Image segmentation

SCC0251/5830 – Image Processing

Prof. Moacir A. Ponti

Instituto de Ciências Matemáticas e de Computação – USP

2020

Agenda

Definition

- Divides the image into parts / regions
  - pixels of a region are correlated according to some criterion
  - usually with some semantics

original

Definition

- Divides the image into parts / regions
  - pixels of a region are correlated according to some criterion
  - usually with some semantics

original

segmented regions

Definition

- Let R be the region occupied by all image
- Segmentation is a process that partitions R in n subregions: $R_1, R_2, \ldots, R_n$, so that
  1. $\bigcup_{i=1}^{n} R_i = R$;
  2. R_i is conected;
  3. $R_i \cap R_j = \varnothing$ for all i, j with $i \neq j$
  4. there is a criterion Q so that
     - $Q(R_i) =$ `true` and
     - $Q(R_i \cup R_j) =$ `false` for two adjacent regions

Segmentation

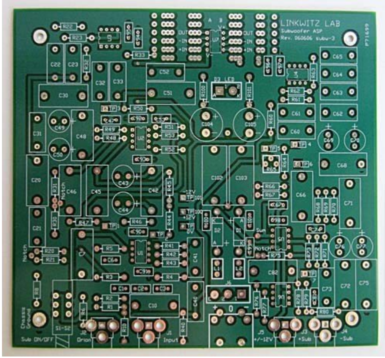
- Subjective and one of the most difficult tasks in image processing
  - application dependent;
  - success often requires *a priori* knowledge.

Segmentation

- Subjective and one of the most difficult tasks in image processing
  - application dependent;
  - success often requires *a priori* knowledge.
- It needs a well-formulated problem so that we use the appropriate method

Example 1: circuit

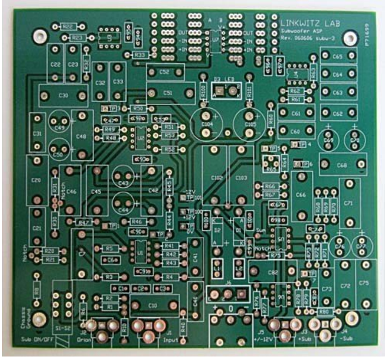
- task: segment terminations and printed components



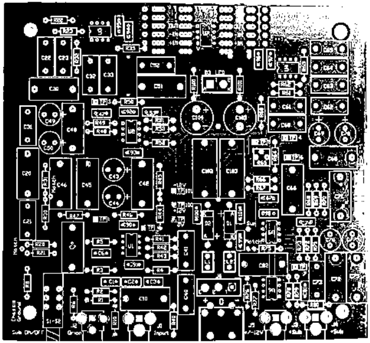
original

Example 1: circuit

- task: segment terminations and printed components



original



segmentation

Example 2: clouds

- segment clouds

Original

Example 2: clouds

- segment clouds

Original

Segmentation

Methodology for segmentation

- **Global knowledge** (*histogram-based*), looks for thresholds in intensities;
- **Edge-based**, search for discontinuities between neighbours;
- **Region-based**, connects pixels that are similar in some neighborhood;
- **Model-based**, search for patterns with a pre-defined model;
- **Connectivity-based**, models image via a graph or network;
- **PDE-based**, tries to solve a numerical solution for a PDE given some criterion.
- ...

Methodology for segmentation

- **Global knowledge** (*histogram-based*), looks for thresholds in intensities;
- **Edge-based**, search for discontinuities between neighbours;
- **Region-based**, connects pixels that are similar in some neighborhood;
- **Model-based**, search for patterns with a pre-defined model;
- **Connectivity-based**, models image via a graph or network;
- **PDE-based**, tries to solve a numerical solution for a PDE given some criterion.
- ...

Agenda

Thresholding

- Separate pixels in regions given their intensity/color.
- Threshold can be obtained in a manual or authomatic method (example: Otsu)
- Adaptive thresholding is also possible, segmenting image by regions

Thresholding

- Separate pixels in regions given their intensity/color.
- Threshold can be obtained in a manual or authomatic method (example: Otsu)
- Adaptive thresholding is also possible, segmenting image by regions

$$T(r) = \begin{cases} V, & \text{if } r \geq L \\ 0, & \text{otherwise} \end{cases}$$

Thresholding

- Separate pixels in regions given their intensity/color.
- Threshold can be obtained in a manual or authomatic method (example: Otsu)
- Adaptive thresholding is also possible, segmenting image by regions

$$T(r) = \begin{cases} V, & \text{if } r \geq L \\ 0, & \text{otherwise} \end{cases}$$

- L defined so that it separates the regions of interest, V is the value set for those pixels above the threshold.

Thresholding: Otsu

- Presented by Nobuyuki Otsu.
- Assumes that image has a bi-modal histogram;
- Computes the optimal threshold separating the intensities in two classes so that it :
  - **minimizes intra-class variance**.
  - **maximizess variance among classes**.

Thresholding: Otsu

- Presented by Nobuyuki Otsu.
- Assumes that image has a bi-modal histogram;
- Computes the optimal threshold separating the intensities in two classes so that it :
  - **minimizes intra-class variance**.
  - **maximizess variance among classes**.

Basic algorithm

1. Compute histogram
2. For each intensity i, compute intra-class variance $\sigma_W^2(i)$
3. Use as threshold intensity $T = \arg\min_i \left[\sigma_W^2(i)\right]$

Thresholding: Otsu

- Intraclass variance (within class variance), for a threshold L of an image with M pixels, is the weighed sum of the variances of class a/b:

$$\sigma_W^2 = W_a\sigma_a^2 + W_b\sigma_b^2$$

Thresholding: Otsu

- Intraclass variance (within class variance), for a threshold L of an image with M pixels, is the weighed sum of the variances of class a/b:

$$\sigma_W^2 = W_a \sigma_a^2 + W_b \sigma_b^2$$

- Weigh for class a computed by the sum of frequencies (histogram) of values 0 to $L-1$:

$$W_a = \frac{1}{M} \sum_{i=0}^{L-1} h(i)$$

Thresholding: Otsu

- Intraclass variance (within class variance), for a threshold L of an image with M pixels, is the weighed sum of the variances of class a/b:

$$\sigma_W^2 = W_a \sigma_a^2 + W_b \sigma_b^2$$

- Weigh for class a computed by the sum of frequencies (histogram) of values 0 to $L-1$:

$$W_a = \frac{1}{M} \sum_{i=0}^{L-1} h(i)$$

- Weigh for class b computed by the sum of frequencies (histogram) of values L to 255:

$$W_b = \frac{1}{M} \sum_{i=L}^{255} h(i)$$

Thresholding: Otsu

- Mean (used in variance) for class *a*:

$$\mu_a = \frac{\sum_{i=0}^{L-1}(i \times h(i))}{\sum_{i=0}^{L-1} h(i)}$$

- Variance of class *a*:

$$\sigma_a^2 = \frac{\sum_{i=0}^{L-1}\left[(i - \mu_a)^2 \times h(i)\right]}{\sum_{i=0}^{L-1} h(i)}$$

Thresholding: Otsu

- Mean (used in variance) for class b:

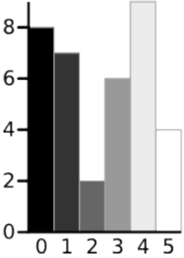
$$\mu_b = \frac{\sum_{i=L}^{255}(i \times h(i))}{\sum_{i=L}^{255} h(i)}$$

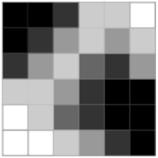
- Variance of class b:

$$\sigma_b^2 = \frac{\sum_{i=L}^{255}\left[(i - \mu_b)^2 \times h(i)\right]}{\sum_{i=L}^{255} h(i)}$$

Thresholding: Otsu — example

Histogram and image 6×6 with 6 colors

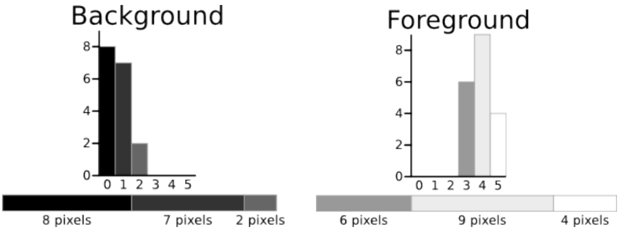




thanks to A.Greenspan (www.labbookpages.co.uk)

Thresholding: Otsu — example

For $L = 3$:



- Within class variance: $\sigma_W^2 = W_a\sigma_a^2 + W_b\sigma_b^2$

thanks to A .Greenspan (www.labbookpages.co.uk)

Thresholding: método de Otsu — example

Within class variance

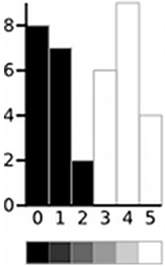
L	0	1	2	3	4	5
σ_W^2	3.1196	1.5268	0.5561	0.4909	0.9779	2.2491

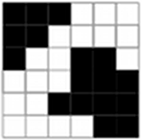
thanks to A.Greenspan (www.labbookpages.co.uk)

Thresholding: método de Otsu — example

Within class variance

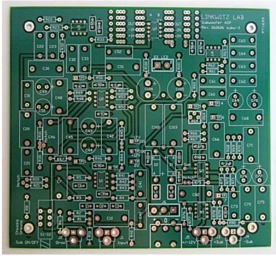
L	0	1	2	3	4	5
σ_W^2	3.1196	1.5268	0.5561	0.4909	0.9779	2.2491



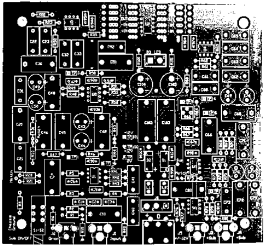


thanks to A.Greenspan (www.labbookpages.co.uk)

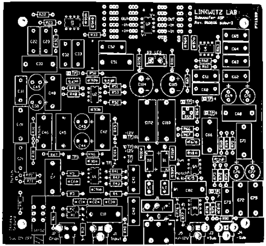
Thresholding: método de Otsu — example



Original



Average threshold

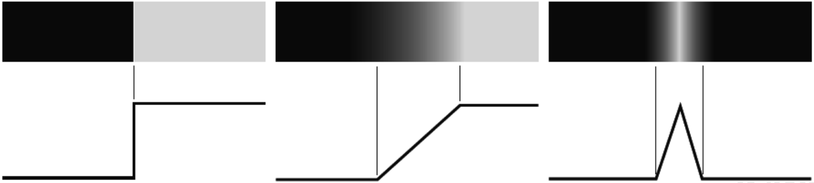


Global optimal threshold

Agenda

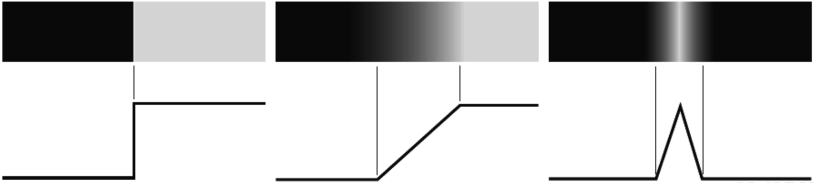
Edge detection

- Use as basis transitions of intensities
- Search for the borders of regions by looking at discontinuities



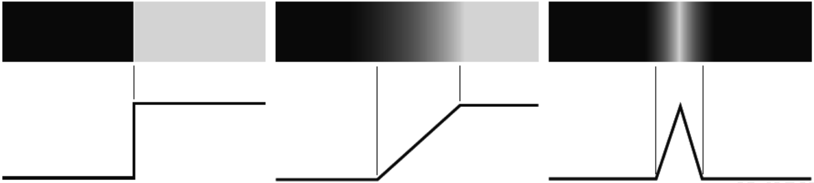
Edge detection

- Use as basis transitions of intensities
- Search for the borders of regions by looking at discontinuities
- Can be seen as a high-pass filter.
  1. Laplacian
  2. Sobel
  3. Prewitt



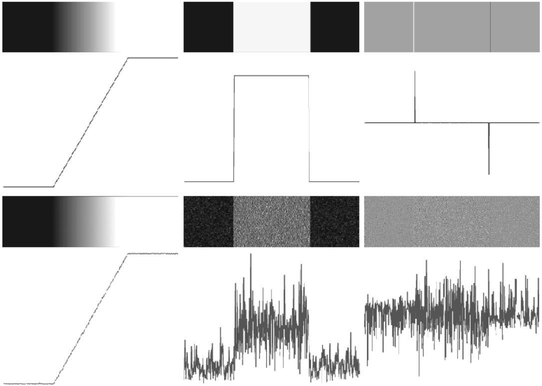
Edge detection

- Use as basis transitions of intensities
- Search for the borders of regions by looking at discontinuities
- Can be seen as a high-pass filter.
  1. Laplacian
  2. Sobel
  3. Prewitt
- After filtering, apply threshold to extract edges.



Edge detection

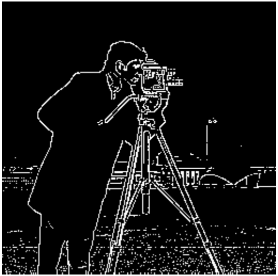
- Because edge detection is based on derivatives, it can easily degrade with varations and noise.



Edge detection

- **Laplacian**: isotropic differential operator

$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix}$$



Edge detection

- Directed edge detection

$$\begin{matrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{matrix} \qquad \begin{matrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{matrix} \qquad \begin{matrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{matrix} \qquad \begin{matrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{matrix}$$

Horizontal $\qquad +45^\circ \qquad$ Vertical $\qquad -45^\circ$

Horizontal

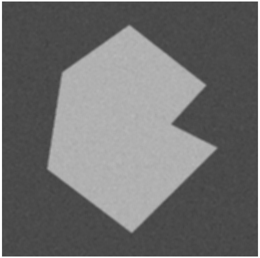
45°

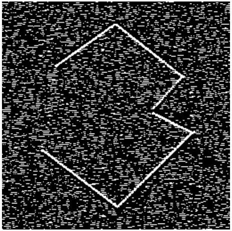
Edge detection

- **Laplacian of Gaussian** (LoG): a combined filter with smoothing followed by a Laplacian filter

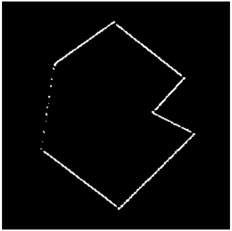
$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]\exp^{-\frac{x^2+y^2}{2\sigma^2}}.$$

Edge detection: example

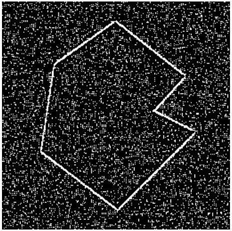




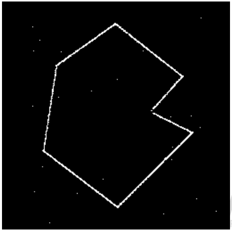
Horizontal



45°



Laplacian



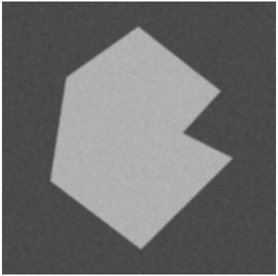
LoG

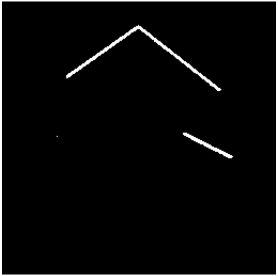
Edge detection: Sobel

- Sobel filters are widely used for edge detection, giving more weight to the central pixels

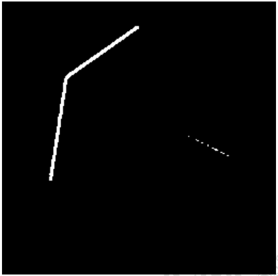
$$\begin{array}{ccc} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{array} \quad \begin{array}{ccc} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{array} \quad \begin{array}{ccc} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{array} \quad \begin{array}{ccc} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{array}$$

Horizontal Vertical Diagonal Diagonal





Horizontal



Diagonal

Agenda

Region-based segmentation

- Find regions directly via pixel inspection
  1. Region growing
  2. Split and merge

Region growing: pixel aggregation

- From seed pixels, allow regions to grow, by adding to the seed neighbour pixels that have similar characteristics

Region growing: pixel aggregation

- From seed pixels, allow regions to grow, by adding to the seed neighbour pixels that have similar characteristics
- Elements to define:
  - Method for seed initilization;
  - Number of seeds per region;
  - Similarity criterion;
  - Stoping condition.

Region growing: pixel aggregation

- Seeds can be chosen manually or based on filtering operations,

Region growing: pixel aggregation

- Seeds can be chosen manually or based on filtering operations,
- Similarity can be intensity difference or also based on a local statistic,

Region growing: pixel aggregation

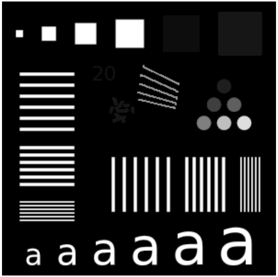
- Seeds can be chosen manually or based on filtering operations,
- Similarity can be intensity difference or also based on a local statistic,
- The standard stopping condition is to end the process when no new pixel can be added to the region. But there are scenarion in which pixels change region in every iteration,

Region growing: pixel aggregation

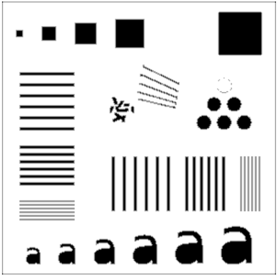
- Seeds can be chosen manually or based on filtering operations,
- Similarity can be intensity difference or also based on a local statistic,
- The standard stopping condition is to end the process when no new pixel can be added to the region. But there are scenarion in which pixels change region in every iteration,
- The original algorithm has recursive nature, but it often safer to implement a virtual stack, in order to avoid stack overflow due to the limit of recursive calls.

Pixel aggregation using 1 seed

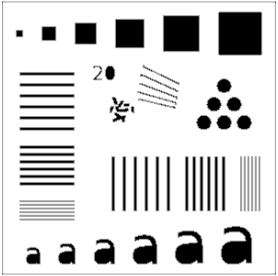
One seed at $(1, 1)$ and using difference of intensities as criterion for inclusion.



Original

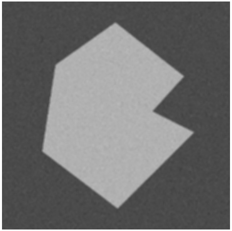


$Q = 5$

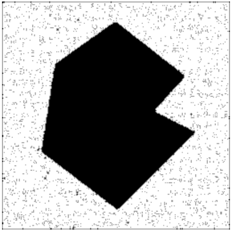


$Q = 20$

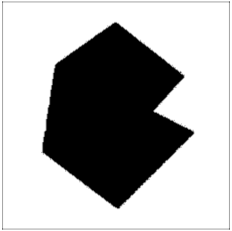
Pixel aggregation using 1 seed



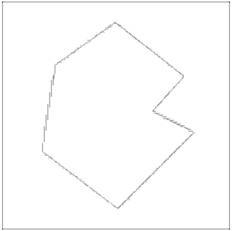
Original



$Q = 3$



$Q = 10$



$Q = 20$

Pixel aggregation using 1 pixel seed

Sometimes it is not adequate to the problem

Original

$Q = 30$

$Q = 40$

Watershed transformation

- Views the image as a topographic surface: low intensities are valleys, high intensities hills or peaks,
- Starts by filling the valleys (local minima) with different "colored water", which are seeds for regions,

Watershed transformation

- Views the image as a topographic surface: low intensities are valleys, high intensities hills or peaks,
- Starts by filling the valleys (local minima) with different "colored water", which are seeds for regions,
- When different water meet, some barrier is build to prevent them form merging,

Watershed transformation

- Views the image as a topographic surface: low intensities are valleys, high intensities hills or peaks,
- Starts by filling the valleys (local minima) with different "colored water", which are seeds for regions,
- When different water meet, some barrier is build to prevent them form merging,
- This process of "filling" and "barrier construction" continues until all peaks are under water, then the barriers represent the segmentation

Hough Transform

- Isolate features of any particular shape in an image, as long as you can represent that shape in a mathematical (parametric) form.
- The *classical Hough* transform is most commonly used for the detection of regular curves: lines, circles and ellipses.

Hough Transform

- Isolate features of any particular shape in an image, as long as you can represent that shape in a mathematical (parametric) form.
- The *classical Hough* transform is most commonly used for the detection of regular curves: lines, circles and ellipses.
- A line can be represented as: $y = mx + c$, or in parametric form as:

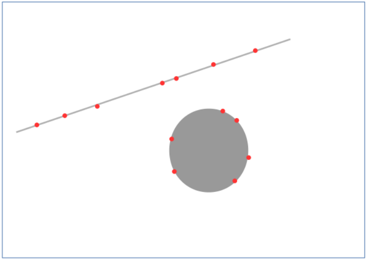
$$\rho = x \cos\theta + y \sin\theta$$

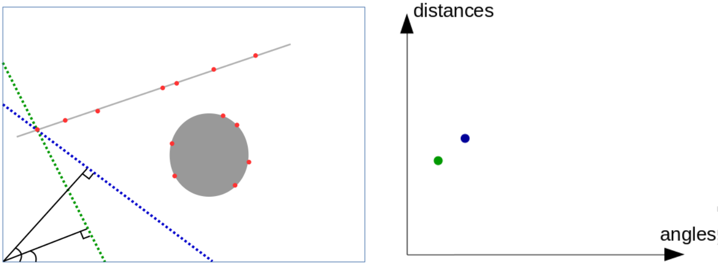
where ρ is the perpendicular distance from origin to the line, θ is the angle formed by the perpendicular line and the horizontal axis.

Hough Transform: Line

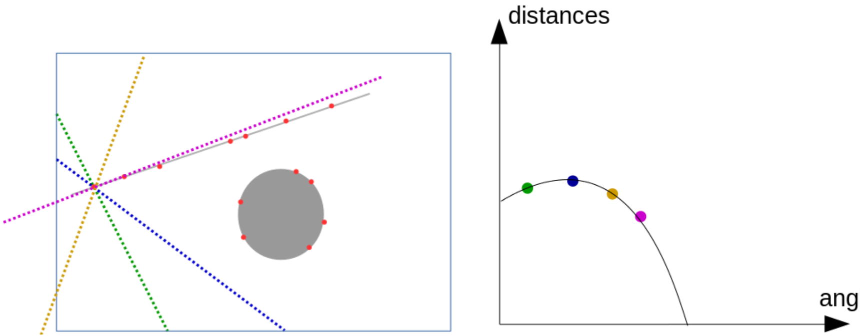
- The **accumulator** (matrix) is used to detect the existence of a line described by $\rho = x\cos\theta + y\sin\theta$, i.e. it is a ρ–θ space.

Hough Transform: Line





Hough Transform: Line



Bibliography I

GONZALEZ, R.C.; WOODS, R.E. ⋆
Processamento Digital de Imagens, 3.ed
Capítulo 10.
Pearson, 2010.

GREENSTED, A.
Otsu Thresholding
http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html