

An Information Theoretic Approach To Edge Detection

Erhan Gökçay¹, Murat Karakaya², Atila Bostan²

¹Software Engineering Department, Atilim University, Ankara, Turkey
erhan.gokcay@atilim.edu.tr

²Computer Engineering Department, Atilim University, Ankara, Turkey
murat.karakaya@atilim.edu.tr, atila.bostan@atilim.edu.tr

Abstract—This work presents an Information Theoretic Approach to Edge Detection named ITEDGE. Edge Detection is a very important abstraction method in Pattern Analysis and Object Recognition as it gives an outline of the objects by removing the remaining details. The basic step in any edge detection method is to detect a change in the intensity and decide if this change can be considered as an edge or not. In this study, an Information Theoretic Approach has been used to detect changes instead of gradient based methods. A simple filtering is applied to reduce the edge information and the strength of each edge is marked which is useful in further processing and elimination. The results are compared to “Canny” edge detector and differences in both methods are discussed.

Keywords — cluster evaluation function; edge detection; entropy; information theory; segmentation.

I. INTRODUCTION

Edge detection is a very important operation in image processing. The abstract information improves the performance of Object Recognition algorithms. Most of the edge detection algorithms depend on gradient methods, where spatial first and/or second derivatives of the image are calculated. Since the gradients are sensitive to noise, in most cases a smoothing operator (low pass filtering) is applied to the image to suppress noisy sections. Usually a Gaussian kernel is used in the smoothing process. There are other approaches which include space-scale detection algorithms and methods based on wavelets where a summary is given below.

In this paper, a different approach has been used to solve the edge detection problem. The main criterion is to detect changes and a change basically shows that there is a difference among regions. Instead of finding the rate of change among these regions, the proposed method will measure how different these regions are from each other by using a clustering measure that is shown to differentiate nonlinear clusters successfully in our previous paper.

II. EDGE DETECTION

A. Edge Definition

It should be noted that, by definition, an Edge is a subjective entity. The edge in very general terms is defined as a sharp intensity change in the image, where the definition of “sharp” is an open question. Certain changes can be considered as fluctuations and others can be considered as edges. Some changes are slow w.r.t. distance and it is very difficult to draw an edge and if any to find the exact position. Although there are methods to evaluate edge detection performance, these methods require a ground truth which is basically subjective. Therefore, we believe that the performance and usefulness of an edge detector depends on the context and the methods should be selected considering the application.

B. Previous Work

Edge detection is a very important topic, thus many different methods have been applied to solve the problem. The most famous edge detector is the Canny Edge detector [1]. The detector performs a detailed analysis after the gradient measurement for thickness and continuity. The image is smoothed to remove noise as the gradient calculation in general is sensitive to noise in the image. After finding the gradients local maxima are marked as edges, thresholding is used to find edges, and a hysteresis is used to track edge continuity. A multiscale approach is given in [4][14], where wavelet decomposition is used to create scales and a Hidden Markov Model is created with Viterbi algorithm to find the most likely sequence. For training the model, EM algorithm was used. A comparison with the Canny Detector was given. Wavelet transformation cannot keep the edge direction information. This deficiency was tried to be solved in [2] by combining the wavelet transform with a-priori shear transformation. On the other hand, a predictive coding and an adaptive thresholding were implemented in [3], where a visual comparison is done with Canny Edge detector. In [6], edge detectors are compared by choosing detector parameters statistically. The ground truth is estimated for comparison. The basic problem of this type of comparisons is that the ground truth, no matter how well it is estimated, is usually

subjective and context dependent. An edge detector using several DSC (discrete singular convolution) filters is designed in an earlier study [7], where the filters form a band-pass filter. In [8] the intensity surface is estimated using Least Square Support Vector Machines and a hybrid approach is developed including gradient and zero-crossing methods. Another approach is using wavelets to create a multi-scale transform, where a single adaptive threshold is calculated as in [9] and [12] and a multi-scale product is calculated in [11]. A survey of extensive use of Gaussian in edge detection is given in [10]. Another analysis after gradient calculation is given in [15] using fuzzy logic. Another solution to noise elimination in canny edge detector is given in [16] using discrete cosine transform. The noise is eliminated in the transformed domain. Another improvement is to calculate the thresholds statistically as in [17].

III. INFORMATION THEORY AND CLUSTERING

Clustering is an unsupervised method to separate data into groups by using a certain metric or distance function in such a way that the groups or clusters are more similar within, compared to other grouping choices. Similar to edge detection, clustering is also a subjective division depending on the distance measure. Some distance measures can only divide the data using a linear boundary whereas others can perform a nonlinear separation. Even the human eye will cluster the same data differently depending on other conditions and context.

Information theory has been used as a distance measure successfully in one of the authors' previous paper [5]. Using an information theoretic measure, nonlinear regions can be separated without any supervision. The derivation of the distance measure or CEF (Cluster Evaluation Function) was given elsewhere [5] and it will not be repeated here. Only the final formulation will be used. In (1), p and q are clusters of size N_1 and N_2 , respectively where $x_i \in p$ and $x_j \in q$. The Gaussian kernel needs a parameter σ for the kernel size. The proper value of this parameter is important in clustering but for edge detection it will be useful in creating a multi-scale approach.

$$CEF(p, q) = \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} G(x_i - x_j, 2\sigma^2) \quad (1)$$

A feature vector needs to be extracted from the data to be clustered. The feature set can be any original and/or transformed subset of the data which provides a good description of data. Using the feature set as an input to the distance measure, labels are assigned to each cluster such that the distance between different labels (clusters) has the maximum value. Searching the optimal labeling may need an exhaustive search and some heuristic algorithms are applied to limit the search space. As explained in [5], the final result is very promising and a nonlinear separation of clusters is possible with this method without any supervision as shown in Fig 1.

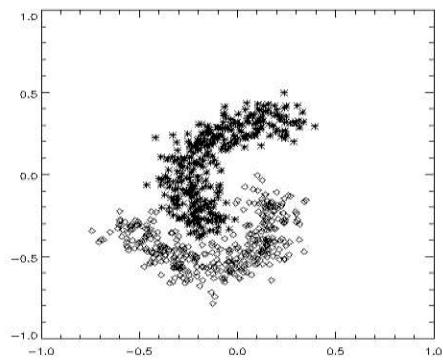


Figure 1: Clusters separated by CEF

IV. PROPOSED METHOD : ITEDGE

A. Local Clusters

The proposed method depends on using the information theoretic distance function used in clustering as a metric to find out local differences in the image. Here the distance measure will be used to find out how different the local clusters are from each other. The underlying assumption is that the distance between local regions (clusters) near an edge will be large, whereas the distance between slowly changing or homogenous regions will be small.

During the calculation we are not trying to measure an intensity change but we are trying to measure local cluster differences. The clustering using CEF as a distance measure depends on choosing the cluster points in such a way that the value of CEF is optimized. In ITEDGE algorithm, instead of optimizing CEF function to decide clusters, the cluster regions are fixed and CEF is used as a distance measure. In Fig [1] an intensity change is given and small, non overlapping clusters are shown, where each cluster consists of 2×2 cells. It is also possible to form 3×3 cells as well. Beyond 3×3 cells, the separation resolution of cluster distances starts disappearing. The reason for the loss of resolution is that when the cluster size increases, the change of having multiple edges in the cluster increases.

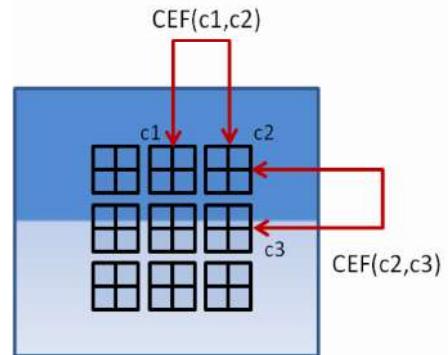


Figure 2

When we use CEF to measure the distance between clusters, for the given image we obtain

$$\text{CEF}(c1, c2) < \text{CEF}(c2, c3) \quad (2)$$

Similar differences can be obtained between the remaining cluster cells. The key feature is that differences in a homogeneous region will be small whereas differences around an edge will be high. A detailed explanation is given below.

B. Motivation

The Laplace operator is used in edge detection frequently where the operator is defined as a second order differential operator in the n-dimensional Euclidean space. The Laplacian of f is defined as:

$$\Delta f = \nabla^2 f = \nabla \cdot \nabla f \quad (3)$$

where ∇ is given as

$$\nabla = \left(\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_n} \right) \quad (4)$$

and

$$\nabla^2 f = \sum_{i=1}^n \frac{\partial^2 f}{\partial x_i^2} \quad (5)$$

For edge detection the image is smoothed using a Gaussian function and the Laplace operator is applied to the smoothed image to detect high frequency changes. The Laplace operator is a high-pass filter but some high-pass components, basically noise, are filtered by smoothing the image. The final result is a band-pass filter.

It can be shown that in the limit case the Laplace operator on Gaussian smoothed image can be approximated by the difference of the same image smoothed by two different Gaussian filters which basically creates a band-pass filter. During the convolution, differences in the image pixels are multiplied with the difference of Gaussian kernels, which is basically weighting the pixel differences nonlinearly. In order to use CEF function for edge detection we expect a similar behavior, i.e., smoothing and high-pass filtering.

The CEF formulation can be analyzed from a different point of view. In the CEF formulation the difference between pixel values are passed to the Gaussian kernel as a parameter, i.e., $\text{CEF} \cong G(\Delta x)$ where $\Delta x = x_i - x_j$ and x_i and x_j are from different clusters. So instead of weighting pixel differences nonlinearly, we are creating a nonlinear mapping from pixel differences and as can be seen from Fig 3, large differences, i.e. edges, are mapped to a low value and only small changes, i.e. smooth regions, are mapped to a high value nonlinearly. The variance of the Gaussian function determines the degree of filtering, where

large variances will map most pixel differences to a high value except very large ones, which basically creates a high-pass filtering and the degree is controlled by the variance. The degree of filtering can be used to create a scale-space representation of the Information Gradient. During the CEF calculation, the differences are calculated for each combination of the pixels in both cluster cells and averaged, which in turn works as a smoothing filter.

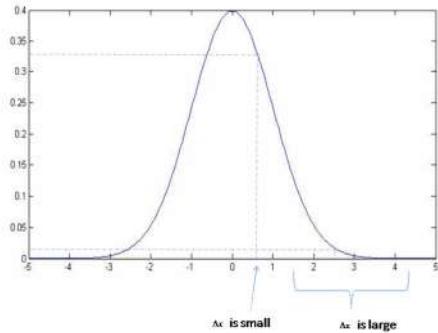


Figure 3

C. Feature Vector

The feature vector to measure the difference between cluster cells is formed as follows. For every pixel at the location (x, y) , the following 3×1 feature vector $f_{x,y}$ will be formed as shown in (6):

$$f_{x,y} = \begin{bmatrix} \text{pixel}_{x-\text{location}} \\ \text{pixel}_{y-\text{location}} \\ \text{pixel}_{\text{gray-value}} \end{bmatrix} \quad (6)$$

When we combine feature vectors of each pixel, we obtain a feature vector for a 2×2 cluster cell of size 3×4 , and the feature vector for a 3×3 cluster cell will be a vector of size 3×9 . In terms of magnitude, gray values can be large compared to pixel locations. This difference will create difficulties in determining the kernel parameters used in CEF distance function. For that reason each row of the feature vector is rescaled separately between 0 and 1.

D. Cluster Cells

There are several options in forming the cluster cells. The first option is the size of each cluster cell. We will consider only 2×2 and 3×3 cluster cells in this study. If we increase the cluster size, more than one edge may fit inside the cluster, which will make the detection of individual edges difficult or even impossible. The cells are defined as in Fig. (4).

During the calculation we want to measure the differences between cluster cells. The second option is whether the cells nearby the target cell overlap each other or not. Overlapping and non overlapping 2×2 corner cells are shown in Fig 4.



Figure 4

Following this definition overlapping and non overlapping 3×3 corner cells are also shown in Fig 5. Side by side cells are also defined similarly.

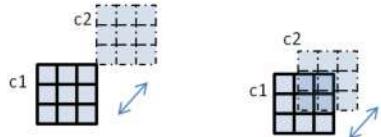


Figure 5

E. Information Gradient (IG)

The algorithm starts by reading the image and converting the image to a feature vector and rescaling the whole image once. All cluster cells are taken from the rescaled image. For each pixel in the image the corresponding cluster cell is formed (2×2 or 3×3) and CEF function is calculated between this cell and all the other cells around it. When all directions are considered, the total number of CEF calculations for a pixel is 8. The calculation is shown in Fig 6. For simplicity only the non overlapping cells are shown. In the image, the cells are touching each other.

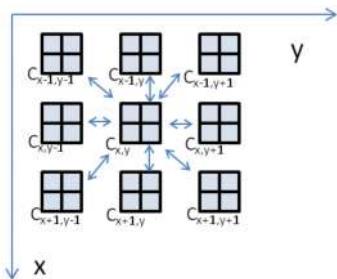


Figure 6

The absolute maximum value (which shows maximum separation of clusters) is taken between 8 calculations and recorded for that pixel as the IG value as given in (7). The calculation is done for each pixel location and at the end, an image is obtained that shows cluster cell differences, which we call the Information Gradient of the image.

$$IG_{x,y} = \max \left\{ \begin{array}{l} CEF(C_{x,y}, C_{x-1,y-1}) \\ CEF(C_{x,y}, C_{x-1,y}) \\ CEF(C_{x,y}, C_{x-1,y+1}) \\ CEF(C_{x,y}, C_{x,y-1}) \\ CEF(C_{x,y}, C_{x,y+1}) \\ CEF(C_{x,y}, C_{x+1,y-1}) \\ CEF(C_{x,y}, C_{x+1,y}) \\ CEF(C_{x,y}, C_{x+1,y+1}) \end{array} \right\} \quad (7)$$

F. Edge Filter

We can consider the operation as an Information Theoretic Edge Filter $ITEF(x, y)$ where the image is convolved with the filter to produce the edge information. The $ITEF$ filter consists of the following template given in Fig. 7 with the center located at (x, y) where the operation at the center of the template is defined as in (3). A cluster given in (7) at any point (n, m) is defined as in (8) where $f_{n,m}$ is the feature vector at the location (n, m) .

$$C_{n,m} = \{f_{n,m}, f_{n-1,m}, f_{n,m-1}, f_{n-1,m-1}\} \quad (8)$$

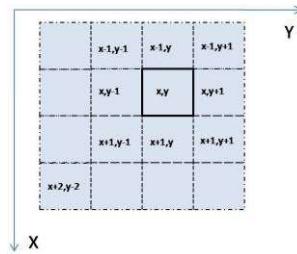


Figure 7

The Information Gradient of the whole image can be found by convolving the template with the image as in (9).

$$IG_{image}(x, y) = (Image * ITEF)(x, y) \quad (9)$$

The algorithm is tested on the image given in Fig. 8.

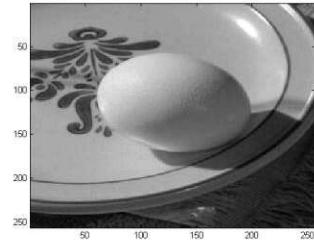
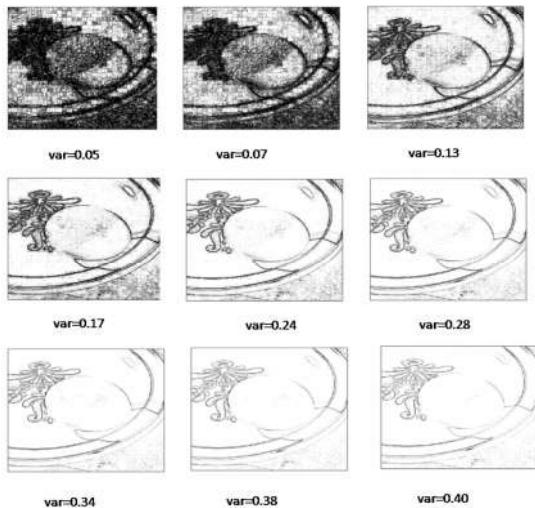
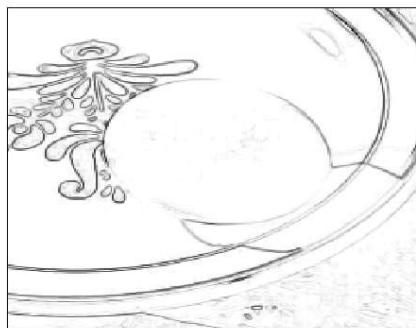
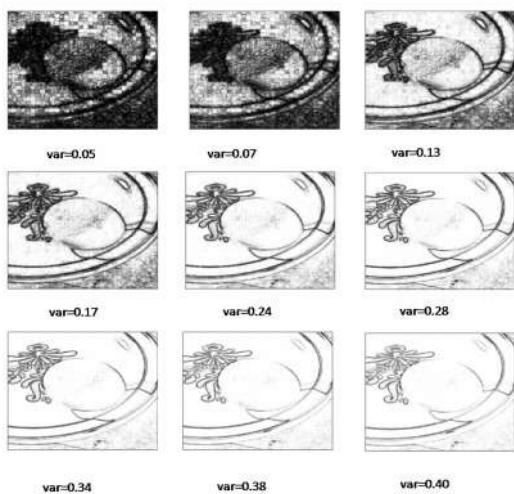


Figure 8

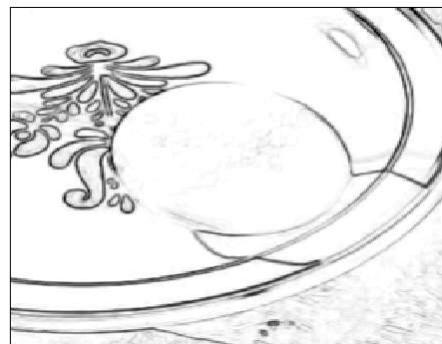
The clusters are formed using 2×2 cells and later using 3×3 cells. The variance of the kernel is also modified in a range. The first test is done using a 2×2 cell and results are given in Fig. 9 and an enlarged picture with variance=0.38 is given in Fig. 10.

**Figure 9****Figure 10**

The second test is done using a 3×3 cell and results are given in Fig. 11 and an enlarged picture with $\text{var}=0.38$ is given again in Fig. 12.

**Figure 11**

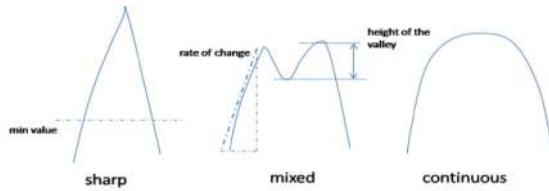
Using 2×2 cells creates a better resolution and separation but using 3×3 cells creates a higher response at edge boundaries. Different responses are expected since using a smaller cell size will reduce the chance of missing edges and averaging is less, which increases the resolution. But on the other hand, increasing the cell size uses more information hence the responses is stronger but at the same time edges found are more blurry since larger cell size tends to average the information processed more than a smaller cell size.

**Figure 12**

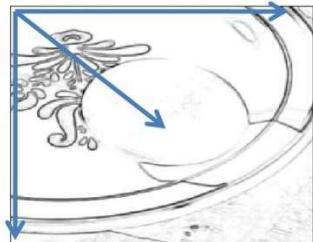
Another important observation is that as the variance changes the dominant structures are preserved whereas small details tend to disappear. This makes the determination of the variance easier where it also gives us a chance to investigate the edges at any level.

G. Edge Evaluation

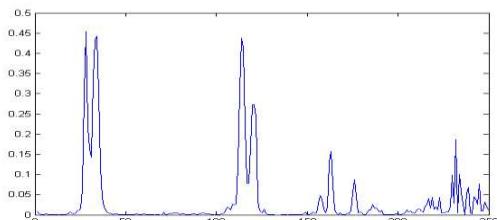
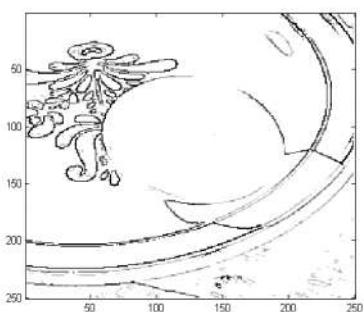
When the Information Gradient of the image is generated using *ITEDGE* algorithm similar to a gradient map of the image, the edges should be extracted from the map. We analyzed all peaks of the map for 3 different cases as shown in Fig. 13 and decided to the edges depending on the strength and on the shape of the map. The rate of change is also considered as a factor to eliminate small changes from being considered as edges. As shown in Fig. 13, the height of the valley at the peak is also checked to differentiate the edge from a continuous edge. By setting a threshold we marked the edges as edges or non-edges, and noise together with small values are eliminated from the map. The final edge pixels are also classified using gray values to indicate the strength of the edge. At this moment no connectivity analysis between pixels/edges has been made but similar analysis of the Information Gradient is always possible.

**Figure 13**

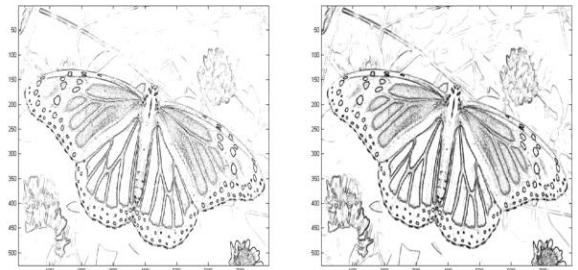
The final Information Gradient is scanned in each axis (horizontal, vertical and diagonal) as shown in Fig. 13 and the pixels are marked as edges or non-edges.

**Figure 14**

A sample single line scan is shown in Fig. 14 where assumed shapes of peaks and edges in Fig. 15 can be seen clearly including noisy edges with low values. The result of the scan is shown in Fig. 16, where weak edges are in lighter colors than strong edges. Although extra thinning or filtering can be applied to the edges, we managed to show that the Information Gradient preserves main structures in the image and the edges are extracted easily from the map.

**Figure 15****Figure 16**

The result of the convolution with the edge filter using another image (Fig. 17) has been given in Fig. 18 with cluster sizes 2 and 3 respectively. The image is converted to grayscale before applying the filter.

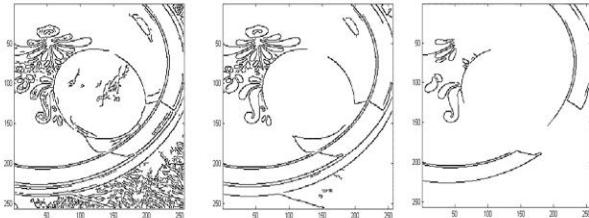
**Figure 17****Figure 18**

H. Comparison with Canny Edge Detector

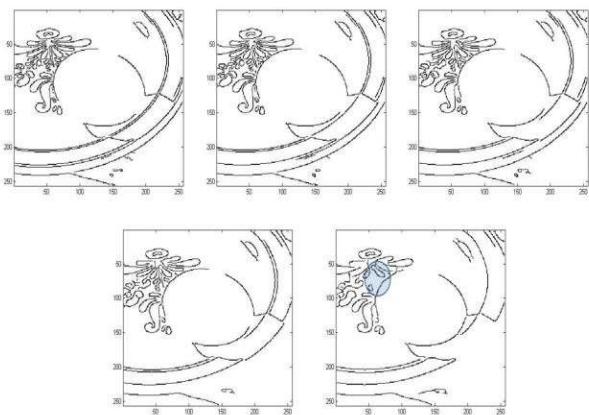
The definition of an edge is a subjective entity except artificially created images with very high contrast. Because of that reason a deterministic comparison of edges is not implemented but the edges are compared visually whether the basic structures in the image are captured or not and how the filter responds to different scales.

Exact threshold values for canny edge detector and for the information edge detector are not given as they are not important in the comparison since there is no ground truth in edge detection in general. The comparison is done based on how the detectors respond to parameter changes. Otherwise the choice of an edge detector and its parameters depend on the subjective need for finding an edge.

Although there are many detectors, the canny edge detector is a good detector to compare our algorithm with. Another reason is that the method is similar to a gradient calculation like the one used in canny edge detector. To see the change of the result with respect to the kernel size and threshold, canny detector has been tested with different parameters. The details, as can be seen in Fig. 19, depend on the threshold. When the threshold is increased some structures disappear as expected.

**Figure 19**

On the other hand, when the variance is increased, the structures tend to join each other forming other structures as seen in Fig. 20. Some structures also disappear as well; an example is marked with a circle in the same figure.

**Figure 20**

ITEDGE algorithm keeps the structures much better than the canny edge detector when the variance has been changed in a range; hence it is better suited to create a scale-space analysis. The edge is calculated depending on the cluster differences; hence the strength of the difference (edge strength) can be controlled by changing the variance and not the threshold. It is obvious that the analysis after finding the Information Gradient can be extended but this is out of the scope of this paper. The results are summarized in Table 1.

Parameter	Canny	ITEdge
Threshold	Threshold decides when to consider the result of the gradient as an edge and hence depending on the threshold several structures may disappear	It is used to eliminate noise and to mark edges and threshold values may still modify the edges but the threshold is used just to mark the edges not depending on its value but also on its shape preserving its value; hence it has less affect on the result

Variance	Change in the variance increases the scope of the kernel in the image convolution; hence several structures may appear and disappear depending on its value.	When modified only weak edges disappear, since variance only modifies the response of the difference calculation (nonlinear difference in clusters) but its range is still limited to the cells. A scale-space analysis can be implemented because of this feature. The edge strength can be better controlled.
Single Pixel Edges	After the gradient, single pixel analysis is implemented	An analysis is done to mark the edges as edge or non-edge but a single pixel analysis is not included. Depending on the requirements, it can be added.
Noise	The image is smoothed before gradient calculation since it is sensitive to noise	The cluster difference calculation removes noise to some extend because of the averaging and kernel used.

Table 1

V. CONCLUSION

In this paper we managed to show that the filter proposed encapsulates a regularization and edge detection feature at the same time and performs nicely in capturing the basic structures in the image in several scales. This is very important in image analysis and pattern recognition. In most algorithms increasing the kernel size will result in blurring of even strong edges, which makes a multi-scale approach very difficult. Using different scales is also easy to implement using our filter, since increasing the kernel size will result in less contribution from weak edges but the filter will not blur and remove strong edges. Only increasing the cluster size in the filter will result in blurring.

VI. FUTURE WORK

ITEDGE algorithm can be improved in many ways. One of the improvements is the running time. Calculation of Information Gradient for each pixel is a time consuming task. The other improvement is the evaluation of edges after the convolution. A more comprehensive analysis will certainly improve the edge quality in terms of smoothness and thickness, and it will eliminate noise better.

REFERENCES

- [1] Canny, John, "A Computational Approach to Edge Detection," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.PAMI-8, no.6, pp.679,698, Nov. 1986
- [2] Pengfei Xu, Qiguang Miao, Cheng Shi, Junying Zhang, Weisheng Li, "An edge detection algorithm based on the multi-direction shear

- transform", Journal of Visual Communication and Image Representation, Volume 23, Issue 5, July 2012, Pages 827-833.
- [3] Yuan-Hui Yu, Chin-Chen Chang, "A new edge detection approach based on image context analysis", Image and Vision Computing, Volume 24, Issue 10, 1 October 2006, Pages 1090-1102
- [4] Junxi Sun, Dongbing Gu, Yazhu Chen, Su Zhang, A multiscale edge detection algorithm based on wavelet domain vector hidden Markov tree model, Pattern Recognition, Volume 37, Issue 7, July 2004, Pages 1315-1324
- [5] Gokcay, E.; Principe, J.C., "Information theoretic clustering," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.24, no.2, pp.158,171, Feb 2002
- [6] Yitzhaky, Y.; Peli, E., "A method for objective edge detection evaluation and detector parameter selection," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.25, no.8, pp.1027,1033, Aug. 2003
- [7] Z.J. Hou, G.W. Wei, "A new approach to edge detection", Pattern Recognition, Volume 35, Issue 7, July 2002, Pages 1559-1570
- [8] Sheng Zheng, Jian Liu, Jin Wen Tian, "A new efficient SVM-based edge detection method", Pattern Recognition Letters, Volume 25, Issue 10, 16 July 2004, Pages 1143-1154
- [9] Wan Yue-Liang; Cao Yuan-da; Li Dun, "A Single Threshold Edge Detection Algorithm Based on Multiscale Fusion," Computational Intelligence and Security, 2006 International Conference on , vol.2, no., pp.1710,1715, 3-6 Nov. 2006
- [10] Basu, M., "Gaussian-based edge-detection methods-a survey," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on , vol.32, no.3, pp.252,260, Aug 2002
- [11] Zhao Xiaoli, "Edge Detection Algorithm Based on Multiscale Product with Gaussian Function", Procedia Engineering, Volume 15, 2011, Pages 2650-2654
- [12] Wei Jiang; Kin-Man Lam; Ting-Zhi Shen, "Efficient Edge Detection Using Simplified Gabor Wavelets," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on , vol.39, no.4, pp.1036,1047, Aug. 2009
- [13] Scott Konishi, Alan Yuille, James Coughlan, "A statistical approach to multi-scale edge detection", Image and Vision Computing, Volume 21, Issue 1, 10 January 2003, Pages 37-48
- [14] P. Vikas, M. S. Lakshmi, M. S. Rajkumar and P. M. K. Prasad, "Edge detection in noisy images using wavelet transform," 2015 National Conference on Recent Advances in Electronics & Computer Engineering (RAECE), Roorkee, 2015, pp. 36-39
- [15] E. Anas, "Edge detection techniques using fuzzy logic," 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, Delhi NCR, India, 2016, pp. 169-173.
- [16] Miaomiao Zhao, Hongxia Liu and Yi Wan, "An improved Canny edge detection algorithm based on DCT," 2015 IEEE International Conference on Progress in Informatics and Computing (PIC), Nanjing, 2015, pp. 234-237.
- [17] S. Bhattacharjee, A. Banerjee, S. Basu and S. Latib, "Automated threshold driven edge detection using 2-way difference calculation and neighborhood comparison," 2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI), Faridabad, 2015, pp. 122-125.