J. ICT Res. Appl., Vol. 13, No. 2, 2019, 133-144

Identification of Image Edge Using Quantum Canny Edge Detection Algorithm

Dini Sundani[1,*], Sigit Widiyanto[2], Yuli Karyanti[3] & Dini Tri Wardani[4]

Computer Science Faculty, Gunadarma University
Margonda Raya 100, Depok, Indonesia
*E-mail: dinisundani@staff.gunadarma.ac.id

Abstract. Identification of image edges using edge detection is done to obtain images that are sharp and clear. The selection of the edge detection algorithm will affect the result. Canny operators have an advantage compared to other edge detection operators because of their ability to detect not only strong edges but also weak edges. Until now, Canny edge detection has been done using classical computing where data are expressed in bits, 0 or 1. This paper proposes the identification of image edges using a quantum Canny edge detection algorithm, where data are expressed in the form of quantum bits (qubits). Besides 0 or 1, a value can also be 0 and 1 simultaneously so there will be many more possible values that can be obtained. There are three stages in the proposed method, namely the input image stage, the preprocessing stage, and the quantum edge detection stage. Visually, the results show that quantum Canny edge detection can detect more edges compared to classic Canny edge detection, with an average increase of 4.05%.

Keywords: *Canny; edge; edge detection; image; quantum; qubit.*

1 Introduction

The stages of image analysis in image processing are feature extraction, segmentation and classification. The main goal of extracting features is to support the identification of edges in an image object. There are various edge detection algorithms, such as Robert, Prewitt, Sobel and Canny. Each edge detection algorithm has advantages and disadvantages. The selection of the right edge detection algorithm affects the number of edges detected.

Various edge detection algorithms have been developed with the aim of identifying more edges. Among edge detection operators, Canny operators have the most optimal edge detection, because Canny operators have the ability to detect not only strong edges but also weak edges. Canny operators have several stages of edge detection that other operators do not have so they can produce clearer edges [1]. Several studies on edge detection using Canny operators have been applied to complex image objects [2]. The results showed that Canny

Received November 13th, 2018, 1st Revision April 8th, 2019, 2nd Revision July 2nd, 2019, Accepted for publication August 21st, 2019.
 ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2019.13.2.4

operators can detect edges optimally because they have high generalization performance and moreover they are multi-step algorithms that can detect edges and suppress noise at the same time.

The development of edge detection algorithms that exist today has mostly been done using classical computational models, where data are expressed in the form of bits that either have the value 0 or 1. Computational models are now evolving towards quantum computing. Such models have a probability property, where data not only contain the value 0 or 1 but can also be 0 and 1 at the same time, so there are many more possible values that can be obtained. With so many possibilities, a measurement is needed to stop the qubit value from becoming deterministic [3]. Research related to the detection of Sobel's quantum edge has been carried out by Sundani, who showed visually that their method could better detect the edges in an image of the human eye than classical computational models [3].

Considering the advantages of using a quantum computing model that can detect edges better, this paper proposes the identification of image edges using a quantum Canny edge detection algorithm. The development of this method was carried out on the edge tracking. It can find all possible pixels that are considered weak that have the potential to be edges by using probability functions that connect weak pixels to strong pixels. Thus, the edges obtained are sharp and clear, and as close as possible to the actual edges. The results of the study were compared with classic Canny edge detection.

2 Related Work

2.1 Edge Detection

Image edges are generally defined as the boundary between an object and a background image or the boundary between two objects with a strong intensity contrast (there is a significant change of intensity or color between one pixel and another pixel). Edge detection is the main stage used in image processing for object identification [4]. Various algorithms or edge detection methods have been developed to get clear and perfect edges. The edge detection processes are grouped according to the first and second derivatives, as shown in Figure 1.

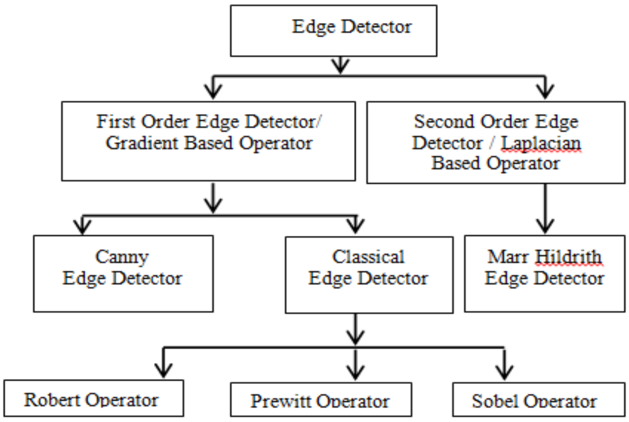


Figure 1 Edge detection classification [1].

Canny edge detection can produce clearer and more perfect edges compared to other edge detection operators (Sobel, Prewitt, Robert). This method uses the Gaussian Derivative Kernel to filter noise from the image to get smooth edge detection results. It consists of two threshold elements, namely a low and a high threshold [5]. The Canny method uses algorithms with several stages to detect edges in the image. The algorithm has a low error rate, localizes edge points (distance of edge pixels) very close to the actual edge, and only gives one response for one edge. The Canny algorithm consists of [1]:

1. *Smoothing*
   This is a process of blurring images to eliminate noise. This process is done using a Gaussian filter.
2. *Finding gradients*
   This process is carried out to determine the edge strength with a change in gray intensity that is maximum.
3. *Non-maximum suppression*
   This process is carried out to determine the edge pixel with the position closest to the location where the pixel value changes between a number of edge pixels detected.
4. *Hysteresis thresholding*
   This process is done to track the remaining pixels by using two threshold values.

2.2 Quantum

Quantum computing is computation based on quantum mechanical properties to operate data, as opposed to classical computing (digital computing). In classical computing, a state can only be in two conditions, expressed in bits, namely 0 or 1.

In classical computing, the information on an event (state) is in the form of the following equations:

$$P(0\ or\ 1) = P(0) + P(1) = 1 \tag{1}$$

P (0) is the probability of state 0 and P (1) is the probability of state 1. If state 0 has a probability equal to $\frac{1}{2}$, then state 1 has a probability equal to $\frac{1}{2}$, so the probability value of state 0 and state 1 is equal to 1, $P(0\ or\ 1) = \frac{1}{2} + \frac{1}{2} = 1$. The position and speed of the classical system are precisely defined for each location and time (deterministic). In quantum equations, bits are expressed in quantum bits (qubits), which can be in more than two states, 0 or 1, namely 0 and 1 simultaneously. Before quantum measurements have been carried out, qubits are 0 and 1 at the same time (probabilistic) [3]. Figure 2 shows the qubit representation for states 0, 1, and 0 and 1.

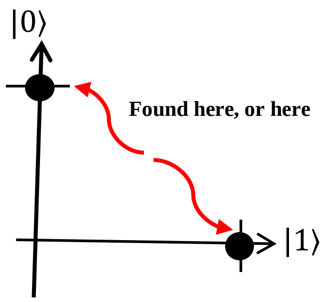


Figure 2 Qubit representation (reality_quantum_entanglement.asp).

The quantum states of a qubit can be modeled mathematically as a linear combination:

$$|\Psi\rangle = \psi|0\rangle + \psi|1\rangle \tag{2}$$

$|\Psi\rangle$ = the total state of the quantum system, which is a function of each state ψ_0 and ψ_1, which represent state 0 and state 1 respectively.

Quantum computing is unique in the process of extracting information from an event, namely by squaring the price of the function of the states [9]. Therefore, if condition 0 has probability equal to $\frac{1}{2}$ then the probability of state 0 is the square of the function of state 0, written as:

$$P(0) = |\psi_0|^2 = \frac{1}{2} \tag{3}$$

so that $\psi_0 = \frac{1}{\sqrt{2}}$ (4)

The same equation applies to the probability of state 1. Based on Eq. (4), the states of a quantum system are obtained as:

$$\Psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \tag{5}$$

To get a certain value, a measurement process for qubits is needed. This action stops the qubit process and forces the system to choose one out of all possible answers [3]. A random number is generated, which limits the measurement result to only one state [3].

3 Proposed Method

The proposed stages of research are illustrated in the following chart.

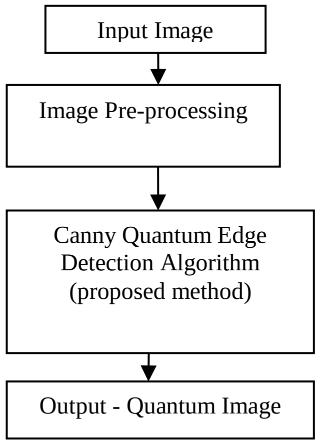


Figure 3 Proposed stages of the research.

3.1 Input Images

The input images that were used as research objects consisted of a number of images that contain complex edges. In this study, the objects used were images with butterflies that have complex wing edges.

Figure 4 Butterfly with complex edges.

3.2 Image Pre-Processing

After selecting the input image, pre-processing of the image is carried out, which consists of image transformation and image enhancement. The transformation process is carried out to change the colors of the input image to grayscale. The image enhancement aims to eliminate noise by using a Gaussian filter. The following is the Gaussian filter used:

$$\frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 21 \end{bmatrix}$$

Figure 5 Gaussian filter.

3.3 Quantum Canny Edge Detection Algorithm

This stage is the development of the proposed method, namely the development of the Quantum Canny Edge Detection Algorithm with the following steps:

1. Gradient magnitude (edge strength)

This stage aims to get point localization of the edge to determine the gradient m (Gm) by doing a gradient search horizontally (Gx) and vertically (Gy) [6]. In this study, point localization is expressed in pixels. The values of Gx, Gy and Gm are described in the following equation:

$$\mathrm{Gx} = (\mathrm{G}(\mathrm{m},\mathrm{n}+1) - \mathrm{G}(\mathrm{m},\mathrm{n}) + \mathrm{G}(\mathrm{m}+1,\mathrm{n}+1) - \mathrm{G}(\mathrm{m}+1,\mathrm{n}))/2 \qquad (6)$$

$$\mathrm{Gy} = \big(\mathrm{G}(\mathrm{m},\mathrm{n}) - \mathrm{G}(\mathrm{m}+1,\mathrm{n}) + \mathrm{G}(\mathrm{m},\mathrm{n}+1) - \mathrm{G}(\mathrm{m}+1,\mathrm{n}+1)\big)/2 \qquad (7)$$

where m, n are the rows and columns of the image in matrix form, which is expressed in pixels as:

$$\mathrm{Gm} = \sqrt{\mathrm{Gx}^2 + \mathrm{Gy}^2} \qquad (8)$$

2. Non-maximum supression.

This stage is done to obtain sharp (strong) edges from the maximum pixel value that has the closest position to the location of the change in pixel value of the detected pixel.

3. Edge tracking by quantum method

In the classic Canny method, edge determination is not only obtained from nonmaximum suppression values, but also by tracing weak pixels and connecting weak pixels to strong pixels (edge tracking by hysteresis) so that the edge is as close as possible to the actual edge. For this stage a weak pixel search was developed using quantum principles. First, all possibilities (probabilities) of pixels are found that are considered weak and have potential as edges, using Eq. (9):

$$f(Gm) = \frac{1}{1+e^{-(Gm-a)/b}} \tag{9}$$

f(Gm) is the search result on a weak edge connected with a strong edge. The values of a and b used in this study were the values 8 and 0.25. From the probability function obtained, there will be many possibilities for weak pixels that have potential to be edges. The probability of an edge is expressed as p1 and expressed in qubit form using Eq. (9):

$$p1 = \sqrt{f(Gm(m,n)} \tag{10}$$

Next is the stage of measuring the probability of weak pixels. This stage is done to change the probalistic form into a deterministic form to get the exact value of the weak pixels has potential as edges. At this stage, Z random numbers are generated in the range of 0-1. If the value of Z obtained is within [0, p1] then the measurement result is expressed as *edge*, and if it is within [p1,1] then it is expressed *not edge*. The following is the qubit measurement algorithm:

a. Generate random number R
b. If (Z < p_1 and Z >= 0) then Q = 1 else Q = 0
   Q is the edge result of quantum edge detection.

4 Result and Discussion

Twenty butterfly images were used as input. The selection of the images was based on the fact that butterfly wings have complex edges. Each input image was processed according to the proposed method, producing images with detected edges. Table 1 shows the visual results of classic edge detection and quantum edge detection. The difference is that there are edges that can be seen in the results of quantum edge detection that were not detected by classic edge detection. This difference can be seen more clearly in Figure 6.

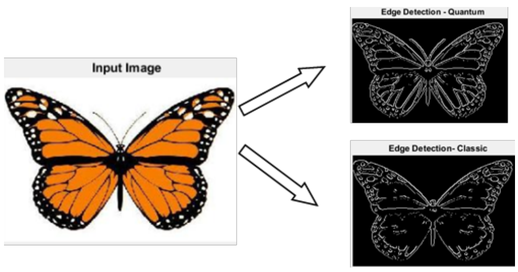
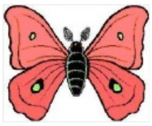


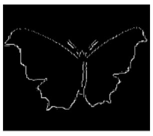
Figure 6 Input image with quantum and classic edge detection.

Table 1 consist of 20 butterfly images called *image1* to *image20.* It can be seen that quantum Canny edge detection can detect more edges compared to classic Canny based on the number of edges produced. Image size shows how many pixels there are in the image. If each pixel expressed as an edge has the value 1, then the number of edges detected can be calculated based on the number of values 1 contained in the image. *Image1* has a 215 x 293 pixel citral size; classic Canny edge detection detected 2760 edges while quantum Canny edge detection detected 5473 edges.

Table 1 Result of quantum and classic canny edge detection.

Input Image	Classic Edge Detection	Quantum Edge Detection	Image Size	Number of Edges Classic Detection	Number of Edges Quantum Detection
			(215x293)	2760	5473
			(233x293)	4498	6105
			(152x200)	1647	3165
			(191x293)	2095	3171
			(264x293)	1664	3074
			(271x293)	2410	5771

Input Image	Classic Edge Detection	Quantum Edge Detection	Image Size	Number of Edges Classic Detection	Number of Edges Quantum Detection
			(248x293)	4088	6165
			(219x293)	3195	7765
			(270x293)	1382	4515
			(206x293)	160	5699
			(168x255)	1784	3499
			(211x293)	931	3128
			(220x293)	1077	2127
			(214x293)	3263	6043
			(200x266)	1224	2895

Input Image	Classic Edge Detection	Quantum Edge Detection	Image Size	Number of Edges Classic Detection	Number of Edges Quantum Detection
			(275x292)	3663	6050
			(218x293)	1415	4812
			(166x247)	1913	3670
			(182x293)	1546	4938
			(205x242)	894	3092

Based on these results, quantum edge detection was able to detect more edges compared to classic edge detection because edge determination in quantum edge detection is based on the probability value of weak pixels that have potential to be edges. The percentages with which the number of edges increased between classic and quantum Canny edge detection are shown in Table 2, which was calculated as follows :

$$\frac{\mathrm{Nq-Nc}}{\mathrm{Si}} * 100\%$$

where N_q, N_c are the number of edges detected with quantum edge detection and the number of edges detected with classic edge detection, respectively, and S_i is the size of the image. For example, in Table 2 under *image1*, the following percentage increase in the number of edges from Canny classic edge detection to Canny quantum edge detection can been seen:

$$\frac{5473-2760}{62995} * 100\% = 4.31\%$$

Table 2 Percentage increase of edge.

Image	Size of Edge	Percentage increase of Edge
Image1	(215x293)	4.31
Image2	(233x293)	2.35
Image3	(152x200)	2.69
Image4	(191x293)	1.92
Image5	(264x293)	1.82
Image6	(271x293)	4.23
Image6	(248x293)	2.85
Image8	(219x293)	7.12
Image9	(270x293)	3.96
Image10	(206x293)	9.17
Image11	(168x255)	4.00
Image12	(211x293)	3.55
Image13	(220x293)	1.63
Image14	(214x293)	4.43
Image15	(200x266)	3.14
Image16	(275x292)	3.34
Image17	(218x293)	5.32
Image18	(166x247)	4.29
Image19	(182x293)	6.36
Image20	(205x242)	4.43
Total percentage		**80.91**

The average percentage increase in the number of edges from the 20 input images was calculated as follows:

$$\frac{80.91}{20} = 4.05\%$$

Based on the percentage results, it can be seen that quantum edge detection could detect more edges compared to classic edge detection, with an average increase of 4.05%.

The development of quantum Canny edge detection in edge tracking can find possible pixels that are considered weak and that have potential to be edges by using probability functions that connect weak pixels to strong pixels. Thus, the edges obtained are sharp and clear, and as close as possible to the actual edges.

5 Conclusion

A new edge detection method was proposed, namely quantum Canny edge detection algorithm. The proposed method performed better edge detection compared to classic Canny edge detection by detecting more edges than classic edge detection based on the number of edges produced, with an average increase of 4.05%. The method can detect edges better in complex images with many edges. In a future research, this method could be further developed for medical purposes such as edge detection in X-rays and lung or breast-USGs.

Acknowledgements

We would like to thank the Indonesian Ministry of Research, Technology and Higher Education (Ristekdikti) for the research grant scholarships 2018. Furthermore, we would also like to thank Gunadarma University for providing the laboratory facilities for data analysis and discussion.

References

[1] Rashmi, Kumar, M., & Saxena, R., *Algorithm and Technique on Various Edge Detection: A Survey*, Signal & Image Processing: An International Journal (SIPIJ), **4**(3), pp. 65-75, 2013.

[2] Jadwa, S.K., *Canny Edge Detection Method for Medical Image*, International Journal of Scientific Engineering and Applied Science (IJSEAS), **2**(8), pp. 54-59, 2016.

[3] Sundani, D., Mutiara, A.B., Juarna, A. & Agushinta, D., *Edge Detection Algorithm for Color Image Based on Principle of Quantum Superposition*, Journal of Theoretical and Applied Information Technology (JATIT), **76**(2), pp. 152-159, 2015.

[4] Reddy, D.N., Mohan, A.R. & Bhat, S., *Canny Edge Detection using Verilog*, International Journal of Engineering Sciences & Technology, **3**(6), pp. 256-260, 2014.

[5] Gupta, Y.K., *A Novel Approach with Various Existing Edge Detection Techniques for Medical Imaging Modalities*, Journal of Advanced Computing and Communication Technologies, **5**(3), pp. 67-71, 2017.

[6] Sulaeman, D., Nugroho, A.S., & Galinium, M., *Iris Segmentation using Gradient Magnitude and Fourier Descriptor for Multimodal Biometric Authentication System*, J. ICT Res. Appl., **10**(3), pp. 209-227, 2016.