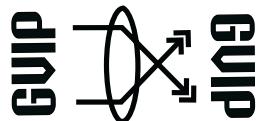


The International Congress for global Science and Technology



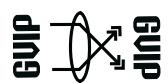
*ICGST International Journal on Graphics, Vision and
Image Processing
(GVIP)*

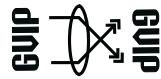
GVIP Special Issue on Edge Detection, 2007

www.icgst.com

© ICGST, 2007

GVIP Journal
ISSN Print 1687-398X
ISSN Online 1687-3998
ISSN CD-ROM 1687-4005
© ICGST 2007





***ICGST International Journal on Graphics, Vision and Image Processing
(GVIP)***

*A publication of the International Congress for global Science and Technology -
(ICGST)*

Dr. Ashraf Aboshosha

www.icgst.com. editor@icgst.com

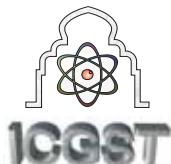
GVIP Committee

Prof. Dr.

- L. ALVAREZ LEON, Departamento de Informatica y Sistemas of Universidad de Las Palmas de Gran Canaria
- M. Bennamoun, School of Computer Science and Software Engineering , The University of Western Australia, Perth, Australia
- A. Farag, Director, Computer Vision & Image Processing Laboratory, CVIP Lab, University of Louisville, USA
- N. Alberto Borghese, Department of Information Science, University of Milan, Italy
- Christos Grecos, Dep.of Electronic and Electrical Engineering, Loughborough University, UK
- G. H. Granlund, head of the Division of Computer Vision, Department of Electrical Engineering, University of LINKOEPING
- L. Itti, iLab, University of southern California
- S. Mohammed, Department of Computer Science, Advanced Technology and Academic Centre, Lakehead University, Canada.
- P. Jonker, Quantitative Imaging Group, Department of Imaging Science and Technology, Faculty of Applied Sciences, Delft University of Technology, The Netherlands
- S.-W. Lee, Center for Artificial Vision Research, Dept. of Computer Science and Engineering, Korea University
- S. Loncaric, Head of Image Processing Group, Faculty of Electrical Engineering and Computing, University of Zagreb
- V. Roberto, Department of Mathematics and Computer Science, University of Udine
- M. Sarfraz, Information & Computer Science Dept., King Fahd University of Petroleum and Minerals, Saudi Arabia
- A. von Wangenheim, Dep. of Computer Sciences, Universidade Federal de Santa catarina, Brazil.
- M. Yeasin, Computer Science Department, Suny Institute of Technology (SunyIt)
- I. El Emari, Faculty of Engineering, Amman Al Ahliyya University, AMMAN, JORDAN

Table of Contents

Papers	Pages
P1150518004 Yacine AIT ALI YAHIA and Abderazek GUESSOUM Parametric Filtering Algorithms for Edge Detection	1--7
P1150535117 F. Moreno and S. Romero and O. Sánchez and E. Cortés Generalization of Frei and Chen's approach to extract the lines and edges of a digital images through orthogonal projections with masks of even dimension	9--14
P1150535188 Tieling Chen An Edge Detection Algorithm using Smoothing Functions	15--20
P115053524 Sang Hyun Kim A Novel Approach to Moving Edge Detection Using Cross Entropy	21--24
P1150542102 B. Bouda and Lh. Masmoudi and D. Aboutajdine A New Model for Edge Detection in Digital Images	25-30
P1150630001 B. Bouda and Lh. Masmoudi and D. Aboutajdine A New Grey Level Corner Detection Based on Electrostatic Model	31--36
P1150649004 B. Bouda and Lh. Masmoudi and D. Aboutajdine Extension of gray-level edge detection model to the RGB images based on virtual electric field	37--43
P1150636001 Raman Maini and J.S. Sohal Performance Evaluation of Prewitt Edge Detector for Noisy Images	45--52
P1150649005 Mohamed Roushdy Comparative Study of Edge Detection Algorithms Applying on the Grayscale Noisy Image Using Morphological Filter	53--59
P1150721005 Tieling Chen Embedded Smoothing Processing in Spatial Gradient Operators, A Comparative Study	61--69



Parametric Filtering Algorithms for Edge Detection

Yacine AIT ALI YAHIA*, Abderazek GUESSOUM**

* National Institute of Computer Science
INI BP 68M Oued Smar 16270, Alger, Algeria
Email: y_ait.ali.yahia@ini.dz

** Department of Electronics Engineering,
University of Blida, BP 270, Blida, Algeria
Email: aguessoum@caramail.com

Abstract

The new family of the digital signal processing techniques to detect image edge is considerable. The overall approach is called parametric filtering, it opens greatly promising ways in edge analysis and segmentation, including, in particular, the signal of the demodulated lag-one autocorrelation, plots of the time correlation analysis and the distortion measure based on the signal. The initial experiments establish the growing interest of the parametric filtering method with their significance and the new diagnoses of various image edge analysis.

Keywords: Edge detection, parametric filter, correlation structure, distortion measures.

1. Introduction

The digital signal processing had a very great impact on coding, image recognition and edge detection. In these three fields, there was the need for detecting the variations of the entering signal such as the estimates of the attributes related to image models (contrast, width and clearness) or the addition of the background noise [2,3].

Image processing must be developed for various reasons according to research in the comprehension of the production process of contour, and it is the key or the precursor in the applications of the image edge detection.

However, the problems of edge detection imply the construction of the time series and the spectral models and the detection of the edge structure changes according to the characteristics of the image (contrast, luminosity, background noise and a number of objects appearing in the image...)

Since this method is successfully applied to speech and vocal segmentation, one will project it, by application, on edge detection to weigh its robustness with the background noise and with the various types of images.

In addition to generalization, parameters or characteristics which the image represents, image processing can also achieve detection of the maxima in the model parameters and in the background noise filtering [1,4,5]. For edge detection, it was always necessary to classify the image as noised or noiseless for various contrast and clearness states, however, the growing interest for edge detection with various methods created the need for a broad range of classification of the image in categories. In particular, if is often approved to detect the border of the object, signal activity, background noise [6, 7].

There is a broad literature in the digital signal processing for image segmentation with



applications to image processing, synthetisation and with edge detection.

The methods of PF (parametric filtering) use the technique of passage by zero, to detect the maxima points of edge and make edge detection more sophisticated. In each of these methods, one may find more indicators, such as the number of passages by zero, the uncertainty of spectral distortion measure error, to detect significant variations.

In this article, we suggest a new method to detecting image contours based on the technique called "Parametric Filtering [8], [9]. To segment the image edge in relatively homogeneous sections, the suggested method combines a parametric filter with the analysis of the lag one autocorrelation of the filtered image signal, once done; it will produce a new characteristic function for the signal image spectrum, based on this new characteristic function. Various distortion measures are proposed as indicators of the spectral change. Initial research showed that these indicators show good performances for edge detection and segmentation and resistance to background noise and fluctuations of dominant spectral peaks.

Being an independent model, these indicators avoid the problem of modeling inaccuracies, shown in the selected model-dependent methods.

2 General View on PF

To put it simply, we classify the methods into two categories: the model-based methods and the model-free ones. For more information refer to [3], [10] and [11].

2.1 Model-based Methods

They depend primarily on modeling *LPC* (auto regressive) due to considerations of the calculation effectiveness. One of the methods is derived from a *GLR* test (A generalized likelihood ratio) by assuming that the signal is part of an auto regressive process led by the Gaussian white-vibration. By giving a field to the signal, the method, in theory, tests the assumption that no change occurs in the field, whereas a variation takes place in this field at a certain time t . Using *LPC* models, the Gaussian likelihood functions are evaluated under the two assumptions and the likelihood ration D_t is calculated for each t . If D_t is maximized in $t=t^*$

and the maximum exceeds a certain threshold, a change in $t=t^*$ happens.

Moreover, the probability likelihood ratios to *LPC* models are also used as spectral estimators in the calculation of distortion measures to detect spectral variation [3],[12],[16]. This leads to methods which avoid using the Gaussian assumption frequently signaled by the likelihood ratio tests.

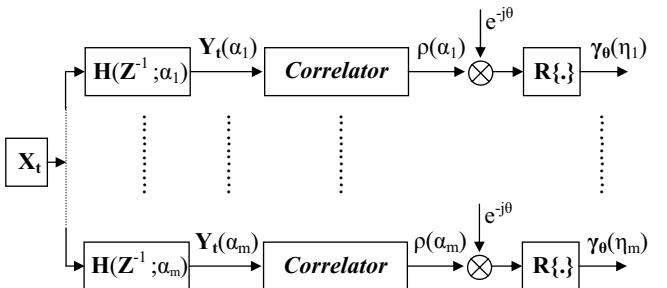


Fig.1. Block diagram of the filter bank.

2.2 Model-free Methods

The existing model-free methods are generally based on the direct use of the image spectrum, coupled with a spectral distortion measure. For example, Dehayes and Picard [9] use the spectral distance from Kolmogorov-Smirnov:

$D = \sup |F_1(\omega) - F_2(\omega)|$ and the spectral

distance from Cramer-Von Mises
 $D = \int |F_1(\omega) - F_2(\omega)|^2 d\omega$, where $F_1(\omega)$ and $F_2(\omega)$ are functions to estimated spectral distributions of two fields of the signal taken in the neighborhood of time t . The variation is declared as taking place in t when distance calculated in t exceeds a threshold.

Other spectral distances (distortion measures) can as well be employed for these goals [12],[13],[17], two examples are the spectral divergence of Kullback-Leibler (KL) [18].

$$D_{KL} = \int_{-\pi}^{\pi} K \left[\frac{f_1(\omega)}{f_2(\omega)} \right] d\omega \quad (1)$$

With $K(u) = u - \log(u) - 1$, and the L_2 distance of \log -

$$\text{spectrum } D_{LL2} = \int \left| \log f_1(\omega) - \log f_2(\omega) \right|^2 d\omega,$$



where $f_1(\omega)$ and $f_2(\omega)$ are the spectral density functions estimated from two frames of signal in neighborhood of t . In the image processing, D_{KL} is also known as the Itakura-Saito distance and D_{LL2} known as the L_2 distance of cepstral coefficients, especially when coupled with the LPC spectral estimator [12],[14].

3 Parametric Filtering Algorithm

Suppose $\{X_t\}$ is real-valued stationary signal with mean equal to zero and autocorrelation function equal to $\rho_k = E(X_{t+k} X_t)/E(X_t^2)$. Let's consider the recursive IIR (all-pole) filter $H(z^{-1}; \alpha)$ as defined by

$$\begin{aligned} Y_t(\alpha) &= \sum_{l=0}^{\infty} \bar{\alpha}^l X_{t-l} \\ &= \bar{\alpha} Y_{t-1}(\alpha) + X_t \end{aligned} \quad (2)$$

Where $\alpha = \eta e^{-j\theta}$ is a complex number with $|\eta| < 1$ and $\theta \in [-\pi, \pi]$, and the overbar represents complex conjugate. Let $\rho(\alpha)$ be the lag-one (first-order) autocorrelation of $\{Y_t(\alpha)\}$, namely

$$\rho(\alpha) = \frac{E(Y_{t+1}(\alpha) \cdot \bar{Y}_t(\alpha))}{E(|Y_t(\alpha)|^2)} \quad (3)$$

For any fixed θ , define the demodulated lag-one autocorrelation of $\{Y_t(\alpha)\}$ as

$$\gamma_\theta(\eta) = \Re(e^{-j\theta} \rho(\alpha)) \quad (4)$$

where $\Re(\cdot)$ represents the real part of a complex number. In this paper, we use $\gamma_\theta(\eta)$ as a new characterization function, complementary to the Fourier spectrum, for representing the correlation structure of $\{X_t\}$. We call this method of analyzing correlation (spectral) structure, the parametric filtering method.

To calculate $\gamma_\theta(\eta)$, we do not impose any parametric models or distributional assumptions on $\{X_t\}$; therefore, the method belongs to the model-free category without explicitly using the spectral densities.

In applications, the method can be easily implemented with a filter bank shown in Fig.1, where η_k can be taken, for example, uniformly from an interval $[\eta_a, \eta_b] \subset [-1, 1]$. For each evaluation of $\gamma_\theta(\eta)$, the number of required multiplications is proportional to the length of the signal, and usually a few evaluations are sufficient for edge detection. If necessary, the analysis can also be carried out with a variable sequence θ , using the prototype filter bank (refer to Fig.1).

3.1 Distortion measures

Given two fields of image signal, say $X_t^{(1)}$ and $X_t^{(2)}$, the characterization property of $\gamma_\theta(\eta)$ can be exploited to derive “distortion measures” that quantify the deviation of $X_t^{(1)}$ and $X_t^{(2)}$ in their correlation structures.

The PF-based distortion measures that have been found effective in our pilot of image edge-detection include the L_p distance of $\gamma_\theta(\eta)$, i.e.:

$$\gamma_\Omega^p = \left(\int_{\Omega} |\gamma_\theta^{(1)}(\eta) - \gamma_\theta^{(2)}(\eta)|^p d\theta d\eta \right)^{1/p} \quad (5)$$

for $p \in (0, \infty)$, and the (symmetrized) KL-type divergence measures

$$\hat{K}_\Omega = \int_{\Omega} \left(K \left[\frac{p_\theta^{(1)}(\eta)}{p_\theta^{(2)}(\eta)} \right] + K \left[\frac{p_\theta^{(2)}(\eta)}{p_\theta^{(1)}(\eta)} \right] \right) d\theta d\eta \quad (6)$$

and

$$\hat{K}_\Omega = \int_{\Omega} \left(p_\theta^{(2)}(\eta) K \left[\frac{p_\theta^{(1)}(\eta)}{p_\theta^{(2)}(\eta)} \right] + p_\theta^{(1)}(\eta) K \left[\frac{p_\theta^{(2)}(\eta)}{p_\theta^{(1)}(\eta)} \right] \right) d\theta d\eta \quad (7)$$

In these expression, Ω is a subset of $[-\pi, \pi] \times [\eta_a, \eta_b]$, $\gamma_\theta^{(1)}(\cdot)$ and $\gamma_\theta^{(2)}(\cdot)$ are the characterization functions obtained from $X_t^{(1)}$ and $X_t^{(2)}$, respectively, and $p_\theta^{(1)}(\cdot)$ and $p_\theta^{(2)}(\cdot)$ are normalized “density functions” on $[\eta_a, \eta_b] \subset [-1, 1]$, taking the form



$$p_\theta(\eta) = \frac{1}{2} \left(\frac{d\gamma_\theta(\eta)}{d\eta} + [\gamma_\theta(\eta_a) + 1]\delta(\eta - \eta_a) + [1 - \gamma_\theta(\eta_b)]\delta(\eta - \eta_b) \right) \quad (8)$$

Note that $p_\theta(\eta)$ also possesses the characterization property because of its equivalence to $\gamma_\theta(\eta)$. In fact, it is easy to see that

$$\int_{\eta_a}^{\eta_b} p_\theta(\lambda) d\lambda = 1 \text{ and}$$

$$\gamma_\theta(\eta) = 2 \int_{\eta_a}^{\eta} p_\theta(\lambda) d\lambda - 1$$

for any $\eta \in (\eta_a, \eta_b)$.

In applications, these distortion measures are discredited using the output $\gamma_\theta(\eta_k)$ from the filter bank in Fig.1.

4. Edge-detection Algorithm

In our preliminary experiments of image edge-detection, the approach of the peak choice (ex: [3],[14],[15]) is used for detection [3]. A typical method of the peak choice takes windows of $2N$ -points of the original image signal, in each point t , a window is centered in t and the other shifted in front of m -points. For the continuation of all the experiments, we take $m=N/2$ (i.e. 50% of overlapping). The two windows are multiplied by N -point Hamming window before they are used in the evaluation of distortion measures. Let the resulting distortion be D_t . Then, the locations of significant peaks in the trajectory of D_t are regarded as locations of spectral changes. These locations may be identified from the zero-crossings of difference $D_t - D_{t-1}$. A peak in D_t is considered significant if its magnitude exceeds a threshold T . Improved performance can be achieved in general by smoothing the trajectory of D_t before the peak-picking step [3],[14]. The parameter filter algorithm for edge detection is illustrated in Figure (2).

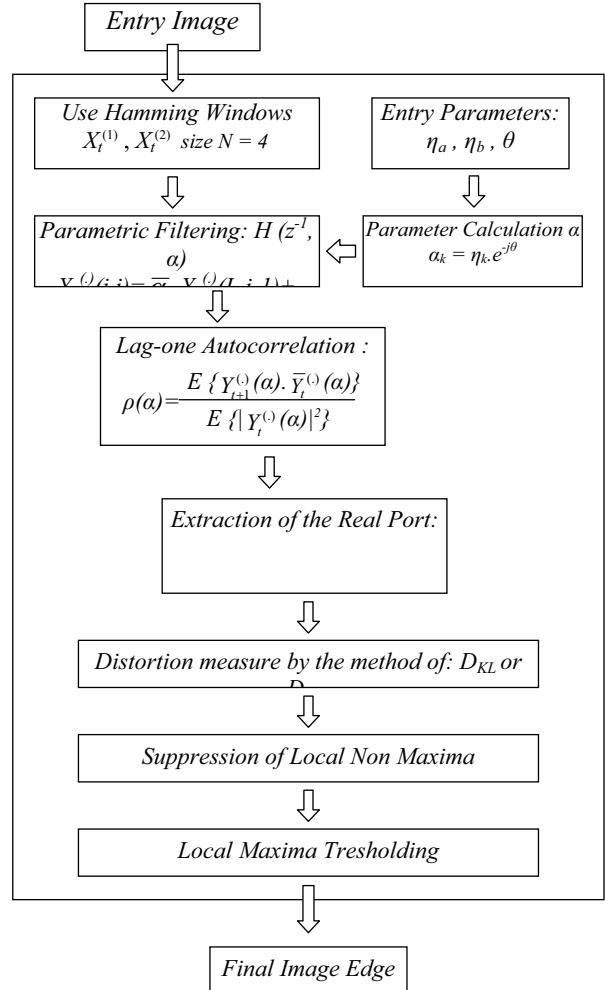


Fig.2. The flowchart of edge detection by PF method

The following stages describe the introduced algorithm:

1. we will calculate the parameter α :

$$\alpha_k = \eta_k e^{j\theta}$$

with $\eta_k = \eta_a + \frac{(k-1)(\eta_b - \eta_a)}{(m-1)}$ for $k=1, \dots, m$

2. we will filter $I(x, y)$ according to lines (X) :

$$Y_x^{(.)} = \bar{\alpha} Y_{x-1}^{(.)} + X_x^{(.)}$$

3. the autocorrelation according to line (X) :

$$\rho(\alpha) = \frac{E(Y_{x+1}(\alpha) \cdot \bar{Y}_x(\alpha))}{E(|Y_x(\alpha)|^2)}$$

and then we take: $Y_\theta^{2(.)}(\eta_k) = \Re(e^{-j\theta} \rho^{(.)}(\alpha))$

4. we fix θ , N and m to measure the distortion by:

- the D_{KL} method following lines (X)



$$\hat{K}_{\theta,x} = \sqrt{N} \cdot \frac{1}{m+1} \sum_{k=0}^m \left(K \left[\frac{P_{\theta,x,k}^{(1)}}{P_{\theta,x,k}^{(2)}} \right] + K \left[\frac{P_{\theta,x,k}^{(2)}}{P_{\theta,x,k}^{(1)}} \right] \right)$$

where: $p_{\theta,x,k}^{(1)} = \gamma_{\theta,x}^{(1)}(\eta_1) + 1$ for $k=0$

and : $p_{\theta,x,m}^{(1)} = 1 - \gamma_{\theta,x}^{(1)}(\eta_m)$ for $k=m$

and : $p_{\theta,x,k}^{(1)} = \gamma_{\theta,x}^{(1)}(\eta_{k+1}) - \gamma_{\theta,x}^{(1)}(\eta_k)$ for: $k=1, \dots, m-1$

- the D_{LL2} method following lines (X):

$$\gamma_{\theta,x}^2 = \sqrt{N} \frac{1}{m} \sum_{k=1}^m \left| \gamma_{\theta,x}^{(1)}(\eta_k) - \gamma_{\theta,x}^{(2)}(\eta_k) \right|^2$$

with: $\gamma_{\theta,x}^{(1)}(\eta_k), \gamma_{\theta,x}^{(2)}(\eta_k)$ are the two filtered signals.

5. we will filter $I(x, y)$ according to columns (Y):

$$Y_y^{(1)} = \bar{\alpha} Y_{y-1}^{(1)} + X_y^{(1)}$$

- 6. the autocorrelation according to columns (Y):

$$\rho(\alpha) = \frac{E(Y_{y+1}(\alpha) \cdot \bar{Y}_y(\alpha))}{E(|Y_y(\alpha)|^2)}$$

and then we take :

$$Y_\theta^{(2)}(\eta_k) = \Re(e^{-j\theta} \rho^{(1)}(\alpha))$$

7. we fix θ, N and m to measure the distortion with:

- the D_{KL} method following columns (Y)

$$\hat{K}_{\theta,y} = \sqrt{N} \cdot \frac{1}{m+1} \sum_{k=0}^m \left(K \left[\frac{P_{\theta,y,k}^{(1)}}{P_{\theta,y,k}^{(2)}} \right] + K \left[\frac{P_{\theta,y,k}^{(2)}}{P_{\theta,y,k}^{(1)}} \right] \right)$$

where: $p_{\theta,y,k}^{(1)} = \gamma_{\theta,y}^{(1)}(\eta_1) + 1$ for $k=0$

and : $p_{\theta,y,m}^{(1)} = 1 - \gamma_{\theta,y}^{(1)}(\eta_m)$ for $k=m$

and : $p_{\theta,y,k}^{(1)} = \gamma_{\theta,y}^{(1)}(\eta_{k+1}) - \gamma_{\theta,y}^{(1)}(\eta_k)$ for: $k=1, \dots, m-1$

- the D_{LL2} method following columns (Y):

$$\gamma_{\theta,y}^2 = \sqrt{N} \frac{1}{m} \sum_{k=1}^m \left| \gamma_{\theta,y}^{(1)}(\eta_k) - \gamma_{\theta,y}^{(2)}(\eta_k) \right|^2$$

with : $\gamma_{\theta,y}^{(1)}(\eta_k), \gamma_{\theta,y}^{(2)}(\eta_k)$ are the two filtered signals.

8. finally, we calculate the average of two distortion measures according to X and Y :

- the D_{KL} method: $\hat{K}_\theta = (\hat{K}_{\theta,x} + \hat{K}_{\theta,y})/2$

- the D_{LL2} method: $\gamma_\theta^2 = \sqrt{|\gamma_{\theta,x}^2| + |\gamma_{\theta,y}^2|}$

we get the following maxima (X) and (Y) of the image to have the desired contour.

5. Experimental results

Fig.3 shows an image of 256*256 dimension, coded on 256 levels of grey, representing various objects and on which we tested the algorithm of the parametric filtering method. Fig.4 shows the image edges when D_{KL} distortion measure is used with the following parameters $\eta_a = 0.0, \eta_b = 0.9, \phi = 0.00063\pi, m = 2$. Fig.5 illustrates the image edges when D_{LL2} distortion measure is used with the parameters $\eta_a = 0.0, \eta_b = 0.9, \phi = 0.00063\pi, m = 2$

Regarding the size of Hamming windows, our tests are carried out on windows of size $N=4$. The tests parameters are: η_a, η_b, ϕ : Parameters of the parametric filter m : size of the parametric filter.

On the practical level, the parameters which were carried out on the audio segmentation [19] are also adequate for edge detection. According to your Ta-Hsin [19], the adequate parameters are:

$$\eta_a = 0.0, \eta_b = 0.9, \phi = 0.00063\pi, m = 2.$$

We note that the chains of edges are continuous (Fig.4 and Fig.5) this allows the statement of good results



Fig.3. Original Image objets.

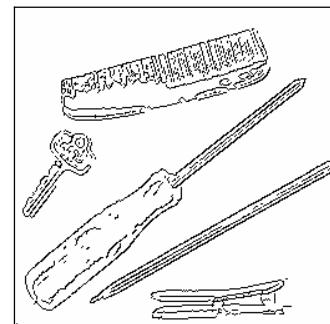
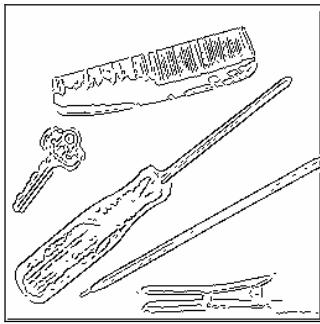


Fig.4. Image edge with D_{KL} Method



Fig.5. Image edge with D_{LL2} Method

6. Conclusion

In this paper, a new method of image edge-detection and characterization is presented. PF method uses a judicious defined filter, which preserves the signal correlation structure as input in the autocorrelation $\gamma_\theta(\eta)$ of the output. This leads to the *TCA* plot showing the temporal evolution of the image correlation structure as well as various distortion measures which quantify the deviation between two zones of the signal (the two Hamming signals) for the protection of an image edge.

Future research will include the experimentation in the other edge-detection algorithms such as the multistage dynamic programming [14]. With new distortion measures as well as the complementary exploration of the *TCA* plots response and distortion measures of the various spectral changes and parameter selection.

Our work opens the field to experimentation and tests on other types of images and with various input parameters.

As the results we obtained are encouraging, we may, consequently, consider some prospects for extension listed hereafter:

1. A study of performance of the parametric filtering method, with a theoretical evaluation as well as an automatic selection of entry parameters for each type of images;
2. The use other distortion measure algorithms.

References

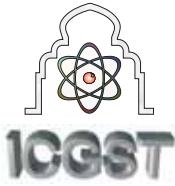
- [1] J.R. Deller, Jr.J. Proakis, and J. H. L. Hansen, *Discrete-Time processing of Speech signals*. New York: Macmillan, 1993.
- [2] R. Adre-Obrech, *A new statistical approaches for automatic segmentation of continuous speech signals*, IEEE Trans. Acoust., Speech, Signal Processing, vol 36, pp 29-40, Jan 1998.
- [3] E. Vidal and A. Marzial, *A review and new approach for automatic segmentation of speech signals*, in Signal Processing V: *Theories and Application*, L. Torres et al. Eds. New York: Elsevier, 1990, vol 1 pp. 43-53.
- [4] L. Rabiner and B.H. Juang, *Fundamental of speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993
- [5] L. R. Rabiner, J.G Wilpon, and F. K. Soong, *high performance connected digit. Recognition using hidden Markov models*. IEEE Trans. Acoust., Speech, Signal Processing, vol. 37, pp 1214-1225, Aug 1989.
- [6] E. Paksoy, K. Sanivasan, and A. Gersho, *Variable rate speech coding with phonetic segmentation*, in Proc. ICASSP, Minneapolis, MN, Apr.1993, vol. 2, pp 155-158.
- [7] S. Wang and A. Gersho, *Improved phonetically-segmentation vector excitation coding at 3.4 kb/s*, in Proc. ICASSP San Francisco, CA, Mar 1992, vol. 1, pp.349-352.
- [8] T.H. Li and J.D. Gibson, *Discrimination of time series of, speech by parametric filtering*, J. Amer. Stat. Assoc, vol. 91, pp. 284-296, Mar. 1996.
- [9] T.H. Li and J.D. Gibson, *Discrimination analysis of speech by parametric filtering*, in Proc. IEEE conf Inform. Science, Systems, Princeton, NJ, Mar. 1994, pp. 575-580.
- [10] M. Basseville and A. Benveniste, *Detection of Abrupt Changes in Signals and Dynamical System*. New York. Springer, 1986.
- [11] M. Basseville and I.V. Nikoforov, *Detection of Abrupt Changes, Theory and Application*. Englewood Cliffs, NJ Prentice-Hall 1993.
- [12] M. Basseville, *Distance measures for signal processing and pattern recognition*, Signal Processing, vol. 18, pp. 349-369, 1989.
- [13] R.M. Gray, A. Bazo, Y. Matsuyama, *Distortion measures for Speech processing*, IEEE Trans. Acoust, Speech, Signal Processing, vol ASSP-28, pp. 367-376, Apr. 1980.
- [14] T. Svendsen and F.K. Soong, *On the automatic segmentation of speech signals*, in Proc. ICASSP. Dallas, TX, Apr. 1987, ^^. 77-80.
- [15] J.G. Wilpon, B.H. Juang, and L.R. Rabiner, *An investigation on the use of acoustic sub-word units for automatic speech recognition*, in Proc. ICASSP, TX, Apr.1987, pp 821-824.
- [16] S.M. Kay, *Modern Spectral Estimation, Theory and Application*. Englewood Cliffs NJ, Prentice-Hall, 1988.
- [17] M. Lavielle, *Detection of changes in the spectrum of a multidimensional process*, IEE Trans. Signal processing, vol. 41, pp. 742-749 Feb. 1993.
- [18] E. Parzen, *Times series, statistics, and information*, in New Directions in Time Series Analysis, P1. 1, D



- Brillinger, Eds New York: Springer-Verlag, 1992, pp
265-286.
- [19] T.H. Li and J.D. Gibson, *Speech Analysis and Segmentation by parametric filtering* IEEE Trans on speech and audio processing. Vol.4, No 3 May 1996 pp 203-213.







Generalization of Frei and Chen's approach to extract the lines and edges of a digital images through orthogonal projections with masks of even dimension

F. Moreno, S. Romero, O. Sánchez, E. Cortés

*Departamento de Matemáticas y Departamento de I. Electrónica de Sistemas Informáticos y Automática, Universidad de Huelva,
Ctra. Huelva-Palos de la Ftra. s/n, 21819 Huelva, Spain,
[franmo, sixto, omar, ecortes]@uhu.es,*

Abstract

The present work provides the generalization to the approach proposed by Frei and Chen to masks of any even dimension for line and edge detection in digital images. In the work presented in ICCS2004 we proposed the generalization to masks of odd dimension. With this work we are completed the decomposition of the euclidian space in direct sum of the three subspaces proposed for Fried and Chen for any dimension of that. It is completed with the application of the designed algorithm to the synthetic and real image.

Keywords: *Frei and Chen approach, multiplier factor of standard deviations, combined detection, orthogonal projections .*

1 Introduction

When we try to extract information from an image whose definition does not allow us to recognize its lines, edges or isolated points, what we must do is its segmentation. This process consists of dividing the image into its mentioned parts.

Segmentation algorithms, [5], for monochromatic images are based mainly on the two fundamental properties of grey levels -discontinuity and similarity. In the former, the abrupt change of the grey level is used for locating lines and edges. Thresholding, region growing and region splitting or merging are used in the latter, [1].

For detecting the three basic types of discontinuity, isolated points, lines and edges, we will use the usual masks for each case. The technique consists of treating each pixel of the original image and creating a new image, [8]. For this purpose, using a 3×3 mask as example, we change the pixel grey level, which matches the central cell of the mask, according to the equation,

1,

$$R = p_1 z_1 + p_2 z_2 + \cdots + p_9 z_9 \quad (1)$$

assigning the R grey level to the mentioned pixel. Being p_i , $i = 1 \dots 9$ the coefficients according to the used mask and z_i , $i = 1 \dots 9$ the pixel grey levels of the nine cells, according to figure 1

p_1	p_2	p_3	z_1	z_2	z_3
p_4	p_5	p_6	z_4	z_5	z_6
p_7	p_8	p_9	z_7	z_8	z_9

Figure 1: A 3×3 mask

Let us suppose 3×3 masks for determining lines and edges. Let us consider the chosen pixel and its eight neighbours as a nine-component vector representing the grey levels

$$\mathbf{z} = (z_1, z_2, z_3, \dots, z_9)^T \quad (2)$$

where z_5 represents the grey level of the chosen pixel, and another vector, also with nine components representing the mask coefficients

$$\mathbf{w} = (p_1, p_2, p_3, \dots, p_9)^T \quad (3)$$

according to equation 1

$$R = \mathbf{w}^T \mathbf{z}. \quad (4)$$

Let us consider each pixel and its neighbours as a nine component vector $\mathbf{z} \in \mathbb{R}^9$ and a series of nine masks $\mathbf{w}_1, \mathbf{w}_2, \dots \in \mathbb{R}^9$ for the detection of edges and lines -combined detection-. By properly choosing the coefficients of the masks so that they represent orthogonal vectors, we can split our 9-dimensional vectorial



space into three orthogonal subspaces of dimensions 4, 4 and 1. This approach was first proposed by Frei and Chen, [7]. The first two are called edge and line subspaces, and the 1-dimensional one is called measurement subspace. Frei and Chen suggested that a measurement of the probability that a pixel belongs to an edge or line is the value of the angle formed by each \mathbf{z} vector with its orthogonal projection onto each of the mentioned subspaces. The smaller the angle, the higher the probability of belonging to the corresponding subspace. Starting from this proposal, our work consisted of calculating the projection matrixes, [12], with the usual masks -Prewitt, Sobel, etc- generalizing the results, as will be shown further on.

In the work presented of ICCS2004, [11], we generalize the results for masks of odd dimension, 3×3 , 5×5 , etc. In this work we propose the generalization to masks of even dimension highlighting the results that they are obtained for masks of 2×2 dimension.

2 Projections matrices onto vectors subspace of an euclidian space of even dimension

Theorem 2.1 Let E , $\dim E = n$, $n = 2k$, $k \in \mathbb{N}$, a euclidian space and let $\mathcal{G} \subset E$ be the $k - 1$ -dimensional vector subspace, called edge subspace, generated for the array $\langle \mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_{k-1} \rangle$ of basic vectors of \mathcal{G} whose symmetric components are opposed and their sum is zero. Then the orthogonal projection matrix onto the \mathcal{G} subspace is the matrix $P_{\mathcal{G}} \in \mathcal{M}_n(\mathbb{R})$:

$$\left(\begin{array}{cccccc} \frac{n-2}{2n} & \frac{-1}{n} & \dots & \frac{-1}{n} & \frac{1}{n} & \dots & \frac{1}{n} & \frac{2-n}{2n} \\ \frac{-1}{n} & \frac{n-2}{2n} & \dots & \frac{-1}{n} & \frac{1}{n} & \dots & \frac{2-n}{2n} & \frac{1}{n} \\ \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots & \vdots \\ \frac{-1}{n} & \frac{-1}{n} & \dots & \frac{n-2}{2n} & \frac{2-n}{2n} & \dots & \frac{1}{n} & \frac{1}{n} \\ \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots & \vdots \\ \frac{1}{n} & \frac{2-n}{2n} & \dots & \frac{1}{n} & \frac{-1}{n} & \dots & \frac{n-2}{2n} & \frac{-1}{n} \\ \frac{2-n}{2n} & \frac{1}{n} & \dots & \frac{1}{n} & \frac{-1}{n} & \dots & \frac{-1}{n} & \frac{n-2}{2n} \end{array} \right) \quad (5)$$

Theorem 2.2 Let E , $\dim E = n$, $n = 2k$, $k \in \mathbb{N}$, a euclidian space and let $\mathcal{G} \subset E$ be the $k - 1$ -dimensional vector subspace, called edge subspace, generated for the array $\langle \mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_{k-1} \rangle$ of basic vectors of \mathcal{G} whose symmetric components are opposed and their sum is zero. Then the orthogonal projection matrix onto the \mathcal{G} subspace is the matrix $P_{\mathcal{G}} \in \mathcal{M}_n(\mathbb{R})$:

$\mathcal{M}_n(\mathbb{R})$:

$$\left(\begin{array}{ccccccccc} \frac{n-2}{2n} & \frac{-1}{n} & \dots & \frac{-1}{n} & \frac{-1}{n} & \dots & \frac{-1}{n} & \frac{n-2}{2n} \\ \frac{-1}{n} & \frac{n-2}{2n} & \dots & \frac{-1}{n} & \frac{-1}{n} & \dots & \frac{n-2}{2n} & \frac{-1}{n} \\ \vdots & \vdots & \dots & \frac{n-2}{2n} & \frac{n-2}{2n} & \dots & \vdots & \vdots \\ \vdots & \vdots & \dots & \frac{n-2}{2n} & \frac{n-2}{2n} & \dots & \vdots & \vdots \\ \frac{-1}{n} & \frac{n-2}{2n} & \dots & \frac{-1}{n} & \frac{-1}{n} & \dots & \frac{n-2}{2n} & \frac{-1}{n} \\ \frac{n-2}{2n} & \frac{-1}{n} & \dots & \frac{-1}{n} & \frac{-1}{n} & \dots & \frac{-1}{n} & \frac{n-2}{2n} \end{array} \right) \quad (6)$$

Theorem 2.3 Let \mathcal{E} , $\dim \mathcal{E} = n$, $n = 2k$, $k \in \mathbb{N}$ an euclidian space and let $\mathcal{V} \subset \mathcal{E}$ the two-dimensional vector subspace, called uniform subspace, generated for the array $\langle (\alpha, \underbrace{\dots, \alpha}_{\frac{k}{2}}, 0, \underbrace{\dots, \alpha}_{\frac{k}{2}}, 0), (0, \underbrace{\dots, 0}_{\frac{k}{2}}, \beta, \underbrace{\dots, \beta}_{\frac{k}{2}}) \rangle$. Then the orthogonal projection matrix onto the \mathcal{V} subspace is the matrix $P_{\mathcal{V}} \in \mathcal{M}_n(\mathbb{R})$:

$$P_{\mathcal{V}} = \left(\begin{array}{cccc|cccc} \frac{2}{n} & \frac{2}{n} & \dots & \frac{2}{n} & 0 & 0 & \dots & 0 \\ \frac{2}{n} & \frac{2}{n} & \dots & \frac{2}{n} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ \frac{2}{n} & \frac{2}{n} & \dots & \frac{2}{n} & 0 & 0 & \dots & 0 \\ \hline 0 & 0 & \dots & 0 & \frac{2}{n} & \frac{2}{n} & \dots & \frac{2}{n} \\ 0 & 0 & \dots & 0 & \frac{2}{n} & \frac{2}{n} & \dots & \frac{2}{n} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 & \frac{2}{n} & \frac{2}{n} & \dots & \frac{2}{n} \end{array} \right) \quad (7)$$

2.1 Norms of projections vectors

In the Frei and Chen's approach the angles they are compared. We will use the norms of projection vectors instead of the angles of the vector with each projections. In the next corollaries we propose the norms of projections vectors onto each one of the three subspaces.

Corollary 2.4 Let E , $\dim E = n$, $n = 2k$, $k \in \mathbb{N}$, a euclidian space, with $\|\cdot\|_2$, and let $\mathbf{x}(x_1, \dots, x_n)^t \in \mathcal{E}$. The norm of the vector projection onto the $\mathcal{G} \subset E$ is

$$\|\mathbf{x}_{\mathcal{G}}\| = \sqrt{\frac{1}{2} \left(\sum_{i=1}^n x_i^2 - \sum_{i=1}^n x_i x_{n+1-i} \right) - \frac{1}{n} (S_2 - S_1)^2} \quad (8)$$

with $S_1 = \sum_{i=1}^{\frac{n}{2}} x_i$ and $S_2 = \sum_{i=\frac{n}{2}+1}^n x_i$.



Corollary 2.5 Let E , $\dim E = n$, $n = 2k$, $k \in \mathbb{N}$, a euclidian space, with $\|\cdot\|_2$, and let $\mathbf{x}(x_1, \dots, x_n)^t \in \mathcal{E}$. The norm of the vector projection onto the $\mathcal{H} \subset E$ is

$$\|\mathbf{x}_{\mathcal{H}}\| = \sqrt{\left(\frac{1}{2} \sum_{i=1}^n x_i^2 + \frac{1}{2} \sum_{i=1}^n x_i x_{n+1-i}\right) - \frac{S^2}{n}} \quad (9)$$

$$\text{with } S = \sum_{i=1}^n x_i$$

Corollary 2.6 Let E , $\dim E = n$, $n = 2k$, $k \in \mathbb{N}$, a euclidian space, with $\|\cdot\|_2$, and let $\mathbf{x}(x_1, \dots, x_n)^t \in \mathcal{E}$. The norm of the vector projection onto the $\mathcal{V} \subset E$ is

$$\|\mathbf{x}_{\mathcal{V}}\| = \sqrt{\frac{2}{n}(S_1^2 + S_2^2)}$$

$$\text{with } S_1 = \sum_{i=1}^{\frac{n}{2}} x_i \text{ and } S_2 = \sum_{\frac{n}{2}+1}^n x_i.$$

2.2 Detection combined with masks of even dimension

We will study the results of applying the results of the sections 2 and 2.1 to masks of 2×2 and 3×2 dimension, figure 2

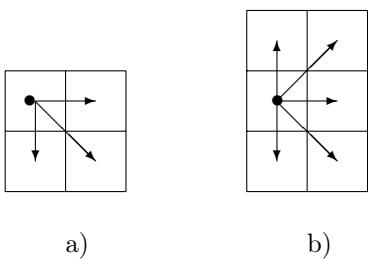


Figure 2: a) Masks 2×2 . b) Mask 3×2 .

2.3 Method details

Following the previous theorems and corollaries, it is not necessary to use any usual masks in order to calculate projection vectors, since projection matrixes are always the same and projection vectors will be obtained when applying such matrixes onto them. It will suffice to choose the mask dimension and we will obtain those vectors.

By calculating the norms of the three projections and choosing the biggest one, we know if the pixel is candidate for an edge, a line or a uniform luminance region. We have constructed an algorithm and, when applied, we did not obtain the expected results.

Most pixels had a projection vector onto the measurement subspace with a much higher norm than with the other two, so that there was no possibility to compare the norms of the projection vectors onto the other two subspaces. In order to remove the pixels belonging to the uniform luminance regions, and to avoid the problem, we calculate the standard deviation of each pixel and its neighbors, according to the mask dimensions, and we calculate the mean of all these deviations. When we scan the image, we compare the standard deviation of each pixel and its neighbors with the previously calculated mean. If it is bigger, it is an edge or line pixel candidate, and if it is smaller, it belongs to a uniform luminance region. This value of the standard deviation may be changed by multiplying it conveniently by a factor that will modify the obtained results. We have called it multiplier factor of the standard deviation -stdf- f in the algorithm. Since the calculation of the standard deviation mean of all image pixels would imply a very high computational time, we have design an algorithm to calculate that value with a much lower number of pixels chosen at random. This number of pixels depends on the image and mask dimensions.

2.4 Algorithm for the proposed method

An algorithm, figure 3, for the proposed methodology is described as follows

Step 1 Input the image I_i -BMP, TIF, JPG format- of size $m \times n$. Also, input the dimension mask and the factor multiplier of standard deviation f .

Step 2 Choose a pixels number distributed aleatorily by the image -subprogram included- and calculate the average of standard deviation, of every one and their neighbors according of dimension mask.

Step 3 To compare the std of each pixel to image with the average, calculate in the step 2, multiplied to f to choose the candidate to belong to an edge or a line.

Step 4 With the pixels elected in the step 3 we calculate the square norms of projections vectors -new proposed method- to know that is the bigger and to belong to an edge or a line.

3 Experimental results and conclusions

Our method, and the algorithm that we propose, detects the edges and lines of a real image although



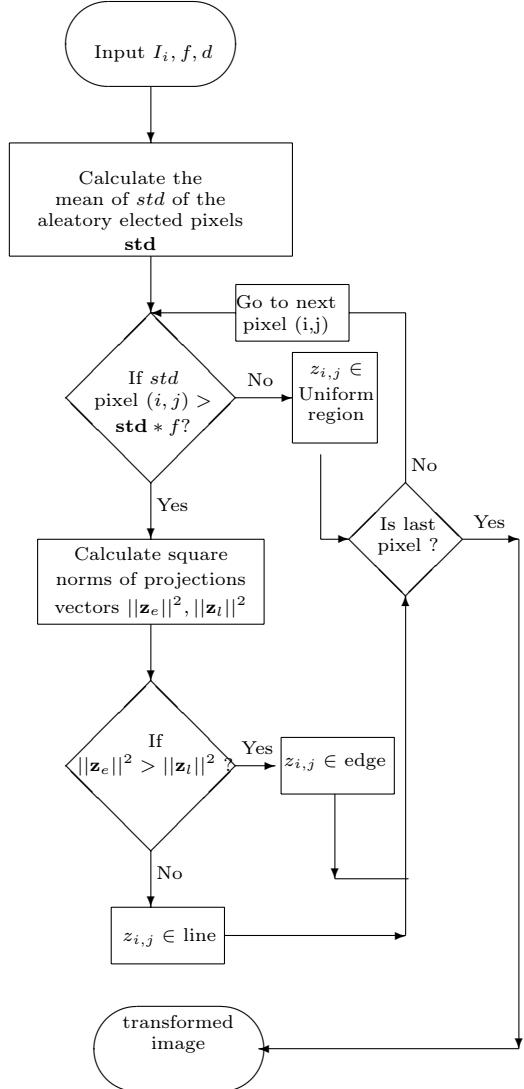
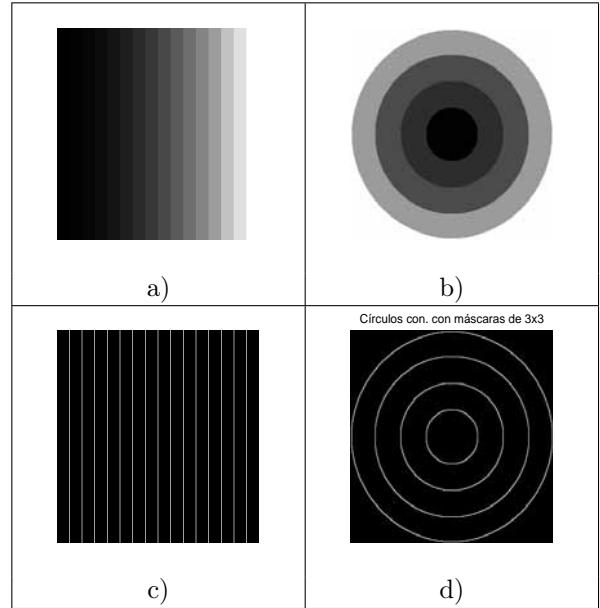


Figure 3: Flow chart of the algorithm

later on it would be necessary to narrow and to prolong or to close this edges to conclude the segmentation process. Contrary to other detection edges methods, surface fitting approach [10], [16], [3], or detectors based on some optimality criteria [2], [14],[15], or using statistical procedures are illustrated in [6], [4], [13] and other approaches include the use of genetic algorithm, neural networks, etc., with our method, based in orthogonal projections, we can obtain the lines and the edges at the same time on the original image. In order to judge the performance of the proposed algorithm we have used two real images and two synthetic images to check the effectiveness of our algorithm. The figure 4 a) and c) show vertical fringes formed by 20 pixels of width, of values $\{0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132, 156, 194, 224, 255\}$, and 199 pixels of height and the edges using dimension masks 2×2 . The figure 4 b) and d) show concentric circles of values $\{1, 45, 75, 155, 245\}$ and the

Figure 4: a) Vertical fringes, b) Concentric circles. c) Edges obtained with masks 2×2 . c) Edges obtained with masks 2×2 .

treated image with the edges. The figure 5 show the principal façade theater of Praga(Czech. Rep.)¹ and the image treated with the proposed algorithm and Canny's algorithm²; in the following figure 6 we show a detail of the same previous image in which we can observe the balcony using the algorithm that we propose and that of Canny; we can check as the algorithm that we use it gets bigger definition of the edges. In the figure³ 7 we show the result of to applied the algorithm with masks of 3×2 dimension and compare the results with other masks dimension; the best results are obtained with masks of 2×2 dimension.

Acknowledgements

Own X Plan of University of Huelva (Spain), PAI (Research Andalusian Plan of the Regional Government) the National Project of Science of Culture Department with reference REN-2000-0903-C08-08CLI, National Project of Science Educational and with reference DIP-2004-07310

References

- [1] T. Blue A. Muñoz and M. Unser. Least squares image resizing using finite differences. *IEEE Transactions on image processing*, 10-9:1365–1378, 2001.

¹photography of authors.

²Canny's algorithm is taken from Matlab ver. 7.0.0.1992 R(14)

³Image downloaded from Matlab ver. 7.0.0.1992 R(14)



- [2] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:679–698, 1986.
- [3] G. Chen and Y. H. H. Yang. Edge detection by regularized cubic b-spline fitting. *IEEE Transactions on Systems, Man and Cybernetics*, 10-9:636–643, 1995.
- [4] E. Chuang and D. Sher. Chi-square test for feature extraction. *Pattern Recognition*, 26-11:1637–1683, 1993.
- [5] S. Djedziri D. Mammmas and F. Nouboud. Pro-topological approach for image segmentation and edge detection. *Journal of mathematical imaging and vision*. Kluwer Academic, 15:169–179, 2001.
- [6] P. de Souza. Edge detection using sliding statistical test. *Comput. Vis., Graph. Image Processing*, 23-1:1–14, 1983.
- [7] W. Frei and C. Chen. Fast boundary detection: A generalization and a new algorithm. *IEEE Trans. Computer*, C-26, num.10:988–998, 1977.
- [8] J. Gómez. An analysis of edge detection by using the jensen-shannon divergence. *Journal of Mathematical Imaging and Vision*, num.13:35–56, 2000.
- [9] R. González and R. Woods. *Digital Image Processing 2nd. ed.* Prentice-Hall. Upper Saddle River., New Jersey, 2002.
- [10] R. M. Haralick. Digital step edges from zero crossing of seconds dierctional derivatives. *IEEE Trans. Pattern Anal Machine Intell.*, PAMI-6:55–68, 1984.
- [11] F. Moreno and S. Romero. An application to the treatment of geophysical images through orthogonal projections. *LNCS*, 3039:213–220, 2004.
- [12] B. Noble and J. Daniel. *Applied Linear Algebra*. Prentice-Hall, New York, 1988.
- [13] P. Qie and S. M. Bhandarkar. An edge detection techniques using local smoothing and statistical hypothesis. *Pattern Recognition Lett.*, 17-8:849–872, 1996.
- [14] S. Y. Sarkar and K. L. Boyer. Optimal infinite impulse reponse zero crossing based edge detector. *Comput. Vis. Graph. Image Processing: Image Understanding*, 54-9:224–243, 1991.
- [15] J. Shen and S. Castan. An optimal linear operator for step edge detection. *Graph. Models Image Processing*, 54-1:112–133, 1992.
- [16] S. S. Sinha and B. G. Shunk. A two stage algorithm for discontinuity preserving surface reconstruction. *IEEE Trans. Pattern Anal. Machine Intell.*, 14:36–55, 1992.



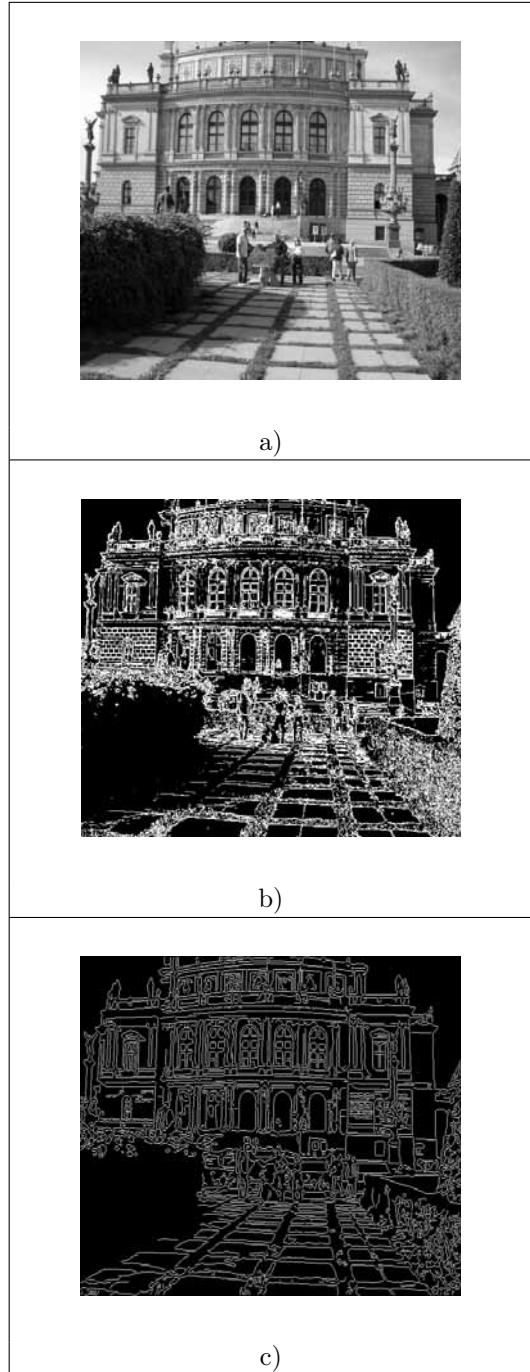


Figure 5: a) Original image. b) Image obtained applying our method. c) Image obtained applying the Canny's method

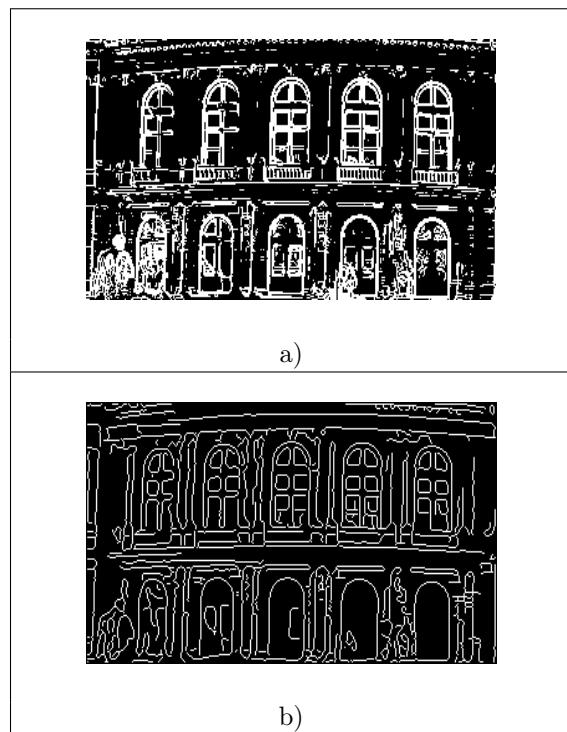


Figure 6: a) Details of figure 7 b). b) Details of figure 7 c).

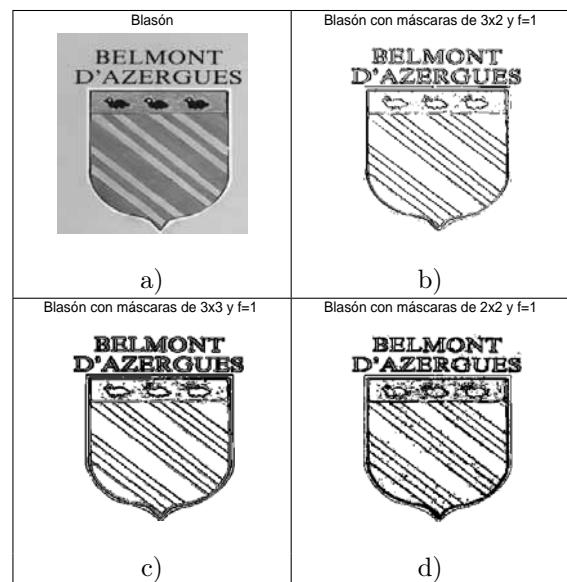


Figure 7: a) Original image b) Image obtained with masks 3×2 . c) Image obtained with masks 3×3 . d) Image obtained with masks 2×2





An Edge Detection Algorithm using Smoothing Functions

Tieling Chen

Mathematical Sciences Department, University of South Carolina Aiken

471 University Parkway, Aiken, SC 29801, USA

tielingc@usca.edu

Abstract

This paper studies the influence of the shape of the smoothing function used in a wavelet transformation based edge detector on the localization of the detected edges. A concrete smoothing function is given and it is compared with the Gaussian function used in Canny's edge detector.¹

Keyword: *Edge detection, wavelet transformation, smoothing function, localization, gradient.*

1. Introduction

Edge detection is a crucial step in edge extraction and object delineation in image processing. An effective edge detector reduces a large amount of data but still keeps most of the important feature of the image. As one of the first-use methods, edge detection is widely used in imagery data processing such as remote sensing data processing. For example, for an aerial image, a high quality edge detector not only obtains a good understanding of the image but also provides a simple but significant input for the process of extracting edges of some specific objects.

The gradient method is an early technique to detect edges (see [7], [6], and [4]). Basically, this method uses some specifically designed mask to traverse the image and detects edges by locating the maxima of the magnitude of the gradient of the image. But the masks have fixed sizes and cannot be dilated. The method is also sensitive to noises if the image is not smoothed first.

Canny's detector ([1]) is a commonly used tool in edge detection. The method first blurs the input image and then finds the gradient of the blurred image. Edges can then be detected by tracing the local maxima of the magnitude of the gradient image. Canny's detector is

optimal for isolated step edges corrupted by white noises and it uses the Gaussian function to smooth the input images. But close edges may affect each other in the process especially when the deviation of the Gaussian function is big, which leads to inaccurate edge locations and some edge losses.

The paper [5] introduces a wavelet transformation method in image processing. Although the paper does not put the focus on edge detection, some ideas can be generalized to design effective edge detectors. A wavelet transformation of a given image produces an image that is proportional to the gradient of the smoothed image, and then edges can be detected by locating the local maxima of the magnitude of the resultant image. As an edge detector, this method is essentially equivalent to Canny's method. The wavelet function used in the method is derived from a smoothing function but the paper does not discuss how smoothing functions affect the process of edge detection.

The wavelet transformation based edge detection method is extensively studied in [2] and [3]. These papers characterize some edges including the step edges through the evolution of the wavelet transformation of the edges at different scales. Importantly, these papers study the interferences of close edges during the wavelet transformations and discover that when a proper dilation scale of the smoothing function is used the edges of the main objects in the image can be detected well while the noises and tiny irrelevant objects can be mostly removed. This method has been effectively used in remote sensing image processing on the purpose of extracting road networks.

In fact, the wavelet transformation based method and Canny's method can be implemented with different smoothing functions. The shape of a smoothing function is an important aspect to be considered when the smoothing function is designed. The motivation of this paper is to study the shapes of smoothing functions and show how they affect edge detection.

¹ This work is partially supported by a grant from the University of South Carolina Research and Productive Scholarship Fund.



2. Edge detection with smoothing functions

Smoothing function and wavelet transformation: We define a smoothing function as a finite supported even function that is differentiable and has an integral value one. The Gaussian function can be treated as a smoothing function in the actual implementation because it is fast descending and the majority of the function is over a finite interval.

The derivative of a smoothing function gives a wavelet function. Suppose $\theta(x)$ is a smoothing function and $\phi(x)$ is the derived wavelet function. The wavelet transformation of a given signal function f is defined as

$$Wf(x) = f * \phi(x) = \int_{-\infty}^{\infty} f(x-t)\phi(t)dt.$$

A smoothing function $\theta(x)$ can be dilated to a scale level $s > 0$ to get a so-called dilated smoothing function $\theta_s(x) = \theta(x/s)/s$. The corresponding dilated wavelet function is defined as $\phi_s(x) = \phi(x/s)/s$.

The wavelet transformation of f at scale level s is given by

$$W_s f(x) = f * \phi_s(x) = \int_{-\infty}^{\infty} f(x-t)\phi_s(t)dt.$$

Computation gives that $W_s f(x) = s \frac{d}{dx} (f * \theta_s)(x)$,

which means the wavelet transformation of f is proportional to the derivative of a smoothed function obtained by blurring f with the smoothing function $\theta_s(x)$ ([5]).

If $\theta(x)$ is a smoothing function, then the function $\Theta(x, y) = \theta(x)\theta(y)$ gives a two-dimensional smoothing function, and its two partial derivatives give two wavelet functions. A two-dimensional wavelet transformation of a given image function $F(x, y)$ is proportional to the gradient of the corresponding smoothed image smoothed by $\Theta(x, y)$ ([5]). The wavelet transformation of $F(x, y)$ at dilation scale $s > 0$ is a vector of two components

$$W_s F(x, y) = ((W_s F(x, \cdot) * \theta_s)(y), (W_s F(\cdot, y) * \theta_s)(x)).$$

Each component in the above is in fact a one-dimensional wavelet transformation in one variable followed by a smoothing processing in the other variable ([2], [3]).

Edge Characterizations: Edges can be characterized through the evolution of a wavelet transformation at different scale levels. For a one-dimensional signal, the support set and the shape of the smoothing function heavily affect the magnitudes of the local extreme values of the corresponding wavelet transformation of the signal. This is also true for two-dimensional images when two-dimensional wavelet transformations are performed.

We can look at this through the wavelet transformation of a step signal. Step signals show abrupt changes of

intensities and they usually give the locations of the edges. For convenience, we consider the step signal $f(x) = 0$ when $x < 0$, and $f(x) = 1$ when $x \geq 1$.

If $\theta(x)$ is the smoothing function, then the wavelet transformation of this signal at scale level s is

$$\begin{aligned} W_s f(x) &= s \frac{d}{dx} (f * \theta_s)(x) = s \frac{d}{dx} \int_{-\infty}^{\infty} f(x-t)\theta_s(t)dt \\ &= s \frac{d}{dx} \int_{-\infty}^x \theta_s(t)dt = s\theta_s(x) = \theta(x/s). \end{aligned}$$

The maximum of this wavelet-transformed signal is $\theta(0)$, which is independent of the scale level value s . But the length of the support interval, which is $(-s, s)$, is increasing in the scalar value s .

A similar situation can be obtained if we consider an opposite step signal $f(x) = 0$ when $x > 0$, and $f(x) = 1$ when $x \leq 1$. The wavelet transformation of this signal at scale level s is $-\theta(x/s)$ and its minimum is $-\theta(0)$, which is also independent of s . The length of the support interval is increasing in s .

A cross section of an image gives a one-dimensional signal, and the edges in the image cut by the cross section give step-like changes or nearly step-like changes. These sudden changes cannot be treated separately, since the wavelet transformations of every two close sudden changes may affect each other.

We can check this with a platform signal. Platform signals are frequently encountered when a cross section of an image is made. For example, a cross section of a road or building in an aerial image basically gives a platform signal. A wavelet transformation of a platform signal is a sum of two opposite bumps as discussed in the above, since the platform signal is a combination of two opposite step signals. When the scale level s of the transformation is large enough, a cancellation of these two bumps occurs. The cancellation not only lowers down the magnitudes of the local extreme values of the transformed signal such that the edge points may not be detected, but also shifts the locations of the local extreme values to the outer sides of the signal and hence gives bad localizations of the edge points if they are detected.

Edge detection and the shape of a smoothing function:

The purpose of edge detection is to precisely locate the positions of sudden intensity changes, but noises should be removed. To this end, we can choose a proper dilation scale level of the smoothing function and perform the corresponding wavelet transformation on the given image. Check every pixel in the resultant image by comparing the magnitude with those on the nearby pixels along and opposite to the gradient direction. If a maximum is obtained then the pixel is an edge pixel.

The magnitudes of the local extreme values along gradient directions caused by tiny objects and noises are lowered down by cancellation, while the magnitudes of



the local extreme values of the main objects are only slightly affected because their edges are separated far apart. The intensities of the edge pixels are closely related to the sizes of objects. Basically, objects with small sizes have dim edges, while big objects delay the decrement of the intensities of edge pixels in the evolution of the wavelet transformation. This process is mainly controlled by the dilation of the corresponding smoothing function. Edges of main objects can be extracted using a threshold.

Three criteria are commonly used to evaluate an edge detection algorithm: low error rate, the edge points are well localized, and there is only one response to a single edge ([1]). Since the wavelet transformation based algorithm is equivalent to locating the extreme values of the magnitudes of the derivative of a smoothed image, there is only one response to every detected edge. Although an uncertainty principle exists for the low error rate and good localization criteria ([1]), the entire effect can be improved by imposing some requirements on the smoothing function.

The requirements are reflected in the design of the shape of the smoothing function. First, the low error rate criterion requires that the smoothing function does not have a sharp top, so high frequency noises can be largely smoothed. The extreme case of this consideration is the difference-of-boxes function, which maximize the signal to high frequency noises ratio but gives ambiguous responses to a single edge. Second, the good localization criterion requires the cancellation or addition of the extreme values of the wavelet transformation of two close edges is delayed in the evolution of the wavelet transformation. This can be reached if the locations of the most rapid changes of the smoothing function are close to the origin. A good smoothing function should take the above two requirements into consideration simultaneously. An example of a smoothing function is given in the next section and compared with the Gaussian function.

Compare with Canny's Edge Detector: The Gaussian function is exclusively used in the Canny's edge detector but it cannot be said the best approximation of a smoothing function. A dilation of a Gaussian function at a large scale easily makes the detected edges move away from the real locations.

According to the discussion in the above section, we compare the Gaussian function $G(x)$ with a specifically designed smoothing function $\theta(x)$. The deviation of $G(x)$ is set to be one third such that the majority of the function is over the interval $[-1, 1]$. The smoothing function $\theta(x)$ is also supported in $[-1, 1]$, defined as

$$\theta(x) = \begin{cases} 7.417x^6 + 44.503x^5 + 111.258x^4 + 148.344x^3 + 111.258x^2 + 44.503x + 7.417 & -1 \leq x < -0.5 \\ 66.755x^6 + 133.510x^5 + 66.755x^4 - 22.252x^3 - 22.252x^2 + 1.854 & -0.5 \leq x < 0 \\ 66.755x^6 - 133.510x^5 + 66.755x^4 + 22.252x^3 - 22.252x^2 + 1.854 & 0 \leq x < 0.5 \\ 7.417x^6 - 44.503x^5 + 111.258x^4 - 148.344x^3 + 111.258x^2 - 44.503x + 7.417 & 0.5 \leq x < 1 \\ 0 & \text{else} \end{cases}$$

The graphs of these two functions and their derivatives are illustrated in Figure 1.

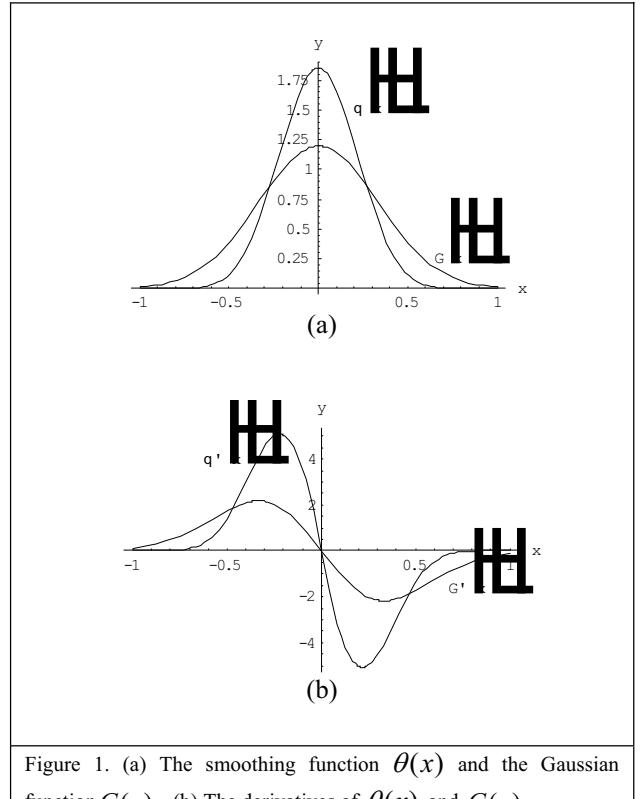


Figure 1. (a) The smoothing function $\theta(x)$ and the Gaussian function $G(x)$. (b) The derivatives of $\theta(x)$ and $G(x)$.

The smoothing function $\theta(x)$ has a closer distance between the locations of its derivative extrema. This makes close edges less affected by each other in the wavelet transformations and hence gives better localization of the edges. Meanwhile, the function $\theta(x)$ still has a good smoothing mechanism.

Figure 2 illustrates the effects of the corresponding wavelet transformations on a platform signal $p(x)$ at two different scale levels, where $p(x)=1$ when $-1 \leq x \leq 1$ and $p(x)=0$ else where.



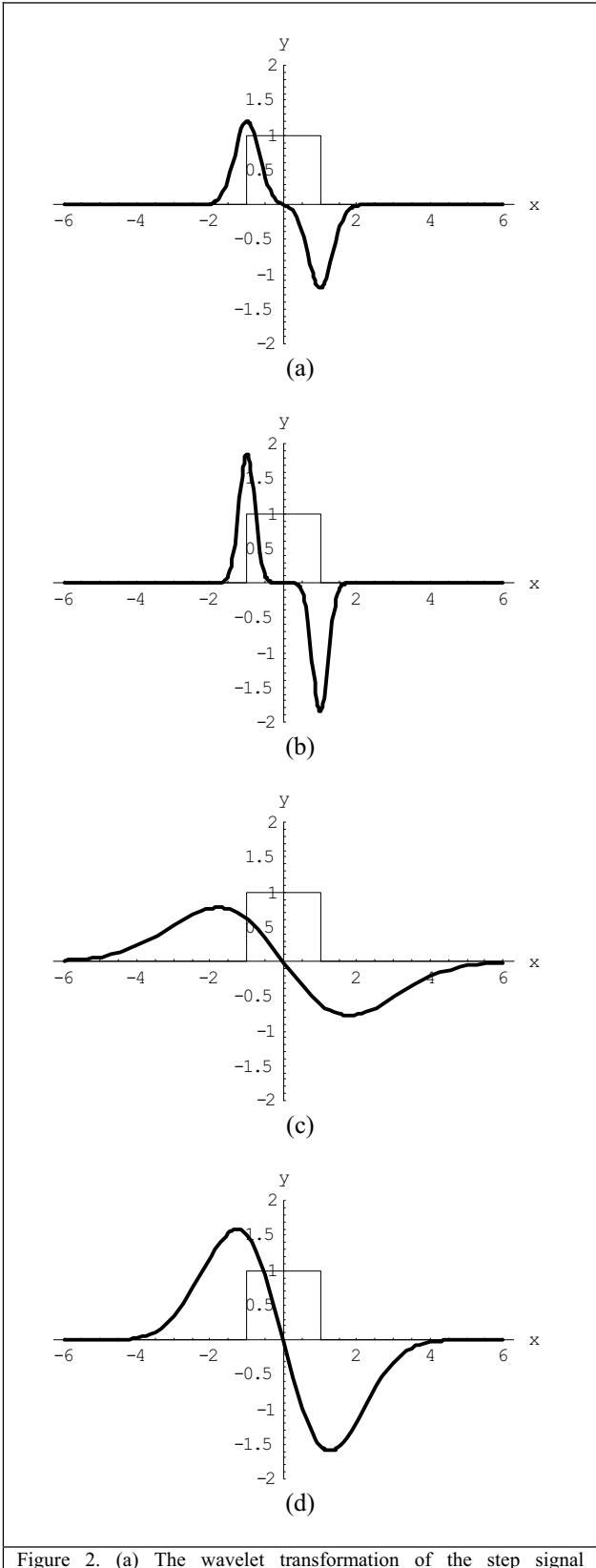


Figure 2. (a) The wavelet transformation of the step signal determined by the Gaussian function at $s = 1$. (b) The wavelet transformation of the step signal determined by the smoothing function at $s = 1$. (c) The wavelet transformation of the step signal determined by the Gaussian function at $s = 5$. (d) The wavelet transformation of the step signal determined by the smoothing function at $s = 5$.

When the dilation scale value s is small such that cancellation does not occur, the locations of the local extrema of both wavelet transformations of the signal

give the accurate edge locations. When the dilation scale is getting bigger and the cancellations occur, the locations of the extrema of the wavelet transformations are more and more separated apart in both cases, but this process is delayed in the evolution corresponding to the smoothing function $\theta(x)$ and hence a better localization is obtained. For the wavelet transformations corresponding to the dilated smoothing function at large scale values, the magnitudes of the extrema are apparently lowered down at each step. This implies that the smoothing mechanism of $\theta(x)$ is not sacrificed.

3. Examples

Edge detection: We use Lena's photo to compare the wavelet transformation based method using the above smoothing function with Canny's detector. Figure 3 (b) shows the edges detected using the smoothing function. The corresponding wavelet transformation is performed at the dilation scale $s = 10$. Figure 3 (c) shows the edges detected by Canny's detector when the Gaussian function is dilated to the same level. Edges in Figure 3 (b) are closer to their real locations, which can be seen from Lena's lips. Noises are removed by some thresholds before Figure 3 (b) and 3 (c) are obtained. We can also see that Figure 3 (b) keeps more features of Lena.

Road network extraction: Road network extraction has important applications in the geography information system. A cross section of a road gives a platform-like image hence the wavelet transformation based method is a proper choice for the road network extraction. Since the roads are the main objects, all other objects like houses and driveways as well as noises are undesirable.

Figure 4 (a) is a remote sensing image of a residential area. After a simple preprocessing such as thresholding, the wavelet transformation based method using the above smoothing function is applied. The resultant is then post-processed to remove noises and tiny objects and then we get the image in Figure 4 (b). The image of the detected edges has a good localization and only a little information of the road network is lost.

4. Conclusion

This paper considers the smoothing functions used in a wavelet transformation based edge detection algorithm and analyzes how the shape of a smoothing function affects the localization of the detected edges. To get a good localization the distance between the locations of the local extrema of the wavelet function derived from the smoothing function cannot be too big. To keep a high signal to noise ratio, the smoothing function cannot have a sharp top in shape. A smoothing function is designed in the paper and it is compared with the Gaussian function used in Canny's edge detector. With this smoothing function, the wavelet transformation based edge detector produces edges that are closer to their real locations, compared with those detected by Canny's detector at the same dilation level, while noises can still be removed with a threshold. This method can be effectively used in the general purpose of object delineation and edge extraction from remote sensing images.





(a)



(b)

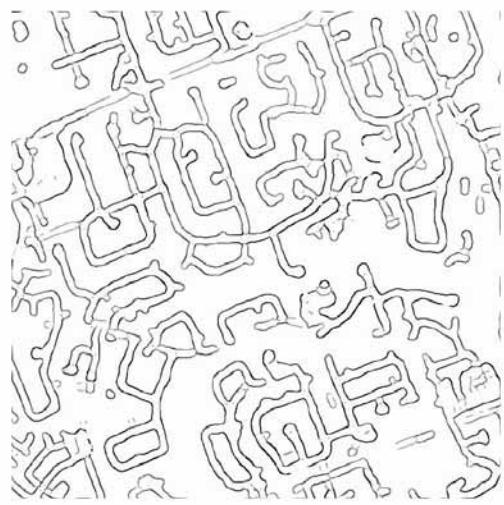


(c)

Figure 3. (a) The original image of Lena. (b) Edges detected using the smoothing function. The dilation level is $s = 10$. (c) Edges detected by Canny's edge detector with the Gaussian function dilated to the same level.



(a)



(b)

Figure 4. (a) A remote sensing image of a residential area. (b) Road network extracted using the wavelet transformation based method with the smoothing function. Some preprocessing and post processing are performed.

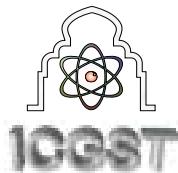
5. References

- [1] J. Canny. A Computational Approach to Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6: 679 – 698, 1986.
- [2] T. Chen, J. Wang, K. Zhang. A wavelet transform based method for road extraction from high resolution remotely sensed data, *Proceeding of the 2002 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2002)*, Toronto, Canada, VI: 3369 – 3371, 2002.
- [3] T. Chen, J. Wang, K. Zhang. A Wavelet Transform Based Method for Road Centerline Extraction, *Photogrammetric Engineering and Remote Sensing*, 70(2): 1423 – 1432, 2004.
- [4] R. Kirsh. Computer determination of the constituent structure of biomedical images, *Computers and Biomedical Research*, 4(3): 315 – 328, 1971.



- [5] S. Mallat, S. Zhong. Characterization of signals from multiscale edges, IEEE transactions on pattern analysis and machine intelligence, 14(7): 710 – 732, 1992.
- [6] J. G. Prewitt. Object enhancement and extraction, Picture Processing and Psychopictorics, B. Lipkin and A. Rosenfeld, Eds. New York: Academic, 75 – 149, 1970.
- [7] L. Roberts. Machine perception of three-dimensional solids, Optical and Electro-optical Information Processing, Cambridge, Massachusetts, MIT Press, 159 – 197, 1965.





A Novel Approach to Moving Edge Detection Using Cross Entropy

Sang Hyun Kim

Dept. of Electronic and Electrical Engineering, Sangju National University

#386, Gajang-Dong, Sangju, Kyungbuk, Korea

shk@sangju.ac.kr

Abstract

We propose a moving edge extraction method using the concept of entropy and cross-entropy, in which the cross-entropy concept is applied to dynamic scene analysis. The cross-entropy concept provides enhancement of detection for the dynamically changed area. We combine the results of cross-entropy in the difference picture (DP) with those of entropy in the current frame so that we can effectively extract moving edges. We also propose the moving edge extraction method by combining the results of cross-entropy and those of Laplacian of Gaussian (LoG).

Keywords: moving edge, entropy, cross-entropy, Laplacian of Gaussian (LoG).

1. Introduction

Segmentation of an image into a number of regions to separate the objects from the background has been an important research topic. Image segmentation can be classified into three groups.¹ The first group is based on the technique called image thresholding, which uses fixed thresholding as a decision criteria to separate an image into different regions. The second group uses the discontinuities between graylevel regions to detect edges within an image. Edges can be very important features for object recognition and identification. The last group separates an image into several different regions based upon an appropriate criteria. For example, pixels which have the same graylevel are grouped together and so on. After an image has been segmented into different regions, it is often necessary to describe these regions using a small set of descriptors, thus reducing the complexity of the image recognition process.

Dynamic scene is the picture that contains moving objects, and to separate moving objects from irrelevant background, sofisticated segmentation to separate stationary edges are necessary. Dynamic scene segmentation methods can be classified into two major categories: feature-based methods and pixel-based methods. In the feature-based methods, to separate moving objects we use object features such as corners,

straight edge segments, and center of gravity of an object. In the pixel-based methods, the moving objects are segmented using the difference picture (DP) method, matching method, or moving edge method.²

For successful dynamic segmentation, we propose a moving edge detection method based on cross-entropy concepts. The edges can be extracted by the detection of regions which involve abrupt changes of brightness and difference operators are usually used to detect edges. In this paper, we employ the difference operators using entropy and also use the commonly-used Laplacian operator. The Laplacian is a second-difference operator which is rotation invariant. The entropy operator calculates the entropy of brightness in a local region of the picture. The entropy value is maximum when the brightness in the region is uniform, and the value is small when the brightness changes abruptly.³ So the regions where the entropy is small contain edges. We can also segment moving objects if we apply a region growing method to regions around detected moving edges.

This paper is organized as follows. The concept of entropy and cross-entropy is presented in Section 2. Proposed moving edge extraction algorithms are described in Section 3, and simulation results and discussions are shown in Section 4. Finally conclusions are given in Section 5.

2. Entropy and Cross-Entropy

Since edges involve abrupt changes of intensity, color, texture, motion, and so on, edges can be extracted by detecting these changes. Shiozaki proposed the entropy operator for detection of these changes.³ This operator calculates the entropy of intensity over the region. The entropy is small (large) when the change of brightness is large (small). Then, edges may be extracted by detecting the regions where the entropy is small. Let a_0 be the intensity value at center (i,j) of the window and $a_i, 1 \leq i \leq n$, be intensity values of its neighbors, where



a_1	a_2	a_3
a_4	a_0	a_5
a_6	a_7	a_8

(a) $N=3$

a_1	a_2	a_3	a_4	a_5
a_6	a_7	a_8	a_9	a_{10}
a_{11}	a_{12}	a_0	a_{13}	a_{14}
a_{15}	a_{16}	a_{17}	a_{18}	a_{19}
a_{20}	a_{21}	a_{22}	a_{23}	a_{24}

(b) $N=5$

Figure 1. $N \times N$ windows with labeled pixels.

n denotes the number of neighbors. Then the entropy H of the intensity in the region is defined as

$$H = -\sum_{i=0}^n p_i \log p_i / \log(n+1), \quad (1)$$

where the normalized intensity p_i is defined by

$$p_i = a_i / \sum_{j=0}^n a_j. \quad (2)$$

The entropy H is normalized so that $0 \leq H \leq 1$. Smooth regions give similar p_i 's, yielding large entropy values. We calculate this entropy value in every region and detect the regions where the entropy value is small. The edges in the current frame can be produced as the regions under a certain threshold.

The cross-entropy C between two probability density functions (pdfs) $p(x)$ and $q(x)$ is defined by

$$C = \int p(x) \log(p(x)/q(x)) dx. \quad (3)$$

This cross-entropy has a unique property as a measure of information dissimilarity, and cross-entropy minimization has unique properties as an inference method.⁴ In essence, minimizing cross-entropy is equivalent to maximizing entropy when the prior distribution is uniform.

The directed divergences of p and q are the two integrals in the functional $F(p, q)$ expressed as⁵

$$F(p, q) = A \int q(x) \log \frac{q(x)}{p(x)} dx + B \int p(x) \log \frac{p(x)}{q(x)} dx \quad (4)$$

in which the sum, with $A = B = 1$, signifies the divergence. In (4), if $A = 0$ and $B = 1$, then $F(p, q)$ represents cross entropy.⁴

Cross-entropy has properties that are desirable as an information measure and can be characterized axiomatically in the continuous cases.⁵ Cross-entropy has been applied primarily to statistics, but also to statistical mechanics, chemistry, pattern recognition, and spectral analysis.

3. Proposed Moving Edge Extraction Based On Cross-Entropy

Two moving edge detection methods are presented depending on the schemes employed for edge detection in the current frame. In the first method, we extract the edges using entropy and in the second, using LoG, and then in both, we used cross-entropy method to detect moving edges.

3.1 Edge Extraction Using Entropy in the Current Frame

First, we calculate entropy in order to obtain edges in the current frame. We use eight neighboring pixels in the 3×3 window or 24 neighboring ones in the 5×5 window, as shown in Fig. 1. The image is raster scanned and the entropy is computed over the region covered by the $N \times N$ window. Then edges are detected by selecting the regions having entropy values smaller than the threshold value.

Second, the cross-entropy between the current and previous frames is calculated, in which the cross-entropy is used as a measure of dissimilarity between two successive frames. Since the probability value is discrete in a digital picture, the cross-entropy defined by

$$C = \sum_{l=0}^n p_l \log(p_l / q_l) \quad (5)$$

is employed, where p_l and q_l are defined as in Eq. (2) with p_l (q_l) computed in the current (previous) frame. The cross-entropy is calculated in every region, then the image is divided into two regions based on it. The region with a small cross-entropy value is the one which has relatively small change, whereas the region with a large cross-entropy value is the one which has large dissimilarities between two frames, possibly containing a part of moving objects. Thus we can obtain the picture similar to the difference picture (DP).

Next we combine the result of entropy with that of cross-entropy in order to obtain moving edges. The pixel which has the entropy value H smaller than the threshold T_1 and the cross-entropy value C larger than the threshold T_2 is detected as a moving edge point. The binary moving edge map $E(i, j)$ can be expressed by

$$E(i, j) = \begin{cases} 1 & (\text{movingedge}), \\ 0 & (\text{non-movingedge}), \end{cases} \quad \begin{matrix} \text{if } (H < T_1) \text{ and } (C > T_2) \\ \text{otherwise}, \end{matrix}$$

where $E(i, j)$ is the binary moving edge map, and threshold values, T_1 and T_2 are tuned experimentally. If T_2 is large value, only the edges with large motion are detected, otherwise the edges even with small motion can be detected. To remove the false edges by noise signals,



we employ Gaussian preprocessing for the current frame $I(i, j)$. The extracted edges using entropy with Gaussian filtering shows remarkably reduced false edges for noisy images. That is, the edges in the current frame $I(i, j)$ can be extracted by the entropy operator H for $G^*I(i, j)$, where G represents Gaussian.

3.2 Edge Extraction Using LoG in the Current Frame

Alternatively, we extract edges in the current frame using LoG which is known as performing efficiently to detect edges. Using this, edges in the current frame $I(i, j)$ satisfy⁶

$$D^2G * I(i, j) = 0$$

where D^2 and G represent the second-derivative operator and Gaussian defined by $G(r) = (\pi\sigma^2 / 2)\exp(-r^2 / 2\sigma^2)$, respectively, where $r = \sqrt{i^2 + j^2}$ and c is Gaussian spread parameter.

So we can extract the moving edge by combining the result of LoG with that of cross-entropy. If the pixel is classified into an edge by LoG, and has the cross-entropy value C larger than the threshold T , it is considered as a moving edge point:

$$E(i, j) = \begin{cases} 1 & \text{if (edge by LoG) and } (C > T) \\ 0 & \text{otherwise.} \end{cases}$$

where $E(i, j)$ is the binary moving edge map. The threshold T can be set depending on the probability of moving edges.

4. Simulation Results and Discussions

Image sequences of each size, 352x240, uniformly quantized at eight bits are generated to test the effectiveness of the proposed method. Fig. 2 shows the simulation results of moving edge extraction with a 3x3 window for the synthesized test sequence using entropy and cross entropy. Fig. 2(a) is the original current frame and Fig. 2(b) shows the DP between current and previous frames. Fig. 2(c) illustrates the edge image of Fig. 2(a) and Fig. 2(d) shows the moving edge map obtained by entropy and cross-entropy. Fig. 2 shows that the proposed method using entropy and cross entropy can exactly extract the moving edges.

Fig. 3 shows the simulation results of moving edge extraction using entropy and cross entropy for the noisy synthesized test sequence. Fig. 3(a) is the original current frame and Fig. 3(b) illustrates the DP between current and previous frames. Fig. 3(c) shows the edge image of Fig. 3(a) with a 3x3 window, and Fig. 3(d) shows the moving edge map obtained by entropy and cross-entropy. Fig. 3(e) shows the edge image after Gaussian filtering of Fig. 3(a) and Fig. 3(f) shows the moving edge map obtained by entropy and cross-entropy after Gaussian filtering. From Fig. 3, it is noted that preprocessing by Gaussian filtering can remarkably reduce false edges.

Figs. 4 and 5 show results of moving edge detection using entropy and cross-entropy with 3x3 and 5x5 windows, respectively.

Figs. 4(a) and 5(a) are the original current frames of the Table Tennis sequence. Figs. 4(b) and 5(b) illustrate the DP between current and previous frames. Figs. 4(c) and 5(c) show edge images of Figs. 4(a) and 5(a), respectively, obtained by thresholding the entropy value. Figs. 4(d) and 5(d) show the moving edge maps obtained using the concept of both entropy and cross-entropy. In Figs. 4 and 5, the edge detection using the 5x5 window shows less false edges compared with that using the 3x3 window.

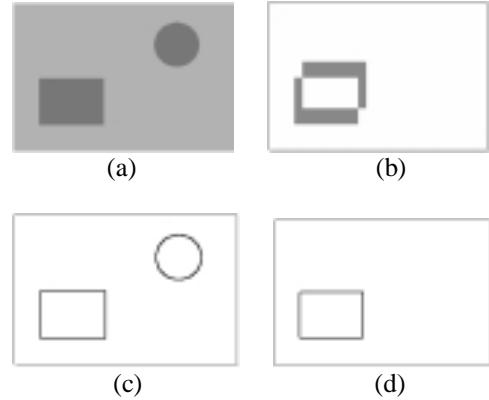


Figure 2. Result of moving edge extraction using entropy and cross-entropy (3x3 window) for the synthesized test sequence. (a) Original image. (b) DP (magnified by a factor of two). (c) Edge image of (a). (d) Moving edge map.

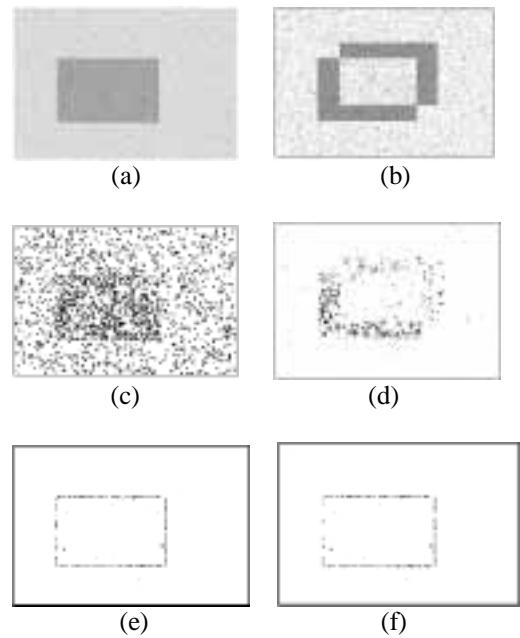


Figure 3. Result of moving edge extraction using entropy and cross-entropy (3x3 window) for the synthesized noisy test sequence. (a) Original image. (b) DP (magnified by a factor of two). (c) Edge image of (a). (d) Moving edge map. (e) Edge image of (a) after Gaussian filtering. (f) Moving edge map after Gaussian filtering.

Fig. 6 also shows the simulation results with a 5x5 window for moving edge extraction using cross entropy and LoG. Fig. 6(a) shows the original current frame of the Table Tennis sequence. Fig. 6(b) illustrates the DP between the current and previous frames. Fig. 6(c) shows the edge image of Fig. 6(a) obtained by LoG. Fig. 6(d) shows the edge image of Fig. 6(a) obtained by LoG. shows the moving edge picture obtained using LoG and cross-entropy. In Fig. 6, the moving edge detection using cross-entropy and LoG can extract the moving edges appropriately by controlling a Gaussian smoothing factor.

As shown in Figures, the moving edge extraction method using cross-entropy and entropy shows thick edges, whereas the moving edge extraction method using cross-entropy and LoG represents thin edges. It is also noted that the proposed moving edge extraction methods using cross-entropy can extract moving edges satisfactorily.

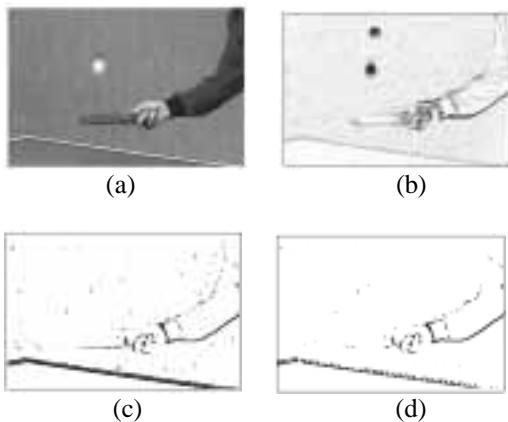


Figure 4. Result of moving edge extraction using entropy and cross-entropy (3x3 window) for the Table Tennis sequence. (a) Original image. (b) DP (magnified by a factor of two). (c) Edge image of (a). (d) Moving edge map.

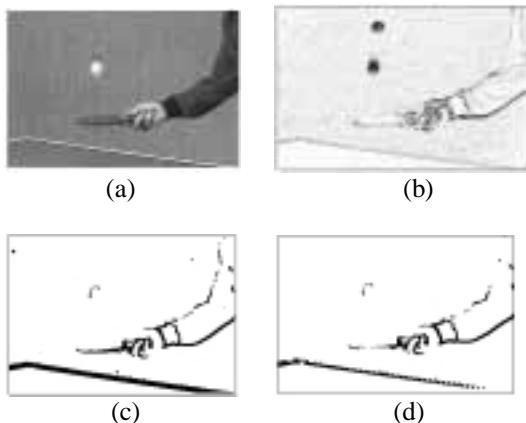


Figure 5. Result of moving edge extraction using entropy and cross-entropy (5x5 window) for the Table Tennis sequence. (a) Original image. (b) DP (magnified by a factor of two). (c) Edge image of (a). (d) Moving edge map.

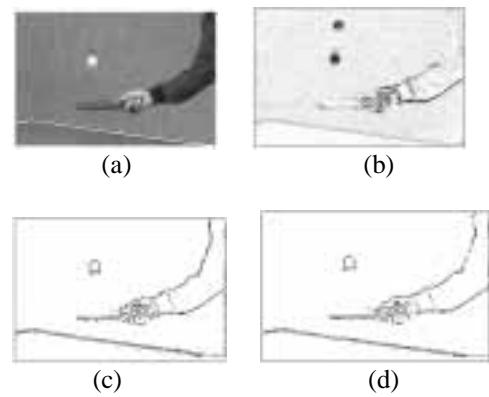


Figure 6. Result of moving edge extraction using LoG and cross-entropy (5x5 window) for the Table Tennis sequence. (a) Original image. (b) DP (magnified by a factor of two). (c) Edge image of (a). (d) Moving edge map.

5. Conclusions

In this paper, the cross-entropy concept is presented to detect the dissimilarities between two frames of dynamics scenes. It is also shown that the moving edge can be obtained by combining the concept of entropy and cross-entropy or combining LoG and cross-entropy. Computer simulations with various test sequences show the effectiveness of the proposed methods. Future work will focus on the analysis of various dynamic scenes using entropy and cross-entropy with verification by computer simulations.

6. References

- [1] R. Weeks, Jr., *Fundamentals of Electronic Image Processing*, SPIE Optical Engineering Press, Bellingham, WA, 1996.
- [2] S. M. Haynes and R. Jain, "Detection of moving edges," *Computer Vision, Graphics, and Image Processing*, vol. 21, pp. 345-367, Apr. 1983.
- [3] A. Shiozaki, "Edge extraction using entropy operator," *Computer Vision, Graphics, and Image Processing*, vol. 36, pp. 1-9, Oct. 1986.
- [4] J. E. Shore and R. W. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Trans. Information Theory*, vol. IT-26, no. 1, pp. 26-37, Jan. 1980.
- [5] R. W. Johnson, "Axiomatic characterization of the directed divergences and their linear combinations," *IEEE Trans. Information Theory*, vol. IT-25, no. 6, pp. 709-716, Nov. 1979.
- [6] R. Kasturi and R. C. Jain, *Computer Vision: Principles*, IEEE Computer Society Press, Los Alamitos, CA, 1991.



A New Model for Edge Detection in Digital Images

B. Bouda, Lh. Masmoudi, D. Aboutajdine

Faculty of Sciences, Mohammad V University,

4 Avenue Ibn Battouta, B.P. 1014 RP, Rabat, Morocco

brahimbouda777@gmail.com, [masmoudi, aboutaj]@ fsr.ac.ma

<http://www.fsr.ac.ma>

Abstract

Edge detection and feature extraction are widely used in image processing and computer vision applications. In this paper we propose a new model for detecting edges in digital images. The basic idea is to model the image as a grid of the electric charges of a plan surface in electrostatic balance and, to affect for each block of size 3x3 pixels, a negative sign to the pixels with gray levels greater than a calculating threshold. The method operates in two phases: the first one consists in establishing the neighbourhood forces exerting on the central pixel for each defined block of the image, and the second one determines the horizontal and vertical masks corresponding to the resulting forces followed by the convolution operation.

In order to validate and to illustrate the interest of the proposed approach, some experiments were made on a collection of synthetic and real images. As an objective quantitative measure, we have used Pratt's Figure of Merit to compare edge preservation performances to other methods.

Keywords: Edge detection, Charge electrostatic, Figure of merit, Masks, Model.

1. Introduction

The edges research in an image is one of the most studied problems since the origin of work in image processing. According to the fixed purpose, many methods were developed in this area. This is due to the very intuitive nature of edges which naturally seem as the ideal visual indices in the greatest part of the situations. They are then effective descriptors of the change of image intensity. These methods can be grouped in two classic approaches: one consists in finding the homogeneous areas of the image and the other which consists in seeking the border of the objects. Although these two approaches are dual, they are also complementary. In this sense, several edges detection algorithms focus on this first aspect were developed, such the Gradient, Sobel, Prewitt, Canny and Deriche. They can be implemented as a series of image convolutions by using variable weight in the convolution kernel. These techniques often analyze

an image in its global nature by using the same convolution kernel, without taking into account the specificities of the various image blocks. As consequence, the edges are well detected in certain zones and badly detected in others within the same image. To achieve the operation of edges detection and remedy this established fact, it was necessary to introduce adaptive methods [10, 11].

The objective of this article is to propose an adaptive approach of edges detection basing on the design of an ideal model. The basic idea consists in establishing for the images a similar model to that of the electrostatics. So, for a given image we determine the average of every block of size 3x3 pixels. The gray levels of pixels which are greater than this threshold (the average) will be assigned by a negative sign. Afterward, we exploit the electrostatics fundamental law and the forces of attraction and repulsion between the electric charges to define homologous equations for the image blocks. With this analogy we build a filter (vertical mask and horizontal mask) relative to each block and adapted to the edges detection. By a convolution product of the blocks image and the determined filters, the edge detection is made. We notice that the modelled filter is satisfying not only on the synthetic images but also on images of the real world. The comparison of our results with those obtained with the Sobel and Canny's algorithm indicates the difference in edges quality; translate the validity of the adopted model.

The rest of this article is organized as follows: Section (2) presents a state of the art and the fundamental tools. The modelling of the image and the construction of the masks in blocks image are presented in section (3). The quantitative evaluation procedure is presented in section (4), whereas, the results of experimentation of the proposed detection approach and the conclusion are presented in sections (5) and (6) respectively.

2. State of the art and fundamental tools

An edge is in general a border which separates the adjacent zones of image having distinct brightness. The development of an edge detector is often based on a



specific characteristic of the image. First order operators: the gradient of Roberts, the masks of Prewitt [13] and of Sobel [14] and the operator of the second order (Laplacian), are well adapted to the images without noise. The very popular mask of Sobel has been proved to be well adapted to the edge detection in this case. However, it is sensitive to a noisy image. The filters of Canny [2] and Deriche [3, 4] are developed while based on an edge model of the level type, whereas, filters of Bourennane [1], Laggoune [9], and of Gouton [5] use the edge of the crawl and peak type respectively. These filters are well adapted to the noisy images by an adjustment of one or more parameters. We propose in this part an adaptive approach of edge detection by using an electrostatics theoretical model.

Lets now define the concept on witch based the proposed approach.

Consider tow electric charges q_0 and q_1 distant of the distance r . there is an electrostatic field between these two fixed charges. The charge q_1 generates an electrostatic field \vec{E}_1

$$\vec{E}_1 = \frac{q_1}{4\pi\epsilon_0} \frac{\vec{r}}{r^3} \quad (1)$$

The force exerted by the field \vec{E}_1 on q_0 is \vec{F}_{q_1}/q_0 such as:

$$\vec{F}_{q_1}/q_0 = q_1 \cdot \vec{E}_1 = \frac{q_0 q_1}{4\pi\epsilon_0} \frac{\vec{r}}{r^3} \quad (2)$$

The equation (2) can be put in the form:

$$\vec{F}_{q_1}/q_0 = k \cdot q_0 \cdot q_1 \frac{\vec{r}}{r^3} \quad \text{where } k = \frac{1}{4\pi\epsilon_0} \quad (3)$$

When we have a whole of charges q_i , $i = 1, 2 \dots m$, distant from the charge q_0 by distances r_i , then each charges exert an electric force on q_0 inversely proportional to r_i .

3. Image modelling by the electrostatic model

We can model the image as a grid points of electric charges uniformly distributed on a plan surface in electrostatic balance, and we divide it in blocks of size 3x3 pixels. For each blocks, we make the following analogy: the gray levels of pixels correspond to the electric charges and the central charge is subjected to the electrostatic forces of the eight neighbouring charges. The components of the electric forces could be calculated as well by using the equation (3). We distinguish between two cases: the repulsive and the attractive forces. In electrostatic balance we can write:

$$\sum_{i=1}^n \vec{F}_i = \vec{0} \quad (4)$$

3.1. Attractive forces

In a reference R (q_0, \vec{X}, \vec{Y}), the charges q_i , $i=1,2,\dots,8$ and q_0 have an opposite sign, q_i exercise an attractive force on q_0 . The figure 1 illustrates this situation.

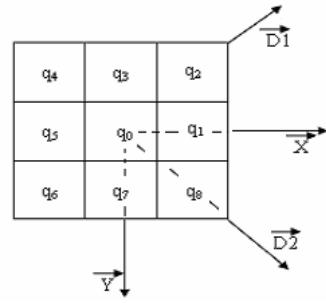


Figure 1. Electric charges q_j and q_0 having the opposite sign.

Let \vec{H} and \vec{V} the respectively horizontal and vertical forces, r : the distance between q_j and q_0 if j is odd and δ : this distance if j is even.

$$\vec{H} = \vec{F}_{q_1}/q_0 + \vec{F}_{q_5}/q_0 = \frac{k q_0}{r^2} (q_5 - q_1) \vec{X} \quad (5)$$

$$\vec{V} = \vec{F}_{q_3}/q_0 + \vec{F}_{q_7}/q_0 = \frac{k q_0}{r^2} (q_3 - q_7) \vec{Y} \quad (6)$$

Let \vec{ED}_1 and \vec{ED}_2 the diagonal forces:

$$\vec{ED}_1 = \vec{F}_{q_2}/q_0 + \vec{F}_{q_6}/q_0 = \frac{k q_0}{\delta^2} (q_6 - q_2) \vec{D}_1 \quad (7)$$

$$\vec{ED}_2 = \vec{F}_{q_8}/q_0 + \vec{F}_{q_4}/q_0 = \frac{k q_0}{\delta^2} (q_4 - q_8) \vec{D}_2 \quad (8)$$

\vec{D}_1 and \vec{D}_2 can be expressed according to X and Y as follow:

$$\vec{D}_1 = \cos(\psi) \vec{X} - \sin(\psi) \vec{Y} \quad \text{and} \quad \vec{D}_2 = \cos(\psi) \vec{X} + \sin(\psi) \vec{Y} \quad (9)$$

Where $\Psi = \pi/4$ and $\delta^2 = 2r^2$

The equations (7) and (8) become respectively:

$$\vec{ED}_1 = \frac{k q_0 \sqrt{2}}{4r^2} (q_6 - q_2) [\vec{X} - \vec{Y}] \quad (10)$$

$$\vec{ED}_2 = \frac{k q_0 \sqrt{2}}{4r^2} (q_4 - q_8) [\vec{X} + \vec{Y}] \quad (11)$$

Let $P(i, j)$ the central pixel of the defined block, we can write:

$$q_0 = P(i, j) \quad (12)$$

The equations (5), (6), (10) and (11) become:

$$\vec{H} = \frac{k P(i, j)}{r^2} (q_5 - q_1) \vec{X} \quad (13)$$

$$\vec{V} = \frac{k P(i, j)}{r^2} (q_3 - q_7) \vec{Y} \quad (14)$$



$$\vec{ED}_1 = \frac{k\sqrt{2} P(i, j)}{4r^2} (q_6 - q_2)[\vec{X} - \vec{Y}] \quad (15)$$

$$\vec{ED}_2 = \frac{k\sqrt{2} P(i, j)}{4r^2} (q_4 - q_8)[\vec{X} + \vec{Y}] \quad (16)$$

Let:

$$\vec{F}_x = F_x \cdot \vec{X} \quad \text{and} \quad \vec{F}_y = F_y \cdot \vec{Y} \quad (17)$$

Thus:

$$F_x \vec{X} = \frac{k P(i, j)}{r^2} [q_5 - q_1 + \frac{\sqrt{2}}{4} (q_6 - q_2 + q_4 - q_8)] \vec{X} \quad (18)$$

$$F_y \vec{Y} = \frac{k P(i, j)}{r^2} [q_3 - q_7 + \frac{\sqrt{2}}{4} (q_2 + q_4 - q_6 - q_8)] \vec{Y} \quad (19)$$

The total force and the norm of this force are:

$$\vec{FT} = F_x \vec{X} + F_y \vec{Y} \quad \text{and} \quad \|\vec{FT}\| = \sqrt{F_x^2 + F_y^2} \quad (20)$$

From the equations (18) and (19) we obtain the two convolution masks G_{xa} and G_{ya} respectively in the direction of X and of Y shown in figure 2.

$\frac{\sqrt{2}}{4}$	0	$-\frac{\sqrt{2}}{4}$
1	0	-1
$\frac{\sqrt{2}}{4}$	0	$-\frac{\sqrt{2}}{4}$

 $G_{xa} =$

$\frac{\sqrt{2}}{4}$	1	$\frac{\sqrt{2}}{4}$
0	0	0
$-\frac{\sqrt{2}}{4}$	-1	$-\frac{\sqrt{2}}{4}$

 $G_{ya} =$

$\frac{\sqrt{2}}{4}$	1	$\frac{\sqrt{2}}{4}$
0	0	0
$-\frac{\sqrt{2}}{4}$	-1	$-\frac{\sqrt{2}}{4}$

Figure 2. Horizontal and vertical filters determined by the attractive forces.

3.2. Repulsive forces

In this case we suppose that charges q_i , $i=1, 2\dots 8$ and q_0 have the same sign. Each charge q_i exerts a repulsive force on q_0 . After the theoretical development of the equations, we ended the two directional filters G_{xr} and G_{yr} shown in the figure 3.

$-\frac{\sqrt{2}}{4}$	0	$\frac{\sqrt{2}}{4}$
-1	0	1
$-\frac{\sqrt{2}}{4}$	0	$\frac{\sqrt{2}}{4}$

 $G_{xr} =$

$-\frac{\sqrt{2}}{4}$	-1	$-\frac{\sqrt{2}}{4}$
0	0	0
$\frac{\sqrt{2}}{4}$	1	$\frac{\sqrt{2}}{4}$

 $G_{yr} =$

$-\frac{\sqrt{2}}{4}$	-1	$-\frac{\sqrt{2}}{4}$
0	0	0
$\frac{\sqrt{2}}{4}$	1	$\frac{\sqrt{2}}{4}$

Figure 3. Horizontal and vertical filters determined by the repulsive forces.

3.3. Generalization

The algorithm of the proposed approach is as follow:

- 1: For each block of size 3x3 pixels of the image, calculate the average (threshold) and attribute a negative sign to the gray levels of pixels greater than the calculated threshold.

2: Calculate the four horizontal and vertical forces due to the four neighbouring pixels according to four connexity by using the equation (13) and (14).

3: Calculate the four diagonal forces due to the four neighbouring pixels according to both diagonal by using the equation (15) and (16).

4: Determine both masks by using the equation (18) and (19).

5: Carry out an operation of convolution of the each block by the two determined filters.

4. Quantitative evaluation procedure

Several comparison methodologies exist for edge detection [7, 8, 12]. As an objective quantitative measure and to compare edge preservation performances we have chosen Pratt's Figure of Merit [12]. This measure has often been used in edge detection evaluation which we will use to compare our results with those of others in case the test images are the same. Pratt's Figure of Merit (FOM) also attempts to balance three types of errors that can produce erroneous edge maps: missing valid edge points, failure to localize edge points and classification of noise fluctuations as edge points. It is defined as:

$$FOM = \frac{1}{\max(I_D, I_I)} \sum_{i=1}^{I_D} \frac{1}{1 + \alpha d_i^2} \quad (21)$$

In this equation, I_D and I_I are the number of detected and ideal edge pixels respectively, and the parameter α is a scaling constant typically set to 1/9, while d is the Euclidean distance from an actual edge pixel to the nearest ideal edge pixel. FOM ranges between 0 and 1, with unity for ideal edge detection. Note that the Pratt's Figure of Merit strongly depends on what method is used to obtain a binary edge map and the truth ground image.

5. Simulations and results

Lena, Tiffany and Office images were selected for our tests because they have within each same image various contours and contrasts densities. The diversity of these images can appreciate the limits of the proposed method.

Our first experiment has for objective to test the proposed algorithm on these selected images and to compare the edge detection to those obtained by the conventional filters such as Sobel and Canny algorithm (figure 4, 5 and 6). The comparison shows the edges quality presented by the proposed approach. To compare edge preservation performances, we adopt the Pratt's Figure of Merit. Note that this calculation is applicable only when the truth ground is known. So, we use a synthetic image which we know the truth ground edge image (figure 7). This image is subjected to the three algorithms of edge detectors and the measure of detection results similarity with the awaited result is given. As shows in table 1, the FOM of the proposed algorithm is slightly higher with the best score.



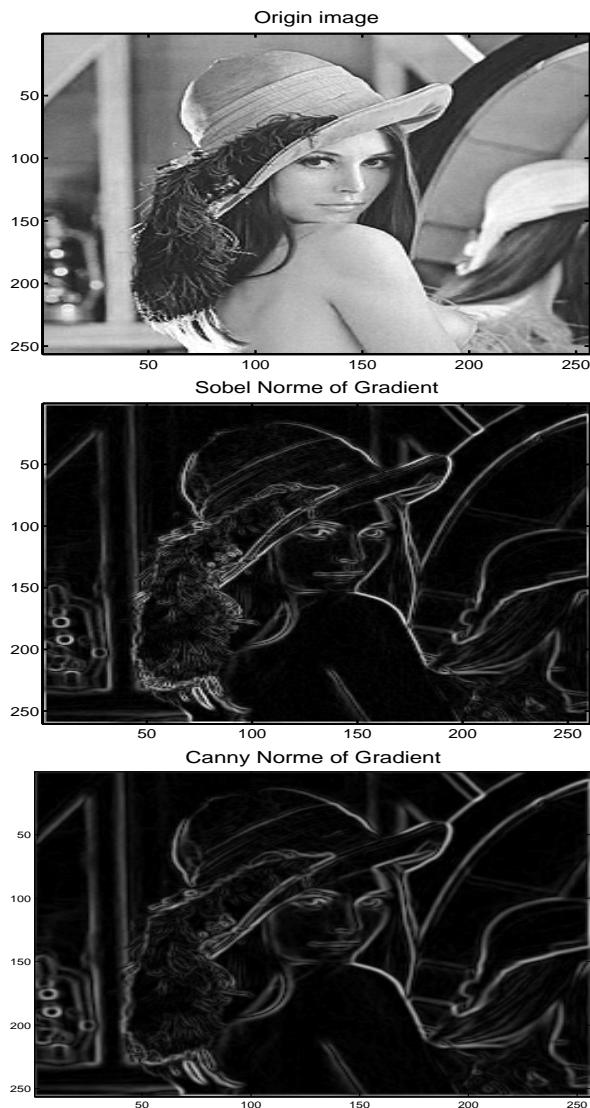


Figure 4. Edge for Lena image, in the order: origin image, Sobel measure, Canny and proposed measure.

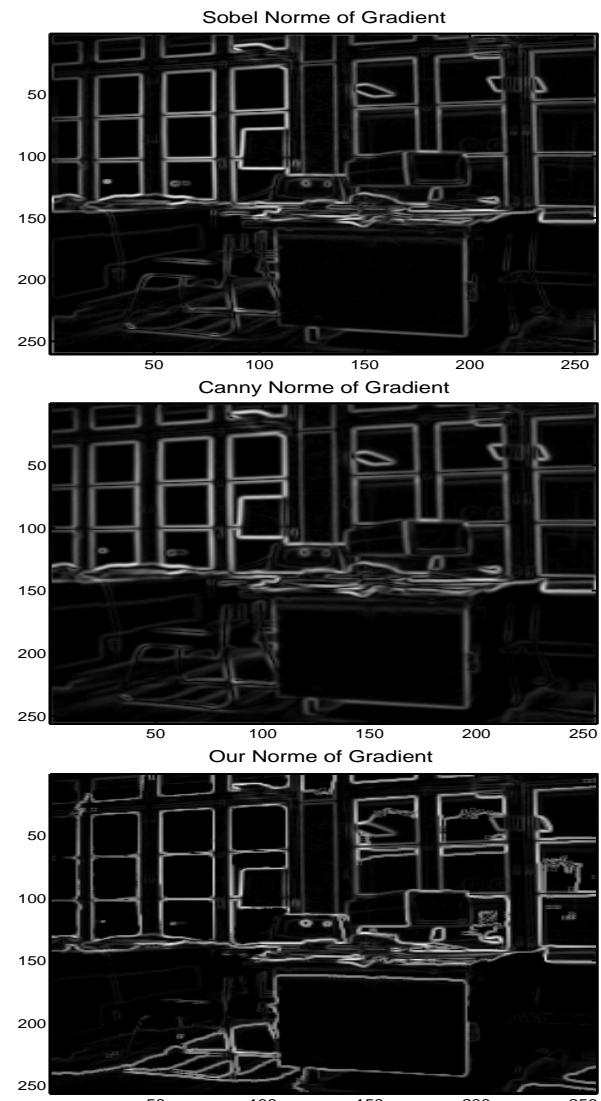
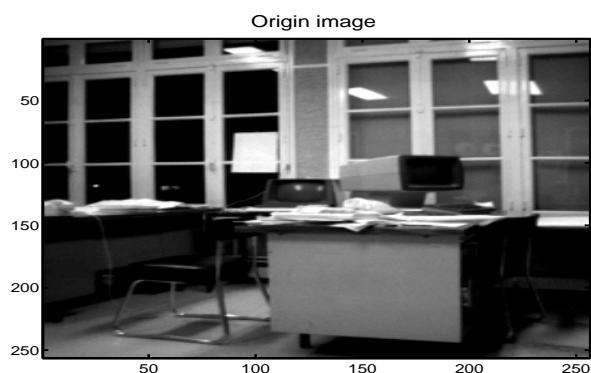
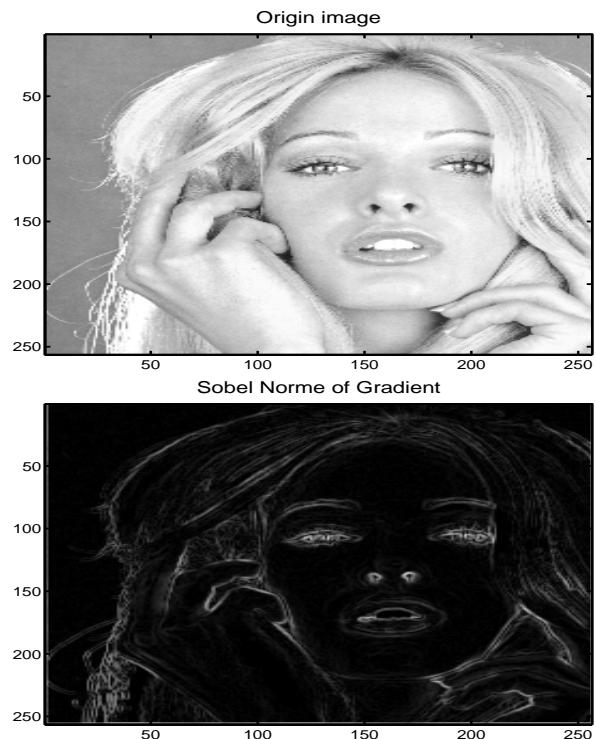


Figure 5. Edge for office image, in the order: origin image, Sobel measure, Canny and proposed measure.



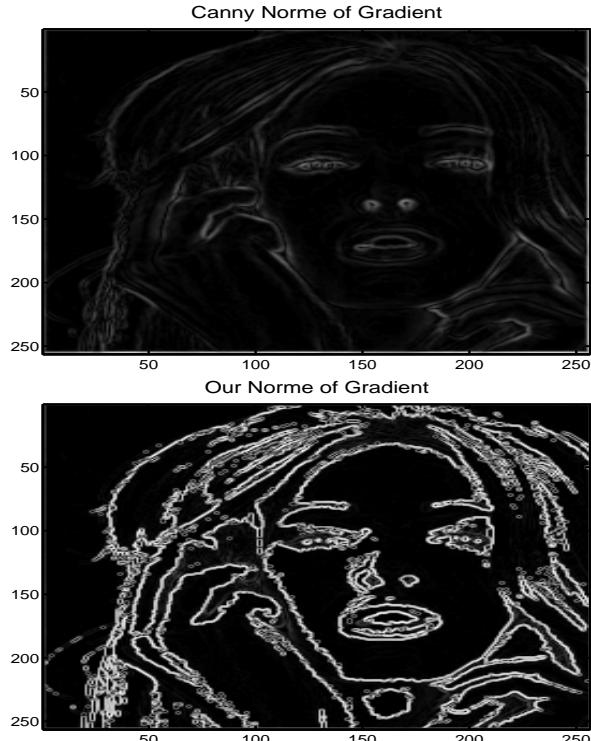


Figure 6. Edge for Tiffany image, in the order: origin image, Sobel measure, Canny and proposed measure.

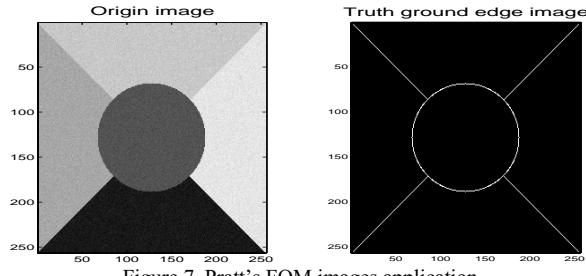


Figure 7. Pratt's FOM images application.

Table 1. Pratt's FOM for norm of gradient methods.

Methods	FOM
Sobel	0.267
Canny	0.068
The proposed method	0.394

Our second experiment (figure 8) has for objective to thin the edges (thinning) after a preliminary thresholding. In order to make better comparison between the three methods we chose the threshold $\alpha = 0.2$ which is adapted to the Canny's algorithm. We indicate that the Canny's parameters are adjusted in order to obtain better possible results. The comparison with the edge detectors shows the quality of the proposed method.

The figure 9 summarizes the edge preservation performance of the three edges detectors techniques versus threshold parameters by using the same synthetic image shown in figure 7. As shown, the best edge detection result according to the FOM score is often obtained with the proposed algorithm. For Sobel and Canny's method, the edge conservation performance degrades considerably as the threshold parameter increases. Consequently the choice of this parameter is very meticulous and difficult to realize. On the contrary,

for the proposed method, the choice of this parameter is not necessary to be determined. The edge preservation performance remains acceptable up to value 0.5 of the threshold parameter with a FOM in the neighborhood of value 0.7.

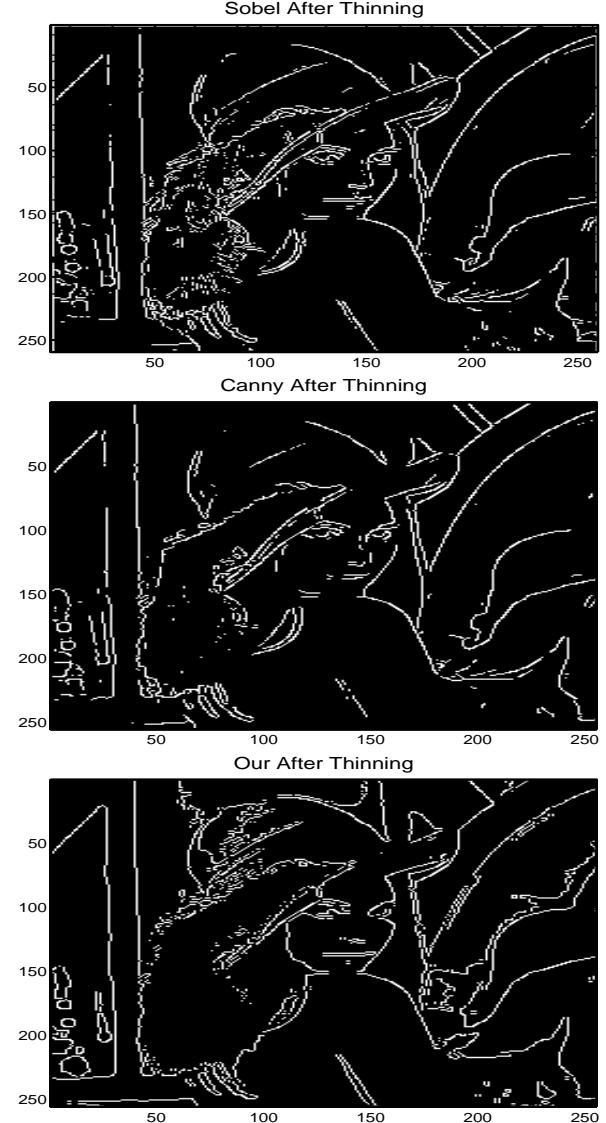


Figure 8. Edges for Lena image after thresholding with $\alpha = 0.2$, in the order: origin image, Sobel measure, Canny and the proposed measure

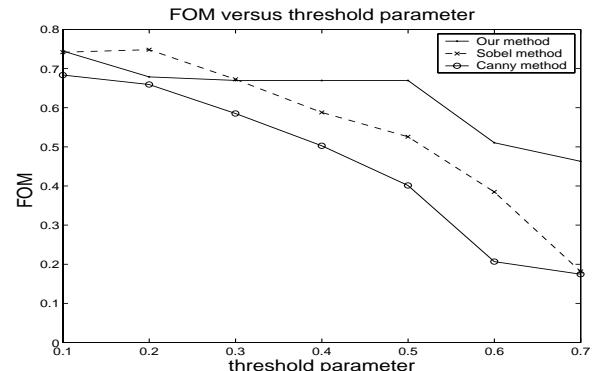


Figure 9. Pratt's FOM for edge methods after thinning versus threshold parameter.



6. Conclusion and perspectives

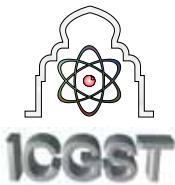
In this work we tested the proposed method for edge detection on several images having different configurations and variant gray levels. The method has been evaluated with the Pratt's procedure, and the obtained results are relatively satisfactory in comparison to the conventional methods.

The proposed model can be also applied for edge detection in colours images and transposed to several contexts in image processing and computer vision applications.

7. References

- [1] E. Bourennane, P. Gouton, M. Paindavoine, F. Truchetet. Generalization of Canny-Deriche filters for detection of noisy exponential edge. *Signal Processing*, 82(10): 1317–1328, October 2002.
- [2] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.
- [3] R. Deriche. Fast algorithms for low-level vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):78–88, January 1990.
- [4] R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2): 167–187, May 1987.
- [5] P. Gouton, H. Laggoune, Rk. Kouassi, M. Paindavoine. Optimal detector for crest lines. *Optical Engineering*, 39(6): 1602–1611, June 2000.
- [6] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer. A robust visual method for assessing the relative performance of edge-detection algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(12):1338–1359, December 1997.
- [7] R. Horaud, O. Monga. Vision par ordinateur, outils fondamentaux. Editions Hermès – Série Informatique: Paris, 2^{ème} édition, Août 1995.
- [8] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer. Comparison of edge detectors: Methodology and initial study. *Computer Vision and Image Understanding*, 69(1):38–54, January 1998.
- [9] H. Laggoune. Contributions à l'étude des filtres optimaux pour des contours 2D et 3D quelconques. Ph.D. thesis, Université de Bourgogne, Dijon, France, Mars 1996.
- [10] S. G. Mallat and S. Zhong. Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(7):710–732, July 1992.
- [11] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.
- [12] W. K. Pratt. *Digital Image Processing*, 2nd Edition, John Wiley & Sons, Inc.: New York, 1991.
- [13] J. M. S. Prewitt. Object enhancement and extraction. In B. S. Lipkin and A. Rosenfeld (Eds.), *Picture Processing and Psychopictorics*, pages 75–149. Academic Press, New York, 1970.
- [14] I. Sobel. An isotropic image gradient operator, In H. Freeman (Eds.), *Machine Vision for Three-Dimensional Scenes*, pages 376–379. Academic Press, 1990.





A New Grey Level Corner Detection Based on Electrostatic Model

B. Bouda¹, Lh. Masmoudi², D. Aboutajdine¹

¹*Faculty of Sciences, Mohammad V University, GSCM-LRIT Laboratory,*

4 Avenue Ibn Battouta B.P. 1014, Rabat, Morocco

²*Faculty of Sciences, Mohammad V University, LETS Laboratory,*

4 Avenue Ibn Battouta B.P. 1014, Rabat, Morocco

brahimbouda777@gmail.com, [masmoudi, aboutaj]@fsr.ac.ma

<http://www.fsr.ac.ma>

Abstract

Corner detection and feature extraction are widely used in image processing and computer vision applications. In this paper, we propose a new grey level corner detection based on electrostatic model having different configurations, which demonstrate the best detection. The basic idea is to model the image as a grid of electric charges of a plan surface in electrostatic balance.

The algorithm is achieved in two main steps. The first one uses an adaptive method for computing directional gradients which permits to build an edge detection operator of 3x3 pixels size. The second one introduces this operator for determining image derivatives in order to detect the corners. The smoothed squared images derivatives are given by using a Gaussian function. Then, the response function proposed by Harris and Stephens is calculated for each pixel, using auto-correlation matrix. The local maxima search is performed as a final step of the method.

Simulation results on synthetic and real images show the performances of the proposed approach compared to that presented by an improved version of Harris detector.

Keywords: Adaptive methods, Corner detection, Edge detection, Models.

1 Introduction

Corner detection is a domain which attracted many researchers and consequently, several methods have been developed in this area. Nevertheless, the choice of an accurate method remains the major problem. These methods can be classified in three categories [6, 7, 16, 18]: Contour approaches; the idea is at first to detect the contours of image before extracting the

corners. Parametric model approaches fit intensity model to signal. They often provide sub-pixel accuracy, but they are limited to specific kinds of corners. For the intensity approaches, the idea is to compute a measure that indicates the presence of a corner directly from the grey values.

The evaluation of corner detectors presented in [6] demonstrates an excellent performance of Harris and Stephens detector [3], compared to other existing approaches [11, 13, 14, 15]. Harris and Stephens improved the approach of Moravec [17] by using the auto-correlation matrix. The use of directions and discrete shifts is avoided. Instead of using a simple sum, a Gaussian is applied to weight the derivatives inside the window. Corners are detected if the auto-correlation matrix has two significant eigenvalues. Several adaptations were proposed to improve this detector. Schmid [6] is one of them that propose an improved version by computing the auto-correlation matrix with Gaussian masks. The localization problem is reduced, the detector is robust with respect to the camera viewpoint and to the illumination variations, but it became sensitive to noise. In [2] we have developed an improved version of the Harris detector by introducing a preprocessing stage based on filtering before computing the corners. The results are compared with those obtained by the Harris and Stephens detector. The method demonstrates the best detection (precision, number of correct points detected and time consuming).

Recently, the authors have proposed a method for detecting edge in grey level images [1]. The method is based on model inspired by the electrostatic one. In this paper, we propose a new corner detector based on this model. It consists to affect for each block of



3x3 pixels size, a negative sign to the pixels with grey levels greater than a calculating threshold. Neighborhood forces exerting on the central pixel for block of the image and filters corresponding to the resulting forces are determined. Directional gradients which permit to build two edge detection operators of 3x3 pixels size are given. To determine image derivatives I_x and I_y , we carried out a convolution of the image by horizontal and vertical edge detector operators. To give smoothed squared image derivatives, we made a convolution of the square image derivatives $(I_x)^2$, $(I_y)^2$ and $I_x * I_y$ by Gaussian filter. The method is achieved by computing for each image pixel the detector response proposed by Harris and Stephens. A local maxima search is performed as a final step of the method.

The rest of this paper is organized as follows: Section (2) presents Harris detector method. The proposed corner detector is presented in section (3), whereas, the results of experimentation of the proposed corner detector and the conclusion are presented in sections (4) and (5) respectively.

2 Harris detector

Harris corner detector method can be described as follows: denoting the image by I , the approximation of its gradient is given using the directional filters by:

$$\begin{cases} dx = [-1 \ 0 \ 1] \\ dy = [-1 \ 0 \ 1]^t \end{cases} \quad (1)$$

We call $I_x(j; l)$ and $I_y(j; l)$ the filtering results respectively of $I(j; l)$ by dx and dy . By the approximation in first order, the average variation of the intensity for the small shift (x, y) using a Gaussian window can be concisely written as:

$$E(x, y) = (x, y) \cdot M \cdot (x, y)^t \quad (2)$$

where the 2x2 symmetric autocorrelation matrix is:

$$M = W(x, y) \otimes \begin{pmatrix} Ix^2 & Ix \cdot Iy \\ Ix \cdot Iy & Iy^2 \end{pmatrix} \quad (3)$$

I_x and I_y are first derivatives of the image and $W(x, y)$ is Gaussian window. The matrix M describes the gradient distribution in a local neighborhood of a point. The method combines the trace and determinant of the second moment matrix to form the following operator:

$$Cornerness_{K_h} = \det(M) - K_h \operatorname{trace}^2(M) \quad (4)$$

K_h is the Harris parameter response. A local maxima search is performed as a final step of the method.

3 Corner Detection in Grey Level Image Using Electrostatic Model

The grey level image I can be modeled as a grid points of electric charges uniformly distributed on a plan surface in electrostatic balance. For each block of 3x3 pixels size we make the following analogy: the grey levels of pixels correspond to the electric charges and the central charge is subjected to the electrostatic forces of the eight neighboring charges.

In a block of 3x3 pixels size, if the q_i charges, $i = \{1, 2, \dots, 8\}$ and q_0 have an opposite (respectively same) sign, q_i exerts an attractive (respectively repulsive) force on q_0 . Figure 1 illustrates this model.

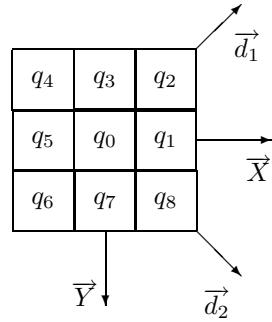


Figure 1: Electrostatic model: q_i and q_0 charges have the opposite (respectively the same) sign.

Let k be a constant and r_i be a distance between an arbitrary electric charge and the central one. Each charge exerts an electric force on q_0 inversely proportional to r_i . Horizontal and vertical forces, when the q_i charges have an opposite sign to q_0 , are given by:

$$\vec{F}_h = \vec{F} q_1/q_0 + \vec{F} q_5/q_0 = \frac{k}{r^2} q_0 (q_5 - q_1) \vec{X} \quad (5)$$

$$\vec{F}_v = \vec{F} q_3/q_0 + \vec{F} q_7/q_0 = \frac{k}{r^2} q_0 (q_3 - q_7) \vec{Y} \quad (6)$$

The diagonal forces are given, respectively, by the following equations:

$$\vec{F}_{d1} = \frac{k}{4r^2} q_0 \sqrt{2} (q_6 - q_2) [\vec{X} - \vec{Y}] \quad (7)$$

$$\vec{F}_{d2} = \frac{k}{4r^2} q_0 \sqrt{2} (q_4 - q_8) [\vec{X} + \vec{Y}] \quad (8)$$

Let $p(i, j)$ the central pixel which corresponds to q_0 charge. The resulting force applied on the central pixel can be represented by their components along the XY coordinate system. Adopt the following inspired formulations:

$$F_X \vec{X} = \frac{k}{r^2} p(i, j) [q_5 - q_1 + \frac{\sqrt{2}}{4} (q_6 - q_2 + q_4 - q_8)] \vec{X} \quad (9)$$



$$F_Y \vec{Y} = \frac{kp(i,j)}{r^2} [q_3 - q_7 + \frac{\sqrt{2}}{4} (q_2 + q_4 - q_6 - q_8)] \vec{Y} \quad (10)$$

From the equations (9) and (10), we deduce two filters dxa and dya shown in figure 2. Following the same development for repulsive forces we get the two filters dxr and dyr shown in figure 3.

$\frac{\sqrt{2}}{4}$	0	$-\frac{\sqrt{2}}{4}$
1	0	-1
$\frac{\sqrt{2}}{4}$	0	$-\frac{\sqrt{2}}{4}$

$\frac{\sqrt{2}}{4}$	1	$\frac{\sqrt{2}}{4}$
0	0	0
$-\frac{\sqrt{2}}{4}$	-1	$-\frac{\sqrt{2}}{4}$

Figure 2: Horizontal filter dxa and vertical filter dya for attractive forces.

$-\frac{\sqrt{2}}{4}$	0	$\frac{\sqrt{2}}{4}$
-1	0	1
$-\frac{\sqrt{2}}{4}$	0	$\frac{\sqrt{2}}{4}$

$-\frac{\sqrt{2}}{4}$	-1	$-\frac{\sqrt{2}}{4}$
0	0	0
$\frac{\sqrt{2}}{4}$	1	$\frac{\sqrt{2}}{4}$

Figure 3: Horizontal filter dxr and vertical filter dyr for repulsive forces.

The method proposed in [1] for detecting edges in grey level images is based on the previous filters. Figure (5) shows an example of edge detection results for the well known house and peppers images depicted in figure (4).



Figure 4: Original grey level images: house, peppers.

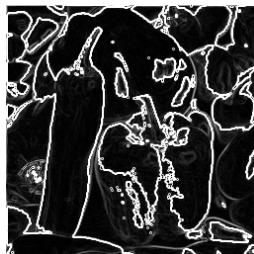


Figure 5: Results of edge detection applied to the grey level images using electrostatic model.

To compare edge preservation performances, the method has been tested and evaluated on several grey

level images using Pratt's Figure of Merit procedure [8]. The edge detection results are relatively satisfactory in comparison to the conventional methods such as Canny [4] and Sobel [5]. Based on these filters, we develop a new corner detection algorithm. It consists to making a convolution operation of the image by using one of both determined filters shown in figure (2) and figure (3). The algorithm is summarized as follows:

- Step 1. Compute the directional filters by using (9) and (10) in order to determine the gradients (the image derivatives) for input grey level image,
- Step 2. Use the Gaussian filter to compute the smoothed squared image derivatives for input grey level image,
- Step 3. Compute the detector response by using (4),
- Step 4. Search the local maximum and use the corner sorting by brightness, then removing the non local maximum.

In step 1, the image derivatives I_{xa} and I_{ya} are calculated and a convolution of the image by a directional filters dxa and dya is made. In step 2, the smoothed squared image derivatives are calculated in order to build the matrix defined in (3). For this reason, we made a convolution of $(I_{xa})^2$ and $(I_{ya})^2$ like $I_{xa} * I_{ya}$ by the Gaussian filter with standard deviation equal to 2. In step 3, the Harris response defined in (4) is calculated for each image pixel. Concerning value of the parameter K_h , experiment shows according to several authors, that an optimum number of corners is obtained by a value $K_h = 0.04$. We adopt this value for processing digital images in the present work. Finally, the local maximum of a pixel in question is calculated and considered when its response is strictly greater than that of the eight neighboring pixels. This research will be followed by corners sorting according of their brightness values.

4 Simulations and results

In this section, we describe results of the experimentation on some kinds of synthetic and real grey level images in two conditions, using the proposed corner detection method:

- Disturbed image: the Gaussian noise of average 0.01 and variance 0.01 is added to original image.
- Affine transformation: the affine transformation of matrix $T_A = [1 \ 0.2; 0.1 \ 1; 0 \ 0]$ is applied to the original image.



We select four grey level images: house and peppers depicted previously in figure (4) and checkerboard images shown in figure (6). They have within each image various contours and contrasts densities. They are subjected to two detectors: the improved Harris detector [2] and the proposed corner detector.

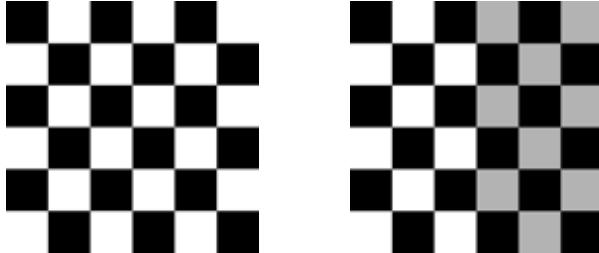


Figure 6: Images of simulation. In the order: checkerboard ((120, 3, 3)>0.5), checkerboard (80, 3, 3).

Firstly, the objective of this experiment is to show how the proposed method provides good corner detection on synthetic images. Figures (7-8) show the corners detected in original images. Figures (9-10) show the corners detected in disturbed images, while figures (11-12) show the corners detected in transformed images. The number of input points for the checkerboard images is 47 and the parameter K_h is 0.04. From the results of these experiments, it can be seen that the proposed corner detector gives good performance and any corners are not missed. The performance of the improved Harris corner detection varies from image to image and condition from condition, and depends on the parameter selections such as size of window and threshold. Therefore, the detector is tested by varying parameter values to obtain the best results. In figures (7-8) many corners missed by the version of Harris are successfully detected by the proposed detector. Noise test is carried out by a Gaussian one as shown in figure (10). Some multiple corners are declared in rounded corner and some others are missed by the version of Harris detector. Two corners are not detected due to view change caused by affine transformation are shown in figure (12).

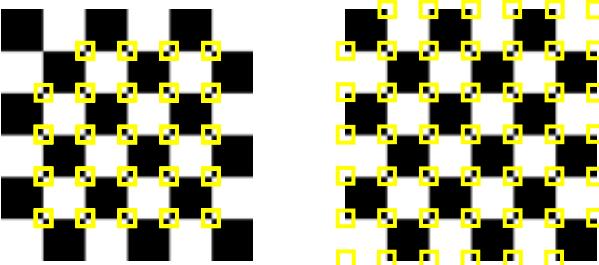


Figure 7: Outputs of corner detectors for original checkerboard ((120, 3, 3)>0.5) image. In the order: improved version of the Harris detector, proposed corner detector.

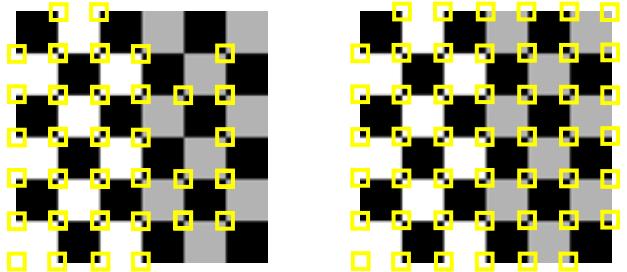


Figure 8: Outputs of corner detectors for original checkerboard (80, 3, 3) image. In the order: improved version of the Harris detector, proposed corner detector.

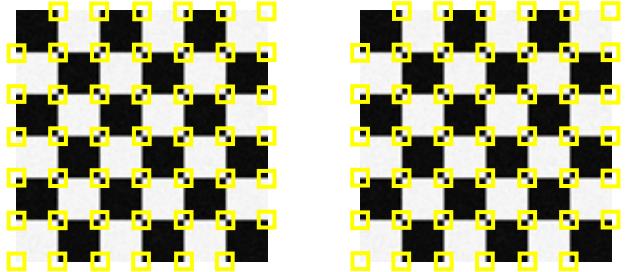


Figure 9: Outputs of corner detectors for a disturbed checkerboard ((120, 3, 3)>0.5) image, in the order: improved version of the Harris detector, proposed detector.

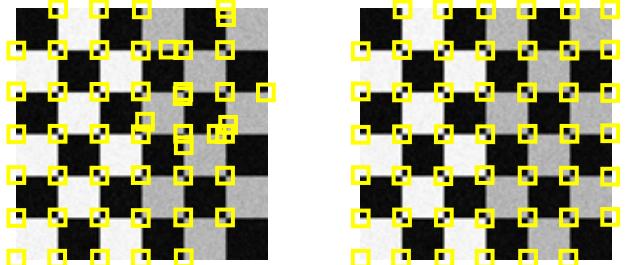


Figure 10: Outputs of corner detectors for a disturbed checkerboard (80, 3, 3) image. In the order: improved version of the Harris detector, proposed detector.

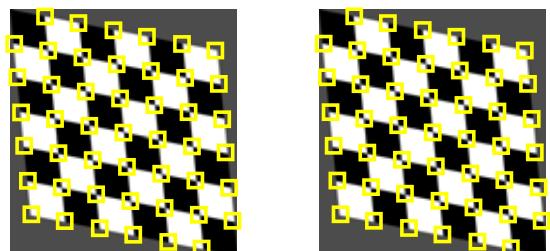


Figure 11: Outputs of corner detectors for a transformed checkerboard ((120, 3, 3)>0.5) image. In the order: improved version of the Harris detector, proposed detector.

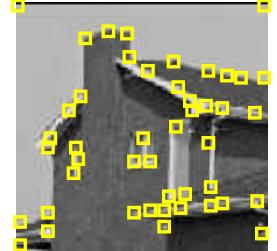
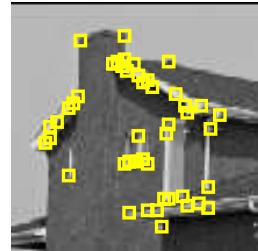
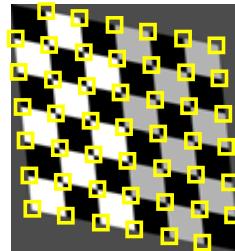
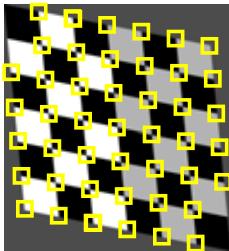


Figure 12: Outputs of corner detectors for a transformed checkerboard (80, 3, 3)image. In the order: improved version of the Harris detector, proposed detector.

In order to test the localization accuracy of detected corners, we vary the sigma of Gaussian smoothing as shown in figure (13).

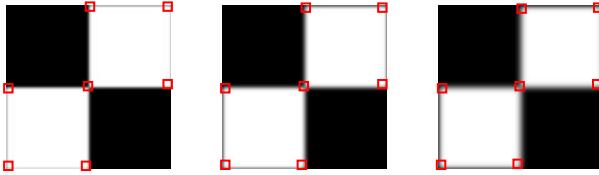


Figure 13: Outputs of proposed corner detector for a checkerboard (80, 1, 1) image with a varying sigma of Gaussian smoothing, in the order: sigma=1, sigma=2, sigma=3.

As sigma increases, obtained corner coordinates varies:

$$\{X_a, Y_a\} = \{(82, 2), (158, 2), (2, 82), (80, 80), (158, 78), (2, 158), (78, 158)\},$$

$$\{X_b, Y_b\} = \{(82, 3), (157, 3), (3, 82), (80, 80), (157, 78), (3, 157), (78, 157)\}$$

and

$$\{X_c, Y_c\} = \{(82, 4), (157, 3), (4, 82), (80, 80), (156, 78), (3, 157), (78, 156)\}.$$

Localization accuracy is satisfactory while some deviation exist.

Secondly, the objective of this experiment is to show how the proposed method provides good corner detection on real images. For this reason, we consider real house image with lots of small details and peppers image with much texture. Figures (14-15) show the outputs of the proposed algorithm and the improved version of the Harris algorithm. Number of input points for house and peppers images are 46 and 200, respectively.

From these figures, it can be seen that the proposed corner detector works well with respect to detection. Due to the texture and details many corners missed by the improved version of the Harris detector are detected by the proposed detector. Also, many multiple corners are declared in rounded corner by the improved version of the Harris.

Figure 14: Outputs of corner detectors for original house image. In the order: improved version of the Harris detector, proposed corner detector.

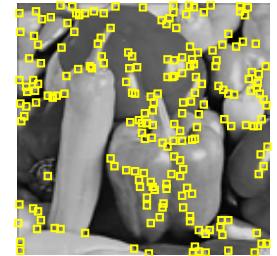


Figure 15: Outputs of corner detectors for original peppers image. In the order: improved version of the Harris detector, proposed corner detector.

5 Conclusion

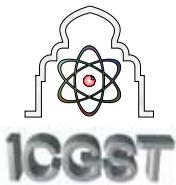
In this work a new approach for corner detection in grey level images has been presented. The method is based on the electrostatic model which has been evaluated for edge detection by Pratt's procedure. The algorithm is easy in implementation because uses the simple convolution operation by filters of 3x3 pixels size. The proposed algorithm gives good performance and obtained results are relatively satisfactory in comparison to the improved version of the Harris detector algorithm. The proposed method can be extended to corner detection in color images. This method can be also transposed to several contexts in image processing and computer vision applications such as edge detection and segmentation.

References

- [1] B. Bouda, Lh. Masmoudi and D. Aboutajdine. A new model for edge detection in digital images. ICGST, International Journal on Graphics Vision and Image Processing, Special issue on edge detection and tracking, March 2006.
- [2] B. Bouda, Lh. Masmoudi and D. Aboutajdine. Une méthode de détection des points d'intérêt basée sur l'algorithme de Harris. 5èmes Journées d'Optique et du Traitement de l'Information, OPTIQUE'06, INPT, Rabat, Morocco, pages 505-509, Avril 2006.

- [3] C. Harris and M. Stephens. A combined corner and edge detector. In Alvey Vision Conference, pages 147-151, 1988.
- [4] J. Canny. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679-698, November 1986.
- [5] F. I. Sobel. An isotropic image gradient operator. In H. Freeman (Eds.). Machine Vision for Three-Dimensional Scenes, Academic Press, pages 376-379, 1990.
- [6] C. Schmid, R. Mohr and C. Bauckhage. Evaluation of interest point detectors. In International Journal of Computer Vision, 37(2):151-172, 2000.
- [7] C. Schmid, R. Mohr and C. Bauckhage. Comparing and Evaluating Interest Points. 6th International Conference on Computer Vision, Bombay, India, January 1998.
- [8] W. K. Pratt. Digital Image Processing. 2nd Edition, John Wiley and Sons, Inc.: New York, 1991.
- [9] D. R. Jérôme and K. Hubert. Adaptation du détecteur de Harris pour l'indexation de textures. Gretsi'01 Traitement du Signal et des Images, 2001.
- [10] H. Asada and M. Brady. The curvature primal sketch. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(1): 2-14, 1986.
- [11] J. C. Cottier. Extraction et appariements robustes des points d'intérêt de deux images non étalonnées Technical Report. LIFIA-IMAG-INRIA Rhône-Alpes, 1994.
- [12] W. Förstner and E. Gülich. A fast operator for detection and precise location of distinct points, corners and centers of circular features. In Intercom-mission Conference on Fast Processing of Photogram metric Data, Interlaken, Switzerland, pages 281-305, 1987.
- [13] H. W. Förstner, A framework for low level feature extraction, In Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden, pages 383-394, 1994.
- [14] F. Heitger, L. Rosenthaler, R. Heydt, E. Peterhans and O. Kuebler. Simulation of neural contour mechanism: From simple to end-stopped cells. Vision Research, 32(5): 963-981, 1992.
- [15] R. Horaud, T. Skordas and F. Veillon, Finding geometric and relational structures in an image, In Proceedings of the 1st European Conference on Computer Vision, Antibes, France, pages 374-384, 1990.
- [16] J. M. Jolion and S. Bres. Multi-resolution contrast based detection of interest points. September 1998.
- [17] H. P. Moravec. Towards automatic visual obstacle avoidance. In Proceedings of the 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts, USA, pages 584, 1977.
- [18] E. Vincent and R. Laganière. Matching feature points in stereo pairs: A comparative study of some matching strategies. Machine Graphics and Vision, 10(3):237-259, 2001.





Extension of Gray-level Edge Detection Model to RGB Images Based on Virtual Electric Field

B. Bouda¹, Lh. Masmoudi², D. Aboutajdine¹

¹*Faculty of Sciences, Mohammad V University, GSCM-LRIT Laboratory,*

4 Avenue Ibn Battouta B.P. 1014, Rabat, Morocco

²*Faculty of Sciences, Mohammad V University, LETS Laboratory,*

4 Avenue Ibn Battouta B.P. 1014, Rabat, Morocco

brahimbouda777@gmail.com, masmoudi@fsr.ac.ma, aboutaj@fsr.ac.ma

<http://www.fsr.ac.ma>

Abstract

Several methods have been developed and used in many areas for edge detection in gray-level images. Also, a considerable attention was paid in RGB images. Nevertheless, the choice of an accurate method remains the major problem.

We present in this paper an extension of gray-level edge detection model to RGB images based on virtual electric field, which demonstrate the best detection. Where, at each component block of 3x3 pixels, an edge detection operator is first given. This is based on the use of forces provided by the neighboring virtual electric charges on the central charge present. The next stage is to integrate the contrast information contained in RGB components into one meaningful result. Indeed, two techniques are used for this: the old monochromatic-based technique and the new vector-valued technique. The results of the proposed operator are finally compared with those obtained by other operators such as Canny and Cumani, using the above techniques.

Simulations are presented in this paper and better results are achieved with the proposed method using vector-valued techniques. This method is shown to be simple and easy to implement.

Keywords: Adaptive methods; Color edge detection; Electric field; Models; RGB images.

1 Introduction

Edge detection techniques based upon differential operators are widely used to process gray-level and color images. They can be grouped in two classic methods: first-order differential and second-order differential methods. Second directional derivative and Lapla-

cian [3,7] are two most operators of the second-order differential methods. Second directional derivative is the most appealing of them in edge detection. The Laplacian method searches for zero crossings [4,10,23] in the second derivative of the image to find edges. In particular, first-order differential method is an early technique to detect edges [9,18,19,22]. Basically, this method uses some specifically designed masks to traverse the image and detects edges by locating the maxima of the magnitude of the gradient in the considered image. But the masks have fixed sizes and cannot be dilated. Consequently, the edges are well detected in certain zones and badly detected in others within the same image. To avoid the above problem, it was suitably to introduce an approach based on adaptive methods [1,14]. The method is also sensitive to noises if the image is not smoothed first. For practical applications, edge detection is only the first step in image analysis, consequently, should be short as possible. Some complicated edge detectors are not used in practice despite having rather good quality. For instance, the Canny edge detection algorithm [2] needs to adjust two thresholds and a standard deviation of Gaussian smooth mask to yield a proper result. In contrast, the first order differential methods are easy in implementation and computationally not very expensive.

Edge in gray-level images is very well known. It has a long history and has been carefully study [2,15]. The basic idea behind edge detection in this case is to find places in an image where the intensity changes rapidly. Based on this idea, an edge detector may either be based on the technique of locating the places where the first derivative of the intensity is greater in magnitude than a specified threshold. The potential edge pixel locations can then be described



by gradient operators in the results. This method works best when the gray-level transition is quite abrupt. Recently, the authors have proposed in [1] a new model based on electrostatic one for detecting edges in gray-level images, which demonstrate the best detection.

As compared to gray-level based edge detection approaches, color based edge detection has not received the same attention. One of the difficulties in edge detection in this case is the formulation of what is an edge. Indeed, in gray-level images a scalar gray-level is assigned to a pixel of image, but in color images, a color vector which consists of several components is assigned to a pixel. Another question is how to integrate the contrast information contained in various components into one meaningful output. Up to now, monochromatic-based techniques of applying a gray-level algorithm to the single components of the image and then combining the obtained results, are the most appealing edge detection methods. These traditional techniques give better results than standard methods. Koschan applied these techniques to the three color channels and performed a comparative study [12]. The author gives a result on different techniques for color edge detection and demonstrates that the performance of the Cumani operator is better [5]. However, many color variants of the Canny operator exist in the literature. Novak and Shafir [17] suggest an extension of this operator for color edge detection. Kanad introduced this color operator in [8]. For edge detection in color and multispectral images, Cumani suggests the extension of procedures based on the second-order derivatives of the images functions. This operator is one of the fundamental work in color edge detection. Opposed to the traditional techniques, vector-valued techniques which treat the color information as color vectors in vector space provided with a vector norm are new and different. Recently, Koschan presented a review of these techniques for the edge detection in color images [11]. After application of the some motioned operators using these last techniques, the author demonstrates that the performance of the Cumani operator is better.

In the present paper, we consider the RGB color model for processing digital images. In the following, we show to extend a gray-level edge detector based on the above mentioned first-order differential methods to RGB images. The novelty of this work is to use a new method for detecting edges based on virtual electric field. Input RGB image is first processed in separate processing to obtain individual components. Then, approximative gradients are obtained by convolving each block of 3x3 pixels of each component with two directional designed filters. This is based on the use of forces provided by the neighboring

virtual electric charges on central charge present. These operators are performed separately for each component and at each pixel position. The next stage is to combine the results into one meaningful output. Two techniques are then used: the first one is the monochromatic-based technique of applying a gray-level algorithm to the single components of the RGB image and then, combining the obtained results according to some logical operators. Here, the proposed logical operator is the gradient maximum. The second one is the vector-valued technique, which treat the color information as color vectors. Some operators mentioned above were implemented and the results are presented in this paper.

The rest of the paper is organized as follows. Section 2 presents virtual electric field theory and proposed edge detection operator. The edges in RGB images will be described in section 3. The experimentation results are presented in section 4. Finally, section 5 concludes the paper.

2 Virtual Electric Field and Edge Detection Operator

The presence of an electric charge produces a force electrostatic on all other charges present. The charges exert an attractive (or repulsive) force on one another by means of disturbances that they generate in the space surrounding them. These disturbances are called electric fields.

Consider an electric charge q_1 placed at point $m(x_m, y_m)$. This charge generates two electrostatic field components E_x and E_y at point $M(x_M, y_M)$ of the electrostatic field \vec{E} in XY coordinates as follows:

$$\vec{E} = \frac{K q_1}{d^2} \left[\frac{2(x_M - x_m)}{d} \vec{X} + \frac{2(y_M - y_m)}{d} \vec{Y} \right] \quad (1)$$

Where; $d = \sqrt{(x_M - x_m)^2 + (y_M - y_m)^2}$ is the Euclidian distance between points $m(x_m, y_m)$ and $M(x_M, y_M)$ and $K = \frac{1}{8\pi\epsilon_0}$ is a constant.

There is a relationship between electric field \vec{E} and electric force \vec{F} exerted on another electric charge q_0 presented at point M in that field. The simplest mathematical description is the following expression:

$$\vec{F} = \frac{K q_0 q_1}{d^2} \left[\frac{2(x_M - x_m)}{d} \vec{X} + \frac{2(y_M - y_m)}{d} \vec{Y} \right] \quad (2)$$

If we have n charges, then the total electric force \vec{F}_{tot} is given by the vector sum of each electric force:

$$\vec{F}_{tot} = \sum_{i=1}^n \vec{F}_i \quad (3)$$

Based on this mathematical description, we proceed to model a gray-level image I as a grid points



of virtual electric charges, uniformly distributed on a plane surface in electrostatic balance. For each block of 3x3 pixels we make the following analogy: the gray-levels of pixels correspond to the virtual electric charges. Hence, the central charge is subjected to the electrostatic forces of the eight neighboring charges.

In a block of 3x3 pixels, if the virtual charges q_i , $i = \{1, 2, \dots, 8\}$ and q_0 have an opposite (respectively same) sign, q_i exerts an attractive (respectively repulsive) force on q_0 . The figure 1 illustrates the virtual electric charges for one block of the image.

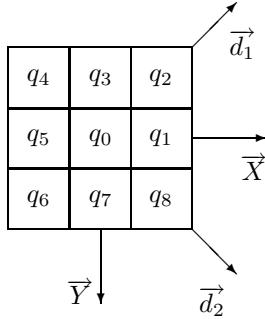


Figure 1: Virtual electric charge model for one block of the image: the charge q_0 and its eight neighboring charges.

Let r be a distance between an arbitrary virtual charge q_i and the central charge q_0 when "i" is odd (see figure 1). Each virtual charge exerts an electric force on q_0 inversely proportional to r . Horizontal and vertical forces when the virtual charges q_i have an opposite sign to q_0 , are given by:

$$\vec{F}_h = \vec{F} q_1/q_0 + \vec{F} q_5/q_0 = \frac{K q_0}{r^2} (2q_5 - 2q_1) \vec{X} \quad (4)$$

$$\vec{F}_v = \vec{F} q_3/q_0 + \vec{F} q_7/q_0 = \frac{K q_0}{r^2} (2q_3 - 2q_7) \vec{Y} \quad (5)$$

The diagonal forces are also given, respectively, by the following equations:

$$\vec{F}_{d1} = \frac{K q_0}{r^2} \left(\frac{\sqrt{2}}{2} q_6 - \frac{\sqrt{2}}{2} q_2 \right) [\vec{X} - \vec{Y}] \quad (6)$$

$$\vec{F}_{d2} = \frac{K q_0}{r^2} \left(\frac{\sqrt{2}}{2} q_4 - \frac{\sqrt{2}}{2} q_8 \right) [\vec{X} + \vec{Y}] \quad (7)$$

Let $I(x_0, y_0)$ be the central pixel which corresponds to virtual charge q_0 . The resulting force components F_X and F_Y are:

$$F_X = \frac{KI(x_0, y_0)}{r^2} [2q_5 - 2q_1 + \frac{\sqrt{2}}{2} (q_6 - q_2 + q_4 - q_8)] \quad (8)$$

$$F_Y = \frac{KI(x_0, y_0)}{r^2} [2q_3 - 2q_7 + \frac{\sqrt{2}}{2} (q_2 + q_4 - q_6 - q_8)] \quad (9)$$

The above analysis shows that the total electric forces are given by a simple convolution of the predefined masks and image. In order to keep the brightness of the image constant we take $K = 1$ and we deduce from the equation (8) and (9) the horizontal mask h_{xa} and the vertical mask h_{ya} shown in figure 2.

$\frac{\sqrt{2}}{2}$	0	$-\frac{\sqrt{2}}{2}$
2	0	-2
$\frac{\sqrt{2}}{2}$	0	$-\frac{\sqrt{2}}{2}$

$\frac{\sqrt{2}}{2}$	2	$\frac{\sqrt{2}}{2}$
0	0	0
$-\frac{\sqrt{2}}{2}$	-2	$-\frac{\sqrt{2}}{2}$

Figure 2: The obtained masks of the virtual electric charge model. On the left h_{xa} , on the right h_{ya} .

By the similarity and following the same development for repulsive forces, we find two masks $h_{xr} = -h_{xa}$ and $h_{yr} = -h_{ya}$. When we have the repulsive and attractive forces at the same time in block of the image, we proceed by the same theory to determine the suitable masks. For more detail regarding the above equations, the reader can refer to the article [1].

To generalize this method for all image components and to appear a negative sign, we determine the threshold of each block of 3x3 pixels. The gray-levels of pixels which are greater than this threshold (the average) will be assigned by a negative sign. Afterward, we exploit the above theory for the determination of filter (vertical and horizontal masks) relative to each block and adapted to edges detection.

As compared to the well known Sobel operator [22], the proposed operator is another way to take into account the pixels' distance. In this case, this is inversely proportional to the factor $\sqrt{2}$ in diagonals which stands for the vector projection. The masks are determined adaptively for each block of 3x3 pixels. On the contrary to the Sobel masks, the proposed masks vary from a block to another block of the image. Other advantage of this method is that easy in implementation and computationally not very expensive.

3 Edges in RGB Images

The techniques used for edge detection in color images can be subdivided on the basis of their principle procedures into two categories [11].

Monochromatic-based techniques: These approaches applies a gray-level algorithm to the single components of the RGB image and then, combining the individual results gained.

Vector-valued techniques: While previous approaches consider the individual color channels, vector-valued techniques treat the color information as color vectors in a vector space provided with a vector norm.



3.1 Vector-valued Techniques

Early approaches to detecting discontinuities in color or multispectral images attempted to combine the response of single-spectral edge detectors applied separately to each of the image component [16,20]. We follow the principled way to compute gradients in vector images as described by [6] and further used in [13], which is summarized as follows:

Let $\Psi(u_1, u_2) : \Re^2 \rightarrow \Re^m$ be a m-band image with components $\Psi_i(u_1, u_2) : \Re^2 \rightarrow \Re$, $i = 1, 2, \dots, m$. For RGB image, we have $m = 3$ components. For the image plane coordinates we have $u_i, i = 1, 2$. The more usual notation (x, y) is that $u_1 \equiv x$ and $u_2 \equiv y$. The value of the image at a given point (u_1^0, u_2^0) is a vector in \Re^m . The difference of image values at two points $P = (u_1^0, u_2^0)$ and $Q = (u_1^1, u_2^1)$ is given by $\Delta\Psi = \Psi(P) - \Psi(Q)$. When the distance $d(P, Q)$ between P and Q is infinite small displacement, the difference becomes the arc element.

$$d\Psi = \sum_{i=1}^2 \frac{\partial \Psi}{\partial u_i} du_i \quad (10)$$

and its squared norm is given by:

$$d\Psi^2 = \sum_{i=1}^2 \sum_{j=1}^2 \frac{\partial \Psi}{\partial u_i} \frac{\partial \Psi}{\partial u_j} du_i du_j \quad (11)$$

The equations (11) can be rewrite by using denote g_{ij} as:

$$d\Psi^2 = \begin{bmatrix} du_1 \\ du_2 \end{bmatrix}^T \begin{bmatrix} g_{11} & g_{12} \\ g_{12} & g_{22} \end{bmatrix} \begin{bmatrix} du_1 \\ du_2 \end{bmatrix} \quad (12)$$

Where; $g_{ij} = \frac{\partial \Psi}{\partial u_i} \frac{\partial \Psi}{\partial u_j}$ and the extrema of the quadratic form (12) are obtained in the directions of the eigenvectors of the matrix $[g_{ij}]$, and the values attained there are the corresponding eigenvalues. Simple algebra shows that the eigenvalues are:

$$\lambda_{\pm} = \frac{g_{11} + g_{22} \pm \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2}}{2} \quad (13)$$

and the eigenvectors are:

$$\eta_{\pm} = (\cos\theta_{\pm}, \sin\theta_{\pm}) \quad (14)$$

Where; θ_{\pm} are given (modulo π) by:

$$\begin{cases} \theta_+ = \frac{1}{2} \arctan \left(\frac{2g_{12}}{g_{11} - g_{22}} \right) \\ \theta_- = \theta_+ + \frac{\pi}{2} \end{cases}$$

Thus, the eigenvectors provide the direction of maximal and minimal changes at a given point in the image, and the eigenvalues are the corresponding rates of change. θ_+ is called the direction of maximal change and λ_+ the maximal rate of change. Similarly, θ_- and λ_- are the direction of minimal change and the minimal rate of change, respectively.

In monochromatic case ($m = 1$), letting $I(x, y) = \Psi_1(x, y)$, we have $\lambda_+ = I_x^2 + I_y^2$ and $\lambda_- = 0$. The gradient is always perpendicular to the level-sets. As a consequence, the strength of an edge in the multispectral case is not simply given by rate of maximal change, λ_+ , but by how λ_+ compares to λ_- . For example, if $\lambda_+ = \lambda_-$, we know that the image change at an equal rate in all directions. Image discontinuities can be detected by defining a function $f = f(\lambda_+, \lambda_-)$ that measures the dissimilarity between λ_+ and λ_- . A possible choice is subtraction $f = f(\lambda_+ - \lambda_-)$ as proposed in Cumani [5]. Based on the results in above theory and the ideas described in previous section, we proceed to propose RGB edge detection. Denoting the RGB gradient by ∇C , $C = \{R, G, B\}$. The gradient for RGB image is as follows:

$$\|\nabla C\| = \sqrt{\lambda_+^C - \lambda_-^C} \quad (15)$$

Where; λ_+^C and λ_-^C are given according to (13) by:

$$\lambda_{\pm}^C = \frac{g_{11}^C + g_{22}^C \pm \sqrt{(g_{11}^C - g_{22}^C)^2 + 4(g_{12}^C)^2}}{2} \quad (16)$$

The values of the matrix $[g_{ij}^C]$ are:

$$\begin{aligned} g_{11}^C &= |\frac{\partial R}{\partial x}|^2 + |\frac{\partial G}{\partial x}|^2 + |\frac{\partial B}{\partial x}|^2, \\ g_{22}^C &= |\frac{\partial R}{\partial y}|^2 + |\frac{\partial G}{\partial y}|^2 + |\frac{\partial B}{\partial y}|^2 \text{ and} \\ g_{12}^C &= \frac{\partial R}{\partial x} \frac{\partial R}{\partial y} + \frac{\partial G}{\partial x} \frac{\partial G}{\partial y} + \frac{\partial B}{\partial x} \frac{\partial B}{\partial y}. \end{aligned}$$

An example of application of the proposed edge detection method is shown in figure 3. Two kinds of RGB images are selected: the 243x243x3 synthetic image (figure 3(a)) and the 256x256x3 real image (figure 3(c)). The edge detection results of these images and their gray-level representation (figure 3(b,d)) are given. Each RGB image is transformed to gray-level image by computing the mean value over the three color components. Our goal here is to extend the method presented in previous section to RGB images using vector-valued technique. For this reason, each input color image is first processed in separate processing to obtain individual components. Then, we compute values of λ_{\pm}^C by using 3x3 masks obtained by the proposed virtual electric field method.

From the obtained results, we see that all visually, the method gives the better outputs. Some edges could be detected in the RGB images (figure 3(e,g)) that were not determined in the gray-level images (figure 3(f,h)). In summary, with the proposed method best quality of edges is given and the obtained results from color images were better than those obtained from intensity images.

The algorithm used to find edges is as follows:

- Compute, for each block of 3x3 pixels of the image, both masks by basing on the virtual electric



field method and proceed to convolution operation in order to determine images derivatives;

- Compute the magnitude of the gradient using equation (15).

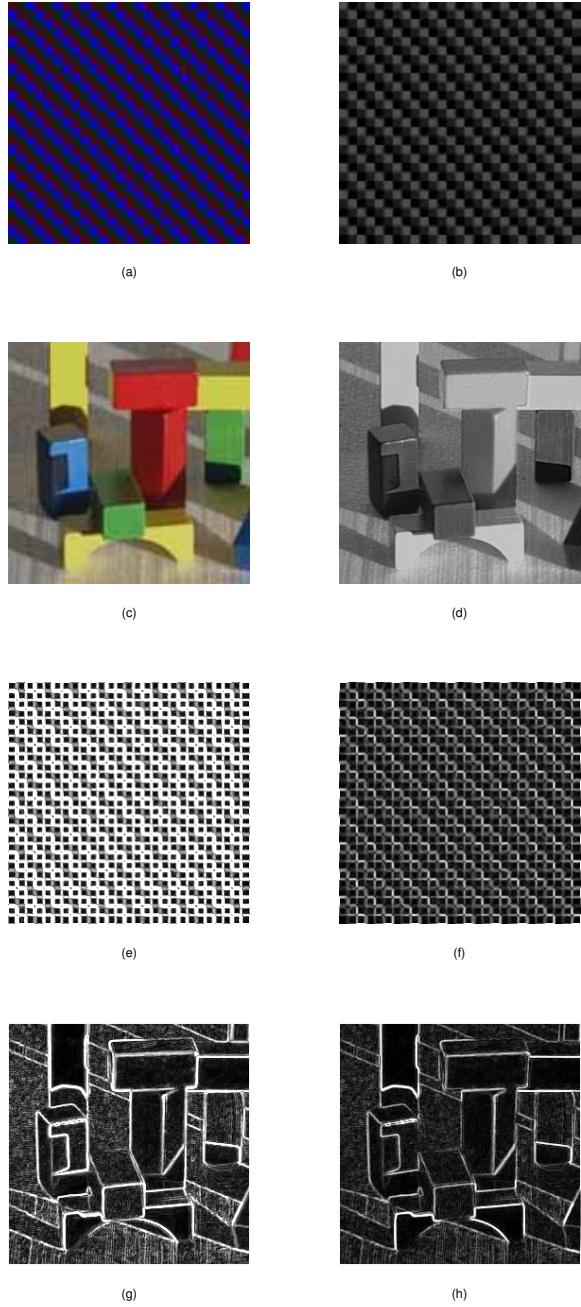


Figure 3: Results of edge detection applied to the RGB Squares and Block images. (a,c) Original RGB images; (b,d) Gray-level representations; (e,g) Proposed edge results for RGB images; (f,h) Proposed edge results for gray-level images.

3.2 Monochromatic-based Techniques

Most of the color edge detection methods are monochromatic-based techniques. Better results are

achieved in general when these methods are applied comparatively with the old standard techniques. These last methods start by transforming the color images into one of the luminance-chrominance models before applying edge detection operation. In this subsection, we focus mainly on monochromatic-based techniques in order to compare their results to these obtained by the proposed method using vector-valued techniques. Hens, we choose the gradient maximum as logical operator. The detectable edges for each RGB component are fused of the three components according to the gradients maximum g_M :

$$g_M = \text{Max}(|g_R|, |g_G|, |g_B|) \quad (17)$$

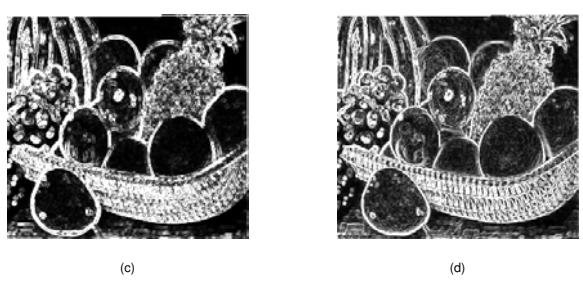
Where; g_R , g_G and g_B are the vector gradients of red, green and blue component, respectively.

An example of edge detection result using the monochromatic-based method is shown in figure 5. Two images are selected: 281x302x3 Fruit and 256x256x3 Speelgoed as shown in figure 4. The obtained results can be interpreted as follows. Many pixels in the image background are determined as edge points by applying this method. Also, many faults and gaps in the contours are shown in the detected edges (figure 5(a,c)). As compared to the obtained edges using the vector-valued techniques developed in previous subsection, we show clearly that the results are improved. Hens, many gaps shown in the detected edges by the traditional method are removed with the virtual electric field model using vector-valued techniques (see figure 5(b,d)).

In summary, best edges are detected with the proposed method based on the virtual electric field using vector valued techniques.



Figure 4: Original color images.



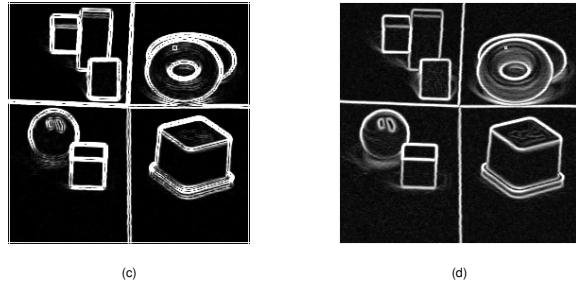


Figure 5: Results of edge detection applied to the color images Fruit and Speelgoed. (a,c) Edge results of the traditional algorithm based on virtual electric field; (b,d) Edge results of the proposed algorithm based on virtual electric field.

4 Simulations and results

In this section, we describe results of our experimentation for 512x512x3 image Lena shown in figure 6 (a). We applied the Canny and the Cumani operator to this selected image. We compared the results with those obtained when applying the proposed method. These operators are chosen in this work since they are given the best results as described in [12]. For instance, a global discussion of several criteria for the evaluation of edge detectors is given in [21] for gray-level images and [11, 12] for gray-level and color images.

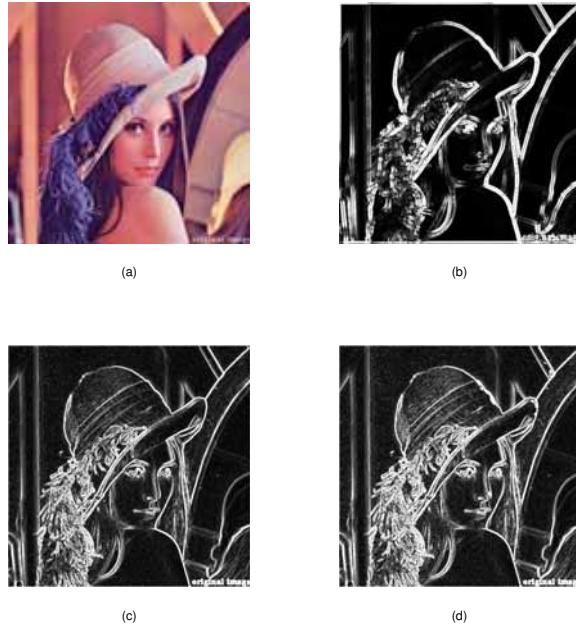


Figure 6: Results of edge detection applied to the color image Lena using vector-valued techniques: (a) Original RGB image; (b) Edge detection results for the Canny operator; (c) Edge detection results for the Cumani operator; (d) Results of the proposed algorithm based on virtual electric field.

The results can be interpreted as follows. From the obtained edges with the Canny operator (figure 6(b)),

it can be remarked the nondetection of some contours that are successfully detected with the Cumani operator indicated in (figure 6(c)). Better results are also achieved with the operator based on virtual electric field (figure 6(d)). More edges were detected with this last method than with Cumani operator. Here, the quality of the results can be improved if the best threshold parameter is chosen. Instead of the average value we can choose the median or the maximum value for each block of 3x3 pixels. In addition, significant difference occurred in computing time between the Canny operator and the proposed operator based upon first order-differential methods.

5 Conclusion

In this paper a new edge detection method for RGB images based on virtual electric field has been proposed. The comparison of this method with some other methods has been presented using traditional techniques and vector-valued techniques.

All implemented methods have been applied to several synthetic and real images. Due to lack of space, this paper only presents the edge detection results of some selected images. The outputs of the Canny and Cumani operator was compared with those obtained by the proposed operator. In this comparison, better results were achieved when the proposed method based on the virtual electric field, using vector-valued techniques was applied.

References

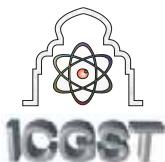
- [1] B. Bouda, Lh. Masmoudi, D. Aboutajdine. "A new model for edge detection in digital images", ICGST, Graphics, Vision and Image Processing, special issue on edge detection and tracking, March 2006.
- [2] J. Canny. "A computational approach to edge detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679-698, November 1986.
- [3] H. Chidiac, D. Ziou. "Classification of image edge", Vision Interface'99, Troise-Rivieres, Canada, 17-24, 1999.
- [4] J. J. Clarkk. "Authenticating edges produced by zero crossing algorithms", IEEE Trans. Patt. Anal. Mach. Intell., 11(1):43-57, 1989.
- [5] A. Cumani. "Edge detection in multispectral images", Comput. Vis. Graph. Image Process.: Graphical Models Image Processing, 53(1):40-51, 1991.
- [6] S. Di Zenzo. "A note on gradient of multi-image", Computer Vison, Graphics and Image Processing, 33:116-125, 1986.



- [7] R. M. Haralick, S.R. Sternberg, X. Zhuang. "Image analysis using mathematical morphology", IEEE Trans. Pattern Anal. Machine Intell., 9(4):532-550, 1987.
- [8] T. Kanade. "Image understanding research at CMU", in Proc. Image Understanding Workshop, 2:32-40, 1987.
- [9] R. Kirsh. "Computer determination of the constituent structure of biomedical images", Computers and Biomedical Research, 4(3):315-328, 1971.
- [10] A. F. Korn. "Toward a symbolic representation of intensity changes in images", IEEE Trans. Patt. Anal. Mach. Intell., 10(5):610-625, 1988.
- [11] A. Koschan, M. Abidi. "Detection and classification of edges in color images", IEEE Signal Processing Magazine, 22(1):64-73, Jan. 2005.
- [12] A. Koschan. "A comparative study on color edge detection", proc. of 2nd Asian Conference on Computer Vision, Singapore, 3:574-578, December 1995.
- [13] H. C. Lee, D. R. Cok. "Detecting boundaries in a vector field", IEEE Transaction on signal processing, 39(5):1181-1194, 1991.
- [14] S. G. Mallat and S. Zhong. "Characterization of signals from multiscale edges", IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(7):710-732, July 1992.
- [15] D. Marr, E. Hildreth. "Theory of edge detection", proc. Royal Soc. London, B207:187-217, 1980.
- [16] R. Nevatia. "A color edge detector and its use in scene segmentation", IEEE Trans. Syst. Man Cybern., 7(11):820-826, November 1977.
- [17] C. L. Novak, S.A. Shafer. "Color edge detection", proceedings DARPA Image Understanding Workshop, Los Angeles, CA, USA, 1:35-37, February 1987.
- [18] J. G. Prewitt. "Object enhancement and extraction", Picture Processing and Psychopictorics, B. Lipkin and A. Rosenfeld, Eds. New York: Academic, 75-149, 1970.
- [19] L. Roberts. "Machine perception of three-dimensional solids", Optical and Electro-optical Information Processing, Cambridge, Massachusetts, MIT Press, 159-197, 1965.
- [20] G. Robinson. "Color edge detection", Optical Engineering, 16(5):479-484, September 1977.
- [21] M. Salotti, F. Bellet, C. Garbay. "Evaluation of edge detectors: Critics and proposal", in Proc. Workshop Performance Characteristics Vision Algorithms, H.I. Christensen, W. Förstner, C.B. Madson, eds., 1996.
- [22] I. Sobel. "An isotropic image gradient operator", in H. Freeman (Eds.), Machine Vision for Three-Dimensional Scenes, Academic Press: 376-379, 1990.
- [23] V. Torre, T. Poggio. "On edge detection", IEEE Trans. Patt. Anal. Mach. Intell., 8(2):147-163, 1986.







Performance Evaluation of Prewitt Edge Detector for Noisy Images

¹Raman Maini, ²J.S.Sohal
*University College of Engineering,
Punjabi University, Patiala-147002(India)
research_raman@yahoo.com*

²Ludhiana College of Engineering & Technology,
Katani Kalan, Ludhiana-141113(India)
dr_jssohal@yahoo.com

Abstract

Since edge detection is in the forefront of image processing for object detection, it is crucial to have a good understanding of edge detection algorithms. This paper evaluates the performance of Prewitt Edge Detector for detection of edges in digital images corrupted with different kinds of noise. Different kinds of noise are studied in order to evaluate the performance of the Prewitt Edge Detector. Further, the various standard test Images are examined to validate our results. The software is developed using MATLAB 7.0.¹ It has been observed that the Prewitt Edge Detector works effectively for the digital images corrupted with Poisson Noise where as its performances reduces sharply for other kinds of noise in digital images. The results of this study are quite promising.

Keywords: Edge Detector, Digital Image Processing, Noise

1. Introduction

Digital image processing is the use of computer algorithms to perform image processing on digital images. Digital image processing has the same advantages (over analog image processing) as digital signal processing has (over analog signal processing) -- it allows a much wider range of algorithms to be applied to the input data, and can avoid problems such as the build-up of noise and signal distortion during processing. Digital image processing allows the use of much more complex algorithms for image processing, and hence can offer both more

sophisticated performance at simple tasks, and the implementation of methods which would be impossible by analog means[1]-[3].

An image may be defined as a two dimensional function , $f(x,y)$, where x and y are spatial(Plane) coordinates, and the amplitude of f at any pair of co-ordinates (x,y) is called the intensity or grey level of the image at that point[3].

When x,y , and the amplitude values of f are all finite, discrete quantities, we can call the image as digital image

Edge detection is a fundamental problem of computer vision and has been widely investigated. The goal of edge detection is to mark the points in a digital image at which the luminous intensity changes sharply. For Computer vision and Image processing Systems to Interpret an Image, they first must be able to detect the edges of each object in the image [4]. Edge representation of an image drastically reduces the amount of data to be processed, yet it retains important information about the shapes of objects in the scene. This description of an Image is easy to integrate into a large no of object recognition algorithms used in computer vision and other image processing applications. Edge detection produces an edge map that contains important information about the image. The Memory space required for storage is relatively small, and the original image can be restored easily from its edge map. This method has proved both effective and powerful and is widely used in applications ranging from satellite imaging to medical radiology [5]-[6].

2. Noise

Noise is considered to be any measurement that is not part of the phenomena of interest. Images are prone to different types of noises. Departure of ideal signal is generally referred to as noise. Noise arises as a result of unmodelled or unmodellable

¹ Raman Maini is corresponding author with e-mail address research_raman@yahoo.com , Mobile no +91-9872220442 and address as Assistant Professor, University College of Engineering, Punjabi University, Patiala-147002 (India), Fax No. +91-1753046324



processes going on in the production and capture of real signal. It is not part of the ideal signal and may be caused by a wide range of sources, e. g., variation in the detector sensitivity, environmental variations, the discrete nature of radiation, transmission or quantization errors, etc. It is also possible to treat irrelevant scene details as if they are image noises, e.g., surface reflectance textures. The characteristics of noise depend on its source, as does the operator which reduces its effects. Many image processing packages contains operators to artificially add noise to an image. Deliberately corrupting an image with noise allows us to test the resistance of an image processing operator to noise and assess the performance of various noise filters. Noise is generally grouped into two categories-independent noise and image data dependent noise.

2.1 Image Data Independent Noise

This type of noise can be described by an additive noise model, where the recorded image, $i(m, n)$ is the sum of the true image $t(m, n)$ and the noise $n(m, n)$:

$$i(m, n) = t(m, n) + n(m, n) \quad (1)$$

The noise $n(m, n)$ is often zero-mean and described by its variance, σ_n^2 . In fact, the impact of the noise on the image is often described by the signal to noise ratio (SNR), which may be given by

$$SNR = \frac{\sigma_t}{\sigma_n} = \sqrt{\frac{\sigma_i^2}{\sigma_n^2} - 1} \quad (2)$$

σ_i^2 and σ_n^2 are the variances of the true image and the recorded image, respectively. In many cases, additive noise is evenly distributed over the frequency domain (white noise), whereas an image contains mostly low frequency information. Therefore, such a noise is dominant for high frequencies and is generally referred as Gaussian noise.

2.2 Image Data Dependent Noise

2.2.1 Detector Noise

Another kind of Gaussian noise, which occurs in all recorded images to a certain extent, is detector noise. This kind of noise is due to the discrete nature of radiation, i.e., the fact that each imaging system is recording an image by counting photons. Allowing some assumptions (which are valid for many applications) this noise can be modeled with an independent, additive model, where the noise has a zero-mean Gaussian distribution described by its standard deviation (σ), or variance[8]. This means that each pixel in the noisy

image is the sum of the true pixel value and a random, Gaussian distributed noise value.

2.2.2 Speckle Noise

Another common form of noise is data drop-out noise commonly referred to as Speckle noise. This noise is, in fact, caused by errors in data transmission [9]. The corrupted pixels are either set to the maximum value, which is something like a snow in image or have single bits flipped over.

2.2.3 Salt & Pepper Noise

This type of noise is also caused by errors in data transmission and is a special case of data drop-out noise when in some cases single, single pixels are set alternatively to zero or to the maximum value, giving the image a salt and pepper like appearance [10]. Unaffected pixels always remain unchanged. The noise is usually quantified by the percentage of pixels which are corrupted.

2.2.4 Poisson Noise

This type of noise is caused by the nonlinear response of the image detectors and recorders. Here the image data dependent (signal dependent) term arises because detection and recording processes involve random electron emission having a Poisson distribution with a mean response value [11]. Since the mean and variance of a Poisson distribution are equal, the signal dependent term has a standard deviation if it is assumed that the noise has a unity variance.

3. Prewitt Edge Detector

A variety of Edge Detectors are available for detecting the edges in digital images. However, each detector has its own advantages and disadvantages. The basic idea behind edge detection is to find places in an image where the intensity changes rapidly. Based on this idea, an edge detector may either be based on the technique of locating the places where the first derivative of the intensity is greater in magnitude than a specified threshold or it may be based on the criterion to find places where the second derivative of the intensity has a zero crossing[12]-[14]. The basic criterion for using Prewitt edge detector for detection of edges in digital images is that image should contain sharp intensity transition and low noise of Poisson type is present. When using Prewitt edge detection the image is convolved with a set of (in general 8) convolution kernels, each of which is sensitive to edges in a different orientation. For each pixel the local edge gradient *magnitude* is estimated with the maximum response of all 8 kernels at this pixel location:

$$|G| = \max (|G_i|: i=1 \text{ to } n) \quad (3)$$



Where \mathbf{G}_i is the response of the kernel i at the particular pixel position and n is the number of convolution kernels. The local edge orientation is estimated with the orientation of the kernel that yields the maximum response. Various kernels can be used for this operation. [15]. Two templates out of the set of 8 are shown in Figure 1:

-1	+1	+1
-1	-2	+1
-1	+1	+1

0

+1	+1	+1
-1	-2	+1
-1	-1	+1

45

Fig. 1 Prewitt edge detecting templates sensitive to edges at 0° and 45° .

The whole set of 8 kernels is produced by taking one of the kernels and rotating its coefficients circularly. Each of the resulting kernels is sensitive to an edge orientation ranging from 0° to 315° in steps of 45° , where 0° corresponds to a vertical edge. The maximum response $|G|$ for each pixel is the value of the corresponding pixel in the output magnitude image. The values for the output orientation image lie between 1 and 8, depending on which of the 8 kernels produced the maximum response. After having calculated the magnitude of the 1st derivative using convolving the image with the mask, we now have to identify those pixels corresponding to an edge. The easiest way is to threshold the gradient image, assuming that all pixels having a local gradient above the threshold must represent an edge. The lower the threshold, the more lines will be detected, and the results become increasingly susceptible to noise, and also to picking out irrelevant features from the image. Conversely a high threshold may miss subtle lines, or segmented lines. So proper threshold value have to be selected so that we get only real edges and false edges are rejected. The selection of a threshold value is an important design decision that depends on a number of factors, such as image brightness, contrast, level of noise, and even edge direction. Typically, the threshold is selected following an analysis of the gradient image histogram. So, Selection of threshold is an important parameter to get better performance for considered noisy images. The output of the thresholding stage is extremely sensitive and there are no automatic procedures for satisfactorily determining thresholds that work for all images. The strengths of using Prewitt edge detector are that it is simple to implement, less computational cost as compared to other edge detectors like LoG(Laplacian of Gaussian), zero-crossing and canny etc. 3x3 filter mask being a mask of larger size provides good smoothing operation and reduces noise to a good

level. So, Prewitt edge detector is an appropriate way to estimate the magnitude and orientation of an edge.

This edge detection method is also called *edge template matching*, because a set of edge templates is matched to the image, each representing an edge in a certain orientation. The edge magnitude and orientation of a pixel is then determined by the template that matches the local area of the pixel the best [16]-[17].

4. Results and Discussion

First, in order to evaluate the performance of the Prewitt Edged Detector, a standard test image of a coin was taken and its edge was detected using Prewitt Edge Detector. The Edge image is used as a reference for the purpose of comparison in subsequent studies. Then the test image was got corrupted with four different kinds of noise, generated using in MATLAB [16] environment with default values. Then for each of the four noisy images, the performance of the Prewitt Edge Detector was examined practically. Predefined default threshold values used by Prewitt Edge Detector for different test images corrupted with different kind of noise are shown in ²Table-1. The results of edge detection are shown in Fig. 2. It has been observed that the Prewitt Edge Detector works well both with the Gaussian as well as Poisson noise corrupted images. Further, it has been observed that out of these two results, the performance of the said detector is much superior in Poisson noise corrupted image as compared to Gaussian noise corrupted image. However, its performance decrease drastically for Salt & Pepper as well as Speckle noise corrupted images. Prewitt edge detector by inherent does the averaging of neighboring pixels. Since the Salt & Pepper noise and speckle pixel values are often very different from the surrounding values, they tend to distort the pixel average calculated by the averaging of neighboring pixels significantly. Therefore the average value calculated will be significantly different from the true value. So, performance of Prewitt edge detector decreases sharply for salt & pepper and speckle type of noise. For Poisson noise, distribution for the values of an each pixel is determined by the nature of light itself. Light isn't a continuous quantity, but occurs in discrete photons. These photons don't arrive in a steady stream, but sometime vary over time. Think of it like a flow of cars on a road-sometimes they bunch together, sometimes they spread out, but in general there is an overall average flow. Therefore, corrupted pixel come

² Tables 1 & 2 and Fig. 2 & 3 are included after the references



together and can be better smoothed by averaging. So Prewitt edge detector which by its inherent property does the averaging of neighboring pixels values reduces this kind of noise accurately. In Gaussian Noise, each pixel in the noisy image is the sum of the true pixel value and a random, Gaussian distributed noise value. So image corrupted by this type of noise is smoothed as a whole by Prewitt edge detector but loses sharp image characteristics to large extent as noise is present in each pixel value but not so well as Poisson noise corrupted image because Poisson noise distribution is for discrete values, not continuous ones which suits the Prewitt edge characteristic of averaging of neighborhood pixels well. Therefore, Prewitt edge detector performance is better for images corrupted with Poisson type of noise as compared to Gaussian noise.

It has also been observed that Prewitt edge detector produces thicker edges and corners are often missed because 1D gradient at corners is usually small.

In order to validate our results about the performance of Prewitt Edge Detector, six different standard test images, each corrupted with Poisson noise are considered. The performance of Prewitt Edge Detector is again examined both for the original as well as noise corrupted images. Predefined default threshold values used by Prewitt Edge Detector for different test images are given in Table-2. The results are shown in Fig.3. From the results, it has again been observed that the performance of the Prewitt Edge Detector is found to be satisfactory for all the six test images corrupted with Poisson noise.

5. Conclusion

In this paper, an attempt is made to evaluate the performance of the Prewitt Edged Detector for noisy images. Experimental results have demonstrated that the Prewitt Edge Detector works quite well for digital images corrupted with Poisson Noise whereas its performance decreases sharply for other kinds of noise. Hence, this type of detector can not be used in practical images which are generally corrupted with Gaussian noise, salt & pepper noise and speckle type of noise. However, these can be used successfully in conjunction with suitable digital filter to reduce the effect of noise substantially before applying the Prewitt Edged Detector. So, firstly noise is to be reduced by convolving the image with a suitable two dimensional digital filter. With the use of Digital Filter Isolated noise points and small structures are filtered out. Therefore, some filter for noise reduction has to be applied before applying Prewitt Edge Detector (PED) otherwise edge detection will not be accurate. The work is under further progress to develop two dimensional digital FIR filter based on two dimensional weighted least squares method to reduce the effect of different kind

of noise in digital images and study its performance in conjunction with Prewitt Edge Detector to find edges in noisy images.

Acknowledgement

The authors are greatly indebted to the Department of Computer Engineering, Punjabi University, Patiala and Ludhiana College of Engineering and Technology, Katani Kalan, Ludhiana for providing excellent lab facilities that make this work possible

References

- [1] B. Chanda and D.D. Majumdar: '*Digital Image Processing and Analysis*', PHI, New Delhi, December, 2002
- [2] S. Ando: 'Consistent Gradient Operators', *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp.252-265, Mar. 2000.
- [3] R.C. Gonzalez and R.E. Woods: '*Digital Image Processing*', Addison-Wesley Publishing company, 1992.
- [4] J. Koplowitz and V. Greco: 'On the Edge Location Error for Local Maximum and Zero-Crossing Edge Detectors', *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.16, no. 12, pp. 1207-1212, Dec. 1994.
- [5] G. Robinson: 'Edge Detection by Compass Gradient Masks', *Computer Graphic Image Processing*, vol. 6, pp. 492-501, 1997.
- [6] Y. Yitzhaky and E Peli.: 'A Method for objective Edge Detection Evaluation and Detector Parameter Selection', *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 1027-1033, August 2003.
- [7] M. Sharifi, M. Fathy and, M.T. Mahmoudi: 'A Classified and Comparative Study of Edge Detection Algorithms', in proc. *IEEE Computer Society International Conference on Information Technology: Coding and Computing*, 2002.
- [8] S.G. Chang, Y. Bin and M. Vetterli: 'Adaptive wavelet thresholding for image denoising and compression', *IEEE Trans. On Image Processing*, vol. 9, no.9, pp. 1532-1546, Sep 2006
- [9] L. Gagnon and F.D. Smaili: 'Speckle Noise Reduction of Airborne SAR Images with Symmetric Daubechies Wavelets', *SPIE Proc. #2759*, pp. 1424, 1996
- [10] Z. Wang and D. Hang, " Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images," *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal processing*, vol. 46, no. 1, pp. 78-80 Jan. 1999
- [11] K. Timmermann and R. Novak: 'Multiscale modeling and estimation of Poisson processes with applications to photon-limited imaging',



- IEEE Trans. Information Theory.*, vol. 45 , no. 3, pp. 846-852., 1999
- [12] H. Chidiac and D. Ziou, 'Classification of Image Edge', Vision Interface'99, Trois-Rivières, Canada, 1999, pp. 17-24.
- [13] J. Mathews: 'An Introduction to Edge Detection: The Sobel Edge Detector', Available at www.generations.org/content/2001/1m01.asp 2002
- [14] J.L. Haralick, R. Shapiro and L. Boeing: 'Morphological Edge Detector', *IEEE Journal of Robotics and Automation*, vol. 3, no. 2, pp. 142-156, Apr. 1987.
- [15] K. Cho, P. Meer and J. Cabera: 'Quantitative Evaluation of Performance through bootstrapping: Edge Detection'. in proc. of International Symposium on Computer Vision, 1995, pp. 491-496.
- [16] R.C. Gonzalez, R.E. Woods and S.L. Eddins: 'Digital Image Processing using MATLAB', Pearson Education Inc., 2004.
- [17] B. Chanda,, B.B. Chaudhari and D.D. Majumdar: 'A Differentiation Enhancement Edge Detector and its Properties', *IEEE Trans. on System, Man and Cybern.*, vol. 15, no 6, pp.162-168, 1985.



Raman Maini received B.Tech(Computer Science & Engineering) from Beant College of Engineering, Gurdaspur, Punjab, India in 1999 and M.Tech(Computer Science & Engineering) from PAU, Ludhiana, India , in 2002. He got Merit certificate

in his M.Tech thesis at PAU
He is currently working as an Assistant Professor in Computer Engineering at University College of Engineering, Punjabi University, Patiala, India. He is a life member of ISTE (Indian Society of Technical Education), India, IETE (Institution of Electronics & Telecommunication Engineers), India. His current area of research is Computer Vision (Specialty Noise Reduction in Medical Images, Edge Detection and Image Enhancement)

He is currently working as a Principal, Ludhiana College of Engineering & Technology, Katani Kalan, Ludhiana, India. Formerly he was Dean FET, Professor and Head DET &DSW, G.N.D.U, Amritsar, Punjab, India, Professor and Head department of Computer Science & Electrical Engineering, PAU, Ludhiana, Punjab, India. He has to his name 80 Research papers in referred national and international journals. He is fellow of FIE, FIETE, LMISTE, LMISAE CHARTERED ENGR. His area of research includes Image Processing, neural networks, and mobile systems.



J.S.Sohal received M.Tech(Electronics & Communication Engineering) from Indian Institute of Technology, Roorkee Uttar Pradesh, India in 1973. He received his Ph.D(Electronics & Communication Engineering) in 1976 from Indian Institute of Technology, Roorkee, Uttar Pradesh, India



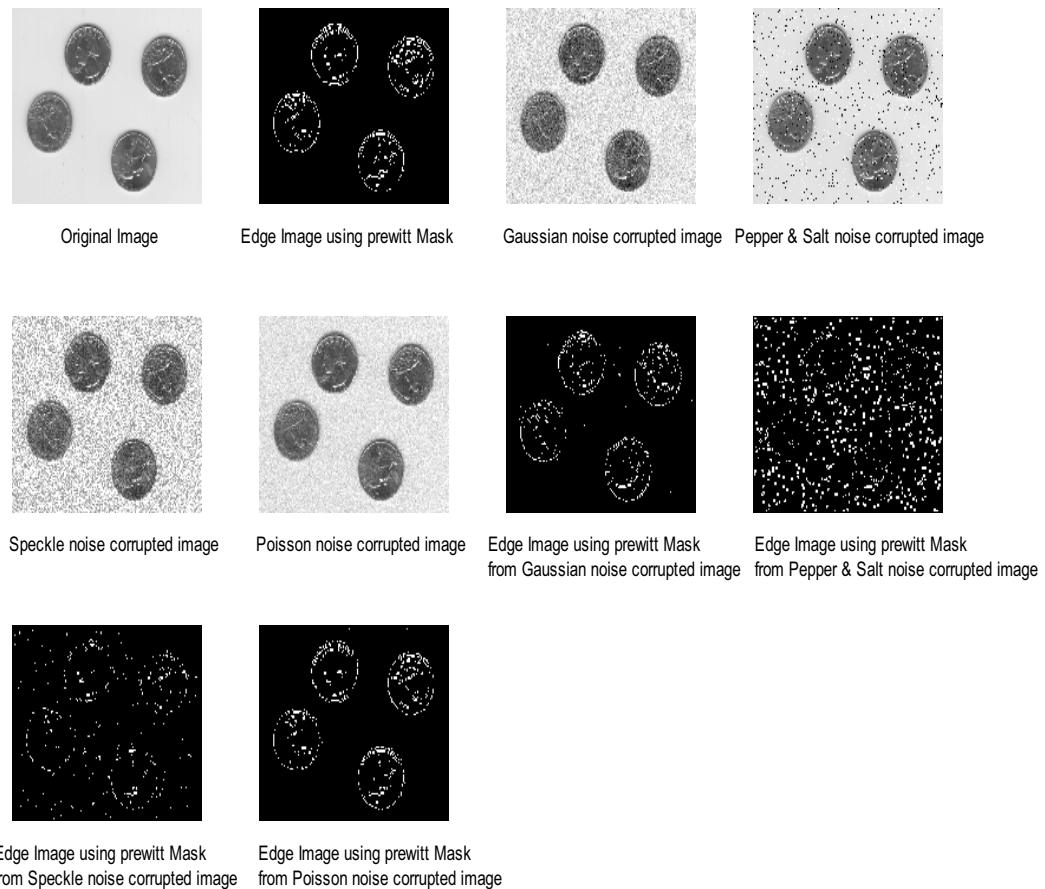


Fig. 2: Performance of Prewitt Edge Detector for an image corrupted with different types of noise. Firstly edges are detected using Prewitt Edge Detector from noise free coin image. Then coin image is corrupted with four different type of Noise. Then again the Prewitt Edge Detector is applied to corrupted coin images to study the performance of Prewitt Edge detector with different Kind of Noise

Table-1
Predefined default threshold values used by Prewitt Edge Detector for different test images corrupted with different kind of noise

S. No.	Image	Threshold value
1	Original	0.0805
2	Gaussian noise corrupted image	0.1380
3	Pepper & Salt noise corrupted image	0.1854
4	Speckle noise corrupted image	0.1880
5	Poisson noise corrupted image	0.1036

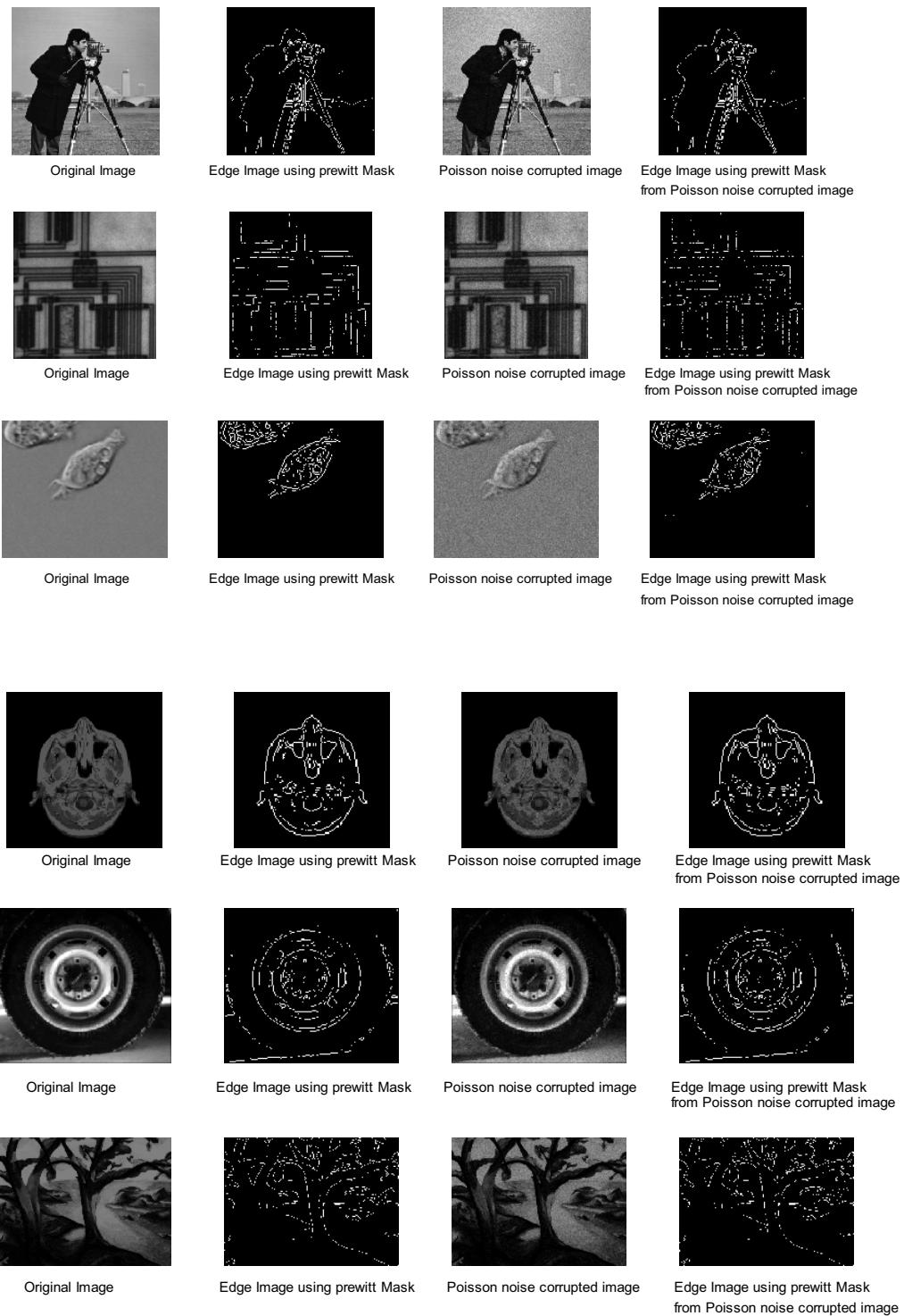


Fig. 3 Performance of Prewitt Edge Detector for different images corrupted with Poisson Noise



S. No.	Image	Original image	Poisson noise corrupted image
1	Coin	0.0805	0.1036
2	Cameraman	0.1441	0.1533
3	Circuit	0.0825	0.0926
4	Cell	0.0539	0.0750
5	MRI	0.0842	0.0876
6	Tire	0.1243	0.1282
7	Tree	0.0909	0.0973

Table -2:- Predefined default threshold values used by Prewitt Edge Detector for Different Images





Comparative Study of Edge Detection Algorithms Applying on the Grayscale Noisy Image Using Morphological Filter

Mohamed Roushdy

Computer Science Department, Faculty of Computer and Information Sciences

Ain Shams University, Abbassia, Cairo, Egypt

miroushdy@hotmail.com

<http://cis.shams.edu.eg>

Abstract

In this paper, classified and comparative study of edge detection algorithms are presented. Experimental results prove that Boie-Cox, Shen-Castan and Canny operators are better than Laplacian of Gaussian (LOG), while LOG is better than Prewitt and Sobel in case of noisy image. Subjective and objective methods are used to evaluate the different edge operators. The morphological filter is more important as an initial process in the edge detection for noisy image and used opening-closing operation as preprocessing to filter noise. Also, smooth the image by first closing and then dilation to enhance the image before the edge operators affect.

Keywords: Edge detection, image processing, morphological filter

1- Introduction

Edge detection is a critical element in image processing, since edges contain a major function of image information. The function of edge detection is to identify the boundaries of homogeneous regions in an image based on properties such as intensity and texture. Many edge detection algorithms have been developed based on computation of the intensity gradient vector, which, in general, is sensitive to noise in the image. In order to suppress the noise, some spatial averaging may be combined with differentiation such as the Laplacian of Gaussian operator and the detection of zero crossing. Canny [1] derived analytically optimal step edge operators and showed that the first derivative of Gaussian filter is a good approximation of such operators. An alternative to gradient techniques is based on statistical approaches. The idea is to examine the distribution of intensity values in the neighborhood of a given pixel and determine if the pixel is to be classified as an edge. In comparison with the differential approaches, less attention has been paid to statistical approaches. However, this method has been approached by some authors, e.g., Bovik et al.

[2] and Yakimovsky [3]. The rest of this paper is organized as follows. Section (2) reviews edge detection operators. Section (3) classifies the edge detection algorithms and gives the advantages and disadvantages of these different operators. Section (4) discusses morphological filtering which is very important process for noisy image. Section (5) presents methods of the performance evaluation for edge detectors. Section, (6) the experimental result of many edge detections obtained and finally in section (7) the discussion and conclusion are given.

2. Review of previous work

In the past two decades several algorithms were developed to extract the contour of homogeneous regions within digital image. A lot of the attention is focused to edge detection, being a crucial part in most of the algorithms. Classically, the first stage of edge detection (e.g. the gradient operator, Robert operator, the Sobel operator, the Prewitt operator) is the evaluation of derivatives of the image intensity. Smoothing filter and surface fitting are used as regularization techniques to make differentiation more immune to noise. Raman Maini and J. S. Sobel [4] evaluated the performance of the Prewitt edge detector for noisy image and demonstrated that the Prewitt edge detector works quite well for digital image corrupted with Poisson noise whereas its performance decreases sharply for other kind of noise. Davis, L. S. [5] has suggested Gaussian pre-convolution for this purpose. However, all the Gaussian and Gaussian-like smoothing filters, while smoothing out the noise, also remove genuine high-frequency edge features, degrade localization and degrade the detection of low- contrast edges. The classical operators emphasize the high frequency components in the image and therefore act poorly in cases of moderate low SNR and/or low spatial resolution of the imaging device. The awareness of this has lead to new approaches in which balanced trade-offs are sought between noise suppression, image deblurring and the ability to resolve interfering



edges, altogether resulting in operators acting like band-pass filters e.g. Canny. Sharifi, M. et al. [6] introduces a new classification of most important and commonly used edge detection algorithms, namely ISEF, Canny, Marr-Hildreth, Sobel, Kirch and Laplacian. They discussed the advantages and disadvantages of these algorithms. Shin, M.C et al. [7] presented an evaluation of edge detector performance using a structure from motion task. They found that the Canny detector had the best test performance and the best robustness in convergence and is one of the faster executing detectors. It performs the best for the task of structure from motion. This conclusion is similar to that reached by Heath et al. [8] in the context of human visual edge rating experiment. Rital, S. et al. [9] proposed a new algorithm of edge detection based on properties of hyper graph theory and showed this algorithm is accurate, robust on both synthetic and real image corrupted by noise. Li Dong Zhang and Du Yan Bi [10] presented an edge detection algorithm that the gradient image is segmented in two orthogonal orientations and local maxima are derived from the section curves. They showed that this algorithm can improve the edge resolution and insensitivity to noise. Zhao Yu-qian et al. [11] proposed a novel mathematic morphological algorithm to detect lungs CT medical image edge. They showed that this algorithm is more efficient for medical image denoising and edge detecting than the usually used template-based edge detection algorithms such as Laplacian of Gaussian operator and Sobel edge detector, and general morphological edge detection algorithm such as morphological gradient operation and dilation residue edge detector. Fesharaki, M.N. and Hellestrand, G.R [12] presented a new edge detection algorithm based on a statistical approach using the student t-test. They selected a 5x5 window and partitioned into eight different orientations in order to detect edges. One of the partitioning matched with the direction of the edge in the image shows the highest values for the defined statistic in that algorithm. They show that this method suppresses noise significantly with preserving edges without a prior knowledge about the power of noise in the image.

3- Edge detection operators

An edge operator is a neighborhood operation which determines the extent to which each pixel's neighborhood can be partitioned by a simple arc passing through the pixel where pixels in the neighborhood on one side of the arc have one predominant value and pixels in the neighborhood on the other side of the arc have a different predominant value. Usually gradient operators, Laplacian operators, zero-crossing operators are used for edge detection. The gradient operators compute some quantity related to the magnitude of the slope of the underlying image gray tone intensity surface of which the observed image pixel values are noisy

discretized sample. The Laplacian operators compute some quantity related to the Laplacian of the underlying image gray tone intensity surface. The zero-crossing operators determine whether or not the digital Laplacian or the estimated second direction derivative has a zero-crossing within the pixel. Edge detection operators are often implemented with convolution masks and discrete approximations to differential operators. These operators may return magnitude and direction information, some return magnitude only. Potential edge points are found by examining the relationship a pixel has with its neighbors; an edge implies a change in gray level.

3.1 Gradient operators

Gradient operators are based on the idea of using the first or second derivative of the gray level. The first derivative will mark edge points, with steeper gray level changes providing stronger edge points (large magnitudes). The second derivative returns two impulses, one on either side of the edge. An advantage of this is that if a line is drawn between the two impulses the position where this line crosses the zero axis is the center of the edge, which theoretically allows us to measure edge location to sub-pixel accuracy. Sub-pixel accuracy refers to the fact that zero-crossing may be at fractional pixel distance. In the traditional edge detector, the gradient of image is calculated using first order deviation [13]. When the gradient is above the threshold, there is an object in the image. As regarding to image $f(x,y)$, the gradient of point (x,y) is defined as follows:

$$\nabla f(x,y) = \begin{bmatrix} G_x & G_y \end{bmatrix} = \begin{vmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{vmatrix} \quad (1)$$

The weight of the vector is

$$\nabla f = mag(\nabla f) = \sqrt{G_x^2 + G_y^2} \quad (2)$$

And its direction as

$$\phi(x,y) = \arctan(G_y / G_x) \quad (3)$$

Gradient of every pixel of the image is calculated using the above three equations. In fact, small region pattern convolution is used to process the image. Gradient operators include Robert, Prewitt and Sobel operator. These operators have simplicity and detect the edges and their orientations but have an inaccurate sensitivity to noise. On the other hand, Laplacian operator uses second derivative, the operator is defined as:

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \quad (4)$$

The Laplacian operator finding the correct places of edge, testing wider areas around the pixel but malfunctioning at corners, curves. Also and where the gray level intensity function varies, not finding the orientation of edge because of using the Laplacian filter. Laplacian of Gaussian (LOG)



combined Gaussian filtering with the Laplacian and defined as:

$$G_{\delta}(x, y) = \frac{1}{2\pi\delta^2} \exp\left(-\frac{x^2 + y^2}{2\delta^2}\right) \quad (5)$$

Using Convolution of Gaussian operator with image $f(x, y)$, the image is smoothed, then the edge is detected using the following equation:

$$\nabla^2 [G_{\delta}(x, y) * f(x, y)] = [\nabla^2 G_{\delta}(x, y) * f(x, y)] \quad (6)$$

Gaussian edge detectors are symmetric along the edge, and reduce the noise by smoothing the image. The significant operators are Canny and Shen-Castan which convolve the image with the derivative of Gaussian for Canny and Shen-Castan. The Canny algorithm is an optimal edge detection method based on a specific mathematical model for edges. The edge model is a step edge corrupted by Gaussian noise. The Canny edge detector was devised to be an optimal edge detector, which satisfies all of the three performance criteria. The first criterion is to minimize the situations of detecting false edges and missing actual edges. The second criterion is to minimize the distance between the detected edges and actual edges. The third criterion is to minimize multiple responses to an actual edge, i.e. to ensure there is only one response for an actual edge point. Boie-Cox algorithm [14], is a generalization of the Canny algorithm using matched filters and Wiener filters. The Shen-Castan algorithm [15], was developed as an optimal solution to a specific mathematical model. Shen and Castan claim that their filter does better than Canny at finding the precise location of the edge pixel. The search includes steps similar to the Canny, but with modifications and extensions.

3.2-Advantages and disadvantages of edge detectors

The classical operator such as Sobel, and Prewitt which uses first derivative has very simple calculation to detect the edges and their orientations but has inaccurate detection sensitivity in case of noise. Laplacian of Gaussian (LOG) operator is represented as another type of edge detection operator which uses second derivative. It finds the correct places of edges and testing wider area around the pixel. The disadvantages of LOG operator is that it can not find the orientation of edge because of using the Laplacian filter. The other type of edge detection operator is the Gaussian edge detectors such as Canny, Shen Castan and Boie-Cox operators which are using probability for finding error rate and localization. Also it is symmetric along the edge and reduces the noise by smoothing the image. So it is performs the better detection in noise condition but unfortunately it has complex computing.

4- Morphological Filtering

Morphology relates to structure or form of objects. Morphological filtering simplified segmented images by smoothing out object outlines using filling small holes, eliminating small projections. Primary operations are dilation and erosion. These operations use a structuring element which determines exactly how object will be dilated or eroded. Dilation process expanding image objects by changing pixels with value of "0" to "1". On the other hand the erosion process shrinking binary objects by changing pixels with a value of "1" to "0". There is also a combination of dilation and erosion called opening and closing. Opening is erosion followed by dilation. Closing is a dilation followed by erosion. Morphological edge detection algorithm selects appropriate structuring element of the processed image makes use of the basic theory of morphology including erosion, dilation, opening and closing operation and synthesization operations of them get clear image edge. The effect of erosion and dilation operations is better for image edge by performing the difference between processed image and original image, but they are worse for noise filtering. As opposed to erosion and dilation, opening and closing operations are better for filtering.

5- Methods of the evaluation of performance for edge detectors

There are two methods to evaluate the performance of edge detectors, subjective methods and objective methods. Subjective methods borrowed from the field of psychology and use human judgment to evaluate the performance of edge detectors. More precisely, these methods involve presenting a series of edge images to several individuals and asking them to assign scores on a given scale. Even if these methods seem easy to put into practice, they have some drawbacks. The number of characteristics a human eye can distinguish is limited. For example, the eye cannot differentiate between two gray levels that are slightly different. As well, the judgment depends on the individual's experience and attachment to the method, as well as on the image type (i.e., multi-spectra, X-ray). The subjective measures are better than the objective measures for image evaluation, if the goal is to achieve high-quality images as defined by our visual perception. Subjective measures can be classified into three categories. The first type is referred to as impairment tests, where the test subjects score the images in terms of how bad they are. The second type is quality test, where the test subject rates the images in terms of how good they are. The third type is called comparison tests, where the images are evaluated on a side-by-side basis. The comparison type tests are considered to provide the most useful results, as they provide a relative measure, which is the easiest metric for most people to determine. Impairment and quality tests require an absolute measure, which is



more difficult to determine in an unbiased fashion. Table(1) explains the subjective fidelity scoring scales [16].

Impairment	Quality	Comparison
5-Imperceptible	A-Excellent	+2 much better
4-Perceptible, not annoying	B-Good	+1 better
3-Somewhat annoying	C-Fair	0 the same
2-Severely annoying	D-Poor	-1 worse
1-Unusable	E-Bad	-2 much worse

Table (1) Subjective fidelity scoring scales

On the other hand, objective methods use to measure the performance of edge detectors using signal to noise ratio and mean square error between the edge detectors images and the original one. The objective methods borrowed from digital signal processing and information theory, and provide us with equations that can be used to measure the amount of error in a processed image by comparison to known image. Although the objective methods are widely used, are not necessarily correlated with our perception of image quality. For instance, an image with a low error as determined by an objective measure may actually look much worse than an image with a high error metric. Commonly used objective measures are the root-mean-square error, e_{RMS} , the root-mean-square signal-to-noise ratio, SNR_{RMS} , and the peak signal-to-noise ratio, SNR_{PEAK} as in equations (7), (8) and (9) respectively.

$$e_{RMS} = \sqrt{\frac{1}{MN} \sum_{r=0}^{M-1} \sum_{c=0}^{N-1} [E(r,c) - O(r,c)]^2} \quad (7)$$

$$SNR_{RMS} = \sqrt{\frac{\sum_{r=0}^{M-1} \sum_{c=0}^{N-1} [E(r,c)]^2}{\sum_{r=0}^{M-1} \sum_{c=0}^{N-1} [E(r,c) - O(r,c)]^2}} \quad (8)$$

$$SNR_{PEAK} = 10 \log_{10} \frac{(L-1)^2}{\frac{1}{MN} \sum_{r=0}^{M-1} \sum_{c=0}^{N-1} [E(r,c) - O(r,c)]^2} \quad (9)$$

Where $O(r,c)$ is the original image, $E(r,c)$ is the reconstructed image and L is the number of gray level equal to 256.

6- Experiment of Edge Detection

In this paper image Lena (grayscale 256x256) which digitized at the University of Southern California was

chosen as one of many possible images for use by the research community. Lena image contains a nice mixture of detail, flat regions, shading, and texture that do a good job of testing various image processing algorithms. The sample resultant images from application of Boie-Cox, Shen-Castan, Canny, LOG, Prewitt and Sobel operators were applied to a Lena gray scale image with salt and pepper noise with probability 0.1 for both salt and pepper. The morphological filter is applied for a noisy image to get a clear image before edge detection of noisy image. Opening-closing operation is firstly used as preprocessing to filter noise. Then smooth the image by first closing and then dilation. The perfect image edge will be got by performing the difference between the processed image by above process and image before dilation.

The visual comparison of the resultant images can lead us to the subjective evaluation of the performances of selected edge detectors. Figure (1) shows the comparison between edge detection operators with salt and pepper noise with probability 0.1 for both salt and pepper without morphological filter. Figure (2) shows the comparison between operators after morphological filter on the noisy image with salt and pepper. The evaluating of edge detection performance obeys the three important criteria. First, the edge detector should find all real edges and not find any false edges. Second, the edges should be found in the correct place. Third, there should not be multiple edges found for a single edge. The same experiment done with Lena image is done with another grayscale image (256x256) designed by author as in figure (3-a). The original ground truth (GT) image in figure (3-c) was compared with the different edge detect operators for the original grayscale noisy image without morphological filter as in figure (3) and with morphological filter as in figure (4).

Objective methods are used in this experiment by calculating signal to noise ratio peak and the root mean square error between the edge detection images and the original ground truth image. Table (2-a,b) shows the peak signal to noise ratio and the root mean square error for different operators on noisy image before and after morphological filter is used.

Operators	SNR _{PEAK} before Morph.	SNR _{PEAK} after Morph.
Sobel	2.123	8.377
Prewitt	2.556	8.754
LOG	5.825	10.164
Canny	7.145	10.193
Shen-Castan	8.741	11.754
Boie-Cox	9.418	10.009

Table(2-a) shows SNR_{PEAK} before and after Morphological filter.



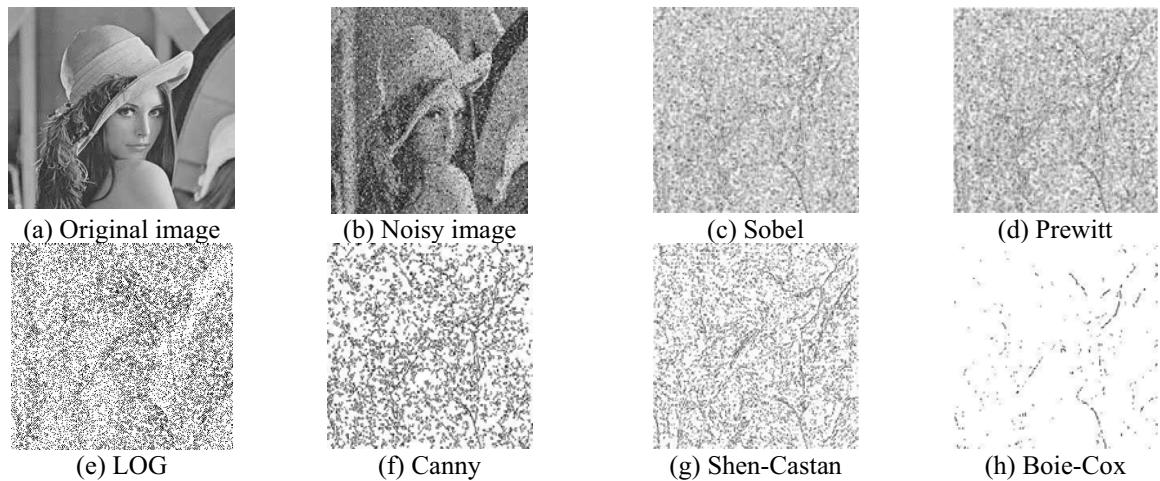


Figure (1) shows the comparison between edge detection operators with salt and pepper noise without morphological filter.

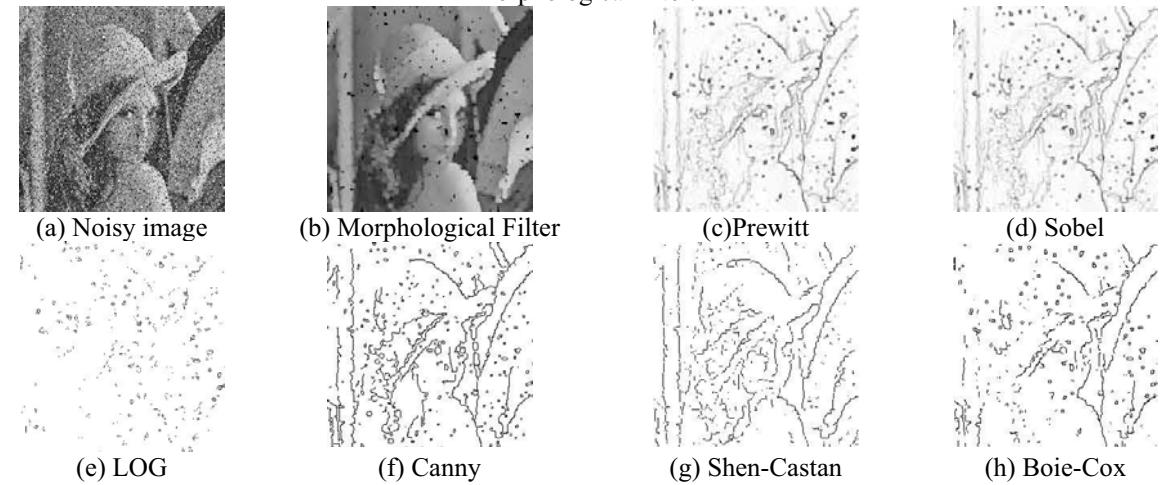


Figure (2) shows the comparison between operators after morphological filter on the noisy image with salt and pepper after morphological filter.

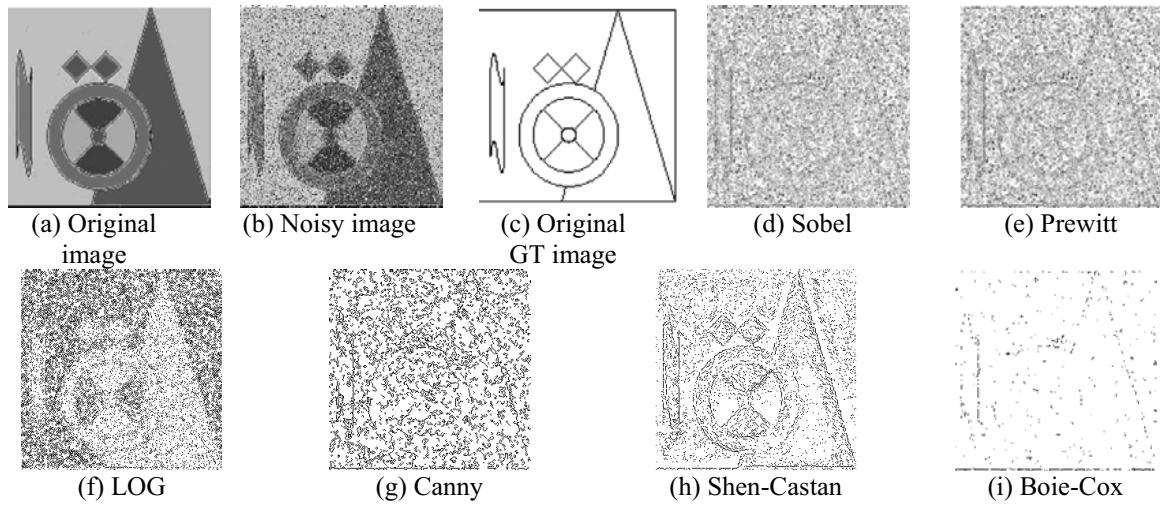


Figure (3) shows the comparison between edge detection operators with salt and pepper noise used an original image ground truth without morphological filter.



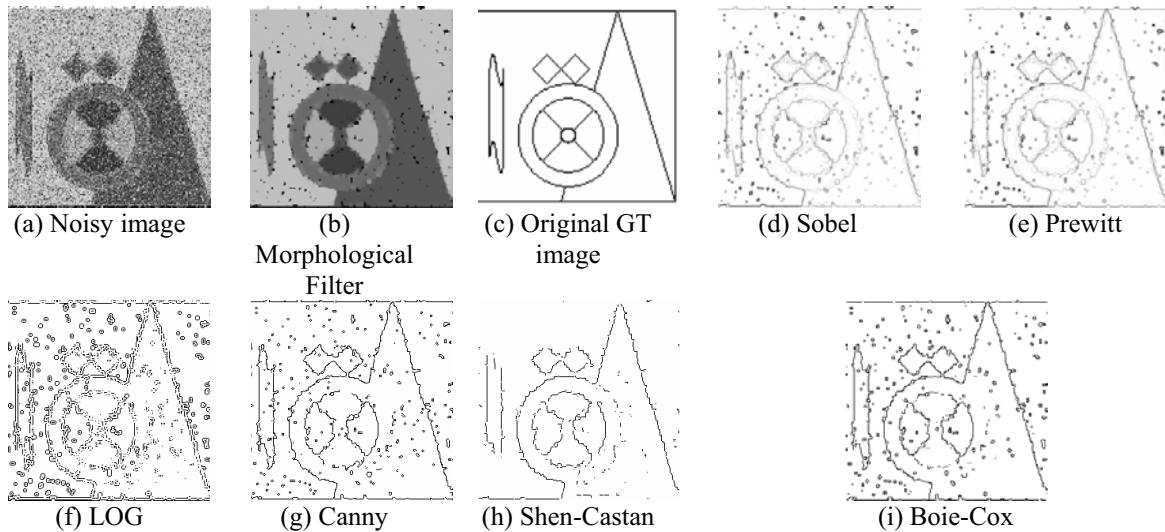


Figure (4) shows the comparison between edge detection operators with salt and pepper noise used an original image ground truth after morphological filter.

Operators	e_{RMS} before Morph.	e_{RMS} after Morph.
Sobel	199.707	97.206
Prewitt	189.994	93.007
LOG	130.408	79.126
Canny	112.023	78.870
Shen-Castan	93.217	65.895
Boie-Cox	85.450	80.558

Table(2-b) shows e_{RMS} before and after Morphological filter.

7- Discussion and conclusion

In this paper, subjective evaluation of edge detection result images show that Boie-Cox, Shen-Castan, Canny, LOG, Prewitt, and Sobel exhibit better performances respectively under noisy conditions. This is due to the Gaussian operators such as Canny and Shen-Castan operators using probability for finding error rate, localization and response. According to table (1) in case of noisy image with salt and pepper with probability 0.1 without morphological filter, the Prewitt and Sobel have poor quality. Laplacian of Gaussian has fair quality, and Canny, Shen-Castan and Boie-Cox have a good quality. After morphological filter the Prewitt and Sobel have fair quality. Shen-Castan, Boie-Cox and Canny have good quality. Shen-Castan, Boie-Cox and Canny are more acceptable than LOG, while LOG is more acceptable than Prewitt, and Sobel in case of noisy image without morphological filter. Boie-Cox, Shen-Castan, Canny and LOG are better than Prewitt and Sobel in case of noisy image after morphological filter. The objective evaluation of edge detection results as in table (2-a,b) agree the subjective evaluation that the Shen-Castan, Boie-Cox and Canny operators are better than LOG, Prewitt and Sobel in case of noisy image. The root mean square error of Canny, Boie-Cox and Shen-

Castan are less than LOG, while LOG less than Sobel and Prewitt. On the other hand the Single to Noise Ratio Peak for Canny, Boie-Cox and Shen-Castan are greater than LOG, while LOG is greater than Sobel and Prewitt in case of noisy image. After morphological filter the peak signal to noise ratio increases for all edge detector operators while the root mean square error decreases. The peak signal to noise ratio and the root mean square error in LOG operator and Canny operator are nearly the same after morphological filter. It means that the subjective and objective evaluations are reliable.

This paper concludes that the subjective and objective evaluations of noisy image shows that Boie-Cox, Shen-Castan, Canny, LOG, Prewitt, and Sobel exhibit better performances respectively. This is because the Gaussian edge detectors are symmetric along the edge and reduce the noise by smoothing the image. Also, it concludes that the morphological filter is more important as an initial process in the edge detection for noisy image.

8 - References

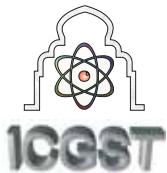
- [1] Canny, J., "A Computational Approach to Edge Detector", IEEE Transactions on PAMI, pp679-698, 1986.
- [2] Bovik, A. C., Huaung, T. S. and JR. D. C. M., "Non-parametric tests for edge detection noise", Pattern Recognition, 19:209-219, 1986.
- [3] Yakimovsky Y., "Boundary and object detection in real world image", Journal ACM, 23:599-618, 1976.
- [4] Raman Maini and J. S. Sobel, "Performance Evaluation of Prewitt Edge Detector for Noisy Images", GVIP Journal, Vol. 6, Issue 3, December 2006. www.icgst.com
- [5] Davis, L. S., "Edge detection techniques", Computer Graphics Image Process. (4), 248-270, 1995.



- [6] Sharifi, M.; Fathy, M.; Mahmoudi, M.T.; " A classified and comparative study of edge detection algorithms", International Conference on Information Technology: Coding and Computing, Proceedings, Page(s):117 – 120, 8-10 April 2002.
- [7] Shin, M.C.; Goldgof, D.B.; Bowyer, K.W.; Nikiforou, S.; " Comparison of edge detection algorithms using a structure from motion task", Systems, Man and Cybernetics, Part B, IEEE Transactions on Volume 31, Issue 4, Page(s):589-601, Aug. 2001.
- [8] Heath M. , Sarker S., Sanocki T. and Bowyer K., " Comparison of Edge Detectors: A Methodology and Initial Study", Proceedings of CVPR'96 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.143-148, 1996.
- [9] Rital, S.; Bretto, A.; Cherifi, H.; Aboutajdine, D.; "A combinatorial edge detection algorithm on noisy images", Video/Image Processing and Multimedia Communications 4th EURASIP-IEEE Region 8 International Symposium on VIPromCom, Page(s):351 – 355, 16-19 June 2002.
- [10] Li Dong Zhang; Du Yan Bi; "An improved morphological gradient edge detection algorithm", Communications and Information Technology, ISCIT 2005. IEEE International Symposium on Volume 2, Page(s):1280 – 1283, 12-14 Oct. 2005.
- [11] Zhao Yu-qian; Gui Wei-hua; Chen Zhen-cheng; Tang Jing-tian; Li Ling-yun; "Medical Images Edge Detection Based on Mathematical Morphology" Engineering in Medicine and Biology Society, IEEE-EMBS. 27th Annual International Conference, Page(s):6492 – 6495, 01-04 Sept. 2005.
- [12] Fesharaki, M.N.; Hellestrand, G.R.; "A new edge detection algorithm based on a statistical approach", Speech, Image Processing and Neural Networks, Proceedings, ISSIPNN '94., International Symposium, Page(s):21 - 24 vol.1, 13-16 April 1994.
- [13] Gonzalez , R and Woods, R., "Digital Image Processing" 2/E, Prentice Hall Publisher, 2002.
- [14] Cox, I.J.; Boie, R.A.; Wallach, D.A.; "Line recognition", Pattern Recognition Proceedings., 10th International Conference on Volume i, Page(s):639 - 645 vol.1, June 1990.
- [15] Castan, S.; Zhao, J. and Shen, J." New edge detection methods based on exponential filter", Pattern Recognition, Proceedings 10th International Conference on Volume i, Issue 16- Page(s):709 - 711, vol.1, Jun 1990.
- [16] Umbaugh, S. , "Computer Imaging: digital image analysis and processing", CRC press book, 2005.







Embedded Smoothing Processing in Spatial Gradient Operators, a Comparative Study

Tieling Chen

University of South Carolina Aiken, 471 University Parkway, Aiken, SC 29801, USA

tielingc@usca.edu

Abstract

This paper analyzes the embedded smoothing processing in some commonly used gradient operators used in edge detection, including Prewitt's operator, Sobel's operator, Canny's detector and a wavelet transformation based method. The effects of the embedded smoothing processing on edge detection are studied and compared. The paper also reduces the design of a gradient operator to the design of one or two one dimensional smoothing functions.

Keywords: smoothing, gradient operator, convolution, edge detection.

1. Introduction

Gradient operators are commonly used to detect abrupt changes of grey level values in digital images. A gradient operator consists of two partial derivatives, which determine the magnitude and the direction of the gradient. Since a digital image is represented by a discrete function, the partial derivatives can only be approximated. Generally, the partial derivatives are implemented as a pair of convolution masks, and most of them contain smoothing processing on the original image. As introduced in [4, 6], commonly used gradient operators include Prewitt's operator for horizontal and vertical edges, consisting of the partial derivative masks

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad (1)$$

and Sobel's operator for horizontal and vertical edges, consisting of

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}. \quad (2)$$

The above two gradient operators actually carry smoothing processing that may influent the computation of the gradient.

Prewitt and Sobel also have masks to detect diagonal edges. They are

$$\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \text{ for Prewitt, and}$$

$$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \text{ for Sobel.}$$

These two pairs can be obtained from (1) and (2) by rotations and they form two gradient operators with embedded smoothing processing. Since the paper studies the changes of grey level values along the horizontal and the vertical directions and decompose a derivative mask into a horizontal row mask and a vertical column mask, the masks detecting changes of the grey level values along the diagonal directions are not considered in the paper.

Canny's detector uses a three-stage algorithm to detect edges [1]. At the first stage, the original image is smoothed by convolving a two dimensional Gaussian function with a proper variance. At the second stage, masks detecting the horizontal, vertical and sometimes the diagonal edges are used to compute the gradient of the smoothed image. Edges are then traced at the third stage. Canny's detector is variously implemented. Some implementations of the second stage may slightly violate Canny's original intention if the above gradient masks are used because the embedded smoothing processing in these masks interferes the Gaussian smoothing processing used at the first stage. This slight variation has been ignored all along. We will check this influence in the next section.

Another important edge detection method is one that uses a wavelet transformation [3, 5]. In the one dimensional case, a smoothing function can derive a wavelet function. The wavelet transformation of a



given signal is proportional to the derivative of the smoothed signal that is smoothed by the smoothing function. When this is generalized to the two dimensional case, the partial derivatives of a two dimensional smoothing function give two wavelet functions. The wavelet transformation of an image is then a vector with two components, and this vector is in fact proportional to the gradient of the smoothed image. Since the wavelet function used in the method is derived from a smoothing function, the corresponding smoothing processing is embedded in the gradient operator.

Smoothing processing is almost always involved in spatial gradient operators. The smoothing processing is determined by the corresponding smoothing function, which plays an essential role in the design of a gradient operator. For Prewitt's or Sobel's gradient operator, each partial derivative mask smoothes the image in a different way that is in favor of edges along some specific direction. Hence, these operators are not really gradient operators according to the strict mathematical meaning.

In the paper we use the direct gradient operator without any embedded smoothing processing to compare with some commonly used gradient operators and then find the effects of their embedded smoothing processing. The design of a gradient operator is then reduced to the design of one or two smoothing functions.

The organization of the rest of the paper is as follows: In section 2, we first introduce the direct gradient operator without any embedded smoothing processing as the basic to be compared with other gradient operators. Then we check Sobel's and Prewitt's gradient operators to find the corresponding smoothing processing imposed on the direct gradient operator. And then the influences of Sobel's and Prewitt's gradient operators on Canny's detector are studied. The section finally introduces a method of designing a gradient operator with embedded smoothing processing. In section 3, we show experimental results on comparing some gradient operators with different embedded smoothing processing. Some general conclusions are given in section 4.

2. Embedded Smoothing Processing

A. Direct gradient operator

There are several ways to define the first order derivative of a one dimensional discrete function. Each way is an approximation of the definition in the continuous system. They might cause a little difference in applications, but the difference is negligible when we detect edges of objects bigger than a certain size, basically those objects causing visual interest in images. Suppose $f(x)$ is a one

dimensional discrete function and we use $Df(x)$ to denote its derivative at point x . Some commonly used definitions of $Df(x)$ includes

$$\begin{aligned} Df(x) &= f(x) - f(x-1), \\ Df(x) &= f(x+1) - f(x), \end{aligned}$$

and

$$Df(x) = (f(x+1) - f(x-1))/2.$$

Since the two dimensional generalizations of the first two versions do not give convolution masks with odd-value lengths, they are rarely put into use when implementing the partial derivative masks. For convenience, we drop the coefficient in the third version and define the derivative of a discrete function as

$$Df(x) = f(x+1) - f(x-1).$$

For an image represented by a two dimensional discrete function $f(x, y)$, its partial derivatives are similarly defined as

$$D_x f(x, y) = f(x+1, y) - f(x-1, y),$$

and

$$D_y f(x, y) = f(x, y+1) - f(x, y-1).$$

The mask forms of the partial derivatives are

$$D_x = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \text{ and } D_y = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}.$$

Therefore the partial derivatives of an image $f(x, y)$ can be obtained by the following convolutions:

$$D_x f(x, y) = (D_x * f)(x, y)$$

and

$$D_y f(x, y) = (D_y * f)(x, y).$$

We call the gradient operator consisting of D_x and D_y the direct gradient operator. No smoothing processing is embedded in the direct gradient operator.

B. Smoothing processing in Sobel's and Prewitt's gradient operators

If a two dimensional operator is decomposed into two separate one dimensional convolutions, we can go down to a deeper level to understand the effects of the operator on each variable. As we know, both Sobel's operator and Prewitt's operator are separable. For example, the partial derivative masks of Sobel's operator shown in (2) are both separable:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \times [1 \ 2 \ 1]$$

and

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times [-1 \ 0 \ 1].$$



In the above, the first separation shows that Sobel's partial derivative operation with respect to x is equivalent to smoothing the given image with respect to y first, with the row smoothing mask $[1 \ 2 \ 1]$, and then taking the derivative with respect to x . Of course we can also understand it as taking the derivative first and then performing the smoothing processing because the order does not matter. In the discussion that follows we will mention the smoothing processing first and then the derivative.

Similarly, the second separation shows that Sobel's partial derivative operation with respect to y is equivalent to smoothing the given image with respect to x first, with the column smoothing mask $[1 \ 2 \ 1]^T$, the transpose of $[1 \ 2 \ 1]$, and then taking the derivative with respect to y . We can expand the row mask and the column mask into the following square masks since this does not change anything in the convolutions:

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Clearly, with Sobel's operator the original image is smoothed in two different ways and the partial derivatives are not taken over the same smoothed image. The situation is quite similar for Prewitt's operator. The two masks in (1) are also separable,

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \times [1 \ 1 \ 1]$$

and

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \times [-1 \ 0 \ 1].$$

Each convolution is equivalent to a two-step operation that first smoothes the original image along one direction and then takes a partial derivative over the smoothed image along the perpendicular direction. The original image is smoothed in two different ways, with the following two different smoothing masks,

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

In both Sobel's and Prewitt's operators, the two partial derivatives are taken over two different smoothed images. Since the gradient is computed from two different sources, these operators are not strict mathematical gradient operators. They are approximations of the gradient operation that are in favor of horizontal and vertical edges.

C. Smoothing processing in Canny's detector

Canny's detector is an edge detection algorithm with three stages. At stage one the original image is smoothed by a Gaussian function and at stage two the gradient of the smoothed image is computed. At stage three the detector uses the information of the gradient obtained at stage two to extract edges. We only check the computation of the gradient and put the focus on the implementation.

There are many ways to implement Canny's detector, including using Sobel's operator to compute the gradient of the smoothed image. The implementation of the gradient of the smoothed image would deviate from Canny's original intention if a gradient operator carrying smoothing processing, such as Prewitt's operator or Sobel's operator, were used in the gradient computation. Because the Gaussian smoothing at stage one is interfered by the embedded smoothing processing in the gradient operator.

Sobel's operator is commonly used to compute the gradient of the smoothed image. Suppose f is the original image, g is the Gaussian function selected in the smoothing processing at stage one, and S_x is Sobel's derivative mask with respect to x . Then the derivative with respect to x found by Sobel's operator is $S_x * (f * g)$. We have

$$\begin{aligned} S_x * (f * g) &= f * (g * S_x) = f * ((g * D_x) * [1 \ 2 \ 1]) \\ &= f * (D_x(g) * [1 \ 2 \ 1]) = D_x(f * g * [1 \ 2 \ 1]). \end{aligned}$$

The final expression in the above shows that the derivative is actually taken after the original image is smoothed by Gaussian and the row mask $[1 \ 2 \ 1]$. Similarly if Sobel's operator is used, the derivative with respect to y is taken after the original image is smoothed by Gaussian and the column mask $[1 \ 2 \ 1]^T$.

Usually in implementations, the convolution operation does not unnecessarily flip one of its operands, as required by the strict mathematical definition. But in mathematical inductions the influence of the flip should be taken into considerations, especially when the mask is not symmetric. In the above mathematical induction, the mask D_x is flipped over in the convolutions.

On the four boundaries of a given image, we pad extra rows and columns of zeros such that the convolution can still be performed on the boundaries.

Now we check $D_x(g)$ with a concrete 3 by 3 Gaussian mask,



$$g = \begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix}. \quad (3)$$

We have

$$D_x(g) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{16} & -\frac{2}{16} & -\frac{1}{16} \\ -\frac{2}{16} & -\frac{4}{16} & -\frac{2}{16} \\ 0 & 0 & 0 \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix}, \quad (4)$$

where the mask D_x is flipped in the convolution. We can compute $D_y(g)$ in a similar way and obtain

$$D_y \begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix} = \begin{bmatrix} -\frac{1}{16} & -\frac{2}{16} & 0 & \frac{2}{16} & \frac{1}{16} \\ -\frac{2}{16} & -\frac{4}{16} & 0 & \frac{4}{16} & \frac{2}{16} \\ -\frac{1}{16} & -\frac{2}{16} & 0 & \frac{2}{16} & \frac{1}{16} \end{bmatrix}. \quad (5)$$

If Sobel's operator is used at stage two, an extra convolution is performed on each mask, resulting in a pair of biased masks, $D_x(g)*[1 \ 2 \ 1]^T$ and

$$D_y(g)*[1 \ 2 \ 1]^T, \text{ which are}$$

$$\begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ 16 & 16 & 16 & 16 & 16 \\ -2 & -8 & -12 & -8 & -2 \\ 16 & 16 & 16 & 16 & 16 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 16 & 16 & 16 & 16 & 16 \\ 1 & 4 & 6 & 4 & 1 \\ 16 & 16 & 16 & 16 & 16 \end{bmatrix} \quad (6)$$

and

$$\begin{bmatrix} -1 & -2 & 0 & \frac{2}{16} & \frac{1}{16} \\ 16 & 16 & 16 & 16 & 16 \\ -4 & -8 & 0 & \frac{8}{16} & \frac{4}{16} \\ 16 & 16 & 16 & 16 & 16 \\ -6 & -12 & 0 & \frac{12}{16} & \frac{6}{16} \\ 16 & 16 & 16 & 16 & 16 \\ -4 & -8 & 0 & \frac{8}{16} & \frac{4}{16} \\ 16 & 16 & 16 & 16 & 16 \\ -1 & -2 & 0 & \frac{2}{16} & \frac{1}{16} \\ 16 & 16 & 16 & 16 & 16 \end{bmatrix}. \quad (7)$$

We can see the difference between the partial derivative masks shown at the right sides of equations (4) and (5) and the partial derivative masks shown in (6) and (7). To see this clearer we separate the above partial derivative masks into products of column masks and row masks. We only look at the mask at the right side of equation (4) and the mask in (6). The comparison of the other pair is similar.

The mask obtained by the direct gradient operator is separated as

$$\begin{bmatrix} -\frac{1}{16} & -\frac{2}{16} & -\frac{1}{16} \\ -\frac{2}{16} & -\frac{4}{16} & -\frac{2}{16} \\ 0 & 0 & 0 \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix} = \begin{bmatrix} -\frac{1}{4} \\ -\frac{2}{4} \\ 0 \\ \frac{2}{4} \\ \frac{1}{4} \end{bmatrix} \times \begin{bmatrix} \frac{1}{4} & \frac{2}{4} & \frac{1}{4} \end{bmatrix} \quad (8)$$

and the mask obtained by Sobel's operator is separated as

$$\begin{bmatrix} -\frac{1}{16} & -\frac{4}{16} & -\frac{6}{16} & -\frac{4}{16} & -\frac{1}{16} \\ -\frac{2}{16} & -\frac{8}{16} & -\frac{12}{16} & -\frac{8}{16} & -\frac{2}{16} \\ 0 & 0 & 0 & 0 & 0 \\ \frac{2}{16} & \frac{8}{16} & \frac{12}{16} & \frac{8}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{4}{16} & \frac{6}{16} & \frac{4}{16} & \frac{1}{16} \end{bmatrix} = \begin{bmatrix} -\frac{1}{4} \\ -\frac{2}{4} \\ 0 \\ \frac{2}{4} \\ \frac{1}{4} \end{bmatrix} \times \begin{bmatrix} \frac{1}{4} & \frac{4}{4} & \frac{6}{4} & \frac{4}{4} & \frac{1}{4} \end{bmatrix}. \quad (9)$$

The column masks at the right hand side of both (8) and (9) are the same, which is a smoothed derivative produced by smoothing $[-1 \ 0 \ 1]^T$ with $\left[\frac{1}{4} \ \frac{2}{4} \ \frac{1}{4}\right]^T$. But the row masks are different. We normalize the row mask in (9) to $\left[\frac{1}{16} \ \frac{4}{16} \ \frac{6}{16} \ \frac{4}{16} \ \frac{1}{16}\right]$ such that the sum of the entries is one. Compared with the row mask $\left[\frac{4}{16} \ \frac{8}{16} \ \frac{4}{16}\right]$ that is same as the one in (8), the mask $\left[\frac{1}{16} \ \frac{4}{16} \ \frac{6}{16} \ \frac{4}{16} \ \frac{1}{16}\right]$ takes a portion of the weight on the center pixel and evenly assigns the weight to the two pixels that are one pixel apart from the center. Obviously this smoothing mask is in favor of the horizontal edges.

To implement Canny's detector correctly, derivative masks involving smoothing processing such as the masks in Sobel's operator and Prewitt's operator should not be used. When a Gaussian function is selected, a smoothing mask can be obtained by



sampling the Gaussian function. Then the smoothed image convolves D_x and D_y directly to compute its partial derivatives and then the gradient. Nevertheless, if derivative masks involving smoothing processing are used some additional effects may be obtained, such as edges along some specific directions are strengthened. But these effects are extra to Canny's operator.

D. Design gradient operators

From the above discussion we find that the embedded smoothing processing in a gradient operator is the characteristic of the operator. For each separable partial derivative mask of a gradient operator, the smoothing processing can be decomposed into row smoothing and column smoothing. One of them is used to smooth the corresponding derivative. Since derivative is sensitive to noise, proper smoothing is good for the computation. But too much smoothing would attenuate the difference among pixel values such that the derivative becomes dull. Hence the row smoothing and the column smoothing do not have to be the same. The original image may not be smoothed the same way when computing the partial derivatives, as in Prewitt's and Sobel's operators; therefore the operator is an approximated gradient operator.

The design of a gradient or an approximated gradient operator is reduced to the design of one or two smoothing functions. The idea can be used in a wavelet transformation based gradient operator. We can start the operator with a one dimensional smoothing function. A smoothing function $\theta(x)$ is differentiable, even and decreasing in $|x|$, has a finite support and integral one. Its derivative $\phi(x)$ gives a wavelet function. By dilating a smoothing function $\theta(x)$ with a dilation scale s one can get a so-called dilated smoothing function $\theta_s(x)$, defined as

$$\theta_s(x) = \frac{1}{s} \theta\left(\frac{x}{s}\right).$$

This leads to a corresponding dilated wavelet function $\phi_s(x) = \phi(x/s)/s = s(\theta_s(x))'$. The wavelet transformation of a function f at dilation level s , denoted by $(W_s f)(x)$, is given by the following convolution:

$$(W_s f)(x) = (f * \phi_s)(x).$$

Since

$$(f * \phi_s)(x) = f * \left(s \frac{d}{dx} \theta_s(x) \right) = s \frac{d}{dx} (f * \theta_s)(x),$$

the wavelet transformation is actually proportional to the derivative of the smoothed function. From the one dimensional smoothing function, we can make a separable two dimensional function that can also be dilated, $\Theta_s(x, y) = \theta_s(x)\theta_s(y)$. Its partial derivatives give two wavelet functions, dilated at scale level s .

The wavelet transformation of $f(x, y)$ at scale level s is a vector with two components,

$$(W_s f)(x, y) = (((f(\cdot, y) * \phi_s)(x) * \theta_s)(y), ((f(x, \cdot) * \phi_s)(y) * \theta_s)(x)),$$

which is equal to

$$s \left(\frac{d}{dx} (f * \Theta_s)(x, y), \frac{d}{dy} (f * \Theta_s)(x, y) \right).$$

Hence the wavelet transformation is equivalent to a gradient operator. This method heavily depends on the design of the smoothing function $\theta(x)$ [2]. A good smoothing function makes the method well meet Canny's criteria for edge detection algorithms [1]. The evolution of the wavelet transformation across dilation scales can be used to characterize different types of edges. In implementations, once a smoothing function is chosen, the dilated smoothing function and the corresponding dilated wavelet function can be sampled dynamically at different scales.

To strengthen the edges along specific directions, the two factors $\theta_s(x)$ and $\theta_s(y)$ of the two dimensional smoothing function do not have to be dilated to the same scale level. For example, to strength the edges along the x direction we can dilate $\theta_s(y)$ to a higher level and keep the dilation of $\theta_s(x)$ at a lower level. The things are just opposite when edges along the y direction are to be strengthened. In this way, the partial derivatives are not taken over the same smoothed image and the method is regarded as an approximation of the gradient method. It is also feasible to use two different one dimensional smoothing functions ξ and η to build two dimensional smoothing functions $\Theta_1(x, y) = \xi(x)\eta(y)$ and $\Theta_2(x, y) = \eta(x)\xi(y)$, which are used in separate partial derivatives. One of ξ and η handles smoothing the derivative and the other handles pure smoothing on the image.

In general, to design partial derivative masks of a gradient operator we can start with a two dimensional smoothing mask. The smoothing mask does not have to be separable. The partial derivative mask with respect to x is given by convolving the smoothing mask with the column mask $[1 \ 0 \ -1]^T$, the flipped D_x , and the partial derivative mask with respect to y is given by convolving the smoothing mask with the row mask $[1 \ 0 \ -1]$, the flipped D_y . The resulting pair is just a gradient operator.

But we have more flexibility if we design separable masks since we can start with one dimensional smoothing masks. Let us use the smoothing mask $[1 \ 3 \ 4 \ 3 \ 1]$, sampled from a Gaussian function,



to illustrate the design procedure. For convenience, the denominator 12 of each entry is dropped. This mask covers a bigger neighborhood of the center than the smoothing mask $[1 \ 2 \ 1]$ embedded in Sobel's operator. The derivative of the mask is

$$[1 \ 3 \ 4 \ 3 \ 1] * [1 \ 0 \ -1] = [-1 \ -3 \ -3 \ 0 \ 3 \ 3 \ 1].$$

Then the two partial derivative masks of the gradient operator are given by

$$[-1 \ -3 \ -3 \ 0 \ 3 \ 3 \ 1]^T \times [1 \ 3 \ 4 \ 3 \ 1] \quad (9)$$

and

$$[1 \ 3 \ 4 \ 3 \ 1]^T \times [-1 \ -3 \ -3 \ 0 \ 3 \ 3 \ 1]. \quad (10)$$

Each of the above separable partial derivative masks is a product of a column matrix and a row matrix. Then a convolution with the derivative mask is reduced to two separate convolutions, one of which handles the smoothing processing along one direction and another handles a smoothed derivative along the perpendicular direction. In many situations we want the derivative to be more sensitive to changes so we do not smooth the derivative too much. In the extreme case, each partial derivative is not smoothed, we get

$$[-1 \ 0 \ 1]^T \times [1 \ 3 \ 4 \ 3 \ 1]$$

and

$$[1 \ 3 \ 4 \ 3 \ 1]^T \times [-1 \ 0 \ 1],$$

which gives a generalized Sobel operator, with the partial derivative masks

$$\begin{bmatrix} -1 & -3 & -4 & -3 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 4 & 3 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & 0 & 1 \\ -3 & 0 & 3 \\ -4 & 0 & 4 \\ -3 & 0 & 3 \\ -1 & 0 & 1 \end{bmatrix}. \quad (11)$$

3. Experimental Results

Now we examine the effects of the gradient operators discussed in the paper with Lena's image in Figure 1. For each operator, only the gradient are checked and further processing to extract edges is not under our considerations. To compare the effects of different operators, the grey level values of each gradient image are rescaled to the range from 0 to 255. When finding the difference between two images, we subtract the second image from the first image and then shift the grey level values of the result to a non-negative range. We call this normalizing the difference.

We will make various comparisons. We will check the differences among Sobel's operator, Prewitt's operator, a wavelet transformation based method, and the operator with direct partial derivatives. We will also check the difference between two different implementations of

Canny's detector. Finally we will check the effects of the operators studied in section 2. D.



Figure 1. Lena's image.

Let us compare the direct gradient operator with Sobel's operator first. Figure 2 (a) is the norm of the gradient of the original image produced by the direct gradient operator. Figure 2 (b) is the corresponding gradient image produced by Sobel's operator. To give clear appearances, Figure 2 (a) and Figure 2 (b) are negated.

The difference between these two gradient images is almost invisible with naked eyes. The pixel by pixel comparison, without negation, is shown in Figure 2 (c). The difference is normalized. A bright pixel means the grey level value at the same pixel in the gradient image corresponding to Figure 2 (a) is bigger than that corresponding to Figure 2 (b). A dark pixel means the opposite. The difference is mainly because the edges produced by Sobel's operator are shifted by the smoothing processing of the operator.

Prewitt's operator shows a similar effect. Figure 3 is the normalized difference between the direct gradient image corresponding to Figure 2 (a) and the gradient image obtained by Prewitt's operator. The difference is mainly resulted from the shifts caused by the smoothing processing in Prewitt's operator.

We use the cubic spline smoothing function

$$\theta(x) = \begin{cases} 8(x+1)^3/3 & -1 \leq x < -1/2; \\ -8x^3 - 8x^2 + 4/3 & -1/2 \leq x < 0; \\ 8x^3 - 8x^2 + 4/3 & 0 \leq x < 1/2; \\ 8(1-x)^3/3 & 1/2 \leq x < 1; \\ 0 & \text{otherwise.} \end{cases}$$

to derive a wavelet function and then dilate it to the

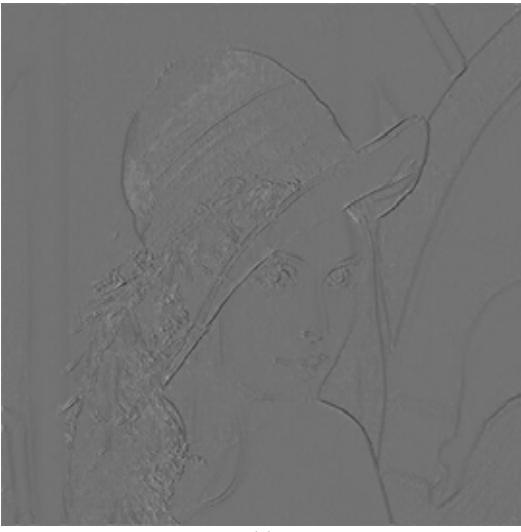




(a)



(b)



(c)

Figure 2. (a) The negated gradient image produced by the direct differentiation without smoothing. (b) The negated gradient image produced by Sobel's operator. (c) The normalized difference between the two gradient images.

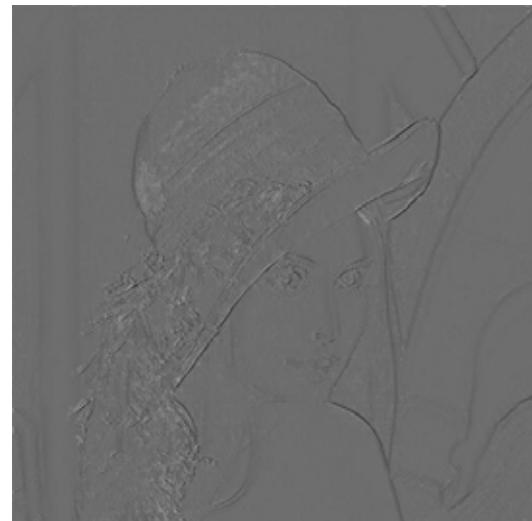


Figure 3. The normalized difference between the gradient image corresponding to Fig 2 (a) and the gradient image obtained by Prewitt's operator.



Figure 4. The normalized difference between the gradient image corresponding to Fig 2 (a) and the gradient image obtained by a wavelet transformation based method.

scale level $s = 3$. The wavelet transformation is then performed on the image in Figure 1 to find its gradient. The normalized difference between the direct gradient image corresponding to Figure 2 (a) and the gradient image obtained by the wavelet transformation is shown in Figure 4. Edge shifts caused by the embedded smoothing processing is still the main reason of the difference.

For Canny's detector, we use mask (3) to smooth the original image at the first stage. We show the difference between two implementations of the gradient. One



implementation is with the direct gradient operator without any extra smoothing. Another one is with Sobel's operator, which carries extra smoothing processing. The normalized difference between the gradient images is quite slight. With naked eyes, the difference between the two gradient images is almost unnoticeable. To see the difference clearly, the normalized difference is negated, shown in Figure 5. The real difference is small in grey level values because the Gaussian smoothing dominates the smoothing in Sobel's operator. The slight difference on locations can also be seen. This is because the extra smoothing processing in Sobel's operator shifts the edges. In the implementations of Canny's detector, many people use Sobel's operator to compute the gradient. The slight difference shown in Figure 5 implies that the approximation is acceptable.

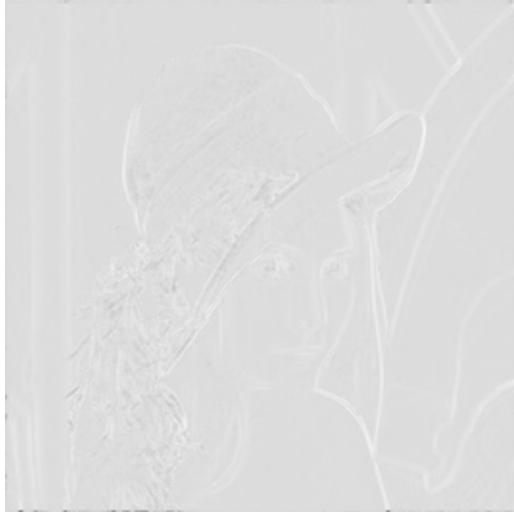


Figure 5. The negated normalized difference between two implementations of Canny's detector. One uses derivatives without smoothing to compute the gradient and the other one uses Sobel's operator to compute the gradient.

Next we look at the gradient operator consisting of the masks (9) and (10). The normalized difference between the gradient image produced by the direct gradient operator without smoothing and the gradient image produced by the gradient operator given by (9) and (10) is shown in Figure 6. The difference in grey level values is apparent since the derivatives in the operator given by (9) and (10) are dull because of the heavy smoothing. The shifts of edges are also clearly seen in the difference image, implying localization errors sacrificed to the smoothing processing.

Finally we check the generalized Sobel operator with partial derivative masks shown in (11). This operator smoothes the original image in two different heavily biased ways. The difference between the gradient image

produced by the classical Sobel's operator and the gradient image produced by the generalized Sobel operator is given in Figure 7 (a). To show the difference more clearly, the image is negated. The normalized difference is slight. In Figure 7 (b), we also show the normalized difference between the gradient image produced by direct differentiation and the gradient image produced by the generalized Sobel operator. The image is negated too. The white and black edges imply shifts of edges caused by the biased partial derivatives.



Figure 6. The normalized difference between two gradient images. One is produced by the direct gradient operator and the other one is produced by the partial derivatives shown in (9) and (10);

4. Conclusion

The paper studies the influences of the embedded smoothing processing in some commonly used gradient operators in the gradient computation. Generally a gradient operator carries smoothing processing, explicitly or implicitly, and the smoothing processing is the characterization of the operator. A separable partial derivative mask can be decomposed as a product of a two one dimensional masks. One of them handles smoothing the original image along one direction and the other one handles a one dimensional derivative along the perpendicular direction. Generally the one dimensional derivative is also smoothed. The smoothing function that smoothes the derivative does not have to be the same as what handles the smoothing processing. The design of a gradient operator is then reduced to the design of one or two smoothing functions.

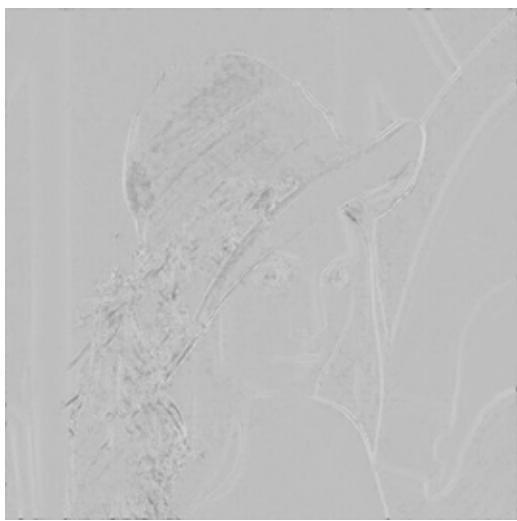
Differences exist among gradient operators using different ways of smoothing and basically they lie in two aspects: grey level value differences and edge location differences. When the sizes of partial derivative masks are small, as in Sobel's operator, Prewitt's



operator, or the direct gradient operator, these differences are not apparent. For Canny's operator, when computing the gradient of the image smoothed by a Gaussian function, the smoothing processing embedded in the gradient operator is mixed with the smoothing processing of Gaussian. If the sizes of the partial derivative masks of the gradient operator are small, like Sobel's operator and Prewitt's operator, the influence of the embedded smoothing processing is also small and probably can be ignored. If the sizes of the partial derivative masks are big, the influence of the embedded smoothing processing can not be ignored.



(a)



(b)

Figure 7. (a) The negated normalized difference between the gradient image produced by the classical Sobel's operator and the gradient image produced by the generalized Sobel's operator. (b) The negated normalized difference between the gradient image produced by direct differentiation and the gradient image produced by the generalized Sobel's operator.

5. References

- [1] J. Canny, A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8(6), pp. 679-698, 1986.
- [2] T. Chen, An Edge Detection Algorithm using Smoothing Functions, GVIP Journal, special issue: Edge Detection, 2006.
- [3] T. Chen, J. Wang, and K. Zhang, A Wavelet Transform Based Method for Road Centerline Extraction, Photogrammetric Engineering & Remote Sensing, vol. 70, no. 12, pp. 1423-1431, 2004.
- [4] R. Gonzalez and R. Woods, Digital Image Processing, 2nd Edition, Prentice Hall, pp. 577-581, 2002.
- [5] S. Mallat and S. Zhong, Characterization of Signals from Multiscale Edges, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14(7), pp. 710-732, 1992.
- [6] M. Roushdy, Comparative study of Edge Detection Algorithms Applying on the Grayscale Noisy Image Using Morphological Filter, GVIP Journal, vol. 6(4), pp. 17-23, 2006.

Biography



Tieling Chen is an Assistant Professor in the Mathematical Sciences Department of the University of South Carolina at Aiken, USA, since 2002. He received a Ph. D. degree in Mathematics in 2001 and a M. S. degree in Computer Science in 2002 from the University of Western Ontario, Canada. His present research interest is mathematical operations in image processing.

