Computing Homotopic Shortest Paths Efficiently

Alon Efrat[1], Stephen G. Kobourov[1], and Anna Lubiw[2]

[1] Dept. of Computer Science, Univ. of Arizona, {alon, kobourov}@cs.arizona.edu
[2] School of Computer Science, Univ. of Waterloo, alubiw@uwaterloo.ca

Abstract

Geometric shortest paths are a major topic in computational geometry; see the survey paper by Mitchell [12]. A shortest path between two points in a simple polygon can be found in linear time using the "funnel" algorithm of Chazelle [3] and Lee and Preparata [10]. A more general problem is to find a shortest path between two points in a polygonal domain. In this case the "rubber band" solution is not unique, or, to put it another way, different paths may have different homotopy types. When the homotopy type of the solution is not specified, there are two main approaches, the visibility graph approach, and the continuous Dijkstra (or shortest path map) approach [12]. In this paper, we address the problem of finding a shortest path when the homotopy type is specified. Colloquially, we have a "sketch" of how the path should wind its way among the obstacles, and we want to pull the path tight to shorten it.

Homotopic shortest paths are used in VLSI routing [4, 8, 11]. A related problem is that of drawing graphs with "fat edges": given a planar weighted graph G, find a planar drawing such that all the edges are drawn as thickly as possible and proportional to the corresponding edge weights. Duncan *et al.* [5] and Efrat *et al.* [7] present an $O(kn + n^3)$ algorithm for this problem, where n is the number of edges and k is the maximum of their input and output complexities. Hershberger and Snoeyink [9] give an algorithm for the homotopic shortest path problem. Their algorithm assumes a triangulation of size n of the polygonal domain, and finds a shortest path homotopic to a given path of k edges in time linear in k plus the number of triangles (with repetition) visited by the input path. This can be nk in the worst case, and our aim is to reduce it when output size permits. Cabello *et al.* [2] consider the related problem of testing if two given paths are homotopically equivalent. Our original work [6] was done independently of theirs and the idea of the first step of the algorithms is the same. The current presentation of our work incorporates their more efficient implementation of this idea.

We now define homotopy, and give a precise description of our problem. Let $\alpha, \beta : [0,1] \longrightarrow \mathbb{R}^2$ be two continuous curves parameterized by arc-length. Then α and β are *homotopic* with respect to a set of obstacles $V \subseteq \mathbb{R}^2$ if α can be continuously deformed into β while avoiding the obstacles; more formally, if there exists a continuous function $h : [0,1] \times [0,1] \to \mathbb{R}^2$ such that:

1. $h(0,t) = \alpha(t)$ and $h(1,t) = \beta(t)$, for $0 \leq t \leq 1$

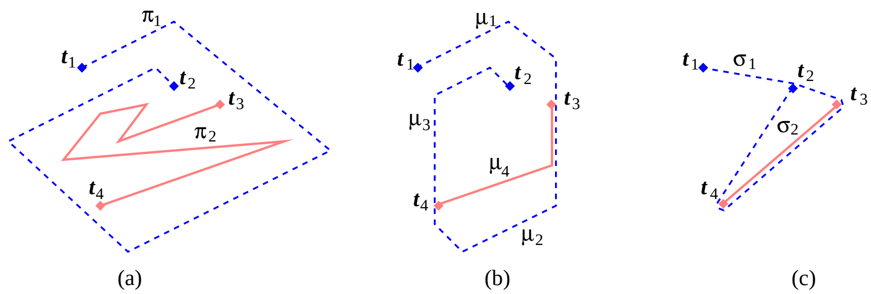

Fig. 1. (a) Paths π_1 and π_2 joining terminals t_1 to t_2 and t_3 to t_4, respectively. (b) After vertical shortcuts, π_1 consists of 3 monotone pieces: μ_1 from t_1 to t_3, μ_2 from t_3 to t_4, and μ_3 from t_4 to t_2; π_2 consists of one x-monotone piece, μ_4, homotopically equivalent to μ_2; (c) Final homotopic shortest paths, σ_1 and σ_2.

2. $h(\lambda, 0) = \alpha(0) = \beta(0)$ and $h(\lambda, 1) = \alpha(1) = \beta(1)$ for $0 \leq \lambda \leq 1$
3. $h(\lambda, t) \notin V$ for $0 \leq \lambda \leq 1, 0 < t < 1$

Let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$ be a set of disjoint, simple polygonal paths and let the endpoints of the paths in Π define the set T of at most $2n$ fixed points in the plane. Note that we allow a path to degenerate to a single point. We call the fixed points of T "terminals," and call the interior vertices of the paths "bends," and use "points" in a more generic sense, e.g. "a point on a path." We assume that no two terminals/bends lie on the same vertical line. Our goal is to replace each path $\pi_i \in \Pi$ by a shortest path σ_i that is homotopic to π_i with respect to the set of obstacles T; see Fig. 1. Note that σ_i is unique. Let $\Sigma = \{\sigma_1, \ldots \sigma_n\}$ be the set of resulting paths. Observe that these output paths may [self] intersect by way of segments lying on top of each other, but will be *non-crossing*.

Let k_{in} be the number of edges in all the paths of Π. Let k_{out} be the number of edges in all the paths of Σ. Note that k_{in} and k_{out} can be arbitrarily large compared to n, and that $k_{out} \leq nk_{in}$. The algorithm of Hershberger and Snoeyink [9] finds homotopic shortest paths in time $O(nk_{in})$. The deterministic algorithm presented in this paper runs in time $O(k_{out} + k_{in} \log n + n\sqrt{n})$, and the randomized algorithm in time $O(k_{out} + k_{in} \log n + n(\log n)^{1+\varepsilon})$. These are improvements except when k_{in} is quite small compared to n.

Our algorithm relies on the algorithm of Bar-Yehuda and Chazelle [1], which uses linear time polygon triangulation and ideas of linear time Jordan sorting, and finds a trapezoidization of n disjoint polygonal chains with a total of k edges in time $O(k + n(\log n)^{1+\varepsilon})$. Replacing this by plane sweep makes our algorithm implementable and yields a running time of $O(k_{out} + (n + k_{in}) \log(n + k_{in}))$ with an extra $O(n\sqrt{n})$ factor for the deterministic version. The algorithm of Cabello *et al.* [2] tests whether two paths are homotopically equivalent under the pin, pushpin, or tack models in $O((n + k_{in}) \log(n + k_{in}))$ time as they opt for the practical plane sweep approach.

References

1. R. Bar-Yehuda and B. Chazelle. Triangulating disjoint Jordan chains. *International Journal of Computational Geometry & Applications*, 4(4):475–481, 1994.
2. S. Cabello, Y. Liu, A. Mantler, and J. Snoeyink. Testing homotopy for paths in the plane. In *18th Annual Symposium on Computational Geometry*, pages 160–169, 2002.
3. B. Chazelle. A theorem on polygon cutting with applications. In *23rd Annual Symposium on Foundations of Computer Science*, pages 339–349, 1982.
4. R. Cole and A. Siegel. River routing every which way, but loose. In *25th Annual Symposium on Foundations of Computer Science*, pages 65–73, 1984.
5. C. A. Duncan, A. Efrat, S. G. Kobourov, and C. Wenk. Drawing with fat edges. In *9th Symposium on Graph Drawing (GD'01)*, pages 162–177, September 2001.
6. A. Efrat, S. Kobourov, and A. Lubiw. Computing homotopic shortest paths efficiently. Technical report, April 2002. `http://www.arXiv.org/abs/cs/0204050`.
7. A. Efrat, S. G. Kobourov, M. Stepp, and C. Wenk. Growing fat graphs. In *18th Annual Symposium on Computational Geometry*, pages 277–278, 2002.
8. S. Gao, M. Jerrum, M. Kaufmann, K. Mehlhorn, W. Rülling, and C. Storb. On continuous homotopic one layer routing. In *4th Annual Symposium on Computational Geometry*, pages 392–402, 1988.
9. Hershberger and Snoeyink. Computing minimum length paths of a given homotopy class. *CGTA: Computational Geometry: Theory and Applications*, 4:63–97, 1994.
10. D. T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14(3):393–410, 1984.
11. C. E. Leiserson and F. M. Maley. Algorithms for routing and testing routability of planar VLSI layouts. In *17th Annual ACM Symposium on Theory of Computing*, pages 69–78, 1985.
12. J. S. B. Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*. 1998.