PERGAMON

Pattern Recognition 34 (2001) 721–725

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

On the Canny edge detector

Lijun Ding, Ardeshir Goshtasby*

Department of Computer Science and Engineering, Wright State University, 303 Russ Engineering Center, Dayton, OH 45435, USA

Received 14 May 1999; received in revised form 26 January 2000; accepted 26 January 2000

Abstract

The Canny edge detector is widely used in computer vision to locate sharp intensity changes and to find object boundaries in an image. The Canny edge detector classifies a pixel as an edge if the gradient magnitude of the pixel is larger than those of pixels at both its sides in the direction of maximum intensity change. In this paper we will show that defining edges in this manner causes some obvious edges to be missed. We will also show how to revise the Canny edge detector to improve its detection accuracy.

Keywords: Edge detection; Canny edge detector; Zero-crossing edges; Missing edges; Edge accuracy

1. Introduction

Edge detection is one of the fundamental operations in computer vision with numerous approaches to it. In an historical paper, Marr and Hildreth [1] introduced the theory of edge detection and described a method for determining the edges using the zero-crossings of the Laplacian of Gaussian of an image. Haralick [2] determined edges by fitting polynomial functions to local image intensities and finding the zero-crossings of the second directional derivative of the functions. Canny [3] determined edges by an optimization process and proposed an approximation to the optimal detector as the maxima of gradient magnitude of a Gaussian-smoothed image. Clark [4] and Ulupinar and Medioni [5] independently found a method to filter out false edges obtained by the Laplacian of Gaussian operator. Bergholm [6] introduced the concept of edge focusing and tracked edges from coarse to fine to mask weak and noisy edges. A curve-fitting approach to edge detection was proposed by Goshtasby and Shyu [7] in which edge contours were represented by parametric curves that fitted to high-gradient image pixels with weights proportional to the gradient magnitudes of the pixels. Recent advances in edge detection include a method by Elder and Zucker [8] to determine edges at multitudes of scales, and an adaptive smoothing method by Li [9] to remove noisy details in an image without blurring the edges. Many other edge detection techniques have been proposed. For a survey and comparison of the edge detectors, the reader is referred to the paper by Heath et al. [10].

Among the edge detection methods proposed so far, the Canny edge detector is the most rigorously defined operator and is widely used. The popularity of the Canny edge detector can be attributed to its optimality according to the three criteria of good detection, good localization, and single response to an edge. It also has a rather simple approximate implementation, which is the subject of this paper. We will show examples where this approximate implementation misses some obvious edges. We will also show how to revise the Canny edge detector to improve its detection accuracy.

2. A revised Canny edge detector

Although the optimization process described by Canny rests on solid grounds, its simple approximation, which almost all implementations are based on, has some

* Corresponding author. Tel.: + 1-937-775-5170; fax: + 1-937-775-5133.

E-mail address: agoshtas@cs.wright.edu (A. Goshtasby).

0031-3203/01/$20.00
PII: S0031-3203(00)00023-6

problems. A typical implementation of the Canny edge detector [11,12] follows the steps below.

1. Smooth the image with an appropriate Gaussian filter to reduce desired image details.
2. Determine gradient magnitude and gradient direction[1] at each pixel.
3. If the gradient magnitude at a pixel is larger than those at its two neighbors in the gradient direction, mark the pixel as an edge. Otherwise, mark the pixel as the background.
4. Remove the weak edges by hysteresis thresholding.

We have narrowed down the source of inaccuracies to Step 3 of the algorithm. This is demonstrated in an example in Fig. 1(a). Fig. 1(a) depicts an image containing four homogeneous regions. A smoothed version of this image obtained by convolving it with a Gaussian of standard deviation 11 pixels is shown in Fig. 1(b). Fig. 1(c) depicts the gradient magnitudes of Fig. 1(b). We can clearly see that information about the edge contours is present in the gradient image as evidenced by the bright horizontal and vertical stripes passing through the image center. An edge detector that uses this gradient image should be able to determine four edge contours that join at the image center and separate the homogeneous regions from each other.

If we determine the edges in Fig. 1(a) using the Canny edge detector with a Gaussian smoother of standard deviation 11 pixels, we obtain the edges shown in Fig. 1(d). We see that edges near the image center are missed. The Canny edge detector cannot detect branching edges. From Fig. 1(d) we see that some other edges are missed also. By closely examining the gradient magnitudes and directions of edges that are missed, we observe that although gradient magnitudes at the missing edges are larger than those of pixels adjacent to them, such maxima are not in the gradient direction. Since gradient maxima in an image form ridges, and since at the ridge points the slopes are either zero or very small, the directions of the slopes cannot be accurately determined. Therefore, computed directions of slopes along a ridge contour may not be normal to the contour. When the direction of a gradient slope at a ridge point is not normal to the ridge contour, the ridge point is not picked as an edge; therefore, the edge is missed.

To alleviate this problem, we propose revising Step 3 of the Canny edge detector as follows: Let us call edges obtained by the Canny edge detector the *major edges*. We will then mark a pixel as a *minor edge* if its gradient magnitude is larger than those adjacent to it and at opposing sides but not necessarily in the gradient direction.

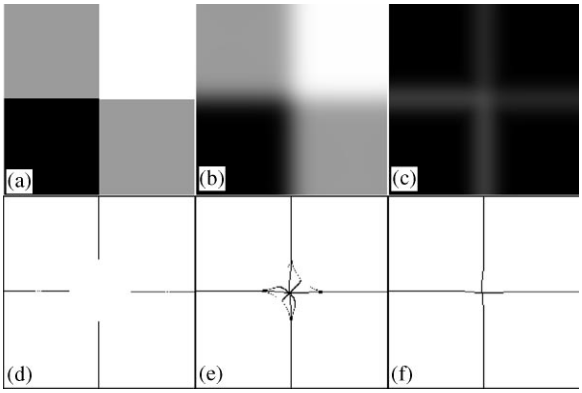

Fig. 1. (a) An image with four well-defined homogeneous regions. (b) Image after smoothing with a Gaussian kernel of standard deviation 11 pixels. (c) Gradient magnitudes of (b). (d) Edges of (a) determined by the Canny edge detector with a Gaussian smoother of standard deviation 11 pixels. These are the major edges. (e) The minor edges. (f) Edges obtained by the revised Canny edge detector.

The minor edges obtained from the image of Fig. 1(c) are shown in Fig. 1(e). We see that major edges are a subset of minor edges. We also see that although some of the minor edges correspond to the true edges, some other minor edges represent false edges. We will need to separate the true minor edges from the false ones. To achieve this, we trace the minor edge contours and keep only the parts of the contours that connect the major edges. To avoid formation of false loops, the minor edge contours are first partitioned at the branch points and branches that do not contain any major edges are removed. Then the remaining contour segments are processed, and portions delimited by major edges are kept.

Combining the major edges and parts of the minor edge contours that connect the major edges, we obtain the result shown in Fig. 1(f). This is the final result of the revised Canny edge detector. Note that the edges shown in Fig. 1(f) represent the edges of Fig. 1(b) and not the edges of Fig. 1(a). When smoothing an image, the Gaussians centered at brighter pixels become taller than the Gaussians centered at darker pixels, and as a Gaussian smoother is made wider, locally maximum intensity changes move away from the brighter regions and toward the darker regions. This phenomenon is observed in Fig. 1(f). As the standard deviation of the Gaussian smoother is increased, edges shift from brighter regions to darker ones.

A second example is shown in Fig. 2. Suppose Fig. 2(a) depicts the gradient magnitudes of an image whose edges have to be determined. This image could represent the gradients of an image containing a bright homogeneous object in a dark homogeneous background. Suppose the objective is to determine the boundary of the object by

[1] Gradient direction is assumed to be the direction of maximum intensity change.

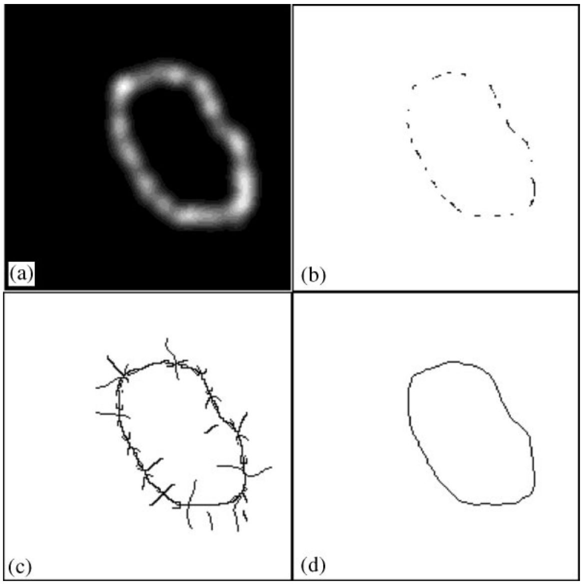

Fig. 2. (a) A synthetically generated gradient image. (b) Edges obtained by the nonmaxima suppression of Canny. (c) The minor edges. (d) Edges obtained by the revised Canny edge detector.

the Canny edge detector. As can be visually observed in Fig. 2(a), the information about the object boundary is present in the image. By applying the nonmaxima suppression of Canny to the image of Fig. 2(a), we obtain the edges shown in Fig. 2(b). Many of the obvious edges are missed. If we determine the minor edges of Fig. 2(a), we obtain the edge contours shown in Fig. 2(c). By keeping only those minor edges that connect the major edges, we obtain the edge contour shown in Fig. 2(d). This is the result of the revised Canny edge detector.

To summarize, our revised Canny edge detector follows the steps below.

1. Smooth the given image with an appropriate Gaussian to reduce desired amount of image details and noise.
2. Determine the gradient magnitude and gradient direction at each pixel.
3. If the gradient magnitude at a pixel is larger than those at its two neighbors in the gradient direction, mark the pixel as a major edge. If the gradient magnitude at the pixel is larger than those pixels adjacent to it in any direction, mark the pixel as a minor edge. Otherwise, mark the pixel as the background.
4. Partition the minor edge contours at the branch points.
5. Remove all branches that do not contain a major edge. Then rename as major edges the portions of minor edge contours that are delimited by major edges. Combine newly obtained major edges with previously obtained major edges.
6. Remove from among the combined major edges those that are sufficiently weak by hysteresis thresholding.

3. Results

To compare results obtained by the original Canny edge detector and the revised Canny edge detector, a number of experiments were carried out. In these experiments, the same Gaussian smoother was used in both edge detectors. Also, the same hysteresis thresholds were used in both edge detectors. Fig. 3(a) shows a synthetic image of homogeneous blocks, and Fig. 4(a) shows an image of printed text. Thirty-two homogeneous regions are present in Fig. 3(a). The boundaries of the regions are clearly defined, and because the image is not noisy, we expect an edge detector to obtain horizontal and vertical lines representing the edge contours in the image. Edges obtained by the Canny edge detector with a Gaussian of standard deviation 2 pixels is shown in Fig. 2(b). As expected, edges near the branch points are missed. It is interesting to note that discontinuities in the edge contours are vertical and not horizontal. Since gradients are stronger vertically, it is desired that the detected edge contours be horizontal. The Canny edge detector has done a good job on that; however, it has disconnected the vertical edge contours.

The minor edges of Fig. 3(a) are shown in Fig. 3(c). Not only are the branch points obtained, but many spurious

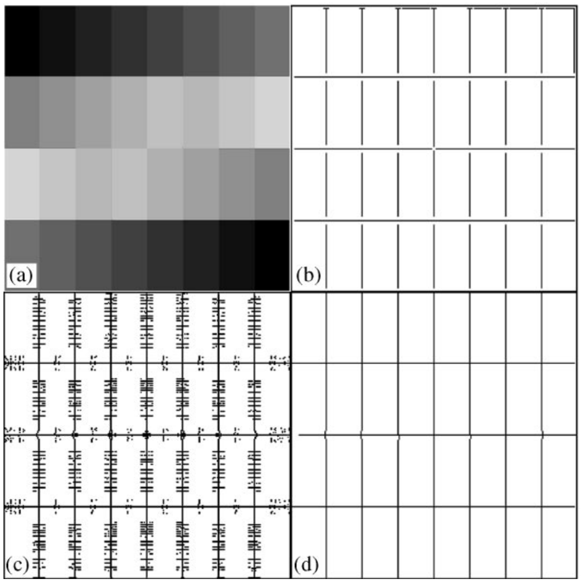

Fig. 3. (a) An image with 32 homogeneous regions. (b) Edges determined by the Canny edge detector. (c) The minor edges. (d) Edges determined by the revised Canny edge detector.

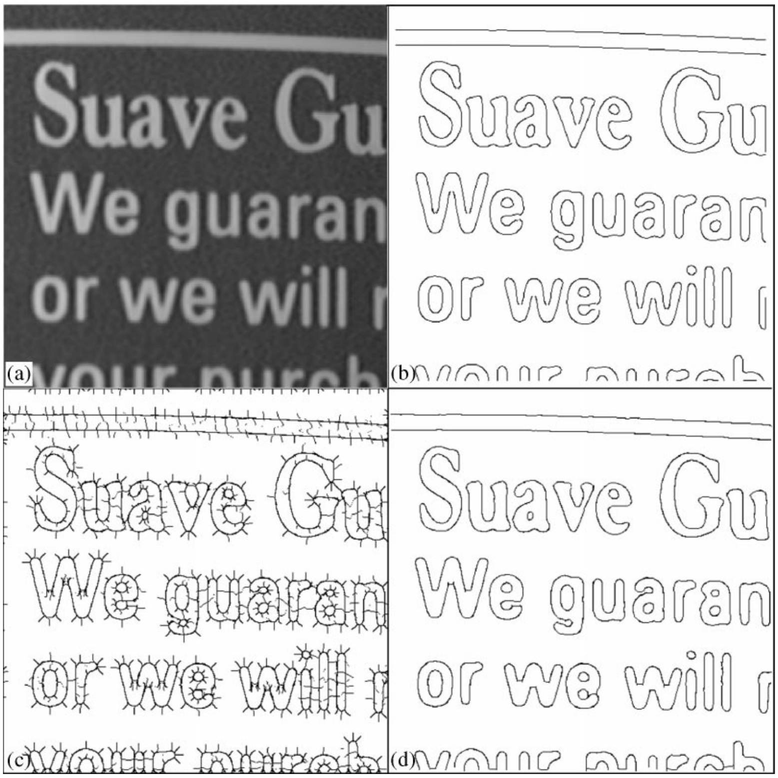

Fig. 4. (a) An image of printed text. (b) Edges determined by the Canny edge detector. (c) The minor edges. (d) Edges determined by the revised Canny edge detector.

edge contours have been obtained also. We believe that these spurious contours are a side effect of the Fourier transform we used to perform the convolutions. The magnitudes of these spurious edges are, however, very small and they can be removed by thresholding. These edge contours are usually harmless, and even if they are not removed by thresholding, they will be discarded by our algorithm since they contain no major edges. Adding parts of the minor edge contours that are delimited by major edges to the major edges determined earlier, we will obtain the final result shown in Fig. 3(d). This image shows the edges of Fig. 3(a) obtained by the revised Canny edge detector.

Fig. 4(a) shows an image of printed text. The characters are well defined, and we expect to obtain the edges contours delineating the characters. Edges obtained by the Canny edge detector are shown in Fig. 4(b). Although an overwhelming majority of the correct edges are detected, a small number of the edges are missed, causing boundaries to become disconnected. The discontinuities can be observed in letters 'u', 'v', and 'w'. Although image noise is not significant and letters 'u', 'v', and 'w' are rather well defined, the Canny edge detector misses some of the edges because of the shapes of these characters. If we examine the minor edges as shown in Fig. 4(c), we see that the edges not picked by the Canny edge detector are among them. Selecting from among the minor edges those that connect the major edges, and combining the selected minor edges with the major edges, we obtain the edges shown in Fig. 4(d). The edge contour delineating each character is now closed.

4. Discussion and conclusion

To ensure that closed edge contours are obtained, one may use the zero-crossings of the Laplacian of Gaussian [1] of the image. The zero-crossings of the Laplacian of Gaussian of Fig. 4(a) after hysteresis thresholding is shown in Fig. 5(a). As can be observed, the characters are well delineated. A small number of the zero-crossings do not correspond to locally maximum intensity changes but rather correspond to locally minimum intensity changes. Edges corresponding to locally minimum intensity changes are referred to as phantom edges [4]. After removing the phantom edges, we obtain the image shown in Fig. 5(b). The phantom edges have appeared only in the 'w's'. Fewer edges have been missed by the Clark method than by the Canny method.

Some differences in the quality of edges detected by the Canny, Clark, and Marr-Hildreth methods can be

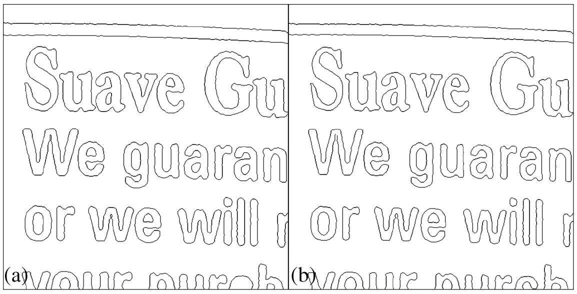

Fig. 5. (a) The zero-crossings of the Laplacian of Gaussian of the printed text image after hysteresis thresholding. (b) The zero-crossing edges after removal of the phantom edges.

observed. Such differences are believed to be due to implementational differences. For example, convolutions may be implemented directly or using the Fourier transform; edge contours may be represented by 4-neighbors or 8-neighbors; edge positions may be determined using locally maximum intensity gradients or zero-crossings of the second derivative of image intensities; and so on.

We have tested two different implementations of the Canny edge detector, one developed locally and one developed by Winder [13] in a package called ImgStar. Although there were some differences in the quality of edges obtained by the two methods, the detected edges were basically similar: both missing some obvious edges.

Edge contours representing object boundaries play a major role in object recognition. An object that is bounded in an image should produce a closed boundary contour. A boundary contour that breaks into pieces loses its effectiveness in shape analysis and object recognition. As demonstrated through a simple example in Fig. 1, locally maximum intensity changes along the gradient direction in a Gaussian-smoothed image do not pick up all the obvious edges in an image. Finding edges according to this definition misses some critical edges. Although the number of edges missed may be small, the effect could be significant. A single missing edge element will turn a closed contour into an open one, disabling an object recognizer from detecting and recognizing an object.

In this paper, a method to recover edges missed by the Canny edge detector was presented. The method involves a step that looks for further image evidence and connects short edge contours into longer ones and converts open contours into closed ones if image evidence supports that.

References

[1] D. Marr, E. Hildreth, Theory of edge detection, Proc. Roy. Soc. London B-207 (1980) 187–217.

[2] R. Haralick, Digital step edges from zero crossing of second directional derivatives, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1984) 58–68.

[3] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698.

[4] J.J. Clark, Authenticating edges produced by zero crossing algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 11 (1) (1989) 43–57.

[5] F. Ulupinar, G. Medioni, Refining edges detected by a LoG operator, Comput. Vision, Graphics, Image Process. 51 (1990) 275–298.

[6] F. Bergholm, Edge Focusing, IEEE Trans. Pattern Anal. Mach. Intell. 9 (1987) 726–741.

[7] A. Goshtasby, H-L. Shyu, Edge detection by curve fitting, Image Vision Comput. 13 (3) (1995) 169–177.

[8] J.H. Elder, S.W. Zucker, Local scale control for edge detection and blur estimation, IEEE Trans. Pattern Anal. Mach. Intell. 20 (7) (1998) 699–716.

[9] S.Z. Li, Closed-form solution and parameter selection for convex minimization-based edge preserving smoothing, IEEE Trans. Pattern Anal. Mach. Intell. 20 (9) (1998) 916–932.

[10] M.D. Heath, S. Sarkar, T. Sanocki, K.W. Bowyer, A robust visual method for assessing the relative performance of edge-detection algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 19 (12) (1997) 1338–1359.

[11] R. Jain, R. Kasturi, B.G. Schunck, Machine Vision, McGraw-Hill, 1995, pp. 169–173.

[12] E. Trucco, A. Verri, Introductory Techniques for 3-D Computer Vision, Prentice-Hall, Englewood Cliffs, NJ, 1988, pp. 71–79.

[13] S.A.J. Winder, ImgStar: Image Processing Toolset and C Library, 1994. http://www.ph.tn.tudelft.nl/PRInfo/PRInfo/software/msg00008.html.

About the Author—LIJUN DING is currently a Ph.D. candidate in the Department of Computer Science and Engineering at Wright State University. He received his master's degree in Electrical Engineering from the Institute of Science, Chinese Academy of Science in 1997 and his bachelor's degree in Electrical Engineering from the University of Science and Technology in China in 1994. His dissertation research is on registration of volumetric images using elastic transformation functions.

About the Author—ARDESHIR GOSHTASBY is an Associate Professor in the Department of Computer Science and Engineering at Wright State University. He received his Bachelor's degree in Electronics Engineering from the University of Tokyo in 1974, his master's degree in computer science from the University of Kentucky in 1975, and his doctorate also in computer science from Michigan State University in 1983. Dr. Goshtasby has worked on various problems in computer vision, image processing, computer graphics, and geometric modeling. His current research interests are in volume image analysis and representation of free-form shapes with parametric curves, surfaces, and volumes.