

# 단순 다각형 내부의 두 가시성 다각형에 대한 집합 연산을 수행하는 최적 알고리즘

## (Optimal Algorithms for the Set Operations of Two Visibility Polygons in a Simple Polygon)

김 수 환<sup>†</sup>

(Soo-Hwan Kim)

**요 약** 단순 다각형  $P$ 의 가시성 다각형은 점이나 에지와 같은  $P$ 에서의 가시원으로부터 가시적인 점들의 집합을 말한다. 가시성 다각형은 점들의 집합이므로 가시성 다각형들에 대해 교집합, 합집합, 차집합 등과 같은 집합 연산을 수행할 수 있다. 두 가시성 다각형의 교집합은 해당되는 두 가시원으로부터 동시에 보이는 점들의 집합이고, 합집합은 하나 이상의 가시원으로부터 보이는 점들의 집합이다. 두 가시성 다각형의 차집합은 하나의 가시원으로부터만 보이는 점들의 집합이다.

모두  $n$ 개의 정점을 가진 두 개의 일반적인 다각형에 대해 집합 연산을 수행하는 기존의 알고리즘으로 가장 효율적인 알고리즘은  $O(n \log n + k)$  시간이 소요된다,  $k$ 는 집합 연산의 출력의 크기이다. 그러나 본 논문에서는 가시성 다각형의 특성을 이용하여  $O(n)$  시간에 교집합, 합집합, 차집합을 구하는 최적인 알고리즘을 제시한다.

**키워드** : 가시성, 가시성 다각형, 화랑 문제, 최적 알고리즘

**Abstract** The visibility polygon of a simple polygon  $P$  is the set of points which are visible from a visibility source in  $P$  such as a point or an edge. Since a visibility polygon is the set of points, the set operations such as intersection, union, or difference can be executed on them. The intersection (resp. union) of two visibility polygons is the set of points which are visible from both (resp. either) of the corresponding two visibility sources. The difference of two visibility polygons is the set of points which are visible from only a visibility source.

Previously, the best known algorithm for the set operations of two polygons with total  $n$  vertices takes  $O(n \log n + k)$  time, where  $k$  is the output size. In this paper, we present  $O(n)$  time algorithms for computing the intersection, the union, and the difference of given two visibility polygons, which are optimal.

**Key words** : Visibility, Visibility Polygon, Art Gallery Problem, Optimal Algorithms

### 1. 서 론

컴퓨터 이론의 한 연구 분야인 계산 기하학(computational geometry)은 기하 문제에 대한 알고리즘의 설계 및 분석, 그리고 이에 필요한 기초 기하 이론에 관한 연구를 수행하는 분야이다[1]. 계산 기하학의 분야 중 가시성 문제(visibility problem)는 평면 또는 공간 상의 물체들 사이의 가시성에 관한 문제로서, 그래픽스, 로보틱스, 영상처리 등의 분야에서 많이 발생한다.

다각형의 내부의 어떤 두 점  $p$ 와  $q$ 를 연결하는 선분

이 다각형의 외부와 교차하지 않으면, “ $p$ 와  $q$ 는 서로 볼 수 있다”라고 말한다. 다각형의 주어진 점  $p$ 에서 볼 수 있는 모든 점들의 집합을  $p$ 에 의한 가시성 다각형(visibility polygon)이라고 부른다. ElGindy와 Avis[2], Lee[3]는 주어진 점에서의 가시성 다각형을 구하는 선형 시간의 알고리즘을 제시하였다.

Avis와 Toussaint[4]은 가시성 개념을 에지로 확장하여 에지 가시성을 제안하였다. 다각형의 에지  $e$ 와 내부의 한 점  $x$ 가 주어질 때,  $x$ 를 볼 수 있는  $e$  상의 점이 존재하면 “ $e$ 는  $x$ 를 볼 수 있다”라고 정의한다. Avis와 Toussaint[4]은 주어진 에지가 다각형의 내부를 모두 볼 수 있는지를 판별하는 선형 시간의 알고리즘을 제시하였다. Toussaint[5], Guibas 등[6], Hoffmann과

<sup>†</sup> 종신회원 : 부산외국어대학교 컴퓨터공학부 교수

shkim@pufs.ac.kr

논문접수 : 2003년 7월 24일

심사완료 : 2003년 12월 22일

Mitchell[7], Kim과 Chwa[8]는 주어진 예지로부터 가시적인 영역인, 예지 가시성 다각형을 구하는 선형 시간의 알고리즘을 제시하였다.

본 논문에서는 다각형 내부에서 점 또는 예지 가시성 다각형들에 대해 집합 연산을 수행하는 알고리즘을 설계하려고 한다. 구체적으로 다각형 내부의 점이나 예지와 같은 가시원(visibility source)에 의한 가시 집합, 즉 가시성 다각형이 두 개 주어질 때, 이들 가시성 다각형에 대한 합집합(union), 교집합(intersection), 차집합(difference)을 구하는 것이다. 이 문제는 화랑 문제(art gallery problem)[9]와도 관련된다. 원래의 화랑 문제는 다각형 내부를 경비할 수 있는(볼 수 있는) 경비원의 최소수를 구하는 문제이다. 지금까지 화랑 문제에서 변형된 많은 문제들이 제시되고 해결되었다[9].

본 논문에서 고려되는 문제를 화랑 문제의 형태로 바꾸면 다음과 같다. 다각형으로 표현된 건물 내부를 경비하는 두 개의 로봇 또는 감시 카메라가 있을 때, 이 들로부터 동시에 경비되는 영역과 이 들로부터 경비되는 전체 영역을 구하라. 이것은 로봇 또는 감시 카메라의 위치에서의 가시성 다각형의 교집합과 합집합을 구하는 문제에 해당된다. 그리고 하나의 로봇 또는 감시 카메라에 의해서만 경비되는 영역을 구하라는 문제는 차집합을 구하는 문제에 해당된다.

일반적인 두 다각형의 교차 영역을 구하는 문제에 대해 Mckenna[10]는  $O(n^2)$  시간의 알고리즘을 제시했고, Ahn[11]은 Mairson[12]의 알고리즘을 이용하여 이 문제를  $O(n \log n + k)$  시간에 해결할 수 있다는 것을 보였다, 여기서  $n$ 은 두 다각형의 정점의 총 개수이고  $k$ 은 출력의 크기로서 그 값은  $O(n^2)$ 이다. 가장 간단한 다각형 구조인 볼록 다각형(convex polygon)에 대한 교차 영역을 구하는 문제는  $O(n)$  시간에 해결될 수 있다[13]. 그러나 비교적 간단한 구조의 다각형 부류인 별형 다각형(star-shaped polygon)이나 예지 가시 다각형(edge-visible polygon) 등에 대한 교차 영역을 구하는 문제와 출력의 크기가  $O(n)$ 으로 제한된 경우의 문제에 대해서도 Ahn의 방법을 이용한  $O(n \log n + k)$  시간의 알고리즘 외에는 다른 특별한 알고리즘은 알려져 있지 않다.

본 논문에서 고려하는 가시성 다각형은 별형 다각형이나 예지 가시 다각형과 같은 구조의 다각형이다. 가시성 다각형에 대한 교집합을 구하는 문제는, 교차 영역에 대한 출력의 크기가  $O(n)$ 이므로(2절의 정리 2 참조), 기존의 방법을 이용하면  $O(n \log n)$  시간에 해결할 수 있다. 그러나, 본 논문에서는  $O(n)$  시간에 교집합뿐만 아니라 합집합과 차집합을 구하는 알고리즘을 제시한다.

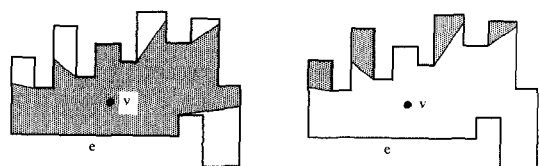
본 논문의 구성은 다음과 같다. 2장에서는 기본 용어를 정의하고, 가시성 다각형에 대한 여러 가지 성질들을

살펴본다. 3장에서는 2장에서 기술한 여러 가지 성질들을 바탕으로 하여 두 가시성 다각형의 교집합, 합집합, 차집합을 구하는 선형 시간의 알고리즘을 제시한다. 마지막으로, 4장에서는 결론과 연구 과제들을 제시한다.

## 2. 용어 정의 및 기본 성질

다각형  $P$ 를 평면 상에 놓인 단순 다각형(simple polygon)이라고 하자.  $P$ 는 정점(vertex)들이 반시계방향으로 주어진 순서 리스트( $v_0, v_1, \dots, v_{n-1}$ )로 표현된다. 편의상  $P$ 의 어떤 세 정점도 일직선 상에 놓여 있지 않다고 가정한다. 두 정점  $v_i$ 와  $v_{i+1}$ 을 연결하는 예지(edge)를  $e_i$ 로 표기하고,  $e_i$ 를  $v_i$ 에서  $v_{i+1}$  방향의 유향 선분(directed line segment)으로 간주한다, 여기서  $v_i$ 는  $e_i$ 의 꼬리(tail)라고 부르고,  $v_{i+1}$ 은  $e_i$ 의 머리(head)라고 부른다.  $P$ 의 정점들 중 내부에서의 각도가  $180^\circ$ 보다 작은 정점을 볼록 정점(convex vertex)이라 부르고,  $180^\circ$ 보다 큰 정점을 오목 정점(convex vertex)이라고 부른다.  $P$ 의 내부와 외부를 구분하는 점들의 집합을  $P$ 의 경계선(boundary)이라고 부르고,  $Bd(P)$ 로 표기한다.  $P$ 의 경계선을 반시계방향으로 따라갈 때 다각형의 내부는 왼쪽에 놓이고 외부는 오른쪽에 놓이게 된다.

$P$ 의 경계선 상의 두 점  $x, y$ 에 대해  $x$ 에서  $y$ 까지 반시계방향으로 따라간 경계선을  $x$ 와  $y$ 사이의 체인(chain)이라고 부르고  $ch_P(x, y)$ 로 표기한다.  $ch_P(x, y)$ 에서 다각형에 대한 혼동이 없다면 간단히  $ch(x, y)$ 로 표기하기로 한다. 두 점  $x$ 와  $y$ 를 연결하고  $x$ 에서  $y$  방향을 가진 유향 선분을  $ls(x, y)$ 라고 나타낸다.  $x$ 와  $y$ 가  $Bd(P)$ 의 점일 때, 양 끝점을 제외한  $ls(x, y)$ 가  $P$ 의 내부에 완전히 포함되면,  $ls(x, y)$ 를  $P$ 의 코드(chord)라고 부른다. 다각형  $P$ 의 한 가시원  $s$ 로부터 가시적인  $P$ 의 영역을  $s$ 로부터의 가시성 다각형이라고 부르고  $V(s)$ 로 표기한다. 앞으로 가시원은 점과 예지만을 고려하기로 한다. 그림 1은 다각형  $P$ 의 가시원인 예지  $e$ 와 점  $v$ 로부터의 가시성 다각형에 대한 교집합과 차집합을 보여 준다.



(a)  $V(e) \cap V(v)$  (b)  $V(e) - V(v)$

그림 1 두 가시성 다각형의 교집합과 차집합의 예

다각형  $P$ 의 어떤 가시원으로부터의 가시성 다각형에 대해 다음의 사실을 관찰할 수 있다.

**관찰 1.** 가시성 다각형의 경계선은 P의 에지 또는 P의 코드  $ls(p, q)$ 로서 구성된다. 단, 코드  $ls(p, q)$ 에서 p 또는 q는 반드시 P의 오목 정점에 해당한다.

일반적인 두 다각형의 교차 영역은  $O(n^2)$ 개의 연결 요소로 구성될 수 있다, 여기서 n은 두 다각형의 정점의 총 개수이다. 단순한 구조인 별형 다각형은 다각형의 내부를 모두 볼 수 있는 점이 존재하는 다각형으로 점 가시성 다각형과 같은 구조이다. 그런데 두 개의 별형 다각형의 교차 영역도 역시  $O(n^2)$ 개의 연결 요소로 구성될 수 있다(그림 2 참조). 그러나 하나의 다각형에서 정의되는 두 가시성 다각형의 교차 영역은 항상 하나의 연결 요소로 구성되며 경계선 상의 에지의 개수는  $O(n)$ 이다. 다음의 소정리 1, 2, 3과 정리 1, 2, 3은 이 사실을 증명한다.

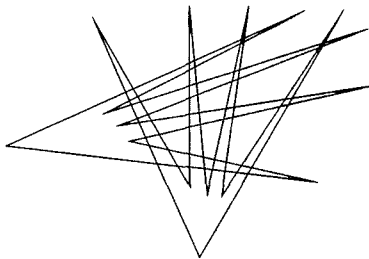


그림 2 두 별형 다각형의 교차 영역

**소정리 1.** 다각형 P의 두 점 p와 q가 P의 점 v로부터 가시적이면, P의 내부에서 정의되는 p와 q 사이의 최단 경로  $\pi(p, q)$ 의 모든 점도 v로부터 가시적이다.

(증명) 점 v가  $\pi(p, q)$ 의 어떤 점 z를 볼 수 없다고 가정하자. v는 p와 q를 볼 수 있으므로 두 선분  $ls(v, p)$ 와  $ls(v, q)$ 는 P에 포함된다. 만일  $\pi(p, q) = ls(p, v) \cup ls(v, q)$ 이면, v가  $\pi(p, q)$ 의 모든 점을 볼 수 있으므로 z를 볼 수 없다는 가정에 위배된다. 그렇지 않은 경우,  $ls(p, v)$ ,  $ls(v, q)$ 와  $\pi(q, p)$ 로 둘러싸인 영역은 단순 다각형을 형성한다. 이 다각형을 Q라고 부르자. Q의 내부에 P의 내부의 점이 포함되어 있으므로, Jordan curve theorem에 의해 Q는 다각형 P에 완전히 포함된다(그림 3(a) 참조). 이 때, 최단 경로의 성질에 의해 p, q를 제외한  $\pi(p, q)$ 의 각 정점은 P의 오목 정점에 해당된다. 따라서 v, p, q를 제외한 Q의 각 정점은 Q의 관점에서 오목 정점이 되어야 한다(그림 3(b) 참조). 그렇지 않으면 Q의 내부에 P의 외부의 점이 포함되기 때문이다. 그리하여  $\pi(p, q)$ 는 삼각형 (p, v, q)에 포함됨을 볼 수 있고, v는 Q의 내부를 통해  $\pi(p, q)$ 의 모든 점을 볼 수 있다. 이것은 v가  $\pi(p, q)$ 의 점 z를 볼 수 없다는 앞의 가정에 위배된다. 그러므로 최단 경로  $\pi(p,$

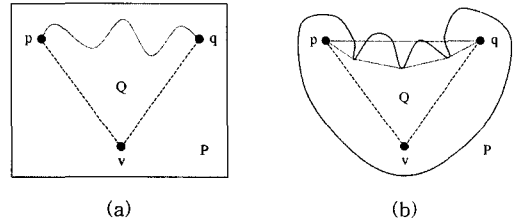


그림 3 소정리 1의 증명: 가시원이 점인 경우

q)의 모든 점이 v로부터 가시적이다. □

**소정리 2.** 다각형 P의 두 점 p와 q가 P의 에지 e로부터 가시적이면, P의 내부에서 정의되는 p와 q 사이의 최단 경로  $\pi(p, q)$ 의 모든 점도 e로부터 가시적이다.

(증명) 먼저, 에지 e가  $\pi(p, q)$ 의 어떤 점 z를 볼 수 없다고 가정하자. 그리고 에지 e에서 p를 볼 수 있는 점을 p', q를 볼 수 있는 점을 q'이라고 하자. p'과 q'이 같은 경우는 가시원이 점인 경우에 해당되므로 소정리 1에 의해 소정리가 성립한다. 그러므로 p'과 q'이 다른 경우만 고려하자. 에지 e가 x축 양의 방향으로 놓여 있다고 가정할 때, p와 q의 상대적인 위치를 다음의 세 가지 경우로 살펴볼 수 있다.

첫째, p, q가 모두 e의 아래쪽에 놓여 있는 경우이다. 이 경우는 p와 q가 e의 꼬리나 머리에서만 가시적이다. 이 때,  $p' \neq q'$ 이라면, e의 꼬리와 머리에서 하나씩 p와 q를 볼 수 있다. 일반성을 잃지 않고(without loss of generality), p'과 q'을 각각 e의 꼬리와 머리라고 하자(그림 4 (a) 참조). 그러면  $\pi(p, q) = ls(p, p') \cup e \cup ls(q', q)$ 이다. 따라서 e가  $\pi(p, q)$ 의 모든 점을 볼 수 있으므로 z를 볼 수 없다는 가정에 위배된다.

둘째, p, q가 모두 e의 위쪽에 놓여 있는 경우이다. 이 때, 만일  $ls(p, p')$ 과  $ls(q, q')$ 이 서로 교차한다면, 그 교차점 v를 가시원으로 둘 때, 소정리 1과 동일한 상태가 된다(그림 4(b) 참조). 따라서 교차점 v에서  $\pi(p, q)$ 의 모든 점을 볼 수 있다. 또한 v에서 볼 수 있는  $\pi(x, y)$ 의 점은  $ls(q', p')$ 에서도 볼 수 있으므로, e가  $\pi(p, q)$ 에 위치한 점 z를 볼 수 없다는 가정은 모순이다. 이제  $ls(p, p')$ 과  $ls(q, q')$ 가 교차하지 않은 경우를 살펴보자(그림 4(c) 참조). 이 경우,  $ls(p, p')$ ,  $ls(p', q')$ ,  $ls(q', q)$ 와  $\pi(q, p)$ 로 둘러싸인 다각형 Q가 사각형 (p, p', q', q)에 포함되고  $\pi(p, q)$ 가 오목 체인이라는 것을 가시원이 점인 경우와 비슷한 방식으로 증명할 수 있다. 따라서 Q의 구조상, 에지 e는 Q의 모든 점을 볼 수 있으므로, e가 z를 볼 수 없다는 앞의 가정은 모순이 된다.

셋째, p, q가 e의 위쪽과 아래쪽에 각각 위치하고 있는 경우이다. 일반성을 잃지 않고 p는 e의 아래쪽에, q는 e의 위쪽에 위치한다고 하고 p'이 e의 꼬리라고 하

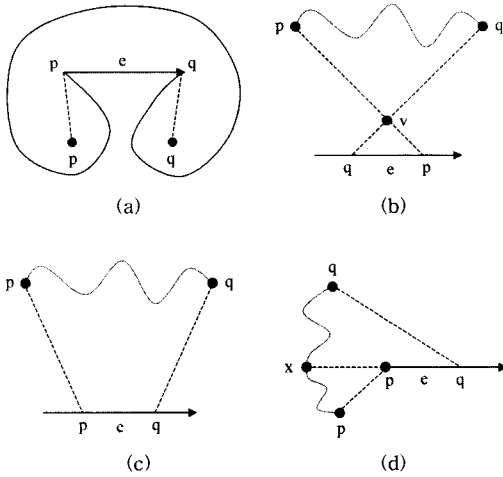


그림 4 소정리 2의 증명: 가시원이 예지인 경우

자(그림 4(d) 참조). 이 때, e의 꼬리에서 x축 음의 방향으로 따라갈 때 처음으로 만나는  $\pi(p, q)$ 의 점을 x라고 하자. 최단 경로의 성질에 의해  $\pi(p, q) = \pi(p, x) \cup \pi(x, q)$ 이다. 점 p와 x는 p'에서 가시적이므로 소정리 1에 의해  $\pi(p, x)$ 는 p'에서 가시적이다. 그리고 점 x와 점 q는 점 q'로부터 가시적이므로 역시  $\pi(x, q)$ 는 q'로부터 가시적이다. 따라서 e는  $\pi(p, q)$ 의 모든 점을 볼 수 있다. 그러므로 e가  $\pi(p, q)$ 의 점 z를 볼 수 없다는 가정은 모순이다.

결과적으로 e는 최단 경로  $\pi(p, q)$ 의 모든 점을 볼 수 있다. □

소정리 1과 소정리 2의 내용을 통합하면 다음의 정리 1과 같다.

**정리 1.** 다각형 P의 두 점 x와 y가 점 또는 예지인 가시원 s로부터 가시적이면, P의 내부에서 정의되는 x와 y 사이의 최단 경로  $\pi(x, y)$ 의 모든 점도 s로부터 가시적이다.

이제 두 가시성 다각형의 교차 영역은 둘 이상의 연결 요소로 구성되지 않는다는 것을 보인다.

**정리 2.** 다각형 P에서 정의되는 두 가시성 다각형의 교차 영역은 항상 하나의 연결된 영역이다.

(증명) 가시 집합인 두 가시성 다각형의 교집합을 S라고 하고, S가 공집합이 아니라고 하자. 그리고 S가 둘 이상의 서로 분리된 연결 요소들로 구성된다고 가정하자.  $S_1$ 과  $S_2$ 가 S의 분리된 두 요소, 즉,  $S_1 \cap S_2 = \emptyset$ 일 때, 점 p와 q를 각각  $S_1$ 과  $S_2$ 의 점이라고 두자. 이 때, p와 q를 연결하는 모든 경로에는 반드시 S에 포함되지 않는 점 z가 존재한다. 그렇지 않으면  $S_1$ 과  $S_2$ 가 서로 분리되어 있다는 사실에 위배된다. 이것은 p와 q를 연결하는 최단 경로  $\pi(p, q)$ 에도 S에 포함되지 않은

점 z가 존재한다는 것을 의미한다. 따라서 z를 볼 수 없는 가시원 s가 존재한다. 그런데 p와 q는 교집합에 속하는 점이므로 s로부터 가시적이므로, 정리 1에 의해  $\pi(x, y)$ 의 모든 점이 s로부터 가시적이어야 한다. 이는  $\pi(x, y)$ 의 점 z가 s로부터 비가시적이라는 사실에 위배된다. 결국 두 가시성 다각형의 교차 영역은 하나의 연결된 영역임이 증명된다. □

다음의 소정리 3과 정리 3은 두 가시성 다각형의 교차 영역이  $O(n)$  개의 정점(예지)만을 가진다는 것을 보인다.

**소정리 3.** A와 B를 다각형 P의 두 가시성 다각형이라고 하자. A의 어떤 예지  $e_A$ 와 B와의 교차 영역은 공집합이 아니면 하나의 선분이다.

(증명) s를 가시성 다각형 B의 가시원이라고 하고,  $e_A$ 와 B와의 교차 영역이 공집합이 아니라고 하자. s가 볼 수 있는  $e_A$ 의 점들 중에서 x에 가장 가까운 점을 p라고 하고, y에 가장 가까운 점을 q라고 하자.  $e_A$ 는 P의 예지이거나 코드이기 때문에 P의 내부에서 정의되는 p와 q 사이의 최단 경로  $\pi(p, q)$ 는  $ls(p, q)$ 이다. 그런데 p와 q가 s로부터 가시적이므로 소정리 1에 의해 최단 경로  $\pi(p, q)$ , 즉,  $ls(p, q)$ 의 모든 점이 s로부터 가시적이다. 이는  $e_A$ 와 B와의 교차 영역이 하나의 선분으로 구성됨을 의미한다. 그러므로  $e_A$ 와 B와의 교차 영역은 공집합이거나 아니면 하나의 선분이여야 한다. □

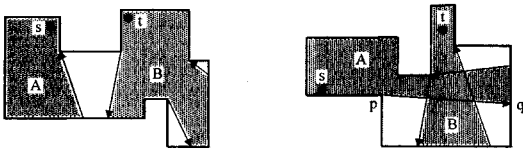
**정리 3.** 다각형 P에서 정의되는 두 가시성 다각형의 교차 영역은  $O(n)$  개의 예지로 구성된다.

(증명) 각 가시성 다각형의 예지의 개수가  $O(n)$ 이고 소정리 3에 의해 각 예지마다 하나 이하의 예지 요소가 생성되므로 교차 영역은  $O(n)$  개의 예지를 가진다. □

두 가시성 다각형 A와 B의 교집합이 공집합일 때, 그 둘의 합집합은 두 개의 연결 요소로 구성되고, A와 B의 차집합은 A 자체가 됨을 볼 수 있다. A와 B의 교집합이 공집합이 아닌 경우, 그 둘의 합집합은 역시 하나의 연결된 요소로 구성되나 차집합은 여러 개의 요소로 분리될 수 있다(그림 1 참조). 합집합과 차집합의 관계선은 교집합의 예지들과 P의 예지들로 구성되므로 그 크기는 역시  $O(n)$ 이다.

마지막으로 알고리즘에서 유용하게 이용되는 성질을 살펴본다. 가시성 다각형 A의 예지가 P의 코드  $ls(p, q)$ 일 때,  $ls(p, q)$ 와  $ch(p, q)$ 로 둘러싸인 영역을 P에서의 포켓(pocket)  $Po(p, q)$ 라고 부르자.  $Po(p, q)$ 는 물론 A의 바깥쪽에 위치하므로 A의 가시원으로부터 비가시적이다. 만일 A와 B가 서로 분리되어 있다면, B는 A의 어떤 예지에 의해 정의되는 포켓에 완전히 포함되어 있고, 반대로 A는 B의 어떤 예지에 의해 정의되는 포켓에 완전히 포함되어 있음을 알 수 있다(그림 5(a) 참조). A

와 B가 동일하지 않으면서 서로 교차한다면, B와 교차하는 A의 에지에 의해 정의되는 포켓이 존재하고 그 반대도 마찬가지이다.  $ls(p, q)$ 를 A의 에지로서 P에서 포켓을 정의한다고 하자(그림 5(b) 참조). 소정리 3에 의해  $ls(p, q)$ 와 B와의 교차 영역은 하나의 선분이다. 따라서  $ls(p, q)$ 와 교차하는 B의 에지는 0개 아니면 2개이다, 여기서 B의 에지의 머리가  $ls(p, q)$  위에 위치하는 경우는 배제한다. 그러므로  $ls(p, q)$ 와 교차하는 B의 에지 중에서 머리가  $Po(p, q)$ 의 내부에 위치하는 에지를  $b_j$ 라고 하고 머리가  $Po(p, q)$ 의 외부에 위치하는 에지를  $b_k$ 라고 할 때,  $b_j$ 에서  $b_k$  사이에 있는 B의 경계선은  $Po(p, q)$ 에 완전히 포함된다.



(a) A와 B가 분리된 경우 (b) A와 B가 교차하는 경우  
그림 5 가시성 다각형과 포켓, s와 t는 각각 A와 B의 가시원일

### 3. 알고리즘

다각형 P와 P의 두 가시성 다각형 A와 B는 각각  $n, r, s$ 개의 에지들로 구성된다고 가정하고, 다음과 같이 표기한다.

P의 에지 집합 =  $\{e_0, e_1, e_2, \dots, e_{n-1}\}$

A의 에지 집합 =  $\{a_0, a_1, a_2, \dots, a_{r-1}\}$

B의 에지 집합 =  $\{b_0, b_1, b_2, \dots, b_{s-1}\}$

모든 알고리즘에서 이용할 수 있는 자료구조로서 다각형 P와 가시성 다각형 A, B가 주어질 때 에지 정보를 효과적으로 검색하기 위한 배열을 구성할 수 있다. 먼저 P의 각 에지  $e_i$ 에 대해서,  $Aedge[i], Ahead[i], Atail[i]$ 는 아래와 같이 정의된다.

$Aedge[i] = e_i$ 에 완전히 포함되는 A의 에지  $a_j$ 가 존재하면 j, 그렇지 않으면 -1

$Ahead[i] = P$ 의 코드인 A의 에지  $a_j$ 의 머리가  $e_i$ 에 놓이면 j, 그렇지 않으면 -1

$Atail[i] = P$ 의 코드인 A의 에지  $a_j$ 의 꼬리가  $e_i$ 에 놓이면 j, 그렇지 않으면 -1

어떤 유향 선분의 머리(또는 꼬리)가 P의 정점  $v_i$ 에 위치한 경우, 이 유향 선분의 머리(또는 꼬리)는 에지  $e_i$ 에 위치한 것으로 둔다, 여기서  $e_{i-1} = ls(v_{i-1}, v_i)$ 이고  $e_i = ls(v_i, v_{i+1})$ 이다, 즉, 에지의 머리는 교차 여부의 고려 대상에서 배제한다.

그리고, A의 각 에지  $a_i$ 에 대해서,  $PedgeA[i], PheadA[i], PtailA[i]$ 는 다음과 같이 정의된다.

$PedgeA[i] = a_j$ 가 P의 에지  $e_j$ 에 완전히 포함되면 j, 그렇지 않으면 -1

$PheadA[i] = a_i$ 가 P의 코드일 때,  $a_i$ 의 머리가  $e_j$ 에 놓이면 j, 그렇지 않으면 -1

$PtailA[i] = a_i$ 가 P의 코드일 때,  $a_i$ 의 꼬리가  $e_j$ 에 놓이면 j, 그렇지 않으면 -1

다각형 P와 가시성 다각형  $A = V(v)$ 가 그림 6과 같을 때, 배열  $Aedge, Ahead, Atail, Pedge, PheadA, PtailA$ 는 표 1과 같이 구성된다. 그림 6에서 경계선의 바깥쪽 번호는 P의 에지 번호이고, 안쪽 번호는 A의 에지 번호이다.

표 1의 자료구조는 다음의 알고리즘을 이용하여 구축할 수 있다. 알고리즘에서 에지 번호(index)에 대한 연산은 항상 modulo 연산을 수행하는 것으로 간주한다.

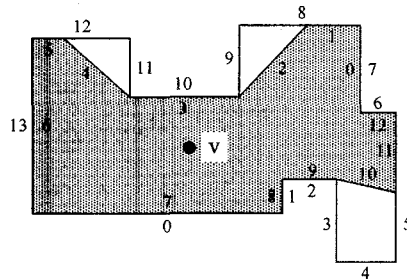


그림 6 다각형 P와 가시성 다각형  $A = V(v)$

표 1 그림 6의 다각형 P와 A에 대한 자료구조의 내용

P	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Aedge	7	8	9	-1	-1	11	12	0	1	-1	3	-1	5	6
Ahead	-1	-1	-1	-1	-1	10	-1	-1	-1	-1	2	-1	4	-1
Atail	-1	-1	-1	10	-1	-1	-1	-1	2	-1	-1	4	-1	-1

A	0	1	2	3	4	5	6	7	8	9	10	11	12
Pedge	7	8	-1	10	-1	12	13	0	1	2	-1	5	6
PheadA	-1	-1	10	-1	12	-1	-1	-1	-1	-1	5	-1	-1
PtailA	-1	-1	8	-1	11	-1	-1	-1	-1	-1	3	-1	-1

**알고리즘 1:** 자료구조를 구축하는 알고리즘

입력: P와 A

출력: 배열 정보 Aedge, Ahead, Atail, PedgeA, PheadA, PtailA

1. 각 배열 Aedge, Ahead, Atail, PedgeA, PheadA, PtailA의 초기값으로 -1을 배정한다.
2.  $a_0$ 와 교차하는 P의 에지  $e_j$ 를 찾는다.
3. while (A의 각 에지  $a_i$ 에 대해서,  $i = 0, \dots, r-1$ )
4.     while ( $a_i$ 와  $e_j$ 가 서로 교차하지 않으면)
5.          $j = j + 1$ ;
6.     endwhile
7.     if ( $a_i$ 가  $e_j$ 에 포함되면)
8.         Aedge[j] = i; PedgeA[i] = j;
9.     else if ( $a_i$ 의 머리가  $e_j$ 에 놓이면)
10.         Ahead[j] = i; PheadA[i] = j;
11.     else //  $a_i$ 의 꼬리가  $e_j$ 에 놓이면
12.         Atail[j] = i; PtailA[i] = j;
13.     endifelse
14. endwhile

알고리즘 1의 1줄과 2줄의 각 문장은  $O(n+r)$  시간에 수행된다. 3줄의 while 문에서 A의 각 에지와 P의 각 에지는 한 번씩만 고려되기 때문에, while 문의 수행 시간은  $O(n+r)$ 이다. 항상  $r \leq n$ 이므로 알고리즘 1의 전체 수행 시간은  $O(n)$ 이다.

자료구조 Bedge, Bhead, Btail, PedgeB, PheadB, PtailB도 같은 방식으로 정의되며 역시  $O(n)$  시간에 구

축할 수 있다. 이 자료구조를 이용하면 다음의 연산들을 상수 시간에 수행할 수 있음을 쉽게 알 수 있다.

- P의 에지  $e_i$ 에 포함되는 A의 에지 또는 B의 에지를 검색하라.
- A의 에지  $a_i$  또는 B의 에지  $b_j$ 가 포함되는 P의 에지를 검색하라.
- A의 에지  $a_i$  또는 B의 에지  $b_j$ 가 P에서 코드인지를 검사하라.
- A의 에지  $a_i$ 가 P에서 코드인 경우,  $a_i$ 의 머리 또는 꼬리가 놓인 P의 에지를 검색하라.
- B의 에지  $b_j$ 가 P에서 코드인 경우,  $b_j$ 의 머리 또는 꼬리가 놓인 P의 에지를 검색하라.
- P의 에지  $e_i$ 에 머리 또는 꼬리가 놓인 A의 에지를 검색하라.
- P의 에지  $e_i$ 에 머리 또는 꼬리가 놓인 B의 에지를 검색하라.

다각형 P의 두 가시성 다각형 A와 B의 교집합, 합집합, 차집합을 구하는 알고리즘은 모두 첫 번째 단계에서 A와 B의 에지들 중 서로 교차하는 에지 쌍 하나를 찾는다. 두 가시성 다각형이 서로 분리되어 있지 않다면 반드시 교차하는 에지 쌍이 존재함을 볼 수 있다. 이를 위해, A와 B의 경계선의 교차 여부를 판별하여 교차하는 경우 교차하는 에지 쌍 하나를 출력하는 알고리즘을

**알고리즘 2:** 교차 에지 쌍을 구하는 알고리즘

입력: P와 A, B

출력: A와 B의 교차 여부 및 교차 에지

1. // A의 경계선을 따라가면서 B의 경계선과 교차 여부를 검사한다.
2. while (A의 각 에지  $a_i$ 에 대해서,  $i = 0, \dots, r-1$ )
3.     if ( $a_i = ls(p, q)$ 가 P의 코드이면) // 포켓  $Po(p, q)$ 를 조사함.
4.          $e_{start}$ 와  $e_{end}$ 를 각각 p와 q가 놓인 P의 에지라고 하자;
5.         while (P의 각 에지  $e_j$ 에 대해서,  $j = start, \dots, end$ )
6.             if ( $e_j$ 와 교차하는 B의 에지  $b_k$ 를 만나면)
7.                 // B의 경계선을 따라가면서  $a_i$ 와 교차 여부를 검사한다.
8.                 while (B의 각 에지  $b_j$ 에 대해서,  $j = k, \dots, s-1, 0, \dots, k-1$ )
9.                     if ( $b_j$ 와  $a_i$ 가 교차하면)
10.                         교차 에지  $a_i$ 와  $b_j$ 의 정보를 출력하고 종료한다;
11.                     endif
12.                 endwhile
13.             “A와 B가 교차하지 않는다”는 정보를 출력하고 종료한다;
14.     endif
15.     endwhile
16.     else //  $a_i = ls(p, q)$ 가 P의 코드가 아닌 경우
17.         if ( $a_i$ 와 교차하는 B의 에지  $b_k$ 를 만나면)
18.             교차 에지  $a_i$ 와  $b_k$ 의 정보를 출력하고 종료한다;
19.         endif
20.     endifelse
21. endwhile
22. “A와 B가 교차하지 않는다”는 정보를 출력하고 종료한다;

먼저 소개한다. 다음의 알고리즘 2는 A와 B의 경계선의 교차 여부를 판별하여 교차하는 경우 교차하는 에지 쌍 하나를 출력한다.

A와 B가 서로 분리되어 있으면, 즉, 교차 에지 쌍이 존재하지 않는 경우는 앞 절의 마지막에 언급된 것처럼 A의 어떤 포켓에 B가 완전히 포함되어 있다. 이 경우는 A의 각 포켓 검사에 의해 결국 알고리즘 2의 6~14줄에서 검사된다. A와 B의 경계선이 교차하는 경우, A와 B의 교차 에지 쌍에는 다음의 두 가지 유형이 있다. 첫째는 두 에지가 서로 겹쳐 있는 유형이고, 둘째는 한 점에서 만나는 유형이다. 첫 번째 유형은 알고리즘의 17~19줄에서 검사되고, 두 번째 유형은 알고리즘의 9~11줄과 17~19줄에서 검사되므로 알고리즘 2는 올바르게 A와 B의 교차 에지 쌍을 구한다.

알고리즘 2의 각 연산은 상수 시간에 수행되고, 다각형의 각 에지는 기껏해야 한 번씩만 검사되기 때문에 알고리즘 2의 수행 시간은 선형 시간이다.

이제 A와 B의 교집합을 구하는 알고리즘을 살펴보자. A와 B는 P의 경계선의 일부를 항상 포함하고 있으므로, A와 B의 경계선이 서로 교차하지 않는다면 A와 B는 서로 분리된 것으로 판단할 수 있다. 따라서 A와 B의 교차 에지 쌍이 없으면 A와 B의 교집합은 공집합이다. 그렇지 않으면 초기 교차 에지 쌍에서부터 A와 B의 경계선을 동시에 따라가면서 교차 영역을 구한다. 이 과정은 알고리즘 3에 자세히 나와 있다.

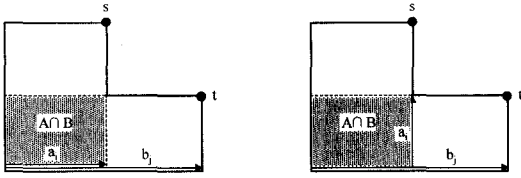
알고리즘 3은 초기 교차 에지 쌍에서부터 시작하여 그 다음 교차 에지 쌍을 계속 찾아나가는 방식으로 구성되어 있다. 두 에지가 서로 겹치는 경우는 그 다음 교차 에지 쌍은 그 인접한 에지들에 해당되고(6~14줄), 한 점에서 교차하는 경우는 해당되는 포켓을 검사하여 그 다음 교차 에지 쌍을 찾게 된다(15~27줄). 교차 영역을 구성하는 에지들을 구하기 위해, 두 에지가 겹치는 경우는 그 겹쳐지는 구간을, 두 에지가 한 점에서 만나는 경우는 더 안쪽에 위치한 에지(P에서의 코드)의 부

### 알고리즘 3: A와 B의 교집합을 구하는 알고리즘

입력: P와 A, B

출력: A와 B의 교집합

1. 알고리즘 2를 이용하여 A와 B의 교차 에지 쌍을 구한다. 만일 그러한 에지 쌍이
2. 존재하지 않으면 “공집합” 정보를 출력하고 종료한다.
3. 두 교차 에지를 각각  $a_p$ 와  $b_q$ 라고 두고, 두 변수  $i$ 와  $j$ 에 대해  $i = p, j = q$ 로 설정한다.
4. 선형 리스트 S를 생성한다; // S에는 교집합의 각 에지가 순서대로 저장된다.
5. repeat //  $a_p$ 와  $b_q$ 가 다시 방문될 때까지 반복함
6.     if ( $a_i$ 와  $b_j$ 가 서로 겹치면)     // 그림 7(a) 참조
7.          $a_i$ 와  $b_j$ 의 교차 구간에 해당되는 선분을 S에 저장한다;
8.         if ( $a_i$ 의 머리가  $b_j$ 의 외부에 위치하면)
9.              $j = j + 1$ ;
10.         else if ( $b_j$ 의 머리가  $a_i$ 의 외부에 위치하면)
11.              $i = i + 1$ ;
12.         else // 두 에지의 머리의 위치가 일치하면
13.              $i = i + 1$ ;  $j = j + 1$ ;
14.         endifelse
15.     else //  $a_i$ 와  $b_j$ 가 한 점에서 만나면, 그림 7(b) 참조
16.         if ( $b_j$ 의 머리가  $a_i$ 를 연장선 선의 오른쪽에 위치하면)
17.              $b_j$ 부터 B의 에지를 따라가면서 처음으로  $a_i$ 와 교차하는 에지  $b_k$ 를 찾는다.
18.             선분  $ls(x, y)$ 를 S에 저장한다, 여기서  $x$ 와  $y$ 는 각각  $a_i$ 와  $b_j$ 의 교차점과  $a_i$ 와  $b_k$ 의 교차점이다.
19.              $j = k$ 로 설정하고, 만일  $a_i$ 의 머리가  $b_j$  위에 위치하면  $i = i + 1$ ;
20.         else //  $a_i$ 의 머리가  $b_j$ 를 연장한 선의 오른쪽에 위치하면
21.              $a_i$ 부터 A의 에지를 따라가면서 처음으로  $b_j$ 와 교차하는 에지  $a_k$ 를 찾는다.
22.             선분  $ls(x, y)$ 를 S에 저장한다, 여기서  $x$ 와  $y$ 는 각각  $a_i$ 와  $b_j$ 의 교차점과  $a_k$ 와  $b_j$ 의 교차점이다.
23.              $i = k$ 로 설정하고, 만일  $b_j$ 의 머리가  $a_i$  위에 위치하면  $j = j + 1$ ;
24.         endifelse
25.     endifelse
26.     endifelse
27.     endifelse
28. until ( $i = p$  이고  $j = q$ )
29. S에 저장된 에지 집합을 출력하고 종료한다;



(a)  $a_i$ 와  $b_j$ 가 서로 겹침 (b)  $a_i$ 와  $b_j$ 가 한 점에서 교차함  
그림 7 가시성 다각형과 포켓, s와 t는 각각 A와 B의 가시원인

분을 저장하게 된다. 교차 영역은 하나의 연결된 영역이므로(정리 2), 초기 교차 에지 쌍을 다시 방문하게 되면 수행 과정은 종료된다.

알고리즘 3의 각 기본 연산은 상수 시간에 수행되고, 각 에지는 기껏해야 한 번씩만 검사되기 때문에 수행 시간은 선형 시간이다.

이제 A와 B의 합집합을 구하는 알고리즘을 살펴보자. A와 B의 교차 에지 쌍이 존재하지 않으면, 두 다각형은 서로 분리되어 있으므로 A와 B의 합집합은 두 개의 연결된 요소 즉, A와 B가 된다. 그렇지 않으면 합집합은 하나의 연결된 영역으로 구성된다. 합집합을 구하는 과정은 교집합을 구하는 알고리즘과 거의 동일하지만, 합집합 영역을 구성하는 에지들을 구하기 위해, 두 에지가

한 점에서 만나는 경우 해당되는 포켓의 내부에 있는 다각형의 에지들을 저장해야 한다. 이 과정은 알고리즘 4에 자세히 나와 있다.

알고리즘 4의 전체적인 구조와 기본 연산은 알고리즘 3과 거의 동일하므로, 알고리즘 4는 선형 시간에 합집합을 구한다는 것을 알 수 있다.

마지막으로 A와 B의 차집합을 구하는 알고리즘을 살펴보자. A와 B가 서로 분리되어 있으면 A와 B의 차집합은 바로 A가 된다. 그렇지 않으면 차집합은 하나 이상의 분리된 연결 요소로 구성된다는 것을 보았다. 차집합을 구하는 과정은 B의 각 포켓에 포함되어 있는 A의 경계선을 추출하여 저장하는 것이다. 이 과정은 알고리즘 5에 자세히 나와 있다.

알고리즘 5도 역시 전체적인 구조와 기본 연산은 알고리즘 3과 거의 동일하므로, 알고리즘 5가 선형 시간에 차집합을 구한다는 것을 알 수 있다.

알고리즘 3, 4, 5에 의해 다음의 정리가 성립한다.

**정리 4.** 단순 다각형 내부의 두 개의 가시성 다각형에 대한 교집합, 합집합, 차집합 연산은 선형 시간에 수행된다.

#### 4. 결론 및 연구 과제

##### 알고리즘 4: A와 B의 합집합을 구하는 알고리즘

입력: P와 A, B

출력: A와 B의 합집합

1. 알고리즘 2를 이용하여 A와 B의 교차 에지 쌍을 구한다. 만일 그러한 에지 쌍이 존재하지 않으면 합집합의 두 연결 요소 A와 B를 각각 출력하고 종료한다.
2. 두 교차 에지를 각각  $a_p$ 와  $b_q$ 라고 두고, 두 변수 i와 j에 대해  $i = p, j = q$ 로 설정한다.
3. 선형 리스트 S를 생성한다; // S에는 합집합의 각 에지가 순서대로 저장된다.
4. repeat //  $a_p$ 와  $b_q$ 가 다시 방문될 때까지 반복함
5.     if ( $a_i$ 와  $b_j$ 가 서로 겹치면)
6.         ... // 알고리즘 3의 7~14줄의 내용이 삽입됨
7.     else //  $a_i$ 와  $b_j$ 가 한 점에서 만나면
8.         if ( $b_j$ 의 머리가  $a_i$ 를 연장선 선의 오른쪽에 위치하면)
9.             선분  $ls(x, y)$ 를 S에 저장한다, 여기서 x는  $a_i$ 와  $b_j$ 의 교차점이고 y는  $b_j$ 의 머리이다.
10.              $b_j$ 부터 B의 에지를 따라가면서 처음으로  $a_i$ 와 교차하는 에지  $b_k$ 를 만날 때까지
11.             방문하는 각 B의 각 에지  $b_k$ 를 S에 저장한다,  $h = j+1, \dots, k-1$ .
12.             선분  $ls(x, y)$ 를 S에 저장한다, 여기서 x는  $b_k$ 의 꼬리이고 y는  $a_i$ 와  $b_k$ 의 교차점이다.
13.              $j = k$ 로 설정하고, 만일  $a_i$ 의 머리가  $b_j$  위에 위치하면  $i = i + 1$ ;
14.             else //  $a_i$ 의 머리가  $b_j$ 를 연장한 선의 오른쪽에 위치하면
15.             ... //  $a_i$ 와  $b_j$  그리고  $a_k$ 와  $b_k$ 가 서로 바뀌어 있는 것을 제외하면
16.             // 16~22줄까지의 내용과 동일한 문장이 삽입됨
17.         endifelse
18.     endifelse
19.     until ( $i = p$  이고  $j = q$ )
20.     S에 저장된 에지 집합을 출력하고 종료한다;



**알고리즘 5: A와 B의 차집합을 구하는 알고리즘**

입력: P와 A, B

출력: A와 B의 차집합

1. 알고리즘 2를 이용하여 A와 B의 교차 에지 쌍을 구한다. 만일 그러한 에지 쌍이
2. 존재하지 않으면 A를 출력하고 종료한다.
3. 두 교차 에지를 각각  $a_p$ 와  $b_q$ 라고 두고, 두 변수  $i$ 와  $j$ 에 대해  $i = p, j = q$ 로 설정한다.
4. 선형 리스트 S를 생성한다; // S에는 차집합의 각 에지가 순서대로 저장된다.
5. repeat //  $a_p$ 와  $b_q$ 가 다시 방문될 때까지 반복함
6.     if ( $a_i$ 와  $b_j$ 가 서로 겹치면)
7.         ... // 알고리즘 3의 8~14줄의 내용이 삽입됨, 즉, 7줄의 문장이 생략됨
8.     else //  $a_i$ 와  $b_j$ 가 한 점에서 만나면
9.         if ( $b_j$ 의 머리가  $a_i$ 를 연장선 선의 오른쪽에 위치하면)
10.              $b_j$ 부터 B의 에지를 따라가면서 처음으로  $a_i$ 와 교차하는 에지  $b_k$ 를 찾는다.
11.              $j = k$ 로 설정하고, 만일  $a_i$ 의 머리가  $b_j$  위에 위치하면  $i = i + 1$ ;
12.             else /\*  $a_i$ 의 머리가  $b_j$ 를 연장선 선의 오른쪽에 위치하면 \*/
13.              $a_i$ 부터 A의 에지를 따라가면서 처음으로  $b_j$ 와 교차하는 에지  $a_k$ 를 만날 때까지
14.             방문하는 각 A의 각 에지  $a_h$ 를 S에 저장한다,  $h = j+1, \dots, k-1$ .
15.             선분  $ls(x, y)$ 를 S에 저장한다, 여기서  $x$ 는  $a_k$ 의 꼬리이고  $y$ 는  $b_j$ 와  $a_k$ 의
16.             교차점이다.
17.             선분  $ls(x, y)$ 를 S에 저장한다, 여기서  $x$ 와  $y$ 는 각각  $b_j$ 와  $a_k$ 의 교차점과  $a_i$ 와
18.              $b_j$ 의 교차점이다.
19.              $i = k$ 로 설정하고, 만일  $b_j$ 의 머리가  $b_j$  위에 위치하면  $j = j + 1$ ;
20.         endifelse
21.     endifelse
22.     until ( $i = p$  이고  $j = q$ )
23.     S에 저장된 에지 집합을 출력하고 종료한다;

본 논문에서는 다각형 내부의 두 개의 가시성 다각형이 주어져 있을 때, 이 둘 사이의 교집합, 합집합, 차집합을 구하는 선형 시간의 알고리즘을 제시하였다. 출력의 크기가  $O(n)$ 인 두 다각형의 교집합을 구하는 문제에 대해 기존의 가장 효율적인 알고리즘의 시간 복잡도가  $O(n \log n)$ 인데 반해, 본 논문의 알고리즘은 가시성 다각형의 특성을 잘 반영함으로써 시간 복잡도를 개선하였다. 가시원이 점이나 에지인 경우 가시성 다각형을 구하는데 선형 시간이 소요되므로, 결국 다각형 내부의 두 개의 가시원이 주어져 있을 때, 이 들로부터 가시적인 영역, 동시에 가시적인 영역, 그리고 한 가시원에 의해 서만 가시적인 영역을 구하는 문제를 선형 시간에 해결할 수 있다.

앞으로의 연구 과제로 세 개 이상의 가시성 다각형이 주어져 있을 때, 이 들 사이의 집합 연산을 효율적으로 수행하는 알고리즘을 개발하는 것이다. 세 개 이상의 가시성 다각형들에 대한 집합 연산을 수행하기 위해서는 본 논문에서 사용했던 방법을 그대로 적용하기가 어렵다. 이 문제에 대한 효율적인 알고리즘을 설계하기 위해서는 가시성 다각형에 대한 새로운 특성들을 밝혀내는 것이 필요하다.

**참고 문헌**

- [1] F. D. Prepatata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, 1985.
- [2] E. ElGindy, D. Avis, and G. T. Toussaint, "A linear algorithm for computing the visibility polygon from a point," *J. Algorithms*, Vol. 2, pp. 190~197, 1981.
- [3] D. T. Lee, "Visibility of a simple polygon," *Computer Vision, Graphics, and Image Processing*, Vol. 22, pp. 207~221, 1983.
- [4] D. Avis and G. T. Toussaint, "an optimal algorithm for determining the visibility from an edge," *The Visual Computer*, Vol. 2, pp. 342~357, 1986.
- [5] G. T. Toussaint, "A linear time algorithm for solving the strong hidden line problem in a simple polygon," *Pattern Recognition Letters*, Vol. 4, pp. 449~451, 1986.
- [6] L. Guibas, J. Hershberger, D. Leven, M. Shri r, and R. E. Tarjan, "Linear time algorithm for visibility and shortest path problems inside triangulated simple polygons," *Algorithmica*, Vol. 2, pp. 209~233, 1987.
- [7] P. J. Heffeman and S. B. Mitchell, "Structured visibility profiles with applications to problems in simple polygons," *Proc. 16th Annual Symp. on Comput. Geometry*, pp. 53~62, 1990.
- [8] S. H. Kim and K. Y. Chwa, "An optimal algorithm for computing the edge visibility

- polygon from an edge of a simple polygon," *Proc. of KISS Conf.*, Vol. 17, No. 2, pp. 769~772, 1990.
- [9] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, 1987.
- [10] M. Mckenna, "Worst-case optimal hidden-surface removal," *ACM Tr. on Graphics*, Vol. 6, pp. 19~28, 1987.
- [11] S. S. Ahn, "Output Sensitive Polygon Intersection," MS thesis, KAIST, Korea, 1991.
- [12] H. G. Mairson and J. Stolfi, "Reporting and counting intersections in the plane," *Tech. Rep.*, Dept. of Computer Science, Stanford University, 1983.
- [13] G. T. Toussaint, "A simple linear algorithm for intersecting convex polygons," *The Visual Computer*, Vol. 1, pp. 118~123, 1985.



김 수 환

1987년 서울대학교 계산통계학과 학사  
 1989년 한국과학기술원 전산학과 석사  
 1995년 한국과학기술원 전산학과 박사  
 1992년~현재 부산외국어대학교 컴퓨터  
 공학부 부교수. 관심분야는 알고리즘 설  
 계와 분석, 계산기하학, 병렬 알고리즘