



Implementation of Image Segmentation Algorithms in Digital Image Processing using MATLAB

G Sumanth Prasad
Associate Professor
Department of Electronics and Communication Engineering
GDMM College of Engineering
Nandigama, Andrapradesh, India
sumanthgdmm@gmail.com

ABSTRACT

Image segmentation has emerged as an important phase in image based applications. segmentation is the process of partitioning a digital image into multiple regions and extracting a meaningful region known as the region of interest to stop regions of interest vary application to stop segmentation of region of interest in the real world images is the first major hurdle for effective implementation of image processing applications as the segmentation process is often difficult. Hence the success or failure of the extraction of ROI, nothing but region of interest, ultimately influences the success of image processing applications in this paper in the implementation of image segmentation process algorithms using MATLAB is presented.

Key words: Image Segmentation, Region of interest, MATLAB.

I.INTRODUCTION

Image segmentation algorithms are based on discontinuity principle for similarity principle. The idea behind the Discontinuity principle used to extract regions that differ in properties such as intensity, colour, texture coming or any other image statistics. Mostly, everything changes in intensity among the regions resulting in extraction of Ages. The idea behind the similarity principle is to group pictures based on common property, to extract a region to stop.

II.CLASSIFICATION OF IMAGE SEGMENTATION ALGORITHMS

There are different ways of classifying the segmentation algorithm figure 1 illustrates the ways. one way is to classify the organisms based on user interaction required for extracting the ROI. Another way is to classify them based on the pixel relationships. Based on user interaction, the segmentation algorithms can be classified into the following three categories. Those are manual, semi-automatic, and automatic.

Robots algorithm and Method can be used interchangeably. In the manual method, the object of interest is observed by experts traces its boundaries as well, with the help of software. Hence, the decisions related to segmentation are made by human observers. Many software systems assist experts in tracing the boundaries and extracting them. By using the software systems, the experts outline the object stop the outline can be either an open or closed contour. Some software systems provide additional Help by connecting the open tracing automatically to give a closed region. Disclosed outlines are then converted into a series of control points. These control points are then connected by spline. The advantage of the control points is that even if there displacement, the software systems ensure that they are always connected. Finally, the software provides help to the user in extracting the closed religions.



Boundary retracing is a subject to process and hence variations in exist among opinions of different experts in the field, leading to the problems in reproducing the same results. In addition, a manual method of extraction is time consuming, highly subjective, prone to human error and has poor intra-observer reproducibility. However, manual methods are still used commonly by experts to verify and validate the result of automatic segmentation algorithms

Automatic segmentation algorithms are a preferred choice as they segment the structures of the objects without any human intervention. They are preferred if the task needs to be carried out for a large number of images.

Semi-automatic algorithms are a combination of automatic and manual algorithms. In semi-automatic algorithms human intervention is required in the initial stages. Normally, an observer is supposed to provide the initial speed points indicating the ROI. Then the extraction process is carried out automatically as dictated by the logic of the segmentation algorithm. Region growing techniques are algorithms where the initial seeds are given by the human observer in the region that needs to be segmented. However, the program process is automatic. These algorithms can be called assisted manual segmentation algorithms.

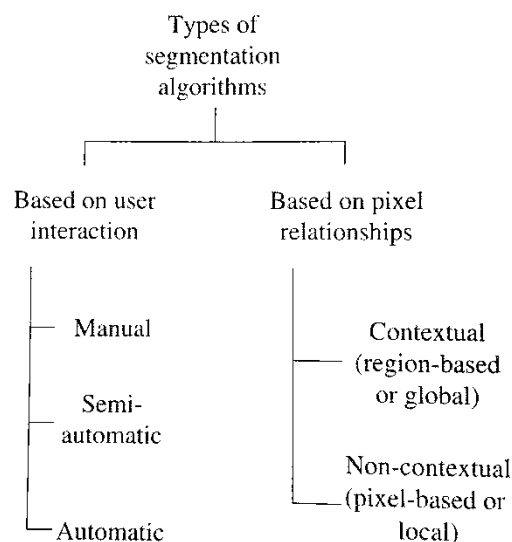


Figure 1: Classification of Segmentation Algorithms

III.CONTEXTUAL AND NON CONTEXTUAL ALGORITHMS

Another way of classifying the segmentation algorithms is to use the Criterion of the pixel similarity relationships with the neighbouring pixels. The similarity relationships can be based on colour texture, brightness, or any other image statistics. On this basis segmentation algorithms can be classified as contextual algorithms and non-contextual algorithms.

Contractual algorithms group pixels together based on common properties by exploiting the relationships that exist among the pixels to stop these are also known as region-based are global algorithm. In region based algorithms, the pixels are grouped based on some sort of similarity that exists between them. Non contextual algorithms are also known as pixel based or local algorithms. These algorithms ignore the relationship that exists between the pixels or features. Kama the idea is to identify the difficulties that are



present in the image such as isolated lines and edges. These are then simply grouped into a region based on some global level property. Intensity based thresholding is a good example of this method.

IV. IMAGE SEGMENTATION ALGORITHM

Here MATLAB supports the Otsu algorithm. A simple thresholding can be implemented using the commands for doing that image segmentation. Adaptive thresholding can be used segment images having bad illumination full stop the threshold for adaptive algorithms can be it mean or contrast or median.

ALGORITHM:

```
clc;
close all;
clear all;
a = imread('grayflower256.jpg');
a = rgb2gray(a);
subplot(3,3,1);
imshow(a); title('Original Image');
level = 0.3;
subplot(3,3,2);
segimage1 = im2bw(a,level);
imshow(segimage1); title('Simple Thresholding at 0.3');
subplot(3,3,3);
imshow(a > 153); title('Simple Thresholding at 0.6');
tmp = a;
[m n]= find(a<26);
for j = 1: length(m)
tmp(m(j),n(j))=0;
end
[m n]= find(a>26 & a <= 230);
for j = 1: length(m)
tmp(m(j),n(j))=0.8;
end
[m n]= find(a>230);
for j = 1: length(m)
tmp(m(j),n(j))=0;
end
subplot(3,3,4);
segimage2 = im2bw(tmp,0);
imshow(segimage2); title('Multiple threshoding(Between 26-230)');
level = graythresh(a);
subplot(3,3,5);
segimage = im2bw(a,level);
imshow(segimage); title('Otsu - Optimal Segmented Image');
b = imread('bluredtxt.jpg');
subplot(3,3,6);
imshow(b); title('Badly illuminated Image');
level = graythresh(b);
subplot(3,3,7);
segimage = im2bw(b,level);
imshow(segimage); title('Otsu - Segmentation for bad illuminated Image');
b = imread('bluredtxt.jpg');
b = rgb2gray(b);
avgfilt = ones(13,13);
```



```
adaptfiltmask = avgfilt/sum(avgfilt);  
im = imfilter(b,adaptfiltmask,'replicate');  
im1 = medfilt2(b,[20 20]);  
thresh = im+18;  
adaptthreshimg = b - thresh;  
subplot(3,3,8);  
imshow(adaptthreshimg > 0);  
thresh1 = im1 + 2;  
adaptthreshimg = b - thresh1;  
subplot(3,3,9);  
imshow(adaptthreshimg > 0);
```

V. RESULTS

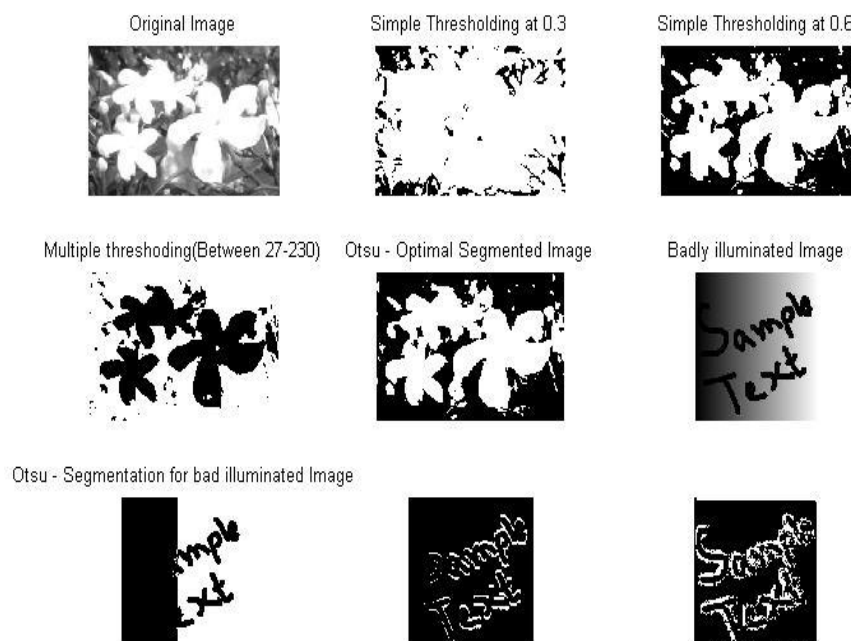


Figure 2: Implementation of Image Segmentation Algorithm

VI. CONCLUSION

It can be observed that finding the ideal threshold value of an image is a difficult exercise. The threshold images of different pressures can be observed and this indicates the difficulty in finding the threshold values. multiple algorithms can be implemented easily by specifying the threshold condition as per the requirements. it can be observed that adaptive thresholding is better than simple thresholding. In the result, it can be seen that adaptive thresholding can retrieve the contents that are not and are covered by the simple thresholding algorithms in badly illuminated images.



REFERENCES:

- [1] "Image Processing analysis and Machine Vision" by Milan Sonka and Vaclav Hlavac and Roger Boyle
- [2] "Fundamentals of Digital Image Processing " by Anil K Jain
- [3] "Digital Image Processing using MATLAB " by Jayaraman