TR-88-030

FINDING ALL SHORTEST PATH EDGE SEQUENCES ON A CONVEX POLYHEDRON

Finding All Shortest Path Edge Sequences on a Convex Polyhedron

Yie–Huei Hwang[1], Ruei-Chuan Chang[1,2] and Hung–Yi Tu[2]

Abstract

In this paper, the problems of computing the Euclidean shortest path between two points on the surface of a convex polyhedron and finding all shortest path edge sequences are considered. We propose an $O(n^6 \log n)$ algorithm to find *All Shortest Path Edge Sequences*, construct n *Edge Sequence Trees*, and draw out $n(n-1)/2$ *Visibility Relation Diagrams* for a given convex polyhedron. According to these data structures, not only can we enumerate all shortest path edge sequences and draw out all maximal ones, but we can also find the shortest path between any two points lying on edges in $O(k+\log n)$ time where k is the number of edges crossed by the shortest path.

Index Terms —— Shortest Path, Shortest Path Edge Sequence.

1. Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, Republic of China.

2. Institute of Information Science, Academia Sinica, Nankang, Taipei, Taiwan, Republic of China.

1. Introduction

Recent interest in the fields of robotics and industrial automation has prompted the study of *Motion Planning.* One of the basic problems is to determine a continuous path for the motion of a given body in an environment that imposes geometric constraints on the body's motion. In this paper we consider the problem of computing the Euclidean shortest path between two points on the surface of a convex polyhedron P [10]. This problem is also of considerable interest in terrain navigation, where a moving vehicle is bound to move along a surface what could be modeled by a polyhedron (here we treat the vehicle as a single moving point) [4]. The shortest path problem on a convex polyhedron can be formally defined as follows [4]:

Let S be the surface of a given convex polyhedron P, defined by a set of faces, edges, and vertices, with each edge occurring in two faces and two faces intersecting either at a common edge, a vertex, or not at all. A *shortest path* between two points A and B on S is the Euclidean shortest path between points A, B along the surface of P. A *shortest path edge sequence* can be defined as an ordered list of edges of P such that any two adjacent edges share a common face, and such that there exists a shortest path traversing the edges in the list. A shortest path edge sequence is said to be *maximal* iff it is not the subset of any other shortest path edge sequence [8]. If the question is to find the shortest path between two fixed points on S, we call it *Discrete Geodesic Problem.* If only one source point (say A) is fixed and we are asked to build a structure which allows one to find out a shortest path from A to any other query point (say B), it is called *Single–Source Discrete Geodesic Problem.* For the general case, if two query points are allowed to be chosen arbitrarily (both are not fixed) on S, we name it *General Geodesic Problem.* We can also make a restriction on the query domain such that the query points can only be chosen on edges. In this way, it is called *Edge–Point General Geodesic Problem.* The enumeration of all shortest path edge sequences on a convex polyhedron is named *All Shortest Path Edge Sequence Problem.*

The *Discrete Geodesic Problem* and *Single–Source Discrete Geodesic Problem* were first posed in [11], where an $O(n^3 \log n)$ algorithm was given for the case of a convex polyhedron. A subsequent result of Mount [5] has reduced the running time to $O(n^2 \log n)$. Both methods are to find the subdivision on the surface of a given convex polyhedron according to one fixed source point, such that any point in the same region has the same shortest path edge sequence to this source point. After building the subdivision, the shortest path problem can be transformed into a standard point location problem and the shortest path from the fixed source point to a given query point can be computed in time $O(k+\log n)$ where k is the number of edges in the corresponding shortest path edge sequence. For the nonconvex case, O'Rourke, Suri, and Booth gave an $O(n^5)$ algorithm [7]. Subsequently, Mitchell [4] improved this result to $O(n^2 \log n)$ by using the "Continuous Dijkstra" technique. He combined the concepts of the original Dijkstra algorithm for finding shortest paths in a graph [2], and the subdivision method in [11]. In [4], edges of the given polyhedron behave like nodes of a graph, but here the distance from the source to an edge is not the unique value. Instead, Continuous Dijkstra Algorithm uses a function that serves as a label for an interval of the edge. Keeping track of the discrete description of these functions, one can subdivide the edge into regions for which the shortest path to points in the region have the same shortest path edge sequence. This method is a generalization of the algorithm proposed in [11].

Since all of the previous algorithms are inefficient to solve *General Geodesic Problem* or even *Edge–Point General Geodesic Problem*, few papers discuss them [4, 11]. The problem of finding all shortest path edge sequences on a convex polyhedron originated from Sharir [10]. He proposed a method to compute shortest paths in 3–D amidst convex obstacles, whose solutions depend on all shortest path edge sequences of these convex obstacles. Sharir [10] gave an $O(n^8 \log n)$ algorithm to compute these edge sequences for each obstacle. He also provided a bound of $O(n^7)$ on the number of edge sequences [10]. Subsequently, Mount [6] had further reduced this bound to $O(n^4)$ and gave an example to show that it is tight. Recently, Schevon and O'Rourke [8] used a graph–theoretic argument to show that the number of maximal sequences of edges traversed by shortest paths is $\theta(n^3)$. This result also provided an alternate proof that the total number of shortest path edge sequences is $O(n^4)$. In the same paper he also proposed an $O(n^7 \log n \cdot 2^{\alpha(n^2)})$ algorithm to compute all shortest path edge sequences of a convex polyhedron, which improved slightly on Sharir's algorithm.

In this paper we shall propose an $O(n^6 \log n)$ algorithm to compute all shortest path edge sequences of a convex polyhedron, by using a data structure with a size of $O(n^4)$. According to this data structure, not only can we enumerate all shortest path edge sequences and draw out all maximal ones, but we can also find the shortest path between any two points lying on edges in $O(k+\log n)$ time where k is the number of edges crossed by the shortest path. Our approach consists of two major parts. We shall first consider all $O(n^4)$ shortest path edge sequences as n edge sequence trees, and use the property of visibility between points on edges to construct these trees. The second part is that, instead of creating the subdivision on the surface of a convex polyhedron [10, 4], for each edge pair (e_s, e_e) we construct the subdivision on domain $Z = e_s \times e_e$ so that any point (A,B) in the same region has the same shortest path edge sequence from point A to point B on S. This approach is the generalization of Continuous Dijkstra Algorithm in [4] and Slice Algorithm in [10].

This paper is organized as follows. In Section 2, we show that all the shortest path edge sequences can be represented as n edge sequence trees. In Section 3, a data structure, called *Visibility Relation Diagram*, is given to maintain the subdivision of each domain $Z=e_s \times e_e$. In Section 4, we propose an algorithm to find all shortest path edge sequences of a convex polyhedron, and show that it can be accomplished in $O(n^6 \log n)$ time. Concluding remarks are given in Section 5.

2. Tree Representation for All Shortest Path Edge Sequences

Let P be a 3–D convex polyhedron with n edges. For each pair of points (A,B) on the surface of P, we denote the shortest path from A to B as $\pi(A,B)$, and the sequence of edges of P crossed by $\pi(A,B)$ as $\xi(\pi(A,B))$. To solve the *Edge–Point General Geodesic Problem* and generate all shortest path edge sequences, we shall first consider the restricted case in which the starting point A lies on an edge e_s, and the ending point B lies on another edge e_e. These two edges, e_s and e_e are called the starting edge and the ending edge respectively. Since the shortest paths on a convex polyhedron cannot cross any edge more than once [11], we can use the brute force approach to form all of the edge sequences as a permutation tree, and then determine which of these sequences are shortest path edge sequences.

For a convex polyhedron P, an *edge sequence tree* T with starting edge e_s is a tree specifying e_s as the root. Each node N_i in T is an edge of P, denoted as $E(N_i)$. Node N_j is a son of node N_i, if $E(N_j)$ shares a common face with $E(N_i)$ on P and $E(N_j)$ is not an ancestor of $E(N_i)$ in T. The path from root e_s to node N_i, denoted as $ES(N_i)$, is an edge sequence of P. If T' is the tree obtained from deleting some nodes of T, such that for every node N_i of T', $ES(N_i)$ is a shortest path edge sequence, we call T' a *shortest path edge sequence tree.* T' is considered *maximal* iff it can not be extended to form another shortest path edge sequence tree.

For example, the edge sequence tree of a tetrahedorn (see Fig. 1a) with starting edge e_1 is shown in Fig. 1b. Hence, the problem to find all shortest path edge sequences with a fixed starting edge is now reduced to the problem to build a maximal shortest path edge sequence tree with this edge.

Lemma 1. If edge sequence $\xi=(e_1e_2 \ldots e_{n-1}e_n)$ is a shortest path edge sequence, then its subsequence $\xi_1=(e_1e_2 \ldots e_{n-1})$ is also a shortest path edge sequence.

Proof: Since ξ is a shortest path edge sequence, there exist two points, say X and Y, on e_1 and e_n respectively, such that the shortest path $\pi(X,Y)$ crosses ξ. Let Z be the intersection of $\pi(X,Y)$ and e_{n-1}. The subpath of $\pi(X,Y)$ from X to Z then, must be the shortest path between X and Z. Otherwise, the concatenation of $\pi(X,Z)$ and the subpath of $\pi(X,Y)$ between Z and Y would be shorter than $\pi(X,Y)$. Therefore, ξ_1 must be the edge sequence crossed by the shortest path from X to Z. ∎

Lemma 1 implies that once we have found a shortest path edge sequence ξ_1, it is very likely that ξ would be another shortest path edge sequence. Thus, the process to find new shortest path edge sequences can be considered as the "expansion" on edge sequence trees. First, we specify the starting edge e_s as the root of T, and add the edges which share a common face with e_s as the children of the root. Then iteratively select a leaf F_i, whose $ES(F_i)$ is a shortest path edge sequence on P, and add edges sharing a common face with $E(F_i)$ as the children of F_i, until all the shortest path edge sequences are found.

In the process of expansion, we are immediately confronted with two problems: to determine which leaf F_i will lead $ES(F_i)$ to be the shortest path edge sequence; and to decide when to stop expanding the edge sequence tree. To decide whether $ES(F_i)$ is a shortest path edge sequence or not, we use the concept of visibility between points on edges [10]. Some definitions are specified as follows. Let $f_1f_2...f_n$ be a sequence of faces on a convex polyhedron P such that edge e_s (resp. e_e) is on the boundary of f_1 (resp. f_n) and f_i, f_{i+1} be adjacent on edge e_i for $i=1,2,..,n-1$. The planar unfolding of P relative to edge sequence $\xi=(e_s e_1 e_2 ... e_{n-1} e_e)$ is obtained by unfolding these faces, one at a time, about the edges that separate them, until they all lie in the plane containing f_1 (with no two adjacent faces overlapping one another, see Fig. 2) [1]. Two points A and B, on starting edge e_s and ending edge e_e respectively, are *visible* to each other in edge sequence ξ if, after the planar unfolding relative to ξ, the straight line from A to B crosses ξ (if ξ is a set of edge sequences, it means that A and B are visible to each other in at least one of these edge sequences). Let $\pi_\xi(A,B)$ be the straight line segment connecting points A and B in this unfolding. $|\pi_\xi(A,B)|$ denotes its length. During expanding edge sequence tree T, the weight of leaf F_i is defined as follows:

$$
\begin{aligned}
W(F_i) = \min(\{\ |\pi_\xi(A,B)| :\ & (A,B)\in e_s\times e_e,\ ES(F_i)=\xi; \\
& \text{and } \forall\ N_j\in T\backslash\{F_i\},\ E(N_j)=E(F_i), \\
& \text{such that } |\pi_\xi(A,B)|\le|\pi_{\xi'}(A,B)| \text{ where } ES(N_j)=\xi' \ \});
\end{aligned}
$$

if the set in function **min** is empty, $W(F_i)$ is set to be infinite.

A leaf F_i is called *with minimal weight* if no other weights of leaves in T are smaller than $W(F_i)$. Roughly speaking, the weight of leaf F_i can reflect the existence of shortest paths between the points on $E(F_i)$ and the points on e_s in the planar unfolding relative to $ES(F_i)$. When $W(F_i)$ goes infinite, it implies that, for every $(A,B)\in e_s\times E(F_i)$

either the points A and B are invisible to each other, or we have already had a node N_j in the expanding edge sequence tree T such that the shortest path from A to B crossing edge sequence $ES(N_j)$ is shorter than the one crossing $ES(F_i)$. In other words, there are no shortest paths crossing edge sequence $ES(F_i)$.

Lemma 2. [11] If points A and B, on edges e_s and e_e respectively, are not visible in edge sequence ξ, then the shortest path edge sequence between A and B can not be ξ.

Lemma 3. In building edge sequence tree T, if F_i is the leaf with minimal weight, then edge sequence $ES(F_i)$ is a shortest path edge sequence.

Proof: Assume that F_i is the leaf with minimal weight and $ES(F_i)=\xi_i$. There must be a pair of points (A,B) such that $W(F_i)=|\pi_{\xi_i}(A,B)|$. By the definition of weight, $|\pi_{\xi_i}(A,B)|$ is the smallest one for all possible (A,B) in the planar unfolding relative to ξ_i. If $\pi_{\xi_i}(A,B)$ is not the shortest path between A and B along the surface of P, then there exists another leaf F_j (let $ES(F_j)=\xi_j$) such that either ξ_j is the shortest path edge sequence between A and B, or ξ_j is just the subsequence of this shortest path edge sequence. In the latter case, we have $W(F_j)<W(F_i)$. This implies that $W(F_i)$ is not the minimal one. In the former case, we have $|\pi_{\xi_j}(A,B)|<|\pi_{\xi_i}(A,B)|$. $W(F_i)$ should not be $|\pi_{\xi_i}(A,B)|$ (by the definition of weight). Thus, $\pi_{\xi_i}(A,B)$ is the shortest path from A to B and ξ_i is its shortest path edge sequence. ■

From **Lemma 2** and **Lemma 3** we know when to stop expanding edge sequence tree T. If all the leaves in T are with infinite weight, there are no leaves to be expanded. **Lemma 3** also tells us which leaf should be included into the shortest path edge sequence tree, and this chosen leaf is also the next one to be expanded. In order to compute the weight for each leaf and find the minimal one quickly, a data structure called *visibility relation diagram* is used to maintain the visible relations between the points on edges.

3. Visibility Relation Diagrams

In this section we shall describe the structure of *visibility relation diagrams* in details. Let T be the currently expanding edge sequence tree with root e_s. Assume that S is a set of edge sequences in which all edge sequences have the same ending edge e_e. In order to determine whether there are shortest paths crossing the edge sequences in S, by **Lemma 2**, we must show the visible relationships between the points on e_s and the points on e_e in the planar unfoldings relative to the edge sequences of S. Our approach is to consider the 2–D space $Z=e_s \times e_s$ of all possible pairs of starting and ending points, and partition it into regions, such that for each such region R_ξ there exists an edge sequence ξ of S such that, for all $(A,B) \in R_\xi$, $\pi_s(A,B)=\pi_\xi(A,B)$. In other words, not only are the points A, B visible to each other in the planar unfolding relative to ξ, but the straight line segment connecting A and B in ξ is also smaller than the others in edge sequences of $S \backslash \{\xi\}$.

Definition. Assume that $S=\{\xi_1,\xi_2,..,\xi_{n-1},\xi_n\}$ is a set of edge sequences with starting edge e_s and ending edge e_e. Let the function $f: e_s \times e_e \longrightarrow S \cup \{\phi\}$ defined by

$f(A,B)=\xi_i$ iff A and B are visible to each other in ξ_i ,

and $\pi_{\xi_i}(A,B) \leq \pi_{\xi_j}(A,B)$ for all $\xi_j \in S \backslash \{\xi_i\}$;

$f(A,B)=\phi$ iff A and B can not be seen from each other in S.

For a pair of edges, e_s and e_e, a *visibility relation diagram* (short for VRD) restricted to S is a partition on domain $Z=e_s \times e_e$ defined by f. We denote the equivalent class corresponding to ξ_i as R_{ξ_i}.

In the following, we first consider the special case in which S contains only one edge sequence, and then we show how to modify the VRD restricted to S to a new VRD when adding an edge sequence into S. In the remained paragraphs, the method to compute weights of leaves from VRD will be proposed.

Initially, let S contain only one edge sequence $\xi=(e_1e_2..e_{n-1}e_n)$. After performing the planar unfolding relative to ξ, we have a polygon, denoted by $G_1(\xi)$, whose boundary is composed of e_1, e_n, and the edges connecting the end points of e_i, e_{i+1} for $i=1,2,..,n-1$ (see Fig. 3a). By using the algorithm in [3], it is easy to find two shortest paths connecting the end points of e_1 and e_n in $G_1(\xi)$ such that these two paths are not crossed with each other (see Fig. 3b). Hence, these two paths together with e_1 and e_n define another simple polygon $G_2(\xi)$. For the points $A \in e_1$ and $B \in e_n$, if they are visible to each other in ξ, their connecting straight line segment should be contained in $G_2(\xi)$. According to visibility between points on e_1 and on e_n, domain $Z=e_1 \times e_n$ can be partitioned into two equivalent classes R_ξ and R_ϕ : for the point (A,B) in R_ξ, A and B are visible to each other in ξ; if (A,B) is in R_ϕ, they can not be seen from each other in ξ. In order to find the boundary between R_ξ and R_ϕ on domain Z, we should formularize the boundary between these two equivalent classes.

Definition. The *boundary–points* of R_ξ are the points $(A,B) \in Z$ where the straight line segment $\overline{AB}$ in $G_2(\xi)$ contains a vertex of $G_2(\xi)$.

Lemma 4. For a fixed vertex of $G_2(\xi)$ the locus of boundary points is a hyperbolic curve on domain $Z=e_1 \times e_n$.

Proof: Let A and B be points respectively on e_1 and e_n, and **c** be the fixed vertex. Parameterize A and B as $\mathbf{a}_1+\mathbf{b}_1 u$ and $\mathbf{a}_2+\mathbf{b}_2 v$ respectively, for appropriate vectors $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{b}_1$, $\mathbf{b}_2$, and real parameters u, v. Then the condition that A, B, **c** are collinear can be written as

$$(A-\mathbf{c})\times(B-\mathbf{c})=0,$$

i.e.,

$$\begin{aligned} 0 &= (\mathbf{a}_1+\mathbf{b}_1 u-\mathbf{c})\times(\mathbf{a}_2+\mathbf{b}_2 v-\mathbf{c}) \\ &= (\mathbf{a}_1-\mathbf{c})\times(\mathbf{a}_2-\mathbf{c}) + [\mathbf{b}_1\times(\mathbf{a}_2-\mathbf{c})]u + [(\mathbf{a}_1-\mathbf{c})\times\mathbf{b}_2]v + (\mathbf{b}_1\times\mathbf{b}_2)uv, \end{aligned}$$

which is an equation of a hyperbola in u–v space [11]. ■

By **Lemma 4**, each vertex of $G_2(\xi)$ has a corresponding hyperbolic curve. The boundary of R_ξ is composed of these cure segments. For example the boundary defined by the polygon in Fig. 4b is shown in Fig. 4c.

Since we have found the equivalent class corresponding to just one edge sequence, our next goal is to show how to modify an existent VRD to a new VRD when adding a new edge sequence to edge sequence trees. Let $S=\{\xi_1,\xi_2,..,\xi_n\}$ be the set of all edge sequences with the same ending edge e_e in currently expanding edge sequence tree T. Assume that we have had a visibility relation diagram VRD_s restricted to S. Whenever a new node N is generated on T, if E(N) is e_e, we should modify VRD_s to show the existence of ES(N), since it is possible that some paths crossing ES(N) are shorter than the ones crossing the other edge sequences already existing in S. To simplify the notation, let ES(N) be ξ. The modification consists of two steps :

Step (1) partition domain Z into R_ξ and R_ϕ;

Step (2) for all points $(A,B)\in R_\xi \cap R_{\xi_i}$, decide whether the shortest path in the planar unfolding relative to ξ is shorter than the one in the planar unfolding relative to ξ_i (determine whether (A,B) should be classified into R_ξ or R_{ξ_i}).

Step (1) can be accomplished by the previous method. For step (2), we perform two planar unfoldings relative to ξ and ξ_i on a common plane such that they share the common e_s. However, point B on e_e will be duplicated to two points in these two unfoldings, say B_ξ and B_{ξ_i} (see Fig. 5). Let the perpendicular bisector of $\overline{B_\xi B_{\xi_i}}$ intersect e_s at point C. This bisector partitions the plane into two halfplanes. One contains B_ξ while the other contains B_{ξ_i}. If A is in the same halfplane with B_ξ, then $|\overline{AB_\xi}|<|\overline{AB_{\xi_i}}|$. In other words, the path from A to B crossing ξ is shorter than the one crossing ξ_i. Hence (A,B) should be classified to R_ξ. On the contrary, if A and B_{ξ_i} are in the same halfplane, (A,B) belongs to R_{ξ_i}. When A is just located on C, we have $|\overline{AB_\xi}|=|\overline{AB_{\xi_i}}|$. It means that if we move point B on the edge e_e (the position of point C is well defined) the locus of (C,B) can partition $R_\xi \cap R_{\xi_i}$ into two regions, where one should be combined into equivalent class R_ξ, while the other should be included into R_{ξ_i}. We name these points (C,B) the *partition–points*. Hence, this new partition on domain Z, obtained by modifying the original VRD_s, is the visibility relation diagram restricted to $S \cup \{\xi\}$. In the same way mentioned in Lemma 4, it is easy to show that the locus of these partition–points is also a hyperbolic curve on domain Z.

Lemma 5. The locus of the partition–points of $R_\xi \cap R_{\xi_i}$ is a hyperbolic curve on Z.

Proof: To make the proof simple. We follow the previous notations. Let $B_\xi = a + bu$, $B_{\xi_i} = a_i + b_i u$, and $C = c + dv$, for appropriate parameters. Since $\angle B_\xi B_{\xi_i} C = \angle B_{\xi_i} B_\xi C$, we have the following equation,

$$(B_{\xi_i}-B_{\xi})\times(C-B_{\xi}) + (B_{\xi}-B_{\xi_i})\times(C-B_{\xi_i}) = 0,$$

which can be simplified as

$$c_0+c_1u+c_2v+c_3uv+c_4u^2=0,$$

where

$$c_0=[(a_i-a)\times(c-a)]+[(a-a_i)\times(c-a_i)]$$

$$c_1=[(a_i-a)\times d]+[(a-a_i)\times d]$$

$$c_2=[(a_i-a)\times(-b)+(b_i-b)\times(c-a)]+[(a-a_i)\times(-b_i)+(b-b_i)\times(c-a_i)]$$

$$c_3=[(b_i-b)\times d]+[(b-b_i)\times d]$$

$$c_4=[(b_i-b)\times(-b)]+[(b-b_i)\times(-b_i)]$$

This equation is also a hyperbola in u–v space. ■

As mentioned in Section 2, VRD is built to show the visible relation between points on edges, and to compute the *weights* of leaves in expanding T. In order to get the weights of leaves from VRD, we should point out which path makes the edge sequence to be the shortest path edge sequence. Let F be a leaf in T and ES(F)=ξ. A *weight–point* (A,B) of F is a point in R_ξ such that $W(F)=|\pi_\xi(A,B)|$. If R_ξ is empty, F has no weight–points. We define the *boundary* of R_ξ as the union of its boundary–points and partition–points.

Lemma 6. If F is a leaf with non–empty R_ξ, there exits a *weight–point* (A,B) of F, which is located on the *boundary* of R_ξ.

Proof: We prove this lemma by contradiction. Assume that all the weight–points of F are neither boundary–points nor partition points. Let (A,B) be one of these weight–points. By definition of weight–point we can find two points on the boundary of

R_ξ, say (A_1,B_1) and (A_2,B_2), such that $\pi_\xi(A_1,B_1)$ and $\pi_\xi(A_2,B_2)$ are both longer then $\pi_\xi(A,B)$, run parallel with $\pi_\xi(A,B)$, and are on the different sides of $\pi_\xi(A,B)$. But this is contrary to the fact that both A_1, A, A_2 are collinear on starting edge, and B_1, B, B_2 are collinear on ending edge (see Fig. 6). Thus, there must be a weight–point on the boundary of R_ξ. ■

With the same geometric analyses used in **Lemma** 4 and **Lemma 5**, the lengths of $\overline{AB}$ and $\overline{CB}$ can be formulized as hyperbolic functions of parameters u and v, too. Since there exists a weight–point on the boundary, we can compute the weight by differentiating these functions. Hence the visibility relation diagrams not only can show the visibility between edges but also can maintain the weights of nodes during expanding the edge sequence trees.

4. The Algorithm and its Time Complexity

In this section we first formally state the algorithm of finding all shortest path edge sequences on a convex polyhedron, and then analyze its time complexity.

We can describe our algorithm formally as follows:

Algorithm: Finding_All_Shortest-Path–Edge–Sequences (**FAS**)

Input: The data structure representing the convex polyhedron P

Output: Visibility Relation Diagrams and Edge Sequence Trees for All Shortest Path Edge Sequences of P

(1) **FOR** each edge e_i on P, use e_i as the starting edge DO :

(2) Let e_i be the root of edge sequence tree T_i;

```
        FOR each edge e_j sharing a common face with e_i DO :
            Construct the VRD on domain Z=e_i×e_j;
            Let e_j be the son of e_i and compute the weight of e_j;
        END of FOR;
        WHILE there exists a leaf whose weight ≠ ∞ DO :
            Find the leaf F with minimal weight;
            FOR each edge e_k sharing a common face with E(F) DO :
                Construct/Modify the VRD on domain Z=e_i×e_k;
                FOR each leaf F' with E(F')=e_k DO :
                    Compute/Recompute the weight of F' END of FOR;
                Let e_k be the son of F;
            END of FOR;
        END of WHILE;
    END of FOR.
```

The correctness of Algorithm FAS can be shown in the following theorem.

Theorem 1. By **Algorithm FAS**, we can construct a one to one correspondence between the shortest path edge sequences on P and the paths from root to internal nodes in edge sequence trees.

Proof: We prove this theorem by induction. Let i be the length of the edge sequence. For i=1 or 2, the statements are obviously true. Assume that the statements are true for $i \leq n-1$. Let $\xi=(e_1e_2..e_n)$ be a shortest path edge sequence on P. By **Lemma 1**, the edge sequence $\xi_1=(e_1e_2..e_{n-1})$ is also a shortest path edge sequence. By inductive hypothesis, there must exist a node N in edge sequence trees such that $ES(N)=\xi_1$. Since ξ is a shortest path edge sequence, N has at least one son, say F, where $E(F)=e_n$, such that W(F) is NOT infinite (ref. **Lemma 2**, **Lemma 3**, and the definition of ***weight*** in Section 2). This implies that ξ is a path from root to node N in edge sequence trees. The argument is clearly reversible; hence the theorem is proved. ∎

The running time of **Algorithm FAS** depends on

(a) the number of nodes in edge sequence trees,

(b) the number of regions in visibility relation diagrams, and

(c) the time to modify visibility relation diagrams during expanding edge sequence trees.

To see this, we examine each separately.

For (a), Mount [6] and O'Rourke [8] have proved that there are $O(n^3)$ shortest path edge sequences from a fixed starting edge to the other edges on P. This implies that the number of internal nodes in each edge sequence tree can be bound to $O(n^3)$. To simplify the analysis, assume that P is triangulated. By the fact that the shortest path can cross a face only once, each internal node has no more than two children. Hence there are overall $O(n^3)$ nodes (including leaves) in an edge sequence tree. Since we construct n edge sequence trees in **Algorithm FAS**, there are totally $O(n^4)$ nodes in n edge sequence trees.

For (b), to count the number of regions in visibility relation diagrams, we first examine the correspondence between the regions and the shortest path edge sequences.

Lemma 7. There are $O(n^2)$ regions in each visibility relation diagram after performing **Algorithm FAS**.

Proof: To prove this lemma, we show that for any two points $A=(A_s,A_e)$, $B=(B_s,B_e)$ on domain $Z=e_s \times e_e$, if A and B are in the same equivalent class, say R_ξ, then there exists a path $P \subset R_\xi$ connecting A and B on domain Z. In other words, R_ξ is path connected. Without loss of generality, we discuss the following cases separately.

CASE 1: If $A_e=B_e$, $\overline{AB}$ is parallel to e_s on domain Z (see Fig. 7a). We shall claim that $\overline{AB} \subset R_\xi$.

Assume that there exists some point $C=(C_s,A_e)$ on $\overline{AB}$ but belonging to $R_{\xi'}$, where $\xi' \neq \xi$. We first perform planar unfoldings relative to ξ and ξ' on a common plan such that they share the common edge e_s. However, point A_e will be duplicated to two points, say A_ξ and $A_{\xi'}$ (see.Fig. 7b). Let the perpendicular bisector to $\overline{A_\xi A_{\xi'}}$ intersect e_s

at point E. This bisector partitions the plane into two hyperplanes. Since $(C_s,A_e) \in R_{\xi'}$ and $(A_s,A_e) \in R_\xi$, we have $|\pi_\xi(C_s,A_e)| > |\pi_{\xi'}(C_s,A_e)|$ and $|\pi_\xi(A_s,A_e)| < |\pi_{\xi'}(A_s,A_e)|$. This implies that respectively $|\overline{C_sA_\xi}| > |\overline{C_sA_{\xi'}}|$ and $|\overline{A_sA_\xi}| < |\overline{A_sA_{\xi'}}|$ in these planar unfoldings. Hence E must be on $\overline{A_sC_s}$,and B_s is in the same hyperplane with C_s, which means $|\pi_\xi(B_s,A_e)| > |\pi_{\xi'}(B_s,A_e)|$. The shortest path edge sequence from B_s and A_e should not be ξ ,but be ξ'. This contradicts to our assumption, $(B_s,A_e) \in R_\xi$. The case of $A_s=B_s$ can also be derived from, instead of create two e_e, duplicating e_s when performing planar unfoldings.

CASE 2: With same notations, if neither $A_e=B_e$ nor $A_s=B_s$, we have two kinds of planar unfoldings relative to ξ (see Fig. 8). In one case $\pi_\xi(A_s,A_e)$ crosses $\pi_\xi(B_s,B_e)$, while in the other case these two paths are not crossed by each other.

For the former case, if $\pi_\xi(A_s,A_e)$ crosses $\pi_\xi(B_s,B_e)$ at point D (see Fig. 8a), we shall claim that the following curve P is a path connecting point A and B in domain Z and $P \subset R_\xi$.

P is a hyperbolic curve in domain $[A_s,B_s] \times [A_e,B_e]$ such that for every point $(P_s,P_e)\in P$, P_s, D, and P_e are collinear in the planar unfolding relative to ξ.

Since A and B belong to R_ξ, $\overline{A_sA_\xi}$ and $\overline{B_sB_\xi}$ should be included in polygon $G_2(\xi)$. Hence, for all points $(P_s,P_e)\in P$, P_s and P_e can be seen by each other in the planar unfolding relative to ξ. Assume that some point $C\in P$ belongs to $R_{\xi'}$, where $\xi'\neq\xi$ and $C=(C_s,C_e)$. This implies that $|\pi_\xi(C_s,C_e)|>|\pi_{\xi'}(C_s,C_e)|$. On the other hand, since A and B belong to R_ξ, we have $|\pi_\xi(A_s,A_e)|<|\pi_{\xi'}(A_s,A_e)|$ and $|\pi_\xi(B_s,B_e)|<|\pi_{\xi'}(B_s,B_e)|$ respectively. A simple geometric analysis can derive that $|\pi_\xi(C_s,C_e)|<|\pi_{\xi'}(C_s,C_e)|$, which contradicts to our assumption that $C\in R_{\xi'}$. Thus every point $C\in P$ must belong to R_ξ.

For letter one, since both A and B belong to R_ξ, $\overline{A_sA_\xi}$ and $\overline{B_sB_\xi}$ should be included in polygon $G_2(\xi)$. Hence, all points on $\overline{A_sA_\xi}$ and $\overline{B_sB_\xi}$ can be seen by each other in the planar unfolding relative to ξ. Let $A'=(A_s,B_e)$ and $B'=(B_s,A_e)$. In the following, we shall claim that either $\overline{AA'\cup A'B}$ or $\overline{AB'\cup B'B}$ (but not both) belongs to R_ξ.

Assume that A' belongs to $R_{\xi'}$, but $\xi'\neq\xi$. This implies that the shortest path edge sequence from A_s to B_e is ξ'. Thus, we have $|\pi_\xi(A_s,B_e)|>|\pi_{\xi'}(A_s,B_e)|$. On the other hand, since both A and B belong to R_ξ, we have $|\pi_\xi(A_s,A_e)|<|\pi_{\xi'}(A_s,A_e)|$ and $|\pi_\xi(B_s,B_e)|<|\pi_{\xi'}(B_s,B_e)|$ respectively. A simple geometric analysis can derive that either $|\pi_\xi(A_s,B_e)|<|\pi_{\xi'}(A_s,B_e)|$ (see Fig. 8b) or $B'\in R_\xi$, but not both. Here, the former one contradicts to our assumption that $A'\in R_\xi$, while the latter one meets $\overline{AB'\cup B'B}\subset R_\xi$ (by *CASE 1*). The relative statements are also true for assumption $B'\notin R_\xi$.

With the analytical results in *CASE 1* and *CASE 2*, it is not difficult to see R_ξ is path connected. Since the number of equivalent classes on domain Z is the same as the number of shortest path edge sequence, the number of regions on domain Z is bound to $O(n^2)$. Hence **Lemma 7** is true. ■

Our next goal is to show that when processing **Algorithm FAS** the number of regions in each visibility relation diagram is also no more than $O(n^2)$. By **Lemma 7** we have known that each internal node has only one corresponding region, but it is possible that, in building edge sequence trees, we have a leaf (or leaves) whose edge sequence has two (or more) corresponding regions in the visibility relation diagram. For this leaf, it will eventually be either an internal node constituting an edge of some shortest path edge sequence, or a leaf with infinite weight. In the former case, the final internal node will contain only one corresponding region, while the other regions will be overlaid by the regions of other internal nodes. The latter one implies that the path from root to this node is not a shortest path edge sequence. Its equivalent class should be empty and all its corresponding regions will be covered by regions of other edge sequences. Thus, during the entire process of **Algorithm FAS**, the number of regions in each visibility relation diagram will be no more than the number of all shortest path edge sequences from a fixed starting edge to another fixed ending edges. The above discussion can be summarized as the following.

Corollary 8. The number of regions in each visibility relation diagram can be bound to $O(n^2)$ during the whole process of **Algorithm FAS**.

For (c), we shall claim that for each time we expand a node in edge sequence trees it takes $O(n^2 \log n)$ time to modify its corresponding visibility relation diagram. Using the notations in Section 2, the planar unfolding relative to some edge sequence [11] and the construction of polygon $G_2(\xi)$ [3] can both be performed in $O(n \log n)$ time. The construction of all intersection regions $R_{\xi} \cap R_{\xi_i}$ can be accomplished by calculating the intersections between the hyperbolic curves, by sorting these points along each of these curves, and then by tracing the boundary of each intersection region. Since by **Corollary** 8 we know there are at most $O(n^2)$ regions in each visibility relation diagram, it needs overall time $O(n^2\log n)$ [10] to draw out all intersection regions. For each of the resulting $O(n^2)$ intersection regions (at most), we must draw a hyperbolic curve to partition it. Since the planar unfoldings relative to ξ and ξ_i have been put on a common plane, this step takes constant time. To compute the weight for a new node, we must differentiate the boundaries of its corresponding regions. It takes $O(n)$ time. With this information, the next time we modify its weight, we needs only constant time. The above discussions can be summarized as follows. To expand a new node in an edge sequence tree, we spend $O(n\log n)$ time to construct $G_2(\xi)$ and region R_ξ, $O(n^2\log n)$ time to find the intersection regions, and $O(n^2)$ time to modify the visibility relation diagram and compute the weights of leaves. Hence, it takes overall $O(n^2\log n)$ time to expand a new node in the edge sequence tree.

By the analytical results to (a), (b), and (c), we can conclude that **Algorithm FAS** totally takes $O(n^6\log n)$ time to construct n edge sequence trees and $n(n-1)/2$ visibility relation diagrams. Since the visibility relation diagrams show us the visibility between points on edges of P, the problem of finding shortest path edge sequences on P can be reduced to a location problem on VRD's. For a pair of given points (A,B) lying on edges e_s and e_e respectively, we need only $O(\log n)$ time to identify its located region in domain $Z=e_s\times e_e$. Thus, its corresponding shortest path edge sequence can be draw out from edge sequence trees immediately.

By concluding above discussions, we give the following theorem.

Theorem 2. Given a convex polyhedron P with n vertices, one can preprocess P by a procedure which runs in $O(n^6\log n)$ time. This procedure produces n edge sequence trees and $n(n-1)/2$ visibility relation diagrams, in each of which has $O(n^2)$ regions. With the aid of these trees and diagrams one can find the shortest path edge sequence between any two specified points lying on edges in $O(k+\log n)$ time where k is the number of edges in the shortest path edge sequence.

5. Conclusions and Remarks

As mentioned in Schevon and O'Rourke's paper [8], the gap between the number of shortest path edge sequences and the time to compute them can be narrowed. This paper has shown it. We transfer the visible relationship between edges into *Visibility Relation Diagrams*, and organize all shortest path edge sequences into n *Edge Sequence Trees* in overall $O(n^6\log n)$ time. This is a new approach in finding all shortest path edge sequences. It is different from Sharir's [10] or Mount's [5] methods, in which they partitioned the surface of a polyhedron into slices. Hence the running time can be reduced. It seems quite likely that the algorithm developed in this paper is much closer to the optimal one, as there are $O(n^4)$ shortest path edge sequences on the polyhedron, and for given two points, without preprocessing, one needs $O(n^2\log n)$ time to find their shortest path edge sequence (the best method up to now). We expect that the time complexity could be reduced to $O(n^6)$ by using some better data structures to maintain visibility relation diagrams. Keeping the ordering of the boundary of each region during constructing visibility relation diagrams could be another approach to reduce the time bound. The data structure of the visibility relation diagram may be of interest in its own right.

The method we used in this paper is a generalization of the continuous Dijkstra technique in [4]. In [4], the Continuous Dijkstra technique was limited to discuss the relationship between a fix point p and the points on each edge e or face f. In our term, these relationships can be characterized as the visibility relation diagrams on domain $Z=p\times e$ or $Z=p\times f$ respectively. For example, It is easy to understand that *Single–Source Discrete Geodesic Problem* can be looked at as a special case of *Edge–Point General Geodesic Problem.* In Algorithm FAS, we simply initialize original starting edge to be a single point, and then proceed exactly as before to construct the visibility relation diagram for each edge. Obviously, each of these visibility relation diagrams is a partition on the corresponding edge. This is actually what Mitchell has done in [4].

We believe that the generalized Continuous Dijkstra algorithm can also be applied to *General Geodesic Problem,* which is important in the study of robotics and terrain navigation. But in the generalization from $Z=e_i\times e_j$ to $Z=f_m\times f_n$, the process to partition Z into equivalent regions will be more complex. It obviously includes a subproblem which is the dynamic point location problem in 4–D. Hence, whether we can develop a good algorithm for this generalized case crucially depends on the method to solve the dynamic point location problem in 4–D.

Acknowledgements

We thank Joseph O'Rourke for his helpful papers, and Micha Sharir for his suggestions and comments.

References

[1] A. Baltson, M. Sharir, On Shortest Paths between Two Convex Polyhedra, *JACM*, Vol. 35, No. 2, April 1988, pp. 267–287.

[2] E. W. Dijkstra, A Note on Two Problems in Connection with Graphs, *Numerische Mathematik*, Vol. 1, 1959, pp. 269–271.

[3] L. Guibas, J. Herchberger, K. Leven, M. Sharir, R. Tarjan, Linear Time Algorithms for Visibility and Shortest Path Problems inside Simple Polygons, *Algorithmica*, Vol. 2, No. 2, 1987, pp. 209–233.

[4] J. S. B. Mitchell, *Planning Shortest Paths*, Ph. D. Thesis, Department of Operations Research, Stanford University, August 1986.

[5] D. M. Mount, On Finding Shortest Paths on Convex Polyhedra, *Technical Report* 1495, Computer Science Department, University of Maryland, College Park, 1984.

[6] D. M. Mount, The Number of Shortest Paths on the Surface of a Polyhedron, *Technical Report*, Computer Science Department, University of Maryland, College Park, MD, 1986.

[7] J. O'Rourke, S. Suri, and H. Booth, Shortest Paths on Polyhedral Surfaces, Manuscript, Johns Hopkins University, 1984.

[8] C. Schevon, J. O'Rourke, The Number of Maximal Edges Sequences on a Convex Polytope, *Proceedings of Allerton Conference*, 1988.

[9] C. H. Papadimitriou, An Algorithm for Shortest-Path Motion in Three Dimensions, *Information Processing Letter*, Vol. 20, No. 5, 12 June, 1985.

[10] M. Sharir, On Shortest Paths Amidst Convex Polyhedra, *SIAM J. Comput.*, Vol. 16, No. 3, June 1987, pp. 561-572.

[11] M. Sharir and A. Schorr, On Shortest Paths in Polyhedral Spaces, *SIAM J. Comput.*, Vol. 15, No. 1, February 1986, pp. 193-215.

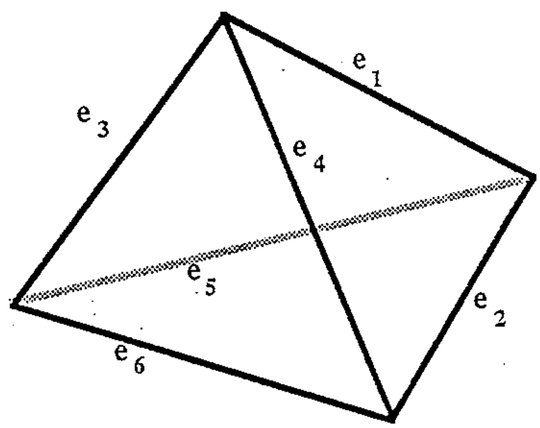

Fig. 1a A given convex polyhedron P.

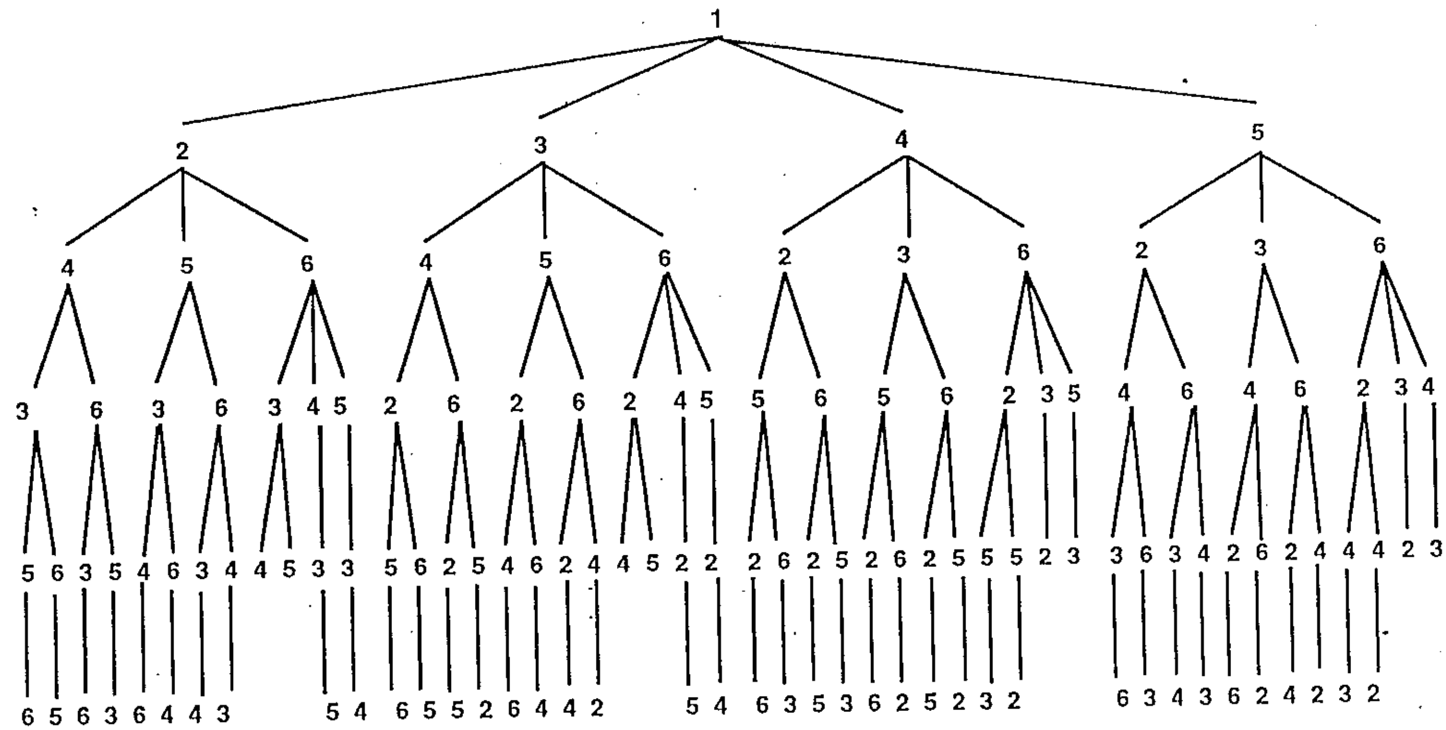

Fig. 1b The edge sequence tree of the convex polyhedron with starting edge e_1

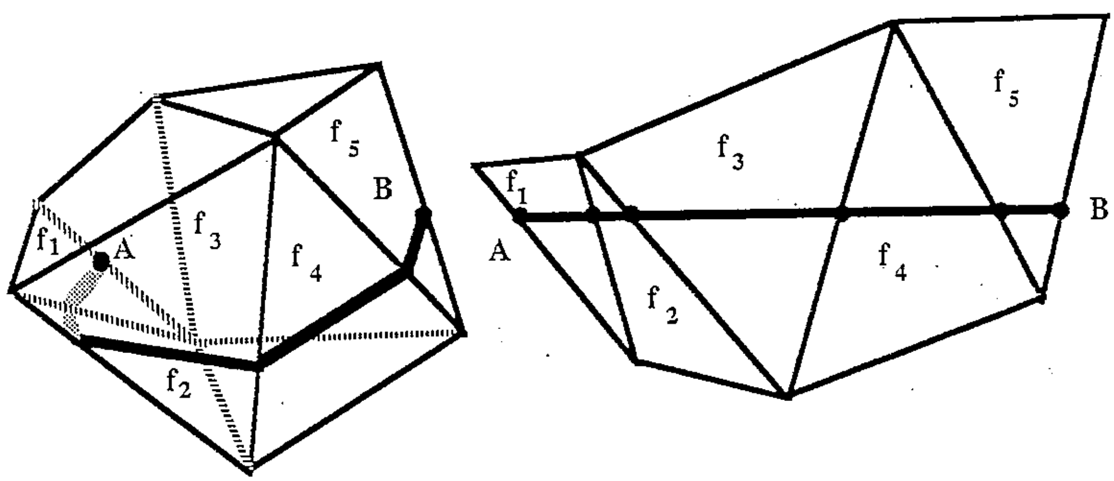

Fig. 2 The planar unfolding relative to edge sequence $(e_S\ e_1\ e_2\ e_3\ e_4\ e_e)$.

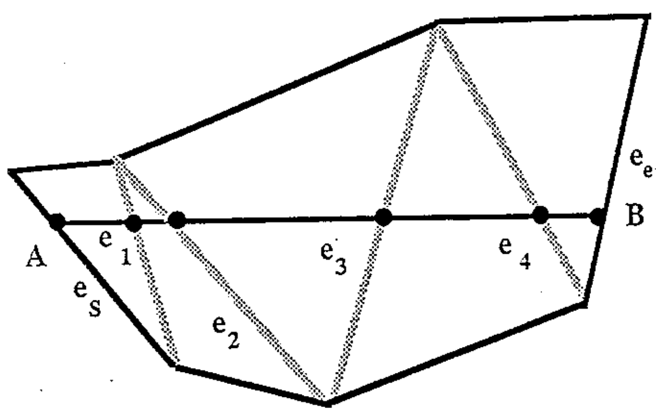

Fig. 3a Simple polygon G_1

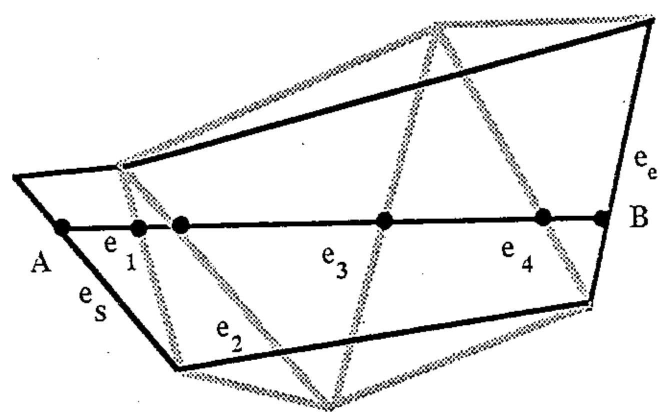

Fig. 3b Simple polygon G_2

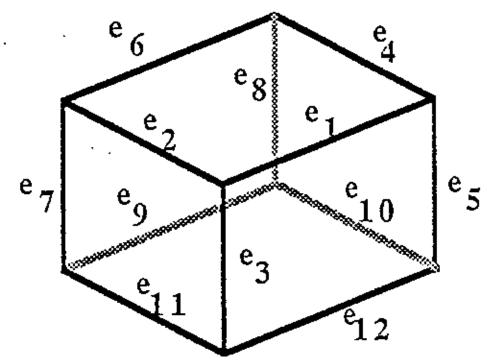

Fig. 4a A given rectangular polyhedron.

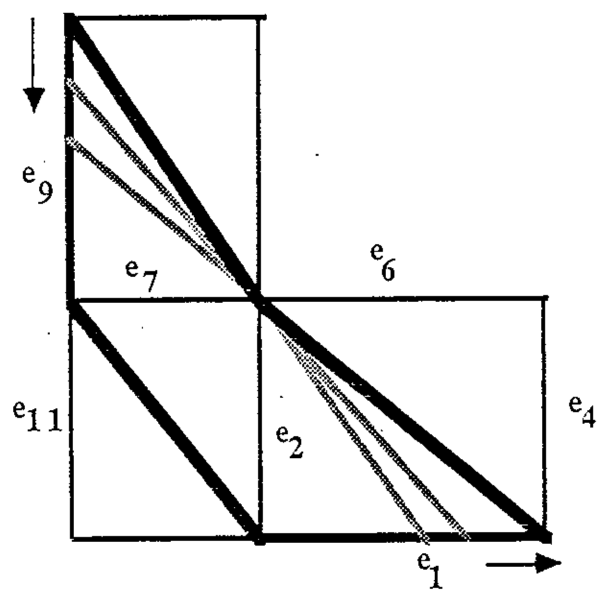

Fig. 4b The planar unfolding relative to edge sequence ($e_1 e_2 e_7 e_9$).

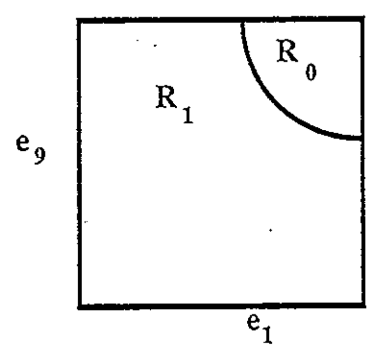

Fig. 4c The visibility relation diagram of Fig. 4b.

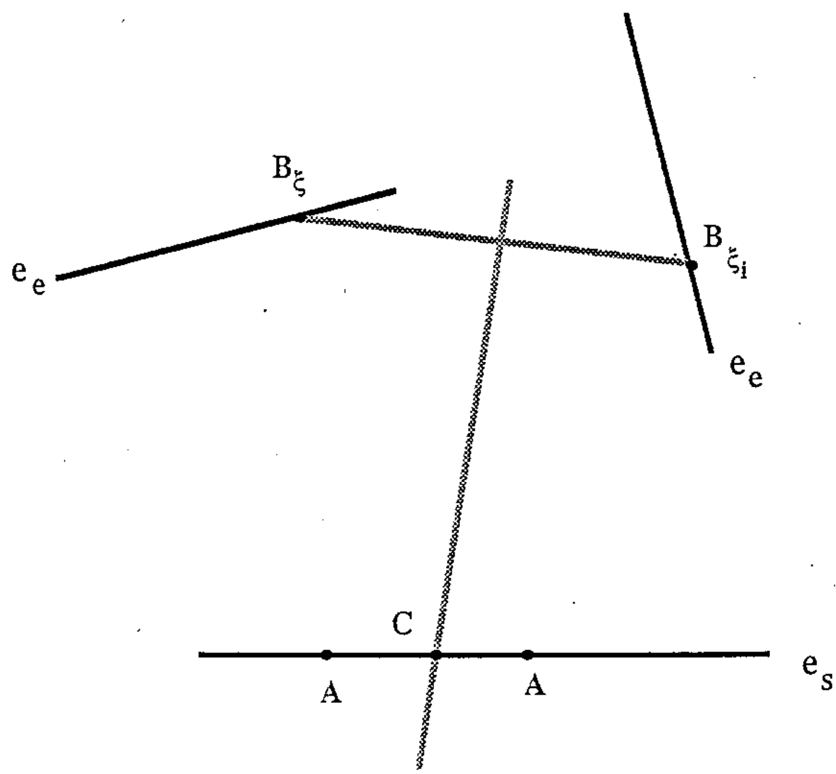

Fig. 5 The planar unfoldings relative to ξ and ξ_i

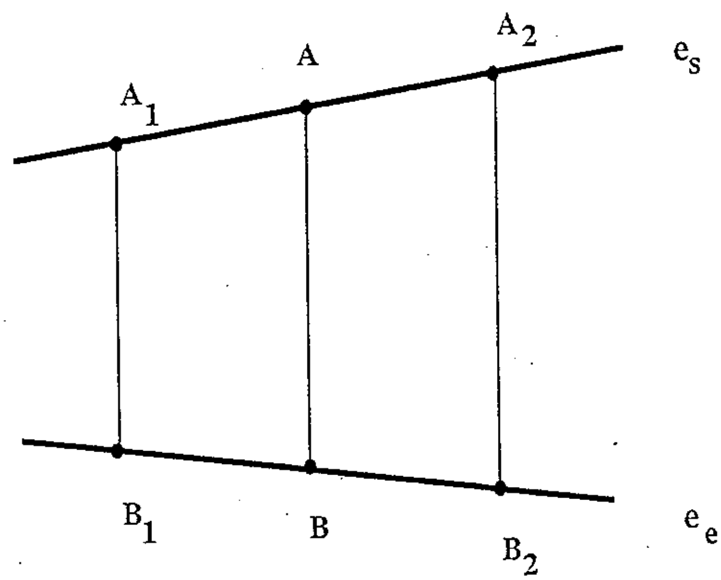

Fig. 6 The weight-points located on the boundary of region.

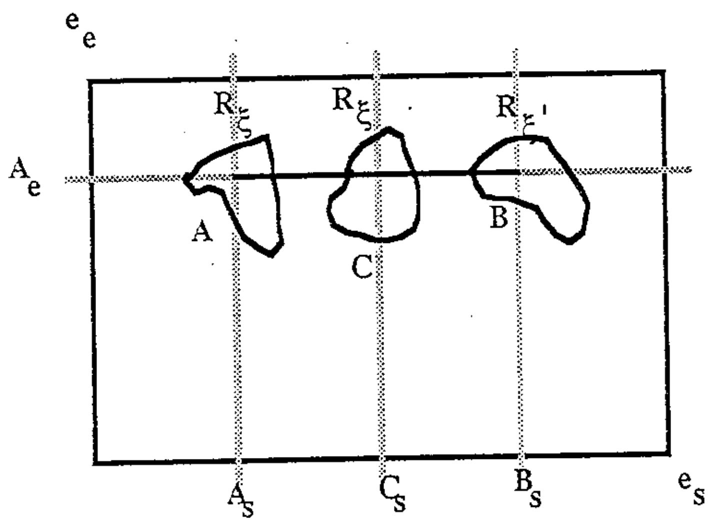

Fig. 7a In CASE 1, $A_e = B_e$

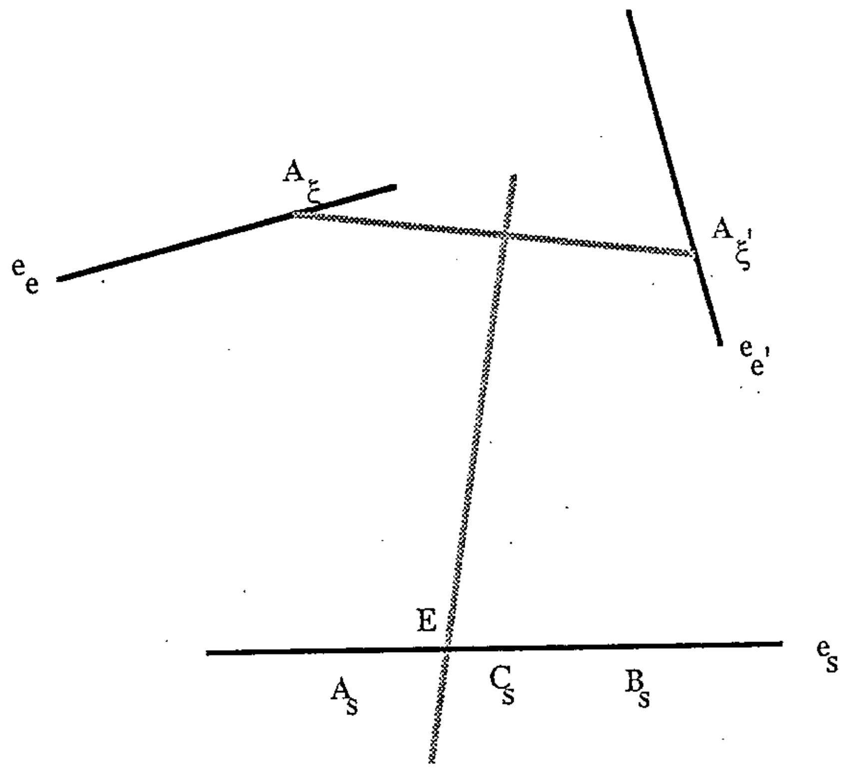

Fig. 7b The planar unfoldings of CASE 1.

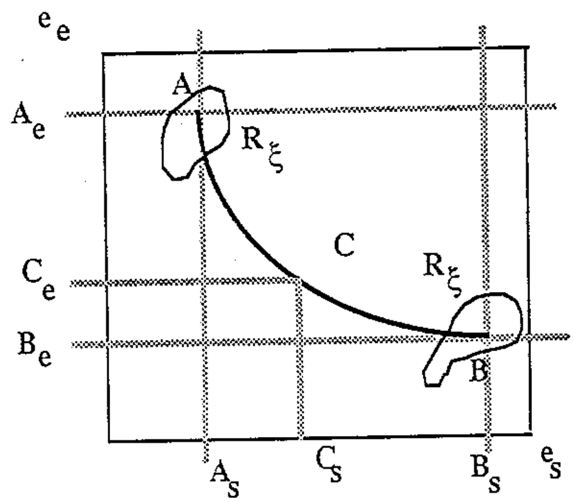

Fig. 8a (1) In CASE 2, A_e is not equal to B_e

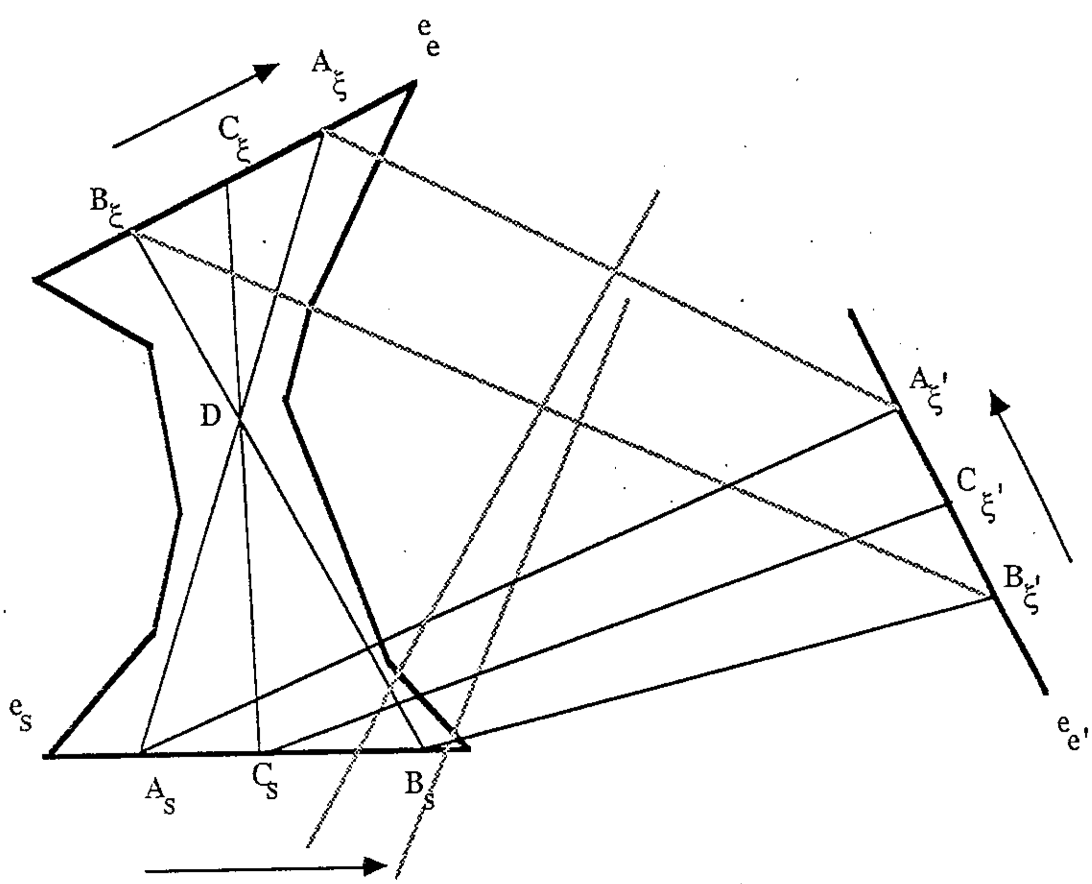

Fig. 8a (2) In CASE 2, two shortest path cross each other.

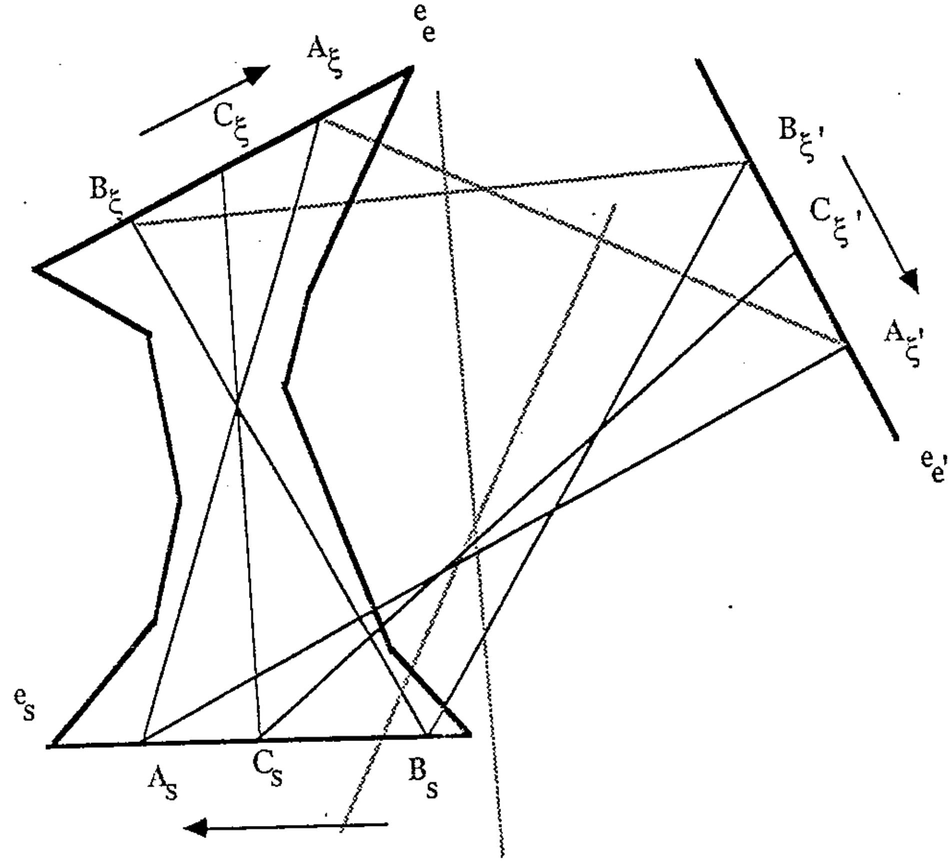

Fig. 8a (3) In CASE 2, four shortest paths cross each other in pairs.

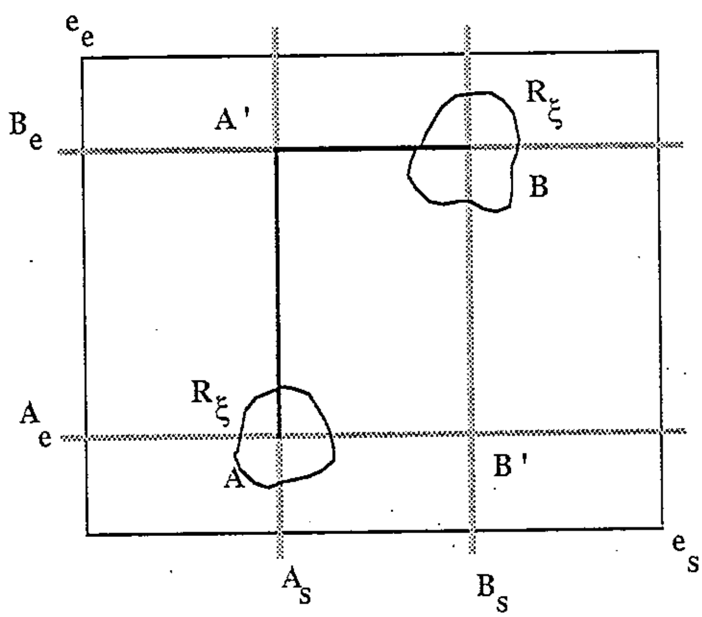

Fig. 8b (1) The path connects A and B by crossing A'.

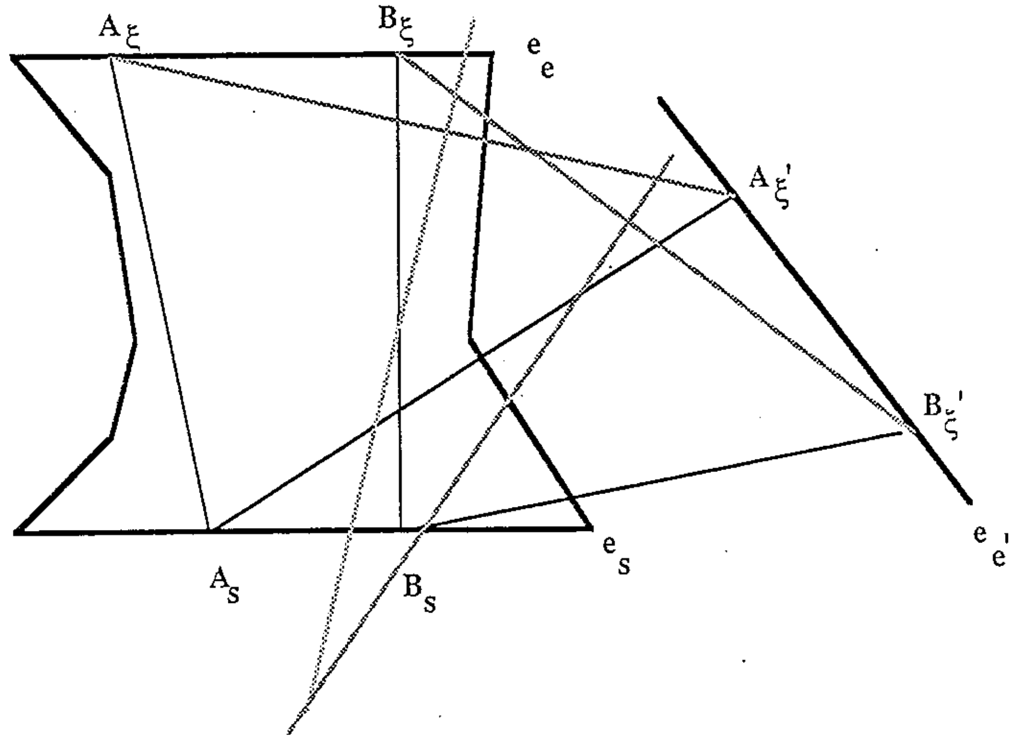

Fig. 8b (2) The unfolding relative to Fig. 8b (1).

January 30, 1989

Professor Joseph O'Rourke
Department of Computer Science
Smith College
Northampton, Massachusetts 01063
U. S. A.

Dear Joseph:

Thank you for your extended abstract and the comments on "finding all shortest path edge sequences." From the day we received your letter (01/18/89), we have been tried our best to modify the original abstract to a new version to meet the general cases. Enclosed please find a copy of this new version. It is our pleasure to have your comments.

I hope the following example can make it clearer.

EXAMPLE:

For a given rectangular polyhedron in Fig. 1, here we show how to construct the visibility relation diagram restricted to $S=\{\xi_1, \xi_2, \xi_3\}$, where $\xi_1=(e_1e_2e_7e_9)$, $\xi_2=(e_1e_2e_{11}e_9)$, and $\xi_3=(e_1e_8e_9)$.

Initially, we construct the visibility relation diagram for edge sequence ξ_1. the polygon $G_2(\xi_1)$ is shown in Fig. 2. Hence, the domain $Z=e_1\times e_9$ can be divided into two equivalent classes, R_{ξ_1} and R_{ϕ}, respectively specified as R_1 and R_0 in Fig. 3.

To add ξ_2, we unfold ξ_2 to the common plane of ξ_1 (see Fig. 4). It is easy to see that the points on e_1 and e_9 are visible to each other in $G_2(\xi_2)$ and hence R_{ξ_2} is the whole Z. Now, R_0 is obviously substituted by a part of R_{ξ_2} (denoted as R_3 in Fig. 5). For region $R_{\xi_1} \cap R_{\xi_2}$ (R_1 in Fig. 3), since the two source images of e_9 relative to ξ_1 and ξ_2 are connected at end point, the perpendicular bisector L is fixed and the partition curve is a vertical line on domain Z (see Fig. 5). As you can see, we now have two equivalent classes R_{ξ_1} and R_{ξ_2}, where $R_{\xi_1}=R_2$ and $R_{\xi_2}=R_1 \cup R_3$ in Fig. 5. The latter one obviously has two disjoint regions. This does not contradict to our Lemma 6, because there must exist some other edge sequences, of which the corresponding equivalent classes will cover either R_1 or R_3, or both. For example, we can add the planar unfolding relative to ξ_3 to the previous two unfoldings (see Fig. 6). As the same with ξ_2, R_{ξ_3} is the whole Z, too. However, R_3 (in Fig. 5) should be included in R_{ξ_3} after drawing the partition curves P_1 and P_2. We get a visibility relation diagram on domain Z, where $R_{\xi_1}=R_2$, $R_{\xi_2}=R_1$, and $R_{\xi_3}=R_3$ as Fig. 7.

This example evidently shows us the following.

(1) Since the surface of a Convex polyhedron is homeomorphic to $S^2=\{x \in R^3: |x|=1\}$, and the image of a connected space under a continuous map is also connected, each equivalent class defined on this space should be connected. For our example, although it is possible that R_{ξ_2} is disconnected during constructing the visibility relation diagram (contains more than one components), it will be finally reduced to only one region. The reason why R_{ξ_2} was divided into two components R_1 and R_2 is

"The visibility between e_1 and e_9 is limited to the area of $G_2(\xi_1)$ and $G_2(\xi_2)$. Hence, R_0 (in Fig. 3) was replaced by R_3 to meet the requirement of VISIBLE."

(2) Since eventually the disconnected components will be replaced by other regions, the number of regions in domain Z will not more than the number of all shortest path edge sequences, $O(n^2)$.

(3) The points on partition curves, in our terminology, is relative to the Voronoi edges in your Allerton papers. Since we use the visibility between points on edges to eliminate lots of event points, there are only $O(n^2)$ events to process when expanding a node on edge sequence trees. Since there are $O(n^2)$ shortest path edge sequences between a fixed pair of

edges, we need execute $O(n^2)$ expansions on edge sequence trees. Thus, we process at most $O(n^4)$ events to construct a visibility relation diagram for a given pair of starting edge and ending edge.

For your Allerton paper, there is also a point we do not understand. Why does the main loop of the algorithm run in $O(n \log n)$ time, with one iteration per event processed ? It appear to us to be gap, but perhaps the sketch just does not include details.

It is our pleasure to have information from you. All members in this research group are encouraged by these discussions. We will also reciprocate everything we write on this topic. Once again, thanks for your comments.

Best regards,

Hung–Yi Tony Tu

Research Group for Computational Geometry
Institute of Information Science, Academia Sinica

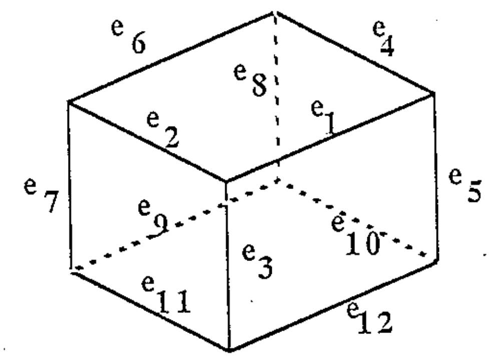

Fig. 1 A given rectangular polyhedron.

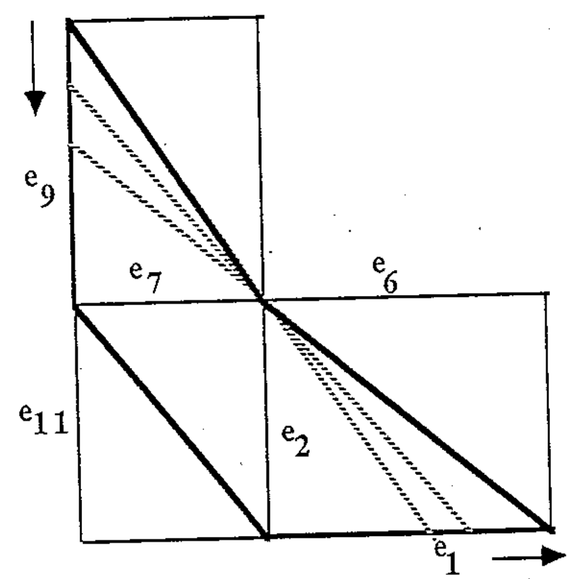

Fig. 2 The planar ufolding relative to edge sequence {1,2,7,9}.

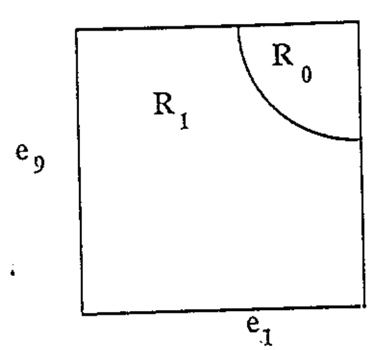

Fig. 3 The visibility relation diagram restricted to edge sequence {1,2,7,9}.

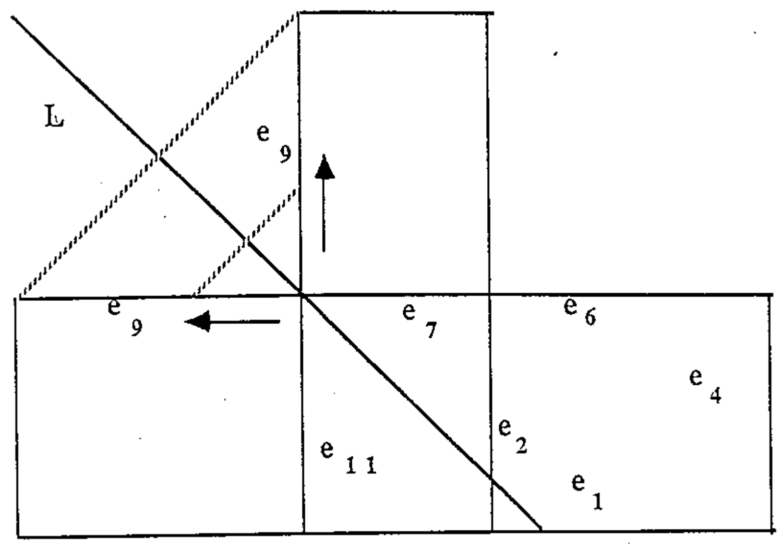

Fig. 4 The planar unfoldings relative to {1,2,7,9} and {1,2,11,9}.

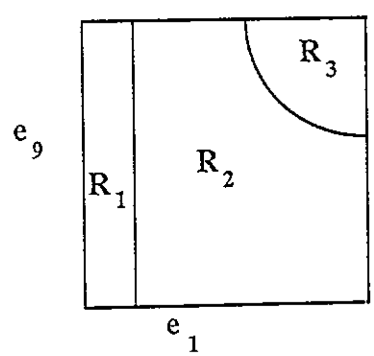

Fig. 5 The visibility relation diagram restricted to edge sequences {1,2,7,9} and {1,2,11,9}.

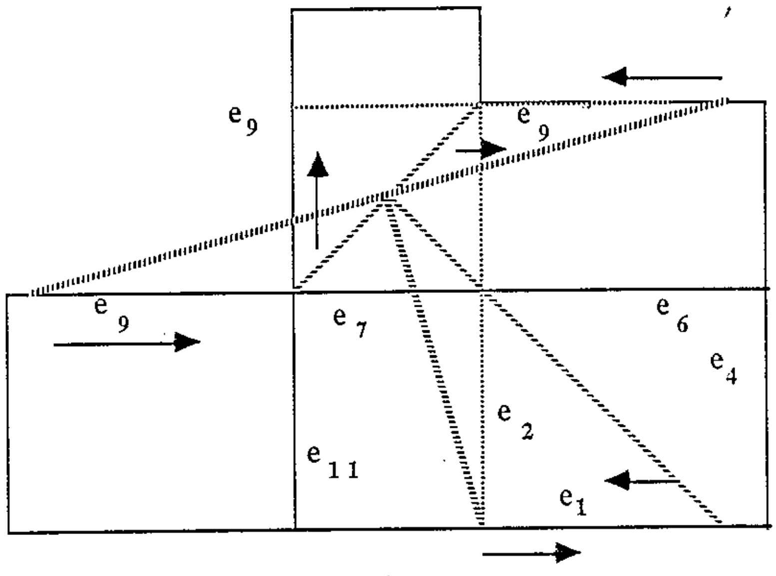

Fig. 6 The planar unfoldings relative to edge sequences {1,2,7,9},{1,2,11,9}, and {1,6,9}.

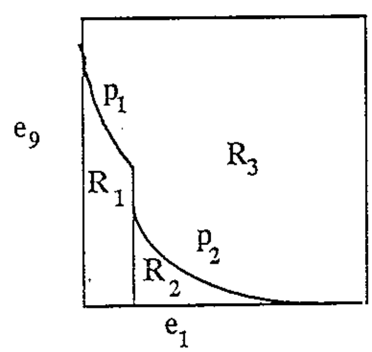

Fig. 7 The visibility relation diagram restricted to edge sequences {1,2,7,9},{1,2,11,9}, and {1,6,9}.