International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 11, November-2014

Edge Detection Based on Otsu Method and Stentiford Algorithm

P Anil Kumar
Asst. professor, ECE department
JCET - Nellore
Nellore,India

P V Vinaya Teja
ECE department
JCET - Nellore
Nellore,India

Abstract: **This paper describes an approach for edge detection wherein otsu thresholding and stentiford algorithm has its vital play. Thresholding sets or determines the intensity separation based on intensity margin set by the user requirement. In order to simplify the margin determination Otsu thresholding is bought to play. Now our determination is to markout the boundaries of the object so the simple sentiford algorithmis utilized. Edge detection is rigorously used in face detection, object recognition during night etc.,**

Keywords :- Edge detection, Thinning, Thresholding, Otsu, Stentiford.

I. INTRODUCTION

An image is an array, or matrix, or square pixels (picture elements) arranged in columns and rows.In imaging science, image processing is simply a kind ofsignal processing wherein input is an image, the output of processing may be either an image or parameters or a set of characteristics related to the image. Image-processing techniques treats image as a 2-D signal [1]and apply standard signal-processing techniques to it. Digital image processing being a division of digital signal processing has many sophisticated advantages over analog image processing,allows wide range of algorithms to apply to the input data, to avoid problems such as signal distortion and build-up of noise during processing with the help of computer algorithms. Edge detection has its vital utilizations in feature extraction and pattern recognition of objects. Here, I will emphasize on the *Otsu's method*and *Stentiford Thinning Algorithm*. Most images contain some amount of redundancies that can sometimes be removed when edges are detected and replaced during reconstructionby thinning. As edges often occur representing objectboundaries, edge detection is extensively used in image segmentation when images are divided into areas corresponding to different objects.

II. APPROACH

The main motto of this paper is to skeletonizing the object. This can be simply obtained by following the below mentioned approach:

A. Capture Image
B. Gaussian Filter
C. Image Gradient
D. Otsu Thresholding
E. Stentiford Thinning Algorithm

In general images are available with many constrains such as noise, intensity levels, blur etc. So the image is pre-processed to meet certain requirements for smooth and accurate edge detection.

A. Capture Image

The image to be processed is captured from camera and invoke to matlab in jpeg, png, bmp, tif etc., formats. The image can also be resized to meet the specifications of requirement. Source images obtained are generally colourbut we know grayscale images are very common and sufficient for the task discussed in this paper. Thereby colour image processing complications are facilitated. Grayscale image signal only carries the information about the intensity. Images of this kind are known as black and white, are composed exclusively of shades of gray, varying from weakest (black) to strongest (white) intensity. Often grayscale possess 256 possible different shades of gray from black to white.

B. Gaussian Filter

Gaussian filtering is used to blur images and remove noise and detail. It is similar to the mean filter, but it uses different kernel that represents the shape of Gaussian (bell-shaped) hump. As working with images the two dimensional Gaussian function is given as Shown graphical representation of 2D Gaussian distribution with mean (0,0) and σ = 1 is shown in above figure 1.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad [2]$$

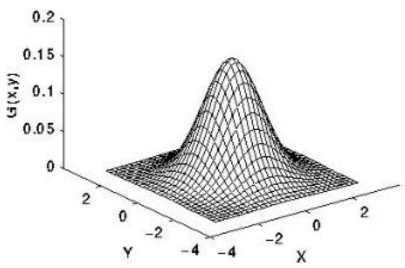

Figure 1 Gaussian distribution

Where σ is the standard deviation of the distribution.

C. Image Gradient

An image gradient is a directional change in the intensity of an image. It may also be used to extract information from images. Also knows as colour progression as it possess even gradiation from low to high values. Image or Pixel direction resembles the direction of the gradient. The direction of maxima change in the image is given by

$$\nabla f = \frac{\partial f}{\partial x}\hat{x} + \frac{\partial f}{\partial y}\hat{y}$$

Gradient can also be applied to RGB image and the direction and orientation of individual colour can be found.

D. OtsuThresholding

Thresholding starts with setting or determining gray value [3]. Otsu method is used to automatically perform cluster-based thresholding[4]. Otsu method is a kind of image segmentation that involves iteration through all the possiblevalues of threshold and calculating a measure of spread for the pixel levels on each side of the threshold, i.e. the pixel either fall in background or foreground. The aim is to find the minimum spread from the sum of foreground and background.

It is based on very simple idea of finding threshold that minimizes the weighted with in class variance, this turns out to be same as maximizing between class variance. As it is operating directly on grayscale image [p(i)], it's fast. A bimodal (histogram) which consists of the values of both object and background (multi modal). Otsu method holds well in case of images that have a uniform distribution (figure 3) of object and background intensities in two different moulds. On the other hand in non-uniform distribution(figure-2) perfect segmentation of object and back-ground intensities into two different moulds in order to fix the threshold value is not appropriate. In case of noisy image also it may not be possible for a better segmentation. But we have many algorithms to de-noise the noisy image to obtain bimodal distribution.

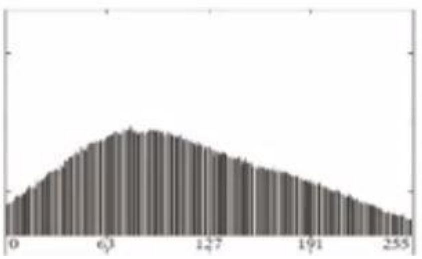

Figure 2 Non-uniform distribution

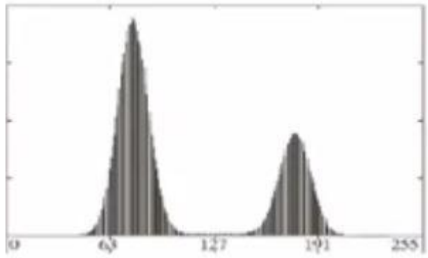

Figure 3 Uniform distribution

Method Illustration

If we observe figure 3, the threshold is going to be applied across, such that the in-class viability is very small, so the threshold is applied at '~127' such that the variance in each one of the classes is as small as possible so that the class is as compact as possible.

We need to minimize the weighted within-class variance

$$\sigma_\omega^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

Where,t is the threshold lies somewhere between 0-255

The class probabilities are estimated as:

$$q_1(t) = \sum_{i=1}^{t} P(i) \quad q_2(t) = \sum_{i=t+1}^{I} P(i)$$

And class means are given by

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \quad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}$$

Finally the individual class variances are

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)}$$

$$\sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

For a given threshold, the total variance is the summation of within-class and between class variance, which is the sum of weighted squared distances between class and grade mean.

$$\sigma^2 = \underbrace{\sigma_\omega^2(t)}_{} + \underbrace{q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2}_{}$$

Within-classBetween-class $\sigma_B^2(t)$

Since the total is constant and is independent of t, the cause of changing the threshold is merely moving the contribution of two terms back and forth. Between-class can be computed very efficiently in a recursive fashion from 0-255. The difference is that within-class cannot be done explicitly in a recursive manner but between-class can be done explicitly in a recursive manner where (t+1) depends on t. So, minimizing the within-class variance is same as maximizing the between-class variance.

E. Stentiford Thinning Algorithm

Now, we are at the final stage of this paper with a threshold image in our hand. To obtain reliable image boundaries we indeed have to perform thinning operation. Thinning is a morphological operation used for the removal of selected foreground pixels from binary image. It preserves the topology of the original image region while throwing away most of the original foreground pixels. The thinning output is mentioned in the following figure 4.

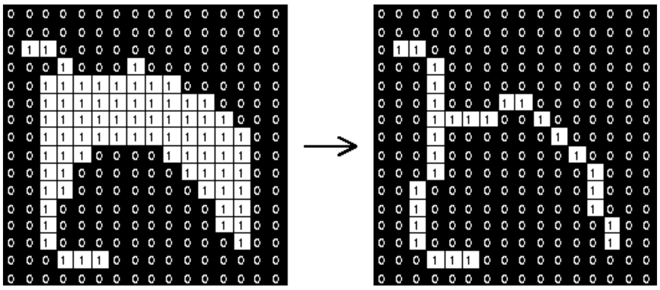

Figure 4 Thinning algorithm

Thinning is somewhat similar to erosion or opening utilized for skeletonizing and medial axis transform. Here, I emphasize with Stentiford thinning algorithm.

Method Illustration

This algorithm uses a set of four templates to scan the image, say T1, T2, T3, and T4 as shown in figure 5.

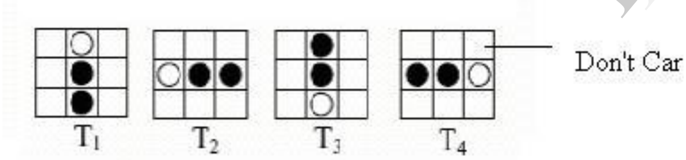

Figure 5: Templates of Stentiford algorithm

The Stentiford Algorithm can be processed as stated below[5]:

1. Initially locate a pixel location (i, j) which matches to template T1. With T1 template every pixels along top side of the image are removed moving from left to right and from top to bottom.
2. If the central pixel isn't an endpoint, and has connectivity number '= 1'; then set this pixel for deletion.
3. Repeat steps 1 and 2 for all pixel locations matching T1.
4. Similarly follow the above mentioned steps 1-3 for the templates: T2, T3, and T4.
5. Template T2, T3, and T4 matches the left, bottom, and right side of the image.
6. Pixels marked for deletion are set to white.

III. EXPERIMENTAL RESULTS

Above proposed method is coded and simulated with the help of matlab from mathworks. And the results obtained on the standard test image 'Lena512' are attached below regarding various stages of the proposal.

The threshold and final result after applying otsu thresholding and stentiford algorithm for standard test image are respectively displayed below in figure 7 and figure 8 along with the original image in figure 6.

Figure 6 Lena 512

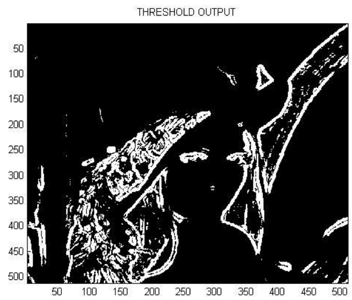

Figure 7 Threshold Output

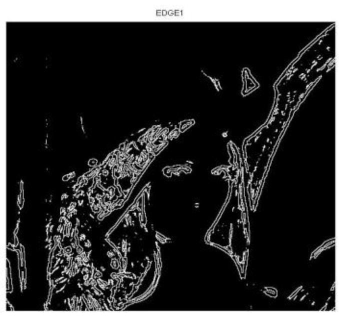

Figure 8 Stentiford output

IV. CONCLUSION

The proposed method to describe the edge detection based on otsu thresholding and stentiford algorithm. Thresholding has its key play in dividing the intensities as per the requirement during the need. Stentiford algorithm is utilized to mark out the borders of the image with the help of various templates. The results obtained after the execution are made convincing for the threshold boundary levels less than 50% of the image intensity levels.

REFERENCES

[1] R. Gonzalez and R. Woods, Digital image processing. Addison Wisley, 1992.

[2] Shapiro, L. G. & Stockman, G. C: "Computer Vision", page 137, 150. Prentice Hall, 2001

[3] Demant C., Streicher-Abel B. and Waszkewitz P. (1999). Industrial Image Processing: Visual Quality Control in Manufacturing. Springer-Verlag. p. 96. ISBN 3-540-66410-6.

[4] M. Sezgin and B. Sankur (2004). "Survey over image thresholding techniques and quantitative performance evaluation". Journal of Electronic Imaging 13 (1): 146–165.

[5] By Mohan palani at http://in.mathworks.com/matlabcentral/fileexchange/34712-stentiford-thinning-algorithm