Available online at www.sciencedirect.com
ScienceDirect
J. Vis. Commun. Image R. 19 (2008) 219–230
Journal of Visual Communication & Image Representation
www.elsevier.com/locate/jvci

Novel efficient two-pass algorithm for closed polygonal approximation based on LISE and curvature constraint criteria

Kuo-Liang Chung *,[1], Po-Hsuan Liao, Jia-Ming Chang

Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei 10672, Taiwan, ROC

Received 2 January 2006; accepted 24 January 2008
Available online 21 February 2008

Abstract

Given a closed curve with n points, based on the local integral square error and the curvature constraint criteria, this paper presents a novel two-pass $O(Fn + mn^2)$-time algorithm for solving the closed polygonal approximation problem where $m(\ll n)$ denotes the minimal number of covering feasible segments for one point and empirically the value of m is rather small, and $F\ (\ll n^2)$ denotes the number of feasible approximate segments. Based on some real closed curves, experimental results demonstrate that under the same number of segments used, our proposed two-pass algorithm has better quality and execution-time performance when compared to the previous algorithm by Chung et al. Experimental results also demonstrate that under the same number of segments used, our proposed two-pass algorithm has better quality, but has some execution-time degradation when compared to the currently published algorithms by Wu and Sarfraz et al.

Keywords: Algorithm; Closed curve; Closed polygonal approximation algorithm; Curvature; Local integral square error; Shortest path algorithm

1. Introduction

Polygonal approximation (PA) is an important method for shape representation [4]. The primary goal of PA is to determine an approximate polygonal curve as the contour representation of one object. Sometimes the approximate polygonal curve can be thought as a special compression method for representing the contour of that object. Given a polygonal curve C with n vertices, $C = \langle P_1, P_2, P_3, \ldots, P_n\rangle$, the PA problem is to find another similar polygon C' with n' vertices, where $C' = \langle P_{1'}, P_{2'}, P_{3'}, \ldots, P_{n'}\rangle$ and $|C'| \leqslant |C|$, such that the obtained approximate polygonal curve satisfies some error criteria to retain an acceptable quality. The determined n' vertices in C' must be a subset of C. Besides the compression benefit, the closed approximate polygonal curve C' can make the manipulation and analysis, e.g., rotation, translation, scaling, clipping, and partitioning, easy and it leads to a computational benefit. In addition, the PA approach has also been applied to the binary image progressive transmission [8] successfully.

In the past three decades, many efficient PA algorithms for open curves have been developed. These developed PA algorithms can be classified into two types, the sub-optimal algorithms [18,14,13,6,7] and the optimal algorithms [10,3,17,12,11,1,9]. These heuristic sub-optimal PA algorithms are quite fast, but the obtained approximate polygonal curves usually are local optimal solutions. To extend the open PA problem to the closed PA (CPA) problem, naturally we exhaustively examine all vertices in C as the possible starting points, and finally determine the global solution within these n possible solutions. For solving the CPA problem, Sato [17] presented an $O(n^3)$-time algorithm in which the selected starting point is the farthest point

* Corresponding author. Fax: +886 2 273 76777.
E-mail address: k.l.chung@mail.ntust.edu.tw (K.-L. Chung).
[1] Supported by National Science Council of R.O.C., under contracts NSC96-2218-E-011-002, NSC96-2219-E-011-001, and NSC96-2221-E-011-027.

1047-3203/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2008.01.004

from the center of gravity. The error criterion of arc lengths between the original curve and the approximating curve is used. Based on the L_∞ metric, Zhu and Seneviratne [21] presented a heuristic $O(n^3)$-time algorithm for solving the CPA problem. Horng and Li [5] presented a heuristic $O(sn^2)$-time algorithm where s denotes the number of specified polygonal segments. Since the above three efficient algorithms for solving the CPA problem are heuristic, the optimization of their results can not be guaranteed. In [12], the global integral square error (ISE) criterion is used. In [14,13,1], the local ISE (LISE) criterion is used. In [1], the presented CPA algorithm takes $O(n^3)$ time and the number of polygonal segments determined is minimal. Basically, the LISE metric can keep more peak information when compared to the global ISE. Besides using the LISE error criterion in the CPA problem, the curvature constraint is also a common error criterion. Under the same k-cosine curvature metric, Wu and Wang [19] presented an efficient coarse-to-fine algorithm for determining dominant points which were connected as the solution of the CPA problem.

Based on the concept of break points, Wu [20] presented an adaptive, improved CPA algorithm. In Wu's algorithm, the set of break points are first filtered out using the curvature criterion. Further, the break points with maximum curvature are selected as the dominant points which are considered in his CPA algorithm. In [16], Sarfraz et al. presented a recursive algorithm for solving the CPA problem. Their algorithm extracts initial break points as a preprocessing step. Among these initial break points, they first consider points with angle of 135° as the dominant points in their CPA algorithm. If no point with 135° angle is found, then the points with angle of 90° are considered. If the points with 90° angle are still not available, then the first break point is considered as the dominant point. Experimental results demonstrated that the CPA algorithm by Sarfraz et al. is quite competitive with the one by Wu. The motivation of this paper is to design a novel, efficient CPA algorithm under the three error criteria, namely the LISE, the curvature constraint, and the longest vertical distance consideration.

This paper presents a novel two-pass $O(Fn + mn^2)$-time algorithm for solving the CPA problem where $m(\ll n)$ denotes the minimal number of covering segments for one point and empirically the value of m is rather small and $F(\ll n^2)$ denotes the number of feasible approximate segments. According to the concept of covering segments for each point, the first pass of our proposed algorithm can be performed in $O(Fn + mn^2)$ time under the given LISE criterion; the second pass of our proposed algorithm can be performed in $O(n)$ time under the given curvature constraint and the longest vertical distance consideration. Because of $Fn + mn^2 \ll n^3$ and considering these criteria, our proposed CPA algorithm has better time performance and quality when compared to the previous CPA algorithm by Chung et al. [1] which takes $O(n^3)$ time complexity and considers only the LISE criterion. Based on two real closed curves for representing French and Italy, experimental results demonstrate that under the same number of segments used, our proposed two-pass algorithm has better quality and execution-time performance when compared to the currently published algorithm by Chung et al. [1]. Based on two real closed curves for representing a semicircle and a chromosome, experimental results demonstrate that under the same number of segments used, our proposed two-pass algorithm has better quality, but has some execution-time degradation when compared to currently published algorithms by Wu [20] and Sarfraz et al. [16].

The rest of this paper is organized as follows. Section 2 introduces the concerned two error criteria, the LISE bound and the curvature constraint. Section 3 presents new methods for determining all feasible segments and covering segment sets which will be used in our proposed two-pass CPA algorithm. Section 4 presents our proposed whole two-pass CPA algorithm. The definition of longest vertical distance consideration will be explained in Section 4. Experimental results are demonstrated in Section 5. Conclusions are addressed in Section 6.

2. Error criteria

In this section, both the LISE and the curvature criteria are introduced. For exposition, the next paragraph introduces the definition of discrete curvature measure, which will be used in the second pass of our proposed CPA algorithm, and how to measure it for each point on the original curve. The curvature constraint accompanied with the longest vertical distance consideration will be introduced in Section 4.2. Next, the definition of LISE is given and it will be used in the first pass of our proposed CPA algorithm.

Curvature can be explained as how much the curve bends at each point on the curve. It has been defined that the original polygonal curve is presented by the set $\{P_i = (x_i, y_i) | i = 1, 2, \ldots, n\}$ and P_i denotes the ith point with coordinate (x_i, y_i) on the original polygonal curve. Following the k-cosine value to estimate the curvature of each point, as shown in Fig. 1, the estimated curvature at point P_i related to two points P_{i-k} and P_{i+k}, is set to be the k-cosine value cos_{ik}.

Definition 1. [15] The k-cosine value at point P_i related to two k-index apart neighboring points P_{i-k} and P_{i+k} is defined by

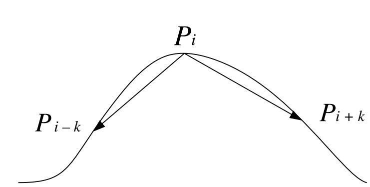

Fig. 1. The depiction for the definition of k-cosine value.

$$\cos_{ik} = \frac{\overrightarrow{P_iP_{i-k}} \cdot \overrightarrow{P_iP_{i+k}}}{|\overrightarrow{P_iP_{i-k}}||\overrightarrow{P_iP_{i+k}}|} \tag{1}$$

where $\overrightarrow{P_iP_{i-k}} = (x_{i-k} - x_i, y_{i-k} - y_i)$, $\overrightarrow{P_iP_{i+k}} = (x_{i+k} - x_i, y_{i+k} - y_i)$, and $|W|$ denotes the vector length of W; the operator '$\cdot$' denotes the inner product operation.

By Definition 1, $\cos_{ik}$ denotes the cosine value of the angle between the vector $\overrightarrow{P_iP_{i-k}}$ and the vector $\overrightarrow{P_iP_{i+k}}$. In our research, only 1-cosine value is enough in the curvature measure.

Lemma 1. *The* 1-*cosine value* $\cos_{i1}$ *can be computed in* $O(1)$ *time.*

The LISE criterion is now defined as follows.

Definition 2. The LISE between the segment passing through two points, P_i and P_j, denoted as $\overline{P_iP_j}$, and the set of points $P_{i+1}, P_{i+2}, \ldots, P_{j-1}$ is expressed by

$$\mathrm{LISE}_{i,j} = \sum_{k=i+1}^{j-1} d^2(P_k, \overline{P_iP_j})$$

where $d(P_k, \overline{P_iP_j})$ denotes the Euclidean distance from the point P_k to the approximate segment $\overline{P_iP_j}$ (see Fig. 2). The convention '$j > i$' is followed through this paper.

The equation of the approximate segment $\overline{P_iP_j}$ can be expressed by $y - y_i = \frac{y_j - y_i}{x_j - x_i}(x - x_i)$. We thus have $(y_i - y_j)x + (x_j - x_i)y + (x_iy_j - x_jy_i) = 0$. Let $a_{i,j} = y_i - y_j$, $b_{i,j} = x_j - x_i$, and $c_{i,j} = x_iy_j - x_jy_i$, when plugging the three parameters, $a_{i,j}$, $b_{i,j}$, and $c_{i,j}$, into the above point-slope form, we have $a_{i,j}x + b_{i,j}y + c_{i,j} = 0$. Consequently, the LISE criterion, which is denoted by the shaded area of Fig. 2, can be written as

$$\begin{aligned}\mathrm{LISE}_{i,j} &= \sum_{k=i+1}^{j-1} d^2(P_k, \overline{P_iP_j}) \\ &= \frac{1}{a_{i,j}^2 + b_{i,j}^2} \sum_{k=i+1}^{j-1} (a_{i,j}x_k + b_{i,j}y_k + c_{i,j})^2 \\ &= \frac{1}{a_{i,j}^2 + b_{i,j}^2} \sum_{k=i+1}^{j-1} [a_{i,j}^2x_k^2 + b_{i,j}^2y_k^2 + c_{i,j}^2 \\ &\quad + 2a_{i,j}b_{i,j}x_ky_k + 2a_{i,j}c_{i,j}x_k + 2b_{i,j}c_{i,j}y_k]\end{aligned} \tag{2}$$

In our proposed two-pass CPA algorithm, the LISE criterion is used in the first pass of the proposed algorithm. The minimal set of approximate segments determined in this pass must satisfy that the LISE value of each approximate segment (see Eq. (2) and the shaded area in Fig. 2) must be less than the specified threshold T_{LISE}. In the second pass, for each determined approximate segment, denoted as $\overline{P_iP_j}$, we first examine 1-cosine values of P_k's for $i + 1 \leqslant k \leqslant j - 1$. Based on the examining result, we further identify these points, each point with high curvature, to be the set of high-curvature points. Empirically, if the curvature of each point is greater than -0.8 [20], it is claimed to be a high-curvature point. We further calculate the vertical distance from each high-curvature point to $\overline{P_iP_j}$. Based on these calculated vertical distances, we select the high-curvature point with the longest vertical distance as the key point. The corresponding selected key points in the first pass will be processed further in the second pass to obtain the final solution.

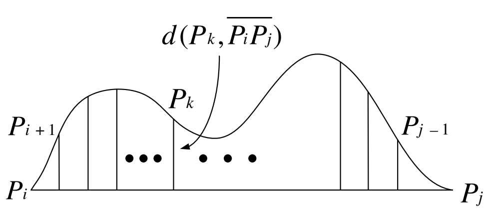

Fig. 2. The distance from P_k to $\overline{P_iP_j}$ and the LISE.

3. Determining feasible approximate segments and covering segment sets

This section first presents a modified fast $O(n^2)$-time method for determining all feasible approximate segments in C. From the determined feasible approximate segments, a novel concept of covering segment sets, which can be realized in $O(Fn)$ time where $F(\ll n^2)$ denotes the number of feasible approximate segments, is presented for speeding up the computation effort in the first pass of our proposed CPA algorithm.

3.1. $O(n^2)$-time method for determining all feasible approximate segments

Instead of using nine parameters and thirty six equalities to help deriving an $O(n^2)$-time method to construct all feasible approximate segments [1], in this subsection, our simpler method only needs five parameters and ten equalities. Let

$$S_1(i,j) = \sum_{k=i+1}^{j-1} x_k^2,$$

$$S_2(i,j) = \sum_{k=i+1}^{j-1} y_k^2,$$

$$S_3(i,j) = \sum_{k=i+1}^{j-1} x_ky_k,$$

$$S_4(i,j) = \sum_{k=i+1}^{j-1} x_k,$$

and

$$S_5(i,j) = \sum_{k=i+1}^{j-1} y_k,$$

then by Eq. (2), we have

$$\mathrm{LISE}_{i,j} = \frac{1}{a_{i,j}^2 + b_{i,j}^2}[a_{i,j}^2 S_1(i,j) + b_{i,j}^2 S_2(i,j) + (j-i-1)c_{i,j}^2 + 2a_{i,j}b_{i,j}S_3(i,j) + 2b_{i,j}c_{i,j}S_4(i,j) + 2c_{i,j}a_{i,j}S_5(i,j)]. \quad (3)$$

Lemma 2. *The values of all $LISE_{i,j}$'s for $1 \leqslant i,j \leqslant n$ can be computed in $O(n^2)$ time in an incremental way.*

Proof. It is known that $S_1(i, j) = \sum_{k=i+1}^{j-1} x_k^2$, $S_2(i, j) = \sum_{k=i+1}^{j-1} y_k^2$, $S_3(i, j) = \sum_{k=i+1}^{j-1} x_k y_k$, $S_4(i, j) = \sum_{k=i+1}^{j-1} x_k$, and $S_5(i, j) = \sum_{k=i+1}^{j-1} y_k$. The calculations of $S_l(i,j+1)$ and $S_l(i+1,j)$ can be obtained in $O(1)$ time using few arithmetic operations via $S_l(i,j)$ for $l = 1, 2, 3, 4, 5$; the related ten formulas for two cases are expressed below:

Case 1 when $j < n$:

$$S_1(i,j+1) = \sum_{k=i+1}^{j} x_k^2 = \sum_{k=i+1}^{j-1} x_k^2 + x_j^2 = S_l(i,j) + x_j^2$$

$$S_2(i,j+1) = \sum_{k=i+1}^{j} y_k^2 = \sum_{k=i+1}^{j-1} y_k^2 + y_j^2 = S_2(i,j) + y_j^2$$

$$S_3(i,j+1) = \sum_{k=i+1}^{j} x_k y_k = \sum_{k=i+1}^{j-1} x_k y_k + x_j y_j = S_3(i,j) + x_j y_j$$

$$S_4(i,j+1) = \sum_{k=i+1}^{j} x_k = \sum_{k=i+1}^{j-1} x_k + x_j = S_4(i,j) + x_j$$

$$S_5(i,j+1) = \sum_{k=i+1}^{j} y_k = \sum_{k=i+1}^{j-1} y_k + y_j = S_5(i,j) + y_j$$

Case 2 when $i + 1 < j$:

$$S_1(i+1,j) = \sum_{k=i+2}^{j-1} x_k^2 = \sum_{k=i+1}^{j-1} x_k^2 - x_{i+1}^2 = S_l(i,j) - x_{i+1}^2$$

$$S_2(i+1,j) = \sum_{k=i+2}^{j-1} y_k^2 = \sum_{k=i+1}^{j-1} y_k^2 - y_{i+1}^2 = S_2(i,j) - y_{i+1}^2$$

$$S_3(i+1,j) = \sum_{k=i+2}^{j-1} x_k y_k = \sum_{k=i+1}^{j-1} x_k y_k - x_{i+1}y_{i+1} = S_3(i,j) - x_{i+1}y_{i+1}$$

$$S_4(i+1,j) = \sum_{k=i+2}^{j-1} x_k = \sum_{k=i+1}^{j-1} x_k - x_{i+1} = S_4(i,j) - x_{i+1}$$

$$S_5(i+1,j) = \sum_{k=i+2}^{j-1} y_k = \sum_{k=i+1}^{j-1} y_k - y_{i+1} = S_5(i,j) - y_{i+1}$$

Initially, by Eq. (3), the value of $\mathrm{LISE}_{1,2}$ can be computed in $O(1)$ time. Then, using the above five equalities in Case 1 for $j < n$, the value of $\mathrm{LISE}_{1,3}$ can be obtained in $O(1)$ time via $\mathrm{LISE}_{1,2}$. Consequently, the values of $\mathrm{LISE}_{1,4}, \ldots,$ and $\mathrm{LISE}_{1,n}$ can be calculated in $O(n)$ time by such an incremental way. Using the above five equalities in Case 2 for $i + 1 < j$, the value of $\mathrm{LISE}_{2,3}$ can be obtained in $O(1)$ time via $\mathrm{LISE}_{1,3}$. Similarly, using the above five equalities in case 1 when $j < n$, the values of $\mathrm{LISE}_{2,4}$, $\mathrm{LISE}_{2,5}, \ldots,$ and $\mathrm{LISE}_{2,n}$ can be calculated in $O(n)$ time in an incremental way. By the same arguments, the values of all $\mathrm{LISE}_{i,j}$'s for $1 \leqslant i,j \leqslant n$ can be computed in $O(n^2)$ time. We complete the proof. □

By Lemma 2, all values of $\mathrm{LISE}_{i,j}$'s corresponding to all possible approximate segments $\overline{P_iP_j}$'s for $1 \leqslant i,j \leqslant n$ can be computed in $O(n^2)$ time. Since the number of all possible approximate segments is bounded by $O(n^2)$ and each $\mathrm{LISE}_{i,j}$ would consider distance-calculations from $(j - i - 1)$ points P_k's for $i + 1 \leqslant k \leqslant j - 1$ to $\overline{P_iP_j}$, the heuristic method can obtain all values of $\mathrm{LISE}_{i,j}$'s for $1 \leqslant i,j \leqslant n$ in $O(n^3)$ time because of $\sum_{i=1}^{n}\sum_{j=i}^{n}(j-i-1) = O(n^3)$.

Lemma 3. *Our proposed $O(n^2)$-time method for computing all values of $LISE_{i,j}$'s for $1 \leqslant i,j \leqslant n$ is faster than the heuristic $O(n^3)$-time method.*

Definition 3. (Feasible approximate segment) Given a specified LISE tolerance T_{LISE}, if $\mathrm{LISE}_{i,j} \leqslant T_{\mathrm{LISE}}$, the approximate segment $\overline{P_iP_j}$ is called a feasible approximate segment; otherwise, it is called an infeasible approximate segment.

By Lemmas 2, 3, and Definition 3, it is easy to verify the following result.

Theorem 1. *Given a closed curve with n points, all the feasible approximate segments can be determined in $O(n^2)$ time using our proposed method which is faster than the heuristic method.*

After describing our proposed method to determine all feasible approximate segments and its time complexity analysis, the proposed covering segment set concept is presented in Section 3.2.

3.2. $O(Fn)$-time method for determining all covering segment sets

Based on the determined feasible approximate segments (see Definition 3), a feasible approximate segment $\overline{P_iP_j}$, $1 \leqslant i,j \leqslant n$, is said to be a covering segment for point P_k, $i \leqslant k \leqslant j$.

Definition 4. (Covering segment set) The covering segment set of point P_k, $1 \leqslant k \leqslant n$, is denoted by $S_{\mathrm{CS}}(P_k) = \{\overline{P_{i_1}P_{j_1}}, \overline{P_{i_2}P_{j_2}} \ldots, \overline{P_{i_l}P_{j_l}}\}$ $\overline{P_{i_m}P_{j_m}}$, $1 \leqslant m \leqslant l \leqslant n$ and $1 \leqslant i_l < j_l \leqslant n$, is a covering segment of point P_k and it satisfies $i_m \leqslant k \leqslant j_m$.

Given an example as shown in Fig. 3 by Definition 4, the covering segment sets for points P_3, P_4, P_5, and P_6 are denoted by $S_{\mathrm{CS}}(P_3) = \{\overline{P_1P_4}, \overline{P_2P_7}, \overline{P_2P_8}\}$, $S_{\mathrm{CS}}(P_4) = \{\overline{P_1P_4}, \overline{P_2P_7}, \overline{P_2P_8}, \overline{P_4P_7}, \overline{P_4P_8}\}$, $S_{\mathrm{CS}}(P_5) = \{\overline{P_2P_7}, \overline{P_2P_8}, \overline{P_4P_7}, \overline{P_4P_8}\}$, and $S_{\mathrm{CS}}(P_6) = \{\overline{P_2P_7}, \overline{P_2P_8}, \overline{P_4P_7}, \overline{P_4P_8}\}$, respectively.

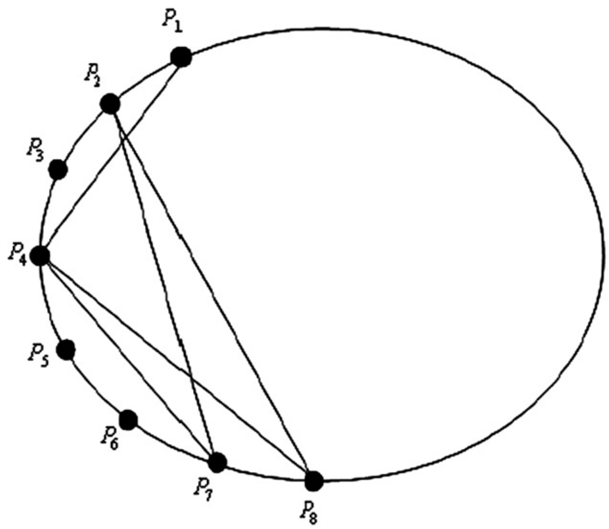

Fig. 3. One example for demonstrating covering segment sets.

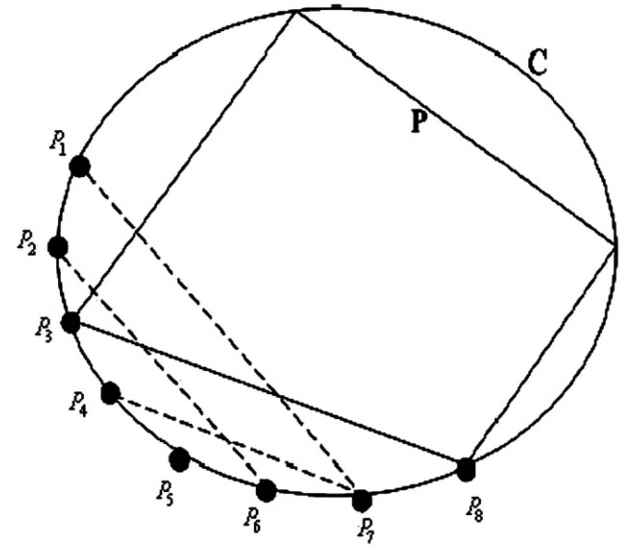

Fig. 4. The depiction for showing Theorem 3.

After introducing the definition of the covering segment set for each point, the time complexity analysis for determining all covering segment sets is shown below.

Theorem 2. *Given a closed curve C with n points, let F denote the number of all feasible approximate segments of C, all the covering segment sets of these n points can be determined in O(Fn) time.*

Proof. For each feasible approximate segment $\overline{P_{i_k}P_{j_k}}$, $1 \leqslant k \leqslant F$ and $1 \leqslant i_k < j_k \leqslant n$, we save the covering segment information $\overline{P_{i_k}P_{j_k}}$ into the points P_{i_k}, P_{i_k+1}, …, P_{j_k-1}, and P_{j_k}. Since each point can keep at most F covering segments, it takes $O(F)$ time in the worst case to save the information of these covering segments and these saved covering segment form the so called covering segment set for that point (see Definition 4). Considering the given n points in C, it takes $O(Fn)$ time to save the information of all concerned covering segment sets. We complete the proof. □

Theorem 3. *Given a closed curve C with n points, each edge of the final approximate polygon belongs to the covering segment sets of at least one original point.*

Proof. For exposition, in Fig. 4, the symbol C' denotes the final approximate polygon with four feasible approximate segments. By Theorem 2, we have $S_{\text{CS}}(P_5) = \{\overline{P_1P_7}, \overline{P_2P_6}, \overline{P_3P_8}, \overline{P_4P_7}\}$.

Considering any feasible approximate segment $\overline{P_iP_j}$ in C', it exists at least one original point P_k, $1 \leqslant k \leqslant n$, such that P_k is covered by the segment $\overline{P_iP_j}$ for $i \leqslant k \leqslant j$. That is, the segment $\overline{P_iP_j}$ belongs to $S_{\text{CS}}(P_k)$. We complete the proof. □

Up to now, each point P_k, $1 \leqslant k \leqslant n$, in the original curve C has known its own covering segment set $S_{\text{CS}}(P_k)$ and the corresponding cardinality $|S_{\text{CS}}(P_k)|$. After checking all the cardinalities of covering segment sets for all points P_k's for $1 \leqslant k \leqslant n$, we have the following result:

Theorem 4. *The minimal covering segment set* $\text{Min}_k\{|S_{CS}(P_k)|\}$ *can be determined in O(n) time.*

4. The Proposed Two-Pass Algorithm for Solving CPA Problem

In this section, the proposed novel two-pass efficient algorithm for solving the CPA problem is presented.

4.1. First-pass of the proposed CPA algorithm

Given an original closed curve C and the specified LISE criterion threshold T_{LISE}, the first pass of the proposed CPA algorithm consists of the following three steps:

Step 1. By Eq. (3), it takes $O(n^2)$ time to compute all LISE pairs $\text{LISE}_{i,j}$'s for $1 \leqslant i,j \leqslant n$. For each $\text{LISE}_{i,j}$ of $\overline{P_iP_j}$, if $\text{LISE}_{i,j} \leqslant T_{\text{LISE}}$, $\overline{P_iP_j}$ is set to a feasible approximate segment with weight 1. Finally, the corresponding directed graph $G = (V,E)$ is constructed where the node set V denotes the original set of points in C and the edge set E denotes the set of all feasible segments.

Step 2. For all points P_k's for $1 \leqslant k \leqslant n$, it takes $O(Fn)$ time to build up all the covering segment sets $S_{CS}(P_k)$'s. Further, it takes $O(n)$ time to determine the minimal covering segment set and the associated point $P_{k'}$. This step takes $O(Fn + n)$ time.

Step 3. Selecting each covering segment in $S_{\text{CS}}(P_{k'})$ as the starting segment, let the two end points of the selected segment be denoted by A and B. Since the given curve is closed, taking the point A as the starting source and taking the point B as the ending target, we run the Dijkstra's single-source and single-target shortest-path algorithm [2] on G to find the temporary approximate polygon in $O(n^2)$ time. Since there are m $(=|S_{CS}(P_{k'})|)$ concerned covering segments, in total, it needs to run the Dijkstra's algorithm m times and hence takes $O(mn^2)$ time to obtain these m possible temporary approximate polygon solutions. Finally, it takes $O(m)$ time to select the temporary approximate polygon solution with minimal cost as the final approximate polygon solution. This step takes $O(mn^2+m)$ time.

We thus have the following result:

Theorem 5. *The first pass of our proposed CPA algorithm can be done in* $O(Fn+mn^2)$ $(=O(Fn+n+mn^2+m))$ *time, where* $F \ll n^2$ *and* $m(\ll n)$ *is rather small empirically.*

Note that under the same LISE criterion, the set of polygonal segments determined by the first pass of our proposed algorithm is minimal and is the same as that obtained by the currently published CPA algorithm [1]. However, the time complexity required in the first pass of our proposed algorithm is $O(Fn+mn^2)(\ll n^3)$ and is less than the previous one by Chung et al. [1] which needs $O(n^3)$ time.

4.2. Second-pass of the proposed CPA algorithm

In the second pass, the curvature constraint associated with the longest vertical distance is used to refine the temporary approximate polygon solution obtained in the first pass in which these n' segments are denoted by the ordered set $TP=\langle S_1,S_2,\ldots,S_{n'}\rangle=\langle\overline{P_{i_1}P_{j_1}},\overline{P_{i_2}P_{j_2}},\ldots,\overline{P_{i_{n'}}P_{j_{n'}}}\rangle$, $1\leqslant i_1<j_1=i_2<j_2=i_3<j_3\ldots<i_{n'}<j_{n'}$. We pick up the first segment $S_1=\overline{P_{i_1}P_{j_1}}$. By Definition 1, the 1-cosine values of P_k's for $i_1+1\leqslant k\leqslant j_1-1$ are computed. If the 1-cosine value of point P_k is greater than the given curvature constraint T_{CUR}, the point P_k is set to be a high-curvature point. Besides adopting the curvature constraint to preserve the peak information of the high-curvature point, we also consider the vertical distance from each high-curvature point P_k to the first segment S_1 in TP. For segment S_1, based on these calculated vertical distances, we select the high-curvature point P_k with the longest vertical distance as the key point (see Fig. 5(a)) and the two new segments $\overline{P_{i_1}P_k}$ and $\overline{P_kP_{j_1}}$ are constructed to replace the temporary segment $\overline{P_{i_1}P_{j_1}}$. For example, in Fig. 5(b), if we only select the high-curvature point P_k as the key point, the two segments $\overline{P_{i_1}P_{k'}}$ and $\overline{P_{k'}P_{j_1}}$ are newly constructed. It is observed that the LISE value of Fig. 5(a) is less than the LISE value of Fig. 5(b). On the other hand, besides considering the LISE criterion in the first pass, taking both the curvature and the longest vertical distance between the high-curvature point to the concerned segment into account could generate better polygonal approximation quality. This is why we consider the LISE criterion, the curvature constraint, and the longest vertical distance consideration in our proposed two-pass CPA algorithm.

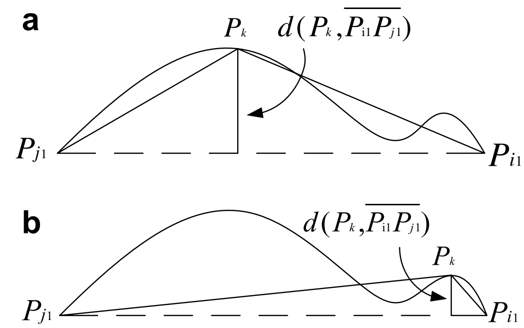

Fig. 5. Taking both the curvature constraint and the longest vertical distance consideration into account. (a) The high-curvature point P_k with the longest vertical distance $d(P_k,\overline{P_{i_1}P_{j_1}})$. (b) The highest-curvature point $P_{k'}$ with the vertical distance $d(P_k,\overline{P_{i_1}P_{j_1}})$ which is not the longest.

Initially, we set $q=1$. The second pass of our proposed CPA algorithm consists of the following four Steps:

Step 1. Consider the approximate segment $S_q=\overline{P_{i_q}P_{j_q}}$. By Definition 1, compute the *1*-cosine values of P_k's for $i_q+1\leqslant k\leqslant j_q-1$. If the curvature of each point P_k is greater than the curvature constraint T_{CUR}, it is set to be a high-curvature point.

Step 2. For each determined high-curvature point P_k in Step 1, we calculate the vertical distance $d(P_k,\overline{P_{i_1}P_{j_1}})$.

Step 3. Select the high-curvature point P_k with the longest vertical distance as the key point and the two new segments $\overline{P_{i_q}P_k}$ and $\overline{P_kP_{j_q}}$ are constructed to replace the temporary segment $\overline{P_{i_q}P_{j_q}}$.

Step 4. $q=q+1$. Repeat Steps 1–4 until $q=n'$.

5. Experimental results

In this section, two testing images, the Italy image and the French image are first used to compare the performance between our proposed two-pass CPA algorithm and the previous CPA algorithm by Chung et al. [1]. The two testing images are illustrated in Fig. 6. Both algorithms are implemented by using Borland C++ Builder 6.0 language and the Pentium 4 3.2 GHz PC with 512 MB RAM. Besides such a comparison, based on the other testing images, the comparison with the currently published CPA algorithms [20,16] is also demonstrated later.

From Fig. 7, it is observed that our proposed two-pass CPA algorithm has better polygonal approximation quality. For each testing image, the quality of the four subpolygonal approximations (denoted by four circles) obtained by using our proposed CPA algorithm is superior to the one by using the previous CPA algorithm by Chung et al. [1]. From Table 1, it is observed that our proposed two-pass CPA algorithm also has better execution-time performance in terms of seconds when compared to the

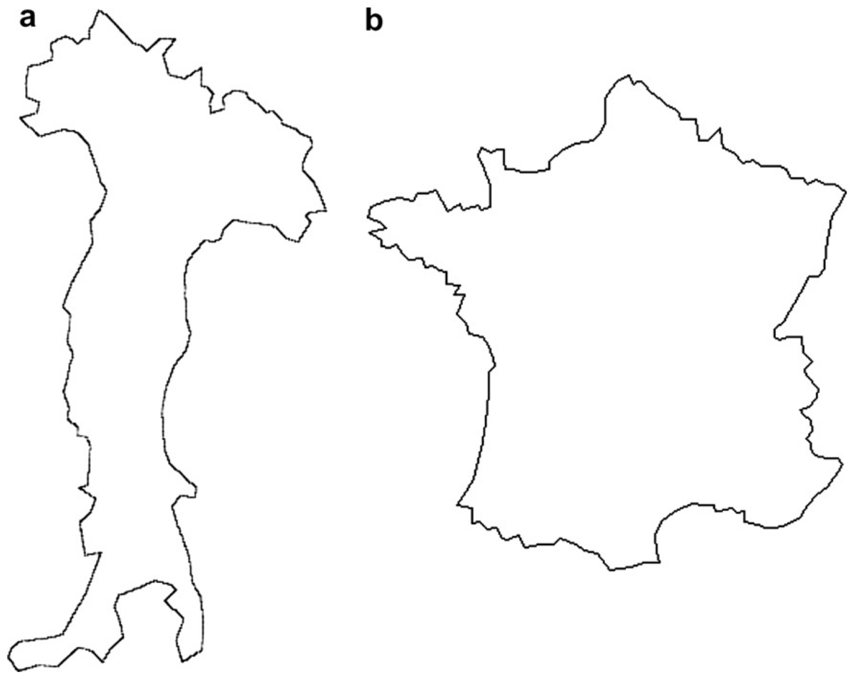

Fig. 6. Two testing images for comparing the performance between the previous CPA algorithm by Chung et al. [1] and our proposed CPA algorithm. (a) Italy image. (b) French image.

previous algorithm by Chung et al. This confirms the time complexity analysis. The average execution-time improvement ratio of our proposed CPA algorithm over the previous algorithm by Chung et al. is 97.5% ($=\frac{T_{\text{Prev}}-T_{\text{Ours}}}{T_{\text{Prev}}}$) where T_{Prev} denotes the execution-time required in the previous algorithm by Chung et al. [1] and T_{Ours} denotes the execution-time required in our proposed two-pass CPA algorithm. Consequently, our proposed two-pass CPA algorithm has the quality and execution-time advantages when compared to the previous CPA algorithm by Chung et al. [1].

Based on the same testing images used in [20], namely the semicircle image (see Fig. 8(a)) and the chromosome image (see Fig. 8(b)), Fig. 9 is used to demonstrate the polygonal approximation quality comparison between the dominant point-based CPA algorithm [20] and our proposed two-pass CPA algorithm. For fair comparison, the relevant thresholds are tuned to guarantee that the number of approximate segments required in each concerned CPA algorithm for one testing image is equivalent. In the implementation of Wu's CPA algorithm, when the curvature constraint threshold T_{CUR} is set to -0.8, the number of required approximate segments is 27(17) for Fig. 9(a) and (c). In the implementation of our proposed two-pass CPA algorithm, when the specified threshold T_{LISE} is set to 40 and the curvature constraint threshold T_{CUR} is set to -0.63, the number of required approximate segments is 27 for Fig. 9(b); when the specified threshold T_{LISE} is set to 120 and the curvature constraint threshold T_{CUR} is set to -0.4, the number of required approximate segments is 17 for Fig. 9(d). From the quality comparison between Fig. 9(a) and (b) and the quality comparison between Fig. 9(c) and (d), it is observed that under the same number of approximate segments used, our proposed two-pass CPA algorithm has better polygonal approximation quality when compared to Wu's CPA algorithm, but takes about four times the execution-time required by Wu.

Based on the above four testing images, finally we compare the performance between our proposed algorithm and the recently published algorithm by Sarfraz et al. Fig. 10 is used to demonstrate the polygonal approximation quality comparison between the Sarfraz et al.'s CPA algorithm [16] and our proposed CPA algorithm. In the implementation of Sarfraz's CPA algorithm, when the threshold value ε is set to 0.18, 0.22, 0.47, and 0.27, the numbers of required approximate segments are 82, 70, 27, and 17 for Fig. 10(a), (c), (e) and (g). From the quality comparison between Fig. 10(a), (b) and 10(c–h), it is observed that under the same number of approximate segments used, our proposed two-pass CPA algorithm has better polygonal approximation quality when compared to Sarfraz et al.'s CPA algorithm, but takes about 4.3 times the execution-time required by Sarfraz et al.

6. Conclusions

The proposed novel two-pass $O(Fn + mn^2)$-time algorithm for solving the CPA problem has been presented where n denotes the number of points in the given closed curve; $m(\ll n)$ denotes the minimal number of covering feasible segments for point and empirically the value of m is rather small; $F(\ll n^2)$ denotes the number of feasible approximate segments. The four testing images in Section 5 with their n, F, and m, respectively, are listed in Table 2 and they demon-

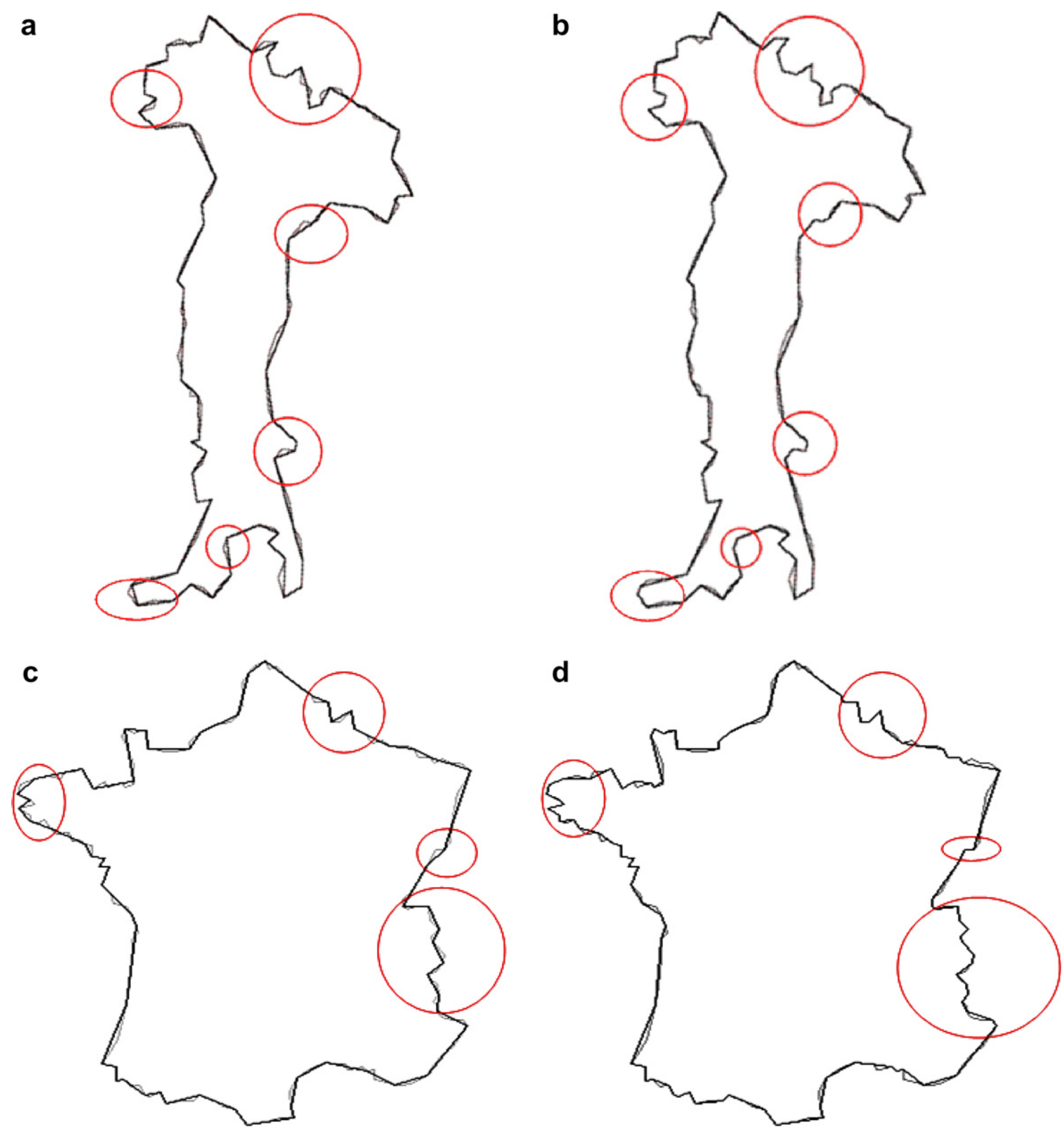

Fig. 7. Polygonal approximation quality comparison between the previous CPA algorithm by Chung et al. [1] and our proposed two-pass CPA algorithm for Italy image and French image. (a) The approximation result using the previous algorithm by Chung et al. [1] for Italy image ($T_{\text{LISE}} = 20$, time = 0.3612 s). (b) The approximation result using our proposed algorithm for Italy image ($T_{\text{LISE}} = 20$, $T_{\text{CUR}} = -0.8$, time = 0.0102 s). (c) The approximation result using the previous algorithm by Chung et al. [1] for French image ($T_{\text{LISE}} = 20$, time = 0.4609 s). (d) The approximation result using our proposed algorithm for French image ($T_{\text{LISE}} = 20$, $T_{\text{CUR}} = -0.8$, time = 0.0112 s).

Table 1
Execution-time (in seconds) comparison between the previous CPA algorithm by Chung et al. [1] and our proposed two-pass CPA algorithm for Italy and French images

	Previous algorithm by Chung et al. [1]		Proposed algorithm	
	Italy image (s)	French image (s)	Italy image (s)	French image (s)
$T_{\text{LISE}} = 20$	0.3612	0.4609	0.0102	0.0112
$T_{\text{LISE}} = 30$	0.3602	0.4606	0.0092	0.0107
$T_{\text{LISE}} = 40$	0.3600	0.4604	0.0090	0.0105
Average execution-time improvement ratio			97.37%	97.64%

strate the applicability of our proposed algorithm. According to the concept of covering segment set for each point, we speed up the first pass of our proposed CPA algorithm under the given LISE criterion. In the second pass of our proposed algorithm, we further consider the curvature constraint and the longest vertical distance consideration to

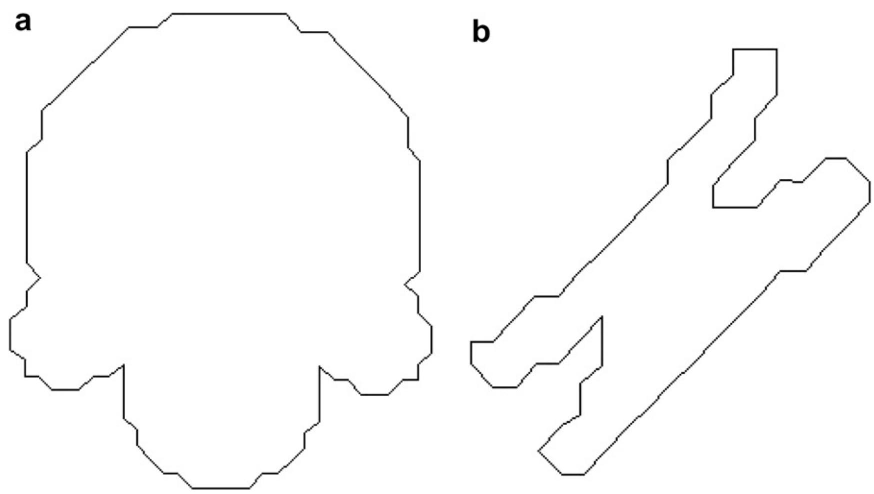

Fig. 8. Two testing images for evaluating the quality performance between the previous CPA algorithm by Wu [20] and our proposed CPA algorithm. (a) Semicircle image. (b) Chromosome image.

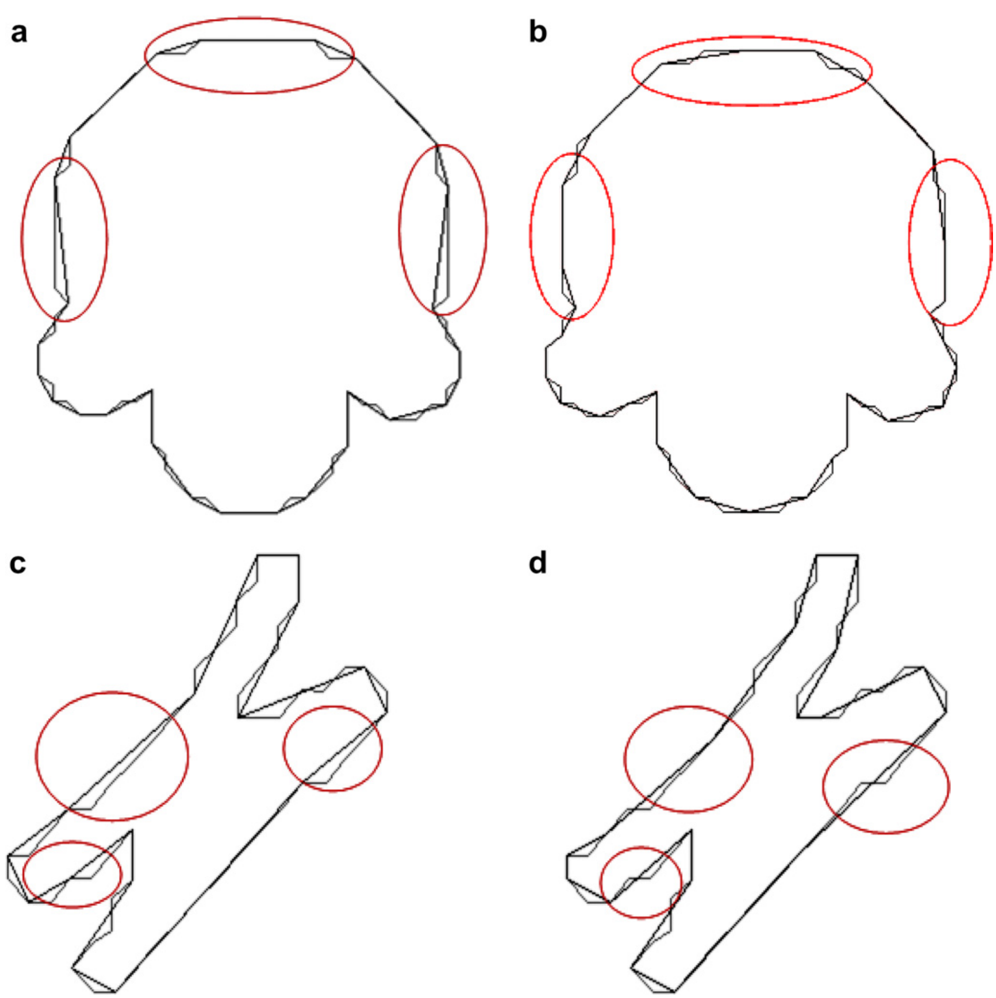

Fig. 9. Quality comparison between the previous dominant point-based CPA algorithm by Wu [20] and our proposed two-pass CPA algorithm for semicircle image and chromosome image. (a) Polygonal approximation result using the previous algorithm by Wu [20] for semicircle image ($T_{CUR} = -0.8$, time = 0.0024 s). (b) Polygonal approximation result using our proposed algorithm for semicircle image ($T_{CUR} = -0.63$, $T_{LISE} = 40$, time = 0.0062 s). (c) Polygonal approximation result using the previous algorithm by Wu [20] for chromosome image ($T_{CUR} = -0.8$, time = 0.0010 s). (d) Polygonal approximation result using our proposed algorithm for chromosome image ($T_{CUR} = -0.4$, $T_{LISE} = 120$, time = 0.0042 s).

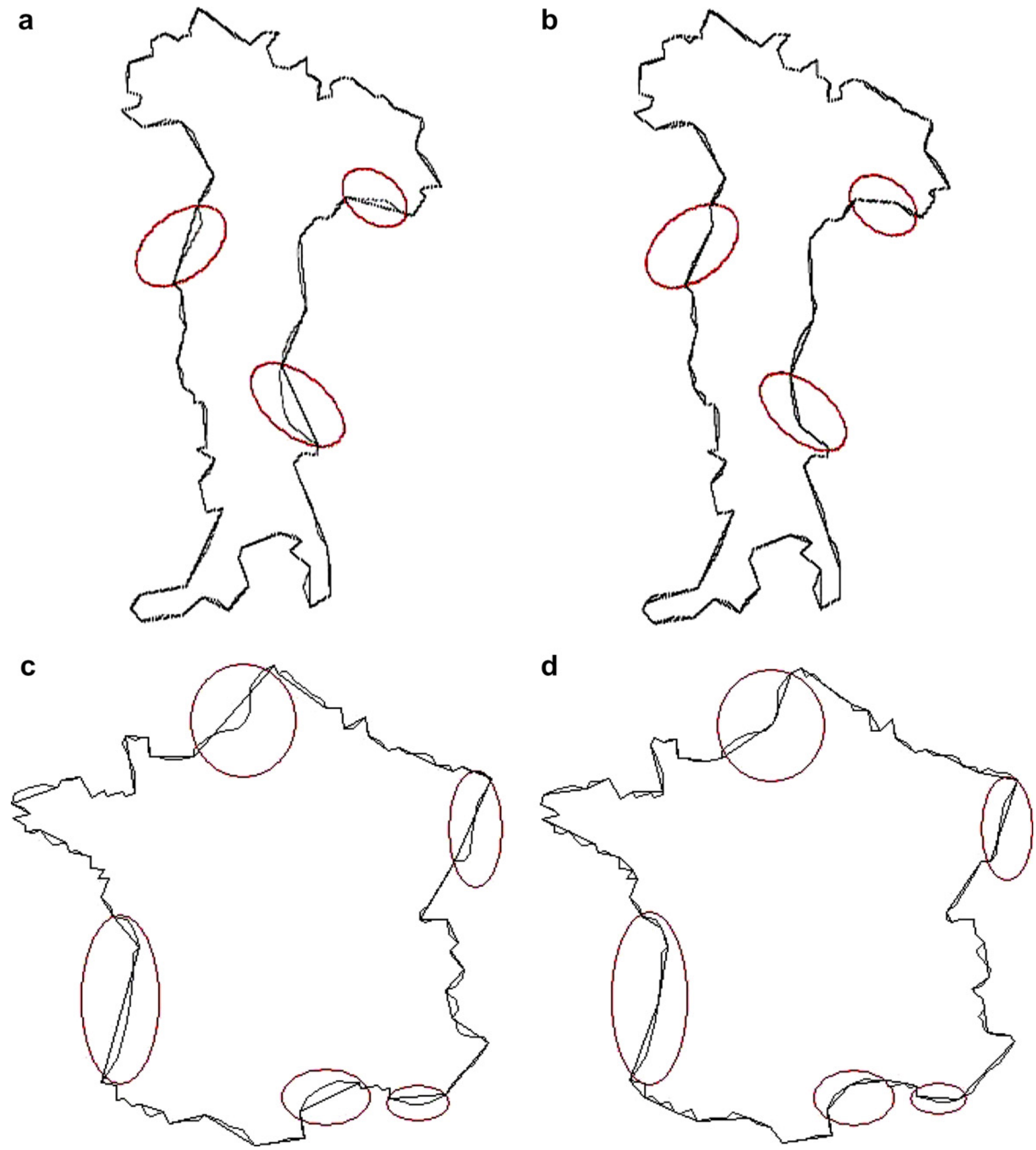

Fig. 10. Quality comparison between Sarfraz et al.'s CPA algorithm and our proposed two-pass CPA algorithm for four testing image. (a) Polygonal approximation result using the previous algorithm by Safraz et al. [16] for Italy image ($\varepsilon = 0.18$, time = 0.002 s). (b) Polygonal approximation result using our proposed algorithm for Italy image ($T_{\mathrm{CUR}} = -0.8$, $T_{\mathrm{LISE}} = 20$, time = 0.0102 s). (c) Polygonal approximation result using the previous algorithm by Safraz et al. [16] for French image ($\varepsilon = 0.22$, time = 0.0041 s). (d) Polygonal approximation result using our proposed algorithm for French image ($T_{\mathrm{CUR}} = -0.8$, $T_{\mathrm{LISE}} = 45$, time = 0.02 s). (e) Polygonal approximation result using the previous algorithm by Safraz et al. [16] for semicircle image ($\varepsilon = 0.47$, time = 0.0019 s). (f) Polygonal approximation result using our proposed algorithm for semicircle image ($T_{\mathrm{CUR}} = -0.63$, $T_{\mathrm{LISE}} = 40$, time = 0.0062 s). (g) Polygonal approximation result using the previous algorithm by Safraz et al. [16] for chromosome image ($\varepsilon = 0.27$, time = 0.0014 s). (h) Polygonal approximation result using our proposed algorithm for chromosome image ($T_{\mathrm{CUR}} = -0.4$, $T_{\mathrm{LISE}} = 120$, time = 0.0042 s).

Table 2
Four testing images ($T_{\mathrm{LISE}} = 200$) with their n, F, and m

	Italy image	French image	Semicircle image	Chromosome image
n	151	181	99	60
F	680	1212	566	256
m	2	3	3	2

find the key point for refining each temporary approximate segment obtained in the first pass. Experimental results demonstrate that under the same number of segments, our proposed two-pass algorithm has better quality and execution-time performance when compared to the previous algorithm by Chung et al. [1]. Under the same number of segments, our

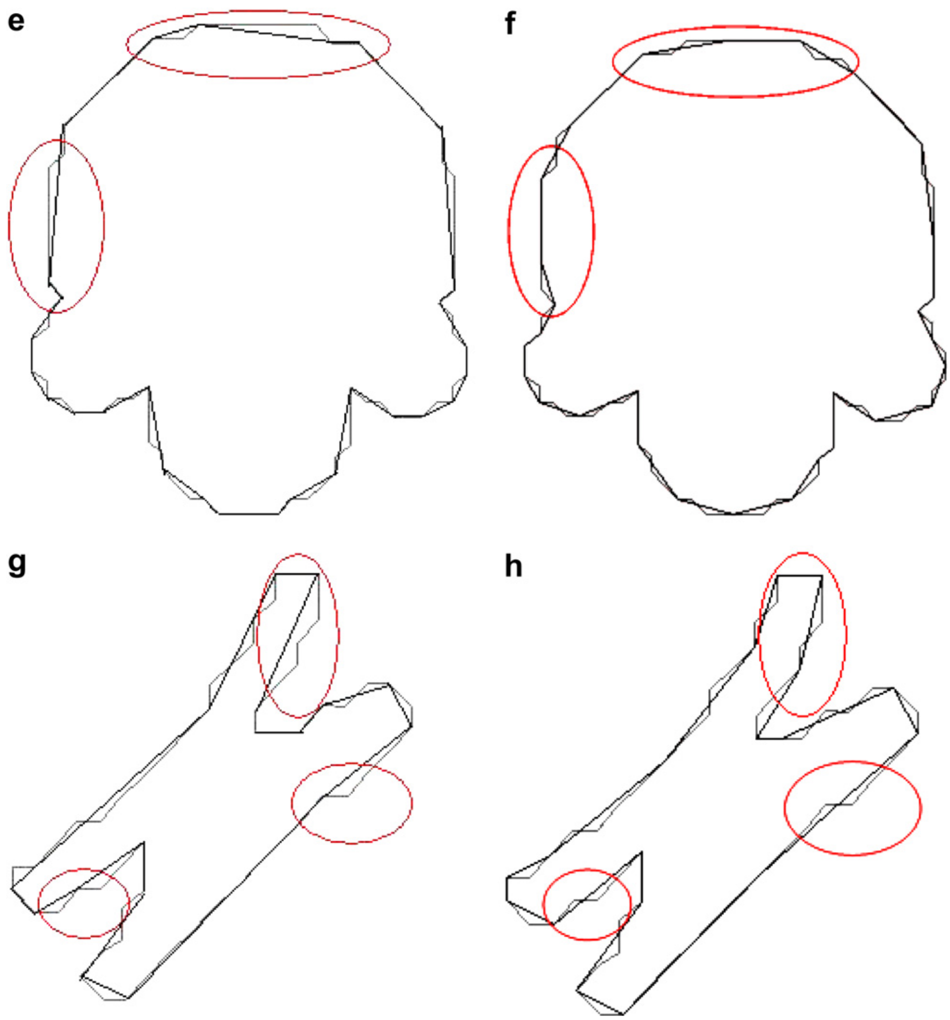

Fig. 10 (*continued*)

proposed two-pass algorithm has better quality, but has some execution-time degradation when compared to currently published algorithms by Wu [20] and Sarfraz et al. [16]. It is still a challenge problem to design efficient CPA algorithm with a good tradeoff between the quality and the execution-time.

References

[1] K.L. Chung, W.M. Yan, W.Y. Chen, Efficient algorithms for 3-D polygonal approximation based on LISE criterion, Pattern Recognition 35 (2002) 2539–2548.

[2] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, Section 25.2: Dijkstra's Algorithm, The MIT Press, Cambridge,MA, 1990.

[3] J.G. Dunham, Optimum uniform piecewise linear approximation of planar curves, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1986) 67–75.

[4] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Section 11:1.2: Polygonal Approximations, second ed., Prentice Hall, New York, 2002.

[5] J.H. Horng, J.T. Li, An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves, Pattern Recognition Letters 23 (2002) 171–182.

[6] J. Hu, H. Yan, Polygonal approximation of digital curves based on the principles of perceptual organization, Pattern Recognition 30 (1997) 701–718.

[7] S.C. Huang, Y.N. Sun, Polygonal approximation using genetic algorithms, Pattern Recognition 32 (1999) 1409–1420.

[8] C.L. B Jordan, T. Ebrahimi, M. Kunt, Progressive content-based shape compression for retrieval of binary images, Computer Vision and Image Understanding 71 (1998) 198–212.

[9] Y. Mayster, M.A. Lopez, Approximating a set of points by a step function, Journal of Visual Communication and Image Representation 17 (2006) 1178–1189.

[10] U. Montanari, A note on minimal length polygonal approximation to a digitized curve, Communications of the ACM 13 (1970) 41–47.

[11] A. Pikaz, I. Dinstein, Optimal polygonal approximation of digital curves, Pattern Recognition 28 (1995) 373–379.

[12] J.C. Perez, E. Vidal, Optimum polygonal approximation of digitized curves, Pattern Recognition Letters 15 (1994) 743–750.

[13] P.L. Rosin, Techniques for assessing polygonal approximations of curves, IEEE Transactions on Pattern Analalysis and Machine Intelligence 19 (6) (1997) 659–666.

[14] B.K. Ray, K.S. Ray, A non-parametric sequential method for polygonal approximation of digital curves, Pattern Recognition Letters 15 (1994) 161–167.

[15] A. Rosenfeld, E. Johnston, Angle detection on digital curves, IEEE Transactions on Computer C-22 (1973) 875–878.

[16] M. Sartfraz, M.R. Asim, A. Masoos, Piecewise polygonal approximation of digital curves, Proc. of the Eighth International Conf. on Information Visualisation, IEEE Computer Society, London, UK, 14–16 July 2004.
[17] Y. Sato, Piecewise linear approximation of planar curves by perimeter optimization, Pattern Recognition 25 (12) (1992) 1535–1543.
[18] C.H. Teh, R.T. Chin, On the detection of dominant points on digital curves, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (8) (1989) 859–872.
[19] W.Y. Wu, M.J. Wang, Detecting the dominant points by the curvature-based polygonal approximation, CVGIP: Graphical Model Image Process 55 (1993) 79–88.
[20] W.Y. Wu, An adaptive method for detecting dominant points, Pattern Recognition 36 (2003) 2231–2237.
[21] Y. Zhu, L.D. Seneviratne, Optimal polygonal approximation of digitized curves, IEE Proc. Vision Image and Signal Processing 144 (1997) 8–14.