A New Approach to Edge Detection

Jan Juszczyk, *Silesian University of Technology*

Abstract

This paper describes an approach to the edge detection based on directional image searching for edges. This is a new idea of detecting edges because the algorithm discussed generates only some selected points of one particular edge. Using for calculation only some of the points surrounding the desired edge results in high computation speed of this system. The presented algorithm has also some specific skills useful in medical imaging such as automatic crossing the artefacts.

The article contains examples of how this system works on synthetic images as well as its application to semiautomatic edge detection in real CT and MR images.

1. Introduction

Edge detection is a very important part of many vision systems and often is used as image pre-processing before a more advanced analysis. It was Canny who first discussed a computational approach to edge detection and proved mathematically that his operator is the best to detect edge in general [1]. Since then many systems have been created on the basis of improvements of Canny's operator [2,3,4,5]. Currently lots of the systems specialize in edge detection only of a particular object such as iris [6]. There are also some biocybernetics systems using "ant colony" or "genetic" algorithms. However, these systems have some disadvantages. They try to find all the edges in an image, which is often unnecessary.

There is also another group of algorithms detecting the desired edges, yet requiring specific kinds of images.

A new system has been created involving a universal edge detection algorithm which can also be adapted to solve more specific problems. This algorithm has been developed on the basis of directional searching.

Searching begins from one point referred to as *rot*. It should be located near the desired edge. Obtaining this point is not irrelevant for this approach. *rot* is the rotation point for a group of points l_n and r_n. Points l_n and r_n are on the line tangent to the circle with center in *rot* and radius defined by vector $\underline{A}$. Points l_n, r_n are rotated by a given angle. The difference between points l_n and r_n is computed. The direction where this difference is maximum is chosen as the edge detection direction.

2. Algorithm

Let $\underline{A}$ be a nonzero 2D vector, and $\underline{R}$ and $\underline{L}$ are nonzero 2D vectors that are perpendicular to $\underline{A}$:

$$\underline{R} \circ \underline{A} = \underline{L} \circ \underline{A} = 0 \tag{1}$$

and

$$\underline{R} = -\underline{L}. \tag{2}$$

Let *rot* will be a point in the image plane. Point *a* is defined as the translation of point *rot* by vector $\underline{A}$:

$$a = T_{\underline{A}}(rot) \tag{3}$$

where:
$T_{\underline{A}}$ is a translation operator

Let l_i and r_i be points defined as the translation of point by vectors $\underline{i \cdot L}$ and $\underline{i \cdot R}$, respectively:

$$l_i = T_{\underline{i \cdot L}}(a) \tag{4}$$

$$r_i = T_{\underline{i \cdot R}}(a)\,. \tag{5}$$

Where $i = 1,2,3,\ldots,n$ and $j = 1,2,3,\ldots,m$ (Fig. 1.a.)

Moreover, α is defined as a directed angle and $Ro_{\alpha,rot}$ is rotation operator of point *rot* by angle α (Fig. 1.b.). This yields:

$$a = T_{\underline{A}}(rot)\,, \tag{6}$$

$$a_{0,\alpha} = Ro_{\alpha,rot}(a)\,, \tag{7}$$

$$l_i = T_{\underline{i \cdot L}}(a)\,, \tag{8}$$

$$l_{i,\alpha} = Ro_{\alpha,rot}\left(T_{\underline{i \cdot L}}(a)\right) \quad (9)$$

$$r_i = T_{\underline{i \cdot R}}(a) \quad (10)$$

$$r_{i,\alpha} = Ro_{\alpha,rot}\left(T_{\underline{i \cdot R}}(a)\right) \quad (11)$$

Function S is defined as:

$$S(rot, l_{1,\alpha}, l_{2,\alpha}, \ldots, l_{n,\alpha}, a_{0,\alpha}, r_{1,\alpha}, r_{2,\alpha}, \ldots, r_{n,\alpha}, \alpha) = \beta$$

where

$$\beta : \beta \in [-\gamma, \gamma] \text{ and}$$

$$\forall \alpha \in [-\gamma, \gamma] :$$

$$\left| \sum_{i=1}^{i=n} l_{i,\alpha} - \sum_{j=1}^{j=m} r_{j,\alpha} \right| \leq \left| \sum_{i=1}^{i=n} l_{i,\beta} - \sum_{j=1}^{j=m} r_{j,\beta} \right| \quad (12)$$

where γ is an angle whose value is fixed.

Let L_α and R_α be defined as:

$$L_\alpha = \sum_{i=1}^{i=n} l_{i,\alpha} \text{ and } R_\alpha = \sum_{j=1}^{j=m} r_{j,\alpha}, \quad (13)$$

therefore

$$S(L_\alpha, R_\alpha, \alpha) = \beta, \quad (14)$$

where

$$\beta : \beta \in [-\gamma, \gamma] \text{ and}$$

$$\forall \alpha \in [-\gamma, \gamma] :$$

$$|L_\alpha - R_\alpha| \leq |L_\beta - R_\beta| \quad (15)$$

The S function can be more advanced. For example, instead of computing $|L_\alpha - R_\alpha|$, differential edge detection can be used locally, only for $l_{i,a}$ and $r_{i,a}$. This might shorten the edge detection computation time.

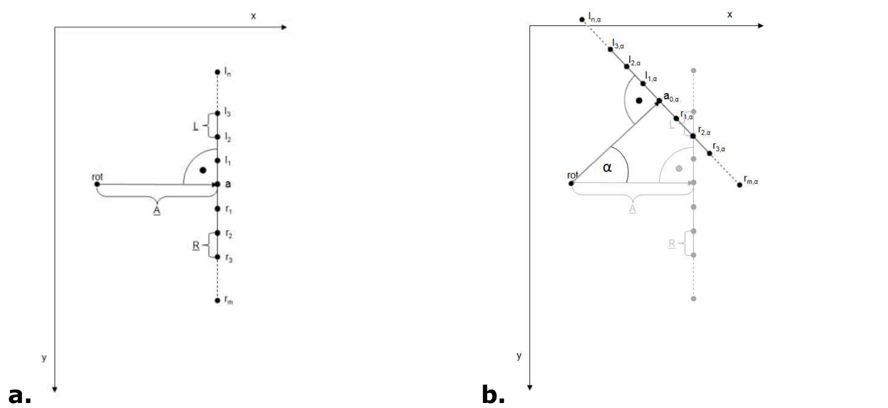


Fig.1. Function geometry. a. First step of defining measurement points, without rotation. b. After rotation.

This method returns only some points on the detected edge. The distance between these selected points equals $|\underline{K}|$; it should be mentioned that $\underline{K}$ is a rotation of $\underline{A}$ by angle β, however for each iteration t the value of β differs:

$$\underline{K}_t = Ro_{\beta_t, rot_t}(\underline{A}_t), \quad (16)$$

Then for each iteration the function S becomes:

$$S(\alpha)_{t+1} = T_{\underline{K}}\left(Ro_{\beta,rot}(S(\alpha)_t)\right). \quad (17)$$

Furthermore, this algorithm returns for each iteration a new rotation point rot_t, which becomes point a_{t-1}. Another important thing is that searching in the next iteration progresses only in one direction determined by angle β calculated in the previous iteration. Therefore the algorithm detects edges only in the relative angle $[-\gamma,\gamma]$ (Fig.2.), because the process cannot go backwards and turns only with the minimum turning radius. (Fig. 5.b.)

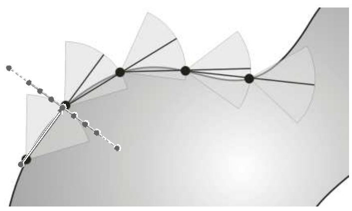


Fig.2. Moving on the image

If the searched edge turns too sharply or simply finishes, function S will reach a homogeneous field, which means that points on the left and right sides of the a_t point have similar values, no matter what angle a we choose. Thus the stop condition is based on the standard deviation definition; let *dist* be a difference in standard deviations of points on both side of point a_t:

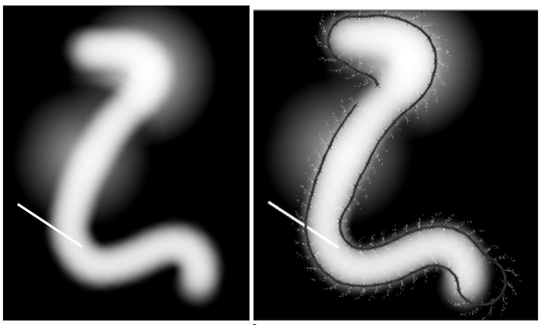
$$distd = \left| \sqrt{\frac{1}{n-1}\sum_{i=1}^{i=n}\left(l_{i,\beta} - \overline{l_\beta}\right)^2} - \sqrt{\frac{1}{m-1}\sum_{j=1}^{j=m}\left(r_{j,\beta} - \overline{r_\beta}\right)^2} \right|. \quad (17)$$

If *distd* is smaller than a given threshold then detecting will stop automatically.

4. Results

Synthetic Data

First, the results of edge detection on synthetic data will be shown. A JPEG grayscale image (Fig. 3.a.), 2048x2048x8bit in size, was used as a sample. Large images of this kind allow the tonal transition to be very smooth and there is lots of data to process. Synthetic image lets us see, better than clinical medical imaging data, how the algorithm works in detail. Fig. 3.b. shows a synthetic image with edges detected. Which one of all the edges in this image will be detected depends on the chosen start point and the given direction of the $\underline{A}$ vector. It means the user can choose which edge is to be detected.



a. b.

Fig.3. a. Synthetic image. b. Synthetic image with an edge detected.

The number of points which has to be calculated in this algorithm is significantly lower than the whole image size (Fig. 4.). Thanks to this solution the computations progress much faster. It is important because in medical imagining one examination has more than only one image (often more than one hundred), so the amount of data to the analyze increases. By using this algorithm the speed of edge detection depends on the length of the desired edge and also on how precise the result should be. In general the detection speed is defined by the object size and does not depend directly on the image size.

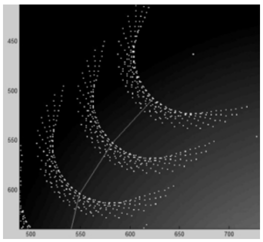
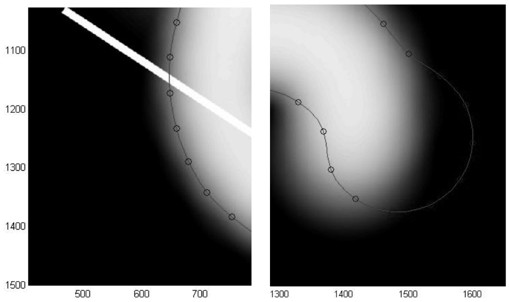


Fig.4. Three iterations of edge detecting with measurement points marked (white points).

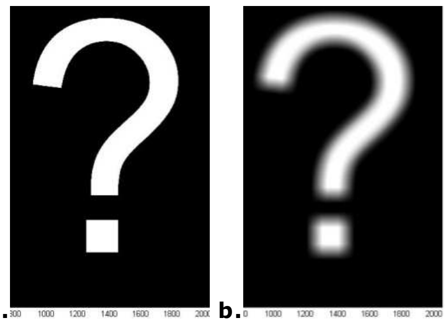
Besides the speed another advantage of this algorithm is its "force ability". This system can cross other objects smaller than the detected object. (Fig. 5.a.). It is very important for CT images, particularly where artifacts from endoprostheses or clothes may occure. The next attribute is its ability to return to the edge after a too double sharp (Fig. 5.b.). If the edge turns too sharply but doesn't end abruptly, the algorithm will try to come back to the correct edge.



a. b.

Fig.5. a. Crossing an obstacle b. Minimum turning radius and coming back to the edge.

To test the algorithm accuracy a TIFF 2048x2048x16bit grayscale image was used. Original image (Fig. 6.a.) was smoothed with the Gaussian filter (Fig. 6.b.). The algorithm was tested on smoothed images.



a. b.

Fig.6. a. Original TIFF image b. Smoothed image

The maximum error of the segmented area on the basis of the detected edge is less than 8%. The maximum difference between the detected edge and the real one is 14 pixels, the object width being 16 pixels, yielding the maximum relative error of 8.5%.

Fig. 7. a. Segmentation result in the original image b. Segmentation in the smoothed image c. The absolute difference of a and b

4.1 Summary and Future Works

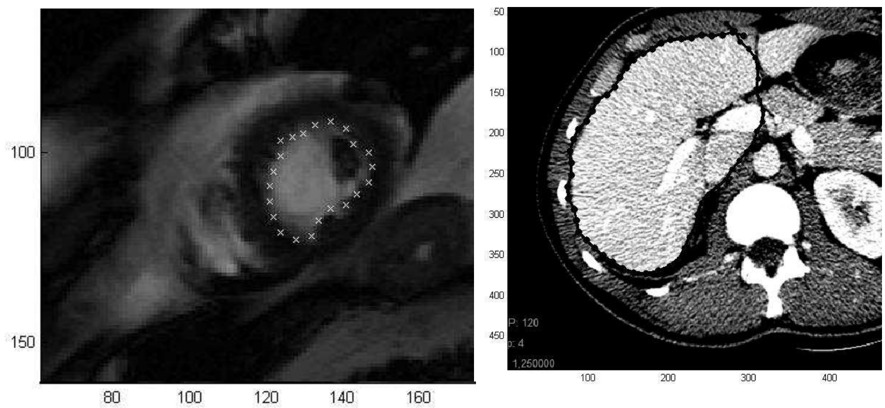
All the edges could be found if only their number in an image is known and the start points are set. The algorithm returns a series of points and therefore is suitable for 3D modelling. The computations speed is the next argument to use the results obtained as input data for real-time modelling.

To test the algorithm on real data MR and CT images were chosen. CT images come from an abdominal CT scan with contrast agent, 512x512x16bit in size. MR images come from 4D, heart MRI 256x256x16bit in size.

The task in CT images is to detect liver edges with only one start point (Fig.6.a)). In MR heart images blood in the left ventricle is detected, with only one start point too (Fig.6.b)). The quality of the edges detected in both cases is sufficient eg. for an implementation of 3D models of these structures.

The idea of the suggested edge detection method has been proved, however many things can be improved. On the basis of the idea presented in [3] the algorithm will be improved to detect edges with a lower range, which means starting from one point two or more edges will be found, first the main edge and then the weaker one.

At this time the algorithm incorporates many parameters which are set by the user. In the next generation of this system the parameters value will be chosen automatically.



a. b.

Fig.6. a. Heart MRI, the edge marked by crosses b. Liver CT, the edge marked by a dotted line.

References

1. Canny J.: *A Computational Approach to Edge Detection*, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 8, No. 6 (1986)
2. Rachid Deriche *Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector*, International Journal of Computer Vision (1987)
3. Ding L., Goshtasby A.: *On the Canny Edge Detector,* Pattern Recognition (34) 721-725 (2001)
4. Hou J., Ye J., Li S.:*Application of Canny Combining and Wavelet Transform in the Bound of Step-Structure Edge Detection*, Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition (2007)
5. Xiangjian He, Wenjing Jia and Qiang Wu: *An Approach of Canny Edge Detection with VirtualHexagonal Image Structure,* 10th Intl. Conf. on Control, Automation, Robotics and Vision (2008)
6. Jing Huang, XingeYou, YuanYanTang, LiangDu, YuanYuan: *A Novel Iris Segmentation Using Radial-Suppression Edge Detection* Signal Processing, Elsevier (2009)
7. Jeffrey J. Tabor, Howard M. Salis, Zachary Booth Simpson, Aaron A. Chevalier, Anselm Levskaya, Edward M. Marcotte, Christopher A. Voigt, and Andrew D. Ellington: *A Synthetic Genetic EdgeDetection Program*, Cell (2009)
8. Yonghua Wu,Yihua, Wuhu Lei, Nanxiang Zhao, and Tao Huan: *Edge Detection of Laser Range Image Based on a Fast Adaptive Ant Colony Algorithm*, Advances in Swarm Intelligence, Springer (2010)

Author:
Jan M. Juszczyk, MSC.
Silesian University of Technology
ul. Akademicka 16, 44-100 Gliwice
tel. (32) 237 17 35, email: jan.juszczyk@polsl.pl