Scene Segmentation and Interpretation
Coursework 1. Image Segmentation (Region Growing)

Flávia Dias Casagrande and Dragutin Vujovic

March 4, 2015

1 Objective

This report presents all the development executed for the Region Growing algorithm implementation in order to segment a given image. The algorithm works for both grey level and color images and its results are compared with the results of the Fuzzy C-Means (FCM) clustering algorithm, which is provided in Matlab.

2 Introduction and problem definition

Image segmentation consists in partitioning the image in different regions that have similar characteristics (which can be for example intensity, color, texture, etc.) and that are meaningful in respect to a particular application. It provides a good tool for locating objects and boundaries in an image.

Each of the regions represents a label, which is a value assigned for the pixels. Therefore, pixels with the same label belong to the same region.

There are many groups of methods to segment an image, and each has several methods.

For instance, the region-based methods, which groups together pixels which are neighbours and have similar values, and split the pixels with different values. Region Growing and Split and Merge method are examples, and the former is one of the implemented algorithms in this coursework.

Clustering-based methods groups pixels considering one or more features and requires a guess for the number of clusters and an initial position for each of them. The other pixels are assigned to the cluster they belong. Examples of this group are image Thresholding, Peak and Valley method, K-Means and Fuzzy C-Means. The latter is the second method implemented by the group.

3 Algorithm analysis

The Region Growing algorithm consists in choosing a so-called seed point. Starting from it, for each pixel in the image, the neighbours are analysed and grouped to the region of the pixel under study if they satisfy a certain criteria. The difference between the mean or the standard deviation for example of the region should be lower than a given threshold, so that the pixel belongs to the region.

In order to avoid the Region Growing to go trough the image line by line, it is common to construct a queue, which allows the regions to grow in any desired direction, which could be for example analysing the A4- and A8-connectivity.

The Region Growing algorithm can be written mostly in three ways: recursive, sequential (analyses pixel by pixel) and concurrent (several seed points, growing simultaneously).

In this coursework the algorithm was implemented in the sequential way, by using the queue according to either A4- or A8-connectivity.

4 Design and implementation of the proposed solution

The algorithms for the Region Growing and the Fuzzy C-Means methods were implemented in Matlab.

The Region Growing algorithm has a main function called *growingRegion* and three other functions, *findNeighbours*, *getDifference* and *getMean*.

growingRegion: It receives three parameters, the image to be segmented, the connectivity (4- or 8-connectivity) and the threshold to take in account when assigning regions. It returns the image colored according to the assigned labels to each pixel ('regions'), the number of regions needed to provide the required segmentation ('regionNumber') and the time to execute the algorithm ('executionTime').

findNeighbours: It receives as parameters the pixel whose neighbourhood will be returned in a vector ('currentPixel'), the connectivity and the size of the image in segmentation process. It returns a vector with the indexes (row and column) for each neighbour.

getDifference: It receives the current pixel and the previous calculated mean of the intensity. The difference between the value of the pixel and the previous mean in the current region should be inside the threshold defined, so that the pixel will also be part of this region, otherwise the region label will be increased in the main function. In order to calculate the difference, the function checks the number of elements in the variable pixel received. If it is one, it is a grey level image, then the difference is simply the intensity of the current pixel and the mean. In the case of a color image, the difference is the Euclidean distance, which is defined by Equation 1.

$$EuclideanDistance = \sqrt{(Pixel_{Red} - \mu_{Red})^2 + (Pixel_{Green} - \mu_{Green})^2 + (Pixel_{Blue} - \mu_{Blue})^2} \quad (1)$$

getMean: It receives the current pixel, the previous mean and a counter, which corresponds to the number of pixels already in a certain region. It calculates then the mean of the region (for each color plane in the case of a color image - red, green and blue - and for the intensity for the grey level image), by considering that each of the previous pixels were valued as the mean value and the current one, divided by the total number of pixels. It returns the a vector with the mean(s) intensity.

The algorithm is described in the following pseudo-code.

Algorithm 1 Region Growing Algorithm

```
procedure REGION GROWING
    dim ← size of image
    visitedPixels ← new matrix, same size as image, to assign the labels
    regionNumber ← 0
    for every pixel in the image do
        if current pixel == 0 then
            regionNumber ← regionNumber + 1.
            meanIntensity ← value of pixel.
            queue ← initialize with indexes of current pixel.
            visitedPixels(i,j) ← currentregionNumber.
            while queue ≠ 0 do
                findNeigbours
                Remove seed from queue
                if getDifference < thresholdandvisitedPixels == 0 then
                    Add neighbours to queue
                    Assign label to pixel
                    getMean
```

The Fuzzy C-Means algorithm uses the function *fcm* of Matlab. Its receives as parameters the image and an initial guess for the number of clusters. It returns a matrix with the center points of each cluster ('center'), a matrix with the percentage of the pixels being part of each cluster (U) and a matrix with the values of the best location for the clusters (OBJ_FCN). Basically, each point in the image belongs to each cluster with a certain degree, and all clusters are characterized by their center of gravity.

Therefore, from the returned variables, the matrix U is the one that was used to generate the labelled output matrix. The detailed pseudo-code is presented below.

In order to make easier the choice of parameters for the segmentation calculation for both methods (connectivity, threshold, number of clusters, etc.) and to provide a better comparison and interaction with the user, an interface was also developed. That said, in order to run the algorithms for the desired images the file *gui.m* should be executed.

Algorithm 2 Fuzzy C-Means Algorithm

procedure FUZZY C-MEANS
  $U \leftarrow$ fcm(image,numberOfClusters)
  $visitedData \leftarrow$ new matrix, same size as image, to assign the labels
  $numberColumns \leftarrow$ U
  for $i :=$ U **do**
    $maxIndex \leftarrow$ biggest percentage in current column of U)
    $visitedData(i) \leftarrow maxIndex$

Besides that, there is a pre-processing option of smoothing the image, with a certain size for the mask applied. The smoothing is done with the function *fspecial* to generate the mask and *imfilter* to convolve the image with it, both from Matlab, by using a Gaussian filter.

5 Experiments

Figures 1 and 2 show the segmentation of the gantry color image provided. With two labels it is already possible to perform a satisfying segmentation in this image, as done by the Fuzzy C-Means method. By performing Region Growing with a low threshold of 30 (Figure 1), the number of regions is 799 and with a threshold of 100 (Figure 2) it reduces to 83. The time for the segmentation by using Region Growing is around 4s, whilst for the FCM it is much lower, around 0.83s.

Therefore, the first characteristic to notice is that the execution time for the FCM algorithm is much faster than Region Growing, since the number of clusters is much smaller and the center points for each cluster are already defined.

Regarding the Region Growing algorithm, it is notable that as the threshold value increases, the number of regions is lower. Although for a image with many different characteristics, a high threshold will eliminate many details and structures.

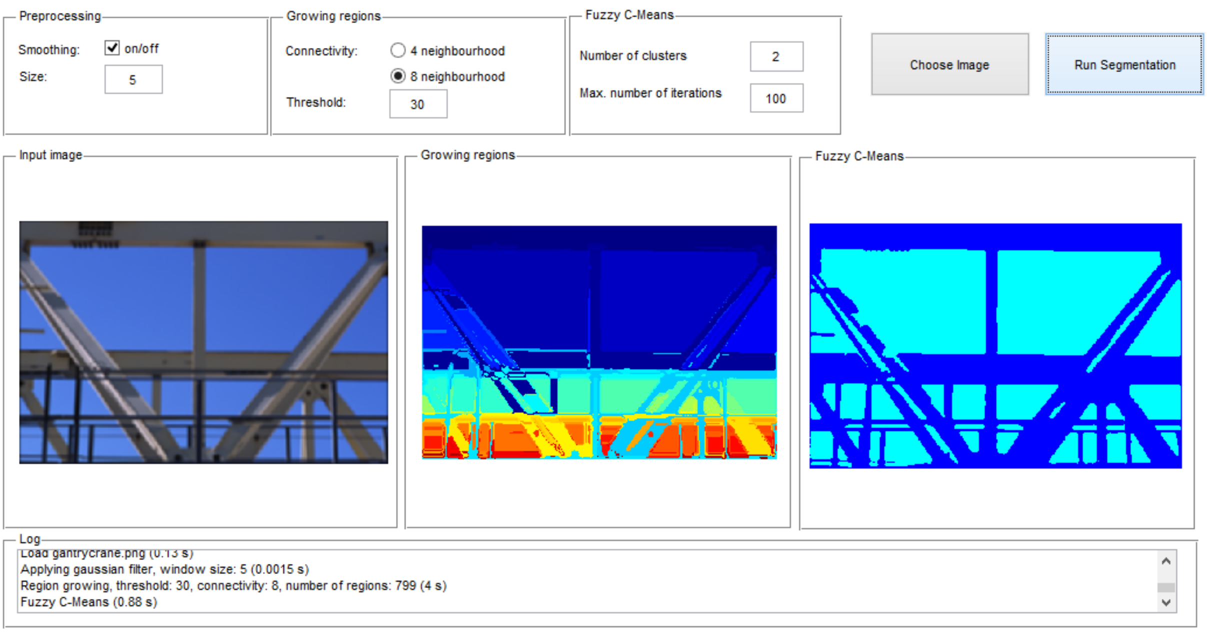

Figure 1: Gantry image segmentation - threshold of 30.

Figures 3 and 4 illustrate the segmentation of the coins grey level image. The experiment performed here is the difference between A4- and A8-connectivity defined for the Region Growing method.

Both connectivities provided the same number of regions for the segmentation, and obviously, the A8-connectivity takes a bit longer execution time.

The difference in the segmented images here is in one of the coins, that is a bit darker in its center. The A8-connectivity (Figure 4) was more precise than the A4-connectivity (Figure 3). Taking in account

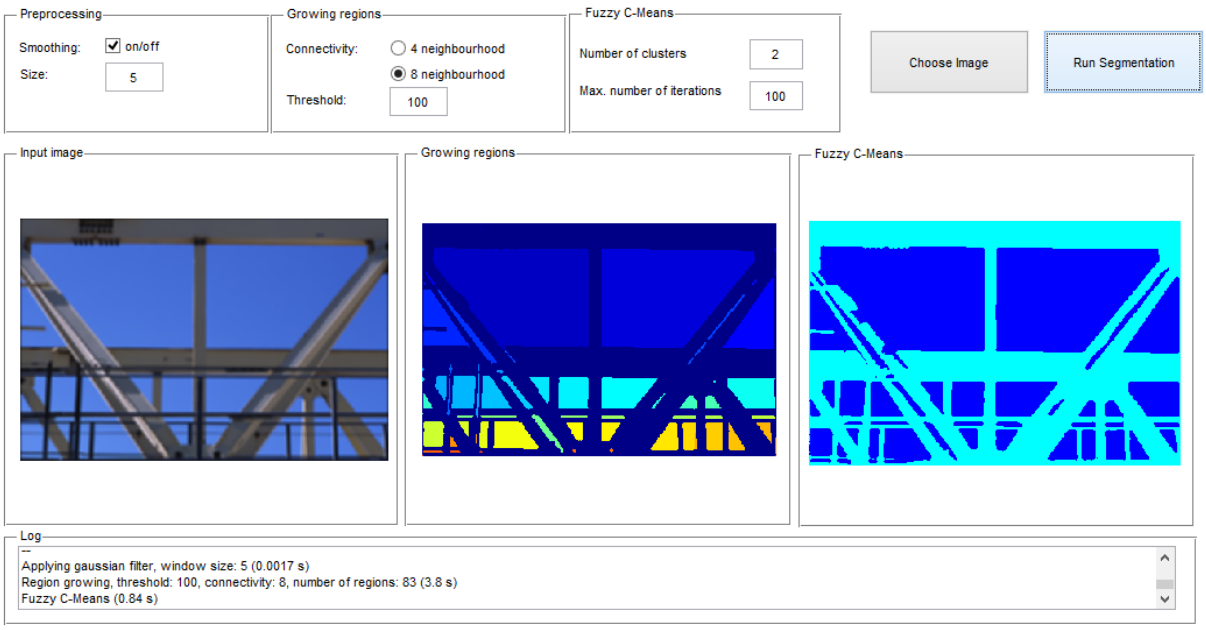

Figure 2: Gantry image segmentation - threshold of 100.

a just a bit lower threshold (Figure 5), the segmentation using A8-connectivity has a similar result to the A4-connectivity, however with one more region.

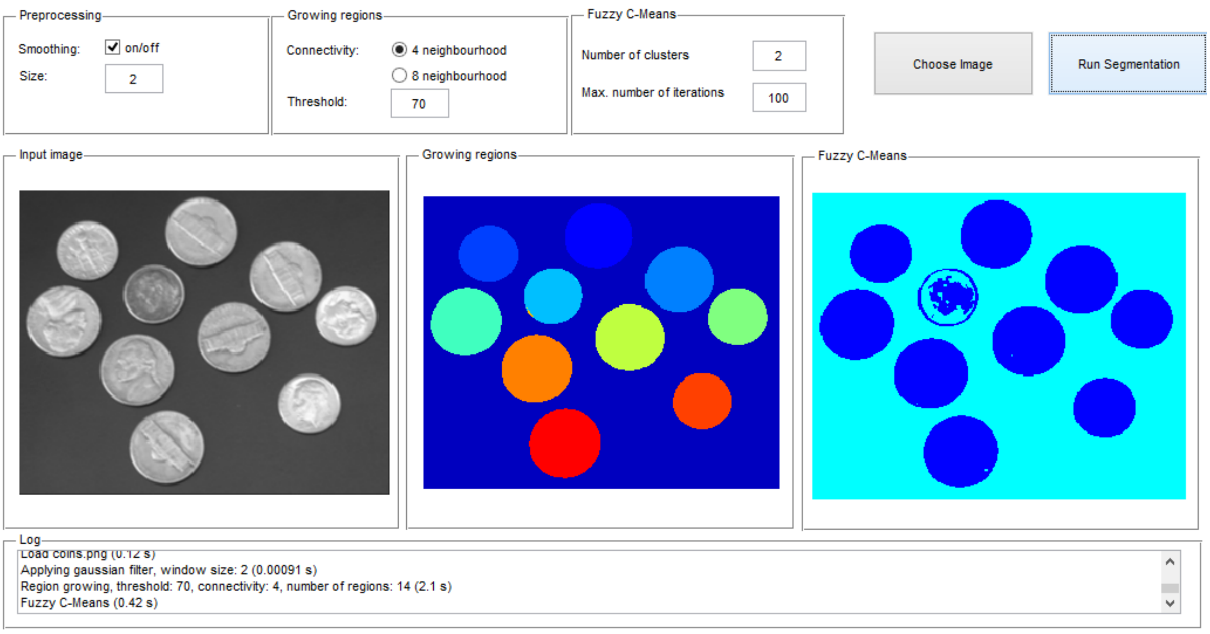

Figure 3: Coins image segmentation - Threshold for Growing Region:70 and A4-connectivity.

The last experiment took in account the influence of the smoothing before performing the segmentation.

Figure 6 and 7 show the segmentation of the woman color image. Without the smoothing of the original image (Figure 6), the Region Growing method resulted in 661 regions, while by applying the smoothing (Figure 7) the number of regions is reduced to 436.

Visually, the segmentation is much better when applying the smoothing, since the noise is reduced and almost not noticed in the segmented image.

Notably, the execution time for the FCM method is normally much faster than the Region Growing, although there is the need of an initial guess for the number of clusters.

A more detailed comparison between the methods applied will be done in next section.

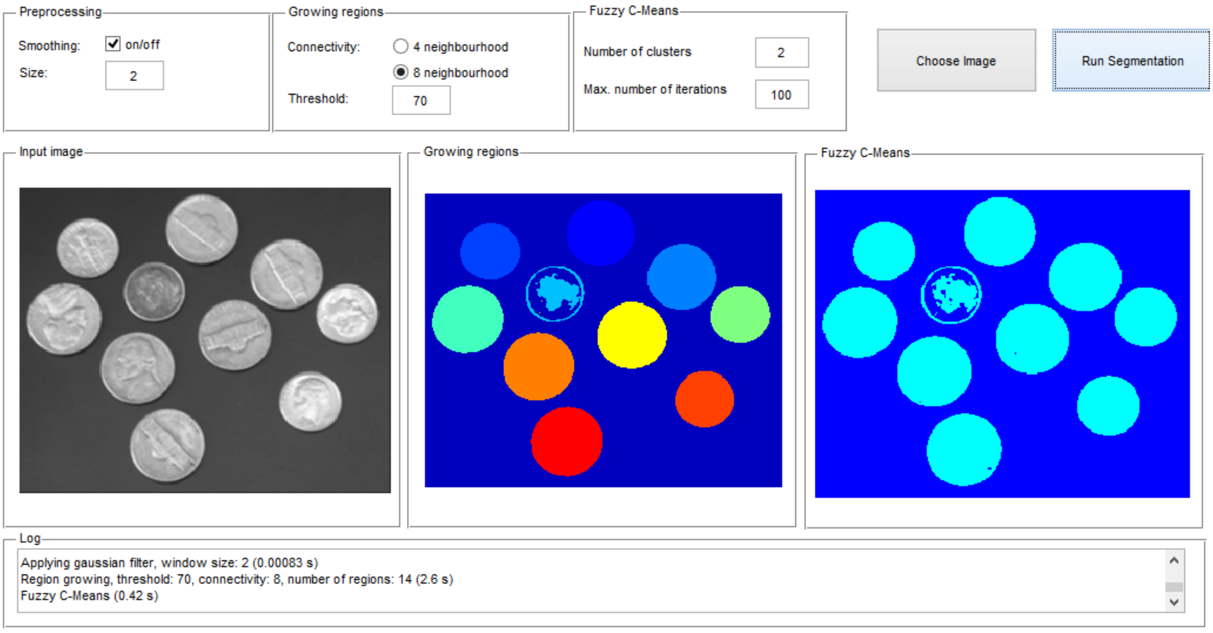

Figure 4: Coins image segmentation - Threshold for Growing Region:70 and A8-connectivity.

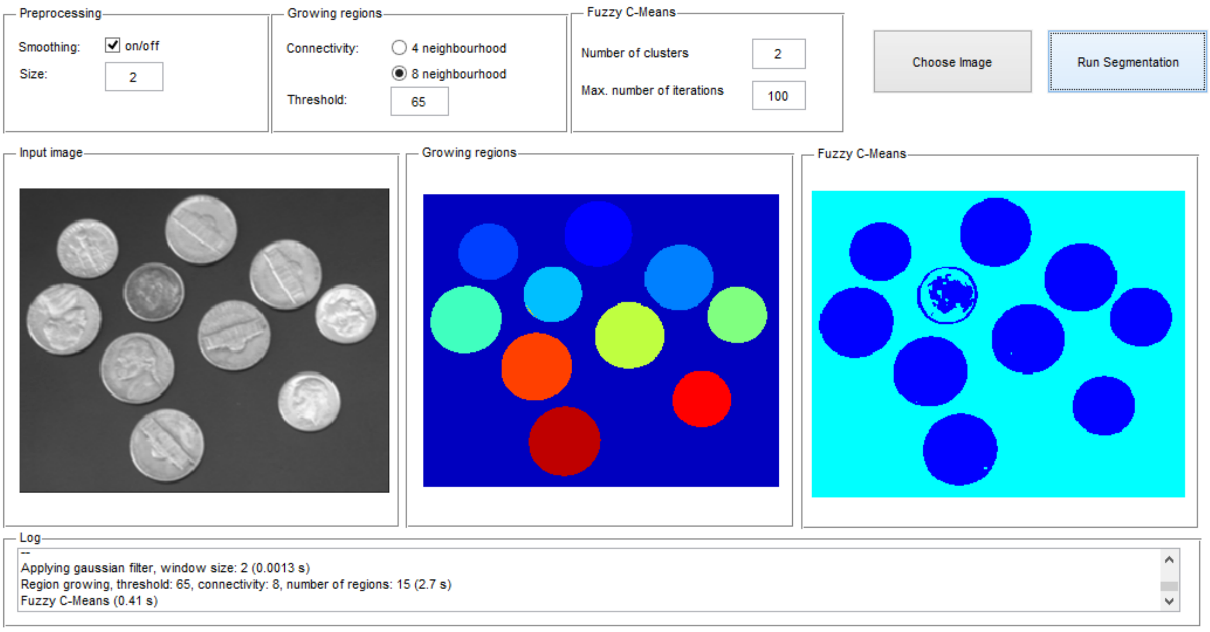

Figure 5: Coins image segmentation - Threshold for Growing Region:65 and A8-connectivity.

6 Comparison of Region Growing with Fuzzy C-Means

The Region Growing algorithm is based on continuity. As previously mentioned, it starts with a seed pixel and adds neighbouring pixels based on similarity to the same region. The selection of the seed point is important as it influences performances and segmentation, besides the fact that it leads to different results. For example, before starting the algorithm, it is possible to find edges and select nearby points as seed points.

Compared to the Fuzzy C-Means algorithm, it is more time consuming and provides a result with a much bigger number of regions, which depending of the application can be a good characteristics or not.

The Fuzzy C-Means has a better performance, robustness and gives best results when the data set is well defined, as is the case of the coins image.

The drawback of this method is that it requires a prior knowledge of the number of clusters. Besides that, randomly choosing initial cluster centres leads to different segmentation results for the same initial

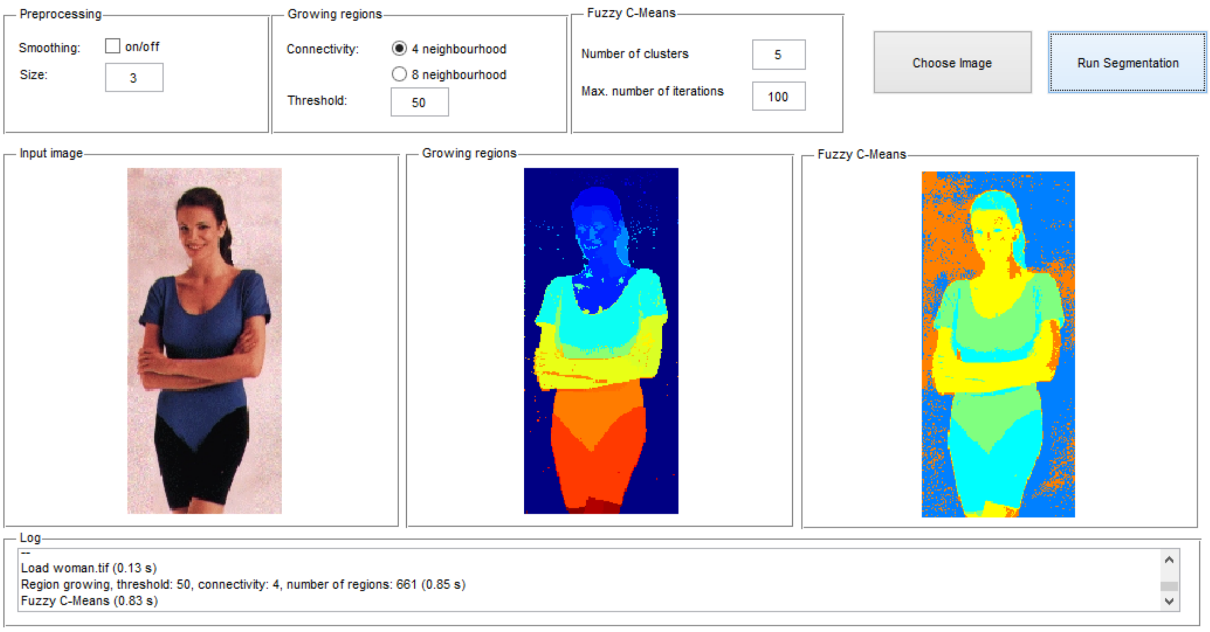

Figure 6: Woman image segmentation - Smoothing option off.

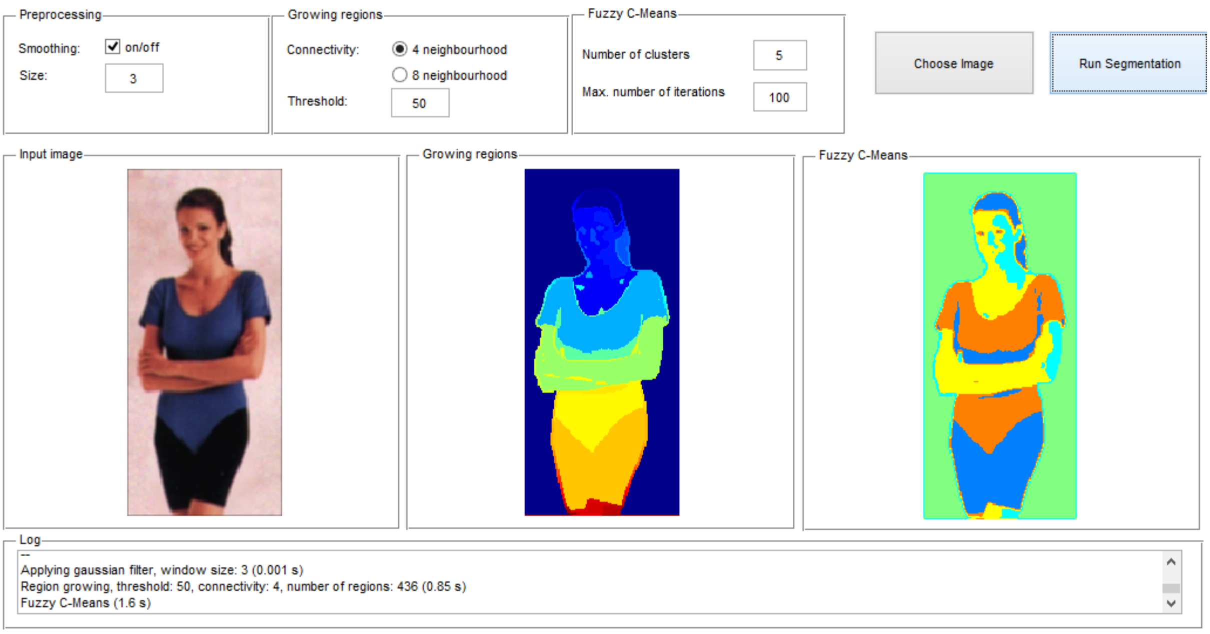

Figure 7: Woman image segmentation - Smoothing option on.

configuration.

A great difference between both methods is that Region Growing requires spatial information while it is not important for FCM, since the clusters and their centres are defined since the beginning.

7 Organization and development of the coursework

The tasks for executing the coursework were quite simple and sequential. First the development of the Region Growing algorithm (two and half hours) and then the Fuzzy C-Means algorithm (one hour). When the algorithms were working correctly, the interface was developed (one hour). The last task was the report writing (three hours).

It is observed then that the real dedication for this coursework was of seven and a half hours. The estimated time dedication for this coursework is four hours (two lab sessions). It is understandable though the additional hours required. Manipulating color images was one of the tasks that took more time and the group was also not so familiar with interface implementation in Matlab.

8 Conclusions

The results of the coursework were successful and achieved the expectations of the group. The segmentation for the images for both methods were done correctly and in an acceptable execution time.

Implementing such methods made much easier their understanding and the comparison between region-based and clustering methods.