The detection of critical points is an important issue in many disciplines such as computer vision, image processing, pattern recognition, computer graphics and cartography/GIS. Numerous algorithms have been developed since the late 1960s. These algorithms can be classified into three major groups, i.e. corner detection, polygonal approximation, and a hybrid technique which is a combination for the first two. This paper aims to provide a review, and to examine the advantages and disadvantages, of these various algorithms in all of these three categories.

An Examination of Algorithms for the Detection of Critical Points on Digital Cartographic Lines

Zhilin Li

School of Surveying and Land Information, Curtin University of Technology, Perth, West Australia

INTRODUCTION

It has been discovered by Attneave (1954) that some points on an object are richer in information than others and these few points are sufficient to characterise the shape of the object. These points are referred to as "dominant points" in the computer graphics and pattern recognition literature and as "critical points" in the cartography/GIS literature.

Critical points of a curve in classical geometry are its maxima, minima, and points of inflection. To this list, Freeman (1978) also added the following: discontinuities in curvature, end points, intersections (junctions), and points of tangency; and these points should be so well-defined that their characteristics will not be affected by scaling, rotation and translation.

Critical points are an important concept in many disciplines such as computer vision, image processing, pattern recognition, computer graphics and cartography/GIS. For example, in computer vision and pattern recognition, the concept of critical points is used in algorithms for feature extraction, shape recognition, point-based motion estimation, and coding. In cartography/GIS, it is used for data compression, for line caricature and misleadingly for line generalisation, as pointed out by Li (1993).

Algorithms for detecting critical points can, in general, be classified into three major groups:

(a) corner detection,
(b) polygonal approximation, and
(c) a hybrid technique, which is a combination for the first two.

It is notable that almost all algorithms in the cartographic literature belong to the category of polygon approximation, except for a few (Thapa, 1988b). A review of those algorithms developed before the mid-1980s has been provided by McMaster (1987) and Thapa (1988a). Here, no attempt is made to repeat such work. Instead, this paper will provide a broader overview and concentrate on more recent developments although, from time to time, some older literature will be cited for clarity. The advantages and disadvantages of these various algorithms will be discussed and one typical example for each group and/or sub-group will also be illustrated, when appropriate.

THE POLYGONAL APPROXIMATION APPROACH

In the polygonal approximation approach, the two commonly used techniques are:

(a) sequential methods; and
(b) iterative methods.

Most algorithms for the detection of critical points appearing in the cartography/GIS literature belong to this category except for a few (e.g. Thapa, 1988b).

Sequential methods

The sequential methods start from a point to find the longest allowable segment, within which all points along the original line can be neglected with a given criterion or tolerance or threshold. The last point on the longest allowable segment is considered as being a critical point and is designated as the next starting point. This process is repeated until the whole curve has been traced. At the end, all these detected points, plus the two end points, are considered as being the critical points of this curved line.

The typical example, which is familiar to the cartographic community, is the algorithm described by Lang (1969), which was used in a drawing device Geagraph 4000 provided to the ECU (Experimental Cartography Unit) in Britain by the German firms AEG and Aristo. Its working principle is illustrated in *Figure 1* in order to systematize the discussion, although it is relatively familiar to the cartography/GIS community.

In *Figure 1*, Point 1 is starting point of the line segment. From this point, a line can be drawn to connect Point 3. The distance from Point 2 from Line $\underline{1\ 3}$ is then compared with the

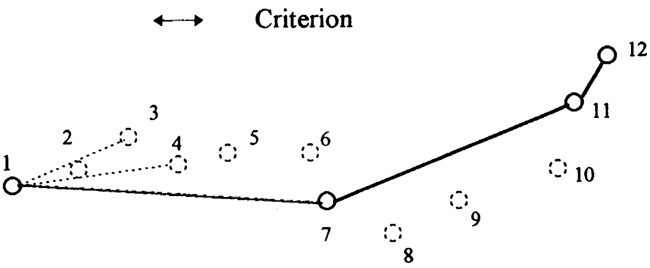

Figure 1. The working principle of a sequential method for polygonal approximation. This algorithm was described by Lang (1969).

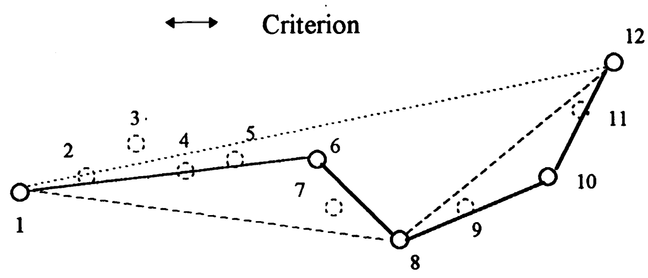

Figure 2. The working principle of Ramer (1972) algorithm (known as Douglas-Peucker (1973) algorithm in the cartography/GIS community).

criterion given to see whether this distance is greater than the criterion. If true, then Point 2 (instead of Point 3) will be selected as a critical point and used as the next starting point. In the case shown in *Figure 1*, it is not true. Then a line is drawn from Point 1 to Point 4 and the distances from Point 2 and Point 3 to this line are compared with the criterion to see whether any of them is greater than the criterion. If it is true, Point 3 (instead of Point 4) will be selected as a critical point and used as the next starting point. In fact, it is not true in this case, so this process continues until Point 8 is tested. That is, the distance from Point 6 to Line 1 8 is greater than the criterion. Now Point 7 is selected and used as the next starting point. The process continues with Point 7 as the new starting point.

Examples of algorithms belonging to this group are the Reumann-Witkam (1974) algorithm and Opheim (1981) algorithm. The only difference between the one described by Lang (1969) and these two algorithms is the way to find the longest allowable segment. For example, in the Reumann-Witkam algorithm, the concept of tolerable band is used. However, this is not much different from the concept of tolerable distance. For a detailed description of these algorithms see McMaster (1987). Other examples outside of the catrographic literature are the Sklansky and Gonzalez (1980) algorithm, the Wall and Danielsson (1984) algorithm, and the Ray and Ray (1991) algorithm.

Sequential methods have drawbacks of missing some important features. In the case of the example given in *Figure 1*, Points 6 and 8 are missed although they are richer in information than Point 7. Due to this serious drawback, in recent years, not much further development of algorithms of this category are known to the author.

Iterative methods

There are two commonly used approaches in iterative methods, i.e. progressive splitting and split-and-merge.

The progressive splitting methods split the curve segment between two initial points into two parts by a partitioning point in between them. This partitioning point is selected as a critical point and will become a new starting point of the two curve segments, which will then be split again until certain criteria are met.

The typical algorithm is the Ramer algorithm. Ramer published this algorithm in *Computer Graphics and Image Processing* in 1972. A year later, the same algorithm was published in *The Canadian Cartographer* by Douglas and Peucker (1973). In the cartography/GIS community, this algorithm is known as the Douglas-Peucker algorithm. Indeed, this algorithm has also been described by Duda and Hart (1973) in their book entitled *Pattern Classification and Scene Analysis* and is referred to as the Forsen Algorithm by them. Again, for the sake of systemization in the discussion, the working principle is illustrated in *Figure 2*.

The algorithm starts with Point 1 (first point) and Point 12 (last point) to find out the greatest distance from all points in between these two points to Line 1 12; that is, the distance from Point 8 in this example. The distance is then compared with the given criterion. It is found that this is greater than the criterion. Point 8 is then selected as a critical point resulting in the curve being split into 2 segments. It also becomes the new starting point for both of the new curve segments. The first segment is again split into two by Point 6 and the second by Point 10. Then points 1, 6, 8, 10 and 12 are the critical points detected.

It can be noted that the critical points detected by this algorithm vary with the locations of the two initial critical points. For the example given in *Figure 2*, if one starts with Point 1 and Point 9, then all points in between these two points will not be detected, although Point 8 is very important even for the curve segment from Point 1 to Point 9 only (see *Figure 3*). In this case, Points 1, 9 and 12 will be the critical points detected. This is the main drawback of this algorithm.

In addition, it is computationally expensive. Li (1988) tries to speed up such a process by using local maxima in both X and Y directions as critical points. After the first round of selection, local maxima between two critical points can then be created by applying a rotation (a particular case of a conformal coordinate transformation) to all points in between two critical points, with the line joining these two critical points becoming the new X axis. In this way, the time required is greatly reduced. In the original paper by Li (1988), the time ratio for that example was given as 3972:4132. However, these figures include the time required to compile the Fortran program, which is a large component. The actual improvement is over 15% in terms of processing time.

Another technique used in iterative methods is the so-called split-and-merge. There are many versions of the split-and-merge algorithm. The following is the working principle of a simple one which is described by Ansari and Delp (1991): (a) assign an arbitrary number of points along the line as initial critical points; e.g. Points 1, 4, 10 and 12 in *Figure 4*, (b) for each pair of the critical points, e.g. Points 4 and 10 in *Figure 4*, the perpendicular distances of all points to the line joining these two critical points are computed. If any of the perpendicular

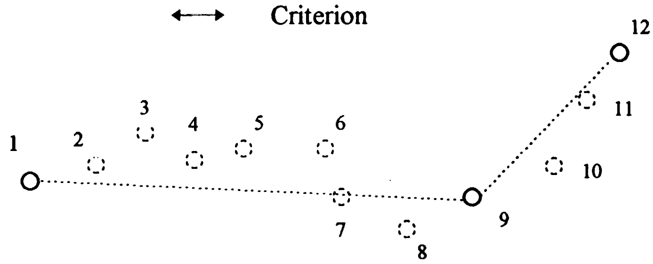

Figure 3. The variation of critical points detected by the Ramer algorithm.

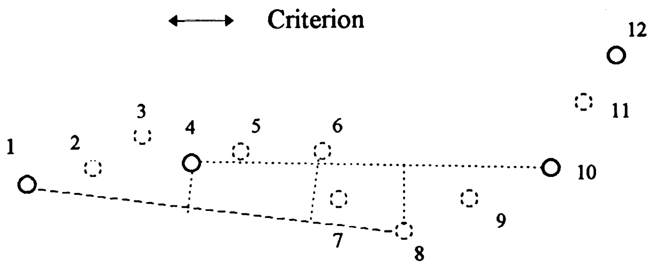

Figure 4. The working principle of split-and-merge algorithm. Points 1, 4, 10, 12 are the arbitrary initial (critical) points. In the 1st splitting, Point 8 becomes new critical point and in the 1st merging, Point 4 is removed from critical point list so that the two adjacent line segment is merged.

distances is greater than the criterion, then the point with the largest perpendicular distance is selected as a critical point, i.e. Point 8 in *Figure 4*. In other words, the curve segment between Point 4 and 10 is split by Point 8, (c) for each pair of adjacent line segments, the three consecutive critical points, e.g. Point 1, 4 and 8 in *Figure 4*, will be checked to see whether the middle critical point, i.e. Point 4 in this case, can be removed from the list of critical points. The perpendicular distance from this point (i.e. Point 4) to the line (i.e. the line joining Point 1 and 8) is then compared with the threshold. If it is shorter than the criterion, then the middle point is removed from the list. In the case of this example, Point 4 is removed from the list and these two line segments are then merged, (d) Repeat (b) and (c) until no further split-and-merge is required. In the case of *Figure 4*, Point 6 will be detected in the next round of splitting. The final list will include Points 1, 6, 8, 10 and 12.

Again, the performance of the split-and-merge methods is very sensitive to the setting of initial "critical" points partitioning the curve. For example, if Points 1, 4, 9 and 12 are used as the initial points, then the results will be identical to that shown in *Figure 3*. That is, Points 6 and 8 are not detected and the final list of critical points will be Points 1, 9 and 12.

This split-and-merge algorithm locates the critical points on the line itself. However, for other methods, the detected points may not lie on the original curve. An example of such algorithms is the one by Pavlidis and Horowitz (1974). In this algorithm, points on a curve are approximately by interpolating straight line segments.

THE CORNER DETECTION APPROACH

The common stages for corner detection techniques are, (a) estimate the curvature for each point on the curve; and (b) locate the points which have local maximum (both positive and negative) curvatures as the corners. Actually, step (b) can also be subdivided into another two steps: (1) some threshold (whether input or computed during the process) is applied to the curvature estimates to eliminate those points whose curvature is absolutely too low to be critical points; and (2) a process of non-maxima suppression is applied to the remaining points to further eliminate any point whose curvature estimate is not a local maxima in a sufficiently large segment (i.e. region of support) on the curve. Finally, the detected corners are considered as the critical points on the curve. Corner detection algorithms might also be classified into two groups, i.e.

(a) Parametric methods, and
(b) Non-parametric methods.

PARAMETRIC METHODS

Parametric methods need one or more input parameters. The classic method for curvature detection is the angle detection

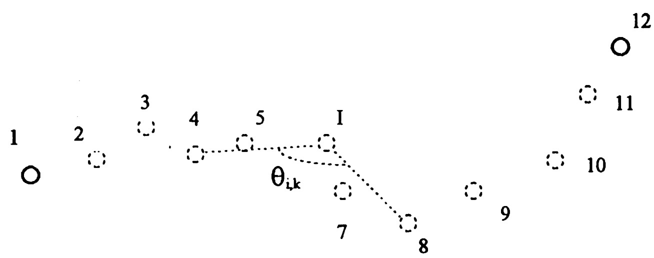

Figure 5. Estimation of curvature using cosine value $-\cos\theta$.

algorithm proposed by Rosenfeld and Johnston (1973). The cosine values $\cos\theta_{i,k}$ are used to approximate the curvature, where the angle $\theta_{i,k}$ is the angle between the line joining Point I and Point I$-$k and the joining Point I and Point I + k, where k = 1, 2, . . . m. Here m is in fact a smoothing factor and its value is an input parameter. In the case shown in *Figure 5*, I = 6, k = 2. The value of $\cos\theta_{i,k}$ is in the range between -1 (a straight line) and $+1$ (sharpest angle, i.e. 0 degree). For each point I, all the possible $\cos\theta_{i,k}$ values will be computed and then a parameter called 'the region of support' needs to be assigned which is the largest k value, K, from Point I where $\cos\theta_{i,m} < \ldots \cos\theta_{i,}K_{+1} \cos\theta_{i,}K \geqslant \cos\theta_{i,}K_{-1}$. Also the value of $\cos\theta_{i,}K$ will be assigned to point I. Finally, point I will be retained as a critical point only if $\cos\theta_{i,}k \geqslant \cos\theta_{j,}k$, for all j within the range from I $-$ K/2 to I + K/2.

Other classic algorithms belonging to this group are the Rosenfeld Weszka (1975) improved angle detection algorithms the Freeman-Davis (1977) algorithm, the Sankar-Sharma (1978) algorithm, and so on. In the cartographic community, the only algorithm belonging this group is the one described by Thapa (1988b). In this algorithm, the chained coded line features are convoluted with the second derivative of the Gaussian function. The points corresponding to the zero-crossing in the convoluted data are selected as critical points. The results will be sensitive to noise and the sigma value used in the Gaussian function.

NON-PARAMETRIC METHODS

Traditionally, it is believed that accurate determination of discrete curvature measures is the key factor in detecting reliable critical points. In most algorithms, one or more input parameters are required, which represents a priori knowledge about the digital curve. These input parameters are used to determine the region of support, from which the curvature of line points is measured. There are two problems. The first is that the object is assumed to be unknown, therefore, it is difficult to determine an appropriate parameter without trial-and-error. The second problem is that the computation of local curvature measures based on this single region of support is ineffective for objects with multi-size features. To overcome this problem. Teh and Chin (1989) designed an adaptive approach to determine the region of support for each points on a curve without any input parameter.

Instead of the cosine values, the ratio between the distance from Point I to the chord joining Points I $-$ K and I + K and the length of the chord itself, d/L as shown in *Figure 6*, is used to determine the region of support for Point I. Then, a measure of significance, e.g. the curvature measure by Rosenfeld and Johnston (1993), is computed for each point using the points within its region of support. Finally, non-maxima in curvature are suppressed. The remaining points are considered as critical points. This algorithm does not require any input parameter and works well on an object curve which is not corrupted with

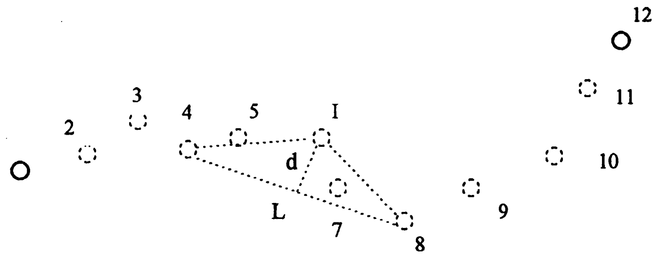

Figure 6. Teh and Chin algorithm, using d/L to determine the region and support.

noise. However, it is sensitive to noise present in the digital curve and false critical piont s will be detected due to noise (Ansari and Huang, 1991). In addition, it is computationally expensive since a region of support needs to be determined for each point.

Ansari and Huang (1991) developed another non-parametric algorithm. In this algorithm, the region of support, as suggested by Teh and Chin (1989), also needs to be determined for each point and then an adaptive Gaussian filter with a width proportional to the region of support is applied to smooth the curve. As a result, this algorithm is less sensitive to noise but its complexity has increased dramatically. Thus, it is computationally more expensive.

To overcome the shortcomings of existing algorithms. Rattarangsi and Chin (1992) as well as Pei and Lin (1992) have recently suggested the use of a technique called scale-space filtering. The scale here refers to the sigma (σ) value in the Gaussian function. Such an algorithm consists of the following steps: (a) the curve represented by x(t) and y(t) where t is the length of path along the curve, (b) the curve, represented by x(t) and y(t), is smoothed with a Gaussian filter to minimise the effect of noise, (c) the curvature at each point is computed, (d) this curvature is convoluted with the Gaussian function, (e) repeat (b) to (d) using different values for t (length of path) and σ (scale) to obtain the different curvature values, (f) the maxima (both positive and negative) can be located using the plot of scale-space image, i.e. the curvature value at the space with one axis being σ and the other being t. Pei and Lin (1992) propose a fast convolution algorithm, and it has been claimed by them that this technique is capable of detecting stable cardinal points accurately, and that the resultant critical points will not change under scaling, translation and rotation.

THE HYBRID APPROACH

Since there are certain advantages and disadvantages with both corner detection algorithms and polygonal approximation algorithms, the natural line of thought is to combine these two together so that the advantages of both can be kept and any shortcomings off-set. Some algorithms using this hybrid approach have been developed and will be examined here.

Hybrid algorithms can also be classified into two groups, i.e.

(a) corner detection followed by a split-and-merge processing, or

(b) corner detection followed by a progressive splitting processing.

Hybrid I: corner detection followed by split-and-merge

It has been noted by Ansari and Delp (1991) that points with extreme curvature are very likely to be critical points. They propose to use these points as the initial set of points for the split-and-merging process. In their algorithm, a Gaussian smoothing filter is first applied to the original curve; then a set of points with maximum positive and negative curvature on the

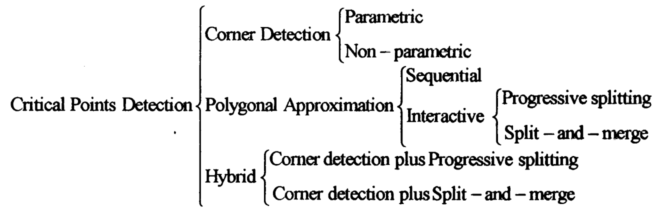

Figure 7. Classification of algorithms for detecting critical points.

Gaussian-smoothed lines is detected; and finally a split-and-merge process is followed to detect more critical points. There is a trade-off in the selection of the width for the Gaussian filter. Too large a width will remove all small variations and too small a width will create false concavities and convexities.

Hybrid II: corner detection followed by progressive splitting

Wu and Wang (1993) also noted the fact that critical points on a curve are usually the points with the largest curvature. They also use these points as initial critical points. Instead of applying a split-and-merge processing in the later stage, they apply the progressive splitting processing; that is, each curve segment between two consecutive critical points is further split until certain criteria are met. Both the initial set of critical points and those detected during progressive splitting are considered as critical points.

There are many possible combinations, considering the fact that there are so many corner detection and polygon approximation algorithms. They found that a combination of the Rosenfeld and Johnston angle detection algorithm and the Ramer polygonal approximation algorithm produces satisfactory results.

CONCLUDING REMARKS

Critical points are important in shape recognition and representation. In cartography/GIS, critical points will be of great help in the efficient display and caricature of line features as well as data compression. Due to this is importance, numerous detection algorithms have been developed. These algorithms can be classified into three main groups, i.e. direct corner (or angle detection) approach, polygonal approximation approach, and hybrid approach, i.e. a combination of the previous two. These three groups can be further divided into a few groups as shown in *Figure 7*. The majority of algorithms in the cartography/GIS literature fall into the second group. However, there are serious shortcomings with these algorithms. The most serious one is that the final result will be dependent on the initial set of critical points.

The current trend in algorithm development is to consider different levels of detail present in a curved line. In more technical terms, different sizes of support region are to be determined for different points on the same curve line due to the different scale. The techniques using scale-space filtering seem promising. Another trend is to look at the semantics of line features so that different algorithms may be applied to different types of line features.

REFERENCES

Ansari, N. and Delp, E., 1991. On detecting dominant points. *Pattern Recognition*, 24(5): 441–451.
Ansari, N. and Huang, K.-W., 1991. Non-parametric dominant point detection, *Pattern Recognition*, 24(9): 849–862.
Attneave, F., 1954. Some informational aspects of visual perception. *Psychological Review*, 61(3): 183–193.
Douglas, D. H. and Peucker, T. K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2): 112–122.

Duda, R. O. and Hart, P. E., 1973. *Pattern classification and scene analysis*. Wiley: New York.
Freeman, H., 1978. Shape description via the use of critical points. *Pattern Recognition*, 10: 159–166.
Freeman, H. and Davis, L. S., 1977. A corner-finding algorithm for chain-coded curves. *IEEE Transactions on Computers*, 26: 297–303.
Lang, T., 1969. Rules for robot draughtsmen. *Geographic Magazine*, 42(1): 50–51.
Li, Zhilin, 1988. An algorithm for compressing digital contour data. *The Cartographic Journal*, 25(2): 143–146.
Li, Zhilin, 1993. Some observations on the issue of line generalisation. *The Cartographic Journal,* 30(1): 68–71.
McMaster, R. B., 1987. Automated line generalisation. *Cartographica*, 24(2): 74–111.
Opheim, H., 1981. Smoothing a digitized curve by data reduction methods. *Eurographics' 81*, 127–135.
Pavlidis, T. and Horowitz, S. L., 1974. Segmentation of plan curves. *IEEE Transactions on Computers*, 23: 860–870.
Pei, S.-C. and Lin, C.-N., 1992. The detection of dominant points on digital curves by scale-space filtering. *Pattern Recognition*, 25(11): 1307–1314.
Ramer, U., 1972. An interactive procedure for polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1: 244–256.
Rattarangsi, A., and Chin, R. T., 1992. Scale-based detection of corners of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(4): 430–449.
Ray, B. K. and Ray, K. S., 1991. A new appraoch to polygonal approximation. *Pattern Recognition Letters*, 12(4): 229–234.
Reumann, K. and Witkam, A. P. M., 1974. Optimizing curve segmentation in computer graphics. *Proceedings of International Computing Symposium*, 1467–472.
Rosenfeld, A. and Johnston, E., 1973. Angle detection of digital curves. *IEEE Transactions on Computers,* 22: 875–878.
Rosenfeld, A. and Weszka, 1975. An improved method of angle detection on digital curves. *IEEE Transactions on Computers*, 24: 940–941.
Sankar, P. V. and Sharma, C. V., 1978. A parallel procedure for detection of dominant points on a digital curve. *Computer Graphics and Image Processing*, 7: 403–412.
Sklansky, J. and Gonzalez, V., 1980. Fast polygonal approximation of digitized curves. *Pattern Recognition*, 12: 327–331.
Teh, C.-H., and Chin, R. T., 1989. On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8): 859–872.
Thapa, K., 1988a. A review of critical points detection and line generalization algorithms. *Surveying and Mapping*, 48(3): 185–205.
Thapa, K., 1988b. Critical points detection and automated line generalization in raster data using zero-crossings. *The Cartographic Journal*, 25(1): 58–68.
Wall, K. and Danielsson, P. E., 1984. A fast sequential method for polygon approximation of digitized curves. *Computer Vision, Graphics and Image Processing*, 28: 220–227.
Wu, W.-Y. and Wang, M.-J., 1993. Detecting the dominant points by curvature-based polygonal approximation. *CVGIP: Graphical Models and Image Processing*, 55(2): 79–88.