Vol.16 No.6 J. Comput. Sci. & Technol. Nov. 2001

CIM Algorithm for Approximating Three-Dimensional Polygonal Curves

YONG Junhai (雍俊海), HU Shimin (胡事民) and SUN Jiaguang (孙家广)

National CAD Engineering Center, Tsinghua University, Beijing 100084, P.R. China
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P.R. China

E-mail: shimin@tsinghua.edu.cn

Received November 29, 1999; revised July 17, 2000.

Abstract The polygonal approximation problem is a primary problem in computer graphics, pattern recognition, CAD/CAM, etc. In R^2, the cone intersection method (CIM) is one of the most efficient algorithms for approximating polygonal curves. With CIM Eu and Toussaint, by imposing an additional constraint and changing the given error criteria, resolve the three-dimensional weighted minimum number polygonal approximation problem with the parallel-strip error criterion (PS-WMN) under L_2 norm. In this paper, without any additional constraint and change of the error criteria, a CIM solution to the same problem with the line segment error criterion (LS-WMN) is presented, which is more frequently encountered than the PS-WMN is. Its time complexity is $O(n^3)$, and the space complexity is $O(n^2)$. An approximation algorithm is also presented, which takes $O(n^2)$ time and $O(n)$ space. Results of some examples are given to illustrate the efficiency of these algorithms.

Keywords polygonal curve, CIM, LS-WMN, approximation, optimization

1 Introduction

Polygonal curves play an important role in many areas. Optimization and approximation algorithms for polygonal curves are widely used. In computer graphics and image processing, these algorithms can be used to obtain skeletons or outlines of objects[1−4]. In pattern recognition, these algorithms can smooth away noise and help analyzing characteristic features[5]. In CAD (Computer-Aided Design), the polygonal curve is a basic geometric entity. In CAM (Computer-Aided Manufacturing), these algorithms reduce the line segments used in CNC (Computerized Numerical Control) paths so as to reduce production cost. In communication, these algorithms can compress data and decrease the transmission time requirements[4]. Above all, almost all kinds of curves have to be converted into polygonal curves before being displayed on screens, and these algorithms can be applied to reduce computer memory requirements and to speed up display[4].

An arbitrary polygonal curve, which can be represented as $P = [p_1, p_2, \dots, p_n]$, is made up of a chain of line segments $[p_1, p_2], [p_2, p_3], \dots, [p_{n-1}, p_n]$. Let each vertex p_i of P have a weight ε_i which indicates the error tolerance of p_i. The polygonal approximation problem is to find an approximating polygonal curve $P' = [p'_1, p'_2, \dots, p'_m]$, such that:

i) The sequence of vertices of P' is a subsequence of the sequence of vertices of P, and the two end vertices of P' are often coincident with those of P, i.e., $p'_1 = p_1$ and $p'_m = p_n$;

ii) Any line segment $[p'_A, p'_{A+1}]$ of P', which substitutes a subchain $[p_B, p_{B+1}, \dots, p_C]$ of P, should satisfy $p'_A = p_B, p'_{A+1} = p_C$, and the given error criteria $EC(p_i, \overline{p'_A p'_{A+1}}) \le \varepsilon_i$, $\forall i$, $B < i < C$;

iii) m is minimum over all approximating polygonal curves that satisfy i) and ii).

Here, ε_i can be zero, which means that p_i is a vertex of P'. The formulae $p'_1 = p_1$ and $p'_m = p_n$ ensure that P' retain some properties of P, e.g., open or close. Constraint i) is useful for vertices of the approximating curve P' are exactly on P. This property can be used to reduce the accumulative

This research was supported by the National Natural Science Foundation of China (No.69902004) and NKBRSF (No.G1998030600).

error. The given formula for $EC(p_i, \overline{p'_A p'_{A+1}})$ determines how the error tolerance is measured, and leads to different approximation algorithms.

Miyaoku and Harada have classified the polygonal approximation problems[6]. This paper discusses LS-WMN problem with L_2 in R^3 (the three-dimensional weighted minimum number polygonal approximation problem with the line segment error criterion under L_2 norm). A brute force algorithm of this problem is as follows:

1) Build the directed graph $G(V, E)$ where V consists of the vertices of P, and E consists of the directed edge that satisfies $EC(p_k, \overline{p_i p_j}) \leq \varepsilon_k$, for all k, $i < k < j$. The number of edges in $G(V, E)$ is $O(n^2)$. This method for generating $G(V, E)$ requires checking the relationship among every three vertices in P, so it requires $O(n^3)$ time.

2) Find the shortest path from p_1 to p_n in $G(V, E)$, which requires $O(n^2)$ time using forward dynamic programming.

The CIM (Cone Intersection Method) is similar to the brute force algorithm except for the procedure of building $G(V, E)$. The basic idea is to check as few vertices as possible. The CIM uses the technique of the cone intersections to check the given error criteria $EC(p_k, \overline{p_i p_j}) \leq \varepsilon_k$, for all k, $i < k < j$. The difficulties are:

(a) how to represent and obtain the cones, and

(b) how to represent and obtain the intersections of these cones.

Eu and Toussaint[1] thought that these two problems in R^3 were too difficult while dealing with PS-WMN problem with L_2 in R^3 (the three-dimensional weighted minimum number polygonal approximation problem with parallel-strip error criterion under L_2 norm). Thus they imposed an additional constraint on the problem, i.e., P should be strictly monotonic along the x-axis, and changed the error tolerance region of the vertices from balls

$$ER(p_i, \varepsilon_i) = \{q | D(p_i, q) \leq \varepsilon_i\},$$

into planar discs

$$ER_1(p_i, \varepsilon_i) = \{q | D(p_i, q) \leq \varepsilon_i \text{ and } X(q) = X(p_i)\},$$

where $D(p_i, q)$ is the distance from p_i to q, $X(p_i)$ and $X(q)$ are the x-coordinates of p_i and q respectively. Then, these planar discs are projected to a common plane, and the results are still planar discs. The intersection of these projective planar discs is used in place of that of the corresponding three-dimension cones. Certainly, the planar discs are much smaller than the balls, and the resulting polygonal curve in general contains more vertices than that of the optimal result. Furthermore, the CIM of Eu and Toussaint cannot be widely used because of the additional constraint.

Without increasing the time, space and implementation complexities, the method in the paper removes the additional constraint and the substitution of the error tolerance criteria. Since the LS-WMN problem is much more common than the PS-WMN problem, we discuss the LS-WMN problem with L_2 in R^3. In fact the PS-WMN problem can also be solved and the answers to the above two questions are presented in this paper.

In Section 5, we also present an approximation algorithm which takes $O(n^2)$ time and $O(n)$ space. In the penultimate section, the results of some examples are given to illustrate the efficiency of these algorithms.

2 Definitions and Notations

The problem discussed in this paper is the LS-WMN problem with L_2 in R^3. Suppose p, q, s, v, st, ed and $c_b \in R^3$, r and $r_b \in R$. Some notations used in this paper are defined as follows:

$X(p)$, $Y(p)$ and $Z(p)$ are the x-coordinate, y-coordinate and z-coordinate of p respectively.

$D(p, q) = \sqrt{(X(p) - X(q))^2 + (Y(p) - Y(q))^2 + (Z(p) - Z(q))^2}$ is the distance between p and q.

$L(p, q) = \{(1 - t)p + tq | t \in R\}$ is a line passing through p and q. When p and q are coincident, $L(p, q)$ is a set containing only one point.

$HL(p, q) = \{(1 - t)p + tq | t \geq 0, t \in R\}$ is a ray that starts from p and passes through q. When p and q are coincident, $HL(p, q)$ degenerates into a one-point set.

$[p,q] = \{(1-t)p + tq | 0 \le t \le 1, t \in R\}$ is a line segment with p and q as the end points. When p and q are coincident, $[p,q]$ is constituted by only one point.

$D(s, L(p,q)) = \min\{D(s,w) | w \in L(p,q)\}$ is the distance between the point s and the line $L(p,q)$.

$D(s, HL(p,q)) = \min\{D(s,w) | w \in HL(p,q)\}$ is the distance between the point s and the ray $HL(p,q)$.

$D(s, [p,q]) = \min\{D(s,w) | w \in [p,q]\}$ is the distance between s and the line segment $[p,q]$. $EC(p_k, \overline{p_i p_j})$ is equal to $D(p_k, [p_i, p_j])$ in this paper.

$B(p,r) = \{q | (X(p) - X(q))^2 + (Y(p) - Y(q))^2 + (Z(p) - Z(q))^2 = r^2\}$ is the surface of a ball with the center point p and the radius r.

$BS(p,r) = \{q | (X(p) - X(q))^2 + (Y(p) - Y(q))^2 + (Z(p) - Z(q))^2 \le r^2\}$ is a solid ball with the center point p and the radius r.

$CB(p,r) = \{w | D(w, HL(c_b, p)) = r,\ w \in B(c_b, r_b),\ w \in R^3\}$ is a circle on the ball $B(c_b, r_b)$ if $p \in B(c_b, r_b)$.

Arc$B(p, r, st, ed)$ is an arc on the ball $B(c_b, r_b)$, which lies on the circle $CB(p,r)$, starting from st and ending at ed in counterclockwise order according to the direction $(p - c_b)$, if $CB(p,r)$ is a circle on $B(c_b, r_b)$.

$DB(p,r) = \{w | D(w, HL(c_b, p)) \le r,\ w \in B(c_b, r_b),\ w \in R^3\}$ is a disc on the ball $B(c_b, r_b)$ if $p \in B(c_b, r_b)$.

$OC(p,q,r) = \{s | w \in HL(p,s),\ w \in BS(q,r)\}$ is called an open cone. Note that $OC(p,q,r)$ is equal to R^3 if $D(p,q) \le r$.

$ER(p_i, \varepsilon_i) = \{w | D(p_i, w) \le \varepsilon_i,\ w \in R^3\}$ is the error tolerance region at p_i, where p_i is a vertex of the polygonal curve P, and ε_i is the weight (i.e. the error tolerance) of p_i.

$PP(p,q) = \{w | q \in HL(p,w),\ w \in B(p,1)\}$ is the set containing the projection point of q on the unit ball $B(p,1)$. Note that if p and q are coincident, $PP(p,q)$ is the whole unit ball $B(p,1)$.

$PR(p,q,r) = OC(p,q,r) \cap B(p,1)$ is the projection of the open cone $OC(p,q,r)$ on the unit ball $B(p,1)$.

3 Theories of the Algorithm

Theorem 1. *Let $[p_A, p_B, p_C]$ be a subchain of P. $[p_A, p_B, p_C]$ can be approximated by $[p_A, p_C]$ if and only if $p_C \in OC(p_A, p_B, \varepsilon_B)$ and $p_A \in OC(p_C, p_B, \varepsilon_B)$.*

Proof. According to the definition of polygonal approximation problem, $[p_A, p_B, p_C]$ can be approximated by $[p_A, p_C]$ if and only if $D(p_B, [p_A, p_C]) \le \varepsilon_B$, i.e., $[p_A, p_C] \cap BS(p_B, \varepsilon_B) \ne \phi$. Since $[p_A, p_C] = HL(p_A, p_C) \cap HL(p_C, p_A)$, $[p_A, p_C] \cap BS(p_B, \varepsilon_B) \ne \phi$ if and only if $HL(p_A, p_C) \cap BS(p_B, \varepsilon_B) \ne \phi$ and $HL(p_C, p_A) \cap BS(p_B, \varepsilon_B) \ne \phi$. According to the definition of the open cone $OC(p,q,r)$ in Section 2, $HL(p_A, p_C) \cap BS(p_B, \varepsilon_B) \ne \phi$ if and only if $p_C \in OC(p_A, p_B, \varepsilon_B)$, and $HL(p_C, p_A) \cap BS(p_B, \varepsilon_B) \ne \phi$ if and only if $p_A \in OC(p_C, p_B, \varepsilon_B)$. Therefore, $[p_A, p_B, p_C]$ can be approximated by $[p_A, p_C]$ if and only if $p_C \in OC(p_A, p_B, \varepsilon_B)$ and $p_A \in OC(p_C, p_B, \varepsilon_B)$. □

Corollary 1. *Let $[p_A, p_{A+1}, \ldots, p_B]$ be a subchain of P. $[p_A, p_{A+1}, \ldots, p_B]$ can be approximated by $[p_A, p_B]$ if and only if $p_B \in \bigcap_{i=A+1}^{B-1} OC(p_A, p_i, \varepsilon_i)$ and $p_A \in \bigcap_{i=A+1}^{B-1} OC(p_B, p_i, \varepsilon_i)$.*

Corollary 2. *Let $[p_A, p_{A+1}, \ldots, p_B, p_{B+1}, \ldots, p_C]$ be a subchain of P, and $A < B < C$. If $\bigcap_{i=A+1}^{B} OC(p_A, p_i, \varepsilon_i) = \phi$ or $\bigcap_{i=B}^{C-1} OC(p_C, p_i, \varepsilon_i) = \phi$, $[p_A, p_{A+1}, \ldots, p_C]$ cannot be approximated by $[p_A, p_C]$.*

From the two corollaries above, it is very easy to obtain the CIM in R^3. However, as mentioned in the first section, the two difficulties of CIM in R^3 remain unresolved. If we use the open cone directly, the computation will be very complex. Fortunately, we do not need to know the exact intersections of the open cones. Only the results of whether the intersections containing the points are required. Therefore, we can substitute the open cones with some discs on a unit ball or the unit ball itself. Furthermore, these discs can be obtained easily, and computing the intersections of these discs is not complex.

Theorem 2. *Let $[p_A, p_B, p_C]$ be a subchain of P. $[p_A, p_B, p_C]$ can be approximated by $[p_A, p_C]$ if and only if $PP(p_A, p_C) \subseteq PR(p_A, p_B, \varepsilon_B)$ and $PP(p_C, p_A) \subseteq PR(p_C, p_B, \varepsilon_B)$.*

Proof. $PP(p_A, p_C) \subseteq PR(p_A, p_B, \varepsilon_B)$ if and only if $HL(p_A, p_C) \subseteq OC(p_A, p_B, \varepsilon_B)$. $HL(p_A, p_C) \subseteq OC(p_A, p_B, \varepsilon_B)$ if and only if $p_C \in OC(p_A, p_B, \varepsilon_B)$. So $PP(p_A, p_C) \subseteq PR(p_A, p_B, \varepsilon_B)$ if and only if $p_C \in OC(p_A, p_B, \varepsilon_B)$. In the same way, $PP(p_C, p_A) \subseteq PR(p_C, p_B, \varepsilon_B)$ if and only if $p_A \in OC(p_C, p_B, \varepsilon_B)$. Then from Theorem 1, we can obtain Theorem 2. □

Corollary 3. *Let $[p_A, p_{A+1}, \ldots, p_B]$ be a subchain of P. $[p_A, p_{A+1}, \ldots, p_B]$ can be approximated by $[p_A, p_B]$ if and only if $PP(p_A, p_B) \in \bigcap_{i=A+1}^{B-1} PR(p_A, p_i, \varepsilon_i)$ and $PP(p_B, p_A) \in \bigcap_{i=A+1}^{B-1} PR(p_B, p_i, \varepsilon_i)$.*

Corollary 4. *Let $[p_A, p_{A+1}, \ldots, p_C]$ be a subchain of P, and $A < B < C$. If $\bigcap_{i=A+1}^{B} PR(p_A, p_i, \varepsilon_i) = \phi$ or $\bigcap_{i\in B}^{C-1} PR(p_C, p_i, \varepsilon_i) = \phi$, then $[p_A, p_{A+1}, \ldots, p_C]$ cannot be approximated by $[p_A, p_C]$.*

This solves the presentation problem of the open cones used by CIM in R^3. The following formula illuminates how to obtain these open cones.

Formula 1. $PR(p_A, p_i, \varepsilon_i)$, i.e., the projection of the open cone $OC(p_A, p_i, \varepsilon_i)$ on the unit ball $B(p_A, 1)$, can be calculated by

$$PR(p_A, p_i, \varepsilon_i) = \begin{cases} DB(q_i, e_i), & D(p_A, p_i) > \varepsilon_i, \\ B(p_A, 1), & D(p_A, p_i) \le \varepsilon_i \end{cases},$$

where $q_i = (p_i - p_A)/D(p_i, p_A) + p_A$, $e_i = \varepsilon_i / D(p_i, p_A)$, and $DB(q_i, e_i)$ is a disc on $B(p_A, 1)$.

The intersection can be represented by a close chain of arcs on the balls, denoted by ArcB defined in Section 2. Getting the intersection of the projections of these open cones is similar to the case of the planar discs in R^2.

Lemma 1. *The intersection of two different circles $CB(q_i, e_i)$ and $CB(q_j, e_j)$, which is on the same ball $B(p_A, r)$, has no more than two vertices (i.e., the joining points of the close chain of the arcs).*

Lemma 1 can be deduced from the facts that three nonlinear points determine one circle, and that three different points on the same circle are nonlinear.

Lemma 2. *$\bigcap_{i=1}^{n} DB(q_i, e_i)$, i.e., the intersection of n discs on a ball, consists of no more than $2(n-1)$ vertices (i.e., the joining points of the close chain of the arcs) if no disc is larger than a half ball.*

The method used to prove Lemma 3.2 in [1] can also be used to prove Lemma 2 in this paper.

Lemma 1 and Lemma 2 will be used to calculate the intersections of the projections of the open cones, i.e., $\bigcap_{i=A+1}^{B-1} PR(p_A, p_i, \varepsilon_i)$ and $\bigcap_{i=A+1}^{B-1} PR(p_B, p_i, \varepsilon_i)$, and illuminate the time complexity of the algorithms.

4 Optimization Algorithm

According to Corollary 3, we can find all edges of $G(V, E)$; and according to Corollary 4, we can determine when to stop working on the successive vertices while checking the error criteria. We describe the algorithm as follows. The direction of each edge in $G(V, E)$ is from the vertex with smaller subscript to that with larger subscript, thus $[p_i, p_j]$, which is used to indicate the line segment, can also be used to represent the directed edge without confusion.

Algorithm. CIM Algorithm for Approximating Three-Dimensional Polygonal Curves

Input: Arbitrary polygonal curve $P = [p_1, p_2, \ldots, p_n]$ in R^3 with associated error tolerances $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$.
Output: Approximating curve $P' = [p'_1, p'_2, \ldots, p'_m]$.

```
1. for $i = 1$ to $n - 1$ do {To build graph $G(V, E)$}
   begin
       $IPR \leftarrow B(p_i, 1)$;
       for $j = i + 1$ to $n$ and while $IPR \neq \phi$ do
       begin
           if $PP(p_i, p_j) \in IPR$ then $G \leftarrow G \cup [p_i, p_j]$
           $IPR \leftarrow IPR \cap PR(p_i, p_j, \varepsilon_j)$
       end {End of inner for-loop}
   end {End of outer for-loop}
```

```
2. for $i = n$ to 2 do {To check edges in graph $G(V, E)$}
   begin
       $IPR \leftarrow B(p_i, 1)$;
       $k \leftarrow \min\{h|[p_h, p_i] \in G\}$;
       for j= $i-1$ to 1 and while $IPR \neq \phi$ do
       begin
           if $[p_j, p_i] \in G$ and $PP(p_i, p_j) \notin IPR$ then $G \leftarrow G - [p_j, p_i]$;
           if $k \geq j$ then
               break; {Jump out of inner for-loop}
           $IPR \leftarrow IPR \cap PR(p_i, p_j, \varepsilon_j)$
           if $IPR = \phi$ then $G \leftarrow G - \{[p_h, p_i]|h < j\}$
        end {End of inner for-loop}
   end {End of outer for-loop}
3. Output the shortest path from $p_1$ to $p_n$ according to $G(V, E)$, using forward dynamic programming.
End of CIM Algorithm
```

Since *IPR* consists of at most $O(n)$ discs according to Lemma 2, testing $PP(p_i, p_j) \in IPR$ requires $O(n)$ time, and obtaining $IPR \cap PR(p_i, p_j, \varepsilon_j)$ also requires $O(n)$ time. Hence, Step 1 has $O(n^3)$ time complexity. The edges of $G(V, E)$ are obtained from Step 1, so they can be ordered. Therefore, determining $\min\{h|[p_h, p_i] \in G\}$ requires only $O(n)$ time, and obtaining $\{[p_h, p_i]|h < j\}$ requires $O(n)$ time. Hence, Step 2 has $O(n^3)$ time complexity. Since $G(V, E)$ has $O(n^2)$ edges, the time complexity of Step 3 is $O(n^2)$.

In practice, the minimization of the vertices is not always mandatory, especially if it is at the expense of requiring much more time or space. An approximation algorithm is given below; it only requires $O(n^2)$ time and $O(n)$ space.

5 Approximation Algorithm

In order to find a fast algorithm, we substitute Theorem 2, which specifies the sufficient and necessary conditions, with Theorem 3, which specifies the sufficient conditions.

Theorem 3. *Let* $[p_A, p_B, p_C]$ *be a subchain of* P. $[p_A, p_B, p_C]$ *can be approximated by* $[p_A, p_C]$ *if* $PP(p_A, p_C) \subseteq PR(p_A, p_B, \varepsilon_B)$ *and* $D^2(p_A, p_C) \geq D^2(p_A, p_B) - \varepsilon_B^2$.

Firstly, according to Theorem 3, we can omit Step 2 of the CIM algorithm in Section 4. Secondly, we do not need to search for every approximating line segment. Therefore, the graph $G(V, E)$ is not needed in the approximation algorithm. The approximation algorithm is described as follows.

Algorithm. Approximation Algorithm

```
Input: Arbitrary polygonal curve $P = [p_1, p_2, \ldots, p_n]$ in $R^3$ with associated error tolerances $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$.
Output: Approximating curve $P' = [p'_1, p'_2, \ldots, p'_m]$.
$i = 1$;
output $p_1$;
while $i < n$ do
begin
    $IPR \leftarrow B(p_i, 1)$;
    for $j = i + 1$ to $n - 1$ do
    begin
        $rt = (p_j - p_i) * (p_j - p_i) - \varepsilon_{j*}\varepsilon_j$;
        if $j = i + 1$ then
            $rr = rt$; {To set the initial value of $rr$}
        else
        begin
            if $(p_j - p_i) * (p_j - p_i) < rr$ or $PP(p_i, p_j) \notin IPR$, then begin
                output $p_{j-1}$;
                break; {Jump out of for-loop}
            end {End of if}
            if $rr < rt$, then $rr = rt$;
        end {End of else}
        $IPR \leftarrow IPR \cap PR(p_i, p_j, \varepsilon_j)$
    end {End of for-loop}
    $i \leftarrow j - 1$;
end {End of while loop}
output $p_n$;
End of Approximation Algorithm
```

The boundary of IPR consists of $O(n)$ ArcBs, so the time complexity of calculating $IPR \cap PR(p_i, p_j, \varepsilon_j)$ is $O(n)$. Although the approximation algorithm has two nested loops, its time complexity is $O(n^2)$. Because IPR requires $O(n)$ space, the space complexity of the algorithm is $O(n)$.

6 Examples

First, we use the points evenly distributed on some Bézier curves as examples, and the results can be checked by readers. The first Bézier curve has eight control points, $(0,0,0)$, $(40,240,50)$, $(100,300,100)$, $(120,40,0)$, $(160,100,0)$, $(220,30,0)$, $(240,300,300)$ and $(300,250,250)$, and is denoted by $C1$. The second one has five control points, $(0,0,350)$, $(100,400,50)$, $(200,350,0)$, $(300,0,400)$ and $(400,300,100)$, and is denoted by $C2$. The third one has nine control points, $(0,0,0)$, $(200,0,0)$, $(400,0,0)$, $(400,200,100)$, $(400,400,200)$, $(200,400,200)$, $(0,400,200)$, $(0,200,100)$ and $(0,0,0)$, and is denoted by $C3$, which cannot be approximated by the algorithm of Eu and Toussaint. The number of sampled points is denoted by M. All of the associated error tolerances have the same value, although they might be different. We have implemented four algorithms: the brute force algorithm, the algorithm of Eu and Toussaint, the optimization algorithm using CIM and the approximation algorithm using CIM. All these algorithms are implemented with Visual C++ 5.0 under Windows 98

Table 1. The Efficiency of the Optimization Algorithm Using CIM

Curve	M	ε_i	Result (vertices)	Time (ms)
$C1$	21	10	7	9
$C1$	101	1	20	130
$C1$	101	1e−1	78	25
$C1$	301	1e−2	240	70
$C1$	1001	1e−3	772	255
$C1$	5001	1e−4	2089	2620
$C2$	21	10	7	8
$C2$	101	1	20	125
$C2$	151	1e−1	68	69
$C2$	401	1e−2	245	145
$C2$	1001	1e−3	777	265
$C2$	5001	1e−4	2071	2570

Table 2. The Efficiency of the Brute Force Algorithm

Curve	M	ε_I	Result (vertices)	Time (ms)
$C1$	21	10	7	2
$C1$	101	1	20	70
$C1$	101	1e−1	78	45
$C1$	301	1e−2	240	385
$C1$	1001	1e−3	772	4191
$C1$	5001	1e−4	2089	105030
$C2$	21	10	7	3
$C2$	101	1	20	70
$C2$	151	1e−1	68	105
$C2$	401	1e−2	245	688
$C2$	1001	1e−3	777	4089
$C2$	5001	1e−4	2071	104980

Table 3. The Efficiency of the Algorithm of Eu and Toussaint

Curve	M	ε_i	Result (vertices)	Time (ms)
$C1$	21	10	10	5
$C1$	101	1	27	80
$C1$	101	1e−1	88	15
$C1$	301	1e−2	267	40
$C1$	1001	1e−3	872	151
$C1$	5001	1e−4	3030	1525
$C2$	21	10	10	5
$C2$	101	1	27	80
$C2$	151	1e−1	107	45
$C2$	401	1e−2	322	80
$C2$	1001	1e−3	882	155
$C2$	5001	1e−4	3351	1570

Table 4. The Efficiency of the Approximation Algorithm Using CIM

Curve	M	ε_i	Result (vertices)	Time (ms)
$C1$	21	10	7	0
$C1$	101	1	20	10
$C1$	101	1e−1	78	5
$C1$	301	1e−2	240	15
$C1$	1001	1e− 3	772	60
$C1$	5001	1e−4	2089	370
$C2$	21	10	7	0
$C2$	101	1	20	10
$C2$	151	1e−1	68	10
$C2$	401	1e−2	245	28
$C2$	1001	1e−3	777	60
$C2$	5001	1e−4	2071	350

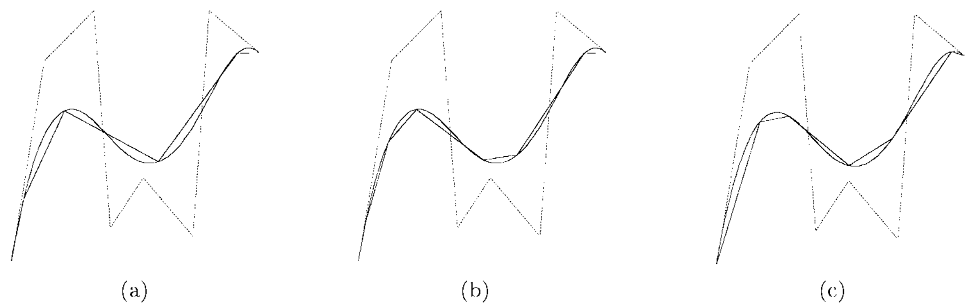

Fig.1. Approximate $C1$ ($M = 101, \varepsilon_i = 10$). (a) Result of the optimization algorithm using CIM or the brute force algorithm (5 segments). (b) Result of the algorithm of Eu and Toussaint (7 segments). (c) Result of the approximation algorithm using CIM (6 segments).

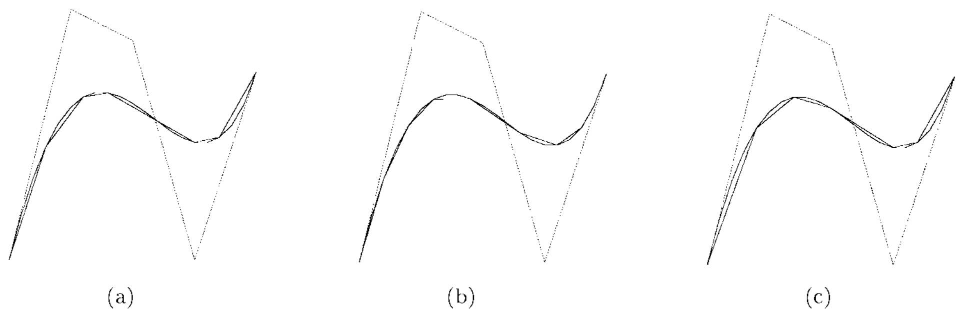

Fig.2. Approximate $C2$ ($M = 21, \varepsilon_i = 10$). (a) Result of the optimization algorithm using CIM or the brute force algorithm (6 segments). (b) Result of the algorithm of Eu and Toussaint (9 segments). (c) Result of the approximation algorithm using CIM (6 segments).

on a personal computer with Pentium II 350 CPU chip and 64M memory (Tables 1–4, Figs.1–3). The results of the brute force algorithm and the optimization algorithm using CIM are the same, but with different efficiency.

Another example is from a military project using the finite element analysis on some large-scale landing crafts. The sampled points on the crafts are preprocessed before the models of these crafts are reconstructed in computers. Fig.4 shows the shape of the boundary curve of the large-scale landing craft. Table 5 shows the preprocessing results for the boundary curve from A to B.

From the above tables, we can find that the time required by the brute force algorithm increases sharply when the number of the sampled

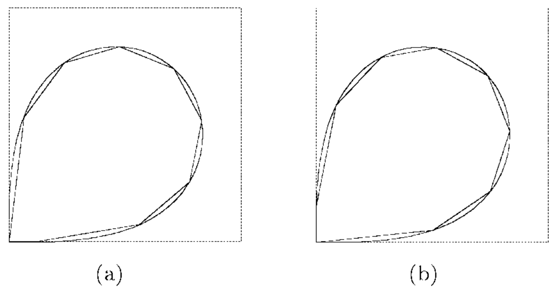

Fig.3. Approximate $C3$ ($M = 101, \varepsilon_i = 10$). Note that this curve cannot be approximated by the algorithm of Eu and Tousssaint. (a) Result of the optimization algorithm using CIM or the brute force algorithm (9 segments). (b) Result of the approximation algorithm using CIM (9 segments).

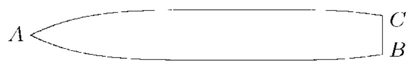

Fig.4. The boundary curve of a large-scale landing craft.

Table 5. Approximation Results of the Boundary Curve

	Optimization algorithm using CIM	Brute force algorithm	Algorithm of Eu and Toussaint	Approximation algorithm using CIM
Result (vertices)	201	201	251	201
Time (ms)	135	300	145	35

points and the number of the resulting points increase. The results of the algorithm of Eu and Toussaint have more points than the other three algorithms, since the disc used by Eu and Toussaint as the error tolerance region of any vertex is smaller than the ball used by the other three algorithms. Additionally, $C3$ cannot be approximated by the algorithm of Eu and Toussaint. According to the tables above, the results of the approximation algorithm are very close to the optimization algorithm, and its cost is much lower.

7 Conclusion

Without any additional constraint, we present the solution to the LS-WMN problem with L_2 in R^3. If the polygonal curve is used to approximate a G^1 curve, and if the minimization of vertices is not mandatory, the approximation algorithm may be preferable, for it has much lower cost and produces only a few more vertices than that of the optimization algorithm.

In this paper, we have made the CIM feasible in R^3, which means that not only a more efficient algorithm is realized, but also the CIM has been proved to be applicable in both R^2 and R^3. The differences only consist of:

1. representations of points, vectors and open cones,
2. methods of deciding whether a vertex lies in the open cone, and
3. methods of computing the intersections of the open cones (or the projections of the open cones).

This property fits well with object-oriented software, thus the codes of CIM can be shared in both R^2 and R^3.

Acknowledgements The authors thank Mr. Zhu Xiang for providing the original data of the last example in the paper. The authors acknowledge with thanks the comments and suggestions of the reviewers.

References

[1] David Eu, Godfried T Toussaint. On approximating polygonal curves in two and three dimensions. CVGIP: Graphical Models and Image Processing, 1994, 56: 231–246.

[2] Montanari U. A note on minimal length polygonal approximation to a digitized contour. *Communication of the ACM,* 1970, 13(1): 41–54.

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

[4] Williams C M. Note an efficient algorithm for the piecewise linear approximation of planar curves. *Computer Graphics and Image Processing,* 1978, 8(2): 286–293.

[5] Kurozumi Y, Davis W. Polygonal approximation by the minimax method. *Computer Graphics and Image Processing,* 1982, 19(3): 248–264.

[6] Kento Miyaoku, Koichi Harada. Note approximating polygonal curves in two and three dimensions. *Graphical Models and Image Processing,* 1998, 60: 222–225.

YONG Junhai received his Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, P.R. China in 2000. He received the B.S. degree from the Department of Computer Science and Technology, Tsinghua University, in 1996. From March to June in 2000, he was a visiting researcher in the Hong Kong University of Science and Technology. His research interests are computer aided design, computer graphics and standardization of product data.

HU Shimin is an associate professor in the Department of Computer Science and Technology, Tsinghua University, Beijing, P.R. China. He received the Ph.D. degree in 1996 from Zhejiang University, P. R. China, and finished postdoctoral research work in 1998 at Tsinghua University. His research interests are computer aided design, computer graphics, fractal geometry and its applications.

SUN Jiaguang is a professor in the Department of Computer Science and Technology, Tsinghua University, Beijing, P.R. China. His research interests are computer graphics, computer aided design, computer aided manufacturing, product data management and software engineering.