Dynamic Thresholding Based Edge Detection

Neeta Nain, Gaurav Jindal, Ashish Garg and Anshul Jain *

Abstract—**Edges are regions of interest and edge detection is the process of determining where the boundaries of objects fall within an image. It is an important concept, both in the area of object recognition and motion tracking. This paper presents an adaptive thresholding based edge-detection method using morphological operators. The novelty of the approach is the adaptive efficient peak detection of the image histogram, thus deciding the modality of image and also the usage of morphological operations in the extraction of one pixel thick $m-$connected boundary which is continuous in a segment.**

Keywords: Peaks,Boundary, Edges, Morphological Operations.

1 Introduction

An edge [1], [2], [3], [4] is usually a step change in intensity in an image. It corresponds to the boundary between two regions or a set of points in the image where luminous intensity changes very sharply. The presence of an edge within a grayscale image indicates that there is a change in the grayscale from one region to another. The derivative of the grayscale levels within an image as a function of the (x, y) position provides a means of detecting the presence of an edge. A large number of vision applications like matching and tracking use edges and lines as primitives. The process of edge detection is usually followed(preceded) by the Thresholding of the edge detected image. The process of thresholding provides a means of separating weak edges from strong edges. Consider two adjacent pixels, if these pixels are having dissimilar gray levels, then there is a good probability that they are separated by gray level discontinuity. This is easier said than done, when dealing with digital pictures, most images are having continuous intensity variation. This leads to edges having a gradual change of pixel levels from one object to the next thus causing the edge to be less defined and harder to distinguish and therefore an optimal threshold value is very important for clearly distinguishing between distinct regions.
The common challenge of good edge detection algorithms is to have a low probability of error i.e; failing to mark edges or falsely marking non-edges with the marked points as close as possible to the center of true edge. Considering the assumption that objects in an image can be distinguished on the basis of their gray levels, image thresholding technique is being used to determine the various regions into which the image can be segmented. The number of regions into which image can be divided depends upon the number of dominant peaks [7] present in the image histogram as explained in section 3.1. After calculating the number of significant peaks, the image is thresholded using Multi-level thresholding technique and than morphological operators(gradient) are used to extract the edges as explained in section 3.3. The edges obtained may be more than one pixel thick on the boundary. They are made one-pixel thick by using the various morphological thinning masks as explained in subsection 3.4. The extracted boundary is then smoothed to fill one-pixel breaks and to remove erroneous single stray pixels as explained in section 3.5. The proposed method has been evaluated both qualitatively and quantitatively over a number of images with different intensity gradation with excellent results as shown in section 3.5. It identifies all significant edges with minimal false positives.

2 Literature Review

Some of the earliest methods of detecting edges like, Roberts [1], Prewitt [2], Sobel [8] etc in images used small convolution masks to approximate the first derivative of the image brightness function. The most popular gradient masks are the Prewitt and Sobel edge detectors. The Sobel edge filter provides good edge detection and is somewhat insensitive to noise present within the image. This is due to the averaging that is performed by this edge detector during the computation of the gradient. The most popularly used edge detector is defined by Canny Edge Detector [9]. Since the Canny edge detector is a significant and widely used contribution to edge detection techniques and it is also used for our experimental comparisons, its principles are explained in detail. Canny proposed a new approach to edge detection that is optimal for step edges corrupted with white noise. Canny's derivation of a new edge detector is based on several ideas.

- The edge detector was expressed for a $1D$ signal. A closed-form solution was found using the calculus of variations.
- The detector is then generalized to two dimensions. A step edge is given by its position, orientation, and possibly magnitude also called strength. Suppose

*Malaviya National Institute of Technology, Department of Computer Engineering, Jaipur, INDIA

ISBN:978-988-98671-9-5

G is a $2D$ Gaussian signal and assume we wish to convolve the image with an operator G_n which is the first derivative of G in the direction n.

$$G_n = \frac{\delta G}{\delta n} = n\dot{\bigtriangledown}G \tag{1}$$

The direction n should be oriented perpendicular to the edge. If f is the image, the normal to the edge n is estimated as

$$n = \frac{\bigtriangledown(G * f)}{|\bigtriangledown(G * f)|} \tag{2}$$

The edge location is then at the local maximum of the image f convolved with the operator G_n in the direction n

$$\frac{\delta}{\delta n} G_n * f \tag{3}$$

Substituting in 3 for G_n from equation 1, we get

$$\frac{\delta^2}{\delta n^2} G * f = 0 \tag{4}$$

The equation 4 illustrates how to find local maxima in the direction perpendicular to the edge; this operation is often referred to as non-maximal suppression. The edge strength (magnitude of the gradient of an image intensity function) is measured as

$$|G_n * f| = |\bigtriangledown(G * f)| \tag{5}$$

- Noise may cause spurious responses to edge detection, called as 'streaking' problem. Usually the output of the edge detector is thresholded to decide which edges are significant, and streaking means the breaking up of the edge contour caused by the operator fluctuating below and above the threshold. Streaking can be removed by thresholding with hysteresis. If the edge response is above a high threshold, those pixels detect definite edges for a particular scale. Individual weak responses usually correspond to noise, but if these pixels are connected to any of the pixels with strong responses, they are more likely to be actual edges in the image. Such connected pixels are treated as edge pixels if their response is above a low threshold. The low and high thresholds are set according to an estimated signal-to-noise ratio.

3 Image Thresholding

Thresholding [2] is the process of separating an image into different regions based upon its gray level distribution. Key to the selection of a threshold value is an image's histogram, which defines the gray level distribution of its pixels. The bimodal nature of this histogram is typical of images containing two predominant regions of two different gray levels as objects and background. When dealing with digital pictures, most images are having continuous intensity variation and if only a single threshold level is used then many important regions are lost. It becomes difficult to identify significant regions of such images having multimodal histogram. A better method of Thresholding the gray level image is thus to use multilevel Thresholding [6] instead of bi level thresholding. This is the approach that is taken in the implementation of optimal thresholding. We calculate the optimal number of threshold levels by computing the number of **significant peaks** from image's histogram as explained below.

3.1 Peak Detection Algorithm

We propose an efficient peak-finding algorithm to determine the number of prominent peaks in the histogram of the image. The number of peaks [7] represents the number of distinct regions, the image can be divided into. The number of significant peaks are computed as:

1. Compute the frequency histogram of the image. Insert gray level values and their corresponding frequencies into the sets G and F respectively. For each gray level i, put the element (i,f_i) into the set $S0$, where f_i is the frequency corresponding to the i^{th} gray level.

$$S0 = \{(i, f_i)\} \tag{6}$$

where $i \epsilon G \, and f_i \epsilon F$.

2. Compute the set $S1$, that contains the elements representing the points of local maxima in the image histogram.

$$S1 = \{(i, f_i) \mid ((f_{i-1} < f_i) \& (f_i > f_{i+1}))\} \tag{7}$$

where $(i, f_i) \epsilon S0$

3. Compute the set $S2$ comprising of elements having frequency higher than 1% of the maximum frequency f_{max} in the set $S1$. This empirical threshold value 1% is taken as the optimal value based on experiments performed over $100's$ of images having various gradual intensity changes varying from bi level to 9 levels.

$$f_{max} = max(F) \tag{8}$$

$$S2 = \{(i, f_i) \mid f_i > f_{max}\} \tag{9}$$

where $(i, f_i) \epsilon S1$

4. Remove the elements having close peaks from set $S2$. This is done by checking the difference between the gray levels of two elements in set $S2$. If the difference is less than 20, then the element with lower frequency is removed. Again 20 is an empirically determined threshold value based on experiments. Construct set $S3$ after the removal of close peaks as:

$$S3 = S2 - \{((((i, f_i)| \, i > j, \; (i-j) \leq 20) \\ \&(f_i = min(f_i, f_j))))\}$$

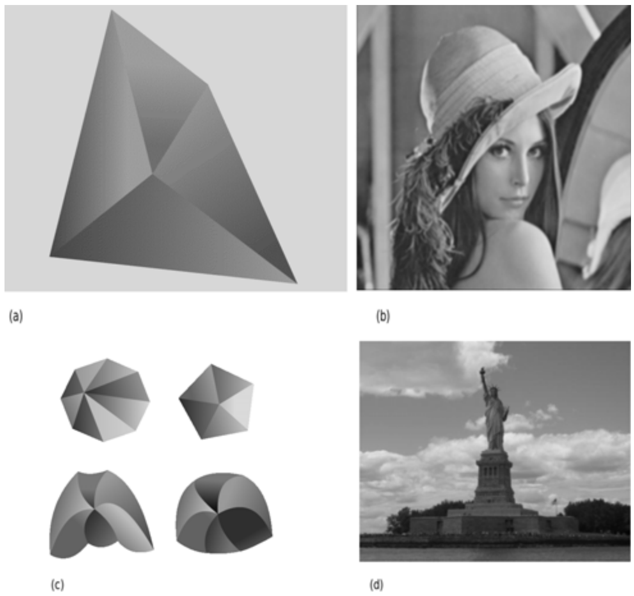

Figure 1: Test images with various gradual variations in intensity a) Object1, image having gradient within the region b)Lena, standard test image with gradual intensity change c) Object2, image having gradient within the region and across the regions d) Statue of liberty with light background

where $(i, f_i) \epsilon S2$ & $j \epsilon G$ &$f_i,\ f_j \epsilon F$

5. The number of elements in set $S3$, denoted as $|S3|$, is equal to the number of significant peaks in the histogram:

$$n = |S3| \tag{10}$$

where n is the number of prominent peaks.

The prominent peaks in image histograms for the images in figure 1 are shown in figure 2.

3.2 Multi-level Thresholding Algorithm

If the input image is a gray level image then the boundary extracted image could be in gray level. To convert the image into a binary image we threshold the image. For thresholding, we compute multilevel thresholds adaptive of local intensity variations as:

1. Compute the frequency histogram of the image. For each gray level i, the probability distribution $p_i = \frac{f_i}{N}$, where N is the total number of pixels and f_i is the frequency of i^{th} gray level .
2. For $M - level$ thresholding; Divide L, the total number of gray levels, into M classes as $C_1[1..t_1]$, $C_2[t_1..t_2]$, ..., $C_M[t_{M-1}..L]$ where the total number of thresholds $= M - 1$ and $M = n$, calculated in equation 10.

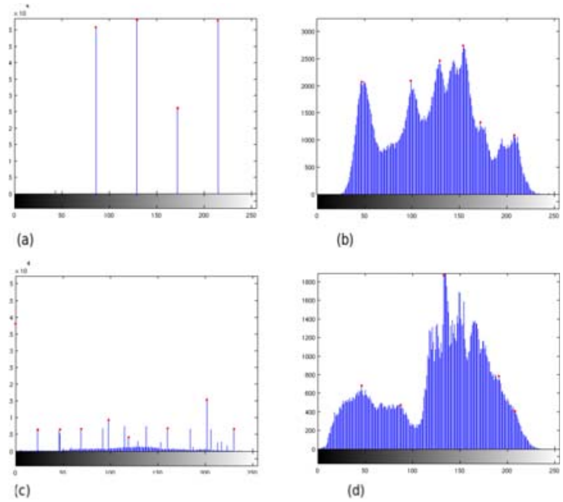

Figure 2: a) Image histogram having four prominent peaks, b) Image histogram having six prominent peaks, c) Image histogram having nine prominent peaks, d) Image histogram having five prominent peaks

3. Compute w_k and μ_k for each class k as $w_k = \sum_{i \epsilon C_k} p_i$, $\mu_k = \sum_{i \epsilon C_k} \frac{ip_i}{w_k}$
4. Compute the variance σ_B^2 as,

$$\sigma_B^2(t_1, t_2, ..., t_{M-1}) = \sum_{k=1}^{M} (w_k \mu_k^2 - \mu_T^2) \tag{11}$$

where $\mu_T = \sum_{k=1}^{M} w_k \mu_k$

5. The optimal thresholds are,

$$(t_1^*, t_2^*, ..., t_{M-1}^*) = Max(\sigma_B'^2(t_1, t_2, ..., t_{M-1})) \tag{12}$$

where $1 \leq t_1 < t_2 < ...... < t_{M-1} < L$ and $\sigma_B'^2 = \sum_{k=1}^{M} w_k \mu_k^2$

The optimal thresholds are calculated by maximizing the class separation while minimizing the areas of the classes as per the approach taken by Otsu's [5] method of adaptive thresholding to select the most optimum threshold value. If the number of threshold classes are less than the number of peaks in the frequency histogram of the image, the intricate details of gradual intensity changes which are though significant are lost. On the other hand increasing the number of threshold classes beyond the peaks increases the noise in the image. Thus the optimal number of thresholds is equal to the number of peaks in the image. The output Figures 3(a-f) shows that the thresholding done at optimal values gives the best results. Threshold levels < optimal threshold level misses significant gradual intensity variations while levels > optimal level generates noise. Figure 3(a), shows there are four peaks in the histogram. Thus, the results of bi-level

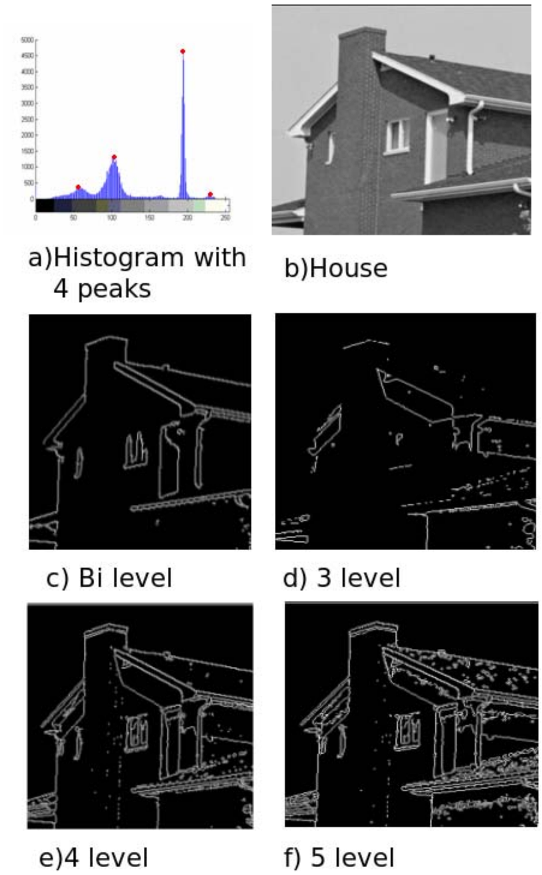

Figure 3: (a) Intensity histogram of original image having 4 peaks, (b) Original image, (c) Two threshold classes, (d) Three threshold classes, (e) Four threshold classes(optimal), (f) Five threshold classes

and 3-level thresholding in Figures 3(c),(d) does not reflect the gradual intensity changes within the house and hence the internal details of the house are missed. Figure 3(e) gives good edges when four threshold classes are used(computed using our optimal threshold method). Increasing the number of thresholds beyond the number of peaks increases the noise in the image as shown in figure 3(f).

3.3 Edge Detection

A powerful set of binary image processing operations developed from a set-theoretical approach comes under the heading of *mathematical morphology* [2]. Although the basic operations are simple, the operations and their variants can be concatenated to produce much more complex effects. In the general case, morphological image processing operates by passing a *structuring element*(StrEl) over the image in an activity similar to convolution. The structuring element can be of any size, and it can contain any complement of $1's$ and $0's$, and a -1 specifies don't care. At each pixel position a specified logical operation is performed between the structuring element and the underlying binary image. The binary result of that logical operation is stored in the output image at that pixel position. The effect created depends upon the size and content of the structuring element and upon the nature of the logical operation. All the morphological operators used here conform to their standard definition [2]. The morphological operations of erosion(conforms to gradient) is used to detect edges. The set difference between the original object and the eroded(dilated) object produces a contour that straddles the inside(outside) of the contour of the original object. The inside contour of the thresholded image or the edge detector is defined as:

$$img = img - (img \ominus StrEl) \tag{13}$$

Where the,

$$\text{StrEl}= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

can be used to erode the image and $\ominus$ is morphological erosion operation.

3.4 Boundary Thinning

The result of step 3.3 is a boundary extracted image which can be two or more pixel thick image. To convert the above two or more pixel thick edge extracted image to one pixel thick image, we search extra pixels in each direction using Hit-and-Miss transform. The set of 3 x 3 masks;

$$Mask1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\text{and } Mask2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

will search extra pixels in the horizontal direction. By rotating these masks by 90° we get the masks for vertical direction and by rotating these masks by 45° we can get the masks for diagonal direction too. $Mask1$ and $Mask2$ finds extra pixels in positive and negative directions respectively. Using these masks we compute the *Hit-and-Miss Transform* of the image and remove the pixels searched by these masks. This step performs thinning(identifies the medial axis) of the boundary and converts the two or more pixel thick boundary to one-pixel thick boundary.

3.5 Remove Noisy Stray Pixels and Fill in Pixel Breaks

The Hit-and-Miss transform may create pixel breaks in the image boundary as shown in Fig 4(a-e) and may

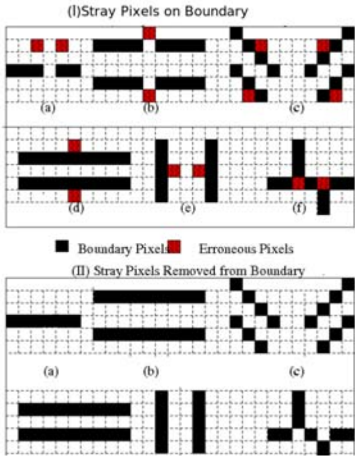

Figure 4: a) Isolated pixel and One pixel break in a straight horizontal or vertical line(one pixel thick), b) One pixel break in a straight horizontal line and there is only one pixel above the break, c) One pixel thick diagonal line with one extra pixel in 4-neighborhood of any of the pixel of this diagonal, d), e) One pixel thick straight line with one extra pixel above any of the pixels in the line, f) A "T" made by a set of pixels

also create single stray pixels. For making a continuous boundary and to reduce noise, these pixel breaks are to be filled in and the isolated noisy pixels are to be removed. To *Thin* and to *Fill gaps* in the binary image we used the set of masks shown in Figure 5(with rotation of 90° four times on each masks). The output after applying the above masks to Figure 4(I) is shown in Figure 4(II). A pixel p at coordinates (x, y) has two horizontal and two vertical neighbors whose coordinates are $(x+1, y), (x-1, y), (x, y+1) and (x, y-1)$. This set of $4-neighbors$ of p denoted as $N_4(p)$ are called 4-connected and similarly the four diagonal neighbors of p having coordinates $(x+1, y+1), (x+1, y-1), (x-1, y+1) and (x-1, y-1)$ are denoted as $N_D(p)$. The union of $N_4(p)$ and $N_D(p)$ are the $8-neighbors$ of p denoted as $N_8(p)$. If a pixel p is both 4-and 8-connected to q it introduces redundancy in path from p to q which can be avoided by converting their connectivity to m-connectivity. Two pixels p and q are m-connected when $q \in N_4(p)$ or $q \in N_D(p)$ AND $N_4(p) \bigcap N_8(q) = \emptyset$. To obtain $m-$connected boundary from the above one pixel thick boundary

–1	–1	–1
–1	–1	–1
–1	–1	–1

Mask 1

0	0	0	0	0
0	–1	–1	–1	0
1	1	–1	1	1
0	–1	–1	–1	0
0	0	0	0	0

Mask 2

0	0	0	0	0
0	–1	–1	–1	0
0	–1	1	–1	0
1	1	–1	1	0
0	–1	–1	–1	0

Mask 3

0	0	0	0	0
1	–1	–1	–1	0
0	1	1	–1	0
0	–1	1	–1	0
0	0	–1	–1	–1

Mask 4

0	0	0	0	0
0	0	–1	0	0
0	–1	1	–1	0
0	0	1	1	0
0	0	0	0	0

Mask 5

0	0	0	0	0
0	0	–1	0	0
0	–1	1	–1	0
0	1	1	0	0
0	0	0	0	0

Mask 6

Figure 5: Masks to Thin and Fill in Gaps

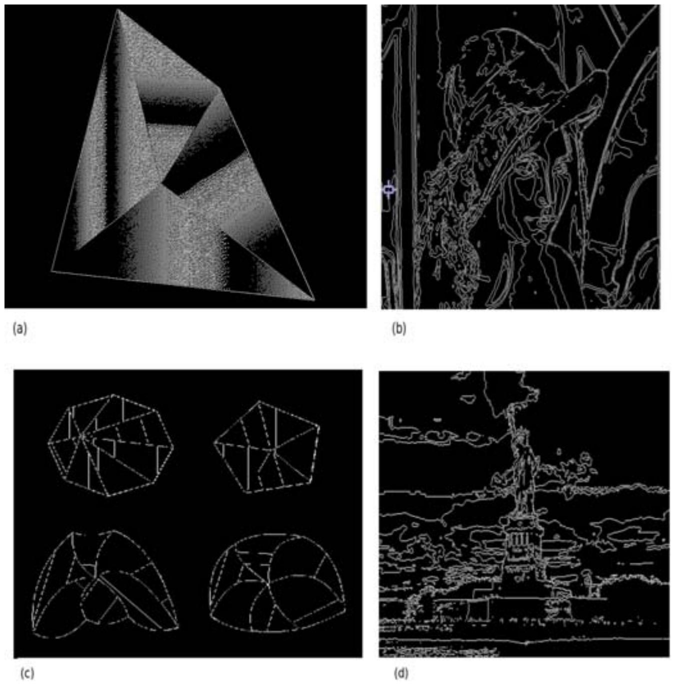

Figure 6: Edge extraction results using Proposed Boundary extractor on test images shown in Figure 1

$$Mask7 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

is used for a Hit-miss transform and the pixels searched by $Mask7$ are subtracted from the boundary as:

$$bdry = bdry - (bdry \otimes Mask7) \tag{14}$$

where $\otimes$ is Hit-Miss transform. This step removes redundancy in connectivity and the boundary pixels becomes m-connected.

4 Test Results

The results obtained using the proposed technique as shown in Figure 6 are comparable to Canny edge extraction method shown in Figure 7.The Canny edge extractor misses intricate details in the case of object1, Figure 7a) and lena images Figure 7b) that are clearly visible in the Figures 6a) and 6b) respectively. In object2 image Canny edge extractor introduces noisy pixels as shown in Figure 7c) whereas in statue image it misses internal details, Figure 7d). The noisy pixels of object2 image are

ISBN:978-988-98671-9-5

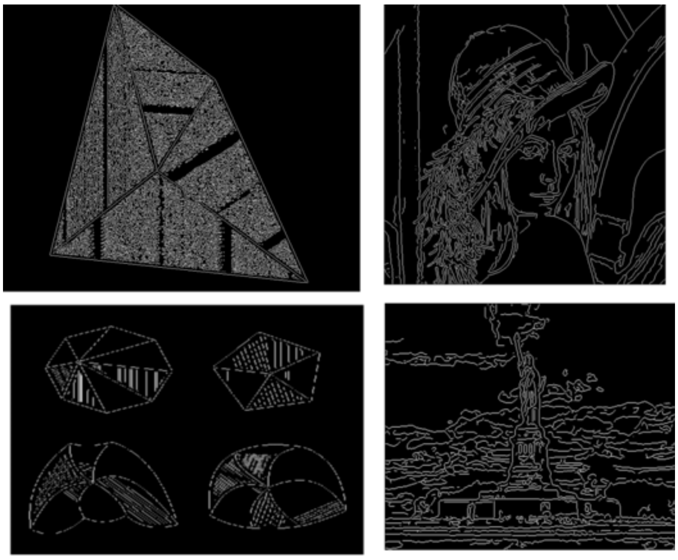

Figure 7: Edge extraction results using Canny Boundary extractor on test images shown in Figure 1

removed and internal details of statue image are detected using the proposed edge extractor, Figures 6c), 6d). The results obtained using the dynamic Multi level thresholding technique show remarkable improvement in contrast to the bi level thresholding used in most of the common edge extractors.

5 Conclusions

This paper proposes a novel and efficient method which uses morphological operations for detection of edges or boundary. All significant edges are identified as Multi-level optimal thresholds are computed. Compared to traditional convolution derivative masks for edge detection morphological operations are used which are efficient, as they are applied on the complete image while convolution masks are applied at every pixel position. Compared to existing Edge detectors like Canny [9] and Roberts, Prewitt [2] etc., our algorithm extracts precise one pixel thick seamless, continuous (in a segment) image boundary which is very important to extract prominent and significant corners [10] in images and also in computing image semantics [11]. This method works on all types of images, noise suppression is handled by Gaussian smoothing with $\sigma = 4.95$. This method can further be extended to handle all type of noises like Poisson, Speckle and Gaussian noise with $\sigma > 1$. Furthermore, while converting boundary single pixel thick, thinning in all directions is done, which could be alternatively done by local non-maxima suppression perpendicular to the edge direction. This could improve edge localization.

References

[1] V Hlavac M Sonka and R Boyle, *Image Processing Analysis and Machine Vision*, Chapman and Hall, 1993.

[2] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Addison Wesley Longman, 2nd ed., 2000.

[3] H Itoh J T Law and H Seki, *Image Filtering, Edge Detection, and Edge Tracing Using Fuzzy Reasoning.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996, vol. 18, pp. 481–491.

[4] D Marr and E Hildreth, *Theory of Edge Detection.* Computer Vision, Los Alamitos, CA, 1991, pp. 77–107.

[5] N Otsu, *A Threshold Selection Method from Gray Level Histograms.* IEEE Transactions on Systems Man Cybernet, 1979, vol. SMC-9, pp. 62–66.

[6] Ping-Sung Liao, Tse-Sheng Chen and Pau-Choo Chung *A Fast Algorithm for Multi-level Thresholding* . Journal of Information Science and Engineering, 2001, vol. 17, pp.713-727.

[7] H Oulhadj A Nakib and P Siarry, *Image Histogram Thresholding Based on Multiobjective Optimization.* Signal Processing, November,2007, vol. 87, pp. 2516–2534.

[8] L G Roberts, *Machine Perception of Three-Dimensional Solids*, Optical and Electro-Optical Information Processing, MIT Press, Cambridge, MA, 1965.

[9] J F Canny, *A Computational Approach to Edge Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, vol. 8(6), pp. 679–698.

[10] Neeta Nain, Vijay Laxmi, Bhavitavya Bhadviya and Arpita Gopal, *Corner Detection Using Difference Chain Code as Curvature.* The Third IEEE International Conference on Signal Image Technology and Internet Based Systems, SITIS'07, 2007, vol. Track III, pp. 766–770.

[11] Neeta Nain, Vijay Laxmi, Deepak Agarwal and Manish Khandelwal, in *Transformation Invariant Shape Descriptors.* International Conference on Image Processing and Computer Vision, IPCV'07, June 25-28, Nevada, USA, 2007, vol. I, pp. 545–550.

ISBN:978-988-98671-9-5