A METHOD FOR DENOISING IMAGE CONTOURS

Ovidiu **COSMA**
Technical University of Cluj-Napoca, North University Center of Baia Mare
ovidiu.cosma@yahoo.com

Keywords: edge detection, noise removal

Abstract*: The edge detection techniques have to compromise between sensitivity and noise. In order for the main contours to be uninterrupted, the level of sensitivity has to be raised, which however has the negative effect of producing a multitude of insignificant contours (noise). This article proposes a method of removing this noise, which acts directly on the binary representation of the image contours.*

1. INTRODUCTION

Contour detection is a fundamental technique of image processing. Edges and contours play a key role in the human visual system and underpin computer vision applications, such as robot guidance, automatic inspection, process control, and medical applications. Contours play a key role in the perception of images. It is often possible to understand the content of an image that has been reduced to a few basic lines. Consequently, we can conclude that a major part of information in an image lays in objects contours.

Despite the importance of the subject, at present there is no known technique of tracing perfect contours. This is due to the fact that there is a need to compromise between the following two complementary features: thin and uninterrupted contour lines and the absence of false contours (no noise).

2. METHODS OF EDGE DETECTION

The contours in an image are made up of pixels for which brightness changes steeply after a certain orientation. The first contour detection methods are based on gradients, which are calculated using first-order derivatives in horizontal and vertical directions [1, 2]. If we look at a digital image as a function of two variables $f(x, y)$, then the gradient vector is defined according to relation (1), and the gradient magnitude is defined in relation (2).

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x}(x,y) \\ \frac{\partial f}{\partial y}(x,y) \end{bmatrix} \tag{1}$$

$$|\nabla f(x,y)| = \sqrt{(\frac{\partial f}{\partial x}(x,y))^2 + (\frac{\partial f}{\partial y}(x,y))^2} \tag{2}$$

The points on the contours are those for which the magnitude of the gradient exceeds a certain threshold. A simple binarization operation can be applied for tracing the contours. The following two filters, known as Roberts filters [1, 2], can be used to estimate the gradient. They determine diagonal contours, but are not very sensitive to orientation. Because they are as small as possible, they can detect the finest contours.

$$H_1^R = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \qquad H_2^R = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3}$$

Any contouring technique has to deal with noise-related issues, because they can be mistakenly interpreted as edges. For this reason, it is preferable to apply a smoothing filter before edge detection [1, 2]. Any smoothing filter would solve this problem, but it is preferable one that does not affect too much the edges. The Gaussian filter defined in relation (4) is frequently used because it is configurable by standard deviation σ, which determines the width of the bell.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

This smoothing step may be omitted if the filters that are used calculate the average values of the gradient. The next two filters (Sobel) calculate at each step the gradient average on 3 lines, respectively on 3 columns [1-3]. *Figure 1.b* shows the contours of the image in *figure 1.a*, drawn by applying the Sobel filters, followed by a thresholding operation.

$$H_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \qquad H_x^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{5}$$

The main problem of all the gradient based techniques is the fact that they rely on the first-order derivative, which is nonzero in ramps. For this reason, they draw relatively thick contours and artificial thinning techniques are required. To solve the problem, edge detection techniques based on the second order derivative were developed [1, 2]. They draw thinner contours, but they have the disadvantage of doubling roof edges.

The Laplace operator for an image $f(x, y)$ is obtained by summing the second order derivatives calculated in the horizontal and vertical directions.

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2}(x,y) + \frac{\partial^2 f}{\partial y^2}(x,y) \tag{6}$$

Figure 2.a was generated using the Laplace operator followed by thresholding. The Laplacian filters can detect fine edges, but they have a high sensitivity to noise. As a result, a pre-smoothing operation is required. For efficiency, the two operations (smoothening and edge detection) can be combined into a single one using a Laplacian of Gaussian (LoG) filter that can be calculated based on the following relation:

$$LoG(x,y) = -\frac{1}{\pi\sigma^2}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{7}$$

Figures 2.b and *2.c* were generated using the LoG filter, with a standard deviation σ of 0.8 and 1.4 respectively.

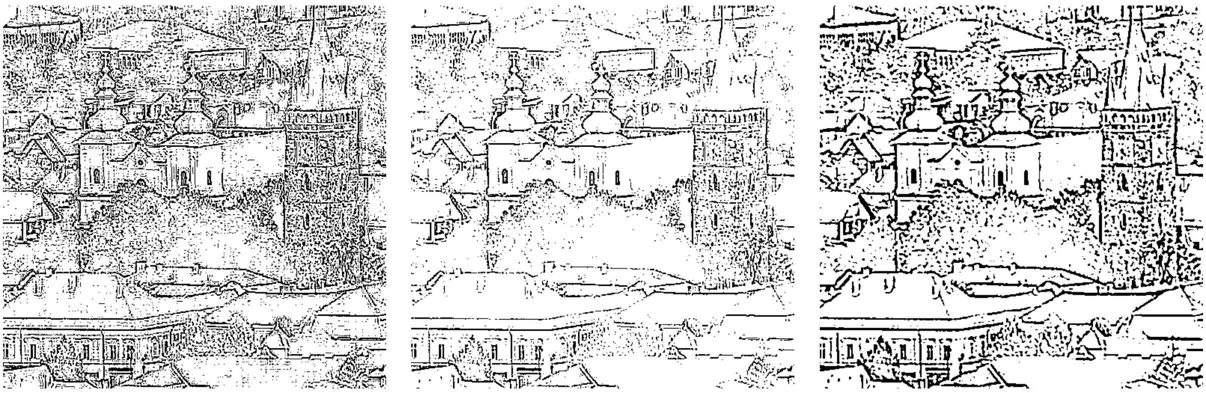
An optimized contour detecting method known as the Canny operator is proposed in [4]. The proposed method comprises the following steps:

1. Apply a Gaussian smoothing filter to the original image, for noise reduction.
2. Use the Sobel filters to determine both gradient magnitude and gradient orientation.
3. Apply a contour-thinning algorithm that reduces all contours to one pixel thickness. In the case of thick contours, only the pixel with maximum gradient magnitude in the direction of the gradient orientation remains. All other pixels will be removed from the contour.
4. To avoid discontinuous contours, a hysteresis thresholding algorithm is proposed. This algorithm uses two thresholds. In the first step, a higher threshold is used, based on which the pixels belonging to contours are set. In the second step, a lower threshold is used, based on which other pixels are added to the contours, provided they are connected to the pixels selected in the first step.

The contour image in *figure 1.c* was generated by applying the Canny operator on the image in *figure 1.a.*

a.) Original image *b.) Sobel filter* *c.) Canny operator*

Fig. 1. Original image and contour images



a.) Laplacian filter *b.) LoG filter,* $\sigma = 0.8$ *c.) LoG filter,* $\sigma = 1.4$

Fig. 2. Image contours

3. RECENT RESULTS

A Laplacian of B-spline operator for edge detecting is proposed in [6]. The authors claim that the new operator outperforms the Canny and the LoG operators in the case of noisy images. In [7] is presented a new method for detecting the boundaries of homogenous regions. The method is based on gradient intensity and texture variations. The wavelet transform is used for calculating a texture representation. A method based on non-linear filtering for structure preserving noise reduction and edge / corner detection is presented in [8]. A multi resolution edge detection technique is proposed in [9]. For increasing accuracy, the algorithm performs edge pattern analysis. A parallel implementation that runs in real-time is presented. In [10] is presenred a new contour detection algorithm that is based on quantum entropy. The efficiency of the method is demonstrated using different examples. The performance is evaluated using the peak-signal-to-noise ratio. An edge detecting method based on fractional Fourier transformation is presented in [11]. The authors show that the proposed method performs better than classical methods in terms of resilience to noise. A method of contour detection based on the Canny operator enhanced with an ant colony optimization algorithm is presented in [12]. A new contour detection method for color images is presented in [13]. The method is based on the vector angle between adjacent pixels. An edge detection method based on a logarithmic image processing model is presented in [14]. The author claims that the proposed method performs better than the classical methods in the presence of noise. A new contour detection method is proposed in [15]. The method is based on gradient calculation, followed by a frequency domain filtering step.

A multiresolution contour detection method is proposed in [16]. The method is based on Bayesian denoising and inhibition of the surroundings. The gradient is calculated at different resolutions then the edges are denoised. In the surrounding inhibition step, the edges that represent texture are recognized and suppressed. A technique based on local and global analysis is proposed in [17]. The contour detection is performed by integrating the local edges detected by filtering with a global saliency map. A histogram difference function is presented, for estimating the probability of the pixels to belong to contours. A multi-scale morphological edge detection algorithm for noisy images is presented in [18]. The key features of the algorithm are robustness and accuracy. An evaluation of the linear methods for image contour detection is presented in [19]. The methods are evaluated on basis of sampling errors, output noise level, and computational complexity.

4. THE PROPOSED METHOD

Of all the filters for edge detection, the Laplacian filter has the most promising results, but they are shadowed by the high level of noise. The only way to reduce noise is to raise the binarization threshold, but this operation may cause important contours to be interrupted. The LoG filter introduces an additional parameter that can adjust the noise level. This is the standard deviation σ that controls the width of the Gaussian filter that performs the initial smoothing operation. But unfortunately, the initial filtering does not completely solve the problem. A large σ also affects the thickness of contour lines.

The proposed method is different from the existing ones by the fact that it acts at the end (after the LoG filter and the thresholding operation), on the binary contour image. The method operation is presented in *figure 3*.

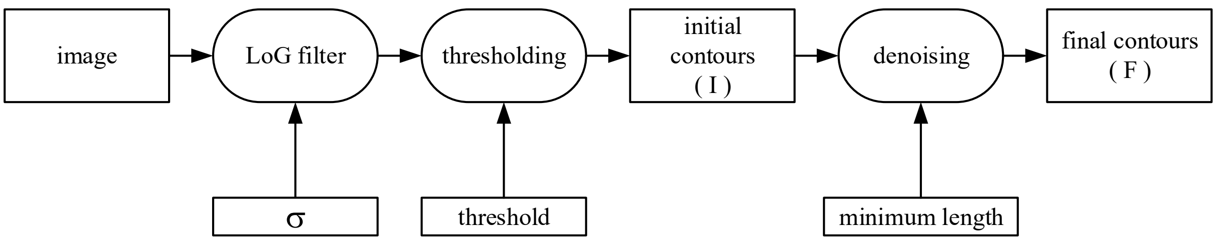


Fig. 3. Operation of the proposed method

The denoising block performs the following operations:

1. All the pixels of the final contours image (F) are initialized with white.
2. The initial contours image (I) is traversed along lines and columns. For each black pixel encountered, the following processing is performed:
   - 2.1. Initialize a list with the position of the pixel,
   - 2.2. Change pixel color to white,
   - 2.3. Add to the list all the black neighbors of the pixel, and their black neighbors, and the black neighbors of their neighbours, etc. Change the color of all pixels added to the list in white. The eight neighbors of the pixel on line y and column x are shown in *figure 4*.
   - 2.4. If the number of items in the list exceeds a certain threshold, then the color of all the pixels in image F that are located in the positions indicated by the list elements will be changed to black.

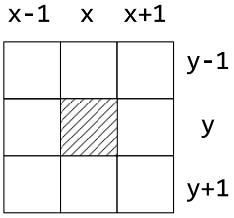
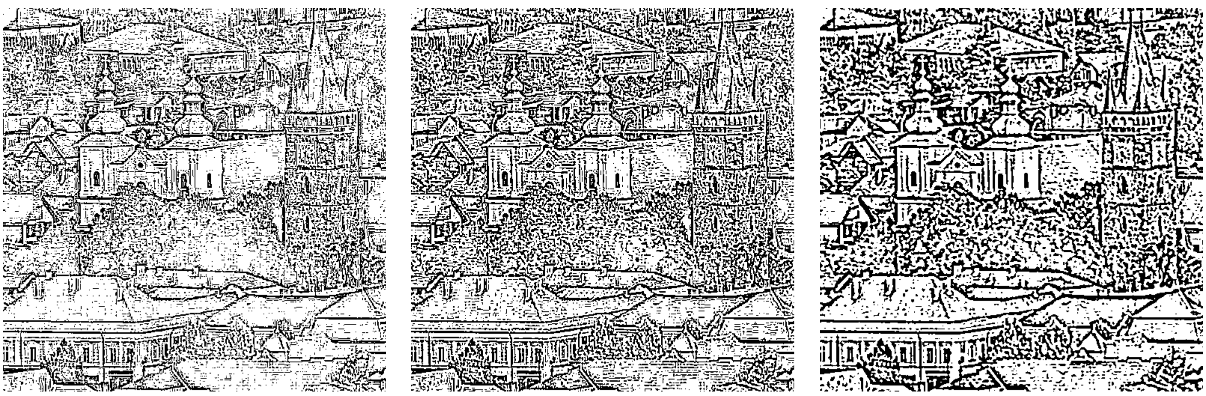


Fig. 4. A pixel neighbors locations

5. EXPERIMENTAL RESULTS

The image in *figure 1.a* was used for demonstrating the performance of the method. The images obtained after applying the LoG filter for different values of the standard deviation σ are shown in *figure 5*. To avoid the interruption of important contours, a relatively small threshold was chosen. This adds a lot of noise in the countours. The result of applying the contour cleaning algorithm is shown in *figure 6*.



a.) $\sigma = 0.6$ *b.)* $\sigma = 0.8$ *c.)* $\sigma = 1.4$

Fig. 5. LoG filter contours

a.) $\sigma = 0.6$ *b.)* $\sigma = 0.8$ *c.)* $\sigma = 1.4$

Fig. 6. Cleaned LoG filter contours

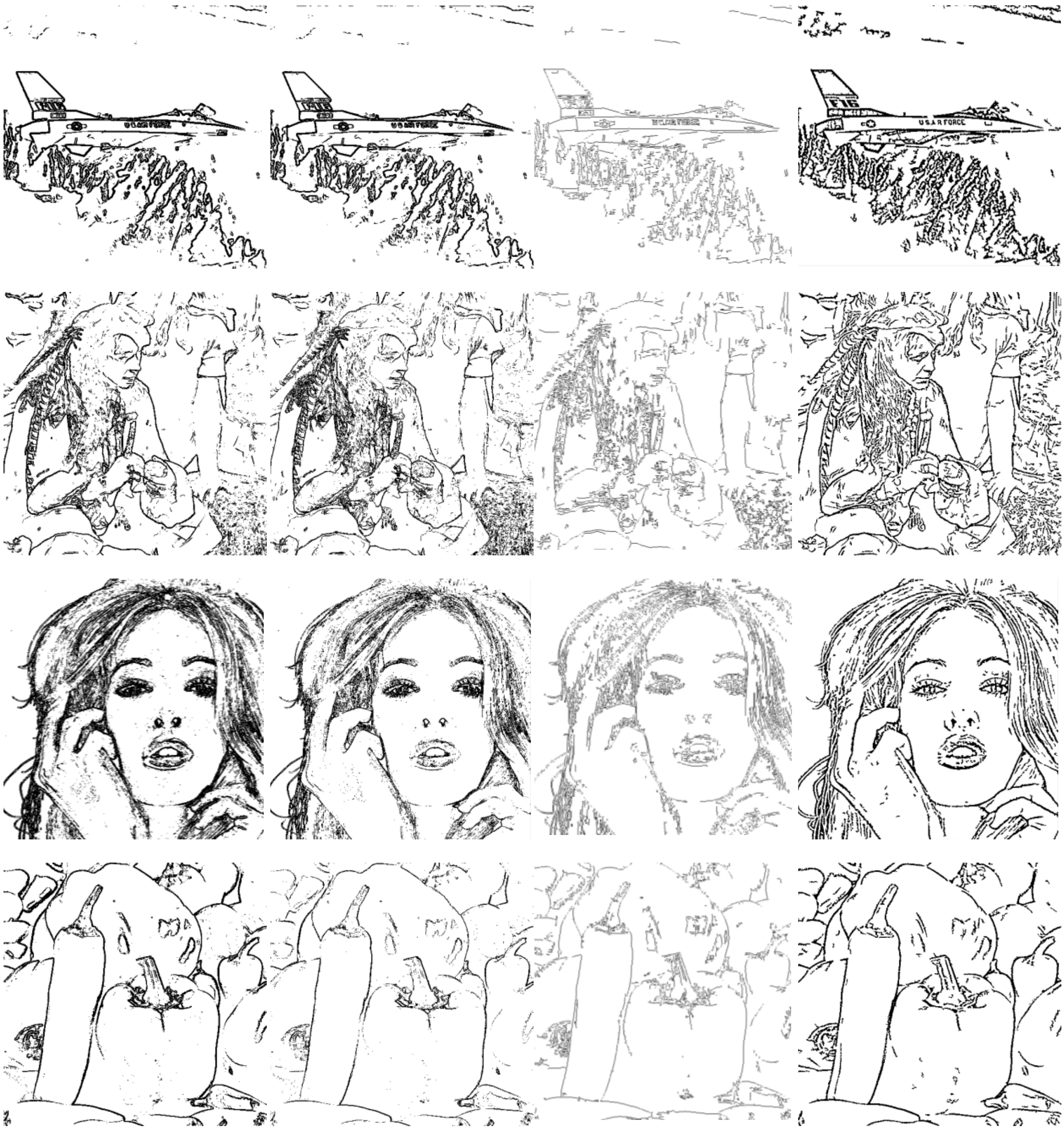
For verifying the properties of the method, it was applied on a set of "standard" images, which are commonly used in literature. The test images are presented in *figure 7*. They were taken from [20]. The contours of these images were drawn using the Sobel and Roberts filters, the Canny operator and the proposed method. The results are presented in *figures 8.1* and *8.2*.

a.) Cameraman *b.) Jetplane* *c.) Pirate* *d.) Blonde woman* *e.) Peppers*

Fig. 7. Set of test images

a.) Sobel filter *b.) Roberts filter* *c.) Canny operator* *d.) Proposed method*

Fig. 8.1. Comparison of the results



a.) Sobel filter *b.) Roberts filter* *c.) Canny operator* *d.) Proposed method*

Fig. 8.2. Comparison of the results

6. CONCLUSIONS

The method presented in this paper can be used to eliminate noise from contours generated with the LoG filter, adding an additional optimization parameter: minimum length (the minimum number of adjacent black pixels). This allows lowering the binarization threshold, to avoid interrupting important contours. The proposed algorithm is relatively simple and acts directly on the binary representation of the image contours. The contours obtained in this way are clean and firm. They are visually superior to those obtained using classical techniques.

REFERENCES

[1] Wilhelm J. Burger, Mark J. Burge, *Principles of Digital Image Processing*, Springer, 2013.

[2] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Prentice-Hall, 2002.

[3] Irwin Sobel, *History and definition of the so-called Sobel Operator*, https://www.researchgate.net/publication/239398674_An_Isotropic_3_3_Image_Gradient_Operator, (accessed 2017-05).

[4] John Canny, *A Computational Approach to Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, Issue 6, November 1986, pp. 679 – 698.

[5] D. Marr; E. Hildreth, *Theory of Edge Detection*, Proceedings of the Royal Society of London, Series B, Biological Sciences, Vol. 207, No. 1167, pp. 187-217, 1980.

[6] Dongqing Xu, Xiuyou Wang, Gang Sun, Huaimin Li, *Towards a novel image denoising method with edge-preserving sparse representation based on laplacian of B-spline edge-detection*, Multimedia Tools and Applications, 2015.

[7] Nasser Chaji, Hassan Ghassemian, *Texture-Gradient-Based Contour Detection*, Journal on Applied Signal Processing, pp. 1–8, 2006.

[8] Stephen M. Smith, J. Michael Brady, *SUSAN—A New Approach to Low Level Image Processing*, International Journal of Computer Vision Vol. 23, Issue 1, pp. 45–78, 1997.

[9] Bo Jiang, *Real-time multi-resolution edge detection with pattern analysis on graphics processing unit*, Journal of Real-Time Image Processing, pp. 1-29, 2014.

[10] S. Abdel-Khalek, Gamil Abdel-Azim, Z. A. Abo-Eleneen, A.-S. F. Obada, *New approach to image edge detection based on quantum entropy*, Journal of Russian Laser Research, Volume 37, Number 2, March, 2016.

[11] Sanjay Kumar, Rajiv Saxena, Kulbir Singh1, *Fractional Fourier Transform and Fractional-Order Calculus-Based Image Edge Detection*, Circuits, Systems, and Signal Processing, Vol. 36 Issue 4, April, pp. 1493-1513, 2017.

[12] P. Hinduja, K. Suresh, B. Ravi Kiran, *Edge Detection on an Image Using Ant Colony Optimization*, Proceedings of the Second International Conference on Computer and Communication Technologies, pp. 593-599, September 2015.

[13] Young-Hyun Baek, Oh-Sung Byun, Sung-Ryong Moon, Deok-Soo Baek, *Edge Detection in Digital Image Using Variable Template Operator*, International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pp. 195-200, 2005.

[14] Guang Deng, *Differentiation-Based Edge Detection Using the Logarithmic Image Processing Model*, Journal of Mathematical Imaging and Vision Vol. 8, pp. 161–180, 1998.

[15] Qu ZhiGuo, Gao YingHui, Wang Ping, Wang Peng, Tan XianSi, Shen ZhenKang, *Contour detection improved by frequency domain filtering of gradient image*, Science China Information Sciences, Volume 57, Issue 1, January, pp. 1–11, 2014.

[16] Giuseppe Papari, Patrizio Campisi, Nicolai Petkov, Alessandro Neri, *A Biologically Motivated*

Multiresolution Approach to Contour Detection, Journal on Advances in Signal Processing, Vol. 2007.

[17] Hong Cheng, Lin Chen, *A Holistic Approach for Efficient Contour Detection*, Journal of computer science and technology, Vol. 29, Issue 6, pp.1038-1047, November 2014.

[18] Tang Zhengjun, Song Jianshe, *A multi-scale morphological approach to SAR image edge detection*, Journal of electronics, Vol. 17, No. 3, July 2000.

[19] M. Yu. Khomyakov, *Comparative Evaluation of Linear Edge Detection Methods,* Pattern Recognition and Image Analysis, Vol. 22, Issue 2, June, pp. 291–302, 2012.

[20] http://www.imageprocessingplace.com/root_files_V3/image_databases.htm, (accessed 2017-07).