

Binarització de la imatge



Introducció

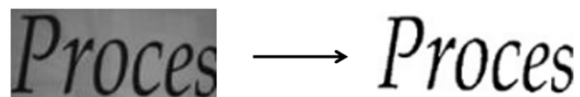
- La segmentació consisteix en dividir la imatge en regions amb característiques similars
- Cada regió es representa com a una vora tancada
- Els problema sorgeix quan les imatges són sorolloses
- 3 grans famílies:
 - Binaritzat
 - Segmentació basada en contorns
 - Segmentació basada en regions



Binaritzat

Binarization

- Reduce number of grey levels to **two** (foreground and background)



- Objects segmented from background
- Textual images or document images
- Much 'information' can be lost by binarization
- Binary images are often simpler to process than grayscale images
- These arguments need to be traded off before choosing between binary and gray scale image processing approaches

Binaritzat

- Procés més senzill per a segmentar
- Ràpid, fàcilment implementable en temps real
- Es discrimina entre objectes i fons usant una constant anomenada *threshold*:

$$\begin{aligned} g(i,j) &= 1 && \text{si } f(i,j) \geq T \\ g(i,j) &= 0 && \text{si } f(i,j) < T \end{aligned}$$

- El threshold es pot fixar a tota la imatge (threshold global) o pot variar en parts diferents de la imatge (threshold local)
- El thresholds globals rarament funcionen bé. La il·luminació no és mai uniforme.

Binaritzat



UPC

Binaritzat

- Podem usar threshold local: La imatge es divideix en subimatges i cada una es processa amb el seu threshold.
- Podem usar doble threshold:

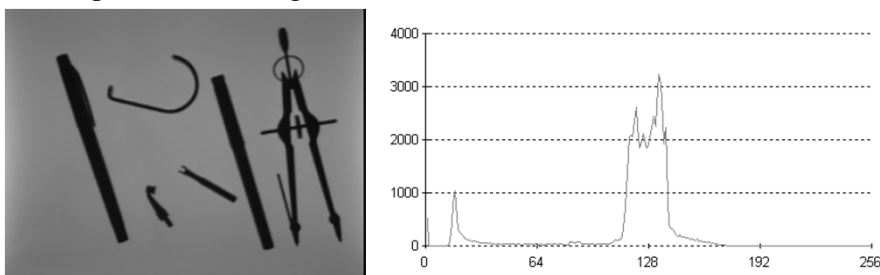
$$\begin{aligned} g(i, j) &= X && \text{si } f(i, j) \geq T_H \\ g(i, j) &= Y && \text{si } f(i, j) < T_L \\ g(i, j) &= Z && \text{si } T_H > f(i, j) \geq T_L \end{aligned}$$

- Una possible millora:
 - 1. Seleccionar dos llindars: T_L i T_H .
 - 2. Dividir la imatge en 3 regions:
 - R_L conté els píxels amb nivell de gris $< T_L$
 - R_M conté els píxels amb nivells de gris entre T_L i T_H
 - R_H conté els píxels amb nivells de gris $> T_H$
 - 3. Recórrer tots els píxels de R_M .
 - Si el píxel té un veí en R_L , llavors cal reassignar-lo a R_L .
 - 4. Repetir el pas 3 fins que no es moguin píxels de lloc.
 - 5. Reassignar tots els píxels que queden en R_M a R_H .

UPC

Selecció del llindar

- **P-tile-thresholding**: escollir el llindar que deixa p% píxels per sobre.
- A partir de l'**histograma bimodal**:



- L'histograma no sempre és bimodal
- És difícil determinar si un histograma és bimodal
- Un histograma bimodal no garanteix una bona segmentació entre objectes i fons.



Selecció del llindar

- **Optimal thresholding**: aproximar l'histograma com la suma ponderada de dues normals.

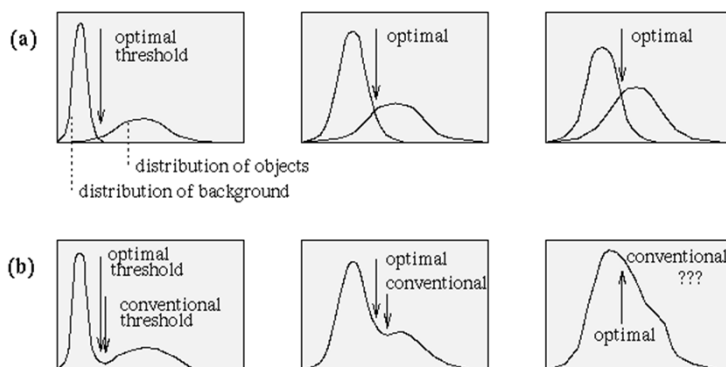


Figure 5.4 Grey level histograms approximated by two normal distributions; the threshold is set to give minimum probability of segmentation error: (a) Probability distributions of background and objects, (b) corresponding histograms and optimal threshold.



Selecció del llindar

- Otsu thresholding

- The aim is to find a threshold that minimizes the intra-class variance (i.e. minimum variation in the foreground + minimum variation in the background)
- Question: How?
- Answer: Exhaustive search.
- Within-class variance = foreground variance + background variance
- Minimizing the within-class variance is the same as maximizing the between-class variance

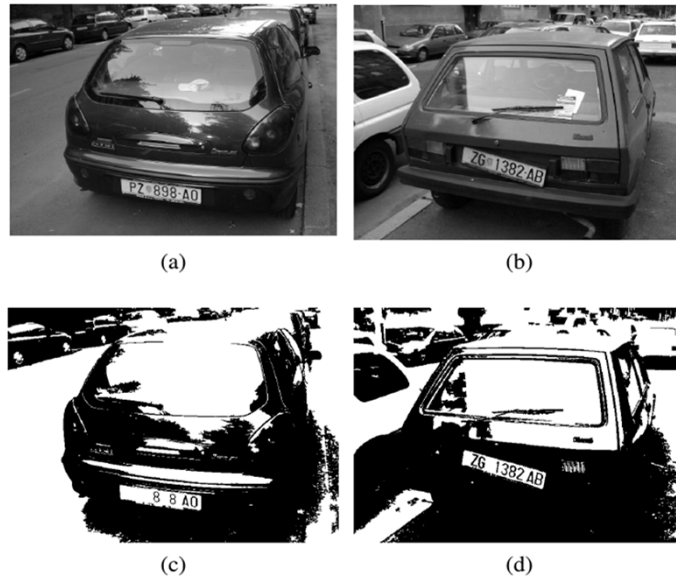
Selecció del llindar

Algorithm: Otsu Thresholding

1. Compute histogram and probabilities of each intensity level
2. Initialize class probabilities $w_i(0)$ and class means $\mu_i(0)$
3. Step through all possible thresholds $t = 1$ to 255
 1. Update w_i and μ_i
 2. Compute $\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t)$
4. The value of t which gives maximum $\sigma_b^2(t)$ is the desired Otsu's threshold
5. $\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t)$
6. $\sigma_b^2(t) = w_1(t)w_2(t)[\mu_1(t) - \mu_2(t)]^2$

<http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>

Otsu's results



UPC

Local Thresholding

➤ Niblack's method: At each pixel position determine:

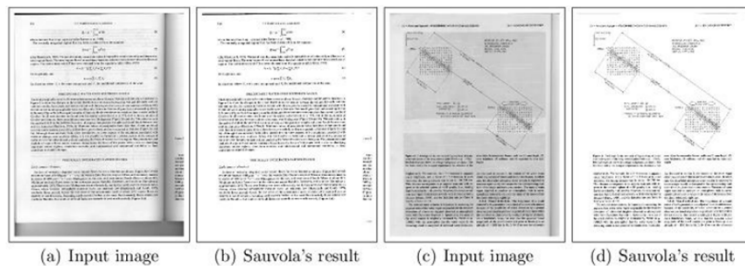
- mean μ in a region of e.g. 15×15 pixels
- standard deviation σ in the same region
- set local threshold to $t = \mu - 0.2\sigma$

➤ Sauvola's method is a variant of Niblack's

$$t = \mu (1 + k(\sigma/r - 1))$$

$$k = 0.5$$

$$r = 128$$



Resum de tècniques

Globals:

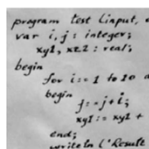
- Usant els pixels de contorn: Wezska (laplacià),
- Selecció iterativa: Ridler, Thrussel, ...
- Basats en histograma: Otsu, Dong, Kittler (min error)...
- Basats en la entropia: Pun, Kapur, ...
- Fuzzy sets: Huang, ...

Locals

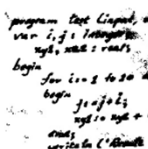
- Chow-Kaneko, moving averages...



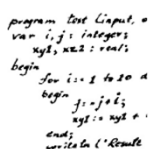
Resum de tècniques



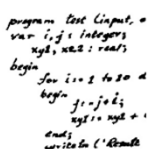
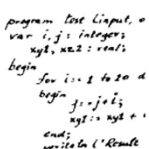
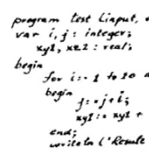
Original



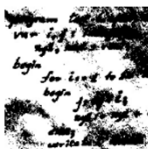
Mean value = 175



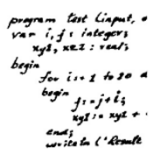
Two peaks = 132

Edge
pixels = 156Iterative
Selection = 121

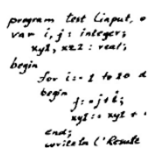
GLH = 122



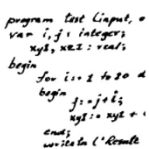
Pun = 184



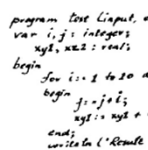
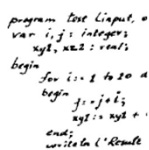
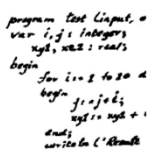
Kapur = 153



Johannsen = 132



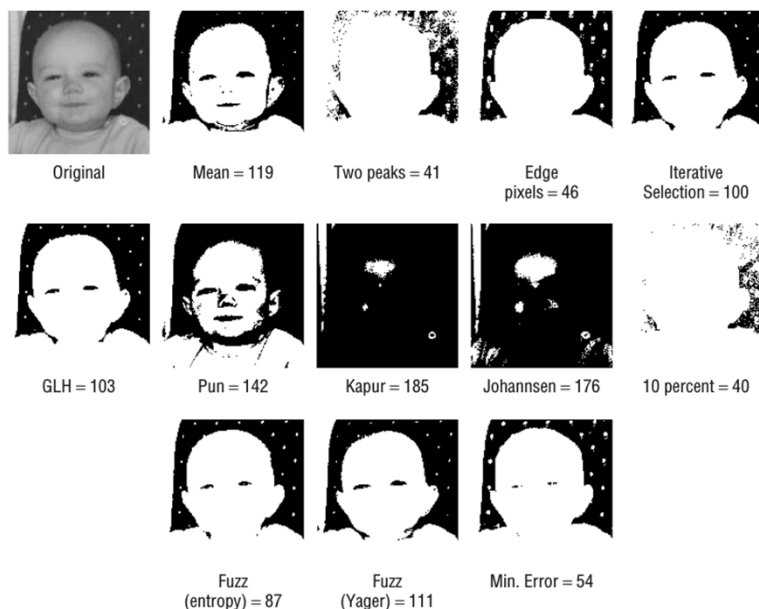
10 percent = 147

Fuzz
(entropy) = 131Fuzz
(Yager) = 124

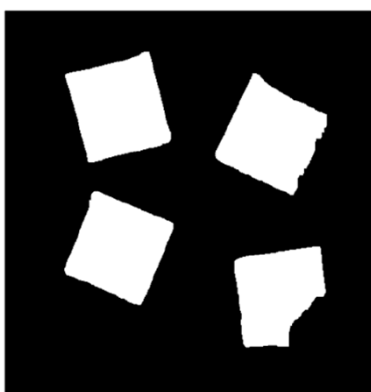
Min. Error = 166



Resum de tècniques



Anàlisi d'imatges binàries

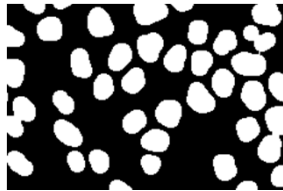


How many objects are in this image?



Labelling

- Operació que assigna un valor únic al píxels que pertanyen a la mateixa regió connexa.



- És un operador de pre-processat. Ens permetrà fer un anàlisi individual de les cel·les posteriorment.

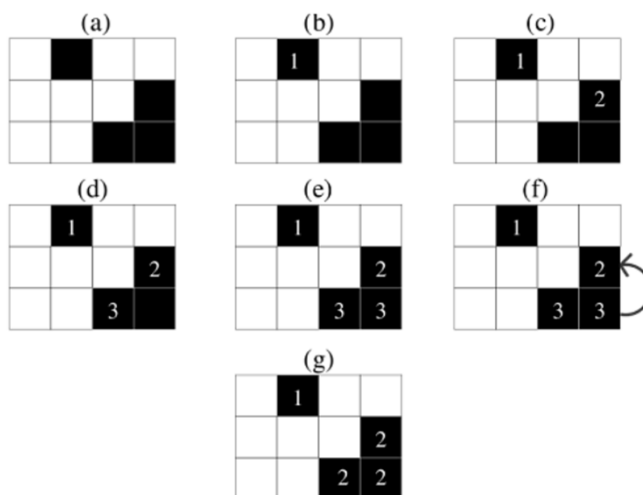


Algorisme de labelling

- Scannejar la imatge per files
- Quan es trobi un nou pixel de foregrounds, assignar-li una etiqueta:
 - a) Si el pixel de l'esquerra era de fons, assignar-li una etiqueta nova.
 - b) Si el pixel de l'esquerra era del blob, assignar-li la mateixa etiqueta.
- Si el pixel veí de dalt era del blob, fusionar les etiquetes.

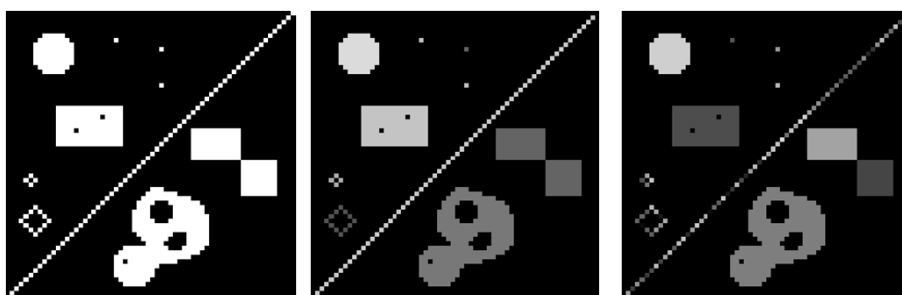


Algorisme de labelling



Labelling

- Connectivitat 4 vs connectivitat 8



Imatge binària

Etiquetada C-8

Etiquetada C-4

- En la imatge etiquetada amb connectivitat 8 apareixen 10 regions connexes més el fons. En la etiquetada amb connectivitat 4 apareixen 86 regions connexes més el fons.

