International Journal of Civil Engineering and Technology (IJCIET)
Volume 9, Issue 1, January 2018, pp. 728–737, Article ID: IJCIET_09_01_070
Available online at http://http://www.iaeme.com/ijciet/issues.asp?JType=IJCIET&VType=9&IType=1
ISSN Print: 0976-6308 and ISSN Online: 0976-6316

COMBINATIONAL EDGE DETECTION USING MULTIPLE COLOR CHANNELS AND GRABCUT

Ajinkya A. Gaitonde
Research Scholar, Department of Computer Science,
Christ University, Bengaluru, India

Vijayalakshmi A., PhD
Assistant professor, Department of Computer Science,
Christ University, Bengaluru, India

ABSTRACT

Identification of edges is very important in feature extraction and pattern recognition. An edge of an image detected by converting it from RGB format to a Grayscale image and can sometimes be inefficient and inaccurate. This inefficiency is caused due to various color differences that get erased or rewritten during the process of grayscale conversion. Edge locations in a colored image are derived by analysing the variations in the multiple color channels and merging the gradients in these channels to compute a single edge snapping vector field, which is derived using the Euclidean distances between two distinct pixels in an image. This lets to retain the various multidimensional characterizations of color channels of an image as the color differences can be closely calculated using human perceptions. Hence this method is proved to provide more accuracy in edge detection. Furthermore, the results are improved when combined with a finer edge detection method GrabCut, which allows the user to detect the edges in an image using reduced iterations. The proposed paper uses a combinational approach to efficiently acquire edges in the image by enhancing the color properties of an image and then using the GrabCut method to retrieve the edges present in an image.

Key words: Edge detection, multiple color channels, multidimensional, GrabCut.

Cite this Article: Ajinkya A. Gaitonde and Vijayalakshmi A, Combinational Edge Detection using Multiple Color Channels and GrabCut. *International Journal of Civil Engineering and Technology,* 9(1), 2018, pp. 728-737.
http://www.iaeme.com/IJCIET/issues.asp?JType=IJCIET&VType=9&IType=1

1. INTRODUCTION

There are numerous software tools that are used to develop and manipulate digital images. Some software allows the users to easily manipulate, modify and adjust the content in an image. These applications allow the user to make global manipulations to the image, which affects the entire content of the image. Through localized manipulation of images only a selected area of the image can be adjusted or manipulated. The desired manipulation of these images can be achieved by a two-step method. First the area or portion to be manipulated needs to be selected and then the manipulations need to be applied on that selected area. This also allows the user to separately copy, adjust or extract any selected portion. For example, if the user wishes to increase the contrast and darken a certain area of the image he can do this by just selecting the respective area and then applying the suitable adjustments to that portion of the image. Selecting the certain portion of the image signifies a procedure of detecting the edge or a boundary of the image. This edge is considered as a distinct separation between the selected portion and the unselected portion of the image, also known as foreground and background.

Edges give significant information from the boundaries of an image. This helps in extracting required data while working with images. There are three types of edges in an image; horizontal, vertical and diagonal edges. Edge detection is created by the changes or the interruptions in an image processing technique for finding the boundaries of objects within images. It is a finely-honed field on its own within image processing and is an image segmentation technique that divides the image into meaningful parts or regions. It mainly works by detecting discontinuities in the brightness of the image. Thus, edge detection is used for image segmentation and data extraction in areas of image processing, computer vision, machine vision, etc. Most of the information about the shape of an image is enclosed in edges. Various edge detections algorithms like Sobel, Prewitt, Roberts, and Canny and Laplacian of Gaussian methods are used for different edge detection and segmentation purposes.

2. LITERATURE REVIEW

Edge detection is considered as an important aspect of digital image manipulation. An edge can be called as a set of connected pixels that form a boundary like separation between the foreground and the background. Each pixel consists of three different sub-pixels that form a digital image. These pixels with different color properties determine the differences in the foreground and the background. In [1][9], the authors study the various edge detection algorithms and their differences that play a vital role in the detection of edges in an image by comparing them over various noisy images. The paper also discusses the various aspects, methodologies and algorithms that are used for edge detection. After the comparison of various edge detection algorithms, it is concluded that canny edge detector displays better chances of edge detection results with less sensitivity towards any noise as compared to the other methods and algorithms. Another aspect discussed in the paper is the combination of edge detection with linearity measure to accurately detect straight edges that are present in images with high noise ratio. The paper also discusses the various advantages of canny operator over the other edge detection operators.

Classification of edges is very important as it simplifies various problems faced during the process of Image Processing, by associating the processing rules to every edge. In [2] the authors discuss about different types of edge detection algorithms and operators such as Gradient edge detectors, Zero crossing, Laplacian of Gaussian (LoG), Gaussian edge detectors and Colored edge detectors, their advantages and their disadvantages. Table 1 depicts the advantages and disadvantages stated by the experimented edge detection algorithms. These variations and their properties help in selecting a much more efficient and fundamental edge detection operator. The authors experiment on edge detection algorithms over thirty different sample images where in only few of the methods display efficiency and acceptable performance.

Table 1 Advantages and Disadvantages of edge detection algorithms [2]

Operator	Advantages	Disadvantages
Classical (Sobel, Prewitt, Krisch,..)	Simplicity, Direction of edges and their orientation	Sensitivity to noise, inaccurate
Zero crossing(Laplacian, Second directional derivative)	Detection of edges and their orientations, having fixed characteristics in all directions	Reresposing to some of the existing edges, sensitivity to noise
Laplacian of Gaussian(LoG) (Marr-Hildreth)	Finding the correct places of edges, testing wider area around the pixel	Malfunctioning at corners, curves and where the gray level intensity function varies, not finding the orientation of edge because of using Laplacian filter
Gaussian(Canny, shen-castan)	Using probability for finding error rate, localization and response, improving signal to noise ratio, better detection specially in noise conditions	Complex computations, false zero crossing, time consuming
Colored edge detectors	Accurate, more efficient in object recognition	Complicated, Complex computations

Colours in the target image also play a vital role in the detection or selection of edges. Paint selection which is paint based local selection method, allows the user to get instant feedback even during the selection process of multi-megapixel images. The efficiency of the algorithm is mostly due to a progressive algorithm, multi-core graph cut and adaptive band up-sampling. This method allows the user to interact with the image through an interface. The feedback is critical for interactive image editing and for accurate and optimal edge detection. The foreground color information is retrieved after each optimization applied to the image. The paper discusses about various progressive painting based algorithm and user interfaces that are used for selection of edges in the image in a localized manner. The proposed method provides an instant feedback to the user via an interactive selection for multi-megapixel images [3].

Threshold is one of the most crucial and of at most important factors in the detection and selection of edges in a target image. In [4] the authors propose an edge detection technique that is considered as a rapid technique as it uses logical operations on circularly shifted images to detect the edges in the image with lesser complexity and higher efficiency. Using three stages of edge detection, the proposed method allows the user to optimally select and manipulate the edges in an image. The proposed algorithm is proved more efficient and that it also preserves the clear edges by comparing it with other existing methods of edge detection.

In [5] authors provide another method, used for edge detection i.e., by using multiple color channels present in the image. For example, the RGB color model is used for sensing, representing and for displaying the color in an electronic system, but these values are often not very feasible in terms of creating or establishing meaningful relations between visual appearance of difference colors as perceived by the human eye. In this research executed, authors state that the proposed method splits and characterizes each of the pixels in the image into three unique color parameters that further define the pixel.

In [6] authors state a more powerful and iterative image optimization method that allows us to simplify the user interactions significantly at which they are needed for obtaining the given quality of result. The author tests the proposed method on various images with different level of camouflage levels in orders to check the detection efficiency of the method. The research paper also states previously used edge detection methods and tools such as Magic Wand, Intelligent Scissors, Bayes matting, Knock Out2, and Graph Cut. These methods and tools do happen to have their respective limitations in terms of optimality, significance and accuracy. The author also speaks about the difficulties faced by the earlier methods and states that the GrabCut method aims at achieving high performance and good quality if edge detection at minimum interaction efforts put by the users. Figure 1 shows the iterative segmentation in grab cut. In this diagram the author depicts the various steps taken to successfully detect the edges using grab cut method.

Initialisation

- User initialises trimap T by supplying only T_B. The foreground is set to $T_F = \emptyset$; $T_U = \overline{T}_B$, complement of the background.
- Initialise $\alpha_n = 0$ for $n \in T_B$ and $\alpha_n = 1$ for $n \in T_U$.
- Background and foreground GMMs initialised from sets $\alpha_n = 0$ and $\alpha_n = 1$ respectively.

Iterative minimisation

1. *Assign GMM components to pixels:* for each n in T_U,
$$k_n := \arg\min_{k_n} D_n(\alpha_n, k_n, \theta, z_n).$$
2. *Learn GMM parameters* from data $\mathbf{z}$:
$$\underline{\theta} := \arg\min_{\underline{\theta}} U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z})$$
3. *Estimate segmentation:* use min cut to solve:
$$\min_{\{\alpha_n :\, n \in T_U\}} \min_{\mathbf{k}} \mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}).$$
4. Repeat from step 1, until convergence.
5. Apply border matting (section 4).

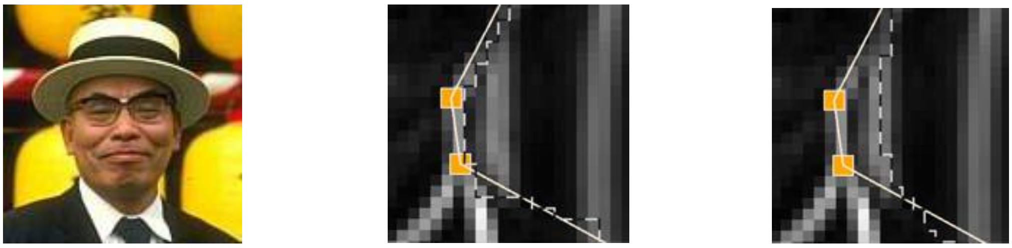
User editing

- *Edit:* fix some pixels either to $\alpha_n = 0$ (background brush) or $\alpha_n = 1$ (foreground brush); update trimap T accordingly. Perform step 3 above, just once.
- *Refine operation:* [optional] perform entire iterative minimisation algorithm.

Figure 1 Iterative image segmentation in GrabCut proposed by authors in [6].

Level Set Method allows the user to set the initial edge from where all the other edges will be detected [7][10]. This initial edge then keeps on evolving and updating to make it more accurate. This method is flexible as the extraction of the desired edge is much simpler and the curves move based on human interactions. Various software is often used for localized edge detection methods to manipulate images and adjust them according to the need of the user. Many visual effects can also be used in-order to manipulate the image furthermore. Accurate edge detection thus becomes an important concern when working with visual effects that include blurring, sharpening, and smoothing effects. The paper discusses the various fundamentals of the level set method. The authors' mention that a zero-level set for a dimensional hypersurface is created that contains the interface. With this the users can control the curves evolution by just controlling the hyperspaces evolution. The authors also state that the cursors' margins surround the main region of the target image, narrowing down the region and perform the edge detection. The results of the paper state that the method is effective for fetching edges the objective and the background by applying it in different ways.

The most concerning issue in the detection of any edge is the difficulty in the visual representation of the edges that are already detected. Lazy Snapping is an interactive cut-out tool that allows us to separate the course of the edge and the fine scale processing of the edge making object specification and detailed adjustments much easier. It also provides an instant visual feedback of the true object boundary along with an interactive user interface tool for flexible control and better user experience. The method allows the user to first mark and then group the desired object in the foreground. Efficiency is improved through the graph cut algorithm which is built on images that are pre-composed over segmentation rather than their pixels [8]. Figure 2 depicts the edge detection using lazy snapping method. In this figure the author states that (a) is the original image where in (b) and (c) are the color gradient of the image region near the neck. In (b) the users can mark the strong edges using polygon soft constraints. The (c) shows the same image but without using the use of the constraints which results in erroneous edges due to ambiguity.



(a) Original image (b) With constraint (c) Without constraint

Figure 2 Edge detection using lazy snapping method [8].

3. PROPOSED METHOD IMPLEMENTATION AND RESULTS

3.1. GrabCut

Normally, an ideal matte detection tool should be able to generate consistent alpha values of the entire region of a sample image. It should also not have the constraints of the values being just 0 or 1 for the alpha matte of the image. This allows automatic solution for various problems of edge detection when it comes to smaller and finer foreground images such as fine hair samples or opacity of the smoke. As GrabCut is based on an iterative edge detection method Graph cut, the first step in GrabCut is to retrieve the hard segmentation by using Graph cut. After that the boundaries detected using hard segmentation are narrowly focused on by applying border matting and the alpha values are obtained. GrabCut is an enhancement to two mechanisms of Graph cut i.e. incomplete labelling and iterative estimation and includes alpha computation to be used or border matting. GrabCut also focuses on the energy consumed during the process of edge detection. This iterative energy minimisation scheme is an enhancement to previously used one-shot algorithm.

As discussed, GrabCut method enhances the procedure followed in Graph cut algorithm by

1. Making use of color instead of grey level. This is executed by replacing the monochrome image by Gaussian Mixture Model.

In GrabCut, two GMMs are considered one for the background and other for the foreground of the image. This lets assigning a unique GMM component to each of the pixel in an image. With these changes, the energy function turns out to be

$$E(\alpha,k,\theta,z) = U(\alpha,k,\theta,z) + V(\alpha,z)$$

Where U(α,k,θ,z) is the data term that is defined by taking into consideration the color GMM models which helps in promoting pixels to belong to the foreground or the background depending on how well they fit into the particular GMM.

The data term is expressed as

$$U(\alpha,k,\theta,z) = \sum D(\alpha n , kn , \theta, zn)$$

$$D(\alpha n , kn , \theta, zn) = - \log p(zn \mid \alpha n ,kn ,\theta) - \log \pi (\alpha n ,kn)$$

In the above term, p(·) is the Gaussian probability distribution. Gaussian probability distribution is described by its mean μ and variance Σ. p(·) and π(·) are mixture weighting coefficients.

From the above definition of the terms, the parameters of the model can be defined as

$$\theta =\{ \pi (\alpha ,k), \mu(\alpha ,k), \Sigma(\alpha,k),\ \alpha =0,1 , k= 1...K \}$$

here, V(α,z) is the smoothness term only using the Euclidean distance in RGB colour space instead of in the monochrome space.

$$V(\underline{\alpha},\mathbf{z}) = \gamma \sum_{(m,n)\in\mathbf{C}} [\alpha_n \neq \alpha_m] \exp -\beta \|z_m - z_n\|^2$$

2. The min cut estimation algorithm is replaced by a iterative procedure

The iterative procedure followed in GrabCut method helps in automatic refinement of the opacities α. This term defines all those pixels that belong to the foreground and background of a given image. Following steps define the iterative minimisation procedure followed in GrabCut.

Step 1: GMM components are applied on to all those unlabelled pixels for each value of n in TU.

$$k_n := \arg\min_{k_n} D_n(\alpha_n, k_n, \theta, z_n)$$

Step 2: For each GMM component k of an image z, the mean value μ(α,k) and covariance value Σ(α,k) are computed from the subset of pixels {zn: kn= k and αn = 0 or 1}.The weights π(α,k) are also computed. This is nothing but the as the ratio between the subset zn and all pixels belonging to the same model.

$$\underline{\theta} := \arg\min_{\underline{\theta}} U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z})$$

Step 3: Solving the parameters obtained from step 1 and step 2, we get

$$\min_{\{\alpha_n:\ n\in T_U\}} \min_{\mathbf{k}} \mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}).$$

Step 4: Step 1 to Step 3 are repeated until the value of E ceases to decrease significantly.

3. GrabCut allows incomplete labelling.

In GrabCut only those labels specified by the user is considered when a user places a bounding box around an object. When a bounding box is considered, all those pixels outside the bounding box is added on to TB and all those pixels inside are added into TU. The inside pixels are used to generate the foreground GMM. These pixels may change to the background after a few iterations of the algorithm. At the same time all those pixels that are outside the bounding box are assigned to TB at the beginning of the algorithm does not change their label.

3.2. Multiple Color Channels

The term ‘edge snapping vector field’ refers to the field corresponding to a pixel array that further forms a digital image. Here the direction of maximum color difference and the magnitude of color difference are calculated for each pixel existing in the array. Various techniques are used in-order to efficiently detect edges in an image by clearly analysing the different variations that are present in the multiple color channels that characterize the image. These techniques further help in proper accurate and efficient detection of edges in the localized area. Color differences get lost during the grayscale conversion and can be prevented by restricting the conversion of the sample image into grayscale image. Even though edge detection plays a vital role in the manipulation of any digital image, the tools provided by the existing software are not very efficient and accurate and fall short to user expectations.

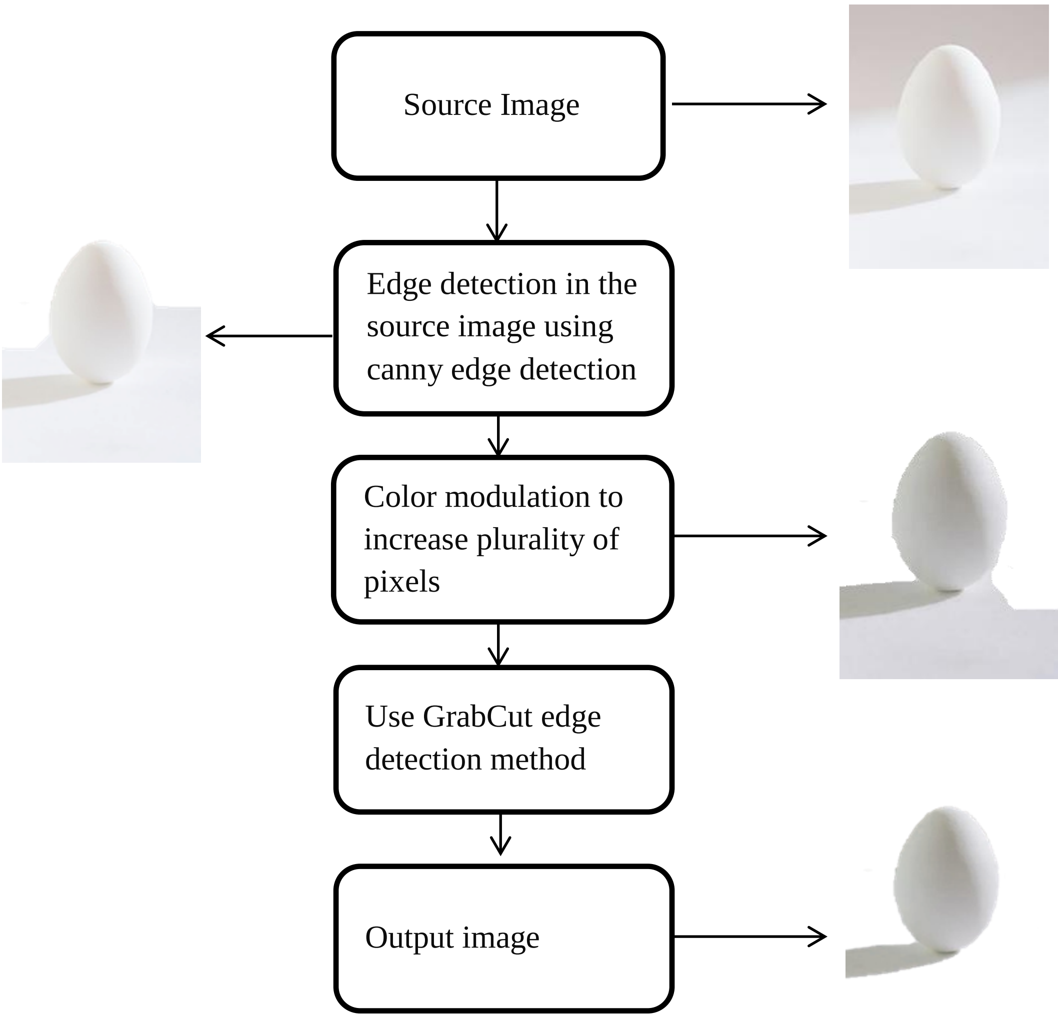
3.3. Proposed Method

In the proposed work, two existing methods that have been widely used for the detection of edges are combined to obtain an efficient detection of the edge. Using GrabCut as a base method and allowing it to access the target image though color correction on multiple color channels allows us to obtain a fine result. The series of implementation goes through three steps that allow the user to make sufficient tweaks and changes accordingly to maintain the feasibility and efficiency of the result. The user’s interaction is of importance as the inputs that are given by the user reduce the possibilities of unnecessary image segmentation and a predefined area can be concentrated upon.

3.4. Architecture of the proposed edge detection method

Various techniques that have been designed for solving matting of the target image are only effective when the adequate dissociation between the background and the foreground. The color distributions may sometimes fail in camouflages. It is not completely possible to acquire and perceive the full matte of the image. In the proposed method we first, we try to increase the possibilities of acquiring the color distributions while in camouflage. This is done by using the multiple color channels that are used to determine the target image.

By modulating the multiple color channels, the image thereby retains all its color differences that might be lost during the conversion of the target image to a grayscale image. This non-conversion method allows human level perception of the image there by allowing much efficient extraction of edge data. For an example if we consider a simple image that has a white object over a white background. Now to extract the required edge of this white object can become tedious. Even when the image gets converted to a grayscale image the possibilities of obtaining the required edge can be of much difficulty.



3.5. Steps taken to formulate the edge detection in proposed method

Step 1: User initialises the sample image.

Step 2: Use of canny method for first level of edge detection.

Step 3: Use color modulation to enhance the plurality in the pixels more better visualization.

Step 4: Use of GrabCut edge detection method to efficiently detect the edge over brighter background.

Step 5: Repeat method until convergence

The use of multi-color channels plays a vital role as we can manipulate the properties of the color distributed along the borders of the foreground object over the similar colored background. This method allows the user to precisely extract the desired local object area for further processing of the image. The Euclidian distance between the two pixels of an image characterized using multi-dimensional color model will be more precise in terms of color perceived by a human eye, than the Euclidian distance between two pixels of an image characterized using a grayscale conversion.

On application of these two methods over a sample image produces an efficient result and better edge detection. The detection of edges over brighter background is made possible after adjusting the color properties of each plurality in pixels present in the image.

(a) (b)

(c) (d)

Figure 4 (a) sample image, (b) canny edge detection method applied for first level of edge detection, (c) color properties in the image manipulated and adjusted for adequate results; saturation and brightness levels adjusted, (d) finally GrabCut edge detected method applied for finer detection of edges over brighter background.

4. CONCLUSIONS

The proposed method consists of steps that involve a multidimensional image with plurality of pixels as a source input. Each of these pixels has color which is further characterized into three color parameters that determine each pixel. By not converting the image into a grayscale image we preserve its color differences and properties. The method further creates an edge snapping vector field based on the three color parameters for the base image. The edge is created and selected on a point where the magnitude of the color in the proposed region is greater than the other available points. This allows a better selection of pixels and edge. Yet there are always flaws when it comes to accurate and efficient detection of edges in various images. Sometimes it is much difficult for complex images. The domain of edge detection is very vast and the effectiveness and efficiency of detection of the edges in an image can be further improved through further research.

REFERENCES

[1] Dr. R. Shiva Ram Prasad and S. Suresh 2016, ‘A Review on Edge Detection Algorithms’, International Journal of Mechanical Engineering and Computer Applications, Vol. 4, No. 1, Jan – Feb, ISSN – 2320-6349.

[2] Mohsen Sharifi, Mahmoud Fathy and Maryam Tayefeh Mahmoudi, ‘A Classified and Comparative Study of Edge Detection Algorithms’ Proceedings of International Conferences of Information and Technology, Coding and Computing (ITCC’02) IEEE 0-7695-1506-1/02

[3] Jiangyu Liu, Jian Sun, Heung – Yeung Shura, ‘Paint Selection’

[4] Kiran Jot Singh, Divneet Singh Kapoor, Anshul Sharma and Amit Kumar Kohli 2016, ‘Multi level threshold-based Edge Detection using Logical Operators’

[5] Byungmoon Kim and Sunil Hadap 2016, ‘Edge Detection using Multiple Color Channels’, United States Patent, US 9,355,457 B1, May 31.

[6] Carsten Rother, Vladimir Kolmogorov and Andrew Blake, ‘GrabCut - Interactive Foreground Extraction using Iterated Graph Cuts’, Microsoft Research Cambridge, UK.

[7] Zhang Junru and Jiang Xuezhong, ‘Research on Edge Extraction with Level Set Method’, Commission IV, WG IV/3.

[8] Yin Li, Jian Sun, chi-Keung Tang and Heung-Yeung Shum, ‘Lazy Snapping’, Microsoft Research Asia

[9] Serene Banerjee, Noushath Staffi and Pulkit Parikh 2012, ‘Edge Detection’, United States Patent Application Publication, US 2012/0093434 A1, April 19

[10] JawGwang Lim, Byungmoon Kim and Sunil Hadap 2014, Selection Editing using a Localized Level Set Algorithm

[11] Deepika Rani Sona, Prayline Rajabai. C, Debopam De y, Akanksha Jain, Rashmi Ranjan Das and Stephen Olatunde Olabiy isi, A Case Study: Edge Detection Techniques Using Hough Transform and Cann y Edge Algorithm, International Journal of Mechanical Engineering and Technology 8(10), 2017, pp. 729–740.

[12] Bhupendra Fataniya, Mekhala Kar, Grishma Joshi, Dr. Tanish Zaveri and Dr. Sanjeev Acharya. Edge Detection of Microscopic Image, International Journal of Electronics and Communication Engineering & Technology, 7 (3), 2016, pp. 01 – 10.

[13] Reena Jangra and Abhishek Bhatnagar. Comparison Analysis of Sensitivity of Noise B/W Various Edge Detection Technique by Estimating Their PSNR Value. International Journal of Computer Engineering and Technology, 6 (10), 2015, pp. 01 - 12 .