Edge detection methods and key parameters study

Qingyu Li
Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Champaign, IL, 61820, United States
qingyul2@illinois.edu

Abstract. Edge is essential for diverse vision processing tasks and it constitutes a crucial part in the image of objects. There has already been a huge amount of researches about edge detection algorithms. The Canny method and the Sobel method are the most widely used techniques because of their relative advantages. Some parameters can drastically affect the performance and results of the two methods, which will be discussed in this paper. In addition, noise is also a factor affecting the number of the extracted edges. Experiment results have demonstrated that the noise filter will achieve better edge detection performance. Further improvements with suitable parameters of different applications for the two methods are possible.

1. Introduction
Edge is the element that separates objects from the surroundings. Finding the edges is crucial to the identification and the location of the objects or image segmentation, and all of these fields are essential for the development of related visual processing technology. Therefore, edge detection has been quite a popular topic since the last century. The underlying rationale behind its complex mechanism is to use the change of gradient to analyze the region and then execute a secondary operation with optimized algorithms and continuity of the edge on it to have a better and more complete result.

There are some traditional edge detection methods. Sobel [1], Prewitt [1] and Robert [1] are ones based on the gradient; Laplacian of Gaussian (also known as LOG) [2], Difference of Gaussian (also known as DOG) [3] and Canny [1] are ones based on Laplacian distribution.

To begin with, these three methods were developed in the early time of edge detection technology. The Sobel method uses the discrete differences between rows and columns and its operator is based on convolving the picture with a small filter. Clearly, it is not suitable for images teemed with complex noise, which could significantly influence the relation between pixels, whereas it works well for moderate noise and the edges detected are usually two-pixel wide or above. In the Robert method, the vertical and horizontal edges are identified individually and put together in the later procedure. Robert is ideal in cases where the relative angle is 45 degrees and is notable for the high definition of edge location. However, its resistance to noise is as poor as that of the Sobel method. As for the Prewitt method, it uses local differential and convolution to find the edges just like the Robert method. This one is best for gradual grayscale change or noisy images, but it couldn’t get rid of fake edges like the others.

Then, there are some derived methods from the methods mentioned above. The LOG method first uses the Gaussian filter to get the processed image and subsequently computes the second derivative to locate the edges where there is any zero-crossing. Since it uses the Gaussian filter to reduce the noise, it could complete a decent job in some noisy pictures at the cost of loss of details and precision. The DOG method uses the difference of two Gaussian functions to determine the edges and could be used to approximate the LOG operator. The Canny method is now the optimal algorithm for edge detection.
It first reduces the noise, then compute the gradient and finally identify the edges with a high threshold value and a low threshold value. So, it can be used in a variety of cases and it has good performance for noisy images.

Besides those traditional methods, some advanced and more complex methods are also invented to better improve the edge detection results involved in various applications, such as the Convolution Neural Network (CNN) [1] which uses machine learning to train a model for identifying different parts of the image. The Backward Propagation Network Based on Genetic Algorithm [4], the Deep Semantic Edge Learning Architecture Based on ResNet [5] and a New Skip-layer Architecture and Medical Ultrasound Image Edge Detection Based on Machine Learning [6] are some novel examples. This kind of edge detection model usually possesses a strong resistance to different noises and could evolve during the training process. As a result, it could be increasingly accurate while it is fed with more data or instances. However, the training process could be a mess given the uncertainty of the parameters and the complexity of the network. Sometimes, it could even be backward evolving if any specific parameter is inappropriately modified. Besides this shortcoming, the complexity of configuring an appropriate network is daunting. Thus, these disadvantages make it impractical to apply neural network edge detection methods to cases in real life.

At length, it becomes ideal to use traditional methods, and the Canny method, possessing more flexible features and potent qualities than the others, automatically turns into the best solution for the practical applications. We will then explore the Canny method and its possible improvements and limitations.

In this paper, we primarily focus on the parameters of the Canny method and the Sobel method that have a significant impact on the results and conduct some comparisons. There is also an additional exploration of the improvements brought by the Gaussian filtering.

The remainder of this paper is organized as the followings. In Section 2, we briefly analyze the algorithms of the Canny method and the Sobel method. Then in Section 3, we carry through the experiments. Finally, in Section 4, we come out of a conclusion for this paper.

2. EDGE detection methods

As a multi-level edge detection algorithm, the Canny method was invented by John F. Canny in 1986. It was a milestone since it founded the computational theory of edge detection to explain how edge detection actually works. The goal of this method is to find the most optimal solution with three main standards. The first is a good detection, which means the algorithm should detect as many real edges as possible, the second is an accurate location, which represents the strong relation between detected edges and actual edges, and the last is a minimal response, which indicates that the edges in the image can only be marked once and cannot come from noises. The Canny method utilizes the Calculus of Variations to fulfill the above standards.

The exact processes involve three main steps. First, the original image needs to be sifted out of the noises. Usually, the Gaussian filter is selected and used to convolve with the original image to get a blurred version of the original one as the formulae below show where \( B \) is the blurred version and \( A \) is the original image.

\[
B = \frac{1}{159} \begin{bmatrix}
2 & 4 & 5 & 4 & 2 \\
4 & 9 & 12 & 9 & 4 \\
5 & 12 & 15 & 12 & 5 \\
4 & 9 & 12 & 9 & 4 \\
2 & 4 & 5 & 4 & 2
\end{bmatrix} \ast A, \quad (1)
\]

\[
H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-(i-(k+1))^2+(j-(k+1))^2}{2\sigma^2}\right); \quad i \leq i, j \leq (2k + 1), \quad (2)
\]

where \( H_{ij} \) denotes a specific entry in the resulting kernel, \( i \) and \( j \) denote the coordinates of the entry, \( \sigma \) denotes the picked parameter and the kernel is of the size \( (2k + 1) \). As a result, the individual pixel noise could impose minimal influence on the detection process.
Then, analyzing the gradients is essential since every edge in the image could be in a different direction, so in the Canny method, it uses four different masks to detect horizontal, vertical and diagonal edges. Every result from the convolution of each mask and the original image is saved, and for each point, we mark the maximum value and the direction of the generated edge. After it, we have the luminance gradient chart and its direction for each point.

For the last step, a relatively high luminance gradient might be the edge, but the threshold value is still unknown to correctly classify the edges and the non-edges, so the Canny method uses the Hysteresis Threshold. Hysteresis threshold needs two values—a high threshold value and a low threshold value. Supposed that the edges are continuous curves, we can thus start with a high threshold value to mark the edges that we are certain about, and then we can use these real edges and their direction information to trace the whole and complete contour. While tracing the complete edge, we can make use of a low threshold value to follow the vague parts of the curves until we are back to the initial point. Finally, we have a binary image to tell us whether the given point belongs to an edge or not.

To be more precise and explicit, the process can be summarized as conducting some comparisons based on the following graph.

Figure 1. Graph for Hysteresis Threshold Process

In the graph, there are eight directions represented as capital directional abbreviations, and P, P1 and P2 are pixel points. First, in order to make the edges slimmer and more concrete, we need to apply the non-maximum value suppression using the gradient linear interpolations \( G_{P1} \) and \( G_{P2} \), which at points P1 and P2 respectively are:

\[
\tan(\theta) = \frac{G_x}{G_y}, \quad (3)
\]

\[
G_{P1} = (1 - \tan(\theta)) \times E + \tan(\theta) \times NE, \quad (4)
\]

\[
G_{P2} = (1 - \tan(\theta)) \times W + \tan(\theta) \times SW, \quad (5)
\]

where \( G \) is the magnitude of the gradient to quantify the speed with which the pixel changes its luminance and subscripts \( x \) and \( y \) stand for the horizontal component and vertical component.

Nevertheless, the edges extracted from the above are usually dispersive, which is undesirable. Hence, there exists the need to further explore the possibilities and come up with some solutions to correctly and accurately separate the continuous edges from the background. Some improved versions of the method involve calculating the second derivative at the direction of the gradient to find the zero-crossings, which could help achieve sub-pixel precision. The way to compute it is listed below, where \( L \) stands for Laplacian and the subscripts \( x \) and \( y \) stand for the horizontal component and vertical component.
Moreover, its third directional derivative along with the gradient direction must satisfy the following inequity.

\[ L_x^3 L_{xxx} + 3L_x^2 L_y L_{xxy} + 3L_x L_y^2 L_{xyy} + L_y^3 L_{yyy} < 0, \] (7)

Then we could get continuous edges and consequently don’t need extra edge tracking using threshold values.

Another alternative edge detection method is the Sobel method or the Sobel operator, which technically is a discrete differential operator, computing an approximation of the gradient of the image intensity function.

The Sobel operator includes two 3x3 matrices, one horizontal and one vertical. These two matrices are convolved with the original image to calculate approximations of derivatives of luminance along the horizontal direction and vertical direction. Then we could get two images \( G_x \) and \( G_y \), which at each point contain the vertical and horizontal derivative approximations from the original image \( A \) as the followings,

\[
G_x = \begin{bmatrix}
-1 & 0 & +1 \\
-2 & 0 & +2 \\
-1 & 0 & +1
\end{bmatrix} \ast A, \quad (8)
\]

\[
G_y = \begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
+1 & +2 & +1
\end{bmatrix} \ast A, \quad (9)
\]

The x-coordinate is increasing from left to right, and the y-coordinate is increasing from up to down. At each point, the magnitude of the gradient \( G \), which is the same as the \( G \) in the Canny method, can be obtained from these approximations with:

\[
G = \sqrt{G_x^2 + G_y^2}, \quad (10)
\]

We could also calculate the gradient’s direction \( \theta \) with:

\[
\theta = \arctan\left(\frac{G_y}{G_x}\right), \quad (11)
\]

For example, the result 0 is a vertical edge which is lighter on the right side.

The Sobel operator has some natural advantages over the other edge detection operator. The first of them is that it decreases the blurring of the edges since it executes weighting operation for the influences of pixels’ positions. Second, due to the form of the filter operator, it could finish the convolving calculation in a short period, so it is fast and effective for edge extracting. Because of this easy-to-use feature, the Sobel operator is widely applied in edge detection.

However, on account of the fact that the Sobel operator is not based on the grayscale of the image and doesn’t strictly follow the physiological characteristic of the human visual system, it could not clearly separate the subjects of the image from the background. As a result, the extracted edges are not always satisfying and accurate.

3. Experiment and result

To test the different performance of existing edge detection methods such as the Canny method and the Sobel method, we experiment with a cartoon image as Figure 2 to find the better method for detection characters’ contour.

With the two edge detection methods, we are going to explore the influence some of the parameters and processes impose on the experiment outcome, and we will find a better solution with proper parameters when extracting the overall contour of the cartoon characters.
Figure 2. Original Image

Starting with the Canny method on the sample image, the kernel size is initially 3x3 with the high threshold value being 150 and the low threshold value being 50 as in Figure 3(a). We could clearly see the overall contour is clear and the redundant edges are not too many. However, when the kernel size is increased to 5x5, the primary edges are overshadowed by the irrelevant edges, and the textures on the clothes are exaggerated and magnified, which makes it impossible to clearly identify the contour of the clothes. When the size is further increased to 7x7, the whole image is barely legible. This result is definitely unacceptable as the analyzed data fed to the computer to accomplish visionary tasks.

Figure 3. Effects of Kernel Size on the Performance of the Canny Edge Detector

Another factor limiting the final result of the Canny method is the threshold values. When the high threshold value is set to 250, there is a drastic drop of detected edges as in Figure 4(a) with the same 5x5 kernel. However, the overall shape of the objects and the contour remain clear and unbroken. When the low threshold value is increased to 85 while keeping the high threshold value at 150, there is also a decrease of irrelevant edges and the disturbance as in Figure 4(b) compared to Figure 3(a).
Subsequently, the Sobel operator starts with a 3x3 kernel and then executes the detection along the $x$-direction as in Figure 5(a) and $y$-direction as in Figure 5(b) separately. Last, we combine the two processed images into one edge detection result as in Figure 5(c).

We can tell that for the edge detection result along the $x$-direction, the vertical outlines of the character’s face are better recognized than the result along the $y$-direction, which is reasonable since the vertical outlines have greater gradient change along the $x$-direction. As for Figure 5(b) which is the result of $y$-direction detection, the toy stick in the character’s mouth is depicted more clearly, on account of the fact that it mostly consists of horizontal lines. At last, the final result is synthesized from the two images to give the complete edges.

There is also a parameter that needs to be carefully selected, which is the kernel size of the Sobel operator. With size 5x5 as in Figure 6, we get a relatively different result, and it is far from being a decent and recognizable outcome. If we look close enough, we could find an inconspicuous word above the character in the middle, of which we couldn’t find any trace in the original image.
From the result of the Canny method with a kernel size of 3x3 and the combined result of the Sobel method, we can identify some characteristics. One of them is the irrelevant edges detected from both methods. Since the Sobel method is directly using the gradients to find the steepest curve of the luminance, and the result is not reworked using secondary algorithms, its result largely reserves the overall information from the original image. On the contrary, the irrelevant edges in the Canny method are more obvious for the reason that the result is amplified after the process of hysteresis threshold, and they are reserved because they are just above the low threshold value. However, with proper adjustment of the low threshold value, we can eliminate a great number of irrelevant edges. Overall, the Canny method suppresses the noise better on account of the fact that it bates much more unconcerned pixel points on the clothes of the characters, the objects in their hands and mouth, and the glasses they wear as in the original image Figure 2. In a word, both methods could extract the overall contour, but the Canny method could have better performance on the suppression of noise.

We can certainly expect some improvements in the characters’ clothes from these extracted images. There is too much noise among the edges, and that noise comes from the texture of the jeans. So, we can add a Gaussian filter before the actual detection happens.

First with the Canny method, initially the Gaussian filter kernel is 3x3 as in Figure 7(a). We could clearly find that when the kernel used in the Gaussian filter is 5x5 as in Figure 7(b), the noises on the clothes are much less than those when the kernel is 3x3 as in Figure 7(a). But when the kernel size is further enlarged to 7x7, the overall contour becomes incomplete and some edges are missing as in Figure 7(c). It is clear that when the size of the kernel continues to increase, the contour will become more fragmentary. Moreover, the titles on the upper right corner reveal that the larger 5x5 kernel has better quality and less noise inside the characters among the three images. Besides, the effect of increasing the Gaussian filter size is significantly better than that of increasing the low threshold value on the noisy edges of the figure’s clothes, since the Gaussian filter possess an inner advantage when dealing with small irrelevant noises. As a result, a properly selected size of the Gaussian filter could contribute to the final edge detection quality.
Then to study how differently sized Gaussian filters influence the results of the Sobel method, we add a 5x5 Gaussian filter before the detection, trying to smooth out the noise on the clothes. The result is shown in Figure 8(a). Compared to Figure 5(c), the noise on the clothes is faded to some extent while keeping the overall contour complete, but it still leaves some traces and marks behind. When we increase the Gaussian filter to the size of 11x11 as in Figure 8(b), the result is further blurred and faded. Even though the noise is nearly invisible, some edges and details are also less obvious. We can conclude that if we keep increasing the size of the filter, the contour of the characters is also cast-off along with the noise on the clothes.

At length, each method has its own features and characteristics. For the Canny method, it has better performance on the reduction of noise, but its parameters require special attention and care to modulate. As for the Sobel method, the noise is nearly inevitable, which means it can only be suppressed and not completely eliminated at the cost of the overall detection quality. However, the Sobel method is easy and handy to use, and adjusting its parameters is more intuitive and perceptual. In a word, according to various application scenarios, we should try the possible methods and harness the suitable and proper one or ones with carefully picked parameters to obtain the optimal result.

4. Conclusions
In this paper, we explore the effects of several main parameters for the Canny method and the Sobel method and compare the results of the two edge detection methods. In total, the Canny method has better quality and less noise for the extracted edges and the Sobel method possesses better performance of its computation time and better overall shape. Moreover, the main parameters include the kernel size, threshold values, and the Gaussian filter size. Each result is different but somehow interconnected. Therefore, to get the optimal quality and performance, we need to choose the right method with fine-tuned parameters to fit various application scenarios. In the future, there is still some possible progress for the edge detection method that can be rather groundbreaking, such as the learning-based methods, and we will study it afterward for better performance.
References

[1] Dr. Krishna Raj, Pragya Gautam, Kapil Dev Tiwari, Vipul Goel, A Review Paper: On Various Edge Detection//International Journal for Research in Applied Science & Engineering Technology Volume 5 Issue VIII, August 2017

[2] Ahmad Sabry Mohamad, Nur Syahirah Abdul Halim, Muhammad Noor Nordin, Roszymah Hamzah, Jameela Sathar, Automated Detection of Human RBC in Diagnosing Sickle Cell Anemia with Laplacian of Gaussian Filter//14-15 Dec. 2018, IEEE

[3] Xinbo Gao, Yuming Fang, Jinjian Wu, Shiqi Wang, Leida Li, Yu Zhou, No-Reference Quality Assessment for View Synthesis Using DoG-Based Edge Statistics and Texture Naturalness//Volume: 28 , Issue: 9 , Sept. 2019, IEEE

[4] Zhengquan He, M Y. Siyal, Edge Detection with BP Neural Networks//Singapore: IEEE, 1998: 1382-1384

[5] Zhiding Yu, Chen Feng, Ming-Yu Liu, Srikumar Ramalingam, CASENet: Deep Category-Aware Semantic Edge Detection//IEEE

[6] Soman A.K., Vaidyanathan P.P., Linear Phase Paraunitary Filter Banks: Theory, Factorizations and Designs//IEEE, Transactions on Signal Processing: A Publication of the IEEE Signal Processing Society, 1993,12(12)