Fast Edge Detection Algorithm for Embedded Systems

KHALID BUKHARI Syed Usama [1], BRAD Remus [2], ZAMFIRESCU Constantin Bălă [2]

[1] Faculty of Engineering, Lucian Blaga University of Sibiu, B-dul Victoriei 10, Sibiu 550024, Romania, syedusama@gmail.com

[2] Computer Science Department, Lucian Blaga University of Sibiu, B-dul Victoriei 10, Sibiu 550024, Romania, remus.brad@ulbsibiu.ro, constantin-bala.zamfirescu@ulbsibiu.ro

Abstract: The processing of images is currently moving from desktop implementation to mobile or embedded ones. In the case of automotive image processing, limited resources in memory or CPU frequency reduce the applicability of nowadays algorithms and possibility of real time processing. In this respect, we have proposed two methods for fast computation of edge detection. The accuracy and speedup were compared with some of basic methods as Canny and Sobel detectors. Also for a solid reference, the Berkeley Computer Vision Group datasets were employed as benchmarks. Good results were obtained over one hundred images from the set. In view of an embedded implementation, two platforms were used for the evaluation of the proposed methods and the references. Performing two times faster and with similar accuracy, our algorithms could have evident implementations in the growing field of embedded devices.

Keywords: edge detection; algorithm complexity; real time computation; embedded systems.

Syed Usama Khalid Bukhari[1] received his BS and MS degrees in Computer Science from Hamdard University, Pakistan in 2002 and COMSATS IIT in 2011 respectively. He is currently an Erasmus Mundus scholar doing his PhD. at Lucian Blaga University of Sibiu, Romania in Computer Vision. His current research interests include computer vision, stereo vision and augmented reality.

Remus Brad[2] received an Engineer Diploma degree in Computer Science from the Lucian Blaga University of Sibiu Romania, a M.S. degree from Université Pierre et Marie Curie Paris, France in Artificial Intelligence and a PhD. from Technical University of Cluj-Napoca, Romania. Since 1994 he has joined the Department of Computer Science and Electrical Engineering at the Lucian Blaga University of Sibiu, Romania. His current research interests include image processing, motion estimation and biomedical imaging. He is a member of the IEEE Signal Processing Society.

Constantin-Bălă Zamfirescu[2] received a DPhil in Automation from "Politehnica" University of Bucharest and a MSc in Computer Science from the Lucian Blaga University of Sibiu. He worked as invited researcher in Austria, Spain and Belgium where he participated in various EU R&D projects. Since 1997 he is associated with the Computer Science Department from the Lucian Blaga University of Sibiu, where he teaches Industrial Informatics, Multi-agent systems and Manufacturing Control courses. He is member of the IFAC TC on LSS and EU COST 298 Action. In 2003 he founded the Effective Decisions, a high-tech research and consulting company that offers independent services to both industry and government.

1. Introduction

In the future industrial applications contextual data and knowledge will be captured and provided by embedded devices that are situated in "smart environments" [1] that tightly integrates computational, physical and social elements. This complex integration is studied under a broad range of emerging concepts, such as "ubiquitous intelligent systems"[2], "Internet of Things"[3], and/or "cyber-physical systems" [4], etc.

Among many enabling technologies for non-intrusive context-aware factory automations (e.g. global or local positioning systems, barcodes, RFID, Bluetooth, NFC, etc.), nowadays computer vision techniques are playing an increasing role in maintenance and quality management (i.e. by providing on special displays real-time repairing information in an augmented reality which is created based on the identification of objects), or even manufacturing (i.e. by giving to the human operator contextualized, step-by-step instructions, on how to execute the operations). Despite their real potential for performance enhancement in industry (see [5] for an exhaustive survey), there are very few applications that have broke out of the lab settings and are regularly used [6]. Most of these augmented reality applications are employing complex hardware (i.e. special camera systems, sensors, displays and eye-tracking devices) and powerful computing systems (i.e. smartphones, tablet computers) that are cost beneficial for very limited areas. Therefore, it is well acknowledged that augmented reality will not be widely adopted by industry as long the information will not be directly sensed and processed by embedded devices that are seamlessly immersed in the environment [5], [6], without being noticed by its potential users.

Moving the image processing applications from mobile to embedded (or even wearable) devices

poses many challenging constraints in terms of limited computational capabilities and real-time processing requirements.

Edge detection is one of the basic processes related to computer vision and image processing, as edges contain a richness of information associated with image contents. Edge appears between two neighboring areas having different level of color / light intensity and draws up the boundaries between objects or objects and background.

As one of the most widely employed method of feature detection, the edge detection had a long history. Since the algorithms of Roberts [7] and Sobel [8] implemented in cellular automata, a number of new and improved methods have been developed. These techniques, including Prewitt [9], Canny [10], Laplacian of Gaussian (LOG) detectors [11], mathematical morphology [12] and wavelet transform [13], [14] were used to achieve better results.

The efficiency of algorithms is normally judged based upon correctness and speed. Classical algorithms with differential operators are easy to implement, however they are sensitive to noise [9]. Several developments have been proposed to the mentioned category in order to improve the accuracy [15]–[17], although, many other adaptive and hybrid techniques concentrate on noise reduction and the correct detection of edges [18]–[20]. A number of speed-up techniques have also been proposed in last few years, as real-time computation is considered an important factor in embedded systems applications. Therefore in [21], the LOG has been optimized to require less computation during run-time and a CUDA (Computer Unified Device Architecture) implementation of Canny algorithm offers up to 61% improvement in speed [22].

A better edge detection algorithm is required in many applications where results are crucial, for example, in the field of embedded systems, especially for automotive [27] or medical image processing purposes [28].

The paper is a step forward in this direction by investigating two methods for one of the most frequently used techniques in feature detection, requiring no kernel computation. One algorithm involves only one subtraction operation to calculate the edges, while the second uses conditional statements and does not necessitate any type of mathematical

operations. As it was proven, both methods require less computation in comparison with other well known edge detection algorithms.

The paper is organized as follows; Section 2 presents the proposed algorithms, while the experimental results and performance comparisons are presented in Section 3. The last section concludes with potential applications for the proposed methods for edge detection.

2. A novel approach for edge detection

First method

The first method exploits the concept that neighbouring pixels often have similar intensities; therefore a significant level of change in intensities among neighbouring pixel is due to the presence of an edge or a noise. To minimize the noise, a Gaussian filter can be used along with a reliable value of the threshold. The proposed algorithm finds the edges in image if the consecutive chain of similar intensities of pixels is broken. Moving from left to right, top to bottom, the algorithm finds the significant change in pixel intensities in the form of ascending and descending sorting order. A change from ascending order intensities to descending order or vice-versa will be considered an edge, as described by the pseudo code below.

$$if \begin{array}{l} \left[I(x,y)-I(x+1,y)>T\right]|| \\ \left[I(x,y)-I(x,y+1)>T\right]|| \\ \left[I(x,y)-I(x-1,y)>T\right]|| \\ \left[I(x,y)-I(x,y-1)>T\right] \end{array}$$

then Flag= des. Chain

else

$$if \begin{array}{l} \left[I(x+1,y)-I(x,y)>T\right]|| \\ \left[I(x,y+1)-I(x,y)>T\right]|| \\ \left[I(x-1,y)-I(x,y)>T\right]|| \\ \left[I(x,y-1)-I(x,y)>T\right] \end{array}$$

then Flag=as.Chain
end

if pFlag !=Flag
    pixel $I(x,y)$ is an edge
    pFlag=Flag;
end

where des.Chain is a flag marking a descending chain of pixels, while as.Chain flags an ascending chain of pixels and pFlag is a Boolean variable that stores information about the change of pixels sequence from ascending to descending or vice-versa. T is a constant used as a threshold. Since the determination of an optimal threshold for each test image conducts to extra calculations, the value of T adopted was 12 for all the experiments.

Second method

The second proposed algorithm finds the edges by subtracting the next pixel from previous one and comparing the difference with a certain threshold. If the value is bigger than T, it considers detecting an edge, as shown by equation 1.

$$\text{if } \begin{bmatrix} \left| I(x,y) - I(x+1,y) \right| > T \end{bmatrix} || \\ \begin{bmatrix} \left| I(x,y) - I(x,y+1) \right| > T \end{bmatrix}$$

then pixel $I(x,y)$ is an edge $\qquad$ (1)

The algorithm is based upon the fact that a significant change in consecutive pixel values will indicate an edge. The amount of change is adjusted using the threshold value, considered as the value 12 for all the experiments.

3. Performance comparison of the proposed methods

Execution time evaluation

Both proposed algorithms have $O(n)$ complexity and require less working memory. These advantages recommend them for embedded systems implementations. In order to demonstrate the running time performance, we have compared them with Canny and Sobel edge detectors in two typical embedded devices.

The experiments were performed on Proteus 8 Professional simulator, using an Atmel 8051 family AT80C31 and a PIC 18F1220. In the case of Atmel, an IAR compiler was used and Hi-Tech C compiler was employed for PIC18 code. The running platform was a 2GHz Core2Duo MacBook running Windows 7.

In view of a standard comparison, the segmentation benchmark dataset provided by Berkeley Computer Vision Group [26] was used in all our evaluations. The graphs display the average, minimum and maximum runtime values in seconds, for 100 images of the BSDS500 database. As it can be seen from the results in figure 1, both algorithms perform faster than Canny and Sobel on either embedded device. In the case of Atmel, Method 1 and Method 2 are 2.59, respectively 6.17 times faster than Sobel edge detector, while for the PIC device; the ratios are 2.25 and 5.27 faster. Those results have been reported to the mean values computed for the BSDS benchmark.

Accuracy evaluation

If our two novel algorithms for edge detection have a speedup of more than 2 times, we have also investigated their accuracy in performing the designated task. In this respect, several measures have been estimated, as the Mean Square Error (*MSE*) and Peak Signal to Noise Ratio (*PSNR*), using the relations in equations (2) and (3). The two compared images having *MxN* size are: $I(x,y)$ the segmented image using the first, second proposed algorithm or the referenced methods and $I'(x,y)$ the ground truth segmentation from BSDS.

$$MSE = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} (I(x,y) - I'(x,y))^2 \qquad (2)$$

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \qquad (3)$$

As a comparison rule, a large value of MSE indicates a low quality, while small values of *PSNR* are also an indication of poor results [23]. In our tests, the *MSE* values for both algorithms were averaged to 0.120 and 0.010 respectively, while the average *PSNR* values obtained were 58.08 and 68.3. Therefore, as it can be seen from figure 2, the proposed methods have obtained good results.
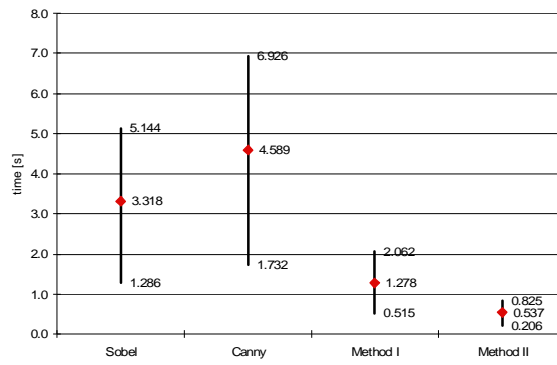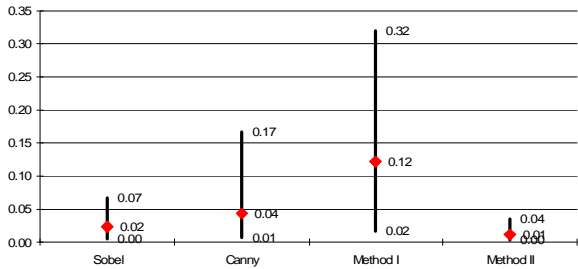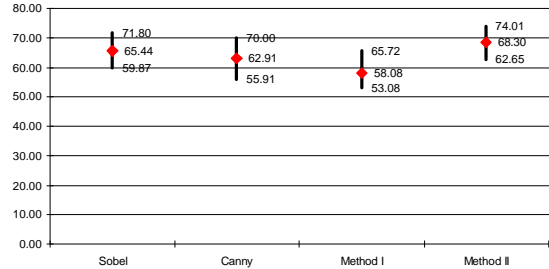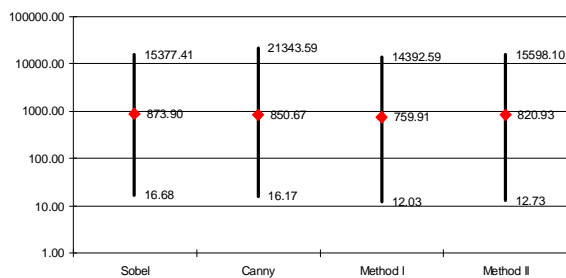
a)
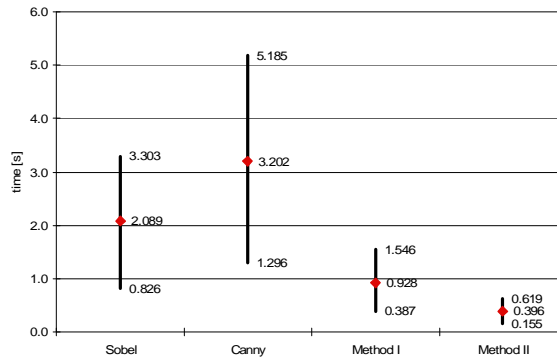
b)

Figure 1. Running time on (a) Atmel and (b) PIC devices, for Sobel, Canny and the proposed algorithms.
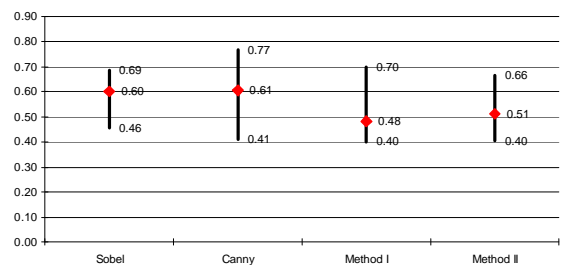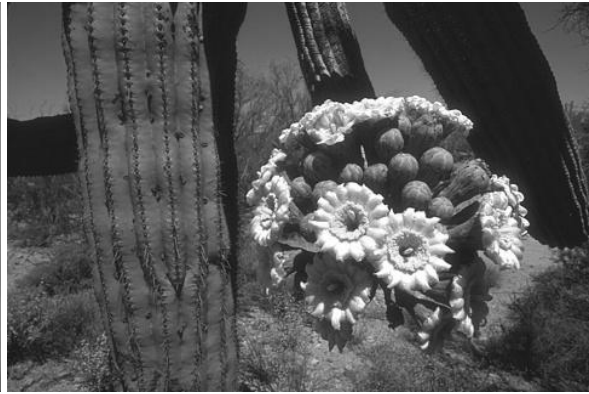

a)

b)

c)

d)

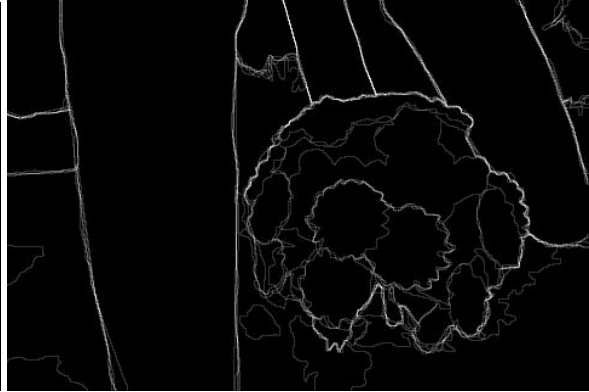**Figure 2.** Measures of edge detection accuracy; (a) MSE, (b) PSNR, (c) BDM and (d) F-measure.
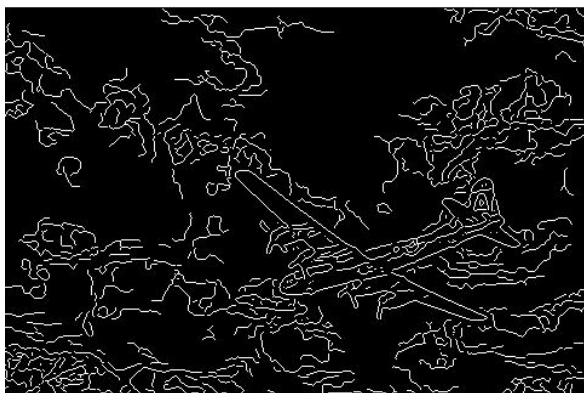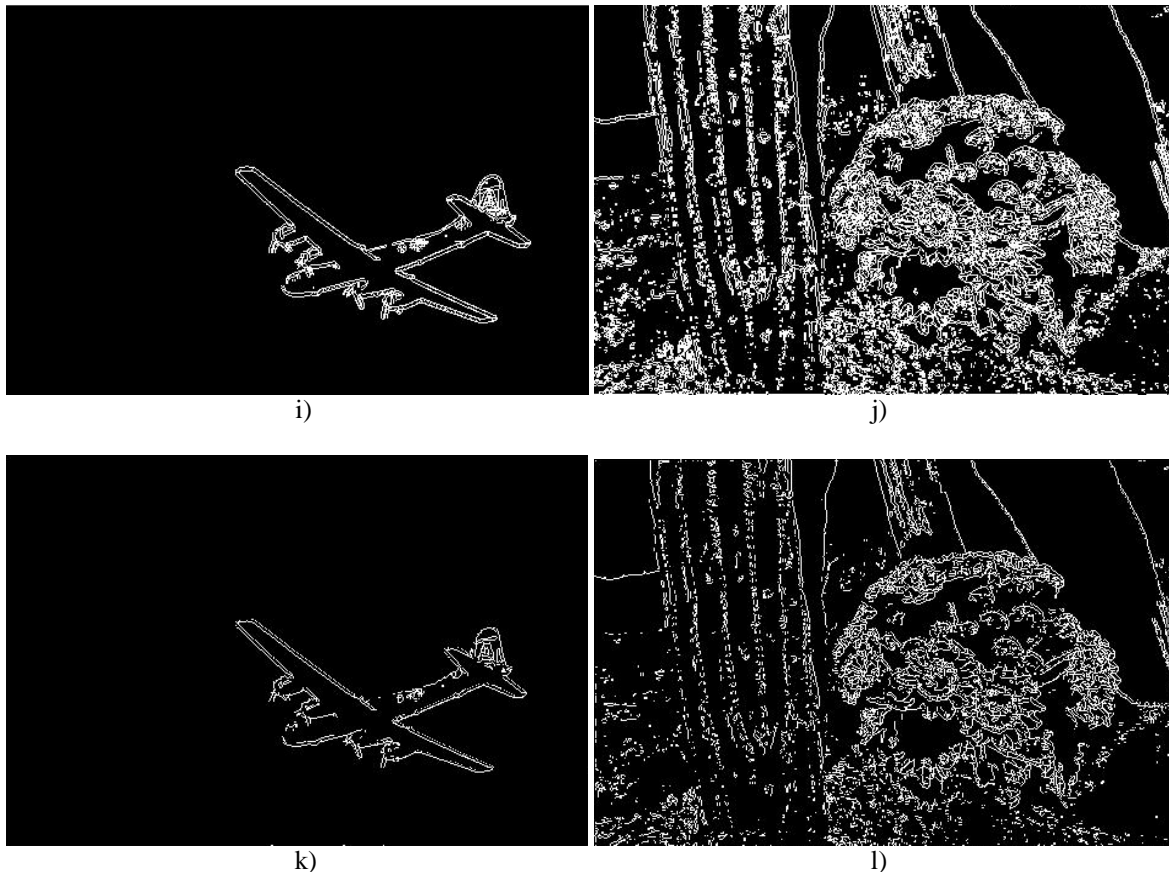
a)

b)

c)

d)

e)

f)

g)

h)

Figure 3. Segmentation results for two images of the BDSD data set (a) and (b); (c) and (d) ground truth; (e) and (f) Sobel; (g) and (h) Canny; (i) and (j) first proposed method; (k) and (l) second proposed method

Another measure used in edge detection evaluation is the Baddeley Delta Metric (*BDM*) that calculates the dissimilarity between two images, in which case, a higher value indicating a large difference [25]. Also, the authors of BSDS are measuring the edge detection by the means of F-measure, an accuracy test that takes into account the precision and recall of detected pixels [24] (equation 4).

$$precison = \frac{t_p}{t_p + f_p}, \quad recall = \frac{t_p}{t_p + f_n} \quad (4)$$

where t_p, f_p and f_n are a true positive, false positive and false negative detection. F-measure is the harmonic mean of precision and recall as in equation 5.

$$F = 2\frac{precision \cdot recall}{precision + recall} \quad (5)$$

The comparisons were performed between the detected image and the human ground truth image provided by the benchmark. The results of all considered measures are presented in figure 2. The average F-Measures of both proposed algorithms are 0.48 and 0.51, while the average BDM values are 759 and 820 respectively, close to the referenced detectors.

For every image in the benchmark, a full search algorithm was used in order to obtain the best parameters of the Canny and Sobel detectors, which have then been used for all the experiments. A visual comparison of the results obtained during our tests, for the case of two images extracted from the BDSD set, is shown in figure 3.

As it can be observed from the qualitative analysis presented above, our detectors are performing respectably well and fast.

4. Conclusions

The paper presents two novel approaches for an effective and fast edge detection algorithm, as their low complexity requires only $O(n)$ computations. The algorithms detect edges without using any convolution kernel, requiring

low mathematical operations per image pixel. Comparisons with Sobel and Canny detectors have been performed, using the Berkeley image segmentation benchmark.

The proposed edge detection algorithms are fast and accurate, recommending them to be used in lowered power systems or situations where the accuracy can be a compromise to speed. The results are showing that they manage to obtain as good results as the classical referenced one, offering evident applications in the embedded systems area.

REFERENCES

1. POSLAD, S., **Ubiquitous Computing Smart Devices, Smart Environments and Smart Interaction**, Wiley (2009).

2. REICHLE, R., WAGNER, M., KHAN, M. U., GEIHS, K., LORENZO, J., VALLA, M., FRA, C., PASPALLIS, N., PAPADOPOULOS, G. A., **A Comprehensive Context Modeling Framework for Pervasive Computing Systems**, Proceedings 8th IFIP WG 6.1 international conference on Distributed applications and interoperable systems, 281–295 (2008).

3. ATZORI, L, IERA, A., MORABITO, G., **The Internet of Things: A survey**, *Computer Networks*, 54(15), 2787–2805, (2010).

4. NIST, **Foundations for Innovation in Cyber-Physical Systems**, Workshop Report, Available at: http://www.nist.gov/el/upload/CPS-WorkshopReport-1-30-13-Final.pdf (2013).

5. MA, D., FAN, X., GAUSEMEIER, J., GRAFE, M. (Eds.), **Virtual Reality & Augmented Reality in Industry**, Springer (2011).

6. FITTE-GEORGEL, P., **Is there a reality in Industrial Augmented Reality?**, Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE, Basel (2011).

7. ROBERTS, L.G., **Machine perception of three-dimensional solids**, No. TR315, degree of Doctor of Philosophy, Massachusetts Inst. of Technology (1963).

8. SOBEL, I., **Camera models and machine perception**, No. AIM-121, Stanford University, California, Dept. of Computer Science (1970).

9. PREWITT, J. M. S., **Object enhancement and extraction**, Picture Processing and Psychopictorics, Academic Press, 75-149 (1970).

10. CANNY, J. A., **Computational approach to edge detection**, IEEE Transactions on Pattern Analysis and Machine Intelligence, 6, 679-698 (1986).

11. BASU, M., **Gaussian-based edge-detection methods-A survey**, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 32(3), 252-260 (2002).

12. LEE, J., HARALICK, R., SHAPIRO, L., **Morphologic edge detection**, IEEE Journal of Robotics and Automation, 3(2), 142-156 (1987).

13. MALLAT, S., ZHONG, S., **Characterization of signals from multiscale edges**, IEEE Transactions on Pattern Analysis and Machine Intelligence, 14, 710-732 (1992).

14. XU, P., MIAO, Q., SHI, C., ZHANG, J., LI, W., **An edge detection algorithm** based on the multi-direction shear transform, Journal of Visual Communication and Image Representation, 23(5), 827–833 (2012).

15. GAO, W., ZHANG, X., YANG, L., LIU, H., **An improved Sobel edge detection**, Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology, Chengdu, China, IEEE, 5 (2010).

16. MA, C., YANG, L., GAO, W., LIU, Z., **An improved Sobel algorithm based on median filter**, Proceedings of the 2nd International Conference on Mechanical and Electronics Engineering, Kyoto, Japan, IEEE, 1 (2010).

17. FARAHANIRAD, H., SHANBEHZADEH, J., PEDRAM, M., SARRAFZADEH, A., **A hybrid edge**

detection algorithm for salt-and-pepper noise, Proceedings of the International MultiConference of Engineeres and Computer Scientists, Hong Kong, March 16-18 (2011).

18. WANG, B., FAN, S., **An improved CANNY edge detection algorithm**, Proceedings of the Second International Workshop on Computer Science and Engineering, Xiamen, China, 1, 497-500 (2009).

19. SAHU, T. L., DUBEY, D.A., **Survey on edge detections and denoise techniques**, International Journal of Science, Engineering and Technology Research, 2(1), 160-166 (2013).

20. DAFU, P., WANG, B., **An improved Canny algorithm**, Proceedings of the 27th Chinese Control Conference, IEEE, Kunming, Yunnan Province, China, July 16-18 (2013).

21. TORRE, V., TOMASO, A. P., **On edge detection**, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2, 147-163 (1986).

22. OGAWA, K., ITO, Y., NAKANO, K., **Efficient Canny edge detection using a GPU**, Proceedings of First International Conference on Networking and Computing, IEEE, Hiroshima, Japan, November 17-19, (2010).

23. VORA, V. S., SUTHAR, A. C., MAKWANA, Y. N., DAVDA, S. J., **Analysis of Compressed Image Quality Assessments**, International Journal of Advanced Engineering & Application, I&II, 230-234 (2010).

24. ARBELAEZ, P., MAIRE, M., FOWLKES, C., MALIK, J., **Contour detection and hierarchical image segmentation**, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33, 898-916 (2011).

25. BADDELEY, A. J., **An error metric for binary images**, in Robust Computer Vision: Quality of Vision Algorithms, W. Forstner, S. Ruwiedel Eds., Wichmann Verlag, Karlsruhe, pp. 59-78 (1992).

26. MARTIN, D., FOWLKES, C., MALIK, J., **Learning to detect natural image boundaries using local brightness, color, and texture cues**, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(5), 530-549 (2004).

27. FAGADAR-COSMA, M., NOURI, M., CRETU, V.I., MICEA, M. V., **A Combined Optical Flow and Graph Cut Approach for Foreground Extraction in Videoconference Applications**, Studies in Informatics and Control, vol. 21 (4), 413-422 (2012).

28. KREJCAR, O., JIRKA, J., JANCKULIK, D., **Use of mobile phones as intelligent sensors for sound input analysis and sleep state detection**, Sensors 11 (6), 6037-6055 (2011).