Scene Segmentation and Interpretation
Image Segmentation using Region Growing

Oscar E. Ramos, Babak Rezaei
MSc in Computer Vision and Robotics
Universitat de Girona, 2010

Abstract. This report presents the usage of the region growing segmentation algorithm for some color images. Details of the implementation and development of the algorithm are presented and its pros and cons are discussed empirically, based on the results obtained.

I. INTRODUCTION

IMAGE segmentation consists in decomposing an image into tessellations based on a certain criteria. It is one of the essential parts of both image processing and computer vision since it acts as a preprocessing step for pattern recognition and object detection. Segmentation is usually applied as an unsupervised method to detect regions of interest with a certain characteristic and the constraint that the segmented regions should be homogeneous and without many holes inside. Equivalently, the similarity criteria should be uniform for pixels inside the same region and should vary among different regions.

Segmentation methods can be broadly categorized into region based methods, such as region growing or split and merge, and clustering methods such as K-means, clustering and EM method. The focus of this report is on the region growing method. Region growing operates by selecting a seed point and analyzing the neighbors, whether they should be included in the same region or not. Growing regions from seed points have the advantage of generating coherent regions and unlike clustering methods they provide the spatial information about the regions. However, the problem with these methods is that they are dependent on the placement and the number of seed points. Besides that, the aggregation criterion for adding a new pixel to a region is not trivial and has to be carefully chosen.

The main goal of this coursework is to implement a region based algorithm that best segments the images provided and compare them with the ground truth, which is a manually segmented image that resembles the way in which a human would segment the image.

II. REGION GROWING ALGORITHM

A. General Issues

Region Growing method starts with one or more seeds and fills the regions starting from the seed points so that the neighbors are visited in a specific order and depending on the fulfillment of the similarity criteria, also called aggregation criteria, they can be added to the region. This process is done in an iterative way for all the pixels in a region. As it can be seen, this algorithm introduces several issues like:

- How to choose the adjacent pixels.
- How to define the similarity or aggregation criteria and what characteristics of the region should be used as the criterion.
- Supposing that $S(p,q)$ is the similarity measure between pixels p and q, where q is the candidate pixel, what can p be?, in other words, what is the representative characteristic of the region.
- How to define the threshold required to render the decision on the similarity measure.

The following parts will propose some solutions to the above mentioned problems.

B. Description of the algorithm

The approach that was taken for the implementation of the region growing algorithm, solving some of the stated issues, is the following.

1. Select the first element of the region. The very first one will be the pixel at the top left corner of the image. Then, start a new empty queue. Another option would be to start the first

point randomly; however, this approach was not considered in this work.

2. Look for the neighbors using 4 adjacency, starting with the top one and going clockwise as shown in Figure 1. Special consideration should be taken with the pixels located at the borders since they don't have all the neighbors. In that case, pick only the available neighbors.

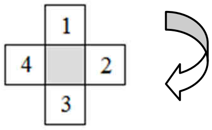


Fig. 1. Way in which neighbors of the current point are chosen

3. If a neighbor point does not belong to any region and it has not been added to the queue yet, it will be queued.
4. Select the following element in the queue (in the order the elements were stored) and verify whether it belongs to the region or not using a predefined criterion. The aggregation criteria used will be explained later. If the new point belongs to the region, i.e. the aggregation criterion holds, check for its neighbors as in step 2.
5. Repeat the process until the queue is empty. Once it is empty increase the region number and start the whole process again with the following pixel that does not belong to any region.

C. Aggregation Criteria

This criterion establishes whether a new point belongs to the current region or not based on certain measurements. For the following part, assume that $I(x,y)$ is the value of the pixel at the position x,y of the image and let the region R have n elements.

1. *Standard deviation criterion.* It consists in obtaining the mean and the standard deviation of the elements in the region and include the new point $I_{n+1}(x,y)$ in the region if it satisfies (1).

$$\mu_n - X\sigma_n < I_{n+1}(x,y) < \mu_n + X\sigma_n \qquad (1)$$

where X is the parameter that determines how far the new point can be from the mean of the region, measured in terms of the standard deviation. Statistically, 99.73% of the points that belong to a region are included if X is considered as 3. The lower the value of X, the fewer points will be added to the region and the more regions there will be in the segmented image.

In the colored image, the criterion has to be satisfied for each one of the planes (R, G, B) simultaneously. In this case a logical AND will be used to express that condition.

For the first pixel, since a zero standard deviation would mean that the following pixel has to be exactly identical to the first one, a not too realistic condition, the criterion of computing the standard deviation of that first point and the one that is going to be evaluated was rather considered.

2. *Threshold criterion.* It consists in obtaining the mean of the region and adding the new point $I_{n+1}(x,y)$ if it is closer to the mean by a fixed threshold X, that is, if it satisfies (2).

$$\mu_n - X < I_{n+1}(x,y) < \mu_n + X \qquad (2)$$

For the color image, the threshold condition has to be satisfied in the three planes simultaneously to be considered in the same region.

3. *Euclidean distance criterion.* This will be only applied to the color image since the three planes will be considered as coordinates that generate the tridimensional color space. In this case, the new point $I(x,y)$ will be added to the region if the Euclidian distance from the mean of the region is shorter than a value X, as stated by (3).

$$\sqrt{(I_R-\mu_R)^2+(I_G-\mu_G)^2+(I_B-\mu_B)^2} < X \qquad (3)$$

where $I_R = I_R(x,y)$, $I_G = I_G(x,y)$ and $I_B = I_B(x,y)$ are each one of the planes of the $n+1$ element.

D. Post processing after segmentation

In order to improve the results of the segmentation, a median filter was applied to the segmented image so that little regions that can be considered noise are removed and only bigger regions are preserved.

For more accurate results, a second segmentation for only the Hue component of the HSV (Hue, Saturation and Value) color space was performed. That is, after the median filter was applied to the segmented image, the regions were mapped onto the original image in the RGB space so that the mean value of each region was displayed instead of the label itself. The new created image is in a certain way a "more softly" segmented image since it preserves the mean of the colors of the original image. Then, the space of this segmented image was transformed onto the HSV color space. The H component was picked and a new segmentation was applied to this single plane, as if it were a gray level image. The result was again filtered using a median filter to remove some additional noise.

III. Design and Implementation

The segmentation algorithm described in the previous section was fully implemented in Matlab. Table I gives a summary of the implemented algorithm. The further post-process of the image was also implemented to improve the segmentation results.

One issue in the implementation was the computational time that for the functions in Matlab. It is well known that this environment, even though easy to program, is not well suited for speed. The code was mainly implemented in one single file since the repetitive calls to separate functions make the whole process to take more time. Another issue for the speed of the computation was the way of calculating certain parameters as described in the following section.

TABLE I
ALGORITHM FOR THE REGION BASED SEGMENTATION

For every pixel $I(x,y)$ in the image, if $I(x,y)$ does not belong to any region

1. Take $I(x,y)$ as starting point and label it.
2. Initialize a new queue and add the neighbors that don't belong to any region to the queue.
3. While the queue is not empty
   a. Pop the first value from the queue
   b. If the adjacency criterion is satisfied
      - Add the pixel to the region (label it).
      - Obtain the neighbors that are not in the queue and don't belong to any region
4. Increase the label of the region.

A. Computation of the mean and standard deviation

When using all the points that belong to the region to compute the mean and standard deviation, the whole process is computationally expensive and slow in time. In order to solve that problem, a recursive way for the computation of these two parameters was implemented. The mean of the first N elements, μ_N, is given by (4).

$$\mu_N = \frac{(N-1)(\mu_{N-1})+I_N}{N} \tag{4}$$

Where I_N is the value of $I(x,y)$ for the N-*th* element. The standard deviation for the N-*th* element, σ_N, is computed using the previous standard deviation σ_{N-1} and the current mean μ_N as in (5)

$$\sigma_N = \sqrt{\frac{(N-2)(\sigma_{N-1})^2+\frac{N}{N-1}(I_N-\mu_N)^2}{N-1}} \tag{5}$$

where the single consideration is that the initial values have to be $\mu_1 = I_1$ and $\sigma_1 = 0$.

B. Functions developed

The functions that were created using the algorithm presented in Table I are *f_regiongrowcolor* for color images and *f_regriongrowgray* for single plane images. The input parameters are the image, the growing criterion ('*s*' for standard deviation, '*t*' for a predefined threshold and '*e*' for the Euclidean distance) and the parameter X according to (1), (2) and (3). For instance, we can have the following statements:

Regions1 = f_regiongrowcolor(image,*'s',3*);

Regions2 = f_regiongrowgray(image,*'t',40*);

In the first example, the criterion is *3* standard deviations for the color image and in the second case it is a fixed threshold of 40 away from the mean for the gray image.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

This section provides the results of the experiment carried out for the evaluation of the segmentation method. Several images have been used as test cases. The segmentation method has been developed both for the 1-D (e.g. grayscale) and 3D (e.g. RGB color space) cases. We varied the parameters such as the coefficient of the standard deviation in the aggregation criterion or the threshold in the thresholding method to achieve better results based on the obtained images.

The implementation of the algorithm has been optimized to provide less computational time. For this matter, statistics were computed incrementally and a static memory allocation has been used. Figure 2 shows the result of the segmentation algorithm for the image containing the coins. This may be the easiest one of the test examples as the image is grayscale. Thus, less computation is necessary for the evaluation of the aggregation criterion. Due to the contrast between the coins and background and uniformity of background and coins, the shapes are extracted with high accuracy. As explained in the implementation section, the segmentation starts from the seed point in position (1,1) which is the background for this image. The uniform grayscale value of the background helps its easy segmentation and all the coins are segmented properly using this method. The only problem encountered was that the edges produce extra segments due to the contrast of the pixels in these areas with the neighbor pixels. This has been solved by applying a median filter with size of 5 which removes the outliers, in this case, the extra regions. It should be noted that the results in Figure 2 have been obtained using the mean of the region as the aggregation criterion with a threshold of 22 as described by (2). The whole segmentation process takes approximately 2.87 seconds. For RGB color images, the single intensity level segmentation does not provide satisfactory results as the intensity level does not offer the adequate discriminatory power for segmenting the different colors in the image.

Figure 4 shows the result of the segmentation of the "plane image" using different parameters and two different approaches but utilizing the standard deviation criterion for both of them. In the first approach, the coefficient of the standard deviation was set to 6 and then a median filter was applied to remove the small regions, as shown in Figures 4.a and 4.b. As it can be seen, some parts of plane have not been correctly detected in Figure 4.b after the filtering of the segmented image. In the second approach the previous steps were followed but in this case with a value of $X=4$ in (1) and the results are depicted in Figure 4.c and Figure 4.d. The improvement proposed was to map the regions onto the original image so that the mean of each region is stored as a new RGB image and then it has been changed to the HSV space. The "Hue" value was integrated and segmented again using a single image plane and the threshold criterion with $X = 22$ in (1). In summary, in this approach the first image is segmented using the standard deviation of RGB levels and then, the segmented image is segmented once again using only the hue value with the thresholding method. The results can be seen in Fig. 4.e and the final result in Fig. 4.f after the mean filter. It should be noted that the computation time of the first approach is 7.019 second while the second approach takes 7.22 seconds. Comparing Fig. 4.b and Figure 4.d it can be observed that decreasing the coefficient of standard deviation leads to over segmentation as more pixels will satisfy the aggregation criteria. This, in combination with a large median filter, can deteriorate the performance and lead to the emergence of small regions.

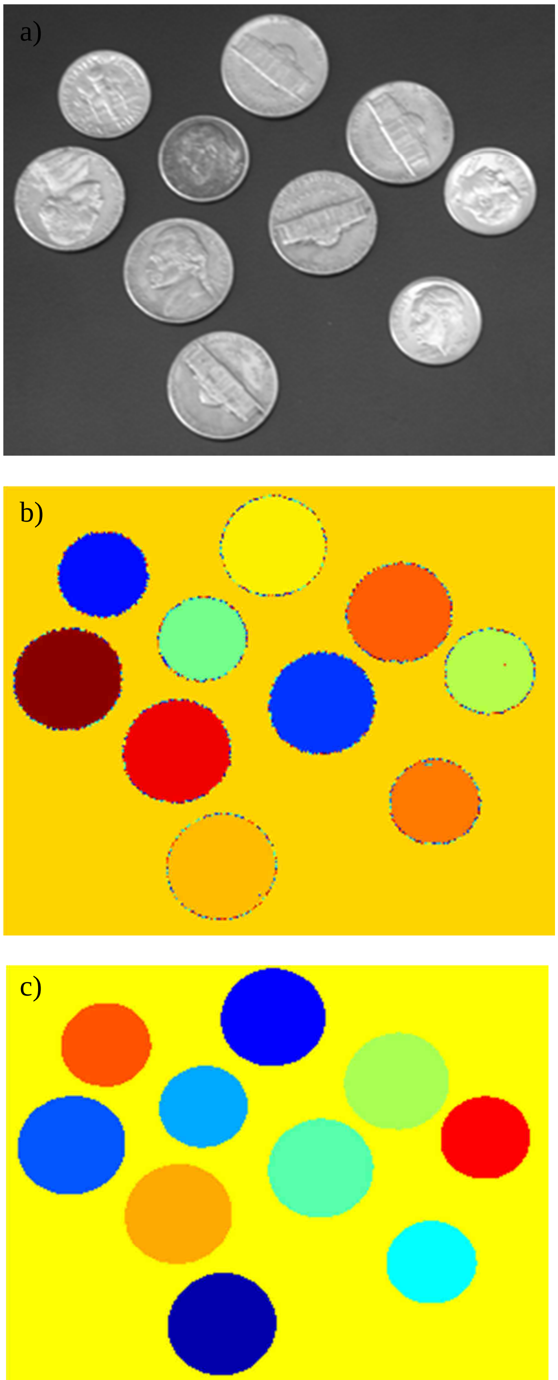


Figure 2. a) Original Image, b) segmented image using grayscale, c) image after applying a median filtering.

The same problem can be observed if an inappropriate threshold is used in the threshold criteria method. Figure 4.f shows the best performance for this image, thus it is used for comparing with the ground truth for finding an evaluation measure (i.e. sensitivity and specificity) as in Figure 3.

Figure 6 and Figure 8 show the result of segmentation for two other test images and, as it can be observed, as we increase the coefficient of the standard deviation *X*, the performance of the segmentation improves. In both cases the last images are the result of the 2 sequential segmentations. First, the segmentation in the RGB planes and then using the result for segmenting using the Hue value from the HSV color space. This merges some regions which have same color but different illumination, thus different RGB value. It can be observed that for Figure 6, the horses are connected as a single region. This is due to the fact that the color of the horses is similar and they are connected with several pixels, so this result can be expected.

A confusion matrix is formed as below, to calculate the evaluation criteria. The first row shows the region labeled in the ground truth and the first column shows the label in the result. As we know the color labels of the segments in both ground truth and result image we can use Table 2 to calculate the required statistics and derive the sensitivity and specificity.

Table II. Confusion matrix for Plane Image

	Object (green)	Background (blue)
Object (green)	True Positive (TP)	False Positive(FP)
Background(blue)	False Negative (FN)	True Negative (TN)

It should be noted that the confusion matrix is well suited for 2 class problems, then, before computing it we should binarize the images into 2 classes. For example, in figure 7 the segments are the horses, the ground and the sky, so, for computing the confusion matrix we binarize the image in such way that we have only horses and background.

TABLE III. PSUDO CODE FOR SEGMENTATION EVALUATION

For every pixel *I(x,y)* in the result image and T(x,y) in Ground truth image.

1. If I(x,y)=green & T(x,y)=green then TP++;
2. If I(x,y)=green & T(x,y)=blue then FP++;
3. If I(x,y)=blue & T(x,y)=green then FN++;
4. If I(x,y)=blue & T(x,y)=blue then TN++;
5. End.
6. Sensitivity $=TP/TP+FN$
7. Sensitivity $=TN/TN+FP$

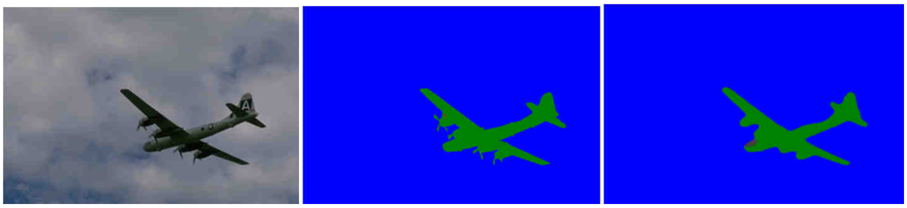
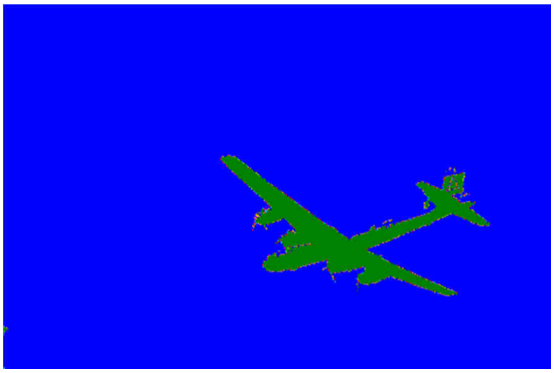


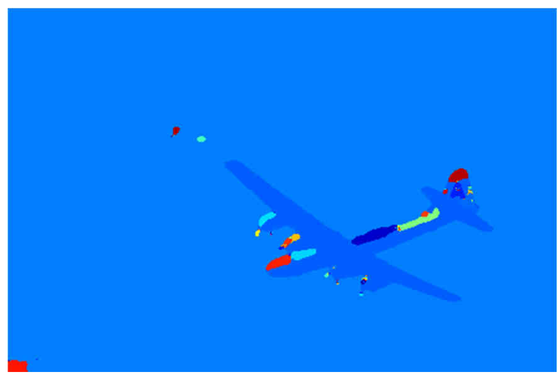
Figure 3. Evaluation of the segmentation method: a) original image, b) ground truth, c) result of segmentation



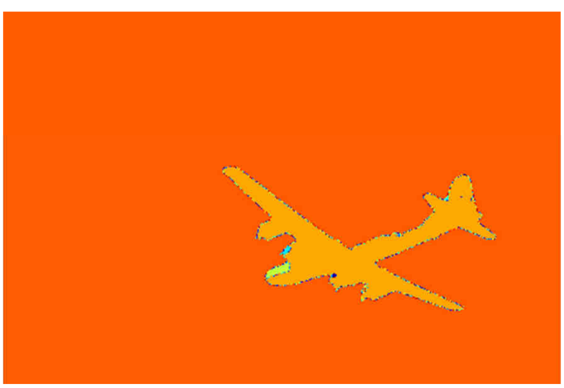
a) Std Coeff (X in eq(1))=6

b) Std Coeff =6 and a median filter of 5x5

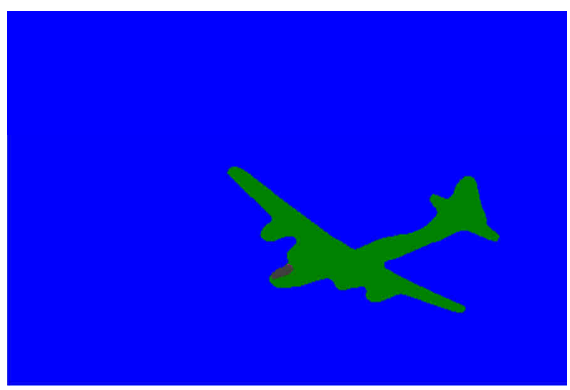
c)Std Coeff (X) = 4



d) Std Coeff (X) = 4 and a median filter of 5x5



e) Hue of d), segmented with threshold = 22



f) Image e) after a median filter of size 9x9

Figure 4. Results of segmentation for “im3096.jpg” image using various parameters.

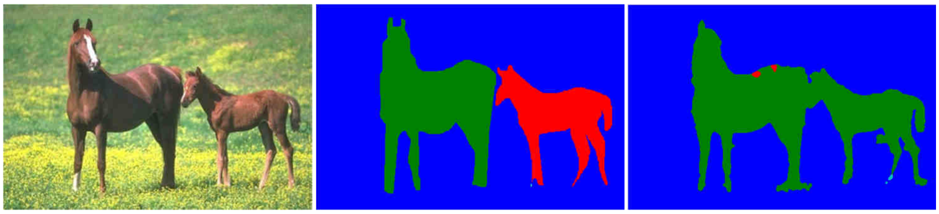
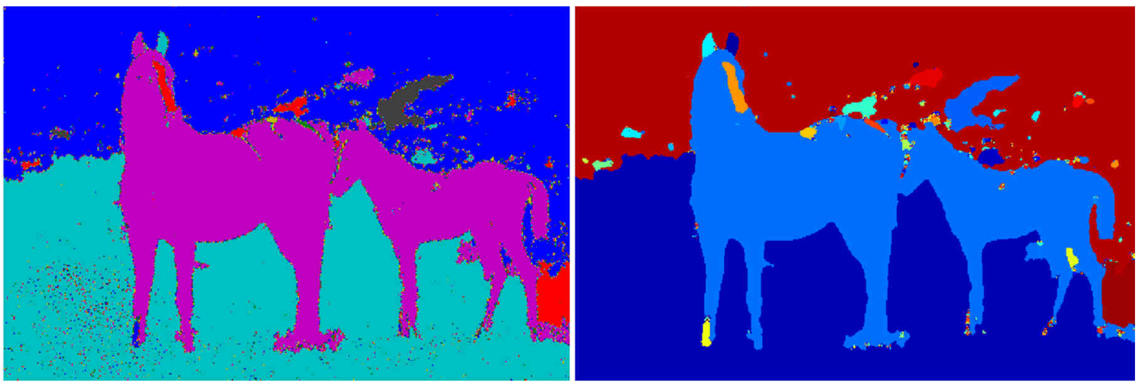
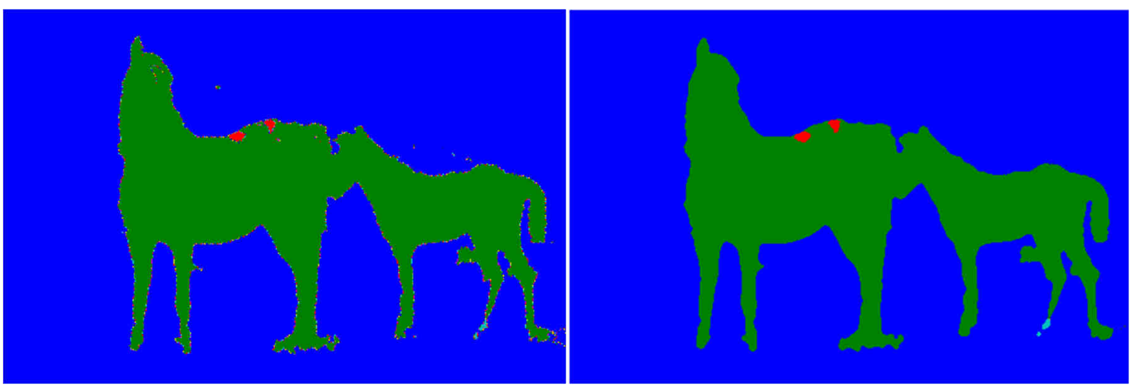


Figure 5. Evaluation of the segmentation method : a) original image, b) ground Truth, c) result of segmentation.



a) std coeff (X) =3

b) atd coeff (X) =3, and median filter 5x5



c) Hue of b) in RGB segmented with threshold = 22

d) Segmented image c)with median filter size 5x5

Figure 6. Results of segmentation for image “im113016.jpg” using various parameters (14.52 seconds)

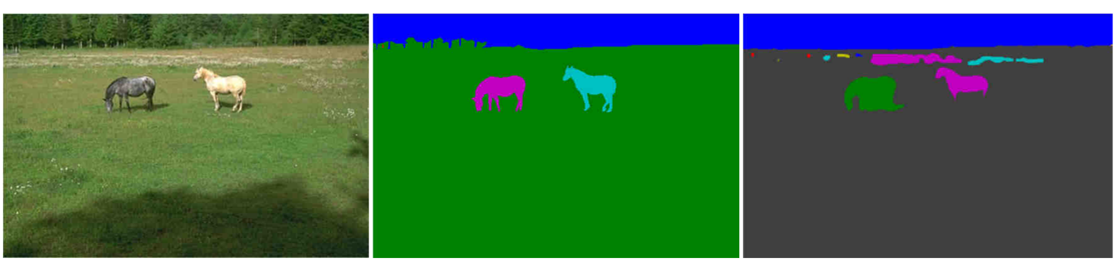
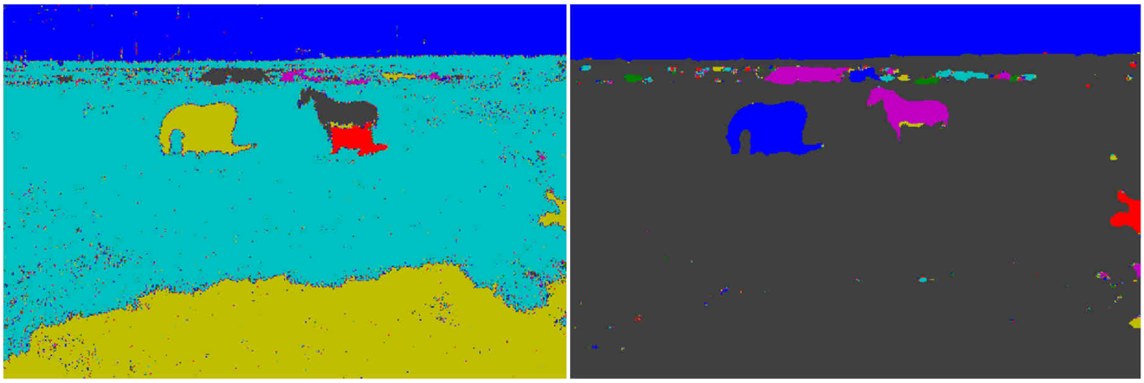
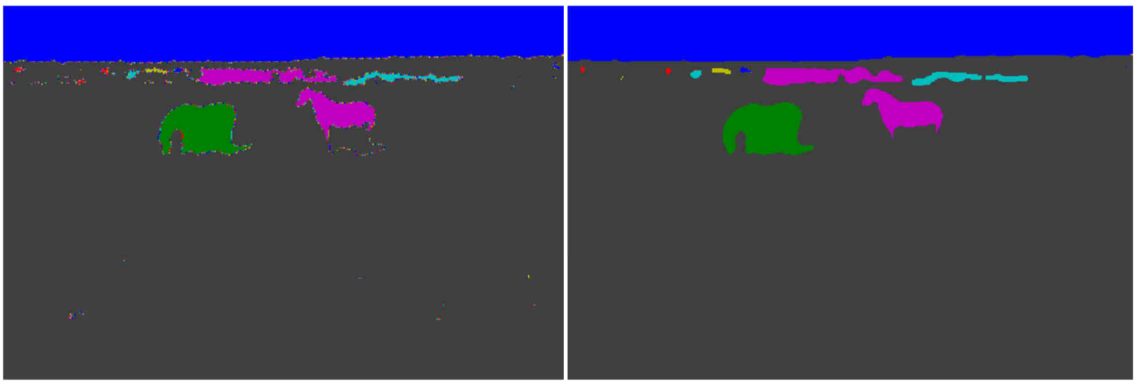


Figure 7. Evaluation of the segmentation method : a) original image, b) ground Truth, c) result of segmentation.



a) *std coeff (X) =3* b) *std coeff (X) =3 and median filter of size 5x5*



c) *Hue component of b) segmented with X=6 in (2)* d) *Segmented Image c)with median filter of 7x7*

Figure 8. Results of segmentation for the image "im28075.jpg" using various parameters (15.23 seconds)

In case the background segment or the segmented object in either the result or the ground truth image consist of several colors, Table II and Table III will change to accommodate such changes. For example, in the case of the last test example, the ground truth has 2 colors for the background and 2 colors for the objects (horses) according to the ground truth image and the final result image. The confusion matrix will be as below:

Table IV. Confusion matrix for Plane Image

	Object (purple) or(light blue)	Background (dark blue) or (green)
Object (green) or (purple)	True Positive (TP)	False Positive(FP)
Background(gray) or(blue)or(light blue) or(red) or yellow	False Negative (FN)	True Negative (TN)

Following the confusion matrix for each test case the sensitivity and the specificity of each image can be calculated as below.

Table V. Table of evaluation criteria for test images

	Sensitivity	specificity
im3096.jpg	0.9205	0.9988
im113016.jpg	0.6984	0.9799
im28075.jpg	0.7280	0.9873

One problem of the proposed method was that it led to the emergence of single pixel regions in all the test cases, which was resolved by the usage of a median filter. However, the use of the median filter changes the position of the edges with respect to the actual edge. The reason for the emergence of single pixel regions is that the contrast between these pixels and the neighbor pixel is high. For this reason we have made a change into the code such that if the region only contains one single pixel it is aggregated into the previous region. Figure 9 shows the difference of the result of our conventional method and the result of the method after the change. In all the cases shown in Figure 9, the parameters (e.g. aggregation criteria, threshold, standard coefficient) are the same. It is noticeable that all the single pixel regions have been removed.

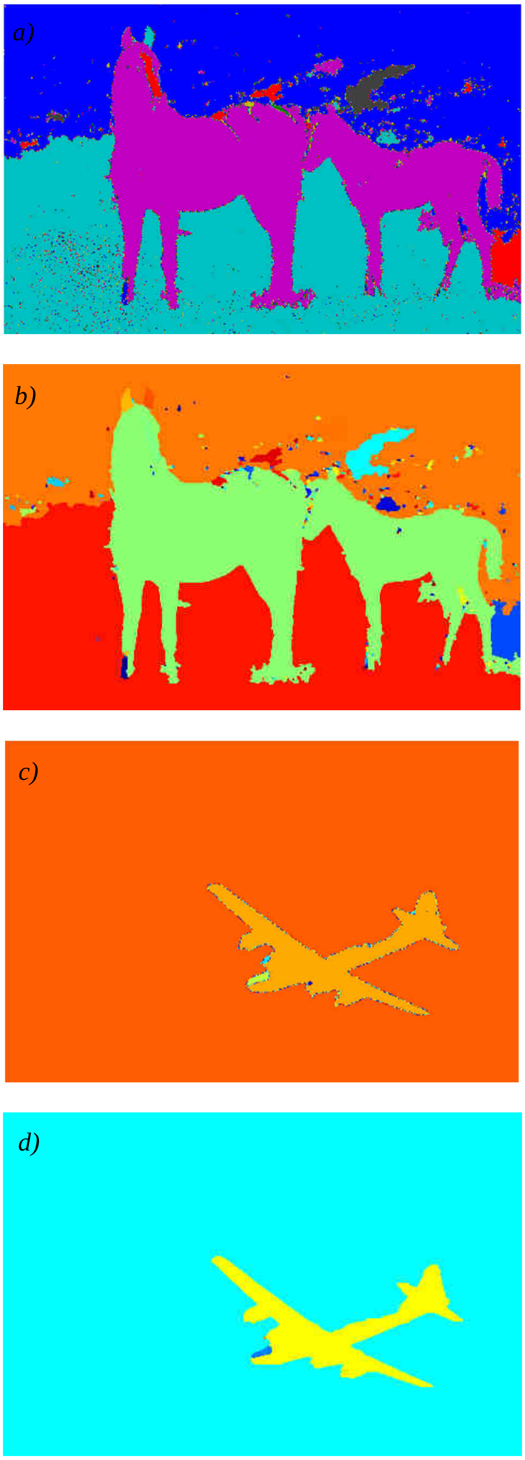


Figure 9. a) and c) conventional method with single pixel regions. b) and d) after changing the method.

V. CONCLUSION

To sum up the experiment we can say that region growing is not a very strong and effective tool for segmentation due to the fact that it is based on first order statistics of the region which may not be representative of the region. Using neighboring pixel for aggregation criteria instead of mean and std as the region model may cause over segmentation. Further, the result is dependent on the seed point placement. Moreover, it is difficult to design an aggregation criterion which can accommodate the needs of some segmentation applications. Further region growing is not an appropriate tool for texture segmentation and will not perform well when the segment does not have a uniform color /intensity distribution.

VI. REFERENCES

[1] R. C. Gonzales and E. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1992.

[2] Bigun, J., *Vision with Direction A Systematic Introduction to Image Processing and Computer Vision*. Springer-Verlag Berlin Heidelberg 2006

[3] E. Zehavi , J.K. Wolf, On runlength codes, IEEE *Transactions on Information* Theory**,** Vol 34 , Is 1, 1988, NJ, USA

[4] J. Bruce, T. Balch, M. Veloso, “Fast and inexpensive Color Image Segmentation for Interactive Robots”, Proceedings of the *IEEE International conference on Intelligent robots and System*., 2000

[5] Y.J.Zhang, A survey on evaluation methods for image segmentation, in *Pattern Recognition*, Vol.29, 1996.

[6] N.R Pala, S.K. Pal, “A review on Image Segmentation Techniques”, in *Pattern Recognition*, Vol.26, Issue 9, 1993.