Straight line detection from remote sensing images by rule-based feature fusion

WANG Min* and ZHANG Qingfeng

Key Laboratory of Virtual Geographic Environment, Ministry of Education, Nanjing Normal University, 1 Wenyuan Road, Nanjing 210046, China

(Received 30 March 2011; final version received 30 June 2011)

Straight line detection is a fundamental problem in target recognition from remote sensing images since many man-made objects have straight boundaries. In this study, an integrated straight line detection method for remote sensing images is proposed. In this method, the edge-based straight lines are extracted using a chain code tracing method and the phase-based straight lines are extracted using a phase grouping method. The two types of lines are combined using a rule-based feature fusion method by removing redundant line extraction. Since this method integrates the specialties of edge- and phase-based straight line detection methods, it can detect straight lines from remote sensing images with high correctness and robustness.

Keywords: straight line detection; edge; gradient phase; remote sensing image

1. Introduction

Straight line detection is a fundamental step in target recognition using remote sensing images since many man-made objects such as airports, roads, buildings, and bridges have straight boundaries. In addition, straight lines are also the basis for many more complex remote sensing image analysis tasks including spatial scene matching, (1, 2) classification, (3, 4) and change detection (5, 6). Because of its importance, straight line detection has been widely studied within the fields of remote sensing image processing, computer vision, and pattern recognition. From the perspective of image feature utilization, state-of-the-art straight line detection methods include two main categories. The first category searches for straight lines associated with image edges, and includes the Hough transform, (7, 8) the Radon transform, (9) heuristic connecting, (10, 11) token-based extraction, (12) and PCA transformation. Since line searching is edge-based, the accuracy of the method typically depends on the effectiveness of edge detector procedure (13). As successful edge detection usually relies on a high gradient across edge pixels, this approach may fail to detect low-contrast straight lines. The second category of line detection is represented by Burn’s gradient phase grouping method (14). A set of same-phase pixels is first grouped into a so-called line support region and then a straight line representing the main regional direction is obtained using a least square fit. While this method can potentially detect low-contrast straight lines from images, the identified lines may deviate from actual line positions due to imprecise fitting. In addition, image noise tends to cause fragmentation of the lines identified.

Remote sensing images have broad spatial coverage and complex ground features. To make the line detection more robust, an integrated straight line detection method is proposed in this study. Edge-based straight lines (EBSLs) are obtained by a chain code searching method, which includes edge detection using a Canny operator, (15) edge grouping, line segmentation, and straight line connection. Phase-based straight lines (PBSLs) are obtained using Burn’s phase grouping method. A feature-level, rule-based straight line fusion is implemented to combine the two types of lines. The two methods supplement each other, creating a more reliable and comprehensive map of straight lines. Furthermore, the repetitively extracted lines are trimmed with reference to the support regions. Since our method integrates the specialties of the edge- and PBSL detection methods, it can detect straight lines from remote sensing images with high completeness and correctness.

2. Methodology

2.1. EBSL detection by chain code tracing

We first get the edge image using the Canny edge detector. Edge pixels are connected into curves by chain code tracing. We then detect the EBSLs using the Douglas–Peucker algorithm (16). An eligible straight line segment must satisfy the condition that the distance $d$ of its vertices to the line connecting its beginning and end nodes is less than a predefined threshold ($d_{\text{max}}$). In addition, it...
must be longer than a certain length threshold \((l_{\text{min}})\) to avoid line segments of trivial length.

### 2.2. PBSL detection by phase grouping

We get the PBSLs using Burn’s method. Gradient magnitude and orientation are detected using a \(2 \times 2\) local window. A dual overlapping partition technique is used to group pixels into line-support regions based on similarity of gradient orientations. Straight lines are then extracted by weighted least squares fitting from each region. In this method, we need to specify two algorithm inputs, the minimum gradient difference and minimum line length.

### 2.3. Straight line fusion

The EBSLs and PBSLs are first combined into a whole straight line layer. Since some lines are repetitively extracted by both methods, the elimination of redundant information is necessary. According to the principle of Burn’s method (1986), two straight edge lines rarely fall into a same line-support region since it would make the regional phases inconsistent and break the region. In Burn’s method, only one PBSL is created for each region. Therefore, if an EBSL falls into a line-support region, it denotes the actual position of its PBSL. In this case, we call such pair of EBSL/PBSL a “homonymous” straight line pair.

Such occurrences of two lines are found to be with small positional differences when their support region is ideal in shape, typically when the region is long, straight, and with high contrast. However, in common cases, a “homonymous” straight line pair has distinct positional, orientation, and length differences between its two parts. This is partially due to the influence of image noise and the limitations of the method including the edge detection, support region searching, and straight line fitting. Furthermore, when a region has an irregular shape, the two lines may have more obvious differences due to fitting error. Since the EBSL represents the actual image edge position, it theoretically has higher positional accuracy than the “virtual” PBSL originated from line fitting. These arguments suggest that we should preferentially keep the EBSL during the trimming process when the line pair has positional inconsistency.

Our feature fusion strategy thus includes the following steps. All EBSLs with no intersections with any eligible line-support regions are kept. All PBSLs with no EBSLs intersecting their support regions are also kept. For each PBSL that has a support region intersected by an EBSL, a set of feature fusion rules which follow a “Delete-PBSL-first” strategy are designed adapting to different intersecting situations.

In Figure 1, a gray region denotes a line-support region, a dotted line is its PBSL; the solid line represents an EBSL intersecting with this region; and \(\theta\) is the angle between the two lines. If \(\theta\) is larger than some threshold \(\theta_c\), the two lines are regarded as nonhomonymous. Figures 1(a)–(c) illustrate three different cases with large \(\theta\). Figure 1(a) is for illustrative purposes only, since it cannot occur in reality. The gradient phases within the region will be inconsistent due to the intersection which will break the support region. Figure 1(b) occurs when inaccurate fitting originating from the irregular support region. In this case, the PBSL is deleted since it is either a false alarm or with low positional accuracy and the EBSL is kept. In Figure 1(c), an EBSL touches an eligible line-support region with a large \(\theta\). This situation can occur in real situations and both lines are kept.

If \(\theta\) is smaller than \(\theta_c\), otherwise, the line pair is regarded as collinear and this homonymous line pair needs to be trimmed. Figures 1(d)–(f) illustrate the cases where the two lines are collinear. They are treated uniformly according to the following rules.

As shown in Figure 2, \(AB\) is an EBSL, \(CD\) is a PBSL, points \(c\) and \(d\) are the projections of nodes \(C\) and \(D\) on \(X\) or \(Y\) axis, \(A'\) and \(B'\) are the projections of points \(A\) and \(B\) on line \(CD\), and points \(a\) and \(b\) are the projections of nodes \(A'\) and \(B'\) on \(X\) or \(Y\) axis. If \(a < c\) and \(b > d\), only \(AB\) is kept; if \(a > c\) and \(b < d\), only \(CD\) is kept; if \(a > c\) and \(d < b\), according to the “Delete-PBSL-first” strategy, \(AB\) and the segment \(CA'\) of \(CD\) are kept and connected into a single straight line; and if \(a < c\) and \(d > b\), then \(AB\) and \(DB'\) are kept.

### 3. Experimental analysis

#### 3.1. Experimental settings

For a Canny edge detector, three inputs are needed; a Gaussian standard deviation \((\sigma)\), a high threshold \((T_1)\), and a
ratio of low threshold to high threshold ($T_2$). For the phase grouping method, only pixels with their gradients higher than a minimum gradient threshold (GT) are involved in searching the line-support regions. If line-support regions are only constructed by the same-phase constraint, this method tends to create many trivial, irregular, or even false support-regions due to the influence of noise. The threshold GT is set so as to minimize such problems. While these inputs can be tuned to suppress false straight lines due to noise, it is difficult to prevent loss of true lines in both methods. In our method, these parameters are tuned to maintain high line-extraction completeness while suppressing noise as much as possible. The assumption is that lost line segments are restored, or partially restored, by the fusion of the complementary method.

Our experimental data came from the Advanced Land Observing Satellite panchromatic images (2.5 m) in Jiangning, Jiangsu province, China acquired on 14 February 2007 (SceneID: ALPSMW103282960; spatial extent: 118.421°E–119.257°E and 31.583°N–32.039°N). Ten subsets of the images were clipped for validation of the proposed method. The algorithm inputs were tuned and uniformly used as follows. For the Canny edge detector, $\sigma$ was set to 1.0, $T_1$ was 0.7, and $T_2$ was 0.7. For straight line tracing, $d_{\text{max}}$ was set to 2.5 pixels and $l_{\text{min}}$ was 10 pixels. For the phase grouping method, GT was 5.0 and minimum line length was 10 pixels. For straight line fusion, $\theta_i$ of $\pi/8$ was found a suitable setting to remove false PBSLs.

### Table 1. Method performance.

| Area | Image size (in pixels) | EBSL length (in pixels) | PBSL length (in pixels) | Fusion line length (in pixels) | EBSL kept | PBSL kept |
|------|------------------------|-------------------------|-------------------------|-------------------------------|-----------|-----------|
|      |                        |                         |                         |                               | Length    | Contribution ratio (%) |
| 1    | 768 × 600              | 34,182                  | 27,885                  | 40,612                        | 31,603    | 77.8      |
| 2    | 1760 × 1788            | 234,448                 | 158,764                 | 267,586                       | 222,029   | 83.0      |
| 3    | 883 × 638              | 42,547                  | 33.711                  | 49,842                        | 39,917    | 80.1      |
| 4    | 482 × 408              | 14,183                  | 9658                    | 15,337                        | 12,299    | 80.2      |
| 5    | 691 × 638              | 31,734                  | 28,391                  | 37,483                        | 28,922    | 77.2      |

|                  |                        |                         | Fusion line length (in pixels) | Contribution ratio (%) | Length    | PBSL kept | Contribution ratio (%) |
|                  |                        |                         |                               |                         |           |           |
|                  |                        |                         |                               |                         | 9009      | 22.2      |
|                  |                        |                         |                               |                         | 45,557    | 17.0      |
|                  |                        |                         |                               |                         | 9925      | 19.9      |
|                  |                        |                         |                               |                         | 3038      | 19.8      |
|                  |                        |                         |                               |                         | 8561      | 22.8      |

![Figure 3](image-url) Straight line detection.

[Figure 3. Straight line detection.]
3.2. Performance of the fusion rules

Table 1 lists five experimental results and indicates that the lengths of the PBSLs are consistently less than those of the EBSLs. This indicates the phase grouping method is more sensitive to image noise since more valid lines are discarded when suppressing the false lines. The line lengths of the fusion method are, as expected, longer than each single method. In addition, the contribution ratios of the EBSLs are far higher than those of the PBSLs, which show that more of the latter are replaced by the former with higher positional precision.

Figure 3 illustrates the result in one experimental area. Figure 3(a) is the experimental area, Figure 3(b) is the edge map obtained by the Canny operator, and Figure 3(c) is the raw EBSLs before straightening. We see that false lines are totally eliminated while some broken lines emerge. Figure 3(d) shows the support regions, while Figure 3(e) exhibits the PBSLs with many fresh line segments not contained in the EBSLs. Figure 3(f) shows the fusion results with mutual supplementary information in many positions (compare the areas indicated by the rectangles in Figure 3(c), (e) and (f)). Figure 4 presents several zoom-in versions of different fusion cases. In Figure 4, the pink lines are the EBSLs, the yellow lines are the PBSLs, the green areas are the line-support regions, and the white lines are the Canny edges.

In Figure 4(a1)-(a4), the support region is associated with a false alarm for a straight line. An EBSL runs across the region with a large θ, as exemplified in Figure 1(b). In this case, the false PBSL is eliminated and the EBSL is kept. The fusion result is shown in Figure 4(a4), which is a more objective representation of the image context.

In Figure 4(b1)-(b4), an EBSL touches a support region with a large angle. In this case, this EBSL is kept according to the rule shown in Figure 1(c); the EBSL and PBSL within the support region are trimmed. The final fusion result is shown in Figure 4(b4).

In Figure 4(c1)-(c4), an EBSL runs completely across a support region. There exist some positional deviations between the PBSL and EBSL pair, while the EBSL has higher positional accuracy. Following the rule as in Figure 1(d), the entire EBSL is kept, the PBSL is fully removed, and the fusion yields a more accurate result.

Figure 4. Different fusion cases. (a1)-(a4) show an actual case corresponding to Figure 1(b); (b1)-(b4) illustrate Figure 1(c); in (c1)-(c4), a PBSL with low positional accuracy is replaced by a more accurate EBSL, which corresponds to Figure 1(d); (d1)-(d4) illustrate a complementary case as in Figure 1(e); and (e1)-(e4) show a case corresponding to Figure 1(f) where a PBSL replaces an incomplete EBSL.
In Figure 4(d1)–(d4), the EBSL is not ideal due to a rupture. While the PBSL has a similar defect, fortunately the two lines are complementary to each other. The straight line is extracted by both methods, which exemplifies Figure 1(e), and the result is shown in Figure 4(d4).

In Figure 4(e1–e4), the straight line is more completely extracted as a PBSL. In this case, the PBSL replaces the EBSL which exemplifies Figure 1(f), and the result is shown in Figure 4(e4).

### 3.3. Method robustness analysis

The fusion also makes the method robust for input variation. As illustrated in Figure 5, the hatched area denotes the duplicate extracted lines and the left and right blank areas denote the lines extracted by a single method only. The whole rectangle denotes the final straight lines, which is the sum of the kept EBSLs and PBSLs. Since some EBSLs may be replaced by more eligible PBSLs, the area of the kept EBSLs may be less than that of the extracted ones.

If the inputs of the edge-based method are changed to reduce the EBSLs, the rectangle of the extracted EBSL may shrink as illustrated in Figure 5(b). If the PBSLs are kept unchanged, the area of the duplicated lines will shrink.

Although the extracted EBSLs are reduced, it is fortunate that the “PBSL only” area expands which compensates for this loss. This causes less loss of the final kept lines after the line fusion than in the edge-based method. A similar conclusion can also be applied to the cases when the inputs of the phase-based method or both methods change. This indicates the fusion method may be more robust to the algorithm setting variation than each single method.

These analyses are verified in Table 2, which lists the responses of the three methods with different settings. We changed the algorithm inputs to reduce the extracted lines for both methods, and arbitrarily combined them to investigate the outputs of the fusion method. The method completeness is presented in the column “Completeness.” Completeness is the ratio of the extracted line length to that of the fusion method obtained with the “ideal” settings as described previously, while the latter is regarded as an estimation of the actual line length in the experimental area (the first row of the table). For example, the completeness of the EBSLs at the second row is 30,650/40,612 = 75.5%, which means about 75% of the actual lines are extracted by the edge-based method with the specified inputs.

The two methods compensate each other. For example, from rows 2 to 4, the length of the extracted PBSLs declines due to the increase of GT, while the kept EBSLs increase which reduces the total loss. The other rows present a similar pattern. We find that under more rigorous conditions, the completeness of the fusion method is evidently higher than each single method alone, denoting that fusion is effective and makes it less dependent on the algorithm inputs.

### Table 2. Method robustness.

| $T_1$ | $T_2$ | $\sigma$ | GT | EBSL extracted | PBSL extracted | EBSL kept | PBSL kept | Fusion line length | Completeness |
|-------|-------|---------|----|----------------|----------------|-----------|-----------|-------------------|--------------|
| 0.70  | 0.7   | 1.0     | 5  | 34,182         | 27,885         | 31,603    | 9009      | 40,612            | 84.2         |
| 0.75  | 0.7   | 1.5     | 5  | 30,650         | 27,885         | 28,503    | 10,539    | 39,042            | 75.5         |
| 0.75  | 0.7   | 1.5     | 10 | 30,650         | 24,769         | 29,161    | 7767      | 36,928            | 75.5         |
| 0.75  | 0.7   | 1.5     | 15 | 30,650         | 21,994         | 29,677    | 5672      | 35,349            | 75.5         |
| 0.75  | 0.7   | 2.0     | 5  | 29,587         | 27,885         | 26,946    | 11,712    | 38,658            | 72.9         |
| 0.75  | 0.7   | 2.0     | 10 | 29,587         | 24,769         | 27,513    | 9326      | 36,839            | 72.9         |
| 0.75  | 0.7   | 2.0     | 15 | 29,587         | 21,994         | 27,970    | 7507      | 35,477            | 72.9         |
| 0.80  | 0.7   | 1.5     | 5  | 26,045         | 27,885         | 23,589    | 12,826    | 36,415            | 64.1         |
| 0.80  | 0.7   | 1.5     | 10 | 26,045         | 24,769         | 24,289    | 9897      | 34,186            | 64.1         |
| 0.80  | 0.7   | 1.5     | 15 | 26,045         | 21,994         | 24,719    | 7460      | 32,179            | 64.1         |
| 0.80  | 0.7   | 2.0     | 5  | 24,304         | 27,885         | 21,173    | 15,078    | 36,251            | 59.8         |
| 0.80  | 0.7   | 2.0     | 10 | 24,304         | 24,769         | 21,798    | 12,240    | 34,038            | 59.8         |
| 0.80  | 0.7   | 2.0     | 15 | 24,304         | 21,994         | 22,340    | 9786      | 32,126            | 59.8         |
4. Conclusion

Straight line detection is an important issue in the fields of remote sensing information extraction and target recognition. Currently, the two main categories of straight line detection methods, including the edge-based and phase-based, both have some limitations. In this study, an integrated straight line detection method was proposed that adopts the advantages of both kinds of straight line detection methods. With carefully designed fusion rules, our method in combination extracts straight lines with better performance than each single method alone. It should be noted that our method offers an open framework for the integration of edge and phase information for straight line detection. It would be possible to replace the edge detector, edge- or phase-based line detection methods, to potentially create a more robust detection method.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 41171321, 40871189); the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (No. 11KJA420001); the Priority Academic Program Development of Jiangsu Higher Education Institutions; the Qin Lan Project of Jiangsu, China.

Notes on contributors

Min Wang, is a professor. His research interests include remote sensing image processing, information extraction, and spatial data mining.

Qingfeng Zhang, is a graduate student. His research interests include remote sensing image processing and information extraction.

References

(1) Nevatia, R.; Price, K.E. Locating Structures in Aerial Images. *IEEE Trans. Pattern Anal. Mach. Intell.* 1982, PAMI-4, 476–484.
(2) Stamos, I.; Leordeanu, M. Automated Feature-Based Range Registration of Urban Scenes of Large Scale. *IEEE Conference on Computer Vision and Pattern Recognition*, Madison, WI, 2003.
(3) Ünsalan, C.; Boyer, K.L. Classifying Land Development in High-Resolution Satellite Imagery Using Hybrid Structural-Multispectral Features. *IEEE Trans. Geosci. Remote Sens.* 2004, 42, 2840–2850.
(4) Ünsalan, C. Gradient-Magnitude-Based Support Regions in Structural Land Use Classification. *IEEE Geosci. Remote Sens. Lett.* 2006, 3, 546–550.
(5) Rowe, N.C.; Grevé, L.L. Change Detection for Linear Features in Aerial Photographs Using Edge-Finding. *IEEE Trans. Geosci. Remote Sens.* 2001, 39, 1608–1612.
(6) Li, W.; Li, X.; Wu, Y.; Hu, Z. A Novel Framework for Urban Change Detection Using VHR Satellite Images. 18th International Conference on Pattern Recognition, Hong Kong, 2006.
(7) Illingworth, J.; Kittler, J. A Survey of the Hough Transformation. *Pattern Recognit.* 1988, 20, 87–116.
(8) Kong, C.W.; Park, R.H.; Lee, K.H. Extraction of Straight Line Segments Using Rotation Transformation: Generalized Hough Transformation. *Pattern Recognit.* 1991, 24, 633–641.
(9) Yan, G.; Li, C.; Zhou, G.; Li, X. Automatic Extraction of Power Lines from Aerial Images. *IEEE Geosci. Remote Sens. Lett.* 2007, 4, 387–391.
(10) Nevatia, R.; Babu, K.R. Linear Feature Extraction and Description. *Comput. Vision. Graph.* 1980, 13, 257–269.
(11) Venkateswar, V.; Chellapa, R. Extraction of Straight Lines in Aerial Images. *IEEE Trans. Pattern Anal. Mach. Intell.* 1992, 14, 1111–1114.
(12) Boldt, M.; Weiss, R.; Riseman, E. Token-Based Extraction of Straight Lines. *IEEE Trans. Syst. Man Cybern.* 1989, 19, 1581–1594.
(13) Lee, Y.S.; Koo, H.S.; Jeong, C.S. A Straight Line Detection Using Principal Component Analysis. *Pattern Recognit. Lett.* 2006, 27, 1744–1754.
(14) Burns, J.B.; Hanson, A.R.; Riseman, E.M. Extracting Straight Lines. *IEEE Trans. Pattern Anal. Mach. Intell.* 1986, PAMI-8, 425–456.
(15) Canny, J.F. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 1986, 8, 679–698.
(16) Douglas, D.; Peucker, T. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *Can. Cartogr.* 1973, 10, 112–122.