A Simple and Effective Edge Detector

C. Cafforio, E. Di Sciascio, C. Guaragnella and G. Piscitelli

Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari
Via E.Orabona, 4 I-70125 Bari, Italy

Abstract. Nonlinear filtering based on a two concentric circular windows operator is introduced as a simple and effective way to find edges in image processing. The dual windows edge detector can operate with no fixed threshold, is isotropic, i.e. its response does not depend on edge orientation and has good noise immunity in the vicinity of edges. Computational requirements are limited and experimental results are good enough.

1 Introduction

Many image processing applications require edge detection. Edges are of extreme importance, as they convey essential information in a picture. Accurate edge detection is necessary for a number of image analysis and recognition techniques. A lot of research has been done on this topic in the past, yet a completely satisfactory algorithm is still to be devised.

An efficient edge detection procedure should comply with some basic requirements: edges should be found with a low probability of false detection due to noise; edges should not be misplaced; algorithms should perform similarly with any image; procedures should lead to an efficient implementation.

Ideally edges are steep transitions between smoothly varying luminance areas. In real scenes transitions are not so steep, thus their extraction requires careful processing. Several edge detection techniques exist and a number of modifications to basic algorithms have been reported in the literature. Usually these algorithms use first or second derivatives of image luminance and are forced to adopt some regularization techniques to control the effects of noise. Marr and Hildreth [1] proposed an algorithm that finds edges at zero-crossing of the image Laplacian, and several modification to their basic approach have been reported. This algorithm guarantees that continuous contours are extracted. The Canny algorithm [2] finds edges at peaks of the first difference in the local gradient direction. A variation on the basic algorithm, due to Boie-Cox [3], uses the maximum first difference and localizes edges with the aid of a directional second difference. A number of other modifications to the approach of Canny also exist [4-8]. Edge detection algorithms based on non linear filters have also been presented in the literature [9]. More recently, techniques have been proposed that characterize edge detection as a fuzzy reasoning problem [10]. There is much literature on the subject and the above references are only a few, not exhaustive, examples.

Most techniques heavily rely on low-pass filtering and the use of thresholds.

Although this is a sure way to drastically reduce noise effects, it leads, most likely, to discontinuities in the detected edges and to contour misplacement. To partially avoid the negative effects of filtering, adaptive directional filters that smooth the image only in the direction orthogonal to local image gradient have been proposed. These filtering schemes are effective, but lead to an increase in the algorithmic complexity without the possibility to assure precise edges localization.

In this paper we present a simple operator that uses two concentric windows. The operator does not require fixed thresholds, is isotropic, i.e. its response is almost independent on contours orientation, and guarantees a good edge localization. Computational requirements are limited.

The proposed algorithm originates from a previous work where no known edge extraction algorithm was accurate enough to allow an efficient extrapolation of high frequency terms from a low passed image [12].

Next section presents the proposed edge detection approach. In section III the obtained results are shown and discussed. The arguments that call for further research are outlined in the final section.

2 Nonlinear Edge Detection

Let X be a gray level image and $x(i,j)$, $i=1,\dots,M$, $j=1,\dots,N$ its samples. It is reasonable to consider contours placed exactly half way along the slopes of transition zones between almost uniform luminance areas. This is a simple task in one dimension; in two dimensions the need for dealing with the orientation of the transition zone makes things more difficult. It is much more so because of the rectangular sampling grid. To correctly locate a contour, it is necessary to estimate the width of the transition zone and to track the position within this zone. This is not achievable with just one local measure.

The proposed algorithm uses two concentric circular windows W_1 and W_2 (as much circular as the rectangular sampling grid allows). The window shape is not an essential feature for our approach: a square window gives similar results, but the response of the operator is no longer isotropic.

Fig. 1. a) A sharp profile and the extracted contour; b) corresponding α_1 and α_2 plots.

$$L_{W_2} = 7, L_{W_1} = 3 \text{ pixels.}$$

Fig. 1 shows the edge detection operator applied on sharp profile. It can be noticed that whatever the external window size L_{W_2} the detected edge width is determined by the inner window size L_{W_1} , in fact it is equal to $(L_{W_1} - 1)$.

The maximum and minimum pixel values are measured within both considered windows:

$$\begin{aligned} \min &= \min_{i,j \in W_1} x(i,j) & MIN &= \min_{i,j \in W_2} x(i,j) \\ \max &= \max_{i,j \in W_1} x(i,j) & MAX &= \max_{i,j \in W_2} x(i,j) \end{aligned} \quad (1)$$

and the differences between maxima and minima in respective windows are then computed:

$$\alpha_1 = MAX - \max \quad \alpha_2 = \min - MIN \quad (2)$$

A contour is considered present wherever $\alpha_1 = \alpha_2 \neq 0$, i.e. when the luminance ranges in W_1 and W_2 coincide.

Fig. 2. A linear profile and the operator response for different outer window sizes: a) $L_{W_2} = 7$, $L_{W_1} = 3$ pixels; b) corresponding α_1 and α_2 plots; c) $L_{W_2} = 19$, $L_{W_1} = 3$ pixels; d) corresponding α_1 and α_2 plots.

The rationale of this procedure is that the larger window identifies the transition zone, while the smaller window, placed exactly at the center of the larger one, tracks the half height point. If the external window is larger than the transition zone, the contour is exactly located. If the transition zone is larger, the condition $\alpha_1 = \alpha_2$ is met, assuming the slope is constant, until one of the extrema of the outer window leaves the discontinuity area.

To clarify how the operator works, let us consider its application in the one dimensional case for two further ideal signals: a linear and a cosine like degrading profiles.

For a linear degrading profile the detected edge width may result thick (see figs. 2.a and 2.b) when a small outer window is used.

A single pixel edge can be obtained if the whole constant slope luminance transition is included within the outer window. This behavior clearly appears in figs. 2.c and 2.d.

Fig. 3. A cosine profile and the detected edge $L_{W_2} = 11, L_{W_1} = 3$ pixels

Fig. 4. A cosine profile with added Gaussian noise ($\sigma = 0.05$, PSNR = 26 dB): a) $L_{W_2} = 11, L_{W_1} = 3$ pixels; b) $L_{W_2} = 31, L_{W_1} = 3$ pixels; c) $L_{W_2} = 41, L_{W_1} = 3$ pixels.

In fig. 3 a cosine degrading 15-pixels wide profile is considered. With this type of signal a single pixel edge is obtainable even with small outer window size. This is because the luminance profile has not an exactly constant slope. In fact the difference $\alpha_1 - \alpha_2$ zeroes only exactly in the middle of the luminance transition, where the contour should be placed.

Condition $\alpha_1 = \alpha_2$ allows edge detection only conceptually. As a matter of fact, in a real environment, a thresholding strategy has to be adopted.

The difficulty lies in the image sampling: it is extremely improbable that the difference $\alpha_1 - \alpha_2$ exactly zeroes on a sampling point. The change in the sign of $\alpha_1 - \alpha_2$ is a much more effective indicator of the presence of a contour. If a two dimensional search of sign changes in $\alpha_1 - \alpha_2$, which can allow a fractional pixel

positioning of contours, is considered a complication, a simplified procedure can be used instead.

Fig. 5. a) Original test image “building”. b) Transition areas in “building”. $\alpha_1 > \alpha_2$ black; $\alpha_1 < \alpha_2$ gray, $\alpha_1 = \alpha_2$ white.

The equality of the two values α_1 and α_2 can be considered verified when their absolute difference is smaller than the estimated luminance change between contiguous pixels or a fraction thereof:

$$\varepsilon = \frac{(max - min)}{W_1} \quad (3)$$

This is a thresholding operation and, though adaptive, it has drawbacks. In fact, it is no longer possible to assure that the extracted contours are closed. In some application this is unacceptable, but discontinuities in detected contours are always an unpleasant effect to consider. On the other hand, continuous contours do not necessarily mean physically meaningful contours.

In fig. 4 the previous cosine degrading profile is considered when an added Gaussian noise is introduced and the adaptive threshold is implemented. By increasing the outer window size it can be noticed that, while the edge position remains unchanged, false edges due to noise are drastically reduced in the area around the true edge position.

The proposed operator can be considered an application to contour detection of the principles of morphological filtering. However, it is possible to give a simpler interpretation of its operation.

The external window is used to circumscribe the area in which the contour is to be looked for, and it determines the local luminance change. A luminance threshold put exactly half-way on the measured luminance change would reproduce ancient techniques used in image binarization. Here a slightly more sophisticated approach is proposed looking for the position where the luminance change within the inner window is centered with respect to that in the outer one. This provides a little (the only one) rejection of the noise inevitably superimposed on the image.

Fig. 6. a) Edge detector output with $L_{W_1}=3$ and $L_{W_2}=11$. b) Edge detector output with $L_{W_1}=3$ and $L_{W_2}=61$.

The noise rejection is more effective with a larger inner window (but this has other drawbacks, as already pointed out) and with a higher local gradient. The noise rejection is maximum when a contour is present: this results in better contour detection and localization. However, the algorithm is completely ineffective in flat luminance areas, where noise can detect a lot of false contours. A threshold on *MAX-MIN* or *max-min* is in some way unavoidable.

The size of the external window, i.e. W_2 , affects the sensitivity of the operator: small external windows will result in extremely accurate contours, but also in a high sensitivity to noise. Larger windows will extend the effect of high contrast contours over a larger area, preventing smaller contrast contour from being detected near higher contrast ones (see fig. 4). As a matter of fact, a larger window provides the large support that is needed to suppress noise, while the inner window has the small support required for edge localization [11].

An effective way to combine the two windowing strategies is possible adopting a simple tracking algorithm. Let us suppose to apply the edge detection algorithm to the same image, with different window sizes. Let image Y_1 contain the contours obtained with the larger external window and image Y_2 contain those obtained with the smaller external window. The algorithm tracks edge samples in Y_1 and, when a break is observed, it checks image Y_2 for contour points that allow to proceed in edge following. This is done with a window 5 pixels wide having at least 3 pixels in Y_1 already marked as edge samples. When such a condition is met other edge samples in Y_2 are marked as edge pixels also in the new edge image. The algorithm obviously avoids to consider isolated edge pixels, i.e. pixels having no edge neighbors within the window. It is clear that the current implementation of the tracking algorithm is extremely simple, although effective in reducing contours discontinuities.

Similar or better results could be obtained with more refined techniques that could operate without the two passes with different window sizes. A dynamically varying external window size could be a better choice for a single pass algorithm.

Fig. 7. a) Edge image obtained with the application of the tracking algorithm on the previous two edge images. b) Edges detected with $L_{W_1}=3$, $L_{W_2}=5$ and $\varepsilon > 5$ for test image building.

3 Experimental Results

All experiments were based on 720×560 pixels gray level images quantized with 8 bits. No filtering whatsoever was used to reduce noise effects, in order to check the noise sensitivity of the algorithm.

Fig. 5.a shows the original test image named “building”. Figure 5.b shows the image obtained considering the difference $(\alpha_1 - \alpha_2)$ in the test image. The image is obtained using $L_{W_1}=3$ and $L_{W_2}=11$, respectively. It is evident that where transitions take place, i.e. where the difference $(\alpha_1 - \alpha_2)$ changes sign, contours should be placed.

The application of the nonlinear operator on the test image produces edge images shown in figs. 6.a and 6.b. The first one is obtained with window sizes $W_1=3$ and $W_2=10$, the second one is obtained with $W_1=3$ and $W_2=61$. The difference is noticeable; figure 6.b has all higher energy contours detected, but some of them are incomplete, while figure 6.a, though having all edges detected, has a lot of noise. The application of the tracking algorithm on images in figures 6.a and 6.b produces the image in figure 7.a, where various contours become closed. Edges are correctly positioned even after the modification of the window size and this is what allows a correct edge tracking. The proposed algorithm is able to detect multiple converging edges, as no assumption on edge direction is made.

Other experiments have been performed introducing a further fixed threshold on ε to reduce the noise effect. The introduction of this threshold allows to reduce used window sizes also producing accurate but discontinuous contours. Fig. 7.b has been obtained using $L_{W_1}=3$, $L_{W_2}=5$ and a fixed threshold $\varepsilon > 5$.

4 Conclusions

Edge detection is a crucial stage in almost any image processing application. This work offers a contribution to the design of simple and reliable edge detectors. The basic nonlinear operator we propose uses two concentric windows; the outer one identifies a transition zone, while the inner one tracks the half point that can be assumed to be an edge sample. A noteworthy advantage of this method is the edge detector requires no fixed threshold; an adaptive threshold, proportional to the luminance change in the inner window has been introduced.

Experimental results obtained with various test images seem to support the theoretical background. Further work is needed to make the algorithm more robust in low contrast areas. Adaptive window size and a fuzzy logic approach to thresholding will be considered.

References

1. D. Marr, E. C. Hildreth, "Theory of edge detection", Proc. Royal Soc. London, B, vol. 207, pp. 187-217, 1980.
2. Canny, "A computational approach to edge detection", IEEE Trans. on PAMI, vol. 8, pp. 679-698, Nov. 1986.
3. A. Boie, I. J. Cox, "Two dimensional optimum edge recognition using matched and Wiener filters for machine vision", Proc. of Int. Conf. On Computer Vision, London, pp. 450-456, 1987.
4. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detection, Int. Journal of Computer Vision, pp. 167-187, 1987.
5. Petrou, J. Kittler, "Optimal edge detectors for ramp edges", IEEE Trans. on PAMI, vol. 13, pp. 1154-1171, Nov. 1991.
6. Sarkar, K. L. Boyer, "On optimal infinite impulse response edge detection filters", IEEE Trans. on PAMI, vol. 13, pp. 483-491, May, 1991.
7. A. Spacek, "Edge detection and motion detection", Image and vision computing, vol. 4, pp. 43-56, Feb. 1986.
8. Ziou, "Line detection using an optimal IIR filter", Pattern Recognition, vol 24, pp. 465-478, 1991.
9. I. Pitas, A. N. Venetianopoulos, "Edge detectors based on nonlinear filters", IEEE Trans. on PAMI, vol. 8, NO. 4, pp. 538-550, 1986.
10. Law, H. Itoh, H. Seki, "Image filtering, edge detection, and edge tracing using fuzzy reasoning, IEEE Trans. on PAMI, vol. 18, No. 5, pp. 481-491, 1996.
11. Gokmen, C. Li, "Edge detection and surface reconstruction using refined regularization", IEEE Trans. on PAMI, vol. 15, No. 5, pp.492-499, 1993.
12. C. Cafforio, E. Di Sciascio, C. Guaragnella, "Spectral extrapolation in sub-band coding", Proceedings of IEEE Workshop on Digital Signal Processing, pp. 13-16, Loen (Norway), 1996.