Using Sliding Window Technique to Explore the Variations of Image Pixels for Edge Detection

Jau-Ji Shen* and Chia-Ming Zen
*Department of Management Information Systems, National Chung Hsing University,
250, Kuo Kuang Rd., Taichung 402, Taiwan, R. O. C.
Institute of Information Management, National Formosa University,
64, Wen Hua Rd., Huwei, Yunlin 632, Taiwan, R. O. C.
jj.shen@msa.hinet.net

Abstract

This paper proposes an edge detection method using sliding window technique. Compared to traditional methods, the major difference is that the size of sliding window is floating. The window size will be modified according to the variance of all pixel values and mean value of the sum of all differences between each pixel value and the minimum pixel value in the sliding window. This dynamic windowing concept will be applied to detect edges from four different directions, hence many windows with different size will be generated. All pixel values in the same window are similar because of their low variations. After all pixels of the image being included into a certain window, four directional edge maps are generated. The proposed method can detect much thinner edges than Sobel method. Compared with Kang and Wang's method, the proposed method can extract all the major edges and is also timesaving.

1. Introduction

In the widespread applications of computer vision and pattern recognition, it is a critical task to distinguish the objects from their background. The complete information of edge in an image must be acquired so that the rough sketch of the objects will be generated. Thus edge detection is essential for image processing. Edge detection, as implied by the name, means extracting accurate edge lines with correct orientation in the considered image. Many papers have been published during the past two decades. One of common criteria for edge detector performance is low error rate. It is significant that edges occurring in the image should be considered edge points and spurious responses should be as less as possible. Moreover, minimizing the distance between marked point and the actual fact of itself is also another performance indicator.

Several well-known edge detection methods use masks to do convolution on the image, and the edges will be detected based on the dramatic changes of the gray level [1]. But the extracted edge is usually thicker under such conditions where the gray level variation in the image is smooth. Canny [2] improves Sobel method [1] by analyzing the vertical and horizontal edge intensities of the pixel to resolve the direction angle of the edge point, and then the non-maxima suppression (NMS) [2] is employed to attain the edges. Kang and Wang [3] also proposed an edge detection method using four edge patterns of different directions to obtain the edge and direction maps, and then utilized NMS to determine the edge points. By using fuzzy sets to detect edges is another approach proposed by [4, 5] but it requires rather large rule sets compared to simpler fuzzy methods [6]. The competitive fuzzy edge detection (CFED) method is proposed by Liang and Looney [7]. CFED splits edge types into six patterns according to a fuzzy classifier. Nevertheless, CFED usually extracts speckles in some more detailed regions of an image.

In this paper, an edge detection method based on sliding window technique is proposed. Compared to traditional methods, the major difference is that the sliding window size is dynamic. Besides, our method is also capable to generate the sketch of an image from its detected directional edge maps.

The rest part of this paper is structured as follows. Section 2 reviews Sobel method and Kang's method. Our proposed method is discussed in Section 3. Section 4 mainly presents experimental results, and conclusion of this paper is in the last section.

2. Backgrounds

Before introducing our proposed method, the popular edge detection method, Sobel method, and the edge detection technique proposed by Kang and Wang [3] will be described in this section.

2.1 Sobel method

Sobel method is a well-known edge detection technique. By applying two 3×3 masks S_x and S_y, to do convolution on a gray level image, the edge intensities E_x and E_y, corresponding to vertical and horizontal directions will be generated (as shown in Fig. 1). After taking the summation of absolute values of E_x and E_y, we will obtain the edge intensity of mask center. By a given appropriate threshold *T*, while the edge intensity of each pixel is larger than *T*, this pixel will be considered an edge point. Unfortunately, the edge extracted by Sobel method is usually thicker than the actual fact of itself [7].

1	0	-1
2	0	-2
1	0	-1

S_x

1	2	1
0	0	0
-1	-2	-1

S_y

Fig. 1 Two convolution masks in Sobel method.

2.2 Kang and Wang's method

Kang and Wang proposed an edge detection method based on four edge patterns for different directions to extract edge map and direction map from the image, and then NMS is employed to attain the edge points.

*p*1	*p*2	*p*3
*p*4	*p*5	*p*6
*p*7	*p*8	*p*9

Fig. 2 The 3×3 mask.

The coefficients in the 3×3 mask (as indicated in Fig. 2) will be divided into two pixel sets (S_0 and S_1) according to four possible direction edges (as shown in Fig. 3). Hence, four edge patterns in accordance with different directions are generated.

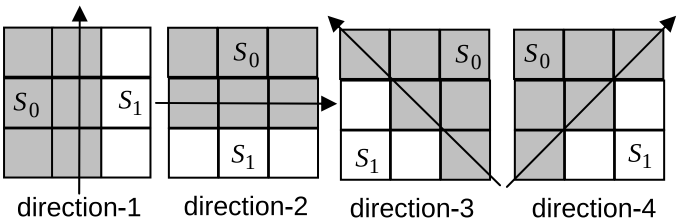


Fig. 3 Four possible direction edges.

Subsequently, four edge intensities of each pixel corresponding to those possible directions will be obtained using a predefined object function. By selecting the maximum edge intensity of each pixel and the direction of the maximum edge intensity, the edge map and direction map are generated. Finally, the non-maxima suppression method is applied to the edge and direction maps to extract the edge points. For more details about the above-mentioned method, readers with interest can refer to the literature [3].

3. Proposed method

The basic idea of our method is using sliding window with dynamic window size to detect edge maps from four different directions. Then six 1×5 and two 1×3 edge selectors are applied to sift out edges from these edge maps, and the final edge image will be decided by majority vote. The dynamic window size will be changed according to the variations of all pixel values in the sliding window. The variance of all pixel values and mean value of the sum of all differences between each pixel value and the minimum pixel value in the sliding window are the major criteria for measuring the variations. The proposed method will be described in more detail in the following sections.

3.1 Directional edge maps generation

For a given gray level image, the window with dynamic size is applied to slide on the image to detect edges from four different directions. The concept of the sliding window of our method is shown in the following figure.

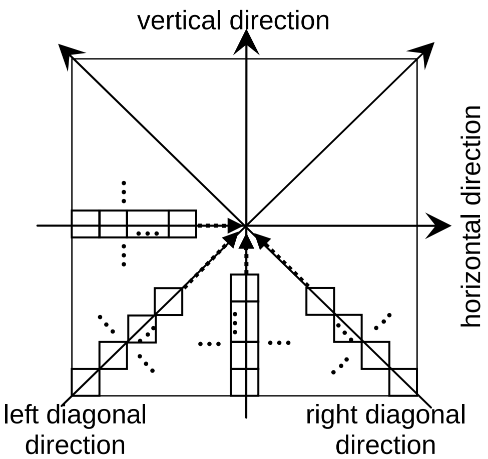


Fig. 4 The concept diagram of the proposed method.

For any sliding direction, all pixels in the image will be included into different windows. Along each direction, a new sliding window is created when any one of the following three cases occurs. Along the sliding direction,

(1) the pixel is the initial value in each column or row,

(2) after including the current pixel value into the sliding window, the variance of all pixel values in this window will exceed T_1, where T_1 is a predefined threshold,

(3) after including the current pixel value into the sliding window, the mean value of the sum of all differences between each pixel value and the minimum pixel value in the sliding window is larger than T_2, where T_2 is the other predefined threshold.

The process above will be performed until all pixels of the image are included into a certain window.

After all pixels in the image have been processed, many windows with different size will be generated. Along the horizontal sliding direction, for example, any two adjacent windows their pixel values will be set to 0 and 1 accordingly. Therefore, a directional edge map with sliding direction based mainly on horizontal direction is generated (as shown in Fig. 5(b)). Similarly, the method is applied to the image to attain the other directional edge maps along three different sliding directions.

Fig. 5 Peppers and its directional edge map from horizontal direction.

3.2 Sketch of an image

During the process of detecting image edges, four directional edge maps of different sliding directions will be obtained. Let any two adjacent pixels in each directional edge map do EOR (Exclusive-OR) operation along its sliding direction. Thus, four rough sketches of an image can be generated. Under such conditions that a pixel in one of the four rough sketches is an edge point, the pixel in the sketch of an image is also regarded as an edge point. Therefore, the sketch of an image is obtained (as shown in Fig. 6(a)).

(a) T_1=10, T_2=50 (b) T_w=4, T_1=10, T_2=50

Fig. 6 Sketches of "peppers" image based on different approaches.

In addition to applying the simple version for generating the sketch of an image (as mentioned above), the following process of sketching image also can achieve a better result.

This approach significantly differs from the above-mentioned method is that the preprocess will be performed before setting all the pixel values in the same sliding window to 0 or 1. After many sliding windows with dynamic size have been generated, all the sliding windows with size equal or exceed a predefined threshold T_w will be found. Subsequently, those selected big sliding windows will be divided into several sub-windows according to T_1, T_2 and T_w. After including the current pixel value into the sub-window, this pixel value will be contained in a new created sub-window under one of the three following conditions: (1) The size of sub-window is larger than T_w. (2) The variance of all pixel values in the sub-window exceeds T_1. (3) The mean value of the sum of all differences between each pixel value and the minimum pixel value in the sub-window is larger than T_2. Thus, many sub-windows are generated with size no greater than T_w.

Before setting all the pixel values in the same sliding window to 0 or 1, the following process is also necessary. Any two or more adjacent sub-windows their pixel values will be integrated into a new sub-window while their size are all equal to T_w -1. Subsequently, along each sliding direction, any two adjoining windows their pixel values will be set to 0 and 1 accordingly. In other words, whether the sliding window is a sub-window or not, all the pixel values in the same sliding window will be set to 0 or 1. Finally, by doing EOR operation to generate four rough sketches and edge points' selection, the sketch of an image will be obtained (as indicated in Fig. 6(b)).

3.3 Edge selectors and vote

During the course of edge detection, four directional edge maps corresponding to different sliding directions can be produced. The following six 1×5 and two 1×3 edge selectors are applied to those edge maps to sift out edges along each sliding direction.

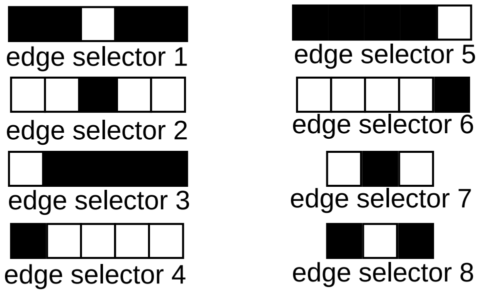


Fig. 7 Six 1×5 and two 1×3 edge selectors.

The step-by-step illustration for selecting edge points is stated as follows. For the given 4 directional edge maps,

Step 1. Perform edge selectors 1-6 to sift out edge points. Edge selectors 1 and 2 are employed to scan the directional edge maps along each sliding direction. Once any five adjacent pixels of the directional maps match those edge selectors, the third pixel will receive a vote. In the same way, while edge selectors 3 and 4 are applied to sift out edges from those directional edge maps, the first of five matched adjoining pixels will also get a vote. For the case of edge selectors 5 and 6, the last of five matched adjacent pixels will receive a vote.

Step 2. Apply edge selectors 7 and 8 to sieve possible edge points. By scanning each directional edge map, any three adjacent pixels matched edge selectors 7 or 8 will be acquired, and then all of the second of three matched adjacent pixels are regarded as possible edge points. All of the possible edge points detected from the directional edge maps are merged into a new image. Subsequently, we should determine which edge points in this new merged image will get a vote. Given D as a threshold, once the length of any line (i.e., a segment of consecutive possible edge points along the sliding direction) in this merged image is no less than D, the first and last pixels will receive a vote, respectively. In other words, pixels of the directional edge map with same position to those pixels receiving votes will also get a vote. Therefore, all pixels of those different directional edge maps have been accomplished its voting.

Step 3. Decide the final edge image by counting the votes. By counting votes of each pixel in those directional edge maps, a pixel with number of votes exceeding a predefined threshold V will be regarded as an edge point.

It is worth to note that since thick edges and speckles are imperfect result, we must perform the preprocessing as mentioned earlier (Step 2) when edge selectors 7 and 8 are applied to sift out edges. Thus, detected edges with thick edges and speckles will be as less as possible.

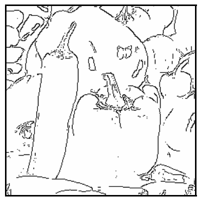
4. Experimental results

Two 256×256 gray images are utilized to compare the proposed method with Sobel method and the method proposed by Kang and Wang. Experiments are described as follows.

(a) original "peppers" image

(b) T_1=30, T_2=200, D=3 and V=3

(c) T=130



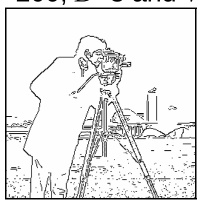
(d)w_1=90, w_2=40 and T=50

Fig. 8 Comparisons of different methods: (a) original image, (b) the proposed method, (c) Sobel method, (d) Kang and Wang's method.

(a) cameraman

(b) our method (T_1=30, T_2=200, D=3 and V=3)

(c) Sobel method (T=110)



(d) Kang and Wang's method (w_1=90, w_2=40 and T=50)

Fig. 9 Detected edges by different methods for "cameraman" image.

Let the "peppers" image (as show in Fig. 8(a)) be the first example and the "cameraman" image (as indicated in Fig. 9(a)) be the second example, respectively. The edges detected by the proposed method (as illustrated in Fig. 8(b) and Fig. 9(b)) are thinner than Sobel method (as indicated in Fig. 8(c) and Fig. 9(c)). Compared to Kang and Wang's method (as shown in Fig. 8(d) and Fig. 9(d)), our method also can extract all the major edges. All the threshold values in Kang and Wang's method are referred to the literature [3], and the optimal threshold value for Sobel method is acquired by a series of parameter tuning.

From the above experiments on ordinary images, there is still not so clear to show how difference of our method to the others. The experiment on one-pixel-width image, Fig. 10(a), is conducted to evaluate these methods. It is clear that double edges are detected by Sobel and Canny methods. Fig 10(e) shows that the edges don't connect very well on each corner, and the edges are thicker than those edges in Fig. 10(d).

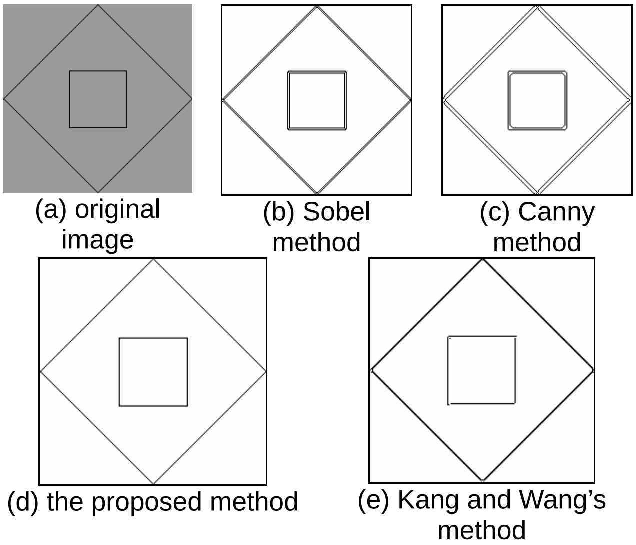


Fig. 10 Line detection comparisons of different edge detectors.

For the algorithm performance, ten different gray images are applied to test the proposed method and Kang and Wang's method. On average, our method spends 104.24 seconds in detecting edges from an image while the Kang and Wang's method takes 141.62 seconds to do the same work.

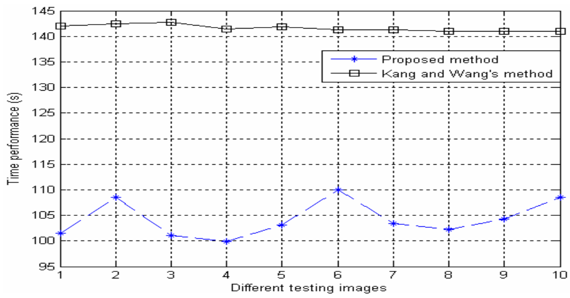


Fig. 11 Comparisons of time performance.

5. Conclusions

In this paper, an edge detection method based on sliding window technique is proposed. The size of the sliding window is floatingly applied on an image to generate four directional edge maps for four different directions. Subsequently, six 1×5 and two 1×3 edge selectors are designed to sieve edge points from those directional edge maps. The larger edge selector is employed to attain the higher variations edge points from those directional edge maps. The smaller ones are also utilized to sift out edge points from those directional edge maps in which any three adjacent pixel values having abrupt changes. Some of the edge points sifted out by the smaller edge selectors must be weeded out so as to prevent thick edges or speckles occurring. Besides, those directional edge maps also can be applied to generate an image's sketch. By the experimental results, the detected edge image of our method is thinner than Sobel method. Compared to Kang and Wang's method, the proposed method is not only extracting all the major edges but also timesaving. However, each sifted edge point receiving a vote is unjust, and how to design appropriate edge selectors for those directional edge maps is also a significant task.

6. References

[1] N. Efford, Digital Image Processing, Addison-Wesley, Reading, MA, 2000.

[2] J. F. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Analysis Mach. Intell. Vol. 8, Issue 6, 1986, pp. 679-698.

[3] C. C. Kang and W. J. Wang, "A novel edge detection method based on the maximizing objective function," Pattern Recognition, 2006.

[4] T. Law, H. Itoh and H. Seki, "Image filtering, edge detection, and edge tracing using fuzzy reasoning," IEEE Trans. Pattern Analysis Mach. Intell. Vol. 18, Issue 5, 1996, pp. 481-491.

[5] F. Russo and G. Ramponi, "Fuzzy operator for sharpening of noisy images," IEE Electronics Letters, Vol. 28, Issue 18, 1992, pp. 1715-1717.

[6] C. G. Looney, "Nonlinear rule-based convolution for refocusing," Real-Time Imaging. Vol. 6, Issue 1, 2002, pp. 27-37.

[7] L. R. Liang and C. G. Looney, "Competitive fuzzy edge detection," Applied Soft Computing, Vol. 3, Issue 2, 2003, pp. 123-137.