Scene Segmentation and Interpretation
Lab 1 Report: Region Growing

Mohammad Rami Koujan
M.Sc. VIBOT
University of Girona

Vishnu Veilu Muthu
M.Sc. Computer Vision
University of Girona

March 3, 2016

1 Introduction and problem definition

Image segmentation is the division of an image into regions or categories which correspond to different objects or parts of objects. Every pixel in an image is allocated to one of a number of these categories. A good segmentation is typically one in which:

- Pixels in the same category have similar characteristics and form a connected region.
- Neighboring pixels which are in different categories have dissimilar characteristics.

Segmentation of nontrivial images is one of the most difficult tasks in image processing. Segmentation accuracy determines the eventual success or failure of computerized analysis procedure. For this reason, considerable care should be taken to improve the probability of rugged segmentation. Two main types of segmentation according to the method followed:

- Region based.
- Clustering based.

The main focus of this lab assignment is on region growing algorithm which is of the first type. So, the objective during this lab session is to design, analyse and implement the Region Growing algorithm in Matlab. Furthermore, a comparison should be drawn with another clustering based algorithm (FCM).

2 Algorithm analysis

As its name implies, region growing is a procedure that groups pixels or subregions into larger regions based on predefined criteria. The basic approach is to start with a set of seed points and from these grow regions by appending to each seed those neighboring pixels that have properties similar to the seed (such as specific ranges of gray level or color).
Selecting a set of one or more starting points often can be based on the nature of the problem. When a priori information is not available, the procedure is to compute at every pixel the same set of properties that ultimately will be used to assign pixels to regions during the growing process. If the result of these computations shows clusters of values, the pixels whose properties place them near the centroid of these clusters can be used as seeds. A critical point for this algorithm is the aggregation criteria. Criteria such as gray level, texture, and color, are local in nature and do not take into account the "history" of region growth. Additional criteria that increase the power of a region-growing algorithm utilize the concept of size, likeness between a candidate pixel and the pixels grown so far (such as a comparison of the gray level of a candidate and the average gray level of the grown region), and the shape of the region being grown.

3 Design and implementation of the proposed solution

As the guidelines of this lab assignment suggest, recursive implementation is going to be a problem due to the maximum number of recursive calls allowed. Therefore, sequential implementation is preferable method of implementation. The following assumptions are made:

- Since no a priori information about the images will be considered available, unsupervised segmentation, the procedure is to iterate through every pixel in order to try to build a new region, if this pixel is still not included with any of the previously constructed regions.

- The aggregation criteria is based on a comparison between the gray level, or the Euclidean distance for color images, of a candidate and the average gray, or colors, level of the grown region. This comparison is drawn with a tolerance variable delta.

- The region under construction will grow in all directions at each level of the aggregation procedure, which would make the results visually more appropriate.

The idea of the proposed implementation is to start from a random pixel, the first one in the provided code, by giving it a label and then obtain the neighbors of this pixel, 4-neighbors connectivity is considered here. Those neighbors will be checked by a separate function, called IsCheckedBefore, if they have any other label or being processed and then inserted to a queue, the aim of which is to organize the procedure in terms of controlling the shape of the region being grown, which is done for example by inserting neighbors clockwise, growing more effectively level by level around the seed, and easing the implementation. Then, each pixel inside the queue will be retrieved and tested to determine if it satisfies the aggregation condition or not. If yes, it will be given the region label and its neighbors will be obtained and passed to IsCheckedBefore function in order to insert them at the back of the queue. Otherwise, the retrieved pixel will not be assigned any label. These process ends when the queue becomes empty which means the current region is not able to grow any more. Subsequently, the previous steps will start again with new unassigned pixel until all the pixels have labels. The implemented function takes into consideration both gray level and color images and gives the segmented image and the number of segmented regions as output variables.

4 Experimental section and results

This section presents the results of testing both region growing and Fuzzy C-Means (FCM) clustering algorithms on a number of gray and RGB images, which are shown below, in terms of speed and quality.

(a) coins (b) color (c) gantrycrane (d) woman

Figure 1: Experimental Images

4.1 FCM

The following table demonstrates the time required for each image to be segmented into 2, 3, 4, and 5 regions by FCM algorithm. The same results are drawn in figure 2.

Table 1: FCM algorithm

Images or characteristics	Type	Size	Elapsed time by FCM algorithm with different number of regions R			
			R=2	R=3	R=4	R=5
gantrycrane.png	RGB	400*264 pixels	T=1.889s	T=5.357s	T=5.729s	T=14.265s
woman.tif	RGB	116*261 pixels	T=0.82s	T= 1.432s	T=2.342s	T=2.832s
color.tif	RGB	275*278 pixels	T=3.598s	T=2.301s	T=2.266s	T=2.586s
coins.png	Gray	300*246 pixels	T=0.842s	T=3.263s	T=3.781s	T=6.580s

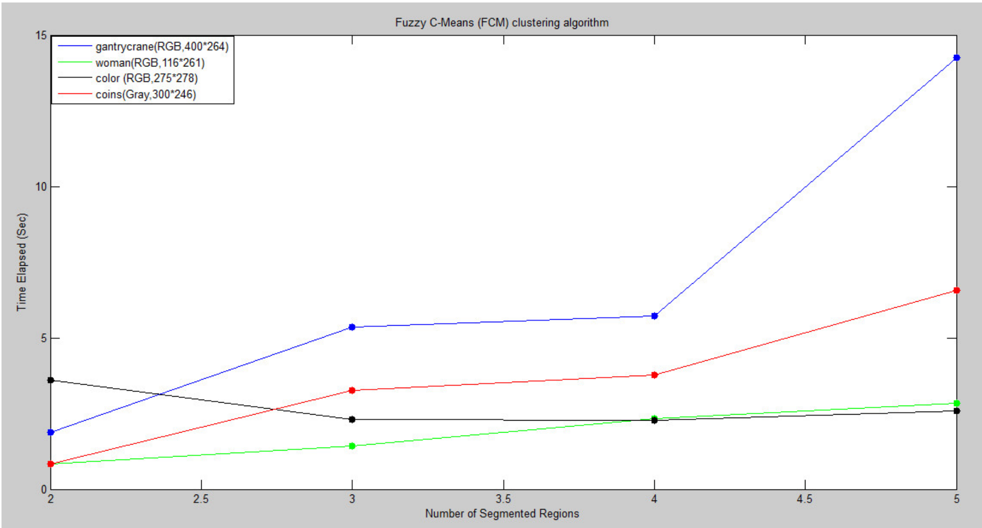


Figure 2: FCM-T-R Graph

As it is evidently shown in figure 2, greater number of segmented regions or bigger size of the image to be segmented implies that more time is required for the FCM algorithm to segment this image. In fact,this difference will be more obvious for higher numbers of regions and pixels. The reason for this increase in time is that more number of regions or pixels means that there are more mathematical operations to be computed and more iterations to be processed since the objective function of the FCM algorithm is highly dependent on these two numbers.
FCM, actually, is based on the minimization of the following objective function.

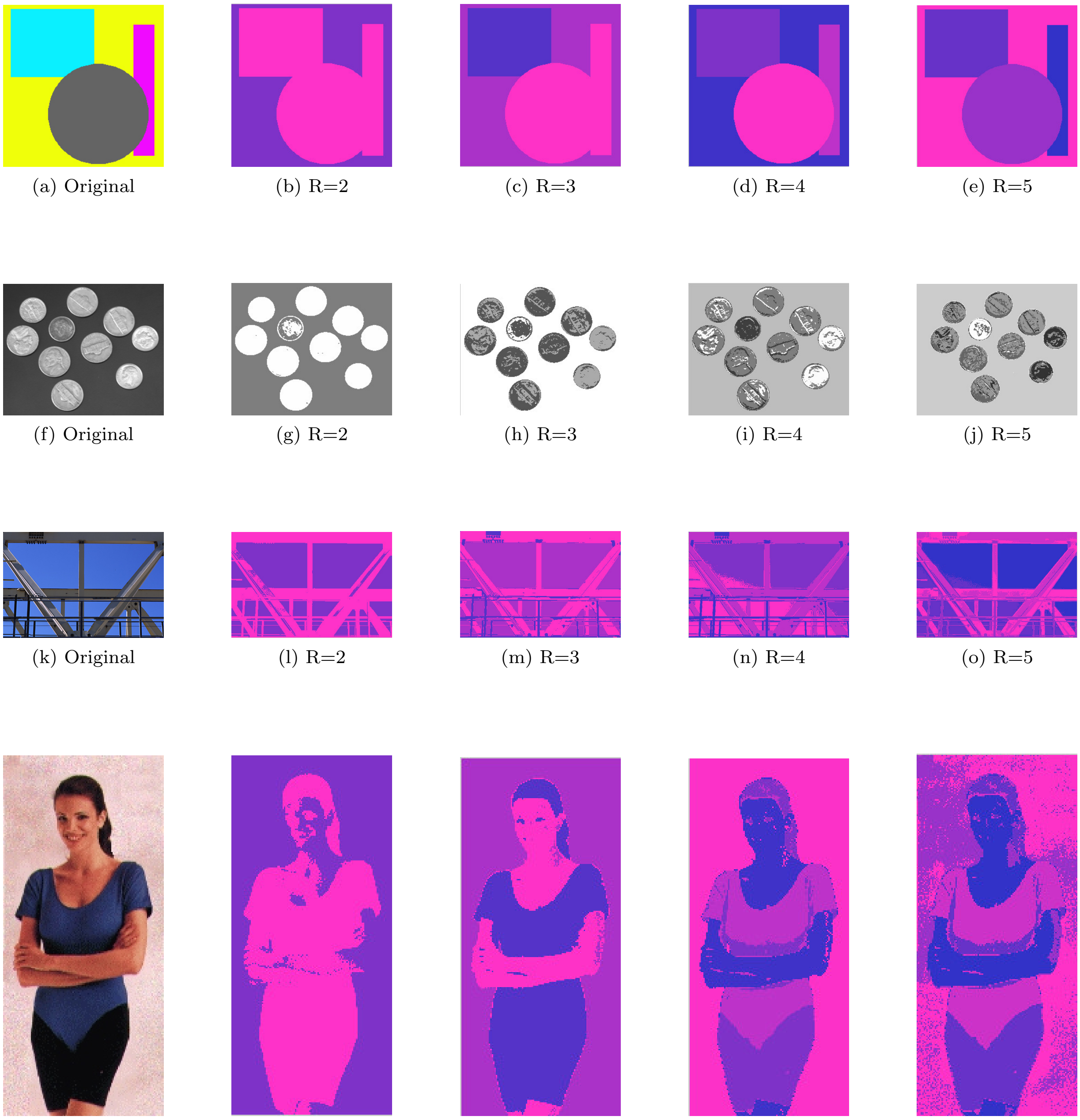
$$J_m = \sum_{i=1}^{D} \sum_{j=1}^{N} \mu_{ij}^m \left\| x_i - c_j \right\|^2$$

where

- D is the number of data points.
- N is the number of clusters.
- m is fuzzy partition matrix exponent for controlling the degree of fuzzy overlap, with m > 1. Fuzzy overlap refers to how fuzzy the boundaries between clusters are, that is the number of data points that have significant membership in more than one cluster.
- x_i is the ith data point.
- c_j is the center of the jth cluster.

- $\mu_i j$ is the degree of membership of x_i in the j th cluster. For a given data point, x_i, the sum of the membership values for all clusters is one.

The following images are the results of the segmentation performed by FCM algorithm and presented in table:1.



(a) Original (b) R=2 (c) R=3 (d) R=4 (e) R=5

(f) Original (g) R=2 (h) R=3 (i) R=4 (j) R=5

(k) Original (l) R=2 (m) R=3 (n) R=4 (o) R=5

(p) Original (q) R=2 (r) R=3 (s) R=4 (t) R=5

Figure 3: Segmented Images

In terms of quality of the segmented images, since we do not have any groundtruth images for the results to be evaluated according to sensitivity and specificity, we can visually judge the quality of the segmented regions. Indeed, they range from undersegmented to oversegmented images depending on the value of R and images themselves, variety of colors in each. For example, R=2 case could be considered as an undersegmentation result for the "woman" and "color", which is a synthetic image with four different colors, images while it is more visually acceptable for "coins" and "gantrycrane" images since the coins and the background are separated in the former one and the crane and the sky in the latter, which is as a result of the existence of two dominant colors in both images.

Figures 4 to 7 show the histogram of the “coins” image with the segmented clusters for different values of R. This, in fact, helps more to understand the results and to see that most of the pixels intensities are concentrated around two main centroid.

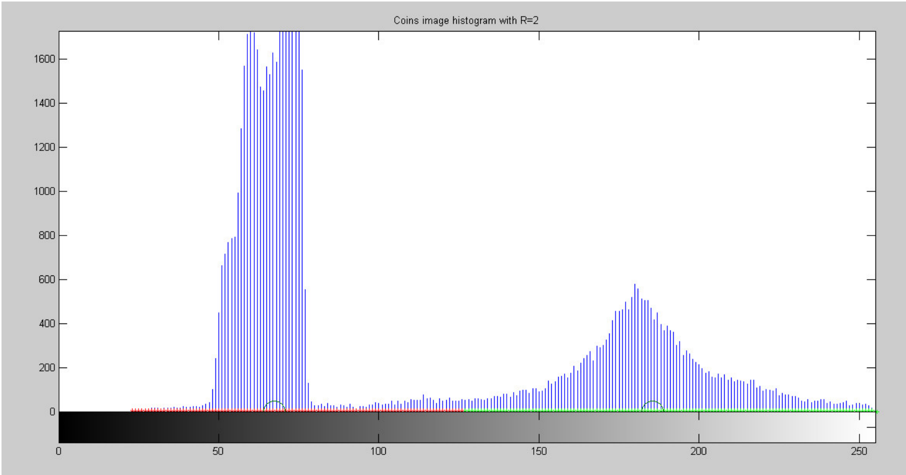


Figure 4: Coin histogram with R=2

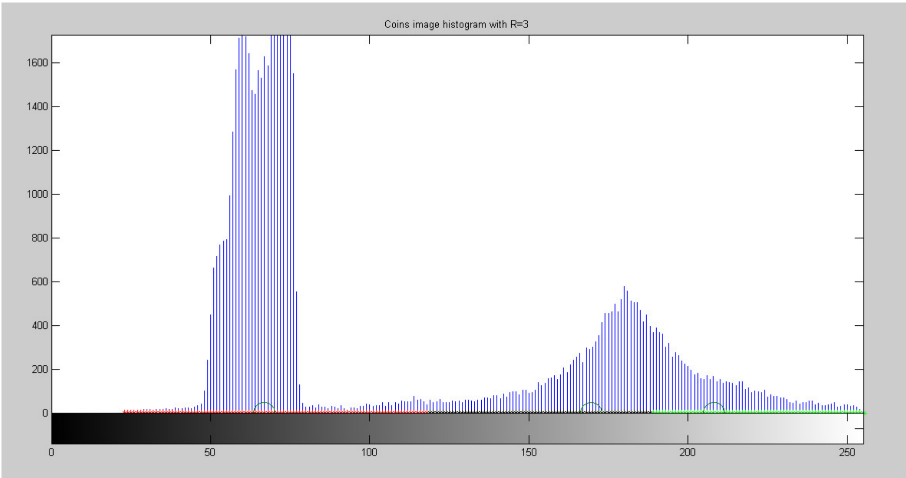


Figure 5: Coin histogram with R=3

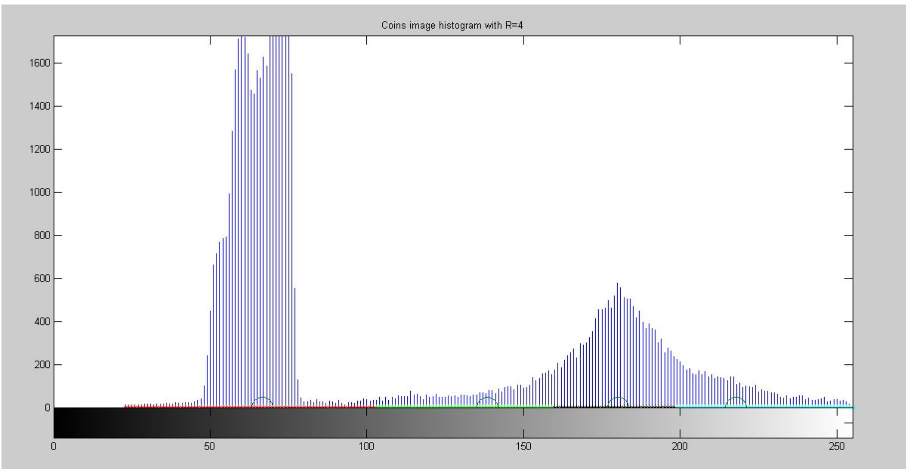


Figure 6: Coin histogram with R=4

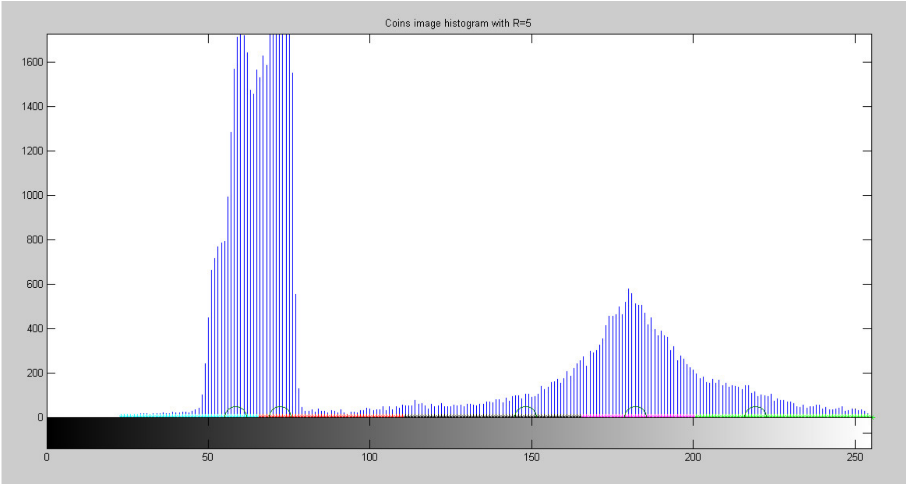


Figure 7: Coin histogram with R=5

Other values of R like 5 could be considered as an oversegmentation result for all of the previous images except the "color" image since there are only four different colors in this image which prevents the FCM algorithm from segmenting it into more than four regions. Furthermore, due to the clustering based inherited nature of FCM algorithm, the segmented regions in the previously studied images are not continuous except in the synthetic image. On final thing to comment on is for the synthetic image, when trying to segment it into more than three regions, the value of the minimization function will tend to at some point to infinity. Therefore, to prevent this case, the value of the fuzzy partition matrix should be set to a number higher than 2, which is the default value, in order to raise the fuzziness. Figures 8 to 16 show pixels in clusters of the three color images ("woman","gantrycrane","color") in RGB space for different values of R, which makes it easier to notice the Euclidean distance between those pixels and consequently comprehending the result of any clustering based algorithm conspicuously. Blue stars in the following figures represent the centriods of the different clusters

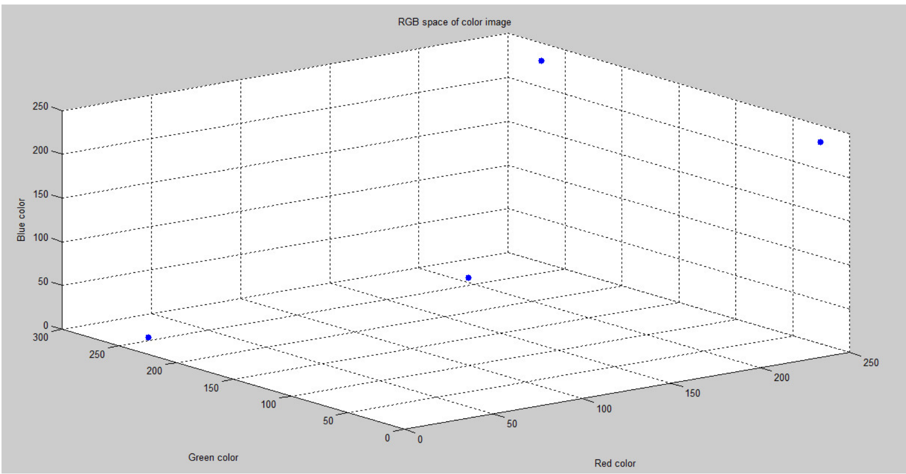


Figure 8: Color image RGB space with R=2,3,4,5 (same result)

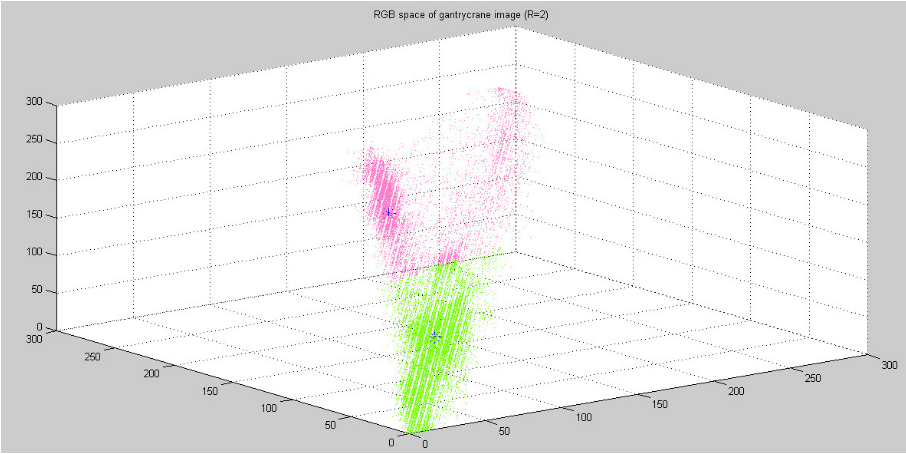


Figure 9: crane RGB space with R=2

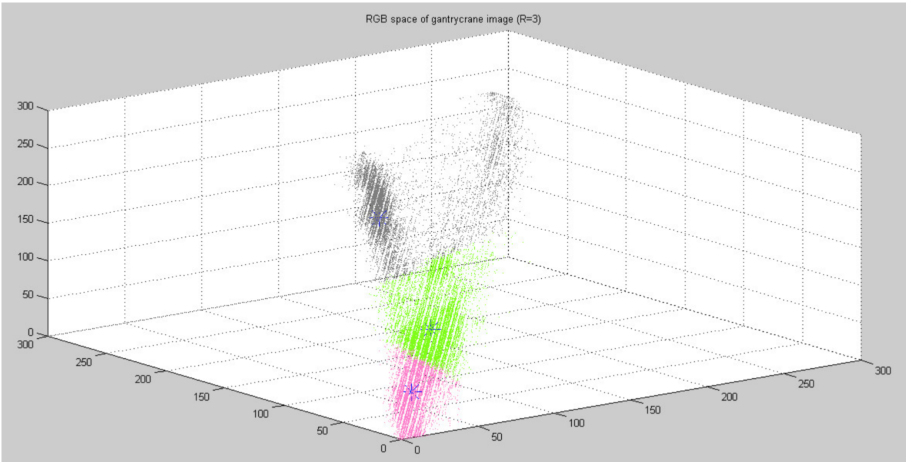


Figure 10: crane RGB space with R=3

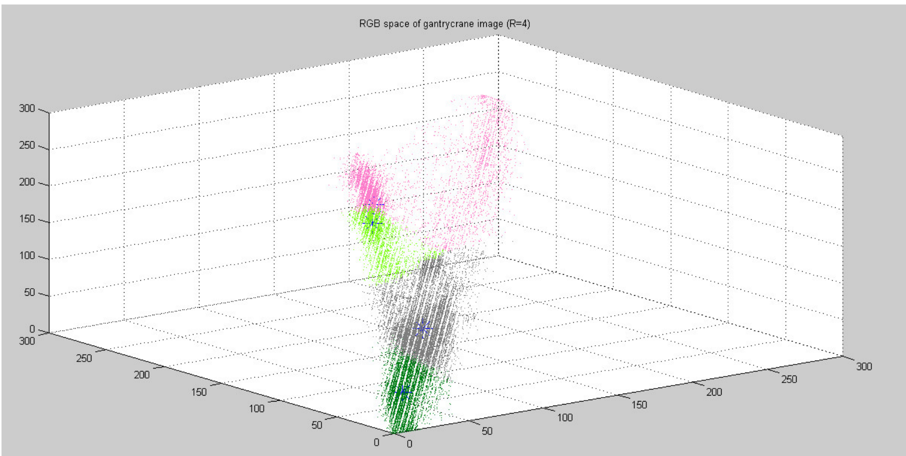


Figure 11: crane RGB space with R=4

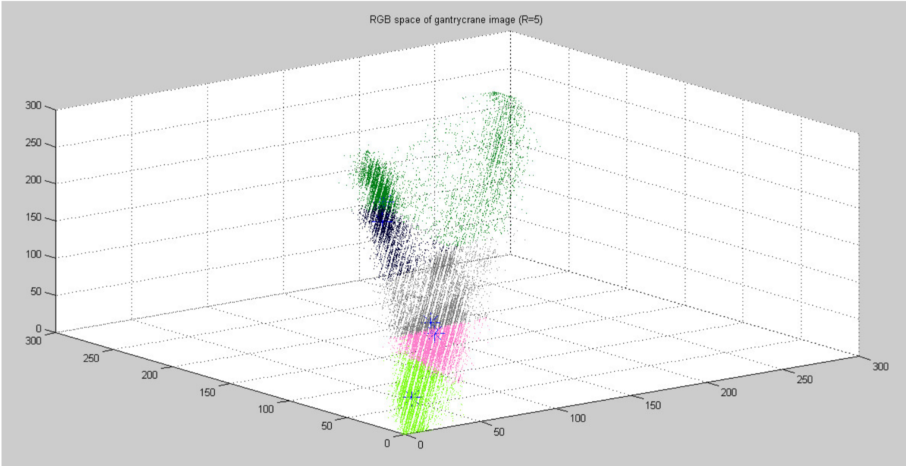


Figure 12: crane RGB space with R=5

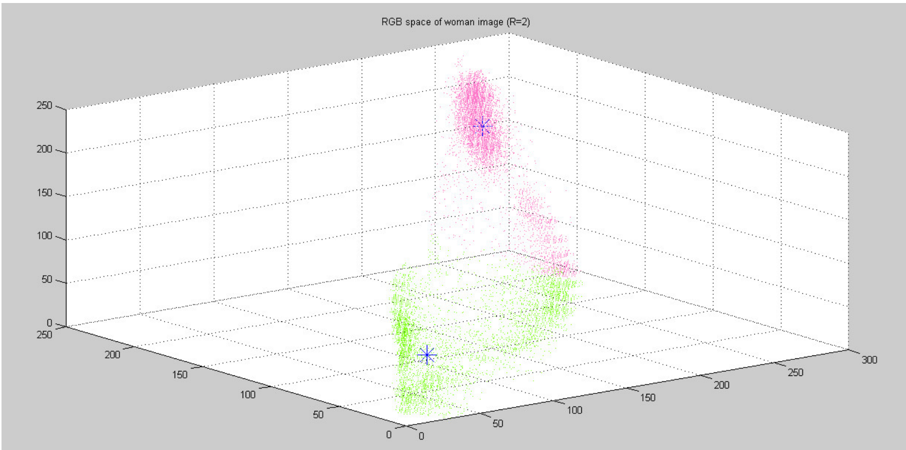


Figure 13: woman RGB space with R=2

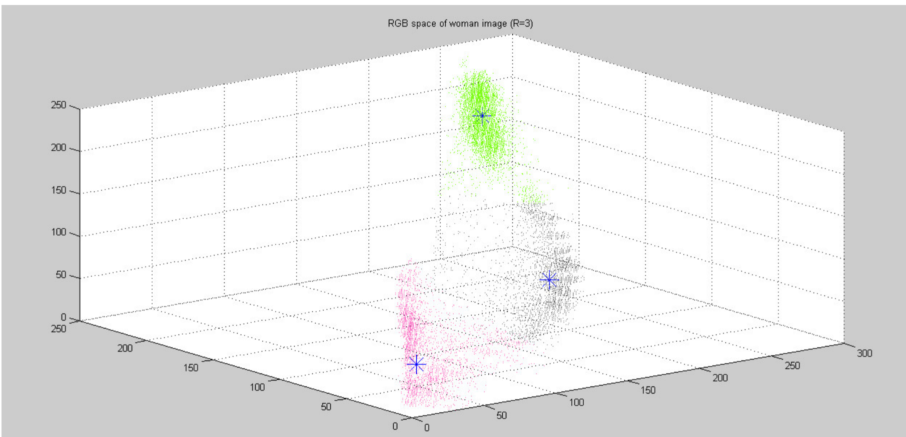


Figure 14: woman RGB space with R=3

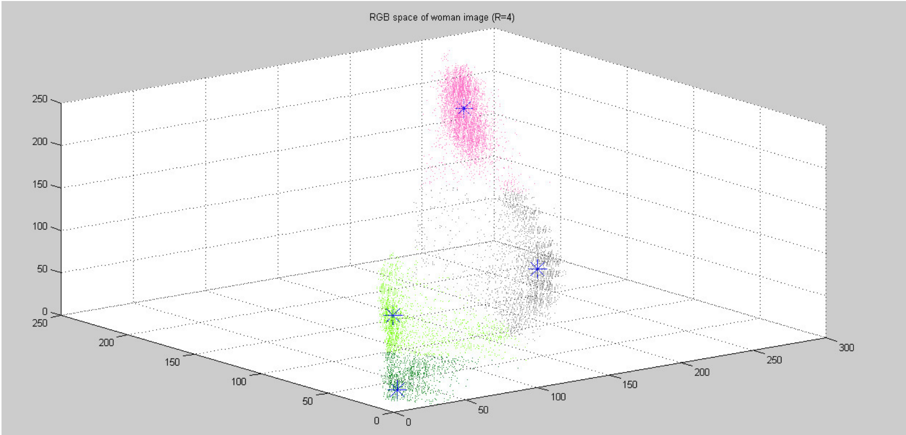


Figure 15: woman RGB space with R=4

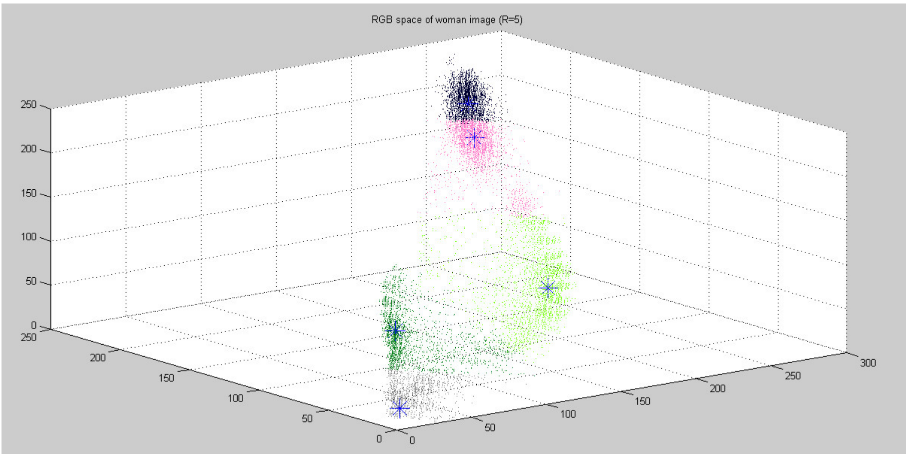


Figure 16: woman RGB space with R=5

4.2 Region Growing algorithm

As the number of segmented regions in the implemented Region-Growing algorithm depends on the value of the parameter delta, the timing results will be presented in terms of delta and image size instead.

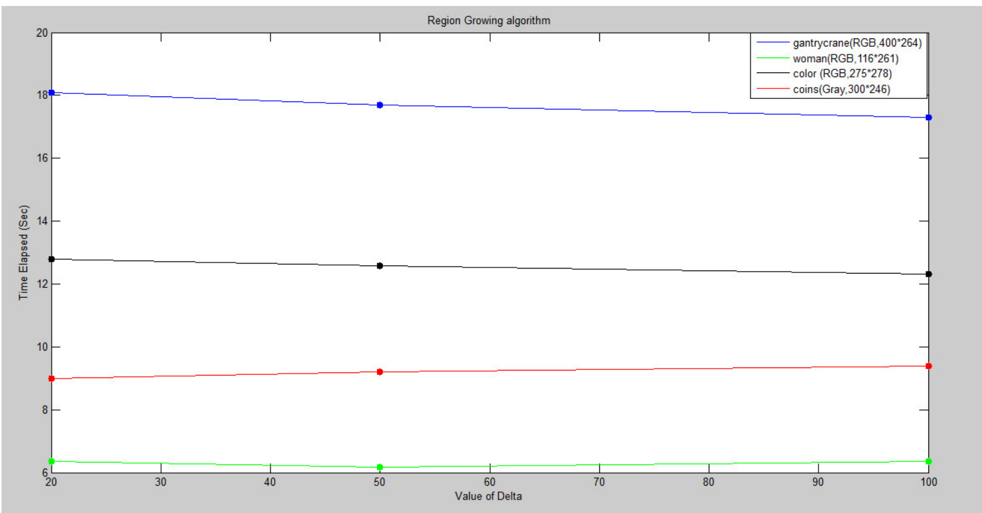


Figure 17: Region growing comparison

Table 2: Region growing Algorithm results

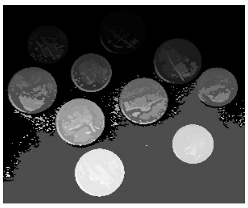
names or diff aspects	Type	Size	Elapsed time by FCM algorithm			R at different D's
gantrycrane.png	RGB	400*264 pixel	D=20 T=18.084s	D=50 T=17.691s	D=100 T=17.294s	R=7039, 2467, 705
woman.tif	RGB	116*261 pixel	D=20 T=6.361	D=50 T=6.184	D=100 T=6.352	R=3489, 666, 104
color.tif	RGB	275*278 pixel	D=20 T=12.802 s	D=50 T=12.577s	D=100 T=12.311	R=4,4,4
coins.png	gray	300*246 pixel	D=20 T=8.985	D=50 T=9.195	D=100 T=9.392	R=583,105, 121

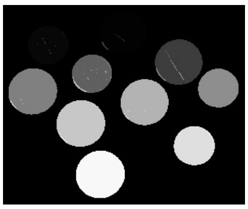
The results demonstrated in table:2 and drawn in figure:17 suggest that the time required for segmenting the studied images is quite independent of the value of Delta and just depends on the size of the original image. This is a straightforward consequence because the main idea of the written algorithm is to iterate through each pixel which makes it reliant on the size of the image even though the value of Delta affects the number of the resultant regions. However, there is only one additional thing to be calculated for every new region which is the initialization phase and this phase is somewhat negligible compared with the other carried operations. The following images are output of the region growing algorithm with the same results shown in table:2.

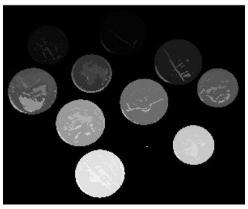

(a) D=20

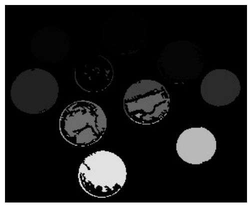

(b) D=50

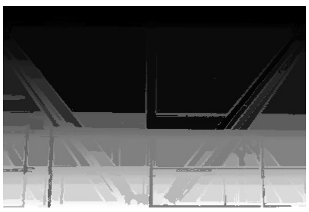

(c) D=100

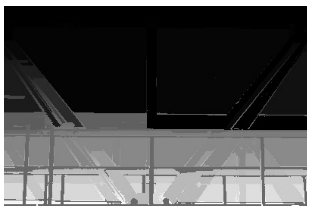

(d) D=10

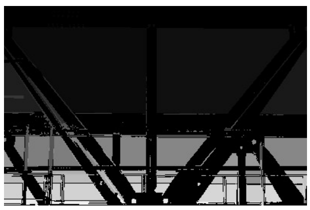

(e) D=50

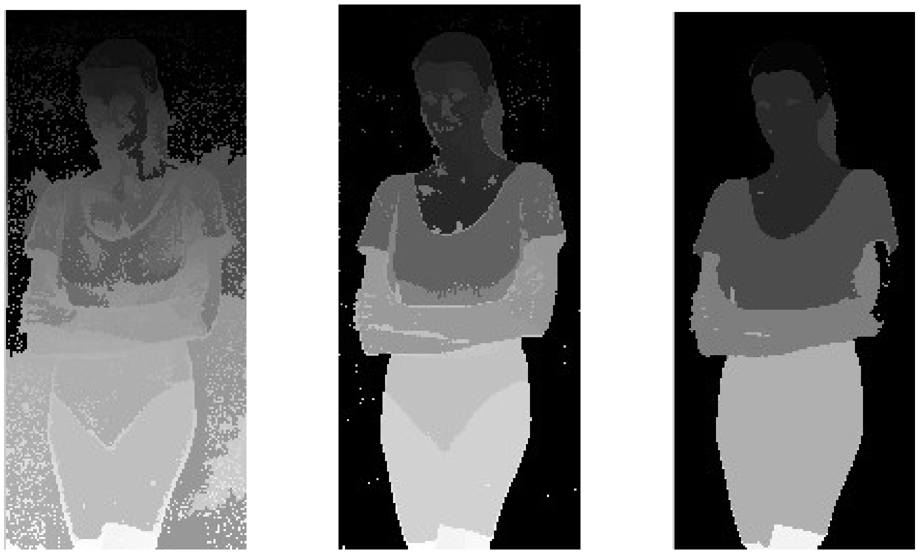

(f) D=20


(g) D=100


(h) D=20


(i) D=50


(j) D=100



(k) D=20 (l) D=50 (m) D=100

Figure 18: Segmented Images

Table:2 states that there are high numbers of regions in the output images for three different values of Delta. The reason why those high numbers exist is that the vast majority of those regions consist of very small number of pixels, most of the time it is only one pixel. What happens usually is that some sparse pixels with high values are surrounded by groups of pixels with comparatively smaller values which makes them constrained and as a result they constitute separate regions. It is noteworthy that when a pixel is surrounded by 4 neighbors of intensities that break the aggregation condition, this pixel will be considered as a separate region which is the case in our program. For example, for the case of "coins" image and value of Delta equal to 50, the output image contains 105 regions with different labels. One region with 50393 pixels out of 73800 in total, 10 regions with 1800 to 2500 pixels, and the rest are regions of only one pixel. Visually speaking, it is not possible to recognize all these regions because the output image is of a gray-level type and there are only 256 levels. That is why a contrast scaling step is done before returning the segmented image. Another noticeable aspect of the output images is that some of the regions are little noisy because of the 4-neighborhood connectivity considered in the implemented algorithm. Nevertheless, all of them are continuous in nature. Finally, visually we can say that the synthetic image was segmented into four regions for three different values of Delta, because it contains only four different colors and they are on a higher Euclidean distance from each other than the used values of Delta. Others, like "coins", "gantrycrane", and "woman" range from oversegmented images for Delta=10 to undersegmented images for Delta=100 although for this value of Delta the women image could be considered fine for some applications.

5 A brief discussion regarding the algorithms

Region growing algorithm, in fact, has a number of prominent advantages, however, it also has some disadvantages. One of the main drawbacks of this algorithm is that it is dependent upon the placement of the initial seeds. This means that uniqueness in the solution is not guaranteed for dissimilar initial seeds positions. In general, it is sensitive to noise and moreover the aggregation condition is quite critical part of this algorithm and considerable attention should be paid here. Obviously, the connectivity or pixel adjacent information is helpful for us to determine the threshold and seed points. Nevertheless, it is an easy algorithm to implement and can correctly separate the regions that have the same properties we define. Finally, it is a region based algorithm making it reliant on the spatial information as well.

On the other hand, Fuzzy C-means algorithm is an algorithm that is very similar to k-means algorithm. The main constraint on this method, as we noticed during this lab, is the determination of the numbers of clusters initially. It works only with the features space, as opposed to region growing algorithm, rather than the spatial information. One of the very important parameters of this method is the exponent of the fuzzy partition matrix. This parameter controls the amount of fuzzy overlap between clusters, with larger values indicating a greater degree of overlap.

If the image pixels intensities set is wide with a lot of overlap between potential clusters, then the calculated cluster centers might be very close to each other. In this case, each data point has approximately the same degree of membership in all clusters. Experimentally we have noticed that improving the clustering results could be done by decreasing this value, which limits the amount of fuzzy overlap during clustering. On the contrary, for clearly separated pixels, the case of the synthetic image, it better to put up this value.

6 Organization and development of the coursework

At the begging, it was very important for us to review what we had learnt previously in the lecture about region based segmentation algorithms in general and specifically about region growing method. Then, we started to think about and discuss the practical steps of the possible implementation. Initially, we implemented what is usually considered in programming world as a quick and dirty solution just to be familiar with the different parameters and the methodology of implementation in general. Our first time response for the coins image, for instance, was about 8 minutes, which is not good. After that, we started to analyze the run time required for each part and tried to separate every task of the algorithm by putting it in a different function, easier for debugging purposes. What we noticed is that our way of building the queue data structure and defining some additional parameters and functions was pretty expensive in terms of time because it was mainly based on many built-in search functions in Matlab. Next, we improved this part and embedded most of the separate functions in only one main function, including the queue, since function calls in Matlab is quite time heavy task. Eventually, we have got a rather optimized version that takes only few seconds to segment the provided images. Therefore, the first session was spent to implement our first version and then we spent the week after that session for optimizing our method. During the second session, we started to play with the FCM algorithm and put the results in tables and figures for the sake of comparison. The final step was to write the report in order to explain what we have done so far.

7 Conclusions

In this lab sessions, we have successfully designed, analyzed and implemented the region growing algorithm in Matlab program and tested it on different types and sizes of gray and color images. We have also drawn a comparison between the implemented algorithm and the Fuzzy C-means algorithm, which has a built in function in Matlab, in terms of many aspects, speed quality and critical differences.