COMPUTER SCIENCE PUBLICATION

APPROXIMATION OF POLYGONAL CURVES WITH

MINIMUM NUMBER OF LINE SEGMENTS OR MINIMUM ERROR

W.S. Chan and F. Chin

Technical Report TR-93-04

March 1993

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF ENGINEERING
UNIVERSITY OF HONG KONG
POKFULAM ROAD
HONG KONG

Approximation of Polygonal Curves with Minimum Number of Line Segments or Minimum Error

W.S. Chan and F. Chin
Department of Computer Science
University of Hong Kong

Abstract

We improve the time complexities for solving the polygonal curve approximation problems formulated by Imai and Iri. The time complexity for approximating any polygonal curve of n vertices with minimum number of line segments can be improved from $O(n^2 \log n)$ to $O(n^2)$. The time complexity for approximating any polygonal curve with minimum error can also be improved from $O(n^2 \log^2 n)$ to $O(n^2 \log n)$. We further show that if the curve to be approximated forms part of a convex polygon, the two problems can be solved in $O(n)$ and $O(n^2)$ time respectively for both open and closed polygonal curves.

1 Introduction

In various situations and applications, images of a scene have to be represented at different resolutions. For pixel images, lower resolution images can be derived from higher ones through a number of multi-resolution techniques[1, 3, 4, 8, 10]. If the image is a line drawing, the problem becomes approximating the original figure with a fewer number of line segments. A number of algorithms[5, 7, 9, 11] have been devised to solve the approximation problem with different constraints and approximation criteria[5, 6, 7]. In this paper we consider the problem of approximating a piecewise linear curve or polygonal curve by another whose vertices are a subset of the original. We improve the time complexity given by Imai and Iri[7] and Melkman and O'Rourke[9] from $O(n^2)$ to $O(n^2 \log n)$ and give efficient algorithms for other variations.

Formally, let $P = (p_0, p_1, \ldots, p_{n-1})$ be a *piecewise linear curve* or *polygonal curve* on a plane, i.e., $p_0, p_1, \ldots, p_{n-1}$ is a sequence of points on a plane and each pair of points p_i and p_{i+1}, $i = 0, 1, \ldots, n-2$, are joined by a line segment (note that the line segments may intersect). Lines segment $\overline{p_r p_s}$ can be used to approximate the polygonal curve $(p_r, p_{r+1}, \ldots, p_s)$ and the *error of a line segment* can be defined as the maximum distance between the segment $\overline{p_r p_s}$ and each point p_k between p_r and p_s, i.e., $r \leq k \leq s$. The distance $d(\overline{p_r p_s}, p_k)$ between a line segment $\overline{p_r p_s}$ and a point p_k is defined to be the minimum distance between $\overline{p_r p_s}$ and p_k , i.e., $d(\overline{p_r p_s}, p_k) = \min_{x \in \overline{p_r p_s}}\{d(x, p_k)\}$ where $d(x, p_k)$ is the Euclidean distance between points x and p_k. Thus, the error of $\overline{p_r p_s}$, $e(\overline{p_r p_s})$, can be defined as $\max_{r \leq k \leq s}\{d(\overline{p_r p_s}, p_k)\}$. Furthermore, we say $P' = (p_{i_0}, p_{i_1}, \ldots, p_{i_m})$ is an *approximate curve* of P if $p_{i_0}, p_{i_1}, \ldots, p_{i_m}$ is a subsequence of $p_0, p_1, \ldots, p_{n-1}$ with $i_0 = 0$, $i_m = n-1$ and $0 \leq m < n$. The *error of an approximate curve* P' is defined as the maximum error of each line segment in P', i.e., $e(P') = \max_{0 \leq k < m}\{e(\overline{p_{i_k} p_{i_{k+1}}})\}$. We say that the *error of* P' *is within* ϵ if $e(P') \leq \epsilon$. Note that $d(\overline{p_{i_j} p_{i_{j+1}}}, p_k) \leq \epsilon$ if and only if p_k lies within the shaded area as shown in Figure 1. Under this error

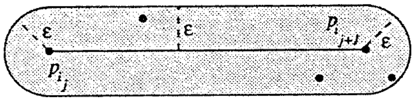

Figure 1: Definition of being within ϵ of a line segment

measurement, we can guarantee that P' will lie totally within the "band" of width 2ϵ running alongside P if $e(P') \leq \epsilon$ (Figure 2). There are two types of optimization

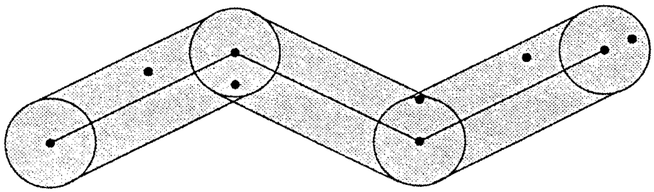

Figure 2: "Band" running alongside a polygonal curve

problems associated with the curve approximation problem.

min-# problem: Given $\epsilon \geq 0$, construct an approximate curve with error within ϵ and having the minimum number of line segments.

min-ϵ problem: Given m, construct an approximate curve consisting of at most m line segments with minimum error.

In this paper, we show that the min-# problem for an open polygonal curve can be solved in $O(n^2)$ time, improving the previous result of $O(n^2 \log n)$ time[7]. We further show that if the polygonal curve forms part of a convex polygon, the min-# problem can be solved in $O(n)$ time for both open and closed polygonal curves. The min-ϵ problem can be solved in $O(n^2 \log n)$ time for a general open polygonal curve, improving the previous result of $O(n^2 \log^2 n)$[7], $O(n^3 \log n)$ time for a general closed polygonal curve, and in $O(n^2)$ time for both open and closed convex curves. The results are summarized in Table 1.

	general		convex	
	open	closed	open	closed
min-#	$O(n^2)$	$O(n^3)$	$O(n)$	$O(n)$
min-ϵ	$O(n^2 \log n)$	$O(n^3 \log n)$	$O(n^2)$	$O(n^2)$

Table 1: Summary of the time complexities of the polygonal approximation algorithms

2 The Min-# Problem

Given a polygonal curve P and an error bound ϵ, this problem can be solved in two steps. The first step is to construct a *directed* graph $G = (V, E)$, where each vertex v_r in V represents a point p_r in P and each edge (v_r, v_s) is in E if and only if the error of the line segment $\overline{p_r p_s}$ is within ϵ, i.e., $V = \{v_0, v_1, \ldots, v_{n-1}\}$ and

$E = \{(v_r, v_s) \mid r < s \text{ and } e(\overline{p_r p_s}) \leq \epsilon\}$. We shall call G the *ϵ-graph* of P. Note that a directed graph is used to model the problem so that it can later be extended to the closed curve case. The second step is to find the shortest path in G from v_0 to v_{n-1} with each edge of unit length. Thus, the path length will correspond to the number of line segments in the approximate curve. Moreover, the shortest path from v_0 to v_{n-1} also reveals the subset of points of P used in the approximate curve P', i.e., if $(v_{i_0}, v_{i_1}, \ldots, v_{i_m})$ is the shortest path with $i_0 = 0$ and $i_m = n - 1$, then the corresponding $P' = (p_{i_0}, p_{i_1}, \ldots, p_{i_m})$ will be the approximate curve with error $\leq \epsilon$ and the minimum number of line segments.

The brute-force method of constructing G is to check for each pair of points, p_r and p_s, whether the error of $\overline{p_r p_s}$ is within ϵ, i.e., $(v_r, v_s) \in E$. There are $O(n^2)$ pairs of points and checking the error of a line segment, corresponding to a pair of points, takes $O(n)$ time. This brute-force method takes $O(n^3)$ time. Since finding the shortest path in G takes no more than $O(n^2)$ time, the min-# problem can be solved in $O(n^3)$ time.

The critical part of the algorithm is the construction of G. Melkman and O'Rourke [9] described an $O(n^2 \log n)$ algorithm to construct G. For each pair of points, p_r and p_s, their algorithm maintains a data structure which represents a region W_{rs} such that $(v_r, v_s) \in E$ if and only if $p_s \in W_{rs}$. Their data structure allows checking of whether $p_s \in W_{rs}$ and updating W_{rs} to $W_{r,s+1}$ in $O(\log n)$ time. Thus G can be constructed in $O(n^2 \log n)$ time and the time complexity of solving the min-# problem is improved to $O(n^2 \log n)$. In this paper, we improve the complexity of constructing G to $O(n^2)$. As a result, the time complexity of the algorithm for solving the min-# problem is $O(n^2)$, an improvement from $O(n^2 \log n)$. In the following, we shall discuss how G can be constructed in $O(n^2)$ time (Section 2.1) and $O(n)$ time if the polygonal curve is convex (Section 2.2). Sections 2.3 and 2.4 will be dealing with closed polygonal curves.

2.1 General Open Polygonal Curves

Before we proceed with the algorithm, let us discuss some *necessary* conditions for $d(\overline{p_r p_s}, p_k) \leq \epsilon$ when $p_r \neq p_s$ and $r < k < s$, i.e., $(v_r, v_s) \in E$.

Condition A $d(\overleftrightarrow{p_r p_s}, p_k) \le \epsilon$ where $\overleftrightarrow{p_r p_s}$ denotes the infinite line extending at both ends of $\overline{p_r p_s}$.

Condition B If $\angle p_k p_r p_s > \pi/2$, $d(p_k, p_r) \le \epsilon$, where $\angle p_k p_r p_s$ denotes the *convex* angle between line segments $\overline{p_k p_r}$ and $\overline{p_r p_s}$(Figure 3).

Condition C If $\angle p_k p_s p_r > \pi/2$, $d(p_k, p_s) \le \epsilon$(Figure 3).

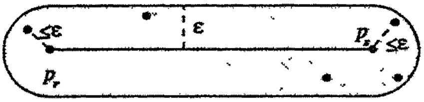

Figure 3: Conditions B and C

Note that Condition A is intuitively straightforward, while Conditions B and C ensure that if p_k lie outside the "range" of $\overline{p_r p_s}$, p_k would not be too far away from p_r or p_s. Moreover, one can easily argue that it is not possible to have both $\angle p_k p_r p_s$ and $\angle p_k p_s p_r$ greater than $\pi/2$, and satisfying these three conditions is also *sufficient* to guarantee that $d(\overline{p_r p_s}, p_k) \le \epsilon$.

Now we are ready to discuss the construction of the ϵ-graph $G = (V, E)$. Basically, G is constructed in two phases, in which $G' = (V, E')$ and $G'' = (V, E'')$ are formed. Note that G, G' and G'' have the same vertex set V and we shall show that $E = E' \cap E''$. G' is a graph generated based on Conditions A and B, while G'' on Conditions A and C.

Let $\overrightarrow{p_r p_s}$ denote the "ray" emanating from p_r and passing through p_s. If $p_r = p_s$, let $\overrightarrow{p_r p_s} = p_r$. $(v_r, v_s) \in E'$ if and only if $r < s$ and for all $r \le k \le s$, $d(\overrightarrow{p_r p_s}, p_k) \le \epsilon$, i.e., p_k lies in the area as shown in Figure 4(a). Similarly, $(v_r, v_s) \in E''$ if and only if $r < s$ and for all $r \le k \le s$, $d(\overrightarrow{p_s p_r}, p_k) \le \epsilon$, i.e., p_k lies in the area as shown in Figure 4(b).

Since segment $\overline{p_r p_s}$ is part of both $\overrightarrow{p_r p_s}$ and $\overrightarrow{p_s p_r}$, it is easy to observe that the shaded area in Figure 3 is the intersection of those shaded areas given in Figures 4(a)

(a) Area for which p_k lies within ϵ of $\overrightarrow{p_r p_s}$

(b) Area for which p_k lies within ϵ of $\overrightarrow{p_s p_r}$

Figure 4: Relationship between $\overline{p_r p_s}$ and its rays

and 4(b). Thus we have $d(\overline{p_r p_s}, p_k) \leq \epsilon$ if and only if both $d(\overrightarrow{p_r p_s}, p_k)$ and $d(\overrightarrow{p_s p_r}, p_k)$ are less than or equal to ϵ, and similarly, $E \subseteq E'$, $E \subseteq E''$ and $E = E' \cap E''$. Instead of constructing E directly, E can then be generated from E' and E''.

Now we describe the construction of E', and the construction of E'' is similar. Consider a particular vertex v_r in V. We want to determine whether $(v_r, v_s) \in E'$ for some s with $r < s$. Let $r < k < s$. If $d(p_r, p_k) > \epsilon$, let a_{rk} and b_{rk} be the two rays emanating from p_r, one at each side of p_k and at a distance ϵ from p_k, i.e., $d(a_{rk}, p_k) = d(b_{rk}, p_k) = \epsilon$, and let D_{rk} be the convex region bounded by a_{rk} and b_{rk}, including a_{rk} and b_{rk} but excluding point p_r (Figure 5). If $d(p_r, p_k) \leq \epsilon$, let D_{rk} be the whole plane. The following lemma shows that $d(\overrightarrow{p_r p_s}, p_k) \leq \epsilon$ as long as $p_s \in D_{rk}$.

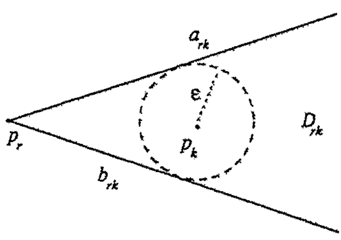

Figure 5: Definition of D_{rk} when $d(p_r, p_k) > \epsilon$

Lemma 2.1 *Assume $r \leq k < s$. $p_s \in D_{rk}$ if and only if $d(\overrightarrow{p_r p_s}, p_k) \leq \epsilon$.*

Proof: The proof is rather straightforward. Basically all cases when $d(p_r, p_k) > \epsilon$, $d(p_r, p_k) \leq \epsilon$, $p_r = p_s$ and $p_r \neq p_s$ have to be considered. □

In order to determine whether $(v_r, v_s) \in E'$, it is necessary and sufficient to check whether $p_s \in D_{rk}$ for all $r \leq k < s$ (from Lemma 2.1). Define $W_{rs} = \bigcap_{r \leq k < s} D_{rk}$ for $r < s$.

Lemma 2.2 *Assume $r < s$. $(v_r, v_s) \in E'$ if and only if $p_s \in W_{rs}$.*

Proof: $(v_r, v_s) \in E'$ iff $d(\overrightarrow{p_r p_s}, p_k) \leq \epsilon$ for all $r \leq k < s$ (from definition)
iff $p_s \in D_{rk}$ for all $r \leq k < s$ (Lemma 2.1)
iff $p_s \in W_{rs} = \bigcap_{r \leq k < s} D_{rk}$ (from definition)
□

From Lemma 2.2, we can determine whether $(v_r, v_s) \in E'$ by testing whether $p_s \in W_{rs}$. For any r, the algorithm would consider all pairs of vertices (v_r, v_s) with s ranging from $r + 1$ to n. The key part of the algorithm is to determine whether a point is in W_{rs} and to update W_{rs} from $W_{r,s-1}$ efficiently.

Since W_{rs} is an intersection of the D_{rk}'s, the D_{rk}'s and the W_{rs} are either the whole plane or a cone-shape region bounded by the two rays emanating from p_r. As long as we can keep track of these two rays(boundaries) of W_{rs}, one can easily check whether $p_s \in W_{rs}$ and update W_{rs} in constant time. Checking whether $p_s \in W_{rs}$ is equivalent to checking whether p_s lies within the two rays. As for updating W_{rs}, $W_{rs} = W_{r,s-1} \cap D_{r,s-1}$. In fact, if $W_{rs} = \phi$, we can conclude immediately that $(v_r, v_k) \notin E'$ for all $k \geq s$. Thus we can determine all the edges in E' incident from v_r in $O(n)$ time and all edges in E' in $O(n^2)$ time. The pseudo-code for constructing E is listed in Figure 6.

Theorem 1 *$G = (V, E)$ can be constructed in $O(n^2)$ time.*

Proof: As both E' and E'' can be constructed in $O(n^2)$ time, and $E = E' \cap E''$ can be generated with another n^2 operations, G can be constructed $O(n^2)$ time. □

Corollary 2.3 *The min-# problem for an open polygonal curve can be solved in $O(n^2)$ time.* □

```
algorithm A
begin
      call procedure e1 to compute E'
      compute E'' in a similar way
      output E = E' ∩ E''
end

procedure e1
begin
      for r = 0 to n - 2 do
      begin
          s := r + 1
          W := the whole plane { W = W_rs }
          while W ≠ ∅ and s < n do
          begin
                if p_s ∈ W then output (v_r, v_s) ∈ E'
                W = W ∩ D_rs
                s := s + 1,
          end { while }
      end { for }
end
```

Figure 6: Algorithm which constructs E in $O(n^2)$ time

2.2 Open Convex Polygonal Curves

We call a polygonal curve $P = (p_0, p_1, \ldots, p_{n-1})$ *convex* if the polygon formed by P with the line segment $\overline{p_{n-1}p_0}$ is convex.

Before the algorithm for the min-# problem of a convex curve is described, we study some special properties of G for a convex polygonal curve.

2.2.1 Properties of Convex Polygonal Curves

Lemma 2.4 *Let $G = (V, E)$ be the ϵ-graph of a convex polygonal curve $P = (p_0, p_1, \ldots, p_{n-1})$. If $r + 1 < s$ and $(v_r, v_s) \in E$, then (v_{r+1}, v_s), $(v_r, v_{s-1}) \in E$.*

Proof: By definition, since $(v_r, v_s) \in E$, $d(\overline{p_r p_s}, p_k) \leq \epsilon$ for all $r \leq k \leq s$. We want to show that $d(\overline{p_{r+1}p_s}, p_k) \leq \epsilon$ for all k, $r + 1 \leq k \leq s$, and $d(\overline{p_r p_{s-1}}, p_k) \leq \epsilon$ for all k, $r \leq k \leq s - 1$. The proofs for these two cases are similar; only the former case will be shown here.

Let x be the point on $\overline{p_r p_s}$ which is nearest to p_k. Since P is convex, it can be proved easily that the line $\overline{p_k x}$ has to intersect line $\overline{p_{r+1}p_s}$ at some point y(Figure7).

Thus $d(\overline{p_{r+1}p_s}, p_k) \leq d(y, p_k) \leq d(x, p_k) = d(\overline{p_r p_s}, p_k) \leq \epsilon$. □

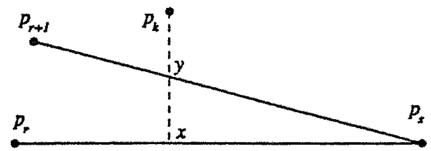

Figure 7: p_k is closer to $\overline{p_{r+1}p_s}$ than to $\overline{p_r p_s}$

The following corollary can be proved by applying Lemma 2.4 repeatedly.

Corollary 2.5 *If* $(v_r, v_s) \in E$, $(v_{r'}, v_{s'}) \in E$ *for all* $r \leq r' < s' \leq s$. □

It follows from Corollary 2.5 that for each vertex v_r where $r < n-1$, there is a vertex v_s such that $(v_r, v_k) \in E$ for all $k, r < k \leq s$ and $(v_r, v_k) \notin E$ for all $k, s < k < n$. We call v_s the *furthest vertex reachable from* v_r. We denote this index s by $f(r)$. Thus G can be completely characterized if $f(r)$ is defined for each $r = 0, 1, \ldots, n-2$. From Corollary 2.5, it can be shown easily that if $r \leq s$, $f(r) \leq f(s)$. So, intuitively, one would like to find $f(r)$ iteratively, i.e., to search for $f(r)$ starting from $f(r-1)$.

Another property is about the distance between p_k and $\overleftrightarrow{p_r p_s}$ for $r < k < s$.

Lemma 2.6 [2] *Let* $(p_r, p_{r+1}, \ldots, p_s)$ *be a convex polygonal curve and* $r \leq k \leq s$. *The function* $d(\overleftrightarrow{p_r p_s}, p_k)$ *is unimodal with respect to* k. □

Let $\gamma(r,s)$ be the smallest index between r and s such that $d(\overleftrightarrow{p_r p_s}, p_{\gamma(r,s)}) = e(\overleftrightarrow{p_r p_s})$, i.e. $\max_{r \leq k \leq s} d(\overleftrightarrow{p_r p_s}, p_k)$. With the unimodal property stated in Lemma 2.6 and the following lemma, $\gamma(r,s)$ can be found in $O(\gamma(r,s) - \gamma(r,s-1))$ or $O(\gamma(r,s) - \gamma(r-1,s))$ time by sequential search starting from $\gamma(r,s-1)$ or $\gamma(r-1,s)$ respectively.

Lemma 2.7 *Let* $(p_0, p_1, \ldots, p_{n-1})$ *be a convex polygonal curve,*

1. $\gamma(r,s) \leq \gamma(r,s+1)$ *for* $0 \leq r < s < n-1$, *2.* $\gamma(r,s) \leq \gamma(r+1,s)$ *for* $0 < r+1 < s \leq n-1$.

Proof: Refer to Figure 8 (a) and (b), $d(\overleftrightarrow{p_r p_s}, p_k) \leq d(\overleftrightarrow{p_r p_s}, p_{\gamma(r,s)})$ implies $d(\overleftrightarrow{p_r p_{s+1}}, p_k) \leq d(\overleftrightarrow{p_r p_{s+1}}, p_{\gamma(r,s)})$ for all k, $r < k < \gamma(r,s)$. Thus $\gamma(r,s) \leq \gamma(r,s+1)$. The proof of $\gamma(r,s) \leq \gamma(r+1,s)$ would be similar. □

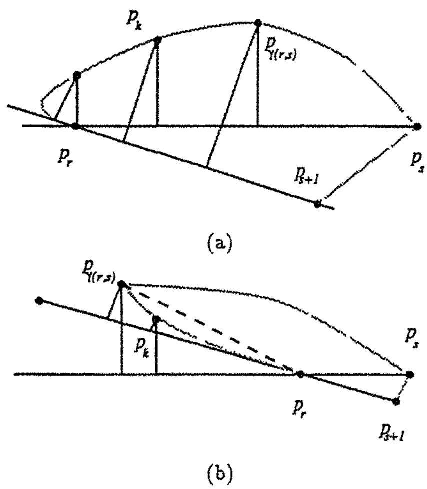

Figure 8: Relationship between $\gamma(r,s)$ and $\gamma(r,s+1)$

2.2.2 ϵ-graph G and Greedy Algorithm

In order to find $f(r)$, Condition A has to be checked. Lemmas 2 6 and 2.7 allow the search of $\gamma(r,s)$ to start from the point previously stopped, i.e., $\gamma(r-1, f(r-1))$: As for the checking of Conditions B and C, we shall keep track of two indices $\alpha(r)$ and $\beta(r)$ for each point p_r such that all points p_k with $r \leq k < \alpha(r)$ (or $\beta(r) < k \leq r$) would have satisfied Condition B (or C) already and $p_{\alpha(r)}$ (or $p_{\beta(r)}$) would be the most likely candidate which would violate Condition B (or C). Note that $d(p_r, p_{\alpha(r)})$

and $d(p_r, p_{\beta(r)})$ must be greater than ϵ.

With $\alpha(r)$ found for $0 \leq r < n-1$ and $\beta(r)$ for $0 < r \leq n-1$ respectively, we can start sketching the algorithm for finding $f(r)$, $0 \leq r < n-1$. Starting with $r = 0$, $s = 1, 2, \ldots$, are tested sequentially as possible candidates for $f(r)$. For each s, condition A is tested, i.e., whether $e(\overleftrightarrow{p_0 p_s}) \leq \epsilon$, by sequential search for $\gamma(0, s)$(Lemma 2.6) starting from $\gamma(0, s-1)$(Lemma 2.7). At the same time, Condition B can be checked in constant time by testing whether $\angle p_s p_0 p_{\alpha(0)} \leq \pi/2$ based on the property that Condition B has already been satisfied for all the points $p_1, p_2, \ldots, p_{\alpha(0)-1}$ and $d(p_0, p_{\alpha(0)}) > \epsilon$. Condition B could not be satisfied if $\angle p_s p_0 p_{\alpha(0)} > \pi/2$. Similarly, Condition C can be checked by testing whether $\angle p_0 p_s p_{\beta(s)} \leq \pi/2$. Candidates for $f(0)$ are checked sequentially until one of these Conditions A, B or C is not satisfied. Thus finding $f(0)$ takes $O(f(0))$ time. The searching of $f(1)$ can be started from $f(0)$ by checking $p_{\gamma(1,s)}$, $p_{\alpha(1)}$ and $p_{\beta(s)}$ for each s sequentially starting from $f(0)$, in particular $\gamma(1, s)$ from $\gamma(0, f(0))$ (Lemma 2.7), and this process takes no more than $O(f(1) - f(0))$ time. Similarly, $f(r)$, for $r = 2, 3, \ldots, n-2$, can be found sequentially from $f(r-1)$ in $O(f(r) - f(r-1))$ time. Thus given $\alpha(r)$, $0 \leq r < n-1$, and $\beta(r)$, $0 < r \leq n-1$, respectively, all the $f(r)$'s, and thus G, can be found in $O(f(0) + \sum_{r=1}^{n-2}(f(r) - f(r-1))) = O(f(n-2)) = O(n)$ time. The algorithm for defining $f(r)$ is shown by Procedure *Compute_f*(n)(Figure 9).

Now we describe the method for computing $\alpha(r)$ for all $0 \leq r < n-1$. $\beta(s)$, $0 < s \leq n-1$, can be computed similarly. $\alpha(0)$ is chosen to be the smallest index s such that $d(p_0, p_s) > \epsilon$. With $\alpha(0)$ computed, we want to find $\alpha(1)$. If $d(p_1, p_{\alpha(0)}) > \epsilon$, then let $\alpha(1)$ be $\alpha(0)$. In general, we can find $\alpha(r)$ by searching sequentially from $\alpha(r-1)$ for the smallest index $s \geq \alpha(r-1)$ such that $d(p_r, p_s) > \epsilon$. If no such s exists, i.e., $d(p_r, p_s) \leq \epsilon$ for all $\alpha(r-1) \leq s \leq n-1$, let $\alpha(r)$ be n. The algorithms to compute $\alpha(r)$ and $\beta(s)$ are shown by Procedures *Compute_*$\alpha(n)$ and *Compute_*$\beta(n)$ respectively(Figure 10).

It is shown in Appendix A.1 that for all r, $0 \leq r < n-1$, all points p_k, $r < k < \alpha(r)$, satisfy Condition B for any line segment $\overline{p_r p_s}$, where $k < s \leq n-1$.

After the ϵ-graph G of a convex polygonal curve has been constructed, we can apply a greedy approach to find the shortest path from v_0 to v_{n-1}. For each step, we go

```
procedure Compute_f(n)
begin
        r := 0
        s := α(0)    {d(p_0, p_k) ≤ ε due to the definition of α(0) }
        γ := 0
        while r < n − 1 do
        begin
                γ := Peak(p_r p_s, h)   { γ is γ(r, s) }
                while s < n and d(p_r p_s, p_h) ≤ ε and      { Condition A }
                        not (α(r) < s and ∠p_α(r) p_r p_s > π/2) and    { Condition B }
                        not (β(s) > r and ∠p_β(s) p_s p_r > π/2) do    { Condition C }
                begin
                        s := s + 1
                        γ := Peak(p_r p_s, γ)
                end
                output f(r) = s − 1
                r := r + 1
        end
end

function Peak(L, γ)  { This function returns γ(r, s), where L = p_r p_s and γ is
                       the starting index for the sequential search based on Lem-
                       mas 2.6 and 2.7 }
begin
        k := h
        while d(L, p_k) < d(L, p_k+1) do
                k := k + 1
        return k
end
```

Figure 9: Procedure *Compute_f*(n)

as "far" as possible until v_{n-1} is reached, i.e., the path is $R(n-1) = (v_{i_0}, v_{i_1}, \ldots, v_{i_m})$ where $v_{i_0} = v_0, v_{i_m} = v_{n-1}, i_{j+1} = f(i_j)$ for $0 \le j \le m-1$. It can be proved (Appendix A.2) that $R(n-1)$ is a shortest path from v_0 to v_{n-1}. Thus we have,

Theorem 2 *Given $\epsilon > 0$, and a convex polygonal curve $P = (p_0, p_1, \ldots, p_{n-1})$, the min-# problem can be solved in $O(n)$ time.* □

2.3 General Closed Polygonal Curves

A polygonal curve is *closed* when there is an edge joining p_{n-1} and p_0, i.e., $p_n = p_0$. The min-# problem for a closed polygonal curve is to find an approximate curve $(p_{i_0}, p_{i_1}, \ldots, p_{i_m})$ with error within ϵ, $p_{i_0} = p_{i_m}$ and m as small as possible.

```
Procedure Compute_α(n)
begin
      r := 0
      s := 1
      while r < n - 1 do
      begin
            while s < n and d(p_r, p_s) ≤ ε do
            begin
                  s := s + 1
            end
            output α(r) = s
            r := r + 1
      end
end
```

```
Procedure Compute_β(n)
begin
      s := n - 1
      r := s - 1
      while s > 0 do
      begin
            while r ≥ 0 and d(p_r, p_s) ≤ ε do
            begin
                  r := r - 1
            end
            output β(s) = r
            s := s - 1
      end
end
```

Figure 10: Computing $\alpha(r)$ and $\beta(r)$

Theorem 3 *Given $\epsilon > 0$ and a closed polygonal curve $P = (p_0, p_1, \ldots, p_{n-1})$, the min-# problem can be solved in $O(n^3)$ time.*

Proof: The problem can be solved by considering n separate open curve problems by breaking up the closed curve at each point p_k. Since each min-# problem can be solved in $O(n^2)$ time, the closed curve min-# problem can be solved in $O(n^3)$ time. □

Similarly, the min-# problem for a closed convex polygonal curve can also be solved in $O(n^2)$ time. We shall show in the following how the problem for a closed convex polygonal curve can be solved in $O(n)$ time.

2.4 Convex Closed Polygonal Curves

First of all, we construct the ϵ-graph G for a closed polygonal curve P_C. Formally, G is redefined for a closed polygonal curve as follows. $G = (V, E)$ where $V = (v_0, v_1, \ldots, v_{n-1})$ and $E = \{(v_r, v_s) \mid d(\overline{p_r p_s}, p_k) \leq \epsilon \text{ for all } k = r, r+1, \ldots, s-1, s\}$. For example, if $n = 9$, $(v_4, v_8) \in E$ if $d(\overline{p_4 p_8}, p_k) \leq \epsilon$ for all $k = 4, 5, 6, 7, 8$. $(v_8, v_4) \in E$ if $d(\overline{p_4 p_8}, p_k) \leq \epsilon$ for all $k = 8, 0, 1, 2, 3, 4$. Thus when G is constructed, the same G can be referenced in order to find the optimal approximate curve of each open curve.

It can be verified easily that Lemma 2.4 and thus Corollary 2.5 can be extended to each open curve $(p_r, p_{r+1}, \ldots, p_0, \ldots, p_r)$. In terms of the ϵ-graph G, these properties can be summarized in the following lemma. Note that modulo n arithmetic is assumed in the indexing of the vertices.

Lemma 2.8 *If $(v_r, v_s) \in E$, then $(v_{r'}, v_{s'}) \in E$ for all $r' = r, r+1, \ldots, s-1$, and $s' = r'+1, r'+2, \ldots, s$.* □

Thus $f(r)$ can still be used to characterise G when it is redefined for a closed curve. $f(r)$, $0 \leq r < n$, is now defined as the index s such that $(v_r, v_k) \in E$ for all $k = r+1, r+2, \ldots, s$ and $(v_r, v_k) \notin E$ for any other k.

It can be shown easily that the ϵ-graph $G_0 = (V, E_0)$ for the open polygonal curve $P_0 = (p_0, p_1, \ldots, p_0)$ where $p_n = p_0$ is always a subgraph of $G = (V, E)$ for P_C. The edges which are in E but not in E_0 are those (v_r, v_s) where $r \geq s$. Lemma 2.9 shows how we can construct G by constructing G_0 first and adding those edges (v_r, v_s) with $r \geq s$ afterwards, i.e., the set of edges $E - E_0$.

Lemma 2.9 *Let g be the smallest index such that $(v_g, v_0) \in E_0$. All edges in $E - E_0$ are contained in the ϵ-graph of the open curve $P_g = (p_g, p_{g+1}, \ldots, p_{n-1}, p_0, \ldots, p_{f(0)})$.*

Proof: Trival. □

Let $f(r)$, $f_0(r)$ and $f_g(r)$ be the index of the furthest vertex reachable by v_r in the graph for P_C, P_0 and P_g respectively. If $f_0(r) = 0$ for some $r \geq g$, there is a possibility that $(v_r, v_{s'}) \in E$ for some s', $0 < s' \leq f(0)$. Thus $f(r) = f_g(r)$. If $f_0(r) \neq 0$, all (v_r, v_s) in E are contained in E_0 and $f(r)$ equals $f_0(r)$. Thus G can be constructed for P_C in $O(n)$ time.

Our next step is to apply greedy approach to find a shortest path P_0' from v_0 to v_0 itself. Let $P_0' = (v_{i_0}, v_{i_1}, v_{i_2}, \ldots, v_{i_m})$. Note that $v_{i_0} = v_{i_m} = v_0$, $i_j = f(i_{j-1})$ for $j = 1, 2, \ldots, m-1$. The following lemmas show that the optimal approximate curve of each open curve, constructed by breaking up P_C at each point, consists of similar number of line segments.

Lemma 2.10 *If P_0' consists of m line segments, the optimal approximate curve P_i' for the open curve $P_i = (p_i, p_{i+1}, \ldots, p_{n-1}, p_0, \ldots, p_i)$ has at least $m-1$ and at most $m+1$ line segments.*

Proof: Assume the contrary that P'_i consists of only $m-2$ line segments. If p_0 is in P'_i, we would break up P'_i at p_0 and arrive at an approximate curve for P_0, which has only $m-2$ line segments. If p_0 is not in P'_i , by Lemma 2.8, we can replace the line segment $\overline{p_{i_j}p_{i_{j+1}}}$ in P'_i, which "covers" p_0, i.e., $i_j < n$ and $0 < i_{j+1}$, by two line segments $\overline{p_{i_j}p_0}$ and $\overline{p_0p_{i_{j+1}}}$ and can also arrive at another approximate curve for P_0 which has $m-1$ line segments. Both cases lead to the contradiction that P'_0 consists of m line segments.

With similar argument as above, we can show that P'_i consists of at most $m+1$ line segments. □

Since P'_C is one of the P'_i's, we have the following corollary.

Corollary 2.11 *If P'_0 consists of m line segments, the optimal approximate curve P'_C for the closed curve P_C has at least $m-1$ line segments.* □

Lemma 2.12 *For any line segment $\overline{p_{i_j}p_{i_{j+1}}}$ in P'_0, $0 \le j < m-1$, the optimal approximate curve P'_C must contain some vertex p_k with $i_j \le k \le i_{j+1}$.*

Proof: Assume the contrary that $\overline{p_{i_j}p_{i_{j+1}}}$ does not "cover" any vertex of the optimal curve. The optimal curve must contain a line segment $\overline{p_rp_s}$ where $r < i_j < i_{j+1} < s$. Thus $(v_{i_j}, v_s) \in E$ by Lemma 2.8 and $f(i_j) \ge s$. This contradicts the fact that $f(i_j) = i_{j+1}$. □

To find P'_C, we can select the line segment with minimum difference $(i_{j+1} - i_j)$ and find the approximate curve for each open curve $(p_k, p_{k+1}, \ldots, p_{n-1}, p_0, p_1, \ldots, p_{k-1}, p_k)$ where $i_j \le k \le i_{j+1}$. Note that by Lemma 2.12, one of these approximate curves would be an optimal approximate curve P'_C for P_C. Since each approximate curve has at most $m+1$ edges, it takes $O(m)$ time to search G to find the shortest path and the approximate curve. As the minimum difference $(i_{j+1} - i_j) \le n/(m-1)$, the total time to find P'_C would be no more than $O(m\frac{n}{m}) = O(n)$ time.

Theorem 4 *Given $\epsilon > 0$ and a closed polygonal curve $P_C = (p_0, p_1, \ldots, p_{n-1})$, the min-# problem for P_C can be solved in $O(n)$ time when P_C is convex.* □

3 The Min-ϵ Problem

Given $m > 0$ and a polygonal curve P, the problem is to find an approximate polygonal curve P' consisting of at most m line segments having minimum error $e(P')$. We denote this minimum error by ϵ^*. We consider open curves in this section first.

3.1 General Open Polygonal Curves

Imai and Iri[7] have proposed a method which makes use of the algorithm for solving the min-# problem. For any line segment $\overline{p_r p_s}$ and point p_k, the distance $d(\overline{p_r p_s}, p_k)$ is one of $d(p_r, p_k)$, $d(p_s, p_k)$ or $d(\overleftrightarrow{p_r p_s}, p_k)$, depending on $\angle p_k p_r p_s$ and $\angle p_k p_s p_r$. Thus ϵ^* of P is contained in a set $S = \{d(p_r, p_s) | 0 \leq r < s < n\} \cup \{e(\overleftrightarrow{p_r p_s}) \mid 0 \leq r < s < n\}$. The set $\{e(\overleftrightarrow{p_r p_s}) \mid 0 \leq r < s < n\}$ can be determined in $O(n^2 \log n)$ time[11] and $\{d(p_r, p_s) \mid 0 \leq r < s < n\}$ in $O(n^2)$ time. Since $|S| < 2n^2$, sorting elements in S needs $O(n^2 \log n)$ time. ϵ^* can be found by binary search for the smallest element in S such that P can be approximated with at most m line segments. For each ϵ in S, we can test whether it is a possible candidate for ϵ^* by using the algorithm for solving the min-# problem with input P and ϵ. Thus this approach takes $O(n^2 \log n + t(n) \log n)$ where $t(n)$ is the time required to solve a min-# problem. If the algorithm discussed in Section 2 is used, t(n) is $O(n^2)$ and the total running time is $(n^2 \log n)$. This is an improvement from $O((n \log n)^2)$, in which $t(n) = n^2 \log n$ where the method of Melkman and O'Rourke [9] is used.

Theorem 5 *Given $m > 0$ and an open polygonal curve with n vertices, the min-ϵ problem can be solved in $O(n^2 \log n)$ time.* □

3.2 The Min-ϵ Problem of a Convex Curve

Imai and Iri[7] have also proposed another approach using graphs to solve the min-ϵ problem. Let $G^* = (V, E^*)$ be a *weighted* directed graph where $V = \{v_0, v_1, \ldots, v_{n-1}\}$ and $E = \{(p_r, p_s) | 0 \leq r < s < n\}$. Let $w(r, s)$, the weight of edge (v_r, v_s), be the error of $\overline{p_r p_s}$, i.e., $w(r, s) = e(\overline{p_r p_s})$. In particular, $w(r, r+1) = 0$ for all $0 \leq r < n - 1$. We shall call G^* the *error-graph* of P_0. For each path $R = (v_{i_0}, v_{i_1}, \ldots, v_{i_m})$, the *weight of* R, $w(R)$, is defined to be the maximum weight of the edges on R, i.e, $\max_{0 \le j < m}\{w(i_j, i_{j+1})\}$. Such path R corresponds to an approximate curve of $P' = (p_{i_0}, p_{i_1}, \ldots, p_{i_m})$ having $e(P') = w(R)$. Thus solving the min-ϵ problem is equivalent to finding a path R from v_0 to v_{n-1} in G^* consisting of at most m edges and having minimum $w(R)$.

In the following, we shall discuss how to take advantage of the convexity of the given curve P to construct G^* in $O(n^2)$ time and to find ϵ^* with an additional $O(nm)$ time.

Recall that in Section 2, we can construct G by considering Conditions A, B and C. In order to compute $w(r,s)$, we consider the following three types of values, $e(\overleftrightarrow{p_r p_s})$, u_{rs} and v_{rs}, where

$u_{rs} = \max_{r \le k \le s}\{d(p_r, p_k) \mid \angle p_k p_r p_s \ge \pi/2\}$, radius of the smallest "hemisphere" centered at p_r to include those vertices p_k, $r < k < s$, lying outside $\overline{p_r p_s}$, and

$v_{rs} = \max_{r \le k \le s}\{d(p_s, p_k) \mid \angle p_k p_s p_r \ge \pi/2\}$, radius of the smallest "hemisphere" centered at p_s to include those vertices p_k, $r < k < s$, lying outside $\overline{p_r p_s}$.

It is easy to show that $w(r,s) = \max\{e(\overleftrightarrow{p_r p_s}), u_{rs}, v_{rs}\}$. We shall show that all $e(\overleftrightarrow{p_r p_s})$, u_{rs} and v_{rs}, $0 \le r < s < n$, can be found in $O(n^2)$ time and thus $w(r,s)$ for $0 \le r < s < n$.

Computing $e(\overleftrightarrow{p_r p_s})$ is not difficult since $\gamma(r,s)$ can be found by sequential search from $\gamma(r, s-1)$ (Lemmas 2.6 and 2.7). $e(\overleftrightarrow{p_r p_s})$, for all s, $r < s < n$, can be found in $O(n)$ time for any particular r and in $O(n^2)$ time for all r and s where $r < s$.

In general, u_{rs} can be computed from $u_{r,s-1}$ and similarly v_{rs} from $v_{r+1,s}$. In the following, only the computation of u_{rs} from $u_{r,s-1}$ will be described. Since P is convex, for each pair of points p_r and p_s where $r < s$, $\angle p_k p_r p_s \ge p_{k'} p_r p_s$ if $r < k < k' \le s$. Thus, there exists a *unique* index $\varphi(r,s)$ such that $\angle p_k p_r p_s \ge \pi/2$ for all $k, r < k < \varphi(r,s)$ and $\angle p_k p_r p_s < \pi/2$ for all $k, \varphi(r,s) \le k \le s$. In other words, $\varphi(r,s)$ is the smallest index k greater than r having $\angle p_k p_r p_s$ less than $\pi/2$. Since P is convex, $\angle p_k p_r p_{s-1} \le \angle p_k p_r p_s$ for all k, $r < k < s$, and thus $\varphi(r, s-1) \le \varphi(r,s)$. $\varphi(r,s)$ can therefore be found by searching from $\varphi(r, s-1)$ towards s for the first k having $\angle p_k p_r p_s < \pi/2$(Figure 11). Therefore u_{rs} can be found without considering all p_k again, i.e., $u_{rs} = \max\{u_{r,s-1}, \max_{\varphi(r,s-1)\leq k<\varphi(r,s)}\{d(p_r,p_k)\}\}$. Thus u_{rs} can be found in $O(\varphi(r,s)-\varphi(r,s-1))$ time. Since $\varphi(r,s) < s < n$, u_{rs} can be defined in $O(n)$ time for all $s, r < s < n$, and in $O(n^2)$ time for all r and s where $r < s$.

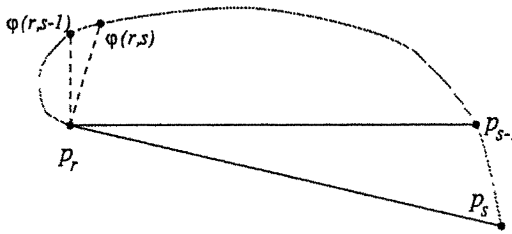

Figure 11: Relationship between $\varphi(r,s)$ and $\varphi(r,s-1)$

Since $e(\overleftrightarrow{p_rp_s})$, u_{rs}, v_{rs} as well as $w(r,s)$ can be found in $O(n^2)$ time. The weighted error-graph G^* can therefore be constructed for a convex polygonal curve in $O(n^2)$ time.

With G^* constructed, we can use it to find ϵ^* and its corresponding approximate curve by binary search on the sorted weights in G^* in $O(n^2\log n)$ time. In the following, we shall describe a dynamic programming method to find ϵ^* in $O(nm)$ time after the error-graph G^* has been constructed.

Let $f(i,k)$ be the minimum weight of a path from v_0 to v_i consisting of *at most* k edges. $f(i,k)$ can be defined as follows:

$$f(i,k) = \begin{cases} \min_{1\leq j\leq i}\{\max\{f(j,k-1), w(j,i)\}\} & \text{if } k > 1 \\ w(0,i) & \text{if } k = 1 \end{cases}$$

The path corresponding to each $f(i,k)$ can be found by keeping track of the index at which the minimum occurs, i.e., the last vertex adjacent to v_i on the minimum-weight path from v_0 to v_i.

If we consider the problem as a general dynamic programming problem, all $f(j,k-1)$, $1 \leq j < i$, have to be referenced in order to compute $f(i,k)$. There are up to $n-2$ such j's. Thus solving the problem in this way takes $O(n^2m)$ time.

However, the following two lemmas, which hold for the convex polygonal curves, allow us to find $f(i,k)$ without considering all these j's.

Lemma 3.1 *Let $G^* = (V, E^*)$ be the error-graph constructed for a convex polygonal curve P. If $(p_r, p_s) \in E^*$, then*

1. $w(r,s) \leq w(r,s+1)$ *if* $0 \leq r \leq s < n-1$; *and*
2. $w(r,s) \leq w(r-1,s)$ *if* $0 < r \leq s \leq n-1$.

Proof: Similar to Lemma 2.4. □

Intuitively, Lemma 3.1 states that the error will be larger for more "separated" vertices. This fact also makes $f(i,k)$ nondecreasing with respect to i.

Lemma 3.2 *Let $P = (p_0, p_1, \ldots, p_{n-1})$ be a convex polygonal curve. $f(i,k) \leq f(i+1,k)$ for $1 \leq i < n$ and $1 \leq k \leq m$.*

Proof: The lemma can be proved by induction on k. The lemma is true for $k = 1$ by Lemma 3.1. Assume the lemma is true for $k-1$ for some $1 < k \leq m$. For the case of k,

$$\begin{aligned}
f(i+1,k) &= \min_{1\leq j\leq i+1}\{\max\{f(j,k-1), w(j,i+1)\}\} \\
&= \min\{\min_{1\leq j\leq i}\{\max\{f(j,k-1), w(j,i+1)\}\}, f(i+1,k-1)\} \\
&\geq \min\{\min_{1\leq j\leq i}\{\max\{f(j,k-1), w(j,i+1)\}\}, f(i,k-1)\} \quad \text{(ind. hypo.)} \\
&\geq \min\{\min_{1\leq j\leq i}\{\max\{f(j,k-1), w(j,i)\}\}, f(i,k-1)\} \quad \text{(Lemma 3.1)} \\
&= f(i,k)
\end{aligned}$$

□

From Lemmas 3.1 and 3.2, $f(j,k-1)$ is non-decreasing and $w(j,i)$ is non-increasing with respect to j. Moreover, $f(1,k-1) = 0 \leq w(1,i)$ and $f(i,k-1) \geq w(i,i) = 0$ for both i and k greater than 1. Therefore the functions $f(j,k-1)$ and $w(j,i)$ must "intersect" at some j, i.e., there exists an index j', such that $f(j',k-1) \leq w(j',i)$ and $f(j'+1,k-1) \geq w(j'+1,i)$. Let $\lambda(i,k)$ be the value j

such that $\max\{f(j,k-1), w(j,i)\}$ attains its minimum. It is easy to see that $\lambda(i,k)$ is either j' or $j'+1$.

Assume that we have found $f(i-1,k)$ for some $i > 2$ and $k > 1$ and $\lambda(i-1,k)$ is known, we can find $\lambda(i,k)$ without considering all values of $f(j,k)$ and $w(j,i)$, $1 \le j \le i$. Since $w(\lambda(i-1,k),i) \ge w(\lambda(i-1,k),i-1)$ by Lemma 3.1, the index j' where $w(j',i)$ and $f(j',k-1)$ "intersect" should not be less than $\lambda(i-1,k)$, i.e., $\lambda(i,k) \ge \lambda(i-1,k)$. Thus we can search sequentially from $\lambda(i-1,k)$ towards i for $\lambda(i,k)$ and find $f(i,k)$ in $O(\lambda(i,k) - \lambda(i-1,k))$ time. Therefore all $f(i,k)$, $i = 1,2,\ldots,n-1$ can be found in $O(n)$ time for a particular k and in $O(nm)$ time for all k, $1 \le k \le m$. Combining the fact that the error-graph of P_C can be constructed in $O(n^2)$ time, we have the following theorem.

Theorem 6 *Given $m > 0$ and a convex polygonal curve with n vertices, the min-ϵ problem can be solved in $O(n^2)$ time.* □

3.3 Min-ϵ Problem of a General Closed Polygonal Curve

For a general closed polygonal curve $P_C = (p_0, p_1, \ldots, p_{n-1})$, the min-$\epsilon$ problem is to find an approximation curve $P'_C = (p_{i_0}, p_{i_1}, \ldots, p_{i_m})$, where $i_0 = i_m = k$ for some $k < n$, and its error $e(P'_C)$ is as small as possible. The problem can be solved by considering n open curve problems. Each problem consists of an open curve formed by breaking up P_C at each point p_k. Solving one open curve problem takes $O(n^2 \log n)$ time. Thus the min-ϵ problem can be solved in $O(n^3 \log n)$ time.

Theorem 7 *Given $m > 0$ and a closed polygonal curve with n vertices, the min-ϵ problem can be solved in $O(n^3 \log n)$ time.* □

3.4 The Min-ϵ of a Convex Closed Polygonal Curve

Now we turn to the closed curve min-ϵ problem for a convex polygonal curve. We shall show in the following that the minimum error ϵ^* can be determined without considering all n open curves.

We extend the definition of the error-graph G^* so that the error of the line segment between any pair of distinct points are included. Now define $G^* = (V, E^*)$

where $V = (v_0, v_1, \ldots, v_{n-1})$ and $E = \{(v_r, v_s) \mid 0 \leq r, s < n\}$. The weight $w(r,s)$ of (v_r, v_s) is defined as $\max_{k=r,r+1,\ldots,s}\{d(\overline{p_r p_s}, p_k)\}$. For example, if $n = 9$, $w(4,8) = \max_{k=4,5,6,7,8}\{d(\overline{p_4 p_8}, p_k)\}$, $w(8,4) = \max_{k=8,1,2,3,4}\{d(\overline{p_4 p_8}, p_k)\}$, and $w(4,4) = 0$.

Since the algorithm discussed in Section 3.2 to find the error-graph for an open curve can be modified to find the error-graph for P_C, error-graph for P_C can be constructed in $O(n^2)$ time. Given the error-graph of P_C, if we find the optimal approximate curve of P_C by considering all n open curves, it will take $O(n(nm))$ time. In the following, we shall show how this problem can be done in $O(n^2)$ time.

The first step is to apply the algorithm described in Section 3.2 to solve the min-ϵ problem for the open convex polygonal curve $P_0 = (p_0, p_1, \ldots, p_{n-1}, p_0)$. Let ϵ_0 be the found minimum error to approximate P_0 with at most m line segments. Note that $\epsilon_0 \geq \epsilon^*$. The second step is to solve the min-# problem for P_0 with the given error ϵ_0 by the greedy algorithm as described in Section 2.2. Let m' be the minimum number of line segments to approximate P_0 with error ϵ_0 and $P'_0 = (p_{i_0}, p_{i_1}, \ldots, p_{i_{m'}})$ be its approximate curve such that $w(i_{j-1}, i_j + k) > \epsilon_0$ for all $k \geq 1$(except the last segment). Note that for any line segment $\overline{p_{i_{j-1}} p_{i_j}}$ of P'_0, $0 < j \leq m' - 1$, there must be a point p_k, $i_{j-1} \leq k \leq i_j$, belonging to an optimal approximate curve of P_C with error ϵ^* and at most m line segments, i.e., the solution of this problem. Assume the the contrary that $\overline{p_{i_{j-1}} p_{i_j}}$ does not "cover" any point of an optimal curve. This optimal curve must contain a line segment $\overline{p_r p_s}$ where $r < i_{j-1} < i_j < s$. Thus $w(i_{j-1}, s) \leq w(r, s) \leq \epsilon^* \leq \epsilon_0$ by Lemma 3.1. This contradicts the fact that $w(i_{j-1}, i_j + k) > \epsilon_0$ for all $k \geq 1$.

The third step is to select the line segment $\overline{p_{i_{j-1}} p_{i_j}}$ of P'_0 with minimum difference $i_{j-1} - i_j$, $j \leq m' - 1$. The last step is to apply the min-ϵ algorithm (Section 3.2) to find the minimum error ϵ_k for each open curve $P_k = (p_k, p_{k+1}, \ldots, p_0, \ldots, p_k)$, where $i_{j-1} \leq k < i_j$. With the previous argument ϵ^* must be equal to one of the ϵ_k's, $\epsilon^* = \min_{i_{j-1} \leq k < i_j}\{\epsilon_k\}$.

As far as the time complexity is concerned, the first three steps takes no more than $O(n^2)$ time. Since the minimum difference $i_{j-1} - i_j \leq n/(m'-1) = O(n/m)$ and $m' \geq m/2$ (Lemma 3.3), the total time to find ϵ^* and its approximate curve would take no more than $O(mn\frac{n}{m}) = O(n^2)$ time.

Lemma 3.3 $m' \geq m/2$.

Proof: By contradiction. Assume $m' < m/2$, we can construct another approximate curve P_0'' with error less than ϵ_0 and the number of line segments is at most m. Basically, each line segment $\overline{p_{i_j} p_{i_{j+1}}}$ of P_0' with $w(i_j, i_{j+1}) = \epsilon_0$ can be replaced by two line segments $\overline{p_{i_j} p_k}$ and $\overline{p_k p_{i_{j+1}}}$, $i_j < k < i_{j+1}$, with $w(i_j, k), w(k, i_{j+1}) < w(i, j)$. Note that the above replacement can always be possible for all edges with at most one exception where the replacement can only be possible with three line segments.

Thus P_0'' can be constructed with at most $2m' + 1 \leq m$ line segments. This contradicts the fact that ϵ_0 is the minimum error. □

Thus we have the following theorem.

Theorem 8 *Given $m > 0$ and a closed convex polygonal curve with n vertices, the min-ϵ problem can be solved in $O(n^2)$ time.* □

4 Conclusion

In this paper, we have shown that the min-# problem formulated in [7] can be solved in $O(n^2)$ time for an open polygonal curve and $O(n^3)$ for a closed curve, where n is the number of vertices of the given polygonal curve. If the given polygonal curve is convex, the problem can be solved in $O(n)$ time for both open and closed curves, which is optimal in terms of time complexity. We have also shown that the min-ϵ problem can be solved in $O(n^2 \log n)$ time for a general open polygonal curve, and $O(n^3 \log n)$ time for a closed curve. This time complexity can be further reduced to $O(n^2)$ for both open and closed convex polygonal curves.

For the general curve approximation problem, results for other error criteria can be found in [5, 7].

References

[1] P.J. Burt. Fast filter transforms for image processing. *Computer Graphics and Image Processing 16*, pp. 20-51,1981.

[2] B.M. Chazelle. *Computational Geometry and Convexity.* Ph.D Thesis, Yale University, 1980.

[3] F. Chin, A. Choi and Y. Luo. Optimal generating kernels for image pyramids by Linear fitting. In *Proc. 1989 Int. Symp. on Computer Architecture and Digital Signal Processing* (Hong Kong, Oct. 11-14, 1989), 612-617.

[4] F. Chin, A. Choi and Y. Luo. Optimal generating kernels for image pyramids by Piecewise fitting. *IEEE Trans. Pattern Anal. Machine Intell.* (to appear).

[5] L. Guibas, J. Hershberger, J. Mitchell and J.S. Snoeyink Approximating Polygons and Subdivisions with Minimum Link Paths. *Technical Report 92-5*, University of British Columbia, March 1992.

[6] S. L. Hakimi and E. F. Schmeichel. Fitting polygonal functions to a set of points in the plane. *CVGIP, Graphical Models and Image Processing*, Vol. 53 (1991), pp. 132-136.

[7] H. Imai and M. Iri. Polygonal Approximations of a Curve-Formulations and Algorithms. In G. T. Toussaint, editor, *Computational Morphology.* North Holland, 1988

[8] P. Meer, E.S. Baughter and A. Rosenfeld. Frequency domain analysis and synthesis of image pyramid generating kernels. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 512-522, July 1987.

[9] A. Melkman and J. O'Rourke. On Polygonal Chain Approximation. In G. T. Toussaint, editor, *Computational Morphology.* North Holland, 1988

[10] A. Rosenfeld. (ed.) *Multiresolution Image Processing and Analysis.* New York, Springer-Verlag, 1984.

[11] G. T. Toussaint. On the Complexity of Approximating Polygonal Curves in the Plane. In *Proc. IASTED, International Symposium on Robotics and Automation, Lugano, Switzerland, 1985*

Appendix

A.1 The Proof that for all $r, 0 \leq r < n-1$, all points p_k, $r < k < \alpha(r)$, satisfy Condition B for any line segment $\overline{p_r p_s}$, where $k < s \leq n-1$

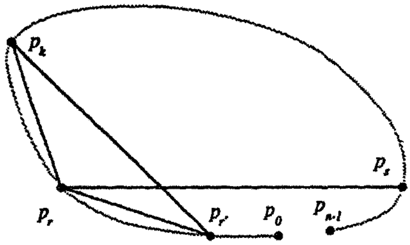

Figure 12: Property of $\alpha(r)$

Lemma A.1 *For all r, $0 \leq r < n-1$, any k, $r < k < \alpha(r)$, and s, $k < s \leq n-1$, if $\angle p_k p_r p_s > \pi/2$, $d(p_r, p_k) \leq \epsilon$.*

Proof: If $\alpha(r-1) \leq k < \alpha(r)$, from Procedure *Compute_α(n)* given in Section 2.2.2, $d(p_r, p_k) \leq \epsilon$ and the lemma is proved. If $k < \alpha(r-1)$ (Figure 12), from Procedure *Compute_α(n)*, there must exist some vertex $p_{r'}$ where $0 \leq r' < r$ with $d(p_{r'}, p_k) \leq \epsilon$. If $\angle p_k p_r p_s > \pi/2$, we must have $\angle p_k p_r p_{r'} > \pi/2$ and it is easy to see that $d(p_r, p_k) \leq \epsilon$. □

Note that $d(p_r, p_{\alpha(r)}) > \epsilon$ if $\alpha(r) \neq n$. The following lemma shows that Condition B for $\overline{p_r p_s}$ is satisfied by all points p_k such that $r < k < s$ if and only if $r < \alpha(r) < s$ and $\angle p_{\alpha(r)} p_r p_s \leq \pi/2$.

Lemma A.2 *Assume $0 \leq r < s < n$. There exists some p_k, $r < k < s$ such that $d(p_r, p_k) > \epsilon$ and $\angle p_k p_r p_s > \pi/2$ if and only if $r < \alpha(r) < s$ and $\angle p_{\alpha(r)} p_r p_s > \pi/2$.*

Proof: *If* part can be proved easily. As $\angle p_{\alpha(r)} p_r p_s > \pi/2$, from the definition of $\alpha(r)$, $\alpha(r) \neq n$, $d(p_r, p_{\alpha(r)}) > \epsilon$.

The *only if* part can be proved by contrapositive. Assume $\alpha(r) \geq s$ or $\angle p_{\alpha(r)}p_r p_s \leq \pi/2$. It follows from Lemma A.1 that $d(p_r, p_k) \leq \epsilon$ for all those points p_k with $\angle p_k p_r p_s > \pi/2$ and $r < k < \alpha(r)$. Since $r < k < \alpha(r)$ has already been covered by the case with $\alpha(r) \geq s$, what remains to be shown is for the assumption when $\angle p_{\alpha(r)}p_r p_s \leq \pi/2$ and $\alpha(r) < s$. Since the given polygon P is convex, $\angle p_k p_r p_s \leq \angle p_{\alpha(r)}p_r p_s \leq \pi/2$ for all points p_k with $\alpha(r) \leq k < s$. Thus the lemma is proved. □

A.2 The correctness proof of the greedy approach in finding the shortest path from v_0 to v_{n-1} in the ϵ-graph G of a convex polygonal curve

Lemma A.3 *Let $G = (V, E)$ be the corresponding ϵ-graph constructed for a convex polygonal curve $P = (p_0, p_1, \ldots, p_{n-1})$. Assume $1 \leq k \leq n-1$. $R(k) = (v_0 = v_{i_0}, v_{i_1}, \ldots, v_{i_m} = v_k)$ with $i_{j+1} = f(i_j)$ for $0 \leq j < m-1$ and $i_m \leq f(i_{m-1})$ is a shortest path from v_0 to v_k.*

Proof: By induction on k. Clearly the lemma is true for $k = 1$. Assume that the lemma is true for all $k' \leq k-1$ with $k \leq n-1$. If $(v_0, v_k) \in E$, $R(k) = (v_0, v_k)$ and the lemma holds trivially. Otherwise any shortest path from v_0 to v_k has at least two edges. Let $R^*(k) = (v_0 = v_{j_0}, v_{j_1}, \ldots, v_{j_{l-1}}, v_{j_l} = v_k)$ be a shortest path from v_0 to v_k. Clearly $(v_{j_0}, v_{j_1}, \ldots, v_{j_{l-1}})$ is a shortest path from v_0 to $v_{j_{l-1}}$. By induction hypothesis, $R(j_{l-1})$ is also a shortest path from v_0 to $v_{j_{l-1}}$. Thus, the new path with $(v_{j_0}, v_{j_1}, \ldots, v_{j_{l-1}})$ in $R^*(k)$ replaced by $R(j_{l-1})$ is also a shortest path from v_0 to v_k. Let $R(j_{l-1}) = (v_0 = v_{i_0}, v_{i_1}, \ldots, v_{i_{l-1}} = v_{j_{l-1}})$. Note that $f(i_{l-2}) < k$, otherwise $(v_0, v_{i_1}, \ldots, v_{i_{l-2}}, v_k)$ would be a path from v_0 to v_k with length shorter than that of $R^*(k)$, contradicting the fact that $R^*(k)$ being the shortest. Since $i_{l-1} \leq f(i_{l-2}) < k$ and $(v_{i_{l-1}}, v_k) \in E$, $(v_{f(i_{l-2})}, v_k) \in E$ by Corollary 2.5. Thus $(v_0 = v_{i_0}, v_{i_1}, \ldots, v_{i_{l-2}}, v_{f(i_{l-2})}, v_k) = R(k)$ as specified in the lemma with $m = l$ is also a feasible path from v_0 to v_k and has the same length as $R^*(k)$. Thus, the lemma is true for k. □

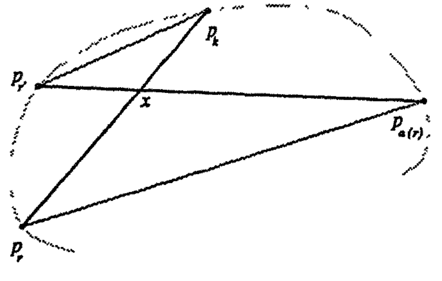

Figure 13: Another property of $\alpha(r)$

A.3 The min-# problem for a closed convex polygonal curve

The method for solving the min-# problem for a closed convex polygonal curve P_C, which can be approximated by at most two lines only, will be described.

Let $P_C = (p_0, p_1, \ldots, p_{n-1})$. Suppose we compute the values of $\alpha(r)$ for the open curve $P_0 = (p_0, p_1, \ldots, p_{n-1})$. Some properties of $\alpha(r)$ can help us to solve the problem.

Lemma A.4 *If $d(p_r, p_{\alpha(r)}) > \epsilon$, $d(p_r, p_k) \leq \epsilon$ for all k, $r \leq k < \alpha(r)$ and $p_{r'}$ is a vertex such that $r < r' \leq \alpha(r)$ and $d(p_{r'}, p_{\alpha(r)}) \leq \epsilon$, then $d(p_{r'}, p_k) \leq \epsilon$ for all k, $r' \leq k \leq \alpha(r)$.*

Proof: $\overline{p_r p_k}$ and $\overline{p_{r'} p_{\alpha(r)}}$ should intersect at some point x(Figure 13). By triangular inequality, $d(p_k, x) + d(p_{r'}, x) \geq d(p_{r'}, p_k)$ and $d(p_r, x) + d(p_{\alpha(r)}, x) \geq d(p_r, p_{\alpha(r)})$. Thus $d(p_r, p_k) + d(p_{r'}, p_{\alpha(r)}) \geq d(p_{r'}, p_k) + d(p_r, p_{\alpha(r)})$. Since $d(p_r, p_{\alpha(r)}) > \epsilon$ and $d(p_{r'}, p_{\alpha(r)})$, $d(p_r, p_k) \leq \epsilon$, $d(p_{r'}, p_k) = d(p_r, p_k) + d(p_{r'}, p_{\alpha(r)}) - d(p_r, p_{\alpha(r)}) \leq \epsilon$. □

Note that there are difference between Lemma A.4 and Lemma A.1. From Lemma A.1, it is obvious that $d(p_r, p_k)$ can be larger than ϵ and nothing is said about $d(p_k, p_{\alpha(r)})$. But in Lemma A.4, it is assumed that $d(p_r, p_k) \leq \epsilon$ for all k, $r \leq k < \alpha(r)$ in order to show $d(p_{r'}, p_k) \leq \epsilon$ for all k, $r < r' \leq k \leq \alpha(r)$.

Now we describe the method to determine whether the optimal approximate curve of P_C consists of a single vertex only. If $d(p_0, p_{\alpha(0)}) > \epsilon$, p_0 cannot approximate

P_C. The next candidate that might approximate P_C can be found by searching $p_1, p_2, \ldots$, sequentially for the first vertex p_r whose distance $d(p_r, p_{\alpha(0)}) \leq \epsilon$. From Lemma A.4, $d(p_r, p_k) \leq \epsilon$ for all k, $r \leq k \leq \alpha(0)$. Thus p_r can be considered as if *Compute_α* was started at p_r instead of p_0 and all p_k, $r \leq k \leq \alpha(0)$ have already been tested. Other vertices are tested in a similar way. The time complexity the the method can be analysed in the following way. Between each evaluation of $d(p_r, p_k)$, either r of k must be incremented by one. r range from 0 to $n-1$. k is "ahead" of k by at most n and so k varies from 0 to $n-1$ at most twice. Thus no more than $3n$ evaluations of $d(p_r, p_k)$ are needed. We can conclude that if P_C can be approximated by at most two line segments, the optimal curve can be determined in $O(n)$ time.