Pseudo-Triangle Visibility Graph: Characterization and Reconstruction

Sahar Mehrpour Alireza Zarei*

Department of Mathematical Sciences
Sharif University of Technology
Tehran, IRAN P.O. Box: 11155-9415
S. Mehrpor: shr_mehrpour@yahoo.com
A. Zarei: zarei@sharif.ir +989123705477

ABSTRACT

The visibility graph of a simple polygon represents visibility relations between its vertices. Knowing the correct order of the vertices around the boundary of a polygon and its visibility graph, it is an open problem to locate the vertices in a plane such that it will be consistent with this visibility graph. This problem has been solved for special cases when we know that the target is a *tower*, a *spiral*, or an *anchor* polygon. Knowing that a given visibility graph belongs to a simple polygon with at most three concave chains on its boundary, a *pseudo-triangle*, we propose a linear-time algorithm for reconstructing one of its corresponding polygons. Moreover, we introduce a set of necessary and sufficient properties for characterizing visibility graphs of pseudo-triangles and propose polynomial algorithms for checking these properties.

Keywords: Computational geometry; Visibility graph; Characterizing visibility graph; Polygon reconstruction; Pseudo-triangle.

1. Introduction

The *visibility graph* of a simple polygon $\mathcal{P}$ is a graph $\mathcal{G}(V, E)$ where V is the vertices of $\mathcal{P}$ and an edge (u, v) exists in E if and only if the line segment uv lies completely inside $\mathcal{P}$, i. e. they are visible from each other. Based on this definition, each pair of adjacent vertices on the polygon boundary is assumed to be visible from each other. This implies that we always have a Hamiltonian cycle in a visibility graph that follows the order of vertices on the boundary of the corresponding polygon.

Computing the visibility graph of a given simple polygon has many applications in computer graphics, computational geometry, and robotics. There are several efficient polynomial-time algorithms for this problem. Asano *et al.* [1] and Welzl [2] proposed $\mathcal{O}(n^2)$-time algorithms for computing the visibility graph of a simple polygon of n vertices. This was then improved to $\mathcal{O}(m + n \log\log n)$ by Hershberger [3], where m is the number of edges in the visibility graph. The term $n \log\log n$ is due to the time required for triangulating a simple polygon. Using the $\mathcal{O}(n)$ time triangulation algorithm of Chazelle [4] reduces the running-time of Hershberger's result to $\mathcal{O}(m + n)$ which is optimal. There are numerous recent results considering properties of the visibility graph [5, 6], visibility graphs of special polygons [7] and visibility graph applications [8].

This concept has been studied in reverse as well: Characterizing a visibility graph is to determine whether a given graph is isomorphic to the visibility graph of some simple polygon, and the reconstruction problem is to build such a simple polygon. Everett showed that these problems are in PSPACE [9], and this is the only result known about the complexity of these problems. Although the problem has not been settled yet for general polygons, it has been solved recently for some variants. Ameer *et al.* proposed a polynomial-time recognition and reconstruction algorithm for pseudo-polygons [10] improving the older related results [11, 12, 13, 14], Casel *et al.* showed that the problem is NP-hard for unit square visibility graphs [15], and Boomari *et al.* proved that the problem is ∃R-complete on 3D terrains [16]. The class ∃R consists of problems that can be reduced in polynomial-time to the problem of deciding whether a system of polynomial equations with integer coefficients and any number of real variables has a solution. It can be easily seen that $NP \subseteq \exists\mathrm{R}$. This class has been studied earlier by other communities and recently, several long-standing open computational geometry problems were proved to be complete for this class [17, 18, 19].

For simple polygons, the recognition and reconstruction problems have been solved only for special cases of *spiral*, *tower*, and *anchor* polygons. These results are obtained by Everett and Corneil [20] for spiral polygons, by Colley *et al.* [21] for tower polygons, and by Boomari and Zarei [22] for anchor polygons. In a spiral polygon, there is at most one concave chain (Fig. 1a), the boundary of a tower polygon is composed of two concave chains and a single edge (Fig. 1b), and an anchor polygon is a tower polygon whose base edge is a convex chain.

Although there is a bit of progress on this type of reconstruction problem, there have been plenty of studies on characterizing visibility graphs [23, 21, 24, 25, 20, 26]. In 1988, Ghosh introduced three necessary conditions for visibility graphs and conjectured their sufficiency [26]. In 1990, Everett proposed a counter-example graph disproving Ghosh's conjecture [9]. She also refined Ghosh's third necessary condition to a new stronger condition [27]. In 1992, Abello *et al.* built a graph satisfying Ghosh's conditions and the stronger version of the third condition which was not the visibility graph of any simple polygon, disproving the sufficiency of these conditions [28]. In 1997, Ghosh added his fourth necessary condition and conjectured that this condition along with his first two conditions and the stronger version of the third condition are sufficient for a graph to be a visibility graph. This was also disproved by a counter-example from Streinu in 2005 [14].

In this paper, we solve the reconstruction problem for pseudo-triangles. A *pseudo-triangle* is a simple polygon whose boundary is composed of three concave chains, called *side-chains*, where each pair shares one convex vertex (called a corner). Let $\mathcal{P}$ be a pseudo-triangle formed by the concave side-chains $AB = [A, \ldots, B]$, $AC = [A, \ldots, C]$, and $BC = [B, \ldots, C]$ where A, B, and C are the corners (Fig. 1c).

According to this notation, a concave side-chain joining corner vertices X and Y is denoted by XY where X and Y are in $\{A, B, C\}$.

Let $\mathcal{H} = < A, \ldots, C, \ldots, B, \ldots, A >$ be the Hamiltonian cycle of the visibility graph of $\mathcal{P}$ which indicates the order of vertices on the boundary of $\mathcal{P}$. Here, we use the same notation for a vertex on the boundary of $\mathcal{P}$ and its corresponding vertex in the visibility graph and $\mathcal{H}$. For a given pair of Hamiltonian cycle $\mathcal{H}$ and visibility graph $\mathcal{G}(V, E)$, we introduce a set of necessary properties on $\mathcal{H}$ and $\mathcal{G}$ when this pair belongs to a pseudo-triangle and prove that these properties are sufficient as well.

Having these properties, we propose a linear-time algorithm for reconstructing a pseudo-triangle $\mathcal{P} =< A, \ldots, C, \ldots, B, \ldots, A >$ with $\mathcal{G}(V, E)$ as its visibility graph. Moreover, we propose algorithms for verifying the properties on a given pair of $\mathcal{H}$ and $\mathcal{G}$. These characterizing algorithms run in linear time in terms of the size of $\mathcal{G}$. Therefore, in this paper, we solve the characterizing and reconstructing problems for another class of polygons called pseudo-triangles.

Since a tower polygon is a special case of a pseudo-triangle, we use the tower reconstruction algorithm [21] as a sub-routine in our algorithm to build the initial part of the polygon.

Our motivation in solving this problem for pseudo-triangles is that every polygon can be partitioned into pseudo-triangles. Then, an idea for solving a general reconstruction problem is to handle these steps:

- Recognize a pseudo-triangle decomposition for the target polygon from $\mathcal{G}(V, E)$ and $\mathcal{H}$.
- Reconstruct each pseudo-triangle separately.
- Attach the reconstructed pseudo-triangles satisfying the pseudo-triangle decomposition and the visibility constraints.

In Section 2, we briefly describe the tower reconstruction algorithm [21] for reconstructing tower polygons which is used as a sub-routine in our algorithm. In Section 3, we introduce a set of necessary conditions (properties) of the visibility graph of pseudo-triangles and in Section 4, we prove the sufficiency of these conditions by proposing a reconstruction algorithm. Finally, we analyze the running time of the reconstruction algorithm and the algorithms required to check the properties.

2. Reconstructing Tower Polygons

Let $\mathcal{G}(V, E)$ be a bipartite graph with partitions U and W, and $<_U$ and $<_W$ are two orderings on respectively (*resp.*) vertices of U and W. The pair $(<_U, <_W)$ is a *strong ordering* on this graph if having $u <_U u'$ and $w <_W w'$ implies that if the edges (u, w') and (u', w) exist in E, then the edges (u, w) and (u', w') also exist in E.

The following theorem by Colley *et al.* [21] indicates the main property of the visibility graph of a tower polygon and guarantees the existence of a tower polygon consistent with such a visibility graph.

Theorem 1. *[21] Removing the edges of the reflex chains from the visibility graph of a tower gives an isolated vertex plus a connected bipartite graph for which the ordering of the vertices in the partitions provides a strong ordering. Conversely, any connected bipartite graph with strong ordering belongs to a tower polygon. Furthermore, such a tower can be constructed in linear time in terms of the number of vertices.*

The outline of the reconstruction algorithm proposed by Colley *et al.* [21] is as follows. As input, it takes the corner vertex $A = u_0 = v_0$ and a connected bipartite graph $\mathcal{G}(V, E)$ with vertices partitioned into two independent sets $AB = \{u_1, \ldots, u_m\}$ and $AC = \{v_1, \ldots, v_n\}$ having strong ordering.

In the first step, the position of the corner A and the vertices u_1 and v_1 are determined as in Fig. 2. In a middle step, suppose that the positions of the vertices $u_0, \ldots, u_{j-1}$ and $v_0, \ldots, v_{k-1}$ have been determined and the directions of the half-lines from u_{j-1} and v_{k-1} which respectively contain u_j and v_k, where $(u_j, v_k) \in E$, are also known. To complete such a middle step, the position of u_j and the half-line from u_j which contains u_{j+1} (where u_{j+1} is visible from v_k) must be determined. For this purpose, u_j is located somewhere on its containing half-line horizontally below the vertex v_c which v_c has the minimum index among vertices of AC which are visible from u_{j+1}. Consider s_{j+1} to be a point on $v_{c-1}v_c$ with an ϵ distance below s_j, when s_j lies on $v_{c-1}v_c$. Then, the containing half-line of u_{j+1} will be the half-line on the supporting line of u_j and s_{j+1} downward from u_j. If s_j does not lie on $v_{c-1}v_c$, then s_{j+1} is a point on $v_{c-1}v_c$ with an ϵ distance below v_{c-1}.

According to this construction, s_j will be the intersection of chain AC and the supporting line of u_j and u_{j-1}. Similarly, r_j will be the intersection of AB and the supporting line of v_j and v_{j-1} (Fig. 2). We say *"will"* because we first fix the position of s_j (resp. r_j) from which the position of vertex u_j (resp. v_j) is determined. We will use this notation once again in Section 4.

3. Properties of Pseudo-Triangle Visibility Graphs

In this section, we describe a set of properties that a pair of $\mathcal{H}$ and $\mathcal{G}$ must have to be the Hamiltonian cycle and visibility graph of a pseudo-triangle.

Any sub-sequence $< v_i, \ldots, v_j >$ on the Hamiltonian cycle is called a chain and is denoted by $[v_i, \ldots, v_j]$. A vertex v_a on a chain $[v_i, \ldots, v_j]$ is a *blocking vertex* for the invisible pair (v_i, v_j) if there is no visible pair of vertices v_l in $[v_i, \ldots, v_{a-1}]$ and v_k in $[v_{a+1}, \ldots, v_j]$. Ghosh showed that for every invisible pair of vertices (u, v) in a visibility graph, there is at least one blocking vertex in $[u, \ldots, v]$ or $[v, \ldots, u]$. Furthermore, every vertex on the shortest Euclidean path between u and v(inside the corresponding polygon) is a blocking vertex for this pair [26]. Note that in a pseudo-triangle the shortest Euclidean path between two invisible vertices turns in only one direction (i.e. clockwise or counterclockwise).

Let AB, AC, and BC be the side-chains of a pseudo-triangle. The order of vertices in these chains is defined with respect to one of their corner vertices. For a vertex u in chain XY, $Ind^X(u)$ is equal to the number of vertices in chain $[X, \ldots, u]$ minus one. According to this definition, $Ind^A(A)$ is zero and $Ind^B(A)$ is $k-1$ where k is the number of vertices in chain AB. Then, based on a given vertex indexing we refer to the previous and next vertices of a given vertex on a side-chain. For a vertex v in chain XY with $Ind^X(v) = i$, we use $N^X(v, j)$ to refer to the vertex $u \in XY$ with $Ind^X(u) = i + j$. Similarly, $P^X(v, j)$ is the vertex $u \in XY$ with $Ind^X(u) = i - j$. For the sake of brevity, we use $N^X(v)$ instead of $N^X(v, 1)$, and $P^X(v)$ instead of $P^X(v, 1)$. Note that in this notation, j can be a positive or negative natural number. For corner vertices that belong to two side-chains, we use N or P notation only when the target chain is known from the context. For a vertex u, $FV^X(u, XY)$ is a vertex on chain XY with the minimum index that is visible from u when the indices start from corner vertex X. Similarly, $LV^X(u, XY)$ is a vertex on chain XY with the maximum index that is visible from u when the indices start from corner vertex X. We have used FV and LV respectively as abbreviations for *first visible* and *last visible*. Fig. 3 depicts this notation.

Lemma 1. *It is always possible to identify at least two corners of a pseudo-triangle $\mathcal{P}$ from its corresponding Hamiltonian cycle and visibility graph.*

Proof. Since a corner is a convex vertex, it cannot be a blocking vertex for its neighbors. On the other hand, every concave vertex blocks visibility of its neighbors. Therefore, in the Hamiltonian cycle of a pseudo-triangle, there are at most three vertices whose adjacent vertices are visible pairs. By traversing the Hamiltonian cycle, these visible pairs, and the corresponding corners, can be identified.

Suppose that this method does not identify all three corners. Without loss of generality, assume that A is an unidentified corner and its adjacent vertices on chains AB and AC are respectively u and v. This means that u and v do not see each other and there must be a blocking vertex for this invisible pair. Due to their concavity, this blocking vertex cannot belong to the chains AB and AC. Consider the shortest Euclidean path between u and v inside the pseudo-triangle(Fig. 4). It is clear that this path is composed of a subchain of BC, say $[w, \ldots, w']$, and two edges (u, w) and (w', v) where $Ind^B(w') \geq Ind^B(w)$ and both edges (u, w) and (w', v) belong to the visibility graph. The polygon formed by $< u, \ldots, B, \ldots, w >$ is a tower polygon with base (u, w) and corner B. The corner of this tower is the isolated vertex obtained by removing the edges of its Hamiltonian cycle from its visibility graph. Therefore, the corner vertex B is detectable. The same argument holds for the tower polygon formed by $< w', \ldots, C, \ldots, v >$ from which the corner C can be identified. This means that if A cannot be identified from the visibility graph, the other two corners will be detectable. □

Consider a pseudo-triangle $\mathcal{P}$ with side-chains AB, AC, and BC, and $\mathcal{G}$ and $\mathcal{H}$ as its visibility graph and Hamiltonian cycle, respectively. Assume that the method described in Lemma 1, identifies only two corners of $\mathcal{P}$. Without loss of generality, assume that A is the unidentified vertex. This means that there is a subchain on BC which blocks the visibility of adjacent vertices of A on chains AB and AC. Then, there is no visibility edge between a vertex from AB and a vertex of AC. By removing the edges of the Hamiltonian cycle from the visibility graph, two isolated vertices B and C and a connected bipartite graph, with parts S and T, is obtained where S consists of vertices of chains AB and AC except the isolated vertices B and C and T consists of vertices of BC except B and C. By adding the isolated vertex B to T, and the boundary edge e to this bipartite graph that connects B to its adjacent vertex on AB, we will have a single isolated vertex C and a bipartite graph with strong ordering. Then, according to Theorem 1 this bipartite graph corresponds to a tower polygon with base edge e and $\mathcal{G}$ and $\mathcal{H}$ as its visibility graph and Hamiltonian cycle, respectively. Fig. 5 shows how such a pseudo-triangle can be interpreted as a tower polygon.

Therefore, we have the following property about the pair of $\mathcal{H}$ and $\mathcal{G}$ of a pseudo-triangle.

Property 1. If $\mathcal{H}$ and $\mathcal{G}$ are respectively the Hamiltonian cycle and visibility graph of a pseudo-triangle $\mathcal{P}$, at least two corners of $\mathcal{P}$ can be identified. Furthermore, if only two corners are detectable, the given $\mathcal{H}$ and $\mathcal{G}$ belong to a pseudo-triangle if and only if there is a tower polygon with $\mathcal{H}$ and $\mathcal{G}$ as its Hamiltonian cycle and visibility graph, respectively.

From this property, we assume for the remainder of this section that the method described in the proof of Lemma 1 identifies all three corners. Otherwise, we can use the tower polygon algorithm to decide whether the given pair of $\mathcal{H}$ and $\mathcal{G}$ belong to a tower polygon(which is a special case pseudo-triangle) and obtain the answer.

An *interval* of a chain with endpoints p and q is the set of points on this chain connecting p to q. Note that in this definition, the endpoints of an interval are not necessarily vertices of the chain. For example, for points p on edge (u_i, u_{i+1}) and q on edge (u_j, u_{j+1}) of a chain AB where $i < j$, the interval defined by p and q is the chain $[p, u_{i+1}, \ldots, u_j, q]$.

Property 2. Every non-corner vertex of a side-chain sees a single non-empty interval from any one of the other side-chains.

Proof. The inner angle of such a vertex is more than π and its inner visibility region cannot be bounded by a single concave chain. Therefore, it will see some parts from any of the other side-chains. The continuity of these visible parts on each side-chain is proved by contradiction. Assume that a vertex $u \in AB$ sees two disjoint intervals $[v_i, \ldots, v_j]$ and $[v_k, \ldots, v_l]$ from AC meaning that the interval $(v_j, \ldots, v_k)$ is not visible from u. Consider an invisible point v' in $(v_j, \ldots, v_k)$. There must be a blocking vertex for the invisible pair (u, v'). This blocking vertex must lie on the third side-chain which will also block either the visibility of u and v_j or u and v_k. □

Property 3. (Fig. 6(a)) For any pair of side-chains AB and AC and a pair of vertices $\{u, v\}$ where $u \in AB$, $v \in AC$, $v \neq A$, and $u = FV^A(v, AB)$, we have $(P^A(v), u) \in E$. In other words, the closest vertex to A on AB which is visible to a vertex $v \in AC$, for $v \neq A$, is also visible from $P^A(v)$.

Proof. Consider the subpolygon $< u, P^A(u), \dots, A, \dots, P^A(v), v >$. If we triangulate this polygon, there is no internal diagonal connected to v which means that$< u, v, P^A(v) >$ must be a triangle in any triangulation. Therefore, the edge $(u, P^A(v))$ is diagonal and this edge must exist in the visibility graph. $\square$

Corollary 1. *(Fig. 6(b)) For any pair of side-chains AB and AC and a vertex $v \in AC$ where $v \neq C$, if $FV^A(v, AB) = u_j$ and $FV^A(N^A(v), AB) = u_k$, then $Ind^A(u_k) \geq Ind^A(u_j)$.*

Corollary 2. *For any pair of side-chains AB and AC and a vertex $v \in AC$ where $v \neq C$, if v does not see any vertex from AB, then $N^A(v)$ does not see any vertex of AB as well.*

Corollary 3. *(Fig. 7(a)) For any pair of side-chains AB and AC and a pair of vertices (u, v) where $u \in AB$ and $v \in AC$ and $k, l > 0$, if both $(P^A(u, k), v)$ and $(u, P^A(v, l))$ exist in E, then $(P^A(u, k), P^A(v, l)) \in E$.*

Proof. Let u' and v' denote $P^A(u, k)$ and $P^A(v, l)$, respectively. Trivially, $Ind^A(u') \geq Ind^A(FV^A(v, AB))$. Applying Corollary 1 iteratively on chain $[v', \dots, v]$ implies that $Ind^A(FV^A(v, AB)) \geq Ind^A(FV^A(v', AB))$. This means that u' lies between two vertices $FV^A(v', AB)$ and u which are both visible from v'. Then, Property 2 implies that u' is also visible from v'. $\square$

Corollary 4. *(Fig. 7(b)) For any pair of side-chains AB and AC and a pair of vertices (u, v) where $u \in AB$ and $v \in AC$, if both $(N^A(u, k), N^A(v, l))$ and (u, v) exist in E where $l, k > 0$, then at least one of the edges $(N^A(u), v)$ or $(u, N^A(v))$ exists in E.*

Proof. We prove this by induction on $k+l$. For the induction base step, assume that $k = l = 1$. If v is not visible from $N^A(u)$, $N^A(v)$ must be equal to $FV^A(N^A(u), AB)$. Then, Property 3 implies that u sees $N^A(v)$.

For the inductive step, assume that the corollary holds for all $k + l < n$ where $n > 2$. Let u' and v' denote $N^A(u, k)$ and $N^A(v, l)$, respectively. If $Ind^A(FV^A(u', AB)) \geq Ind^A(v)$, v sees both vertices u and u' which according to Property 2 sees $N^A(u)$ as well. Otherwise, according to Property 3, $FV^A(u', AB)$ is visible from $P^A(u')$. If $P^A(u') = u$, then u sees v and $FV^A(u', AB)$ which is farther from A than v and means that u and $N^A(v)$ see each other. Finally, when $P^A(u') \neq u$ we obtain a smaller version of the problem with parameters $k - 1$ and l which holds by induction. $\square$

Corollary 5. *(Fig. 7(c)) For any pair of side-chains AB and AC and a pair of vertices $u \in AB$ and $v \in AC$, where $(u, v) \in E$ and none of the edges $(N^A(u), v)$ and $(u, N^A(v))$ exist in E, all visible vertices of AC from $N^A(u, k)$ are also visible from $N^A(u, k-1)$ (for any $k > 0$). This implies that $LV^A(N^A(u, k), AC)$ must lie above v.*

Proof. Any visible vertex v' must belong to $[A, \dots, P^A(v)]$. Otherwise, according to Corollary 4 either $(N^A(u), v)$ or $(u, N^A(v))$ must exist. According to Corollary 1, $FV^A(u, AC)$ is closer to A than $FV^A(N^A(u, k), AC)$, and because of the continuity of the chain that is visible from u (Property 2) , v' will be visible from u. This implies that v' is visible from all vertices of the chain $[u, \dots, N^A(u, k)]$. □

For each pair of vertices $u \in AB$ and $v \in AC$, the diagonal edge (u, v) in the visibility graph of a pseudo-triangle specifies a tower formed by the boundary vertices $< u, \dots, A, \dots, v >$. The vertices of this tower satisfy the strong ordering defined earlier. This strong ordering can be derived from Property 2 and corollaries 3 and 4. Therefore, we do not specify this as a new property.

Property 4. For any pair of side-chains AB and AC and a pair of vertices $u \in AB$ and $v \in AC$, where $(u, v) \in E$ and none of the edges $(N^A(u), v)$ and $(u, N^A(v))$ exist in E,

a. (Fig. 8(a)) there is a non-empty subchain of the third side-chain BC which is visible from both u and v.
b. (Fig. 8(b)) let $[w, \dots, w']$ be the maximum subchain of BC visible to both u and v where $w' = N^B(w, l)$, $l \geq 0$. Then, w' is not closer to B than $LV^B(N^A(u), BC)$, or formally, $Ind^B(w') \geq Ind^B(LV^B(N^A(u), BC))$.
c. (Fig. 8(b)) $FV^B(N^A(v), BC)$ is not closer to B than $LV^B(N^A(u), BC)$, or formally, $Ind^B(FV^B(N^A(v), BC)) \geq Ind^B(LV^B(N^A(u), BC))$.

Proof. (a) Triangulating $\mathcal{P}$ using the edge (u, v), the adjacent triangle of this edge on the opposite side of A must have its third vertex on BC. This is due to the invisibility of $(N^A(u), v)$ and $(u, N^A(v))$ pairs. Therefore, this chain contains at least one vertex. From Property 2 we know that the visible part of BC from any one of vertices u and v is continuous and the intersection of these parts will be continuous as well.

(b) From (a) we know that the subchain $[w, \dots, w']$ is non-empty. For the sake of contradiction, assume that $w'' = LV^B(N^A(u), BC)$ is farther from B than w'. Then, the segments $(w'', N^A(u))$ and (w', v) intersect each other inside the pseudo-triangle. Let p be this intersection point. The sub-polygon formed by the boundary vertices $< u, N^A(u), p, v >$ must be a convex polygon that completely lies inside the pseudo-triangle. Otherwise, w' will prevent $N^A(u)$ and w'' from seeing each other. So, the diagonal edge $(N^A(u), v)$ must exist in E which is a contradiction.

(c) Let v' be $N^A(v)$ and u' be $N^A(u)$. For the sake of contradiction, assume that $FV^B(v', BC)$ is closer to B than $LV^B(u', BC)$. Then, the edges $(v', FV^B(v', BC))$ and $(u', LV^B(u', BC))$ intersect within the pseudo-triangle. Let p be this intersection point. The sub-polygon formed by the boundary vertices $< u, v, v', p, u' >$ must be a convex polygon that completely lies inside the pseudo-triangle. Otherwise, $FV^B(v', BC)$ will prevent u' and $LV^B(u', BC)$ from seeing each other. So, all diagonal edges (u, v'), (u', v), and (u', v') must exist in E which is a contradiction. □

Corollary 6. *For any side-chain BC, there exists at least one vertex $w \in BC$ that sees some vertices from both of the other side-chains. Furthermore, every vertex $P^B(w, k)$ where $k > 0$, sees at least one vertex from AB.*

Proof. If there is a pair of vertices $u \in AB$ and $v \in AC$ satisfying Property 4(a), the first part holds for the vertices of the subchain of the third side-chain BC which is visible from both u and v. If there is no such a pair of vertices, without loss of generality assume that B sees some vertices of AC and $v = LV^A(B, AC)$. Trivially, the adjacent vertex of B on side-chain BC sees both $B \in AB$ and $v \in AC$. This can be obtained directly from Property 4(a) by imaginary cloning B as two separate vertices on AB and AC. and adding new corner vertex B as a point on the supporting line of B and v in the opposite side of v.

Having a vertex satisfying the first part, the second part follows from Property 3. □

Property 5. (Fig. 9) For any side-chain BC and a vertex $w \in BC$ with distinct vertices $u = FV^A(w, AB)$ and $v = FV^A(w, AC)$, the vertices u and v are visible from each other.

Proof. Let $\mathcal{P}'$ be the subpolygon with $< A, \ldots, u, w, v, \ldots, A >$ as its boundary vertices. The vertex w does not see any other vertex of $\mathcal{P}'$ which means that the diagonal uv must be used to triangulate $\mathcal{P}'$. This means that u and v must be visible from each other. □

Property 6. (Fig. 10) For any side-chain BC, let u and v be respectively the closest vertices on AB and AC to A which are visible from some vertex (not necessarily the same) of BC. Then, there exists a non-empty subchain $[w, \ldots, w']$ in BC , $w' = N^B(w, l)$ and $l \geq 0$, that either all vertices of this subchain are visible from both u and v, or, (w, w') is an edge of BC and w sees v and w' sees u.

Proof. It is simple to show that $(u, v) \in E$. Assume that no vertex on BC sees both vertices u and v. Then, we first show that there is a pair of vertices w and $w' = N^B(w, l)$ where w sees v and w' sees u. Let w be $FV^C(v, BC)$ and w' be $FV^B(u, BC)$. Trivially, $w \neq w'$ and w is closer to B than w' (otherwise, u and v will be visible to both w and w'). To complete the proof, it is enough to show that $w' = N^B(w)$. This is done by showing that any vertex w'' between w and w' on

BC must see at least one of the vertices u and v which contradicts the definition of w and w'.

Assume that there is a vertex w'' between w and w' and it sees neither u nor v. In the tower polygon formed by boundary $< u, \ldots, B, \ldots, w'', w' >$, the blocking vertex for the invisible pair (w'', u) must lie on AB. Similarly, in the tower formed by boundary $< w, w'', \ldots, C, \ldots, v >$, the blocking vertex for the invisible pair (w'', v) must lie on AC. Therefore, at least one of the side-chains AB and AC must be convex which is a contradiction. So, w'' must see at least one of the vertices u and v. □

Corollary 7. *(Fig. 11) If w and w' satisfy the conditions of Property 6, then for $k > 0$:*

- $u_i = FV^A(P^B(w', k), AB)$ *is not closer to A than* $u_j = FV^A(P^(w', k-1), AB)$.
- *If there are vertices* $v_i = FV^A(P^B(w, k), AC)$ *and* $v_j = FV^A(P^B(w, k-1), AC)$*, then v_i is not closer to A than v_j.*

These mean that as we move from w' to w the topmost visible vertices of AB and AC go down along these chains.

Proof. For the sake of contradiction, assume that u_i is closer to A than u_j. The diagonal edge $(w', FV^A(w', AB))$ along with vertices $< w', \ldots, B, \ldots, FV^A(w', AB) >$ form a tower polygon which contains the vertices u_i and u_j, and satisfies strong ordering. When both edges $(P^B(w', k), u_i)$ and $(P^B(w', k-1), u_j)$ exist in the visibility graph, the edge $(P^B(w', k-1), u_i)$ must also exist in E.

We prove the second part by contradiction. Let $P^B(w, l)$ be the closest vertex of BC to B which sees at least one vertex from AC ($l \geq 0$). For $l \geq k > 0$, assume that v_i is closer to A than v_j. Since $FV^A(w, AC)$ is not farther from A than v_i, Corollary 3 implies that v_i sees w. According to Property 2, v_i is also visible from $P^B(w, k-1)$ which is a contradiction. □

Property 7. Let $[w_i, \ldots, w_j]$ be the subchain of BC satisfying Property 6 and for any vertex $w \in BC$, $u = FV^A(w, AB)$ and $v = FV^A(w, AC)$ are the closest vertices to A which are visible to w. Then:

a If $w \in [w_i, \ldots, w_j]$, then at least one of the pairs $(N^A(u), P^A(v))$ and $(P^A(u), N^A(v))$ are invisible.

b If $w \in [B, \ldots, w_j]$ and $(N^A(u), P^A(v))$ are invisible vertices, then this happens for all vertices in $[B, \ldots, w]$ and $LV^A(w, AC)$ is not farther from A than $LV^A(N^B(w), AC)$. This is symmetrically true when $w \in [w_i, \ldots, C]$ and $(P^A(u), N^A(v))$ are invisible vertices.

c If $w \neq B$ is closer to B than w_i, then $(N^A(u), P^A(v))$ is an invisible pair. Symmetrically, $(P^A(u), N^A(v))$ are invisible vertices when $w \neq C$ is closer to C than w_j.

Proof. (a) Consider the subpolygon $\mathcal{P}'$ with boundary $< w, v, \ldots, A, \ldots, u >$. The pairs $(w, P^A(v))$ and $(w, P^A(u))$ are invisible. These pairs share the same blocking vertex. If u is the blocking vertex, then $(N^A(u), P^A(v))$ is an invisible pair, and if v is the blocking vertex, then the pair $(P^A(u), N^A(v))$ is invisible.

(b) Assume that $(N^A(u), P^A(v))$ are invisible from each other. This means that the visible vertices of AC from w are bounded from above by vertices of AB. This will happen for all vertices in $[B, \ldots, w]$ as well. A similar argument holds when $(P^A(u), N^A(v))$ is the invisible pair.

(c) It is clear that at least one of the vertices $FV^A(w_i, AB)$ and $FV^A(w_i, AC)$ is farther from A than u and v. For the sake of contradiction, assume that $(N^A(u), P^A(v))$ is a visible pair. Then, in subpolygon $\mathcal{P}' =< w, v, \ldots, A, \ldots, u >$, v must be the blocking vertex for the pairs $(w, P^A(v))$ and $(w, P^A(u))$. This vertex also blocks the pairs $(N^B(w), P^A(u, i))$ and $(N^B(w, l), P^A(v, j))$. But, for some $l > 0$ and i and $j \geq 0$, $N^B(w, l) = w'$, $P^A(u, i) = FV^A(w', AB)$, and $P^A(v, j) = FV^A(w_i, AC)$ which contradicts the definition of w_i. □

As mentioned earlier, Ghosh introduced four necessary conditions for a visibility graph of a simple polygon. It is simple to show that these conditions are derived from the properties described in this section which means that these properties include Ghosh's conditions.

4. Pseudo-Triangle Reconstruction

In this section, $\mathcal{G}(V, E)$ denotes the visibility graph of a pseudo-triangle $\mathcal{P}$ with AB, AC, and BC side-chains and the order of vertices on the boundary of $\mathcal{P}$ is specified by a Hamiltonian cycle $\mathcal{H} =< A, \ldots, C, \ldots, B, \ldots, A >$ in $\mathcal{G}$. We assume that the inputs $\mathcal{G}$ and $\mathcal{H}$ satisfy the properties 1 to 7. We propose an algorithm for reconstructing a pseudo-triangle corresponding to the given pair of $\mathcal{G}$ and $\mathcal{H}$.

In order to reconstruct the pseudo-triangle $\mathcal{P}$, we divide $\mathcal{P}$ into four subpolygons $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, and $\mathcal{Z}'$ as shown in Fig. 12 and reconstruct each one separately. For the sake of brevity, $u_i = N^A(A, i)$ on side-chain AB, $v_j = N^A(A, j)$ on side-chain AC, and $w_k = N^B(B, k)$ on side-chain BC where $i, j, k \geq 0$. We assume that AB and AC have respectively $\alpha + 1$ and $\delta + 1$ vertices.

The subpolygon $\mathcal{X}$ is formed by subchains $[A, \ldots, u_\nu]$ and $[A, \ldots, v_\mu]$ and edge (u_ν, v_μ) where $LV^A(u_\nu, AC) = v_\mu$ and $LV^A(v_\mu, AB) = u_\nu$. The vertices u_ν and v_μ are identified by walking alternatively on side-chains AB and AC from corner vertex A towards B and C. As a step of this trace, assume that we are at vertices u_i and v_j and want to go one step further on AB. If u_i is the last vertex on AB or v_j does not see u_{i+1} we fix u_i as u_ν. Otherwise, we go to u_{i+1} in this step. Walking on side-chain AC is done similarly. The sub-polygon $\mathcal{X}$ is a tower polygon with strong ordering in its visibility graph. Note that $u_{\nu+1}$ or $v_{\mu+1}$ exists only when the side-chain BC has more than one edge, otherwise, two identified adjacent corners u_ν and v_μ compose the base of a tower polygon which can be constructed by the tower reconstruction algorithm. So, we assume that BC has more than one edge.

The subpolygon $\mathcal{Y}$ is identified as follows: Let $[w_i, \ldots, w_j]$ be the maximum subchain of BC visible from both u_ν and v_μ. According to Property 4(a), this chain is non-empty and continuous. Let $LV^B(u_{\nu+1}, BC) = w_k$ and $FV^B(v_{\mu+1}, BC) = w_l$. From Property 4(b), $k \leq j$ and $l \geq i$ and from Property 4(c), $k \leq l$. We define M and N as $\max(k, i)$ and $\min(l, j)$, respectively. It is clear that chain $[w_M, \ldots, w_N]$ contains at least one vertex. Then, $\mathcal{Y}$ is defined to be the polygon with $< u_\nu, w_M, \ldots, w_N, v_\mu >$ as its boundary.

The subpolygon $\mathcal{Z}$ is formed by subchains $[u_\nu, \ldots, B)]$ and $[B, \ldots, w_M]$ and edge (u_ν, w_M). Similarly, subchains $[v_\mu, \ldots, C]$ and $[C, \ldots, w_N]$ and edge (v_μ, w_N) specify the subpolygon $\mathcal{Z}'$. It is clear that $\mathcal{P}$ is the union of $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, and $\mathcal{Z}'$.

Our reconstruction algorithm first builds $\mathcal{X}$ using the tower reconstruction algorithm in such a way that vertices of AB lie to the left of vertices of AC. Then, we extend this polygon to build $\mathcal{Y}$ (Section 4.1) and build and attach $\mathcal{Z}$ and $\mathcal{Z}'$ parts to this polygon (Section 4.2) to complete the construction procedure.

4.1. *Reconstructing* $\mathcal{Y}$

In this step, we build the subpolygon $\mathcal{Y} =< u_\nu, w_M, \ldots, w_N, v_\mu >$. We know the position of vertices u_ν and v_μ from the previous step, which are also on the boundary of $\mathcal{Y}$. To locate positions of other vertices, we show that there are non-empty regions in which these vertices can be placed.

For any vertex w_j from $\mathcal{Y}$ which $FV^A(w_j, AB) = u_i$ and $FV^A(w_j, AC) = v_l$, we define a region $\mathscr{W}^j_{i,l}$ from which each point sees all vertices in the subchains $[u_i, \ldots, u_\nu]$ and $[v_l, \ldots, v_\mu]$. Therefore, w_j can be placed in $\mathscr{W}^j_{i,l}$ satisfying the visibility constraints between w_i and vertices of $\mathcal{X}$. We use $\mathscr{W}^j$ instead of $\mathscr{W}^j_{i,l}$ whenever i and l indices are not important. The region $\mathscr{W}^j$ is determined as follows: Since w_j sees u_ν and v_μ, the vertices u_i and v_l always exist and are well-defined. If u_i and v_l are identical, then $i = l = 0$ and the region $\mathscr{W}^j = \mathscr{W}^j_{0,0}$ is defined to be the part of the cone formed by the lines through (A, u_1) and (A, v_1) restricted to the underneath of the line through points u_ν and v_μ. Trivially, each point of $\mathscr{W}^j$ sees all vertices $u_\nu, \ldots, A, \ldots, v_\mu$.

Let $\mathcal{F}_z(x, y)$ be the 'z' half-plane defined by the line through x and y where 'z' is 'b' (bottom), 'r' (right), or 'l' (left). If u_i and v_l are distinct vertices, according to Property 7, at least one of the pairs (u_{i+1}, v_{l-1}) and (u_{i-1}, v_{l+1}) do not see each other. The invisible pair is determined by applying Corollary 7 and Property 7.

Assume that (u_{i+1}, v_{l-1}) is the invisible pair. Then, $\mathscr{W}^j_{i,l}$ is defined to be $\mathcal{F}_r(s_{i+1}, u_i) \bigcap \mathcal{F}_r(v_l, u_i) \bigcap \mathcal{F}_l(u_{i-1}, u_i) \bigcap \mathcal{F}_l(v_{l-1}, u_i) \bigcap \mathcal{F}_b(v_\mu, u_\nu)$ (Fig. 13). As defined in Section 2, s_{i+1}, u_i and u_{i+1} are collinear. We used $\mathcal{F}_r(s_{i+1}, u_i)$ instead of $\mathcal{F}_r(u_{i+1}, u_i)$ here because at least for $i = \nu$ we do not know the position of u_{i+1} yet.

Any one of these half-planes forces some visibility constraints for w_j. $\mathcal{F}_b(v_\mu, u_\nu)$ implies that w_j sees both $u_n u$ and v_μ; $\mathcal{F}_r(s_{i+1}, u_i)$ implies that w_j sees all vertices $< u_i, \ldots, u_\nu >$; $\mathcal{F}_r(v_l, u_i)$ implies that w_j sees all vertices $< v_l, \ldots, v_\mu >$; $\mathcal{F}_l(u_{i-1}, u_i)$ prevents w_j from seeing vertices $< A, \ldots, u_{i-1} >$; and $\mathcal{F}_l(v_{l-1}, u_i)$ prevents w_j from seeing vertices $< A, \ldots, v_{l-1} >$. Therefore, all points in this region satisfy the visibility constraints from w_j to vertices $< u_\nu, \ldots, A, \ldots, v_\mu >$.

Concavity of AB and AC implies that intersections $\mathcal{F}_r(s_{i+1}, u_i) \bigcap \mathcal{F}_l(u_{i-1}, u_i)$ and $\mathcal{F}_r(v_l, u_i) \bigcap \mathcal{F}_l(v_{l-1}, u_i)$ are not empty. Therefore, $\mathscr{W}^j_{i,l}$ will be empty only when $\mathcal{F}_r(s_{i+1}, u_i) \bigcap \mathcal{F}_l(v_{l-1}, u_i)$ is empty or $\mathcal{F}_r(v_l, u_i) \bigcap \mathcal{F}_l(u_{i-1}, u_i)$ is empty. The first case is impossible, because otherwise, u_{i+1} must be visible from v_{l-1} which is in contradiction with invisibility assumption of (u_{i+1}, v_{l-1}). The second case is also impossible because then, the pair u_i and v_l must be invisible. But, according to Property 5, u_i and v_l must be visible from each other.

Therefore, the region $\mathcal{F}_r(s_{i+1}, u_i) \bigcap \mathcal{F}_r(v_l, u_i) \bigcap \mathcal{F}_l(u_{i-1}, u_i) \bigcap \mathcal{F}_l(v_{l-1}, u_i)$ is non-empty and some part of this intersection lies in half-plane $\mathcal{F}_b(v_\mu, u_\nu)$.

According to the above discussion, $\mathscr{W}^j$ is defined by $\mathcal{F}_b(v_\mu, u_\nu)$ and two half-planes of $\{\mathcal{F}_r(s_{i+1}, u_i), \mathcal{F}_r(v_l, u_i), \mathcal{F}_l(u_{i-1}, u_i), \mathcal{F}_l(v_{l-1}, u_i)\}$. The apex of $\mathscr{W}^j$ is defined to be the intersection of the corresponding lines of these two half-planes which is u_i.

The above discussions were for the assumption that (u_{i+1}, v_{l-1}) is the invisible pair. The description for the cases where (u_{i-1}, v_{l+1}) is the invisible pair is symmetric: $\mathscr{W}^j_{i,l}$ is $\mathcal{F}_l(r_{l+1}, v_l) \bigcap \mathcal{F}_l(v_l, u_i) \bigcap \mathcal{F}_r(v_{l-1}, v_l) \bigcap \mathcal{F}_r(u_{i-1}, v_l) \bigcap \mathcal{F}_b(v_\mu, u_\nu)$ and the apex of $\mathscr{W}^j$ will be v_l.

If the apex of $\mathscr{W}^j$ lies on AB, Property 7 implies that the apex of $\mathscr{W}^{j-1}$ will lie on AB as well. Furthermore, Corollary 7 implies that $\mathscr{W}^{j-1}$ is either completely coinciding $\mathscr{W}^j$ or is completely on its left. Similarly, if the apex of $\mathscr{W}^j$ lies on AC, then the apex of $\mathscr{W}^{j+1}$ lies on AC as well, and $\mathscr{W}^{j+1}$ is either coinciding $\mathscr{W}^j$ or is completely on its right.

Then, we can place the vertices $w_M, \ldots, w_N$ of $\mathcal{Y}$ on an arbitrary concave chain inside $\mathcal{F}_b(v_\mu, u_\nu)$ in such a way that $w_j \in \mathscr{W}^j$. This placement satisfies the visibility constraints for $\mathcal{X}$ and $\mathcal{Y}$. However, to guarantee the reconstruction of $\mathcal{Z}$ and $\mathcal{Z}'$, we define some constraints on this concave chain which is described in the rest of this section.

Let s'_i $(i > \nu)$ be the intersection of AC and the line through u_i and $LV^B(u_i, BC)$, r'_k $(k > \mu)$ be the intersection of AB and the line through v_k and $FV^B(v_k, BC)$, t'_j $(j < M)$ be the intersection of AC and the line through w_j and w_{j+1}, and t'_j $(j > N)$ be the intersection of AB and the line through w_j and w_{j-1} (see Fig. 14).

Note that although we have not yet determined positions of vertices defining s'_i, r'_k, and t'_j, we determine their containing edges from the visibility information as follows: for $i > \nu$, if u_i sees at least one vertex from AC, s_i lies on the segment connecting $P^A(FV^A(u_i, AC))$ and $FV^A(u_i, AC)$ and s'_i lies on the segment connecting

($LV^A(u_i, AC)$ and $N^A(LV^A(u_i, AC))$. On the other hand, if u_i sees no vertex from AC, then for $k \geq i$, both s_k and s'_k lie on the segment connecting $P^A(LV^A(u_j, AC))$ and $LV^A(u_j, AC)$ where u_j has the highest index among the vertices of AB that see at least one vertex from AC. Corollary 5 implies that all these points lie on boundary edges of $\mathcal{X}$, except when $i = \nu + 1$ and w_{M-1} is visible to both u_ν and v_μ, for which both s_k and s'_k for $k \geq i$ lie on $(v_\mu, v_{\mu+1})$. The same situation happens for r_l and r'_l when $l > \mu$.

The containing edge of t'_j for $j < M$ is determined as follows: If w_j sees at least one vertex from AC, then t'_j lies on the segment connecting $LV^A(w_j, AC)$ and $N^A(LV^A(w_j, AC))$, otherwise, it lies on the containing edge of s'_α (Note that according to our assumption at the beginning of Section 4, α and δ are respectively the greatest indices of vertices u_i and v_j on AB and AC side-chains.). Similarly, for $j > N$, if w_j sees at least one vertex from AB, then t'_j lies on the segment connecting $LV^A(w_j, AB)$ and $N^A(LV^A(w_j, AB))$, and otherwise, it lies on the containing edge of r'_δ. Property 7 implies that all these points lie on boundary edges of $\mathcal{X}$ or edges $(v_\mu, v_{\mu+1})$ and $(u_\nu, u_{\nu+1})$.

The containing edges of s'_α and r'_δ are respectively called *"the floating edge in AC"* and *"the floating edge in AB"*. We call these edges floating because we increase their length and reposition their underneath vertices to enforce the concavity in building $\mathcal{Z}$ and $\mathcal{Z}'$.

We define the vertices w_{M^*} and w_{N^*} as follows: If BC has two edges, then w_M and w_N are both equal to w_1 (the middle vertex of BC), and w_{M^*} and w_{N^*} are also defined to be w_1. When BC has more than two edges, M^* is defined to be M when the apex of $\mathscr{W}^M$ does not lie on a vertex of AC below its floating edge. Otherwise, M^* is defined to be j where j is the maximum index for which the apex of $\mathscr{W}^j$ lies above the floating edge of AC (this apex may lie on AB). If the index of $Ind^A(FV^A(w_{M^*}, AB))$ is greater than ν, the apex of $\mathscr{W}^{M^*}$ is temporarily assumed to be u_ν and $\mathscr{W}^{M^*}$ is defined to lie between $\mathcal{F}_r(s_{i+1}, u_\nu)$ and $\mathcal{F}_l(u_{\nu-1}, u_\nu)$. The index N^* is defined similarly. It is clear that at least one of the equalities $w_{M^*} = w_M$ or $w_{N^*} = w_N$ holds.

We use $\mathcal{R}(x, y)$ to denote the ray from x towards y. In addition, $\mathcal{R}_a(x, y)$ denotes the ray from a and parallel to $\mathcal{R}(x, y)$ (Fig. 15).

Despite our definition of the regions $\mathscr{W}^i$ for all vertices $w_i \in BC$, we refine this definition for $\mathscr{W}^{N^*}$ (*resp.* $\mathscr{W}^{M^*}$) when $N^* \neq N$ (*resp.* $M^* \neq M$) or the floating edge of AC (*resp.* AB) lies under the line through u_ν and v_μ. At most one of the floating edges lies under $\mathcal{R}(u_\nu, v_\mu)$. Because otherwise, either $v_m u$ will see $u_{\nu+1}$ or u_ν will see $v_{\mu+1}$ which is in contradiction with the selection of u_ν and v_μ. Let v be a point on $\mathcal{R}_{v_\mu}(r_{\mu+1}, v_\mu)$ when the floating edge of AC lies under $\mathcal{R}(u_\nu, v_\mu)$, or be v_μ otherwise. Similarly, u is defined to be either u_ν or a point on $\mathcal{R}_{u_\nu}(s_{\nu+1}, u_\nu)$. The regions $\mathscr{W}^{N^*}$ and $\mathscr{W}^{M^*}$ are restricted to lie under the line through u and v. Moreover, we know that at most one of the indices M^* and N^* is not equal to its corresponding index M or N. Without loss of generality, assume that $N^* \neq N$.

Then, we additionally restrict the region $\mathscr{W}^{N^*}$ as follows (this restriction is not applied when we reconstruct $\mathcal{Z}$ or $\mathcal{Z}'$). Let p be a point inside the intersection of $\mathscr{W}^N$ and $\mathcal{F}_b(u,v)$ and with an arbitrary positive distance from $\mathcal{R}(u,v)$. We determine t'_{N^*} on its edge and with ϵl distance above the lower endpoint of this edge where $\epsilon > 0$ and l is the number of vertices in AC and BC whose $r'_{(\cdot)}$'s and $t'_{(\cdot)}$'s lie on this edge. The region $\mathscr{W}^{N^*}$ is restricted to lie under the line through t'_{N^*} and p (see Fig. 16).

Let s_α be a point on its edge and with ϵk distance below the upper endpoint of this edge where $\epsilon > 0$ and k is the number of vertices in AB whose $s_{(\cdot)}$'s lie on this edge. Similarly, let s'_α be a point on its edge and with ϵm distance above the lower endpoint of this edge where $\epsilon > 0$ and m is the number of vertices in AB and BC whose $s'_{(\cdot)}$'s and $t'_{(\cdot)}$'s lie on this edge. The value of ϵ is small enough such that s_α lies above s'_α. The points r_δ and r'_δ are defined similarly.

As shown in Fig. 15, let $\mathcal{S}$ (*resp.* $\mathcal{T}$) be the strip defined by the supporting lines of $\mathcal{R}(s_\alpha, u)$ and $\mathcal{R}_{s'_\alpha}(s_\alpha, u)$ (*resp.* $\mathcal{R}(r_\delta, v)$ and $\mathcal{R}_{r'_\delta}(r_\delta, v)$).

Lemma 2. *It is always possible to enlarge the floating edges of AC and AB and re-position the vertices that lie under the enlarged edges such that* $\mathscr{W}^{M^*} \bigcap \mathcal{S}$ *and* $\mathscr{W}^{N^*} \bigcap \mathcal{T}$ *are not empty and the new positions of vertices of* $\mathcal{X}$ *satisfy their visibility relations in the visibility graph.*

Proof. Assume that the intersection of $\mathscr{W}^{M^*}$ and $\mathcal{S}$ is empty. According to the definition of M^*, the apex of $\mathscr{W}^{M^*}$ either lies above the floating edge of AC or lies on AB. This implies that enlarging the floating edge of AC only affects half-plane $\mathcal{F}_b(v_\mu, u_\nu)$ that defines up-side of $\mathscr{W}^{M^*}$. Then, we can enlarge the floating edge of AC in such a way that the lower defining ray of $\mathcal{S}$ and the upper defining half-plane of $\mathscr{W}^{M^*}$ intersect inside $\mathscr{W}^{M^*}$ which means that the intersection of $\mathscr{W}^{M^*}$ and $\mathcal{S}$ is not empty. Moreover, when this intersection is not empty, this extension will just increase the intersection. On the other hand, enlarging this edge changes the position of vertices of $\mathcal{X}$ which lie under this edge. For these vertices, we have their corresponding points r's. By enlarging the floating edge of AC, the new positions will be computed according to their definition (for a vertex v_i it must lie on the supporting line of r_i and v_{i-1}) to satisfy the visibility relations in the visibility graph reduced to vertices of $\mathcal{X}$. To complete the proof, it is simple to see that extending the floating edge of AB will again increase the intersection of BC^{M^*} and $\mathcal{S}$.

The proof for w_{N^*} is analogously the same. □

After locating the position of vertices in $\mathcal{X}$ (by possibly extending the floating edges), we place the vertices of $\mathcal{Y}$ as follows: If $N^* \neq N$, then we set p as w_N and place $w_{M^*} = w_M$ inside the intersection of $\mathscr{W}^M$ and $\mathcal{S}$ in such a way that both w_M and w_N be visible to u and v; neither w_M blocks the visibility of w_N, nor w_N blocks the visibility of w_M. When $M^* \neq M$, w_M and w_N are positioned analogously. Finally, if $M^* = M$ and $N^* = N$, we select a point from $\mathcal{S} \cap \mathscr{W}^M$ as w_M and a point from $\mathcal{T} \cap \mathscr{W}^N$ as w_N again in such a way that both see u and v. Then, we put the vertices $w_{M+1}, \ldots, w_{N-1}$ on a slightly concave chain from w_M to w_N in such a way that each w_j $(M \leq j \leq N)$ lies inside $\mathscr{W}^j$ and sees u and v.

Based on the definition of $\mathscr{W}^i$'s regions and the specified positions of vertices inside these regions, this setting is compatible with the visibility graph restricted to the vertices of $\mathcal{X}$ and $\mathcal{Y}$.

4.2. *Reconstructing $\mathcal{Z}$ and $\mathcal{Z}'$*

In this step, we place the vertices of $\mathcal{Z}$ and $\mathcal{Z}'$ to complete the reconstruction procedure. As said before, $\mathcal{Z}$ (resp. $\mathcal{Z}'$) is a part of the target pseudo-triangle with $< u_\nu, u_{\nu+1}, \ldots, B, \ldots, w_M >$ (resp. $< v_\mu, v_{\mu+1}, \ldots, C, \ldots, w_N >$) boundary vertices. Here, we only describe how to build $\mathcal{Z}$. The construction of $\mathcal{Z}'$ is symmetrically the same.

Location of a vertex $u_i \in \mathcal{Z}$ is determined by the intersection point of the rays $\mathcal{R}(s_i, u_{i-1})$ and $\mathcal{R}(s'_i, LV^B(u_i, BC))$ and location of a vertex $w_h \in \mathcal{Z}$ is an arbitrary point on $\mathcal{R}(t'_h, w_{h+1})$ inside the region $\mathscr{W}^h$. Therefore, to construct $\mathcal{Z}$ we start from $u_{\nu+1}$ and w_{M-1}, and in each step, we determine the position of one of the vertices and go forward to the next vertex. This is done by incrementally determining direction of the rays $\mathcal{R}(s_i, u_{i-1})$, $\mathcal{R}(s'_i, LV^B(u_i, BC))$, and $\mathcal{R}(t'_h, w_{h+1})$ as well as $\mathscr{W}^h$ regions.

Consider the edges of the pseudo-triangle on which the points s_i, s'_i, r_j, r'_j, and t'_l for $i > \nu$, $j > \mu$, and $l < M$ and $l > N$ lie. Keep an upper point and a lower point for each edge. Initialize the upper point with the upper endpoint of that edge or the latest located $s_{(\cdot)}$ or $r_{(\cdot)}$ on this edge. Initialize the lower point with the lower endpoint of the edge. Position of each $s_{(\cdot)}$, $r_{(\cdot)}$, $s'_{(\cdot)}$, $r'_{(\cdot)}$, and $t'_{(\cdot)}$ is determined whenever we need the rays passing through them. We place the points $s'_{(\cdot)}$, $r'_{(\cdot)}$, and $t'_{(\cdot)}$, with $\epsilon > 0$ distance above the current lower point of their edges and place the points $s_{(\cdot)}$ and $r_{(\cdot)}$, with $\epsilon > 0$ distance below the upper point of their edges. Whenever a new $s_{(\cdot)}$, $r_{(\cdot)}$, $s'_{(\cdot)}$, $r'_{(\cdot)}$, or $t'_{(\cdot)}$ point is located on an edge, the upper or lower point of that edge is updated properly.

More precisely, assume that we have already determined positions of vertices $u_\nu, u_{\nu+1}, \ldots, u_{i-1}$ $(i > \nu)$ as well as the vertices $w_M, w_{M-1}, \ldots, w_{j+1}$ $(j < M)$. To determine the position of one of the vertices u_i and w_j we do as follows: Let w_k be $LV^B(u_i, BC)$. If $k < j$, then we have already located the position of w_k, and directions of the rays $\mathcal{R}(s_i, u_{i-1})$ and $\mathcal{R}(s'_i, LV^B(u_i, BC))$ are known. We will show in Lemma 3 that these rays intersect. So, u_i is located on the intersection point of these rays (Fig. 17a). Otherwise, we must first determine position of w_j which lies on $\mathcal{R}(t'_j, w_{j+1})$ and inside $\mathscr{W}^j$ (Fig. 17b). The position of w_{j+1} is already known and t'_{j+1} is determined according to the above paragraph. From these two points the direction of $\mathcal{R}(t'_j, w_{j+1})$ is obtained. The region $\mathscr{W}^j$ is determined as follows: Suppose that $FV^A(w_j, AB) = u_k$ and $FV^A(w_j, AC) = v_l$. We define $\mathscr{W}^j$ as in the previous section with the exception that it may be possible that only one of the vertices u_k and v_l exists. By Corollary 6, for $j < M$, u_j always exists. If w_j sees no vertex from AC, then it would see a part of the floating edge of AC. Hence, we consider the upper endpoint of this edge as v_{l-1}. From properties 5 and 7 we know that $\mathscr{W}^j$ is not empty and lies to the left of $\mathscr{W}^{j+1}$. Moreover, it will be shown in Lemma 3 that $\mathcal{R}(t'_j, w_{j+1})$ intersects $\mathcal{R}_{s'_\alpha}(s_\alpha, u)$. Since $\mathcal{R}_{s'_\alpha}(s_\alpha, u)$ passes through all $\mathscr{W}^{(\cdot)}$, $\mathcal{R}(t'_j, w_{j+1})$ passes through $\mathscr{W}^j$. Therefore, we can determine the position of w_j.

Note that the definitions of s_i's and t'_j's enforce the concavity of the vertices on AB and BC, respectively. According to the definition of $\mathcal{R}(s_i, u_{i-1})$ and $\mathcal{R}(s'_i, LV^B(u_i, BC))$ for u_i and $\mathcal{R}(t'_j, w_{j+1})$ and $\mathscr{W}^j$ for w_j, in both cases (locating u_i or w_j), visibility of the newly located vertex is the same as its visibility in the visibility graph (restricted to the vertices of $\mathcal{X}$, $\mathcal{Y}$, and the constructed part of $\mathcal{Z}$). This means that at the end of this construction where vertices B and C are located the visibility graph of the constructed polygon is consistent with the input visibility graph.

Lemma 3. *The rays $\mathcal{R}(s_i, u_{i-1})$ and $\mathcal{R}(s'_i, LV^B(u_i, BC))$ for $i > \nu$ are convergent inside $\mathcal{S}$.*

Proof. Remember that $\mathcal{S}$ is the strip defined by the supporting lines of $\mathcal{R}(s_\alpha, u)$ and $\mathcal{R}_{s'_\alpha}(s_\alpha, u)$. By Corollaries 1 and 5, we know that s_i lies above the strip $\mathcal{S}$ and s'_i lies below this strip. Then, it is enough to show that for $i > \nu$, $\mathcal{R}(s'_i, LV^B(u_i, BC))$ crosses $\mathcal{R}_{s'_\alpha}(s_\alpha, u)$ and $\mathcal{R}(s_i, u_{i-1})$ crosses $\mathcal{R}(s_\alpha, u)$. We first prove that $\mathcal{R}_{s'_\alpha}(s_\alpha, u)$ intersects $\mathcal{R}(s'_i, LV^B(u_i, BC))$. Let $LV^B(u_i, BC) = w_h$. For $M^* \leq h \leq M$, it can be easily shown by induction that w_h is located above $\mathcal{R}(t'_{M^*}, w_M)$. Moreover, it is simple to see that s'_i must lie below t'_{M^*}. Then, knowing that $\mathcal{R}(t'_{M^*}, w_M)$ crosses $\mathcal{R}_{s'_\alpha}(s_\alpha, u)$ implies that $\mathcal{R}(s'_i, w_h)$ intersects $\mathcal{R}_{s'_\alpha}(s_\alpha, u)$ as well. From the fact that w_{M^*} lies inside $\mathcal{S}$, it can also be shown by induction that w_h for $h < M^*$ lies inside $\mathcal{S}$ which means that $\mathcal{R}(s'_i, w_h)$ crosses $\mathcal{R}_{s'_\alpha}(s_\alpha, u)$.

To complete the proof, we prove by induction on i that $\mathcal{R}(s_i, u_{i-1})$ crosses $\mathcal{R}(s_\alpha, u)$. It is clear that $s_{\nu+1}$ is located above s_α which means that $\mathcal{R}(s_{\nu+1}, u_\nu)$ intersects $\mathcal{R}(s_\alpha, u)$. From the previous paragraph we know that $\mathcal{R}_{s'_\alpha}(s_\alpha, u)$ intersects $\mathcal{R}(s'_{\nu+1}, LV^B(u_{\nu+1}, BC))$. Therefore, $\mathcal{R}(s_{\nu+1}, u_\nu)$ and $\mathcal{R}(s'_{\nu+1}, LV^B(u_{\nu+1}, BC))$ will intersect at a point within $\mathcal{S}$. Since we put $u_{\nu+1}$ at this intersection point, as the induction step, assume that u_{i-1} lies inside $\mathcal{S}$ where $i > \nu+1$. Then, $\mathcal{R}(s_i, u_{i-1})$ intersects $\mathcal{R}(s_\alpha, u)$. □

5. Analysis

In the previous sections, we proved several properties on the visibility graph of a pseudo-triangle and proposed an algorithm that constructs a pseudo-triangle for a given pair of the visibility graph $\mathcal{G}(V, E)$ and Hamiltonian cycle $\mathcal{H}$ when this pair supports these properties. In this section, we analyze the running-time of algorithms required to check these properties and the running time of the reconstruction algorithm.

To check Property 1, we need a linear time trace on vertices of $\mathcal{G}$ according to their order in $\mathcal{H}$. This is done in $\mathcal{O}(n)$ time, where $n = |V|$. If two corners are identified in this way, the existence of a tower polygon corresponding to the pair of $\mathcal{G}$ and $\mathcal{H}$ can also be verified in linear time [21]. Property 2 can be verified in $\mathcal{O}(|E|)$ time by a simple trace of the edge list of the visibility graph. Precisely, for each vertex $u \in AB$ we maintain the minimum index, maximum index, and number of vertices of the other side-chains AC and BC which are visible from u. After finishing this trace, from these triple of parameters (minimum index, maximum index, number of visible vertices) the Property 2 is checked in $\mathcal{O}(n)$ time. To verify the rest of the properties, it is required to know the visible subchains from each vertex. These subchains are obtained as by-products using the method proposed for checking Property 2. Having these subchains for each vertex, Property 3 can be verified in $\mathcal{O}(n)$ time.

In Property 4, for each pair (AB, AC) of side-chains, we must find all pairs of visible vertices $(u \in AB, v \in AC)$ such that $(N^A(u), v)$ and $(u, N^A(v))$ are invisible. Having the visible subchains for each vertex, this check is done in constant time for each edge $(u, v) \in E$. Therefore, all pairs of vertices $(u \in AB, v \in AC)$ satisfying the assumption of this property can be obtained in $\mathcal{O}(|E|)$ time. Then, for each pair, the three necessary conditions are checked in constant time using the maintained visible subchains of u and v vertices. Checking properties 5, 6 and 7 can be done in $\mathcal{O}(n)$ time by simple trace on the side-chains. Therefore, all properties can be verified in $\mathcal{O}(|E|)$ time.

To complete the analysis, we compute the running time of the reconstruction algorithm presented in Section 4. Assume that $\mathcal{G}$ satisfies all of the properties introduced in Section 3 and we know the visible subchains of each vertex according to their order in $\mathcal{H}$. The side-chains of the target pseudo-triangle are identified in linear time according to the algorithm described in the proof of Lemma 1. Reconstructing $\mathcal{X}$ is done using the tower reconstruction algorithm whose running time is linear in terms of the number of edges in the visibility graph reduced to $\mathcal{X}$. To reconstruct $\mathcal{Y}$, the algorithm needs to determine the floating edges of AB and AC which can be done in constant time. Computing the $\mathscr{W}$-type regions (for each vertex $w_i \in BC$) and determining the vertices w_{N^*} and w_{M^*} needs $\mathcal{O}(n)$ time. If the conditions of Lemma 2 are not satisfied, the floating edges of AB and AC must be extended which is done in $\mathcal{O}(1)$ time: A lower bound for the increase in floating edges can be computed by using Thales' theorem and trigonometric functions. Locating each vertex of $\mathcal{Y}$ is also done in constant time. Finally, placing each vertex of $\mathcal{Z}$ and $\mathcal{Z}'$ takes constant time, as well. Therefore, the total running time of the algorithm is $\mathcal{O}(|E|)$ time. We can combine all results as:

Theorem 2. *The visibility graph and the boundary vertices of a pseudo-triangle satisfy properties 1 to 7, and conversely, for any pair of graph $\mathcal{G}$ and Hamiltonian cycle $\mathcal{H}$ satisfying these properties, there is a pseudo-triangle $\mathcal{P}$ whose visibility graph and boundary vertices are respectively isomorphic to $\mathcal{G}$ and $\mathcal{H}$. Checking these* *properties and reconstructing such a polygon can be done in $\mathcal{O}(|E|)$ time.*

6. Conclusion

In this paper, we considered properties of the visibility graph of a pseudo-triangle and obtained a set of necessary and sufficient conditions that such graphs must have. Then, we propose an algorithm to reconstruct a polygon from a given visibility graph which supports these properties. This characterizing and reconstructing problem, despite its long history, is still at the start of its way to be solved for all polygons.

References

1. Asano, T., Asano, T., Guibas, L., Hershberger, J., and Imai, H. "Visibility of disjoint polygons", *Algorithmica 1*, pp. 49–63 (1986).
2. Welzl, E. "Constructing the visibility graph for n-line segments in $o(n^2)$ time", *Information Processing Letters 20*(4), pp. 167–171 (1985).
3. Hershberger, J. "Finding the visibility graph of a simple polygon in time proportional to its size", *Third Annual Symposium on Computational Geometry* , pp. 11–20 (1987).
4. Chazelle, B. "Triangulating a simple polygon in linear time", *Discrete and Computational Geometry 6*, pp. 485–424 (1991).
5. Davies, J., Krawczyk, T., McCarty, R., and Walczak, B. "Coloring polygon visibility graphs and their generalizations", *Journal of Combinatorial Theory, Series B 161*, pp. 268–300 (2023).
6. Caughman, J. S., Dunn, C. L., Laison, J. D., Neudauer, N. A., and Starr, C. L. "Area, perimeter, height, and width of rectangle visibility graphs", *Journal of Combinatorial Optimization 46*(3), p. 18 (2023).
7. Schmidt, J., and Köhne, D. "A simple scalable linear time algorithm for horizontal visibility graphs", *Physica A: Statistical Mechanics and its Applications 616*, pp. 128–601 (2023).
8. Lee, W., Choi, G.-H., and wan Kim, T. "Visibility graph-based path-planning algorithm with quadtree representation", *Applied Ocean Research 117*, pp. 102–887 (2021).
9. Everett, H. *Visibility Graph Recognition.* PhD thesis, University of Toronto, Department of Computer Science, 1990.
10. Ameer, S., Gibson-Lopez, M., Krohn, E., and Wang, Q. "On the visibility graphs of pseudo-polygons: Recognition and reconstruction", *18th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2022)* , vol. 227 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 7:1–7:13 (2022).
11. Gibson, M., Krohn, E., and Wang, Q. "A characterization of visibility graphs for pseudo-polygons", *23rd Annual European Symposium (ESA 2015)* , pp. 607–618 (2015).
12. O'Rourke, J., and Streinu, I. "Vertex-edge pseudo-visibility graphs: Characterization and recognition", *13'th Annual Symposium on Computational Geometry (SoCG 1997)* , pp. 119–128 (1997).
13. O'Rourke, J., and Streinu, I. "The vertex-edge visibility graph of a polygon", *Computational Geometry 10*(2), pp. 105–120 (1998).
14. Streinu, I. "Non-stretchable pseudo-visibility graphs", *Computational Geometry 31*(3), pp. 195–206 (2005).

15. Casel, K., Fernau, H., and Grigoriev, A. "Combinatorial properties and recognition of unit square visibility graphs", *Discrete Computational Geometry 69*, p. 937–980 (2023).
16. Boomari, H., Ostovari, M., and Zarei, A. "Recognizing visibility graphs of triangulated irregular networks", *Fundamenta Informaticae 179*(4), pp. 345–360 (2021).
17. Cardinal, J., and Hoffmann, U. "Recognition and Complexity of Point Visibility Graphs", *31st International Symposium on Computational Geometry (SoCG 2015)* , vol. 34, pp. 171–185 (2015).
18. Abrahamsen, M., Adamaszek, A., and Miltzow, T. "The art gallery problem is ∃r-complete", *Journal of the ACM 69*(1), pp. 1–70 (2021).
19. Schaefer, M. *Journal of Graph Algorithms and Applications 27*(9), p. 803–841 (2024).
20. Everett, H., and Corneil, D. G. "Recognizing visibility graphs of spiral polygons", *Journal of Algorithms 11*(1), pp. 1–26 (1990).
21. Colley, P., Lubiw, A., and Spinrad, J. "Visibility graphs of towers", *Computational Geometry 7*, pp. 161–172 (1997).
22. Boomari, H., and Zarei, A. "Visibility graphs of anchor polygons", *J. Graph Algorithms Appl. 26*(1), pp. 15–34 (2022).
23. Abello, J., Egecioglu, O., and Kumar, K. "Visibility graphs of staircase polygons and the weak Bruhat order I: From visibility graphs to maximal chains", *Discrete & Computational Geometry 14*(3), pp. 331–358 (1995).
24. Coullard, C. R., and Lubiw, A. "Distance visibility graphs", *International Journal of Computational Geometry & Applications 2*(4), pp. 349–362 (1992).
25. ElGindy, H. *Hierarchical decomposition of polygons with applications.* PhD thesis, McGill University, Department of Computer Science, 1985.
26. Ghosh, S. K. "On recognizing and characterizing visibility graphs of simple polygons", *1st Scandinavian Workshop on Algorithm Theory (SWAT 88)* , Springer Berlin Heidelberg, pp. 96–104 (1988).
27. Ghosh, S. K. "On recognizing and characterizing visibility graphs of simple polygons", *Discrete & Computational Geometry 17*(2), pp. 143–162 (1997).
28. Abello, J., Lin, H., and Pisupati, S. "On visibility graphs of simple polygons", *Congressus Numerantium*(90), pp. 119–128 (1992).

Figures

Fig. 1: (a) A Spiral polygon, (b) a tower polygon, and (c) a pseudo-triangle.
Fig. 2: Constructing a tower polygon.
Fig. 3: Notation used for vertices.
Fig. 4: A pseudo-triangle with B and C as its detectable corners.
Fig. 5: Interpreting a pseudo-triangle as a tower polygon: (a) initial pseudo-triangle, (b) equivalent tower polygon.
Fig. 6: (a) Property 3: $P^A(v)$ and $FV^A(v, AB)$ see each other, (b) Corollary 1: $FV^A(v', AB)$ cannot be closer to A than $FV^A(v, AB)$.
Fig. 7: (a) Corollary 3: $P^A(u, 2)$ and $P^A(v)$ must see each other, (b) Corollary 4: u and $N^A(v)$ see each other, (c) Corollary 5: Visible vertices of AC from $N^A(u, k)$ are also visible to $N^A(u, k-1)$.
Fig. 8: (a) Property 4(a): u and v must see common vertices on mW, (b) Property 4(b): w' is not closer to B than $LV^B(N^A(u), BC)$, Property 4(c): $FV^B(N^A(v), BC)$ is not closer to B than $LV^B(N^A(u), BC)$.
Fig. 9: Property 5: $FV^A(w, AB)$ and $FV^A(w, AC)$ must see each other.
Fig. 10: Different cases of Property 6.
Fig. 11: Corollary 7: u_i (resp. v_i) is not closer to A than u_j (resp. v_j).
Fig. 12: The partitions of the initial polygon in reconstruction algorithm: the light-gray region is $\mathcal{X}$, the dark-gray is $\mathcal{Y}$ and the white parts are $\mathcal{Z}$ and $\mathcal{Z}'$.
Fig. 13: $\mathscr{W}_{i,l}^{j}$ is the shaded region.
Fig. 14: Points $s_{(\cdot)}$, $s'_{(\cdot)}$, $r_{(\cdot)}$, $r'_{(\cdot)}$, and $t'_{(\cdot)}$.
Fig. 15: The rays $\mathcal{R}(s_\alpha, u)$, $\mathcal{R}_{s'_\alpha}(s_\alpha, u)$, $\mathcal{R}(r_\delta, v)$, and $\mathcal{R}_{r'_\delta}(r_\delta, v)$.
Fig. 16: Restricting $\mathscr{W}^{N^*}$.
Fig. 17: (a) Determining u_i, (b) Determining w_j.

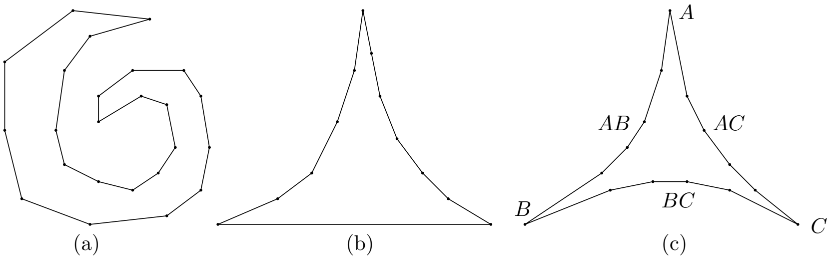

Fig. 1: (a) A Spiral polygon, (b) a tower polygon, and (c) a pseudo-triangle.

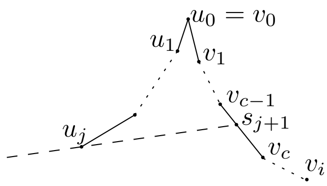

Fig. 2: Constructing a tower polygon.

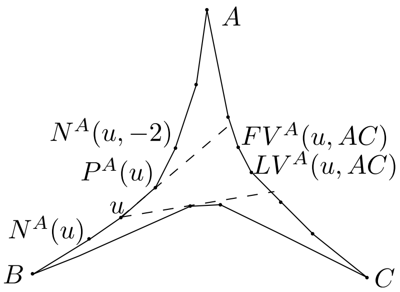

Fig. 3: Notation used for vertices.

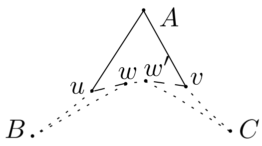

Fig. 4: A pseudo-triangle with B and C as its detectable corners.

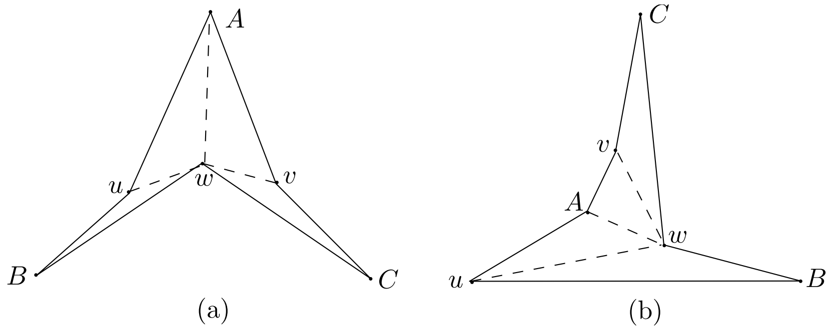

Fig. 5: Interpreting a pseudo-triangle as a tower polygon: (a) initial pseudo-triangle, (b) equivalent tower polygon.

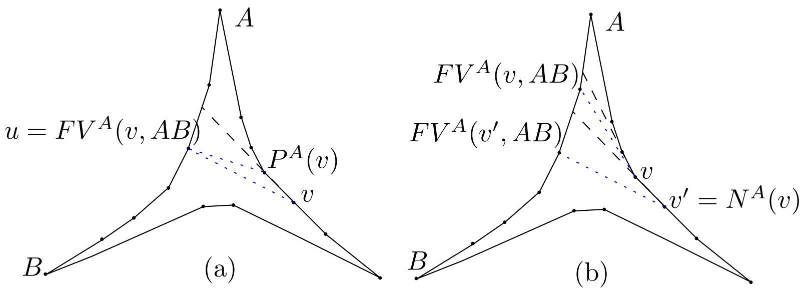

Fig. 6: (a) Property 3: $P^A(v)$ and $FV^A(v, AB)$ see each other, (b) Corollary 1: $FV^A(v', AB)$ cannot be closer to A than $FV^A(v, AB)$.

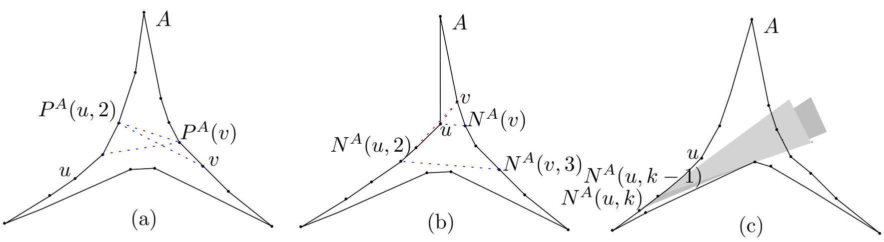

Fig. 7: (a) Corollary 3: $P^A(u, 2)$ and $P^A(v)$ must see each other, (b) Corollary 4: u and $N^A(v)$ see each other, (c) Corollary 5: Visible vertices of AC from $N^A(u, k)$ are also visible to $N^A(u, k-1)$.

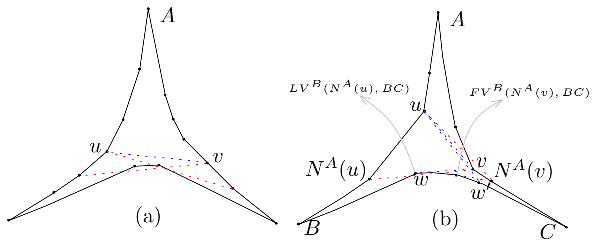

Fig. 8: (a) Property 4(a): u and v must see common vertices on mW, (b) Property 4(b): w' is not closer to B than $LV^B(N^A(u), BC)$, Property 4(c): $FV^B(N^A(v), BC)$ is not closer to B than $LV^B(N^A(u), BC)$.

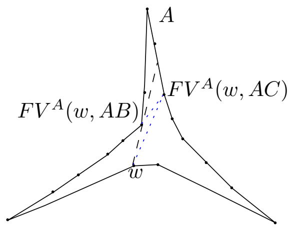

Fig. 9: Property 5: $FV^A(w, AB)$ and $FV^A(w, AC)$ must see each other.

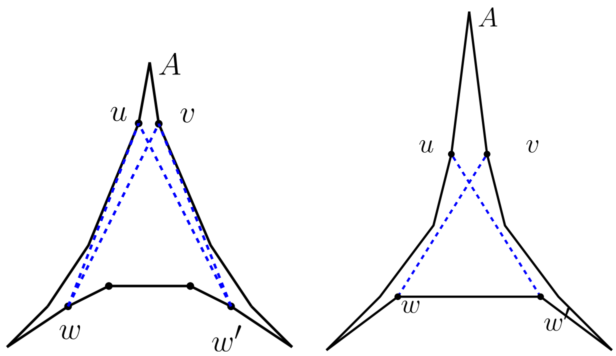

Fig. 10: Different cases of Property 6.

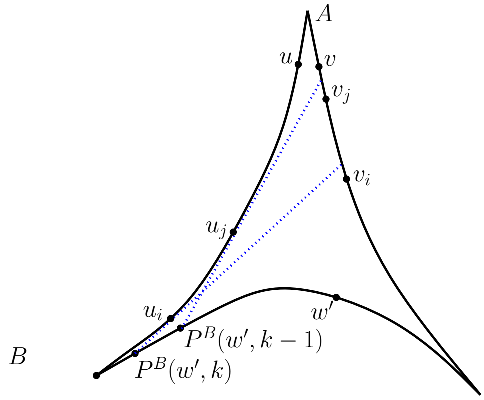

Fig. 11: Corollary 7: u_i (resp. v_i) is not closer to A than u_j (resp. v_j).

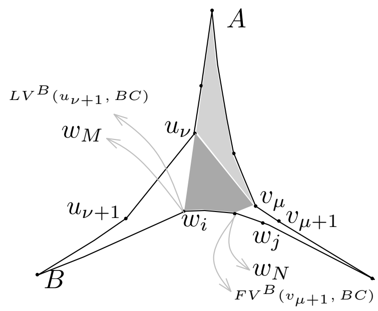

Fig. 12: The partitions of the initial polygon in reconstruction algorithm: the light-gray region is $\mathcal{X}$, the dark-gray is $\mathcal{Y}$ and the white parts are $\mathcal{Z}$ and $\mathcal{Z}'$.

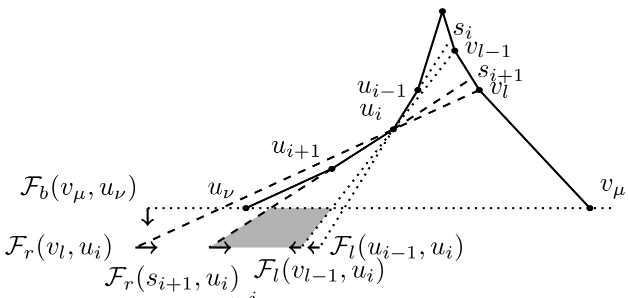

Fig. 13: $\mathscr{W}_{i,l}^{j}$ is the shaded region.

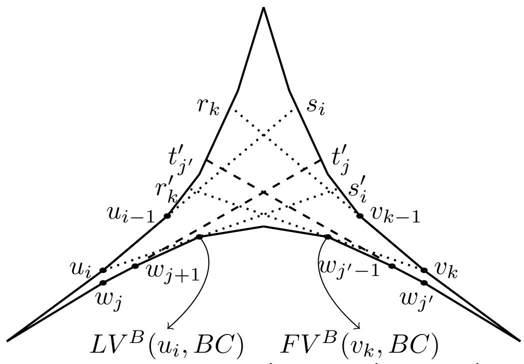

Fig. 14: Points $s_{(\cdot)}$, $s'_{(\cdot)}$, $r_{(\cdot)}$, $r'_{(\cdot)}$, and $t'_{(\cdot)}$.

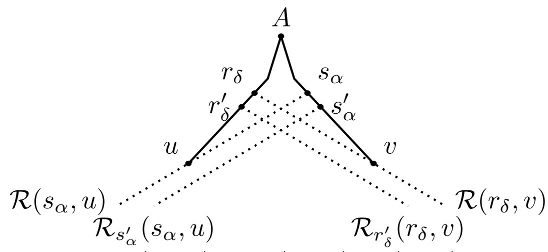

Fig. 15: The rays $\mathcal{R}(s_\alpha, u)$, $\mathcal{R}_{s'_\alpha}(s_\alpha, u)$, $\mathcal{R}(r_\delta, v)$, and $\mathcal{R}_{r'_\delta}(r_\delta, v)$.

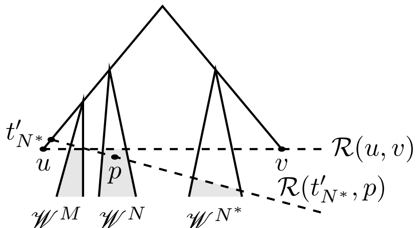

Fig. 16: Restricting $\mathscr{W}^{N^*}$.

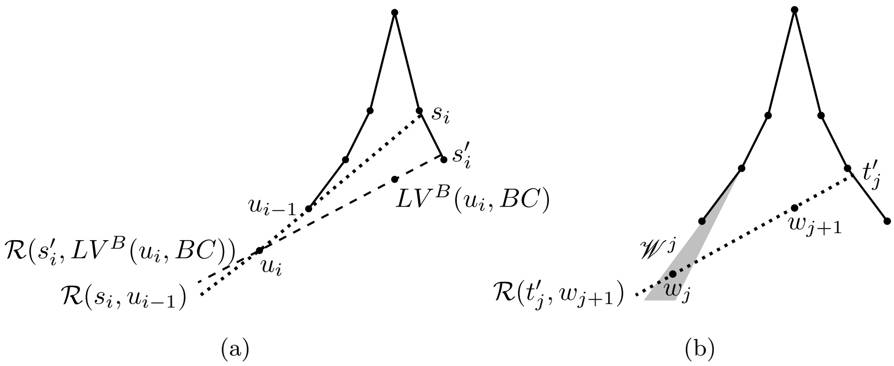

Fig. 17: (a) Determining u_i, (b) Determining w_j.

Sahar Mehrpour received her master from Sharif University of Technology in computer science in 2016. She is currently a Ph.D. candidate in Computer Science at George Mason university. Her research is lying at the intersection of software engineering and human-computer interaction

Alireza Zarei is a computer science faculty member at Sharif University of Technology since 2009. His main research interest is algorithms and complexity, especially on geometric problems. Analyzing problem complexity and designing efficient algorithms and data structures are his core research directions. Complexity theory, graph and discrete mathematics, probability and random processes are other his related research interests.