Precision-Sensitive Euclidean Shortest Path in 3-Space [*,†]

Jürgen Sellen [‡] Joonsoo Choi [§] Chee-Keng Yap [¶]

Abstract

This paper introduces the concept of *precision-sensitive algorithms*, in analogy to the well-known output-sensitive algorithms. We exploit this idea in studying the complexity of the 3-dimensional *Euclidean shortest path* problem. Specifically, we analyze an incremental approximation approach based on ideas in [CSY], and show that this approach yields an asymptotic improvement of running time. By using an optimization technique to improve paths on fixed edge sequences, we modify this algorithm to guarantee a relative error of $O(2^{-r})$ in a time polynomial in r and $1/\delta$, where δ denotes the relative difference in path length between the shortest and the second shortest path.

Our result is the best possible in some sense: if we have a *strongly precision-sensitive* algorithm then we can show that *USAT* (unambiguous SAT) is in polynomial time, which is widely conjectured to be unlikely.

Finally, we discuss the practicability of this approach. Experimental results are provided.

1 Introduction

1.1 Precision-Sensitivity versus Output-Sensitivity

The complexity of geometric algorithms generally falls under one of two distinct computational frameworks. In the *algebraic framework*, the (time) complexity of an algorithm is measured by the number of algebraic operations (such

[*] **An Extended Abstract of this paper has been published in Procceedings 11th ACM Symposium on Computational Geometry, 1995, pp. 350-359.**

[†] Work on this paper by Sellen has been supported by a postdoctoral fellowship from the DAAD, Germany. Choi and Yap have been supported by NSF grant #CCR-9402464.

[‡] ESG GmbH, Einsteinstrae 174, D-81675 Munich.

[§] Department of Computer Science, Kookmin University, 861-1, Chongnung-Dong, Songbuk-Gu, Seoul 136-702.

[¶] Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012.

as $+, -, x, \div, \sqrt{\cdot})$ on real-valued variables, assuming exact computations. In simple cases, the input size has one parameter n corresponding to the number of input values. In the *bit framework*, (time) complexity is measured by the number of bitwise Boolean operations, assuming input values are encoded as binary strings. The input size parameter n above is usually supplemented by an additional parameter L which is an upper bound on the bit-size of any input value. See [CSY].

Currently, practically every computational geometry algorithm is based on the algebraic model. For instance, we usually say that the planar convex hull problem can be solved in optimal $O(n \log n)$ time. This presumes the algebraic framework. What about the bit framework? One can easily deduce that the bit complexity is $O(n \log n \mu(L))$ where $\mu(L)$ is the bit complexity of multiplying two L-bit integers. However, it is not clear that this is optimal. Thus the possibility for faster planar convex hull algorithms seems wide open in the bit model. Of course, the situation with other problems in computational geometry is similar.

This paper is interested in bit complexity, and may be seen as a follow-up on [CSY]. Besides its inherent interest, there are other reasons for believing that the bit model will become more important for computational geometry in the future. As the field now begins to address implementation issues in earnest, it must focus on low-level operations (what was previously dismissed as "constant time operations"). In low-level operations, it is the bit size of numbers that is the main determinant of complexity. Second, there are reasons to think that "exact computation" (see [Ya]) will be an important paradigm for future implementations of geometric algorithms. [The emphasis here is on "implementations" since exact computation is already the de facto standard in theoretical algorithms.] In exact computation, complexity crucially depends on the bit-sizes of input numbers.

The main conceptual contribution in this paper is the idea of *precision-sensitive* algorithms. Today, the concept of *output-sensitive algorithms* has become an important pillar of computational geometry. But output-sensitivity is basically a concept in the algebraic framework. We suggest that precision-sensitivity is the analogous concept in the bit framework. As in output-sensitive algorithms, we may define some implicit parameter $\delta = \delta(I)$ for any input instance I. Instead of measuring the output size, δ now measures the "precision-sensitivity" of I. Intuitively, the parameter δ measures the precision or number of bits needed for output. We seek to design algorithms that can take advantage of this parameter δ. (Our idea is related to recent work in numerical analysis which quantifies the distance from an input instance to the nearest singularity.)

As an example, consider the well-studied 2-dimensional Euclidean shortest path problem. In the algebraic model, the time complexity of this problem was recently shown to be $O(n \log n)$ [HS], a significant improvement upon the previous $O(n^2)$ techniques. But little is known about this problem in the bit model. Here, the question reduces to whether we can compare the sums of n square roots of integers in polynomial time. This problem may require exponential time because the difference between two such sums, as far as we know, may be as small as $2^{-2^{Cn}}$ for some $C > 0$. Blömer [Bl, Bl2] considers this problem* and its extensions. We may let the precision-sensitive parameter δ be the difference in path length between the sought shortest path and the next shortest path. In practical situations, the gap δ is unlikely to be exponentially small. For such inputs, it may be possible to compute the shortest path in time polynomial in n (the number of obstacle vertices), L (the bit length of input numbers) and δ, provided our algorithm is "precision-sensitive".

The introduction of precision-sensitivity paves the way for studying problems that were previously considered hopeless or "solved". Notice that the same situation arises with the introduction of output-sensitivity. To take one example, the hidden surface elimination which is trivially $\Theta(n^2)$ in the usual complexity model (ergo "uninteresting") becomes very interesting when we consider output-sensitive algorithms. See [Berg, Bern] for some interesting results that exploit output-sensitivity in this problem.

1.2 Precision-Sensitive Approach to 3ESP

This paper focuses on the 3-dimensional *Euclidean shortest path* (3ESP) problem: given a collection of polyhedral obstacles in $\mathbf{R}^3$, and source and target points $s, t \in \mathbf{R}^3$, construct an obstacle-avoiding polygonal path

$$p_{min} = (s, x_1, \ldots, x_k, t), \tag{1}$$

$k \geq 0$, from s to t with minimal Euclidean length. Here, the x_i's are called *breakpoints* of the path, and are required to lie on edges of the obstacles. This problem is ideal for introducing precision-sensitivity because conventional approaches are doomed to failure due to its NP-hardness, a result of Canny and Reif [CR]. It is also useless to introduce output-sensitivity here because the output-size is $O(n)$.

On the other hand, something interesting is going on in the bit model: the algebraic numbers that describe the lengths of the shortest paths may have exponential degrees (see subsection 2.2). This means that to compare the lengths of two *combinatorially distinct* shortest paths may require exponentially many bits. "Combinatorially distinct" means that the respective paths pass through different sequences of edges, and each is shortest for its edge sequence. In this paper, we use the relative difference between the length d_1 of a shortest path and the length d_2 of the combinatorially distinct next shortest path as our measure of "precision-sensitivity"

$$\delta = \delta(I) := (d_2 - d_1)/d_1. \tag{2}$$

*Interestingly, Blömer and Yap [Bl, Bl2] noted that the equality of two sums of square roots can be decided in polynomial time.

It should be noted that δ may be 0. One possibility for $\delta = 0$ is when the shortest path passes through a concave corner. Taking into account of δ is a crucial step towards a practical 3ESP algorithm, but it is not enough.

First we clarify some further aspects of 3ESP. The exponential behavior of 3ESP has two sources: not only is the bit complexity apparently exponential, the number of combinatorially distinct shortest paths can also be exponential. In fact, Canny and Reif's NP-hardness construction exploits the latter property of 3ESP. We can separate the combinatorial aspects from the algebraic aspects as follows. Define the *combinatorial 3ESP problem* which, with input as in 3ESP, asks for a *shortest edge sequence*

$$S_{min} = (e_1, \ldots, e_k), \tag{3}$$

such that $x_i \in e_i$ for $i = 1, \ldots, k$, where the x_i are the breakpoints of some shortest path p_{min} given by (1). Once S_{min} is obtained, there are effective numerical methods to zoom into the actual breakpoints $x_1, \ldots, x_k$, as we shall see. Thus the "purely" numerical part of ESP is delegated to a subsequent phase of computation.

How hard is the combinatorial 3ESP problem? Define the implicit parameter $s(I)$ of an input I to 3ESP to be $s = s(I) = \lfloor \log(|d_1 - d_2|) \rfloor$. We say an algorithm for the combinatorial 3ESP problem is *strongly precision-sensitive* if it is polynomial-time in the parameters n, L, s. By a careful analysis of the Canny-Reif proof, we show:

Theorem 1 *If there exists a strongly precision-sensitive algorithm for the combinatorial 3ESP problem then USAT can be solved in polynomial-time.*

Here *USAT* is the *unambiguous satisfiability problem*, commonly believed not to be in polynomial-time [Pa2, VV]. Note that the parameter $s(I)$ is an absolute measure while our sensitivity parameter $\delta(I)$ is a relative one. But this difference is not crucial. What is more important is the fact that $s(I)$ is roughly logarithmic in $\delta(I)$. In some sense, this theorem justifies our choice of $\delta(I)$.

1.3 Towards a Practical Algorithm

In hopes of developing a "practical algorithm", Papadimitriou [Pa1] introduces the *approximate 3ESP* problem. The input is as in 3ESP plus a new input parameter $\varepsilon > 0$. The problem is to compute an *ε-approximate shortest path*, i.e., one whose length is at most $(1 + \varepsilon)$ times the length of the shortest path. The bit-complexity of this approach is resolved in [CSY], yielding an algorithm with time

$$T(n, M, W) = O((n^3 M \log M + (nM)^2) \cdot \mu(W)), \tag{4}$$

where $M = O(nL/\epsilon)$, $W = O(\log(n/\epsilon) + L)$ and $\mu(W) = O(W \log W \log\log W)$ is the complexity of multiplying two W-bit numbers. Despite initial hopes, this result is still impractical, even for small examples, because the stated complexity is, roughly speaking, achieved for every input instance. Our goal is to remedy this by introducing precision-sensitivity.

Recall that Papadimitriou's approach is to subdivide each obstacle edge into *segments* in a clever way and, by treating these segments as nodes in a weighted graph, to reduce the problem to finding the shortest path in a graph.

In order to introduce precision-sensitivity, we exploit the alternative scheme introduced in [CSY] for subdividing edges into segments. The subdivision is parameterized by a choice of $\epsilon > 0$. Our scheme has the property that the $\epsilon/2$-subdivision is a refinement of the ϵ-subdivision, hence we can incrementally reduce the approximation error. The idea is to discard - in each refinement step - all segments that are provably not used by the shortest path; what remains are called *essential* segments. While it is obvious that such an implementation can drastically decrease running time in practice, we show that – depending on the parameter δ – this improvement is also asymptotical.

Assuming *non-degeneracy* (see section 2.1) of S_{min} in (3), we prove the following theorem:

Theorem 2 *There is an incremental algorithm to compute an ε-approximate shortest path in time that is polynomial in $1/\delta$ and $1/\varepsilon$. Omitting logarithmic factors, the dependency on $1/\varepsilon$ is only linear rather than quadratic.*

In case the shortest path sequence S_{min} is unique (*i.e.*, $\delta > 0$), we can use techniques from mathematical optimization as soon as we have reached a refinement in which only S_{min} is left. The convergence depends on the spectral bounds

$$\mu, \quad \rho$$

corresponding to the minimum and maximum (respectively) eigenvalue of the Hessian H of the path length function $l(\lambda_1, \ldots, \lambda_k)$, where $\lambda_1, \ldots, \lambda_k \in \mathbf{R}$ parameterize the points $x_1, \ldots, x_k$ on S_{min}.

Theorem 3 *The length of the shortest path can be approximated to relative error ε in time polynomial in $1/\delta, \log(1/\varepsilon), n, L$ and the spectral bounds μ, ρ.*

This theorem, and the remark in theorem 2 about a linear dependency on $1/\varepsilon$ are of practical significance.

It is important to note that the given running times in theorem 2 and 3 are upper bounds, they are tight only for $\varepsilon \leq \delta$. For $\varepsilon > \delta$, and in particular for $\delta = 0$, the running time of both algorithms can be bounded by the running time of the non-incremental approach in [CSY].

In section 5, we shall provide some experimental results, addressing the practicability of the incremental technique.

2 Preliminaries

Throughout the paper, we assume that the input is given by a source point s, a target point t, and a set of pairwise disjoint polyhedral obstacles, with a total of less than n edges. For each obstacle edge e, denote its endpoints by $s(e)$, $t(e)$ and write $e = \overline{s(e)t(e)}$. Let $[e]$ denote the infinite line through e. We assume that s, t as well as endpoints of edges are specified by L-bit rational numbers. For any point $q \in \mathbf{R}^3$, $||q||$ denotes its Euclidean norm. The scalar product of two k-tuples x, y is denoted $\langle x, y \rangle$.

2.1 Basic Properties

We assume the notation in the preceding introduction. In particular, p_{min} is a *global shortest path* from s to t in the free space FS defined by the obstacles. Here, FS is defined as the closure of the complement of the union of the obstacles.

First we fix an *edge sequence* $S = (s, e_1, \ldots, e_k, t)$. The sequence S is *degenerate* if $s \in [e_1]$, $t \in [e_k]$, or $[e_j] = [e_{j+1}]$ for some $j \in \{1, \ldots, k-1\}$. Note that non-degeneracy of S excludes two edges e_i and e_j from lying in a common line $[e_i] = [e_j]$ only when $|i - j| = 1$, but not if $|i - j| > 1$.

A path

$$p = (s, x_1, \ldots, x_k, t), \tag{5}$$

is called an *S-path* if $x_i \in [e_i]$ for all i. An S-path p is *admissible* if $x_i \in e_i$ for all i.

A breakpoint x_i of p that lies on the line between its neighboring vertices, $x_i \in \overline{x_{i-1}x_{i+1}}$, is called *redundant*. W.l.o.g. we may assume that p_{min} in (1) contains no redundant vertices.

We will parameterize points $x_i \in [e_i]$ by a scalar λ_i according to the equation

$$x_i = s(e_i) + \lambda_i u(e_i), \text{ with } u(e_i) = \frac{t(e_i) - s(e_i)}{||t(e_i) - s(e_i)||}.$$

Let $x_0 = s$ and $x_{k+1} = t$. Then the polygonal path $p = (s, x_1, \ldots, x_k, t)$ over S has length

$$l_S(\lambda_1, \ldots, \lambda_k) = \sum_{i=0}^{k} ||x_{i+1} - x_i||.$$

We also write $|p|$ for $l_S(\lambda_1, \ldots, \lambda_k)$. Let $p_{min}(S)$ be defined to be the path p over S that minimizes the function $l_S(\lambda_1, \ldots, \lambda_k)$, *without consideration of the obstacles and without requiring admissibility.*

A necessary condition for $l_S : \mathbf{R}^k \to \mathbf{R}$ to take its global minimum at $\lambda = (\lambda_1, \ldots, \lambda_k)$ is that all partial derivatives vanish at λ. This condition can be interpreted as *Snell's law*, and, as the next lemma will reveal, is also a sufficient condition to specify shortest paths:

Lemma 1 *The function $l_S : \mathbf{R}^k \to \mathbf{R}$ is convex. If the shortest path over the lines $[e_i]$ has no redundant breakpoints, then l_S has a unique minimizer $\zeta \in \mathbf{R}^k$.*

Proof: (1) Let $l = l_S = \sum_{i=0}^{k} l_i$, where

$$l_i(\lambda_1, \dots, \lambda_k) := ||x_{i+1} - x_i||.$$

We may interpret l_i as a function in 2 variables λ_i and λ_{i+1} (unless $i = 0$ or k, in which case l_i depends on a single variable λ_1 or λ_k).

To show that l is convex, it suffices to show that each of the l_i is convex (the sum of convex functions is convex). The convexity of l_i is a special case of a general result in convex analysis: for any norm $||.|| : \mathbf{R}^k \to \mathbf{R}$ and any linear function $f : \mathbf{R}^m \to \mathbf{R}^k$, the function $||f|| : \mathbf{R}^m \to \mathbf{R}$ is convex (see e.g. [Ro]).

(2) The convexity of l guarantees that every local minimum of l is a global minimum, say, d_S, and that the set of points $\lambda \in \mathbf{R}^k$ satisfying $l(\lambda) = d_S$ (the set of minimizers) is convex.

Assume that there are two distinct minimizers $\zeta_1, \zeta_2 \in \mathbf{R}^k$. Then every $\mu(t) = (\mu_1(t), \dots, \mu_k(t)) := \zeta_1 + t(\zeta_2 - \zeta_1)$, $t \in [0,1]$, is a minimizer, and hence $l(\mu(t)) \equiv const$. But

$$l(\mu(t)) = \sum_{i=0}^{k} l_i(\mu(t)),$$

where

$$l_i(\mu(t)) = \sqrt{A_i(t - B_i)^2 + C_i}$$

with $A_i \geq 0$ and $C_i \geq 0$.
This can be constant only if each of the functions $l_i(\mu(t))$ is linear in t, i.e., $A_i = 0$ or $C_i = 0$ (this directly follows from $\partial^2 l(\mu(t))/\partial t^2 \equiv 0$).

Now let j be the first index for which $\mu_i(t)$ is not constant, i.e., $\mu_i(t) \equiv const$ $\forall\ i = 1, \dots, j-1$ and $\mu_j(t) \not\equiv const$ (j may be equal to 1).

The fact that $l_j(\mu(t))$ is linear then implies that $x_{j-1} \in [e_j]$: the point $x_j(t) := s(e_j) + \mu_j(t)u(e_j)$ is moving on the line $[e_j]$ while keeping distance $l_j(\mu(t))$ to the fixed point $x_{j-1} = s(e_{j-1}) + \mu_{j-1}(t)u(e_{j-1})$.

Thus x_{j-1} and $x_j(t)$ lie on the same line $[e_j]$ for all $t \in [0,1]$. From Snell's law it follows that also $x_{j+1}(t)$ must lie on this line, showing that the vertex $x_j(t)$ is redundant. □

Note that, in contrast to this proof, the known proof for the uniqueness of the shortest path [SS] (see also [Ya, appendix]) uses geometrical arguments.

Lemma 2 *Let $S = (s, e_1, \dots, e_k, t)$ be non-degenerate.*
(i) If $[e_i] \cap [e_{i+1}] = \emptyset$ for all i then the Hessian $H = H(\lambda)$ of $l_S = l_S(\lambda)$ is positive-definite.
(ii) If a path $p = (s, x_1, \dots, x_k, t)$ is such that

$$x_i \notin [e_{i-1}] \cup [e_{i+1}], \quad i = 1, \dots, k,$$

then $H(\lambda)$ is locally positive-definite at p.

Proof: The Hessian $H = H(\lambda)$ of $l = l_S$ is a tridiagonal $k \times k$-matrix

$$H = \begin{pmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & b_2 & \ddots & & \\ & & & & b_{k-1} \\ & & & b_{k-1} & a_k \end{pmatrix}.$$

Let H_i be the Hessian of the function $l_i = ||x_{i+1} - x_i||$, interpreted as function over λ_i and λ_{i+1} if $1 \le i \le k-1$, and over λ_1 (resp., λ_k) if $i = 0$ (resp., $i = k$):

$$H_0 = (a_1^-), \quad H_i = \begin{pmatrix} a_i^+ & b_i \\ b_i & a_{i+1}^- \end{pmatrix}, \quad H_k = (a_k^+),$$

with

$$a_i^+ = \frac{\partial^2 l_i}{\partial \lambda_i^2}\ , \ a_{i+1}^- = \frac{\partial^2 l_i}{\partial \lambda_{i+1}^2}\ , \ b_i = \frac{\partial^2 l_i}{\partial \lambda_i \partial \lambda_{i+1}}.$$

Then H and the H_i are related by $a_i = a_i^- + a_i^+$. Abusing the notation, we may write $H = H_0 + \ldots + H_k$.

To prove that H is positive-definite, it remains to show that the determinant of H is not zero. Write

$$H = \begin{pmatrix} a_1 & b_1 & \\ b_1 & & \\ & & H' \\ & & \end{pmatrix}, \quad H^+ = \begin{pmatrix} a_1^+ & b_1 & \\ b_1 & & \\ & & H' \\ & & \end{pmatrix}.$$

where H' is the matrix obtained by deleting the first row and first column of H. Then

$$det(H) = det(H^+) + a_1^- \, det(H').$$

As all H_i are positive-semi-definite, it follows $H^+ = H_1 + \ldots + H_k$ is positive-semi-definite, and thus $det(H^+) \ge 0$.
This implies

$$det(H) \ge a_1^- \, det(H').$$

Continuing recursively, we finally get

$$det(H) \ge a_1^- \cdot \ldots \cdot a_k^-.$$

Abbreviating

$$v_i = \frac{x_{i+1} - x_i}{||x_{i+1} - x_i||},$$

we have

$$a_i^- = \frac{1 - \langle v_{i-1}, u(e_i)\rangle^2}{||x_i - x_{i-1}||}.$$

This is strictly positive under conditions (i) or (ii) in the statement of the lemma. Hence $det(H) > 0$. $\square$

2.2 Bit Complexity

The goal of this subsection is to provide some background on the algebraic complexity of 3ESP.

First, we specify shortest paths over edge sequences algebraically. Let $S = (s, e_1, \ldots, e_k, t)$ be a fixed edge sequence.

For given intervals $I_i \in \{ \{0\}, \{|e_i|\}, [0, |e_i|] \}$, we define a Boolean formula (in the free variables $\lambda_1, \ldots, \lambda_k$)

$$B_S(I_1, \ldots, I_k) : \quad \bigwedge_{i=1}^{k} (Essential(e_i) \ \wedge \ Optimal(e_i))$$

with the predicate

$$Essential_i \Leftrightarrow \langle \frac{x_{i+1} - x_i}{||x_{i+1} - x_i||}, \frac{x_i - x_{i-1}}{||x_i - x_{i-1}||} \rangle^2 \neq 1$$

specifying that x_i is non-redundant, and the predicate

$$Optimal_i \Leftrightarrow \begin{cases} Snell(e_i)\,, I_i = [0, |e_i|] \\ \lambda_i = 0 \quad\ , I_i = \{0\} \\ \lambda_i = |e_i|\,, I_i = \{|e_i|\} \end{cases}$$

specifying that – according to I_i – the point x_i is fixed at an endpoint of e_i or obeys Snell's law $Snell(e_i)$ at some point in the relative interior of e_i (so I_i only serves as a 'flag').

It is obvious that the predicates $Essential_i$ and $Optimal_i$ can be written as a Boolean combination of a constant number of polynomial inequalities of bounded degree over the variables $\lambda_{i-1}, \lambda_i, \lambda_{i+1}$ and a constant number of 'additional variables'. (Note that roots $\sqrt{r}$ can be eliminated by introducing a new variable a, substituting $\sqrt{r}$ by a and adding $(a^2 = r \ \wedge \ a \geq 0)$ to the formula. The variable a may be introduced by the quantifier $\exists$.)

As a corollary of lemma 1, we get

Lemma 3 *For any fixed edge sequence S and intervals $I_1, \ldots, I_k$, the formula $B_S(I_1, \ldots, I_k)$ is satisfied by at most one algebraic point $\lambda = (\lambda_1, \ldots, \lambda_k)$. If $p_{min}(S)$ is a shortest path over S with non-redundant vertices $x_i = s(e_i) + \lambda_i u(e_i)$, then there exist intervals $I_1, \ldots, I_k$ such that λ satisfies $B_S(I_1, \ldots, I_k)$.*

In particular, there exists a sequence $S = (s, e_1, \ldots, e_k, t)$ and intervals $I_1, \ldots, I_k$ such that the shortest path p_{min} is parameterized by the single solution λ of $B_S(I_1, \ldots, I_k)$.

To derive some upper bounds on the bit complexity of the 3ESP problem, we shall use several results on quantifier elimination and root separation (see e.g. [Re]):

- Quantifier elimination:
  Given a Tarski formula with free variable y

$$\text{(P)} \quad \exists x \in \mathbf{R}^n : \bigvee_i \bigwedge_j p_{ij}(x, y) \Delta_{ij} 0$$

  with $\leq m$ polynomials $p_{ij}(x, y)$ each of degree $\leq d$, with integer coefficients of bit-size $\leq L$, and $\Delta_{ij} \in \{>, \geq, =\}$, then there exists an equivalent predicate

$$\text{(P')} \quad \bigvee_i \bigwedge_j h_{ij}(y) \Delta'_{ij} 0$$

  with $(md)^{O(n)}$ polynomials h_{ij} of degree $(md)^{O(n)}$ and coefficients of bit-size $L(md)^{O(n)}$. ('Equivalent' means that (P) is true for a fixed $y = c$ if and only if (P') is true for $y = c$.) The predicate (P') can be constructed in time polynomial in L and $(md)^{O(n)}$.

- Cauchy's bound:
  Given any univariate polynomial $A(y) = \sum_{i=0}^{d} a_i y^i$ with integer coefficients a_i of bit-size $\leq L$, every root $\alpha \neq 0$ of A satisfies $|\alpha| \geq 2^{-2L}$.

- Root separation:
  If α and β are two distinct roots of $A(y)$, then $|\alpha - \beta| \geq (d2^L)^{-Cd}$ (for some $C > 0$ that does not depend on A). Isolating intervals for all roots of $A(y)$ can be computed in time polynomial in L and d.

Let us consider the formula $B_S(I_1, \ldots, I_k)$ from lemma 3. With the above results we immediately get:

Lemma 4 *Every coordinate λ_i of a parameter tuple λ satisfying $B_S(I_1, \ldots, I_k)$ has a defining polynomial h_{λ_i} of degree $n^{O(n)}$ with integer coefficients of bit-size $Ln^{O(n)}$. The polynomial h_{λ_i} together with an isolating interval for λ_i can be computed in time polynomial in L and $n^{O(n)}$.*

Proof: Consider the formula

$$\text{(P1)} \quad \exists\lambda_1 \ldots \exists\lambda_{i-1} \exists\lambda_{i+1} \ldots \exists\lambda_k : \; B_S(I_1, \ldots, I_k).$$

The formula (P1) contains $O(n)$ polynomials in $O(n)$ variables of bounded degree with rational (resp., integer) coefficients of size $O(L)$. The stated quantifier elimination result provides $n^{O(n)}$ polynomials h_{ij} of degree $n^{O(n)}$, with integer coefficients of bit-size $Ln^{O(n)}$.

If there is a tuple λ satisfying (P1), then λ_i will be a root of one of the polynomials h_{ij} (recall that λ is unique).

Now consider the product

$$h := \prod_{i,j} h_{ij}.$$

Using root separation, we compute isolating intervals for each of the $n^{O(n)}$ roots of h. For each root, we can check if it satisfies (P1).

Clearly, the whole computation can be done in time polynomial in L and in $n^{O(n)}$. □

To determine which choice of intervals $I_1, \ldots, I_k$ specifies the shortest path over a given sequence S, and to determine the shortest path p_{min}, we need to compute and compare shortest path lengths.

Lemma 5 *Let λ and λ' be solutions of $B_S(I_1, \ldots, I_k)$ and $B_{S'}(I'_1, \ldots, I'_{k'})$, respectively, and let p and p' be the corresponding paths. Then $|p| = |p'|$ or*

$$|(|p| - |p'|)| \geq 2^{-Ln^{Cn}}$$

(for a global constant $C > 0$).

Proof: The difference in path length between p and p' is the unique solution y of

$$\begin{aligned} &\text{(P2) } \exists \lambda_1 \ldots \lambda_k \exists \lambda'_1 \ldots \lambda'_{k'} : \\ &\quad (y = \sum_{i=0}^{k} ||x_{i+1} - x_i|| - \sum_{i=0}^{k'} ||x'_{i+1} - x'_i||) \\ &\quad \wedge\ B_S(I_1, \ldots, I_k)\ \wedge\ B_{S'}(I'_1, \ldots, I'_{k'}) \end{aligned}$$

The formula (P2) can again be written as a Tarski formula. Analogous to lemma 4, one can use quantifier elimination to obtain a polynomial h with $h(y) = 0$. The coefficients of h are of bit-size $Ln^{O(n)}$. The claim follows immediately from Cauchy's bound. □

In order to actually compute the shortest path p_{min}, we have to filter out those shortest paths, or solutions to $B_S(I_1, \ldots, I_k)$, which would collide with obstacles. Having calculated the parameter λ satisfying $B_S(I_1, \ldots, I_k)$, this amounts to answering the query '$\overline{x_i x_{i+1}} \in FS$?', for $i = 0, \ldots, k$. But this query can be expressed as Tarski sentence in a fixed number of variables, and can be decided in time polynomial in L and $n^{O(n)}$. We finally obtain:

Theorem 4 *It is possible to compute algebraic representations of all combinatorially distinct shortest paths in time polynomial in L and $n^{O(n)}$.*

Here, we may assume that each shortest path is represented by a sequence $(S, I_1, \ldots, I_k, \lambda_1, \ldots, \lambda_k)$ where $S = (s, e_1, \ldots, e_k, t)$ is an edge sequence, the I_j's are interval flags for S, and the λ_j satisfy the formula $B_S(I_1, \ldots, I_k)$. Furthermore, each λ_j is represented by one of its isolated interval representations.

3 Combinatorial 3ESP is as hard as USAT

Recall that the exponential complexity of the 3ESP problem has a combinatorial and an algebraic source. We give evidence that 3ESP remains intractable even after eliminating the algebraic source of complexity.

We briefly review the Canny-Reif construction ([CR], section 2.5): Given a 3SAT-formula f in conjunctive form with m clauses and n variables $b_1, \ldots, b_n$, it is possible to construct an environment $E(f)$ such that the following holds for a fixed "reference length" $l = 2^{3n}$, and $\Delta = 2^{-nm-3n-4}$: To each instantiation of $(b_1, \ldots, b_n)$, there corresponds an edge sequence $S = S(b_1, \ldots, b_n)$ such that the shortest path p over S lies in free space and satisfies

$$|p| \in \begin{cases} [l, l+\Delta] & \text{if } f(b_1, \ldots, b_n) = 1, \\ [l+2\Delta, \infty) & \text{if } f(b_1, \ldots, b_n) = 0. \end{cases}$$

The number of edges of $E(f)$ as well as the maximal bit-size of coordinates is polynomial in n and m. Deciding the satisfiability of f is reduced to deciding if the shortest path in $E(f)$ has length $\leq l + \Delta$.

A careful analysis shows the following property of $E(f)$: if the formula f is unique satisfiable, i.e., by exactly one instantiation of $(b_1, \ldots, b_n)$, then the shortest path in $E(f)$ is unique and the gap in length between this path and any path that passes over a different edge sequence is single-exponential (i.e., $> c^{-nm}$ for some $c > 1$).

The argument is as follows: The basic construction elements in [CR] are parallel, 2-dimensional plates with (for ease of description) 1-dimensional slots. The construction is based on a scene with 2^n shortest paths, with length $l' \leq l + \Delta$. In the final step, obstacles are introduced which stretch all paths that correspond to non-satisfying instances by at least Δ. It remains to verify that there are no further locally shortest paths that use other edge sequences and have length close to l'. The use of parallel plates ensures that these paths would have additional legs between slots. The spacing between plates and between the break points of the shortest paths in the slots gives a lower bound on the additional length, and is again roughly Δ. Finally, the gap Δ is single-exponential.

Now assume that we have a strongly precision-sensitive algorithm as defined in subsection 1.2. Consider the satisfiability problem restricted to 3SAT formulas that are satisfiable by at most one variable instance, known as the *unambiguous satisfiability problem USAT*. Assume we are given such a formula f. By constructing $E(f)$ and running our algorithm, we would be able to decide the satisfiability of f in polynomial time. This proves theorem 1.

4 Approximation

For simplicity, we shall describe algorithms in this section in the algebraic framework. It is important to note that the hardness result of section 3 is not valid in this model. However, as in [CSY], the technique extends to the bit framework. In particular, it suffices to compute intermediate numbers to precision $W = O(\log(n/\varepsilon) + L)$.

We review the approximation scheme for 3ESP in [CSY]: the algorithm mainly consists of three steps. In the first step, the edges are subdivided into segments using a method that depends on some given parameter $\varepsilon' > 0$. This *ε'-subdivision* (as it is called) satisfies the following properties:

Lemma 6 ([CSY])
(1) Each edge is divided into $O(L/\varepsilon')$ segments.
(2) Each segment σ of the subdivision satisfies $|\sigma| \leq \varepsilon' dist(s, \sigma)$.
(3) The $\varepsilon'/2$-subdivision is a refinement of the ε'-subdivision.

In the second step of the algorithm, the *visibility graph* $G_0 = (V_0, E_0)$ of the segments is constructed. The nodes of the graph comprise the subdivision segments including s and t. The edges comprise pairs (σ, σ') of segments that can "see each other", meaning that there exists $x \in \sigma$ and $x' \in \sigma'$ such that $\overline{xx'} \in FS$. In the third step, the visibility graph G_0 is weighted by assigning to each edge (σ, σ') the Euclidean distance between the midpoints of σ and σ'. Finally, the shortest path Σ in G_0 is computed by running Dijkstra's shortest path algorithm. This path is a *segment sequence* $\Sigma = (s, \sigma_1, \ldots, \sigma_k, t)$. Its "weight' according to the midpoint distances is further denoted as $|\Sigma|$.

The following lemma relates $|\Sigma|$ to $|p_{min}|$ (and shows the correctness of the approximation scheme):

Lemma 7 *For $\varepsilon' = \varepsilon/Cn$, C a given constant, Σ satisfies $|\Sigma| \leq (1+2\varepsilon/C)|p_{min}|$ and $|p_{min}| \leq (1 + 2\varepsilon/C)|\Sigma|$.*

Proof: Consider the path Σ_{min} in G_0 which corresponds to p_{min} (Σ_{min} is equivalent to a path that connects the midpoints of the segments used by p_{min}). By the triangle inequality, the weight of each leg (σ_i, σ_{i+1}) of Σ_{min} can be bounded by the length of the corresponding leg of p_{min}, plus the length of the segments σ_i and σ_{i+1}. Hence, we obtain

$$|\Sigma| \leq |\Sigma_{min}| \leq |p_{min}| + 2\sum_{j=1}^{k} |\sigma_j|.$$

With $k \leq n$, $|\sigma_j| \leq \varepsilon' \mathrm{dist}(s, \sigma_j)$, and $\mathrm{dist}(s, \sigma_j) \leq |p_{min}|$, we get

$$|\Sigma| \leq (1 + 2\varepsilon/C)|p_{min}|.$$

To prove the second inequality, we consider the path p over Σ which connects pairwise visible points $x_i^1 \in \sigma_i$, $x_{i+1}^2 \in \sigma_{i+1}$, and which connects the points x_i^1, x_i^2 on each σ_i by additional legs. By the triangle inequality, we obtain

$$|p_{min}| \le |p| \le |\Sigma| + 2\sum_{j=1}^{k} |\sigma_j|.$$

With $k \le n$, $|\sigma_j| \le \varepsilon' \mathrm{dist}(s, \sigma_j)$, and $\mathrm{dist}(s, \sigma_j) \le |\Sigma|$, we finally get

$$|p_{min}| \le (1 + 2\varepsilon/C)|\Sigma|.$$

□

4.1 An Incremental Algorithm

The above algorithm uses a fixed subdivision. In the following, we shall exploit property (3) in lemma 6 by successively halving the error bound ε, and by refining only those segments which the global shortest path could potentially use.

Let $\varepsilon_i = 2^{-i}$ and $\varepsilon_i' = \varepsilon_i/Cn$, for the fixed constant $C = 32$. (This is a significant improvement to the conference version of this paper where we divide by Cn^2 instead of Cn.) Let $G_i = (V_i, E_i)$ be the weighted visibility graph for any set of segments V_i fulfilling the basic inequality (2) of lemma 6, and let l_i denote the length of the shortest path from s to t in G_i. By lemma 7, we get $l_i \le (1 + \varepsilon_i/16)|p_{min}|$ and $|p_{min}| \le (1 + \varepsilon_i/16)l_i$.

Lemma 8 *If $\Sigma = (s, \sigma_1, \ldots, \sigma_k, t)$, $k \le n$, is a path in G_i with $|\Sigma| > (1+\varepsilon_i/4)l_i$, then any path p over Σ satisfies $|p| > |p_{min}|$.*

Proof: Assume $|p| \le |p_{min}|$. By the triangle inequality, we get

$$|\Sigma| \le |p| + 2\sum_{j=1}^{k} |\sigma_j|.$$

With $k \le n$, $|\sigma_j| \le \varepsilon_i \mathrm{dist}(s, \sigma_j)/32n$ and $\mathrm{dist}(s, \sigma_j) \le |p| \le |p_{min}|$, we get

$$|\Sigma| \le (1 + \varepsilon_i/16)|p_{min}|.$$

With $|p_{min}| \le (1 + \varepsilon_i/16)l_i$, we finally get

$$|\Sigma| \le (1 + \varepsilon_i/4)l_i,$$

contradiction. □

We define the *essential subgraph* $G_i^{ess} = (V_i^{ess}, E_i^{ess})$ of G_i to be the subgraph which is spanned by the union of all (s,t)–paths Σ in G_i with $|\Sigma| \le (1 + \varepsilon_i/4)l_i$.

Corollary 1 *If p_{min} leads over a segment sequence $\Sigma = (s, \sigma_1, \ldots, \sigma_k, t)$ in G_i, then Σ is in G_i^{ess}.*

To approximate a shortest path p_{min} by successive refinement, we need thus only to consider the segments in V_i^{ess} in the next step.

We can compute G_i^{ess} as follows: run Dijkstra's single source shortest path algorithm on G_i twice, starting at s and starting at t, and assign to each $\sigma \in V_i$ the distances $d_s(\sigma)$ (resp., $d_t(\sigma)$) to s (resp., t) in G_i. This implies $l_i = d_s(t) = d_t(s)$. Let the weight of edge (σ, σ') in G_i be denoted by $\omega(\sigma, \sigma')$. Then we can choose E_i^{ess} to be the set of all $(\sigma, \sigma') \in E_i$ that satisfy

$$d_s(\sigma) + d_t(\sigma') + \omega(\sigma, \sigma') \leq (1 + \varepsilon_i/4)l_i.$$

In practice, G_i^{ess} should be significantly smaller than G_i. In fact, it approaches the 1-dimensional skeleton formed by all global shortest paths as $\varepsilon_i \to 0$. In the next lemma we show that, depending on the precision-sensitivity parameter δ, the incremental construction of G_i^{ess} will eventually resolve the edge sequence S_{min} of the global shortest path p_{min}:

Lemma 9 *Let $\varepsilon_i < \delta$ and let (σ, σ') be an arbitrary edge of G_i^{ess} with $\sigma \in e, \sigma' \in e'$ (e and e' are obstacle edges). Then either $e = e'$ or (e, e') is an edge of S_{min}.*

Proof: The graph G_i^{ess} contains a path $\Sigma = \Sigma_1 \cdot (\sigma, \sigma') \cdot \Sigma_2$, where Σ_1 is a shortest path from s to σ and Σ_2 is a shortest path from σ' to t. By construction of G_i^{ess}, Σ satisfies $|\Sigma| \leq (1 + \varepsilon_i/4)l_i$. Let $p = p_1 \cdot (x, x') \cdot p_2$ be an admissible path over Σ, i.e., a path which realizes the visibility relation (p is a zig-zag path which uses additional legs on segments). As Σ_m, $m = 1, 2$, is a shortest path in G_i^{ess}, Σ_m enters and leaves an obstacle edge e at most once (else, there would be a cycle and a shortcut on e). Hence, each Σ_m leads over $k \leq n$ segments σ_j. By the triangle inequality, we get

$$|p| \leq |\Sigma| + 4 \sum_{j=1}^{k} |\sigma_j|.$$

With $|\sigma_j| \leq \varepsilon_i \mathrm{dist}(s, \sigma_j)/32n$ and $\mathrm{dist}(s, \sigma_j) \leq |\Sigma|$, we get

$$|p| \leq (1 + \varepsilon_i/4)|\Sigma|.$$

With $|\Sigma| \leq (1 + \varepsilon_i/4)l_i$ and $l_i \leq (1 + \varepsilon_i/4)|p_{min}|$, we obtain $|p| < (1 + \varepsilon_i)|p_{min}|$. By definition of δ, p must lie on S_{min}. Finally, the edge (σ, σ') used by p must lie on the same obstacle edge e of S_{min} or must correspond to an edge (e, e') of S_{min}. □

We are now ready to formulate the *incremental algorithm* to get a relative error of $\varepsilon = 2^{-r}$:

```
i := 0; ε'_0 := 1/Cn;
Compute the initial ε'_0-subdivision V_0;
Repeat
    Construct the visibility graph G_i = (V_i, E_i);
    Compute G_i^ess = (V_i^ess, E_i^ess);
    Compute V_{i+1} by refining V_i^ess;
    i := i + 1; ε'_i := ε'_{i-1}/2;
Until i = r + 1.
```

4.2 Spectral Analysis

Our first goal in this subsection is to characterize the behavior of the incremental algorithm for a fixed edge sequence S.

Let $l = l_S$, and let again H be the Hessian of l with spectral bounds μ and ρ. Let $\zeta = (\zeta_1, \ldots, \zeta_k)$ be the parameter tuple specifying $p_{min}(S)$, and z_j the break point of $p_{min}(S)$ on e_j, specified by ζ_j. Our goal is to show that a path p over S whose length differs only slightly from $|p_{min}(S)|$ must also have a parameter λ which is close to ζ. Taylor's theorem shows that for any $\lambda \in \mathbf{R}^k$, there exists $\tau \in \mathbf{R}^k$ such that

$$l(\lambda) - l(\zeta) = \langle \nabla l(\zeta), \lambda - \zeta \rangle + \frac{1}{2} \langle \lambda - \zeta, H(\tau)(\lambda - \zeta) \rangle. \tag{6}$$

The first term is equal to zero as ζ minimizes the function l_S (see section 2.1). The second term can be bounded by the "spectrum" of H:

$$\mu ||\lambda - \zeta||^2 \;\leq\; \langle \lambda - \zeta, H(\tau)(\lambda - \zeta) \rangle \;\leq\; \rho ||\lambda - \zeta||^2. \tag{7}$$

This implies

$$l(\lambda) - l(\zeta) \geq \frac{\mu}{2} ||\lambda - \zeta||^2.$$

Thus, the parameter λ of any path p over S with $|p| - |p_{min}(S)| \leq \tilde{\varepsilon}$ satisfies $||\lambda - \zeta||^2 \leq 2\tilde{\varepsilon}/\mu$ (for any $\tilde{\varepsilon} > 0$).

In the next lemma, we consider G_i^{ess} after the edge sequence S_{min} of the unique shortest path has been resolved:

Lemma 10 *Let $\varepsilon_i < \delta$, let $\sigma \in G_i^{ess}$ be a segment with $\sigma \subseteq e$, let $x \in \sigma$ be an arbitrary point, and let $z \in e$ be the breakpoint of the shortest path p_{min} on e. Then*

$$||x - z|| \leq (2\varepsilon_i \frac{|p_{min}|}{\mu})^{\frac{1}{2}}.$$

Proof: By lemma 9, we can assume that $x = x_j$ is the j-th vertex of a path p over S_{min} with $|p| \leq (1 + \varepsilon)|p_{min}|$. Accordingly, let $z = z_j$ be the j-th vertex of p_{min}. Now, let λ be the parameter of p, and let ζ be the parameter of p_{min}.

Setting $\tilde{\varepsilon} = \varepsilon_i |p_{min}|$, we get

$$\begin{aligned} ||x_j - z_j|| &= |\lambda_j - \zeta_j| \le ||\lambda - \zeta|| \\ &\le (2\varepsilon_i \frac{|p_{min}|}{\mu})^{\frac{1}{2}}. \end{aligned}$$

□

Now assume we run our incremental algorithm for the fixed edge sequence S_{min}, and are in step i. Then on each edge e_j, those segments whose distance from z_j is more than $const \cdot \sqrt{\varepsilon_i}$ will automatically not be considered. Here, $const = \sqrt{2|p_{min}|/\mu}$ depends on S but not on i.

By construction, each segment σ on e_j has length $|\sigma| \ge a_j \varepsilon_i / 32n$ (with a_j the distance from e_j to the source s). Thus we refine at most

$$C(S) \cdot n{\varepsilon_i}^{-\frac{1}{2}}$$

segments on each e_j in the i-th step, with

$$C(S) = \frac{32}{a}\sqrt{2\frac{|p_{min}|}{\mu}}$$

and $a = min_j\{a_j\}$.

Let $\varepsilon = 2^{-r}$ be the desired relative error. Summing over $i = 1, \ldots, r$, we produce a total of $O(r\sqrt{1/\varepsilon_r})$ segments. This is a significant improvement to the original (non-precision sensitive) scheme, which would produce $O(1/\varepsilon_r)$ segments.

The described effect occurs in the overall algorithm as soon as $\varepsilon_i < \delta$. Lemma 9 and lemma 10 then imply that the essential subgraph G_i^{ess} contains less than $O(nC(S_{min})2^{i/2})$ segments per edge. On the other hand, if $\varepsilon_i > \delta$, then G_i^{ess} contains $O(nL/\varepsilon_i) = O(nL/\delta)$ segments per edge. (This is the number of segments produced by the non-incremental scheme in [CSY].) This yields:

Lemma 11 *The essential subgraph G_i^{ess} contains less than*

$$M_i = O(n(\frac{L}{\delta} + C(S_{min}) \cdot 2^{\frac{i}{2}}))$$

segments per edge.

We note that the number of segments per edge which are produced by the algorithm in [CSY] is also an upper bound for M_i.

The visibility relation between segments can be computed separately for each of the $O(r)$ refinement steps by a sweep algorithm, as described in [CSY]. The cost of this algorithm dominates the computation of G_i^{ess}. Thus, the running time of the i-th step is $T(n, M_i, W)$, with T as in equation (4). The running time of the total algorithm can be bounded by

$$O(r \cdot T(n, M_r, W)).$$

Thus, we have proven theorem 2.

4.3 Path Optimization

With the incremental approach above, we have a tool to determine S_{min} in (3) in time polynomial in $1/\delta$. As soon as there is only one possible edge sequence (or only a few combinatorially distinct sequences) left, it is however more efficient to use an optimization technique to approximate the actual shortest path. We propose to use a steepest descend method (see e.g. [Go, section C-5]):

Let the spectrum of H be bounded below by $\mu > 0$ and above by ρ (choose μ as the smallest, and ρ as the biggest eigenvalue of H). We can derive explicit values for these bounds (especially for ρ) as described in subsection 4.4.

Let $I = [\frac{1}{2\rho}, \frac{3}{2\rho}]$. Define the sequence

$$\lambda^{i+1} = \lambda^i - \kappa \nabla l(\lambda^i)$$

with $\kappa \in I$ and λ^0 the known approximation. Then this sequence converges to the unique minimizer ζ of l at the rate of a geometric progression with ratio $q = 1 - \mu/\rho$:

$$||\lambda^{i+1} - \zeta|| \leq q||\lambda^i - \zeta|| \leq q^{i+1}||\lambda^0 - \zeta||.$$

With

$$||l(\lambda^i) - l(\zeta)|| \leq \frac{\rho}{2}||\lambda^i - \zeta||^2$$

and

$$||\lambda^0 - \zeta||^2 \leq \frac{2}{\mu}(l(\lambda^0) - l(\zeta)) \leq \frac{2\delta}{\mu} l(\zeta),$$

we get

$$||l(\lambda^i) - l(\zeta)|| \leq \frac{\rho\delta}{\mu} q^{2i} |p_{min}|.$$

To achieve $||l(\lambda^i) - l(\zeta)|| < 2^{-r}|p_{min}|$, it is sufficient to choose $i > N$ with

$$N = \Theta(\frac{\rho}{\mu}(r + |\log \delta| + |\log \frac{\rho}{\mu}|)).$$

Again, it is important to note that this method – e.g., because of the freedom of choice for κ – easily extends to the bit framework. The running time of the whole algorithm can be resolved as

$$O(\log(1/\delta) \cdot T(n, M_\delta, W) + Nn\mu(W)),$$

where $M_\delta = O(nL/\delta)$. This finally proves theorem 3.

4.4 Spectral Bounds

In this subsection, we discuss two different methods to get bounds on the spectrum of the Hessian H of the path length function $l = l_{S_{min}}$. We shall use the notations of section 2.

By the theory of Gerschgorin circles, the eigenvalues of H are bounded above by $\rho = max_i\{ a_i + |b_i| + |b_{i-1}| \}$, with a_i and b_i as in the proof of lemma 2.

With the help of ρ, we can directly give a bound on μ. As the determinant of H is equal to the product of all k eigenvalues of H, we get

$$\mu \geq \frac{det(H)}{\rho^{k-1}},$$

where the determinant of H satisfies the inequality

$$det(H) \geq \prod_{i=1}^{k} a_i^- .$$

A crucial deficiency of the above bound is that it is exponential in k, the number of intermediate vertices of p_{min}.

A bound on μ not depending on k can be obtained by a method based on the theorem of Courant-Fischer (see [Wi], pp. 101-102):

Let A, B and C be positive-semi-definite symmetric matrices with $C = A + B$, and α (resp., β) the smallest eigenvalue of A (resp., B). Then each eigenvalue γ of C satisfies $\gamma \geq \alpha + \beta$. Assuming k to be even (the case of odd k can be similarly treated), we split $H = A + B$ according to

$$A = \sum_{i=0}^{k/2} H_{2i} \ , \quad B = \sum_{i=1}^{k/2} H_{2i-1}.$$

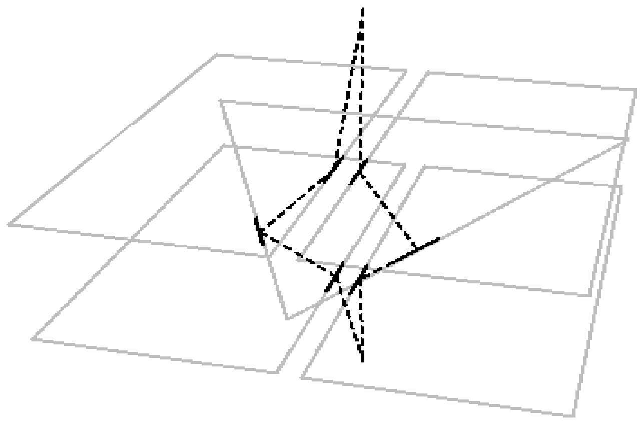
The matrices A and B are block matrices, and the eigenvalues are the eigenvalues of the H_i. It follows that

$$\mu \geq min\{ \mu_i \ ; \ i = 0, \ldots, k \},$$

where μ_i is the smallest eigenvalue of H_i. This bound has the nice property that it depends only on pairs of edges of S_{min}.

5 Experimental Results

The preceding algorithms for the approximate 3ESP problem have a high polynomial dependency on the number of edges n or the desired error bound ε. But these theoretical bounds need not reflect the "average behavior" or practical situations. This suggests some empirical studies.



Horizontal Obstacles 1

Steps	0	2	4	6	8	10
Error	1.6771	0.4193	0.1048	0.0262	0.0066	0.0016
Length	2.4852	2.3746	2.3290	2.3252	2.3250	2.3250
Segments	19	76	224	186	366	728

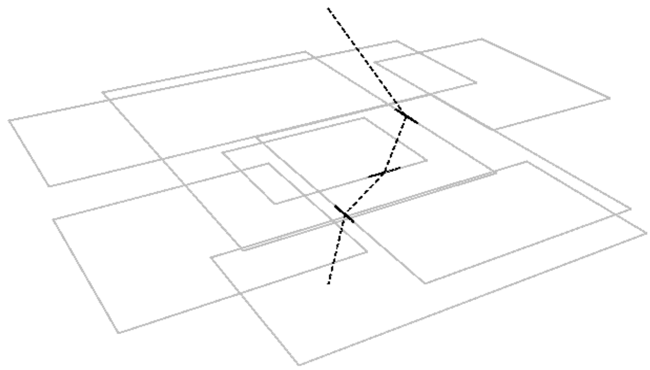
Table 1

To verify the practicability of our incremental approach, we implemented a simplified version of the proposed algorithm. The simplification is based on the observation that for certain special cases, the visibility relation between segments can be replaced by the visibility of segment midpoints:

Let the obstacles be 2-dimensional facets arranged in h parallel planes separating the start and target point. Let $\varepsilon' = \varepsilon/h$, and consider the subdivision defined in section 4. Then the following holds:

Lemma 12 *There exists a free polygonal path p from s to t, which connects segment midpoints and satisfies $|p| \leq (1+\varepsilon)|p_{min}|$.*

Proof: The shortest path p_{min} from s to t is strictly monotone in the direction of the normal vector of the planes containing the obstacle facets. Now pick an arbitrary vertex v of p_{min} and move this vertex on the incident edge in either direction while keeping the other path vertices fixed. We continue this deformation until the (deformed) path hits another obstacle facet or until v hits a segment midpoint. In the first case, we consider the intersection point as a



Horizontal Obstacles 2

Steps	0	2	4	6	8	10
Error	2.2500	0.5625	0.1406	0.0352	0.0088	0.0022
Length	2.4116	2.3263	2.3085	2.3070	2.3069	2.3069
Segments	28	112	290	334	368	430

Table 2

further path vertex, and in the latter case we continue the process by picking another path vertex until all vertices coincide with segment midpoints. It is easy to see that this process terminates after at most h deformation steps, introducing an absolute error of at most $h\varepsilon'|p_{min}|$. □

Further, we used a uniform subdivision for each edge, by starting with the edges as segments and successively halving the segments.

Tables 1 and 2 show the result of the incremental algorithm for the situation in the figures Horizontal Obstacles 1 and 2. The figures visualize the situation after 10 iteration steps: the essential segments are the solid black parts of the edges, and the shortest paths from start to goal determined so far are drawn dashed. In the first example there are two shortest paths, which are resolved after the 8th iteration step. In the second example, the unique shortest path is resolved after the 10th step.

The tables show the guaranteed relative error in path length, the length of the shortest path in the current visibility graph, and the number of the essential segments. The running time of the algorithm is mostly quadratic in this number, and was – for these examples – in the range of seconds on a state-of-the-art workstation.

The following behavior has been typical for the examples we tried: until the error bound ε_i is small enough to discard, the number of essential segments is doubled per step. Then comes a phase where the segment number does not change significantly. Once the shortest paths are resolved, the number of essential segments is doubled every 2 iteration steps, as predicted by the theoretical results.

6 Final Remarks

We have developed the first precision-sensitive algorithms for 3ESP. Beyond its intrinsic interest, it demonstrates a critical exploitation of precision-sensitivity. We conjecture that other previously intractable problems may likewise yield to this approach.

If the sensitivity parameter δ is zero, we can modify our approach to take advantage of the "next sensitivity" parameter, namely the gap between the second and the third shortest path, etc. A general treatment of this may be interesting.

We note that attention has to be paid to degenerate situations in this problem. But it seems unavoidable to take this into account because degeneracy seems to be one cause of intrinsic complexity in 3ESP. Note that there has been some recent literature on degeneracy in geometric problems.

The merits of the incremental 3ESP seem evident: in our examples, there would have been no chance to detect the shortest path by the exhaustive approach [CSY]. Our algorithm is a useful tool when a researcher needs to determine the real shortest path in a particular small environment.

References

[Berg] M. de Berg, "Efficient algorithms for ray shooting and hidden surface removal", *PhD Dissertation*, Department of Computer Science, Utrecht University, the Netherlands, 1992.

[Bern] M. Bern, "Hidden surface removal for rectangles", *J. Computer and System Sciences*, Vol. 40, 1990, pp. 49-59.

[Bl] J. Blömer, "Computing Sums of Radicals in Polynomial Time", *Proc. IEEE Symposium on Foundations of Computer Science*, 1991, pp. 670-677.

[Bl2] J. Blömer, "Simplifying expressions involving radicals", *PhD Dissertation*, Department of Mathematics, Free University Berlin, Germany, 1992.

[CR] J.F. Canny, J. Reif, "New Lower Bound Techniques for Robot Motion Planning Problems", *Proc. 28th IEEE Symposium on Foundations of Computer Science*, Los Angeles 1987, pp. 49-60.

[CSY] J. Choi, J. Sellen, C.-K. Yap, "Approximate Euclidean Shortest Path in 3-Space", *Proc. 10th ACM Symposium on Computational Geometry*, 1994, pp. 41-48.

[Go] A.A. Goldstein, *Constructive Real Analysis*, Harper & Row Publishers, New York, 1967.

[HS] J. Hershberger, S. Suri, "Efficient Computation of Euclidean Shortest Paths in the Plane", *Proc. 34th IEEE Symposium on Foundations of Computer Science*, 1993, pp. 508-517.

[Re] J. Renegar, "On the Computational Complexity and Geometry of the First-Order Theory of the Reals", Technical Report, School of Operations Research Report, Cornell University, 1989.

[Pa1] C.H. Papadimitriou, "An Algorithm for Shortest-Path Motion in Three Dimensions", *Information Processing Letters*, Vol. 20, 1985, pp. 259-263.

[Pa2] C.H. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, Massachusetts, 1994.

[Ro] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.

[SS] M. Sharir, A. Schorr, "On Shortest Paths in Polyhedral Spaces", *SIAM Journal on Computing*, Vol. 15 No. 1, 1986, pp. 193-215.

[VV] L.G. Valiant, V.V. Vazirani, "NP is as Easy as Detecting Unique Solutions", *Theoretical Computer Science*, 47, 1986, pp. 85-93.

[Wi] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press Oxford, 1965.

[Ya] C.-K. Yap, "Towards Exact Geometric Computation", *Computational Geometry: Theory and Applications*, Vol. 7, 1997, pp. 3-23 (Invited talk, Proceed. 5th Canadian Conference on Computational Geometry, 1993, pp. 405-419).