

Binarització de la imatge

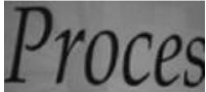

Introducció

- La segmentació consisteix en dividir la imatge en regions amb característiques similars
- Cada regió es representa com a una vora tancada
- 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



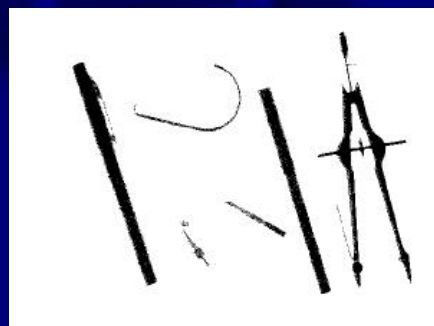
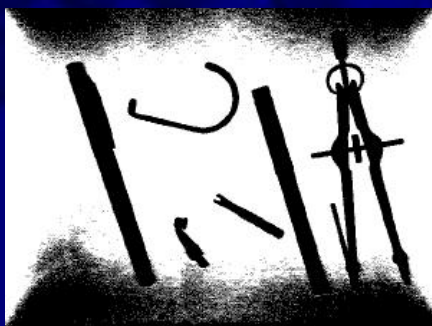
Binaritzat

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

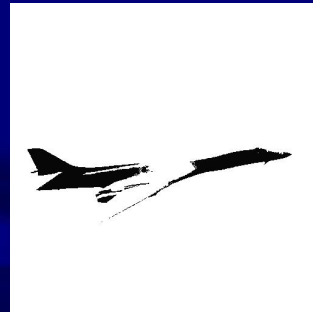
$$\begin{aligned} g(i, j) &= 1 && \text{si } f(i, j) \geq T \\ g(i, j) &= 0 && \text{si } f(i, j) < T \end{aligned}$$

- El thresholds globals rarament funcionen bé. La il·luminació no és mai uniforme.

Binaritzat per llindar fixe



Binaritzat per llindar fixe



- No sol funcionar.
(de fet, no funciona gaire bé mai)

Binaritzat per llindar. Alternatives

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

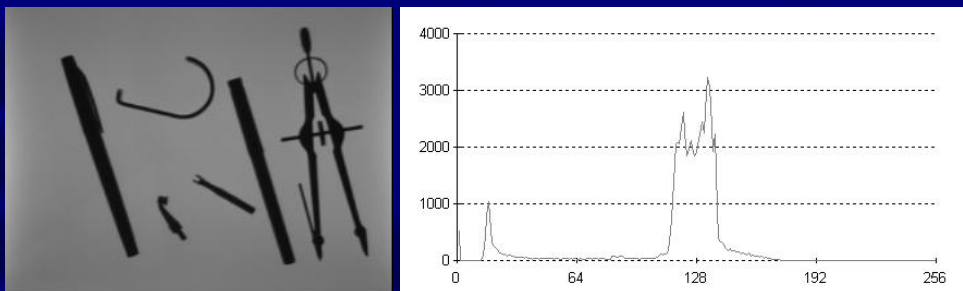
- Podem usar doble threshold:
- $$\begin{aligned} g(i, j) &= X && \text{si } f(i, j) \geq T_H \\ g(i, j) &= Y && \text{si } f(i, j) < T_L \\ g(i, j) &= Z && \text{si } T_H > f(i, j) \geq T_L \end{aligned}$$

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 .

Selecció del llindar

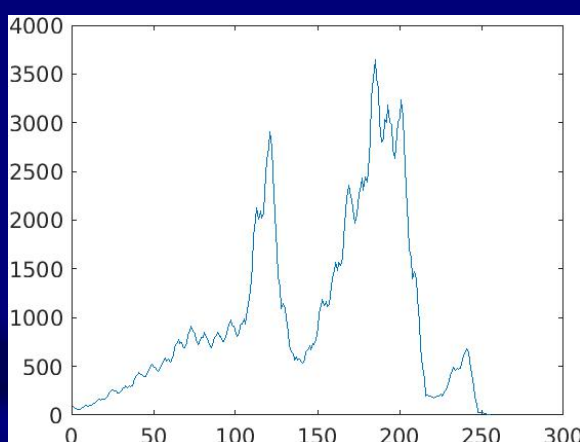
- P-tile-thresholding: llindar que deixa p% píxels per sobre.

- A partir de l'histograma bimodal:



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

Selecció del llindar



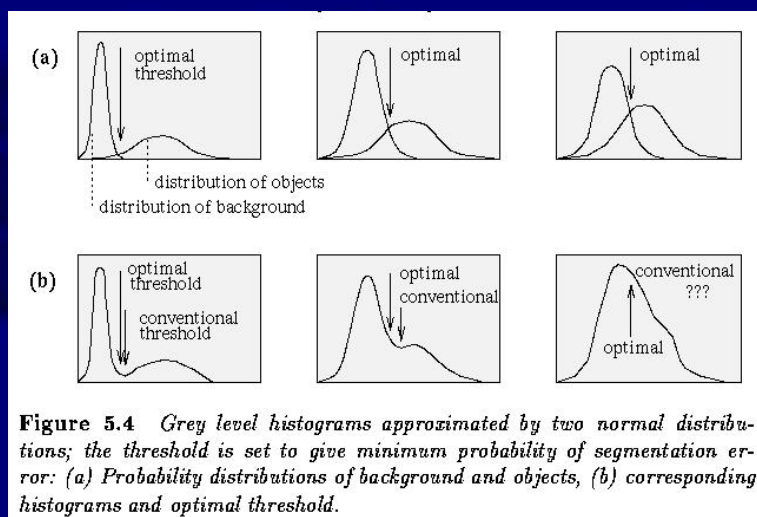
Bimodal?

Trimodal?

Volen dir algo els modes?

Selecció del llindar

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

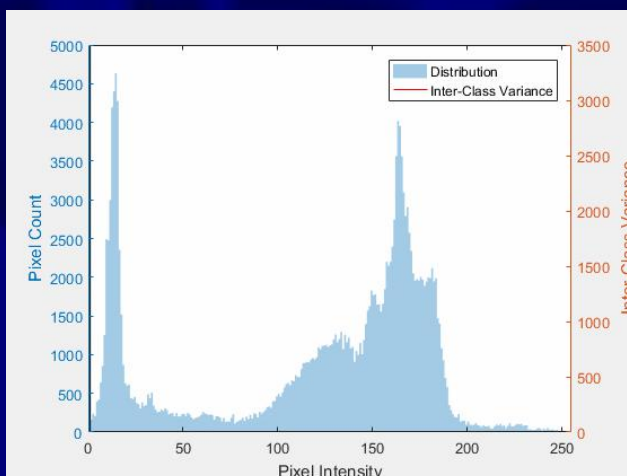


Otsu thresholding

Objectiu: Trobar el llindar que maximitza la variància inter-classe

Com???? Probant-los tots

(Maximitzar la variància inter-classe és el mateix que minimitzar la variància intra-classe)



Otsu thresholding

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



(a)



(b)



(c)



(d)

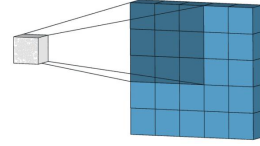


Moving averages

Local Thresholding

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

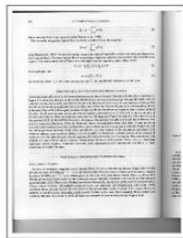
- mean μ in a region of e.g. 15 x 15 pixels
- standard deviation σ in the same region
- set local threshold to $t = \mu - 0.2\sigma$



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

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

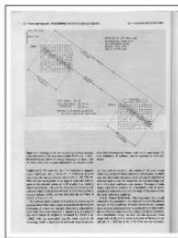
$$k = 0.5$$
$$r = 128$$



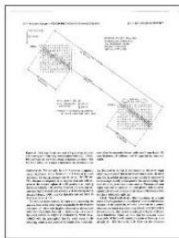
(a) Input image



(b) Sauvola's result



(c) Input image



(d) Sauvola's result

Tècniques diverses per trobar el llindar

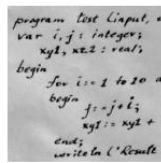
Globals:

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

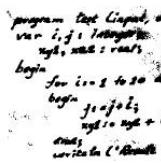
Locals:

- Chow-Kaneko, moving averages...

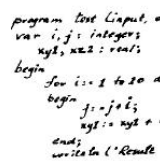
Resultats



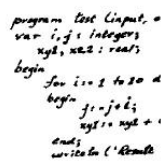
Original



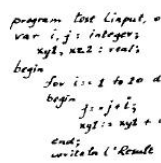
Mean value = 175



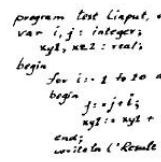
Two peaks = 132



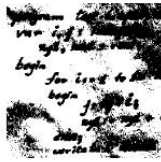
Edge
pixels = 156



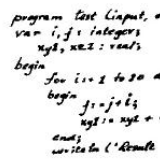
Iterative
Selection = 121



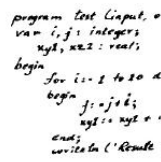
GLH = 122



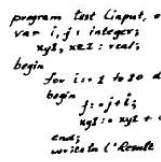
Pun = 184



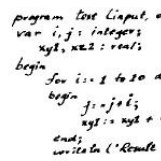
Kapur = 153



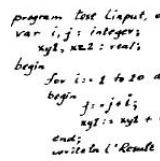
Johannsen = 132



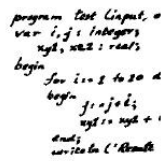
10 percent = 147



Fuzz
(entropy) = 131



Fuzz
(Yager) = 124



Min. Error = 166

Resultats



Original



Mean = 119



Two peaks = 41



Edge
pixels = 46



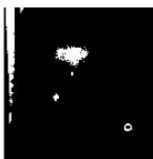
Iterative
Selection = 100



GLH = 103



Pun = 142



Kapur = 185



Johannsen = 176



10 percent = 40



Fuzz
(entropy) = 87

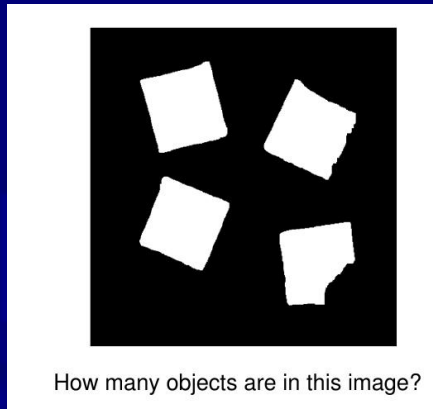


Fuzz
(Yager) = 111



Min. Error = 54

Anàlisi d'imatges binàries



- I quines propietats geomètriques tenen? Àrea, perímetre? Excentricitat?...
- Cal etiquetar els blobs abans de fer aquestes anàlisis

Labelling

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

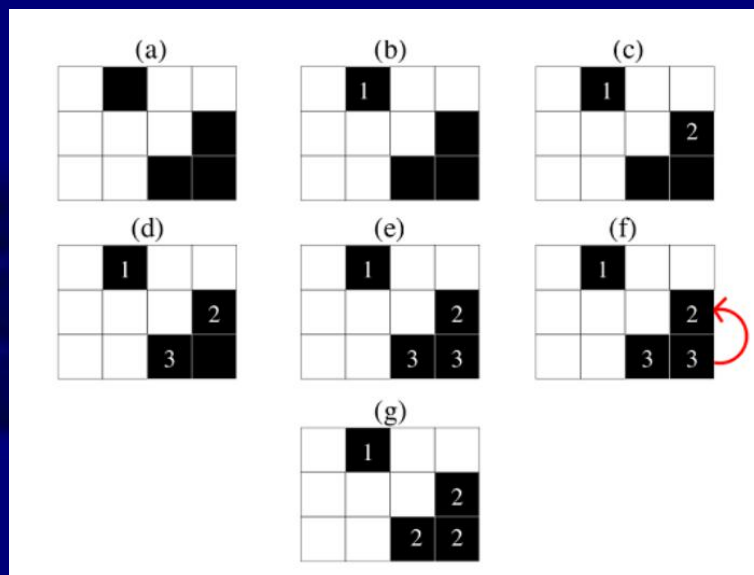


- És un operador de pre-processat. Ens permetrà fer un anàlisi individual de les cel.les posteriorment.

Algorisme de labelling

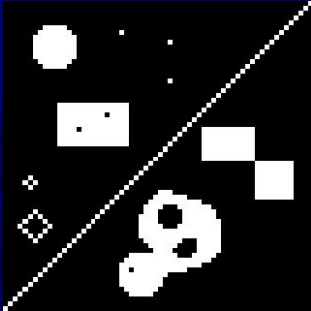
- Scannejar la imatge per files
- Quan es trobi un nou pixel de foreground no etiquetat, assignar-li una etiqueta:
 - a) Si el pixel de l'esquerra era de background, assignar-li una etiqueta nova.
 - b) Si el pixel de l'esquerra era d'algun blob etiquetat, assignar-li la mateixa etiqueta.
- Si el pixel veí de dalt era d'un blob amb diferent etiqueta, fusionar les etiquetes.

Algorisme de labelling

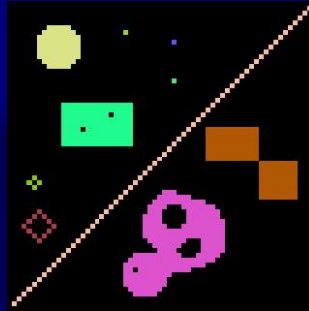


Labelling

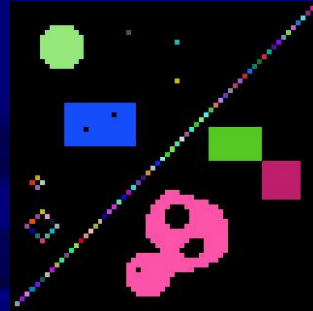
- Connectivitat 4 vs connectivitat 8



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.

