IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Image Segmentation By Threshold-Based Segmentation Method using Otsu Algorithm

[1]Girish S. Katkar, [2]Pooja C. Ukey

[1]Assistant Professor, [2]Research Scholar

[1]Department of Computer Science,

[1]Taywade College, Mahadula-Koradi, Nagpur(M.S.), India

Abstract: To understand images and extract useful information from images to finish some works is an important part in digital image technology and for understanding the image, image segmentation is a first step. Image segmentation has an important part in image processing. Segmentation is the process of partitioning a digital image into multiple regions. Extraction of a meaningful region known as the region of interest. Region of interest used to avoid the processing of irrelevant image points and speed up the processing. Threshold-Based Segmentation method divides pixels based on their intensity relative to a given threshold. Thresholding algorithm suitable for segmenting with higher intensity. Hence the success of extraction of ROI of the image processing applications, in this paper image segmentation process algorithm i.e. Otsu algorithm using MATLAB is presented.

Index Terms- Image Segmentation, MATLAB, Region of Interest, Threshold

I. INTRODUCTION

II. An image contains a large amount of useful information. In digital image technology, for understanding the image, image segmentation is a first step. Segmentation in image processing is a crucial technique for partitioning an image meaningful segments or regions. There are two ways of classifying the segmentation algorithm. One way is the user interaction required for extracting the ROI. Another way is pixel relationships. Based on user interaction, it has three categories: manual, semi-automatic and automatic. A manual method of extraction is highly subjective, time consuming and poor intra-observer reproducibility. Manual methods are commonly used for verifying by experts check the result of automatic segmentation algorithms. Automatic segmentation algorithms segment the structures of the objects without any human interruption. They used for a large number of images. Semi-automatic algorithms are a combination of manual and automatic algorithms. In the initial stages of semi-automatic algorithms human interruption is required. Another way is to use criterion of pixel similarity with neighboring pixels. This is classified as a contextual algorithm and non-contextual algorithm. Contexual algorithm are grouping pixels together on common properties. Non-contexual algorithm ignore relationship between existing pixels. Difficulties occur in the present image such as isolated lines and edges. These are grouped into a region based on global level. Intensity based thresholding is a good example of this method.

Image Segmentation Techniques

1)Edge-Based Segmentation

Edge-Based Segmentation identifies the edges objects in the given images. It helps to find features of related objects in the image. Edge detection helps strip images of redundant data, reducing size. It identifies edges based on texture, color, contrast and saturation. Segmentation accurately represents the borders of objects in a given image.

2)Region-Based Segmentation

Region-Based Segmentation involves distributing an image into regions. Each region is group of pixels. The algorithm is located by seed point. If the algorithm finds the seed points, it can enlarge regions by adding more pixels and merging all pixels with other points.

3) Cluster-Based Segmentation

Clustering algorithms are help to identify hidden information in images. Clustering algorithm are unsupervised classification algorithm. They extend human vision by isolating shading clusters and structures. This algorithm divides images in clusters of pixels and slipting data elements, grouped that similar element into clusters.

4)Watershed Segmentation

Watershed Segmentation used grayscale images. They deal with images like topographic maps, with pixel brightness, and figure out height. It separates images into multiple areas based on pixel height and grouping pixels with the same gray value.

5) Threshold-Based Segmentation

Threshold-Based Segmentation method divides pixels based on their intensity relative to a given threshold. Thresholding algorithm suitable for segmenting with higher intensity. Thresholding Segmentation algorithm is one of the most commonly used in Image Segmentation. It automatically determines the optimal threshold according to reliable criterion, and uses these pixels according to gray level to accomplish clustering.

Image Segmentation Algorithm

Here In MATLAB Otsu algorithm is used. Otsu selects a globally optimal threshold by maximizing the variance between classes. A simple thresholding can be executed using commands for image segmentation. The calculation of the threshold method is simple and speed of operation is faster. Adaptive threshold can be mean of median and can be used to segment images that are having bad illumination.

Algorithm:

1. Select an initial estimate for the global threshold T.
2. Segment the image using T. This will produce two groups of pixels: G_1. consisting of all pixels with intensity values greater than T and G_2,consisting of pixels with values less than or equal to T.
3. Compute the average intensity values m_1 and m_2 for the pixels in regions G_1 and G_2,respectively.
4. Compute a new threshold value:

$T=1/2(m_{1+} m_2)$

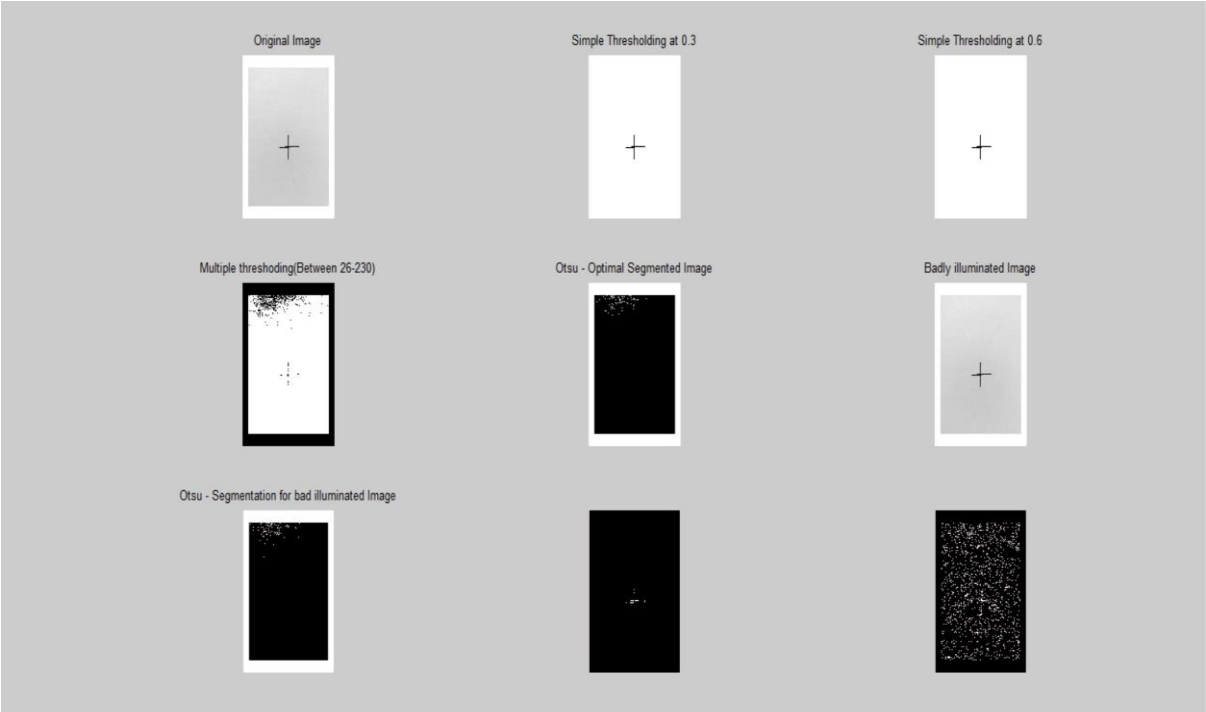
5. Repeat steps 2 through 4 until the difference in T in successive iterations is smaller than a predefined value, ΔT.
6. Segment the image using function im2bw:

g=im2bw (f, T/den)

MATLAB Code:

```
clc;
close all;
clear all;
a = imread('Step 1.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('Step 1.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('Step 1.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);
```



Conclusion:

It is observed that the ideal threshold value of an image is difficult to find. Different pressures of threshold images can be observed and this points out difficulty for finding threshold values. It is observed that adaptive thresholding is more effective than simple thresholding. Multiple algorithms executed by describing threshold conditions as per needed. In the result, it is observed that adaptive thresholding can recover the contents which are not covered by simple thresholding algorithms in badly illuminated images.

References:

[1] S. Yuheng, Y. Hao, "Image Segmentation Algorithms Overview", SiChuan University, SiChun, ChengDu,Vol.1.
[2]"Digital Image Processing Using MATLAB" by Rafael C. Gonzalez.
[3]G.Ijemaru,A.Nwajana, "Image processing system using MATLAB-based analytics",Bulletin of Electrical and Informatics,Vol.10, No.5,October 2021,pp.2566-2577, ISSN:2302-9285.
[4]"Fundamental of Digital Image Processing" by Anil K Jain.
[5]S.Prabhu, J.Gnanasekar, "A Study on Image Segmentation Method for Image Processing",Vol.1.