A Novel Line Detection Algorithm based on Endpoints Estimation

Weili Ding
School of Electrical Engineering
Yanshan University,YSU
Qinhuangdao, China

Wenfeng Wang
College of Vehicles and Energy
Yanshan University,YSU
Qinhuangdao, China

Abstract—**Line segment is the most important feature for shape analysis and object recognition. In this study, a novel line detection method starting from endpoints is proposed. Firstly, the junctions and breakpoints are detected from the edge map. Then the connected edge pixels between any two junctions or breakpoints will be extracted, and taken as a line segment or a curve by using small eigenvalue analysis. Next, the corners are detected from each curve. In this way, each curve is divided into several parts by the corners, and each part between two corners is identified by using the small eigenvalue analysis. We have tested our algorithm on a wide collection of images and compared it with a lot of proposed line detectors. The results show that our algorithm is efficient for extracting line segments from an image.**

Keywords-Line detection;corner detection; small eigenvalue analysis;principal component analysis

I. INTRODUCTION

Line segment is the most basic feature for shape and object identification. How to extract them from a digital image is a classic problem which has been studied for decades.

Hough transform (HT)[1] is the most commonly used method. The main idea of HT is "parameter space transform", which projects the edge pixels on a X-Y plane onto the n-D parameter space to find the predefined shape by a voting procedure. As a feature extraction technique, HT is practical, general and is effective to extract the lines from the noised images, however the length and end-point positions of the line segment are lost after transformation, and the peaks are difficult to be determined, so it is very hard to estimate accurately the line parameters. Therefore, a lot of improved algorithms have been proposed in the last two decades, such as fast HT [2], adaptive HT [3], randomized HT [4], multi-resolution HT [5], and others [6]. However, all improved methods are not available to identify whether the extracted line segments belong to a part of a smooth curve when the equation of the curve is not given.

Region Growing (RG) is another technique commonly used in the proposed line detection algorithms. The RG-based line detection methods use the connected gray pixels or edge pixels with similar gradient orientations, and then combine them into a line with a certain criterion. For example, the algorithm proposed by Burns et al[7] combines the connected gray pixels into a line-support region firstly, and detects the lines with the straightness criterion. However, Burns' method is sensitive to noise, so a huge number of short lines will be generated during the line detection, so a few improved algorithms are proposed, and LSD[8] is the representive one. LSD is real-time algorithm with a false detection control criterion [8], but it takes smooth curves as the concatenations of small straight line segments. Another classic algorithm in this group starts edges, and then the chosen set of edge points will be determined whether they lie in a straight line or a curve by using orientation consistency[9], least square fitting [10] or the small eigenvalue analysis[11, 12]. These algorithms can extract most of line segments from complex images with a satisfying accuracy rate, but its computational time is long and its effect depends much on the chosen set of edge points.

In this study, we propose an efficient method which starts from endpoints estimation. Here, the breakpoints and junction points are taken as the endpoints and estimated firstly. Then the corners are estimated and taken as another type of endpoints. Finally, the connected edge pixels between any two estimated endpoints will be taken as a straight line or a smooth curve by using the small eigenvalue analysis.

Section 2 gives an overall diagram of the proposed algorithm, including the endpoints estimation, edge decomposition and edge segment identification. In the Section 3, the endpoints estimation algorithm is described in detail firstly; and then the proposed line detection algorithm is introduced. The experimental results and discussion on the performance of the proposed algorithm are given in Section 4. And the conclusion is given in Section 5.

II. OVERALL DIAGRAM OF THE ALGORITHM

Given an input image *I* and its edge map *E* detected by the Canny detector [13], we observe that if an edge pixel is a key point such as a junction, a breakpoint, a corner, it is more likely an endpoint of a straight line or a smooth curve in the image. So we take these points as the most probable endpoints and break the edge map at these endpoints, and then identify the connected edge pixels between two estimated endpoints as a line segment or a smooth curve.

Fig.1 shows the process of line detection in our algorithm, which mainly includes two levels of detection. In the first level, the junctions and the breakpoints will be detected in the edge

This work was supported by NSFC (61005034), the Natural Science Foundation of Hebei Province (F2012203185) and the Qinhuangdao city science and technology research and development plan(2012021A054).

.

978-1-4799-2764-7/13/$31.00 ©2013 IEEE

map, and the connected edge pixels between any two junctions or breakpoints will be traced, and taken as an edge chain. Then the straight lines contained in the edge chains will be identified firstly. In the second level, the corners contained in each curved chain will be detected, and the edge chain will be further broken into several parts by corners. Here, we call each part as an edge segment. Because there is no corner in the edge segment, the edge segment may be a straight line segment or a smooth curve, and we identified the line segment finally.

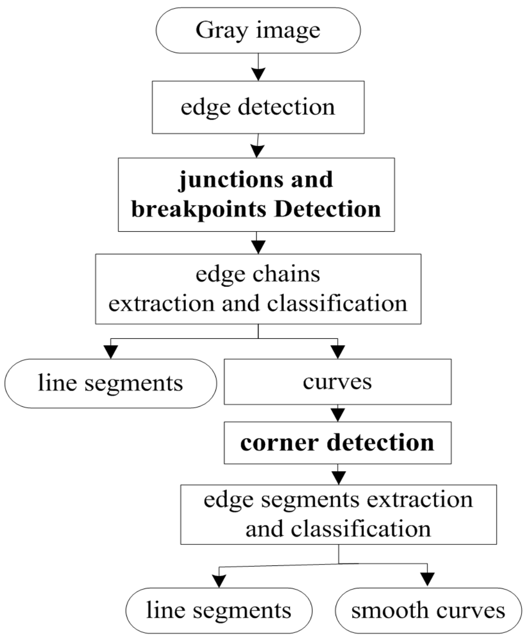

Figure.1. Processing of line detection in our algorithm.

III. ENDPOINTS ESTIMATION AND LINE DETECTION

A. *Endpoints estimation and edge decomposition*

Given an input image I and its edge map E detected by the Canny detector. We estimate the junctions and breakpoints in the following way[14].

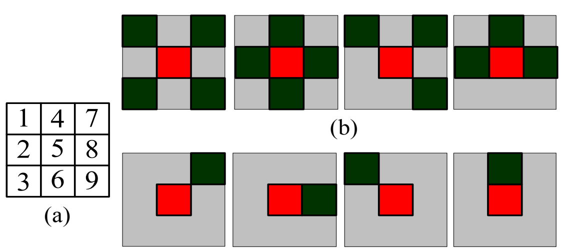

Figure 2. Diagram of the points. a) The label of pixels; b)junction points; c) breakpoints.

As shown in Fig.2, we consider a non-zero pixel (the red pixel in Fig.2.b and Fig.2.c) in the edge map E and its 3-by-3 neighborhood whose label is shown in Fig.2.a. Then we define the following function:

$$f(x) = \sum_{j=1}^{8} (|a_j - b_j|), \quad (1)$$

where $a = [x(1), x(2), x(3), x(6), x(9), x(8), x(7), x(4)]$, $b = [x(2), x(3), x(6), x(9), x(8), x(7), x(4), x(1)]$ and $x(i) = \{0,1\}, (0 < i \le 9)$ is the value of each pixel in the 3-by-3 neighborhood matrix. The non-zero pixel x is a junction point (See fig.2.b) when $f(x) \ge 6$, and it is a breakpoint(See fig.2.b) when $f(x) = 2$.

After detecting the junctions and breakpoint, a set of edge chains will be extracted from the edge map by using the edge linking algorithm[14]. Here, we denote the edge chain as $e_i, (0 < i \le n)$, where n is the pixels number of the extracted edge chain. Then CSS algorithm[15] is used to detect corners in the edge chain, and we set the parameters of CSS as the default values. And further, if we define m_i is the number of corners and the stagnation points in each edge chain, the edge chain will be further broken into several edge segments $e_{ik}, (0 < k \le m_i)$, and the edge map E can be rewritten as a set of edge segments:

$$E = \sum_{i=1}^{n} \sum_{t=1}^{m_i+1} e_{it}, n \ge 1, m_i \ge 1, \quad (2)$$

where the edge segments satisfy the following constrains:

$$e_{it} \subset e_i, e_i \cap e_j = \Phi, e_{it} \cap e_{is} = \Phi, (1 \le i, j \le n, 1 \le t, s \le m_i) \quad (3)$$

Fig.3 shows an example of endpoints estimation and edge decomposition in our algorithm. The junctions and the breakpoints are shown as the red and green crosses, and each edge chain is shown with a different color. From Fig.3.b we see that the blue edge chain is a straight line, and other edge chains are curves. In Fig.3.c, the red edge chain in Fig.4.b is considered, and the detected corners are shown as the blue points. We see that the edge chain is divided into 7 parts by the corners, and most of them are straight lines. Therefore, it is reasonable to decompose the connected edge pixels at corners, junctions and breakpoints, and it is easier to extract straight line from edge segments.

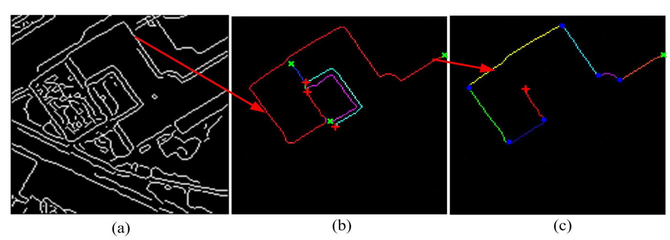

Figure 3. Endpoints estimation and edge decomposition. (a) The edge map extracted by canny detector; (b) Breakpoints, junctions estimation and edge chains tracked by using edge linking algorithm [14]; (c) Corners estimation and edge segments extracted from the red edge chain in (b).

B. *Straight line identification criteria*

Given an edge segment $e_{ik} = \{p(x_{ik}, y_{ik}) | k = 1 \cdots m_i\}$, its covariance matrix can be described by [11]:

$$\mathbf{C}=\begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \tag{4}$$

where $s_{11}=\dfrac{1}{m_i}\sum_{k=1,\ldots,m_i}(x_{ik}-x_m)^2$, $s_{22}=\dfrac{1}{m_i}\sum_{t=1,\ldots,m_i}(y_{ik}-y_m)^2$,

$s_{12}=s_{21}=\dfrac{1}{m_i}\sum_{k=1,\ldots,m_i}(x_{ik}-x_m)(y_{ik}-y_m)$,

and $x_m=\dfrac{1}{m_i}\sum_{t=1,\ldots,m_i}x_{ik}$, $y_m=\dfrac{1}{m_i}\sum_{t=1,\ldots,m_i}y_{ik}$.

The eigenvectors and eigenvalues of $\mathbf{C}$ can be calculated as follows:

$$\begin{cases} \mathbf{p}=[\mathbf{p}_1,\mathbf{p}_2]=\begin{bmatrix} \dfrac{s_{12}}{\sqrt{(\lambda_1-s_{11})^2+s_{12}^2}} & \dfrac{s_{12}}{\sqrt{(\lambda_2-s_{11})^2+s_{12}^2}} \\ \dfrac{\lambda_1-s_{11}}{\sqrt{(\lambda_1-s_{11})^2+s_{12}^2}} & \dfrac{\lambda_2-s_{11}}{\sqrt{(\lambda_2-s_{11})^2+s_{12}^2}} \end{bmatrix} \\ \lambda_i=\dfrac{1}{2}\left(s_{11}+s_{22}\pm\sqrt{(s_{11}-s_{22})^2+4s_{12}^2}\right),(i=1,2)(\lambda_1>\lambda_2) \end{cases} \tag{5}$$

where $\mathbf{p}_1$ is the main direction, $\mathbf{p}_2\perp\mathbf{p}_1$, and λ_1,λ_2 means the length of the base axes. In [11] and [12], the $\lambda_2\le T$ (is T a threshold value close to zero) can be taken as a sufficient and necessary condition of the straight line detection. So we use this condition to identify whether an edge segment is a straight line or not. That is, *if the small eigenvalue of the covariance matrix of any edge segment is less than* T*, it will be taken as a straight line, otherwise it will be taken as a smooth curve.* Here, T is an important parameter to identify the straight line, we set it to 0.25, because this value has been discussed in [12] as a suitable value for line detection.

C. Line detection

Strictly speaking, not all corners detected by CSS are the true endpoints of the straight line. As shown in Fig.4, we divided the detected corners on the edge chain into two types: L-corner and S-corner. The L-corner is defined as the corner connected with two straight lines (See Fig.4.a), and S-corner is defined as the corner connected with one or two curves (See Fig.4.b).

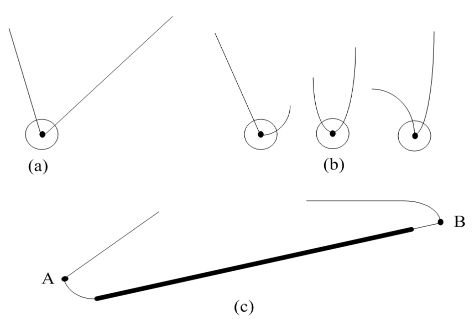

Figure 4.Type of corner. a) L-corner,which is the corner connected with two straight lines; b) S-corner,which is the corner connected with one or two smooth curves; c) the straight line between S-corners.

It is obvious that if the edge segment starts with an S-corner (e.g the edge segment between point A and point B in Fig.4.c), it will not satisfy the small eigenvalue condition, because the edge segment is curved around the S-corner. In order to avoid this case, we only consider the small eigenvalue of the points in the middle of the edge segment. In this study, we go over all edge segment, and compute the small eigenvalue of each edge segment as $e_{ik}=\{p(x_{ik},y_{ik})\,|\,k=m_i/5\cdots4m_i/5)$, if its small eigenvalue is less then T, we will take it as an straight line.

IV. EXPERIMENTAL RESULTS

We have tested our proposed algorithm with different kinds, size and noise levels images, which were downloaded from various research papers. Fig.5 shows some results.We see that almost all long and short line segments are founded, and the line segment results are similar with human vision system because of the smooth curves are filtered successfully from the images.

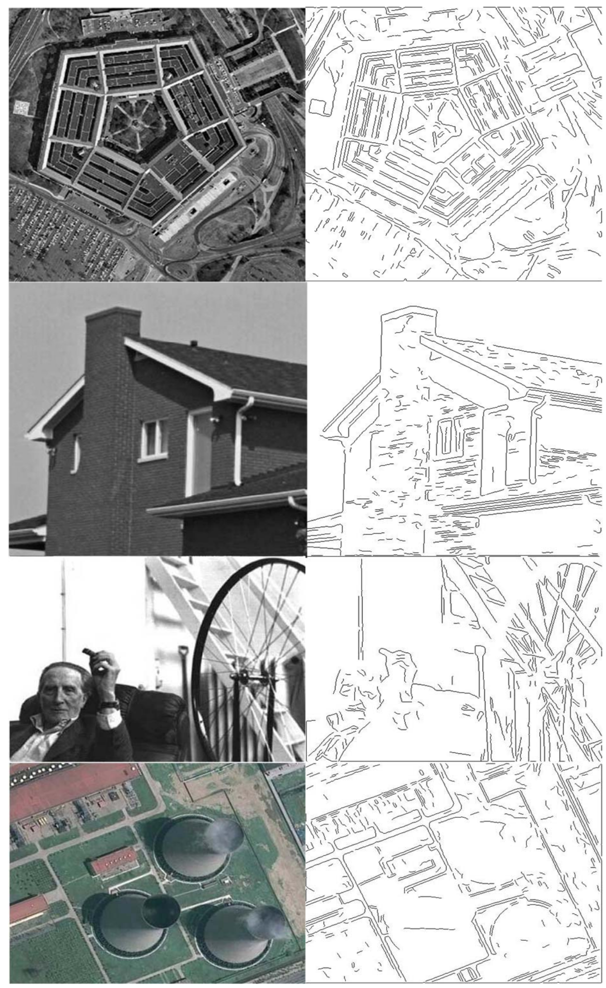

Figure 5.Results of the proposed algorithm on different kinds of nature images.

Fig.6 shows a comparison of various line detection algorithms including Standard HT[10], Burns method[18], , the algorithm using principal componentan analysis(PCA)[12], the algorithm with small eigenvalue analysis(SEA)[11], LSD [8], Edlines[9] and our proposed algorithm. As shown in the figure, we can see that the classic algorithm such as SHT and Burns et al extract line segments with a high false rate. SHT takes the texture structure as a long straight line and misses some short lines, while Burns method extract too many short lines. Other algorithms proposed in recent years such as LSD, EDlines, PCA, and SEA give similar line detection results with low false rate, but they take curved edges as a group of line segments and fail to distinguish between smooth curves and line segments. Compared with them, our algorithm extract the true straight line exactly.

Figure 6. A comparison of seven line detection methods

Fig.7 demonstrates the effects of gauss noise in line segment detection by different algorithms. When the gauss noises with different variances are added in the test images (As shown in Fig.7.a and Fig.7.b), the straight lines are extracted by using five algorithms proposed in recent years. We note that a few line segments can be detected by Guru's method and LSD algorithm, and Lee's method can extract more line segments because it takes some noise edges as line segments. EDlines algorithm and our algorithm shows an encouraging result, but EDlines algorithm takes arc segments as line segments. Our algorithm shows a good result in removing the smooth curves at all resolutions, so it is still promising.

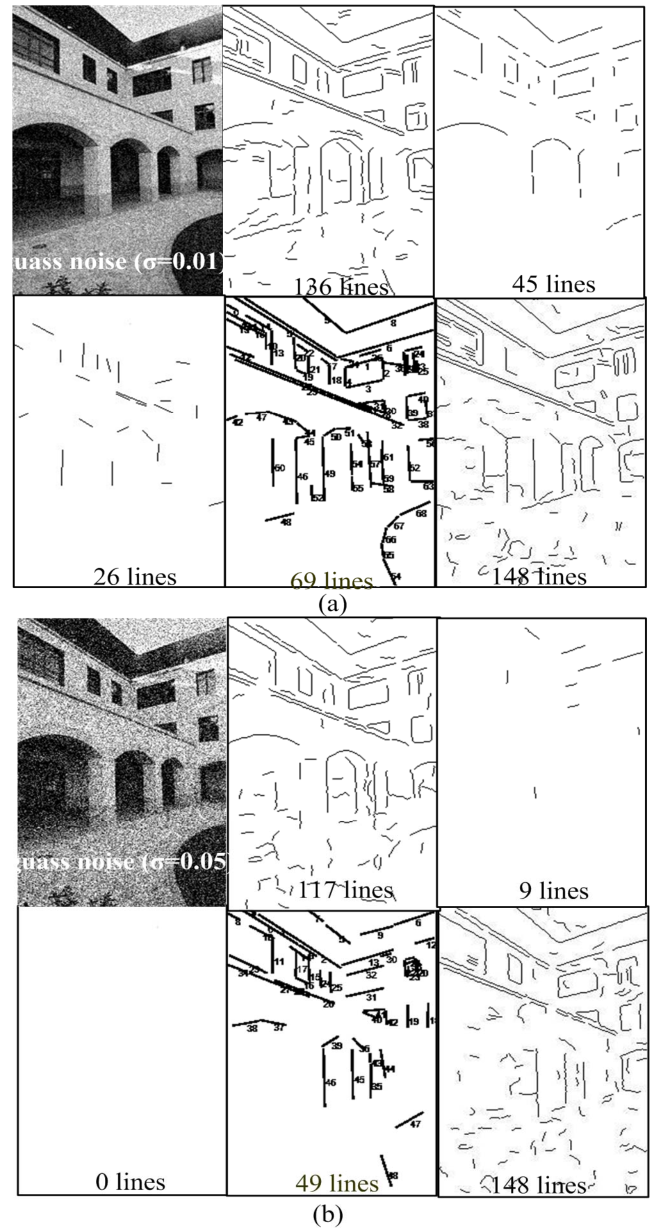

Figure 7. Analysis of a noisy natural image by five line segment detection algorithms proposed in recent years. (a) The noisy image with σ=0.1 and the the line detection results on PCA, SEA, LSD, EDLines, and our algorithm; (b) The noisy image with σ=0.05, and the line detection results on PCA, SEA, LSD, EDLines, and our algorithm

V. CONCLUSION

In this study, we presented a novel line detection algorithm that starts endpoints. Experimental results on nature images indicate that the proposed method has a satisfactory result. However, our algorithm can not avoid the interface of all smooth curves. Some arcs with large radius will still be taken as straight lines. In addition, our algorithm depends on the result of the corner detection, so if the corner is not detected exactly, the straight line connect with it will not be detected. So the future work will be done to detect corners more accurately

REFERENCES

[1] P. V. C. Hough, "Method and Means for Recognizing Complex Patterns," U.S. Patent, 1962

[2] H. Li, et al., "Fast Hough transform: A hierarchical approach," Computer Vision, Graphics, and Image Processing, 1986,.vol. 36, pp. 139-161.

[3] J. Illingworth, Kittler, J., "The adaptive Hough transform," IEEE Trans. PAMI,1987 , vol. 9, pp. 690-698..

[4] L. Xu, et al., "A new curve detection method: Randomized Hough transform (RHT)," Pattern Recognition Letters, 1990, vol. 11, pp. 331-338.

[5] L. Xu, et al., "A new curve detection method: Randomized Hough transform (RHT)," Pattern Recognition Letters, 1990, vol. 11, pp. 331-338.

[6] S. El Mejdani, et al., "Old and new straight-line detectors: Description and comparison," Pattern Recognition, 2008, vol. 41, pp. 1845-1866.

[7] J. B. H. Burns, A.R. and Riseman,E.M. , "Extracting Straight Lines," IEEE Trans. PAMI, 1986, vol. 8, pp. 425-455.

[8] R. G. J. von Gioi, J.; Morel, J.-M.; Randall, G., "LSD: A Fast Line Segment Detector with a False Detection Control " IEEE Trans. PAMI, 2010, vol.32, pp. 722 - 732.

[9] C. Akinlar and C. Topal, "EDlines: A real-time line segment detector with a false detection control," Pattern Recognition Letters, 2011, vol. 32, pp. 1633-1642.

[10] W. L. Ding, W.W.Feng, X.G.Zhang, L.C.Su, Extracting Straight Lines from Building Image Based on Edge Orientation Image. Acta Optica Sinica, 2010. vol .30: pp. 2904-2910.

[11] D. S. Guru, et al., A simple and robust line detection algorithm based on small eigenvalue analysis, Pattern Recognition Letters, 2004, vol. 25, pp. 1-13.

[12] Y.-S. Lee, et al., A straight line detection using principal component analysis, Pattern Recognition Letters, 2006, vol. 27, pp. 1744-1754.

[13] J. Canny, A Computational Approach To Edge Detection. IEEE Trans. Pattern Analysis and Machine Intelligence, 1986. vol.8: pp. 679-698.

[14] http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/index.html.

[15] X. C. He, N.H.C. Yung, Curvature scale space corner detector with adaptive threshold and dynamic region of support in Proceedings of the 17th International Conference on ICPR 2004. . 2004. pp. 791 - 794.