Use of Satellite Imagery for Establishing Road Horizontal Alignments

Said M. Easa, M.ASCE[1]; Haibin Dong[2]; and Jonathan Li[3]

Abstract: Generating fast and inexpensive digital road maps and databases from high-resolution satellite imagery is becoming possible for various applications. This paper presents a new method for establishing road horizontal alignment using IKONOS 1 m spatial resolution imagery. Road extraction algorithms were developed for two types of horizontal curves: Simple circular curves and reverse circular curves. The method requires only two and three unknown parameters for simple and reverse curves, respectively. Unlike existing methods of circle detection, the proposed method performs the search procedures in a much smaller area than the image size and achieves faster computations. The derived curve parameters represent useful inputs into a geographic information system database. The developed method has been tested using IKONOS images for simple and reverse curves. The results show that the proposed method converges in all cases and can be used for accurately establishing road horizontal curves.

DOI: 10.1061/(ASCE)0733-9453(2007)133:1(29)

CE Database subject headings: Satellites; Imaging techniques; Roads; Curvature; Alignment.

Introduction

There is an increasing need for efficient data acquisition and updating for geographic information systems in transportation (GIS-T). However, the acquisition of road information has been based on conventional mapping techniques, mostly using aerial photographs that heavily depend on weather and photographic scales. Aerial photographs are time-consuming and expensive to obtain on a regular basis. This is particularly true for citywide road database updating and change detection in areas with rapid development. In contrast, the new generation of commercial high-resolution satellite imagery, such as IKONOS 1 m spatial resolution imagery, provides a means by which large areas can be mapped with high accuracy and more economically. Although both data availability and user demands are increasing, the lack of fast and efficient tools impedes the practical applications of satellite images in different areas, including road design and road safety. Therefore, new challenges exist in the area of image analysis for automatic extraction of information from satellite images.

Road horizontal curves represent an important element of surveying engineering (Anderson and Mikhail 1998; Kavanagh and Bird 2000; Wolf and Brinker 2000). The extraction of the road geometric elements from IKONOS imagery has drawn considerable attention recently. The approaches fall into two broad categories: Automatic approach and semiautomatic approach. A typical automatic system for road extraction consists of road finding, road following/tracking/tracing, and road linking. An example of the automatic approach that uses a texture-based classification method can be found in Dial et al. (2001).

The semiautomatic approach requires the interaction between the algorithm and an operator. An operator interactively provides some information to control the extraction. An example of the semiautomatic approach can be found in Zhao et al. (2002), in which the image is first classified using commercial remote sensing software to generate a road mask image that excludes nonroad pixels. Edges are then extracted and traced from straight lines. Long lines with slow direction change are taken as road seeds. After a starting point is supplied by an operator, the next road point is determined by matching a template with the road mask image and road seeds. If the matching is poor, the operator is required to assign a control point, and road extraction is repeated by taking the control point as a new starting point.

In high-resolution satellite imagery, there are two basic types of road curves. The first is an intersection turning roadway (e.g., a right-turn at a signalized intersection), that is only visible on high-resolution satellite images such as 1 m resolution IKONOS or 60 cm spatial resolution QuickBird. In low-to-medium resolution satellite images, such as 30 m resolution Landsat TM or 10 m resolution SPOT, this type of curve may not be discernable since the corner is almost similar to a right angle. That is, the radius is too small in pixel size. The other type is a horizontal curve at road mainlines or freeway interchange ramps that would appear in all-resolution images since road alignment is generally discernable. This study focuses on the second type of road horizontal curves. The geometric parameters of such curves can be identified from a georeferenced or rectified satellite imagery. Once the parameters of the curves are established, the road network can be established simply by connecting the curves because only straight lines exist between the curves. These parameters are also useful as reference data for traffic collisions and future road improvements.

The correctness of curve extraction depends on the results of the linear feature extraction. The standard Hough transform (HT)

---

[1] Professor, Dept. of Civil Engineering, Ryerson Univ., Toronto ON, Canada M5B 2K3.

[2] Formerly, Graduate Student, Dept. of Civil Engineering, Ryerson Univ., Toronto ON, Canada M5B 2K3.

[3] Associate Professor, Dept. of Civil Engineering, Ryerson Univ., Toronto ON, Canada M5B 2K3.

Note. Discussion open until July 1, 2007. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on September 9, 2002; approved on February 22, 2005. This paper is part of the ***Journal of Surveying Engineering***, Vol. 133, No. 1, February 1, 2007. ©ASCE, ISSN 0733-9453/2007/1-29–35/$25.00.

J. Surv. Eng., 2007, 133(1): 29-35

is intended for arbitrary images, while the probabilistic HT is designed for fast computations (Trucco and Verri 1998). However, the probabilistic HT has some limitations, including the possibility of producing inconsistent results. The HT has been extended to detect circle features in imagery (Kumar et al. 1994; Guil and Zapata 1997). Most existing algorithms, however, assume that the center of the circle is located within the image size. Thus, the search area and the range of the radius are limited. These methods may not be feasible for establishing road horizontal alignments since they may involve curves with very large radii resulting in centers outside the image size. In addition, due to the larger number of unknown parameters (three for simple curves and five for reverse curves), circle detection involves complex computations and large memory. This makes such methods impractical for large-size imagery. For these reasons, a new algorithm was developed in this study to achieve faster and more efficient computations. The algorithm involves only two and three unknown parameters for simple and reverse arcs, respectively, and a search area that is much smaller than the image size.

The following sections first describe the image preprocessing which is needed for road extraction and the proposed method of extracting simple and reverse horizontal curves. Sample applications using IKONOS imagery are then presented, followed by concluding remarks.

Image Preprocessing

The proposed road extraction algorithm requires some image preprocessing to convert a colored image to an edge (binary) image. The preprocessing involves first converting the colored image to a gray image as required by the Canny edge detector method implemented in this study (Canny 1986). Then, the Canny method is used to generate an edge image from the gray image. The Canny method identifies edges by locating the changes in the intensity (brightness) function; edges are associated with pixels where this function changes abruptly. An edge, a local property of an individual pixel, is calculated from the image function behavior in a neighborhood of that pixel. It is useful to describe briefly the following basic steps of the Canny method (Intel 2001; Sonka 1999):

Step 1: *Smoothing*. The image is preprocessed by a Gaussian filter to eliminate the noise (e.g., small dots) or other small fluctuations in the image and to smooth the coarse appearance of the image features.

Step 2: *Differentiation*. The purpose of this step is to identify the edges between two homogenous areas with different intensity. The smoothed image is differentiated with respect to the x- and y-directions of the image; derivatives are greater where the intensity function undergoes rapid changes. From the computed gradient values x and y, the magnitude and angle of the gradient are calculated.

Step 3: *Nonmaximal Suppression*. After the gradients are calculated, the edges can be located at the pixels with local maximal gradients. Those pixels with local nonmaximal gradients are excluded. Thus, edge pixels that mark areas of different brightness are strengthened and nonedge pixels with homogenous brightness are dimmed.

Step 4: *Thresholding*. The Canny edge detector uses double intensity thresholds (lower and upper limits) to decide which edges are significant. That is, if the pixel intensity value is above the upper limit, it is accepted as the edge pixel. If the pixel intensity value is below the lower limit, it is rejected. A pixel with values between the upper and lower limits is considered to be edge if it is connected to any of the pixels that exhibit strong responses.

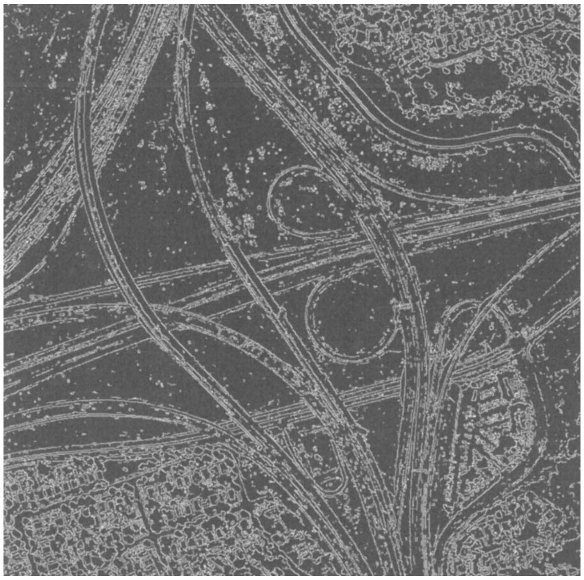

Fig. 1. Example of an edge (binary) image generated by the Canny method

Fig. 1 shows an example of an edge image that has been generated from a gray image. The edge pixels are shown in white and correspond to the sharp brightness variation in the gray image. The dark areas represent nonedge pixels. Only major edges appear in the image since only the edges for large-scale features (roads) are needed. Although the edges are not ideally continuous and are separated from other features at some locations, they are still good enough for further line and curve detection.

Proposed Road Extraction Method

The extraction of road horizontal curves and associated tangents from the edge image involves the use of the Hough transform. It is useful to describe first this transform for both straight lines and circles.

Hough Transform

The Hough transform, a popular algorithm for detecting features from raster images, was used to detect the tangents (straight lines) and the corresponding horizontal curves. For manmade objects in satellite imagery (e.g., roads), the shapes of the objects are generally simple, regular, and primitive, which can be accurately described mathematically. For horizontal alignments, where the information may be stored in a GIS database, it is necessary that the extracted curve parameters be consistent. Since the probabilistic HT may produce inconsistent results, the standard HT was used in this study. The standard HT for detecting straight lines can be represented by (Trucco and Verri 1998)

$$\rho = x \cos \theta + y \sin \theta \tag{1}$$

where ρ=distance from the origin to the line to be detected and θ=angle between the x-axis and a line passing the origin perpendicular to the line to be detected (Fig. 2).

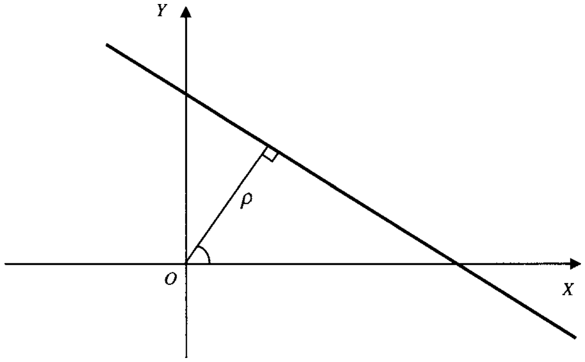

Fig. 2. Geometric properties of a straight line

In an edge image every pixel, whose location is (x,y) and value is not 0, may belong to many lines (with parameters ρ and θ) passing through this pixel. A counter is incremented by 1 for every line passing through this pixel. Thus, after scanning all image pixels, the accumulator contains the number of pixels every line has. Any line having pixels greater than a given threshold is a candidate line in this image. Although a line in an image may not be a perfect line (somewhere broken, noisy, or distorted), such imperfections can be filtered out statistically. A line obvious to the human eyes should have a large quantity of associated pixel members. The accumulator for saving the counting is a two-dimensional (2D) array, $A(\rho,\theta)$, that is initialized to zero.

For circles, the equation of the desired curve is given by

$$(x-x_o)^2+(y-y_o)^2=R^2 \tag{2}$$

where x_o, y_o=coordinates of the center of the circle and R=radius of the circle (Fig. 3). In an edge image, every pixel, whose location is (x,y) and value is not 0, may belong to many circles whose perimeter passes through it. Similar to line detection, the proposed method first constructs an accumulator initialized with zero, then scans all pixels to find the candidate circles. This time, there are three unknown parameters (x_o, y_o, and R) and, therefore, a three-dimensional (3D) accumulator array is used. An arc is only part of a circle in the image. Since no perfect circle exists, many separate arcs may belong to one circle. Thus, arc detection and circle detection are similar.

Extracting Simple Curves

A simple curve is a circular curve connecting two tangents that intersect at the point of intersection (PI). The beginning of the curve is the point of curvature (PC) and the end of the curve is the point of tangency (PT). In practice, finding specific tangents (lines) and curve (arc) connecting them is more localized. That is, not all lines and arcs in an image are needed. There are many of them and the difficulty is deciding the approximate threshold for the image. Even if a road is short and its length is below the threshold, it still could be a candidate for investigation. Therefore, automatic tangent and curve detection is not feasible in this study, and instead a semiautomatic method is used.

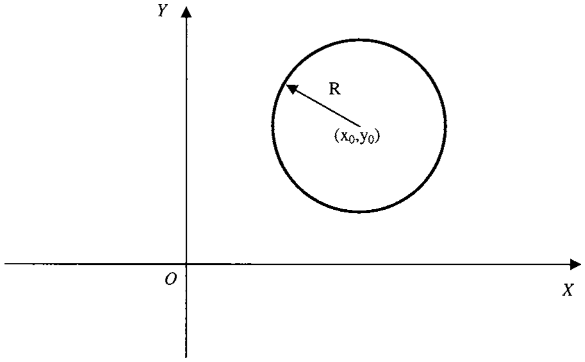

Fig. 3. Geometric properties of a circle

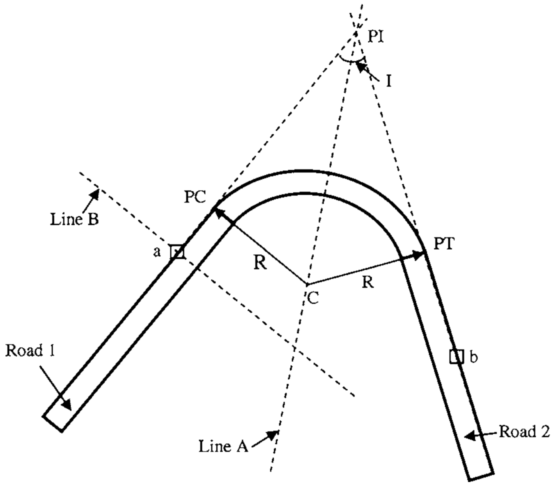

Fig. 4. Schematic representation for establishing a simple horizontal curve

From a road safety perspective, a horizontal curve must have a minimum radius that ensures vehicle stability. The minimum radius depends on road design speed, superelevation (road side slope), and maximum side friction factor. Based on the law of mechanics, the radius of a circular curve is given by (Easa 2002)

$$R=V^2/127(e+f) \tag{3}$$

where R=minimum radius (m); V=vehicle speed (km/h); e=superelevation (m/m); and f=side friction factor. To determine the minimum radius of a horizontal curve, the variables used in (3) would be the design speed, maximum superelevation, and maximum side friction factor for V, e, and f, respectively.

Tangent Detection

To detect an arc, the user clicks on the tangents of the curve, as shown by the two squares (search windows) in Fig. 4. Each square represents the area where the search for the best line is performed. In this study, the search window is taken as 9 × 9 pixels. To increase the accuracy of the search results and satisfy computation and computer storage requirements, a self-adaptive threshold is introduced to find a candidate tangent line. Two filters are applied to the search: (1) the HT line must pass through the selected search window; and (2) the calculated direction of the HT line must be the same as (or nearest to) the direction of the edge in the search window. Thus, a tangent line that resides in the search window can be found.

Curve Detection

Once the two tangents are established, the curve connecting them could be easily found. Its center must reside on the centerline of the intersection angle I (*Line A*), and its radius R must be the

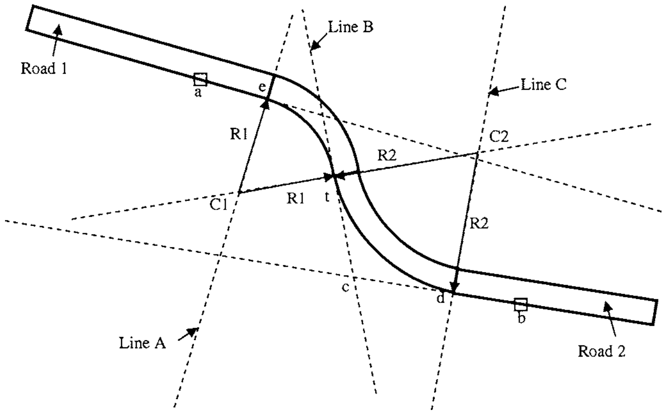

Fig. 5. Schematic representation for establishing a reverse horizontal curve

distance from the curve center to either of the two tangent lines. Now only the center Point *C* is an independent unknown parameter, and *C* lies on Line *A*. The amount of computation and memory also could be greatly reduced. The search is performed for only the circles whose centers are on Line *A*. To further reduce the problem to a one-dimensional (1D) array, it is required that the user clicks an arbitrary point on each tangent of the curve, not on the curve itself [Figs. 4(a and b)]. The point with the smaller distance from PI [Fig. 4(a)] is then marked as the initial point for the algorithm procedure. Then, the software automatically constructs an assistant line perpendicular to the road line on which this initial point lies (Line B). The intersection point of Lines A and *B*, $C_{\max}$, is the maximum possible center for the curve being searched.

To find the candidate center, a 1D accumulator to record the rank of each possible curve is allocated. Each possible curve is the one whose center is between PI and $C_{\max}$, and whose radius is the distance between its center and either of the two tangents. Only one independent unknown is involved because the center lies on Line *A*. A statistical method like HT can tolerate imperfection of the image and determine the most likely "actual" arc. All pixels that each curve passes are scanned (if not zero) and the counter value for this curve is incremented by one. The larger the radius, the more pixels would be scanned. To ensure that the best curve is properly selected, the density (rather than the number) of edge pixels along the curve is used as a measure of fitness. The density is calculated by dividing the counter value for each curve by its length. The curve that has the maximum relative hit count is selected. This results in only one curve that is the best curve connecting the tangents. Once the best curve is determined, all corresponding parameters are recorded, including its radius, beginning point, and end point.

Extracting Reverse Curves

A reverse horizontal curve consists of two consecutive circular arcs in opposite directions with a common tangent point. These curves cannot be established using the algorithm developed for simple curves because this algorithm requires two straight lines for extracting the simple curve. To establish reverse curves, another algorithm was developed. The procedure for establishing reverse horizontal curves is illustrated in Fig. 5. If the road is wide, two reverse horizontal curves will be shown in the image, one for each side of the road. This is the case in high-resolution imagery where a road is shown as a band rather than a line. The objective is to find the starting tangent point, ending tangent point, and common tangent point along one road edge of the reverse curve as well as the radii and centers of the two arcs.

The algorithm involves the following steps to establish the reverse curve:

1. Find two straight lines representing the start and end tangents of the reverse curve.
2. Select the first initial point, Fig. 5(a). To reduce the amount of computations, it is recommended that the initial point be located on the side with the curve that has a smaller radius (first arc). As shown in Figs. 5(a and b), two initial points are marked on the two tangents of the reverse curve.
3. Select every point on the first straight line [from Fig. 5(a) toward the first arc] to be the candidate of the starting tangent point. Make a line perpendicular to the starting line and through the candidate tangent point, such as Line *A*, on which the center of the candidate first arc, C1, must be located.
4. Try all possible radii and all possible common tangent points on the circumference. Using every possible common tangent point (such as *t*), make a common tangent shared by both arcs, such as Line *B*, which intersects with the end tangent at *c*. The distance *ct* must equal the distance *cd*. Thus, the position of the candidate end tangent point *d* can be located.
5. Draw a line perpendicular to the end tangent at *d*, such as Line *C*. The center of the second arc must be on Line *C*, and also on the line perpendicular to the common tangent *B*. These two lines generate an intersection point, which is the center of the second arc, *C*2.
6. Calculate the percentage of match in the image of the arc under consideration.

There are many possibilities of such combinations. By examining all candidates, and comparing their percentages of match in the image, the combination with the maximum possibility is selected as the final reverse curve. The logical flow of the above procedure, which involves a 3D loop (distance over tangent, radius on Line *A*, and central angle), is shown in Fig. 6. The algorithm is applicable to all practical orientations of the first and second tangents of the reverse curve, including parallel tangents. Note that the algorithm requires specifying reasonable starting values of the radius and central angle of the first arc, C_o and θ_o. In addition, The final central angle, final radius of the first arc, and the final point on the first tangent are also set at reasonable practical limits.

The presented method was coded into a computer software using C^{++}. The developed software was based on the computer-vision source code available from Intel, known as OPENCV (Intel 2001). The software implements all the aspects required for road extraction presented in this paper: image preprocessing to develop an edge image from a gray image (using the Canny method), the standard Hough transform for lines and circles, and the proposed method of extracting simple and reverse curves.

Applications

To test the proposed algorithm, an experiment was conducted to extract two types of horizontal curves from IKONOS imagery. One standard frame IKONOS imagery, acquired in August 2001 by the Space Imagine's IKONO-2 imaging satellite launched in 1999, was provided by the Greater Toronto Airports Authority.

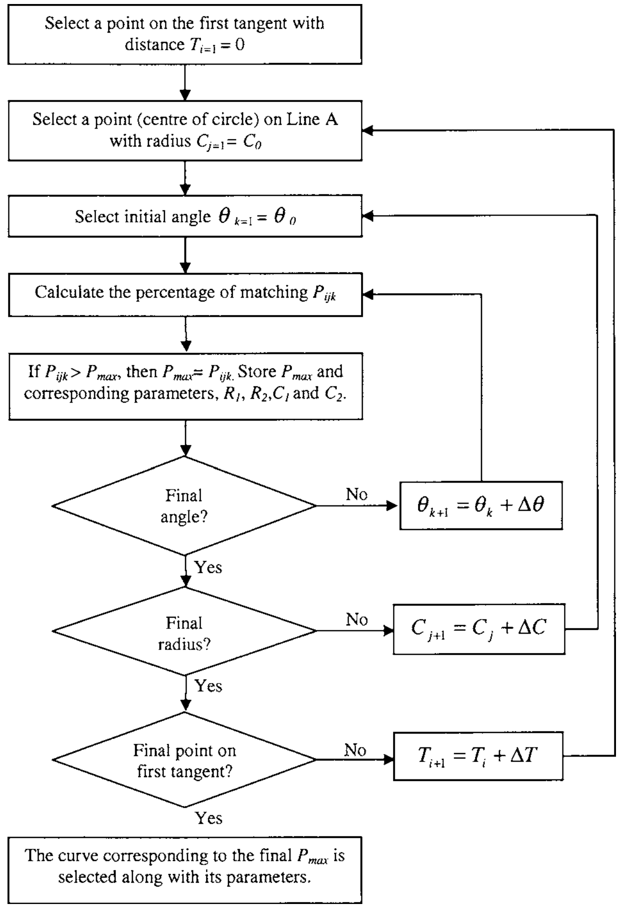

Fig. 6. Flow diagram for establishing a reverse horizontal curve

The IKONOS imagery was geometrically corrected in two dimensions (X and Y) in the Universal Tranverse Mercator (UTM) projection coordinate system, zone 17° in the Geodetic Reference System of 1980 (GRS80) and North American Horizontal Datum of 1983 (MAD83). Two subsection of this IKONOS image, one showing a freeway intersection and the other showing a residential area, were used. The two types of horizontal curves were tested in this study: Simple curve and reverse curve.

The image for the simple horizontal curve (Application 1) is a freeway interchange that has many simple curves (Fig. 7). As previously mentioned, before applying the proposed algorithm, the color image is converted to a gray image which is used by the Canny method to derive the edges (roadsides). The gray image corresponding was shown previously as Fig. 1. The curves derived from the gray image are shown in the color image of Fig. 7. As noted, all interchange curves were accurately identified using the proposed method. The identification was done sequentially (one curve at a time). The parameters identified for each curve included the center, radius, start angle, and end angle, which are automatically calculated by the computer software. The radii of the simple curves are shown in Table 1.

The image for the reverse curve (Application 2) is a Space Imaging's IKONOS satellite image (Fig. 8). As noted, the algorithm correctly identified the two arcs of the reverse curve. The established radii of arcs A and B are 104 and 51 m, respectively. Note that the first and second tangents of the reverse curve are parallel. Similar to simple curves, the algorithm can extract the parameters of the curve from the image, including the coordinates of the centers, radii, start point, end point, and common point.

Fig. 7. Results of establishing a simple horizontal curve at a freeway interchange

Table 1. Established Curve Radii by the Algorithm

Curve name	Radius (m)
A	501
B	393
C	455
D	432
E	273
F	152
G	87
H	52
I	102
J	69
K	402
L	22

Concluding Remarks

This paper has presented a new method for establishing road horizontal curves from satellite imagery. Extraction algorithms were developed for two types of horizontal curves: Simple curves and reverse curves. The established geometric parameters of the curves can enrich GIS databases with a more detailed structure. Based on this study, the following comments are offered:

1. The application examples showed that the developed method can accurately establish simple and reverse curves, even for a complex freeway interchange. The algorithms were found to converge in all cases studied. The algorithm presented for reverse curves can be easily adapted for compound horizontal curves that involve two or more curves. The same principles of tangent and curve detection can be applied with some modifications.
2. The existing method of circle detection involves complex computation and memory consumption (involving three and five unknown parameters for simple and reverse curves, respectively), which makes it impractical for large-size imagery. The new algorithm developed in this paper involves only two and three parameters for simple and reverse arcs, respectively. In addition, the search areas are much smaller than the image size in most cases. Thus, the proposed method would achieve faster and more efficient computation.
3. This paper has focused on establishing horizontal curves for only one side of the road (inside or outside edge). Clearly, the presented method can be applied twice to establish the curve(s) of each edge. In this case, however, there is no guarantee that the outside-edge curve will have the same center as the inside-edge curve, as it should. Therefore, further research is required to extend the presented method to extract one edge curve based on the information obtained for the other edge curve, and the known width of the road.
4. This study has addressed only simple and reverse curves without spirals. Spiral curves are normally used on high-type facilities to provide a smooth transition from the tangent to the curve (and vice versa), among other functions. A spiral curve has a radius of infinity at the tangent that gradually decreases to the radius of the connecting circular curve. The extraction of spiraled horizontal curves is presented in a companion paper (Dong et al. 2007).

Acknowledgments

The writers wish to thank Anthony Sani of Spatial Geo-link in Toronto for his support with the latest version of ERDAS Imagine. The IKONOS imagery was provided by the Greater Toronto Airports Authority. This research was supported by a discovery

Fig. 8. Results of establishing a reverse horizontal curve in a residential area

grant from the Natural Sciences and Engineering Research Council of Canada.

References

Anderson, J. M., and Mikhail, E. M. (1998). *Surveying: Theory and practice*, McGraw-Hill, New York.

Canny, J. (1986). "A computational approach to edge detection." *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6), 679–698.

Dial, G., Gibson, L., and Poulsen, R. (2001). "IKONOS satellite imagery and its use in automated road extraction." *Automatic extraction of manmade objects from aerial and space images (III)*, E. P. Baltsavias, A. Grün, and L. V. Gool, eds., Balkema, Lisse, 357–367.

Dong, H., Easa, S. M., and Li, J. (2007). "Approximate extraction of spiralled horizontal curves from satellite imagery." *J. Surv. Eng.*, 133(1), 36–40.

Easa, S. M. (2002). "Geometric design." *Civil engineering handbook*, W. F. Chen, and J. Y. Liew, eds., CRC, Boca Raton, Fla., Chap. 63.

Guil, N., and Zapata, E. (1997). "Lower order circle and ellipse Hough transform." *Pattern Recogn.*, 30(10), 1729–1744.

Intel reference manual. (2001). "Open source computer vision library." ⟨www.intel.com⟩.

Kavanagh, B. F., and Bird, S. J. (2000). *Surveying—Principles and applications*, Prentice-Hall, Columbus, Ohio.

Kumar, S., Ranganathan, N., and Goldgof, D. (1994). "Parallel algorithms for circle detection in images." *Pattern Recogn.*, 27(7), 1019–1028.

Sonka, M., Hlavac, V., and Boyle, R. (1999). *Image processing, analysis, and machine vision*, Brooks/Cole, New York.

Trucco, E., and Verri, A. (1998). *Introductory techniques for 3D computer vision*, Prentice-Hall, New York.

Wolf, P. R., and Brinker, R. C. (2000). *Elementary surveying*, Harper Collins College Publishers, New York.

Zhao, H., Kumagai, J., Nakagawa, M., and Shibasaki, R. (2002). "Semi-automatic road extraction from high-resolution satellite image."*Int. Archives of photogrammetry and Remote Sensing*, 34(Part 3B), 406–411.

J. Surv. Eng., 2007, 133(1): 29-35