Edge detection algorithm based on valley lines and watershed

Bo Wang1*, Zhiying Yang1

1 School of Automobile and Traffic Engineering, Wuhan University of Science and Technology, Wuhan, Hubei, China

*Corresponding author’s e-mail: wangbosofia@gmail.com

Abstract. The purpose of edge detection algorithm is to highlight the edge in the image, which is also the premise and key of image segmentation. However, the traditional edge detection algorithm will amplify the influence of noise in the process of derivative operation, and the noisy edges will often appear, which is not conducive to the accurate extraction of edge. To solve this problem, we propose an edge detection algorithm combining valley lines and watershed algorithm. In this algorithm, the bilateral filter is used to replace the Gaussian filter to achieve edge preserving and denoising. The rewritten Canny operator is used to calculate the low value points and get the valley lines of the image. This process can achieve the first denoising. Then, the valley lines are used to replace the discrete points of water injection and the watershed algorithm is used to calculate the segmentation results of the image. At this time, there are many pseudo edges around the main boundary, and the pseudo edges are deleted by setting the threshold, so as to achieve the effect of secondary denoising. The experimental results show that our method has good performance in noise resistance and edge continuity compared with other and traditional edge detection algorithms.

1. Introduction

Image edges usually refer to the set of pixel points where the pixel gray value varies discontinuously. In the image processing, complete image edges can well represent the shape of an object and speed up the computational efficiency. Therefore, edge detection techniques have a crucial role in the fields of target recognition, image segmentation, object modeling, and computer vision [1].

The traditional edge detection operators include Roberts operator, Prewitt operator, Sobel operator and Canny operator. Compared with the other three operators, the Canny operator has the advantages of high accuracy and high signal-to-noise ratio, but it is susceptible to the interference of pepper noise as well as the poor adaptability of double threshold selection. Therefore, many improvement methods have been proposed by domestic and foreign scholars for the Canny operator. Liu [2] uses guided filtering for noise removal and uses the Otsu algorithm to select the dual thresholds in the Canny operator to improve the adaptability; a hybrid filter is proposed by mixing the median filter and the geometric mean filter and uses this to replace the Gaussian filter for image smoothing operation [3]; a multi-scale Canny operator technique [4] is used to enhance the real edges and eliminate spurious edges, the algorithm is more robust to noise, but it is slow and inefficient; some methods combine more image information for optimization, although they can extract more accurate edges, they have low computational efficiency [5,6,7,8]; a method improves the threshold selection and optimizes the extraction of weak edges, which combines the knowledge of machine learning and uses genetic algorithms, but it requires a large number of samples for training [9,10].

In this paper, we propose an edge detection algorithm combining valley lines and watershed algorithm. The method smoothes the image with bilateral filtering instead of Gaussian filtering to
achieve the first denoising, then extracts the valley lines of the image by the rewritten Canny operator. We calculate the image segmentation result by replacing the local minima points in the watershed algorithm with valley lines, and define the pseudo-edges and remove them by combining the prior knowledge of the SFE algorithm to obtain the final result. Our method is compared with Canny operator and the method in [11] for experiments and the experimental results are evaluated quantitatively. The experimental results show that our method has good performance in terms of noise immunity performance and extracted edge continuity.

2. Traditional Canny and watershed algorithm

2.1. Analysis of traditional Canny algorithm

In 1986, John Canny proposed a multilevel edge detection algorithm and gave three major criteria for edge detection [12].

- **Signal-to-noise criterion.** This criterion means that the true edge must be detected as much as possible. The mathematical expression for the signal-to-noise ratio is

\[
\text{SNR} = \frac{\int_{-\infty}^{+\infty} G(-x)f(x)dx}{\sigma \sqrt{\int_{-\infty}^{+\infty} f^2(x)dx}}
\]

(1)

where \(G(-x)\) is the edge function; \(f(x)\) is the impulse response of the filter; and \(\sigma\) is the mean squared deviation of the Gaussian noise.

- **Positioning accuracy criterion.** This criterion means that the distance from the detected edge point to the true edge point must be minimized as much as possible. The mathematical expression for the localization accuracy is

\[
L = \frac{\int_{-\infty}^{+\infty} G'(x)f'(x)dx}{\sigma \sqrt{\int_{-\infty}^{+\infty} f'^2(x)dx}}
\]

(2)

- **Single-edge response criterion.** This criterion means that the algorithm must return only one edge point as much as possible. To ensure a single-edge response, the zero-crossing average distance of the impulse response derivatives of the detection operator should satisfy

\[
D(f) = \pi \left( \frac{\int_{-\infty}^{+\infty} f'^2(x)dx}{\int_{-\infty}^{+\infty} f'^2(x)dx} \right)
\]

(3)

Based on the three criteria, the traditional Canny edge detection algorithm is divided into four main steps.

- A Gaussian filter is used to smooth the image to eliminate noise.
- Calculating of gradient amplitude and direction by using finite difference.
- Running non-maximum suppression according to the gradient direction. Detecting whether the gradient amplitude of each pixel point in a certain gradient direction is a maximum value.
- The Canny algorithm sets high threshold \(T_h\) and low threshold \(T_l\), and compares the gradient magnitude with the threshold. If the amplitude is higher than \(T_h\), the pixel is an edge point; if the amplitude is lower than \(T_l\), the pixel is a non-edge point; if the amplitude is between \(T_l\) and \(T_h\), find whether there is a point with amplitude higher than \(T_h\) near the pixel, if so, it is considered an edge point, otherwise not.

Compared with the traditional differential operator, the Canny operator is widely used because of the advantages of fast operation and high detection accuracy, but there are also many shortcomings, such as the Gaussian filtering in the first step to can smooth the noise, but at the same time it will blur the edges, and lose a lot of detailed information, and it is difficult to eliminate the interference of pepper noise.
The setting of the double threshold in the fourth step needs to rely on manual experience, and the adaptivity is not strong.

2.2. Analysis of watershed algorithm

The watershed algorithm is implemented based on a simulated water immersion process [13]. According to the topographic theory of gray scale maps, the whole image can be considered as a 3D image, where the gray scale value indicates the information in the third dimension. Assuming that the image is submerged in water, a hole is punched at the minimum point, and water will slowly flow from the hole into the catchment basin of the image. As the water surface rises in the two catch basins, at some point the water surfaces will touch each other and pool, at this time a dike is built to stop the water from pooling. When the submergence process is over, all the dike collections form a watershed.

According to the principle of traditional watershed, it is assumed that \( P_1, P_2, \ldots, P_m \) are the minimal value points in the image \( I(x, y) \), and \( S(P) \) is the set of coordinates of pixel points in the water accumulation basin corresponding to the minimal value points. \( T[n] \) represents the set of points in the image that lie below the plane \( I(x, y) = n \). When water is immersed from the injection point, the \( xy \) plane produces a binary image \( L_n(P) \), which can be expressed as (5).

\[
T[n] = \{(s, t) | I(s, t) < n\} \quad (4)
\]

\[
L_n(P) = L(P) \cap T[n] \quad (5)
\]

The algorithm starts injecting water from the minimal value point, and the initial condition is \( L[\min + 1] = T[\min + 1] \), and then the algorithm performs iterative recursion. \( L[n] \) is obtained by computing \( L[n-1] \), and assuming that \( Q \) denotes the set of connected components in \( T[n] \), for each \( q \in Q \), there are three possibilities as follows:

- \( q \cap L[n-1] \) is the empty set;
- \( q \cap L[n-1] \) contains one connected component in \( L[n-1] \);
- \( q \cap L[n-1] \) contains at least one connected component in \( L[n-1] \).

The process of computing \( L[n] \) by \( L[n-1] \) depends on one of the three conditions mentioned above. The algorithm satisfies condition one when a new minimum value appears, at which \( q \) is merged into \( L[n-1] \) to generate \( L[n] \); the algorithm satisfies condition two when \( q \) is located in some catchment basin, at that time the operation of condition (1) is repeated; the algorithm satisfies condition three when more than two catchment area ridgelines appear, which if continued inundation would result in catchment regions to merge, so a dike needs to be constructed within \( q \) to prevent water level pooling between separate catchment basins.

3. Edge detection algorithm based on valley lines and watershed

3.1. Valley lines detection algorithm

3.1.1. Introduction of bilateral filtering for image smoothing. To address the phenomenon that Gaussian filters can lead to blurred edges, we consider a bilateral filter to replace the Gaussian filter for image smoothing. The bilateral filter is a nonlinear null domain filter that considers the effects of both spatial proximity and gray scale similarity. Compared with noise, the gray scale variation of edges will be greater, so the bilateral filter can adjust the weight of a neighboring pixel point by judging whether it is located at an edge to achieve edge-preserving denoising. The mathematical expression of the bilateral filter is
\[ I = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_p} \| p - q \| G_{\sigma_q} \| I_p - I_q \| I_q \]  
\[ (6) \]

Where \( W_p \) is the normalization function whose mathematical expression is

\[ W_p = \sum_{q \in S} G_{\sigma_p} \| p - q \| G_{\sigma_q} \| I_p - I_q \| \]  
\[ (7) \]

where \( I \) is the processed image; \( q \) is the central pixel; \( S \) is the neighborhood centered on pixel \( q \); \( p \) is a pixel in the neighborhood of \( S \); \( G_{\sigma_p} \) is the spatial proximity function, which indicates the influence degree of the distance from a pixel in a certain neighborhood to the central pixel, the farther the distance, the lower the weight; \( G_{\sigma_q} \) is the gray scale similarity function, which indicates the influence degree of the gray scale difference between a pixel in a certain neighborhood and the central pixel, the greater the gray difference, the lower the weight.

### 3.1.2. Image segmentation based on valley lines.

The traditional watershed algorithm is sensitive to the noise, and the noise makes very many local minima points in the image, which leads to over-segmentation of the image. In order to further eliminate the effect of noise, we replace the local minima points in the traditional watershed algorithm by valley lines. Applying this method to segmentation, the over-segmentation phenomenon caused by the noise induced minima can be avoided. The traditional Canny operator employs non-maximal suppression, which allows it to compute points with large gray scale variations in the image. In this paper, we rewrite the non-maximal suppression to keep the points with the smallest local gradient variation so that it can compute the pixel points with small gray scale variation, and we call it the low-value points. With the low-value point as the center, all pixel points in the 8 neighborhoods near the low-value point are labeled and defined with different intensity values. The pixel intensity value is inversely proportional to the distance from the low-value point, and the low-value points with intensity values smaller than a given threshold are deleted, and the set of all remaining pixel points is the valley line of the image. Subsequently, the smoothed image is segmented using the valley lines as local minima points.

### 3.1.3. Valley lines extraction results.

The following figure shows the results of the image valley lines extracted by the valley line detection algorithm, where the original image is on the left and the corresponding valley line results are on the right.

![Figure 1](image_url)

**Figure 1.** Example of the image valley lines. (a): ‘Lena’ image. (b): The valley lines of ‘Lena’.

### 3.2. Define pseudo edge and delete

The image processed by the Watershed algorithm generates some pseudo-edges due to the influence of noise. To solve this problem, this paper redefines pseudo-edges with the boundaries obtained by the
Structured Forest Frontier Extraction (SFE) algorithm as a priori knowledge. The method of defining pseudo-edges is as follows.

- We record the number of all pixel points on each SFE edge, and when the distance between a point on an edge and the SFE edge feature point is greater than a given threshold, the point is considered a marker point. When the number of marker points is greater than half of all the pixel points on the edge, the edge is considered as a pseudo edge.
- We consider that the edge intersecting the SFE edge is a pseudo edge.

Removing the pseudo-edges in the above two cases can achieve secondary denoising and get more accurate image edges.

4. Experimental results and analysis

To further test the edge detection effectiveness of our algorithm, the experiments are done on an Intel Core i5-9300HF CPU with 16GB RAM computer and programmed by MATLAB R2018a. In our paper, two classical images in figure 2 are used for experiments to verify the effectiveness of our method in image edge detection in terms of both visual observation and quantitative analysis, and to compare with the traditional Canny edge detection algorithm and the method mentioned in [11].

Figure 2. Two experimental images. (a): ‘Cameraman’ image. (b): ‘House’ image.

To verify the noise-resistance performance of the algorithm in this paper, we set up two sets of control experiments. Zero-mean Gaussian noise with variance of 0.005 and 0.05 is added in experiment 1, and pepper noise with density of 1% and 5% is added in experiment 2.

As can be seen from figure 3 and figure 4, the detection effect of the Canny algorithm is very unsatisfactory, and the detection is disturbed more severely as the noise concentration increases. In contrast, the algorithm of [11] has the best denoising effect, but at the same time, it will remove many major edges, and its overall detection effect is not good. Although the algorithm in our paper will have some noise points, the main boundary can be extracted well. We can get more continuity edges in our algorithm, and the edges are more accurate.

Figure 3. Results of edge detection of the proposed algorithm. (a) ‘House’ image with Gaussian noise with variance of 0.005. (b) Canny algorithm. (c) The algorithm in [11]. (d) Our algorithm.
Figure 4. Results of edge detection of the proposed algorithm. (a) ‘House’ image with Gaussian noise with variance of 0.05. (b) Canny algorithm. (c) The algorithm in [11]. (d) Our algorithm.

From figure 5 and figure 6, it can be seen that when the pepper noise density is 1%, the detection results of the traditional Canny algorithm will be interspersed with many noise points, which is because the traditional Canny algorithm uses a Gaussian filter for image smoothing, and this filter does not have good anti-noise performance for pepper noise. Both the algorithm of [11] and our algorithm have good performance, but the phenomenon of partial edge loss and breakage is obvious in [11], which is due to the fact that the median filter can filter the pretzel noise well, but it will lose the local detail features. When the pepper noise density is 5%, the traditional Canny algorithm can no longer effectively detect the image edges, but the algorithm of literature [11] and the algorithm of this paper are almost unaffected.

Figure 5. Results of edge detection of the proposed algorithm. (a) ‘Cameraman’ image with pepper noise with density of 1%. (b) Canny algorithm. (c) The algorithm in [11]. (d) Our algorithm.

Figure 6. Results of edge detection of the proposed algorithm. (a) ‘Cameraman’ image with pepper noise with density of 5%. (b) Canny algorithm. (c) The algorithm in [11]. (d) Our algorithm.

A method to evaluate the performance of the edge detection algorithm is proposed in [14]. This method is quantified by the ratio between the total number of extracted edge pixels A, the number of 4-connections B and the number of 8-connections C. The value of C/A is used to evaluate the goodness of the edge continuity, the smaller its value, the stronger the continuity of the detected edge; the value of C/B is used to evaluate whether the edge is a single pixel edge, the smaller its value, the better the single-edge. Using the noise-free ‘Cameraman’ image and ‘House’ image as the experimental objects, the detection results of Canny algorithm, the algorithm of [11] and our algorithm are counted by this
method, and the statistical results are shown in Table 1. From Table 1, it can be seen that our algorithm has the lowest C/A and C/B values in both images, which indicates that the extraction effect of our algorithm is better. The extracted edges have good performance in both continuity and intermittent, and the single edge response is better.

Table 1. Evaluation of performance.

| Algorithm          | Cameraman |       | House |       |
|--------------------|-----------|-------|-------|-------|
|                    | C/A       | C/B   | C/A   | C/B   |
| Canny              | 0.0278    | 0.2065| 0.0229| 0.1725|
| Algorithm in [11]  | 0.0114    | 0.1034| 0.0064| 0.0650|
| Ours               | 0.0091    | 0.0812| 0.0053| 0.0476|

5. Conclusions
In this paper, we proposed an edge detection algorithm based on valley lines and watersheds. The method first uses bilateral filtering instead of Gaussian filtering in the traditional Canny operator to smooth the image while protecting the edge information of high frequencies. At the same time, the Canny operator is rewritten to calculate the low-value points of the image and generate the valley lines of the image. Then we replace the local minimal value points with the valley lines and using the watershed algorithm to get the segmentation of the image. Finally, by defining pseudo-edges, we set the corresponding thresholds and remove the pseudo-edges, the accurate image edge results are obtained. After the simulation experiment comparison, ours algorithm have good anti-noise performance no matter under the influence of Gaussian noise or pepper noise. Through quantitative analysis, our algorithm has a greater advantage in the continuity, intermittency and single-edge response of edges.

Our method still needs to be studied in depth, such as the process of removing pseudo-edges by using threshold, which still needs to be selected manually, the degree of self-adaptation needs to be strengthened, and the edge detection effect of the image depends on a priori knowledge. Therefore, how to perform adaptive threshold will be the focus of the next work.

Acknowledgments
We thank to Prof. Li Hu, Qiliang Yang for their useful discussion and comments. The authors acknowledge supports from National Natural Science Foundation of China (Grant No. 51905389).

References
[1] Li B.C., Peng T.Q., and Peng B. (2004) Intelligent image processing technology. Publishing House of Electronics Industry, Beijing. pp. 153–155.
[2] Liu L.X., Li B.W., and Wang Y.P. (2019) Remote sensing image segmentation based on improved canny edge detection. Computer Engineering and Applications, 55: 54-58.
[3] Song R.J., Liu C., and Wang B.J. (2018) An adaptive canny edge detection algorithm. Journal of Nanjing University of Posts and Telecommunications: Natural Science Edition, 38: 72-76.
[4] Yitzhaky Y., Peli E. (2003) A method for objective edge detection evaluation and detector parameter selection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25: 1027-1033.
[5] Leordeanu M., Sukthankar R., and Sminchisescu C. (2014) Generalized boundaries from multiple image interpretations. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36: 1312-1324.
[6] Lei T., Fan Y.Y., and Wang Y. (2014) Colour edge detection based on the fusion of hue component and principle component analysis. IET Image Processing, 8: 44-55.
[7] Dollar P., Zitnick C.L. (2015) Fast edge detection using structured forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37: 1558-1570.
[8] Lawend H.O., Muad A.M., and Hussain A. (2017) Robust edge detection based on canny algorithm for noisy images. Journal of Theoretical and Applied Information Technology, 95: 5104-5114.

[9] Dong Y., Gao Y.B., and Liu X. (2014) Application of the improved genetic algorithm in the threshold selection of canny operator. Journal of Lanzhou Jiaotong University, 33: 1-5.

[10] Fu W.L., Johnston M., and Zhang M.J. (2014) Low-level feature extraction for edge detection using genetic programming. IEEE Transactions on Cybernetics, 44: 1459-1472.

[11] Liu Q.M., Zhang L., and Li X. (2016) Chip image edge detection based on improved canny algorithm. Computer Engineering and Design, 37: 3063-3117.

[12] Canny J. (1986) A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 8: 679-698.

[13] Wang X.P. (2008) Principles and applications of morphological image analysis. Tsinghua University Press, Beijing.

[14] Lin H., Zhao C.S., and Shu N. (2003) Edge detection based on canny operator and evaluation. Journal of Heilongjiang Institute of Technology, 17: 3-6.