Perspectives in Communication, Embedded-Systems and Signal-Processing (PiCES) – An International Journal
ISSN: 2566-932X, Vol. 2, Issue 4, July 2018
Proceedings of National Conference on Emerging Trends in VLSI, Embedded and Networking (NC-EVEN 18), April 2018

A Survey on Improved Canny-Edge Detection Algorithm

Mohammed Sabah
2nd Semester M.Tech (Electronics), Sir. M Visvesvaraya Institute of Technology, Krishnadeveraya Nagar, Hunasamaranahalli, International Airport Road, Via-Yelahanka, Bangalore-562157, msabah38@gmail.com

R. Sundaraguru
Professor & Head, Department of Electronics and communication Engineering, Sir .M Visvesvaraya Institute of Technology, Krishnadeveraya Nagar, Hunasamaranahalli, Bangalore, sugursg@gmail.com

***Abstract:** In this paper we propose to study various papers and article which are based on edge detection algorithm and analyze each one of them. Edge detection is a very important technique in image Processing. Edge detection is used to detect the boundaries of objects within images. It works based on principle of detecting discontinuities in an image. Edge detection is used for image segmentation and data extraction in domains such as image processing, computer vision, and machine vision. There are various algorithms existing in order to perform edge detection such as Robert, Prewitt, Sobel, Zero-Cross, and Marr-Hildreth. All these are suitable for different conditions and situation. Out of all canny edge detection has the capability to work and is adaptable to various environmental conditions. After John Canny introduced a canny algorithm, many problems were faced during real-time implementation in FPGA/CPLD, hence many research and finding were done by many scholars and they came up with an improved canny algorithm which solved the complexity of the algorithm and helped in implementing real-time applications.*

***Keywords:** Canny edge detection; FPGA; Real-time application*

I. INTRODUCTION

The domain area for this project is Image Processing. Image processing is mathematical manipulation of image pixel values in order to obtain fruitful information which could be utilized by user or machine. Edge detection comes beneath image segmentation block which is a principal tool in image processing, computer vision, and machine vision, especially in the zones of feature detection and feature extraction. Edge detection incorporates an assortment of scientific strategies that point at distinguishing focuses in a digital image at which the image brightness changes strongly or, more formally, has discontinuities. The issue of discovering discontinuities in 1-D signals is known as step location and the issue of finding signal discontinuities over time is known as change detection. The reason of distinguishing sharp changes in image brightness is to capture crucial events and changes in properties of the real world. Applying an edge detection calculation to an image may basically diminish the whole of data to be arranged and may be along these lines, channel out information that may be regarded as less germane, while securing the imperative auxiliary properties of an image. Most of the shape information of an image is encased in edges. So to start with we distinguish these edges in an image and by utilizing these channels and at that point by overhauling those zones of image which contains edges, sharpness of the image will increase and image will ended up clearer.

Classification of edge detection is as shown in fig (1).

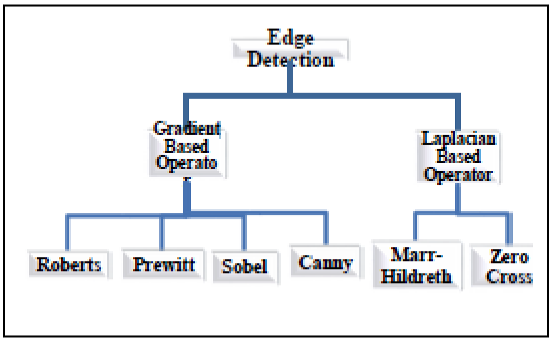

Fig 1. Classification of edge detection

II. LITERATURE SURVEY

In paper [1] all the various edge detection techniques are compared using both subjective and objective analysis, subjective analysis refers to how human perceives an image, different person may perceive differently, for example a flower may be beautiful to one person but may not be beautiful to another person, hence such subjective analysis always have contradictions, but when majority of people consider an image as good, it is wholly considered as good image. Another way of analysis is objective analysis wherein mathematical models and formulas are employed in order to compare various techniques.

From fig (2) we come to conclusion that if we use canny edge detection algorithm we are able to get more number of sharp edges even at the corner, hence canny stands first in subjective analysis.

By objective analysis, we use three parameters (i) Mean square error (MSE) (ii) Root mean square error (RMSE) (iii) Peak signal to noise (PSNR). Hence the result obtained is that for sobel operator we get more PSNR value and for Robert we get less number of PSNR, more the value of PSNR refers to high quality image. Canny produces moderate quality images.

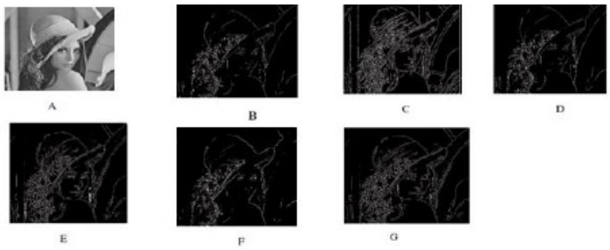

Fig 2. Subjective analysis for all edge detection operator (a) original (b) Sobel (c) Canny (d) Prewitt (e) Log (f) Robert (g) Zero-cross images.

The formulas of MSE, PSNR, RMSE are as follows.

$$MSE = \frac{\Sigma_{M,N}[I1(m,n)-I2(m,n)]^2}{M\ X\ N} \text{...................(1)}$$

$$RMS = \sqrt{MSE} \text{...........................(2)}$$

$$PSNR = \frac{10\ X\ \log(\frac{M\ X\ N}{MSE})}{\log(10)} \text{.....................(3)}$$

Table 1. MSE Comparison Values

	Sobel	Canny	Prewitt	LOG	Roberts	ZC
Sobel	0	0.0543	2.4033e^-04	0.0221	0.0019	0.0221
Canny	0.0543	0	0.0545	0.0322	0.0563	0.0322
Prewitt	2.4033e^-04	0.0545	0	0.0223	0.0017	0.0223
LOG	0.0221	0.0322	0.0223	0	0.024	0
Roberts	0.0019	0.0563	0.0017	0.024	0	0.024
ZC	0.0221	0.0322	0.0223	0	0.024	0

Table 2. RMSE Comparison Values

	Sobel	Canny	Prewitt	LOG	Roberts	ZC
Sobel	0	0.233	0.0155	0.1486	0.0441	0.1486
Canny	0.233	0	0.2336	0.1795	0.2372	0.1795
Prewitt	0.0155	0.2336	0	0.1494	0.0413	0.1494
LOG	0.1486	0.1795	0.1494	0	0.155	0
Roberts	0.0441	0.2372	0.0413	0.155	0	0.155
ZC	0.1486	0.1795	0.1494	0	0.155	0

Table 3. PSNR Comparison Values

	Sobel	Canny	Prewitt	LOG	Roberts	ZC
Sobel	0	66.8369	90.3774	70.7455	81.2951	70.7455
Canny	66.8369	0	66.8177	69.1033	66.6814	69.1033
Prewitt	90.3774	66.8177	0	70.6985	81.86777	70.6985
LOG	70.7455	69.1033	70.6985	0	70.3768	∞
Roberts	81.2951	66.6814	81.8677	70.3788	0	70.3788
ZC	70.7455	69.1033	70.6985	∞	70.3788	0

In paper [2] which has been written John Canny, who is the brain child for the algorithm known as canny algorithm, has deeply explained about his algorithm, wherein the key objectives of canny edge detection are

1. Low error rate of detection.
2. Localization of edges.
3. Single response

There are five basic steps required to perform canny edge detection which are as follows.

Step [1] Colour conversion

Step [2] Gaussian smoothing

Step [3] Calculation of image gradient

Step [4] Non-maxima suppression of gradient value.

Step [5] Edge determination & connection.

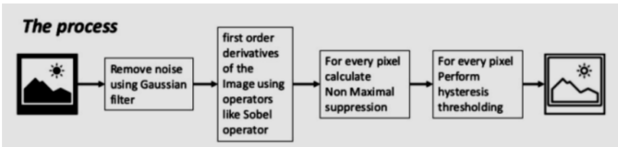

Fig 3. Canny Edge detection process

In paper [3], the author implemented traditional canny algorithm directly on Altera FPGA board and connected 5MP terasic camera module and implemented edge detection. The results what they obtained as astonishing as delay found was very negligible, and hence we could conclude that even if traditional canny is directly implemented it doesn't cause much noticeable delays. But for those hard-real time systems we could design the system with no delay.

The hardware implementation has been shown in fig (4).

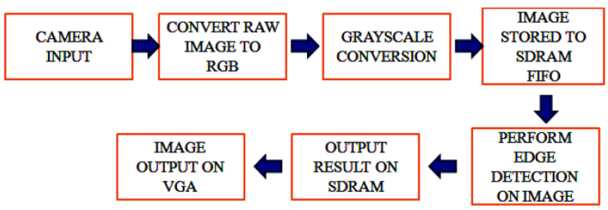

Fig 4. Hardware Implementation

Although the traditional canny algorithm is quite complex hence requires more memory space requirements but the error rate is very minimal.

In paper [4] the author has tried to improve the existing traditional canny edge detection methodology by replacing the traditional Gaussian filter with discrete cosine transform in order to obtain denoising effect.

Here by using discrete cosine transform [DCT] we transform the 2D discrete pixels to continuous DCT domain, through DCT transformation to get the DCT coefficients and for estimating noises, and then we make corrections of DCT coefficient so that we can obtain the smooth image by IDCT.

The subjective analysis has been done by comparing this algorithm with traditional algorithm which has been depicted in fig (5).

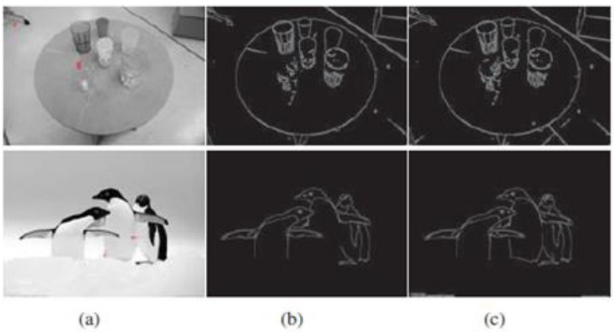

Fig 5. (a) Original image (b) Traditional Canny (c) Proposed Canny

As seen from the above figure we can compare the traditional canny algorithm image with the proposed canny image, where we find that the edges has been detected very well in the proposed canny algorithm. Thus the author concludes that by using this technique, we could obtain more continuous edges.

The author has also used objective analysis to compare both methodologies by using the parameters such as

1. Total edge point number.
2. Peak signal to noise ratio [PSNR]
3. Bit error rate.

Table 4. Total edge point numbers of traditional and proposed methods

	Low-threshold	High-threshold	Traditional edge pixels	Proposed edge pixels	ξ
Desk image	33	89	9466	10495	30
Penguins image	40	120	6329	9513	60

Table 5. PSNR and bit error rate of traditional and proposed methods

image	Traditional PSNR	Proposed PSNR	Traditional Bit-error	Proposed Bit-error
Desk image	40.51	45.29	0.027	0.026
Penguins image	41.05	44.60	0.024	0.022

The more the edge points better is the performance of edge detection. Hence the proposed algorithm has more number of edge points than traditional canny. Similarly more the PSNR value and lower the bit rate error better is the quality of the images.

In paper [5] again some improvement have been proposed in the traditional canny edge detection algorithm wherein the traditional guassian filter have been replaced by an adaptive filter.

The benefit of using adaptive filter is that it selects accordingly based on the feature of the gray values of images.

With the help of adaptive filter we get refined edge points and single pixel level edges especially at corner edges which cannot be obtained from traditional canny algorithm.

Here also the author has performed subjective analysis and compared the proposed methodology with the traditional one.

Before applying the edge detection algorithm, the author first adds some noise to the image as shown in fig (6).

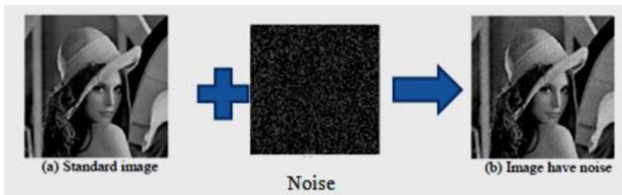

Fig 6. Adding noise to image

After which canny edge algorithm is applied and the result of traditional canny and proposed canny algorithm is seen as shown in fig (7).

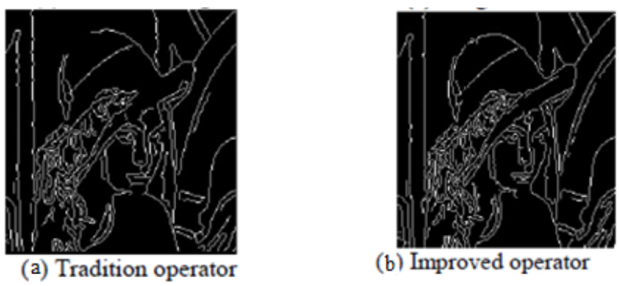

Fig 7. Comparison of traditional and improved canny

As we can see in fig (7) near the cap, the region is discontinuous in traditional operator whereas the region is neatly covered by using improved operator.

By objective analysis if we compare traditional canny and improved canny for 4 neighbor and 8 neighbor we get

Perspectives in Communication, Embedded-Systems and Signal-Processing (PiCES) – An International Journal
ISSN: 2566-932X, Vol. 2, Issue 4, July 2018
Proceedings of National Conference on Emerging Trends in VLSI, Embedded and Networking (NC-EVEN 18), April 2018

more connectivity in improved algorithm at the cost of time consumption as depicted by table (6).

	Time	A	B	B/A
Tradition algorithm	57	8263	843	0.1021
Improved algorithm	63	9027	897	0.0994

Table 6. Time consumption comparison

Thus we can see that the improved algorithm consumes more time than the traditional algorithm, thus it is one of the limitation of this algorithm.

In paper [6] the author tries to modify the usage of guassian filter with a single value of σ for the whole image.

The equation of guassian filter is given as

$$h(x,y)=exp\ (-\pi(x^2+y^2)/\sigma)\text{---------------}(4)$$

The value of σ and threshold is fixed in traditional canny algorithm, whereas in the proposed methodology the value of σ and threshold are changed and varied, and for different parts of image different values of $\sigma$and threshold are used.

Hence the first step in the proposed method is to divide the images into sub-images, and then for each sub-image mean pixel value is calculated, and then depending on the mean pixel value the value of σ and threshold are set for each sub-image.

Hence by using this technique the author says that the entropy is increased. Entropy is nothing but amount of information in an image.

The limitation of this technique is that dividing the image into different sub-image and then applying the algorithm may lead to more time consumption and delay.

Subjective analysis shown in fig (8) depicts that by using variable value of sigma we are able to get proper edges with zero error distortions.

For objective analysis entropy calculation is performed, Entropy is defined as measure of average uncertainty and is given by the formula:

$$H = -\Sigma\, p\ log_2(p)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (5)$$

where p is the histogram count.

Lowest entropy means more information.

In paper [7] they have proposed to minimize the computational complexity and high hardware cost by making use of pipelining architecture, approximation method and sliding windows techniques to compute high and low threshold values.

Here the ways in which image gradients are computed are altered in order to reduce complexity.

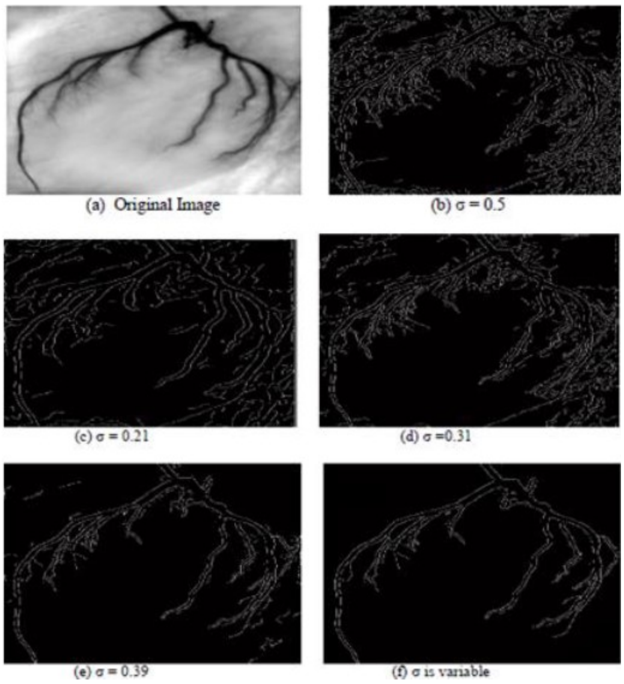

Fig 8. Angiography image with fixed and variable value of sigma and threshold

Image	Image Entropy (H)
Original image	5.6599
σ=0.5	4.0351
σ=0.21	3.9420
σ=0.31	3.5035
σ=0.39	3.3720
Proposed method	3.1581

Table 7. Entropy calculation

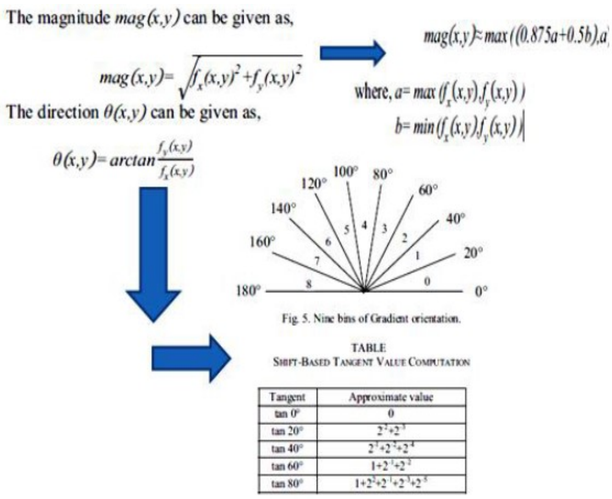

Fig 9. Magnitude and direction computation

By using such techniques we can achieve edge detection on 512 X 512 image in within 1ms.

In paper [8], the author has implemented the concept developed by paper [7] in FPGA vertex-5 board and the results obtained are appreciable. Hence we can say that by using pipelining architecture we can obtain accurate results within less time.

The subjective analysis compares images of iris edge map by both conventional canny algorithm and the proposed canny algorithm. As seen in fig 10 we get proper edges in image where proposed canny algorithm is applied.

The objective analysis based on time consumption is also shown in table (8).

As depicted in table (8) the proposed algorithm requires minimum time and slices for computation of a 512 X 512 image.

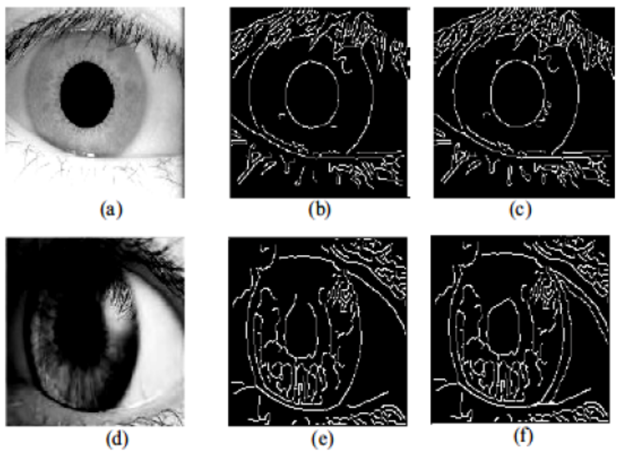

Fig 10. (a),(d) 512 X 512 Iris Image; (b),(c) edge map using the conventional canny edge detector [5]; (c), (f) edge map of the proposed canny edge detector

Edge Detection Methods	Image Size	FPGA Device	Used Slices	Norm. Time (ms)
[7]	256x256	Xilinx Vertex-E	-	2.688
[8]	256x256	Altera Stratix II	1530/48532	2.64
[9]	512x512	Xilinx Virtex-5	4553/71680	1.669
[10]	360x280	Altera Cyclone	-	0.72
[11]	512x512	Xilinx Virtex-5	-	0.47
[12]	512x512	Xilinx Virtex-5	23904/37440	0.372
Proposed	512x512	Xilinx Virtex-5	5980/28495	0.456

Table 8. Comparison of resource utilization and time

In paper [9], the author suggests replacing the method for calculation of low and high threshold for determining edges, which is a new self-adapt threshold calculation, is proposed in which it automatically sets the threshold based on the environment and illumination condition.

In order to do well in real-time they have used pipeline architecture for high speed process. The total time consumption in FPGA based environment is 2.5ms for detection 360 X 280 size gray scale image.

In paper [10] it suggests to replace the guassian filter by an improved median filter so that the performance and reliability of the system is improved when dealing with noise.

A shifting look up table [LUT] based direction calculation is proposed to improve the processing speed and to reduce hardware consumption.

By subjective analysis as shown in fig (11), (12), (13) we see that in fig (11) images are applied with noises with 1%, 5%, 10%. Fig (12) shows canny edge detector without median filtering and fig (13) shows canny edge detector with median filtering.

Thus even in noisy condition, output comes out very well. The time required for computation is 5.24ms which is very high.

Fig 11. Original image contaminated by salt and pepper noise with an intensity of: (a) One percent; (b) Five percent (c) Ten percent

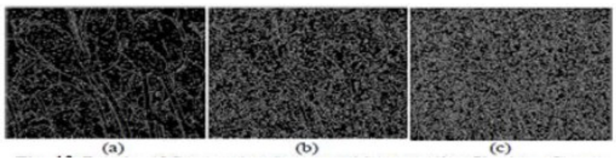

Fig 12. Results of Canny edge detector without median filtering. Canny edge with (a) 1% (b) 5% (c) 10% salt and pepper noise respectively

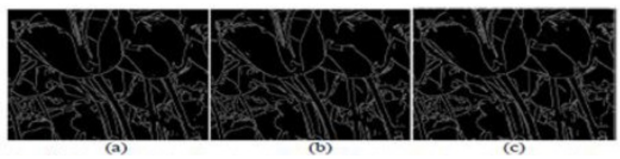

Fig 13. Results of Canny edge detector with improved median filter. Canny edge with (a) 1% (b) 5% (c) 10% salt and pepper noise respectively

III. COMPARISON

In paper [1] it compares all edge detection techniques and concludes that canny edge detection algorithm is the best of all other algorithm. In paper [2] canny himself elaborates his algorithm and the estimated time for completion is 5.7ms. In paper[3] author has implemented traditional canny algorithm in real-time on an FPGA board. In paper [4] guassian filter is replaced by DCT transformation, whereas in paper [5] guassian filter uses different values of sigma and threshold for each sub-images. In paper [6] guassian filter is replaced with adaptive filter which gives output within 6.3ms. In paper [7] & [8] pipelining architecture, sliding windows and approximation methods are used, which help in achieving output within 1ms for 512 X 512 image. In paper [9] the thresholding calculation is replaced with self adapting threshold and by the use of pipelining which gives output at 2.5ms for 360 X 280 gray scale image. Paper [10] replaces guassian filter by improved median filter and shifting LUT which gives output at 5.24ms.

IV. CONCLUSION

At last we would like to conclude by stating that each method has its own advantages, and depends on which situation and condition we would like to implement. For real time application we would like to have the algorithm which gives good accurate and sharp image in a shortest period, which paper [7] has done by using pipelining architecture.

REFERENCES

[1] S.Simranjit and S.Rakesh, “Comparision of various edge detection techniques” 2nd International conference on computing for sustainable global development (INDIACom), 2015.

[2] J.F Canny, “A computational approach to edge detection” IEEE Trans. Pattern Anal. Machine Intell, vol. PAMI-8 no.6, pp.679-697,1986.

[3] R.Jeyakumar, M.Prakash, S.Sivanantham, and K.Sivasankaran, “FPGA implementation of edge detection using canny algorithm” Online International Conference on Green Engineering and Technologies (IC-GET 2015).

[4] Z.Miaomiao, L.Hongxia, and W.Yi, “An Improved canny edge detection algorithm based on DCT” IEEE 2015.

[5] W.Bing and F.ShaoSheng, “An Improved canny edge detection algorithm” Second International workshop on computer science and engineering 2009.

[6] M.H Shokhan, “An efficient approach for improving canny edge detection algorithm” International journal of advances in engineering & technology, March 2014.

[7] D.Sangeetha and P.Deepa, “An efficient hardware implementation of canny edge detection algorithm” 29th International conference on VLSI design and 2016 15th Internal conference on embedded system 2016.

[8] S.Suman and R.Sundaraguru, “Implementation of pipelined canny edge detection algorithm”, International Journal of advance electrical and electronics engineering 2017.

[9] H. Wenhao and Y.Kui, “An improved canny edge detector and its realization on FPGA” Proceeding of 7th World Congress on intelligent control and automation 2008.

[10] L.Xiaoyang, J.Jie and F.Qiayun “An improved real-time hardware architecture for canny edge detection based on FPGA” Third International Conference on Intelligent Control and Information Processing 2012