



Approximating a neuron with cylindrical segments
by Wenhao Lin

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in
Computer Science

Montana State University

© Copyright by Wenhao Lin (2003)

Abstract:

We study a 3D geometric problem originated from computing, neural maps in the computational biology community: Given a set S of n points in 3D, compute K cylindrical segments (with different radii, orientations and lengths) enclosing S such that the sum of their radii is minimized. There is no known result in this direction except when $K = 1$. When K is not part of the inputs, this problem is strongly NP-hard. In this thesis, we present new approximation algorithms for $K = 1$. We attack the general problem by taking input as a reconstructed 3D polyhedron from the sample points from practice. We apply a semi-automatic method to divide one polyhedron into K parts, so that we can apply our approximation algorithms on each part. At last, we present the design of one software system using these ideas.

APPROXIMATING A NEURON WITH CYLINDRICAL SEGMENTS

by

Wenhao Lin

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

May 2003

N378
L6305

APPROVAL

of a thesis submitted by

Wenhao Lin

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

Binhai Zhu



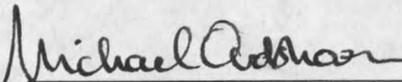
(Signature)

Apr, 21, 03

Date

Approved for the Department of Computer Science

Michael Oudshoorn



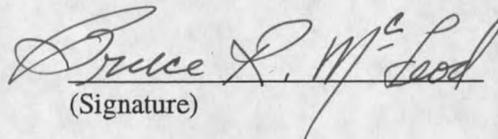
(Signature)

4/21/03

Date

Approved for the College of Graduate Studies

Bruce McLeod



(Signature)

5-19-03

Date

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U. S. Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copyright holder.

Signature Wenhao Lin

Date May 14, 2003

TABLE OF CONTENTS

LIST OF TABLES	v
LIST OF FIGURES	vi
1. INTRODUCTION	1
2. APPROXIMATION ALGORITHMS	4
Approximation Algorithm 1	4
Approximation Algorithm 2	6
Approximation Algorithm 3	10
Approximation Algorithm 4	11
Comparison of Approximation Algorithms	16
3. SYSTEM DESIGN	18
Data Structure	18
Preprocessing	20
3D Rendering	24
Building Local Coordinate System of A 3D Object	27
Building Local Coordinate System of the Virtual Camera	30
Implementing Human-computer Interaction	31
Implementing Navigation Using Keyboard	31
Implementing Picking Using Mouse	32
Implementing Picking-rotation around the Coordinate Origin Using Mouse	35
Implementing Picking-rotation around A Coordinate Axis Using Mouse	37
Implementing Animation	38
Connecting Small Disconnected Components	38
Computing the Smallest Enclosing 2D Disk	39
Heuristic Cutting Algorithm	40
User Interface	43
Further Work	46
4. CONCLUSION	48
REFERENCES CITED	49

LIST OF TABLES

Table	Page
1. Test group 1(1025 points)	17
2. Test group 2(127 points)	17
3. Test group 3(1479 points)	17
4. Contraction edge types and operations	23
5. Navigation function keys	32

LIST OF FIGURES

Figure	Page
1. Approximating a neuron with cylindrical segments	1
2. case 1	5
3. case 2.....	6
4. Illustration for the proof of Lemma 1	7
5. Illustration for the proof of Theorem 3, part 1	8
6. Illustration for the proof of Theorem 3, part 2	9
7. The worst case of Algorithm 4.....	12
8. Projection of one cylinder on a plane	13
9. Local rotation of the center vector	15
10. Data structure	19
11. Edge contraction	21
12. An example scene graph	25
13. Basic rendering concepts in computer graphics.....	26
14. Scene graph of this system	27
15. Object's local coordinate system.....	28
16. Picking	33
17. Matrices in scene graph	36
18. Rotation around an axis	37
19. Project 3D points onto a plane	39
20. Define two points to run the cutting algorithm	41

LIST OF FIGURES — CONTINUE

21. Main window	44
22. Relations among some important classes.....	45

ABSTRACT

We study a 3D geometric problem originated from computing neural maps in the computational biology community: Given a set S of n points in 3D, compute K cylindrical segments (with different radii, orientations and lengths) enclosing S such that the sum of their radii is minimized. There is no known result in this direction except when $K = 1$. When K is not part of the inputs, this problem is strongly NP-hard. In this thesis, we present new approximation algorithms for $K = 1$. We attack the general problem by taking input as a reconstructed 3D polyhedron from the sample points from practice. We apply a semi-automatic method to divide one polyhedron into K parts, so that we can apply our approximation algorithms on each part. At last, we present the design of one software system using these ideas.

CHAPTER 1

INTRODUCTION

In computational biology and medical science, researchers are interested in computing and simulating the behavior of neurons. The geometric shape of a neuron may be complex, but researchers want to use simple objects to approximate it. For example, they can use a set of cylindrical segments to approximate one neuron. Of course, they hope that the errors of such approximation should be minimal. In other words, they want to find the optimal K cylindrical segments to enclose a neuron, such that the sum of their radii is minimal. The idea is illustrated in Figure 1.

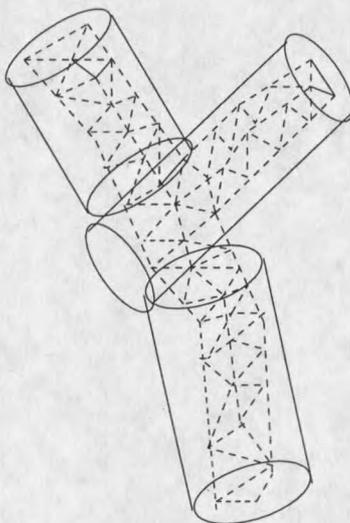


Figure 1. Approximating a neuron with cylindrical segments.

To be more formal, we define the *KCSC* problem as follows.

Optimal K cylindrical segments covering problem(*KCSC*):

Given a 3D point set P , we need to find a set $C = \{c_i, i = 1..K | c_i \text{ is a cylindrical segment, radius of } c_i \text{ is } r_i\}$, s.t. C encloses P and $\sum_{i=1}^K r_i$ is minimized.

In *KCSC*, we not only need to find the number K , but also need to find the K optimal cylindrical segments. This is a hard problem.

Theorem 1. *The KCSC problem is strongly NP-hard [Zhu02].*

In practice, researchers in computational biology and medical science actually solve the *KCSC* problem using a manual method. They use electronic devices to obtain a dense sampling of a neuron's surface. Then they use some commercial software to reconstruct an approximate 3D polyhedron from these sampling points and manually compute several enclosing cylindrical segments which 'seem' to fit the polyhedron best. This is a very time-consuming and error-prone process.

In our project, we use a semi-automatic method. We have designed new approximation algorithms to compute an approximate smallest cylindrical segment enclosing one 3D point set ($K = 1$). Our software system logically can be divided into three modules. One module, with users' input, can divide one polyhedron into K parts. In our system, K is decided by users according to the topology of the polyhedron, i.e, we try to solve a special version of the *KCSC* problem when K is fixed. Another module will compute an approximate smallest enclosing cylindrical segment for each part. The third module is responsible for 3D rendering and user interaction. We have compared the performance of our different approximation algorithms. According to the results of our tests, we have chosen Approximation algorithm 4 with heuristic local rotation as our major working algorithm in this project.

There are some related researches for computing the smallest enclosing cylindrical segment of one point set. Agarwal *et al.* showed an $O(n^{3+\epsilon})$ algorithm and a factor $(1 + \delta)^4$ approximation algorithm with running time $O((1 + 1/\delta^2)n)$ [AAS97]. Schömer *et al.* showed an $O(n^4 \log n^{O(1)})$ algorithm and a factor $(1 + \epsilon)$ approximation algorithm with running time $O(n\epsilon^{-2} \log \epsilon^{-1})$ [SSTY00]. But, when the point set's size is large these algorithms become impractical. Besides, Welzl showed an $O(n)$ algorithm for computing the smallest enclosing disk of a 2D point set [We91]. We will use his algorithm as a subroutine. Lau *et al.* showed a real-time 3D model simplification algorithm using edge contraction [LGTW98]. We will use their algorithm in the preprocessing step.

In the next chapter, we introduce our approximation algorithms for computing the smallest enclosing cylindrical segment of one 3D point set.

¹In this case, a factor $(1 + \delta)$ approximation algorithm means its result is bounded by $(1 + \delta)$ times the optimal result.

CHAPTER 2

APPROXIMATION ALGORITHMS

In practice, when we are faced with very hard problems (e.g, NP-hard or NP-complete problems), instead of trying to find the optimal solutions, we usually design algorithms which can return near-optimal solutions. Such kind of algorithms are called approximation algorithms.

In this chapter, we discuss four approximation algorithms, which compute an approximate smallest cylindrical segment enclosing one part of a polyhedron ¹ (which can be seen as a 3D points set).

The input of these approximation algorithms is a set of 3D points, the output is an enclosing cylindrical segment. For our convenience, let us give some definitions first.

Definitions: *The diameter $D(P)$ of a point set P is the longest distance between any two points in P . The aspect ratio α of a cylinder is its length divided by its width.*

Approximation Algorithm 1

Given a set P of n points, let C^* be the smallest enclosing cylindrical segment of P , A be the approximate smallest enclosing cylindrical segment. We first present a simple approximation algorithm.

Algorithm 1 [Zhu02]:

Step 1 Compute the diameter $D(P)$ of P , let $D(P)$ be $d(p_1, p_2)$.

¹Approximation algorithm 1, 2 were designed by Zhu, implemented and tested by me. Approximation algorithm 3, 4 were joint work of Zhu, Jacobs, Orser and myself.

Step 2 Use p_1p_2 as the center of the approximate cylindrical segment A . The maximal distance between points $q \in P$ and p_1p_2 is the radius of A .

Theorem 2. *Algorithm 1 presents a factor 2 approximation for the smallest enclosing cylindrical segment problem in $O(n \log n)$ time.*

Proof: Let us assume that the smallest cylindrical segment C^* has width r , its center is \bar{C} . Also assume that the point furthest from p_1p_2 is q . The smallest cylindrical segment enclosing just three points p_1, p_2, q has width $d(q, p_1p_2)$. The smallest cylindrical segment enclosing the whole point set must have a larger width. So $d(q, p_1p_2) \leq r, 2 * d(q, p_1p_2) \leq 2 * r$. We use $2 * d(q, p_1p_2)$ as the width of A , so the approximation factor is 2. \diamond

We have two examples, where the worst case happens (the approximate smallest enclosing cylindrical segment's width = $2 * d(q, p_1p_2)$):

Case 1: p_1p_2 is on C^* 's surface, $\triangle qp_1p_2$ contains the center \bar{C} (refer to Figure 2).

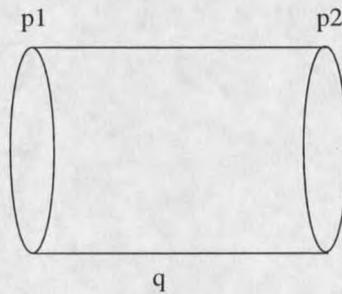


Figure 2. case 1.

Case 2: In this case, there are two lines p_1w and p_2v , which are both on C^* 's surface and parallel to center \bar{C} . The distance between line p_1w and p_2v is r (refer to Figure 3).

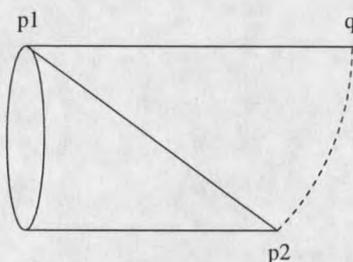


Figure 3. case 2.

Algorithm 1 is not good enough, its approximate enclosing cylindrical segment is too 'fat' according to our tests. So, we want to find a better approximation algorithm. This effort results in Approximation algorithm 2. We introduce the following lemma first.

Lemma 1. *Let u^*v^* be the center of the smallest enclosing cylindrical segment C^* of P . Then the smaller of the angles between p_1p_2 and u^*v^* , θ is at most $\arcsin \frac{1}{\alpha}$.*

Proof: Let us assume that the smallest enclosing cylindrical segment is C^* , the approximate enclosing cylindrical segment computed by Algorithm 1 is C , its center vector is $\overline{p_1p_2}$. We have one important observation. If we know the center vector u^*v^* of C^* , then we can project all the points of P onto a plane vertical to u^*v^* and compute the diameter of the smallest enclosing disk of these 2D projection points. This diameter actually equals to the width of C^* . Let us also assume that there is an angle θ between u^*v^* and p_1p_2 and C has width d . We project points p_1 and p_2 onto a plane vertical to u^*v^* . The projections of p_1, p_2 have distance $L * \sin \theta$. The smallest disk enclosing all the projection points must have diameter $\geq L * \sin \theta$. Then, we have $width(C^*) \geq L * \sin \theta$. We have $width(C) \geq width(C^*) \geq L * \sin \theta$, in other words, $L * \sin \theta \leq d$. So $\sin \theta \leq \frac{1}{\alpha}$. \diamond

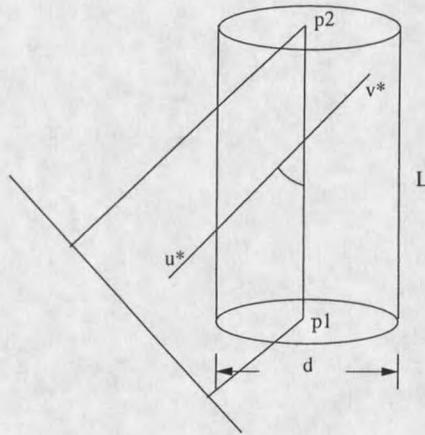


Figure 4. Illustration for the proof of Lemma 1.

Approximation Algorithm 2

In Approximation algorithm 1, we use a point set's diameter as the smallest enclosing cylindrical segment's center vector. But in general, they are different. According to Lemma 1, we know that the intersections of the smallest enclosing cylindrical segment's center vector and the two base planes of the approximate cylindrical segment A computed by Approximation algorithm 1 are bounded. So we try to find the smallest enclosing cylindrical segment's center vector using a grid method.

Algorithm 2 [Zhu02]:

Step 1 Compute the approximate enclosing cylindrical segment A of P using Algorithm 1.

Step 2 Increase the radii of the two circular bases of A by a factor of $t = \frac{\alpha}{\sqrt{\alpha^2 - 1}}$, generate a grid of $\frac{2\sqrt{2}t}{\delta} \times \frac{2\sqrt{2}t}{\delta}$ points on each enlarged base of A . Let the sets of these grid points be G_1 and G_2 respectively.

