Incremental Convex Hull as an Orientation to Solving the Shortest Path Problem

Phan Thanh An and Tran Van Hoai

Abstract—The following problem is very classical in motion planning: Let \( a \) and \( b \) be two vertices of a polygon and \( P \) (\( Q \), respectively) be the polyline formed by vertices of the polygon from \( a \) to \( b \) (from \( b \) to \( a \), respectively) in counterclockwise order. We find the Euclidean shortest path in the polygon between \( a \) and \( b \). In this paper, an efficient algorithm based on incremental convex hulls is presented. Under some assumption, the shortest path consists of some extreme vertices of the convex hulls of subpolylines of \( P \) (\( Q \), respectively), first to start from \( a \), advancing by vertices of \( P \), then by vertices of \( Q \), alternating until the vertex \( b \) is reached. Each such convex hull is delivered from the incremental convex hull algorithm for a subpolyline of \( P \) (\( Q \), respectively) just before reaching \( Q \) (\( P \), respectively). Unlike known algorithms, our algorithm does not rely upon triangulation and graph theory. The algorithm is implemented by a C code then is illustrated by some numerical examples. Therefore, incremental convex hull is an orientation to determine the shortest path. This approach provides a contribution to the solution of the open question raised by J. S. B. Mitchell in J. R. Sack and J. Urrutia, eds, Handbook of Computational Geometry, Elsevier Science B. V., 2000, p. 642.

Index Terms—Motion planning, Euclidean shortest path, convex hull algorithm, convex hull.

I. INTRODUCTION

The problem to determine the Euclidean shortest path between two points in a simple polygon is very classical in motion planning. To date, all methods for solving this problem, as presented in [1], [2], [3], etc, rely on starting with a rather complicated, but linear-time triangulation of a simple polygon. This leads to the open question below raised by J. S. B. Mitchell in [3]: “Can one devise a simple \( O(n) \) time algorithm for computing the shortest path between two points in a simple polygon (with \( n \) vertices), without resorting to a (complicated) linear-time triangulation algorithm?”

In 1987, the Steiner’s problem of finding the inpolygon of a given convex polygon with minimal circumference was solved completely by the method of orienting curves [4]. In 2008, the method was used to determine the convex hull of a finite set of points in the plane [5]. Efficient algorithms for determining convex ropes in robotics (for determining convex hulls, respectively) were introduced in [6] and [7] ([8], respectively). These problems are variations of the shortest path problem and thus can be solved without resorting to a linear-time triangulation algorithm and without resorting to graph theory.

Geometrically, we determine the shortest path connecting two points \( a \) and \( b \) that avoids the obstacles - polylines \( P \) and \( Q \). Assume without loss of generality that \( a \) and \( b \) are the first and the final vertices of \( P \) and \( Q \), respectively. In this paper, an \( O(|P||Q|) \) time algorithm for determining the shortest path, without resorting to a linear-time triangulation algorithm and without resorting to graph theory, is presented, using the method of incremental convex hull, where \( |P| \) (\( |Q| \), respectively) is the number of vertices of \( P \) (\( Q \), respectively). Under an assumption on links to \( P \) and \( Q \), the shortest path consists of the extreme vertices of the convex hulls downward, first advancing on one convex hull formed by vertices of \( P \) including \( a \), then on the other formed by vertices of \( Q \), alternating until the vertex \( b \) is reached. Each such convex hull is delivered from the incremental convex hull algorithm for a subpolyline of \( P \) (\( Q \), respectively) just before reaching \( Q \) (\( P \), respectively). Therefore, incremental convex hull is an orientation to determine the shortest path. The algorithm is implemented by a C code and is illustrated by some numerical examples. This paper also provides a contribution to the solution of the Mitchell’s open question above.

II. PRELIMINARIES

For a simple polyline \( X=x_{u_0}u_1\ldots u_l \), \( [u_iu_j] \) is called the first edge, \( u_0 \) and \( u_l \) are called the first and the final vertex of the polyline \( X \), respectively. If \( i \leq j \) then we say \( u_i \) is before \( u_j \) (\( u_j \) is after \( u_i \)) in the polyline \( X \). \( u_{i+1} \) (\( u_{i+1} \), respectively) is the next vertex of \( u_i \) (\( u_i \), respectively) and \( u_{j-1} \) (\( u_{j-1} \), respectively) is the previous vertex of \( u_j \) (\( u_j \), respectively). \( u \) is an extreme vertex of the convex set \( M \) if \( u \in [p,q] \subseteq M \) implies \( u=p \) or \( u=q \).
A simple polygon is represented by two polylines of vertices $P$ and $Q$ such that the first vertices (final vertices, respectively) of $P$ and $Q$ coincide. Two adjacent vertices define an edge of the polygon. Assume that the polygon is given in an orientation as follows: the interior of the polygon lies to the right (left, respectively) as the edges are traversed in the given order of $P$ ($Q$, respectively) (see Fig. 1). Then we say $P$ ($Q$, respectively) is a counterclockwise (clockwise, respectively) polyline. Assume without loss of generality that $a$ and $b$ respectively are the first and the final of $P$ and $Q$. Henceforth, $PQ$ denotes the polygon. We can say that the polygon is formed by polylines $P$ and $Q$ at their first and final vertices $a$ and $b$, respectively. Furthermore, we assume that the vertices $u_j$ of $PQ$ are supposed to be in general position (no three collinear).

A. Incremental Strategy for Finding Convex Hulls

Many algorithms for determining the convex hull of the polyline $X = \langle u_0, u_1, \ldots, u_n \rangle$ use a basic incremental strategy. At the $j$-th stage, they have constructed the convex hull $H_{j-1}$ of the first $j$ vertices $u_0, u_1, \ldots, u_{j-1}$ of $X$, incrementally add the next vertex $u_j$ and then compute the next convex hull $H_j$ (see [9]).

Most convex hull algorithms construct $H_j$ from $H_{j-1}$ in a similar manner. Namely, they find the right and left tangents from $u_j$ to $H_{j-1}$, say $[u_j, u^r_j]$ and $[u_j, u^l_j]$, respectively and use these as new edges for $H_j$ in case $u_j \notin H_{j-1}$. From now, $u^r_j$ ($u^l_j$, respectively) is labelled $\overline{u}_j$ if $X$ is counterclockwise (clockwise, respectively) and we will use the phrase “convex hull” to mean “the set of extreme points of the convex hull”.

Many algorithms for determining the convex hull of the polyline $X=\langle u_0, u_1, \ldots, u_n \rangle$ use a basic incremental strategy. At the $j$-th stage, they have constructed the convex hull $H_{j-1}$ of the first $j$ vertices $u_0, u_1, \ldots, u_{j-1}$ of $X$, incrementally add the next vertex $u_j$ and then compute the next convex hull $H_j$ (see [9]).

Most convex hull algorithms construct $H_j$ from $H_{j-1}$ in a similar manner. Namely, they find the right and left tangents from $u_j$ to $H_{j-1}$, say $[u_j, u^r_j]$ and $[u_j, u^l_j]$, respectively and use these as new edges for $H_j$ in case $u_j \notin H_{j-1}$. From now, $u^r_j$ ($u^l_j$, respectively) is labelled $\overline{u}_j$ if $X$ is counterclockwise (clockwise, respectively) and we will use the phrase “convex hull” to mean “the set of extreme points of the convex hull”.

If there is no vertex of $Y$ in $H_{j-1}$ and there is some vertex $v_j$ of $Y$ in $H_j$ ($H_{j-1}$) (see Fig. 3) then $\text{conv}X$ is said to be firstly intersected by $v_j \in Y$ between vertices $u_j$ and $u_i$ (for short, $\text{conv}X$ is firstly intersected by $Y$ between $u_i$ and $u_j$). Hence, there is at least the vertex $v_j$ of $Y$ trapped in the domain bounded by rays $u^l_j, u^l_{j-1}$ and the path formed by the convex hull $H_{j-1}$ (before pushing $u_j$ on it). Let $Y^*$ be the set (polylines) of vertices in the domain bounded by rays $u^l_j, u^l_{j-1}, u^r_{j-1}$ and the path formed by the convex hull $H_{j-1}$ ($Y^*$ follows the order of $Y$). Let $u^*$ be a vertex of the polyline (formed by $H_{j-1}$) from $u^l_{j-1}$ to $u^r_{j-1}$ (denoted by $X^*$) and $v^* \in Y^*$ such that apart from $u^*$, all vertices of $X^*$ are left (right, respectively) to the line $u^*v^*$. In addition, if there is no vertex of $Y^*$ inside the shaded area $F(u^*, X^*, Y^*)$ formed by the rays $u^*v^*$, $u^*_j, u^r_{j-1}$ and the subpolyline $\langle u^*, u^*, \ldots, u^* \rangle$ of $X^*$ then $[u^*, v^*]$ is called a link to $X^*$ and $Y^*$. $[u^*, v^*]$ is also called a link to $X$ and $Y$. $u^*$ and $v^*$ are referred to as link points.

B. Links and Tangent Polylines

We assume that $Y$ is a clockwise (counterclockwise, respectively) polyline forming a simple polyline such that this polyline does not intersect the polyline formed by $X$ and the final vertices of $X$ and $Y$ coincide.

If there is no vertex of $Y$ in $H_j$ for all $i \geq j$ (see Fig. 2) then $\text{conv}X$ is said to be not intersected by $Y$ before or at the vertex $u_j$.

We define the counterclockwise (clockwise, respectively) tangent polyline $\mathbf{TP}(X)$ of the counterclockwise (clockwise, respectively) polyline $X$. $\mathbf{TP}(X)$ is a stack and it allows one to "push" or "pop" on the top of the tangent polyline during the incremental strategy in finding the convex hull of $X$.

1) Firstly, $\mathbf{TP}(X) = \langle u_0, u_1 \rangle$.

2) Let the left tangent (right tangent, respectively) from $u_j$ to $H_{j-1}$ be $[u_j, u_j^r] \ (\leq j)$. For each $u_j$, if $\text{conv}X$ is not intersected by $Y$ before or at $u_j$ then vertices of $\mathbf{TP}(X)$ after $u_j$ are popped and $u_j$ is pushed on $\mathbf{TP}(X)$.

3) For some vertex $u_j$ of $X$, $\text{conv}X$ is firstly intersected by $Y$ between $u_{j-1}$ and $u_j$. Let the link to $X^*$ and $Y^*$ be $[u^*, v^*]$ (to find such link will be presented in Section 3.1). Then, all vertices of $\mathbf{TP}(X)$ after $u^*$ are popped (i.e., if $\overline{u}_j = u_j^l$ ($u_j^r$, respectively) all vertices of the tangent polyline between $u^*j$ ($u^r_j$, respectively) and $u^*$ are popped and therefore $u^*$ is the final vertex of $\mathbf{TP}(X)$.

Assume that $u_0 \in \mathbf{TP}(X)$. For simplicity, we consider the left tangents, counterclockwise $X$ and clockwise $Y$ case only. The right tangents, clockwise $X$ and counterclockwise $Y$ case is considered similarly. The first vertex $u_0$ of $X$ is an extreme point of $\text{conv}X$.

In Fig. 1, $[p_2, p_3]$ is the link to $X=P$ and $Y=Q$, $[q_2, q_3]$ is the link to $X=q_0, \ldots, q_1 \rangle$ and $Y=q_2, \ldots, q_p \rangle (q_0=p_0)$.

III. THE ALGORITHM

We denote $|X|$ the number of vertices of $X$ and $<X, Y>$ the polyline delivered from a vertex $x$ and a polyline $X$ ($x$ not belong to $X$) such that $x$ is before every vertex of $X$. For $x$ in $X$, $X_x$ is delivered from $X$ by discarding vertices before $x$. Therefore, if $X=<u_0, u_1, \ldots, u_n>$ then $X_x=<u_0, u_1, \ldots, u_i>$ and $<x, X_x>=<x, u_i, u_{i+1}, \ldots, u_n>$.

We need the following procedure.

A. Procedure $\mathbf{TPL}(X, Y)$ ($\mathbf{TPL}$ Stands for Tangent Polyline and Link)

Given counterclockwise polyline $X=<u_0, u_1, \ldots, u_n>$ and clockwise polyline $Y$ such that the final vertex $u_0$ of $X$ and the final vertex of $Y$ coincide and $u_0$ is an extreme point of $\text{conv}X$.

$\mathbf{TPL}(X, Y)$ finds the tangent polyline $\mathbf{TP}(X)$ and the link $[u^*, v^*]$ ($v^*$, respectively) is the link point of the polyline $X$ ($Y$, respectively). It takes $|X||Y|$ time.
Fig. 3. \( H = \text{conv} X \) is firstly intersected by \( Y \) between \( u_{j-1} \) and \( u_j \). \( [u^*,v^*] \) is a link to \( X^* \) and \( Y^* \) and is called the link to \( X \) and \( Y \).

We begin at \( u_0 \). At any given stage of the incremental strategy given in Sec. II for determining the convex hull of \( X \), we step along \( X \), examining vertices \( u_i \) of the polyline in the order they appear along the polyline. Vertices \( \tilde{u}_i = u_i^L \) and \( u_j \) defined by the left tangents are pushed on the tangent polyline TP\((X)\).

The tangent polyline TP\((X)\) is determined by Section 2.2 a), b) and c)), where convX is firstly intersected by \( v_j \) in \( Y \) between vertices \( u_{j-1} \) and \( u_j \) (see Fig. 3). To do so, we determine if there is some vertex of \( Y \) inside the triangle \( \tilde{u}_i u_j u_{j-1} \) or on \([u_i \tilde{u}_i]\) which takes \(|Y|\) time. Therefore, the processing necessary to determine if convX is firstly intersected by \( Y \) between vertices \( u_{j-1} \) and \( u_j \) takes \(|X||Y|\) time in the worst case.

Take \( u^* \in X^* \) and \( v^* \in Y^* \). Our search to find whenever \([u^*,v^*] \) is the link to \( X^* \) and \( Y^* \) has to move on both \( X^* \) and \( Y^* \). Note that there is some vertex of \( Y^* \) inside \( F(u^*,X^*,Y^*) \) iff there is some vertex of \( Y^* \) inside the area formed by rays \( u^*u_j \) and \( u^*v^* \). Then this procedure can take \( O(|X^*||Y^*|) \) time.

Since the optimal incremental algorithm takes \( O(|X|) \) time, TP\((X)\) and \([u^*,v^*]\) are constructed in \( O(|X|) + O(|X||Y|) + O(|X^*||Y^*|) \) time. Hence, TPL\((X,Y)\) takes \( O(|X||Y|) \) time. In fact, vertices between \( \tilde{u}_j \) and \( u^* \) on the hull-so-far of \( X \) without any vertex of \( Y \) inside is maintained in TP\((X)\). Moreover, if \( v^* \) coincides with the final vertex \( u_l \) of \( X \) and \( Y \) then \( v^* \) and vertices of TP\((X)\) are extreme vertices of convX.

B. Main Algorithm

The algorithm constructs the shortest path, \( Z \), between \( a \) and \( b \) in the simple polygon \( PQ \) which consists of the set of tangent polylines formed by vertices of \( P=\langle a=p_0,p_1,...,p_l=b \rangle \) and \( Q=\langle a=q_0,q_1,...,q_l=b \rangle \) (i.e., the set of some extreme edges of the convex hulls of subpolylines of \( P \) and \( Q \)) and links between these subpolylines. We also use \( Z \) to label the polyline formed by ordered vertices of the path \( Z \). Denote \( V \) a subpolyline of \( P \) (or \( Q \), respectively) and \( V' \) a subpolyline of \( Q \) (or \( P \), respectively). If \([u^*,v^*]\) is a link to \( P \) and \( Q \), assume that \( v^* \) in TPL\((V,a)\) for each subpolyline \( V \) of \( P \) (or \( Q \)).

1) Begin at \( a=p_0=q_0 \). Set \( l:=0 \), \( V:=P \), \( u^*=p_0 \) and \( v^*=q_0 \).

2) Call TPL\((V_0,a),V_0,v^*\) to obtain the tangent polyline TP\((V_0)\) and the link \([u^*,v^*]\) to \( V_0 \) and \( V^v_0 \). Let \( Z_0 \) be the path formed by TP\((V_0)\) and \([u^*,v^*]\). If \( v^*=b \), then \( Z:=\bigcup_{j=0}^{j=l} Z_0 \) STOP. Else, set \( l:=l+1 \) and \( V:=V^v_0 \), go to step 2.

In step 1, we choose \( V:=P \). This selection (\( V:=P \) or \( V:=Q \)) is not crucial in the algorithm and does not effect the result. Because each tangent polyline TP\((X)\) consists of extreme vertices of the convex hull convX from \( u_0 \) to \( u^* \). \( Z \) is determined by the extreme vertices of the convex hulls downw0rd, first advancing on one convex hull formed by vertices of \( P \) including \( a \), then on the other formed by vertices of \( Q \), alternating until the vertex \( b \) is reached (see Fig. 5).

Example 3.1: Consider the simple polygon \( PQ \) in Fig. 1, where \( P=\langle a=p_0,p_1,p_2,p_3,p_4,p_5,p_6,p_7,p_8,p_9=b \rangle \) and \( Q=\langle a=q_0,q_1,q_2,q_3,q_4,q_5,q_6,q_7,q_8,q_9=q_0=b \rangle \). We find the shortest path, \( Z \), between \( a \) and \( b \) in \( PQ \). \( Z \) is determined by left tangent polyline \( <p_0,p_1,p_2> \), right tangent polyline \( <q_4,q_5,q_6> \), left tangent polyline \( <p_7,p_8> \) and 3 links \([p_3,p_4],[q_4,p_5],[p_5,q_6]\). \( Z \) includes the set of the extreme vertices \( p_0,p_1,p_2 \) of the convex hull of \(<p_0,p_1,p_2,p_3,p_4,p_5>\subset P \), the set of the extreme vertices \( q_4,q_5,q_6 \) of the convex hull of \(<q_4,q_5,q_6,q_7,q_8>\subset Q \), and the set of the extreme vertices \( p_7,p_8,p_9 \) of the convex hull of \(<p_7,p_8,p_9>\subset P \) (see Fig. 4).

Fig. 4. The shortest path \( Z \) between \( a \) and \( b \) in the simple polygon \( PQ \) is determined by the set of the extreme vertices of the shaded convex hulls of subpolylines of \( P \) and \( Q \) respectively.

C. Correctness of the Algorithm

Lemma 3.1: Assume that \( Y \) is clockwise and at the \( j \)-th stage in the incremental strategy, the convex hull \( H_j \) of the first \( j \) vertices \( u_0,u_1,...,u_{j-1} \) of clockwise \( X \) is constructed by Section 2.1) and \( u_0 \) in TP\((X)\). Let \([u_j,u_{j-L}] \) and \([u_j,u_{j-R}] \) be the left tangent (right tangent, respectively) from \( u_j \) to \( H_j \) and \( X^* \) be the set of vertices of the counter clockwise convex hull \( H_j \) from \( u_L \) to \( u_R \) and \( u^* \) in \( X^* \) be a link point of a link to \( X \) and \( Y \). Assume that \( u^* \) is the next vertex and \( u^* \) is the previous vertex of \( u^* \) in \( X^* \) and \( Y^* \) is the polyline of vertices of \( Y \) belonging to the domain bounded by the lines \( u^*u^*+u^*-u^* \) and \( u^*u^* \). Then

1) \( u^* \) is an extreme vertex of the convex hull of \([u^*,Y^*]\) in \( P \).

2) The first edge (if viewed from \( u^* \)) of the clock-wise convex hull of \([u^*,Y^*]\) is a link to \( X \) and \( Y \). It follows that a link \([u^*,v^*]\) is the first edge of the right tangent polyline (left tangent polyline, respectively) of the polyline \([u^*,Y^*]) \).

By induction, we conclude that the path, \( Z \), delivered from the algorithm in Section 3.2) determines a set of tangent polylines of subpolylines of \( P \) and \( Q \) and links between these subpolylines.
Proposition 3.1: The path, $Z$, delivered from the algorithm in Section 3.2) determines sets of the extreme vertices of the convex hulls downward, first advancing on one convex hull formed by vertices of $P$ including a, then on the other formed by vertices of $Q$, alternating until the vertex $b$ is reached.

Clearly, the number of these convex hulls is best possible.

Theorem 3.1: The algorithm presented in Section 3.2) computes the shortest path between two vertices $a$ and $b$ in the simple polygon $PQ$ in $O(|P||Q|)$ time.

Proof: First, we prove that $Z$ is the shortest path between two points, $a$ and $b$, in the simple polygon $PQ$. Suppose the shortest path is $Z^*$. Take the links $[u_1,v_1]$ and $[u_2,v_2]$ corresponding to $Z_0$ and $Z_1$, respectively. Then, $Z^*$ intersects with $[u_1,v_1]$ and $[u_2,v_2]$ at some $z_1$ and $z_2$, respectively.

We now consider a new polyline, $Z^*_1$, constructed by $Z_l$ and points $z_1$ and $z_2$ as follows: $z_1$ is inserted on the first position and $z_2$ is pushed on the final position of the polyline $Z_l$. Thus, $Z^*_1 = \langle z_1, v_1 \rangle \cup Z_l \cup \langle u_1, z_2 \rangle$ and therefore the vertices of $Z^*_1$ between $z_1$ and $z_2$ are of $Z_l$.

By Lemma 3.1 b), for the polyline $\langle u_1, v_1 \rangle \cup Z_l$, an interior angle made by a vertex and its previous and next vertices is less than $\pi$. It follows that this property holds true for the polyline $Z^*_1$. Furthermore, the length of $Z^*_1$ between $z_1$ and $z_2$ is bigger or equal the path formed by the polyline, which is the path $Z$ between $z_1$ and $z_2$ (see Fig. 5).

Thus, the path $Z^*_1$ between $z_1$ and $z_2$ coincides with the path $Z$ between $z_1$ and $z_2$ and therefore $Z^*$ coincides with $Z$.

At each step of the algorithm presented in the Section 3, $|V_{zl}|$ and $|V_{zl}'|$ are decreasing. In the next step, $V_:V_{zl}'$. Hence, the algorithm only advances, never backs up, and the number of steps is therefore limited by the number $min\{||P||Q\}$. As shown in Section 3.1, each step takes $O(|V_{zl}'\cup V_{zl}|)$ time. The worst case occurs when $V_{zl}=P$ and $V_{zl}'=Q$ and therefore there is only one link to $P$ and $Q$. This case takes $O(|P||Q|)$ time.

II. IMPLEMENTATION

The algorithm is implemented by a C code, in which the incremental strategy is the Melkman’s convex algorithm. The random simple polygon $PQ$ in case $|P|+|Q|=300$ vertices is computed by RPG’s “X-Monotone” heuristic in [10] in the square of size 1. $a$ ($b$, respectively) is the leftmost vertex, i.e. its $x$ coordinate is minimum (rightmost vertex, i.e. its $x$ coordinate is maximum, respectively). The shortest path is presented in Fig. 6.

![Fig. 6. The shortest path between two vertices $a$ and $b$ in the polygon $PQ$ with the number of random vertices being $|P|+|Q|=300$.](image)

REFERENCES