EDGE DETECTION

U.J.Lin
Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C
Email address：r95942115@ntu.edu.tw

ABSTRACT

In this paper, first, I will give a brief concept about what the edge and corner is, and for what purpose of catching that information. Then I introduce several popular methods to detect edges and corners and give some experimental results. In early researches, usually the edge enhancement operators have been derived heuristically. We will see that an analytic approach to the design of these operators, and recent algorithms to detect the corners, edges, ridges, valleys, isolated dots, saddles, and plains. These are important physical features for the intensity distribution of an natural image. Besides, the accuracy in detecting these discontinuities and efficiency in implementing these operations are also quite important criteria for using an algotithm in the patch of computer vision.

1. INTRODUCTION

Local discontinuities in image luminance from one level to another are called luminance edges, which are considered as the boundaries between different textures. Since the edges for an image are always the important characteristics, they offer an indication for a higher frequency. The gray level corner is represented as the junction between two or more straight-line edges. Detection of edges and corners for one image may help for reducing the amount of data stream, and also help for well matching, such as image reconstruction and so on.

The most popular way of detection approach is differential detection which computes the gradient from two orthogonal orientations. If the gradient is sufficiently large, an edge along the specified axis is denoted, as in Fig. 1.

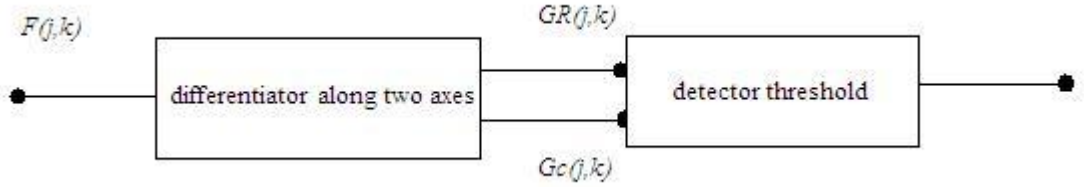

Fig. 1 Block diagram for detection

And as the definition of differentiation in continuous domain

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\Delta x \longrightarrow 0} \frac{f(x+\Delta x, y) - f(x,y)}{\Delta x} \tag{1}$$

since in discrete domain of a digital image, Δx is equal to one pixel, we have

$$\frac{\partial f(j,k)}{\partial j} = f(j+1,k) - f(j,k) = G_R(j,k)$$

$$\frac{\partial f(j,k)}{\partial k} = f(j,k+1) - f(j,k) = G_C(j,k) \tag{2}$$

where $G_R(j,k)$ denotes the gradient from row-axis, and $G_C(j,k)$ denotes the gradient from column-axis. Because of horizontal differentiation can only react on the variations along the vertical direction and vertical differentiation can only react upon the variations along the horizontal direction, we combine those two gradients together for a well detection. As a result we have a square-root gradient that combine those two directions, which is called a spatial gradient amplitude：

$$G(j,k) = \sqrt{G_R(j,k)^2 + G_C(j,.k)^2} \tag{3}$$

But we have pay attention to that the two orientations can be given at any orthogonal vectors with angles with respect to the row axis. The optimal orientation to detect always varies with each natural image.

Assume that there is an ideal smooth valley, as in Fig. 2, and define

$$G_R(j,k) = f(j,k+1) - f(j,k-1)$$
$$G_C(j,k) = f(j+1,k) - f(j-1,k) \tag{4}$$

As an example, the row gradient along the center row of and the column gradient along the center column of this smooth valley model are the following, as in Fig. 2.

$$0 \quad \frac{-a}{2} \quad \frac{-a}{4} \quad \frac{a}{2} \quad \frac{a}{4} \qquad\qquad \frac{-a}{2} \quad \frac{-a}{4} \quad \frac{a}{2}$$

Fig. 2 The gradient for row and column

If the threshold is on a/2, the edge is labeled on which gradient is equal to or over than a/2.

$$\begin{matrix} a & a & a & a & a & b & c \\ a & a & a & a & b & c & a \\ a & a & a & b & c & a & a \\ a & a & b & c & a & a & a \\ a & b & c & a & a & a & a \end{matrix}$$

Fig. 3 Smooth valley $b = \frac{a}{2}, c = \frac{a+b}{2}$

2. SEVERAL MODELS FOR DETECTION

The operators on the pixel neighborhoods that form the row and the column gradients can actually be expressed as the convolution relationships

$$G_R(j,k) = f(j,k) \otimes H_R(j,k)$$
$$G_C(j,k) = f(j,k) \otimes H_C(j,k) \quad (5)$$

Due to the convolution type, we have the 180° rotation upon the definition of the gradient as an impulse response. Larger size of the impulse response models will provide a smoother result because of the gradient impulse response always combine with noise removal impulse response. Such small luminance fluctuations will be ignored to increase the accuracy on detection.

As an example as the *Sobel* detector model, is a two by two box filter follow by a two by two gradient impulse response, as in Fig . 4.

$$H_R(j,k) = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

$$H_C(j,k) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

Fig. 4 *Sobel* detector model for column and row gradients

Both two impulse responses are actually convolution of a box filter for noise removal by the edge detector impulse response.

In Table 1, we have few operators that are 3x3 and 7x7 model size gradient detectors. Note that row operator and column operators are transpose relation to each other; as a result,

usually we only create one model that is enough on implementation of MATLAB. By computing the row gradient and column gradient, the spatial gradient magnitude is final result.

Table 1 Common impulse responses respect to their edge operators

OPERATOR	ROW GRADIENT DETECTOR	COLUMN GRADIENT DETECTOR
Separated pixel difference	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$
Roberts (diagonal difference)	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$
Prewitt	$\frac{1}{3}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$
Sobel	$\frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
7x7 truncated pyramid operator	$\frac{1}{34}\begin{bmatrix} 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 2 & 2 & 0 & -2 & -2 & -2 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 \\ 1 & 2 & 3 & 0 & -3 & -2 & -1 \\ 1 & 2 & 2 & 0 & -2 & -2 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \end{bmatrix}$	$\frac{1}{34}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 2 & 2 & 1 \\ 1 & 2 & 3 & 3 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -3 & -3 & -3 & -2 & -1 \\ -1 & -2 & -2 & -2 & -2 & -2 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$
7x7 boxcar operator	$\frac{1}{21}\begin{bmatrix} 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \end{bmatrix}$	$\frac{1}{21}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$

Now we consider the neighborhood Gaussian- shaped weighting functions as a means of noise suppression. It gives decreasing importance for pixels on neighborhood far from the center

$$g(x,s)=\frac{1}{\sqrt{2\pi s^2}}\exp\{-\frac{x^2}{2s^2}\} \tag{6}$$

The *Argyle* operator for horizontal gradient detector model can be expressed as a sampled version on the continuous domain impulse response, by taking the central pixel as zero index

$$H_R(j,k)=\begin{cases}-2g(x,s)g(y,t) & for\ x>0\\ 0 & for\ x=0\\ 2g(x,s)g(y,t) & for\ x<0\end{cases} \tag{7}$$

where *s* and *t* are two variances. The vertical impulse response function can be expressed in the same way. The *Macleod* operator horizontal gradient impulse response function is given by

$$H_R(j,k)=[g(x+s,s)-g(x-s,s)]g(y,t) \tag{8}$$

From previous discussion, we see that the gradient impulse response sometimes have the convolution form of a smooth filter by a differentiation filter. Since this concept and we know the larger size for the model can detect with lots of accuracy, we consider a large model that is compound with one gradient impulse response in the Table 1 and smooth filter of same size

$$H(j,k)=H_G(j,k)\otimes H_S(j,k) \tag{9}$$

3. CRITERIONS ON DETECTION

Although we see lots of gradient impulse response models, we still do not know how to choose appropriate one to detect given image. *Canny* has taken an analytic approach to the design of such operators, which is assumed that edge detection is performed by convolving a one-dimensional continuous domain noisy edge signal $f(x)$ based on three steps

1. Good detection

   The amplitude signal-to-noise ratio of the gradient is maximized to obtain a well accuracy.

$$\mathbf{SNR}=\frac{h_E S(h)}{\sigma_n}, \tag{10}$$

   where

   h_E denotes an amplitude of one-dimensional continuous domain model of step edge

   σ_n denotes a standard deviation of a plus additive white Gaussian noise

   $h(x)$ denotes the gradient impulse response function with the range $[-w,w]$

$$S(h) = \frac{\int_{-w}^{0} h(x)dx}{\int_{-w}^{w} [h(x)]^2 dx} \tag{11}$$

2. Good localization

The marked edge points should be as close to the center of the real edge as possible.

$$\mathbf{LOG} = \frac{h_E L(h)}{\sigma_n}, \tag{12}$$

where

$$L(h) = \frac{h'(0)}{\int_{-w}^{w} [h'(x)]^2 dx} \tag{13}$$

3. Single response

To improve the resolution of the edges, the distance between peak gradient amplitudes, denoted as x_m, is set to some fraction k of the width of gradient impulse response.

$$x_m = kw \tag{14}$$

Another approach is to compute gradients in a large number of directions with a set of template gradient impulse response arrays. Each gradient impulse response of template is rotational by the angle decided by the number of the edge directions to each other, as in Table 2. Then we choose the maximum magnitude of those gradients to involve in detection

$$G_i(j,k) = F(j,k) \otimes H_i(j,k)$$
$$G(j,k) = MAX\{|G_1(j,k)|, |G_2(j,k)|, \ldots\ldots, |G_m(j,k)|\} \tag{15}$$

Table 2 *Kirsch* template generation

EAST H1	NORTHEAST H2	NORTH H3	NORTHWEST H4
$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$
WEST H5	SOUTHWEST H6	SOUTH H7	SOUTHEAST H8
$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$	$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$

In a particular way, we can pick each *m* by *m* block of given image for template operating and choose each maximum gradient magnitude, since the directions of edges are always not same in entire image.

4. SECOND-ORDER DERIVATIVE EDGE DETECTION

As in the first-order derivative detection, we label edge on the point which gradient magnitude is over than some threshold is local maximum in the gradient magnitude. This can be detected by second-order derivative on a zero crossing. *Laplacian* is one of the second-order derivative types

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{16}$$

defined

$$G(x,y) = -\nabla^2\{F(x,y)\} = -\{\frac{\partial^2}{\partial x^2}F(x,y) + \frac{\partial^2}{\partial y^2}F(x,y)\} \tag{17}$$

The right side of the equation (17) is equal to zero if $F(x,y)$ is a constant or changing linearly in gradient magnitude. In the discrete domain, the simplest approximation to the continuous *Laplacian* is to compute the difference of slopes along each axis

$$\begin{aligned} G(j,k) = {} & [F(j,k+1) - F(j,k)] - [F(j,k) - F(j,k-1)] \quad for\ row \\ & + [F(j+1,k) - F(j,k)] - [F(j,k) - F(j-1,k)] \quad for\ column \end{aligned} \tag{18}$$

From this relation, we have the gradient impulse response

$$H = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{19}$$

One of the effective *Laplacian* responses is the *Laplacian of Gaussian* (LOG) edge detection operator. Since the *Laplacian* and the convolution are linear computation on the original image, we can put the *Laplacian* inside the convolution computation on the *Gaussian* operation

$$\begin{aligned} H(x,y) &= -\nabla^2\{g(x,s)g(y,s)\} \\ &= \frac{1}{\pi s^2}\left(1 - \frac{y^2}{s^2}\right)g(x,s)g(y,s) + \frac{1}{\pi s^2}\left(1 - \frac{x^2}{s^2}\right)g(x,s)g(y,s) \end{aligned} \tag{20}$$

This continuous domain LOG response can also be approximate closely by a *difference of Gaussian* (DOG) operator response

$$H(j,k) = g(j,s_1)g(k,s_2) - g(j,s_2)g(k,s_1) \tag{21}$$

which is often called the *Mexican hat filter* for the curve of the continuous domain response

that it is an even function, as Fig. 5.

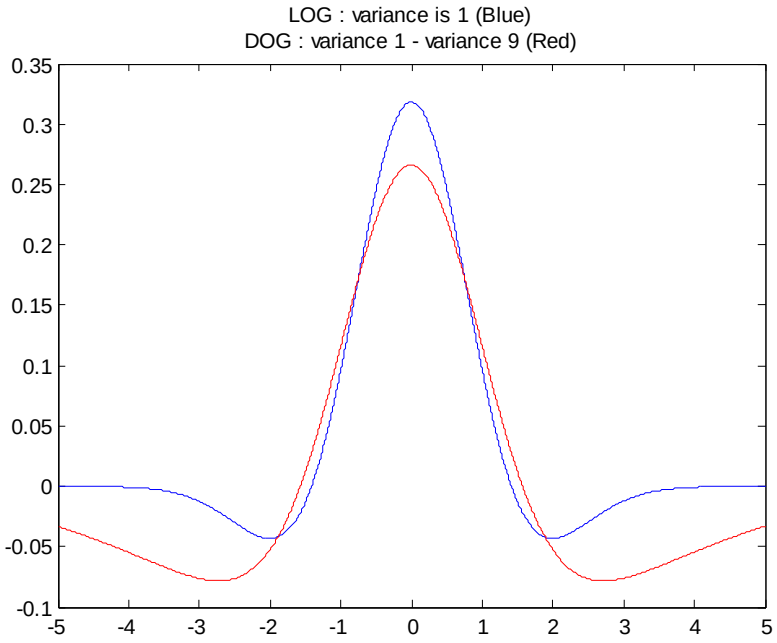

Fig. 5 The LOG and its approximation, DOG. Blue line denotes the LOG and red line denotes the DOG. In this figure, the LOG is on variance equal to 1; the DOG is on s_1 equal to 1 and s_2 equal to 3

This is so called the four-neighborhood *Laplacian* impulse response. Actually, we will see other types of *Laplacian* impulse response for the directed second-order derivative method. In the Fig. 6, the corner detection impulse response is implemented by the *Laplacian* response. Note that a zero crossing will not exist at the single-pixel transition response model. The edge corner should be marked at a pixel that is with a positive response and one of neighborhoods with negative response.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -3/2 & -3/2 & -3/2 & -3/2 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -3/2 & 1 & 2 & 3/2 & 3/2 & 3/2 \\ 0 & -3/2 & 0 & 3/2 & 0 & 0 & 0 \\ 0 & -3/2 & 0 & 3/2 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 6 single pixel transition *Laplacian* response

We also can directly find the second-order derivative of a continuous domain $F(x, y)$ by estimating the edge angle θ with respect to horizontal axis

$$F''(x,y)=\left\{\frac{\partial F(x,y)}{\partial x}\cos\theta+\frac{\partial F(x,y)}{\partial y}\sin\theta\right\}'$$

$$=\frac{\partial^2 F(x,y)}{\partial x^2}\cos^2\theta+2\frac{\partial^2 F(x,y)}{\partial x\partial y}\cos\theta\sin\theta+\frac{\partial^2 F(x,y)}{\partial y^2}\sin^2\theta \qquad (22)$$

In discrete domain, one approximation to this function is to employ the first-order derivative edge detection of row and column gradient relation to estimate the direction of the edge. Then take the angle information into the *like-Laplacian* method to approach in the discrete domain. Another approach, proposed by *Haralick*, involves approximating $F(x,y)$ by a two-order polynomial for the pixel neighborhood that takes the central pixel as zero index. Then we do the second-order differentiation directly on this approximated polynomial

$$F(r,c)=k_1+k_2r+k_3c+k_4r^2+k_5rc+k_6c^2+k_7rc^2+k_8r^2c+k_9r^2c^2 \qquad (23)$$

Consequently, we multiply the coefficients of the tree by tree response model to the pixel position of the approximated coefficients, as in Fig. 7, that can have some conditions for coefficients in response model. There are many polynomials can be an approximation, but we prefer to this kind of orthogonal polynomials for its low computation. Others like the *Chebyshev orthogonal polynomials* are also good choices

$$F(r,c)=k_1+k_2r+k_3c+k_4\left(r^2-\frac{2}{3}\right)+k_5rc+k_6\left(c^2-\frac{2}{3}\right)$$
$$+k_7c\left(r^2-\frac{2}{3}\right)+k_8r\left(c^2-\frac{2}{3}\right)+k_9\left(c^2-\frac{2}{3}\right)\left(r^2-\frac{2}{3}\right) \qquad (24)$$

$k_1-k_2-k_3+k_4+k_5+k_6-k_7-k_8+k_9$	$k_1-k_2+k_4$	$k_1-k_2+k_3+k_4-k_5+k_6-k_7+k_8+k_9$
$k_1-k_3+k_6$	k_1	$k_1+k_3+k_6$
$k_1+k_2-k_3+k_4-k_5+k_6+k_7-k_8+k_9$	$k_1+k_2+k_4$	$k_1+k_2+k_3+k_4+k_5+k_6+k_7+k_8+k_9$

Fig. 7 The 3x3 two-order approximated coefficient block

6. CONCLUSIONS

I have fully studied about the references about edge and corner detections. However, for the constraints on the pages and my purpose, I only wrote a proportional summarization but not the detail histories and algorithms about edge and corner detections. I believe that I have completed the fundamentals of the detections. On the later research, there are still many enhanced algorithms to develop .My future works may still keep on finding the latest

algorithms but in the color image. Since the definition for an edge of a color image would be in several ways and pixels have not only intensity component but also hue and saturation components that form the vector differences between R, G, B.

7. REFERENCES

[1] C. Harris and M. Stephens, "A combined corner and edge detection", Proc. 4th *Alvey Vision Conference*, pp. 189-192, 1988.

[2] Soo-Chang Pei and Jian-Jiun Ding, "New corner detection algorithm by tangent and vertical axes and case table", *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 1, pp. I-365-368, 2005.

[3] Robert M. Haralick and Linda G. Shapiro, "Computer and robot vision volume I", *Addison-Wesley Publishing Company*, New York, 1992.

[4] William K. Pratt, "Digital Image Processing (Third Edition)", *John Wiley & Sons, Inc*, Manhattan ,2002.