Approximate shortest paths in moderately anisotropic regions

Siu–Wing Cheng [*] Hyeon–Suk Na [†] Antoine Vigneron [‡]

September 22, 2005

Abstract

We want to find an approximate shortest path for a point robot moving in a planar subdivision. Each face of the subdivision is associated with a convex distance function that has the following property: its unit disk contains a unit Euclidean disk, and is contained in a Euclidean disk with radius ρ. Obstacles are allowed, so there can be regions that the robot is not allowed to enter. We give an algorithm that, given any two points s and t, finds an approximate shortest path between s and t whose length is at most $(1+\varepsilon)$ times the length of the shortest path. When n is the number of vertices in the input subdivision, the running time of our algorithm is $O((\rho n^3/\varepsilon^2)\log(\rho)\log(n\rho/\varepsilon))$. This bound does not depend on any other parameters, in particular it does not depend on the minimum angle in the subdivision. As special cases, we can solve the following two problems within the same time bound:

- the weighted region problem where all weights are in $[1, \rho] \cup \{+\infty\}$,
- shortest paths in the flow field when the speed of the robot is 1 and the speed of the flow is at most $(\rho - 1)/(\rho + 1)$.

1 Introduction

Previous work on related shortest paths problems [1, 2, 3, 6, 7, 8, 9, 10, 11, 12]. Convex distance functions [4].

2 Convex distance functions

Let $\mathcal{B}$ denote a convex subset of $\mathbb{R}^2$ containing the origin in its interior. The *convex distance function* $\mathrm{d}_\mathcal{B}$ associated with $\mathcal{B}$ is defined as follows:

$$\forall a, b \in \mathbb{R}^2,\ \ \mathrm{d}_\mathcal{B}(a, b) = \inf\{\lambda \mid \lambda \geqslant 0 \text{ and } b \in a + \lambda\mathcal{B}\}.$$

We say that $\mathcal{B}$ is the *unit ball* of $\mathrm{d}_\mathcal{B}$. Note that in general, it is not a metric because $\mathrm{d}_\mathcal{B}(a, b)$ is not necessarily equal to $\mathrm{d}_\mathcal{B}(b, a)$. However, such a function satisfies the triangle inequality:

$$\forall a, b, c \in \mathbb{R}^2,\ \ \mathrm{d}_\mathcal{B}(a, b) + \mathrm{d}_\mathcal{B}(b, c) \leqslant \mathrm{d}_\mathcal{B}(a, c).$$

[*]Department of Computer Science, HKUST, Kong Kong, China. Email: `scheng@cs.ust.hk`

[†]School of Computing, Soongsil University, Seoul, Korea. Email: `hsnaa@computing.ssu.ac.kr`

[‡]Department of Applied Mathematics and Computer Science, INRA Jouy–en–Josas, France. Email: `antoine.vigneron@polytechnique.org`

When $\rho \geqslant 1$, we say that the convex distance function $\mathrm{d}_{\mathcal{B}}$ is a ρ–convex distance function if $\mathcal{B}$ contains the unit Euclidean disk centered at the origin and if $\mathcal{B}$ is contained in the Euclidean disk with radius ρ centered at the origin.

A convex set $\mathcal{B}$ is *strictly* convex if, for any two points a and b on its boundary, the interior of the line segment $\overline{ab}$ is contained in the interior of $\mathcal{B}$. We will say that $\mathrm{d}_{\mathcal{B}}$ is a *ρ–strictly convex distance function* if it is ρ–convex and if $\mathcal{B}$ is strictly convex. In particular, it means that, using the distance function $\mathrm{d}_{\mathcal{B}}$, and for any two points a and b, the line segment $\overline{ab}$ is the (unique) shortest path between a and b.

We denote by $\mathrm{d}(a, b)$ the Euclidean distance between a and b. For any $\rho \geqslant 1$, this distance is a ρ–strictly convex distance function.

3 Shortest paths in an anisotropic triangulation

Let $\rho \geqslant 1$ denote a real number. Let $\mathcal{T}$ be a triangulation in $\mathbb{R}^2$ with n faces, and possibly with holes. Each face f of $\mathcal{T}$ is associated with a ρ–strictly convex distance function d_f. A point robot moves within $\mathcal{T}$, and the time needed by the robot to move from a point $a \in f$ to a point $b \in f$ (without leaving f) is $\mathrm{d}_f(a, b)$. We are interested in finding approximate shortest paths for this robot when it moves from a point s that lies on an edge of $\mathcal{T}$ to another point t that lies on an edge of $\mathcal{T}$.

We do not deal explicitly with the case where an edge is associated with a convex distance function. This cases can be handled by conceptually replacing such an edge by a triangle with a zero–length edge. Similarly, we do not allow edges to be obstacles, we can replace such an edge by a degenerate triangular hole. When the robot moves along an edge e bounding faces f and f', we will consider that its speed is given by d_f or d'_f, whichever is most favorable in his direction of travel. This is to modelize the fact that the robot can choose to travel arbitrarily close to e, either in the interior of f or the interior of f'.

We are interested in optimal paths when the robot moves from s to t.

Lemma 1 *Any optimal path is a polyline, and each vertex of this path lies on an edge of $\mathcal{T}$.*

Proof: As we noted above, the shortest path within a face of $\mathcal{T}$ is a straight line segment, which proves the second part of the statement.

ANTOINE COMPLAINS: *Actually I don't have a proof for the first part (the shortest path is a polyline). There is a proof for the (anisotropic) weighted case in the article by Mitchell and Papadimitriou [8], Lemma 3.1, but it's not convincing. Take for instance the boat–sail problem where winds are spiraling around t and the boat has speed 0. Then the unique path, and thus shortest, is a spiral with an infinite number of edges, and the boat–sail distance is still finite. (Of course this example does not apply to our situation because the unit balls do not contain the origin in their interior, but it shows that the problem is non–trivial). Still I think it works in our case, but it may require more than arguments from discrete geometry. (functional analysis, control theory?)*

ANTOINE COMPLAINS: *If we cannot solve it we may forget about the boat–sail case, which would be a pity, or we may be able prove an approximate version of Lemma 2 that does not require Lemma 1: there is an ε–approximate shortest path with length polynomial in n, ρ and $1/\varepsilon$.* □

Assume that $\mathrm{p}_{s \to t}$ is a polyline starting at s and ending at t, with all its vertices on edges of $\mathcal{T}$. We will denote the sequence of vertices of $\mathrm{p}_{s \to t}$ by $(p_1, p_2, \ldots, p_{k+1})$, where $s = p_1$,

$t = p_{k+1}$ and each p_i lies on an edge of $\mathcal{T}$. So $\mathrm{p}_{s\to t}$ is the concatenation of the line segments $p_i p_{i+1}$. We denote by $\mathrm{w}(\mathrm{p}_{s\to t})$ the weighted length of $\mathrm{p}_{s\to t}$. In other words, if we denote by d_i the convex distance function associated with the face of $\mathcal{T}$ that contains $p_i p_{i+1}$, then we have

$$\mathrm{w}(\mathrm{p}_{s\to t}) = \mathrm{w}(p_1, p_2, \ldots, p_{k+1}) = \sum_{i=1}^{k} \mathrm{d}_i(p_i, p_{i+1}).$$

We are interested in the number of edges that such a path $\mathrm{p}_{s\to t}$ with smallest weighted length $\mathrm{w}(\mathrm{p}_{s\to t})$ can have.

Lemma 2 *There exists a shortest path* $\mathrm{p}_{s\to t}$ *with at most* $18n^2$ *edges.*

Proof: Remember that $\mathcal{T}$ is a triangulation with n faces, therefore it has at most $2n$ edges and $3n$ vertices. A shortest path cannot cross the same point twice, so it contains at most $3n$ vertices of $\mathcal{T}$. We consider a shortest path $\mathrm{p}_{s\to t}$ that contains a maximum number of vertices of $\mathcal{T}$.

Let $p_{ij} = (p_i, p_{i+1}, \ldots, p_j)$ be a subpath of $\mathrm{p}_{s\to t}$ such that no point in $\{p_{i+1}, p_{i+2}, \ldots, p_{j-1}\}$ is a vertex of $\mathcal{T}$. We will show that, for any edge e of $\mathcal{T}$, at most two points in $\{p_{i+1}, p_{i+2}, \ldots, p_{j-1}\}$ lie on e. Then the lemma follows from the fact that $\mathcal{T}$ has at most $2n$ edges and $3n$ vertices.

So let us assume, for a contradiction, that three vertices of p_{ij} lie inside e. The subpath p_{ij} intersects e along line segments or isolated points. So there must be two vertices p_a and p_b of p_{ij} contained in e, with $a < b$, and such that the open line segment ab does not intersect p_{ij}. Now we distinguish between several cases:

(i) Suppose that p_a and p_b are two isolated points along $p_{ij} \cap e$. For any $\delta \in \mathbb{R}$, we consider the point $p_a(\delta)$ that lies on the line ab, and such that the signed distance $p_a p_a(\delta)$ is δ (we orient the line ab such that $p_a p_b > 0$). Let e' denote the edge of $\mathcal{T}$ containing p_{a+1} and let ℓ' be the support line of e'. We define the point $p_{a+1}(\delta)$ as the point in ℓ' such that $p_a p_{a+1}$ is parallel to $p_a(\delta) p_{a+1}(\delta)$. By repeating this process, we construct a path $p_{ab}(\delta) = (p_a, p_a(\delta), p_{a+1}(\delta), \ldots, p_b(\delta), p_b)$ such that, for all $k \in [a, b]$, the edge $p_k(\delta) p_{k+1}(\delta)$ is parallel to $p_k p_{k+1}$ and the point $p_k(\delta)$ belongs to the support line of the edge of $\mathcal{T}$ that contains p_k. Let $[\delta_-, \delta_+]$ be the maximal interval containing 0 such that, for all $\delta \in (\delta_-, \delta_+)$, and for all $k \in [a, b]$, the point $p_k(\delta)$ is not a vertex of $\mathcal{T}$. It is easy to see that $\mathrm{w}(p_{ab}(\delta))$ is an affine function of δ. Therefore $\mathrm{w}(p_{ab}(\delta))$ achieves its minimum over $[\delta_-, \delta_+]$ at δ_- or δ_+. Then the path obtained from $\mathrm{p}_{s\to t}$ by replacing p_{ab} by $p_{ab}(\delta_-)$ or $p_{ab}(\delta_+)$ has weight at most the weight of $\mathrm{p}_{s\to t}$, and contains a larger number of vertices of $\mathcal{T}$, a contradiction.

(ii) (sketch) Suppose that the line segment $p_b p_{b+1}$ is contained in e. It's easy to see that $p_{b-1} p_b$ and $p_{b+1} p_{b+2}$ are contained in the same face. Applying the same transformation at in case (i), we have to consider the case where the length of $p_b p_{b+1}$ shrinks to 0. It also yields a contradiction, because then $p_{b-1}(\delta) p_{b+1} p_{b+2}$ is a path with a corner that lies entirely within a face of $\mathcal{T}$, so it cannot be a subpath of an optimal path.

(iii) (sketch) Suppose that the line segments $p_{a-1} p_a$ and $p_b p_{b+1}$ are contained in e. Replace p_{ab} by $(p_{a-1}, p_a(\delta), p_{a+1}(\delta), \ldots, p_b(\delta), p_{b+1})$ and reach a contradiction.

□

4 Algorithm

4.1 Algorithm

ANTOINE COMPLAINS: *This is just a sketch. We need to introduce the geodesic path.* Compute the (non–weighted) geodesic distance $\mathrm{d_g}(s,t)$. Consider the squares S_i centered at the midpoint of st with edge length $\rho\,\mathrm{d_g}(s,t)/2^i$ for $i \in \{0, 1, \dots, \lceil \log(\rho) \rceil\}$. For all i and for all face f of $\mathcal{T}$, place Steiner points equally spaced along the boundary of $f \cap S_i$, with spacing $\varepsilon\,\mathrm{d_g}(s,t)/(36n^2 2^i)$. For each face f construct a $(1+\varepsilon)$–spanner of the union of the vertices of f and the Steiner points on the boundary of f (use the Yao graph, see section 4.1 in Eppstein's survey [5]). Compute a shortest path $\mathrm{p}^{\varepsilon}_{s\to t}$ from s to t in the union G of these spanners (use for instance the algorithm *shortespathtree* in Tarjan's book [13]). Output $\mathrm{p}^{\varepsilon}_{s\to t}$.

4.2 Analysis

In total, we placed $O((\rho n^3/\varepsilon)\log\rho)$ Steiner points, so G has $O((\rho n^3/\varepsilon^2)\log\rho)$ edges. So our algorithm runs in time $O((\rho n^3/\varepsilon^2)\log(\rho)\log(\rho n/\varepsilon))$.

4.3 Correctness

Let $\mathrm{p}_{s\to t} = (p_1, p_2, \dots, p_{k+1})$ be a shortest path with $k \leqslant 18n^2$. First notice that its weight is at most the weight of the geodesic path from s to t, so the Euclidean length of $\mathrm{p}_{s\to t}$ is at most $\rho\,\mathrm{d_g}(s,t)$ and thus $\mathrm{p}_{s\to t}$ is contained in S_0. Let i_0 denote the largest index i such that $\mathrm{p}_{s\to t}$ is contained in S_i. Then we know that the Euclidean length of $\mathrm{p}_{s\to t}$ is at least $(\rho/2^{i_0})\,\mathrm{d_g}(s,t)$, and therefore

$$\mathrm{w}(\mathrm{p}_{s\to t}) \geqslant \frac{1}{2^{i_0}}\,\mathrm{d_g}(s,t) \tag{1}$$

For each p_i lying in the interior of an edge e of $\mathcal{T}$, let p'_i denote the Steiner point on e that is closest to p_i. Notice that

$$\mathrm{d}(p_i, p'_i) \leqslant \frac{\varepsilon}{36n^2 2^{i_0}}\,\mathrm{d_g}(s,t). \tag{2}$$

Let $\mathrm{p}'_{s\to t} = (p'_1, p'_2, \dots, p'_{k+1})$ (clearly $p_1 = p'_1 = s$ and $p_{k+1} = p'_{k+1} = t$). For all i we denote by d_i the convex distance function associated with the face containing p_i, p_{i+1} ,p'_i and p'_{i+1}. Then the triangle inequality gives us

$$\begin{aligned}
\mathrm{w}(\mathrm{p}'_{s\to t}) &= \sum_{i=1}^{k} \mathrm{d}_i(p'_i p'_{i+1}) \\
&\leqslant \left(\sum_{i=1}^{k} \mathrm{d}_i(p'_i p_i) + \mathrm{d}_i(p_{i+1} p'_{i+1})\right) + \sum_{i=1}^{k} \mathrm{d}_i(p_i p_{i+1}) \\
&\leqslant \left(\sum_{i=1}^{k} \mathrm{d}(p'_i p_i) + \mathrm{d}(p_{i+1} p'_{i+1})\right) + \mathrm{w}(\mathrm{p}_{s\to t}).
\end{aligned}$$

By Equation (2), it yields

$$\mathrm{w}(\mathrm{p}'_{s\to t}) \leqslant \left(\sum_{i=1}^{k} 2\frac{\varepsilon}{36n^2 2^{i_0}}\,\mathrm{d_g}(s,t)\right) + \mathrm{w}(\mathrm{p}_{s\to t}).$$

Remember that $k \leqslant 18n^2$, so

$$\mathrm{w}(\mathrm{p}'_{s\to t}) \leqslant \frac{\varepsilon}{2^{i_0}}\,\mathrm{d_g}(s,t) + \mathrm{w}(\mathrm{p}_{s\to t})$$

and thus by Equation (1) we obtain

$$\mathrm{w}(\mathrm{p}'_{s\to t}) \leqslant (1+\varepsilon)\,\mathrm{w}(\mathrm{p}_{s\to t}).$$

5 Applications

Extend from triangulation to arbitrary planar subdivision. Application to flow fields and weighted regions. Explain how to handle non strictly convex distance functions.

6 Conclusion

We can improve the running time of our algorithm by guessing i_0, and thus replace the $\log\rho$ factor by $\log\log\rho$ in our time bound. This improvement is rather small; the challenge is to remove completely the dependency in ρ at least in the isotropic case, which would give the first strongly polynomial time approximation scheme for the weighted region problem—Mitchell and Papadimitriou gave a polynomial time approximation scheme, but it depends on the bit complexity of the input.

References

[1] O. Aichholzer, F. Aurenhammer, D. Chen, D. Lee, and E. Papadopoulou. Skew Voronoi diagrams. *International Journal of Computational Geometry and Applications*, 9(3):235–247, 1999.

[2] L. Aleksandrov, A. Maheshwari, and J.-R. Sack. Approximation algorithms for geometric shortest path problems. In *ACM Symposium on Theory of Computing*, 2000.

[3] L. Aleksandrov, A. Maheshwari, and J.-R. Sack. Determining approximate shortest paths on weighted polyhedral surfaces. *Journal of the ACM*, 52, 2005.

[4] P. Chew and R. Dyrsdale. Voronoi diagrams based on convex distance functions. In *Proc. 1st Symposium on Computational geometry*, pages 235–244, 1985.

[5] D. Eppstein. Spanning trees and spanners. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 425–461. Elsevier Science Publishers, Amsterdam, 2000.

[6] M. Lanthier, A. Maheshwari, and J.-R. Sack. Shortest anisotropic paths on terrains. In *Annual International Colloquium on Automata, Languages and Programming*, 1999.

[7] C. Mata and J. Mitchell. A new algorithm for computing shortest paths in weighted planar subdivisions. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 264–273, 1997.

[8] J. Mitchell and C. Papadimitriou. The weighted region problem: Finding shortest paths through a weighted planar subdivision. *Journal of the ACM*, 38, 1991.

[9] T. Nishida and K. Sugihara. Voronoi diagram in the flow field. In *14th International Symposium on Algorithms and computation*, volume 2906 of *LNCS*, pages 26–35, 2003.

[10] T. Nishida and K. Sugihara. Approximation of the boat–sail voronoi diagram and its application. In *14th International Symposium on Algorithms and computation*, volume 3045 of *LNCS*, pages 227–236, 2004.

[11] J. Reif and Z. Sun. Movement planning in the presence of flows. *Algorithmica*, 39, 2004.

[12] Z. Sun and J. Reif. On finding approximate optimal paths in weighted regions. *Journal of Algorithms (to appear)*, 2005.

[13] R. E. Tarjan. *Data Structures and Network Algorithms*. SIAM, Philadelphia, 1983.