A Direction Change-based Algorithm for Polygonal Approximation

Han Liu, Xiangliang Zhang, Alyn Rockwood
King Abdullah University of Science and Technology
{han.liu, xiangliang.zhang, alyn.rockwood}@kaust.edu.sa

Abstract

A linear-time algorithm is proposed for polygonal approximation of digital curves. The direction changes of the x- and y-coordinates are traced to generate a new, compact representation of curves. The algorithm, Direction Change-based Polygonal Approximation (DCPA), has two advantages: linear time complexity and insensitivity to parameter setting. Benchmark results demonstrate the competitive performance of DCPA using standard assessment techniques.

1. Introduction

Polygonal approximation of digital curves has been an important research topic since the beginning of the digital image [2,3,5,13]. The main idea is to find the dominant points (DPs), or landmarks of a curve, which are usually discrete endpoints of straight line segments. The boundary curve of a shape is then represented by enough points for a visually satisfying approximation. It has wide applications in computer vision, such as shape matching and video tracking.

Many methods have been developed to create the approximation. Perez [8] developed the optimum algorithm for minimizing global error with complexity $O(MN^2)$, where N and M are the number of points and DPs respectively. Nguyen [9] designed a threshold-free method using the width of blurred segments in $O(NlogN)$. Davis [3] applied the Haar transform to the differences of x- and y-coordinates with different tolerances to filter harmonics in linear time. Bhwmick [1] introduced a linear time complexity algorithm, which depends on a certain approximation criterion and error tolerance. These algorithms either have superlinear time complexities or depend on certain criteria or different kinds of parameters.

Our approach improves on the complexity of the polygonal approximation and parameter independence. The proposed algorithm detects DPs based on the direction changes of neighboring points, thus the name *Direction Change-based Polygonal Approximation* (DCPA). Based on two scans in x- and y-coordinates, DCPA has a linear time complexity. The parameters of DCPA are set according to the intrinsic characteristics of given digital curves. Therefore, DCPA is generally applicable to various digital curves.

2. DCPA: Direction Change-based Polygonal Approximation

Before introducing the proposed DCPA algorithm, we first formalize the polygonal approximation problem. Assume that a digital curve has N points, each of which has only two neighbors in its 8-neighborhood. Polygonal approximation is basically discovering dominant points for constructing a polygon with minimum deviation from the original curve and with a high compression ratio $CR = \frac{N}{N_{DP}}$, where N_{DP} is the number of DPs.

The x- or y-coordinates of each point have three types of direction, i.e., +1, 0 and -1. A digital curve is then represented by two sets of direction labels, each of which is a sequence denoted by $\boldsymbol{S}$. For example, the curve shown in Figure 1(a) has S_x= {-----0--} and S_y ={++--0-00} respectively. Since the x- and y-coordinates will be processed in the same way, we use $\boldsymbol{S}$ for simplicity in the following analysis.

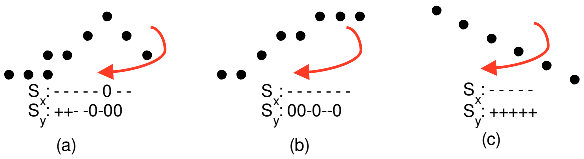

Figure 1. Direction changes of curves in x- and y-coordinates (clockwise)

A straight line has no changes in directions. Note that 0 in $\boldsymbol{S}$ means no direction change. Hence, it does not imply a dominant point. The changes of direction labels (+ → −, or − → +) signify a dominant point, which separates two straight lines. Therefore, $\boldsymbol{S}$ can be simplified to a shorter direction change sequence S_D with counts of identical occurrences S_N**.** For example, $\boldsymbol{S}$ = {++0000++---00} is simplified to S_D= {+0+-0} with S_N = {24232}. This expression shares a similar idea to run-length encoding [14]. In the next two subsections, we will introduce how DCPA identifies dominant points by scanning S_D and S_N twice.

2.1. Scan 1: single direction change

In a digital curve, line segments can be formed by a set of points with single direction changes. In **Scan 1**, we determine DPs by identifying the two endpoints of a segment with at least μ single direction changes. For example in Figure 2(a), the segments (in dots) have more than μ single direction changes in x- or y-coordinates, and have different directions from other segments (in triangles) of the curve. The DPs identified are marked by red squares.

In the DCPA algorithm, μ is empirically set to 4 according to observations in experiments where μ < 4 leads to over-fitting issues and μ > 4 causes under-fitting issues. When scanning i^{th} component of S_D, if its corresponding $S_N(i) \geq \mu$, the location of DPs (two endpoints) are given as:

Starting point: $L_{DP} = 1 + \sum_{j=1}^{i-1} S_N(j)$ (1)

Ending point: $L_{DP} = 1 + \sum_{j=1}^{i} S_N(j)$ (2)

Note that 1 is added at right-hand side of equation (1) and (2) as $S_N(j)$ only records the number of successive direction changes.

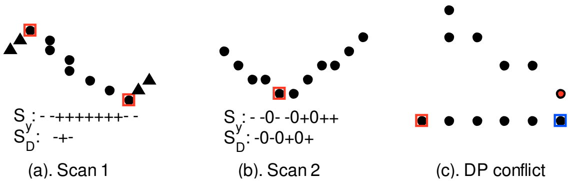

Figure 2. Examples of scans and conflict DPs

2.2. Scan 2: quasi-single direction change

In **Scan 1**, we identify ideal line segments that are composed of a series of contiguous points with single direction change. In **Scan 2**, we seek DPs for the segments with quasi-single direction changes.

When scanning S_D, the first non-zero direction is recorded as $S_{D_{this}}$. A dominant point is marked if i^{th} component has an opposite direction $S_D(i) = -S_{D_{this}}$. The location L_{DP} is calculated as:

$$L_{DP} = 1 + \sum_{j=1}^{i-1} S_N(j) \quad (3)$$

For example in Figure 2(b), $\boldsymbol{S}$ (clockwise) in the y-coordinate is --0--0+0++, and its corresponding S_D = {-0-0+0+}, and S_N = {2121112}. The first recorded direction is -, and the direction change happens at the seventh point (+) in $\boldsymbol{S}$ (i=5 the fifth point in S_D), which is labeled as a DP (red square). The location of DP is $L_{DP} = 1 + \sum_{j=1}^{5-1} S_N(j) = 7$. Note that the scan order is reversed if given in a counterclockwise order.

A special case has to be considered when calculating locations of DPs. Assume that a DP is identified at $S_D(i)$, and there are more than two successive points with the same x- or y-coordinates before it, i.e., $S_D(i-1) = 0$ and $S_N(i-1) \geq 2$. To minimize the error between the approximation polygon and the curve, the location of DP is adjusted to the middle of the successive point set. The threshold μ in **Scan 1** is to determine if the adjustment is necessary.

$$L_{DP} = \sum_{j=1}^{i-2} S_N(j) + \frac{S_N(i-1)}{2}, if\ 1 < S_N(i-1) < \mu \quad (4)$$

To suppress the duplicated DPs on independent scanning of x- and y-coordinates, we first scan x, and skip the neighbors of already identified DPs ($\mathcal{P}$, initialized by empty set ϕ when scanning x) from scan results of x when scanning y. Such an example is shown in Figure2(c), the point marked in red dot is found by Scan 2 on x, and two points marked by squares are found by Scan 1 on y, so the one marked by a blue square will not be considered as a DP.

2.3. Refinement of dominant points

The two independent scanning processes can find redundant dominant points, which will be eliminated to attain a high compression. This refinement process considers the included angle θ formed by connecting neighboring DPs. The DP at the vertex of angle θ can be deleted when: 1) $170° \leq \theta \leq 180°$; or 2) $\theta < 170°$, but there is a negligible effect (<10°) on its neighbor-induced angles before and after deleting this DP.

The DCPA algorithm is described in Figure 3, including the functions for finding and refining dominant points.

2.4. Complexity analysis

This section analyzes the time complexity of DCPA. As described in Figure 3, DCPA involves 4 steps. The runtime of step 1 is $O(N + N_S)$, where N_s is the number of $\boldsymbol{S}$. Step 2 takes $O(N + N_S log_2 N_{DP_x})$ for searching the neighbors in DP_x when scanning y. $N_{DP_x}$is the number of DPs after scanning x. Step 3 and 4 cost $O(N_{DP})$. The overall time complexity of DCPA is $O(N + N_S log_2 N_{DP_x})$. Since $N_S = N$ in the worst case and $N_{DP_x} \ll N$, the total time complexity is linear.

3. Experimental Evaluation

In this section, we present our experimental results evaluated by two widely used criteria.

3.1. Benchmarks and evaluation criteria

We use three of the most popular benchmarks for polygonal approximation of digital curves, semicircle, leaf, chromosome [2] to evaluate the proposed DCPA.

The first evaluation criterion is the most frequently used Sarkar's figure of merit ($FOM = \frac{CR}{ISE} = \frac{N}{N_{DP} \times ISE}$), integrating compression ratio (CR) and integral square error (ISE) [12]. Rosin [11] proposed *Merit* to compare approximation methods with the optimum algorithm[8].

```
DCPA: Direction Change based Polygonal Approximation
Input: direction sequence: S
Output: Approximation polygon
Main:
    $DP_x$ = find_DP ($S_x$, $\phi$)
    $DP_y$ = find_DP ($S_y$, $DP_x$)
    $DPs$ = refine ($DP_x$, $DP_y$)
    Construct a polygon by connecting all DPs;
// function of find_DP
    DP = find_DP(S, $\mathcal{P}$) ($\mathcal{P}$: generated DPs during scans,
                                      initialized as $\phi$)
    Convert S to $S_D$ and $S_N$, $N_s$ = length(S)
    Set $S_{D_{this}}$ the first non-zero $S_D$
    next = i+1;   prev = i-1;
    if i==$N_s$   then next = 1;
    if i==1      then prev = $N_s$ ;
    for i = 1to $N_s$
        if $S_D$ (prev) $\notin \mathcal{P}$ and $S_D$ (next) $\notin \mathcal{P}$ then
            if $SN_i \geq \mu$ then
                use eq. (1) and (2) to determine DP
            if $S_D(i) = -S_{D_{this}}$ then
                use eq.(3) or (4) to determine DP;
// function of refine
    DP = refine ($DP_x$, $DP_y$)
    $L_{DP}$ = location of $(DP_x \cup DP_y)$;
    $N_{DP}$ = length($L_{DP}$ );
    for i = 1 to $N_{DP}$
        next = i+1;
        if i==$N_{DP}$  then next = 1;
        $\theta(i) = arctan\frac{y_{next}-y_i}{x_{next}-x_i}$ ;
        if 180°- $\theta$ (i) < 10°  then
            delete DP at $L_{DP}$ (i);
        else
            prev = i-1;
            if i==1  then prev = $N_{DP}$;
            calculate $\theta'$(prev) , $\theta'$(next) assuming that
            a DP at $L_{DP}$ (i) is deleted;
            if |$\theta'$(prev)- $\theta$(prev)| < 10°
              & |$\theta'$(next)- $\theta$(next) |< 10°  then
                delete DP at $L_{DP}$ (i);
```

Figure 3. Algorithm DCPA

Merit is computed from Fidelity $= \frac{E_{opt}}{E_{approx}} \times 100$, and Efficiency $= \frac{M_{opt}}{M_{approx}} \times 100$, where E_{opt} and E_{approx} are errors of the optimum and approximation algorithm respectively with the same N_{DP}. M_{approx} is the number of DPs produced by the approximation algorithm, and M_{opt} is the number of DPs that the optimum algorithm needs to produce the same error, approximated by interpolation of two closest integer values. Approximation algorithms are ranked according to Merit $= \sqrt{Fidelity \times Efficiency}$.

3.2. Evaluation results

Figure 4 shows a comparison between the proposed method DCPA and Bhowmick's optimal ADSS [1], which is also a linear-time algorithm but sensitive to the parameter settings of an error tolerance τ and N_{DP}. It is worth to note that our algorithm does not rely on the setting of thresholds, and can be directly applied to any input curves without parameter adjustments. More importantly, on the two benchmarks (Chromosome and Semicircle) shown in Figure 4, our proposed algorithm can generate a better approximation with less error.

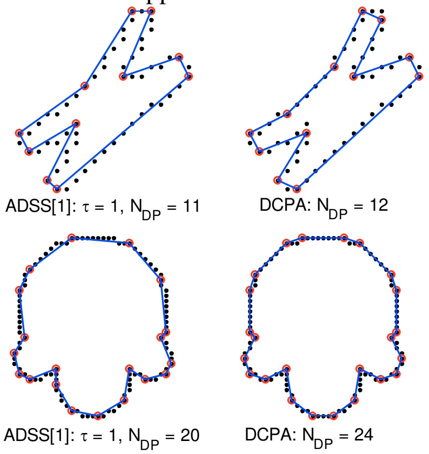

Figure 4. Benchmark comparisons

Figure 5 shows polygonal approximation results of DCPA on Leaf benchmark and two other digital curves. The well-fitting results demonstrate DCPA is applicable to digital curves in various shapes and on different scales.

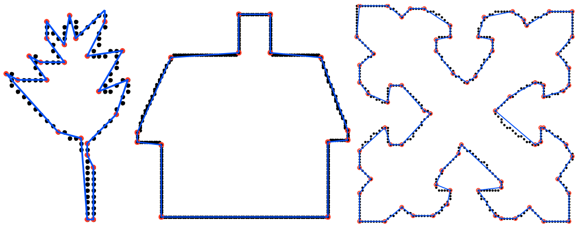

Figure 5. Polygonal approximations of DCPA

Table 1 compares DCPA with other popular algorithms [5-7] evaluating on three benchmarks by the first criterion. We show the assessment of N_{DP}, *CR, ISE, MaxE* (Maximal Error), and *FOM*. Since *FOM* relates to N_{DP}, we compare our results with those having a similar value of N_{DP}. DCPA outperforms other algorithms on chromosome and semicircle by improving more than 37% on *FOM*, and has a lower *MaxE* and similar *ISE* on the leaf benchmark.

Table 1. Evaluating on Benchmarks by FOM

Benchmark	Algo.	N_{DP}	CR	ISE	MaxE	FOM
Chromo-some	Marji [5]	12	5	8.03	0.895	0.623
	Masood[6]	12	5	7.76	0.88	0.65
	Nguyen[7]	15	4	5.69	0.686	0.703
	DCPA	12	5	4.93	0.417	**1.013**
Leaf *Figure5(a)*	Marji [5]	22	5.45	13.21	0.78	0.413
	Masood[6]	23	5.22	10.61	0.74	0.49
	Nguyen[7]	24	5	9.989	0.728	0.50
	DCPA	25	4.8	**10.20**	**0.626**	0.471
Semicircle	Marji [5]	26	3.92	9.01	0.74	0.435
	Masood[6]	22	4.64	8.61	0.72	0.54
	Nguyen[7]	22	4.68	10.31	0.784	0.454
	DCPA	24	4.25	5.714	0.329	**0.744**

Table 2 is a comparison between our DCPA and five high-rank techniques on the semicircle benchmark evaluated by *Merit*. DCPA ranked behind the first two algorithms [4] and [11], which have few dominant points, leading to loss of information and large *ISE* (about 4 and 26 times of our *ISE* ≅ 5.7).

Table 2. Rosin's assessments on semicircle

Algo.	N_{DP}	ISE	Fidel.	Effi.	Merit	Rank
Lowe[4]	13	21.66	95.7	98.6	97.1	1
Banerjee[2]	6	150.53	93.3	98.7	96.0	2
DCPA	24	5.7141	88.93	95.76	**92.28**	-
Massod[6]	21	9.82	817	95.8	88.5	-
Ramer[10]	26	5.27	76.9	92.6	84.4	3
Nguyen[7]	22	10.31	67.99	86.36	76.6	-

In section 2, we claimed that the threshold μ is fixed to 4 and experimental results verified its effectiveness. Figure 6(a) and (b) demonstrate issues of over-fitting and under-fitting when μ is not set to 4.

Our future work will focus on analyzing special patterns of digital curves referring to methods like change-point estimation [15]. For example in Figure 6(c) where a digital curve has zig-zag order of pixels with more than 4 pixels in each line, DCPA has limitations on identifying the correct approximation shown in Figure 6 (d).

4. Conclusion

In this paper, a new efficient algorithm for polygon approximation, DCPA, is proposed. Through scanning the direction changes on x- and y-coordinates, DCPA can efficiently find effective dominant points in linear time. The insensitivity to parameter setting makes DCPA an easy-to-use algorithm. Evaluation results on benchmarks demonstrate the proposed method outperforms other popular techniques by improving the value of *FOM* and *Merit* and reducing the approximation error.

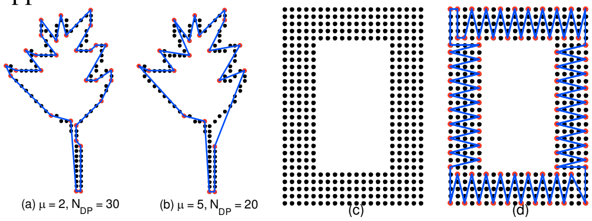

Figure 6. Different μ and Extreme cases

References

[1] P. Bhowmick and B. Bhattacharya. Fast polygonal approximation of digital curves using relaxed straightness properties. *PAMI*, 29(9):1590–1602, 2007.

[2] S. Banerjee, W. Niblack, and M. Flickner. A minimum description length polygonal approximation method. *IBM Research Division*, 1996.

[3] T. Davis. Fast decomposition of digital curves into polygons using the haar transform. *Trans. Pattern Analysis and Machine Intelligence*,21(8):786–790, 1999.

[4] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial intelligen*ce, 31(3):355–395, 1987.

[5] M. Marji and P. Siy. A new algorithm for dominant points detection and polygonization of digital curves. *Pattern recognition*, 36(10):2239–2251, 2003.

[6] A. Masood. Dominant point detection by reverse polygonization of digital curves. *Image and Vision Computing*, 26(5):702–715, 2008.

[7] T. Nguyen and I. Debled-Rennesson. Fast and robust dominant points detection on digital curves. *International Conference on Image Processing,* 953–956. 2009.

[8] J. Perez and E. Vidal. Optimum polygonal approximation of digitized curves. *PRL*, 15(8):743–750, 1994.

[9] T. Nguyen, I. Debled-Rennesson, et al. A multiscale approach to decompose a digital curve into meaningful parts. In *ICPR* 1072-1075, 2010.

[10] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, 1972.

[11] P. Rosin. Techniques for assessing polygonal approximations of curves. . *PAMI*, 19(6):659–666, 1997.

[12] D. Sarkar. A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves. *PRL*, 14(12):959–964, 1993.

[13] C. Teh and R. Chin. On the detection of dominant points on digital curves. *PAMI*, 11(8):859–872, 1989.

[14] M. Burrows and D. Wheeler. A block sorting lossless data compression algorithm. *Technical Report 124, Digital Equipment Corporation*, 1994.

[15] A. Goldenshluger, A. Tsybakov, and A. Zeevi. Optimal change-point estimation from indirect observations. *The Annals of Statistics*, 34(1): 350-372, 2006.