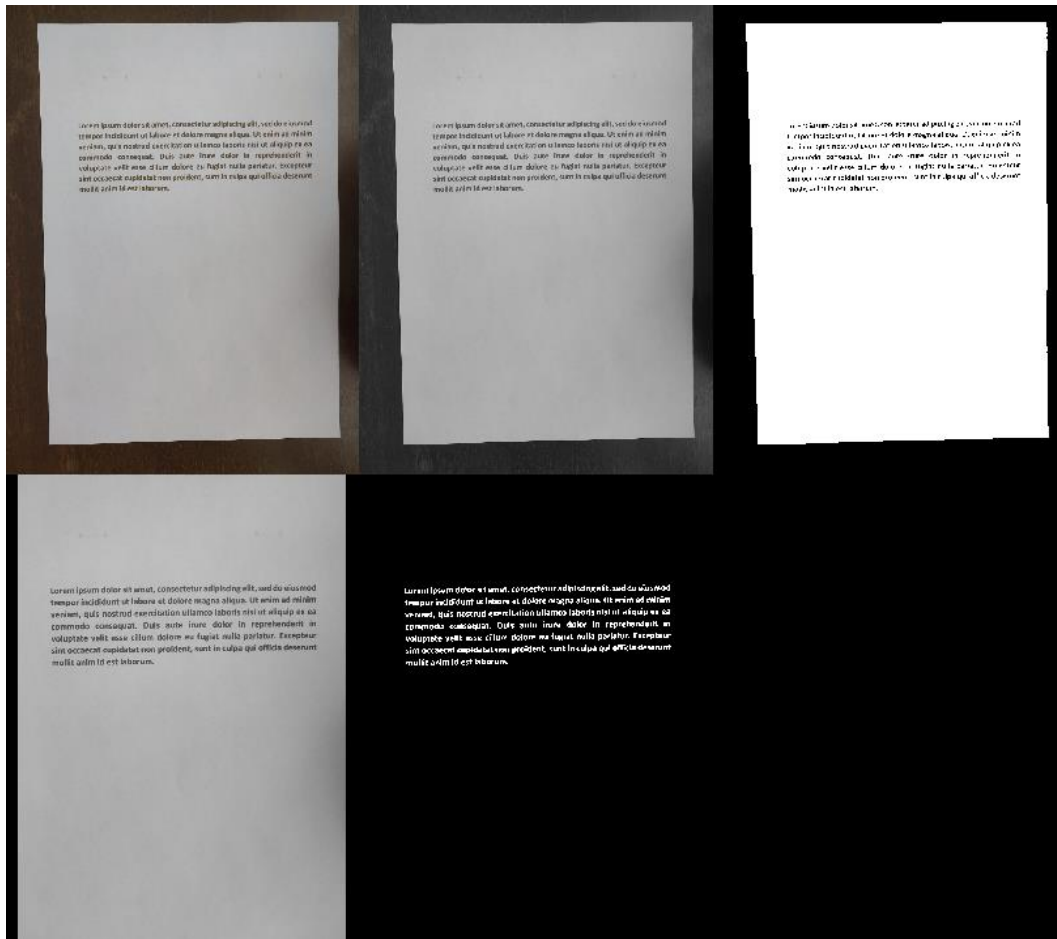


VC: Informe de Laboratori 6

Segmentació



Pere Ginebra Solanellas

3/4/2021 – Q2 Curs 2020-21

Visió per Computador, FIB UPC

1. Introducció

En aquesta sessió de laboratori treballarem la segmentació d'imatges analitzant el text d'una fotografia i aïllant-ne els seus símbols i caràcters.

2. Exercici

Per començar llegim la imatge, la passem a escala de grisos i analitzem el seu histograma:

```
img = imread('LoremIpsum.jpeg');  
imgGray = rgb2gray(img);  
h = imhist(imgGray);  
plot(h);
```

Figura 2.1. Lectura i processat inicial de la imatge

A continuació binaritzem la imatge per distingir el full de paper del fons i utilitzem el resultat per a retallar el fons de la imatge original i quedar-nos solament amb el text:

```
%binarització del full de paper  
BW = imgGray > 95;  
figure  
imshow(BW);  
  
%retallat del full de paper  
imgCrop = imcrop(imgGray, [153 69 977 1405]);  
figure  
imshow(imgCrop);
```

Figura 2.2. Binaritzat general i retallat del full de paper

Fet això utilitzem *colfilt* per aplicar un filtre de mitjana per columnes sobre la imatge amb una finestra de 35x15 (la mida d'en caràcter mitjà i les seves separacions corresponents). Ara podem binaritzar localment la imatge posant a blanc els píxels que són més foscos que la mitjana de la seva finestra multiplicada per un coeficient K:

```

%filtre de columna, colFunc calcula la mitja de la finestra
F = colfilt(imgCrop, [35 15], 'sliding', @colFunc);

%binaritzat local a partir del filtre de F
K = 0.85;
imgDouble = double(imgCrop);
letters = imgDouble < (K * F);

```

Figura 2.3. Filtrat de columna i binaritzat local a partir d'aquest

Ara netegem la imatge resultant de petits punts erronis amb un *imopen* amb un element estructural puntual i analitzem els components connexos de la imatge amb *bwconncomp* per aïllar els símbols del text. *NumObjects* ens indica el nombre de components independents que ha trobat la funció:

```

%neteja de la imatge amb imclose i anàlisis dels components connexos
SE1 = strel('sphere', 1);
letters = imopen(letters, SE1);
figure
imshow(letters);
cons = bwconncomp(letters);
chars = cons.NumObjects

```

Figura 2.4. Neteja de la imatge i anàlisis dels components connexos

Finalment podem representar els resultats intermitjos amb un *montage* i el final amb un *imshow*. Declarant *hold on* podem pintar rectangles al voltant de els components connexos que hem trobat anteriorment utilitzant les funcions *regionprops* i *rectangle* que ens donen les capses i ens les pinten respectivament:

```

%montage dels passos i imatge binària final
montage({img, imgGray, BW, imgCrop, letters})
figure
imshow(imgCrop);
hold on

%encuadrament dels símbols detectats sobre la imatge original
contorns = regionprops(cons, 'BoundingBox');
for i = 1:chars
    rectangle('Position', contorns(i).BoundingBox, 'EdgeColor', 'r', 'LineWidth', 1);
end

```

Figura 2.5. Representació dels resultats intermitjos i el final

3. Resultats

A continuació podem veure el *montage* dels passos intermitjos de la segmentació, d'esquerra a dreta i de dalt a baix son: Original, escala de grisos, binarització general, retallat i resultat final:

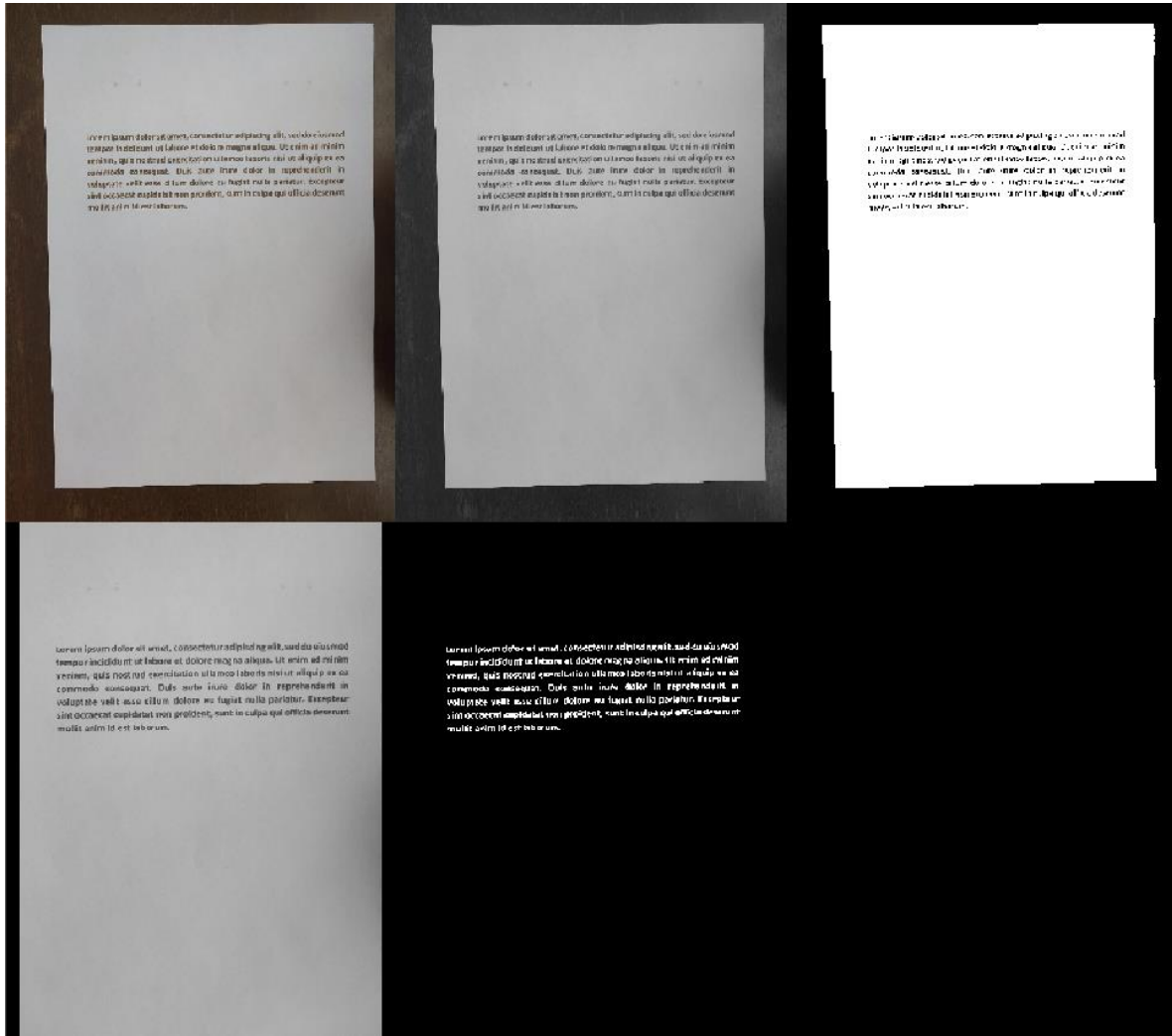


Figura 3.1. Resultats intermitjos de la segmentació

La funció *bwconncomp* ens troba 402 components connexos independents, un valor prou proper als 377 caràcters del text, tenint en compte que aquest conté 42 'i' que sovint es divideixen en dos:



Figura 3.2. Nombre de components connexos obtinguts amb *bwconncomp*

El resultat final marcant les caixes contenidores es el següent:

Dorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
commodo consequat. Duis aute irure dolor in reprehenderit in
voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur
sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt
mollit anim id est laborum.

Figura 3.3. Resultat final

4. Annexos / Script

A continuació tenim el script explicat a l'exercici:

```
clear
close all

img = imread('LoremIpsum.jpeg');
imgGray = rgb2gray(img);
h = imhist(imgGray);
plot(h);

%binarització del full de paper
BW = imgGray > 95;
figure
imshow(BW);

%retallat del full de paper
imgCrop = imcrop(imgGray, [153 69 977 1405]);
figure
imshow(imgCrop);

%filtre de columna, colFunc calcula la mitja de la finestra
F = colfilt(imgCrop, [35 15], 'sliding', @colFunc);

%binaritzat local a partir del filtre de F
K = 0.85;
imgDouble = double(imgCrop);
letters = imgDouble < (K * F);

%neteja de la imatge amb imclose i anàlisi dels components connexos
SE1 = strel('sphere', 1);
letters = imopen(letters, SE1);
figure
imshow(letters);
cons = bwconncomp(letters);
chars = cons.NumObjects

%montage dels passos i imatge binària final
montage({img, imgGray, BW, imgCrop, letters})
figure
imshow(imgCrop);
hold on

%encuadrament dels símbols detectats sobre la imatge original
contorns = regionprops(cons, 'BoundingBox');
for i = 1:chars
    rectangle('Position', contorns(i).BoundingBox, 'EdgeColor', 'r', 'LineWidth',
1);
end
```

Figura 4.1. Script utilitzat durant la sessió

I la funció colFunc utilitzada per calcular la mitjana de la finestra del *colfilt*:

```
function [y] = colFunc(x)
    [f c] = size(x);
    y = sum(x)/f;
end
```

Figura 4.2. Funció utilitzada per calcular la mitjana d'una finestra

5. Bibliografia / Documentació

Per dibuixar les caixes contenidores per sobre de la imatge original:

- <https://es.mathworks.com/matlabcentral/answers/110230-drawing-a-rectangle-on-top-of-an-image>
- <https://es.mathworks.com/matlabcentral/answers/158151-how-to-draw-boundingbox-rectangle-transparent-over-an-image>