University of Girona

Image Segmentation and Interpolation

Lab 1

Introduction to ROS

Author:
Di MENG Darja STOEVA

Supervisors:
Dr. Xavier Lladó
Dr. Arnau Oliver

March 2, 2017

1 Introduction and Problem Definition

The idea behind image segmantation is to divide the pixels of an image into regions, where each pixel belonging to a region shares particular characteristic such as color, texture, etc. Segmenting an image in different regions by marking them with labels accordingly, has a wide range of application in different fields since it emphasizes the edges present in the image and makes it easier to analyse. As an image processing tool, image segmentation plays an important role in the Copmuter Vision field. There are several approaches to split an image such as: using a threshold on the pixel properties such as the intensity values, using discontinuity and similarity in intensity levels or finding the regions directly [1]. One of the well known methods to preform an image segmentation is the Region Growing algorithm which is considered to be a fast, easy to implement while providing goood results.

The aim of this lab was to get familiar with how the Region Growing algorithm works by implementing and testing it on several images. The implemented algorithm was then compared with the MATLAB Fuzzy C-means clustering algorithm which was tested on the same images.

2 Algorithm Analysis

The Region Growing algorithm groups pixels into regions based on predefined criteria for growth. The algorithm starts from an initial point, known as the 'seed' point, and grows regions by checking the pixel neighbours, if the neighbours of the seed point have similar properties it groups them together with the same label. The method is repeated for each pixel that is not label until all the pixels of the image belong to a region [1].

When checking the neighbours, 4 or 8 neighbours can be considered. The choice of which connectivity to implement depends on the input image and what we want to extract or analyse. The pseudo code for the algorithm can be seen in Figure 1. The next section will explain our approach and how we implemented the Region Growing Algorithm.

```
Data: Input Image
Result: Labeled Image
labelvalue = 1
for all the pixels(x,y) of the input image do
    if pixel(x,y) is not visited) then
        enqueue (x,y)
        while queue not empty do
            p ← dequeue
            if p then
                add p to segment(labelvalue)
                enqueue neighbours of p
                mark p as visited
            end
            labelvalue ← labelvalue + 1
        end
        Mark pixel(x,y) as visited
    end
end
```

Algorithm 1: Pseudo code: Region Growing Algorithm

3 Design and Implementation of the Proposed Solution

About the implementation of this algorithm, the code basically does the following steps:

1. Determine that if the image is gray or color, choose their own way to do segmentation. (The only difference of segmenting gray or color images is the method of computing the mean pixel values. Because the gray image has one channel and color image has three.)

2. Create a new empty image which has the same size with input image for saving labels.

3. Go through all the pixels in the label image. If the current pixel is not labeled, then put current position in the queue and label it. Assign the initial mean value with the first pixel value in the queue.

4. Go through all the pixels in the queue. Check the values of 8 neighbors of each pixel, if the difference with mean value is smaller than the given threshold, that means the current pixel is belonging to the same region, label it. If not, leave it there as zero.

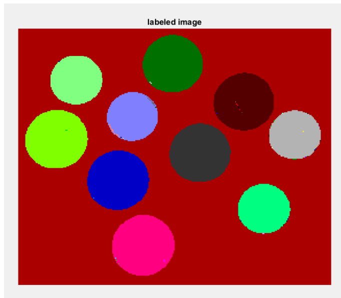
5. When a queue is empty, that means one region is finished searching. Then go to the next pixel without label, redo the former steps until the queue's got empty.

6. When there's no pixels with zero values in the label image, the program goes over. The image has been segmented completely.

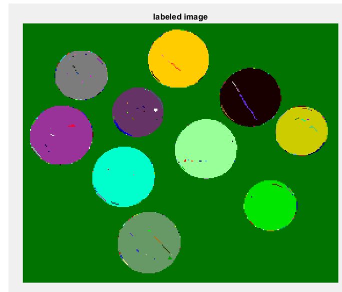
4 Experimental Section and Results

4.1 'coins' - Region growth

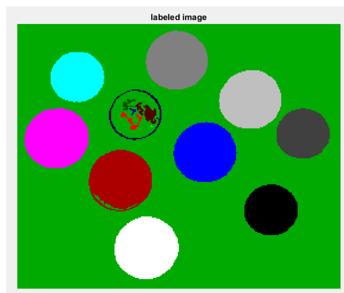
	Threshold	**Labels**	**Performance**
No filter	69	56	Proper
0.954195 s	50	322	Over segmented
	80	17	Under segmented
Filter	60	133	Proper
0.943825 s	50	245	Over segmented
	70	23	Under segmented



(a) No filter, threshold: 69

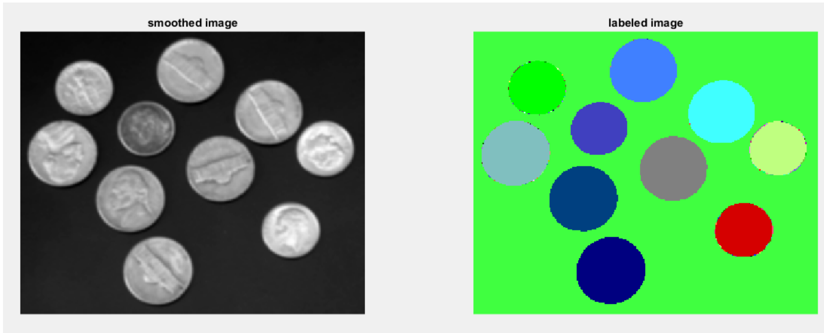


(b) No filter, threshold: 50

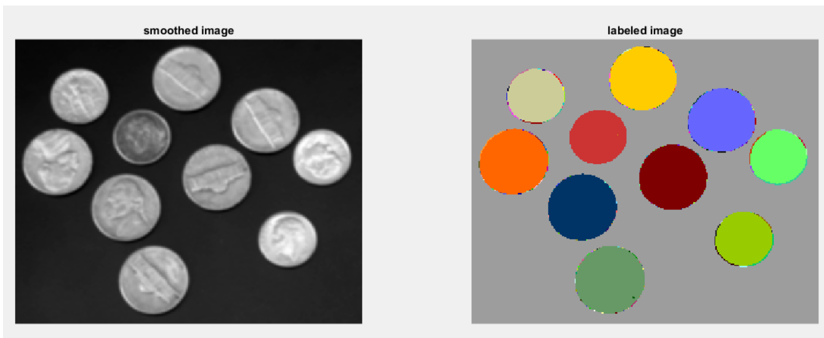


(c) No filter, threshold: 80

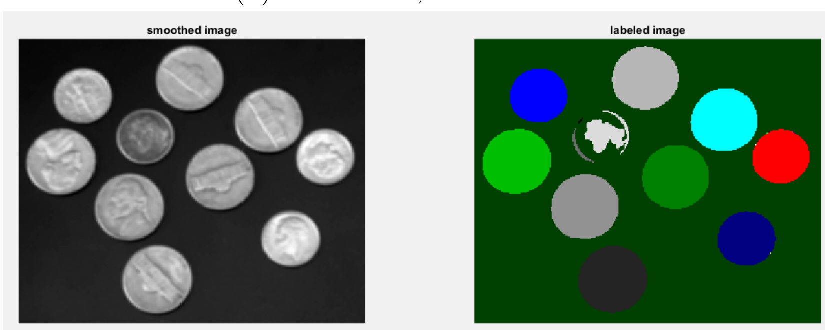
Figure 1: Segmentation on coins without smoothing



(a) With filter, threshold: 60



(b) With filter, threshold: 50



(c) With filter, threshold: 70

Figure 2: Segmentation on coins with smoothing

4.2 'coins' - Fuzzy C means

Number of regions : 2

Elapsed time: 18.315670 s

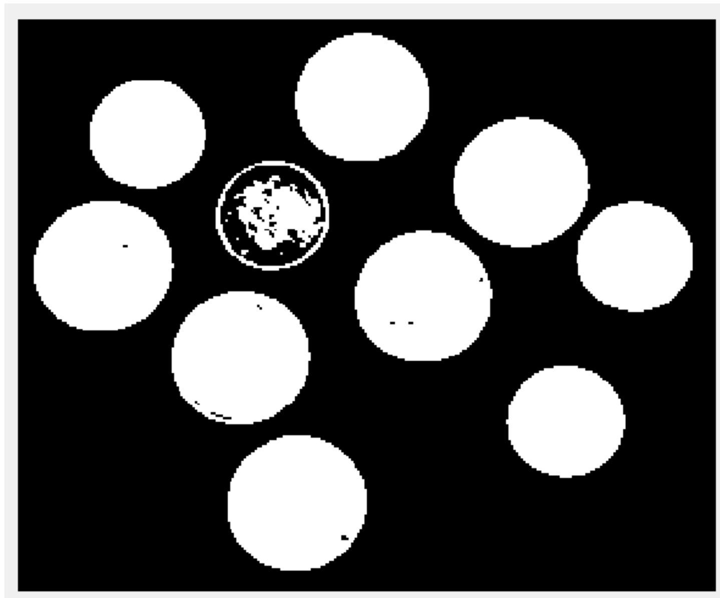
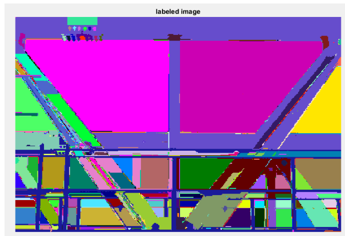


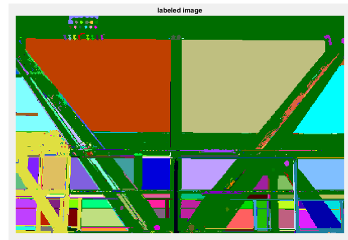
Figure 3: Fuzzy c means : coins

4.3 'gantrycrane' - Region growth

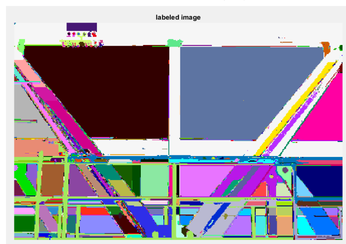
	Threshold	Labels	Performance
No filter	50	1415	Proper
1.350282 s	85	782	Under segmented
	40	2137	Over segmented
Filter	50	427	Proper
1.307392 s	85	144	Under segmented
	40	659	Over segmented



(a) No filter, threshold: 50

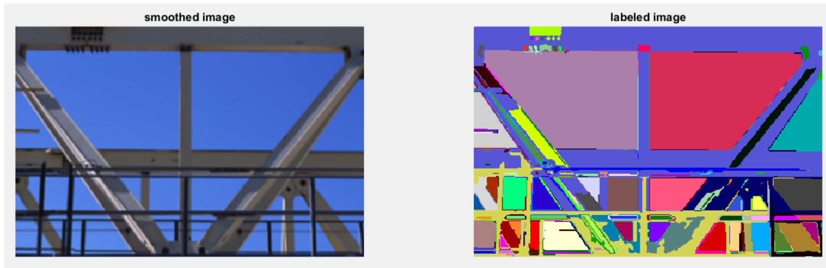


(b) No filter, threshold: 85

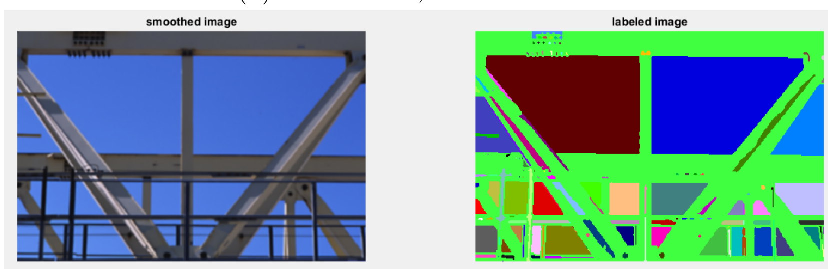


(c) No filter, threshold: 40

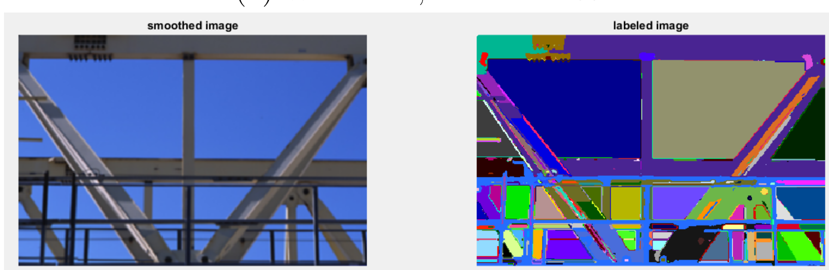
Figure 4: Segmentation on gantrycrane without smoothing



(a) With filter, threshold: 50



(b) With filter, threshold: 85



(c) With filter, threshold: 40

Figure 5: Segmentation on gantrycrane with smoothing

4.4 'coins' - Fuzzy C means

Number of regions : 3

Elapsed time: 6.290390 s

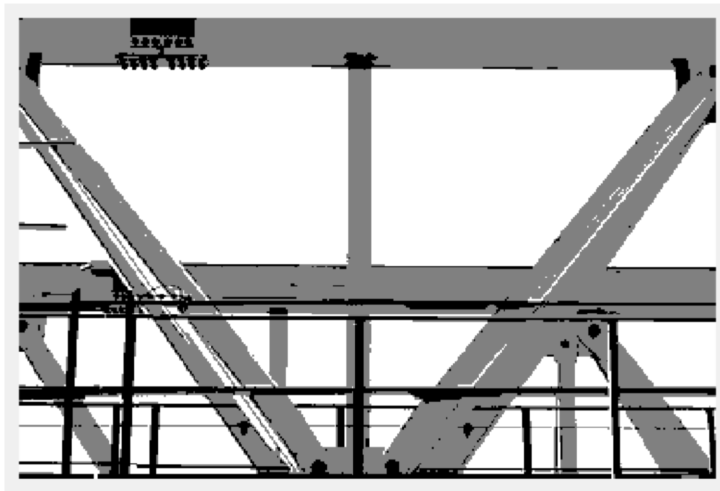


Figure 6: Fuzzy c means : gantrycrane

4.5 'woman' - Region growth

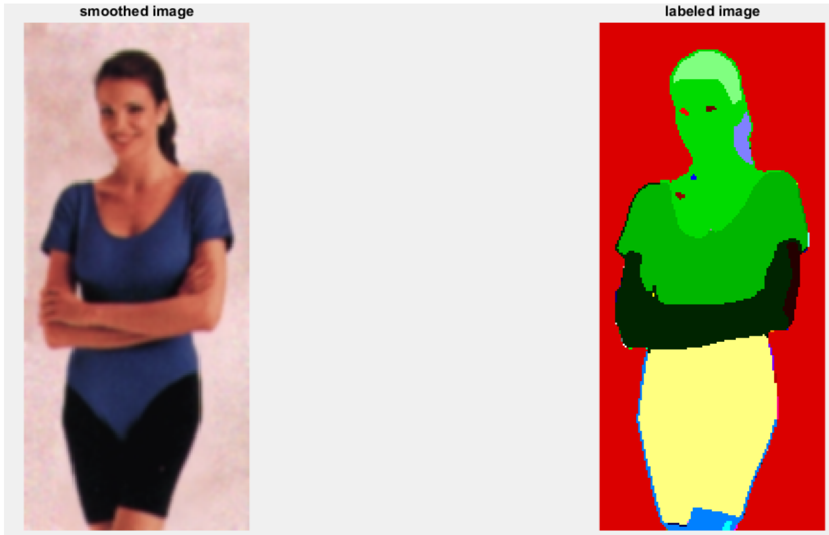
	Threshold	**Labels**	**Performance**
No filter	80	179	Proper
0.517793 s	100	91	Under segmented
	70	269	Over segmented
Filter	80	47	Proper
0.501369 s	100	21	Under segmented
	70	134	Over segmented

(a) No filter, threshold: 80

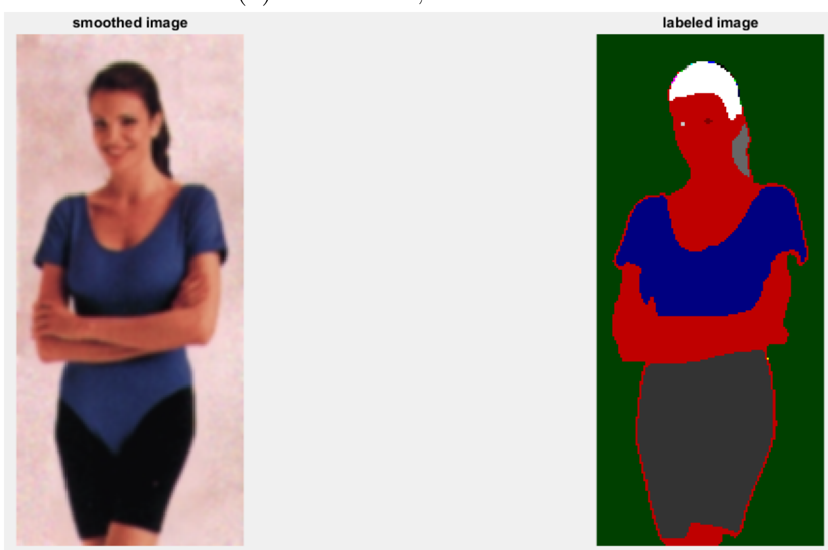
(b) No filter, threshold: 100

(c) No filter, threshold: 70

Figure 7: Segmentation on woman without smoothing



(a) With filter, threshold: 80



(b) With filter, threshold: 100

(c) With filter, threshold: 70

Figure 9: Fuzzy c means : woman

4.6 'woman' - Fuzzy C means

Number of regions : 4

Elapsed time: 5.947931 s

5 Discussion

As it can be seen in the Results section, both of the algorithms give different results dependingly of the comosition of the image and the threshold value for the Region Growing or the number of labels for the Fuzzy C-means.

The Region Growing method separates the regions good when the edges are well defined. In order to improve the performance of the algorithm, preprocessing of the image was applied with a Gaussion kernel. When the image is smoothed, the elapsed time is smaller and the final results show better labeling of the regions. One of the holdbacks of this algorithm is the choice of the threshold. We had to try multiple threshold values for each image in order to find the optimal threshold, and avoid oversegmentation and undersegmentation.

On the other hand the Fuzzy C-mean cluster gave a bit better results. One of the reasons for this might be the need to indicate how many regions we want to be labeled. So in order to have a good result we have to know how many labels we need, but even then depending on the shadows and the quality of the image, the algorithm sometimes does oversegmentation or undersegmentation. When compared to the Region Growing, this algorithm has a higher elapsed time and less freedom on the way of clustering the pixels.

6 Conclusion

In conclusion, during this lab we have seen that the Region Growing algorithm is easy to implement and works better when smoothing is applied to the input image. Very important feature of this algorithm is the threshold, while the Fuzzy C-means has a big dependency on the number of regions that are chosen.

References

[1] Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins. *Digital Image Processing Using MATLAB*. 2nd.