www.icgst.com

Embedded Smoothing Processing in Spatial Gradient Operators, a Comparative Study

Tieling Chen
University of South Carolina Aiken, 471 University Parkway, Aiken, SC 29801, USA
tielingc@usca.edu

Abstract
This paper analyzes the embedded smoothing processing in some commonly used gradient operators used in edge detection, including Prewitt's operator, Sobel's operator, Canny's detector and a wavelet transformation based method. The effects of the embedded smoothing processing on edge detection are studied and compared. The paper also reduces the design of a gradient operator to the design of one or two one dimensional smoothing functions.

Keywords: *smoothing, gradient operator, convolution, edge detection.*

1. Introduction
Gradient operators are commonly used to detect abrupt changes of grey level values in digital images. A gradient operator consists of two partial derivatives, which determine the magnitude and the direction of the gradient. Since a digital image is represented by a discrete function, the partial derivatives can only be approximated. Generally, the partial derivatives are implemented as a pair of convolution masks, and most of them contain smoothing processing on the original image. As introduced in [4, 6], commonly used gradient operators include Prewitt's operator for horizontal and vertical edges, consisting of the partial derivative masks

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \qquad (1)$$

and Sobel's operator for horizontal and vertical edges, consisting of

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}. \qquad (2)$$

The above two gradient operators actually carry smoothing processing that may influent the computation of the gradient.

Prewitt and Sobel also have masks to detect diagonal edges. They are

$$\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \text{ for Prewitt, and}$$

$$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \text{ for Sobel.}$$

These two pairs can be obtained from (1) and (2) by rotations and they form two gradient operators with embedded smoothing processing. Since the paper studies the changes of grey level values along the horizontal and the vertical directions and decompose a derivative mask into a horizontal row mask and a vertical column mask, the masks detecting changes of the grey level values along the diagonal directions are not considered in the paper.

Canny's detector uses a three-stage algorithm to detect edges [1]. At the first stage, the original image is smoothed by convolving a two dimensional Gaussian function with a proper variance. At the second stage, masks detecting the horizontal, vertical and sometimes the diagonal edges are used to compute the gradient of the smoothed image. Edges are then traced at the third stage. Canny's detector is variously implemented. Some implementations of the second stage may slightly violate Canny's original intention if the above gradient masks are used because the embedded smoothing processing in these masks interferes the Gaussian smoothing processing used at the first stage. This slight variation has been ignored all along. We will check this influence in the next section.

Another important edge detection method is one that uses a wavelet transformation [3, 5]. In the one dimensional case, a smoothing function can derive a wavelet function. The wavelet transformation of a given signal is proportional to the derivative of the smoothed signal that is smoothed by the smoothing function. When this is generalized to the two dimensional case, the partial derivatives of a two dimensional smoothing function give two wavelet functions. The wavelet transformation of an image is then a vector with two components, and this vector is in fact proportional to the gradient of the smoothed image. Since the wavelet function used in the method is derived from a smoothing function, the corresponding smoothing processing is embedded in the gradient operator.

Smoothing processing is almost always involved in spatial gradient operators. The smoothing processing is determined by the corresponding smoothing function, which plays an essential role in the design of a gradient operator. For Prewitt's or Sobel's gradient operator, each partial derivative mask smoothes the image in a different way that is in favor of edges along some specific direction. Hence, these operators are not really gradient operators according to the strict mathematical meaning.

In the paper we use the direct gradient operator without any embedded smoothing processing to compare with some commonly used gradient operators and then find the effects of their embedded smoothing processing. The design of a gradient operator is then reduced to the design of one or two smoothing functions.

The organization of the rest of the paper is as follows: In section 2, we first introduce the direct gradient operator without any embedded smoothing processing as the basic to be compared with other gradient operators. Then we check Sobel's and Prewitt's gradient operators to find the corresponding smoothing processing imposed on the direct gradient operator. And then the influences of Sobel's and Prewitt's gradient operators on Canny's detector are studied. The section finally introduces a method of designing a gradient operator with embedded smoothing processing. In section 3, we show experimental results on comparing some gradient operators with different embedded smoothing processing. Some general conclusions are given in section 4.

2. Embedded Smoothing Processing

A. Direct gradient operator

There are several ways to define the first order derivative of a one dimensional discrete function. Each way is an approximation of the definition in the continuous system. They might cause a little difference in applications, but the difference is negligible when we detect edges of objects bigger than a certain size, basically those objects causing visual interest in images. Suppose $f(x)$ is a one dimensional discrete function and we use $Df(x)$ to denote its derivative at point x. Some commonly used definitions of $Df(x)$ includes

$$Df(x) = f(x) - f(x-1),$$
$$Df(x) = f(x+1) - f(x),$$

and

$$Df(x) = (f(x+1) - f(x-1))/2.$$

Since the two dimensional generalizations of the first two versions do not give convolution masks with odd-value lengths, they are rarely put into use when implementing the partial derivative masks. For convenience, we drop the coefficient in the third version and define the derivative of a discrete function as

$$Df(x) = f(x+1) - f(x-1).$$

For an image represented by a two dimensional discrete function $f(x, y)$, its partial derivatives are similarly defined as

$$D_x f(x, y) = f(x+1, y) - f(x-1, y),$$

and

$$D_y f(x, y) = f(x, y+1) - f(x, y-1).$$

The mask forms of the partial derivatives are

$$D_x = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \text{ and } D_y = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}.$$

Therefore the partial derivatives of an image $f(x, y)$ can be obtained by the following convolutions:

$$D_x f(x, y) = (D_x * f)(x, y)$$

and

$$D_y f(x, y) = (D_y * f)(x, y).$$

We call the gradient operator consisting of D_x and D_y the direct gradient operator. No smoothing processing is embedded in the direct gradient operator.

B. Smoothing processing in Sobel's and Prewitt's gradient operators

If a two dimensional operator is decomposed into two separate one dimensional convolutions, we can go down to a deeper level to understand the effects of the operator on each variable. As we know, both Sobel's operator and Prewitt's operator are separable. For example, the partial derivative masks of Sobel's operator shown in (2) are both separable:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

and

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}.$$

In the above, the first separation shows that Sobel's partial derivative operation with respect to x is equivalent to smoothing the given image with respect to y first, with the row smoothing mask $[1 \quad 2 \quad 1]$, and then taking the derivative with respect to x. Of course we can also understand it as taking the derivative first and then performing the smoothing processing because the order does not matter. In the discussion that follows we will mention the smoothing processing first and then the derivative.

Similarly, the second separation shows that Sobel's partial derivative operation with respect to y is equivalent to smoothing the given image with respect to x first, with the column smoothing mask $[1 \quad 2 \quad 1]^T$, the transpose of $[1 \quad 2 \quad 1]$, and then taking the derivative with respect to y. We can expand the row mask and the column mask into the following square masks since this does not change anything in the convolutions:

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Clearly, with Sobel's operator the original image is smoothed in two different ways and the partial derivatives are not taken over the same smoothed image. The situation is quite similar for Prewitt's operator. The two masks in (1) are also separable,

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \times [1 \quad 1 \quad 1]$$

and

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \times [-1 \quad 0 \quad 1].$$

Each convolution is equivalent to a two-step operation that first smoothes the original image along one direction and then takes a partial derivative over the smoothed image along the perpendicular direction. The original image is smoothed in two different ways, with the following two different smoothing masks,

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

In both Sobel's and Prewitt's operators, the two partial derivatives are taken over two different smoothed images. Since the gradient is computed from two different sources, these operators are not strict mathematical gradient operators. They are approximations of the gradient operation that are in favor of horizontal and vertical edges.

C. Smoothing processing in Canny's detector

Canny's detector is an edge detection algorithm with three stages. At stage one the original image is smoothed by a Gaussian function and at stage two the gradient of the smoothed image is computed. At stage three the detector uses the information of the gradient obtained at stage two to extract edges. We only check the computation of the gradient and put the focus on the implementation.

There are many ways to implement Canny's detector, including using Sobel's operator to compute the gradient of the smoothed image. The implementation of the gradient of the smoothed image would deviate from Canny's original intention if a gradient operator carrying smoothing processing, such as Prewitt's operator or Sobel's operator, were used in the gradient computation. Because the Gaussian smoothing at stage one is interfered by the embedded smoothing processing in the gradient operator.

Sobel's operator is commonly used to compute the gradient of the smoothed image. Suppose f is the original image, g is the Gaussian function selected in the smoothing processing at stage one, and S_x is Sobel's derivative mask with respect to x. Then the derivative with respect to x found by Sobel's operator is $S_x * (f * g)$. We have

$$S_x * (f * g) = f * (g * S_x) = f * ((g * D_x) * [1 \quad 2 \quad 1])$$
$$= f * (D_x(g) * [1 \quad 2 \quad 1]) = D_x(f * g * [1 \quad 2 \quad 1]).$$

The final expression in the above shows that the derivative is actually taken after the original image is smoothed by Gaussian and the row mask $[1 \quad 2 \quad 1]$. Similarly if Sobel's operator is used, the derivative with respect to y is taken after the original image is smoothed by Gaussian and the column mask $[1 \quad 2 \quad 1]^T$.

Usually in implementations, the convolution operation does not unnecessarily flip one of its operands, as required by the strict mathematical definition. But in mathematical inductions the influence of the flip should be taken into considerations, especially when the mask is not symmetric. In the above mathematical induction, the mask D_x is flipped over in the convolutions.

On the four boundaries of a given image, we pad extra rows and columns of zeros such that the convolution can still be performed on the boundaries.

Now we check $D_x(g)$ with a concrete 3 by 3 Gaussian mask,

$$g = \begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix}. \qquad (3)$$

We have

$$D_x(g) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{-1}{16} & \frac{-2}{16} & \frac{-1}{16} \\ \frac{-2}{16} & \frac{-4}{16} & \frac{-2}{16} \\ 0 & 0 & 0 \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix}, \qquad (4)$$

where the mask D_x is flipped in the convolution. We can compute $D_y(g)$ in a similar way and obtain

$$D_y \begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix} = \begin{bmatrix} \frac{-1}{16} & \frac{-2}{16} & 0 & \frac{2}{16} & \frac{1}{16} \\ \frac{-2}{16} & \frac{-4}{16} & 0 & \frac{4}{16} & \frac{2}{16} \\ \frac{-1}{16} & \frac{-2}{16} & 0 & \frac{2}{16} & \frac{1}{16} \end{bmatrix}. \qquad (5)$$

If Sobel's operator is used at stage two, an extra convolution is performed on each mask, resulting in a pair of biased masks, $D_x(g)*[1 \quad 2 \quad 1]$ and $D_y(g)*[1 \quad 2 \quad 1]^T$, which are

$$\begin{bmatrix} \frac{-1}{16} & \frac{-4}{16} & \frac{-6}{16} & \frac{-4}{16} & \frac{-1}{16} \\ \frac{-2}{16} & \frac{-8}{16} & \frac{-12}{16} & \frac{-8}{16} & \frac{-2}{16} \\ 0 & 0 & 0 & 0 & 0 \\ \frac{2}{16} & \frac{8}{16} & \frac{12}{16} & \frac{8}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{4}{16} & \frac{6}{16} & \frac{4}{16} & \frac{1}{16} \end{bmatrix} \qquad (6)$$

and

$$\begin{bmatrix} \frac{-1}{16} & \frac{-2}{16} & 0 & \frac{2}{16} & \frac{1}{16} \\ \frac{-4}{16} & \frac{-8}{16} & 0 & \frac{8}{16} & \frac{4}{16} \\ \frac{-6}{16} & \frac{-12}{16} & 0 & \frac{12}{16} & \frac{6}{16} \\ \frac{-4}{16} & \frac{-8}{16} & 0 & \frac{8}{16} & \frac{4}{16} \\ \frac{-1}{16} & \frac{-2}{16} & 0 & \frac{2}{16} & \frac{1}{16} \end{bmatrix}. \qquad (7)$$

We can see the difference between the partial derivative masks shown at the right sides of equations (4) and (5) and the partial derivative masks shown in (6) and (7). To see this clearer we separate the above partial derivative masks into products of column masks and row masks. We only look at the mask at the right side of equation (4) and the mask in (6). The comparison of the other pair is similar.

The mask obtained by the direct gradient operator is separated as

$$\begin{bmatrix} \frac{-1}{16} & \frac{-2}{16} & \frac{-1}{16} \\ \frac{-2}{16} & \frac{-4}{16} & \frac{-2}{16} \\ 0 & 0 & 0 \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix} = \begin{bmatrix} -\frac{1}{4} \\ -\frac{2}{4} \\ 0 \\ \frac{2}{4} \\ \frac{1}{4} \end{bmatrix} \times \begin{bmatrix} \frac{1}{4} & \frac{2}{4} & \frac{1}{4} \end{bmatrix} \qquad (8)$$

and the mask obtained by Sobel's operator is separated as

$$\begin{bmatrix} \frac{-1}{16} & \frac{-4}{16} & \frac{-6}{16} & \frac{-4}{16} & \frac{-1}{16} \\ \frac{-2}{16} & \frac{-8}{16} & \frac{-12}{16} & \frac{-8}{16} & \frac{-2}{16} \\ 0 & 0 & 0 & 0 & 0 \\ \frac{2}{16} & \frac{8}{16} & \frac{12}{16} & \frac{8}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{4}{16} & \frac{6}{16} & \frac{4}{16} & \frac{1}{16} \end{bmatrix} = \begin{bmatrix} -\frac{1}{4} \\ -\frac{2}{4} \\ 0 \\ \frac{2}{4} \\ \frac{1}{4} \end{bmatrix} \times \begin{bmatrix} \frac{1}{4} & \frac{4}{4} & \frac{6}{4} & \frac{4}{4} & \frac{1}{4} \end{bmatrix}. \qquad (9)$$

The column masks at the right hand side of both (8) and (9) are the same, which is a smoothed derivative produced by smoothing $[-1 \quad 0 \quad 1]^T$ with $\begin{bmatrix} \frac{1}{4} & \frac{2}{4} & \frac{1}{4} \end{bmatrix}^T$. But the row masks are different. We normalize the row mask in (9) to $\begin{bmatrix} \frac{1}{16} & \frac{4}{16} & \frac{6}{16} & \frac{4}{16} & \frac{1}{16} \end{bmatrix}$ such that the sum of the entries is one. Compared with the row mask $\begin{bmatrix} \frac{4}{16} & \frac{8}{16} & \frac{4}{16} \end{bmatrix}$ that is same as the one in (8), the mask $\begin{bmatrix} \frac{1}{16} & \frac{4}{16} & \frac{6}{16} & \frac{4}{16} & \frac{1}{16} \end{bmatrix}$ takes a portion of the weight on the center pixel and evenly assigns the weight to the two pixels that are one pixel apart from the center. Obviously this smoothing mask is in favor of the horizontal edges.

To implement Canny's detector correctly, derivative masks involving smoothing processing such as the masks in Sobel's operator and Prewitt's operator should not be used. When a Gaussian function is selected, a smoothing mask can be obtained by sampling the Gaussian function. Then the smoothed image convolves D_x and D_y directly to compute its partial derivatives and then the gradient. Nevertheless, if derivative masks involving smoothing processing are used some additional effects may be obtained, such as edges along some specific directions are strengthened. But these effects are extra to Canny's operator.

D. Design gradient operators

From the above discussion we find that the embedded smoothing processing in a gradient operator is the characteristic of the operator. For each separable partial derivative mask of a gradient operator, the smoothing processing can be decomposed into row smoothing and column smoothing. One of them is used to smooth the corresponding derivative. Since derivative is sensitive to noise, proper smoothing is good for the computation. But too much smoothing would attenuate the difference among pixel values such that the derivative becomes dull. Hence the row smoothing and the column smoothing do not have to be the same. The original image may not be smoothed the same way when computing the partial derivatives, as in Prewitt's and Sobel's operators; therefore the operator is an approximated gradient operator.

The design of a gradient or an approximated gradient operator is reduced to the design of one or two smoothing functions. The idea can be used in a wavelet transformation based gradient operator. We can start the operator with a one dimensional smoothing function. A smoothing function $\theta(x)$ is differentiable, even and decreasing in $|x|$, has a finite support and integral one. Its derivative $\phi(x)$ gives a wavelet function. By dilating a smoothing function $\theta(x)$ with a dilation scale s one can get a so-called dilated smoothing function $\theta_s(x)$, defined as

$$\theta_s(x) = \frac{1}{s}\theta\left(\frac{x}{s}\right).$$

This leads to a corresponding dilated wavelet function $\phi_s(x) = \phi(x/s)/s = s\left(\theta_s(x)\right)'$. The wavelet transformation of a function f at dilation level s, denoted by $(W_s f)(x)$, is given by the following convolution:

$$(W_s f)(x) = (f * \phi_s)(x).$$

Since

$$(f * \varphi_s)(x) = f * \left(s\frac{d}{dx}\theta_s(x)\right) = s\frac{d}{dx}(f * \theta_s)(x),$$

the wavelet transformation is actually proportional to the derivative of the smoothed function. From the one dimensional smoothing function, we can make a separable two dimensional function that can also be dilated, $\Theta_s(x,y) = \theta_s(x)\theta_s(y)$. Its partial derivatives give two wavelet functions, dilated at scale level s. The wavelet transformation of $f(x,y)$ at scale level s is a vector with two components,

$$(W_s f)(x,y) =$$
$$\left(\left((f(\cdot,y)*\phi_s)(x)*\theta_s\right)(y), \left((f(x,\cdot)*\phi_s)(y)*\theta_s\right)(x)\right),$$

which is equal to

$$s\left(\frac{d}{dx}(f*\Theta_s)(x,y), \frac{d}{dy}(f*\Theta_s)(x,y)\right).$$

Hence the wavelet transformation is equivalent to a gradient operator. This method heavily depends on the design of the smoothing function $\theta(x)$ [2]. A good smoothing function makes the method well meet Canny's criteria for edge detection algorithms [1]. The evolution of the wavelet transformation across dilation scales can be used to characterize different types of edges. In implementations, once a smoothing function is chosen, the dilated smoothing function and the corresponding dilated wavelet function can be sampled dynamically at different scales.

To strengthen the edges along specific directions, the two factors $\theta_s(x)$ and $\theta_s(y)$ of the two dimensional smoothing function do not have to be dilated to the same scale level. For example, to strength the edges along the x direction we can dilate $\theta_s(y)$ to a higher level and keep the dilation of $\theta_s(x)$ at a lower level. The things are just opposite when edges along the y direction are to be strengthened. In this way, the partial derivatives are not taken over the same smoothed image and the method is regarded as an approximation of the gradient method. It is also feasible to use two different one dimensional smoothing functions ξ and η to build two dimensional smoothing functions $\Theta_1(x,y) = \xi(x)\eta(y)$ and $\Theta_2(x,y) = \eta(x)\xi(y)$, which are used in separate partial derivatives. One of ξ and η handles smoothing the derivative and the other handles pure smoothing on the image.

In general, to design partial derivative masks of a gradient operator we can start with a two dimensional smoothing mask. The smoothing mask does not have to be separable. The partial derivative mask with respect to x is given by convolving the smoothing mask with the column mask $\begin{bmatrix}1 & 0 & -1\end{bmatrix}^T$, the flipped D_x, and the partial derivative mask with respect to y is given by convolving the smoothing mask with the row mask $\begin{bmatrix}1 & 0 & -1\end{bmatrix}$, the flipped D_y. The resulting pair is just a gradient operator.

But we have more flexibility if we design separable masks since we can start with one dimensional smoothing masks. Let us use the smoothing mask $\begin{bmatrix}1 & 3 & 4 & 3 & 1\end{bmatrix}$, sampled from a Gaussian function, to illustrate the design procedure. For convenience, the denominator 12 of each entry is dropped. This mask covers a bigger neighborhood of the center than the smoothing mask $\begin{bmatrix}1 & 2 & 1\end{bmatrix}$ embedded in Sobel's operator. The derivative of the mask is

$$\begin{bmatrix}1 & 3 & 4 & 3 & 1\end{bmatrix} * \begin{bmatrix}1 & 0 & -1\end{bmatrix} = \begin{bmatrix}-1 & -3 & -3 & 0 & 3 & 3 & 1\end{bmatrix}.$$

Then the two partial derivative masks of the gradient operator are given by

$$\begin{bmatrix}-1 & -3 & -3 & 0 & 3 & 3 & 1\end{bmatrix}^T \times \begin{bmatrix}1 & 3 & 4 & 3 & 1\end{bmatrix} \quad (9)$$

and

$$\begin{bmatrix}1 & 3 & 4 & 3 & 1\end{bmatrix}^T \times \begin{bmatrix}-1 & -3 & -3 & 0 & 3 & 3 & 1\end{bmatrix}. \quad (10)$$

Each of the above separable partial derivative masks is a product of a column matrix and a row matrix. Then a convolution with the derivative mask is reduced to two separate convolutions, one of which handles the smoothing processing along one direction and another handles a smoothed derivative along the perpendicular direction. In many situations we want the derivative to be more sensitive to changes so we do not smooth the derivative too much. In the extreme case, each partial derivative is not smoothed, we get

$$\begin{bmatrix}-1 & 0 & 1\end{bmatrix}^T \times \begin{bmatrix}1 & 3 & 4 & 3 & 1\end{bmatrix}$$

and

$$\begin{bmatrix}1 & 3 & 4 & 3 & 1\end{bmatrix}^T \times \begin{bmatrix}-1 & 0 & 1\end{bmatrix},$$

which gives a generalized Sobel operator, with the partial derivative masks

$$\begin{bmatrix}-1 & -3 & -4 & -3 & -1\\ 0 & 0 & 0 & 0 & 0\\ 1 & 3 & 4 & 3 & 1\end{bmatrix} \text{ and } \begin{bmatrix}-1 & 0 & 1\\ -3 & 0 & 3\\ -4 & 0 & 4\\ -3 & 0 & 3\\ -1 & 0 & 1\end{bmatrix}. \quad (11)$$

3. Experimental Results

Now we examine the effects of the gradient operators discussed in the paper with Lena's image in Figure 1. For each operator, only the gradient are checked and further processing to extract edges is not under our considerations. To compare the effects of different operators, the grey level values of each gradient image are rescaled to the range from 0 to 255. When finding the difference between two images, we subtract the second image from the first image and then shift the grey level values of the result to a non-negative range. We call this normalizing the difference.

We will make various comparisons. We will check the differences among Sobel's operator, Prewitt's operator, a wavelet transformation based method, and the operator with direct partial derivatives. We will also check the difference between two different implementations of Canny's detector. Finally we will check the effects of the operators studied in section 2. D.

Figure 1. Lena's image.

Let us compare the direct gradient operator with Sobel's operator first. Figure 2 (a) is the norm of the gradient of the original image produced by the direct gradient operator. Figure 2 (b) is the corresponding gradient image produced by Sobel's operator. To give clear appearances, Figure 2 (a) and Figure 2 (b) are negated.

The difference between these two gradient images is almost invisible with naked eyes. The pixel by pixel comparison, without negation, is shown in Figure 2 (c). The difference is normalized. A bright pixel means the grey level value at the same pixel in the gradient image corresponding to Figure 2 (a) is bigger than that corresponding to Figure 2 (b). A dark pixel means the opposite. The difference is mainly because the edges produced by Sobel's operator are shifted by the smoothing processing of the operator.

Prewitt's operator shows a similar effect. Figure 3 is the normalized difference between the direct gradient image corresponding to Figure 2 (a) and the gradient image obtained by Prewitt's operator. The difference is mainly resulted from the shifts caused by the smoothing processing in Prewitt's operator.

We use the cubic spline smoothing function

$$\theta(x) = \begin{cases} 8(x+1)^3/3 & -1 \le x < -1/2; \\ -8x^3 - 8x^2 + 4/3 & -1/2 \le x < 0; \\ 8x^3 - 8x^2 + 4/3 & 0 \le x < 1/2; \\ 8(1-x)^3/3 & 1/2 \le x < 1; \\ 0 & \text{otherwise.} \end{cases}$$

to derive a wavelet function and then dilate it to the

(a)

(b)

(c)

Figure 2. (a) The negated gradient image produced by the direct differentiation without smoothing. (b) The negated gradient image produced by Sobel's operator. (c) The normalized difference between the two gradient images.

Figure 3. The normalized difference between the gradient image corresponding to Fig 2 (a) and the gradient image obtained by Prewitt's operator.

Figure 4. The normalized difference between the gradient image corresponding to Fig 2 (a) and the gradient image obtained by a wavelet transformation based method.

scale level $s = 3$. The wavelet transformation is then performed on the image in Figure 1 to find its gradient. The normalized difference between the direct gradient image corresponding to Figure 2 (a) and the gradient image obtained by the wavelet transformation is shown in Figure 4. Edge shifts caused by the embedded smoothing processing is still the main reason of the difference.

For Canny's detector, we use mask (3) to smooth the original image at the first stage. We show the difference between two implementations of the gradient. One implementation is with the direct gradient operator without any extra smoothing. Another one is with Sobel's operator, which carries extra smoothing processing. The normalized difference between the gradient images is quite slight. With naked eyes, the difference between the two gradient images is almost unnoticeable. To see the difference clearly, the normalized difference is negated, shown in Figure 5. The real difference is small in grey level values because the Gaussian smoothing dominates the smoothing in Sobel's operator. The slight difference on locations can also be seen. This is because the extra smoothing processing in Sobel's operator shifts the edges. In the implementations of Canny's detector, many people use Sobel's operator to compute the gradient. The slight difference shown in Figure 5 implies that the approximation is acceptable.

Figure 5. The negated normalized difference between two implementations of Canny's detector. One uses derivatives without smoothing to compute the gradient and the other one uses Sobel's operator to compute the gradient.

Next we look at the gradient operator consisting of the masks (9) and (10). The normalized difference between the gradient image produced by the direct gradient operator without smoothing and the gradient image produced by the gradient operator given by (9) and (10) is shown in Figure 6. The difference in grey level values is apparent since the derivatives in the operator given by (9) and (10) are dull because of the heavy smoothing. The shifts of edges are also clearly seen in the difference image, implying localization errors sacrificed to the smoothing processing.

Finally we check the generalized Sobel operator with partial derivative masks shown in (11). This operator smoothes the original image in two different heavily biased ways. The difference between the gradient image produced by the classical Sobel's operator and the gradient image produced by the generalized Sobel operator is given in Figure 7 (a). To show the difference more clearly, the image is negated. The normalized difference is slight. In Figure 7 (b), we also show the normalized difference between the gradient image produced by direct differentiation and the gradient image produced by the generalized Sobel operator. The image is negated too. The white and black edges imply shifts of edges caused by the biased partial derivatives.

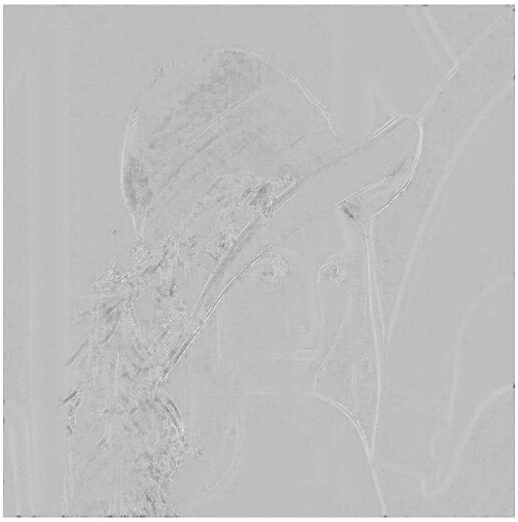
Figure 6. The normalized difference between two gradient images. One is produced by the direct gradient operator and the other one is produced by the partial derivatives shown in (9) and (10);

4. Conclusion

The paper studies the influences of the embedded smoothing processing in some commonly used gradient operators in the gradient computation. Generally a gradient operator carries smoothing processing, explicitly or implicitly, and the smoothing processing is the characterization of the operator. A separable partial derivative mask can be decomposed as a product of a two one dimensional masks. One of them handles smoothing the original image along one direction and the other one handles a one dimensional derivative along the perpendicular direction. Generally the one dimensional derivative is also smoothed. The smoothing function that smoothes the derivative does not have to be the same as what handles the smoothing processing. The design of a gradient operator is then reduced to the design of one or two smoothing functions.

Differences exist among gradient operators using different ways of smoothing and basically they lie in two aspects: grey level value differences and edge location differences. When the sizes of partial derivative masks are small, as in Sobel's operator, Prewitt's operator, or the direct gradient operator, these differences are not apparent. For Canny's operator, when computing the gradient of the image smoothed by a Gaussian function, the smoothing processing embedded in the gradient operator is mixed with the smoothing processing of Gaussian. If the sizes of the partial derivative masks of the gradient operator are small, like Sobel's operator and Prewitt's operator, the influence of the embedded smoothing processing is also small and probably can be ignored. If the sizes of the partial derivative masks are big, the influence of the embedded smoothing processing can not be ignored.

(a)



(b)

Figure 7. (a) The negated normalized difference between the gradient image produced by the classical Sobel's operator and the gradient image produced by the generalized Sobel's operator. (b) The negated normalized difference between the gradient image produced by direct differentiation and the gradient image produced by the generalized Sobel's operator.

5. References

[1] J. Canny, A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8(6), pp. 679-698, 1986.

[2] T. Chen, An Edge Detection Algorithm using Smoothing Functions, GVIP Journal, special issue: Edge Detection, 2006.

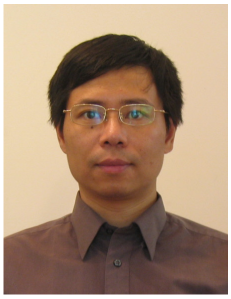
[3] T. Chen, J. Wang, and K. Zhang, A Wavelet Transform Based Method for Road Centerline Extraction, Photogrammetric Engineering & Remote Sensing, vol. 70, no. 12, pp. 1423-1431, 2004.

[4] R. Gonzalez and R. Woods, Digital Image Processing, 2nd Edition, Prentice Hall, pp. 577-581, 2002.

[5] S. Mallat and S. Zhong, Characterization of Signals from Multiscale Edges, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 4(7), pp. 710-732, 1992.

[6] M. Roushdy, Comparative study of Edge Detection Algorithms Applying on the Grayscale Noisy Image Using Morphological Filter, GVIP Journal, vol. 6(4), pp. 17-23, 2006.

Biography



Tieling Chen is an Assistant Professor in the Mathematical Sciences Department of the University of South Carolina at Aiken, USA, since 2002. He received a Ph. D. degree in Mathematics in 2001 and a M. S. degree in Computer Science in 2002 from the University of Western Ontario, Canada. His present research interest is mathematical operations in image processing.