



HAL
open science

Visibility Graphs of Anchor Polygons

Hossein Boomari, Alireza Zarei

► **To cite this version:**

Hossein Boomari, Alireza Zarei. Visibility Graphs of Anchor Polygons. 1st International Conference on Theoretical Computer Science (TTCS), Aug 2015, Tehran, Iran. pp.72-89, 10.1007/978-3-319-28678-5_6 . hal-01446265

HAL Id: hal-01446265

<https://inria.hal.science/hal-01446265>

Submitted on 25 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Visibility Graphs of Anchor Polygons

Hossein Boomari and Alireza Zarei

Sharif University of Technology,
Department of Computer Science

Abstract. Visibility graph of a polygon corresponds to its internal diagonals and boundary edges. For each vertex on the boundary of the polygon, we have a vertex in this graph and if two vertices of the polygon see each other there is an edge between their corresponding vertices in the graph. Two vertices of a polygon see each other if and only if their connecting line segment completely lies inside the polygon. Recognizing visibility graphs is the problem of deciding whether there is a simple polygon whose visibility graph is isomorphic to a given graph. Another important problem is to reconstruct such a polygon if there is any. These are well-known and well-studied, but yet open problems in geometric graphs and computational geometry. However, these problems have been solved efficiently for special cases where the target polygon is known to be a tower or a spiral polygon. In this paper, we solve these recognizing and reconstruction problems for another type of polygons, named *anchor polygons*.

Keywords: Visibility graph, polygon reconstruction, recognizing visibility graph, anchor polygon.

1 Introduction

Visibility graph of a simple planar polygon is a graph in which there is a vertex for each vertex of the polygon and for each pair of visible vertices of the polygon there is an edge between their corresponding vertices in this graph. Two points in a simple polygon are visible from each other if and only if their connecting segment completely lies inside the polygon. In this definition, each pair of adjacent vertices on the boundary of the polygon are assumed to be visible from each other. This implies that we have always a Hamiltonian cycle in a visibility graph which determines 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 [10], computational geometry [9] and robotics [11]. There are several efficient polynomial time algorithms for this problem [9].

This concept has been studied in reverse as well: Is there any simple polygon whose visibility graph is isomorphic to a given graph and if there is such a polygon, is there any way to reconstruct it (find positions for its vertices on the plain)? The former problem is known as recognizing visibility graphs and the latter one is known as reconstructing polygon from visibility graph. Both

these problems are widely open. The only known result about the computational complexity of these problems is that they belong to *PSPACE* [1] complexity class and more specifically belong to the class of *Existence theory of reals* [8]. This means that it is not even known whether these problems are *NP-Complete* or can be solved in polynomial time. Even, if we are given the Hamiltonian cycle of the visibility graph which determines the order of vertices on the boundary of the target polygon, the exact complexity class of these polygons are still unknown.

However, these problems have been solved efficiently for special cases of tower and spiral polygons. In these special cases, we know that the given graph and the Hamiltonian cycle correspond to a tower polygon or a spiral one. A tower polygon consists of two concave chains on its boundary whose share one vertex and the other end point of these chains are connected by a segment (See Fig. 1.a). A spiral polygon has exactly one concave and one convex chain on its boundary (See Fig. 1.b). The recognizing and reconstruction problems have been solved for tower polygons in linear time in terms of the size of the graph [3]. It has been shown in [3] that a given graph is the visibility graph of a tower polygon if and only if by removing the edges of the Hamiltonian cycle from the graph, an isolated vertex and a connected bipartite graph are obtained and the bipartite graph has *strong ordering* following the order of vertices in the Hamiltonian cycle. A strong ordering on a bipartite graph $G(V, E)$ with partitions U and W is a pair of $<_U$ and $<_W$ orderings on respectively U and W such that if $u <_U u'$, $w <_W w'$, and there are edges (u, w') and (u', w) in E , the edges (u', w') and (u, w) also exist in E . Graphs having such ordering are also called *strong permutation graphs*. The recognizing and reconstruction problems have been solved efficiently for spiral polygons, too [2]. Because we need this method in our algorithm, we describe this method in more details in Section 2.

Although there is a bit progress on the recognizing and reconstruction problems, there have been plenty of studies on characterizing visibility graphs [2, 3, 5, 6, 12–14]. In 1988, Ghosh introduced three necessary conditions for visibility graphs and conjectured their sufficiency [5]. In 1990, Everett proposed a graph that rejects Ghosh's conjecture [1]. He also refined the third necessary condition of Ghosh to a new stronger condition [6]. 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 [15] rejecting the sufficiency of these conditions. Again, 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. Finally, in 2005 Streinu proposed a counter example for this conjecture [7]. Alongside these tries in 1994, Abello *et al* proposed four necessary constraints for a graph to be visibility graph of a polygon and they conjectured that these constraints are verifiable efficiently [17]. Later in 1995, Abello *et al* showed these constraints are sufficient for recognizing and reconstruction of 2-spiral polygons¹, without much contribution to their computational complexity [16].

¹ polygons with at most 2 concave chains

In this paper, we consider these problems for another type of polygons called *anchor polygons*. The boundary of an anchor polygon is composed of two concave chains and a convex one (See Fig. 1.c). We characterized these polygons with some efficiently realizable constraints and show that both recognizing and reconstruction problems for a given pair of visibility graph and Hamiltonian cycle of an anchor polygon can be solved in $O(n^2)$ time where n is the number of vertices of the graph, or equivalently, the number of vertices of the anchor polygon.

In the remainder of this paper, we first introduce the algorithm of solving reconstruction and recognizing problems for spiral polygons in Section 2. Also, in this section we present some preliminaries and definitions used in next sections. In Section 3, we give an overview of our algorithm and extract key features from the graph to be used in our reconstruction algorithm. In Section 4 we present the algorithm and analyze its efficiency in Section 5.

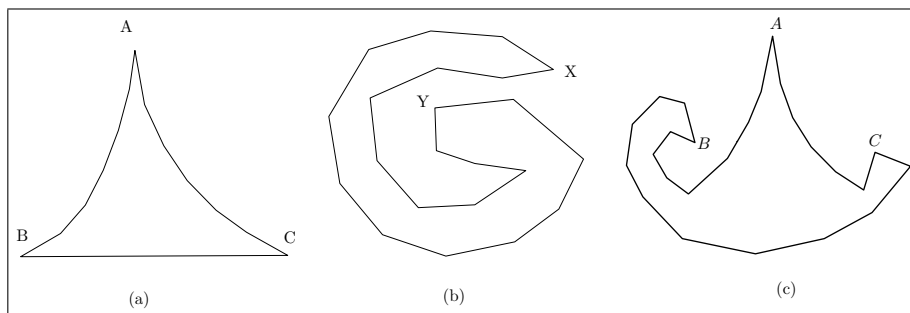


Fig. 1. a) Tower polygon, b) Spiral polygon, c) anchor polygon.

2 Preliminaries and Definitions

In this section, we first briefly describe the recognizing and reconstruction algorithm for spiral polygons. We need these details in some parts of our algorithm. Then, we introduce some definition and basic facts to be used in next sections.

2.1 Spiral Polygons

Assuming that a pair of visibility graph and Hamiltonian cycle belong to a spiral polygon, Everret and Corneil proposed an efficient method to solve recognizing and reconstruction problems in these cases [2]. Here, we briefly describe their method.

The visibility graph of a spiral polygon is a limited subclass of *interval graphs* [2]. This means that any interval graph satisfying another necessary condition corresponds to the visibility graph of a spiral polygon and vice versa (For

the sake of brevity we skip describing this extra condition). This property helps to solve the recognizing problem for the visibility graphs of spiral polygons.

In any interval graph, and equivalently in the visibility graph of a spiral polygon, there are at least two vertices which form cliques with all their neighbors. Moreover, by removing one of these vertices, the remaining graph is still an interval graph. In a spiral polygon two joint vertices that connect its convex and concave chains have this property (form a clique with their neighbors) and by removing one of these vertices the residual graph will be another spiral polygon. Performing this *elimination scheme* from one of the joint vertices toward the other one will finally give us an ordered sequence of removed vertices and a subset of vertices which make a clique composed of the other joint vertex and its neighbors.

Assume that $\langle v_1, v_2, \dots, v_k \rangle$ is the ordered sequence of removing vertices and $\{v_{k+1}, \dots, v_n\}$ is the set of remaining vertices which make a clique. Assume that v_n is a joint vertex and v_{k+1} is the only concave vertex in this set. We put $\{v_{k+1}, \dots, v_n\}$ on the boundary of an arbitrary convex polygon. Then, the position of the vertices $\langle v_1, v_2, \dots, v_k \rangle$ are located in reverse order inductively as follows: Assume that v_c is the last located convex vertex before v_l and v_r is the last located concave vertex before v_l . For the induction base step, we set $v_r = v_n$ and $v_c = v_{k+2}$. To locate the position of v_{l-1} in an inductive step, assume that v_t is the closest convex vertex to v_r which sees v_{l-1} . By closest we mean that v_t is the first vertex on the Hamiltonian cycle when we move from v_r along the reconstructed part of the concave chain and then go along the reconstructed part of the convex chain. If v_l is a convex vertex, v_{l-1} is located somewhere in the angle $\widehat{v'_t v_r v'_{t-1}}$ (see Fig. 2.a) where $v_r v'_t$ (resp. $v_r v'_{t-1}$) is the half line from v_l along $v_r v_t$ (resp. $v_r v_{t-1}$) and in opposite side of v_t (resp. v_{t-1}) and in such a way that the new boundary do not cross itself. Otherwise (if v_l is a concave vertex), v_{l-1} is located somewhere in angle $\widehat{v'_t v_l v'_{t-1}}$ (see Fig. 2.b) where $v_l v_t$ and $v_l v_{t-1}$ half lines are defined similarly.

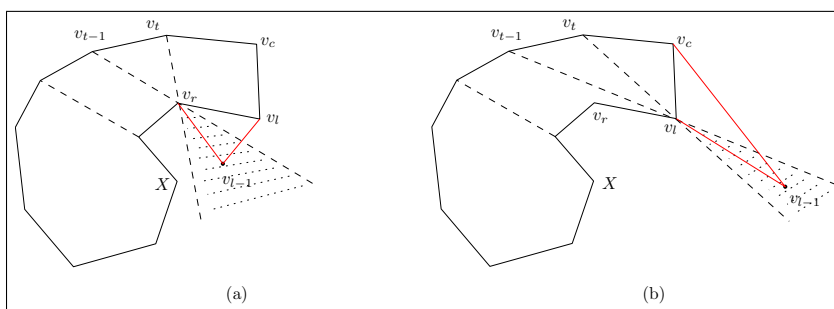


Fig. 2. The reconstruction algorithm for spiral polygons

An important feature of this reconstruction algorithm is that starting from the initial convex polygon v_{k+1}, \dots, v_n , the remainder of the spiral polygon can be reconstructed in an arbitrary close area of the concave vertex of this convex polygon. We use this feature in our reconstruction algorithm.

2.2 Definitions

In an anchor polygon, there are three specific vertices joining the three chains on the boundary of such a polygon. As shown in Fig 1.c, the joint vertex between the concave chains is named A and the other joint vertices are named B and C . For simplicity, we assume that we have a left concave chain from A to B and a right concave chain from A to C and an underneath convex chain from B to C . We may refer to A as top joint vertex and to B and C as the left and right joint vertices, respectively. These names are consistent in all figures and inside the text to help readers have better perspective about the target anchor polygon.

We use term PQ^i for the i^{th} vertex of the boundary of the polygon when we move from vertex P to vertex Q which both lies on the same chain. For example, AB^0 will be vertex A and AB^1 is the first vertex after A on the left concave chain. We also use $UV(P)$ as the closest vertex to U on chain UV which is visible to vertex P . In this notation, U and V can be any of the joint vertices A , B or C .

2.3 Basic Facts

From the convexity or concavity of the chains we have the following basic observations.

Observation 1 *Adjacent vertices (in the given Hamiltonian cycle) of each vertex $a \notin \{A, B, C\}$ of concave chains are not visible from each other.*

Observation 2 *Two vertices of the convex chain BC see each other if and only if no vertex of the concave chains blocks their visibility.*

Observation 3 *If a vertex v_i on the convex chain sees another vertex v_j on this chain, all vertices from v_i to v_j on this chain see each other. Moreover, if a vertex v_i on the convex chain does not see another vertex v_j on this chain and v_i is closer to B on chain BC , then none of the vertices from B to v_i see any one of the vertices from v_j to C .*

Observation 4 *The vertices AB^1 and AC^1 are always visible from each other.*

Observation 5 *At least one of the pairs of vertices (BC^1, BA^1) and (CB^1, CA^1) are visible from each other.*

Proof. The blocking vertices of (BC^1, BA^1) must lie on chain AC and blocking vertices of (CB^1, CA^1) must lie on chain AB . If BC^1 and BA^1 are not visible from each other, then none of the vertices of the chain AB can block the visibility of CB^1 and CA^1 . \square

3 Recognizing Algorithm: Determining joint vertices

We propose a constructive algorithm to solve both recognizing and reconstruction problems for anchor polygons. In this algorithm, we first determine the three chains on the given Hamiltonian cycle. During this process, some necessary conditions of the recognizing algorithm are verified. Then, the area of the target polygon is decomposed into four sub-polygons: a tower polygon, a convex polygon and two spiral polygons. The tower polygon is reconstructed first. Then, the convex polygon is constructed under the base edge of the built tower, and, finally the spiral polygons are built and attached to the sides of the constructed tower and convex polygons (See Fig. 5). Again, during this reconstruction process, the necessary conditions of the given visibility graph are checked to have a recognizing algorithm as well as the reconstruction one.

The details of the decomposition and reconstruction phases are described in Section 4. Here, we give a method for identifying the joint vertices A , B and C from which the three chains on the boundary of the target polygon are obtained. To do this, we first assume that we know vertex A and propose an algorithm for finding vertices B and C . Then, we propose a method for identifying candidate vertices for A .

3.1 Finding joint vertices B and C

When we move from A on the Hamiltonian cycle in both directions, from Observation. 1 and 5 we can find at least one of the joint vertices B or C . This is the first visited vertex in these walks whose adjacent vertices in the Hamiltonian cycle see each other. Our algorithm for finding the other vertex is exactly the same: Walk along the Hamiltonian cycle from A in both directions until a vertex with this property (its adjacent vertices in Hamiltonian cycle see each other) is found in each direction. This algorithm will successfully find correct vertices as B and C if both pairs (BA^1, BC^1) and (CA^1, CB^1) are visible from each other. But, in some cases one of these pairs are not visible. Then, it seems that, our algorithm is failed to find joint vertex B or C .

We assume that both concave chains has at least one vertex other than the joint vertices. Otherwise, the target polygon will be a spiral one and can be recognized and reconstructed by algorithm proposed in [2]. Assume that $G(V, E)$ and H are the given pair of visibility graph and Hamiltonian cycle. The following theorem shows that the joint vertices B and C obtained by our algorithm along with A are the joint vertices of an anchor polygon whose visibility graph and Hamiltonian cycle are equivalent to the given pair of $G(V, E)$ and H if and only if $G(V, E)$ and H belongs to an anchor polygon.

Theorem 1. *Assume that for a given visibility graph $G(V, E)$ and Hamiltonian cycle H and a vertex A , the vertices B and C are the first visited vertices on H when we walk from A in both sides whose adjacent vertices see each other. Then, G and H correspond to an anchor polygon with top vertex A if and only if there is an anchor polygon with joint vertices A , B and C whose pair of visibility graph and Hamiltonian cycle are respectively isomorphic to G and H .*

Note that the proof, we proposed for this theorem is too long. Therefore, we present it in the Appendix section.

3.2 Determining joint vertex A

As a main part of our recognition algorithm, we describe a method for identifying the joint vertex A . If one of the concave chains AB or AC has only one edge (two vertices), the target polygon will be a spiral one and recognizing and reconstruction problems can be solved in such cases using the method proposed in [2]. Therefore, we assume that both chains AB and AC have at least one non-joint vertex. Then, the following observations are true for the joint vertex A .

Observation 6 *The pairs (A, AB^2) and (A, AC^2) do not see each other, but, the pair (AB^1, AC^1) are visible from each other.*

Observation 7 *All visible vertices from A see each other and along with A form a clique in the visibility graph.*

From these observations we have necessary conditions to find candidate vertices for A . We use these conditions in the first phase of our algorithm by moving along the Hamiltonian cycle and finding those vertices whose adjacent vertices see each other, but do not see vertices of distance 2 in the Hamiltonian cycle. For any one of the vertices passing this check we also check Observation 7. Then, we use our algorithm for finding other joint vertices (B and C) corresponding to any one of the candidate vertices for A . Clearly, for any candidate vertex p for A we must find corresponding joint vertices B_p and C_p where chain B_pC_p is convex. Each pair of visible vertices in convex chain B_pC_p must satisfy Observation 3. We show that there are at most three candidate vertices for A which pass the above conditions.

Assume that the given pair of visibility graph and Hamiltonian cycle belongs to an anchor polygon P with joint vertices A_P, B_P and C_P . Then, we have the following theorems.

Theorem 2. *If our algorithm find another candidate vertex for the top joint vertex A other than A_P, B_P and C_P , both chains $A_P B_P$ and $A_P C_P$ in P must lie completely on the same side of the line through vertices $A_P B_P^1$ and $A_P C_P^1$.*

Proof. Assume that a vertex v satisfies all conditions we check in our algorithm for finding candidate vertex A . While $v \notin \{A_P, B_P, C_P\}$, B_P and C_P are distinct vertices. This implies that v and its corresponding other joint vertices B_v and C_v (obtained by our algorithm for finding vertices B and C for a given vertex A) must lie on the convex chain $B_P C_P$ of P . Then, both chains $A_P B_P$ and $A_P C_P$ must lie on the convex chain $B_v C_v$ of the corresponding top joint vertex v . For the sake of a contradiction, assume that the chain $A_P B_P$ or $A_P C_P$ does not completely lie on one side of the line through $A_P B_P^1$ and $A_P C_P^1$. Without loss of generality (W.l.o.g.), assume that $A_P B_P^2$ lies below this line (See Fig. 3).

From convexity of the chain $B_P C_P$, no vertex will block the visibility of $A_P B_P^2$ and $A_P C_P^1$. On the other hand, these vertices lie on the convex chain $B_v C_v$ of the top joint vertex v . Therefore, according to Observation 3 A_P and $A_P B_P^2$ must also see each other which is a contradiction. \square

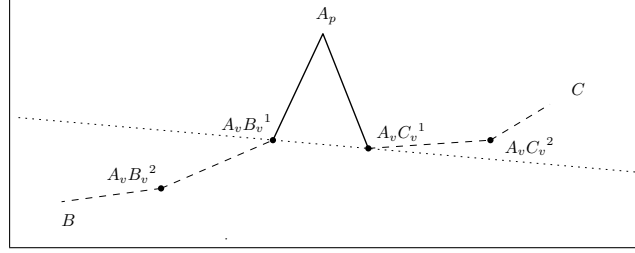


Fig. 3. Both chains $A_P B_P$ and $A_P C_P$ must completely lie on the same side of the line through $A_P B_P^1$ and $A_P C_P^1$.

Theorem 3. *Our algorithm finds at most one candidate vertex for the top joint vertex A out of $\{A_P, B_P, C_P\}$. Moreover, if any one of the joint vertices B_P and C_P be a candidate vertex for A , there can be no more candidate vertex on the convex chain $B_P C_P$ out of B_P and C_P .*

Proof. For the sake of a contradiction, assume that our algorithm finds two candidate vertices A_1 and A_2 for the top joint vertex A out of $\{A_P, B_P, C_P\}$. As said in the proof of Theorem 2, both these vertices and their corresponding other joint vertices must lie on the convex chain $B_P C_P$ in P . W.l.o.g, assume that A_2 lies between B_P and A_1 on this convex chain (See Fig. 4). From the definition of joint vertices B and C and conditions for the top joint vertex, A_2 and $A_2 B_P^2$ must be invisible and $B_P A_P^1$ and $B_P C_P^1$ must be visible pairs. This forces that there must be at least one vertex between B_P and A_2 which means that $A_2 B_P^1$ can not be equal to B_P . While A_2 and $A_2 B_P^2$ are an invisible pair on the convex chain of P , there must be a blocking vertex b on $A_P B_P$ or $A_P C_P$ chains preventing their visibility. Clearly, b must be visible to A_2 . On the other hand, both of the corresponding joint vertices of the top joint vertex A_1 (according to our algorithm) lies between vertices A_2 and C_P on the convex chain of P . This implies that all vertices of chain $B_P C_P$ from A_2 to B_P and vertices of the chains $A_P B_P$ and $A_P C_P$ in P lies on the convex chain of the candidate top joint vertex A_1 . From Observation 3, when two vertices A_2 and b on this convex chain see each other, all vertices from A_2 to b , including $A_2 B_P^2$, must also see each other and form a clique which is a contradiction.

By the same argument we can prove that if the joint vertex C_P (or B_P) be a candidate for the top joint vertex, there cannot be any candidate vertex for the top joint vertex on the convex chain $B_P C_P$ out of $\{B_P, C_P\}$. \square

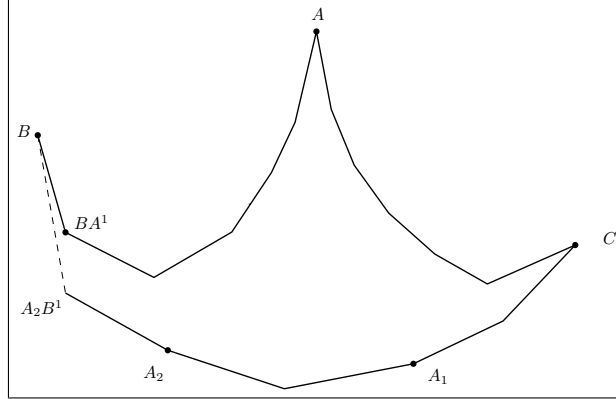


Fig. 4. A_2 and A_2B^2 must be invisible and A_2B^1 and A_2C^1 must be visible pairs.

From the above theorems, we conclude that according to our algorithm there will be at most three candidates for the top joint vertex A . Precisely, if there was any other candidate other than A , either it is a vertex A' on $B_P C_P$ ($A' \notin \{B_P, C_P\}$) or we have at most two candidates from B and C . From observations 3 and 6 we can conclude that for the latter case the candidate vertices does not see any of the vertices of convex chain $B_P C_P$ except the adjacent one in the Hamiltonian cycle.

4 Reconstruction Algorithm

In this section, we assume that we are given a pair of visibility graph, $G(V, E)$ and Hamiltonian cycle, H , and three joint vertices A , B and C and the goal is to obtain an anchor polygon $G(V, E)$ corresponding to these graph and cycle with A , B and C as its top, left and right joint vertices, respectively. Moreover, we assume that the visibility graph and the joint vertices satisfy conditions described in previous observations and conditions of previous algorithms (the joint vertices have been obtained by the algorithms described in section 3). From previous section, we know that there are at most three options for these joint vertices. Therefore, to solve the recognizing algorithm we may run the following algorithm at most three times and if one of these runs leads to an anchor polygon it will be returned as a solution and if none of them produce a polygon it means that $G(V, E)$ and H do not belong to an anchor polygon.

Our reconstruction algorithm consists of two phases. Initially we decompose the target polygon into at most four regions and then these regions are reconstructed to build the final anchor polygon.

4.1 Anchor polygon decomposition

We define a line d as a bi-tangent line for both chains AB and AC if it passes through vertices M and M' on AB and AC , respectively, and both chains lie

completely on the same side of it (See Fig. 5). From the visibility graph edges we can find such a bi-tangent line: There is no edge from vertices of AM (resp. AM') to vertices of $M'C$ (resp. MB) except the edge MM' . Also, the polygon with boundary AM , AM' and edge MM' is a tower polygon.

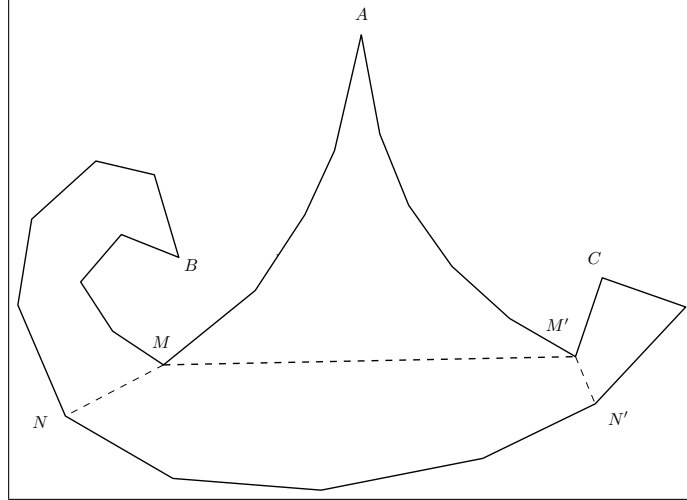


Fig. 5. Decomposition of an anchor polygon

Observation 8 *Each anchor polygon has exactly one bi-tangent line.*

Observation 9 *The bi-tangent line of an anchor polygon passes through its joint vertices B and C if and only if B and C see each other. In these cases, the convex chain BC lies completely on the opposite side of the bi-tangent line compare to A and all of the vertices of this convex chain are visible from each other, and so, they form a clique in the visibility graph.*

Let N and N' be vertices $BC(M')$ and $CB(M)$, respectively (See Fig. 5). As we stated before, the polygon with boundary vertices $\langle M, \dots, B, \dots, M' \rangle$ is a tower polygon and polygon with boundary vertices $\langle M, N, \dots, N', M' \rangle$ is a convex one. Also, both polygons with boundary $\langle M, \dots, B, \dots, N \rangle$ and $\langle M', \dots, C, \dots, N' \rangle$ are spiral polygons and there is no edge between the vertices of one of them to the other one or the tower sub-polygon, except edges have an end point in $\{N, N', M, M'\}$ (See Fig. 5). Otherwise, we report that the pair $G(V, E)$ and H does not correspond to any anchor polygon with joint vertices A, B and C . Note that based of the shape of the anchor polygon, some of these four sub-polygons may not exist (it may be only a point or an edge).

This decomposition of the anchor polygon can be obtained from a given $G(V, E)$ and H and the three joint vertices A, B and C . After obtaining the

bi-tangent line and the tower polygon as discussed above, the vertices N and N' are obtained from $G(V, E)$ and H according to their definition ($N = BC(M')$ and $N' = CB(M)$). Now (after checking the previous observations), because some of the vertices of these spiral sub-polygons may see some of the convex ones, we will extend boundary of them to $\langle M, \dots, B, \dots, N, \dots, N', M' \rangle$ and $\langle M', \dots, C, \dots, N', \dots, N, M \rangle$, respectively (clearly both of them are spiral polygons yet). The visibility graph of any one of these sub-polygons must satisfy the sub-polygon conditions. Precisely, the induced sub-graph of G on vertices of the tower polygon (resp. spiral polygons) must have necessary conditions of the visibility graph of a tower polygon (resp. spiral polygon) with these boundary vertices, and, the induced sub-graph of G on the convex sub-polygon must be a complete graph. Otherwise, we report that the pair $G(V, E)$ and H does not belong to an anchor polygon with the given joint vertices A , B and C .

4.2 Reconstructing sub-polygons

Now, we are ready to propose the final step of our constructive algorithm for solving both recognizing and reconstruction problems. If we consider the union of the tower and convex sub-polygons, in the decomposition phase, as a single polygon, it will be an anchor polygon as well. But, this anchor polygon has this property that its bi-tangent passes through its non-top joint vertices B and C . We call such anchor polygons *simple anchor polygons*. The visibility graph of a simple anchor polygon with joint vertices A , B and C has the following properties.

Observation 10 *Each concave vertex of an anchor polygon sees one continuous sub-chain of the convex chain.*

Observation 11 *The joint vertices B and C of a simple anchor polygon see the whole convex chain.*

Observation 12 *For each concave vertex p of a simple anchor polygon, the vertices of the convex chain which are visible to pA^1 are a subset of the vertices visible from p .*

Observation 13 *If both convex vertices p and pB^1 of a simple anchor polygon lie on the right side of the line through A and AB^1 , the set of visible concave vertices from p is a subset of such set for pB^1 (See Fig. 6.a). Symmetrically, this is true for p and pC^1 if both lie on the left side of the line through A and AC^1 .*

Observation 14 *Assume that $q = AB(p)$ is the closest vertex of the concave chain AB to A which is visible to a convex vertex p on the left side of the line through A to AC^1 in a simple anchor polygon. Then, none of the vertices of the sub-chain from A to $s = AC(q)A^1$ is visible from p and all vertices of the sub-chain from C to $t = AC(qB^1)$ are visible from p (See Fig. 6.b). It means that $AC(p)$ must be one of the vertices of the left concave chain from s to t .*

Symmetrically, for a convex vertex p lying on the right side of the line through A and AB^1 and $q = AC(p)$, $AC(p)$ must be one of the vertices of the right concave chain from $AB(q)A^1$ to $AB(qC^1)$.

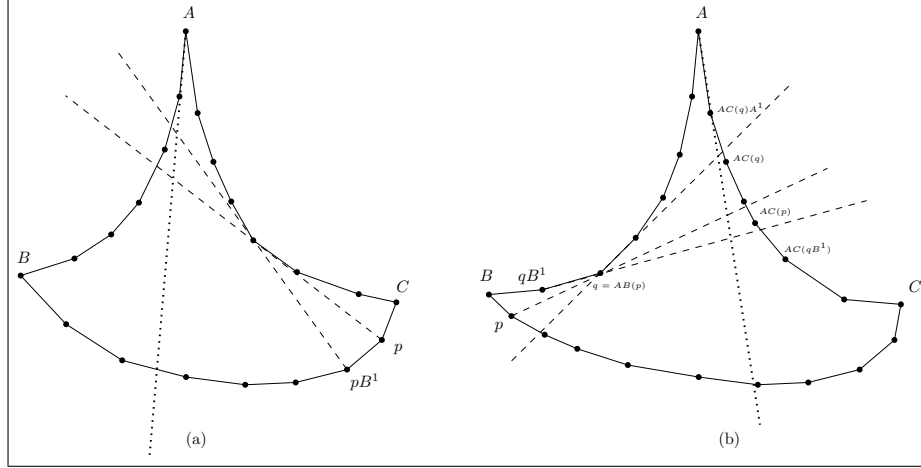


Fig. 6. a) Visible points from p is a subset of the visible points from pB^1 b) Visible and invisible vertices of concave chain AC from p

Trivially all above observations must hold on the visibility graphs induced to the vertices of the tower and convex sub-polygons in our decomposition algorithm presented in Section 4.1. If these conditions hold, we reconstruct simple anchor polygon which corresponds to the obtained tower and convex sub-polygons.

To reconstruct a simple anchor polygon, we first reconstruct the tower polygon using the method presented in [3] (we use the method as a block-box procedure). Then the vertices of the convex chain are put on a convex curve from B to C supporting their order on the Hamiltonian cycle and the visibility graph constraints. To do this, we divide these vertices into these groups: The first group, called V_A , contains those vertices that see all vertices of both concave chains. From the above observations, these vertices must lie on the convex curve between the lines passing through A and AB^1 , and A and AC^1 (See Fig. 7.a). The other groups are the sets V_B and V_C as shown in Fig. 7. To locate an arbitrary vertex $v \in V_B$ it must satisfy two conditions: assume that $p = AB(v)$ and $q = AC(v)$ are respectively, the top most vertices on chains AB and AC which are visible to v . According to the visibility graph constraints, p must lie on the left of the line through p and pA^1 and to the right of the line through p and q . Moreover, v does not see qA^1 and p is a blocking vertex for this invisibility. Therefore, if qA^1 is visible from p then v must lie to the left of the line through p and qA^1 . Otherwise, as v lies to the left of the line through p and pA^1 , the vertex p will block the visibility of v and qA^1 and there is no need to add more constraint

to restrict position of v on the convex curve. Therefore, the intersection of the convex curve and this region must be non-empty. If this happens, we can put p on an arbitrary point of this part of the convex curve and for all points p that must be located in this region, we put them according to their order in Hamiltonian cycle. As the last point of our algorithm, we must show that the intersection of the convex curve and constructed region of v is not empty. The region is restricted to lines d_1 and d_2 or lines d_2 and d_3 (See Fig. 7). It is simple to show that in both cases q is visible from p and in the latter one qA^1 is visible from p and lies above q . This implies that in both cases the region, and consequently, the intersection is not empty. Note that finding the corresponding regions and intersections for vertices of set V_C can be done similarly.

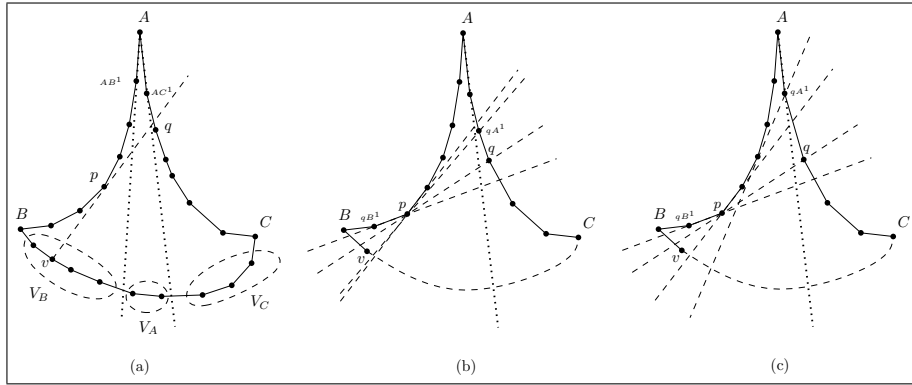


Fig. 7. a) V_A , V_B and V_C b) Invisibility of v and qA^1 needs to be cared c) Invisibility of v and qA^1 do not need to be cared

After reconstructing the simple anchor polygon of the tower and the convex sub-polygons, we must build and attach the spiral sub-polygons to the sides of this simple anchor polygon. Recall that the boundary of the right and left sub-polygons are $\langle M, \dots, B, \dots, N, \dots, N', M' \rangle$ and $\langle M', \dots, C, \dots, N', \dots, N, M \rangle$, respectively. Moreover, remember that in these polygons there is no edge between the vertices of $\langle M, \dots, B, \dots, N \rangle$ and $\langle M', \dots, C, \dots, N' \rangle$ except edges that have an end point in $\{N, N', M, M'\}$. This helps to build these parts independently. Note that we can apply this independency by locating the remained vertices of these spiral polygons above the line through M and M' . We describe how to build the left spiral polygon and the right one can be built Symmetrically. If we apply the elimination scheme starting from the joint vertex B , we find a sequence of removed vertices, which includes all vertices of the left spiral polygon except vertices of the convex sub-polygon. Moreover, the remained vertices of this spiral polygon (which make a convex sub-polygon with respect to the Hamiltonian cycle) are the vertices of our convex sub-polygon $\langle M, M', N', N \rangle$, which is already reconstructed. The spiral polygon reconstruction algorithm de-

scribed in Section 2 says that we can start from an arbitrary convex polygon for the remained vertices and the sequence of the removed vertices can be put in an arbitrary small neighborhood of the only concave vertex (here it is M). This means that, by considering the convex sub-polygon as the starting convex polygon, we can reconstruct the left spiral polygon arbitrary close to M without intersecting the constructed tower polygon. Note that by using this method all vertices of the removed sequence are forced to be located above the line through M and M' .

5 Complexity Analysis

In this section we analyse the time complexity of our algorithm for recognizing and reconstruction of an anchor polygon from its visibility graph and Hamiltonian cycle.

Before beginning the analysis we assume that for each vertex we know its maximal cliques with its previous and successor vertices according to their order in Hamiltonian cycle separately. It means that for each vertex p we know how many vertices consecutively after (resp. before) p will make a clique with it as a number denoted by $C^+(p)$ (resp. $C^-(p)$). We can calculate these numbers in $O(n^2)$ for all the vertices using *Dynamic Programming*[4] where n is number of vertices in the visibility graph. In addition, we denote by $D^+(p)$ (resp. $D^-(p)$) the distance between p and the first visible vertex after (resp. before) its maximal clique, visible to p .

The first part of our algorithm finds candidate vertices. For this purpose our algorithm iterate on each vertex of the visibility graph for checking necessary conditions to be a candidate vertex. For each vertex p checking for visibility of the vertices adjacent to it in Hamiltonian cycle and checking for invisibility of (pB^1, pB^2) and (pC^1, pC^2) (B and C used here to illustrate vertices after and before vertex p) needs $O(1)$ for each vertex and $O(n)$ overally. Then, Finding vertex B_p and C_p will take $O(n)$ time for each vertex p and $O(n^2)$ overally. After finding these vertices, we should check for necessary sight condition between convex vertices which could be done in $O(n)$ for each vertex, using functions C^- , C^+ , D^- and D^+ , and $O(n^2)$ for all vertices. So, we can check necessary conditions for candidate vertices and finding them in $O(n^2)$ and will begin the reconstruction for each of them independently. As the number of candidate vertices are in $O(1)$, the time complexity of the reconstruction algorithm is the time required for one candidate set of A , B and C .

The reconstruction algorithm, take at most $O(n^2)$ for finding the bi-tangent line and then decomposing polygon into a simple anchor polygon and two spiral polygons. After reconstructing the tower polygon of that simple anchor polygon in $O(|E|)$ [3], it will take $O(n^2)$ for checking necessary and sufficient conditions for recognizing simple polygon and reconstructing it. Finally, recognizing each spiral polygon will take $O(n^2)$ using the algorithm of [2]. Summing all, our algorithm will recognize and reconstruct an anchor polygon in $O(n^2)$ time, where n is the number of vertices of the input visibility graph.

References

1. Everett, H, Visibility graph recognition, Phd. Dissertation, University of Toronto (1990)
2. Hazel Everett, Derek G. Corneil: Recognizing Visibility Graphs of Spiral Polygons. *J. Algorithms* 11(1): 1-26 (1990)
3. Paul Colley, Anna Lubiw, Jeremy P. Spinrad: Visibility Graphs of Towers. *Comput. Geom.* 7: 161-172 (1997)
4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduction to Algorithms (3. ed.). MIT Press 2009, ISBN 978-0-262-03384-8, pp. I-XIX, 1-1292
5. Subir Kumar Ghosh: On Recognizing and Characterizing Visibility Graphs of Simple Polygons. *SWAT* 1988: 96-104
6. Subir Kumar Ghosh: On Recognizing and Characterizing Visibility Graphs of Simple Polygons. *Discrete & Computational Geometry* 17(2): 143-162 (1997)
7. Ileana Streinu: Non-stretchable pseudo-visibility graphs. *Comput. Geom.* 31(3): 195-206 (2005)
8. Jürgen Richter-Gebert: Mněvs universality theorem revisited. *Séminaire Lotaringien de Combinatoire* (1995)
9. Subir Kumar Ghosh: Visibility algorithms in the plane. Cambridge University Press 2007, ISBN 0521875749
10. Seth J. Teller, Pat Hanrahan: Global visibility algorithms for illumination computations. *SIGGRAPH* 1993: 239-246
11. Calin Belta, Volkan Isler, George J. Pappas: Discrete abstractions for robot motion planning and control in polygonal environments. *IEEE Transactions on Robotics* 21(5): 864-874 (2005)
12. James Abello, Ömer Egecioglu, Krishna Kumar: Visibility Graphs of Staircase Polygons and the Weak Bruhat Order, I: from Visibility Graphs to Maximal Chains. *Discrete & Computational Geometry* 14(3): 331-358 (1995)
13. Collette R. Coullard, Anna Lubiw: Distance visibility graphs. *Int. J. Comput. Geometry Appl.* 2(4): 349-362 (1992)
14. Hossam Ahmed Elgindy: Hierarchical decomposition of polygons with applications. McGill University (1985).
15. James Abello, Lin Hua, Pisupati Sekhar: On visibility graphs of simple polygons. *Congressus Numerantium*: 119-119 (1992)
16. James Abello, Krishna Kumar: Visibility Graphs of 2-Spiral Polygons (Extended Abstract). *LATIN* 1995: 1-15
17. James Abello, Krishna Kumar: Visibility Graphs and Oriented Matroids. *Graph Drawing* 1994: 147-158

6 Appendix

6.1 Proof of Theorem 1

Proof. Trivially, the theorem is true when B and C are the joint vertices of the target polygon of G and H . Moreover, the theorem is trivially true when there is no anchor polygon with visibility graph G and Hamiltonian cycle H . therefore, it is enough to prove the theorem for the cases where G and H belong to an anchor polygon P with joint vertex A and at least one of the other joint vertices of P is not in $\{B, C\}$. According to Observation. 5, assume that C is a joint vertex of P and the other joint vertex is another vertex $B' \neq B$. This means that B' lies on the convex chain of P , and equal to a vertex BC^i where $i > 0$, while $B' = BC^0$ in anchor polygon P . We prove the theorem by induction on i . For $i = 1$, it means that when BA^1 and BC^1 do not see each other in P , we can consider the joint vertex B as a vertex of the left concave chain of P and considering BC^1 as the left joint vertex, and the visibility graph of this anchor polygon is still equivalent to G . This has been shown in Fig. 9 where (a) is the original polygon and (b) is the new one with BC^1 as a joint vertex.

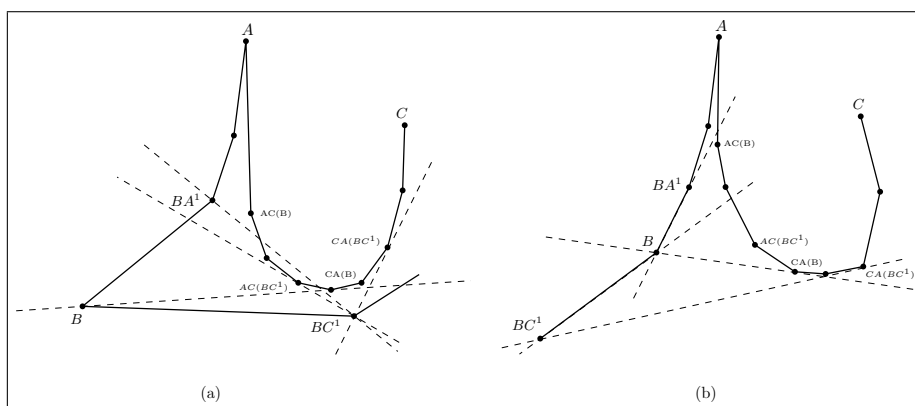


Fig. 8. Considering BC^1 as a joint vertex in (b) while BC^1 and BA^1 are invisible in (a).

To complete the proof we must show that it is always possible to locate BC^1 on the left concave chain of some anchor polygon without disturbing the visibility graph constraints. This is done by first proving that the induced visibility graph on vertices $U = \langle A, AB^1, \dots, B, BC^1 \rangle$ and $W = \langle A, AC^1, \dots, CA(BC^1) \rangle$ have strong ordering with partitions U and W (Then, we can build a tower polygon on these vertices in which B is a concave vertex on the left chain and BC^1 is the vertex on this concave chain) and second, proving that it is possible to place other vertices $\langle BC^2, \dots, C \rangle$ and $\langle CA(BC^1), \dots, C \rangle$ supplying the visibility graph edges.

For the first one, we know from P that the induced visibility graph on $U' = \langle A, AB^1, \dots, B \rangle$ and $W' = \langle A, AC^1, \dots, CA(B) \rangle$ have strong ordering. Then it is enough to consider the pairs (BC^1, w) and (u, w') , where $w, w' \in W$ and $w <^W w'$, w is visible from BC^1 , and $u \in U$ is visible from w' . As shown in Fig. 8 if BC^1 sees w and w' is further than w from A , then u' must also be visible from BC^1 , which means that (BC^1, w') exists in the graph. On the other hand, while BC^1 and BA^1 are invisible from each other, u can only be the vertex B . Then, if $u = B$ sees w' it will also see all closer vertices to A than w' which are visible from BC^1 . This complete the existence of the strong ordering on U and W .

For the second one, we show that after building the tower polygon on U and W , we can add remained vertices to find an anchor polygon with A, BC^1 and C as its joint vertices with the same visibility graph as P . These remaining vertices are the vertices of the right concave chain of P from C to $CA(BA^1)$ and the vertices of convex chain from C to BC^1 . Later we will add the first set as a concave chain of a spiral polygon starts from $CABC^1$ and above the line passing from vertices BC^1 and $CA(BC^1)$. This is consistent with the visibility graph because none of these vertices see anyone of the vertices of the tower (See Fig. 8.b). The convex vertices from C to $BC^1 = B'$ are divided into three parts:

- (V1) The visible vertices from B
- (V2) The visible vertices from B' but not visible to B
- (V3) The invisible vertices from B'

It is simple to check that none of these vertices are visible from BA^1 and all of them visible from B are also visible from B' . According to Fig. 9, assume that d_1 is the line from BA^1 and $CA(BA^1)$, d_2 is the line from B to $CA(B)$ and d_3 is the line from B' to $CA(B')$. The vertices of $V1 \cup V2 \cup V3$ are put on a convex chain from B' to C in such a way that vertices of $V1$ lie inside α , vertices of $V2$ lie inside β and vertices of $V3$ lie above d_3 . As the slope of d_3 is more than d_2 and slope of d_2 is more than d_1 so both α and β are non-empty. Therefore, by considering these constraints, our placement supply edges between vertices of left chain and convex vertices in visibility graph. So we have to consider some constraints to supply edges between right chain and convex vertices. We know that all vertices of the right chain between A and $AC(BA^1)$, are invisible to convex chain (Because $AC(BA^1)$ blocks visibility of B' and BA^1). So, by placing convex vertices above d_1 (includes both regions α and β) these invisibilities can be supplied and we do not need to add more constraints for it. Moreover, Any convex vertex which are visible to B' (equivalently $V1 \cup V2$) sees $CA(B')$ (because all vertices of the right concave chain are above the line through B' and $CA(B')$ and can not block the visibility of them). Therefore, for any convex vertex $p \in V1 \cup V2$, p sees all vertices of vertices of the right chain between $CA(B')$ and $CA(p)$. Moreover, $CA(p)$ blocks the visibility of p and any vertex of vertices of the right chain between A and $CA(p)$. Hence, p must be placed somewhere above the line, d_p , through $CA(p)A^1$ and $CA(p)$ (See Fig. 9). For any $p' \in V1$ (resp. $p' \in V2$), which is closer to C than p , slope of the

line through $CA(p')A^1$ and $CA(p')$ is more than d_p and d_1 (resp. d_2) and less than d_2 (resp. d_3). Moreover all these lines will intersect each other above d_3 . Consider an arbitrary convex curve from B' , that lies completely below d_3 and intersect d_1 , d_2 and $d_{CB(B')}$ strictly below d_3 . So, By considering all constraints above, we can place all vertices of the set $V1 \cup V2$ on this curve, so that, they supply edges of the induced visibility graph on vertices of the sub-chain from A to $CA(B')$ and vertices of $V1 \cup V2$, and form a convex chain with respect to their order in the Hamiltonian cycle.

The convex vertices from B' to C and the concave vertices from $CA(B')$ to C build a spiral polygon and we have already build a convex sub-polygon from it with boundary vertices $\{B', CA(B')\} \cup V1 \cup V2$ (with B' as one of its joint vertices, and other vertices as all neighbors of B'). So the remaining vertices of this spiral polygon (equivalently, the remaining vertices of the anchor polygon) can be reconstructed according to the method described in Section 2 in a close neighborhood of $CA(B')$, so that its boundary do not intersect the rest of the anchor polygon. Note that this reconstruction forces the vertices of $V3$ to be located above d_3 . This completes our induction proof for $i = 1$.

Now we can prove the induction step of our proof. From the proof of the base step of the induction, we conclude that for any anchor polygon P , with pair of $G(V, E)$ as its visibility graph, H as its Hamiltonian cycle and A , B and C as its top, left and right joint vertices, respectively, and BC^i as the left joint vertex found by our algorithm, we have another anchor polygon P' such that, its visibility graph and Hamiltonian cycle are isomorphic to $G(V, E)$ and H , respectively, and A , $B' = BC^1$ and C as its top, left and right joint vertices, respectively. So in P' our algorithm will find $B'C^{i-1}$ as its left joint vertex. So, by repeating this step, there will be a polygon P'' with visibility graph and Hamiltonian cycle isomorphic to $G(V, E)$ and H , respectively and A , BC^i and C as its top, left and right joint vertices, respectively. So the theorem is true. \square

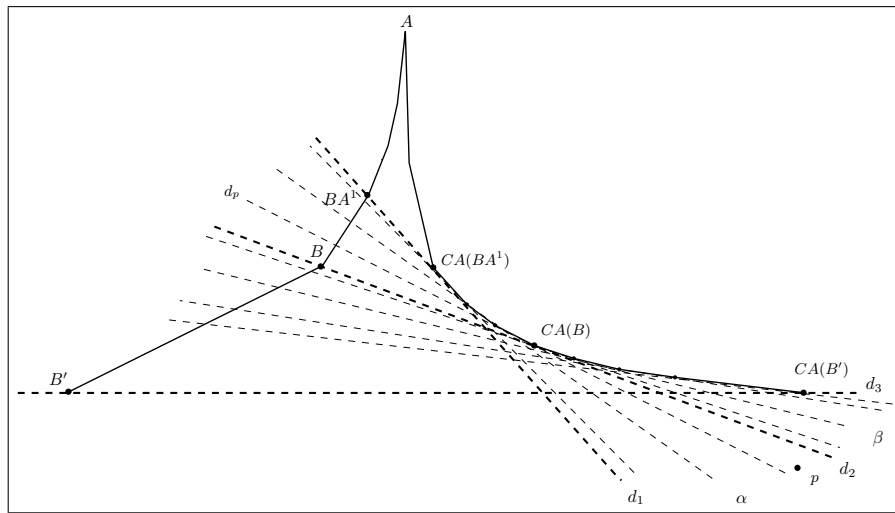


Fig. 9. Lines d_1 and d_2 , parts α and β and other constraints