Shortest Paths in Portalgons

Maarten Löffler
Department of Information and Computing Sciences, Utrecht University, The Netherlands
Department of Computer Science, Tulane University, New Orleans, LA, USA

Tim Ophelders
Department of Information and Computing Sciences, Utrecht University, The Netherlands
Department of Mathematics and Computer Science, TU Eindhoven, Tthe Netherlands

Rodrigo I. Silveira
Department de Matemàtiques, Universitat Politècnica de Catalunya, Barcelona, Spain

Frank Staals
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Abstract

Any surface that is intrinsically polyhedral can be represented by a collection of simple polygons (*fragments*), glued along pairs of equally long oriented edges, where each fragment is endowed with the geodesic metric arising from its Euclidean metric. We refer to such a representation as a *portalgon*, and we call two portalgons equivalent if the surfaces they represent are isometric.

We analyze the complexity of shortest paths. We call a fragment *happy* if any shortest path on the portalgon visits it at most a constant number of times. A portalgon is happy if all of its fragments are happy. We present an efficient algorithm to compute shortest paths on happy portalgons.

The number of times that a shortest path visits a fragment is unbounded in general. We contrast this by showing that the intrinsic Delaunay triangulation of any polyhedral surface corresponds to a happy portalgon. Since computing the intrinsic Delaunay triangulation may be inefficient, we provide an efficient algorithm to compute happy portalgons for a restricted class of portalgons.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Polyhedral surfaces, shortest paths, geodesic distance, Delaunay triangulation

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.48

Related Version *Full Version*: `https://arxiv.org/abs/2303.08937` [18]

Funding *Tim Ophelders*: Partially supported by the Dutch Research Council (NWO) under project no. VI.Veni.212.260.
Rodrigo I. Silveira: Partially funded by MICINN through project PID2019-104129GB-I00/ MCIN/ AEI/ 10.13039/501100011033.

Acknowledgements We are grateful to the anonymous reviewers for the detailed and valuable feedback provided, which helped us to improve the paper considerably.

1 Introduction

We define a *portalgon* $\mathcal{P}$ to be a collection of simple polygons (*fragments*) with some pairs of edges identified, see Figure 1. When we stitch together all fragments of a portalgon we obtain a two-dimensional surface Σ, whose *intrinsic metric* is *polyhedral* [7, 19]. Note that Σ is not necessarily embeddable in $\mathbb{R}^3$ with flat faces or without self-intersections, and not necessarily orientable. We say that $\mathcal{P}$ is a *representation* of Σ; crucially, the same surface may in principle have many different representations. Portalgons can be seen as a generalization of simple polygons, polygons with holes, polyhedral surfaces, and even developable surfaces.

We are interested in the *computational complexity* of computing *shortest paths* on portalgons. In particular, we analyze the *shortest path map* $\mathrm{SPM}(s)$; a representation of all shortest

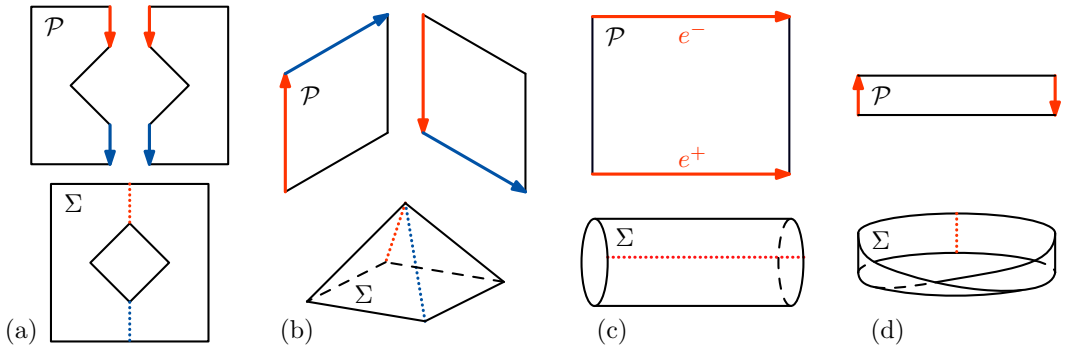

Figure 1 Examples of portalgons. Arrows of the same color represent portals that are identified with each other. (a) A portalgon that represents a polygon with a hole. (b) A portalgon that represents the surface of a bottomless pyramid. (c) A portalgon that represents the surface of a cylinder. (d) A portalgon that represents a Möbius strip, a non-orientable surface.

paths from a source point s to all other points in the portalgon. Our main insights are:

- The complexity of a shortest path on a surface Σ, represented by a given portalgon $\mathcal{P}$, may be *unbounded* in terms of the combinatorial properties of Σ and $\mathcal{P}$.
- The complexity of a shortest path depends on a parameter of the particular portalgon $\mathcal{P}$ representing the surface that we refer to as its *happiness* h. In particular, we show that the maximum complexity of a shortest path is $\Theta(n + hm)$, where n is the total number of vertices in the portalgon, $m \leq n$ is the number of portals.
- Given a source point in $\mathcal{P}$, the complexity of its shortest path map is $O(n^2 h)$. Moreover, if $\mathcal{P}$ is triangulated, it can be computed in $O(\lambda_4(k) \log^2 k)$ time, where k is the output complexity, and $\lambda_4(k)$ the length of an order-4 Davenport-Schinzel sequence on k symbols.
- Every surface with a polyhedral intrinsic metric admits a representation as a portalgon where the happiness h is constant. Specifically, one such representation is given by its intrinsic Delaunay triangulation. In such portalgons shortest paths have complexity $O(n)$.
- Since the intrinsic Delaunay triangulation is not easy to compute, we investigate the problem of transforming a given portalgon of happiness h into one with constant happiness. We present an algorithm to do so in $O(n + \log h)$ time, for a restricted class of portalgons. The question of how to compute such a good representation, in general, remains open.

1.1 Comparison to related work

Shortest paths have been studied in many different geometric settings, such as simple polygons, polygonal domains, terrains, surfaces, and polyhedra (see e.g. [5, 11, 13, 24]; refer to [21] for a comprehensive survey). The efficient *computation* of shortest paths is a fundamental

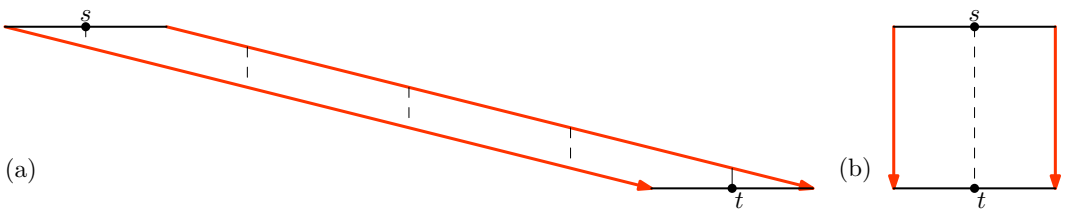

Figure 2 (a) A portalgon where a shortest path between s and t (dashed) has unbounded complexity. (b) A different representation of the same surface where the path has constant complexity.

problem in computational geometry [20, 23, 12, 5, 27, 26]. When the environment is a simple polygon the situation is well understood [12, 11]. For polygons with holes, efficient solutions have also been known for quite a while [13], recently culminating in an optimal $O(n + k \log k)$ time algorithm, where n is the total number of vertices of the polygon, and k the number of holes [26]. For more complex environments, like the surface of a convex polyhedron, several algorithms have been developed [5, 22, 28], and even implemented [14, 15]. However, for more general surfaces the situation is less well understood.

Portalgons generalize many of the geometric settings studied before. Hence, our goal is to unify existing shortest paths results. To the best of our knowledge, this has not been attempted before, even though several questions closely related to the ones addressed in this work were posed as open problems in a blog post almost two decades ago [9]. Instead, portalgons are a rather unexplored concept which, though it has been long adopted into popular culture [6, 8, 25], have only been studied from a computational point of view in the context of annular ray shooting by Erickson and Nayyeri [10].

When measuring the complexity of a shortest path, we can distinguish between its intrinsic complexity, and complexity caused by the representation of the underlying surface. For example, a shortest path π on a convex polyhedron Σ in $\mathbb{R}^3$ with n vertices, may cross (and thus bend) at $O(n)$ edges. Hence, a description of π on Σ has complexity $O(n)$. However, it is known that any convex polyhedron in $\mathbb{R}^3$ can be unfolded into a simple planar polygon P_Σ, and in such a way that a shortest path π corresponds to a line segment in P_Σ [1, 2, 5]. Hence, π actually has a constant complexity description in P_Σ. It is easy to see that some portalgons may have shortest paths of *unbounded* complexity; as illustrated in Figure 2(a). This unbounded complexity, however, is completely caused by the representation, and indeed there is another equivalent portalgon without this behavior (Figure 2(b)). We introduce a parameter that explicitly measures the potential for shortest paths to have high complexity, which we call *happiness* – refer to Section 2 for a formal definition.

In Section 3, we analyze the complexity of shortest paths in terms of the happiness. Our first main result is that, if we have a portalgon with n vertices and happiness h, then the complexity of its shortest path map from a given source point is $O(n^2 h)$. Moreover, we show that, for triangulated portalgons, it can be computed in an output-sensitive fashion: if k is the complexity of the shortest path map, then it can be computed in $O(\lambda_4(k) \log^2 k)$ time.

It is worth noting that our analysis of the shortest path map has similarities with techniques used to compute shortest paths on polyhedral surfaces, most notably [5, 22]. However, the fact that portalgons are more general implies important differences with previous methods. We need to handle surfaces of non-zero genus (e.g., see Figure 3), while

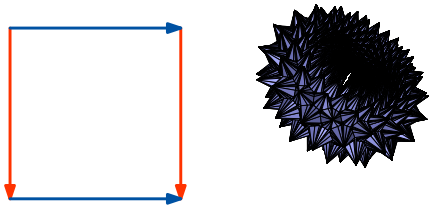

Figure 3 (a) A portalgon representing a flat torus. (b) Interestingly, a flat torus *can* be embedded isometrically in $\mathbb{R}^3$; however, the resulting embedding has very high complexity [4, 16] (image from `https://www.imaginary.org/es/node/2375`).

Chen and Han [5] require genus zero to compute the map in the interior of triangles (see the proof of their Theorem 4). Moreover, Σ may be non-embeddable in Euclidean space with flat triangles. Another difference is that shortest paths in portalgons can cross the same portal edge multiple times, something that is often (explicitly or implicitly) assumed to be impossible in algorithms for polyhedral surfaces (e.g., in [22]).

The fact that the complexity of the shortest path map can be upper bounded by a function of the happiness, leads to the following natural question: for a given surface Σ, can it always be represented by a portalgon $\mathcal{P}$ with *bounded happiness*? In Section 4, we prove that the answer to this question is "yes". In particular, our second main result is that for any portalgon, its *intrinsic Delaunay triangulation* [3] has constant happiness.

In turn, this then leads to another natural question: given a surface Σ, can we actually efficiently *compute* a portalgon representing it that has bounded, preferably constant, happiness? Clearly, the answer to this question depends on how Σ is represented. When Σ is given as a portalgon $\mathcal{P}$, possibly with unbounded happiness, this question then corresponds to the problem of transforming $\mathcal{P}$ into an equivalent portalgon with constant happiness. This is the problem we study in Section 5. Given the above result, the natural approach is to try to construct the intrinstic Delaunay triangulation of $\mathcal{P}$. Unfortunately, it is unknown if it is possible to do that efficiently (that is, with guarantees in terms of n and h). Our third main result shows that for a restricted class of portalgons we can give such guarantees. In particular, if the input portalgon has only one portal, n vertices, and happiness h, we can construct an equivalent portalgon that has constant happiness in $O(n + \log h)$ time.

2 Definitions and observations

Portalgons. We define a *portalgon* $\mathcal{P}$ to be a pair $(\mathcal{F}, P)$, where $\mathcal{F}$ is a collection of simple polygons, called *fragments*, and P is a collection of portals. A *portal* is an unordered pair $e = (e^-, e^+)$ of directed, distinct, equal length, edges from some fragment(s) of $\mathcal{F}$. We refer to e^- and e^+ as *portal edges*, see Figure 1(c), and require that each portal edge appears in one portal. If p^- is a point on e^-, then p^+ will denote the corresponding point on e^+.

Let n be the total number of vertices in the fragments of $\mathcal{P}$, and let m be the number of portal edges; the number of portals is $m/2$. Note that $m \leq n$. We denote the number of vertices and the number of portal edges in a fragment $F \in \mathcal{F}$ by n_F and m_F, respectively.

If we "glue" the edges of the fragments along their common portal edges, then the portalgon $\mathcal{P}$ describes a surface (2-manifold with boundary) $\Sigma = \Sigma(\mathcal{P})$, see Figure 1. Specifically, Σ is the space obtained from $\mathcal{P}$ by taking the collection $\mathcal{F}$ and identifying corresponding pairs of points on portal edges. We can write Σ as a quotient space $(\bigcup_{F\in\mathcal{F}} F)/\sim$, where $\sim$ is an equivalence relation that glues corresponding portal edges e^- and e^+.

Fragment Graph. A portalgon $\mathcal{P} = (\mathcal{F}, P)$ induces a (multi)graph that we refer to as the *fragment graph* G of $\mathcal{P}$ (see Figure 4). Each fragment $F \in \mathcal{F}$ is a node in G, and there is a link between F_1 and F_2 in G if and only if there is a portal e with e^- in F_1 and e^+ in F_2, or vice versa. Note there may be multiple portals connecting F_1 and F_2.

Paths and shortest paths. A path π from $s \in \Sigma$ to $t \in \Sigma$ is a continuous function mapping the interval $[0, 1]$ to Σ, where $s = \pi(0)$ and $t = \pi(1)$. For two points $p = \pi(a)$ and $q = \pi(b)$, we use $\pi[p, q]$ to denote the restriction of π to the interval $[a, b]$. The fragments of $\mathcal{P}$ split π into a set Π of maximal, non-empty, subpaths $\pi_1, .., \pi_z$, where (the image of) each π_i is contained in a single fragment. To be precise, for each fragment F, the intersection of (the

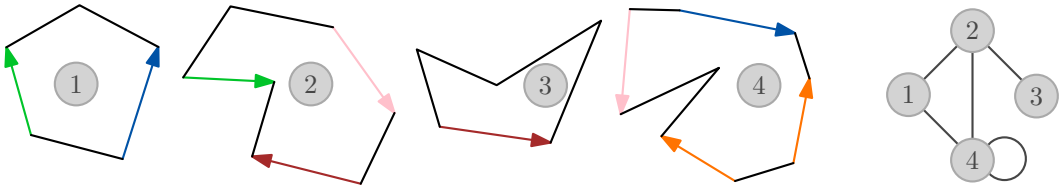

Figure 4 The fragment graph of a portalgon with four fragments and five portals.

image of) π with F is a set of maximal subpaths, and Π is the union of those sets over all fragments F[1]. We define the length of π as the sum of the lengths of its subpaths, and the distance $d(s,t)$ between $s \in \Sigma$ and $t \in \Sigma$ as the infimum of length over all paths between s and t. We inherit the property that $d(s,t) = 0$ if and only if $s = t$ from the metric in each fragment. It then follows that d is also a metric. Moreover, (Σ, d) is a geodesic space.

Observe that if π is a shortest path between s and t, each subpath π_i is a polygonal path whose vertices are either endpoints of π_i or vertices of the fragment containing π_i. Furthermore, when π crosses a portal the path does not bend (otherwise we can again locally shortcut it). It then follows that a shortest path π is also polygonal and can be uniquely described by an alternating sequence $s = v_1, E_1, v_2, .., E_k, v_{k+1} = t$ of vertices (s, t, or portalgon vertices) and sequences of portal edges crossed by the path. We refer to such a description as the *combinatorial representation* of the path, and to the total length of these sequences as the *complexity* of π. In the remainder of the paper, we will use $\pi(s,t)$ to denote an arbitrary minimum complexity shortest path between s and t. As we observed before (see Figure 2(b)), a shortest path may still intersect a single portal edge many times, and hence the complexity of a shortest path may be unbounded in terms of n and m.

Isometry. A map $f\colon X \to Y$ between metric spaces X and Y is an *isometry* if $d_X(x,x') = d_Y(f(x), f(x'))$ for all $x, x' \in X$. We say that f is a *local isometry* at a point x if there exists an open neighborhood U_x of x such that the restriction of f to U_x is an isometry.

Equivalent portalgons. Given a portalgon $\mathcal{P}$, there are many other portalgons that describe the same surface Σ. For instance, we can always cut a fragment into two smaller fragments by transforming a chord of the fragment into a portal, or, assuming this does not cause any overlap, we can glue two fragments along a portal, see Figure 5. More formally, two portalgons $\mathcal{P}$ and $\mathcal{Q}$ are *equivalent*, denoted $\mathcal{P} \equiv \mathcal{Q}$, if there is a bijective isometry between them (i.e., if for any pair of points $s, t \in \mathcal{P}$, their distance in $\mathcal{P}$ and $\mathcal{Q}$ is the same).

[1] Note that if π passes through a vertex of $\mathcal{P}$ that appears in multiple portals, Π contains subpaths for which the image consists of only a single point; the vertex itself. We later restrict our attention to minimum complexity paths, which allows us to get rid of such singleton paths.

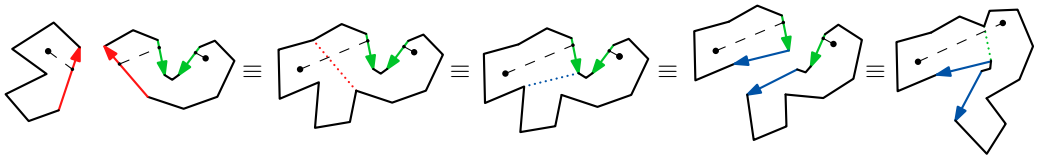

Figure 5 Five equivalent portalgons, with the shortest path between (the same) two points.

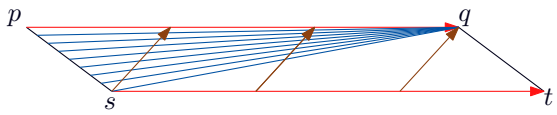

Figure 6 A simple h-happy portalgon in which a shortest path has complexity $\Omega(hm)$ as it crosses $\Omega(m)$ portal edges (the blue portal edges) $\Omega(h)$ times each.

Happiness. Our ultimate goal will be to find, for a given input portalgon, an equivalent portalgon such that all its shortest paths have bounded complexity. To this end, we introduce the notion of a *happy portalgon*, and more specifically, a *happy fragment* of a portalgon.

Let $\mathcal{P} = (\mathcal{F}, P)$ be a portalgon, let $F \in \mathcal{F}$ be a fragment in $\mathcal{F}$, and let $\Pi(p, q)$ denote the set of all shortest paths between $p, q \in \Sigma$. We define $c(X)$ as the number of connected components in X. The *happiness* $\mathcal{H}(F) = \max_{p,q\in\Sigma} \max_{\pi\in\Pi(p,q)} c(F \cap \pi)$ of fragment F is defined as the maximum number of times a shortest path π between any pair of points $p, q \in \Sigma$ can go through the fragment. The happiness of $\mathcal{P}$ is then defined as $\mathcal{H}(\mathcal{P}) = \max_{F\in\mathcal{F}} \mathcal{H}(F)$ the maximum happiness over all fragments. We call a portalgon h-happy when $\mathcal{H}(\mathcal{P}) \leq h$. Further, we sometimes refer to an $O(1)$-happy portalgon as *happy* (without an h value).

▶ **Lemma 1.** *Let $\mathcal{P} = (\mathcal{F}, P)$ be an h-happy portalgon, and let $\mathcal{P}' = (\mathcal{F}', P')$ be a triangulation of $\mathcal{P}$. The happiness of $\mathcal{P}'$ is at most h; that is, $\mathcal{P}'$ is an h-happy portalgon.*

Note that if $\mathcal{P}$ has n vertices and m portals, any triangulation $\mathcal{P}'$ of $\mathcal{P}$ consists of n vertices and $O(m+n)$ portals. The fact that in an h-happy portalgon a shortest path crosses every portal at most $h-1$ times implies the following.

▶ **Lemma 2.** *Let $\mathcal{P}$ be an h-happy portalgonand let s and t be two points in $\mathcal{P}$. A shortest path $\pi(s, t)$ between s and t has complexity $O(n + hm)$. This bound is tight in the worst case.*

Proof. The vertices of $\pi(s, t)$ are either: (i) s or t itself, (ii) vertices of $\mathcal{P}$, or (iii) points in which $\pi(s, t)$ crosses a portal edge. There are only two vertices of type (i) on $\pi(s, t)$. A shortest path can visit any vertex of $\mathcal{P}$ at most once; hence, there are at most n vertices of type (ii). Finally, since $\mathcal{P}$ is h-happy, $\pi(s, t)$ crosses every fragment $F \in \mathcal{F}$ at most h times. A shortest path of complexity $\Omega(n)$ is easy to attain in a fragment without portals and a chain with $\Omega(n)$ reflex vertices. We get the $\Omega(hm)$ term using a portalgon like in Figure 6. For any h, we can choose the length of the red portal so that the portalgon is h-happy. ◀

Shortest path map. Given a portalgon $\mathcal{P}$, a source point $s \in \Sigma$, and a region $\mathcal{X} \subseteq \Sigma$ the shortest path map $\mathrm{SPM}_{\mathcal{X}}(s)$ of s is a subdivision of $\mathcal{X}$ into maximally connected regions, such that for all points q in the interior of a region $R \in \mathrm{SPM}_{\mathcal{X}}(s)$ the shortest path from s to q is unique, and has the same combinatorial structure, i.e., visits the same sequence of vertices and portal edges of $\mathcal{P}$. Note that the complexity of $\mathrm{SPM}_{\mathcal{X}}(s)$ depends on the representation of the surface $\Sigma = \Sigma(\mathcal{P})$, that is, the portalgon $\mathcal{P}$. So changes to $\mathcal{P}$ may affect the complexity of $\mathrm{SPM}_{\mathcal{X}}(s)$. For example, splitting faces of $\mathcal{P}$ increases the complexity of $\mathrm{SPM}_{\mathcal{X}}(s)$. Hence, when Σ is fixed, an important problem is to find a good portalgon (i.e. one for which $\mathrm{SPM}(s) = \mathrm{SPM}_{\Sigma}(s)$ has low complexity) representing it.

Intrinsic Delaunay triangulation. A triangulation of a portalgon is an equivalent portalgon whose vertex set is the same, and all of whose fragments are triangles. In particular, among all such triangulations, the intrinsic Delaunay triangulation is such that for any interior edge of the triangulation, for the two triangles t and t' incident to that edge, the corners of t and t' not incident to that edge sum up to at most 180 degrees [3].

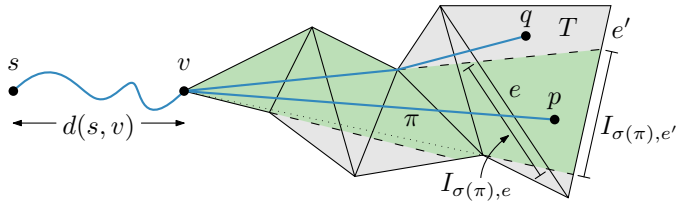

Figure 7 The length of the path π from s to point p in triangle T is $d(s,v) + \|p - v\|$. The signature of π defines intervals $I_{\sigma(\pi),e}$ and $I_{\sigma(\pi),e'}$ on edges e and e' of T, as well as a distance function $d_{\sigma(\pi)}$ illustrated by the path to point q in T.

3 Shortest paths in portalgons

In this section we sketch how to compute the shortest path map $\mathrm{SPM}_{\mathcal{P}} := \mathrm{SPM}_{\mathcal{P}}(s)$ of a vertex s of a triangulated h-happy portalgon $\mathcal{P}$ (a full description is in the full version [18]).

For a path π starting from the source s, define its signature, denoted $\sigma(\pi)$, to be the sequence of vertices and portals it passes through. Note that paths may simultaneously pass through a vertex and a portal (or multiple portals incident to that vertex). In this case, we break ties in the sequence by placing vertices before portals, and portals in the order that the path would pass through them if it were perturbed away from vertices (in a consistent way), where we always perturb the start of the path into a fixed triangle T_s incident to s.

If π is a shortest path from the source s to a point p in a triangle T, and v is the last vertex on $\sigma(\pi)$, then the length of π is $d(s,v) + d(v,p)$, where $d(v,p)$ can be expressed as the length of a line segment. We think of T as being embedded locally isometrically in the Euclidean plane, and by unfolding the fragments that π passes through after v, we can compute a copy of v in the plane, as well as copies of the portals and triangles that π passes through after v; see Figure 7. The locations of these copies in the plane depend only on $\sigma(\pi)$ (but not on p or π itself). Then, $d(v,p)$ is the Euclidean distance between the copies of v and p in the plane. Let e be an edge of T, and define $I_{\sigma(\pi),e}$ to be the interval of points p on e for which the segment $\overline{vp}$ passes through all the unfolded copies of portals of $\sigma(\pi)$ after v. For a point q in T, define

$$f_{\sigma(\pi)}(q) = \begin{cases} d(s,v) + \|\overline{vq}\| & \text{if the segment } \overline{vq} \text{ passes through } I_{\sigma(\pi),e}, \\ \infty & \text{otherwise} \end{cases}$$

Define $d_{\sigma(\pi)}(q)$ to be the infimum length over paths from s to q with signature $\sigma(\pi)$. This infimum is not necessarily realized by a path with the same signature, but is realized by a path that potentially bends around additional vertices, which are therefore inserted in its signature. We say that such a signature *reduces* to $\sigma(\pi)$. Note that if $f_{\sigma(\pi)}(q)$ is finite, then $f_{\sigma(\pi)}(q) = d_{\sigma(\pi)}(q)$. If π is a shortest path from s to p, then π has length $f_{\sigma(\pi)}(p)$. For a portal e of T, let $f_{\sigma(\pi)|e} \colon e \to \mathbb{R} \cup \{\infty\}$ be the restriction of $f_{\sigma(\pi)}$ to points on e.

3.1 A data structure for maintaining a lower envelope

Let F be a set of m partial functions, each pair of which can intersect at most twice. We describe a data structure storing the lower envelope env_F of F, that supports the operations:

NextLocalMinimum(δ): report the smallest local minimum of env_F that is larger than δ.
NextVertex(f, q): given a function $f \in F$ that realizes env_F at point q, find (if it exists) the lowest endpoint (v, δ') of the segment of env_F containing $(q, f(q))$ for which $\delta' > f(q)$.

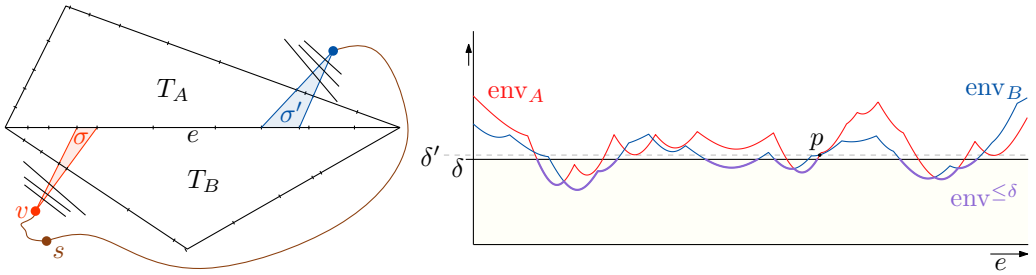

Figure 8 The information that we maintain while computing the SPM of the edges.

Insert(f): insert a new function into F.

▶ **Lemma 3.** *We can maintain* env_F *of a set* F *of partial functions, each pair of which intersects at most twice, in a data structure so that any sequence of* m **Insert** *operations and* k **NextLocalMinimum** *and* **NextVertex** *queries take* $O(k \log^2 m + \lambda_4(m) \log m)$ *time.*

3.2 Computing a shortest path map

We first compute the shortest path map of s, restricted to the edges of the portalgon. As in earlier work [22, 23] we propagate a wavefront of points at distance δ from s from 0 to ∞. However, it will be more convenient to view this as a collection of simultaneous sweepline algorithms. For each portal edge e, we sweep a horizontal line at height δ upward through the "position along e × distance from s" space (see Figure 8), while we construct the part of SPM_e below the sweep line. The main challenge is computing the next event –the first vertex in the lower envelope of the distance functions above the sweep line– in time.

For each portal e connecting two fragments T_A and T_B, we maintain the following:

1. Let $S_A(e,\delta)$ (resp. $S_B(e,\delta)$) be the set of signatures of shortest paths from s to points on the boundary of T_A (resp. T_B), where the last element of the signature is not e, and the length of the path is at most δ. We represent each signature $\sigma \in S_A(e,\delta) \cup S_B(e,\delta)$ implicitly by storing the interval $I_{\sigma,e}$, the position of (the unfolded copy of) the last vertex v on σ, and $d(s,v)$, so that we can compute $f_{\sigma|e}$ in constant time.
2. We store the lower envelope $\text{env}_A(e,\delta)$ (resp. $\text{env}_B(e,\delta)$) of the functions $f_{\sigma|e}$, where σ ranges over the signatures $S_A(e,\delta)$ (resp. $S_B(e,\delta)$), in the data structure of Lemma 3.
3. Let $\text{env}(e,\delta)$ be the lower envelope of the functions $\text{env}_A(e,\delta)$ and $\text{env}_B(e,\delta)$. We maintain only the part of $\text{env}(e,\delta)$ that lies below the sweep line, denoted $\text{env}^{\leq\delta}(e,\delta)$.
4. We maintain a binary search tree $\text{env}^{=\delta}(e,\delta)$ storing the intersection points of $\text{env}_A(e,\delta)$ and $\text{env}_B(e,\delta)$ with the sweep line, in order along the sweep line. For each intersection point we store the function(s) from $\text{env}_A(e,\delta)$ or $\text{env}_B(e,\delta)$ realizing this intersection.
5. Finally, we maintain a set of events pertaining to the edge e. We aggregate the events of all edges in a global priority queue, and use it to advance the sweep line algorithm to the next relevant value. The events that we store for an edge e are the values $\delta' > \delta$ such that
   a. δ' corresponds to a minimum of a function $f_{\sigma|e}$ with $\sigma \in S_A(e,\delta) \cup S_B(e,\delta)$,
   b. δ' corresponds to a vertex of $\text{env}_A(e,\delta)$ or $\text{env}_B(e,\delta)$, or
   c. δ' corresponds to an intersection between functions $f_{\sigma|e}$ on $\text{env}_A(e,\delta)$ and $f_{\sigma'|e}$ on $\text{env}_B(e,\delta)$, where $f_{\sigma|e}$ and $f_{\sigma'|e}$ are neighbors in $\text{env}^{=\delta}(e,\delta)$.

   For each event, we also keep track of the type and corresponding functions.

▶ **Lemma 4.** $\text{env}^{\leq\delta}(e,\delta)$ *encodes the shortest paths of length* $\leq \delta$ *to points on the edge* e.

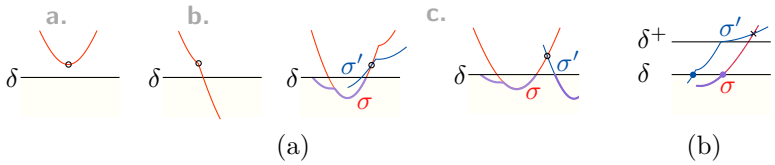

Figure 9 (a) The different event types: **a.** local minima in env_A or env_B, **b.** breakpoints of env_A or env_B, and two events of type **c.** first intersections of env_A and env_B. (b) We will detect the intersection between env_A and env_B once $f_{\sigma|e}$ and $f_{\sigma'|e}$ become neighbors in $\text{env}^{=\delta}$ at time δ^+.

Event handling. At an event (of any type), the order in which the functions of $\text{env}_A(e,\delta)$ and $\text{env}_B(e,\delta)$ intersect the sweep line changes. We therefore update $\text{env}^{=\delta}(e,\delta)$ by removing and inserting the appropriate functions associated with this event, and additionally make sure that we discover any additional events. To this end, we can use the **NextVertex** and **NextLocalMinimum** queries on the data structures storing $\text{env}_A(e,\delta)$ and $\text{env}_B(e,\delta)$.

As a result of the event, a new function, say $f_{\sigma'|e} \in \text{env}_B(e,\delta)$, may have appeared on $\text{env}^{\leq\delta}(e,\delta)$. We insert it into $\text{env}^{\leq\delta}(e,\delta)$, and propagate σ', extended by edge e, into the sets $S_B(e',\delta)$ of the other two edges e' incident to T_A. We therefore call **Insert** to insert a new function into $\text{env}_A(e',\delta)$, and **NextLocalMinimum**(δ) to update the next local minimum. [2]

Analysis. Let $k = |\,\text{SPM}_{\partial\mathcal{P}}\,|$ denote the complexity of the shortest path restricted to the edges. Let k_A and k_B be the number of signatures in $S_A(e,\infty)$ and $S_B(e,\infty)$, respectively. We argue that the total number of events on e is $O(\lambda_4(k_A)+\lambda_4(k_B))+|\,\text{SPM}_e\,|$ (essentially by charging them to $\text{env}_A(e,\infty)$ and $\text{env}_B(e,\infty)$). A signature σ appears in $S_B(e,\delta)$ only when there is a shortest path to another edge e' of T_B. Since every such edge e' propagates to the edges of the two triangles incident to e' it then follows that $\sum_{e\in\partial\mathcal{P}}(|S_A(e,\delta)|+|S_B(e,\delta)|) = O(|\,\text{SPM}_{\partial\mathcal{P}}\,|)$. Thus, the total number of events, over all edges e, is $O(\lambda_4(k))$. Since we can handle every event in $O(\log^2 k)$ time, we can then compute $\text{SPM}_{\partial\mathcal{P}}$ in $O(\lambda_4(k)\log^2 k)$ time.

▶ **Lemma 5.** *For an h-happy portalgon, the complexity of $\text{SPM}_{\partial\mathcal{P}}$ is $O(n^2h)$.*

Extension into the Interior. For each triangle T, we can now compute SPM_T from $\text{SPM}_{\partial T}$ reusing the algorithm from Mitchell et al. [22]. We therefore obtain the following result.

▶ **Theorem 6.** *Let $\mathcal{P}$ be a triangulated h-happy portalgon with n vertices and m portals, and let s be a given source point. The shortest path map $\text{SPM}_{\mathcal{P}}(s)$ of s has complexity $k = O(n^2h)$ and can be computed in $O(\lambda_4(k)\log^2 k)$ time.*

4 Existence of happy portalgons

In this section, we show that for every portalgon there exists an equivalent $O(1)$-happy portalgon; specifically, the *intrinsic Delaunay triangulation* [3] $\mathcal{T}$ of Σ induces a happy

[2] Mitchell, Mount, and Papadimitriou [22] use a similar overall algorithm. They define a notion of T_A*-free* paths that arrive at edge e from T_B. They prove that these paths act sufficiently like "real" shortest paths so that they can explicitly maintain the set of shortest T_A-free and T_B-free paths. Unfortunately, some of the arguments used hold only when a shortest path may cross a portal edge at most once (i.e., when $\mathcal{P}$ is 1-happy). In case of the weighted region problem, Mitchel and Papadimitriou show how to deal with this by extending this notion of T_A*-free* paths to *locally* T_A*-free* paths [23]. However, it is unclear how to bound the number of such paths when the genus may be non-zero.

portalgon, whose fragments correspond to the triangles of $\mathcal{T}$. The resulting portalgon may have more fragments than the original, but its total complexity is still linear.

4.1 Intrinsic Delaunay triangulation

Let $\mathcal{T}$ be a triangulated portalgon, and let Σ be its surface; let V be the set of vertices of Σ. Intuitively, the intrinsic Delaunay triangulation of Σ has a (straight) edge between two vertices $u, v \in V$ when there exists a circle with u and v on its boundary that contains no other vertices of V when we "unfold" Σ, for example by identifying edges of $\mathcal{T}$ as in Section 3.

To simplify geometric arguments, we derive a simply-connected space $\hat{\Sigma}$ from Σ. We can think of $\hat{\Sigma}$ as the universal cover of $\Sigma \setminus V$, with vertices reinserted at the corresponding locations. More formally, we can define $\hat{\Sigma}$ by considering the directed fragment graph G of $\mathcal{T}$. For any portal e of $\mathcal{T}$, we denote by $T_{e^-}, T_{e^+} \in \mathcal{F}$ the triangles (nodes of G) that contain the respective portal edge; G has a link $\dot{e}$ from T_{e^-} to T_{e^+} and a link $\dot{e}^{-1}$ from T_{e^+} to T_{e^-}. We say that $\dot{e}$ and $\dot{e}^{-1}$ are inverses of each other. A *walk* in G from a triangle T to a triangle T' is a (possibly empty, if $T = T'$) sequence of links of G, such that T is the source of the first link, and the source of the $(i+1)$-st is the target of the i-th link, and the target of the last link is T'. For a walk w from T to T', we write $w\colon T \to_G T'$ and say that a walk is *backtracking* if it contains two consecutive links that are inverses of each other. Fix an arbitrary root triangle T_0, and for a walk $w\colon T_0 \to_G T$, let T_w be a copy of the target triangle T placed in the Euclidean plane by unfolding $T_0, T_1, \ldots, T_w$ along their common portals. Let $\hat{\mathcal{F}} = \bigsqcup\{T_w \mid T \in \mathcal{F}, w\colon T_0 \to T, w \text{ is not backtracking}\}$ be the disjoint union of Euclidean triangles of target triangles of walks starting at T_0. We can now define $\hat{\Sigma}$ as $\hat{\mathcal{F}}/\hat{\sim}$, where for any two walks w and w', where w' is obtained from w by removing its final link, $\hat{\sim}$ glues $T_{w'}$ to T_w along the sides corresponding to the last link of w. Let $q\colon \hat{\Sigma} \to \Sigma$ be the map that sends points of $\hat{\Sigma}$ to their corresponding point in Σ. A map $\hat{g}\colon X \to \hat{\Sigma}$ is a *lift* of $g\colon X \to \Sigma$ if $g = q \circ \hat{g}$. Any path $\pi\colon [0,1] \to \Sigma$ has a lift $\hat{\pi}$ in $\hat{\Sigma}$.

There is a map $f\colon \hat{\Sigma} \to \mathbb{R}^2$ whose restriction to any triangle of $\hat{\Sigma}$ is an isometry, and whose restriction to any pair of adjacent triangles is injective. We can think of f as unfolding $\hat{\Sigma}$ so that it lies flat in the plane and is locally isometric everywhere except at the vertices (vertices are the only source of curvature), see Figure 10.

▶ **Observation 7.** *Let $B_r(x,y) = \{p \in \mathbb{R}^2 \mid \|p - (x,y)\| < r\}$ be the open disk in the plane of*

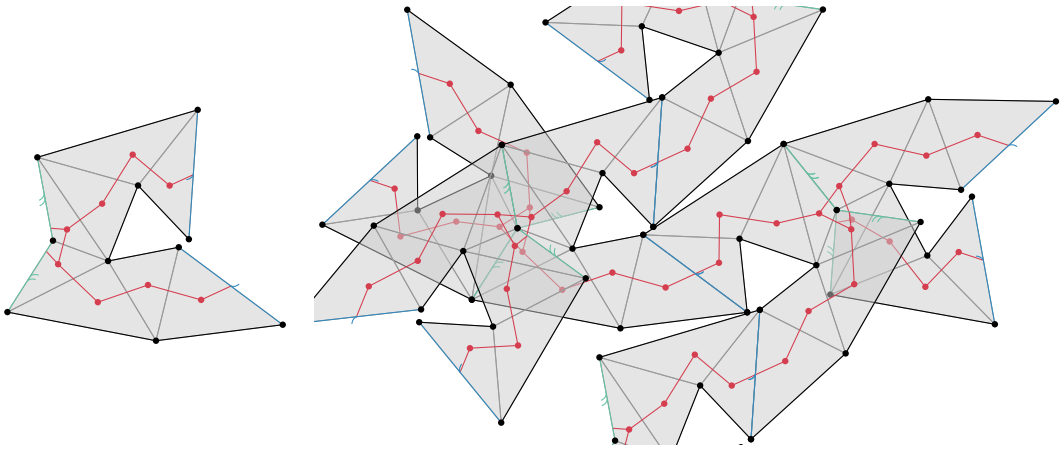

Figure 10 Σ (left) and a local region of $\hat{\Sigma}$ (right). The dual graph of $\hat{\Sigma}$ is an infinite tree (red).

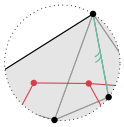

Figure 11 Although f is not injective in general, for a triangle T of a Delaunay triangulation, the restriction of f to $C(T)$ is injective and contains no vertices of the triangulation in the interior.

radius r centered at (x, y). If a component U of $f^{-1}(B_r(x, y))$ contains no vertices, then the restriction of f to the closure of U is an isometry.

For a triangle T of $\hat{\Sigma}$, let $D(T)$ be the open disk bounded by the circumcircle of $f(T)$, and let $C(T)$ be the closure of the component of $f^{-1}(D(T))$ that contains the interior of T. A triangle T of $\hat{\Sigma}$ is *Delaunay* if $C(T)$ does not contain any vertices of triangles adjacent to T in its interior. $\mathcal{T}$ is an *intrinsic Delaunay triangulation* of Σ if and only if all triangles of $\hat{\mathcal{F}}$ are Delaunay. Lemma 8 generalizes a well-known property of Delaunay triangulations in $\mathbb{R}^2$.

▶ **Lemma 8.** *[Bobenko et al. [3]] If $\hat{\mathcal{T}}$ is an intrinsic Delaunay triangulation of $\hat{\Sigma}$, then for any $T \in \hat{\mathcal{T}}$, $C(T)$ contains no vertices in its interior.*

Together with Observation 7 this implies the following corollaries, see also Figure 11.

▶ **Corollary 9.** *For any triangle T of an intrinsic Delaunay triangulation of $\hat{\Sigma}$, the restriction of f to $C(T)$ is injective, and shortest paths intersect $C(T)$ in straight segments.*

▶ **Corollary 10.** *For any triangle T of an intrinsic Delaunay triangulation of $\hat{\Sigma}$, $\partial f(C(T))$ is a union of chords and circular arcs of $\partial D(T)$, where each chord is a boundary edge.*

4.2 Intrinsic Delaunay triangulations are happy

We are now ready to prove the main result of this section: the intrinsic Delaunay triangulation of any portalgon has happiness less than 7.

Let the portalgon $\mathcal{T}$ be an intrinsic Delaunay triangulation of Σ. We want to bound the number of intersections between a shortest path π on Σ and portals (edges of the triangulation). For this, let π be a shortest path between two given points on Σ, and among all such paths, assume that π has a minimum number of crossings with portals of $\mathcal{T}$.

Now consider an arbitrary edge e between two vertices u and v in V and let m be its midpoint. We will show that π intersects e only constantly often. For a contradiction, suppose that π intersects e at least 7 times. To bound the total number of intersections with e we analyze the geometry of $\hat{\pi}$ in a local neighborhood of a lift $\hat{e}$ of e; in particular, we will argue that there cannot be many other copies of e in the neighbourhood of $\hat{e}$ that are all crossed by π. Since clearly there can also not be any copies of e far away that are crossed by π (since then it would be better to walk along e), the result will follow. Here, we give a sketch of the arguments, the full proof can be found in the full version [18].

Arbitrarily fix one of the triangles T incident to e and consider the neighborhood $C(\hat{T})$ of $\hat{e}$ for the corresponding lift $\hat{T}$ incident to $\hat{e}$. By Corollary 9, the restriction of f to $C(\hat{T})$ is an isometry, and $\hat{\pi}$ is locally a straight line segment. Let p_i be the i-th point of intersection of π with e, and define λ_i such that $\pi(\lambda_i) = p_i$. Let $\hat{e}_i$ be a lift of e containing $\hat{\pi}(\lambda_i)$, and let $\hat{p}_i$, $\hat{T}_i$, $\hat{u}_i$, $\hat{m}_i$, $\hat{v}_i$ be the respective lifts of p_i, T, u, m, v incident to $\hat{e}_i$, see Figure 12 (left). Let $D_i := D(\hat{T}_i)$, and let f_i be the restriction of f to $C_i := C(\hat{T}_i)$. Let c_i be the center of

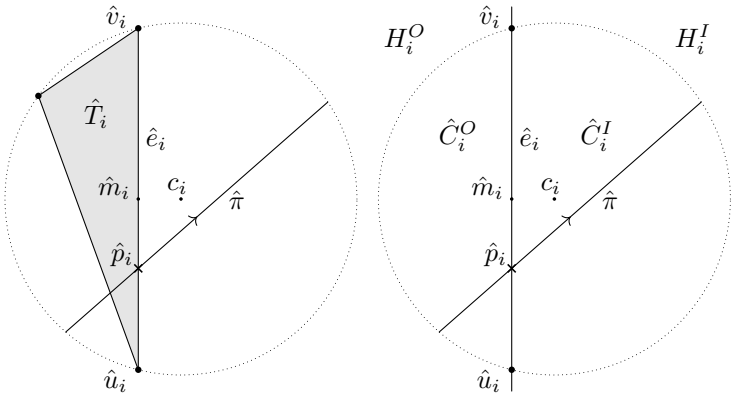

Figure 12 The crossing p_i of π with e is inward.

D_i, i.e., the circumcenter of $f(\hat{T}_i)$. Define H_i^I and H_i^O to be the two half-planes bounded by the line through $f(\hat{e}_i)$, such that H_i^I contains c_i (if $c_i = f(\hat{m}_i)$, label the half-planes by H_i^I and H_i^O arbitrarily). We respectively call H_i^I and H_i^O the *inner* and *outer* half-plane of $\hat{e}_i$, and define $C_i^I := f_i^{-1}(H_i^I)$ to be the *inner* component of $\hat{e}_i$, and $C_i^O := f_i^{-1}(H_i^O)$ to be the *outer* component of $\hat{e}_i$. We call the crossing p_i *inward* if p_i is a non-transversal crossing or $\hat{\pi}$ crosses $\hat{p}_i$ from C_i^O to C_i^I, see Figure 12 (right).

Let s be the segment of e from u to m. Assume without loss of generality that at least four of the (at least seven) crossings of π with e lie on s (otherwise relabel u and v).

▶ **Lemma 11.** *For any inward crossing p_i of π with s, none of the crossings p_j with $j > i$ lie on the segment of s between u and p_i.*

If p_i of π is not inward, then it is inward on the reverse of π. Corollary 12 follows.

▶ **Corollary 12.** *If some crossing p_i of π with s is not inward, then none of the crossings p_h with $h < i$ lie on the segment of s between u and p_i.*

Lemma 11 implies that for the sequence of intersections p_i of π with s, the distance function from u to p_i along s has only one local minimum. So there exists a subsequence of at least three crossings p_h, p_i and p_j $(h < i < j)$ of π with s such that p_i lies between p_h and p_j. Assume without loss of generality that the distances (along s) from u to p_h, p_i, and p_j are increasing (the other case follows by considering the reverse of π). We now observe:

▶ **Observation 13.** $f(\hat{m}_i) \neq c_i$

By Corollary 12, p_i is an inward crossing. We show how the possible locations of the subsequent crossings inside C_i are constrained. Define D_i' to be the disk concentric with D_i, whose boundary passes through $f(\hat{m}_i)$. Any chord of D_i that passes through D_i' is at least as long as e. Let $\overline{e}_h := f(\hat{e}_h) \cap D_i$. We observe:

▶ **Lemma 14.** *$\overline{e}_h$ cannot intersect the closure of D_i'.*

We will arrive at a contradiction to our initial assumption by showing that:

▶ **Lemma 15.** *If π crosses e at least 7 times then $\overline{e}_h$ must intersect the closure of D_i'.*

We conclude that π intersects e at most 6 times.We thus obtain:

▶ **Theorem 16.** *Let $\mathcal{P}$ be a portalgon with n vertices. There exists a portalgon $\mathcal{P}' \equiv \mathcal{P}$ with $O(n)$ vertices that is $O(1)$-happy.*

5 Making portalgons happy

While the approach in Section 4 is, in principle, constructive, its running time depends on the number of edge flips required to reconfigure an initial triangulation of $\mathcal{P}$ into the intrinsic Delaunay triangulation. This number does not (only) depend on the input complexity, except for the case where there is a bound on the minimum angle on the input triangles [17].

In this section, we analyze what can be done when there is no such bound. We study conditions for a portalgon $\mathcal{P}$ to be happy, and present a method to rearrange an unhappy portalgon into an equivalent one that is happy. In the full version [18], we show we can reduce any portalgon whose fragment graph has at most one cycle to a portalgon with just one fragment and one portal. Thus, here we only consider one portal, and give an efficient method to compute an equivalent happy portalgon. For brevity, we present the case where the portal edges are parallel. We also note that if the angle between the two edges is at least a constant, the fragment is already happy. Hence, the interesting case is that of nearly-parallel edges.

Single parallel portal analysis. The happiness of a fragment with a single portal depends on the *shift* of the two (parallel) portal edges. The shift of two portal edges e^-, e^+ is the distance between the two perpendiculars to the portal edges that go through the start vertices of e^- and e^+, respectively. In the following, we assume without loss of generality that the portals are horizontal, thus the shift Δ is simply the difference in x-coordinate between the start vertices of the portal edges. Let v denote the vertical distance between e^- and e^+.

▶ **Lemma 17.** *Let F be a fragment with exactly two portal edges e^-, e^+, which are parallel and belong to the same portal. If the shift Δ of e^-, e^+ is 0, then the fragment is 2-happy.*

If $\Delta \neq 0$, a fragment might be happy or not. Next we present a method to transform a fragment that is not happy into an equivalent portalgon that is happy. The idea is to create a new portal by cutting F through a line in the direction orthogonal to the line through the two start vertices of the portals. First we present the idea for the case where F is a parallelogram with two parallel portal edges e^- and e^+. Assume without loss of generality that e^- is above e^+, and that e^- starts to the left of e^+. In this case, the slope of the line in the direction orthogonal to the portal start points is $z = -\Delta/v$. We begin at the leftmost vertex of e^+, and shoot a ray with slope z in the interior of F until we hit the boundary. Every time the ray crosses the portal, we "cut" along this ray, creating a new portal along it. This creates several smaller fragments, which we glue together along the pieces of the original portals, into one fragment. The resulting fragment is a rectangle F'. See Figure 13 for an illustration. Note that, by definition of z, this new fragment F' now has shift zero. Hence:

▶ **Lemma 18.** *For any parallelogram with two parallel portal edges, there is an equivalent parallelogram that has $\Delta = 0$ and therefore is 2-happy.*

For a fragment with two parallel portals and non-zero shift, there is a unique equivalent fragment with zero shift, which is the one obtained by cutting along the perpendicular ray. However, if F is not a parallelogram, it may occur that when gluing together the new smaller fragments we obtain a non-simple polygon. Therefore, we may not be able to transform F into a single equivalent happy fragment, as shown in Figure 14. In that case, cutting along the perpendicular ray produces a non-simple fragment.

Fortunately, we can transform any fragment with two parallel portal edges into a small constant number of happy fragments.

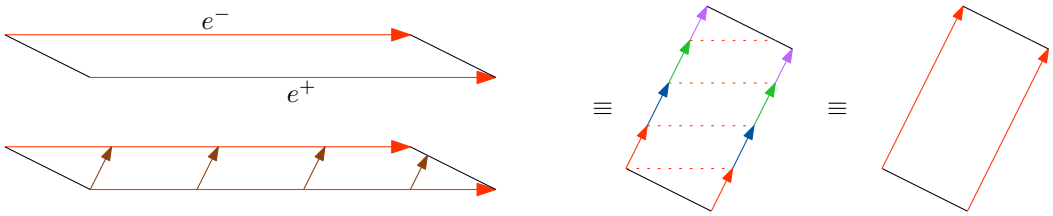

Figure 13 Left: a fragment with two parallel edges and non-zero shift. Center: result of cutting the fragment along a ray orthogonal to the line through the two start vertices of the portals, resulting in a new fragment with several portals. The latter is equivalent to a fragment with one portal and zero shift (right).

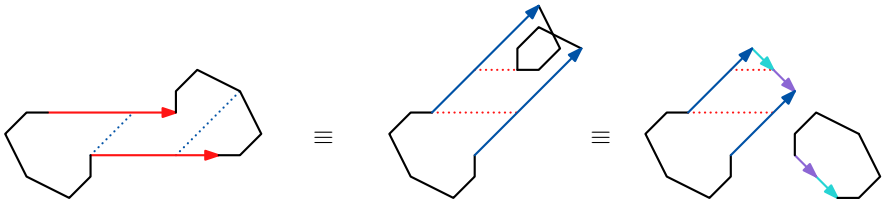

Figure 14 Example of a fragment (left) where the technique of cutting along the perpendicular ray produces a non-simple polygon (center). This can be fixed by using one more fragment (right).

▶ **Lemma 19.** *Let $\mathcal{P}$ be a portalgon with one fragment F with n vertices, and one portal whose edges are parallel. There exists a 5-happy portalgon $\mathcal{P}'$ equivalent to $\mathcal{P}$ consisting of at most three fragments and total complexity $O(n)$.*

Proof sketch. Assume w.l.o.g. that e^- and e^+ are horizontal and oriented left-to-right. We argue that when no three vertices of F are colinear, and F is not already 5-happy, we can split F into at most seven 4-happy fragments of complexity $O(n)$. See Figure 15.

Let m be the line through the start points of e^- and e^+. If there is no translate of m whose intersection with F contains a segment connecting e^- to e^+, F is already 5-happy. Let m_ℓ be the leftmost such translate of m and m_r the rightmost such translate; m_ℓ contains a vertex ℓ of F and m_r contains a vertex r of F (possibly, ℓ or m is an endpoint of e^- or e^+). Let a^- and a^+ be the intersection points of m_ℓ with e^- and e^+, and let b^- and b^+ be the intersection points of m_r with e^- and e^+. We cut the parallelogram $Z = a^-b^-a^+b^+$ from F, which splits F into at most seven fragments.

We now transform Z into a 2-happy fragment using Lemma 18. Let T and B be the fragments containing the starting points of e^- and e^+, respectively. We argue that T is 4-happy (other arguments are symmetric): consider the maximal connected components of a

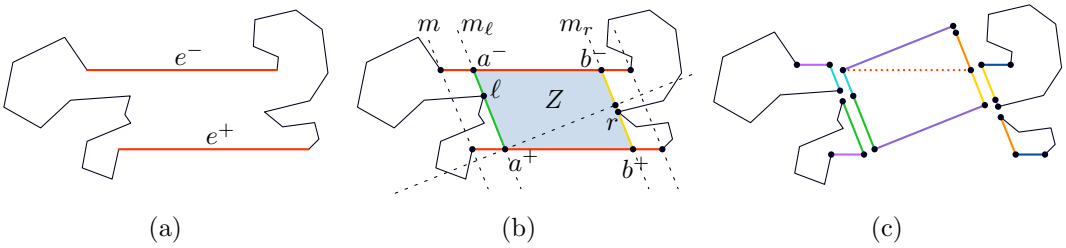

Figure 15 (a) Fragment with two parallel portals e^- and e^+. (b) Lines m_ℓ and m_r define a parallelogram Z, which splits F in at most seven fragments. (c) The resulting 5-happy fragments.

shortest path $\pi = \pi(s,t)$ with F. Such a component either: (i) contains s, (ii) contains t, or (iii) connects a point p_i on e^- to a point q_i on e^+. Each such component can intersect $\overline{a^-a^+}$ at most once, so each such component can intersect T at most once.

We can further classify the type (iii) components into three types, depending on whether p_i lies on the part of e^- in T and whether q_i lies on the part of e^+ in B. A case-by-case analysis shows that there are up to four components of $\pi \cap F$ that intersect T, and each of them intersects T in one consecutive subpath. Hence, T is 4-happy. ◀

▶ **Lemma 20.** *Let F be a parallelogram with two parallel portal edges. We can compute an equivalent 2-happy parallelogram F' in $O(1)$ time.*

Proof sketch. To compute the new, equivalent fragment F', we could explicitly generate the ray with slope z, and compute its intersection points with the portal edges, tracing the ray until it hits a non-portal edge of the fragment. However, this would result in a running time linear in the number of such intersections. Instead, we show in the full version that after cutting along the ray and gluing the pieces together, the result is always a rectangle (as in Figure 13, right) that can be computed directly in $O(1)$ time. ◀

Single non-parallel portal analysis. In the full version [18], we address the case where the fragment is not a parallelogram, and, more interestingly, where the two portal edges are not parallel. The proof of the equivalent of Lemma 19 proceeds analogously, but is more technical. Lemma 20, however, does not have a direct analogue. In particular, the time required to compute an equivalent happy portalgon depends also on the bit complexity of the input coordinates. However, we show that we can achieve $O(\log h)$ time for h the initial happiness of the portalgon. We do this by encoding a single pass through the portal in a transformation matrix M, and computing M^h by performing an exponential search in h. This result is summarized as follows.

▶ **Theorem 21.** *Let $\mathcal{P}$ be an h-happy portalgon with n vertices and fragment graph G, such that G has at most one simple cycle. We can transform $\mathcal{P}$ into an equivalent 5-happy portalgon $\mathcal{P}'$ of total complexity $O(n)$ in $O(n + \log h)$ time.*

References

1. Pankaj K. Agarwal, Boris Aronov, Joseph O'Rourke, and Catherine A. Schevon. Star unfolding of a polytope with applications. *SIAM Journal on Computing*, 26(6):1689–1713, 1997. doi:10.1137/S0097539793253371.
2. Boris Aronov and Joseph O'Rourke. Nonoverlap of the star unfolding. *Discrete & Computational Geometry*, 8(3):219–250, 1992.
3. Alexander I Bobenko and Boris A Springborn. A discrete Laplace–Beltrami operator for simplicial surfaces. *Discrete & Computational Geometry*, 38(4):740–756, 2007.
4. Vincent Borrelli, Saïd Jabrane, Francis Lazarus, and Boris Thibert. Flat tori in three-dimensional space and convex integration. *Proceedings of the National Academy of Sciences*, 109(19):7218–7223, 2012. doi:10.1073/pnas.1118478109.
5. Jindong Chen and Yijie Han. Shortest paths on a polyhedron. *Int. J. Comput. Geom. Appl.*, 6(2):127–144, 1996. doi:10.1142/S0218195996000095.
6. Valve Corporation. Portal, 2007. Video game.
7. Yu. D. Burago V. A. Zalgaller (Eds.). *Geometry III: Theory of Surfaces*. Springer Verlag, 1993.
8. H Ellison. The city on the edge of forever, 1967. Star Trek, season 1, episode 28.

9 Jeff Erickson. Ernie's 3d pancakes: Shortest paths on pl surfaces, 2006. March 14, 2023. URL: https://3dpancakes.typepad.com/ernie/2006/03/shortest_paths_.html.

10 Jeff Erickson and Amir Nayyeri. Tracing compressed curves in triangulated surfaces. *Discret. Comput. Geom.*, 49(4):823–863, 2013. doi:10.1007/s00454-013-9515-z.

11 Leonidas Guibas and John Hershberger. Optimal shortest path queries in a simple polygon. *Journal of Computer and System Sciences*, 39(2):126–152, 1989. doi:10.1016/0022-0000(89)90041-X.

12 Leonidas Guibas, John Hershberger, Daniel Leven, Micha Sharir, and Robert E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2(1):209–233, 1987. doi:10.1007/BF01840360.

13 John Hershberger and Subhash Suri. An Optimal Algorithm for Euclidean Shortest Paths in the Plane. *SIAM Journal on Computing*, 28(6):2215–2256, 1999.

14 Biliana Kaneva and Joseph O'Rourke. An implementation of Chen & Han's shortest paths algorithm. URL: http://cs.smith.edu/~jorourke/Papers/shortest.ps.gz.

15 Stephen Kiazyk, Sébastien Loriot, and Éric Colin de Verdière. Triangulated surface mesh shortest paths. URL: https://doc.cgal.org/latest/Surface_mesh_shortest_path/index.html.

16 Francis Lazarus and Florent Tallerie. A Universal Triangulation for Flat Tori. *CoRR*, March 2022. URL: https://arxiv.org/abs/2203.05496.

17 Yong-Jin Liu, Chun-Xu Xu, Dian Fan, and Ying He. Efficient Construction and Simplification of Delaunay Meshes. *ACM Transactions on Graphics*, 34(6):1–13, 2015. doi:10.1145/2816795.2818076.

18 Maarten Löffler, Tim Ophelders, Rodrigo I. Silveira, and Frank Staals. Shortest paths in portalgons. *CoRR*, 2023. URL: https://arxiv.org/abs/2303.08937.

19 A.D. Milka. Multidimensional spaces with polyhedral metric of nonnegative curvature I. *Ukrain. Geom. Sb.*, 5(6):103–114, 1968. In Russian.

20 Joseph S. B. Mitchell. A new algorithm for shortest paths among obstacles in the plane. *Annals of Mathematics and Artificial Intelligence*, 3(1):83–105, March 1991. doi:10.1007/BF01530888.

21 Joseph S. B. Mitchell. Shortest paths and networks. In Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Toth, editors, *Handbook of Discrete and Computational Geometry, Third Edition*, pages 811–848. Chapman and Hall/CRC, 2017.

22 Joseph S. B. Mitchell, David M. Mount, and Christos H. Papadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16(4):647–668, 1987. doi:10.1137/0216045.

23 Joseph S. B. Mitchell and Christos H. Papadimitriou. The weighted region problem: Finding shortest paths through a weighted planar subdivision. *J. ACM*, 38(1):18–73, 1991. doi:10.1145/102782.102784.

24 Yevgeny Schreiber. An optimal-time algorithm for shortest paths on realistic polyhedra. *Discret. Comput. Geom.*, 43(1):21–53, 2010. doi:10.1007/s00454-009-9136-8.

25 A. Wachowski and L. Wachowski. Matrix revolutions, 2003. Warner Bros. Motion picture.

26 Haitao Wang. A new algorithm for Euclidean shortest paths in the plane. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 975–988, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3406325.3451037.

27 Haitao Wang. Shortest paths among obstacles in the plane revisited. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 810–821. SIAM, 2021. doi:10.1137/1.9781611976465.51.

28 Shi-Qing Xin and Guo-Jin Wang. Improving Chen and Han's algorithm on the discrete geodesic problem. *ACM Trans. Graph.*, 28(4), September 2009. doi:10.1145/1559755.1559761.