Straight line extraction algorithm by Hough transform combining edge grouping

WANG Jingxue[1], ZHU Qing[2], WANG Weixi[3], ZHAO Like[1]

1. *School of Geometrics*, *Liaoning Technical University*, *Fuxin* 123000, *China*;
2. *Faculty of Geosciences and Environmental Engineering*, *Southwest Jiaotong University*, *Chengdu* 610031, *China*;
3. *Shenzhen Innovation Research Centre of Land and Resources*, *Shenzhen* 518034, *China*

Abstract: This paper analyzes and discusses the main problems of line detection and extraction by traditional Hough transform in detail. Thus, it proposes an algorithm of straight line extraction by Hough transform combining edge grouping. This algorithm first adopts an edge tracking based on eight-neighborhood to group the detected edge points by Canny operator. It then separately p erforms the Hough transform to each edge group obtained by grouping, and individually determines the origin of the Hough transform and the range of parameter. This algorithm uses the iterative vote scheme to determine the single peak and the corresponding points to be deleted. The experimental results prove that the proposed algorithm is simple in principle and can effectively solve problems in the traditional Hough transform, such as low precision and complex computation. The proposed algorithm has robustness, can process different content images, and is suitable for parallel processing.

Key words: straight line extraction, Hough transform, edge grouping, edge tracking

CLC number: P237 **Document code:** A

Citation format: **Wang J X, Zhu Q, Wang W X and Zhao L K. 2014. Straight line extraction algorithm by Hough transform combining edge grouping. Journal of R emote Sensing, 1 8(2): 378 – 389 [DOI: 10.11834/jrs.20143127]**

1 INTRODUCTION

Hough transform was proposed by Hough (1962) and is widely applied for straight line extraction and edge detection. This algorithm has many advantages, such as ease of geometrical analysis, insensitive to noise, robust, and able to deal with local occlusion and cover. However, traditional Hough transform also has many problems when it is directly used for extracting lines from image. These problems include discretization parameters that can lead to inaccurate results, the distribution effect can easily cause false line, an accumulator peak in the parameter a-pace is difficult to detect, endpoints and length of a line segment are easily lost in the process of the transform, the c omputation complexity and the space complexity are relatively high, and its speed is low, especially for large images. The related improvements of the algorithm include the following: randomized Hough transform (Xu, et al., 1990), adaptive fuzzy Hough transform (Tang, et al., 2004), the method of double thresholds (Han, et al., 2004; Zuo, et al., 2012), cluster grouping method (Yuan, et al., 2005; Zhu, et al., 2009; Duan, et al., 2010), gradient fractions look-up table method (Teng, et al., 2008), p re-estimated parameter (Bonic, et al., 2005), weight matrix (Gao, et al., 2000), and multi-scale Hough transform (Atiquzzaman, 1992; Xu, 2007; Xu, et al., 2008; Fu, et al., 2012). Even though the current improved algorithms have acquired certain a chievements, the algorithm is still complex. First, this paper a nalyzes the problem of straight line extraction within the traditional Hough transform in detail, which can be divided into c ertainty and uncertainty problems. Second, this paper improves the Hough transform for straight line extraction by combining edge grouping to solve the problems.

2 DESCRIPTION OF PROBLEMS

2.1 Certainty problem

(1) Probability Event. Hough transform is a straight line d etection algorithm in a strict mathematical sense. However, the correction of detected lines from an image by Hough transform belongs to a probability event because mapping a point from the image space to the parameter space is one-to-many. If the number of the quantization level θ in parameter space is M, then mapping a point from the image space to the parameter space will separately fall on the M accumulators. This scenario means that image points corresponding to the accumulator (cell) peak will also fall on other accumulators, and these points join with other discrete points (edge points or noise points) and vote to generate

Received: 2013-05-24; **Accepted:** 2013-10-15; **Version of record first published:** 2013-10-22

Foundation: National Basic Research Program (973 Program) (No. 2010CB731801); National Natural Science Foundation of China (No. 41101452, 41201454); Open Research Fund Program of the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing of Wuhan University (No. 11I02); Research Fund for the Doctoral of Higher Education of China (No. 20112121120003)

First author biography: WANG Jingxue (1981—), female, Lecturer. Her research interest is image matching, as well as three-dimensional reconstruction. E-mail: x iaoxue1861@163.com

another pseudo-peak. The corresponding line to this peak is pseudo-line. As shown in Fig. 1(a), peak cell 2 is pseudo-peak that is composed of a subset of the edge points on the c orresponding line to peak cell 1 and other edge points. This s ituation usually happens in the vicinity of the local peak cell. When the cell position is closer to local peak cell, both cell values are also closer and vice versa. As a result, the cell value of the local peak is difficult to distinguish from its neighbor's cell values, making the detection of the correct peak difficult as well (Xu, 2007). When noises are observed in the image, the probability of the false peak will increase, and the increase will also lead to the increase in the probability of detecting pseudo-lines.

(2) Over zero-clearing. Most of the existing methods use the zero-clearing strategy to avoid generating the above false peak in the neighborhood range of the local peak cells. First, it recodes the local peak cell coordinate in parameter space, and then sets the value of the peak cell and its certain neighborhood cells to zero. The zero-cleaning process can decrease the false peak p henomena to a certain extent, but it also generates a new problem. If two adjacent lines in the image space exist, the peak position of the parameter space corresponding to the two lines is also close. The zero-cleaning process can clear a peak cell that corresponds to one of the two straight lines. As a result, over zero-clearing will undoubtedly miss straight lines in the image space.

(3) The absolute horizontal and vertical lines. In Hough transform, the value of θ can be taken in $[0, \pi]$ or $[-\pi/2, \pi/2]$. For the $[0, \pi]$ case, the absolute vertical lines will generate a double peak value, symmetrically locate the position of 0 and π, and the value of ρ is the opposite. For $[-\pi/2, \pi/2]$, the absolute horizontal lines will generate a double peak value, s ymmetrically locate the position of $-\pi/2$ and $\pi/2$, and the value of ρ is the opposite.

(4) Determine the endpoints and length of a straight line. Hough transform detects straight lines in the image space, but does not determine their endpoint and lengths.

2.2 Uncertainty problem

(1) The quantization interval in the parameter space. The quantization interval of a parameter space directly determines the accuracy and computational speed of the Hough transform. When the quantization interval is large, the aggregation effect in the parameter space is poor, line positioning is not accurate, and a certain number of lines cannot be detected. A small quantization interval will increase computation complexity and storage space, and the detected lines will appear as a staggered phenomenon as shown in Fig. 1(b), which is more serious for longer oblique lines.

(2) Over-connecting. Traditional Hough transform uses binary edge image for global transform. One disadvantage of this algorithm lies in its heavy calculating burden and low computing speed. It will generate over-connecting lines and pseudo-lines, and the threshold of peak detection is difficult to control. Over-connecting lines caused by traditional Hough transform includes three cases: (Case 1) edge points from the edges of different o bjects or different lines of the same object are identified as a straight line, as shown in Fig. 1(c); (Case 2) other points, such as noise points and/or points from other lines, are considered as edge points for a straight line, as shown in Fig. 1(d); (Case 3) non-edge points are detected as a straight line, as shown in F ig. 1(e). Aiming at the over-connecting problem, existing solutions are mostly based on sub-regional Hough transform. First, this method divides an image into non-overlapping sub-images and then extracts straight lines by Hough transform from the s ub-images. This method cannot completely eliminate the over-connecting phenomenon and also makes detecting the lines located on the boundary of two blocks or across two blocks d ifficult. Xu, et al. (2007) kept the 10% overlap between adjacent image blocks divided to solve the problem, but it increases the amount of calculation for line connecting in sequence. O ver-connecting and pseudo-lines can usually be avoided by s etting the minimum distance of the line connection and the m inimum length of the straight line by double thresholds. For the global image Hough transform or rule sub-regional Hough transform, the double thresholds scheme still cannot solve this problem completely.

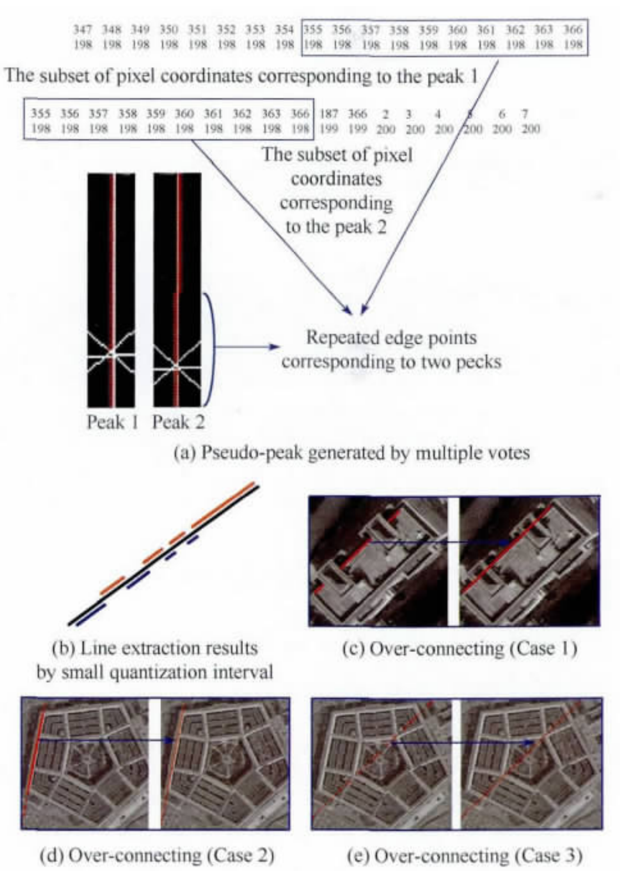


Fig. 1 Problems of straight line extraction by Hough transform

3 HOUGH TRANSFORM STRAIGHT LINE E XTRACTION ALGORITHM COMBINING EDGE GROUPING

For the certainty and uncertainty problem of detected straight lines by Hough transform, this paper proposes an algorithm of straight line extraction by Hough transform combining edge grouping. The flow chart of the proposed algorithm is shown in Fig. 2. The algorithm includes three steps: to detect the edge by Canny operator (Canny, 1986), to group the edge pixels based on the eight-neighborhood tracking method, and to extract the straight lines from each edge group by improved Hough transform.

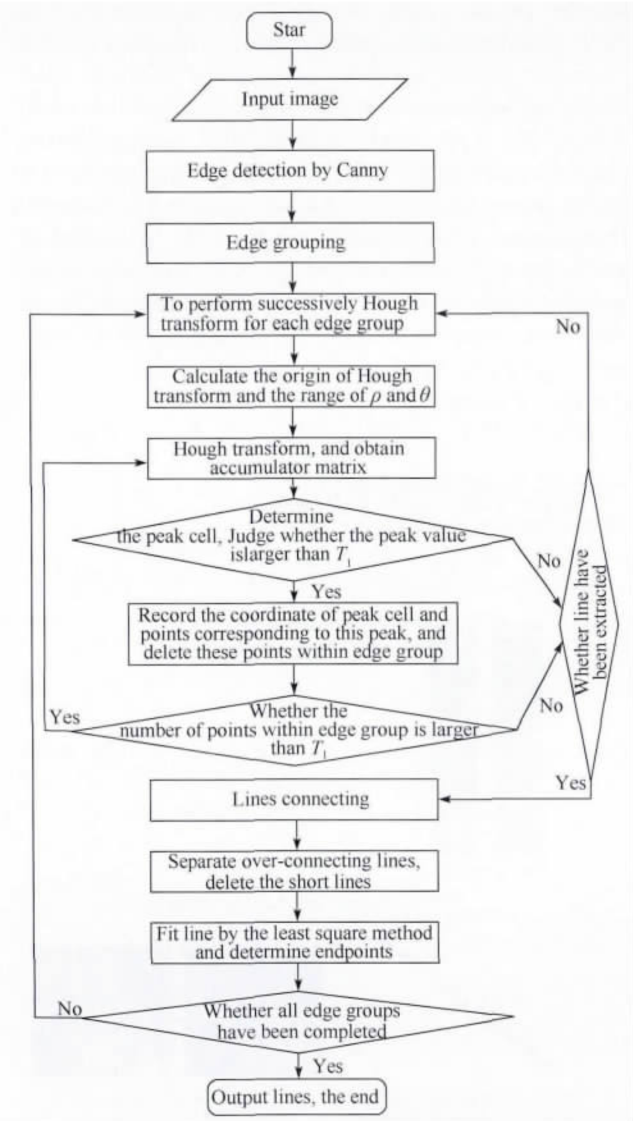


Fig. 2 Flow chart of the proposed algorithm

3.1 Edge grouping

In order to speed up the algorithm, avoid over-connections and false lines caused by global Hough transform, this paper a dopts the improved eight-neighborhood edge tracking method, which obtains several sets of independent unconnected edges, to group the detected edge pixels. Only the coherence among points in the tracking process, rather than restrict the edge direction, and the tracking priority are considered. The procedure of the method is as below.

Step 1 Search the initial pixel $(x, y)_i$. From top to bottom and from left to right, consider the first edge pixel scanned as the i nitial pixel. Establish the edge group, $\boldsymbol{EdgeGroup}_m = \{(x, y)_i\}$, where $m(m \geqslant 1)$ and i are the indexes of edges and edge-points, respectively.

Step 2 Consider the initial pixel as the current point. At the same time, set its pixel value to zero in the binary edge image and detect the presence of edge points in its eight-neighborhood. If no edge points are detected, return to Step 1 to look for the new initial pixel. If so, generate the empty temporary pixel group, ***NewPtGroup***.

Step 3 Update the edge group. Successively add the edge points in the eight-neighborhood of the current point to the edge group, $\boldsymbol{EdgeGroup}_m = \{(x, y)_i, (x, y)_{i+1}, \cdots, (x, y)_{i+n}\}$. Set their pixel values in the binary edge image to zero to avoid being scanned in the following step. Update the temporary pixel group, $\boldsymbol{NewPtGroup} = \{(x, y)_{i+1}, (x, y)_{i+2}, \cdots, (x, y)_{i+n}\}$, which records the newly added pixels.

Step 4 Set the last point in the temporary pixel group to the current point, search the eight-neighborhood of the current point to find new edge points, and then delete the current point in the temporary pixel group. Update the temporary pixel group, $\boldsymbol{NewPtGroup} = \{(x, y)_{i+1}, (x, y)_{i+2}, \cdots, (x, y)_{i+n-1}\}$. If edge points in the eight-neighborhood of the current point exist, return to Step 3. Otherwise, continue the above procedure until the temporary pixel group is empty.

Step 5 Repeat Step 1 to Step 4 until not finding an initial pixel after the image scanning is completed.

Finally, delete the independent pixels and the short edges, that is, the groups in which the total number of pixels is smaller than the threshold T_1. The deleted pixels do not participate in the subsequent Hough transform.

3.2 Straight line extraction by Hough transform

Hough transform will be executed on each edge group to extract straight lines.

(1) For each edge group, determine the origin of the Hough transform and the range of the ρ value. Assume that $\boldsymbol{G}_m = \{G_m(x_i, y_i) \mid i = 1, 2, \cdots, n\}$ is the *m-th* edge group that consists of connected edge points, and the number of edge points is n. In Fig. 3, the curve corresponding to the edge group is obtained by edge tracking. First, calculate the minimum, $x_{\min}$ and $y_{\min}$, for all points in the edge group, that is, $x_{\min} = \min\{x_1, x_2, \cdots, x_n\}$ and $y_{\min} = \min\{y_1, y_2, \cdots, y_n\}$. Take $(x_{\min} - 2, y_{\min} - 2)$ as the origin of the Hough transform of the edge group. The corresponding pixel coordinates are updated as $\boldsymbol{G}_m' = \{G_m'(x_i', y_i') \mid i = 1, 2, \cdots, n\}$, where $x_i' = x_i - (x_{\min} - 2)$ and $y_i' = y_i - (y_{\min} - 2)$. Second, calculate the maximum x value and the maximum y value of all points in the edge group in the new coordinate origin. These values are r ecorded as $x_{\max}' = \max\{x_1', x_2', \cdots, x_n'\}$ and $y_{\max}' = \max\{y_1', y_2', \cdots, y_n'\}$, respectively. Thus, the width and the height of the Hough transform domain are $width = x_{\max}' + 2$ and $height = y_{\max}' + 2$, r espectively. The range of ρ is $-\sqrt{width^2 + height^2} \leqslant \rho \leqslant \sqrt{width^2 + height^2}$. The range of θ is 0 to π. The quantization i nterval in the parameter space setting is set at $\Delta\rho = 1$, $\Delta\theta = \arctan(1/\max(width, height))$ (Mirmehdi, et al., 1991).

(2) To perform Hough transform for each edge group s uccessively. This is an iterative procedure. Take an edge as an example. First, transform each point in an edge group into a curve in the parameter space by polar coordinate equation, then obtain an accumulator matrix in the parameter space by voting.

Second, detect the peak on the accumulator matrix, then select the largest peak or one of the largest global peaks in case of multiple largest peaks. If the peak value is larger than T_1, record this peak coordinate in the parameter space as (r, c), where r and c correspond to the values of ρ and θ and these points in the edge group corresponding to this peak. At the same time, these points will be deleted from the edge group and not take part in the Hough transform afterwards. Judge the number of points within the edge group. If the number is more than T_1, perform the Hough transform for the rest points in the edge group until the maximum peak of the accumulator matrix or the number of the points in the edge group becomes less than T_1.

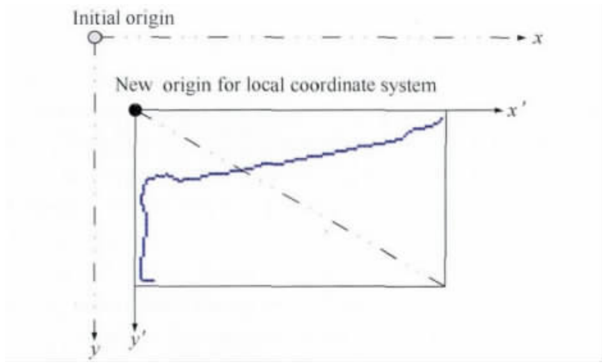


Fig. 3 Origin of the Hough transform and the range of ρ for the edge group

(3) Lines connecting. Compare the peak coordinates (r, c) from the lines extracted by the Hough transform from the single edge group. The points corresponding to two peaks, (r_i, c_i) and (r_j, c_j), which simultaneously satisfy two conditions, $|r_i - r_j| = 1$ and $c_i - c_j = 0$, that is, $|\rho(r_i) - \rho(r_j)| = \Delta\rho$ and $\theta(c_i) - \theta(c_j) = 0$ are regarded as located on the same line.

(4) Separation of over-connecting lines. In practice, a straight line detected by the Hough transform may still be two or more lines, which is the so-called over-connecting problem. The edge grouping strategy can solve the last two cases mentioned in subsection 2. 2, but its first case is still an open problem. Hence, further separating an over-connecting line into edge lines is necessary.

First, order the points in a group corresponding to a line a ccording to its slope. If the absolute value of the slope is less than 1/2, the order of the points is based on their X coordinates. On the contrary, if the absolute value of the slope is larger than or equal to 1/2, the order of the points is based on their Y coordinates. Compute the distance d between the adjacent points. If d is less than or equal to T_2, then the two adjacent points belong to the same line. Otherwise, the two points will be considered as the endpoints of two straight lines and split the group. Judge the number of points in each straight line obtained by the split. If the number is less than the threshold T_3, the straight line is deleted.

(5) Fit the line by the least square method, then determine the endpoints. Straight lines can be extracted after performing the above steps, and each of these lines corresponds to a set of points. The line parameters are obtained by fitting these points u sing the least square scheme. The first and the last points in the group are also the endpoints of the line.

4 EXPERIMENTAL RESULTS AND DISCUSSION

Testing is performed with both the synthetic and aerial digital images to verify the reliability and efficiency of the straight line extraction algorithm. Fig. 4(a) shows the synthetic image with a size of 367 × 397 pixels and uniform background. Fig. 5 (a) gives the aerial image of a pentagon with a size of 4 81 × 489 pixels and with complex object edges. First, traditional Hough transform and the proposed algorithm are e xecuted on both images, and the line detection results are c ompared. Second, the proposed algorithm is used for the aerial image (Fig. 6 (a)) and the picture (Fig. 7), each with complex background hidden rich information about the edges and the curve edges. The proposed algorithm parameters are set as $T_1 = 15$, $T_2 = 8$, and $T_3 = 5$ in the experiment.

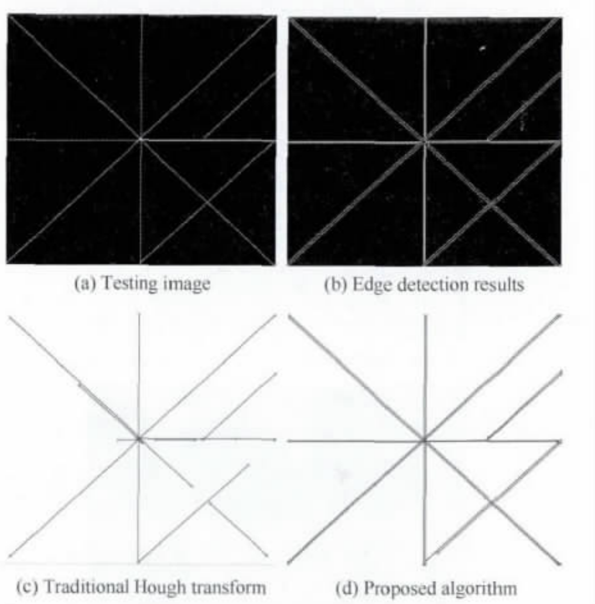
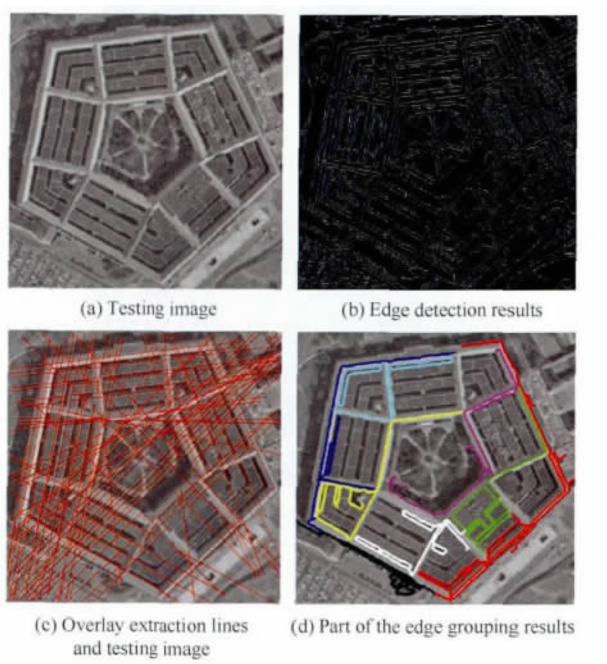
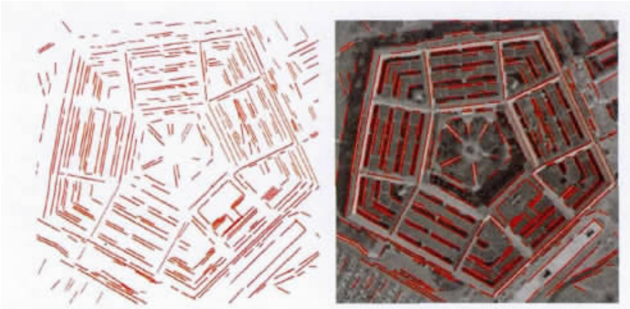


Fig. 4 Lines extracted from the synthetic image by two Hough transform algorithms

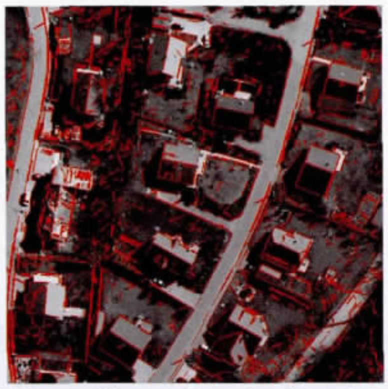




(e) Straight line extraction results by the proposed algorithm (f) Overlay extraction lines and testing image

Fig. 5 Straight line extraction results by two Hough transform algorithms

(a) Straight line extraction results utilizing the proposed algorithm



(b) Extracted lines superimposed on the test image

Fig. 6 Straight line extraction results from the digital aerial image utilizing the proposed algorithm

Fig. 7 Line extraction results by the proposed algorithm

Experiment 1 First, Canny operator is used for the edge e xtraction from the image in Fig. 4(a), and the result is shown in Fig. 4(b). Second, use the traditional Hough transform algorithm and the proposed algorithm for straight line extraction from the binary edge image. Fig. 4(c) and Fig. 4(d) show the results of the operations. Traditional Hough transform extracts 11 lines with θ in $[-\pi/2, \pi/2]$, in which two horizontal lines completely overlapping with each other exist. Given the repeated voting, a short straight line is e ntirely within the line. Considering that the accumulator near a peak is zero-cleared, only one of the edges in the double-edge lines is detected. Edge grouping of the above binary edge image results in one edge group, and 12 lines are extracted on the edge group by the Hough transform proposed in this paper. The e xtracted lines match the edge points completely and have better connectivity. The running time is almost the same as that of the traditional Hough transform.

Experiment 2 First, extract edges from Fig. 5(a) by Canny operator. The resulting binary edge image is shown in Fig. 5(b). Traditional Hough transform is used for the line extraction in Fig. 5(b), and the results are shown in Fig. 5(c). In Fig. 5(c), 168 straight lines were extracted, and the running time was 85 s. G iven the simple background and fewer lines in Experiment 1, most of the lines extracted by the traditional Hough transform are correct. However, the problems mentioned in Section 2 (e. g., the over-connecting problem) exist in the global Hough transform on the complex background image in this experiment. The proposed algorithm first adopts the eight-neighborhood edge tracking method to group the detected edge pixels. Approximately 286 edge groups were obtained. The eight edge groups, which include the largest number of edge points, are i llustrated in d ifferent colors in Fig. 5(d). Then, extract straight lines for each edge group using the improved Hough transform algorithm. The extracted straight lines are shown in Fig. 5(e) and overlaid on the testing image (Fig. 5(f)). The results show that the proposed algorithm can extract straight lines efficiently, even for images with complex backgrounds.

Experiment 3 The digital aerial image with a size of 1800 × 1 800 from the ISPRS website was used to validate the correctness of the proposed algorithm. The number of edge groups obtained by edge tracking is 4568, and the lines extracted from these groups are 2632 (Fig. 6(a)). For visual evaluation, the extracted lines are superimposed on the test image (Fig. 6(b)). As a result, the outlines of the objects such as building roofs and road edges can be detected, are clear and continuous, and with only small details lost. In the vegetation area, some edges corresponding to non-manmade objects are detected as lines. Although they are not considered as useful edges in future p rocessing, the extracted results are still correct.

Experiment 4 The proposed algorithm is applied to the i mage with curved edges. The extracted straight lines are overlaid on the testing image, as shown in Fig. 7. This algorithm can smoothly fit the curve edges utilizing the extracted lines.

Table 1 shows the number of extracted lines, running time, number of false lines, and extraction accuracy by the proposed algorithm in the four experiments. Experimental results show that the proposed algorithm can effectively detect straight lines and curve edges in the images. The algorithm is also suitable for d ifferent types of image data and has linear computation c omplexity depending on the size and content of image.

Table 1 Accuracy and running time of straight line extraction

	Number of extracted lines	Time /s	Number of false lines	Extraction accuracy/%
Experiment 1	12	0.81	0	100
Experiment 2	476	42	5	98.9
Experiment 3	2632	107	39	98.5
Experiment 4	1216	73	9	99.2

5 CONCLUSIONS

This paper aims to solve the problems in the traditional Hough transform when it is used for straight line extraction. It proposes an improved straight line extraction algorithm combining edge grouping and the Hough transform. This algorithm has the following advantages.

(1) The edge grouping ensures the continuity of extracted straight lines, solves the over-connecting problems, and improves the efficiency of edge extraction.

(2) By deleting the short edge groups, the effects of independent pixels or short edge groups on the straight line extraction are not only eliminated, but the running time and storage space can also be decreased.

(3) Individually determines the origin of the local Hough transform and the range of ρ for each edge group, which greatly decreases computation complexity and enhances the reliability of line extraction.

(4) Considering the use of the iterative vote scheme to d etermine the single peak and the corresponding points to be d eleted in process of the Hough transform, the proposed algorithm can effectively avoid false peaks and over zero-clearing caused by multiple voting.

Through theoretical analysis and experimental verification, the results demonstrate that the proposed algorithm is simple in principle, insensitive to noise, and reliable for straight line e xtraction. The proposed algorithm can build a solid foundation for processes such as image matching, camera calibration, and three-dimensional reconstruction.

REFERENCES

Atiquzzaman M. 1992. Multiresolution Hough transform-an efficient method of detecting patterns in images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(11): 1090 - 1095 [DOI: 10.1109/34.166623]

Bonci A, Leo T and Longhi S. 2005. A bayesian approach to the Hough transform for line detection. IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, 35(6): 945 - 955 [DOI: 10.1109/TSMCA.2005.853481]

Canny J. 1986. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(6): 679 - 698 [DOI: 10.1109/TPAMI.1986.4767851]

Duan R J, Zhao W, Huang S L and Chen J Y. 2010. Fast line detection algorithm based on improved Hough transformation. Chinese Journal of Scientific Instrument, 31(12): 2774 - 2780

Fu X Y, You H J, Fu K and Zhang H M. 2012. An approach to extract L-shaped feature of building from VHR SAR image using multi-scale Hough transform. Geomatics and Information Science of Wuhan University, 37(4): 394 - 398

Gao J and Li C. 2000. Pre-storage weight matrix quick Hough transform. Computer Engineering, 26(6): 1 - 3

Han Q L, Zhu M and Yao Z J. 2004. Line segment feature extraction based on improved Hough transform. Chinese Journal of Scientific Instrument, 25(4S): 436 - 439

Hough P V C. 1962. Method and means for recognizing complex patterns. USA, United States Patent, 3069654

Mirmehdi M, West G A W and Dowling G. R. 1991. Label inspection using the Hough transform on transputer networks. Microprocessors and Microsystems, 15(3): 167 - 173 [DOI: 10.1016/0141-9331(91)90137-5]

Tang L, Xie W X, Huang J J and Huang J X. 2004. Adaptive fuzzy Hough transform. Acta Electronica Sinica, 32(6): 946 - 949

Teng J Z and Qiu J. 2008. Fast and precise detection of straight line with Hough transform. Journal of Image and Graphics, 13(2): 394 - 398

Xu L, Oja E and Kultanen P. 1990. A new curve detection method: randomized Hough transform (RHT). Pattern Recognition Letters, 11(5): 331 - 338 [DOI: 10.1016/0167-8655(90)90042-Z]

Xu S H. 2007. Straight line extraction method for stereoimage feature matching. Wuhan: Wuhan University, 40 - 41

Xu S H, Zhu Q, Liu J P, Han L T, Zhao X L and Zhang L H. 2008. Straight line extraction via multi-scale Hough transform based on pre-storage weight matrix. Acta Geodaetica et Cartographica Sinica, 37(1): 83 - 88

Yuan G L, Xue M G, Li C L and Han Y S. 2005. Fast Hough transform line detection based on image block according to phase group. Computer Measurement and Control, 13(7): 647 - 649

Zhu F F, Gu H B and Sun J. 2009. A line detection algorithm based on improved Hough transformation. Computer Technology and Development, 19(5): 19 - 22

Zuo L, Li M, Zhang X W and Luo X Y. 2012. Small-target detection in sea clutter based on improved Hough transform. Journal of Electronics and Information Technology, 34(4): 923 - 928

结合边缘编组的 Hough 变换直线提取

王竞雪[1]，朱庆[2]，王伟玺[3]，赵丽科[1]

1. 辽宁工程技术大学 测绘与地理科学学院，辽宁 阜新 123000;
2. 西南交通大学 地球科学与环境工程学院，四川 成都 610031;
3. 深圳市国土资源创新研究中心，广东 深圳 518034

摘 要: 针对传统 Hough 变换用于直线检测存在的问题进行了细致的分析和归纳总结,在此基础上,提出一种结合边缘编组的 Hough 变换直线提取算法。该算法首先采用基于 8 邻域的边缘跟踪算法对 Canny 算子检测得到的边缘点进行编组; 然后对每一个边缘组分别进行 Hough 变换,单独确定 Hough 变换原点和参数的取值范围。Hough 变换过程中,采用迭代的“投票”方式,每次确定单一峰值点并删除对应像素。实验证明,该算法原理简单,能有效解决传统 Hough 变换存在的精度不高、计算复杂等问题。同时该算法具有较强的鲁棒性,可以有效处理不同类型的影像数据,适用于并行处理。

关键词: 直线提取,Hough 变换,边缘编组,边缘跟踪

中图分类号: P237 文献标志码: A

引用格式: 王竞雪,朱庆,王伟玺,赵丽科. 2014. 结合边缘编组的 Hough 变换直线提取. 遥感学报,18(2): 378 –389

Wang J X, Zhu Q, Wang W X and Zhao L K. 2014. Straight line extraction algorithm by Hough transform combining edge grouping. Journal of Remote Sensing, 18(2): 378 –389 [DOI: 10.11834/jrs.20143127]

1 引 言

自 Hough 变换提出以来(Hough,1962),被广泛应用于直线提取和边缘检测。该算法具有几何解析性简单,受噪声影响小,鲁棒性好,能够较好处理局部遮挡、覆盖等优点。但传统 Hough 变换直接用于图像上直线提取存在许多问题,如参数离散化导致结果不精确,分布效应容易产生虚假直线,参数空间峰值难以检测,此外变换过程中丢失了线段的端点和长度信息,计算复杂度和空间复杂度比较高,运算速度慢,尤其对于较大影像。相关改进包括: 随机 Hough 变换(Xu 等,1990)、自适应模糊 Hough 变换(唐亮 等,2004)、双阈值法(韩秋蕾 等,2004; 左磊 等,2012)、聚类编组法(袁广林 等,2005; 朱芳芳 等,2009; 段汝娇 等,2010)、斜率分式查表法(滕今朝和邱杰,2008)、预先估算(Bonci 等,2005)、权值矩阵法(高隽和李成,2000) 和多尺度 Hough 变换(Atiquzzaman,1992; 徐胜华,2007; 徐胜华 等,2008; 傅兴玉 等,2012) 等。现有改进算法取得了一定的研究成果,但算法较为复杂。本文首先对传统 Hough 变换用于直线提取存在的问题进行详细分析,将其归结为确定性问题和不确定性问题两个方面。然后从解决问题出发,提出一种结合边缘编组的 Hough 变换直线提取算法。

2 问题描述

2.1 确定性问题

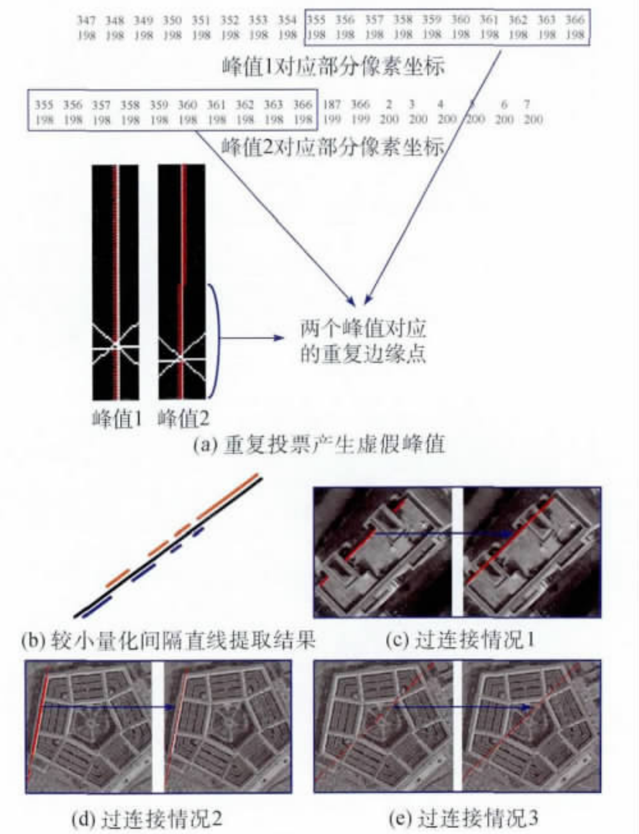
(1) 概率事件: Hough 变换是严格数学意义上的直线检测算法,但是该算法用于图像上直线检测,结果的正确性属于一种概率事件。因为图像空间的点映射到参数空间是“一对多”的映射模式。如果参数空间 θ 的量化值共 M 个,那么图像空间中任意点映射到参数空间分别落到 M 个累加器上。也就是说累加器峰值对应的图像上的点也会落到其他累加器上,这些点可能会同其他离散的点(边

收稿日期: 2013-05-24; 修订日期: 2013-10-15; 优先数字出版日期: 2013-10-22

基金项目: 国家重点基础研究发展计划(973 计划)(编号: 2010CB731801); 国家自然科学基金项目(编号: 41101452,41201454); 武汉大学测绘遥感信息工程国家重点实验室开放基金资助(编号: 11I02); 高等学校博士学科点专项科研基金资助课题(编号: 20112121120003)

第一作者简介: 王竞雪(1981—),女,讲师,主要研究方向是影像匹配、3 维重建理论与方法研究。E-mail: xiaoxue1861@163.com

缘上的点或者噪声点) 共同投票产生另一个虚假峰值,而该峰值对应的是一条伪直线。如图 1(a) 所示,峰值 2 是由峰值 1 对应直线上的部分边缘点和其他边缘点组成的伪峰值。这种情况通常发生在局部峰值点附近。当累加器距离局部峰值较近时,累加器的值就越接近局部峰值; 当累加器距离局部峰值较远时,累加器的值就越远离局部峰值。从而造成真实的局部峰值和其周围的累加器的值不易区分,难以检测出正确的峰值(徐胜华,2007) 。除此之外,当图像上存在噪声时,产生虚假峰值的概率就会增加,从而检测出伪直线的概率也会随之增加。



(a) 重复投票产生虚假峰值

(b) 较小量化间隔直线提取结果 (c) 过连接情况1

(d) 过连接情况2 (e) 过连接情况3

图 1 Hough 变换用于直线提取存在问题

(2) 过清零: 为了避免上述局部峰值点邻域范围内产生的虚假峰值,现有方法多采用清零法。首先记录局部峰值点在参数空间坐标,然后将该峰值点及其一定邻域范围内的累加器的值设置为零。这种邻域范围内清零方法能在一定程度上削弱了上述虚假峰值现象,同时也产生了一个新的问题。即当图像空间中有两条距离较近的直线时,这两条直线对应的参数空间的峰值点的位置是邻近的,而上述清零法会将其中一条直线对应的峰值点也同时清零。这样,无疑会使得图像空间中本该检测出来的直线由于过清零而被漏检。

(3) 绝对水平线和绝对垂直线: Hough 变换过程中,θ 的取值通常有 $[0,\pi]$ 和 $[-\pi/2,\pi/2]$ 两种情况。如果选择 $[0,\pi]$,绝对垂直线会出现双峰值,对称位于 0 和 π 位置,ρ 值相反; 如果选择 $[-\pi/2,\pi/2]$,绝对水平线会出现双峰值,对称位于 $-\pi/2$ 和 $\pi/2$ 位置,ρ 值相反。

(4) 直线端点及长度的确定: Hough 只检测出图像空间中存在的直线,没有对检测直线的端点和长度进行确定。

2.2 不确定性问题

(1) 参数空间量化间隔: 参数空间量化间隔直接决定着 Hough 变换的精度以及计算的速度。当量化间隔较大时,则参数空间的聚集效果差,直线定位不精确,同时会存在部分直线漏检; 反之,当量化间隔较小时,会增加计算量和存储空间,直线检测结果会出现如图 1(b) 所示交错现象,对于较长的斜线更加严重。

(2) 过连接: 传统 Hough 变换把整个边缘检测后的二值图像用于全局 Hough 变换。这种方式一方面是计算量大,运算速度慢。另一方面会产生过连接直线和伪直线,峰值检测的阈值难以把握。Hough 变换中过连接直线包括 3 种情况: 一是,将属于不同物体边缘或同一物体不同部分的直线被检测为同一直线,如图 1(c) 所示; 二是,一些边缘和其他点(噪声点或者其他边缘上的点) 被检测为同一直线,如图 1(d) 所示; 三是,一些非边缘点被检测为同一直线,如图 1(e) 所示。针对过连接问题,现有解决方法主要有分区域 Hough 变换。该方法首先将图像划分为互不重叠的规则的子影像,然后在每一个子影像上分别进行 Hough 变换提取直线。该方法不能完全消除过连接现象,同时使得刚好处于两块之间或跨越两块的直线段难以被检测。鉴于这一情况,(徐胜华等,2008) 将划分的相邻的图像块之间保留 10% 的重叠度,但该方式增加了后面直线合并的计算量。针对过连接和伪直线问题,通常还会采用设定直线连接的最小距离,和检测直线的最小长度的双阈值方法来避免。但对于整幅图像 Hough 变换或者规则的分区域 Hough 变换,双阈值法仍不能很好地解决这一问题。

3 结合边缘编组的 Hough 变换直线提取算法

针对 Hough 变换用于直线检测存在的确定性

问题和不确定性问题,本文提出一种结合边缘编组的 Hough 变换直线提取算法,算法总体流程如图 2 所示。该算法包括 3 个步骤,首先利用 Canny 算子对图像进行边缘检测(Canny,1986);然后利用基于 8 邻域的边缘跟踪方法对检测后的边缘点进行编组;最后利用改进的 Hough 变换方法对每个边缘组分别进行直线提取。

3.1 边缘编组

为了避免全局 Hough 变换运算速度慢,容易产生过连接和虚假直线,首先采用基于 8 邻域的边缘跟踪方法对检测后的边缘像素进行编组,得到若干互不相连的独立边缘组。跟踪过程中只考虑点与点之间的连贯性,并不对边缘的方向加以约束和限制,也不考虑跟踪的优先级。具体步骤如下:

(1) 扫描初始像素点 $(x,y)_i$。按照从上到下,从左到右的顺序,将扫描到的第一个边缘像素点作为初始像素点。建立边缘组 $\boldsymbol{EdgeGroup_m} = \{(x,y)_i\}$,$m$ 表示第 $m(m \geqslant 1)$ 条边缘,i 表示边缘组内第 i 个点。

(2) 将初始像素点作为当前点,同时将二值图像中该像素点的值设为 0,分别检测它的 8 邻域内是否存在边缘点。如果不存在,返回步骤(1),寻找下一组的初始像素点。如果存在,建立一个空的临时像素组 *NewPtGroup*。

(3) 更新边缘组,将当前点 8 邻域内存在的像素点依次加到边缘组 $\boldsymbol{EdgeGroup_m} = \{(x,y)_i, (x,y)_{i+1}, \cdots, (x,y)_{i+n}\}$,并将二值图像中这些边缘点的像素值设为 0,后续不再被扫描。更新临时像素组 $\boldsymbol{NewPtGroup} = \{(x,y)_{i+1}, (x,y)_{i+2}, \cdots, (x,y)_{i+n}\}$,记录新增加到边缘组内的点。

(4) 将临时像素组内最后一个点作为当前点继续搜索,并将该点从临时像素组内删除,临时像素组更新为 $\boldsymbol{NewPtGroup} = \{(x,y)_{i+1}, (x,y)_{i+2}, \cdots, (x,y)_{i+n-1}\}$。如果当前点 8 邻域内存在边缘点,转到步骤(3);如果不存在,继续上述过程,直到临时像素组为空。

(5) 重复步骤(1)—步骤(4),直到图像扫描完毕没有发现初始像素点。

最后,去除独立像素点和短边缘,即将边缘组内总像素数目小于阈值 T_1 的边缘组去除,不参与后续的 Hough 变换。

3.2 Hough 变换提取直线

对跟踪后得到的每一边缘组分别进行 Hough 变换,提取直线。

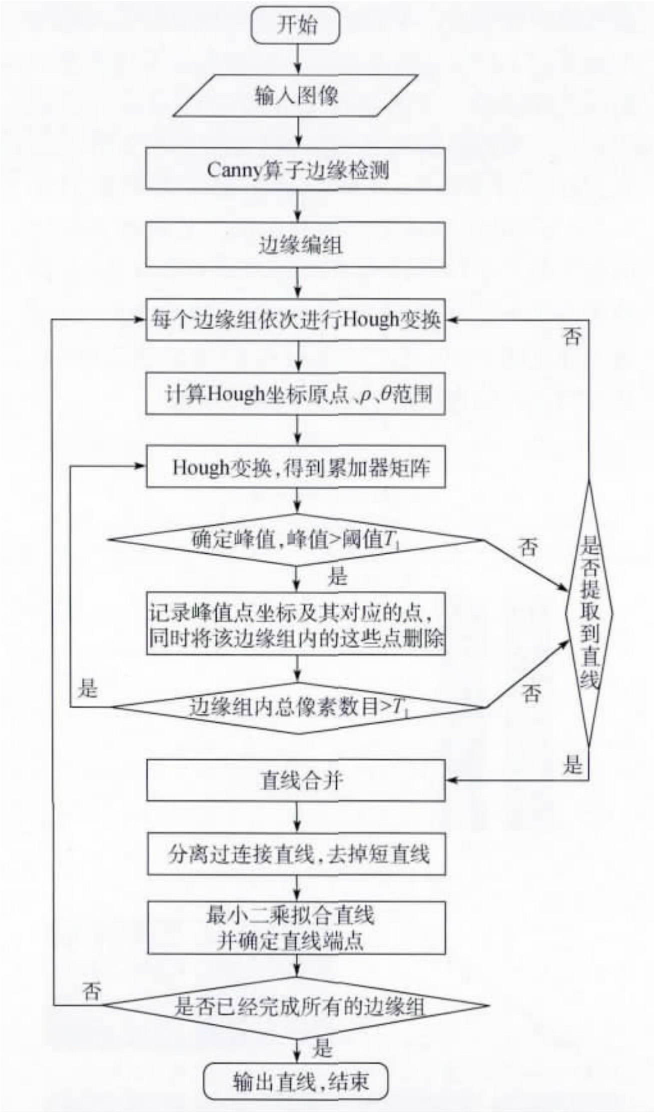


图 2 本文算法流程图

(1) 针对每一条边缘,分别确定 Hough 变换的原点和 ρ 的取值范围。假定 $\boldsymbol{G}_m = \{G_m(x_i, y_i) | i = 1, 2, \cdots, n\}$ 为第 m 条边缘,共包括 n 个相互连接的边缘点。图 3 中曲线为跟踪得到的一条边缘。首先计算该边缘组内所有点的最小的 x 值和 y 值,分别记为 $x_{\min} = \min\{x_1, x_2, \cdots, x_n\}$ 和 $y_{\min} = \min\{y_1, y_2, \cdots, y_n\}$,并以 $(x_{\min} - 2, y_{\min} - 2)$ 作为该边缘组进行 Hough 变换新的原点,对应的组坐标更新为 $\boldsymbol{G}'_m = \{G'_m(x'_i, y'_i) | i = 1, 2, \cdots, n\}$,其中 $x'_i = x_i - (x_{\min} - 2)$,$y'_i = y_i - (y_{\min} - 2)$。再分别计算在新的坐标原点下所有点的最大的 x 值和 y 值,分别记为 $x'_{\max} = \max\{x'_1, x'_2, \cdots, x'_n\}$ 和 $y'_{\max} = \max\{y'_1, y'_2, \cdots, y'_n\}$。则 Hough 变换的域宽和域高为别取 $width = x'_{\max} + 2$,$height = y'_{\max} + 2$。ρ 的取值范围为 $-\sqrt{width^2 + height^2} \leqslant \rho \leqslant \sqrt{width^2 + height^2}$。$\theta$ 的取值为 $[0, \pi]$。参数空间量

化间隔的设定为 $\Delta\rho = 1$,$\Delta\theta = \arctan(1/\max(width, height))$(*Mirmehdi* 等,1991)。

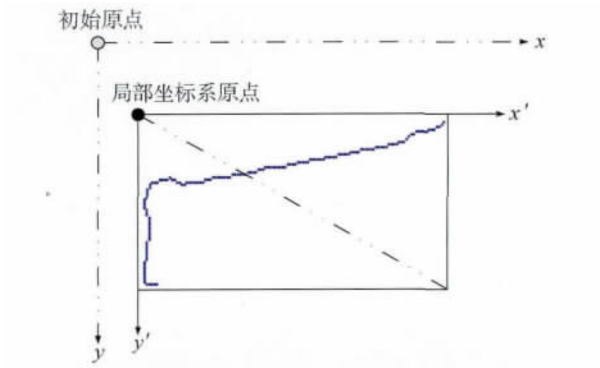


图 3　边缘组 Hough 变换的原点及 ρ 的取值范围

(2) 依次对每一边缘组进行 Hough 变换。这是一个迭代计算的过程。下面以一个边缘为例。首先将边缘组内每一个像素点分别按极坐标方程变换到参数空间,通过投票得到参数空间的累加器数组。然后对累加器矩阵进行峰值检测,仅选择一个最大的全局峰值。如果同时存在多个相同的最大峰值,同样只选择其中一个峰值。若该值大于 T_1,记录该峰值在参数空间坐标(r,c)(r、c 分别对应 ρ、θ 的值) 以及该峰值对应的边缘组内的像素点。同时将这些点从该边缘组内删除,不再参与后续的 Hough 变换。判断边缘组内总像素的数目是否大于 T_1。如果大于再对边缘组内剩下的点进行 Hough 变换,直到累加器数组的最大峰值小于 T_1 或者边缘组内总像素的数目 $< T_1$ 为止。

(3) 直线合并: 对单一边缘经过 Hough 变换后得到的直线进行比较,比较各直线组的峰值坐标(r,c)。经过实验分析,同时满足 $|r_i - r_j| = 1$、$c_i - c_j = 0$ 这两个条件的两个峰值(r_i,c_i)、(r_j,c_j) 对应的像素点为同一条直线上的点,将两个直线合并。上面的条件也就是满足 $|\rho(r_i) - \rho(r_j)| = \Delta\rho$、$\theta(c_i) - \theta(c_j) = 0$ 这两个条件。

(4) 过连接直线分离: 对边缘组进行 Hough 变换后,每个峰值对应的一组像素点理论上位于同一条直线段上,但是针对图像上物体边缘而言,它仍然可能存在过连接问题。边缘分组基本解决了 2.2 节所述的过连接问题的后两种情况,但过连接问题的第 1 种情况仍然可能存在。因此需要进一步分离过连接直线。

首先根据直线斜率确定该组像素点的排序方式。如果斜率绝对值小于 1/2,按 x 值大小对该组像素点进行从小到大的排序,反之,如果斜率绝对值大于等于 1/2,按 y 值大小排序。依次计算组中相邻点距离 d,如果满足 d 小于等于阈值 T_2,则说明此相邻两点属于同一线段; 否则,认为这两个点分别为两条直线段的端点,并在该位置处对组点进行分裂。判断分裂后每一条直线段包含总的像素数目,如果小于阈值 T_3,则删除该直线。

(5) 最小二乘直线拟合及端点确定: 经过上述步骤处理后得到直线提取的结果,每条直线对应一组像素点。对这些点进行最小二乘拟合获取直线参数。同时该组像素点的第一个像素点和最后一个像素点即为该直线的端点。

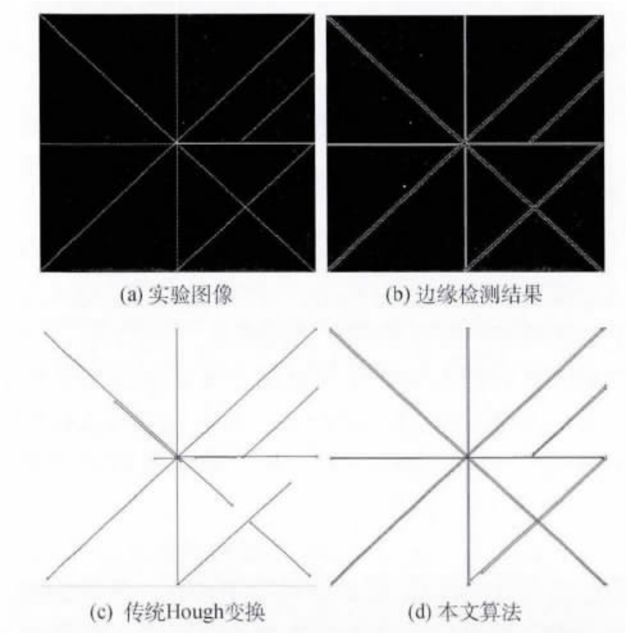
4　实验结果与分析

为了验证本文算法提取直线的可靠性和有效性,先后对模拟图像和航空数字图像进行直线提取实验。图 4(a) 为背景单一的模拟图像,大小为 367 ×397。图 5(a) 为 481 ×489 大小的边缘复杂的五角大楼航拍影像。首先采用传统 Hough 变换和本文算法分别对两张影像进行直线提取,并对结果进行比较。然后将本文算法分别用于具有复杂背景的较大信息量的航空影像图 6(a) 和带有曲线边缘的影像图 7。实验过程中本文算法参数取值分别为 $T_1 = 15$,$T_2 = 8$,$T_3 = 5$。

实验 1: 首先采用 Canny 边缘检测算子对图 4(a) 影像进行边缘检测,得到结果如图 4(b)。然后分别采用传统的 Hough 变换算法和本文算法对边缘检测后的二值图像进行直线提取,得到结果分别如图4(c) 和图 4(d) 所示。传统 Hough 变换共提取到 11 条直线,在 θ 的取值 $[-\pi/2, \pi/2]$ 情况下,其中存在 2 条水平直线出现完全重合现象。同时由于重复投票,另有一条短直线完全位于该直线上。此外,由于对峰值附近累积单元采用局部清零操作,对边缘检测后的双边缘图像大部分只检测到其中一个边缘的直线。本文算法对边缘跟踪后得到的一个边缘组进行 Hough 变换,共提取到 12 条直线。直线提取结果与边缘检测的结果基本吻合,直线连贯性好。运行时间与传统 Hough 变换相当。

实验 2: 首先采用 Canny 算子对原始影像图 5(a) 进行边缘检测,得到二值边缘图像如图 5(b) 所示。采用传统 Hough 变换算法对其进行直线提取,结果如图 5(c) 所示。提取得到 168 条直线,运行 85 s。从结果可以看出,在实验 1 中,由于背景单一,影像上直线较少,传统 Hough 变换提取到的直线基本

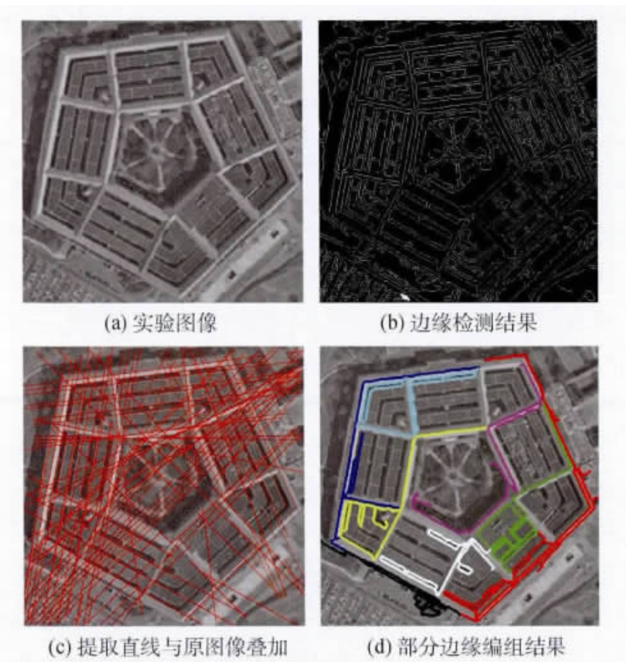
正确,但是在实验 2 中,对于背景复杂影像进行全局 Hough 变换存在第 2 节中分析的两个方面问题,其中最严重的是过连接问题。本文算法首先采用基于 8 邻域的边缘跟踪算法对边缘检测得到的边缘像素进行编组,共得到 286 组边缘组。图 5(d) 为部分边缘组的显示结果,仅显示像素数目最多的前 8 个边缘组,每一边缘组分别用不同的颜色表示。然后采用改进的 Hough 变换算法对每一边缘组进行直线提取,结果如图 5(e) 所示,图 5(f) 为提取直线与原图叠加结果。从中可以看出,对于复杂背景图像,本文算法也能得到理想的直线提取结果。



(a) 实验图像 (b) 边缘检测结果

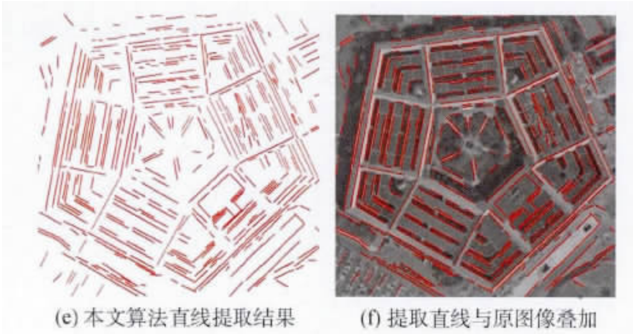
(c) 传统Hough变换 (d) 本文算法

图 4 用两种 Hough 变换算法对模拟影像提取直线



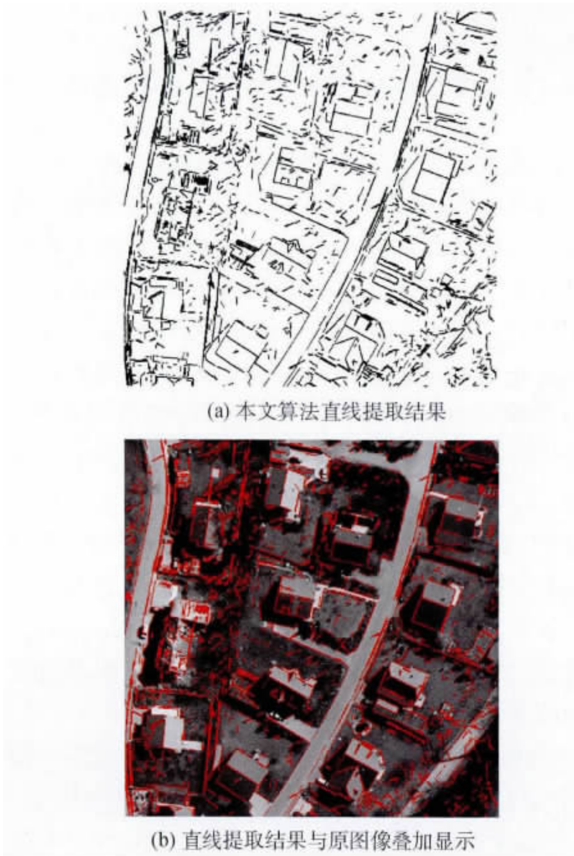
(a) 实验图像 (b) 边缘检测结果

(c) 提取直线与原图像叠加 (d) 部分边缘编组结果



(e) 本文算法直线提取结果 (f) 提取直线与原图像叠加

图 5 两种 Hough 变换算法提取直线结果



(a) 本文算法直线提取结果

(b) 直线提取结果与原图像叠加显示

图 6 本文算法对数字航空影像提取直线结果

实验 3:为了进一步验证本文算法的正确性,选自 International Society for Photogrammetry and Remote Sensing (ISPRS) 官方网站提供的数字航空影像进行实验,影像大小为 1800 × 1800。边缘跟踪得到边缘组数目为 4568,提取得到直线的数目为 2632,直线提取结果如图 6(a)。为了目视判断,将提取直线与原图像叠加显示如图 6(b)。从图 6 上可以看出:地物轮廓基本上被检测出来,如建筑物屋顶,道路边缘等,其轮廓清晰,连续,线段损失少。同时,在植被覆盖区域,一些非人工建筑物边缘线也被提取出来,尽管这些边缘在后续处理中认为是没有意义的,但提取结果是正确的。

实验 4: 将本文算法应用于具有曲线边缘的影像。直线提取结果与原图像叠加显示如图 7 所示,可以看出本文算法提取得到地直线可以较好地拟合曲线边缘。

图 7　本文算法直线提取结果

表 1 为上述 4 组实验中本文算提取直线的数目,运行时间,错误直线的数目及提取精度。实验结果表明,本文算法能较好地检测图像中的直线及曲线边缘。该算法可以有效处理不同类型的影像数据,而算法的计算量与影像大小及影像内容的复杂度呈线性关系。

表 1　直线提取精度及运行时间

	提取直线数目	时间/s	错误直线数目	提取精度/%
实验 1	12	0.81	0	100
实验 2	476	42	5	98.9
实验 3	2632	107	39	98.5
实验 4	1216	73	9	99.2

5　结　论

本文针对传统 Hough 变换用于直线提取存在的问题,提出一种结合边缘编组的 Hough 变换直线提取算法。该方法具有以下特性:

(1) 边缘编组不仅保证直线提取的连贯性,同时有效改善过连接问题,提高运行效率;

(2) 通过删除短的边缘组来消除独立像素点或者短边缘对直线提取的影响,减少了计算量和存储空间;

(3) 针对每一边缘组单独确定 Hough 变换的原点和 ρ 的取值范围,大大减小运算量,提高直线提取可靠性;

(4) Hough 变换过程中采用单峰值确定及对应像素删除的迭代“投票”方式,能有效避免重复投票产生的虚假峰值和清零操作。

通过理论分析和实验验证,结果表明本文算法原理简单,具有良好的抗噪声干扰性和可靠的直线提取结果。为基于直线段的影像匹配、相机标定、3 维重建等后续处理提供了良好基础。

参考文献(References)

Atiquzzaman M. 1992. Multiresolution Hough transform-an efficient method of detecting patterns in images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(11): 1090 – 1095 [DOI: 10.1109/34.166623]

Bonci A, Leo T and Longhi S. 2005. A bayesian approach to the Hough transform for line detection. IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, 35(6): 945 – 955 [DOI: 10.1109/TSMCA.2005.853481]

Canny J. 1986. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(6): 679 – 698 [DOI: 10.1109/TPAMI.1986.4767851]

段汝娇,赵伟,黄松岭,陈建业. 2010. 一种基于改进 Hough 变换的直线快速检测算法. 仪器仪表学报, 31(12): 2774 – 2780

傅兴玉,尤红建,付琨,张翰墨. 2012. 利用多尺度 Hough 变换提取高分辨率 SAR 图像建筑物 L 型结构. 武汉大学学报(信息科学版), 37(4): 394 – 398

高隽,李成. 2000. 预存储权值矩阵的快速 Hough 变换. 计算机工程, 26(6): 1 – 3

韩秋蕾,朱明,姚志军. 2004. 基于改进 Hough 变换的图像线段特征提取. 仪器仪表学报, 25(4S): 436 – 439

Hough P V C. 1962. Method and means for recognizing complex patterns. USA, United States Patent, 3069654

Mirmehdi M, West G A W and Dowling G. R. 1991. Label inspection using the Hough transform on transputer networks. Microprocessors and Microsystems, 15(3): 167 – 173 [DOI: 10.1016/0141-9331(91)90137-5]

唐亮,谢维信,黄建军,黄敬雄. 2004. 自适应模糊 Hough 变换. 电子学报, 32(6): 946 – 949

滕今朝,邱杰. 2008. 利用 Hough 变换实现直线的快速精确检测. 中国图象图形学报, 13(2): 394 – 398

Xu L, Oja E and Kultanen P. 1990. A new curve detection method: randomized Hough transform (RHT). Pattern Recognition Letters, 11(5): 331 – 338 [DOI: 10.1016/0167-8655(90)90042-Z]

徐胜华. 2007. 面向立体影像特征匹配的直线提取方法. 武汉: 武汉大学, 40 – 41

徐胜华,朱庆,刘纪平,韩李涛,赵雪莲,张立华. 2008. 基于预存储权值矩阵的多尺度 Hough 变换直线提取算法. 测绘学报, 37(1): 83 – 88

袁广林,薛模根,李从利,韩裕生. 2005. 基于相位编组图像分块的快速 Hough 变换直线检测. 计算机测量与控制, 13(7): 647 – 649

朱芳芳,顾宏斌,孙瑾. 2009. 一种改进的 Hough 变换直线检测算法. 计算机技术与发展, 19(5): 19 – 22

左磊,李明,张晓伟,罗小云. 2012. 基于改进 Hough 变换的海面微弱目标检测. 电子与信息学报, 34(4): 923 – 928