

# A contour detector with improved corner detection

Li Gun<sup>1</sup> · Zhu Ran<sup>1</sup> · Chai Honglei<sup>1</sup>

Received: 27 April 2015 / Revised: 24 June 2015 / Accepted: 1 July 2015 /

Published online: 19 July 2015

© Springer Science+Business Media New York 2015

**Abstract** This paper mainly studies the image contour detection algorithm which can distinguish edges of different strengths. Based on the study of Probability-of-Boundary operator, we find that defects such as response suppression and offset exist in the algorithm during the detection of corners and curved edges, thus an improved algorithm is proposed. This algorithm retains the advantage in Probability-of-Boundary algorithm which can effectively distinguish the edge strength, while improves the above-mentioned defects. And an improved algorithm is proposed to characterize the strength of boundary reasonably, making the test results in line with human subjective recognition results.

**Keywords** Contour detection · Image segmentation · Computer vision

## 1 Introduction

Digital image processing includes three levels [7]. The input and output in the low-level processing are both images. The low-level processing includes image denoising and enhancement. The outputting features in the middle-level processing such as contour and region of input image will be used as input in high-level processing. The high-level processing aims to achieve tasks such as target recognition, matching and tracking. The detection of edge and contour belongs to middle-level image processing. An ideal result of edge and contour detection provides guarantee for higher processing.

Edge is the region where brightness and colour mutate. Contour is a subset of edge, and is a boundary between the objects. There is not any clear boundary between contours and edges. The edges can be divided into strong part and weak part according to the change of gradient

---

✉ Li Gun  
ligun@uestc.edu.cn

<sup>1</sup> School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu, China

and the size of the area inside the edges. The contour can be seemed as the strong edge, which is considered as more important information than weak edge.

The classic differential detection method is the mainstream among a variety of edge detection algorithms. In 1965, Robert [17] proposed the first detection method based on partial differential operators (Robert operator). After that Sobel operator [24] and Prewitt operator [14] have been proposed based on Robert operator. These methods use the difference between adjacent pixels to characterize the image edges, and execute convolution using the image and specific convolution kernel. Then the image will be made a linear transformation to the gray space. These methods are simple and have a small amount of computation, but they are only based on the local image characteristics which would be affected by noise easily. They usually cannot obtain an edge with a pixel width but generate a wide edge response. To resolve this problem, Rosenfeld proposed Laplacian [18] operator according to the properties that the image edge's derivative is maximum and its second derivative is zero. The Laplacian operator can get only one response to a single edge, but it is more sensitive to the isolated noise. With the presence of fake edge, breakpoint will also appear.

Marr and Hildreth proposed LoG [12] operator, combining the Gauss filter with Laplacian operator. They used Gauss filter to eliminate the noise in order to overcome the inherent limitation of the Laplacian operator, and obtained a good result, but the method is still unable to avoid breakpoint in edge.

In 1986, Canny [4] summarized previous edge detection algorithms, and put forward three criteria for edge detection: the high SNR criterion, high position accuracy criterion and single edge response criterion. He proved that a derivative of Gauss function is an ideal convolution kernel operator which is closest to the optimal filter, and put forward an improved edge detection process which can ensure the single response and connectivity of edge detection.

In recent years, with the development of artificial intelligence technology, great progress has been made in image segmentation, object recognition and other high level image processing applications, proposing new demands to the edge segmentation of middle level. Some methods of image processing at high level expect the processing in the middle level to distinguish the edges with “strength” in order to extract the contours of objects. The classical methods of edge detection like Canny can only convolute kernel operation with adjacent pixels. Contour and texture details inside the object are generally regarded as the same edges' output, which cannot meet the demands of high level application.

To resolve this problem, it is necessary to consider the correlation of the pixels in greater range near the edge. An idea is clustering the pixels in the same region, and extracts the edge between the pixel block as the contour. Z.TU [25] and J. Malik [11] used the regional merger idea, obtaining the main contour by combining the small areas which have similar image features. Both the mean-shift algorithm proposed by P. Meer [5, 6] and Normalized Cut algorithm used by J. Shi [23] and S. Maji [10] achieved clustering of internal pixel. R. Achanta [1] proposed SLIC algorithm as superpixel clustering, in order to get different levels of edge by iteration. A. Levinstein [9] used the geometric flow to accelerate pixel clustering.

Another idea is to distinguish different types of edge features. Under the framework of the Canny detection, Ruzon and Tomasi [19, 20] proposed the edge detection algorithm based on histogram statistics using a larger scale of convolution kernel, and

can effectively avoid the texture details detected by the traditional method like Canny edge detection. On this basis, combining with the research of J. Puzicha [15] et al, D. Martin [13] analyzed the characteristics of object contour edge and texture of non object contour edge, and proposed  $\chi^2$  function as the objective function which can better characterize the strength of contour. They combined luminance, colour and texture information of the image and compared the detection results by human, and proved that the method is close to the result of human's identification. But this method has unstable defect on different scales. Ren [16] pointed out that Martin's shortcomings of the algorithm optimization at multiple scales, and gave a correct method of multi scale fusion. Arbelaez [2] put forward the optimized algorithm based on global information of the image.

Based on the studies of D. Martin and Arbelaez *et al*, this paper points out that suppressive detection exists in this kind of algorithm in corners and bending, cross edges, and gives a solution. Experiments show that the improved algorithm can effectively reduce the distortion of contour detection.

This paper is organized as follows: the second section describes the classic Canny edge detection algorithm and Probability-of-Boundary contour detection algorithm. The third section points out some defects of Probability-of-Boundary algorithm and proposes the improved wedge template operator. The fourth section proposes improved mean distance function to measure the strength of the edge. The fifth section proposes weighted histogram which makes the results more consistent with human subjective vision. The sixth section concludes the whole thesis.

## 2 Existing edge and boundary detection algorithms

### 2.1 Canny detection algorithm

The Probability-of-Boundary algorithm introduced in this paper is an improvement of Canny detection algorithm. Canny algorithm is divided into the following four steps: Gauss filter, computing the gradient magnitude and direction, non maximum suppression and double threshold edge connection.

#### 2.1.1 Gauss filter

Canny proved that a derivative of Gauss function in second-dimensional direction gradient is the closest convolution kernel operator to the optimal filter:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{x^2 + y^2}{2\sigma^2}\right) \quad (1)$$

The filtered image can be expressed as:

$$I(x, y) = F(x, y) * G(x + i, y + j) |_{i^2 + j^2 < r^2} \quad (2)$$

Where  $F(x, y)$  is a second-dimensional matrix of gray image. And  $r$  is the radius of the template. A smaller value of  $r$  can obtain better location performance, while a larger one can avoid false detection caused by noise more effectively,  $r$  generally takes  $2\sigma$  to  $3\sigma$ .

### 2.1.2 The calculation of the gradient magnitude and gradient direction

If the point  $(x, y)$  in the edge meet the condition that  $|\nabla(G*I)|$  takes local maximum in the direction of gradient, for digital images, there is:

$$|\nabla(G*I)| = \sqrt{|P_x|^2 + |P_y|^2} \quad (3)$$

Among them, there is:

$$P(x, y) = G(x, y)I(x, y) \quad (4)$$

$$P_x = P(x, y) - P(x-1, y) \quad (5)$$

$$P_y = P(x, y) - P(x, y-1) \quad (6)$$

where  $P(x, y)$  is the gray level of point  $(x, y)$ . A unit vector in the direction of the gradient is,

$$\frac{\nabla P(x, y)}{|\nabla P(x, y)|} = \frac{|p_x|^2 \vec{i} + |p_y|^2 \vec{j}}{|p_x|^2 + |p_y|^2} \quad (7)$$

where  $\vec{i}$  is the unit vector in x-axis, and  $\vec{j}$  is the unit vector in y-axis.

### 2.1.3 Non maximum suppression in gradient direction

The gradient magnitude of each point can be regarded as the probability of this point being boundary, which is a roof running through the real edge shown in the image. In order to avoid multiple edge responses, the peak point should be chosen at the roof, that is, gradient magnitude is higher than a certain threshold and is greater than the critical point at the direction of the gradient.

### 2.1.4 Dual threshold connection

Outputting the results of the third step directly will bring such a problem: single threshold selection is difficult. A low threshold will bring a large number of over detection, while a high threshold will bring edges defect. Canny detection is based on double threshold connecting method: looking for points of which the value is higher than low threshold in the neighbourhood of points where the value is higher than high threshold and connecting them in turn. This method can ensure connectivity with noise being suppressed.

## 2.2 Probability-of-Boundary algorithm

Based on Canny detection algorithm, Probability-of-boundary algorithm aims to ignore the texture details in the image to find the main contour. The method is improved according to the fact that gradient magnitude and direction of Canny detection method can only detect the difference between adjacent pixels.

As shown in Fig. 1, the sail and water form a remarkable contour line. For any point in this contour line, pixels in its neighbourhood (on both sides of the semi-circular area which are marked red and blue in the figure) have significant difference in gray level distribution. Unlike the detail texture (such as ripples on the water in figure) which is only different in gray level between a few pixels adjacent to each other, the difference in gray level distribution nearby the boundary exists in larger region. Literature [2] presented two semi-circular area statistics for the histogram, and output difference between two histograms as gradient magnitude response of the point. Since the semi-circular area is much larger than the Canny operator, this method makes the boundary obtain more significant response than detail texture, and makes the Probability-of-Boundary response more consistent with human subjective vision. The more significant the contours are, the stronger the Probability-of-Boundary response is. Different intensity contours can be selected by setting different Probability-of-Boundary thresholds.

There are multiple methods to characterize differences between the two semicircles. Literature [15] proposed that characterization of the differences between two regions by  $\chi^2$  would have better recognition performance than those by Euclidean distance, norm-1, and EMD methods.

$$\chi^2(x, y, \theta) = \frac{1}{2} \sum_{i=1}^N \frac{(g(i) - h(i))^2}{g(i) + h(i)} \quad (8)$$

$N$  is image quantization.  $g(i)$  and  $h(i)$  are histograms of each semicircle, which mean the number of pixels in each semicircle templates at the value of  $i$ . Literature [15] verified the  $\chi^2$  distance could be used to characterize the difference of the image grayscale or colour distribution between two areas.

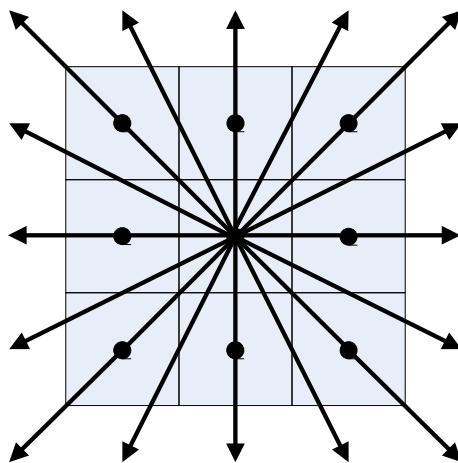
$\chi^2(x, y, \theta)$  can be regarded as possibility-of-boundary of point  $(x, y)$ . The possibility that the pixel is detected as the boundary is determined by the largest value of  $\chi^2(x, y, \theta)$ , that is:

$$Pb(x, y) = \max_{\theta} \chi^2(x, y, \theta) \quad (9)$$

Unlike Canny operator which can only calculate the gradient magnitudes of vertical and horizontal directions, the Probability-of-Boundary algorithm allows more directions, so that more accurate calculation of the gradient direction can be made. But as each pixel having only 8 pixels in adjacent, non maximum suppression only corresponds to four gradient directions. Probability-of-Boundary uses interpolation to solve this problem. As shown in Fig. 2, when  $\theta$



**Fig. 1** The Probability-of-Boundary histogram algorithm



**Fig. 2** The gradient direction for non maximum suppression

is  $22.5^\circ$ ,  $67.5^\circ$ ,  $112.5^\circ$  or  $157.5^\circ$ , the adjacent pixels of the actual gradient direction should interpolate according to the pixels on two adjacent directions.

The larger convolution kernels make it necessary to change the filtering algorithm. In some classic detection algorithms such as Canny, Gauss filter was used to eliminate false edges caused by noise. For Probability-of-Boundary detection, a large detection window makes the result non-sensitive to the isolated noise points in the image, but sensitive to the gradient direction. The gradient direction of the contour has a strong spatial correlation. The gradient directions of adjacent pixels are close to each other. Orientation energy filtering algorithm uses this feature to eliminate the noise of gradient direction. The filter is defined as:

$$OE_{\theta,\sigma} = \left( I * f_{\theta,\sigma}^e \right)^2 + \left( I * f_{\theta,\sigma}^o \right)^2 \quad (10)$$

where,  $I$  is the image matrix.  $f^e$  is the second derivative of the Gaussian function.  $f^o$  is the Hilbert transformation. The radius of the filter window takes 2 to 3 times of the Probability-of-Boundary operator radius. The filter can effectively strengthen the adjacent edge points which are consistent with the gradient direction.

In actual images, contour may get high Probability-of-Boundary value in one or several channels among gray, colour or texture. Unlike classical detection algorithms such as Canny, the output response of Probability-of-Boundary algorithm is not in a certain range. Different image channels, quantization levels, and radiiuses of the window will get different values. Data in multiple channels, multiple scales should be normalized before outputting. Literature [2, 13] studied the multi-channel, multi-scale data fusion methods. Literature [13] used the following equation to achieve normalization, in order to ensure that all pixels are in the allowed range of values.

$$IM(x,y) = \frac{N}{1 + e^{-\lambda - \sum_{i=1}^n \partial_i Pb_i(x,y)}} \quad (11)$$

Here,  $N$  is the quantization of the output image, which is 255 for 8-bit grayscale image.  $\lambda$  is a large positive to ensure that the output is close to zero when input is zero.  $i$  means different

channels and with different scales.  $\partial_i$  is a positive number. Assuming that  $N=1$ , Eq. (11) is a function of shape as shown in Fig. 3.

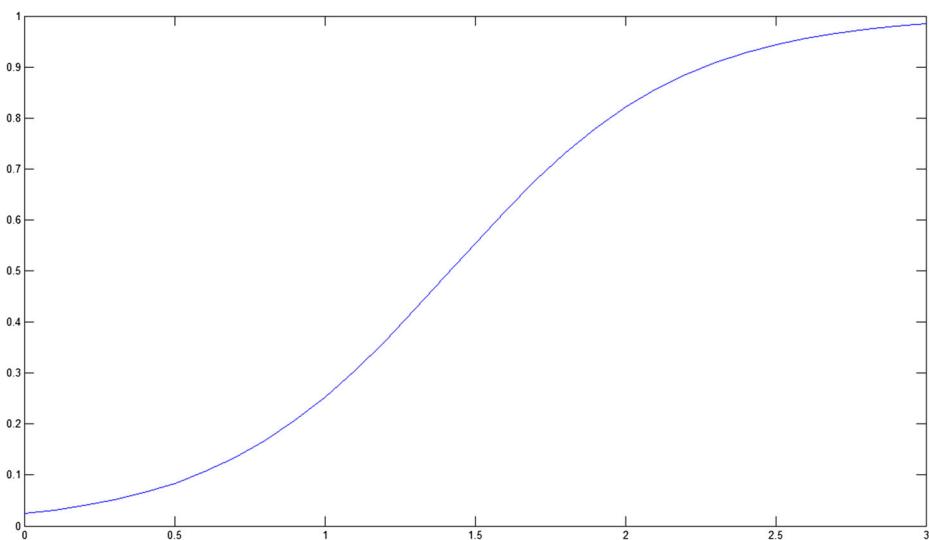
The output value of the function is limited to (0, 1), and the output monotonically increases with the input, which has a maximum value of derivative at the input of medium size. For any pixel, the suitable  $\lambda$  and  $\partial_i$  enable that the pixel with high response values in one or more channels obtains a higher value than those without high response in any channels, and does not exceed the scope of image quantization. This fusion method is in line with the human eyes and subjective visual. The points with any obvious differences between the two neighborhoods in any channels will be regarded as boundary. Literature [13] contrasted multiple images in which boundaries are artificially drawn, and trains multiple sets of values for different channels and scales.

### 3 Improved boundary detection operator

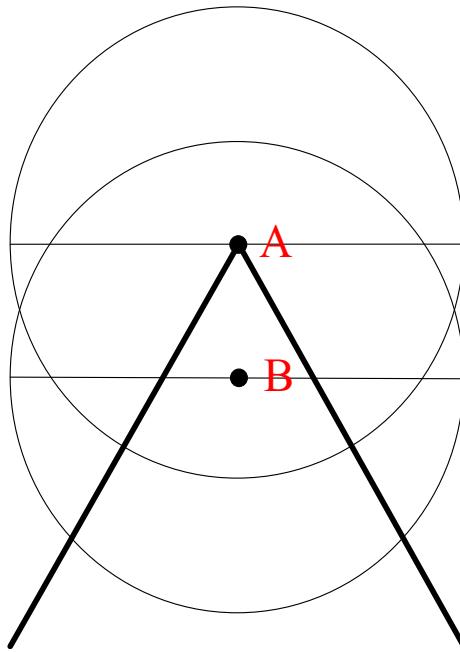
#### 3.1 The shortcomings of the traditional Probability-of-Boundary algorithm

The Probability-of-Boundary algorithm uses the convolution kernel of larger size, so that different response values are obtained for different strengths of the edge response. Literature [2, 16] proposed methods to fuse the results from a plurality of convolution kernel scales to further improve the detection performance, but greater convolution kernel brings side effect as well.

As shown in Fig. 4, point A is in the corner of the real edge, but both sides of the two semicircles in any directions are not completely divided the edge, while point B away from the true edge could get higher Probability-of-Boundary response and form a pseudo edge. Therefore, the value of Probability-of-Boundary algorithm at a corner is less than straight edge, and tends to be more rounded than the real edge.



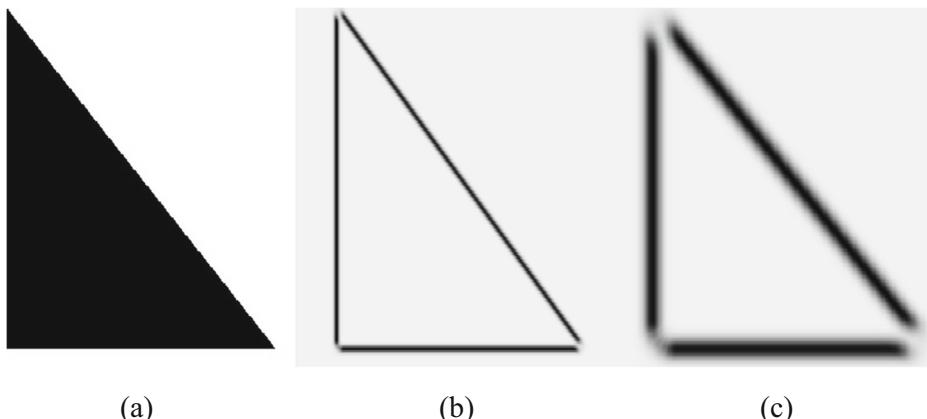
**Fig. 3** The transformation for image output



**Fig. 4** The true edge point at a corner may not get a maximum response

As shown in Fig. 5, the original image is the left one, the one in the middle is the detection result of Probability-of-Boundary algorithm with the radius of 5 pixels (without Non maximum suppression thinning), and the right is the result with the radius of 15 pixels. The corners are round and the responses are suppressed. These defects will be great when the radius increases.

Curved, staggered edges lead to the same defect, and this defect is determined by the geometric structure of the Probability-of-Boundary operator. In fact, due to the assumption that



**Fig. 5** An example of corner suppression: **a** the original image, **b** detection result with the radius of 5 pixels, **c** detection result with the radius of 15 pixels

the edge gradient direction is consistent in the template, a majority of methods using template convolution have the similar defects. Since the convolution kernel is smaller, the defect in classical algorithms such as the Canny operator is not obvious. Literature [8] proposed to detect the corners independently and to repair the defect by curve fitting in the Canny detection. But for Probability-of-Boundary detection, the defects are too large to repair. Improved detection algorithm should be done for itself, making straight edges and corners get close response.

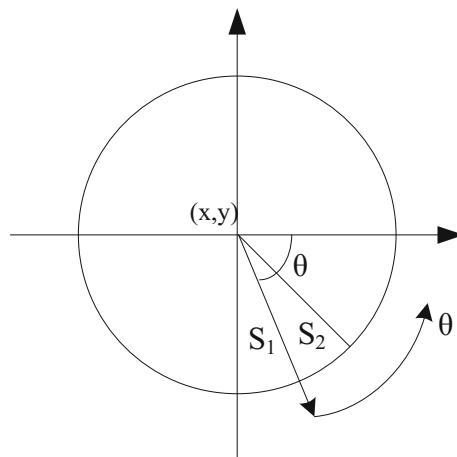
### 3.2 Improved operator using wedge template

By dividing the circle into a plurality of wedge, we can effectively improve the performance of detection of the Probability-of-Boundary algorithm in the corner edge.

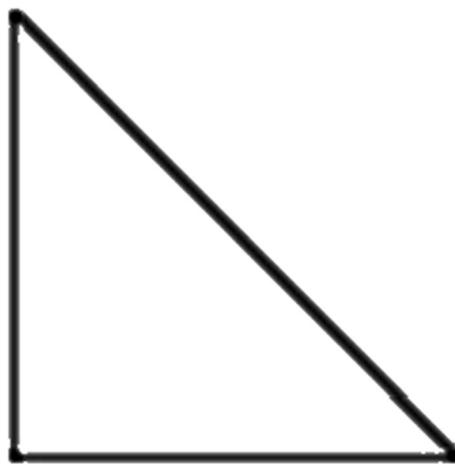
As shown in Fig. 6, Probability-of-Boundary value of point  $(x, y)$  at the corners is determined by two wedges between the ray in the direction of  $\theta$ . If  $Pb(x, y, \theta)$  gets a maximum value at  $\theta$ ,  $\theta$  is the direction of the ray through point  $(x, y)$ . If the angles of the two wedge convolution kernels are small enough, the two convolution kernels can be covered generally by each side of the edge at the right  $\theta$ . Therefore, the Probability-of-Boundary value generated by the convolution kernel will not be different whether point  $(x, y)$  is at the straight edges or corners, which makes an improvement for boundary detection at corners. Figure 7 shows the improved result without thinning.

As shown in Fig. 8,  $\theta_1$  and  $\theta_2$  at both sides of  $(x, y)$  will generate maximum value of  $Pb_\theta$ . Assuming that they are  $Pb(x, y, \vec{\theta}_1)$  and  $Pb(x, y, \vec{\theta}_2)$ , in a straight edge,  $\theta_1, \theta_2$  form an angle of  $180^\circ$ . But for corners or curved edges, the gradient direction of actual edge should be calculated by the following equation.

$$\vec{\theta} = \frac{Pb(x, y, \theta_1) \cdot (\theta_1 \vec{\pi}) + Pb(x, y, \theta_2) \cdot (\theta_2 \vec{\pi})}{Pb(x, y, \theta_1) + Pb(x, y, \theta_2)} \quad (12)$$



**Fig. 6** Divide the half-disk template into wedge template



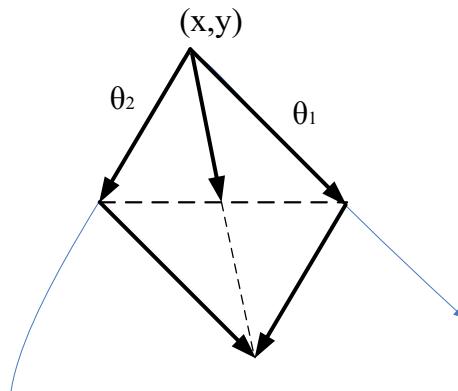
**Fig. 7** The detection result of wedge template

$Pb(x, y, \vec{\theta})$  in the direction should be

$$Pb(x, y, \vec{\theta}) = \sqrt{\frac{Pb^2(x, y, \vec{\theta}_1) + Pb^2(x, y, \vec{\theta}_2) + 2Pb(x, y, \vec{\theta}_1)Pb(x, y, \vec{\theta}_2)\cos(|\theta_1 - \theta_2|)}{2}} \quad (13)$$

The value of  $Pb(x, y, \vec{\theta})$  should be in the range of  $Pb(x, y, \vec{\theta}_1)$  to  $Pb(x, y, \vec{\theta}_2)$ . Therefore,  $Pb(x, y, \vec{\theta})$  will be used in the thinning of the output image while the final output should be  $\max(Pb(x, y, \vec{\theta}_1), Pb(x, y, \vec{\theta}_2))$ .

Improved algorithm optimizes the detection for corners and curved edges, etc, but this advantage will be weakened by the orientation energy filter. Because the consistency of gradient orientation at corner is lower than that of the straight edge. The



**Fig. 8** The calculation of gradient orientation

response at corners will be suppressed with the orientation energy filter. So we use threshold orientation energy filter to avoid this defect. The value will be output when the Probability-of-Boundary response is higher than a certain threshold. The above process is shown in Fig. 9.

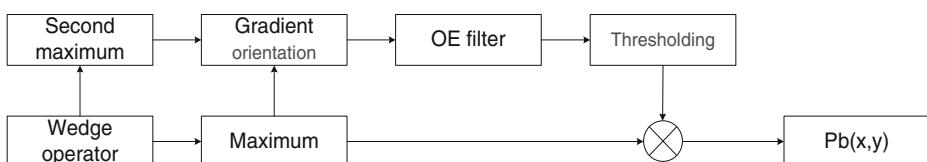
### 3.3 Experimental results and analysis

Figure 10 shows the comparison of results between this algorithm and the traditional Probability-of-Boundary algorithm (without normalization and thinning, taking colour-map for convenience to make a comparison). These images are taken from BSDS300. The detection uses brightness channel, and the radius of convolution kernel is 5 pixels. In the image on the left top, there are rocks and trees with similar brightness, which should have similar boundary strength. But in the image on the middle top, trees have litter response due to their curved contour, and the result is fuzzy and defective. The image on the right top is the improved result. In the foreground, the outlines of the rocks and the trees are marked in red with similar value, and the outline of the trees is much clearer.

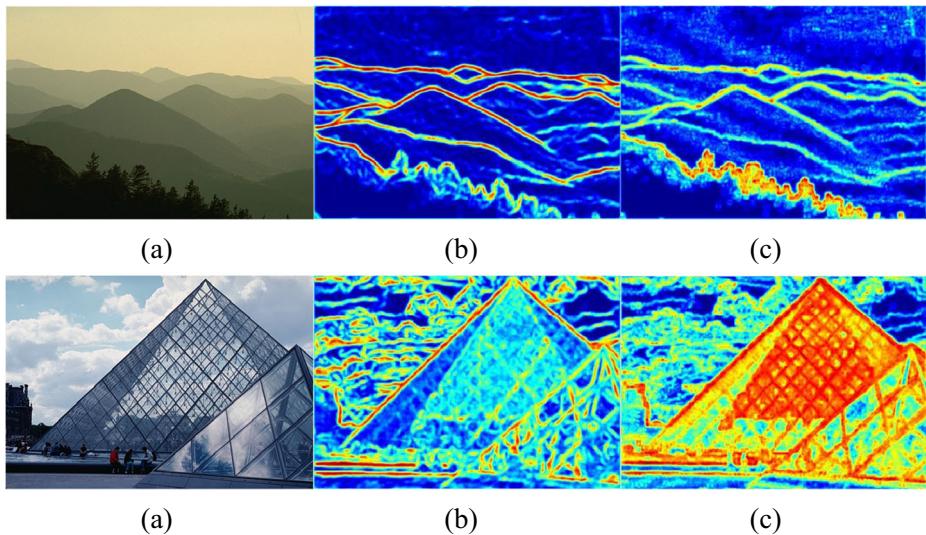
The image on the right bottom shows improvement for dense edge. On the middle bottom in the image, the inner edges of pyramid glass are detected as bending and messy curve, and the values of them are suppressed. The image on the right bottom uses improvement operator and get the result with cross straight line, but it still has a defect: the response of the inner edges should be less than that of the outer boundary, but in the result they have similar value. This defect is due to  $\chi^2$  function, and the corresponding improvement will be presented in the next section.

Figure 11 contrasts contour detection results after thinning. The Canny detection algorithm is on the left. The detection results on the main contour and detail texture edge cannot be distinguished. Some intensive edges such as those of the tree generate multiple responses. While it gets complete detection of outer contour, the Probability-of-Boundary algorithm gets many messy curves for internal edges of the glass Pyramid with suppressed value. Right as the improved algorithm, the linear features and integrity of internal contour are basically retained. But there are still some noise and burr. It is mainly because at the same radius, wedge template contains significantly less pixels than half circular template so that the test results are sensitive to noise. The template radius may be appropriately increased to solve this problem. In addition,  $Pb(x,y,\theta)$  in wedge template changes sharply when  $\theta$  changes and it is sensitive to the calculated error of the gradient direction, which is also the cause of burr. It may be feasible to use the fusion method in Section 2.2 to get fusional results with two kinds of operator.

Different radiiuses of the wedge template will bring different results. Figure 12 shows contour detection results with different radiiuses of template before thinning. The angle of the wedge template is 45°.

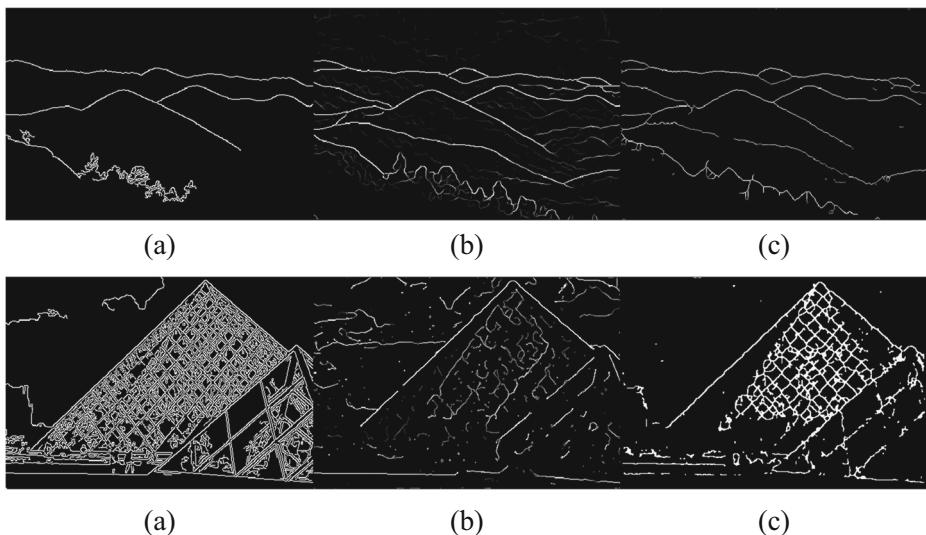


**Fig. 9** OE filtering for wedge template

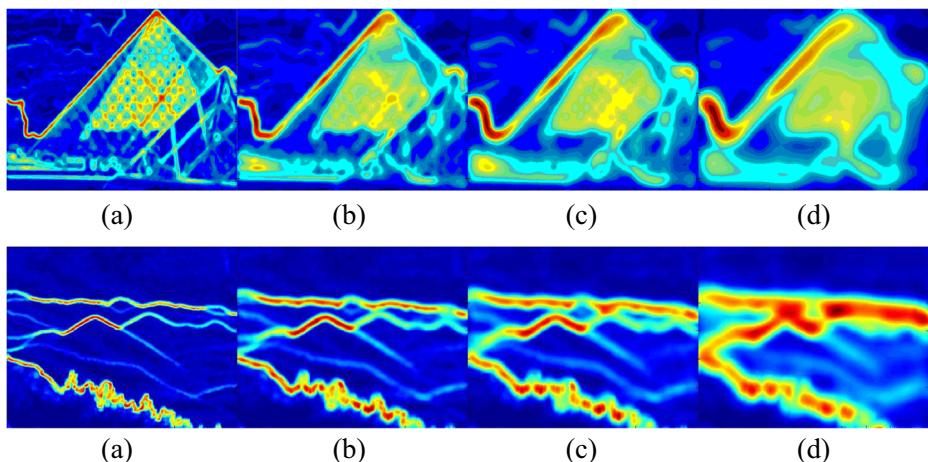


**Fig. 10** The detection results without thinning. **a** the original image, **b** the results with semicircle template, **c** the results with wedge template

In Fig. 12, the results with small template are similar to edge detection. The contour line is thin, which means high positioning accuracy. But the results are sensitive to texture. The pixels in inner texture may get higher response than those in outer contour. While the results with big template are only sensitive to the main contour, they may lose the positioning accuracy since their contour line is thick. So the fusion algorithm of multiscale detection introduced in Section 2.2 can be used to strike a balance.



**Fig. 11** The results after thinning: **a** the results with Canny detection, **b** the results with semicircle template, **c** the results with wedge template

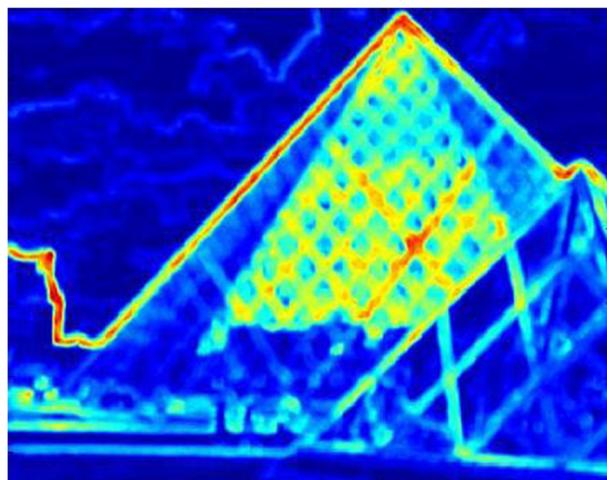


**Fig. 12** The results with different radius: **a** radius is 5 pixels, **b** radius is 10 pixels, **c** radius is 15 pixels, **d** radius is 25 pixels

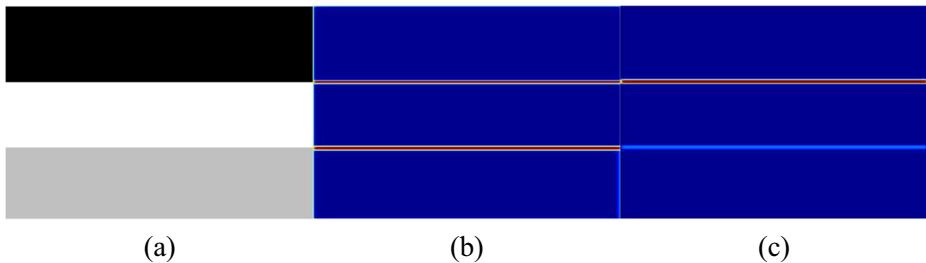
While another experiment shows that results with different angles of wedge template do not make many differences. Figure 13 is the result when the radius of template is 5 pixels and the angle of the wedge templates is  $22.5^\circ$ , which is similar to image (a) in Fig. 12. So there is no need to fuse the result with different angles.

#### 4 An improved approach to measure the strength of boundary

Literature [15] proposed  $\chi^2$  function to measure the differences between two templates, which had a good recognition performance. But in the process of edge detection, a higher recognition performance does not always mean a better one. The part with slightly difference in colour or gray channels should not generate a high Probability-of-Boundary response. But  $\chi^2$  exactly has this defect.



**Fig. 13** The results with different angle of the templates



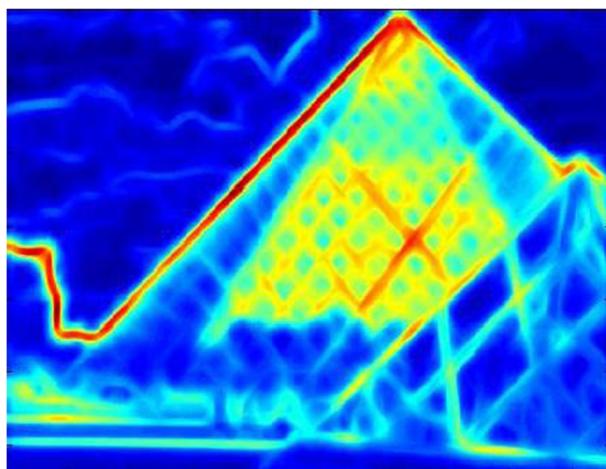
**Fig. 14** Edges with different strength may obtain a same response: **a** the original image, **b** the results with function, **c** the results with mean distance function

$$\chi^2(x, y, \theta) = \frac{1}{2} \sum_i \frac{(g(i) - h(i))^2}{g(i) + h(i)} \quad (14)$$

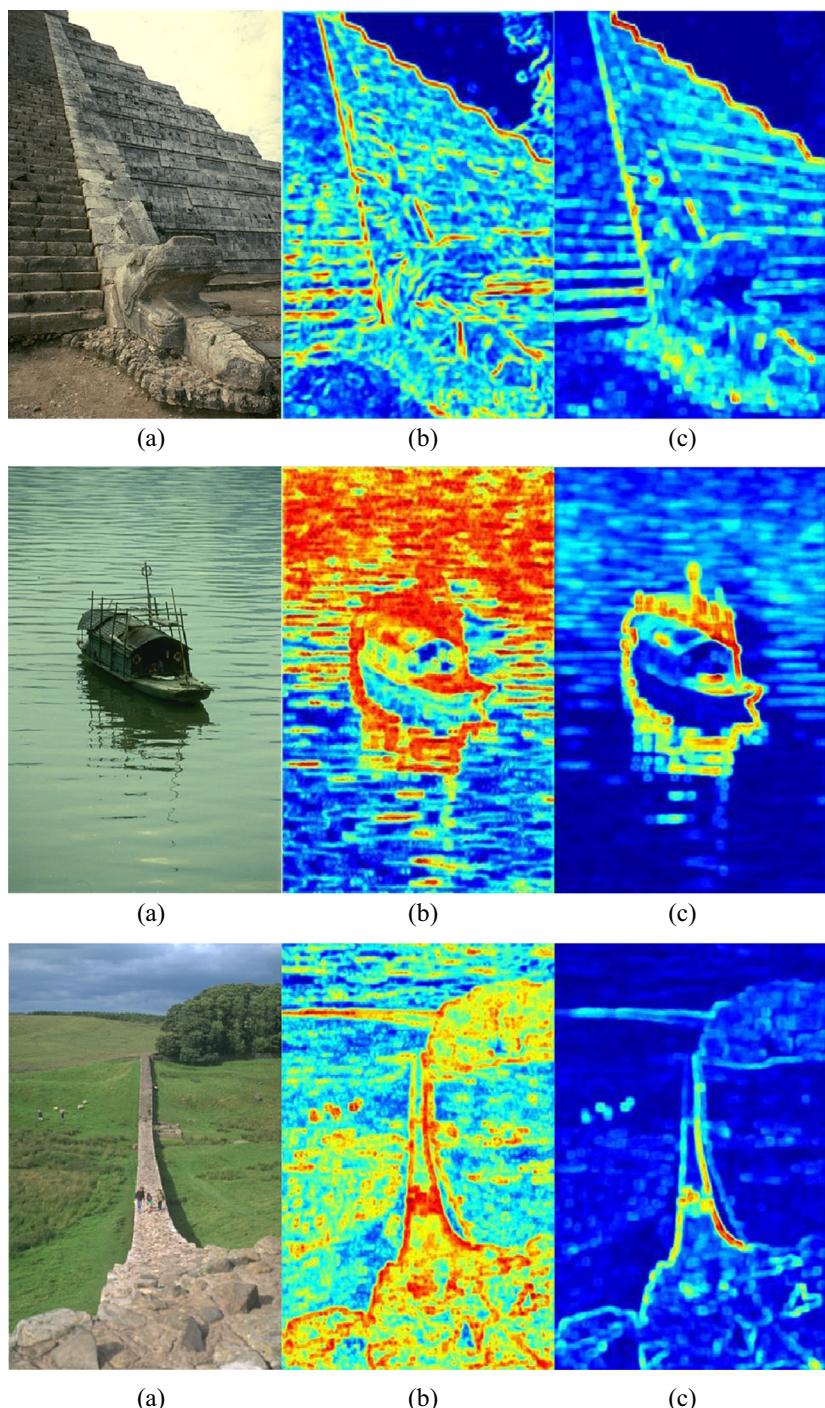
where,  $g(i)$  and  $h(i)$  are histograms of each operator.  $\chi^2$  uses differences on each quantized values in gray or colour histogram to measure the global differences between the two regions. The lower contact ratio of histograms is, the higher output will be generated, not meaning high strength of boundary. As shown in Fig. 14, the boundary between white and black and the other between white and gray generate the same strength of Probability-of-Boundary response since their contact ratios in gray histogram are both zero.

This goes against common sense. Difference between white and black is greater, so that the Probability-of-Boundary response of it should be higher. The way to measure the strength of boundary should be based on concentration of colour and gray scale. Difference on the quantized values which is far away from the mean value makes greater contribution to Probability-of-Boundary response than others. So the mean distance function is proposed to measure the strength of boundary.

$$MD(x, y, \theta) = \sum_{i=1}^n |g(i) - h(i)| |i - avg(x, y)| \quad (15)$$



**Fig. 15** The result using mean distance function



**Fig. 16** Additional results on BSDS300: **a** the original image, **b** the results with function, **c** the results with mean distance function

Here,  $\text{avg}(x,y)$  is the average gray level in the operator:

$$\text{avg}(x,y) = \frac{\sum_{i=1}^n i(g(i) + h(i))}{\sum_{i=1}^n g(i) + h(i)} \quad (16)$$

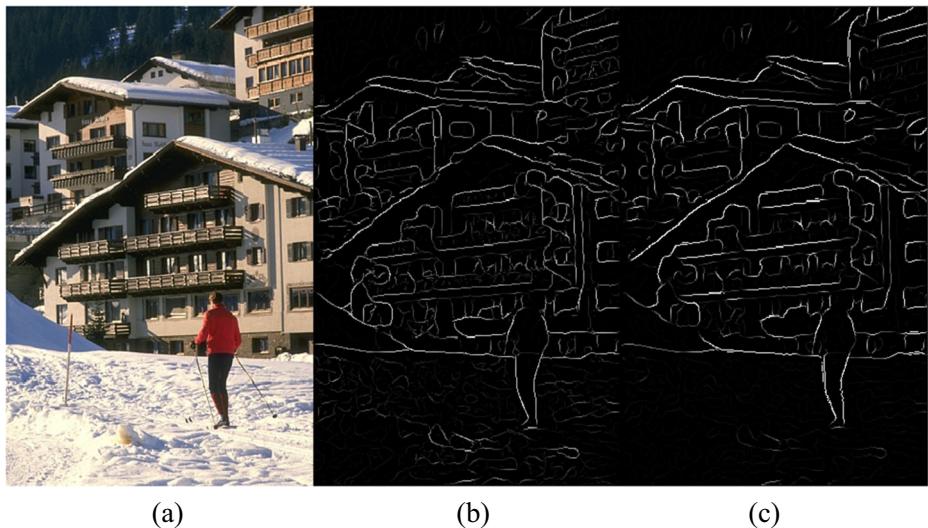
In Fig. 14, the picture on the right shows the detection results obtained by the mean distance function where the difference of the two edges is reflected.

In Fig. 10, the boundary and the texture of the glass Pyramid generate close Probability-of-Boundary value for the same reasons. The result of improved method is given in Fig. 15. The outer contour and inner edges of the glass Pyramid and textures in the background are marked with different colours, which is significant to distinguish.

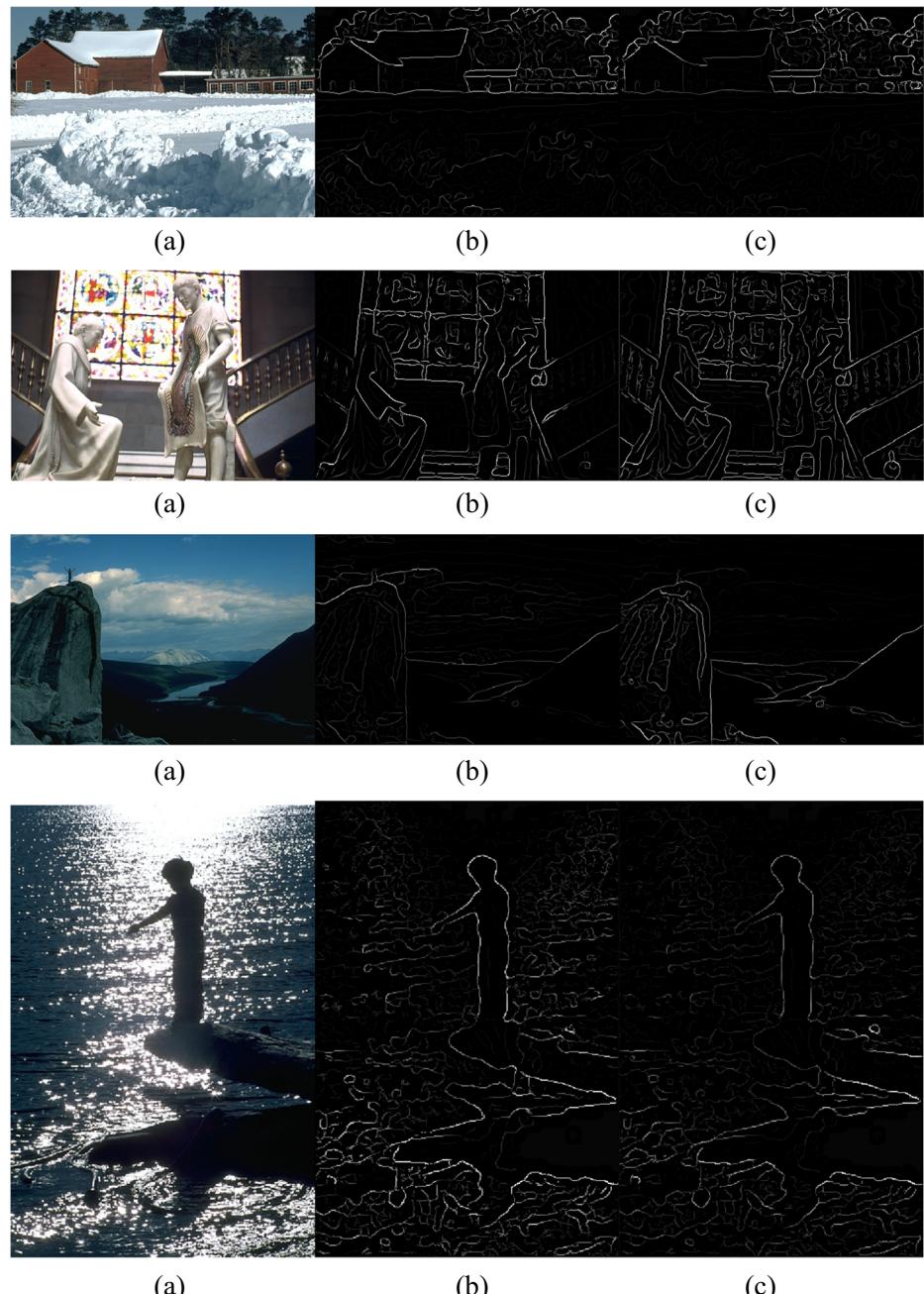
More experimental results are shown as follows in Fig. 16. The left images are the original image, in the middle there are traditional detection results. The right images are obtained by the mean distance function (Fig. 16).

## 5 Weighted histogram

We know that  $g(i)$  and  $h(i)$  are unweighted histograms. This is not in accord with the subjective vision [21, 22]. According to Weber-Fechner law, the subjective visual in human eyes have a



**Fig. 17** The result with weighted histogram. **a** the original image, **b** the results with original histogram, **c** the results with weighted histogram



**Fig. 18** Additional results on BSDS300: **a** the original image, **b** the results with original histogram, **c** the results with weighted histogram

logarithmic relationship with the actual brightness, which means the human eyes are more sensitive to differences in the low gray area but not sensitive to differences in the high gray

value. Therefore in the generation of gray histogram, low gray part should have greater weights.  $g(i)$  and  $h(i)$  in the histogram should be weighted:

$$G(x, y, i) = \frac{g(x, y, i)}{(\ln i + C)(\ln(\text{avg}(x, y)) + C)} \quad (17)$$

$$H(x, y, i) = \frac{h(x, y, i)}{(\ln i + C)(\ln(\text{avg}(x, y)) + C)} \quad (18)$$

where  $C$  is a constant,  $G(x, y, i)$  and  $H(x, y, i)$  are weighted histograms to replace  $g(x, y, i)$  and  $h(x, y, i)$  to calculate  $MD(x, y, \theta)$ .  $\ln i + C$  is the weighting of different gray values in the same template, while  $\ln(\text{avg}(x, y)) + C$  is the weighting among different templates. According to literature [3], the grayscale generated by image sensor has a linear relationship with the actual light intensity. The value of  $C$  which cannot be uniquely determined is related to the actual imaging. According to the literature, it takes 1.39.

Figure 17 shows the weighted histogram improve the improvement. The textures in high brightness part of the image such as the snow are not significant in human subjective vision. But the traditional Probability-of-Boundary algorithm will output these textures because of the high gray value. In the improved algorithm these details are correctly suppressed.

More experimental results are shown as follows in Fig. 18, the left images are the original images. In the middle there are traditional detection results. The right images are obtained by weighted histogram (Fig. 18).

## 6 Conclusions

The traditional Probability-of-Boundary contour algorithm can effectively distinguish edges with different strength, but there are still some defects. The paper points out that there are some defects in the bent, staggered edges, and proposes an improved wedge operator. In addition, we also propose the improved mean distance function and the weighted histogram algorithm. Experiments show that these two improvements effectively improve the representation of contour and solve the problem of over detection in high brightness region.

**Acknowledgments** This work was supported by the National Natural Science Foundation of People's Republic of China (Grant No. 91026005), I wish to thank Professor Wang LingYan who has contributed to the paper improvement.

## References

1. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Sussstrunk S (2010) SLIC superpixels. EPFL Technical Report
2. Arbelaez P, Maire M, Fowlkes C et al (2011) Contour detection and hierarchical image segmentation. IEEE Trans Pattern Anal Mach Intell 33(5):898–916
3. Bin Z (2004) Liu Bingqi: the research on brightness gain of CCD for image intensifier[J]. J Transducer Technol 23(12):20–24
4. Canny J (1986) A computational approach to edge detection. PAMI 8(6):679–698

5. Christoudias C, Georgescu B, Meer P (2002) Synergism in low-level vision, vol IV. 16th International Conference on Pattern Recognition. Quebec City, Canada, p 150–155
6. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. *IEEE Trans Pattern Anal Machine Intell* 24:603–619
7. Gonzalez RC, Woods RE (2002) Digital image processing. Publishing House of Electronics Industry, Beijing
8. Harris C, Stephens MJ (1988) A combined corner and edge detector. Proc Fourth Alvey Vision Conf 147–151
9. Levinstein A, Stere A, Kutulakos KN, Fleet DJ, Dickinson S J, Siddiqi K (2009), TurboPixels: fast superpixels using geometric flows. *PAMI*
10. Maji S, Vishnoi N, Malik J (2011) Biased normalized cut. *CVPR* 2057–2064
11. Malik J, Belongie S, Leung T, Shi J (2001) Contour and texture analysis for image segmentation. *Int J Comput Vis* 43(1):7–27
12. Marr D, Hildreth E (1980) Theory of edge detection. In: Proceeding of the Royal Society of London B(27):187–217
13. Martin C, Fowlkes C, Malik J (2004) Learning to detect natural image boundaries using local brightness, color and texture cues. *PAMI*
14. Prewitt JMS (1970) Object enhancement and extraction. In: SLipkin B, Rosenfeld A (eds) Picture processing and psychohistories
15. Puzicha J, Rubner Y, Tomasi C, Buhmann J (1999) Empirical evaluation of dissimilarity measures for color and texture. *Proc Int Conf Comput Vision*
16. Ren X (2008) Multi-scale improves boundary detection in natural images. *ECCV*
17. Roberts LG, (1965) Mthine perception of three-dimension solids. *Optimal and Electro-Optimal Information Processing*. MA: MIT Press, 99–197
18. Rosenfeld A, Kak AC (1976) Digital picture processing. Academic, New York
19. Ruzon M, Tomasi C (1999) Color edge detection with the compass operator. *Proc IEEE Conf Comput Vision Pattern Recognit*
20. Ruzon M, Tomasi C (1999) Corner detection in textured color images. *Proc Int Conf Comput Vision* 1039–1045
21. Shen J(J) (2003) On the foundations of vision modeling I. Weber’s law and Weberized TV (total variation) restoration. *Physica D: Nonlinear Phenomena* 175(3/4):241–251
22. Shen J(J), Jung Y-M (2006) Weberized Mumford–Shah model with Bose–Einstein photon noise. *Appl Math Optim* 53(3):331–358
23. Shi J, Malik J (2000) Normalized cuts and image segmentation. *PAMI*
24. Sobel IE (1970) Camera models and machine perception, Ph.D. dissertation, Stanford University, Palo Alto, Calif, 1970
25. Tu Z, Zhu S, Shum H (2001) Image segmentation by data driven Markov chain Monte Carlo. *Proc Int Conf Comput Vis* 2:131–138



**Li Gun** received the B.S. degree in theoretical physics from the Northwest Normal University, China, in 2001, and received the Ph.D. degree in Astrometry and Celestial Mechanics from Graduate University of Chinese Academy of Sciences, China, in 2007. During 2007–2015, he was a teacher at the University of Electric Science and Technology of China. His current main research interests include time synchronization, satellite navigation, signal processing and plasmas physics etc.



**Zhu Ran** received the B.E. degree in University of Electronic Science and Technology of China, in 2012, where he is currently pursuing the Master Degree. His areas of interests are computer vision and multimedia processing.



**Chai Honglei** received the B.E. degree in University of Electronic Science and Technology of China, in 2012, where he is currently pursuing the Master Degree. His research interests include image processing and signal processing.