Multimed Tools Appl (2016) 75:16665–16682
DOI 10.1007/s11042-016-3835-y

Line detection algorithm based on adaptive gradient threshold and weighted mean shift

Yi Wang[1] · Liangliang Yu[1] · Houqi Xie[1] · Tao Lei[2,3] · Zhe Guo[1] · Min Qi[1] · Guoyun Lv[1] · Yangyu Fan[1] · Yilong Niu[4]

Received: 27 January 2016 / Revised: 25 July 2016 / Accepted: 2 August 2016 /
Published online: 15 August 2016

Abstract Line detection is a classical problem in computer vision and image processing, and it is widely used as a basic method. Most of existing line detection algorithms are based on edge information, whose discontinuity limited the detection result. Meanwhile, some other algorithms only use gradient magnitudes, and neglect the function of gradient directions. In this paper, an adaptive gradient threshold and omni-direction line growing method based on line detection with weighted mean shift procedure and 2D slice sampling strategy (referred to as LSWMSAllDir) is proposed. It makes full use of the magnitudes and directions of the gradient to detect lines in the image. Experiments on synthetic data and real scene image data showed that the improve algorithm was the most accurate when compared with Progressive

✉ Yilong Niu
yilong_niu@nwpu.edu.cn

Yi Wang
wangyi79@nwpu.edu.cn

Liangliang Yu
yuliangliang@mail.nwpu.edu.cn

Houqi Xie
jxfcxhq@163.com

Tao Lei
leitaoly@163.com

Zhe Guo
guozhe@nwpu.edu.cn

Min Qi
drqimin@nwpu.edu.cn

Guoyun Lv
lvguoyun101@nwpu.edu.cn

Yangyu Fan
fan_yangyu@nwpu.edu.cn

Probabilistic Hough Transform (PPHT), line segment detector (LSD), parameter free edge drawing (EDPF) and original line segment detection using weighted mean shift (LSWMS) algorithms.

Keywords Line detection · Weighted mean shift · Adaptive gradient threshold · Omni-direction searching

1 Introduction

Lines represent the main edge information in images. To locate and recognize targets through line feature plays an important role in the field of computer vision and image processing. Line detection has great practical application value in various fields by combining machine vision with the real environment, such as camera calibration [4], industrial inspection [17, 28], target recognition [26], remote sensing [16], material crack detection [21], image compression [10], and satellite image index [12, 24].

Hough Transform (HT) is the most widely used line detection method. Hough Transform maps collinear points in image space to the intersection line in parameter space by the duality of the point and the line. On the contrary, collinear points in image space always correspond to the line intersection in parameter space. The algorithms based on Hough Transform involve methods with gradient information, such as line detection algorithm introduced in literature [13], which uses segment voting weighted by surround suppression and gradient information. Some other algorithms like Randomized Hough Transform (RHT) [29], Fast Hough Transform (FHT) [18], Adaptive Hough Transform (AHT) [15], and Combinatorial Hough Transform (CHT) [3], which are suitable for sparse image. In 2000, Matas et al. proposed a Progressive Probabilistic Hough Transform (PPHT) [11, 20] that accelerates the calculation of HT by the selection of random edge points, and solves the computational redundancy of traditional HT. In 2008, a kernel-based voting algorithm was used for the HT [9], which improves the performance of the voting scheme, makes the transform more robust to the detection of spurious lines. But this method only got information of line in polar coordinates. In 2013, Soto-Pinto et al. [25] presented a new numerical method for automatic detection and analysis of changes in lineament patterns caused by seismic and volcanic activities, which used the HT to detect the main lineaments. In 2015, a framework for automatic enhancement and identification of the linear patterns of geological fault structures was proposed [23], which was demonstrated the higher performance and the robustness than HT.

HT needs the binary image as the input, so this kind of line detection algorithm is limited by the edge detection operator, and cannot get ideal results. To get out of this shortcoming, Rafael

[1] School of Electronics and Information, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China

[2] School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China

[3] College of Electrical & Information Engineering, Shaanxi University of Science & Technology, Xi'an, Shaanxi 710021, China

[4] School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China

et al. utilized the gradient and level-line of pixels, and proposed a linear-time line segment detector (LSD) with a false detection controlling [27] in 2010. LSD does not need binary image as the input, avoiding the loss of information in edge detection, and has a good real-time performance. In 2012, Akinlar et al. implemented a parameter free edge drawing (EDPF) algorithm [1, 2], and then combining it with the line verification method of false detection controlling in LSD, proposed a line segment detection method, named EDPFLines, which gives accurate results, requires no parameter tuning, and runs up to 11 times faster than the fastest line segment detector before, known as LSD [27]. But EDPLines is still based on edge detection, cannot step over the edge detection operator. The line segment based on mean shift makes full use of gradient vector to perform the extraction of line feature pixels in images, which can achieve the requirements in real time and line continuity. But it has a small and single searching scope in line growing, and is easy to miss some important line points that influence the continuity of the line. As for its deficiency, an improved algorithm of adaptive gradient threshold and omni-direction line growing method is proposed in this paper, referred to as LSWMSAllDir.

2 Line Segmentation based on weighted mean shift

In 2011, Nieto et al. introduced a new line segment detection approach, line segment detection using weighted mean shift (LSWMS) algorithm, by the use of weighted mean shift procedures on a 2D slice sampling strategy [22]. It works without any prior knowledge of image, and does not need the tuning of input parameters in the process of detecting lines. LSWMS speeds up the sample searching process with slice sampling, and refines the sample with weighted mean shift procedures. In this section, the theory of weighted mean shift is introduced firstly, and then the improved LSWMS method is discussed.

2.1 Weighted mean shift procedures of LSWMS

Firstly, LSWMS takes slice sampling to the gradient map of each image in order to obtain the sample points. In this process, it visits pixels in gradient map iteratively until pixel's gradient value satisfies $\hat{p}(x,y) > \mu$, where μ is the average of gradients $\hat{p}(x,y)$ in the whole map. And then it classifies the sample points according to its linear likelihood [22]. For every point $\boldsymbol{x}_k$, the probability $p(x,y)$ in which it belongs to one line can be calculated. However, there may be some other points in the neighborhood of current point. The enhancement step uses weighted mean shift algorithm (wMS) to find such points $\hat{\boldsymbol{x}}_k$. Mean shift algorithm is an iterative, non-parametric algorithm [14] that can be used to find local maxima from a set of available samples whose density function is unknown. Each iteration begins from the start point and searches along the most densely populated area among its neighborhood, then verifies whether this pixel can be replaced by a new one.

LSWMS uses the linear likelihood value of the sample to carry out the wMS algorithm. For a given sample $\boldsymbol{x}_k$, its related weighted factor is $\omega_k = \hat{p}(x,y)$. Therefore, the wMS can find the maximum $\hat{p}(x,y)$ in the neighborhood $\boldsymbol{z}_k$ of the sample $\boldsymbol{x}_k$. This is mainly because the probability $\hat{p}(x,y)$ of a position (x,y) on the image is uniformly distributed, but the weighting factor of wMS for each pixel is not uniform. And on the other hand, the direction searching procedure guides wMS towards those edge points whose directions are similar, and whose probability may not be the maximum $\hat{p}(x,y)$ in the neighborhood. A larger $\hat{p}(x,y)$ means that

the sample is more likely to belong to a straight line. The pixels belonging to a straight line with larger weights need to be processed by wMS.

When a set of samples, $\{\boldsymbol{x}_i\}_{i=1,2,3,\ldots,N}$, are considered, the position density$f(\boldsymbol{x})$ is unknown. Since the $f(\boldsymbol{x})$is important to calculate the probability, so the multivariate weighted kernel estimator of $f(\boldsymbol{x})$ is defined as [22]

$$\widehat{f}_{\omega}(\boldsymbol{x}) = \frac{1}{\sum_{i=1}^{N}\omega_i} \sum_{i=1}^{N} \omega_i \boldsymbol{K}_{\boldsymbol{H}}(\boldsymbol{x}-\boldsymbol{x}_i) \tag{1}$$

in which,

$$\boldsymbol{K}_{\boldsymbol{H}}(\boldsymbol{x}) = \frac{1}{|\boldsymbol{H}|^{1/2}} K\left(\frac{\boldsymbol{x}}{\boldsymbol{H}^{1/2}}\right) \tag{2}$$

where $\boldsymbol{H}$, a symmetric positive definite 3 × 3 matrix, specifies the "width" of the kernel at each dimension. The bandwidth matrix $\boldsymbol{H}$ is chosen for our approach as a diagonal matrix containing the corresponding bandwidth for each dimension: $\boldsymbol{H} = diag\left[h_x^2, h_y^2, h_\theta^2\right]$. K is obtained from the product of symmetric univariate kernels

$$K(\boldsymbol{x}) = \prod_{d=1}^{3} c_d \kappa(x_d) \tag{3}$$

where x_d (d = 1, 2, 3) are three components of the vector $\boldsymbol{x}$, c_d (d = 1, 2, 3) are normalized constants, and the function κ is the Epanechnikov kernel [6]. So a result can be calculated as follows:

$$\widehat{f}_{\omega}(\boldsymbol{x}) = \frac{|\boldsymbol{H}|^{-1/2} C}{\sum_{i=1}^{N}\omega_i} \sum_{i=1}^{N} \omega_i \prod_{d=1}^{3} \kappa\left(\left(\frac{x_d - x_{i,d}}{h_d}\right)^2\right) \tag{4}$$

where $x_{i,d}$ is the dth element in the sample $\boldsymbol{x}_i$, and h_d is corresponding bandwidth.

When the modes in $f(\boldsymbol{x})$ are located at the zeros of the gradient , $\nabla f(\boldsymbol{x})$, the estimator of $\nabla f(\boldsymbol{x})$ can be represented as follows:

$$\nabla\widehat{f}_{\omega}(\boldsymbol{x}) = \boldsymbol{A}\left[\sum_{i=1}^{N} \omega_i \boldsymbol{G}(\boldsymbol{x}-\boldsymbol{x}_i)\right] \times \left[\left(\sum_{i=1}^{N} \omega_i \boldsymbol{G}(\boldsymbol{x}-\boldsymbol{x}_i)\right)^{-1} \left(\sum_{i=1}^{N} \omega_i \boldsymbol{G}(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{x}_i\right) - \boldsymbol{x}\right] \tag{5}$$

where $\boldsymbol{A}$ is a matrix gathering the constants and the bandwidth matrix:

$$\boldsymbol{A} = \frac{2|\boldsymbol{H}|^{-1/2} C}{\sum_{i=1}^{N}\omega_i} \boldsymbol{H}^{-1} \tag{6}$$

and $\boldsymbol{G}(\boldsymbol{x})$ is a diagonal 3 × 3 matrix defined as:

$$\boldsymbol{G}(\boldsymbol{x}) = \mathrm{diag}\left[-\kappa'\left(\frac{x^2}{h_x^2}\right), -\kappa'\left(\frac{y^2}{h_y^2}\right), -\kappa'\left(\frac{\theta^2}{h_\theta^2}\right)\right] \tag{7}$$

where $\kappa^{'}$ is defined as the differentiation of the kernel κ. The last bracket in Eq. (5) is the mean shift vector, representing the difference between the weighted mean of the data samples and the center of the kernel:

$$m_\omega(\boldsymbol{x}_i) = \left(\sum_{i=1}^{N} \omega_i \boldsymbol{G}(\boldsymbol{x}-\boldsymbol{x}_i)\right)^{-1} \left(\sum_{i=1}^{N} \omega_i \boldsymbol{G}(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{x}_i\right) - \boldsymbol{x} \tag{8}$$

So the mean of samples is

$$S_{\text{mean}}(\boldsymbol{x}) = \left(\sum_{i=1}^{N} \omega_i \boldsymbol{G}(\boldsymbol{x}-\boldsymbol{x}_i)\right)^{-1} \left(\sum_{i=1}^{N} \omega_i \boldsymbol{G}(\boldsymbol{x}-\boldsymbol{x}_i)\boldsymbol{x}_i\right) \tag{9}$$

Through Eq. (5) and Eq. (8), Eq. (8) is rewritten as

$$m_\omega(\boldsymbol{x}_i) = \nabla \widehat{f}_\omega(\boldsymbol{x}) \times \boldsymbol{A}^{-1} \left(\sum_{i=1}^{N} \omega_i \boldsymbol{G}(\boldsymbol{x}-\boldsymbol{x}_i)\right)^{-1} \tag{10}$$

In Eq. (10), $\boldsymbol{G}$ is equivalent to the kernel K in Eq. (2). When Eq. (1) is substituted into Eq. (9), Eq. (8) is rewritten again as:

$$m_\omega(\boldsymbol{x}_i) = \frac{\sum_{i=1}^{N} \omega_i}{\boldsymbol{A}} \frac{\nabla \widehat{f}_\omega(\boldsymbol{x})}{\widehat{f}_\omega(\boldsymbol{x})} \tag{11}$$

It will be seen from Eq. (11) that the mean shift vector calculated at $\boldsymbol{x}_i$ by kernel $\boldsymbol{G}$ is positive proportional to the weighted kernel estimator at this position. The numerator of Eq. (11) is the weighted kernel estimator, i.e. the gradient estimator, representing the maximum density change direction. So it can be concluded that the direction of the mean shift vector always agrees with maximum density growth direction [30].

In order to express the physical meaning of m_ω more vividly, let $\boldsymbol{G}(\boldsymbol{x}) = 1$ for mean shift vector with $\omega_i = 1$. Eq. (8) is rewritten as Eq. (12).

$$m_\omega(\boldsymbol{x}_i) = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{x}) \tag{12}$$

Figure 1 illustrates the significance of the Eq. (12). The center of the circle (the larger black spot) in the figure is the position of $\boldsymbol{x}$, which is the kernel $\boldsymbol{G}$'s center. The small black spots around represent the sample points $\boldsymbol{x}_i$ and their arrows indicate the samples' offset vectors relative to the center. Average offset is pointing to the direction of dense sample points, i.e., gradient direction [5]. From the graph, it is obvious that the mean of sampling value $S_{mean}(\boldsymbol{x})$ is always changing along the sample point $\boldsymbol{x}_i$ that offsets most to the center point $\boldsymbol{x}$, i.e., the changing direction of the density gradient.

The mean shift procedure is a recursive process that calculates the mean shift vector $m_\omega(\boldsymbol{x}_i)$, and updates the central position of the kernel $\boldsymbol{G}$ according to $m_\omega(\boldsymbol{x}_i)$. Then the highest point of the probability distribution can be found through this recursive iteration, which is the local maximum point.

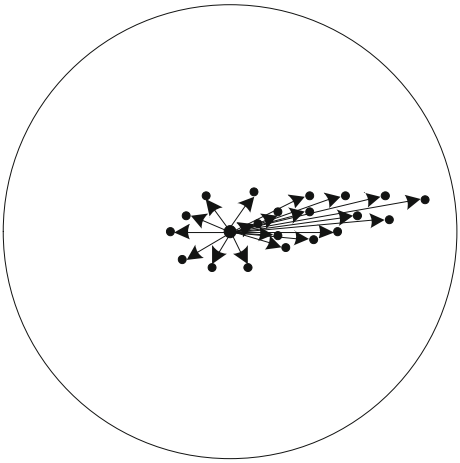


Fig. 1 Schematic diagram of mean shift

2.2 Calculation of adaptive gradient threshold

In literature [8], Desolneux used non-parametric method of Helmholtz principle to define and calculate image edge and boundary (close to the edge). In 2015, Lu utilized a similar method, proposing a non-parametric Canny edge detection operator [19], which defined a concept of "minimum meaningful gradient" and "maximum meaningful gradient" to determine the two thresholds of the Canny operator. It can extract more meaningful edge information through this kind of operator, which improves the performances of the LSD algorithm and EDLines algorithm. In LSWSM, the threshold gradient μ is initialized by the mean value and applied to the whole image. In this paper, since each image was divided into different line support regions according to line segment likelihood, an adaptive threshold μ' was used for each line support region rather than the whole image. The difference is that it only uses "minimum meaningful gradient" to calculate the value of μ', getting more complete and meaningful gradient information. When estimating the lengths of all the line segments, the longest one in the image is the diagonal one. The length of this longest line segment is $L_{\max}$, defined as:

$$L_{\max} = \sqrt{N^2 + M^2} \tag{13}$$

where M and N are the length and width of the image. At this condition, the whole image is a line edge segment, which contains $L_{\max}(L_{\max} - 1)$ edge chain points, i.e. the number of valid pixels for line segment N_p in literature [19]. In LSD, NFA is defined as the number of false events (segment detections) under a noise model. $NFA \leq 1$ means that the current edge is meaningful, otherwise it is meaningless. In the LSWMS algorithm, greater the gradient amplitude is, more obvious the straight line feature of the edge is. So it only need to calculate "minimum meaningful gradient" $l_{\min}$ and ignore "maximum meaningful gradient" $l_{\max}$. The $l_{\min}$ is:

$$l_{\min} = -4\log\left(N'\right)/\log(\rho) \tag{14}$$

where N' is larger one of the width and length of the image, ρ is a constant, which is always set to 1/8 according to LSD [19, 27]. The procedure of getting μ' is similar to the method in literature [19].

The specific steps of the algorithm are as follows:

(1) calculate the gradient magnitude image with Sobel operator;
(2) calculate the gradient magnitude histogram *Hist* by the step size of 1 pixel;
(3) calculate the probability distribution $P(i)$ corresponding to the gradient histogram;
(4) calculate N_p, $N_p = L_{\max} * (L_{\max} - 1)$;
(5) in the case of the assumption $NFA = 1$, calculate $\mathrm{Hist}(g_{\min}) = \left(1/N_p\right)^{1/l_{\min}}$, where $g_{\min}$ is the minimal meaningful gradient magnitude;
(6) start from $i = 1$, traverse histogram probability distribution $P(i)$, calculate the cumulative probability $p = p + p(i)$, where the initial value of p is 0, and the traversal stops when $p \geq Hist(g_{\min})$. Then we can think of $\mu' = i$.

This adaptive gradient threshold μ' introduced to replace the average gradient value μ in each region can prevent the pseudo noise and the loss of some meaningful edge pixels. After line support regions are obtained according to the likelihood of each pixel, μ' is calculated iteratively until all regions are visited. It achieves the purpose of reducing wrong lines in line growth process caused by the noise, and detecting more complete line edge.

2.3 Searching in Omni-direction for line growing

In LSWMS algorithm, the line growing process starts from a point (x, y) with gradient information (G_x, G_y), and then next point for line growing is searched along the direction and the opposite of the current. But the searching scope is single, easy to miss points near other directions but belonging to the current line, or current point in the direction may not be the best candidate point for the line. Thus, the detected line is not the exact line but deviated to interference points around. As shown in Fig. 2a, it is assumed that the current point (x, y) is the origin of coordinates. Then the direction of growing from this point can be direction 1, 2, 3, 4 or any directions. LSWMS searches along the direction of (x^+, y^+) (i.e., direction ①) and the opposite direction (x^-, y^-) (i.e., direction ②), while ignores points that may exist in other directions. So the default line direction is ① or ②. But the exact line may be the rest of the arbitrary direction (actually not arbitrary, but in some deviation along the main direction). This kind of growing method ignores the possibility of deviating direction, although growing is in the guide of gradient probability, but not explicit to rule out the condition.

Figure 2b is the region division for line growing. The basic idea to determine the direction of line growing is according to the gradient direction vector (G_x, G_y) of the current point. The tangent value and the direction angle of the vector (G_x, G_y) are calculated to determine which region the current point belongs to. In theory, all directions are possible, and the area classification can save the searching time. In the original algorithm, the main directions are ①/②, ③/④, x axis, and y axis, each point is assigned to one of the four directions. Differ from it, an improved method is proposed here. Firstly, the right side is divided into three parts: y^--④, ④-① and ①-y^+, the corresponding angle region are $(-\pi/2, -\pi/4]$, $(-\pi/4, \pi/4]$ and $(\pi/4, \pi/2)$. And we believe that line growing directions corresponding to the three angle regions are ③/④, x^+ and ①/②. The left side is divided symmetrically. Thus, two regions are obtained: one is surrounded by ①/② and y (i.e. the shaded part of vertical lines), while the other is ③/④ and y (i.e. the shaded part of horizontal lines). It not only ensures the characteristics of a straight line, but also takes all possible directions in a scope into account. Moreover, the region ①-x^+-④ represents a region with the intersection angle of 90 degrees (i.e. a blank area

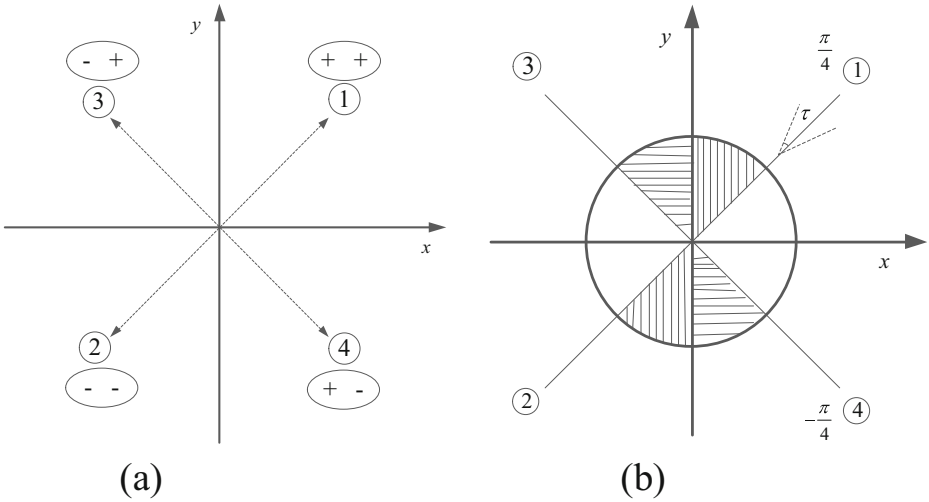


Fig. 2 The region of line growing **a** is the direction of growing; **b** is the region division

surrounded by ① and ④, including x^+, blank area at right side), different from the previous two areas. If divided individually, it will lose the characteristics of straight line growing. But if the region ①-x^+-④ and its symmetric region ②-x^--③ are merged, the span is too large, which is easy to introduce the non-line point, interfere the point set of straight line.

For the rest of the part, if ③-x^- and its symmetric region ④-x^+, or ②-x^- to ①-x^+, are merged as one region, it will be confused with the area divided before. So the direction x^+, x^- were set as benchmark direction of line growing separately to detect a group of straight line results. Then we combine the result with results of the rest directions to realize the detection in omni-direction. Its results significantly improve those with the original algorithm of single direction detection. The only drawback is that its time consumption of growing is twice of original algorithm's. In the process of line growing, line point is searched along main direction within a small angle range. As illustrate in Fig.2b, the tolerance angle is τ, which is 15 degrees referring to literature [27]. It could also be adjusted according to personal data.

During this process, searching considers the current point merely, ignores the direction of prior and next point in the line support region. When these two points are considered, orientation principle is preferred. As shown in Fig.3a, three direction arrows represent the current point, the prior one, and the next respectively. The angle between the prior and the current point is $\angle A$. Similarly, the angle between the current point and the next point is $\angle B$. If $\angle A < \angle B$ (Fig.3a), growing will be along the direction of prior point. Otherwise, if $\angle A \geq \angle B$ (Fig. 3b), growing will be along the direction of current point. Considering the direction association of prior and next points can avoid missing correct pixels, check and correct line detection to tend to the exact line simultaneously, improve the accuracy of results further.

Figure 4 is the flowchart of the proposed algorithm, LSWMSAllDir. The overall framework is similar to the flowchart of the algorithm introduced in literature [22]. One improvement point of our method is the calculation of adaptive μ' for each line support region [28]. The other point is the line growing procedure in the dashed frame as Fig.4, which uses omni-direction line growing method to search line points belonging to a line as many as possible. It can get a series of line points in every line support region R_i that is divided according to line segment likelihood from gradient map.

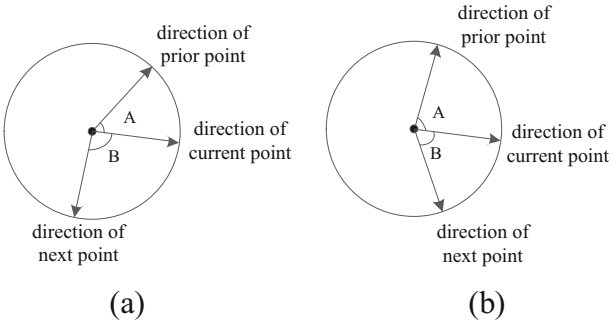


Fig. 3 Preferred orientations. **a** is$\angle A < \angle B$; **b** is $\angle A > \angle B$

2.4 Evaluation methods

To objective evaluate the performance of these line detection algorithms, the pixel overlap rate and false detection rate of line are introduced here.

The overlap rate of line pixels R_o is defined as:

$$R_o = \frac{N_o}{N_r} \tag{15}$$

where N_o represents the number of pixels that have the same positions in detected line segments as those in the ground truth line segments, i.e. the number of overlapped pixels at the same position. N_r is the number of pixels in the ground truth line segments. Larger the overlap rate is, better the detection results are.

In addition, detected line pixels that cannot be found in ground truth line segments also need to be considered to evaluate the performance of the algorithm. The false detection rate, R_{error}, is defined as:

$$R_{\text{error}} = \frac{N_d - N_o}{N_r} \tag{16}$$

where N_d is the total number of pixels on detected line segments.$N_d - N_o$ indicates the number of line pixels detected whose positions are not agreed with pixels' positions in the ground truth line segments. Lower the false detection rate is, better the detection results are. R_{error} may be

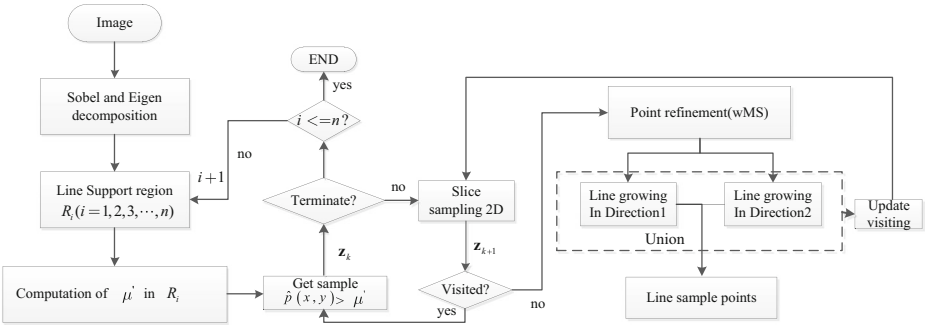


Fig. 4 Flowchart of LSWMSAllDir

larger than 1, which means that the number of false pixels is larger than the number of pixels in the ground truth line segments. Here, N_r is used as the denominator instead of N_d because it is a fixed value for each image.

3 Experiments and results

In order to compare the performance of LSWMSAlldir and current main line detection algorithms, OpenCV 2.4.9 are configured in Studio 2010 compiler and Windows 7 system. All the algorithms are run on 16G Lenovo S30 Think Station.

3.1 Results on synthetic data

In this paper, synthetic data is used as in Fig. 5(a), 18 black line segments with single pixel width are drawn on a 100 pixels × 100 pixels image with white background. The intersection angle between two adjacent line segments is 20 degree. The length of each line segment is 40 pixels. The detection results (red line segments) of PPHT, LSD, EDPFLines, LSWMS and LSWMSAllDir algorithms can be seen in Fig. 5(b)~(f). The overlap rate, false detection rate and running time of line segments detected are shown in Table 1.

From Fig.5, it can be seen that the last four algorithms almost detect all the line segments. However, the red line segments detected by the LSD and EDPFLines algorithms are double. As shown in Table 1, although LSD and EDPFLines algorithms have higher value regarding to the overlap rate, the false detection rate is higher too because of the double line segments. By contrast, the proposed algorithm have a high overlap rate and relatively low false detection rate.

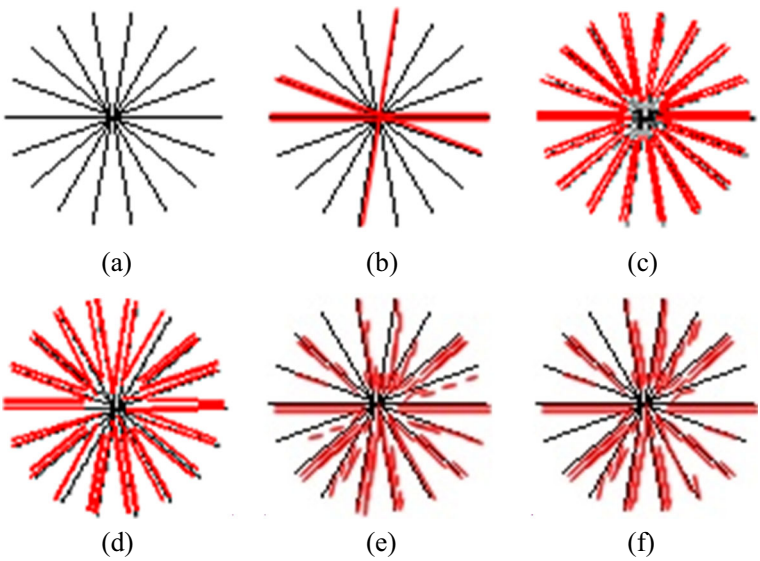


Fig. 5 The synthetic data and line detection results **a** is synthetic data; **b** is the result of PPHT; **c** is the result of LSD; **d** is the result of EDPFLines; **e** is the result of LSWMS; **f** is the result of LSWMSAllDir

Table 1 Values of evaluation metrics

Algorithms	Overlap rate	Flase detection rate	Running time(s)
PPHT	0.3722	0.1719	0.018439
LSD	0.858	1.0009	0.105903
EDPFLines	0.8454	1.0245	0.019262
LSWMS	0.6814	0.6404	0.125903
LSWMSAllDir	0.8466	0.8648	0.091336

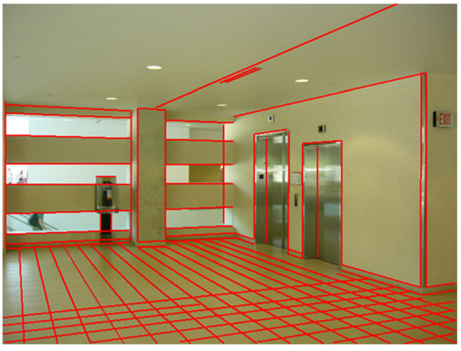
3.2 Results on standard line segment images

In this paper, the standard line segment images in York Urban Database (YUDB) (http://www.elderlab.yorku.ca/YorkUrbanDB/) are also used as the test images to compare the performance of the proposed and original algorithms. The York Urban Database consists of 102 (45 indoor and 57 outdoor scenes) images of urban environments scenes mostly from the campus of York University and downtown Toronto, Canada. The images are 640 pixels × 480 pixels in size and were taken with a Panasonic Lumix DMC-LC80 digital camera [7]. The author of data developed an interactive MATLAB program to manually identify lines in the image with sub-pixel precision as the ground truth. As shown in Fig. 6, the red lines plotted on the image are the ground truth lines described by the author.

In this paper, the line detection results of PPHT, LSD, EDPFLines, LSWMS and LSWMSAllDir algorithms are compared in the subjective and the objective aspects. In subjective aspect, the detection results are compared with the ground truth lines visually. Figure 7 is the detection result of these algorithms. The proposed LSWMSAllDir algorithm has a great advantage in the line accuracy and anti-interference of the noise. More details can be detected accurately from Fig.7(e).

To prove the effect of the proposed algorithm and its two improved aspects, the adaptive gradient threshold and Omni-direction searching for line growing, the mean value of overlap rate, false detection rate, running time of the whole YUDB of 102 images are calculated on PPHT, LSD, EDPFLines, LSWMS, LSWMS-1 (only with adaptive gradient), LSWMS-2 (only with Omni-direction searching) and LSWMSAllDir algorithms. As shown in Table 2, the proposed algorithm has the best performance on the accuracy. For the overlap rate, LSWMSAllDir has a higher value than others.

Fig. 6 Ground truth lines



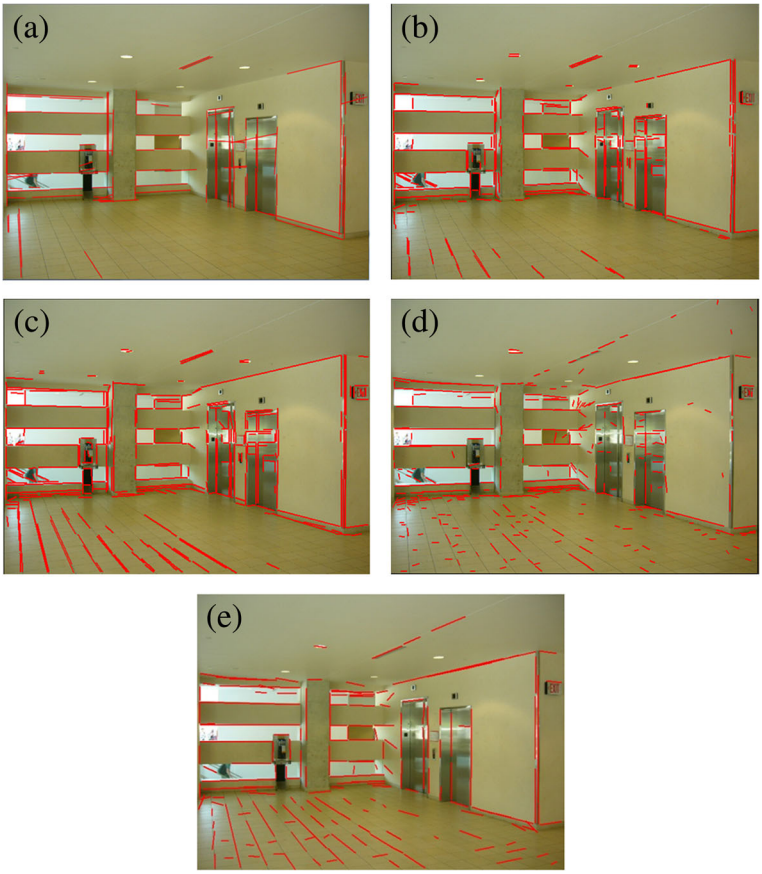


Fig. 7 Results of all algorithms. **a** is the result of PPHT; **b** is the result of LSD; **c** is the result of EDPFLine; **d** is the result of LSWMS; **e** is the result of LSWMSAllDir

4 Conclusion

In this paper, the principle and process of weighted mean shift algorithm are introduced firstly. LSWMS algorithm is non-supervised and non-parametric without prior knowledge. And it is based on the sampling strategy of candidate line points to improve the speed of the algorithm,

Table 2 Average values of evaluation metrics

Algorithms	Overlap rate	Flase detection rate	Running time(s)
PPHT	0.5209	3.4945	0.0420
LSD	0.5710	1.8506	0.1663
EDPFLines	0.5830	1.9048	0.0306
LSWMS	0.3634	1.3200	0.0702
LSWMS-1	0.3685	1.3687	0.0519
LSWMS-2	0.5817	2.2784	0.0711
LSWMSAllDir	0.5841	2.3741	0.0692

reduce the amount of computation. But the growing procedure ignores the possibility of the tiny angle deviation of lines. So an improved LSWMS algorithm (LSWMSAllDir) is proposed, which uses omni-direction searching for line growing based on adaptive gradient threshold. Experiments on synthetic data and standard real scene images proved the advantage of the proposed algorithm in the accuracy.

Acknowledgments This work was supported by the National Nature Science Foundation of China (Grant Nos. 61461025, 61402371); Natural Science Basic Research Plan in Shaanxi Province of China (Grant No. 2015JM6317, 2013JQ8039); Fundamental Research Funds for the Central Universities (Grant No. 3102014JCQ01060); NPU Foundation for Fundamental Research (Grant No. JCY20130130); The Seed Foundation of Innovation and Creation for Graduate Students in NPU (Grant No. Z2016024, Z2016121).

References

1. Akinlar C, Topal C (2011) EDLines: a real-time line segment detector with a false detection control. Pattern Recogn Lett 32(13):1633–1642
2. Akinlar C, Topal C (2012) EDPF: a real-time parameter-free edge segment detector with a false detection control. Int J Pattern Recognit Artif Intell 26(1):3898–3898
3. Ben-Tzvi D, Sandler MB (1990) A combinatorial Hough transform. Pattern Recogn Lett 11(3): 167–174
4. Caprile B, Torre V (1990) Using vanishing points for camera calibration. Int J Comput Vis 4(2): 127–139
5. Collins R T. (2003) Mean-shift blob tracking through scale space[C]//Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 I.E. Computer Society Conference on. IEEE, 2: II-234-40 vol. 2
6. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619
7. Denis P, Elder JH, Estrada F (2008) Efficient Edge-Based Methods for Estimating Manhattan Frames in Urban Imagery[M]. Springer, Berlin Heidelberg, pp. 197–210
8. Desolneux A, Moisan L, Morel JM (1999) Meaningful alignments. Int J Comput Vis 40(1):7–23
9. Fernandes LA, Oliveira MM (2008) Real-time line detection through an improved hough transform voting scheme, pattern recognition. Pattern Recogn 41:299–314
10. Fränti P, Ageenko EI, Kälviäinen H, et al. (1998) Compression of line drawing images using Hough transform for exploiting global dependencies. Proc 4th Jt Conf Inf Sci (JCIS'98) 1998:433–436
11. Galambos C, Kittler J, Matas J (2001) Gradient based progressive probabilistic Hough transform//Vision, Image and Signal Processing, IEE Proceedings. IET 148(3):158–165
12. Grompone R, Jakubowicz J. (2007) Geometry-based unsupervised urban-area detection. IEEE Geoscience and Remote Sensing Letters
13. Guo S, Kong Y, Tang Q, et al. (2008) Hough transform for line detection using segment voting weighted by surround suppression. Visual Information Engineering, 2008. VIE 2008. 5th International Conference on. IET,:47–51.
14. Han B, Comaniciu D, Zhu Y, Davis LS (2008) Sequential kerneldensity approximation and its application to real-time visualtracking. IEEE Trans Pattern Anal Mach Intell 30(7):1186–1197
15. Illingworth J, Kittler J (1987) The adaptive Hough transform. IEEE Trans Pattern Anal Mach Intell 5:690–698
16. Karnieli A, Meisels A, Fisher L, et al. (1996) Automatic extraction and evaluation of geological linear features from digital remote sensing data using a Hough transform. Photogramm Eng Remote Sens 62(5):525–531
17. Lee H J, Ahn H J, Song J H, et al.(2001) Hough transform for line and plane detection based on the conjugate formulation//photonics west 2001-electronic imaging. International Society for Optics and Photonics: 2e44–252
18. Li H, Lavin MA, Le Master RJ (1986) Fast Hough transform: A hierarchical approach. Comput Vis, Graph Image Process 36(2):139–161
19. Lu X, Yao J, Li K, et al. (2015) CANNYLINES: A PARAMETER-FREE LINE SEGMENT DETECTOR Image Processing (ICIP), 2015 I.E. International Conference on. IEEE 2015:507–511
20. Matas J, Galambos C, Kittler J (2000) Robust detection of lines using the progressive probabilistic Hough transform. Comput Vis Image Underst 78(1):119–137

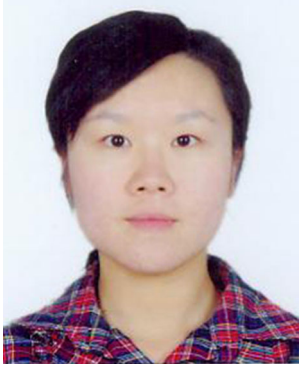
21. Meksen T M, Boudraa M, Drai R. (2006) Detection of cracks in materials using the randomized Hough transform on ultrasonic images//Proc. of the 6th WSEAS Int. Conf. on Signal Processing, Computational Geometry & Artificial Vision: 202–206
22. Nieto M, Cuevas C, Salgado L, et al. (2011) Line segment detection using weighted mean shift procedures on a 2D slice sampling strategy. Pattern Anal Applic 14(2):149–163
23. Panagiotakis C, Kokinou E (2015) Linear pattern detection of geological faults via a topology and shape optimization method, IEEE trans. On geoscience and. Remote Sens 8(1):3–12
24. San DK, Turker M (2010) Building extraction from high resolution satellite images using Hough transform. International archives of the photogrammetry, remote sensing and spatial information. Science 38(1):1063–1068
25. Soto-Pinto C, Arellano-Baeza A, Sánchez G (2013) A new code for automatic detection and analysis of the lineament patterns for geophysical and geological purposes (ADALGEO). Comput Geosci 57:93–103
26. Strzodka R, Ihrke I, Magnor M (2003) A graphics hardware implementation of the generalized hough transform for fast object recognition, scale, and 3d pose detection//image analysis and processing, 2003. Proceedings. 12th international conference on. IEEE:188–193
27. Von Gioi RG, Jakubowicz J, Morel JM, et al. (2010) LSD: a fast line segment detector with a false detection control[J. IEEE Trans Pattern Anal Mach Intell 32(4):722–732
28. Voon LFLY, Bolland P, Laligant O, et al. (1997) Gradient-based Hough transform for the detection and characterization of defects during nondestructive inspection[C]//electronic Imaging'97. International Society for Optics and. Photonics:140–146
29. Xu L, Oja E, Kultanen P (1990) A new curve detection method: randomized Hough transform (RHT). Pattern Recogn Lett 11(5):331–338
30. Yang C, Duraiswami R, Davis L (2004) Similarity Measure for Nonparametric Kernel Density Based Object Tracking. In Eighteenth Conf Neural Inf Proces Syst 2004:13–16

Yi Wang is an Associate Professor at School of Electronics and Information, Northwestern Polytechnical University, P. R. China. She received her Ph. D degree in Signal and Information Processing from the same university in 2007. Currently her research focuses on image processing and artificial intelligence. She has published over 80 academic articles. She is a member of council and Deputy secretary general in Shaanxi Provincial Association of Image & Graphics as well as a member of council in Shaanxi Provincial Association of Biomedical Engineering. Email: wangyi79@nwpu.edu.cn.

Liangliang Yu is graduate student at School of Electronics and Information, Northwestern Polytechnical University, P. R. China. Currently his research focuses on image processing, virtual reality. E-mail: yuliangliang@mail.nwpu.edu.cn.

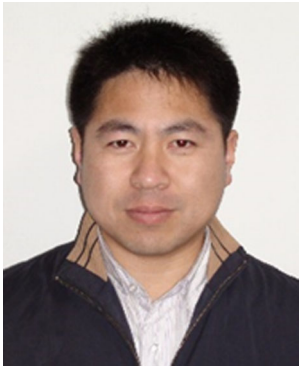
Houqi Xie is graduate student at School of Electronics and Information, Northwestern Polytechnical University, P. R. China. Currently his research focuses on image processing. E-mail: jxfcxhq@163.com.

Tao Lei received his PhD degree in information and communication engineering from Northwestern Polytechnical University, Xi'an, in 2011. He is currently a postdoctor at school of computer science, Northwestern Polytechnical University. He is also a professor at college of electrical and information Engineering, Shaanxi University of science & technology. His current research interests include image processing, computer vision and pattern recognition. Dr. Lei is a member of IEEE SPS, CCF (China). E-mail: leitaoly@163.com.



Zhe Guo was awarded the B.S., M.S., and Ph.D. degree in computer science from Northwestern Polytechnical University in 2005, 2008, and 2012. She is now a lecturer of Northwestern Polytechnical University. Her research interests include virtual reality, bioinformatics, and computer vision. E-mail: guozhe@nwpu.edu.cn.

Min Qi got Ph. D in systems engineering, Northwestern Polytechnical University in 2000. The research interests include image processing, pattern recognition and virtual reality. She is an associate professor in signal and information processing, school of electronics and information, Northwestern Polytechnical University, and published the book *Introduction to Pattern Recognition* (Beijing, China: Tsinghua University Press,2009). She is the secretary-general and executive member of the council, Shaanxi Society of Image and Graphics, P. R. China. She is also the member of the technical committee of Virtual Reality and Visualization Technology, China Computer Federation. E-mail: drqimin@nwpu.edu.cn.



Guoyun Lv is an Associate Professor at School of Electronics and Information, Northwestern Polytechnical University, P. R. China. He received his Ph. D degree in computer science and technology from the same university in 2008. Currently his research interests include image processing, virtual reality, and speech signal processing. E-mail: lvguoyun101@nwpu.edu.cn.

Yangyu Fan received his Ph.D. from Northwestern Polytechnical University in December 1999. He is currently a Professor at the School of Electronics and Information of the University and the head of Multimedia & Virtual Reality Laboratory, the Institute of Signal Processing &Wireless optical Communication. He has published more than 300 papers on journals and conferences, got 4 invent patents in China. His research interests include image processing, signal processing, virtual reality, pattern recognition, wireless optical signal processing. He is a member of the Information Processing Branch Association and the DSP Branch Association of China Electronics Society, one of the principals of the Shaanxi Signal Processing Society. E-mail: fan_yangyu@nwpu.edu.cn

Yilong Niu Yilong Niu is an Associate Professor at School of Marine Science and Technology, Northwestern Polytechnical University, P. R. China. He received his PhD. degree in Signal and Information Processing from the same university in 2009. His research interest is signal processing, target detection and artificial intelligence. E-mail: yilong_niu@nwpu.edu.cn