Dpto. Ingeniería de Sistemas y Automática
Universidad de Málaga

COMPUTER VISION

EXERCISE 4a: Image segmentation with region growing

Concepts: Region growing, seed

The code shown below extracts a region of the image by a growing process. This process starts with a seed in the (*x*,*y*) coordinates and adds pixels to the region recursively. Implement a code to:

- Load the *rice.tif* image and initializes the global variables **region**, **media**, **points_in_region** and **im1**.
- Permits the user to indicate a seed and initializes, using that seed, the growing process. You can use the **ginput** command for that. *Note: when testing the code, choose the seed in such a way that it falls into a rice grain.*

```
imshow(im1);
[x,y]=ginput(1);
x=uint8(x);
y=uint8(y);
region = zeros(size(im1));
media=double(im1(y,x));
points_in_region=0;
```

- Execute the region growing function (**crec_recur**), and draw the extracted region in the initial image as it is shown in the result image.

```
crec_recur(x,y,30);
im=im1;
mask = logical(region);

im_modulo_r = uint8(im);
im_modulo_g = uint8(im);
im_modulo_b = uint8(im);

im_modulo_r(mask) = 0;
im_modulo_g(mask) = 255;
im_modulo_b(mask) = 0;

im_modulo_rgb(:,:,1) = im_modulo_r;
im_modulo_rgb(:,:,2) = im_modulo_g;
im_modulo_rgb(:,:,3) = im_modulo_b;

imshow(im_modulo_rgb);
```

Code

```
% Esta función realiza de manera recursiva el crecimiento de una región de la imagen
% partiendo de una semilla en (x,y). El crecimiento se hace sobre sus ocho-vecinos.
%
% J. Gonzalez
function crec_recur(x,y,toler)

% Variables globales para no pasar tantos argumentos a la funcion
global region;           % Region segmentada. Es una matriz del tamaño la imagen,
con
                         %      todo a cero salvo la region, a uno.
global media;            % Media (dinámica) de la region que va creciendo (DOUBLE!).
global points_in_region; % Número de puntos actual de la region segmentada
global im1;              % Imagen de entrada en escala de grises

% Comprobacion que no estamos fuera de la imagen
if (x <= 0 | x > size(im1,2) | y <= 0 | y > size(im1,1) )
    return;
end
if (region(y,x) == 1) % si esta ya marcado (=1) no hacemos nada y salimos de la
funcion
    return;
end

if abs(double(im1(y,x)) - media) < toler % Condicion de añadir pixel
    region(y,x) = 1;
    points_in_region = points_in_region + 1;
    media = (media * points_in_region + double(im1(y,x)))/(points_in_region+1);
%calculo dinámico de la media

    % Recursión sobre los 8-vecinos
    crec_recur(x-1,y+1,toler);
    crec_recur(x-1,y,toler);
    crec_recur(x-1,y-1,toler);
    crec_recur(x,y+1,toler);
    crec_recur(x,y-1,toler);
    crec_recur(x+1,y+1,toler);
    crec_recur(x+1,y,toler);
    crec_recur(x+1,y-1,toler);
end

return;
```

Results

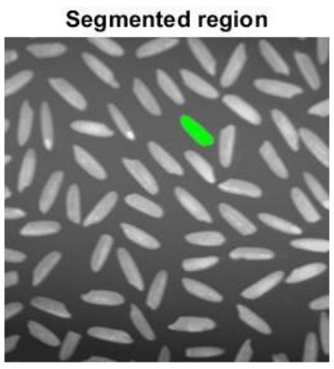


Image 1

As we see in the image, the green part is a region of one of the rice grains. This region is obtained from one segmentation method: the region growing. This method uses an algorithm which see the neighbors and add to the region for each neighbor if the threshold is higher than each one.

EXERCISE 4b: Image segmentation using K-means

Concepts: K-means

The K-means algorithm is used for clustering, since it permits us to group data (in our case, pixels) relying on a certain criteria, for example, a similar intensity level. In the code example below it is illustrated how to use the Matlab function **kmeans** employing as features the image pixel intensities for the segmentation of the image *torre_monica.jpg* (in grayscale) into two clusters.

- Execute the code. What's happening?

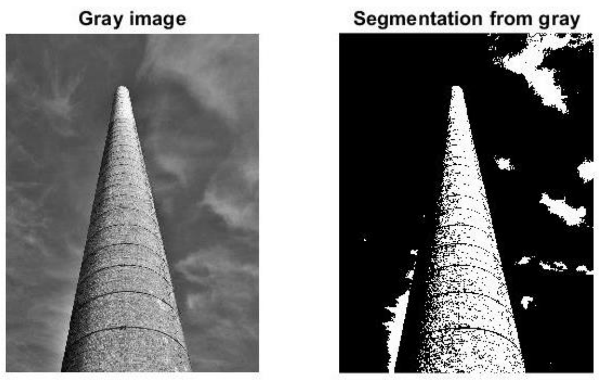


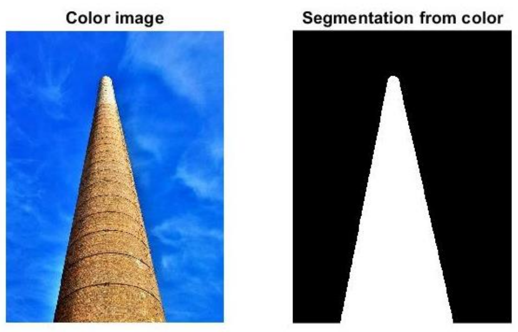
Image 2

The program is trying to segment the image into two parts from the intensity.

- Change the number of iterations of the *for* loop until the **kmeans** algorithm reaches convergence. How many iterations are needed?

Until the 9th iteration, the kmean algorithm does not convergence.

- Now, modify the code to segment objects in the RGB 3D space, in other words, the feature of each pixel is its RGB color. *Pay special attention to the reshape function; get help with **help reshape** in Matlab.*
- Compare the resultant segmentation with the one obtained for the same image but in grayscale.



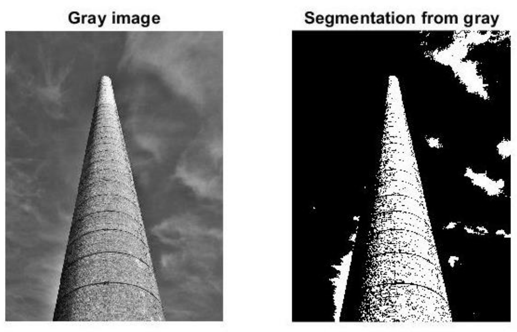


Image 3 *Image 4*

In both images we can see a huge difference in their results. From the colour one, the result is segmented by the colour, so we can appreciate which part is the object and which the sky. From the other, the result is segmented by the intensity.

Note: *for more information about* ***kmeans*** *you can check the Matlab help and the lectures' material.*

Example:

```
clear variables
close all
clc

im = imread('torre_monica.jpg'); % Load the image
im_gray = rgb2gray(im); % Convert it to grayscale
figure, subplot(1,2,1), imshow(im_gray), title('Initial image') % Show it
nPixels = prod(size(im_gray)); % Get the number of pixels to ...
data = reshape(im_gray, nPixels, 1); % ... reshape the image as a vector
seed1 = [10,10]; % Set the seeds
seed2 = [400,240];
seeds = [seed1(1)*size(im_gray,2)+seed1(2);seed2(1)*size(im_gray,2)+seed2(2)];
% seeds = repmat(seeds,[1,3]); % Uncomment for the RGB image

for i=1:10
    fprintf('Performing k-means segmentation with %d maximum iterations\n',i);
    pert = kmeans(double(data), 2,'Start',seeds,'MaxIter',i); % kmeans!
    clus = reshape(pert, size(im_gray)); %Vector Image back to a matrix
    im_clust = uint8(255*(clus-1)/(max(max(clus))-1)); % Clusters are 1 and 2
    subplot(1,2,2), imshow(im_clust), title('Segmented image') % Show results
    pause;
end
```

EXERCISE 4c: Image segmentation with EM (Expectation Maximization)

Concepts: EM

The EM algorithm assigns to each region or cluster a Gaussian probability distribution and enables us to group data (in our case, pixels) by maximizing the probability of the data (pixels) belonging to each region.

- Segment the *torre_monica.jpg* into two regions using **segmentation_em**, available as material for the course, using a different number of algorithm iterations. Concretely try *numIter=2*, *numIter =5* and *numIter =10*.

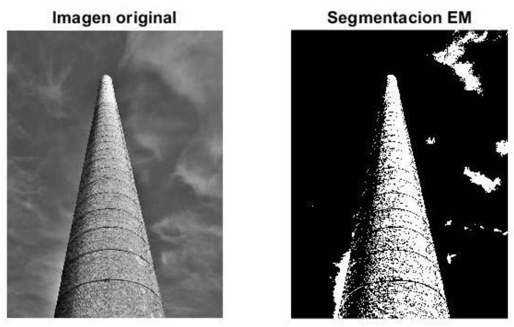


Image 5. EM segmented image with numIter=2

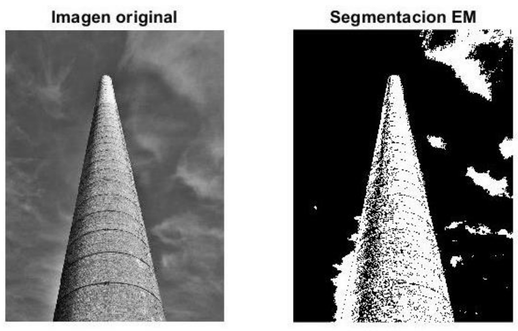


Image 6. EM segmented image with numIter=5

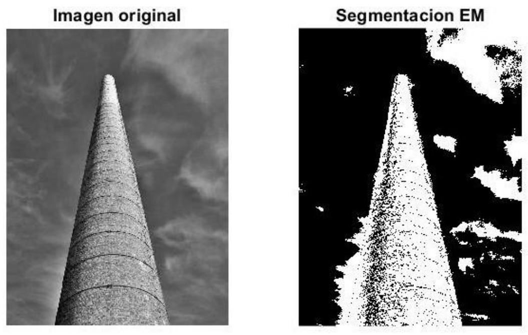
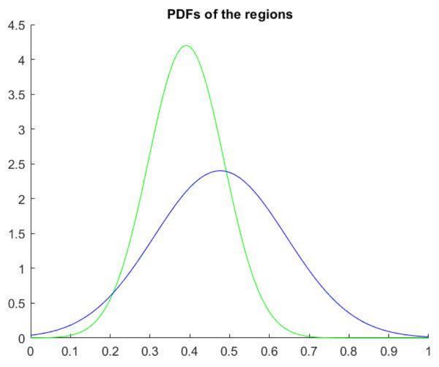
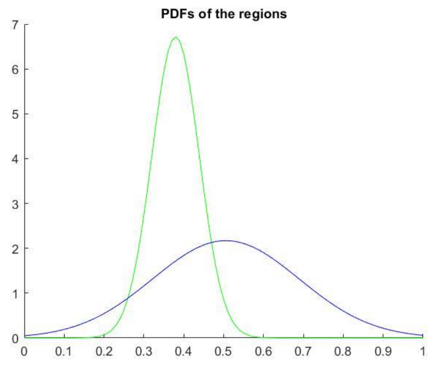


Image 7. EM segmented image with numIter=10

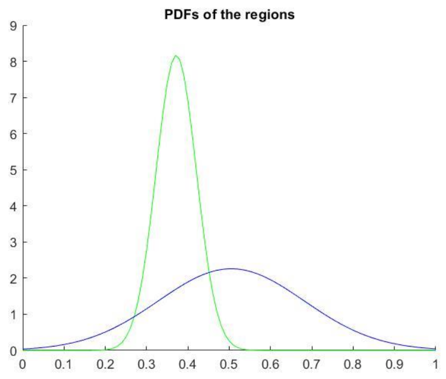
As we see in the tree images, the EM algorithm segments in region from the intensity. Furthermore, when it has more number the iteration, more sections are segmented.

- Interpret the probability density functions of the obtained regions for the different executions. *Note: these densities are automatically plotted by the* ***segmentation_em*** *function.*





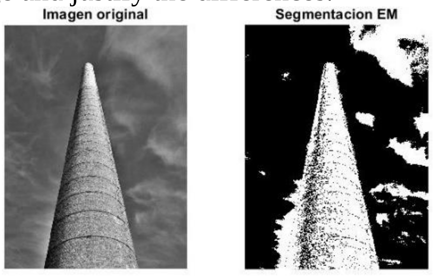
Plot 1. PDFs of EM segmented image with numIter=2 *Plot 2. PDFs of EM segmented image with numIter=5*



Plot 3. PDFs of EM segmented image with numIter=10

For all the plots, the green pdf is from the black segmented part and the blue one, the black segmented part. Also, it indicates the probability that have that segmentation for been in that segmentation. Furthermore, as we see, when it has a greater number of iterations, the variance in each pdf change. In this chase, the change is the green pdf have less and less variance and the blue pdf have more and more variance.

- Compare the results with the output of the K-means algorithm working with the grayscale image and justify the differences.



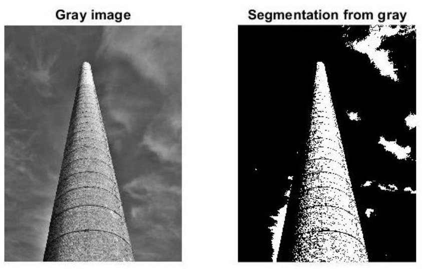


Imagen 8 *Imagen 9*

The difference in both images is that the EM segmentation segmented more than the kmean segmentation. This is due to the EM algorithm which is like the kmean algorithm, but it more generic. Both changes their means in each iteration, but in EM have a extra parameter, the covariance, which also change in each iteration.

Example:

```
clear variables
clc

% Prepare the data
num_seg = 2;
im = imread('torre_monica.jpg');
im = rgb2gray(im);
im = im2double(im);

numIter= 2;

clus = segmentation_em(im, num_seg, numIter); %Image with segm. result
im_clust = uint8(255*(clus-1)/(max(max(clus))-1));

% Show the figure
figure
subplot(1,2,1);
imshow(im);
title('Imagen original');

% Show the results
subplot(1,2,2);
imshow(im_clust);
title('Segmentacion EM');
```