IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 9, 2019, accepted January 18, 2019, date of publication January 24, 2019, date of current version February 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2895016

Extracting Centerlines From Dual-Line Roads Using Superpixel Segmentation

YILANG SHEN, TINGHUA AI, AND MIN YANG
School of Resource and Environment Sciences, Wuhan University, Wuhan 430079, China

Corresponding author: Tinghua Ai (tinghuaai@whu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 41531180, and in part by the National Key Research and Development Program of China under Grant 2017YFB0503500.

ABSTRACT Extracting centerlines from dual-line roads is very important in urban spatial analysis and infrastructure planning. In recent decades, numerous algorithms for road centerline extraction based on the vector data have been proposed by various scholars. However, with the continual development of computer vision technology, advances in the corresponding theories and methods, such as superpixel segmentation, have provided new opportunities and challenges for road centerline extraction. In this paper, we propose a new algorithm called superpixel centerline extraction (SUCE) for dual-line roads based on the raster data. In this method, dual-line roads are first segmented using a superpixel algorithm called simple linear iterative clustering. Then, the superpixels located at road intersections are merged to generate connection points from their skeletons. Finally, the centerlines of roads are generated by connecting the center points and edge midpoints of each superpixel. To test the proposed SUCE method, the vector data of roads at a scale of 1:50 000 from Shenzhen, China, and the raster data of roads at the 18th level from the Tiandi map are used. Compared with a traditional method in ArcGIS software (version 10.2) based on the vector data and four existing thinning algorithms based on the raster data, the results indicate that the proposed SUCE method can effectively extract centerlines from dual-line roads and restore the original road intersections while avoiding burrs and noises, both for simple and complex road intersections.

INDEX TERMS Centerline extraction, dual-line roads, image data, superpixel segmentation.

I. INTRODUCTION

Centerline extraction (also called skeleton line, axis line, or medial line extraction) is an interesting problem in pattern recognition, spatial analysis, map generalization, and urban infrastructure planning [1], [2]. Roads are among the most abundant elements on many maps, and the centerline extraction of roads is necessary for spatial analyses and other purposes. For example, urban road infrastructure location is an analysis mainly based on street networks. Good centerline extraction algorithms that can preserve the correct road intersections are very important for business applications such as route planning, vehicle navigation and public facilities accessibility. Map generalization is a process that simplifies the representation of geographical data to produce a map with a smaller scale than the original data [3], [4]. During map generalization, centerline extraction is necessary because geographical features, such as rivers and roads, must be represented with single lines when the scale is sufficiently small. Thus, centerlines are the key elements used in the dissolve and collapse operators for map generalization.

According to different data types, studies of the skeleton extraction of area features can be divided into raster-based and vector-based methods. For vector data, scholars mainly use Delaunay triangulation [5]–[16] and straight skeletons [17]–[21] for the skeleton extraction of area features. DeLucia and Black [9] first used constrained Delaunay triangulation for the extraction of polygonal skeletons. In this method, the triangles in a triangular mesh are divided into three types according to the number of polygon edges. Then, the skeletons of the objects can be generated by different types of triangles. This method based on constrained Delaunay triangulation provided the foundation for subsequent research. Zou and Yan [11] proposed a method for the extraction of polygon skeletons using constrained Delaunay triangulation. Based on the method proposed by Zou and Yan [11], Morrison and Zou [12] developed an improved method for the extraction of polygon skeletons.

Jones *et al.* [15] adjusted the type-III triangles to extract the T-shaped junctions by calculating the direction of the three associated skeleton line branches. However, this method has shortcomings in handling '+'-shaped branches. Subsequently, Penninga *et al.* [13] modified the method proposed by Jones *et al.* [15] to solve the issues associated with the extraction of '+'-shaped branches. Gao and Minami [16] developed an approach to address X- and Y-shaped junctions based on constrained Delaunay triangulation. In addition, McAllister and Snoeyink [7] and Regnauld and Mackaness [8] developed methods of extracting river centerlines based on Delaunay triangulation. In summary, the existing methods based on Delaunay triangulation can solve only relatively simple problems related to polygon skeletons, and they display shortcomings in certain complex situations, such as a scenario with skeleton branches in different directions [22]. Another method based on the straight skeleton method [17]–[21] was first proposed by Aichholzer *et al.* [17]. They extracted skeletons by connecting straight-line segments consisting of angular bisectors of polygon boundaries. The original method based on a straight skeleton could address only some simple polygons. Thus, Das *et al.* [18] and Eppstein and Erickson [19] improved the original straight skeleton method for complex polygons and based on the extremes of binary functions. Haunert and Sester [20] used the straight skeleton method for area collapse in map generalization, and the method was proven effective for area-line geometry changes. In addition, Lee [23] and Montanari [24] applied Voronoi diagrams to obtain the skeleton lines of polygon boundaries.

Studies based on raster data for centerline or skeleton extraction mainly include image thinning algorithms [25]–[43]. Deng *et al.* [28] developed a one-pass parallel asymmetric thinning algorithm that can generate 8-connected skeleton outputs without serious erosion and with good noise resistance at a relatively fast speed. Ahmed and Ward [32] proposed a thinning algorithm based on a rule system. They developed 20 rules in an inference engine that are concurrently applied to each pixel in an image. This method is very efficient for symmetrical thinning and maintaining the topology of letters and symbols. Wu and Tsai [39] proposed a one-pass parallel thinning method for binary images. In this method, template matching is used to iteratively remove edge points. The proposed method has the advantages of obtaining skeletons with 8-connected properties and preserving the topology of the original shape. In addition, it is insensitive to noise. Zhou *et al.* [40] proposed a sequential thinning method in which both a bitmap and flag map are used to delete the boundary pixels, and the final skeleton is smoothed using smoothing templates. The Zhang and Suen (ZS) algorithm [31] is a classic and popular iterative parallel thinning algorithm based on a 3 × 3 neighborhood that is simple, proven and efficient. Based on the ZS algorithm, many scholars proposed improved methods for image thinning [41]–[43]. Due to acute angles, the ZS algorithm can generate redundant segments; Chen *et al.* [41] proposed a thinning method for removing the redundant segments based on the ZS algorithm. This method was proven effective for solving the problems in the ZS algorithm while maintaining its characteristics. Lü and Wang [42] developed a fast parallel thinning method based on the ZS algorithm. The method can maintain the advantages of the ZS algorithm and overcome some disadvantages in the ZS algorithm. In addition, this method is faster than the original algorithms. Boudaoud *et al.* [43] proposed a modified ZS (MZS) algorithm for image thinning. The MZS method was compared against seven existing thinning algorithms, and the results showed that the MZS thinning method is more than 21 times faster than the existing CPU sequential version. Thomas [44] proposed another method of centerline generation for roads by transforming vector data to raster data. Nevertheless, the current image thinning algorithms struggle to meet the demands of map generalization, such as maintaining road intersections when extracting centerlines.

As the number of unstructured data sets has grown, map generalization based on image data has become increasingly necessary [45]. For example, Shen *et al.* [46] proposed a method for polygonal boundary simplification based on image processing in 2018. The existing raster-based methods used for centerline extraction have shortcomings for maintaining original road intersections, especially in complicated situations. Moreover, with the continual development of computer vision technology, advanced theories and methods, such as superpixel segmentation, have provided new opportunities and challenges for centerline extraction. Thus, in this article, we develop a new method for road centerline extraction based on superpixel segmentation technology.

The outline of this paper is organized as follows. Section 2 compares the superpixel and centerline extraction to lay a theoretical foundation for the centerline extraction of roads based on raster data. In section 3, we present the methods for road centerline extraction, which mainly include superpixel segmentation of the original roads, superpixel merging for generating connection points from skeletons, and centerline generation by connecting the center points and edge midpoints of each superpixel. Section 4 presents the experiment results that illustrate the proposed SUCE method and provides an analysis and discussion of the proposed SUCE method compared with the traditional methods in ArcGIS software and the ZS thinning algorithms. Finally, section 5 presents the conclusions.

II. SUPERPIXEL AND CENTERLINE EXTRACTION

The concept of the superpixel was first announced by Ren and Malik [47] at the Ninth IEEE International Conference on Computer Vision. By analyzing the intensities, colors, textures and other characteristics of the adjacent standard pixels, a pixel block called a superpixel, larger than normal pixels with uniform color and brightness can be generated. These standard pixels consist of a superpixel locally have the similar features. Using the superpixel technology, the complexity of the original image can be effectively reduced and the expression of the original image can be more simplified. The motivations of superpixel segmentation are that [47]:

(1) the standard pixels are only a discrete representation of objects, they cannot represent natural entities; and (2) it is difficult for optimization at the normal pixel level due to huge number of pixels. Thus, the superpixels with locally coherent features can preserve the main information of images at the level of interest.

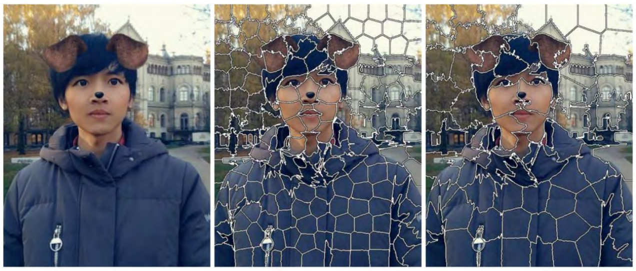

FIGURE 1. An example of superpixels.

Basically, superpixel segmentation is a process of simplifying the representation of images based on clustering. Figure 1 shows a typical example of superpixel segmentation at two different scales. The size of the original image in Figure 1 (left) is 570 × 720, The sizes of the superpixels used for the segmentation are S = 2000 (middle), and S = 5000 (right) pixels, respectively. By increasing the size of each superpixel, the original objects are divided into larger segments with similar features. After the superpixel segmentation, the expression of the original human is simplified using different superpixel sizes at different scales. In the process of segmentation, the main information that we are interested in is preserved and similar details are aggregated into superpixels by clustering. Analogously, centerline extraction is a simplification process that converts two-dimensional information to one-dimensional information. Common characteristics exist between centerline extraction and superpixel segmentation. Notably, both are simplified representations of objects in which multiple or multidimensional objects are simplified. In this process, the minor details are removed, but the main information is maintained. Superpixel segmentation technology was first used by Shen *et al.* [48] to simplify polygonal and linear features on a map. The method proposed by Shen *et al.* [48] is called simplification using superpixel segmentation (SUSS). However, they applied the superpixel technology only to objects with the same dimensions. Because the basic principle of superpixel segmentation and centerline extraction is the same, in this article, image analysis and processing technologies based on superpixel segmentation are applied for road centerline extraction with dimensional changes.

III. METHODOLOGIES FOR ROAD CENTERLINE EXTRACTION

As shown in Figure 2, the methodologies for road centerline extraction can be divided into three steps: (1) the superpixel segmentation of the original roads using a superpixel algorithm called simple linear iterative clustering (SLIC) [49]; (2) merging superpixels located at road intersections, which mainly includes corner detection for selecting the superpixels and dilation and erosion for merging; and (3) the generation of road centerlines by connecting the center points and edge midpoints of the superpixels, which mainly includes extracting the center points of type-M superpixels based on skeleton lines and extracting the center points of type-N superpixels using image moment.

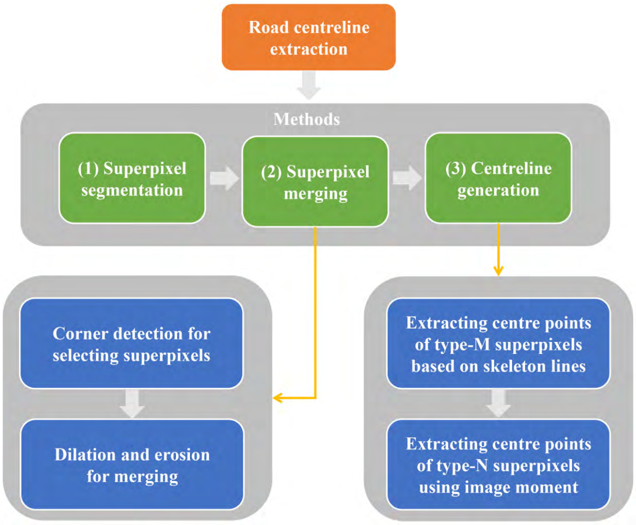
Road centreline extraction
Methods
(1) Superpixel segmentation
(2) Superpixel merging
(3) Centreline generation
Corner detection for selecting superpixels
Dilation and erosion for merging
Extracting centre points of type-M superpixels based on skeleton lines
Extracting centre points of type-N superpixels using image moment

FIGURE 2. Steps in road centerline extraction based on superpixel segmentation.

A. SUPERPIXEL SEGMENTATION BASED ON THE SLIC ALGORITHM

Using superpixel technology, the original normal pixels can be clustered into different subareas of interest. Thus, the redundancy of the original image is reduced. Superpixel segmentation technology can improve various image processing tasks, such as depth estimation [50], body model estimation [51], object localization [52], and object segmentation [53]–[55]. Since the superpixel concept was first proposed, many methods of generating superpixels have been presented [56]–[60]. However, in this study, the superpixel segmentation algorithm called SLIC is applied to segment polygonal features. This method has two distinct advantages: (1) boundaries are maintained better than in other algorithms, and (2) SLIC is fast, efficient and easy to implement. The SLIC algorithm applies a k-means clustering method to produce superpixels. A weighted distance that considers both spatial proximity and color is used to cluster the local pixels. However, because considering color when extracting road centerlines is meaningless, we remove the color distance and consider only the spatial distances between pixels when using the SLIC algorithm.

Figure 3 shows an example of the superpixel segmentation of roads. The size of the original image used to generate the superpixels is 2500 × 2500 pixels. The sizes of the superpixels used for the segmentation in Figures 3b-3d are S = 1000,

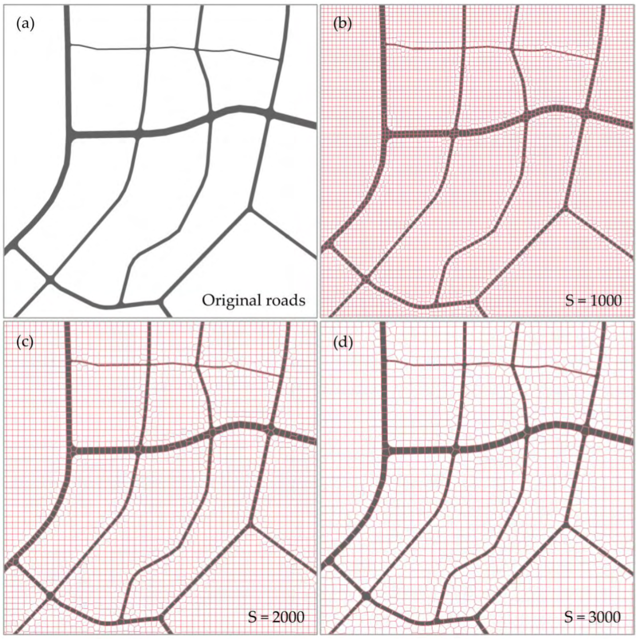

FIGURE 3. Superpixel segmentation based on the SLIC algorithm.

S = 2000, and S = 3000 pixels, respectively. As the size of each superpixel increases, the original roads are divided into more complete segments.

B. MERGING SUPERPIXELS LOCATED AT ROAD INTERSECTIONS

After performing superpixel segmentation using the SLIC algorithm, the original roads are divided into many superpixels that are regularly arranged in the direction of the original roads, as shown in the regions marked in green in Figure 4a. However, the superpixels located at road intersections should be merged because the original intersection features are segmented by several superpixels, as shown the regions marked in red in Figure 4a. Figure 4b shows the merging results for the superpixels in Figure 4a.

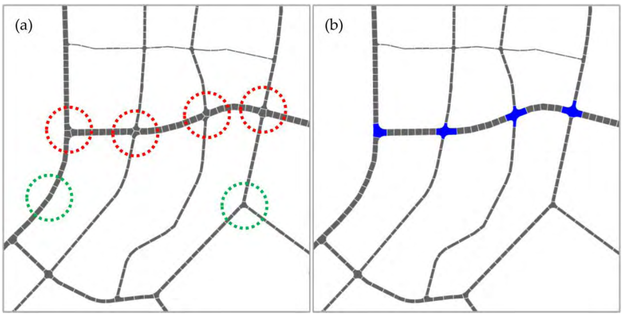

FIGURE 4. Merging superpixels located at road intersections.

The detailed merging of superpixels is based on a method involving the corner detection of superpixels. In general, a corner can be considered as a local point with two different edge directions. Corners have various forms, such as a separate point with a locally maximal or minimal intensity, a point on a curve with a local curvature maximum or a line end. Corners are usually used to represent local critical features of images. Corner detection is very useful for many applications, such as image matching, object recognition and motion tracking [61]–[63]. Therefore, many scholars have proposed a number of methods for corner detection [64]–[68]. The corners of each superpixel are detected using the method proposed by He and Yung [69]. The method of corner detection based on curvature scale space describes three important parameters: (1) angle (A): represents the maximum obtuse angle used for screening true corners; (2) ratio (R): represents the minimum ratio of the major axis to the minor axis of an ellipse; and (3) endpoint (E): represents a flag to decide whether to consider the end points on a curve as corners. When E = 1, it means yes. When E = 0, it means no. The default values of these three parameters are A = 162, R = 1.5 and E = 1. The values of these three parameters are not fixed but are relatively reasonable in this experiment. More information about these three parameters can be found in the research of He and Yung [69].

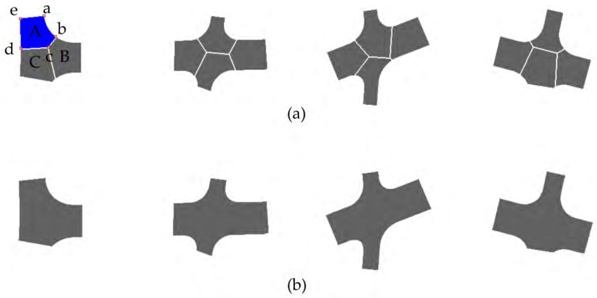

FIGURE 5. Steps in merging superpixels.

Next, superpixel selection is performed. If the corner of a superpixel is located in the interior of an original road, the superpixel should be selected to be merged. For example, as shown in Figure 5a, there are five corners (a, b, c, d and e) of superpixel A marked in red. Corner c is located in the interior of the original road pixels, so superpixel A should be selected for merging. Next, dilation and erosion operations are used to merge the selected superpixels. Figure 5b shows the detailed merging results of the superpixels from Figure 5a.

C. EXTRACTING THE CENTER POINTS OF SUPERPIXELS AND GENERATING THE CENTERLINES OF ROADS

Two types of superpixels exist after merging superpixels: type-M and type-N superpixels. When there are more than two intersecting line segments between a superpixel and the interior of an original road, the superpixel is considered type M. As shown in Figure 6, the superpixel marked in yellow is a type-M superpixel because it contains four intersecting line segments between the superpixel and the interior of the original road, which are marked in blue. The superpixel marked in green is a type-N superpixel because it contains only two intersecting line segments between the superpixel

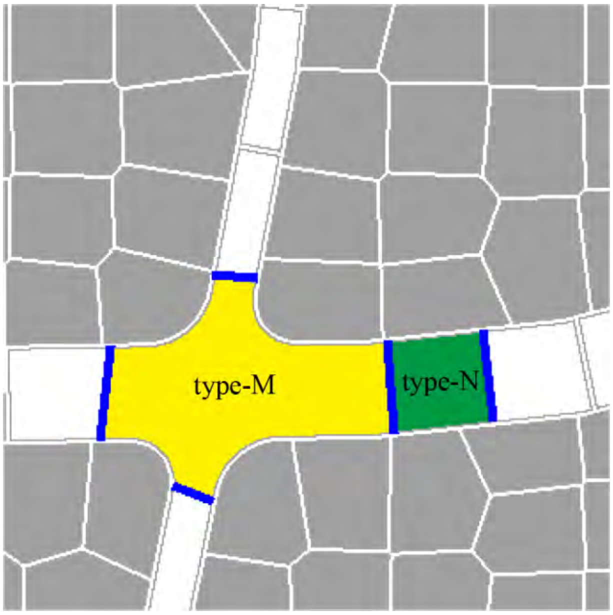

FIGURE 6. Type-M and Type-N superpixels.

and the interior of the original road, which are marked in blue. When there are only two intersecting line segments between a superpixel and the interior of an original road, the superpixel is classified as type-N. The methods used for extracting the center points of type-M superpixels are as follows.

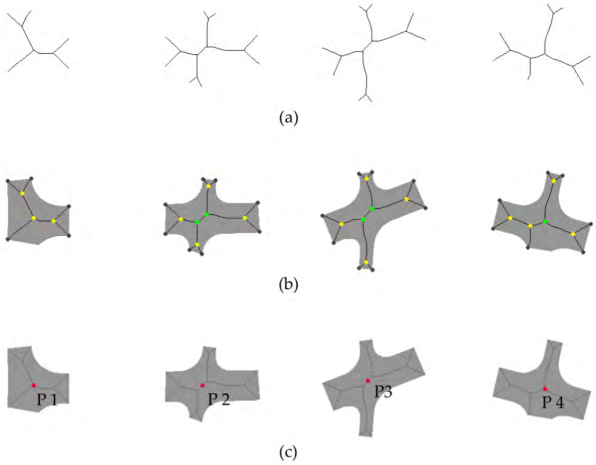

FIGURE 7. Steps for generating the center points of merged superpixels.

First, the skeleton lines of each type-M superpixel are extracted using the algorithm proposed in [70]. Figure 7a shows the skeleton lines of the four merged superpixels.

Then, three types of nodes along the skeleton lines are identified. Type-I nodes are located at the boundaries of the original roads, such as the nodes marked in gray in Figure 7b. Type-II nodes, such as the nodes marked in yellow in Figure 7b, directly connect to type-I nodes via skeleton lines. Type-III nodes are all the remaining nodes connected to the type-II nodes, such as those marked in green in Figure 7b. Finally, the center points of each type-M superpixel can be determined by the following rules. (1) When the number of type-III nodes is zero and the number of type-II nodes is odd, the type-II node located in the middle will be selected as the center point, such as the red point P1 in Figure 7c. When the number of type-III nodes is zero and the number of type-II nodes is even, the midpoint of the two type-II nodes located in the middle will be selected as the center point. (2) When the number of type-III nodes is one, this type-III node will be selected as the center point, such as the red point P4 in Figure 7c. (3) When the number of type-III nodes is two, the midpoint of these two nodes will be selected as the center point, such as the red points P2 and P3 in Figure 7c. (4) When the number of type-III nodes is more than two and the number of type-III nodes is odd, the type-III node located in the middle will be selected as the center point. When the number of type-III nodes is greater than two and the number of type-III nodes is even, the midpoint of two type-III nodes located in the middle will be selected as the center point.

For the type-N superpixels, the center points can be confirmed by calculating the image moment [71]. In the field of image processing, an image moment represents a particular weighted average of image pixel intensities and is specifically used to describe an image object [71]. By calculating the image moments, some properties of the image, such as the area, centroid or orientation, can be acquired. A central moment is the moment of a probability distribution of a random variable about the mean of the random variable. This moment is the expected value of a specified integer power of the deviation of the random variable from the mean. For a digital image f(x,y) of size m × n, the moment of order (p+q) can be defined as follows.

$$M_{pq} = \sum_{x=1}^{m} \sum_{y=1}^{n} (x - \bar{x})^{p} (y - \bar{y})^{q} f(x, y) \tag{1}$$

Moments can be used to generate a series of values that are useful for characterizing the properties of a probability distribution. Compared with ordinary moments, central moments are more commonly used when computing deviations from the mean rather than from zero. Specifically, high order central moments are related to only the shape and spread of the distribution, rather than the location. Central moments can be defined as:

$$u_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^{p} (y - \bar{y})^{q} f(x, y)\, dxdy \tag{2}$$

where $\bar{x}$ and $\bar{y}$ are the components of the centroid. The components can be formulated as follows.

$$X = \bar{x} = \frac{M_{10}}{M_{00}}, Y = \bar{y} = \frac{M_{01}}{M_{00}} \tag{3}$$

Thus, (X, Y) are the coordinates of the center point of each type-N superpixel.

As shown in Figure 8, the centerlines of original roads can be generated by connecting the center points and edge midpoints of superpixels. To obtain the edge midpoints of

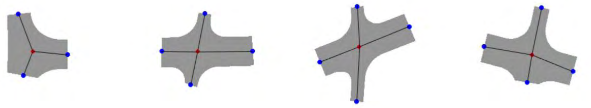

FIGURE 8. Connecting center points and edge midpoints.

the superpixels, the following two steps should be performed. First, the intersecting line segments between the superpixel and the interior of the original road of each superpixel should be calculated by overlaying. An example of intersecting line segments between the superpixel and the interior of the original road can be seen in Figure 6 (blue line segments). Then, the coordinates of the edge midpoints of each intersecting line segment can be acquired by calculating the midpoint between the two endpoints of each intersecting line segment.

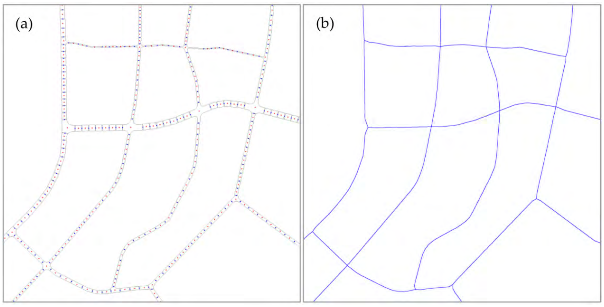

FIGURE 9. Final extraction results of road centerlines.

Figure 9a shows the center points and edge midpoints of all superpixels. The center points of the superpixels are marked in red and the edge midpoints of the superpixels are marked in blue. Figure 9b shows the final extraction results of centerlines from the original dual-line roads.

IV. EXPERIMENTS AND EVALUATIONS

A. CASE STUDY OF ROADS WITH SIMPLE INTERSECTIONS

In this experiment, the original road data with simple intersections used to test the proposed SUCE method are from the southeastern part of the Futian District of Shenzhen, China. The original scale of the data is 1:50000. Because the original roads are vector data, ArcGIS software (ESRI, Redlands, CA, USA) is used to convert the vector data to raster data. The corresponding toolbox for rasterization in ArcGIS is located in the following subdirectory: ''Toolboxes - Conversion Tools - To Raster - Polyline to Raster''. The cell size used in the ArcGIS conversion toolbox is 1, which means 1 pixel in raster data represents 1 m in vector data. As shown in Figure 10, the size of the image after conversion is 9922 × 7371 pixels. The standard road centerlines are provided by the surveying and mapping department of Shenzhen, China, and are shown in Figure 11.

To evaluate the proposed SUCE method, we used two algorithms to perform comparative experiments. The first is the ZS thinning algorithm, which is one of the most common methods used in the field of image processing. The other is called the ''collapse dual lines to centerline'' method in ArcGIS software. This approach is widely applied in the field of map generalization. Figure 11 shows the extraction results for road centerlines using the proposed SUCE method, from which we can see that the proposed SUCE method can effectively extract centerlines from dual-line roads. Figure 12 shows the extraction results of road centerlines using the two

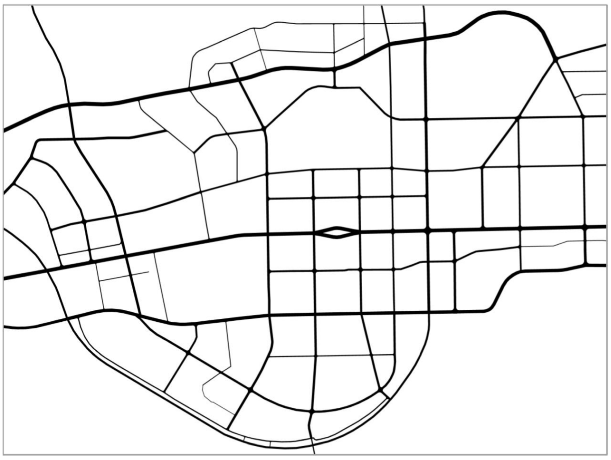

FIGURE 10. Original road data with simple intersections.

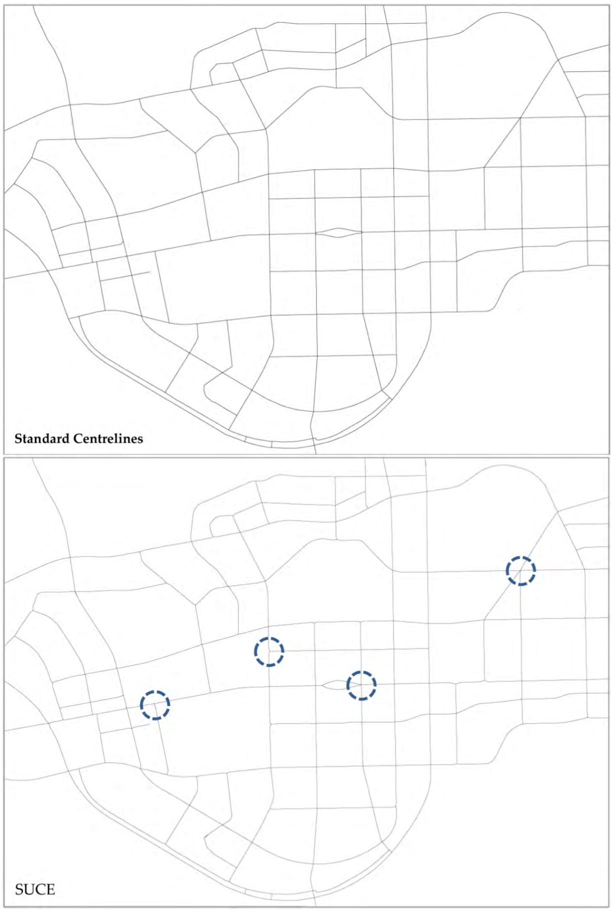

FIGURE 11. Extraction results for road centerlines using the proposed SUCE method.

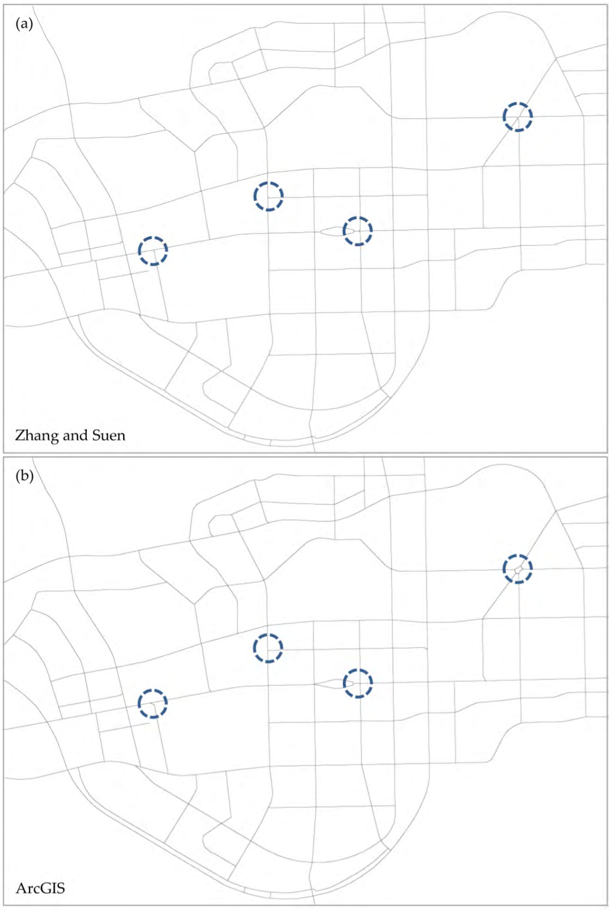

FIGURE 12. Extraction results for road centerlines using comparative methods.

comparative methods. Although these methods can be used to extract road centerlines, they yield different extraction results at road intersections. It is extremely important to properly handle branches when extracting the centerlines of roads, especially in regions with more than two branches. To better illustrate the differences in the results of the various methods, we select some typical regions to display the corresponding detailed information, such as the regions marked with blue circles in Figures 11 and 12. Figure 13 shows the detailed information at road intersections in these typical regions using the proposed SUCE method and the two comparative methods.

Figure 13 shows that for simple branches, the results from ArcGIS tend to generate vertical or inclined intersections, such as the road intersections in Figures 13a and 13d. The results of the ZS and SUCE methods tend to generate centerlines closer to the centers of the original roads, such as the results in Figures 13b-c and 13e-f. Additionally, for complicated intersections, the results from ArcGIS cannot maintain the original road junctions, as in Figure 13g, and in some cases, centerlines are not generated, such as in Figure 13j. Although the ZS method can generate centerlines, the original road junctions are not always maintained and branches still exist in centerline results, as shown in Figures 13h and 13k.

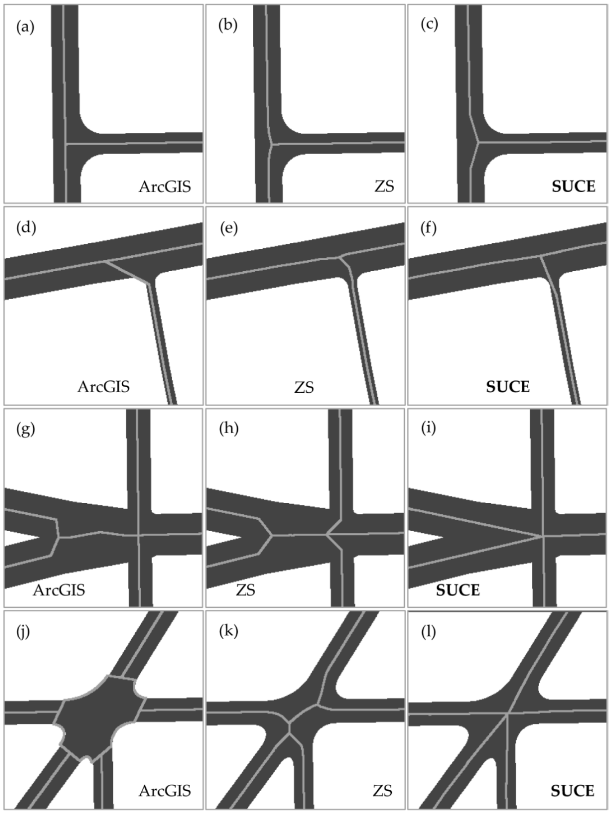

FIGURE 13. Detailed information at road intersections.

However, the proposed SUCE method can effectively extract centerlines from dual-line roads in the case of complicated road intersections and effectively preserve road junctions. Additionally, no redundant branches exist in the results, as shown in Figures 13i and 13l.

To evaluate the restoration accuracy for the roads with simple intersections, we compared both the geometric accuracy and number of incorrect road intersections of these three methods. To evaluate the geometric accuracy of centerlines, the method of measuring the positional accuracy of linear features proposed by Goodchild and Hunter [72] is applied. Because the original roads and centerline results from the ArcGIS method are vector data, we convert them to raster data using the ArcGIS conversion toolbox. The cell size used in the ArcGIS conversion toolbox is 1, which means 1 pixel in raster data represents 1 m in vector data. Based on the standard centerlines, buffers of different sizes (1 pixel, 2 pixels, 3 pixels and 4 pixels) are created. Then, the length percentages of roads located in the buffers are calculated. Table 1 shows the results of the evaluation of geometric accuracy for the ArcGIS, ZS and SUCE methods. Notably, as the buffer size increases, the length percentages for the three methods increase. Although the geometric accuracy of the ArcGIS method is better than that of the other two methods, the percentages obtained with the proposed SUCE method are still above 90%, and the results are considered acceptable. In addition, the ArcGIS method performs better

TABLE 1. Evaluation in the case of simple road intersections.

Types		ArcGIS	ZS	SUCE
Geometric accuracy	1 pixel	95.9 %	93.9 %	90.0 %
	2 pixels	97.4 %	96.6 %	93.0 %
	3 pixels	98.0 %	97.7 %	95.2 %
	4 pixels	98.1 %	98.2 %	96.3 %
Number of incorrect road intersections		8	32	0

because the standard centerline data tend to maintain vertical intersections rather than being closer to the centers of original roads in the case of simple branches, such as in the case of Figure 13a, compared to the other methods. We also calculated the number of incorrect road intersections. As shown in Table 1, the ZS method produces the most incorrect road intersections, which reaches up to 32. The ArcGIS method also produces 8 incorrect road intersections. However, the number of incorrect road intersections produced by the proposed SUCE method is zero, which means that the SUCE method can effectively restore the original road intersections.

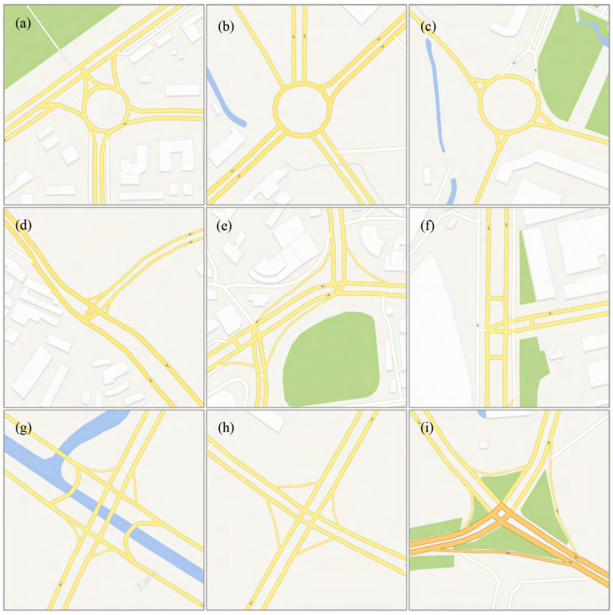

FIGURE 14. Original road data with complex intersections.

B. CASE STUDY OF ROADS WITH COMPLEX INTERSECTIONS

In this experiment, the original road data with complex intersections used to test the proposed SUCE method are from the Tiandi map in China, which can be downloaded for free on the Internet according to the Uniform Resource Locator (URL). The tile map is organized by a pyramid rule [45]. The road data on the tile map used to test the proposed SUCE method are at the 18th level. The size of each tile is 256 × 256 pixels. As shown in Figure 14, several test regions with complex road intersections, such as in a region containing interchanges or street gardens, were chosen to perform the experiment.

Before executing the SUCE method, the roads should first be extracted from the original tile map. Usually, the colors of the roads on the Tiandi map are uniformly yellow or white according to their grades. However, due to overlays of other geographical elements, such as text or arrows, it is still difficult to perfectly extract roads from the Tiandi map. The method used for extracting roads can be found in the study of Shen and Ai [45] and is effective for filling the holes or connecting the fractures caused by the overlays of other geographical elements. Since this study focuses on the centerline extraction from dual-line roads, further discussion about extracting roads from raster maps will not be provided. Figure 15 shows the road extraction results, which can be directly used for the proposed SUCE method.

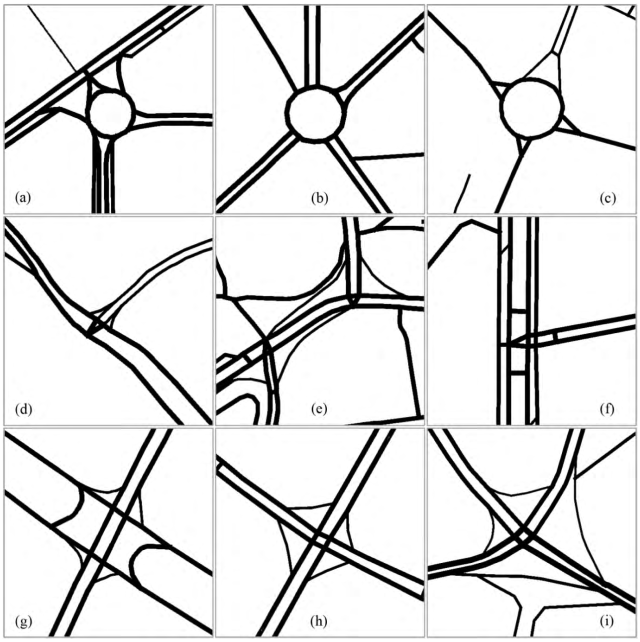

FIGURE 15. Road extraction from the original tile map.

Figure 16 shows the extraction results for the road centerlines using the proposed SUCE method. For the centerline extraction of road data with complex intersections, two important additional steps of the SUCE method should be emphasized. (1) Since the road intersections are too close in some complex cases, under-segmentation may occur. To solve this problem, the following rules should be applied. When the number of first merging superpixels is above a fixed threshold, such as 6, the superpixel after the first merging should be segmented by SLIC again. The number of superpixels used for segmentation for the second time is usually twice as many as the number of superpixels for the first time. For example, Figure 17a shows the original road, and the result of the first superpixel segmentation is shown in Figure 17b. 10 superpixels marked in red are detected to be merged, and the result of the first merging is shown

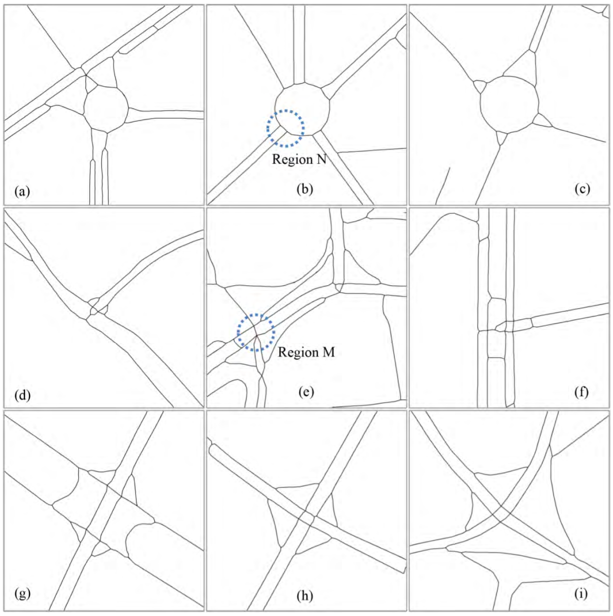

FIGURE 16. Extraction results for road centerlines using the proposed SUCE method.

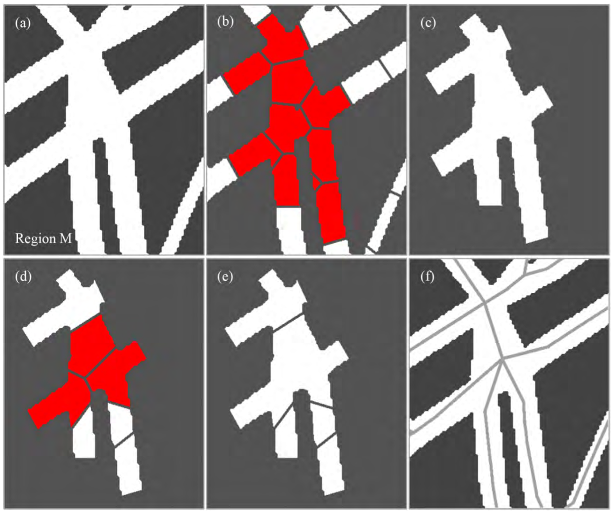

FIGURE 17. Under-segmentation of complex road intersections.

in Figure 17c. Figure 17d shows the result of the superpixel segmentation for the second time. Figure 17e shows the result of the second superpixel merging, and Figure 17f shows the final result of the road centerline extraction. (2) Another problem may appear when two intersections are too close. As shown in Figure 18a, since the distribution of superpixels is homogeneous, the superpixels marked in red are superpixels that are detected to be merged, but in this case, only one intersection can be generated. However, two intersections actually exist in the original roads. To maintain the correct number of road intersections, the following rules should be applied. The numbers in Figure 18a, such as 1 and 2, represent the number of corners of each superpixel located in the interior of the original roads. When a superpixel with

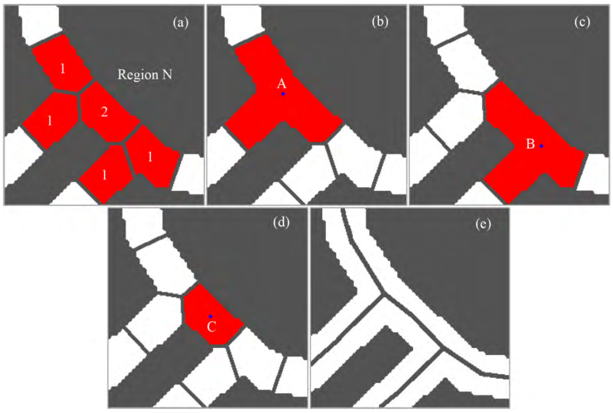

FIGURE 18. Remerging of superpixels.

2 corners located in the interior of the original roads connects four superpixels with 1 corner located in the interior of the original roads, this superpixel with 2 corners can be called a connecting superpixel. In this case, three center points should be calculated, which include two center points of the merged superpixel that consists of one connecting superpixel and two superpixels with 1 corner located in the interior of the original roads and one center point of the connecting superpixel, as shown in Figure 18b-d. Figure 18e shows the final result of the road centerline extraction in this case.

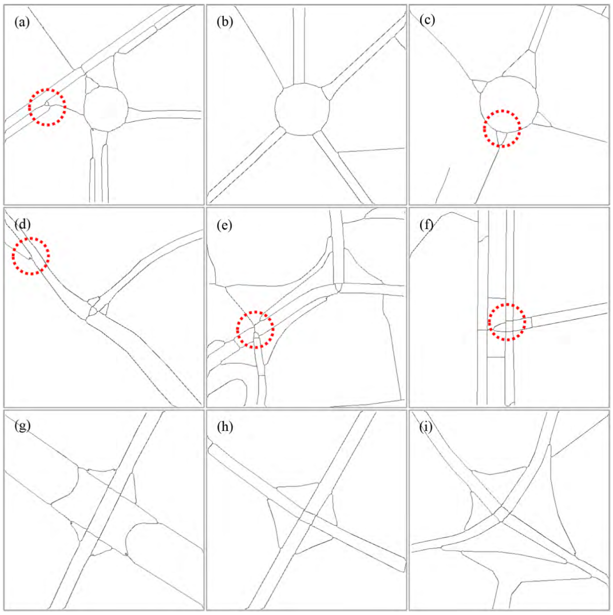

FIGURE 19. Extraction results for road centerlines using ZS.

To evaluate the ability of the SUCE method for extracting the centerlines of roads with complex intersections, four other image thinning algorithms, including the ZS, Rosenfeld [73], Pavlidis [74] and morphological [75] methods, were applied in contrast experiments. Figures 19-22 show the extraction results for the road centerlines using the ZS, Rosenfeld, Pavlidis and morphological methods, respectively.

TABLE 2. Evaluation in the case of complex road intersections.

Methods	ZS	Rosenfeld	Pavlidis	Morphological	SUCE
Number of redundant intersections	8	24	10	4	2
Number of broken intersections	0	0	31	79	0
Number of incorrect intersections	8	24	41	83	2
Number of burrs	10	0	20	0	0
Noise	No	No	No	Yes	No
Average execution time	53.2795 ms	93.1920 ms	68.4507 ms	23.4587 ms	5.7001 s

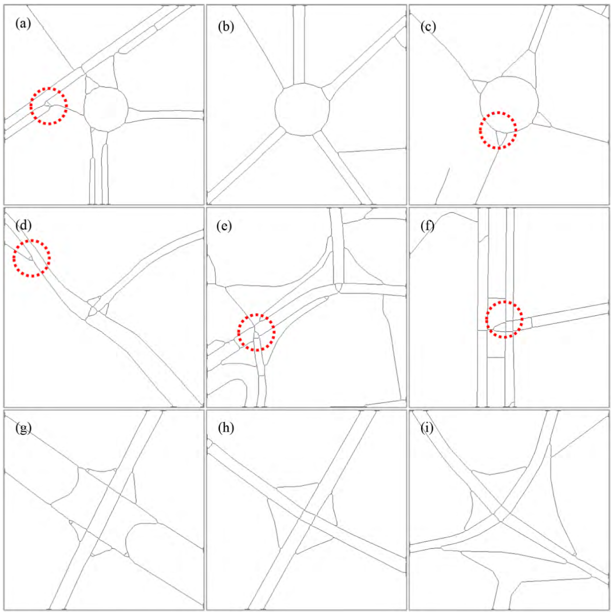

FIGURE 20. Extraction results for road centerlines using Rosenfeld method.

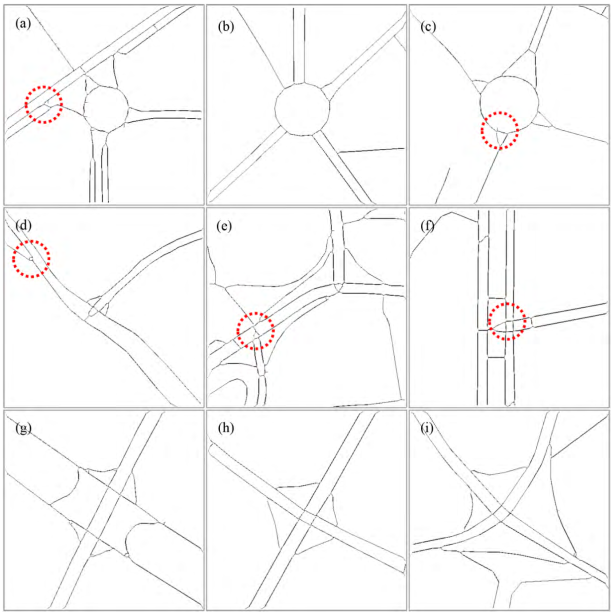

FIGURE 21. Extraction results for road centerlines using Pavlidis method.

To clearly display the differences in the capacity to handle complex road intersections between these methods, some typical regions marked in red circles in Figures 19-22 were chosen. Figure 23 shows the centerline extraction differences of complex road intersections between the ZS, Rosenfeld, Pavlidis, morphological and proposed SUCE methods, which reveal the following: in the case of complex road intersections, the ZS method tends to generate incorrect circular road intersections (such as in Figure b1), burrs (such as in Figure c1) and redundant road intersections (such as on the left side in Figure e1); the Rosenfeld method tends to generate incorrect circular road intersections (such as in Figures b2 and c2); the Pavlidis method tends to generate incorrect fractured road intersections (such as in Figures c3, d3 and e3) and burrs (such as in Figures a3 and b3); and the morphological method tends to generate incorrect fractured road intersections (such as in Figures b4 and e4) and noise (such as in Figures a4, c4 and d4). To summarized, the structure and topological relations of the original road network were destroyed by these four image thinning algorithms. However, as shown in Figures a5, b5, c5, d5 and e5, the proposed SUCE method can effectively maintain the correct road intersections and the topological relations of the original road. Thus, the structure of the original road network can be effectively preserved by the proposed SUCE method, even in the case of complex road intersections.

To evaluate the five methods for extracting the centerlines of roads with complex intersections, as shown in Table 2, the number of redundant intersections, number of broken intersections, number of incorrect intersections, number of burrs, noise, and average execution time were calculated. The Rosenfeld method produces the most redundant intersections, which reaches up to 24. The numbers of redundant intersections of the ZS and Pavlidis methods are close, which are 8 and 10, respectively. The SUCE method produces the fewest (only 2) redundant intersections. The morphological

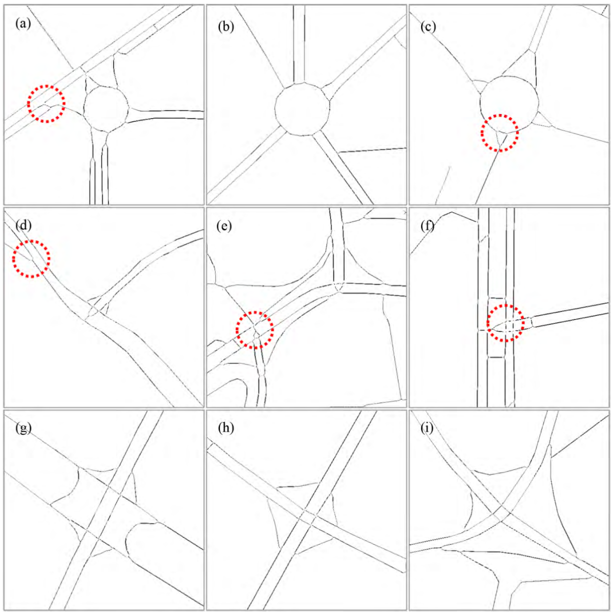

FIGURE 22. Extraction results for road centerlines using morphological method.

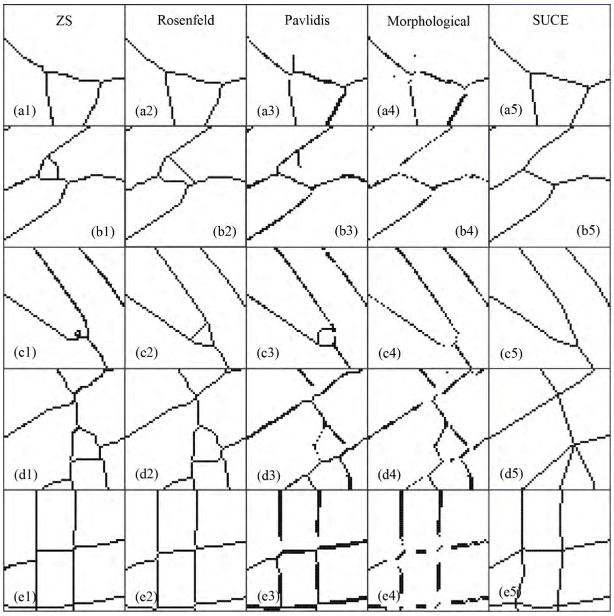

FIGURE 23. Differences of capacity for handling complex road intersections between five methods.

method produces the most broken intersections, which reaches up to 79. The Pavlidis method contains 31 broken intersections. The numbers of broken road intersections produced by the ZS, Rosenfeld and SUCE methods are zero. In total, the morphological method produces the most incorrect intersections, which reaches up to 83. The numbers of incorrect road intersections produced by the ZS, Rosenfeld and Pavlidis methods are 8, 24 and 41, respectively. However, the number of incorrect road intersections produced by the proposed SUCE method is only 2, which means that the SUCE method can effectively restore the original road intersections even in the case of complex road intersections. In addition, the ZS and Pavlidis methods tend to produce burrs, but the other three methods can effectively avoid burrs, and the morphological method is the only method that tends to produce noise. The average execution time of the morphological method performed the best of all five methods. Since the proposed SUCE method is a process of integrating multiple technologies including superpixel segmentation, corner detection and image moments, the average execution time is the longest.

V. CONCLUSIONS

This study proposes an innovative method of extracting centerlines from dual-line roads using superpixel segmentation based on raster data. In the traditional methods of road centerline extraction, vector data are used as the primary experimental data. However, our study focuses on centerline extraction using the SUCE method based on superpixel technology. The original roads are first segmented by a superpixel algorithm called SLIC. Then, the superpixels located at road intersections are merged. Finally, the road centerlines are generated by connecting the center points and edge midpoints of superpixels. Compared with the existing four thinning algorithms based on raster data and the traditional ArcGIS method based on vector data, the results show that the proposed SUCE method can effectively extract centerlines from dual-line roads and effectively restore the original road intersections, even in case of complicated road intersections. In addition, the proposed SUCE can effectively avoid burrs and noise. Since the parameter setting of the number of superpixels is very important, as neither too large nor too small is good, how to adaptively set more accurate parameters is a topic for future research.

REFERENCES

[1] A. H. J. Christensen, "Cartographic line generalization with waterlines and medial-axes," *Cartogr. Geogr. Inf. Sci.*, vol. 26, no. 1, p. 19–32, 1999.

[2] M. Heller, "Triangulation algorithm for adaptive terrain modeling," in *Proc. 4th Int. Symp. Spatial Data Handling*, Zurich, Switzerland, 1990, pp. 163–174.

[3] T. Ai, S. Ke, M. Yang, and J. Li, "Envelope generation and simplification of polylines using delaunay triangulation," *Int. J. Geograph. Inf. Sci.*, vol. 31, no. 2, pp. 297–319, 2017.

[4] X. Liu, F. B. Zhan, and T. Ai, "Road selection based on Voronoi diagrams and 'strokes' in map generalization," *Int. J. Appl. Earth Observ.*, vol. 12, pp. S194–S202, Sep. 2010.

[5] R. Chithambaram, K. Beard, and R. Barrera, "Skeletonizing polygons for map generalization," in *Proc. Tech. Papers, ACSM-ASPRS Conv., Cartogr. GIS/LIS*, 1991, p. 4.

[6] P. M. V. D. Poorten and C. B. Jones, "Characterisation and generalisation of cartographic lines using delaunay triangulation," *Int. J. Geograph. Inf. Sci.*, vol. 16, no. 8, pp. 773–794, 2002.

[7] M. McAllister and J. Snoeyink, "Medial axis generalization of river networks," *Cartogr. Geograph. Inf. Sci.*, vol. 27, no. 2, pp. 129–138, 2000.

[8] N. Regnauld and W. A. Mackaness, "Creating a hydrographic network from its cartographic representation: A case study using ordnance survey MasterMap data," *Int. J. Geograph. Inf. Sci.*, vol. 20, no. 6, pp. 611–631, 2006.

[9] A. A. DeLucia and R. T. Black, "A comprehensive approach to automatic feature generalization," in *Proc. 13th Conf. Int. Cartograph.*, 1987, pp. 168–191.

[10] Z. Li, *Algorithmic Foundation of Multi-Scale Spatial Representation*. Boca Raton, FL, USA: CRC Press, 2006.

[11] J. J. Zou and H. Yan, "Skeletonization of ribbon-like shapes based on regularity and singularity analyses," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 3, pp. 401–407, Jun. 2001.

[12] P. Morrison and J. J. Zou, "Triangle refinement in a constrained Delaunay triangulation skeleton," *Pattern Recognit.*, vol. 40, no. 10, pp. 2754–2765, 2007.

[13] F. Penninga, E. Verbree, W. Quak, and P. V. Oosterom, "Construction of the planar partition postal code map based on cadastral registration," *GeoInformatica*, vol. 9, no. 2, pp. 181–204, 2005.

[14] H. Uitermark, A. Vogels, and P. van Oosterom, "Semantic and geometric aspects of integrating road networks," in *Interoperating Geographic Information Systems*. Berlin, Germany: Springer, 1999, pp. 177–188.

[15] C. B. Jones, G. L. Bundy, and M. J. Ware, "Map generalization with a triangulated data structure," *Cartogr. Geograph. Inf. Syst.*, vol. 22, no. 4, pp. 317–331, 1995.

[16] P. Gao and M. M. Minami, "Raster-to-vector conversion: A trend line intersection approach to junction enhancement," in *Proc. 11th Int. Symp. Autocarto*, 1993, pp. 1–7.

[17] O. Aichholzer, F. Aurenhammer, D. Alberts, and B. Gärtner, "A novel type of skeleton for polygons," *J. Universal Comput. Sci.*, vol. 1, no. 2, pp. 752–761, 1996.

[18] G. K. Das, A. Mukhopadhyay, S. C. Nandy, S. Patil, and S. V. Rao, "Computing the straight skeleton of a monotonepolygon in $O(nlogn)$ time," in *Proc. 22nd Can. Conf. Comput. Geometry*, 2010, pp. 1–6.

[19] D. Eppstein and J. Erickson, "Raising roofs, crashing cycles, and playing pool: Applications of a data structure for finding pairwise interactions," *Discrete Comput. Geometry*, vol. 22, no. 4, pp. 569–592, 1999.

[20] J.-H. Haunert and M. Sester, "Using the straight skeleton for generalisation in a multiple representation environment," in *Proc. 8th ICA Workshop Gen. Multiple Represent.*, 2004, pp. 1–20.

[21] J.-H. Haunert and M. Sester, "Area collapse and road centerlines based on straight skeletons," *GeoInformatica*, vol. 12, no. 2, pp. 169–191, 2008.

[22] T. Ai, X. Zhang, Q. Zhou, and M. Yang, "A vector field model to handle the displacement of multiple conflicts in building generalization," *Int. J. Geograph. Inf. Sci.*, vol. 29, no. 8, pp. 1310–1331, 2015.

[23] D.-T. Lee, "Medial axis transformation of a planar shape," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-4, no. 4, pp. 363–369, Jul. 1982.

[24] U. Montanari, "A method for obtaining skeletons using a quasi-Euclidean distance," *J. ACM*, vol. 15, no. 4, pp. 600–624, 1968.

[25] L. Lam, S.-W. Lee, and C. Y. Suen, "Thinning methodologies—A comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 9, pp. 869–885, Sep. 1992.

[26] A. Jagna, "An efficient image independent thinning algorithm," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 3, no. 10, pp. 8309–8311, 2014.

[27] C. Louhou, "Contribution à l'analyse topologique des images: Étude d'algorithmes de squelettisation pour images 2D et 3D, selon une approche topologie digitale ou topologie discrète," Ph.D. dissertation, Dept. Comput. Eng., Univ. Marne-la-Vallée, Champs-sur-Marne, France, 2001.

[28] W. Deng, S. S. Iyengar, and N. E. Brener, "A fast parallel thinning algorithm for the binary image skeletonization," *Int. J. High Perform. Comput.*, vol. 14, no. 1, pp. 65–81, 2000.

[29] A. Rosenfeld and A. Kak, *Digital Picture Processing*. New York, NY, USA: Elsevier, 1976.

[30] P. Tarabek, "A robust parallel thinning algorithm for pattern recognition," in *Proc. 7th IEEE Int. Symp. Appl. Comput. Intell.*, May 2012, pp. 75–79.

[31] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[32] M. Ahmed and R. Ward, "A rotation invariant rule-based thinning algorithm for character recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1672–1678, Dec. 2002.

[33] L. B. Boudaoud, A. Sider, and A. Tari, "A new thinning algorithm for binary images," in *Proc. 3rd Int. Conf. Control, Eng. Inf.*, May 2015, pp. 1–6.

[34] A. Jagna and V. Kamakshiprasad, "New parallel binary image thinning algorithm," *ARPN J. Eng. Appl. Sci.*, vol. 5, no. 4, pp. 64–67, 2010.

[35] D. Kocharyan, "An efficient fingerprint image thinning algorithm," *Amer. J. Softw. Eng. Appl.*, vol. 2, no. 1, pp. 1–6, 2013.

[36] K. Palágyi, "A 3-subiteration 3D thinning algorithm for extracting medial surfaces," *Pattern Recognit. Lett.*, vol. 23, no. 6, pp. 663–675, 2002.

[37] C. Song, Z. Pang, X. Jing, and C. Xiao, "Distance field guided L_1-median skeleton extraction," *Vis. Comput.*, vol. 34, no. 2, pp. 243–255, 2016.

[38] Y. Y. Tang and X. You, "Skeletonization of ribbon-like shapes based on a new wavelet function," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1118–1133, Sep. 2003.

[39] R.-Y. Wu and W.-H. Tsai, "A new one-pass parallel thinning algorithm for binary images," *Pattern Recognit. Lett.*, vol. 13, no. 10, pp. 715–723, 1992.

[40] R. Zhou, C. Quek, and G. S. Ng, "A novel single-pass thinning algorithm and an effective set of performance criteria," *Pattern Recognit. Lett.*, vol. 16, no. 12, pp. 1267–1275, 1995.

[41] W. Chen, L. Sui, Z. Xu, and Y. Lang, "Improved Zhang-Suen thinning algorithm in binary line drawing applications," in *Proc. Int. Conf. Syst. Inform.*, May 2012, pp. 1947–1950.

[42] H. Lü and P. S. P. Wang, "A comment on 'a fast parallel algorithm for thinning digital patterns,'" *Commun. ACM*, vol. 29, no. 3, pp. 239–242, 1986.

[43] L. B. Boudaoud, B. Solaiman, and A. Tari, "A modified ZS thinning algorithm by a hybrid approach," *Vis. Comput.*, vol. 34, no. 5, pp. 689–706, 2018.

[44] F. Thomas, "Generating street center-lines from inaccurate vector city maps," *Cartogr. Geograph. Inf. Syst.*, vol. 25, no. 4, pp. 221–230, 1998.

[45] Y. Shen and T. Ai, "A hierarchical approach for measuring the consistency of water areas between multiple representations of tile maps with different scales," *ISPRS Int. J. Geo-Inf.*, vol. 6, no. 8, p. 240, 2017.

[46] Y. Shen, T. Ai, and Y. He, "A new approach to line simplification based on image processing: A case study of water area boundaries," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 2, p. 41, 2018.

[47] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Nice, France, Oct. 2003, pp. 1–10.

[48] Y. Shen, T. Ai, L. Wang, and J. Zhou, "A new approach to simplifying polygonal and linear features using superpixel segmentation," *Int. J. Geograph. Inf. Sci.*, vol. 32, no. 10, pp. 2023–2054, 2018.

[49] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal., Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[50] C. L. Zitnick and S. B. Kang, "Stereo for image-based rendering using image over-segmentation," *Int. J. Comput. Vis.*, vol. 75, no. 1, pp. 49–65, 2007.

[51] G. Mori, "Guiding model search using segmentation," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, Beijing, China, Oct. 2005, pp. 17–21.

[52] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *Proc. 12th IEEE Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 670–677.

[53] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," in *Proc. ACM SIGGRAPH Papers (SIGGRAPH)*, Los Angeles, CA, USA, Aug. 2004, pp. 8–12.

[54] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *Int. J. Comput. Vis.*, vol. 80, pp. 300–316, Dec. 2000.

[55] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes, "Layered object detection for multi-class segmentation," in *Proc. 9th IEEE Int. Conf. Vis. Pattern Recognit.*, Jun. 2010, pp. 3113–3120.

[56] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[57] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[58] A. Vedaldi and A. Soatto, "Quick shift and kernel methods for mode seeking," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 705–718.

[59] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: Fast superpixels using geometric flows," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 12, pp. 2290–2297, Dec. 2009.

[60] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *Proc. 11th Eur. Conf. Comput.*, 2010, pp. 211–224.

[61] G. S. K. Fung, N. H. C. Yung, and G. K. H. Pang, "Vehicle shape approximation from motion for visual traffic surveillance," in *Proc. IEEE 4th Int. Transp. Syst.*, Aug. 2001, pp. 608–613.

[62] B. Serra and M. Berthod, "3-D model localization using high-resolution reconstruction of monocular image sequences," *IEEE Trans. Image Process.*, vol. 6, no. 1, pp. 175–188, Jan. 1997.

[63] G. S. Manku, P. Jain, A. Aggarwal, L. Kumar, and S. Banerjee, "Object tracking using affine structure for point correspondence," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 704–709.

[64] L. Kitchen and A. Rosenfeld, "Gray level corner detection," *Pattern Recognit. Lett.*, vol. 1, no. 2, pp. 95–102, 1982.

[65] S. M. Smith and J. M. Brady, "SUSAN—A new approach to low-level image processing," *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 45–78,1997.

[66] A. Rosenfeld and E. Johnston, "Angle detection on digital curves," *IEEE Trans. Comput.*, vol. C-22, no. 9, pp. 875–878, Sep. 1973.

[67] H.-C. Liu and M. D. Srinath, "Corner detection from chain-code," *Pattern Recognit.*, vol. 23, nos. 1–2, pp. 51–68, 1990.

[68] F. Chabat, G.-Z. Yang, and D. M. Hansell, "A corner orientation detector," *Image Vis. Comput.*, vol. 17, no. 10, pp. 761–769, 1999.

[69] X.-C. He and N. H. C. Yung, "Curvature scale space corner detector with adaptive threshold and dynamic region of support," in *Proc. 17th Int. Conf. Pattern Recognit.*, Aug. 2004, pp. 791–794.

[70] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*. Reading, MA, USA: Addison-Wesley, 1992, pp. 170–171.

[71] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inf. Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962.

[72] M. F. Goodchild and G. J. Hunter, "A simple positional accuracy measure for linear features," *Int. J. Geograph. Inf. Sci.*, vol. 11, no. 3, pp. 299–306, 1997.

[73] A. Rosenfeld and A. Kak, *Digital Picture Processing*. New York, NY, USA: Academic, 1982.

[74] T. Pavlidis, *Algorithms for Graphics and Image Processing*. Rockville, MD, USA: Computer Science Press, 1982.

[75] Félix Abecassis. *OpenCV—Morphological Skeleton*. Accessed: Sep. 20, 2011. [Online]. Available: http://felix.abecassis.me/2011/09/opencv-morphological-skeleton/

YILANG SHEN was born in Xianning, Hubei, China, in 1991. He received the B.S. degree in geographical information systems from Wuhan University, in 2013, where he is currently pursuing the Ph.D. degree with the School of Resource and Environmental Sciences. His research interests include computer vision, map generalization, and the change detection of spatial data.

TINGHUA AI was born in Yidu, Hubei, China, in 1969. He received the B.S. and M.S. degrees in cartography and the Ph.D. degree in cartography from the Wuhan University of Surveying and Mapping Science and Technology, Wuhan, in 1994 and 2000, respectively. He is currently a Professor with the School of Resource and Environmental Sciences, Wuhan University. His research interests include the multi-scale representation of spatial data, map generalization, spatial cognition, and spatial big data analysis.

MIN YANG was born in Hubei, China, in 1985. He received the B.S. and Ph.D. degrees in cartography from Wuhan University, Wuhan, in 2007 and 2013, respectively, where he is currently an Associate Professor with the School of Resource and Environmental Sciences. His research interests include the change detection of spatial data, map generalization, and spatial big data analysis.