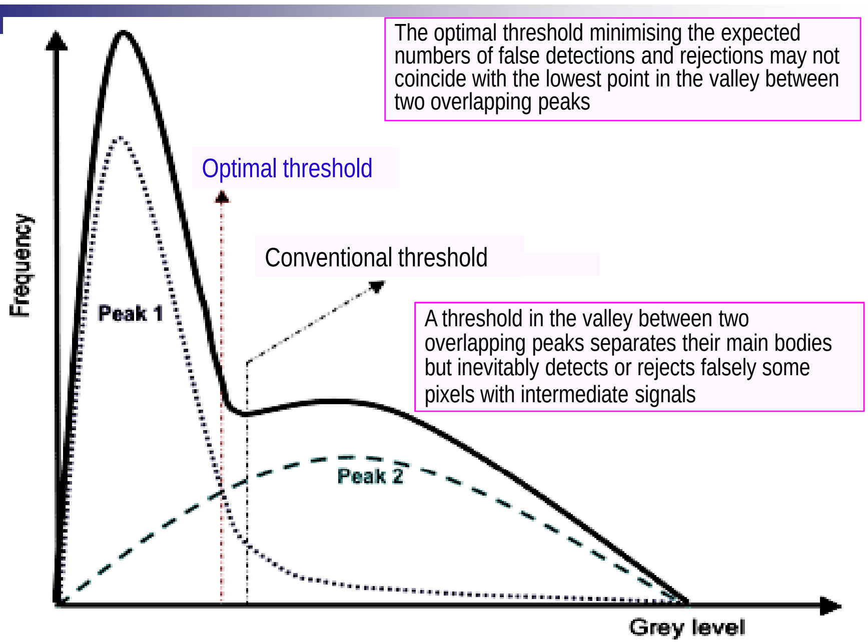
- For a single threshold, the binary map contains two possibly disjoint regions: one with pixel values smaller than a threshold and another with the pixel values at or above the threshold
  - The regions are labelled with 0 and 1, respectively
  - The segmentation depends on the image feature being compared to a threshold and on how the threshold is chosen
- Generally, two or more thresholds can produce more than two types of regions
  - Ranges of pixel values related to region type are separated by thresholds
  - In principle, one region may combine several ranges of pixel values: e.g. $g(x,y) = 0$ if $f(x,y) < \tau_1$ OR $f(x,y) > \tau_2$ and $g(x,y)=1$ if $\tau_1 \leq f(x,y) \leq \tau_2$

Simple Thresholding



Simple Thresholding

- **Main problems**: whether it is possible and, if yes, how to choose an adequate threshold or a number of thresholds to separate one or more desired objects from their background
- In many practical cases the simple thresholding is unable to segment objects of interest
- **General approach to thresholding**:
  - Images are assumed being **multimodal**, i.e. different objects of interest relate to distinct peaks, or modes of the 1D empirical signal histogram
  - The thresholds have to optimally separate these peaks in spite of typical overlaps between the signal ranges corresponding to individual peaks



Adaptive Thresholding

- Threshold separating a background from an object has to equalise two kinds of expected errors:
  - of assigning a background pixel to the object (*false alarm*)
  - of assigning an object pixel to the background (*missed object*)
- **Adaptive non-contextual separation** takes account of empirical probability distributions of object (e.g. dark) and background (bright) pixels
- More complex adaptation: a spatially variant threshold to account for local context ("background normalisation")

Adaptive Thresholding

- Simple iterative adaptation of the threshold: successive refinement of the estimated peak positions
- Basic assumptions:
  1. Each peak coincides with the mean grey level for all pixels that relate to that peak
  2. Pixel probability decreases monotonically on the absolute difference between the pixel and peak values both for an object and background peak
  3. Each grey level is associated with a peak by the threshold being on the half-way between the peaks

Adaptive Thresholding

- Iterative change of the threshold at each iteration j
  - Classification of each grey level $f(x,y)$ using the threshold T_j being computed at previous iteration:

$$(x,y) \in C_{j,\mathrm{ob}} \text{ if } f(x,y) \leq T_j; \quad (x,y) \in C_{j,\mathrm{bg}} \text{ if } f(x,y) > T_j$$

  - Mean grey values for each class:

$$\mu_{j,\mathrm{ob}} = \frac{1}{|C_{j,\mathrm{ob}}|} \sum_{(x,y)\in C_{j,\mathrm{ob}}} f(x,y); \quad \mu_{j,\mathrm{bg}} = \frac{1}{|C_{j,\mathrm{bg}}|} \sum_{(x,y)\in C_{j,\mathrm{bg}}} f(x,y)$$

  - Computation of the new threshold:

$$T_{j+1} = \frac{\mu_{j,\mathrm{ob}} + \mu_{j,\mathrm{bg}}}{2}$$

Where $|C_{j,\mathrm{bg}}|, |C_{j,ob}|$ represents the number of pixels in the background, respectively object, regions at iteration j

Adaptive Thresholding

- Only an image histogram is to be used

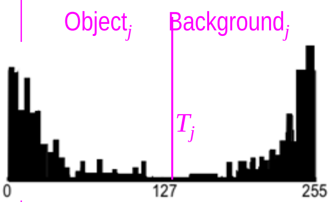
Input : an image histogram $\mathbf{h} = \{h(q) : q = 0,\ldots,255\}$

Initialisation : $j = 0;\quad N = \sum_{q=0}^{255} h(q);\quad T_0 = \frac{1}{N}\sum_{q=0}^{255} q h(q)$

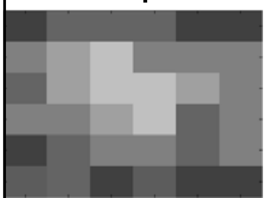
while $T_{j+1} \neq T_j$ **do**

$$\mu_{j,\text{ob}} = \frac{\sum_{q=0}^{T_j} q h(q)}{\sum_{q=0}^{T_j} h(q)};\quad \mu_{j,\text{bg}} = \frac{\sum_{q=T_j+1}^{255} q h(q)}{\sum_{q=T_j+1}^{255} h(q)};\quad T_{j+1} = \frac{\mu_{j,\text{ob}} + \mu_{j,\text{bg}}}{2}$$

end while



Adaptive Thresholding – Example (slide 16-18 LN 27)



$$\mu_{j,\text{ob}} = \frac{1}{|C_{j,\text{ob}}|}\sum_{(x,y)\in C_{j,\text{ob}}} f(x,y);\quad \mu_{j,\text{ob}} = \frac{1}{|C_{j,\text{bg}}|}\sum_{(x,y)\in C_{j,\text{bg}}} f(x,y)$$

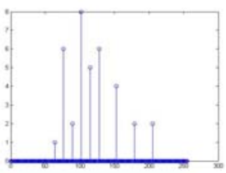
$$T_{j+1} = \frac{\mu_{j,\text{ob}} + \mu_{j,\text{bg}}}{2}$$

$$T_0 = round((64 + 6*76 + 2*89 + 8*102 + 5*115 + 6*128 + 4*153 + 2*179 + 2*205)/36) = 118$$

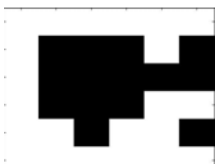
Value	Count
64	1
76	6
89	2
102	8
115	5
128	6
153	4
179	2
205	2

$$\mu_{0,\text{ob}} = round((64 + 6*76 + 2*89 + 8*102 + 5*115)/22) = 95$$

$$\mu_{0,\text{bg}} = round((6*128 + 4*153 + 2*179 + 2*205)/14) = 153$$



$$T_1 = \frac{\mu_{0,\text{ob}} + \mu_{0,\text{bg}}}{2} = 124$$



Colour Thresholding

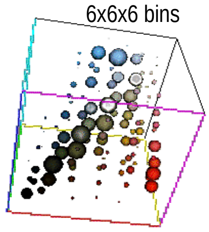
- Colour segmentation is more accurate due to more information per pixel
  - RGB colour space: correlated components
  - HSI (HSV) colour space: more stable and illumination-independent segmentation
- Segmentation ⇔ partitioning of the colour space
  - Thresholding of distances from pixel values $(R(x,y), G(x,y), B(x,y))$ to a reference colour $(R_0G_0B_0)$

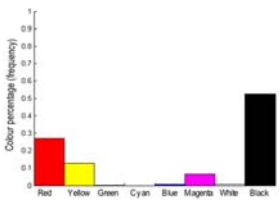
From http://www.matrix-vision.com/products/software

$$g(x,y) = \begin{cases} 1 & \text{if } d(x,y) \le d_{\max} \\ 0 & \text{if } d(x,y) > d_{\max} \end{cases}; \quad d(x,y) = \sqrt{(R(x,y) - R_0)^2 + (G(x,y) - G_0)^2 + (B(x,y) - B_0)^2}$$

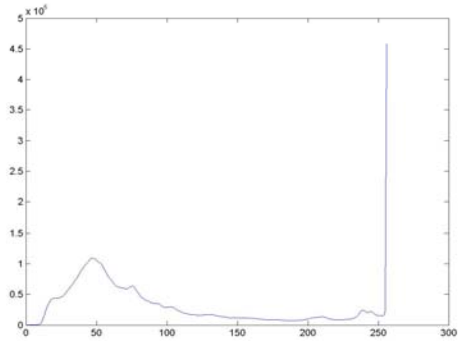
Colour Thresholding

- Colour histogram (3D or 2D projection)
  - Partitioning the colour space into bins (each bin - similar colours)
  - Dominant colours corresponding to peaks in a histogram
  - Pre-selected colours, e.g. the primary R, G, B, Y, C, M, white, and black





Example-Skyline-Histogram



Example-Skyline-Threshold

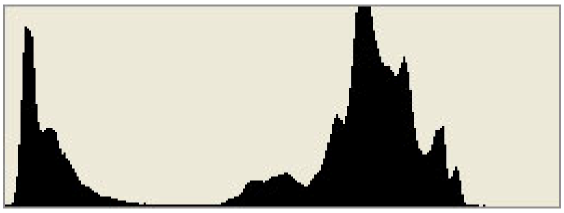
Combined R, G, and Binary image after thresholding: 0 if f(x,y)<128 and 1 if f(x,y)>=128

$$g(x,y) = \begin{cases} 1 & \text{if } d(x,y) \le d_{\max} \\ 0 & \text{if } d(x,y) > d_{\max} \end{cases};$$

$$d(x,y) = \sqrt{(\mathrm{R}(x,y) - \mathrm{R}_0)^2 + (\mathrm{G}(x,y) - \mathrm{G}_0)^2 + (\mathrm{B}(x,y) - \mathrm{B}_0)^2};$$

$$d_{\max} = \sqrt{3} * 128$$

Thresholding

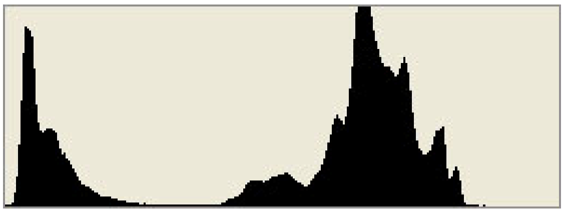


The histogram for the teapot image above exhibits a bi-modal distribution. What value(s) might be suitable for thresholding it into a binary image?

Thresholding

The binary image to the right was produced with a threshold value of 80.

Do you see anything wrong with the threshold we used?



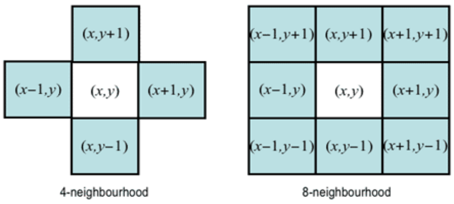
Pixel Neighbourhood

- Normal rectangular sampling pattern ⇔ a digital image on a finite arithmetic lattice
- Two types of neighbourhood of a pixel (x,y) in a lattice:
  - 4-neigbourhood:
  - 8-neighbourhood:

$$\{(x, y) : x = 1,2,\ldots, X\,;\, y = 1,2,\ldots, Y\}$$

$$\{(x, y \pm 1), (x \pm 1, y)\}$$

$$\{(x-1, y \pm 1), (x, y \pm 1), (x+1, y \pm 1), (x \pm 1, y)\}$$



Pixel Connectivity

- A **4-connected path** from a pixel p_1 to another pixel p_n is the sequence of pixels $\{p_1, p_2, \ldots, p_n\}$ such that p_{i+1} is a 4-neighbour of p_i for all $i = 1,\ \ldots,\ n$□1
  - The path is **8-connected** if p_{i+1} is an 8-neighbour of p_i
- A set of pixels is a **4-connected region** if there exists at least one 4-connected path between any pair of pixels from that set.
  - The **8-connected region** has at least one 8-connected path between any pair of pixels from that set.

Contextual Segmentation

Basic Approaches

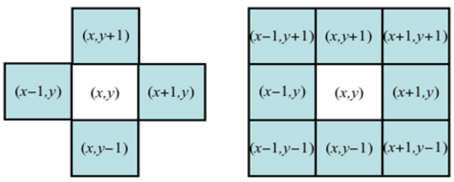
- **Non-contextual thresholding** groups pixels with no account of their relative locations in the image plane
- **Contextual segmentation** can be more successful in separating individual objects because it accounts for closeness of pixels that belong to an individual object.
- Two basic approaches to contextual segmentation:
  - based on **signal discontinuity**
  - based on **signal similarity**

Basic Approaches

- **Discontinuity-based segmentation** attempts to find complete boundaries of relatively uniform regions
  - **Assumption**: abrupt signal changes across each boundary
- **Similarity-based segmentation** attempts to directly create these uniform regions by grouping together ***connected*** pixels that satisfy certain *similarity criteria*
- Both the approaches mirror each other, in the sense that a complete boundary splits one region into two

Pixel Neighbourhood

- Normal rectangular sampling pattern ⇔ a digital image on a finite arithmetic lattice $\{(x, y) : x = 1,2,..., X ; y = 1,2,..., Y\}$
- Two types of neighbourhood of a pixel (x,y) in a lattice:
  - 4-neigbourhood: $\{(x, y \pm 1), (x \pm 1, y)\}$
  - 8-neighbourhood: $\{(x-1, y \pm 1), (x, y \pm 1), (x+1, y \pm 1), (x \pm 1, y)\}$



4-neighbourhood 8-neighbourhood

Pixel Connectivity

- A **4-connected path** from a pixel p_1 to another pixel p_n is the sequence of pixels $\{p_1, p_2, ..., p_n\}$ such that p_{i+1} is a 4-neighbour of p_i for all $i = 1, ..., n-1$
  - The path is **8-connected** if p_{i+1} is an 8-neighbour of p_i
- A set of pixels is a **4-connected region** if there exists at least one <u>4-connected path</u> between any pair of pixels from that set.
  - The **8-connected region** has at least one <u>8-connected path</u> between any pair of pixels from that set.

Region Similarity

- Uniformity / non-uniformity of pixels in a connected region is represented by a **uniformity predicate**
  - Logical statement, or condition being **true** if pixels in the regions are similar with respect to some property such as colour, grey level, edge strength, etc
- Common predicate: restricted signal variations over a pixel neighbourhood in a connected region R
  - Predicate $P(R)$ is **TRUE** if $|f(x,y) - f(x+\xi,y+\eta)| \leq \Delta$ and **FALSE** otherwise
    - (x,y) and $(x+\xi,y+\eta)$ - coordinates of the neighbouring pixels in R

Region Similarity

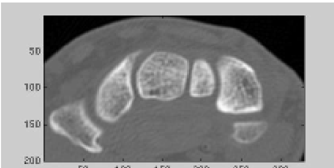
- The above simple predicate does not restrict the grey level variation within an entire region
  - Small changes in neighbouring signal values can accumulate over the region
- Intra-region signal variations can be restricted with a similar but non-local predicate:
  - $P(R)$ = **TRUE** if $|f(x,y) - \mu_R| \leq \varepsilon$ and **FALSE** otherwise
  - (x,y) is a pixel from the region *R* and μ_R is the mean value of signals $f(x,y)$ over the entire region R

Region Growing Segmentation

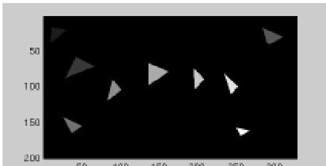
- The **bottom-up** algorithm
- **Initialisation:** a set of seed pixels defined by the user
- **Region growth:** sequentially add a pixel to a region under the following conditions:
  - The pixel has not been assigned to any other region
  - The pixel is a neighbour of that region
  - Addition of the pixel does not impact the uniformity of the growing region

Region Growing Segmentation

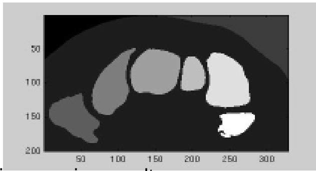
- Region growing is simple but unstable
  - It is very sensitive to a chosen uniformity predicate: small changes of the uniformity threshold may result in large changes of the regions found
  - Very different segmentation maps under different routes of image scanning, modes of exhausting neighbours of each region, seeds, and types of pixel connectivity



Greyscale image



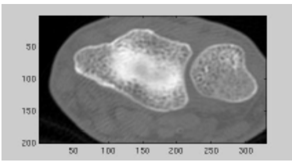
Seed regions

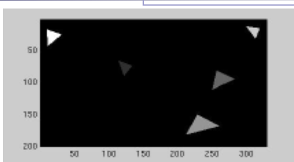


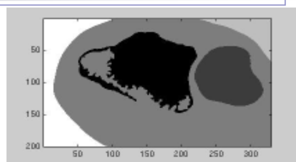
Region growing results

From http://www.lems.brown.edu/~msj/cs292/assign5/segment.html

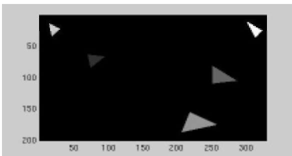
From http://www.lems.brown.edu/~msj/cs292/assign5/segment.html

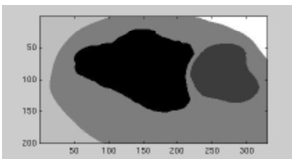






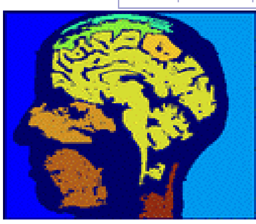
Region growing from two variants of seed regions

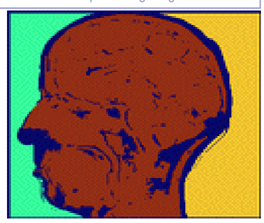




Growth of 4-connected and 8-connected regions

From http://www.comp.leeds.ac.uk/ai21/examples/images/rgrow.html





Complete Segmentation Criteria

1. **All pixels have to be assigned to regions**
2. **Each pixel has to belong to a single region only**
3. **Each region is a connected set of pixels**
4. **Each region has to be uniform with respect to a given predicate**
5. **Any merged pair of adjacent regions has to be non-uniform**

Region Growing: Properties

- **Region growing** satisfies the 3rd and 4th criteria (each region is connected and uniform), but not the others
- The 1st and 2nd criteria are not satisfied
  - In general, the number of seeds may not be sufficient to create a region for every pixel
- The 5th criterion may not hold
  - Regions grown from two nearby seeds are always regarded as distinct, even if those seeds are chosen within a potentially uniform part of the image