A Fast Shortest Path Algorithm on Terrain-like Graphs

Vincent Froese¹ · Malte Renken¹

Received: 5 July 2019 / Revised: 8 June 2020 / Accepted: 11 June 2020 / Published online: 4 August 2020
© The Author(s) 2020

Abstract

Terrain visibility graphs are a well-known graph class in computational geometry. They are closely related to polygon visibility graphs, but a precise graph-theoretical characterization is still unknown. Over the last decade, terrain visibility graphs attracted considerable attention in the context of time series analysis (there called time series visibility graphs) with various practical applications in areas such as physics, geography, and medical sciences. Computing shortest paths in visibility graphs is a common task, for example, in line-of-sight communication. For time series analysis, graph characteristics involving shortest paths lengths (such as centrality measures) have proven useful. In this paper, we devise a fast output-sensitive shortest path algorithm on a superclass of terrain visibility graphs called terrain-like graphs (including all induced subgraphs of terrain visibility graphs). Our algorithm runs in $O(d^* \log \Delta)$ time, where d^* is the length (that is, the number of edges) of the shortest path and Δ is the maximum vertex degree. Alternatively, with an $O(n^2)$ -time preprocessing our algorithm runs in $O(d^*)$ time.

Keywords Output-sensitive algorithm · Polygon visibility graphs · Time series visibility graphs · Vertex ordering

Mathematics Subject Classification 68W40 · 68R10 · 05C12 · 05C62 · 05C45 · 05C85

Editor in Charge: Kenneth Clarkson

Malte Renken was supported by the Deutsche Forschungsgemeinschaft (Project NI 369/17-1).

Vincent Froese
vincent.froese@tu-berlin.de

Malte Renken
m.renken@tu-berlin.de

¹ Algorithmics and Computational Complexity, Faculty IV, TU Berlin, Berlin, Germany

1 Introduction

Visibility graphs are a fundamental concept in computational geometry. For a given set of geometrical objects (e.g. points, segments, rectangles, polygons) they encode which objects are visible to each other. To this end, the objects form the vertices of the graph and there is an edge between two vertices if and only if the two corresponding objects “can see each other” (for a specified notion of visibility). Visibility graphs are well studied from a graph-theoretical perspective and find applications in many real-world problems occurring in different fields such as physics [11,12,24], robotics [6], object recognition [27], or medicine [3,15].

In this work, we consider so-called *terrain-like graphs* [5] which form a superclass of the visibility graphs of 1.5-dimensional terrains, that is, x -monotone polygonal chains (see Fig. 1). Terrain visibility graphs have been studied since the 90’s [1,2,9,10], yet a precise graph-theoretical characterization is still open. While certain necessary conditions for being a terrain visibility graph are known (see Sect. 2), it was recently shown that these conditions are not sufficient [4]. Terrain-like graphs are only required to satisfy one of these necessary conditions. They include all induced subgraphs of terrain visibility graphs.

Recently, terrain visibility graphs found numerous applications in analyzing and classifying time series [11,12,15,23,24,28]. Here, the so-called *complex network approach* is based on studying graph-theoretic measures of visibility graphs of time series. Examples of graph-theoretic measures used in time series analysis include *closeness centrality*, *betweenness centrality*, or *average path length* [11], all of which involve (unweighted) shortest paths. Hence, a fast algorithm for unweighted shortest path computation might speed up time series analysis tasks. Moreover, understanding the structure of shortest paths in terrain visibility graphs might explain in which cases these measures perform well.

Another natural example for the occurrence of terrain visibility graphs is a network of stations communicating via line-of-sight links (e.g. radio signals). Here, determining the distance between two stations is a common task. If the length of a path is measured

Fig. 1 A terrain visibility graph drawn in two different ways (with a corresponding terrain on the left). The shortest path of length three from the second leftmost to the rightmost vertex is drawn with thick lines

via Euclidean distance, then simply moving into the direction of the other station as far as possible is optimal. But a more realistic distance measure in the above scenario is the number of edges, as edge travel times can be negligible in comparison to the processing times at the stations. This setting is more difficult since it might now be better to move in the opposite direction first (as in Fig. 1). Moreover, in general there might also be terrain vertices which obstruct communication but are not communication stations themselves. Then the communication network is not a terrain visibility graph itself but an induced subgraph of a terrain visibility graph.

We develop an output-sensitive algorithm computing shortest paths in terrain-like graphs in $O(d^* \log \Delta)$ time, where d^* is the length of the shortest path and Δ is the maximum degree (also $O(d^*)$ time is possible with an $O(n^2)$ -time preprocessing). Our algorithm is based on a detailed analysis of the structure of a shortest path. Output-sensitivity is indeed favorable in cases where many shortest paths are of a length much smaller than the number of vertices (which might be the case in terrain visibility graphs).

Related Work For a general overview on visibility graphs and related problems see the survey by Ghosh and Goswami [18]. As regards the origin of terrain visibility graphs, Abello et al. [2] studied visibility graphs of staircase polygons which are closely related to terrain visibility graphs as Colley [9] showed that they are in one-to-one correspondence with the core induced subgraphs of staircase polygon visibility graphs. Abello et al. [2] described three necessary properties that are satisfied by every terrain visibility graph and conjectured that these are actually sufficient. Moreover, they showed that one can reconstruct (a possible) slope ordering from the graph with given vertex ordering. Evans and Saeedi [13] gave a simpler proof for the results of Abello et al. [2]. Ameer et al. [4] proved that the necessary properties described by Abello et al. [2] are in fact not sufficient. Abello and Egecioğlu [1] showed that visibility graphs of staircase polygons with unit step-length can be recognized via linear programming and that not all staircase polygon visibility graphs can be represented with unit steps. Choi et al. [8] studied another subclass of terrain visibility graphs called *funnel visibility graphs* which is linear-time recognizable (also studied by Colley et al. [10]). Regarding possible induced subgraphs, it is known that terrain visibility graphs can contain arbitrary large holes but no antiholes of size larger than five [14]. Ashur et al. [5] introduced terrain-like graphs and studied approximation algorithms computing dominating sets.

Lacasa et al. [23] introduced terrain visibility graphs (under the name of *time series visibility graphs*) in the context of time series analysis. A variant (called *horizontal visibility graphs*) where two vertices can only see each other horizontally was later introduced by Luque et al. [26]. Horizontal visibility graphs were fully characterized by Gutin et al. [20] who showed that these are exactly the outerplanar graphs with a Hamiltonian path. Moreover, Luque and Lacasa [25] showed that certain canonical horizontal visibility graphs are uniquely determined by their degree sequence (in contrast to terrain visibility graphs [14]).

As regards shortest path computation (see also the book by Ghosh [17]), it is known that Euclidean shortest paths from a fixed source vertex to all other vertices inside a simple (triangulated) polygon can be computed in $O(n)$ time [19]. Also, an unweighted

shortest path between two vertices of a simple polygon can be computed in $O(n)$ time [16,22,29]. Note that these algorithms require the polygon vertices to be given. To the best of our knowledge, no specific algorithms on (arbitrary induced subgraphs of) the visibility graph of a polygon or terrain (with given vertex ordering) have been developed so far.

2 Preliminaries

We assume the reader to be familiar with basic concepts and classes of graphs (refer to Brandstädt et al. [7] for an overview).

Terrain-like graphs are motivated by terrain visibility graphs, which we introduce first. A (1.5-dimensional) *terrain* is an x -monotone polygonal chain in the plane defined by a set V of *terrain vertices* with pairwise different x -coordinates. For two terrain vertices p, q , we write $p < q$ if p is “left” of q , that is, $p^x < q^x$, where p^x denotes the x -coordinate of p . Furthermore, we define the interval $[p, q] := \{x \in V \mid p \leq x \leq q\}$. The corresponding *terrain visibility graph* is defined on the set of terrain vertices where two vertices p and q are adjacent if and only if they *see each other*, that is, there is no vertex between them that lies on or above the line segment connecting p and q (see Fig. 1 for an example). Formally, there exists an edge $\{p, q\}$, for $p < q$, if and only if all terrain vertices r with $p < r < q$ satisfy

$$r^y < p^y + (q^y - p^y) \frac{r^x - p^x}{q^x - p^x}.$$

Hershberger [21] gave an algorithm to compute the visibility graph of a given terrain with a running time that is linear in the size of the graph.

Clearly, every terrain visibility graph contains a Hamiltonian path along the order of the terrain vertices. Moreover, the following two conditions are necessary for being a terrain visibility graph as shown by Abello et al. [2] (we stick to the names coined by Evans and Saeedi [13]).

X-property: Let p, q, r, s be four terrain vertices with $p < q < r < s$. If p sees r and q sees s , then p sees s .

Bar-property: If p, q are two non-consecutive terrain vertices that can see each other, then there is a vertex r between p and q that can see both of them.

It was conjectured that the existence of a vertex ordering satisfying the above two properties fully characterizes terrain visibility graphs. However, Ameer et al. [4] recently disproved the conjecture and gave a counterexample.

Now, a *terrain-like* graph is simply defined to have a vertex ordering fulfilling the X-property [5]. Note that every induced subgraph of a terrain visibility graph is terrain-like. Interestingly, the converse is not true since $\overline{C}_6$ (that is, the complement of a 6-cycle) can satisfy the X-property (see Fig. 2) but cannot be an induced subgraph of a terrain visibility graph [14, Thm. 3.5].

Fig. 2 An ordering of the vertices of the graph $\overline{C}_6$ that satisfies the X-property. This graph is terrain-like but not a terrain visibility graph

Fig. 3 Two sketches of a shortest path containing the edges $\{v, v'\}$, $\{v, w'\}$ and $\{w, w'\}$. The dashed edge $\{x, x'\}$ cannot also be on the path since this would imply the existence of the other dashed edge

3 Fast Shortest Path Computation

In this section we develop an efficient shortest path algorithm on terrain-like graphs. In the following, let G be an input graph with a (known) vertex ordering satisfying the X-property. Furthermore, let s and t be two vertices of G with $s < t$ and $\text{dist}(s, t) = d^* < \infty$, where $\text{dist}(s, t)$ denotes the length (that is, the number of edges) of a shortest path from s to t . Two edges $\{p, q\}$ and $\{r, s\}$ are said to be *crossing* if $p < r < q < s$. We start with the crucial observation that a shortest path contains at most one pair of crossing edges.

Lemma 3.1 *If a shortest s - t path P contains a pair of crossing edges $\{v, v'\}$, $\{w, w'\}$ with $v < w < v' < w'$, then it also contains the edge $\{v, w'\}$. Furthermore, this is the only pair of crossing edges in P and $V(P) \subseteq [v, w']$.*

Proof Let P be a shortest s - t path and $\{v, v'\}$, $\{w, w'\}$ two crossing edges with $v < w < v' < w'$. By the X-property, G contains the edge $\{v, w'\}$. Since P is a shortest path and thus also an induced path, it must contain that edge (see Fig. 3). We claim that no other edge $\{x, x'\}$, $x < x'$, of P can cross $\{v, v'\}$. If $x < v < x' < v'$, then $\{x, x'\}$ would also cross $\{v, w'\}$. Hence, by the X-property, P would have to contain the edge $\{x, w'\}$, which is not possible since w' would be incident to three edges of P . Otherwise, if $v < x < v' < x'$, then P would contain the edge $\{v, x'\}$, which is again not possible. By symmetry, no other edge of P can cross $\{w, w'\}$.

Consequently, P contains only vertices in $[v, w']$ since otherwise P would contain an edge from a vertex $u \notin [v, w']$ to a vertex x with $v \leq x \leq w'$. Since v, w' are both already contained in two edges of P , this would imply $v < x < w'$ and thus the edge $\{u, x\}$ would cross $\{v, v'\}$ or $\{w, w'\}$. It follows that P cannot contain a second pair of crossing edges since the same argument would apply to that pair. $\square$

We denote a shortest s - t path P of length d^* by its vertices $s = p_0, p_1, \dots, p_{d^*} = t$ and define $\text{li}(P)$ to be the index of the leftmost vertex of P . Analogously, $\text{ri}(P)$ is the index of the rightmost vertex of P . The following can be obtained from Lemma 3.1 (see also Fig. 6 (iii)).

Lemma 3.2 *Let P be a shortest s - t path. Then, $\text{ri}(P) < \text{li}(P)$ holds if and only if P contains a pair of crossing edges. Moreover, if P contains a pair of crossing edges, then $\text{ri}(P) = \text{li}(P) - 1$ and $p_i < p_j$ holds for all $i < \text{ri}(P) < \text{li}(P) < j$.*

Proof If $\text{ri}(P) < \text{li}(P)$, then P must contain a pair of crossing edges between the subpaths from s to $p_{\text{ri}(P)}$ and from $p_{\text{li}(P)}$ to t . Conversely, if P contains a pair of crossing edges, then, by Lemma 3.1, the two crossing edges must be $\{p_{\text{ri}(P)-1}, p_{\text{ri}(P)}\}$ and $\{p_{\text{li}(P)}, p_{\text{li}(P)+1}\}$. This implies that P contains the edge $\{p_{\text{ri}(P)}, p_{\text{li}(P)}\}$, which implies $\text{ri}(P) = \text{li}(P) - 1$. Then, for all $i < \text{ri}(P) < \text{li}(P) < j$, it holds that $p_i < p_j$ since P has no other crossing edges and $p_{\text{ri}(P)-1} < p_{\text{li}(P)+1}$. $\square$

Clearly, Lemma 3.2 implies that $\text{li}(P) < \text{ri}(P)$ if and only if P contains no crossing edges. Moreover, the following holds (see Fig. 6, (i) and (ii)).

Lemma 3.3 *Let P be a shortest s - t -path with $\text{li}(P) < \text{ri}(P)$. Then, $p_i < p_j$ holds for all $i < \text{li}(P) < j$ and all $i < \text{ri}(P) < j$.*

Proof We prove both cases simultaneously. Assume towards a contradiction that $p_j < p_i$, then we have $p_{\text{li}(P)} < p_j < p_i < p_{\text{ri}(P)}$. Consider the subpath P_1 of P that connects p_i with $p_{\text{li}(P)}$ and the subpath P_2 of P that connects p_j with $p_{\text{ri}(P)}$. These two subpaths are vertex-disjoint and must thus be crossing, meaning that there is a pair of crossing edges e_1, e_2 where e_1 is an edge of P_1 and e_2 an edge of P_2 . By Lemma 3.2, this implies $\text{ri}(P) < \text{li}(P)$, which is a contradiction. $\square$

In combination, Lemmas 3.2 and 3.3 imply the following corollary.

Corollary 3.4 *For a shortest s - t path P , it holds that all vertices p_i with $i < \text{ri}(P)$ satisfy $p_i < t$ and all vertices p_i with $i > \text{li}(P)$ satisfy $p_i > s$.*

The results derived above characterize the global structure of a shortest s - t path. Next, we will investigate the local structure. The most important consequence of the X-property is that large steps are usually better than small steps. We formalize this in the following.

Define $\alpha_s(k)$ as the *leftmost* and $\beta_s(k)$ as the *rightmost* vertex that can be reached from s by a path of length at most k that only uses vertices $v \leq t$. We symmetrically define $\alpha_t(k)$ as the *rightmost* and $\beta_t(k)$ as the *leftmost* vertex reachable from t by a path of length at most k using only vertices $v \geq s$. Clearly $\alpha_s(k) \leq s \leq \beta_s(k) \leq t$ and $s \leq \beta_t(k) \leq t \leq \alpha_t(k)$ for all $k \geq 0$ (compare also Fig. 4).

The following lemma shows that there is a shortest path from s to $\alpha_s(k)$ that uses only vertices in $\{\alpha_s(i), \beta_s(i) \mid 0 \leq i \leq k\}$. Here, $N[v] := \{v, w \mid \{v, w\} \in E(G)\}$ denotes the closed neighborhood of v .

Lemma 3.5 *Let $k \geq 1$. Then $N[\alpha_s(k)]$ contains $\alpha_s(k-1)$ or $\beta_s(k-1)$.*

Fig. 4 Example graph with all α - and β -vertices labeled. Note that $\alpha_s(2) = \alpha_s(3) = \dots$, $\beta_s(2) = \beta_s(3) = \dots$, $\alpha_t(1) = \alpha_t(2) = \dots$, and $\beta_t(4) = \beta_t(5) = \dots$. Furthermore, $s = \alpha_s(0) = \beta_s(0)$ and $t = \alpha_t(0) = \beta_t(0)$

Proof Clearly, the statement trivially holds for $k = 1$ and if $\alpha_s(k) = \alpha_s(k-1)$. Hence, assume that $k \geq 2$ and $\alpha_s(k) \neq \alpha_s(k-1)$. In the following, we only consider vertices to the left of t , which is why we assume that t is the rightmost vertex for the remainder of this proof. Let $\{x, \alpha_s(k)\}$ be the last edge of a shortest path from s to $\alpha_s(k)$. By definition, we have $\alpha_s(k-1) \leq x$.

If $x < s$, then let Q be a shortest path from s to $\alpha_s(k-1)$. If x is a vertex of Q , then $x = \alpha_s(k-1)$ (otherwise we have a path from s to $\alpha_s(k)$ of length at most $k-1$ implying $\alpha_s(k) = \alpha_s(k-1)$) and we are done. Otherwise, Q contains an edge $\{y, y'\}$ with $y < x < y'$. By the X-property, there is an edge between $\alpha_s(k)$ and y' , which yields a path of length at most $k-1$ between s and $\alpha_s(k)$, implying $\alpha_s(k-1) = \alpha_s(k)$.

If $x > s$, then let Q be a shortest path from s to $\beta_s(k-1)$. If x is a vertex of Q , then $x = \beta_s(k-1)$ (otherwise we have a path from s to $\alpha_s(k)$ of length at most $k-1$ implying $\alpha_s(k) = \alpha_s(k-1)$) and we are done. Otherwise, Q contains an edge $\{y, y'\}$ with $y < x < y'$. By the X-property, there is an edge between $\alpha_s(k)$ and y' , which yields a path of length at most $k-1$ between s and $\alpha_s(k)$, implying $\alpha_s(k-1) = \alpha_s(k)$. $\square$

Due to symmetry, Lemma 3.5 also holds if one exchanges α and β , and also when s is replaced by t . With the following lemma, we will be able to restrict our search for a shortest s - t path mostly to the α and β vertices.

Lemma 3.6 *There is a shortest s - t path $P = p_0, \dots, p_{d^*}$ such that*

- (i) $p_i \in \{\alpha_s(i), \beta_s(i)\}$ for all $i < \text{ri}(P)$, and
- (ii) $p_i \in \{\alpha_t(d^* - i), \beta_t(d^* - i)\}$ for all $i > \text{li}(P)$.

Proof We begin by showing that there always exists a shortest s - t path satisfying (i) (see Fig. 5 for an illustration). Assume towards a contradiction, that no shortest s - t path satisfies (i). Then, let $P = p_0, p_1, \dots, p_{d^*}$ be a shortest s - t path chosen so that it maximizes i , where $i < \text{ri}(P)$ is the smallest index such that $p_i \notin \{\alpha_s(i), \beta_s(i)\}$. Since $i < \text{ri}(P)$, we know that all of $p_0, \dots, p_i$ are to the left of t by Corollary 3.4. We assume that $s < p_i$ (the case $s > p_i$ is similar and uses that $p_i \neq \alpha_s(i)$). Since $p_i \neq \beta_s(i)$, it holds $p_i < \beta_s(i)$. Note that also $\beta_s(i) < t$, since $\beta_s(i) = t$ would contradict $\text{dist}(s, \beta_s(i)) = i < d^* = \text{dist}(s, t)$. By Lemma 3.5, there exists a shortest path $Q = q_0, q_1, \dots, q_i$ from $s = q_0$ to $\beta_s(i) = q_i$ where $q_j \in \{\alpha_s(j), \beta_s(j)\}$ for

Fig. 5 Left: The situation described in the proof of Lemma 3.6. Note that the proof shows that $\mu = i$ (that is, $q_{\mu} = \beta_s(i)$) and $v = i + 1$ (that is, $p_{v-1} = p_i$). The zigzag line from s to $q_{\mu-1}$ indicates a path of length $i - 1$, whereas the zigzag line from p_v to t indicates a path of length $d^* - i - 1$. Right: The (similar) situation for the case $p_i < s$. Note that also here $\mu = i$ and $v = i + 1$ holds (that is, $p_{v-1} = p_i$ and $q_{\mu} = \alpha_s(i)$)

all $0 \leq j \leq i$. Let $\mu \leq i$ be an index such that $q_{\mu-1} < p_i < q_{\mu}$. Further, let $v > i$ be the minimal index for which $p_v \notin [q_{\mu-1}, q_{\mu}]$ (that is, $p_{v-1} \in [q_{\mu-1}, q_{\mu}]$). Then, $\{q_{\mu-1}, q_{\mu}\}$ and $\{p_{v-1}, p_v\}$ are crossing edges. By the X-property, G contains the edge $\{q_{\mu'}, p_v\}$, where either $\mu' = \mu$ (if $p_v < q_{\mu-1} < p_{v-1} < q_{\mu}$) or $\mu' = \mu - 1$ (if $q_{\mu-1} < p_{v-1} < q_{\mu} < p_v$). Hence, we obtain an s - t walk $P' = q_0, \dots, q_{\mu'}, p_v, \dots, p_{d^*}$ of length $\mu' + 1 + (d^* - v)$. Clearly, this length must be at least d^* , which implies $\mu' + 1 \geq v$. Since $\mu' \leq i$ and $v \geq i + 1$, this is only possible for $\mu' = \mu = i$ and $v = i + 1$. Hence, P' has length d^* , which means that P' is a shortest path from s to t (with $i < \text{ri}(P')$). Recall that, by construction, $q_j \in \{\alpha_s(j), \beta_s(j)\}$ holds for all $0 \leq j \leq i$, which contradicts the choice of P . This proves the existence of a shortest s - t path satisfying (i).

Before moving on, we show that if the chosen shortest path P in the above argument satisfies (ii), then also P' does so. This observation will be helpful for the second part of the proof. To this end, note that we concluded above that $\mu' = \mu = i$ and $v = i + 1$, which implies that $p_{i+1} < p_i$. Assume towards a contradiction that $p_j < p_{i+1}$ for some $j < i$. Then, there must be a pair of crossing edges from the subpaths from p_j to p_i and from p_{i+1} to t . By the X-property, there is now an edge between two non-consecutive vertices of P , contradicting the fact that P is an induced path. Thus, p_{i+1} must be to the left of $p_0, \dots, p_i$. This implies $i < \text{li}(P)$, and thus, $p_{\text{li}(P)}$ is also a vertex of P' . Furthermore, all of $q_0, \dots, q_i$ must be to the right of $p_{\text{li}(P)}$ because otherwise P' would contain a pair of crossing edges from the subpaths from s to q_i and from $p_{\text{li}(P)}$ to t . Thus, $\text{li}(P) = \text{li}(P')$. We conclude that $i < \text{li}(P) = \text{li}(P')$, that is, P' also satisfies (ii).

Now, assume towards a contradiction that no shortest s - t path satisfies both (i) and (ii). Then, let $P = p_0, p_1, \dots, p_{d^*}$ be a shortest s - t path that satisfies (i) and which is chosen to minimize $i > \text{li}(P)$, which is the largest index such that $p_i \notin \{\alpha_t(d^* - i), \beta_t(d^* - i)\}$. Now, if we reverse the vertex ordering of G , swap s and t , and reverse the direction of P , then we obtain a shortest s - t path $\tilde{P}$ satisfying (ii) but not (i). Applying the above arguments to $\tilde{P}$ now yields a contradiction. $\square$

Fig. 6 The three cases distinguished by Lemma 3.7 (only the vertices of a shortest path are drawn)

Note that the only case in which Lemma 3.6 does not restrict all vertices to be α or β vertices is when $ri(P) < li(P)$, in which case Lemma 3.2 applies. This will allow us to show that, when searching for a shortest s - t path, it is sufficient to check for three possible cases (illustrated in Fig. 6). In the following, we define the *left horizon* of a vertex v as the furthest neighbor in that direction, that is, $lhorizon(v) := \min(N[v])$ and $rhorizon(v) := \max(N[v])$. Remember that $d^* = \text{dist}(s, t)$.

Lemma 3.7 *Each of the following conditions implies that $d^* \leq d$.*

- (i) $\beta_s(\sigma) = \beta_t(\tau)$ with $\sigma + \tau = d$.
- (ii) G contains an edge between $\alpha_s(\sigma)$ and $\alpha_t(\tau)$ with $\sigma + \tau = d - 1$.
- (iii) There exist $v \in \{\alpha_s(\sigma), \beta_s(\sigma)\}$ and $w \in \{\alpha_t(\tau), \beta_t(\tau)\}$ with $v < w$ and $\sigma + \tau = d - 3$ such that $rhorizon(v) \geq t$ and $lhorizon(w) \leq s$.

Proof For (i) and (ii), the existence of an s - t path of length at most d follows directly from the definition of α and β . Assume now that (iii) is true. We may also assume $s < w$ since $s = w$ would directly imply $d^* = \tau < d$. Define $h_v := rhorizon(v)$ and $h_w := lhorizon(w)$. If the edges $\{v, h_v\}$ and $\{w, h_w\}$ are crossing, then the existence of an s - t path of length $\sigma + \tau + 3$ follows from the X-property. Otherwise, let $v \leq h_w \leq s < w$ (the case $v < t \leq h_v \leq w$ is symmetric). Let P be a shortest s - v path. Then, P either crosses the edge $\{w, h_w\}$ or contains w or h_w . In any case, we obtain an s - w path of length at most $\sigma + 1$ and thus an s - t path of length at most $\sigma + \tau + 1 < d$. $\square$

Lemma 3.8 *At least one of the conditions (i)–(iii) in Lemma 3.7 holds for $d = d^*$.*

Proof Let $P = p_0, p_1, \dots, p_{d^*}$ be a shortest s - t path satisfying conditions (i) and (ii) as described in Lemma 3.6.

If $li(P) < ri(P) - 1$, then, for $\sigma := li(P) + 1$, it holds $0 < \sigma < d^*$ and $p_\sigma \in \{\alpha_s(\sigma), \beta_s(\sigma)\} \cap \{\alpha_t(d^* - \sigma), \beta_t(d^* - \sigma)\}$. By definition, for all j , we have

$$\alpha_s(j) \leq s < t \leq \alpha_t(j), \quad s \leq \beta_s(j) \leq t, \quad s \leq \beta_t(j) \leq t.$$

Clearly, $p_\sigma \notin \{s, t\}$, thus we must have $p_\sigma = \beta_s(\sigma) = \beta_t(d^* - \sigma)$, and therefore (i) is true with $\tau := d^* - \sigma$.

If $\text{li}(P) = \text{ri}(P) - 1$, then P contains the edge $\{p_{\text{li}(P)}, p_{\text{ri}(P)}\}$. Moreover, $p_{\text{li}(P)} = \alpha_s(\text{li}(P))$ and $p_{\text{ri}(P)} = \alpha_t(d^* - \text{ri}(P))$ clearly holds. Hence, (ii) is true with $\sigma := \text{li}(P)$ and $\tau := d^* - \text{ri}(P)$.

If $\text{ri}(P) < \text{li}(P)$, then, by Lemma 3.2, the two edges $\{p_{\text{ri}(P)-1}, p_{\text{ri}(P)}\}$ and $\{p_{\text{li}(P)}, p_{\text{li}(P)+1}\}$ are crossing, that is, $p_{\text{ri}(P)-1} < p_{\text{li}(P)+1}$ and $\text{ri}(P) = \text{li}(P) - 1$. Thus, (iii) holds with $v := p_{\text{ri}(P)-1}$, $w := p_{\text{li}(P)+1}$, $\sigma := \text{ri}(P) - 1$, and $\tau := d^* - (\text{li}(P) + 1)$. Note that $\text{rhorizon}(p_{\text{ri}(P)-1}) \geq p_{\text{ri}(P)} \geq t$ and $\text{lhhorizon}(p_{\text{li}(P)+1}) \leq p_{\text{li}(P)} \leq s$. $\square$

We are now ready to present our shortest path algorithm (Algorithm 1). It uses the following two definitions of the neighbors of a vertex v that are closest to another vertex a .

$$\begin{aligned} \text{rclosest}_a(v) &:= \min \{x \in N[v] \mid x \geq a\}, \\ \text{lcloset}_a(v) &:= \max \{x \in N[v] \mid x \leq a\}. \end{aligned}$$

If the sets on the right hand side are empty, we treat these values as ∞ resp. $-\infty$. For given vertices v and a , these two neighbors can be computed in $O(\log(\deg(v)))$ time using binary search on the (sorted) adjacency list of v . It is also possible to compute them for a fixed vertex v and all other vertices a in $O(n)$ time.

Algorithm 1 iteratively computes the values $\alpha_s(k)$, $\beta_s(k)$, $\alpha_t(k)$, and $\beta_t(k)$ for increasing values of k and checks each time, whether one of the three cases of Lemma 3.7 holds. Condition (i) is easy to test. To test condition (iii), the algorithm stores the values

$$\begin{aligned} q_s &:= \min \{\sigma \leq k \mid \text{rhorizon}(\alpha_s(\sigma)) \geq t \vee \text{rhorizon}(\beta_s(\sigma)) \geq t\} \quad \text{and} \\ q_t &:= \min \{\tau \leq k \mid \text{lhhorizon}(\alpha_t(\tau)) \leq s \vee \text{lhhorizon}(\beta_t(\tau)) \leq s\}. \end{aligned}$$

In contrast, checking condition (ii) requires slightly more effort. To test for the existence of an edge $\{\alpha_s(\sigma), \alpha_t(\tau)\}$ with $\sigma \leq \tau$, we store the computed values for $\alpha_s(\sigma)$ in a list Q_s , sorted in such a way that the vertices $r_\sigma := \text{rclosest}_t(\alpha_s(\sigma))$ are ordered from left to right. Whenever the iteration reaches a value τ such that $\alpha_t(\tau)$ is to the right of the leftmost of these r_σ , we have an edge of the form $\{\alpha_s(\sigma), \alpha_t(\tau)\}$ with $\sigma \leq \tau$ (by the X-property). To test for edges $\{\alpha_s(\sigma), \alpha_t(\tau)\}$ with $\sigma \geq \tau$, we use a symmetrical procedure with a list Q_t .

Theorem 3.9 *Algorithm 1 computes $d^* = \text{dist}(s, t)$ in $O(d^*)$ time (excluding the time to compute all $\text{lcloset}_a(v)$ and $\text{rclosest}_a(v)$).*

Proof We begin by showing the following invariants which are maintained throughout the computation.

1. At the beginning of each iteration of the for loop (line 8), we have $\alpha_s = \alpha_s(k)$, $\alpha_t = \alpha_t(k)$, $\beta_s = \beta_s(k)$, and $\beta_t = \beta_t(k)$. This is ensured by the EXTENDSEARCHRANGE procedure (based on Lemma 3.5).

Algorithm 1 Shortest Path

Input: A terrain-like graph G with a vertex ordering satisfying the X-property and two vertices $s < t$.
Output: $\text{dist}(s, t)$.

```

1:  $\alpha_s \leftarrow \beta_s \leftarrow s$                                  $\triangleright \alpha_s(k), \beta_s(k)$  for the current value of  $k$ 
2:  $\alpha_t \leftarrow \beta_t \leftarrow t$                              $\triangleright \alpha_t(k), \beta_t(k)$  for the current value of  $k$ 
3:  $d_s[v] \leftarrow d_t[v] \leftarrow \infty \forall v \in V(G)$          $\triangleright$  Distances from  $s$  resp.  $t$ 
4:  $d \leftarrow \infty$                                            $\triangleright$  Smallest  $s$ - $t$  distance found so far
5:  $Q_s \leftarrow [s]$                                         $\triangleright$  Candidate vertices for test (ii) ( $\sigma \leq \tau$ ), sorted by ascending  $\text{rclosest}_t(\cdot)$ 
6:  $Q_t \leftarrow [t]$                                         $\triangleright$  Candidate vertices for test (ii) ( $\sigma \geq \tau$ ), sorted by descending  $\text{lclosest}_s(\cdot)$ 
7:  $q_s \leftarrow q_t \leftarrow \infty$                          $\triangleright$  Distances for test (iii)
8: for  $k \leftarrow 0, 1, \dots, d$  do
9:   UPDATEDISTANCES( $k$ )                                 $\triangleright$  Updates  $d_s, d_t, Q_s, Q_t, q_s$ , and  $q_t$ 
10:  if  $\beta_s \geq \beta_t$  then                                   $\triangleright$  Test (i)
11:     $d \leftarrow \min \{d, d_s[\beta_s] + d_t[\beta_s], d_s[\beta_t] + d_t[\beta_t]\}$ 
12:  end if
13:  while  $\alpha_t \geq \text{rclosest}_t(\text{head}(Q_s))$  do               $\triangleright$  Test (ii) for  $\sigma \leq \tau$ 
14:     $d \leftarrow \min \{d, d_s[\text{head}(Q_s)] + d_t[\alpha_t] + 1\}$ 
15:    remove the head from  $Q_s$ 
16:  end while
17:  while  $\alpha_s \leq \text{lclosest}_s(\text{head}(Q_t))$  do               $\triangleright$  Test (ii) for  $\sigma \geq \tau$ 
18:     $d \leftarrow \min \{d, d_t[\text{head}(Q_t)] + d_s[\alpha_s] + 1\}$ 
19:    remove the head from  $Q_t$ 
20:  end while
21:  if  $q_s < \infty$  and  $q_t < \infty$  then                     $\triangleright$  Test (iii)
22:     $d \leftarrow \min \{d, q_s + q_t + 3\}$ 
23:  end if
24:  EXTENDSEARCHRANGE                                     $\triangleright$  Updates  $\alpha_s, \beta_s, \alpha_t$ , and  $\beta_t$  to  $k + 1$ 
25: end for
26: return  $d$ 

```

2. If $d_s[v] < \infty$ holds at some point, then $d_s[v] = \text{dist}(s, v)$ (the same holds for $d_t[v]$). This is ensured by the **UPDATEDISTANCES** procedure.
3. At every point, $d \geq d^*$ holds. This is obvious for line 11. For line 14, let $\alpha_s(h) = \text{head}(Q_s)$, $r_h = \text{rclosest}_t(\alpha_s(h))$, and let P be a shortest path from t to $\alpha_t(k)$. Then, either P contains r_h or crosses the edge $\{\alpha_s(h), r_h\}$ in which case the X-property implies the existence of an edge between $\alpha_s(h)$ and another vertex on P . In any case $\text{dist}(t, \alpha_s(h)) \leq \text{dist}(t, \alpha_t(k)) + 1$. Thus, $\text{dist}(s, t) \leq \text{dist}(s, \alpha_s(h)) + \text{dist}(t, \alpha_t(k)) + 1$. A symmetrical argument works for line 18. For line 22, if $q_s + q_t < \infty$, then, according to **UPDATEDISTANCES**, we found vertices $v_s \in \{\alpha_s(q_s), \beta_s(q_s)\}$ and $v_t \in \{\alpha_t(q_t), \beta_t(q_t)\}$ with $\text{dist}(s, v_s) = q_s$ and $h_s := \text{rhorizon}(v_s) \geq t$, and $\text{dist}(t, v_t) = q_t$ and $h_t := \text{lhhorizon}(v_t) \leq s$. We can assume that $\beta_s(k) < \beta_t(k)$, since otherwise test (i) would already have succeeded and set $d \leq q_s + q_t$. Therefore, we have $v_s \leq \beta_s(k) < \beta_t(k) \leq v_t$. Thus, condition (iii) of Lemma 3.7 holds which implies $\text{dist}(s, t) \leq q_s + q_t + 3$.

Having established the above invariants, we claim that $d \leq d^*$ holds after the for loop has reached $k = d^*$. By Lemma 3.8, we know that one of the conditions (i)–(iii) is true.

If condition (i) is true, then there are σ and τ with $\sigma + \tau = d^*$ and $\beta_s(\sigma) = \beta_t(\tau)$. Then, d will be set to d^* in line 11 as soon as k reaches $\max\{\sigma, \tau\}$.

Algorithm 2 UPDATEDISTANCES

```

1: procedure UPDATEDISTANCES( $k$ )
2:    $d_s[\alpha_s] \leftarrow \min\{k, d_s[\alpha_s]\}$ 
3:    $d_t[\alpha_t] \leftarrow \min\{k, d_t[\alpha_t]\}$ 
4:    $d_s[\beta_s] \leftarrow \min\{k, d_s[\beta_s]\}$ 
5:    $d_t[\beta_t] \leftarrow \min\{k, d_t[\beta_t]\}$ 
6:   if  $\max\{\text{rhorizon}(\alpha_s), \text{rhorizon}(\beta_s)\} \geq t$  then
7:      $q_s \leftarrow \min\{q_s, k\}$ 
8:   end if
9:   if  $\min\{\text{lhhorizon}(\alpha_t), \text{lhhorizon}(\beta_t)\} \leq s$  then
10:     $q_t \leftarrow \min\{q_t, k\}$ 
11:  end if
12:  if  $\text{rclosest}_t(\alpha_s) < \text{rclosest}_t(\text{head}(Q_s))$  then
13:    add  $\alpha_s$  to the front of  $Q_s$ 
14:  end if
15:  if  $\text{lcloset}_s(\alpha_t) > \text{lcloset}_s(\text{head}(Q_t))$  then
16:    add  $\alpha_t$  to the front of  $Q_t$ 
17:  end if
18: end procedure

```

Algorithm 3 EXTENDSEARCHRANGE

```

1: procedure EXTENDSEARCHRANGE
2:    $\alpha'_s \leftarrow \min_{x \in \{\alpha_s, \beta_s\}}(\text{lhhorizon}(x))$ 
3:    $\beta'_s \leftarrow \max_{x \in \{\alpha_s, \beta_s\}}(\text{lcloset}_t(x))$ 
4:    $\alpha_s \leftarrow \alpha'_s$ 
5:    $\beta_s \leftarrow \beta'_s$ 
6:    $\alpha'_t \leftarrow \max_{x \in \{\alpha_t, \beta_t\}}(\text{rhorizon}(x))$ 
7:    $\beta'_t \leftarrow \min_{x \in \{\alpha_t, \beta_t\}}(\text{rcloset}_s(x))$ 
8:    $\alpha_t \leftarrow \alpha'_t$ 
9:    $\beta_t \leftarrow \beta'_t$ 
10: end procedure

```

If condition (ii) is true, then there are σ and τ with $\sigma + \tau = d^* - 1$ such that $\alpha_s(\sigma)$ and $\alpha_t(\tau)$ are connected by an edge. As soon as $k = \sigma$, the vertex $\alpha_s(\sigma)$ will be added to Q_s by UPDATEDISTANCES and as soon as $k = \tau$, the vertex $\alpha_t(\tau)$ will be added to Q_t . Therefore, when k reaches the value $\max\{\sigma, \tau\}$, then d will be set to d^* either in line 14 or line 18 (depending on whether $\sigma \leq \tau$).

If condition (iii) is true, then there are σ and τ with $\sigma + \tau = d^* - 3$ and $v \in \{\alpha_s(\sigma), \beta_s(\sigma)\}$, $w \in \{\alpha_t(\tau), \beta_t(\tau)\}$ such that $\text{rhorizon}(v) \geq t$ and $\text{lhhorizon}(w) \leq s$. Then, q_s will be set to σ by UPDATEDISTANCES when $k = \sigma$ and q_t will be set to τ when $k = \tau$. Therefore, when k reaches the value $\max\{\sigma, \tau\}$, then d will be set to $q_s + q_t + 3 = d^*$ in line 22.

Since $d \geq d^*$ is an invariant, the for loop will at some point reach $k = d^*$, and thus, Algorithm 1 outputs exactly d^* . It remains to prove the running time. As shown above, the for loop is repeated at most $d^* + 1$ times. Apart from the inner while loops, each iteration only takes constant time. Since at most one vertex is added to Q_s and Q_t in each of these iterations, the inner while loops are also iterated at most $d^* + 1$ times overall. Note that initializing all $d_s[v]$ and $d_t[v]$ with ∞ in line 3 is only done

for ease of notation. In practice, these distance values only need to be stored once they get updated by `UPDATEDISTANCES`. $\square$

How does the running time change if we incorporate the computation of $\text{lcloset}_a(v)$ and $\text{rcloset}_a(v)$? Algorithm 1 requires knowledge of these values for all $a \in \{s, t\}$ and $v \in \{\alpha_b(i), \beta_b(i) \mid b \in \{s, t\}, i \leq d^*\}$. This amounts to $O(d^*)$ many computations, each taking $O(\log(\Delta))$ time where Δ is the maximal degree of any vertex. The overall running time is thus in $O(d^* \cdot \log(\Delta))$. Alternatively, one can precompute these values for all $a, v \in V(G)$ in $O(n^2)$ time and then run Algorithm 1 in output-optimal time $O(d^*)$ for any pair s, t .

Note that even though we always assumed $\text{dist}(s, t) < \infty$, Algorithm 1 can easily be extended to handle the case $\text{dist}(s, t) = \infty$ by additionally testing whether any of the vertices $\alpha_s, \beta_s, \alpha_t, \beta_t$ was updated in `EXTENDSEARCHRANGE`. If not, then the algorithm terminates.

4 Conclusion

We developed an output-sensitive algorithm for unweighted shortest path computation in terrain-like graphs. We conclude with some open questions. First, a fast algorithm for finding shortest paths with respect to Euclidean distances on terrain-like graphs (or on induced subgraphs of terrain visibility graphs) might be of interest. Second, our algorithm assumes that a vertex ordering satisfying the X-property is known. Is it possible to compute shortest paths without knowing the vertex ordering (or can one compute such an ordering efficiently)? Third, can the algorithm be extended to compute all shortest paths from a single source vertex to all other vertices? Finally, it might be worth to search for more efficient algorithms to compute other graph characteristics used for time series analysis in practice such as clustering coefficients or centrality measures.

Acknowledgements Open Access funding provided by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abello, J., Egecioglu, Ö.: Visibility graphs of staircase polygons with uniform step length. *Int. J. Comput. Geom. Appl.* **3**(1), 27–37 (1993)
2. Abello, J., Egecioglu, Ö., Kumar, K.: Visibility graphs of staircase polygons and the weak Bruhat order, I: from visibility graphs to maximal chains. *Discrete Comput. Geom.* **14**(3), 331–358 (1995)

3. Ahmadi, M., Adeli, H., Adeli, A.: New diagnostic EEG markers of the Alzheimer's disease using visibility graph. *J. Neural Transm.* **117**(9), 1099–1109 (2010)
4. Ameer, S., Gibson-Lopez, M., Krohn, E., Soderman, S., Wang, Q.: Terrain visibility graphs: persistence is not enough (2020). [arXiv:2004.00750](https://arxiv.org/abs/2004.00750)
5. Ashur, S., Filtser, O., Katz, M.J., Saban, R.: Terrain-like graphs: PTASs for guarding weakly-visible polygons and terrains. In: 17th International Workshop on Approximation and Online Algorithms (Munich 2019). Lecture Notes in Computer Science, vol. 11926. Springer, Cham (2019)
6. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Computational Geometry. Algorithms and Applications. Springer, Berlin (2008)
7. Brandstädt, A., Le, V.B., Spinrad, J.P.: Graph Classes: A Survey. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, Philadelphia (1999)
8. Choi, S.-H., Shin, S.Y., Chwa, K.-Y.: Characterizing and recognizing the visibility graph of a funnel-shaped polygon. *Algorithmica* **14**(1), 27–51 (1995)
9. Colley, P.: Recognizing visibility graphs of unimonotone polygons. In: 4th Canadian Conference on Computational Geometry (St. John's 1992), pp. 29–34. Memorial University of Newfoundland, St. John's (1992)
10. Colley, P., Lubiw, A., Spinrad, J.: Visibility graphs of towers. *Comput. Geom.* **7**(3), 161–172 (1997)
11. Donner, R.V., Donges, J.F.: Visibility graph analysis of geophysical time series: potentials and possible pitfalls. *Acta Geophys.* **60**(3), 589–623 (2012)
12. Elsner, J.B., Jagger, T.H., Fogarty, E.A.: Visibility network of United States hurricanes. *Geophys. Res. Lett.* **36**(16), # L16702 (2009)
13. Evans, W., Saeedi, N.: On characterizing terrain visibility graphs. *J. Comput. Geom.* **6**(1), 108–141 (2015)
14. Froese, V., Renken, M.: Advancing through terrains (2019). [arXiv:1904.08746](https://arxiv.org/abs/1904.08746)
15. Gao, Z.-K., Cai, Q., Yang, Y.-X., Dang, W.-D., Zhang, S.-S.: Multiscale limited penetrable horizontal visibility graph for analyzing nonlinear time series. *Sci. Rep.* **6**, # 35622 (2016)
16. Ghosh, S.K.: Computing the visibility polygon from a convex set and related problems. *J. Algorithms* **12**(1), 75–95 (1991)
17. Ghosh, S.K.: Visibility Algorithms in the Plane. Cambridge University Press, New York (2007)
18. Ghosh, S.K., Goswami, P.P.: Unsolved problems in visibility graphs of points, segments, and polygons. *ACM Comput. Surv.* **46**(2), # 21 (2013)
19. Guibas, L., Hershberger, J., Leven, D., Sharir, M., Tarjan, R.E.: Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica* **2**, 209–233 (1987)
20. Gutin, G., Mansour, T., Severini, S.: A characterization of horizontal visibility graphs and combinatorics on words. *Physica A* **390**(12), 2421–2428 (2011)
21. Hershberger, J.: Finding the visibility graph of a simple polygon in time proportional to its size. In: 3rd Annual Symposium on Computational Geometry, pp. 11–20. ACM, New York (1987)
22. Hershberger, J., Snoeyink, J.: Computing minimum length paths of a given homotopy class. *Comput. Geom.* **4**(2), 63–97 (1994)
23. Lacasa, L., Luque, B., Ballesteros, F., Luque, J., Nuno, J.C.: From time series to complex networks: the visibility graph. *Proc. Nat. Acad. Sci. USA* **105**(13), 4972–4975 (2008)
24. Liu, C., Zhou, W.-X., Yuan, W.-K.: Statistical properties of visibility graph of energy dissipation rates in three-dimensional fully developed turbulence. *Physica A* **389**(13), 2675–2681 (2010)
25. Luque, B., Lacasa, L.: Canonical horizontal visibility graphs are uniquely determined by their degree sequence. *Eur. Phys. J. Special Top.* **226**(3), 383–389 (2017)
26. Luque, B., Lacasa, L., Ballesteros, F., Luque, J.: Horizontal visibility graphs: exact results for random time series. *Phys. Rev. E* **80**(4), # 046103 (2009)
27. Shapiro, L.G., Haralick, R.M.: Decomposition of two-dimensional shapes by graph-theoretic clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1**(1), 10–20 (1979)
28. Stephen, M., Gu, C., Yang, H.: Visibility graph based time series analysis. *PLoS ONE* **10**(11), # e0143015 (2015)
29. Suri, S.: A linear time algorithm for minimum link paths inside a simple polygon. *Comput. Vis. Graph. Image Process.* **35**(1), 99–110 (1986)