A Simple Shortest Path Algorithm for *L*-visible Polygons

Soo-Hwan Kim, *Member, KIMICS*

***Abstract*— The shortest path between two points inside a simple polygon *P* is a minimum-length path among all paths connecting them which don't pass by the exterior of *P*. A linear time algorithm for computing the shortest path in a general simple polygon requires triangulating a given polygon as preprocessing. The linear time triangulating is known to very complex to understand and implement it. It is also inefficient in case that the input without very large size is given because its time complexity has a big constant factor.**

Two points of a polygon *P* are said to be *L*-visible from each other if they can be joined by a simple chain of at most two rectilinear line segments contained in *P* completely. An *L*-visible polygon *P* is a polygon such that there is a point from which every point of *P* is *L*-visible. We present the customized optimal shortest path algorithm for an *L*-visible polygon. Our algorithm doesn't require triangulating as preprocessing and consists of simple procedures such as construction of convex hulls and operations for convex polygons, so it is easy to implement and runs very fast in linear time.

***Index Terms*— Visibility, *L*-visible Polygon, Shortest Path, Convex Hull.**

I. INTRODUCTION

The Euclidean shortest path problem is a basic geometric problem which often appears in computer application area such as computer graphics, robotics, and sensor networks. There are many possible versions of the problem. In this paper, we consider the shortest path inside a simple polygon. The shortest path between two points inside a polygon *P* is a minimum-length path among all paths connecting them which don't pass by the exterior of *P*. Guibas et al.[1] presented a linear time shortest path algorithm for a triangulated polygon. Chazelle[2] presented a linear time algorithm for triangulating a polygon, but it is known to very difficult to understand and implement it. It is also inefficient in case that the input without very large size is given because of a big constant factor in time complexity.

Notifying some structural knowledge of a given polygon enables us to solve geometric problems more efficiently. For example, the shortest path between two points inside a convex polygon is always a line segment connecting them, so computing the shortest path is straightforward. Visibility is a useful tool by which we can notify some structural knowledge and classify polygons[3].

Two points in a polygon are said to be *visible* from each other if the line segment connecting them doesn't pass by the exterior of the polygon. Star-shaped polygons, edge-visible polygons, and segment-visible polygons are well known as the polygon classes according to visibility. A polygon *P* is said to be *star-shaped* if there exists a point in *P* from which every point of *P* is visible. A polygon *P* is said to be *edge-visible* (resp. *segment-visible*) if there exists an edge (resp. a line segment) in *P* from which *P* is weakly visible. A set *T* is *weakly-visible* from a set *S* if for every point *p* in *T* there exists a point of *S* from which *p* is visible. An edge-visible polygon is a special type of a segment-visible polygon. Kim[4] showed that the shortest path between two points of a segment-visible polygon can be found in linear time without triangulating step as preprocessing.

In this paper, we consider *L*-visibility which is a specific version of *m*-visibility[5]. Two points of a polygon *P* are said to be *L-visible* from each other if they can be joined by a simple chain of at most two rectilinear line segments contained in *P* completely. An *L*-visible polygon *P* is a polygon such that there is a point *z* in *P* from which every point of *P* is *L*-visible. The set of points such as a point *z* is said to be *L-kernel* of *P*. Kim and Chwa[6] presented a linear time algorithm for finding the *L*-kernel of a rectilinear polygon. Fig. 1 shows an example of an *L*-visible polygon which is not segment-visible.

We present a linear time shortest path algorithm for a given *L*-visible polygon which doesn't require triangulating step as preprocessing and construction of complex data structures.

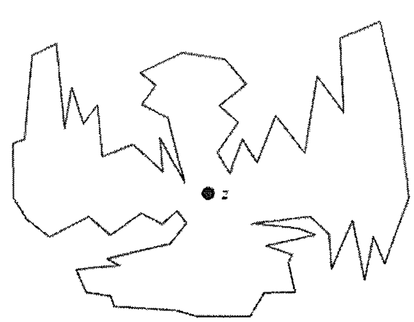


Fig. 1. An example of an *L*-visible polygon.

Manuscript received January 5, 2010; revised January 9, 2010; accepted January 17, 2010.

Soo-Hwan Kim is with the Department of Embedded IT, Pusan University of Foreign Studies, Busan, 608-738, Korea (Email: shkim @pufs.ac.kr)

II. DEFINITIONS AND PRELIMINARIES

Let P be a simple polygon specified by an ordered list $(v_0, v_1, \ldots, v_{n-1})$ of its n vertices in the counter-clockwise order. We normally assume that the vertices of P are in general position, i.e., no three vertices are collinear. A vertex is called *convex* if its internal angle is strictly less than 180°, otherwise, it is called *reflex*. The edge of P joining v_i and v_{i+1} is denoted by $e_i = (v_i, v_{i+1})$ (index calculation taken modulo n). $Bd(P)$ and $Int(P)$ denote the boundary and the interior of P, respectively so that $P = Int(P) \cup Bd(P)$. A chain $ch(p, q)$ is the portion of $Bd(P)$ from p to q in the counter-clockwise sense. A chain $ch(v, w)$ is said to be *convex* if all vertices except for v and w are convex.

The shortest path between two points p and q of P is a minimum-length path among all paths connecting them which don't pass by the exterior of P. The shortest path between p and q is denoted by $\pi(p, q)$. A path $\pi(p, q)$ is said to be *convex* if we don't meet any left-turn (or any right-turn) when we traverse $\pi(p, q)$ from p to q. A line and a line segment connecting two points p and q is denoted by $l(p, q)$ and $ls(p, q)$, respectively.

A chain $ch(v, w)$ is said to be *tidy* if there exists a convex chain $ch'(w, v)$ such that the union of $ch(v, w)$ and $ch'(w, v)$ forms a simple polygon $P'(v, w)$. A chain $ch(v, w)$ is said to be *very tidy* if for every chain $ch(p, q) \subset ch(v, w)$, there exists a simple polygon $P'(p, q)$ and a convex chain $ch'(w, v)$ such that $P'(p, q) \subset P'(v, w)$. The following lemma 1 shows that the shortest path between two points on a very tidy chain is always convex and can be computed in linear time.

Lemma 1 [4]: A shortest path $\pi(p, q)$ between p and q on a very tidy chain $ch(v, w)$ of a simple polygon P is always convex. Also, $\pi(p, q)$ can be computed in linear time not using the triangulating step.

III. SHORTEST PATH ALGORITHMS

Let z be a point in the L-kernel of an L-visible polygon P. For convenience, we assume that z and every vertex of P differ in both x-coordinate and y-coordinate. Let a (resp. b, c, and d) be the first point on $Bd(P)$ hit by a ray starting from z in the direction right (resp. up, left, and down). P is partitioned by four regions R_1, R_2, R_3, and R_4 using four points a, b, c, and d as shown in Figure 2.

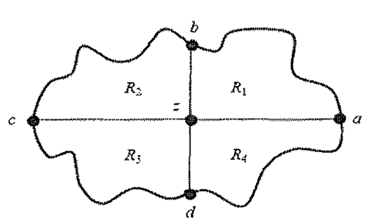


Fig. 2. A partition of a polygon P.

All chains $ch(a, b)$, $ch(b, c)$, $ch(c, d)$, and $ch(d, a)$ are very tidy (see Lemma 2), but it is not true that every chain of P is very tidy (see Fig. 3). Hereafter, the statement "A point p is visible from a line segment ls." means "there is a point q on ls such that x-coordinates (or y-coordinates) of p and q are equal."

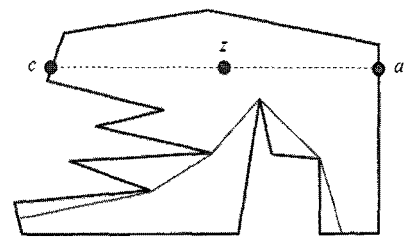


Fig. 3. $ch(c, a)$ is not very tidy.

Lemma 2: All chains $ch(a, b)$, $ch(b, c)$, $ch(c, d)$, and $ch(d, a)$ are very tidy

Proof. We only focus on $ch(a, b)$. The remaining chains can be proved in a symmetrical manner. In order to prove that $ch(a, b)$ is very tidy, we have to show that for $ch(p, q) \subset ch(a, b)$, there is a convex chain $ch'(q, p)$ which doesn't intersect with $ch(a, b)$ except for p and q. We consider two cases depending on the relative locations of p and q.

First, we consider in case that both of p and q are visible from one of $ls(z, a)$ and $ls(z, b)$. Let p' and q' be points of $ls(z, a)$ or $ls(z, b)$ from which p and q are visible, respectively. $ls(p, p') \cup ls(p', q') \cup ls(q', q)$ is a convex chain which doesn't intersect with $ch(a, b)$ except for p and q. Hence the lemma holds for this case.

Next, we consider in case that p is visible from a point p' on $ls(z, a)$ and q is visible from a point q' on $ls(z, b)$ (see Fig. 4). If $ls(p, p')$ and $ls(q, q')$ don't intersect, $ls(p, p') \cup ls(p', z) \cup ls(z, q') \cup ls(q', q)$ is a convex chain which proves that $ch(a, b)$ is very tidy (see Fig. 4 (a)). Otherwise, let r be a point at which $ls(p, p')$ and $ls(q, q')$ intersect. Then $ls(p, r) \cup ls(r, q)$ is a convex chain which proves that $ch(a, b)$ is very tidy (see Fig. 4 (b)). Therefore, the lemma holds. □

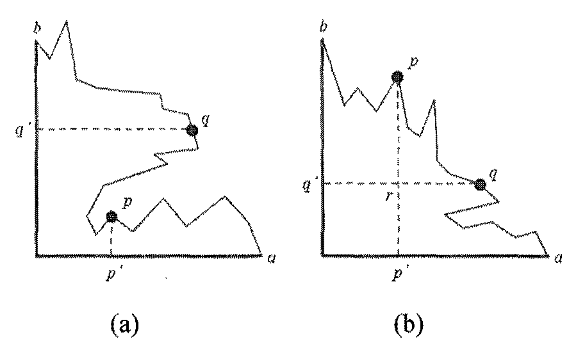


Fig. 4. Illustration for the proof of Lemma 2.

Consider the following three cases depending on the relative locations of two points s and t in P.

Case 1. Two points s and t are located in a same region, i.e., $s, t \in R_i$, where $i = 1, 2, 3$, and 4.

Case 2. Two points s and t are located in symmetric regions with respect to z, i.e., $s \in R_i$, $t \in R_j$, where $(i, j) = (1, 3)$ and $(2, 4)$.

Case 3. Two points s and t are located in adjacent regions, i.e., $s \in R_i$, $t \in R_j$, where $(i, j) = (1, 2)$, $(2, 4)$, $(3, 4)$, and $(4, 1)$.

The following lemmas show that the shortest path between two points of P can be computed in linear time not using triangulating step for each case.

Lemma 3: The shortest path $\pi(s, t)$ between two points s and t of P is convex and is computed in linear time not using triangulating step in case that s and t are corresponding to Case 1.

Proof. We only consider in case that s and t are located in R_1 since the remaining cases are symmetrical. Suppose that $ls(s, t)$ is not contained in P, otherwise, the lemma holds.

Let p (resp. q) be the closest point on $ch(a, b)$ from s (resp. t) among the intersections of $ls(s, t)$ and $ch(a, b)$. The rest of this proof is quite similar to Lemma 2's. In consequence, if $ch(p, q) \subset ch(a, b)$, then the chain $ls(s, p) \cup ch(p, q) \cup ls(q, t)$ is very tidy. If $ch(q, p) \subset ch(a, b)$, then the chain $ls(t, q) \cup ch(q, p) \cup ls(p, s)$ is very tidy. Hence $\pi(s, t)$ is contained in R_1 completely. So, by Lemma 1, $\pi(s, t)$ is convex and is computed in linear time not using triangulating step. □

Lemma 4: The shortest path $\pi(s, t)$ between s and t of P is computed in linear time not using triangulating step in case that s and t are corresponding to Case 2.

Proof. Without loss of generality, we consider that $s \in R_3$ and $t \in R_1$. Suppose that $ls(s, t)$ is not contained in P. Otherwise, the lemma holds. Consider the following two cases.

Case 2-1. s and t are visible from $ls(z, d)$ and $ls(z, a)$ (or $ls(z, c)$ and $ls(z, b)$), respectively.

Case 2-2. s and t are visible from $ls(z, c)$ and $ls(z, a)$ (or $ls(z, d)$ and $ls(z, b)$), respectively.

First, we consider Case 2-1. Assume that s and t are visible from $ls(z, d)$ and $ls(z, a)$, respectively. The other case can be proved in a symmetrical manner. Both of $\pi(z, s)$ and $\pi(z, t)$ are convex by Lemma 3 (see Fig. 5 (a)). Let s_1 (resp. t_1) be the point on $\pi(z, s)$ (resp. $\pi(z, t)$) which the common supporting line of $\pi(z, s)$ and $\pi(z, t)$ passes through. Then $\pi(s, t) = \pi(s, s_1) \cup \pi(s_1, t_1) \cup \pi(t_1, t)$ since $\pi(s_1, t_1)$ is contained in the region surrounded with $ls(s_1, t_1)$, $\pi(z, s_1)$, and $\pi(z, t_1)$.

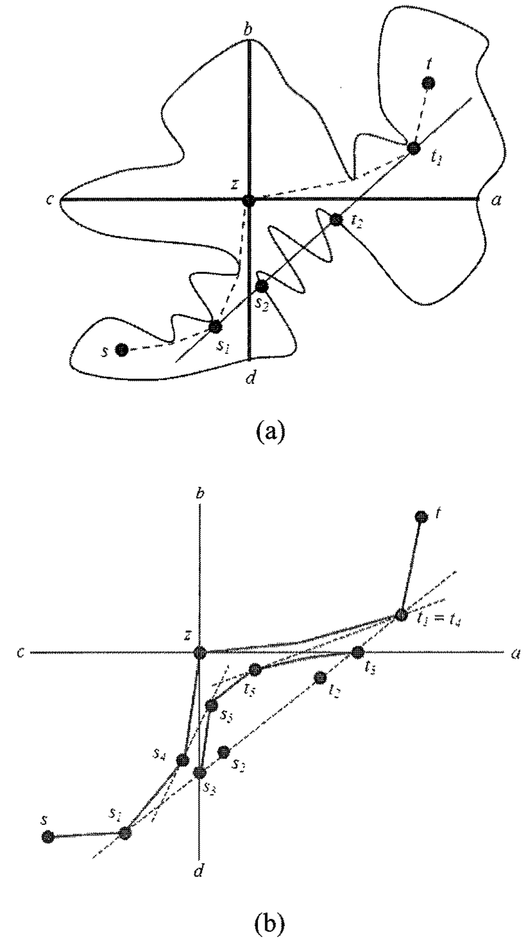


Fig. 5. Illustration for Case 2-1.

If $\pi(s_1, t_1) = ls(s_1, t_1)$, the lemma holds. Otherwise, let s_2 (resp. t_2) be the closest point on $Bd(P)$ from s_1 (resp. t_1) among the intersections of $ls(s_1, t_1)$ and $Bd(P)$. Also, let s_3 (resp. t_3) be the intersection of $ls(z, d)$ and $ls(s_1, s_2)$ (resp. $ls(z, a)$ and $ls(t_1, t_2)$). Since $ch(s_2, t_2) \subset ch(d, a)$, $ch(s_2, t_2)$ is very tidy and $\pi(s_3, t_3)$ is contained in the triangle (z, s_3, t_3). So, $\pi(s_3, t_3)$ is convex and can be computed in linear time by Lemma 3.

Now, we can see that $\pi(s_1, t_1)$ is contained in the region surrounded with $\pi(s_1, z)$, $\pi(z, t_1)$, $\pi(s_3, t_3)$, $ls(s_1, s_3)$, and $ls(t_1, t_3)$ (see Fig 5 (b)). By the properties of convexity, $\pi(s_1, t_1)$ can be constructed using the following manner. Let s_4 (resp. s_5) be the point on $\pi(s_1, z)$ (resp. $\pi(s_3, t_3)$) which the common supporting line of $\pi(z, s)$ and $\pi(s_3, t_3)$ passes through. Then $\pi(s_1, s_5) = \pi(s_1, s_4) \cup ls(s_4, s_5)$. Similarly, let t_4 (resp. t_5) be the point on $\pi(z, t_1)$ (resp. $\pi(s_3, t_3)$) which the common supporting line of $\pi(z, t_1)$ and $\pi(s_3, t_3)$ passes through. Then $\pi(t_5, t_1) = ls(t_5, t_4) \cup \pi(t_4, t_1)$. Hence, $\pi(s_1, t_1) = \pi(s_1, s_4) \cup ls(s_4, s_5) \cup \pi(s_5, t_5) \cup ls(t_5, t_4) \cup \pi(t_4, t_1)$. Since the common supporting line for two convex polygons can be computed in linear time[7], $\pi(s, t)$ is computed in linear time.

Now, we consider Case 2-2. Assume that s and t are visible from $ls(z, c)$ and $ls(z, a)$, respectively. The other case can be proved in a symmetrical manner. Both of $\pi(z, s)$ and $\pi(z, t)$ are convex by Lemma 3 (see Fig. 6 (a)). Let s_1 be the point on $\pi(z, s)$ adjacent to z, and let t_1 be the point of contact on $\pi(z, t)$ where a tangent line of $\pi(z, t)$ passes through s_1. Then $\pi(s, t) = \pi(s, s_1) \cup \pi(s_1, t_1) \cup \pi(t_1, t)$ since $\pi(s_1, t_1)$ is contained in the region surrounded with $ls(s_1, z)$, $\pi(z, t_1)$, and $ls(s_1, t_1)$.

If $\pi(s_1, t_1) = ls(s_1, t_1)$, the lemma holds. Otherwise, let t_2 be the closest point on $Bd(P)$ from t_1 among the intersections of $ls(s_1, t_1)$ and $Bd(P)$. Also, let t_3 be the intersection of $ls(z, a)$ and $ls(t_1, t_2)$. Two chains $ch(s_1, d)$ and $ch(d, t_2)$ are very tidy and $\pi(s_1, t_3)$ is contained in the triangle (z, s_1, t_3). By Lemma 3, $\pi(s_1, d)$ and $\pi(d, t_2)$ are convex and computed in linear time. Also, $\pi(s_1, t_3)$ is the union of $\pi(s_1, d)$ and $\pi(d, t_2)$. Since the union of two convex polygons can be constructed in linear time[7], $\pi(s_1, t_3)$ can be computed in linear time. Next, let t_4 (resp. t_5) be the point on $\pi(z, t_1)$ (resp. $\pi(s_1, t_3)$) which the common supporting line of $\pi(z, t_1)$ and $\pi(s_1, t_3)$ passes through. Then $\pi(t_5, t_1) = ls(t_5, t_4) \cup \pi(t_4, t_1)$. Hence, $\pi(s_1, t_1) = \pi(s_1, t_5) \cup ls(t_5, t_4) \cup \pi(t_4, t_1)$ (see Fig. 6 (b)). Hence $\pi(s, t)$ is computed in linear time. Therefore, the lemma holds true. □

Lemma 5: The shortest path $\pi(s, t)$ between s and t of P is computed in linear time not using triangulating step in case that s and t are corresponding to Case 3.

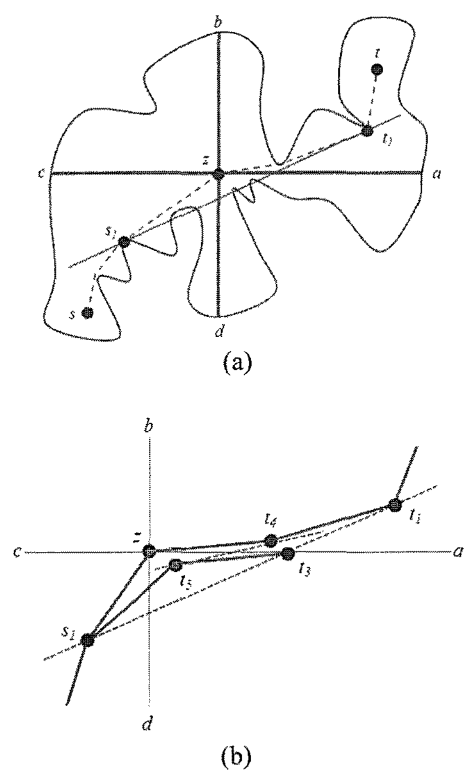


Fig. 6. Illustration for Case 2-2.

Proof. Without loss of generality, we consider that $s \in R_1$ and $t \in R_2$. Suppose that $ls(s, t)$ is not contained in P. Otherwise, the lemma holds. Consider the following three cases.

Case 3-1. s and t are visible from $ls(z, a)$ and $ls(z, c)$, respectively.
Case 3-2. Both of s and t are visible from $ls(z, b)$.
Case 3-3 s and t are visible from $ls(z, a)$ and $ls(z, b)$ (or $ls(z, b)$ and $ls(z, c)$), respectively.

In Case 3-1, both of $\pi(z, s)$ and $\pi(z, t)$ are convex by Lemma 3. Let s_1 (resp. t_1) be the point on $\pi(z, s)$ (resp. $\pi(z, t)$) adjacent to z (see Fig 7). Since $ch(s_1, b)$ and $ch(b, t_1)$ are very tidy and $\pi(s_1, t_1)$ is contained in the triangle (z, s_1, t_1), $ch(s_1, t_1)$ is very tidy. So, $\pi(s_1, t_1)$ is convex and is computed in linear time. Hence $\pi(s, t) = \pi(s, s_1) \cup \pi(s_1, t_1) \cup \pi(t_1, t)$ can be computed in linear time.

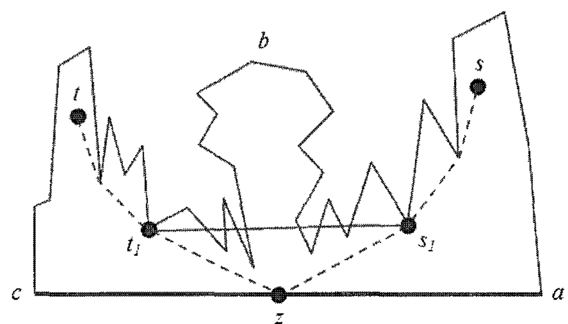


Fig. 7. Illustration for Case 3-1.

In Case 3-2, let s_1 (resp. t_1) be the point on $ls(z, b)$ from which s (resp. t) is visible, and let z_1 be the middle point of $ls(s_1, t_1)$. If z_1 is regarded as z, this case is corresponding to Case 2-2 in Lemma 4 (see Fig. 8 (a)). Hence the lemma holds true for this case.

In Case 3-3, considering the locations and the structures of $\pi(z, s)$ and $\pi(z, t)$ (see Fig 8 (b)), we can prove that the lemma holds using the proof of Cases 2-2 in Lemma 4. Therefore, $\pi(s, t)$ can be computed in linear time not using triangulating step. □

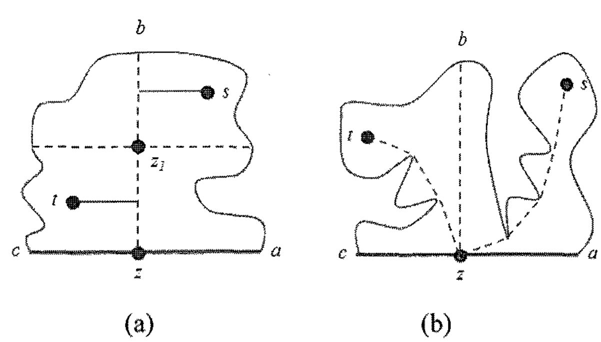


Fig. 8. Illustration for Case 3-2 and Case 3-3.

From above lemmas, we can get the following theorem. Also, the proofs of above lemmas imply an algorithm for computing the shortest path.

Theorem 1: The shortest path between two points of an L-visible polygon P can be computed in linear time not using triangulating step.

IV. CONCLUDING REMARKS

In this paper, we presented a linear time algorithm for computing the shortest path between two points of an L-visible polygon. The algorithm don't require triangulating as preprocessing and consists of simple procedures such as construction of convex hulls and operations for two convex polygons, so it easy to implement and runs fast in optimal time.

REFERENCES

[1] L. Guibas, J. Hershberger, D. Leven, M. Shrir, and R. Tarjan, "Linear Time Algorithm for Visibility and Shortest Path Problems inside Triangulated Simple Polygons," *Algorithmica*, Vol. 2, No. 1-4, 1987, pp. 209-233.

[2] B. Chazelle, "Triangulating a Simple Polygon in Linear Time," *Discrete Comput. Geom.*, Vol. 6, No. 1, 1991, pp. 485-524.

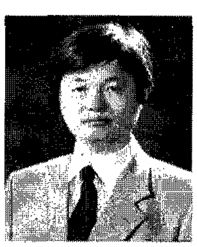
[3] J. O'rourke, *Art Gallery Theorems and Algorithms*, Oxford Univ. Press, 1987.

[4] S.-H. Kim, "A Fast Algorithm for Computing the Shortest Path between Two Points inside a Segment-Visible Polygon," *Journal of KIMICS*, Vol. 14, No. 2, 2010. (in Korean)

[5] H. ElGindy, D. Avis, and G. T. Toussaint, "Application of a two dimensional hidden line algorithm to other geometric problems," *Computing*, Vol. 31, 1983, pp. 191-202,

[6] S.-H. Kim and K.-Y. Chwa, "An Optimal Algorithm for Finding the L-Kernel of a Rectilinear Polygon, *Journal of KIISE*, Vol. 17, No. 2, 1990, pp. 200-211. (in Korean)

[7] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Velag, 1985.



Soo-Hwan Kim received his B.S. degree in Computer Science & Statistics from Seoul National University, Seoul, Korea, in 1987, M.S. and Ph.D. degrees in Computer Science from KAIST, Korea, in 1989 and 1995, respectively. In 1992, he joined the faculty at Pusan University of Foreign Studies, Busan, Korea, where he is currently a professor in the department of Embedded IT. His research interests include the design of sequential and parallel algorithms, computational geometry, and image processing.