

단순 다각형의 두 에지 사이의 가시성 판별을 위한 상수 시간 RMESH 알고리즘

(A Constant Time RMESH Algorithm for Determining the Visibility between Two Edges of a Simple Polygon)

김 수 환 †

(Kim, Soo-Hwan)

요약 본 논문에서는 에지 가시성에 관련된 문제를 재구성가능한 메쉬(RMESH)에서 상수 시간에 해결하는 것을 고려한다. 에지 가시성에 관련된 기본적인 문제들로 다음의 세 가지 문제를 살펴볼 수 있다. 첫째, 주어진 다각형이 어떤 에지로부터 가시적인가를 판별하라. 둘째, 주어진 다각형을 볼 수 있는 모든 에지를 구하라. 셋째, 주어진 다각형에서 어떤 에지로부터의 가시적인 영역, 즉, 가시성 다각형을 구하라. 이들 문제를 상수 시간에 해결하기 위하여 본 논문에서는 두 에지 사이의 가시성에 관한 다음의 문제를 고려한다: 단순 다각형 P의 두 에지 e와 f가 주어졌 있을 때 e로부터 가시적인 f의 영역을 구하라. 본 논문에서는 이 문제를 $N \times N$ RMESH에서 상수 시간에 해결하는 알고리즘을 제시한다, 여기서 N은 P의 정점의 개수이다. 이 알고리즘을 이용하면 에지 가시성에 관련된 기본적인 문제들을 모두 RMESH에서 상수 시간에 해결할 수 있다. 특히, 세 번째 문제를 $N \times N^2$ RMESH에서 상수 시간에 해결하는 것이 가능하다.

키워드 : 재구성가능한 메쉬, 상수시간 알고리즘, 에지 가시성

Abstract In this paper, we consider the problems related to the edge visibility on a reconfigurable mesh(in short, RMESH). The following basic problems related to the edge visibility are considered: First, determine if a given polygon is visible from a specific edge. Second, find all edges from which a given polygon is visible. Third, compute the visibility polygon from a specific edge of a given polygon. In this paper, we consider the following problem in order to solve these problems in constant time: given two edges e and f of a simple polygon P, compute the maximal interval of f which is visible from e. We present a constant time algorithm for the problem on an $N \times N$ RMESH, where N is the number of vertices of P. Applying the algorithm, we can solve the above three problems in a constant time on a reconfigurable mesh. Specially, we can solve the third problem in a constant time on an $N \times N^2$ RMESH.

Key words : Reconfigurable Mesh, RMESH, Constant Time Algorithms, Edge Visibility

1. 서론

재구성가능한 메쉬(Reconfigurable Mesh; 줄여서 RMESH)는 실행 중에 동적으로 변하는 버스 시스템인 재구성가능한 버스 시스템으로 프로세서들을 메쉬 형태로 연결한 병렬 컴퓨터 모델이다[1, 2, 3]. RMESH의

가장 큰 특징이자 매력은 많은 문제를 상수 시간에 해결할 수 있다는 것이다. 다른 계산 모델에서 상수 시간에 해결되는 문제는 대부분 사소한 해를 갖는 것이지만, RMESH에서는 알고리즘적인 여러 기법을 통해 문제들을 다항 개수의 프로세서들만으로 상수 시간에 해결할 수 있다.

RMESH의 기본 구조는 프로세서들을 재구성가능한 버스 시스템에 의해 메쉬 형태로 연결된 것이다. 4×4 RMESH의 예가 그림 1에 나와 있다. 각 프로세서는 동(E), 서(W), 남(S), 북(N)의 4개 포트를 가지며, 알고리즘의 실행중에 버스 스위치에 의해 각 포트 사이를 연

· 본 연구는 한국과학재단 목적기초연구(R05-2001-000-00997-0) 지원으로 수행되었음.

† 통신회원 : 부산외국어대학교 컴퓨터공학부 교수
shkim@pufs.ac.kr

논문접수 : 2002년 5월 24일

심사완료 : 2002년 11월 1일

결하거나 또는 차단하는 것이 가능하다. 프로세서의 포트 연결을 적절히 조절하여 프로세서들을 여러 버스 조각(subbus)으로 분할할 수 있다. 한 순간에 하나의 프로세서만이 버스 조각에 대한 방송(broadcast)을 할 수 있고, 같은 버스 조각에 연결된 모든 프로세서들은 방송된 자료를 상수 시간에 읽을 수 있다. $N \times N$ RMESH의 각 프로세서는 $O(\log N)$ 비트 크기의 워드(word)를 $O(1)$ 개 저장할 수 있고, 사칙연산을 비롯한 기본 연산을 $O(1)$ 시간에 수행할 수 있다. 또한, 각 프로세서는 자신이 속한 행과 열을 인지할 수 있다.

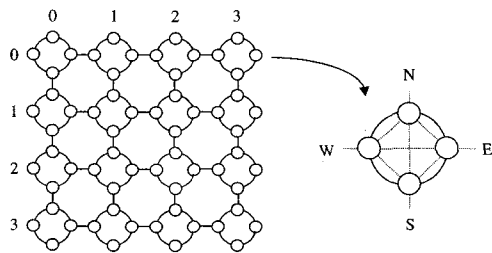


그림 1 4x4 RMESH의 구조

본 논문에서는 계산 기하학(computational geometry)의 기본적인 연구 분야에 해당되는 에지 가시성에 관한 문제를 다룬다. 다각형에 포함되는 두 점 p 와 q 가 있을 때, p 와 q 를 연결하는 선분이 P 에 완전히 포함되면 p 와 q 는 서로 볼 수 있다고 말한다. 에지 e 와 객체(점, 에지, 다각형 등) Q 가 있을 때 Q 의 각 점에 대해 그 점을 볼 수 있는 e 의 점이 존재하면, “ Q 는 e 로부터 약가시적(weakly visible)이다” 또는 그냥 간단히 “ Q 는 e 로부터 가시적이다” 또는 “ e 는 Q 를 볼 수 있다”라고 말한다[4, 5]. 에지 가시성에 관한 기본적인 문제는 다음과 같다.

1. 주어진 다각형이 어떤 에지로부터 가시적인지를 판별하라.
2. 주어진 다각형을 볼 수 있는 모든 에지를 찾아라.
3. 주어진 다각형에서 어떤 에지로부터의 가시적인 영역, 즉, 가시성 다각형을 구하라.

위 문제들에 대해 모두 선형 시간의 순차 알고리즘이 나와 있다[5, 6, 7, 8]. 그러나 에지 가시성에 관한 RMESH 알고리즘에 대해서는 아직 알려진 바가 없다. 본 논문에서는 서로 다른 두 에지 e 와 f 가 주어졌을 때, e 로부터 가시적인 f 의 영역을 구하는 상수 시간 RMESH 알고리즘을 제시한다. 이 알고리즘을 이용하면

위의 세 문제를 모두 상수 시간에 해결할 수 있다.

2. 용어 정의 및 에지 가시성의 성질

단순 다각형 P 가 평면 상에 놓여 있다고 하자. P 는 정점들이 반시계방향으로 주어진 순서리스트(v_0, v_1, \dots, v_{n-1})로 나타낸다. 두 정점 v_i 와 v_{i+1} 을 연결하는 에지는 e_i 또는 (v_i, v_{i+1}) 로 나타내고, 편의상 v_i 에서 v_{i+1} 방향의 유향 선분으로 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 경계선 상의 두 점 p, q 에 대해 p 에서 q 까지 반시계방향으로 따라간 경계선을 p 와 q 사이의 체인(chain)이라고 부르고 $ch(p, q)$ 로 나타낸다. 알고리즘에 대한 설명을 간편하게 하기 위해, 이 논문에서 다루어지는 단순 다각형은 어느 세 정점도 일직선상에 있지 않다고 가정한다.

단순 다각형 P 의 에지 e 와 점 q 가 주어졌을 때, q 로부터 가시적인 e 의 영역을 $vp(e, q)$ 로 나타낸다. P 의 두 에지 e 와 f 가 주어졌을 때, e 로부터 가시적인 f 의 영역을 $vp(f, e)$ 로 나타낸다.

단순 다각형의 한 점으로부터 가시성 다각형을 구하는 Kim의 알고리즘[9]을 이용하면 다음의 소정리 1이 성립한다.

소정리 1 [9]. 단순 다각형 P 의 에지 e 와 점 q 가 주어졌을 때, $vp(e, q)$ 는 $N \times N$ RMESH에서 $O(1)$ 시간에 구할 수 있다, 여기서 N 은 P 의 정점의 개수이다.

다각형 P 의 서로 다른 두 에지를 각각 $e_a = (v_a, v_{a+1})$ 와 $e_b = (v_b, v_{b+1})$ 이라고 하자. 이 절에서 다루는 문제는 e_a 가 볼 수 있는 e_b 의 영역, 즉, $vp(e_b, e_a)$ 를 구하는 것이다. v_{b+1} 과 v_a 를 연결한 선분을 L 이라고 하고, v_{a+1} 과 v_b 를 연결한 선분을 R 이라고 하자. $vp(e_b, e_a)$ 는 e_a 와 e_b 의 상대적인 위치에 따라 다음과 같이 결정된다.

소정리 2. L 과 R 이 교차하는 경우 e_b 가 e_a 를 연장한 선의 왼쪽 편에 놓여 있다면 $vp(e_b, e_a)$ 는 공집합 또는 e_b 의 양 끝점 중 하나이다.

(증명) L 과 R 이 교차한다면, $p \in e_a$ 와 $q \neq v_b$ 이고 $q \neq v_{b+1}$ 인 $q \in e_b$ 인 두 점 p, q 에 대해서, p 와 q 를 연결하는 선분은 $ch(v_{a+1}, v_b)$ 또는 $ch(v_{b+1}, v_a)$ 와 교차하게 되어 P 의 외부를 지나게 된다(그림 2 참조). 따라서 p 와 q 는 서로 볼 수 없다. 만일 $vp(e_a, v_b) \neq \emptyset$ 이면 $vp(e_b, e_a) = v_b$ 이고(그림 2 (a)), $vp(e_a, v_{b+1}) \neq \emptyset$ 이면 $vp(e_b, e_a) = v_{b+1}$ 이다(그림 2 (b)). □

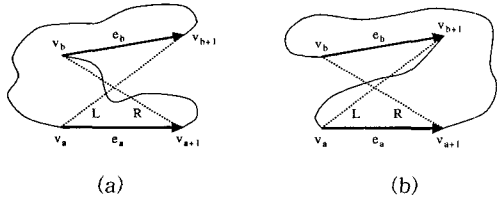


그림 2 소정리 2의 증명

소정리 3. L과 R이 교차하는 경우 e_b 가 e_a 를 연장한 선의 오른쪽 편에 놓여 있다면 $vp(e_b, e_a)$ 는 $vp(e_b, v_a)$ 이거나 $vp(e_b, v_{a+1})$ 이다.

(증명) 소정리 1의 증명에서 e_a 와 e_b 의 상대적인 위치를 서로 바꾸어 생각해 보면 된다. □

L과 R이 교차하지 않은 경우는 네 선분 e_a, R, e_b, L 은 사각형의 경계선을 형성한다. 이 사각형을 $Q = (v_a, v_{a+1}, v_b, v_{b+1})$ 이라고 하자.

소정리 4. 에지 e_a 의 방향이 Q의 경계선에서 볼 때 시계방향이면 $vp(e_b, e_a)$ 는 공집합 또는 e_b 의 양 끝점 중 하나이다.

(증명) 먼저, 두 점 p, q 가 $p \neq v_a$ 이고 $p \neq v_{a+1}$ 이며 $q \neq v_b$ 이고 $q \neq v_{b+1}$ 인 점들이라고 하면, p 와 q 는 서로 볼 수 없다는 것을 보인다. P의 경계선의 방향이 반시계방향이므로 그림 3과 같이 사각형 Q는 항상 P의 외부와 교차하게 된다. Q는 e_a 와 e_b 의 오른쪽 편에 놓여 있으므로 p 와 q 를 연결하는 선분 $l(p, q)$ 에서 p 와 q 에 근접한 점은 P의 외부에 위치해야 한다. 따라서 $l(p, q)$ 는 항상 P의 외부와 만나게 되므로 p 와 q 는 서로 볼 수 없다.

만일 선분 L이 P에 포함되면, v_a 는 v_{b+1} 을 볼 수 있고(그림 3 (a)), 선분 R이 P에 포함되면, v_{a+1} 은 v_b 를 볼 수 있다(그림 3 (b)). □

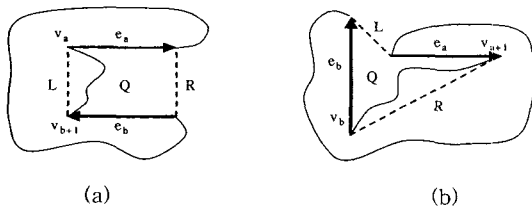


그림 3 소정리 4의 증명

소정리 5. 에지 e_a 의 방향이 Q의 경계선에서 볼 때 반시계방향이고 P의 경계선이 Q의 내부와 교차하지 않음

면 $vp(e_b, e_a)$ 는 e_b 이다.

(증명) 이 경우 Q는 P에 완전히 포함된다. 따라서 e_a 의 모든 점에서 e_b 의 모든 점을 볼 수 있다(그림 4 (a) 참조). □

소정리 6. P의 경계선이 아래의 (i)이나 (ii)에 해당되는 체인 $ch(v_i, v_j)$ 를 포함하고 있으면, 즉, Q의 내부를 관통하는 체인을 포함하고 있으면, $vp(e_b, e_a)$ 는 공집합이다.

(i) $ch(v_i, v_j) \subset ch(v_{b+1}, v_a)$ 이고 $e_i \cap L \neq \emptyset$ 이고 $e_{j-1} \cap R \neq \emptyset$ 이며 e_{j-1} 이 v_i 에서 출발하여 P의 경계선을 따라갈 때 처음으로 Q의 외부와 만나는 예지인 경우

(ii) $ch(v_i, v_j) \subset ch(v_{a+1}, v_b)$ 이고 $e_i \cap R \neq \emptyset$ 이고 $e_{i-1} \cap L \neq \emptyset$ 이며 e_{i-1} 이 v_i 에서 출발하여 P의 경계선을 따라갈 때 처음으로 Q의 외부와 만나는 예지인 경우

(증명) (i), (ii)에 해당되는 체인이 존재하면 $p \in e_a$ 와 $q \in e_b$ 인 두 점 p, q 에 대해서, p 와 q 를 연결하는 선분은 항상 P의 외부로 지나게 됨을 알 수 있다(그림 4 (b) 참조). 따라서 e_a 와 e_b 는 서로 볼 수 없다. □

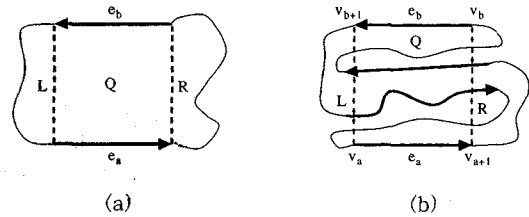


그림 4 Q를 관통하는 체인

이제, Q에서 e_a 의 방향이 반시계방향이고 Q의 내부와 P의 경계선이 교차하지만 Q의 내부를 관통하는 P의 체인이 없는 경우를 고려하자. $ch(v_{b+1}, v_a)$ 와 L의 교차점들을 $l_0=v_{b+1}, l_1, \dots, l_{s-1}=v_a$ 라고 하고, $ch(v_{a+1}, v_b)$ 와의 교차점들을 $r_0=v_{a+1}, r_1, \dots, r_{t-1}=v_b$ 라고 하자. 점 l_i 가 에지 e_i 에 놓여 있다면 $e(l_i)$ 를 e_j 로 정의하고, 점 r_j 가 에지 e_j 에 놓여 있다면 $e(r_j)$ 를 e_s 로 정의하자.

정의 1. 체인 $ch(l_i)$ 와 $ch(r_j)$ 를 다음과 같이 정의한다, $0 \leq i \leq s-1$ 이고 $0 \leq j \leq t-1$.

- (1) $e(l_i)$ 의 머리가 Q의 외부에 놓이면 $ch(l_i) = \emptyset$ 이다.
- (2) $e(l_i)$ 의 머리가 Q의 내부에 놓이면 $ch(l_i) = ch(l_i, l_k)$ 이다, 여기서 l_k 는 l_i 로부터 출발하여 P의 경계선을 반시계방향으로 따라갈 때 L과 맨 처음으로 만나는

점이다.

- (3) $e(r_j)$ 의 머리가 Q 의 외부에 놓이면 $ch(r_j) = \emptyset$ 이다.
- (4) $e(r_i)$ 의 머리가 Q 의 내부에 놓이면 $ch(r_j) = ch(r_j, r_k)$ 이다, 여기서 r_k 는 r_j 로부터 출발하여 P 의 경계선을 반시계방향으로 따라갈 때 R 과 맨 처음으로 만나는 점이다.

Q 의 내부를 관통하는 P 의 체인이 없으므로 모든 $ch(l_i)$ 와 $ch(r_j)$ 는 Q 에 포함된다(그림 5 참조).

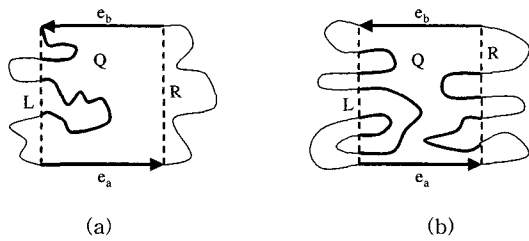


그림 5 $ch(l_i)$ 와 $ch(r_j)$ - 굵은 선으로 표현된 체인

정점 집합 C 를 v_a 와 v_{b+1} , 그리고 $ch(l_i)$ 에 놓여 있는 P 의 정점들로 구성된 집합이라고 하고, 정점 집합 D 를 v_{a+1} 과 v_b , 그리고 $ch(r_j)$ 에 놓여 있는 P 의 정점들로 구성된 집합이라고 하자, $0 \leq i \leq s-1$ 이고 $0 \leq j \leq t-1$. 집합 C 에 대한 볼록외피(convex hull)를 $CH(C)$ 라고 하고, 집합 D 에 대한 볼록외피를 $CH(D)$ 라고 하자.

소정리 7. 두 볼록외피 $CH(C)$ 와 $CH(D)$ 가 교차하면 $vp(e_b, e_a)$ 는 공집합이다.
(증명) 두 볼록외피가 서로 교차하면 두 에지 e_a 와 e_b 는 서로 볼 수 없다는 것은 자명하다(그림 6 (a) 참조). □

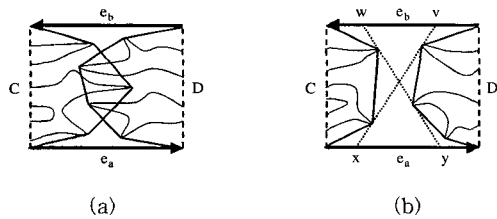


그림 6 볼록외피 $CH(C)$ 와 $CH(D)$

소정리 8. 두 볼록외피 $CH(C)$ 와 $CH(D)$ 가 서로 교차하지 않은 경우, 두 볼록외피에 동시에 접하는 두 접선과 e_b 가 만나는 점을 각각 v, w 라고 하자, 여기서 w 는

v 보다 e_b 의 머리에 가까운 점이다. 이 때, $vp(e_b, e_a)$ 는 선분 (v, w) 이다.

(증명) 두 볼록외피에 동시에 접하는 두 접선과 e_a 가 만나는 점을 각각 x, y 라고 하자, 여기서 y 는 x 보다 e_a 의 머리에 가까운 점이다. 그러면 선분 (x, y) 와 선분 (v, w) 는 서로 가시적임을 알 수 있다(그림 6 (b) 참조). □

3. 알고리즘

본 절에서는 단순 다각형 P 의 서로 다른 두 에지 e_a 와 e_b 가 주어졌을 때, e_a 가 볼 수 있는 e_b 의 영역, 즉, $vp(e_b, e_a)$ 를 구하는 상수 시간 $N \times N$ RMESH 알고리즘을 제시한다, 여기서 N 은 다각형 P 의 정점의 개수이다. 앞으로 RMESH RM에서의 i 번째 행과 j 번째 열에 위치한 프로세서를 $RM(i, j)$ 로 표시한다. $RM(\cdot, j)$ 는 열 번호가 j 인 모든 프로세서를 나타내고, $RM(i, \cdot)$ 은 행 번호가 i 인 모든 프로세서를 나타낸다.

알고리즘의 입력으로 다각형 $P = (v_0, v_1, \dots, v_{N-1})$ 과 두 에지 e_a, e_b 가 주어진다고 하자. 알고리즘이 시작되기 전의 RMESH의 상태와 종결 후의 상태는 다음과 같다.

- 초기 상태:** P 의 각 에지가 $N \times N$ RMESH RM의 0번째 열에 차례로 배치된다. 즉, $RM(i, 0)$ 에는 e_i 가 배치된다, $0 \leq i \leq N-1$. 그리고 $RM(0, 0)$ 에 e_a 와 e_b 가 배치된다.
- 종결 상태:** $RM(0, 0)$ 에 $vp(e_b, e_a)$ 가 배치된다.

알고리즘의 개요는 다음과 같다(용어는 2절 참조).

알고리즘 1: 서로 다른 두 에지 사이의 가시 영역 계산

1. $RM(0, 0)$ 은 두 선분 $L = (v_a, v_{b+1})$ 과 $R = (v_{a+1}, v_b)$ 의 교차 여부를 검사한다. L 과 R 이 교차하지 않으면 단계 2를 수행한다. 그렇지 않은 경우, RM은 소정리 2와 3의 조건에 따라 $vp(e_b, e_a)$ 를 구해 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다.
2. $RM(0, 0)$ 은 사각형 $Q = (v_a, v_{a+1}, v_{b+1}, v_b)$ 에서 e_a 의 방향을 검사한다. e_a 의 방향이 Q 에서 반시계방향이면 단계 3을 수행한다. 그렇지 않은 경우, RM은 소정리 4의 조건에 따라 $vp(e_b, e_a)$ 를 구해 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다.
3. RM은 P 의 경계선이 Q 의 내부와 교차하는가를 검사한다. 교차하면 단계 4를 수행한다. 그렇지 않은 경우, $vp(e_b, e_a) = (v_b, v_{b+1})$ 을 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다.

4. RM은 Q의 내부를 관통하는 P의 체인이 있는가를 검사한다. 그러한 체인이 존재하면 $vp(e_b, e_a) = \emptyset$ 을 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다. 그렇지 않은 경우, 정점 집합 C와 D를 구한다.
5. 두 블록외피 $CH(C)$ 와 $CH(D)$ 를 구한 후 $CH(C)$ 와 $CH(D)$ 의 교차 여부를 검사한다. 두 블록외피가 교차하면 $vp(e_b, e_a) = \emptyset$ 을 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다. 그렇지 않은 경우, RM은 두 블록외피의 두 접선을 구한다. 그리고 두 접선과 e_b 의 교차점 v, w 를 구한다. 마지막으로 $vp(e_b, e_a) = (v, w)$ 를 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다.

정리 1. 단순 다각형 P와 두 에지 e_a 와 e_b 가 주어졌을 때, $vp(e_b, e_a)$ 를 $N \times N$ RMESH에서 상수 시간에 구할 수 있다, 여기서 N은 P의 정점의 개수이다.

(증명) 알고리즘 1의 각 단계를 수행하는 상세한 과정의 설명과 수행 시간의 분석을 통해 $N \times N$ RMESH에서 $vp(e_b, e_a)$ 를 상수 시간에 구할 수 있음을 보인다.

단계 1: /* 소정리 2와 3의 조건에 따른 $vp(e_b, e_a)$ 의 계산 */

- 1.1 $RM(0, 0)$ 은 두 선분 $L = (v_a, v_{b+1})$ 과 $R = (v_{a+1}, v_b)$ 를 구한 후 교차 여부를 검사한다. 교차하지 않으면 단계 2를 수행한다.
- 1.2 $RM(0, 0)$ 은 e_a 와 e_b 의 상대적인 위치를 검사한다.
- 1.3 e_b 가 e_a 를 연장한 선의 오른쪽 편에 놓여 있으면 단계 1.4를 수행한다. 그렇지 않으면, RM은 $vp(e_a, v_b)$ 와 $vp(e_a, v_{b+1})$ 을 구한다. $vp(e_a, v_b)$ 가 공집합이 아니면 $vp(e_b, e_a) = v_b$ 로 두고, $vp(e_a, v_{b+1})$ 이 공집합이 아니면 $vp(e_b, e_a) = v_{b+1}$ 로 둔다. 둘 다 공집합이면 $vp(e_b, e_a) = \emptyset$ 로 둔다. RM은 $vp(e_b, e_a)$ 를 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다.
- 1.4 RM은 $vp(e_b, v_a)$ 와 $vp(e_b, v_{a+1})$ 을 구한다. $vp(e_b, v_a)$ 가 공집합이면 $vp(e_b, e_a) = vp(e_b, v_{a+1})$ 로 두고, 그렇지 않으면 $vp(e_b, e_a) = vp(e_b, v_a)$ 로 둔다. RM은 $vp(e_b, e_a)$ 를 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다.

소정리 2와 3에 의해 단계 1이 올바르게 수행된다는 것을 알 수 있다. 단계 1.3과 1.4에서 $vp(e, p)$ 를 구하는 것은, e 는 에지이고 p 는 정점, 소정리 1에 의해 $N \times N$ RMESH에서 상수 시간에 수행됨을 알 수 있다. 따라서 단계 1은 상수 시간에 수행된다.

단계 2: /* 소정리 4의 조건에 따른 $vp(e_b, e_a)$ 의 계산 */

- 2.1 $RM(0, 0)$ 은 사각형 $Q = (v_a, v_{a+1}, v_{b+1}, v_b)$ 에서 e_a 의 방향을 검사한다.
- 2.2 e_a 의 방향이 Q에서 반시계방향이면 단계 3을 수행한다. 그렇지 않은 경우, RM은 $vp(e_a, v_b)$ 와 $vp(e_a, v_{b+1})$ 을 구한다. $vp(e_a, v_b)$ 가 공집합이 아니면 $vp(e_b, e_a) = v_b$ 로 두고, $vp(e_a, v_{b+1})$ 이 공집합이 아니면 $vp(e_b, e_a) = v_{b+1}$ 로 둔다. 둘 다 공집합이면 $vp(e_b, e_a) = \emptyset$ 로 둔다. RM은 $vp(e_b, e_a)$ 를 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다.

소정리 4에 의해 단계 2가 올바르게 수행됨을 알 수 있다. 단계 2.2는 단계 1.3과 동일한 방식으로 수행됨으로 단계 2는 상수 시간에 수행된다.

단계 3: /* 소정리 5의 조건에 따른 $vp(e_b, e_a)$ 의 계산 */

- 3.1 RM은 $Q = (v_a, v_{a+1}, v_{b+1}, v_b)$ 와 e_j 를 각 $RM(0, j)$ 에 배치한다, $0 \leq j \leq N-1$.
- 3.2 각 $RM(0, j)$ 는 e_j 와 Q의 교차 여부를 검사한다, $0 \leq j \leq N-1$.
- 3.3 각 $RM(0, j)$ 는 스위치 연결 상태를 $\{EW, S, N\}$ 로 초기화한다, $0 \leq j \leq N-1$.
- 3.4 Q와 교차하는 에지 e_j 를 가진 $RM(0, j)$ 는 자신의 E 포트를 끊고 특별한 기호를 발송한다, $0 \leq j \leq N-1$.
- 3.5 $RM(0, 0)$ 가 특별한 기호를 읽으면 단계 4를 수행한다. 그렇지 않은 경우, $vp(e_b, e_a) = (v_b, v_{b+1})$ 로 둔다. RM은 $vp(e_b, e_a)$ 를 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다.

단계 3.3에서 스위치 연결 상태를 $\{EW, S, N\}$ 으로 둔다는 것은 RMESH의 버스(bus) 연결을 수평 방향으로만 설정하고 수직 방향은 단절된 상태로 둔다는 것을 의미한다[10]. 단계 3.1에서 3.4까지는 P의 경계선과 Q가 교차하는지를 검사하는 과정이다. 이것은 P의 각 에지와 Q와의 교차 여부를 검사함으로써 수행된다. 단계 3.4에서 Q와 교차하는 에지 e_j 를 가진 $RM(0, j)$ 가 자신의 E 포트를 끊는 것은 RMESH의 동일한 버스 조각에는 주어진 시간에 단 하나의 프로세서만 발송할 수 있다는 가정을 만족시키기 위한 것이다. Q와 교차하는 에지가 하나라도 있으면 결국 $RM(0, 0)$ 은 특별한 기호를 읽을 수 있다. 교차하는 에지가 하나도 없으면 소정리 5에 따라 $vp(e_b, e_a) = (v_b, v_{b+1})$ 이 된다. 각 단계가 모두 상수 시간에 수행되기 때문에 단계 3은 상수 시간에 수행된다.

단계 4: /* Q를 관통하는 체인 검사와 정점 집합 C와 D의 계산 */

4.1 RM은 Q와 e_j 를 각 $RM(i, j)$ 에 배치한다, $0 \leq i \leq N-1, 0 \leq j \leq N-1$.

4.2 /* Q를 관통하는 체인 검사 및 정점 집합 C의 계산 */

4.2.1 모든 프로세서는 스위치 연결 상태를 {EW, S, N}으로 초기화한다.

4.2.2 각 $RM(i, j)$ 는 $e_j \subset ch(v_{b+1}, v_a)$ 이고 $e_j \cap L \neq \emptyset$ 이며 v_{j+1} 이 유향 선분 $L = (v_{b+1}, v_a)$ 을 연장한 선의 왼쪽에 놓이면 W 포트를 끊고, v_{j+1} 이 Q의 외부에 놓이면 E 포트를 끊는다, $0 \leq i \leq N-1, 0 \leq j \leq N-1$.

4.2.3 각 $RM(i, j)$ 는 자신의 W 포트가 끊어져 있으면 기호 'L'을 방송한다, $0 \leq i \leq N-1$. 모든 프로세서는 방송된 기호를 읽는다. 그 다음 각 $RM(i, j)$ 는 자신의 E 포트가 끊어져 있으면 기호 'L'을 방송한다, $0 \leq i \leq N-1$. 모든 프로세서는 방송된 기호를 읽는다.

4.2.4 방송된 기호 'L'을 읽은 $RM(i, j)$ 는 e_j 와 $R = (v_{a+1}, v_b)$ 와의 교차 여부를 검사한다, $0 \leq i \leq N-1, 0 \leq j \leq N-1$. RM은 단계 3.3~3.4와 같은 방식으로 교차 정보를 각 $RM(i, 0)$ 에 전달한다, $0 \leq i \leq N-1$.

4.2.5 각 $RM(i, 0)$ 는 스위치 연결 상태를 {E, W, SN}로 초기화한다, $0 \leq i \leq N-1$. Q와 교차하는 에지 e_i 를 가진 $RM(i, 0)$ 는 자신의 S 포트를 끊고 특별한 기호를 방송한다, $0 \leq i \leq N-1$. $RM(0, 0)$ 가 이 기호를 읽으면, RM은 $vp(e_b, e_a) = \emptyset$ 를 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다.

4.2.6 각 $RM(i, j)$ 는 기호 'L'을 읽었고 $v_j \in Q$ 이면 $RM(0, j)$ 에 v_j 를 배치하고, 그렇지 않으면 $RM(0, j)$ 에 \emptyset 를 배치한다, $0 \leq i \leq N-1, 0 \leq j \leq N-1$. RM은 0번째 행에 배치된 정점들 중 공집합이 아닌 정점들과 v_a, v_{b+1} 을 RM의 0번째 열에 배치한다.

4.3 /* Q를 관통하는 체인 검사 및 정점 집합 D의 계산 */

4.2절과 동일한 방식으로 수행하여 Q를 관통하는 체인이 존재하면 $vp(e_b, e_a) = \emptyset$ 를 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다. 그렇지 않

으면 정점 집합 D를 구해 RM의 0번째 열에 배치한다.

단계 4.2.2와 4.2.3은 $ch(v_{b+1}, v_a)$ 에서 Q를 관통하는 체인과 정점 L의 $ch(l_i)$ 를 찾는 과정을 나타낸 것이다. 4.2.2에서 W 포트와 E 포트를 끊는 이유는 사각형 Q의 내부에 관통하는 P의 체인 또는 $ch(l_i)$ 에 해당되는 P의 체인들에 대해 그림 7과 같이 RMESH에서 버스 조각들을 형성하기 위한 것이다.

$i > j$ 인 체인 $ch(i, j)$, 즉, $ch(i, j) = ch(i, N-1) \cup ch(0, j)$ 이 존재할 때, 이 체인에 해당되는 버스 조각은 RMESH의 왼쪽과 오른쪽 두 부분으로 분리된다. 단계 4.2.3에서 두 번의 방송을 하는 이유는 이 분리된 버스 조각에 대해서도 기호 'L'을 올바르게 방송하기 위해서이다.

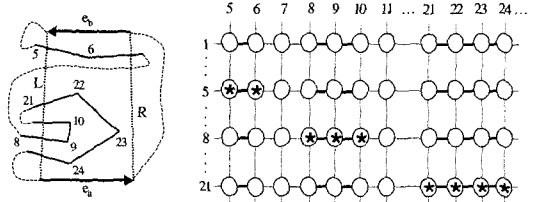


그림 7 Q를 관통하는 체인 및 $ch(l_i)$ 들

$ch(v_i, v_j)$ 가 Q를 관통하는 체인이거나 $ch(l_i)$ 에 해당된다면 $RM(i, k), i \leq k \leq j-1$,는 기호 'L'을 읽게 된다(그림 7 참조). 단계 4.2.4는 소정리 6의 조건이 만족하는가를 검사하는 과정이다. 소정리 6의 조건을 만족하면 $vp(e_b, e_a) = \emptyset$ 이 된다. $N \times N$ RMESH의 한 행에 있는 원소들을 임의의 순서대로 재배치하는 것은 상수 시간에 수행될 수 있기 때문에[10], 단계 4.2.6에서 정점 집합 C에 속하는 정점들을 RM의 0번째 열에 배치하는 것은 상수 시간에 수행된다. 따라서 단계 4의 각 과정은 상수 시간에 수행된다.

단계 5: /* 소정리 7과 소정리 8의 조건에 따른 $vp(e_b, e_a)$ 의 계산 */

5.1 RM의 0번째 열에 저장된 정점 집합 C에 대해서 블록외피를 구해 RM의 0번째 열에 배치한다.

5.2 RM의 0번째 열에 저장된 정점 집합 D에 대해서 블록외피를 구해 RM의 0번째 열에 배치한다.

5.3 두 불록다각형 $CH(C)$ 와 $CH(D)$ 가 교차하는지를 검사한다.

5.4 두 불록다각형이 교차하지 않으면 단계 5.5를 수

행한다. 교차하는 경우, RM은 $vp(e_b, e_a) = \emptyset$ 를 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다.

5.5 두 블록다각형 $CH(C)$ 와 $CH(D)$ 의 두 접선 L_1 과 L_2 를 구한다.

5.6 $RM(0, 0)$ 은 e_b 와 L_1, L_2 와의 교차점 v, w 를 각각 구한다.

5.7 RM은 $vp(e_b, e_a) = (v, w)$ 를 $RM(0, 0)$ 에 배치하고 알고리즘을 종료한다.

0번째 열에 배치된 점들에 대한 블록의피는 $N \times N$ RMESH에서 상수 시간에 구할 수 있다[10]. 또한, 블록 다각형의 교차 여부도 Kim[11]의 알고리즘을 이용하면 $N \times N$ RMESH에서 상수 시간에 판별할 수 있다. 두 블록다각형의 두 접선은 Miller 등[12]의 알고리즘을 이용하면 $N \times N$ RMESH에서 상수 시간에 구할 수 있다. 따라서, 단계 5의 각 과정은 상수 시간에 수행된다.

위의 과정을 통해 $N \times N$ RMESH에서 $vp(e_b, e_a)$ 를 상수 시간에 올바르게 구한다는 것이 증명된다. □

4. 결 론

본 논문에서는 두 에지 사이의 가시 영역을 구하는 상수 시간 $N \times N$ RMESH 알고리즘을 제시하였다. 이 알고리즘을 이용하면 1절에서 언급한 에지 가시성에 관한 세 문제들을 모두 상수 시간에 해결할 수 있다. 특히, 알고리즘 1은 세 번째 문제를 해결하는데 의미가 있다. 단순 다각형 P에서 주어진 에지 e로부터의 가시적인 영역 $V(P, e)$ 를 구하는 것은 다음과 같다:

1. P의 모든 에지 e_i 에 대해 $vp(e_i, e)$ 를 구한다, $0 \leq i \leq N-1$.
2. 공집합 아닌 $vp(e_i, e)$ 들의 양 끝점들을 순서대로 나열한다, 이 때 중복된 정점들을 삭제한다(그림 8 참조), $0 \leq i \leq N-1$.

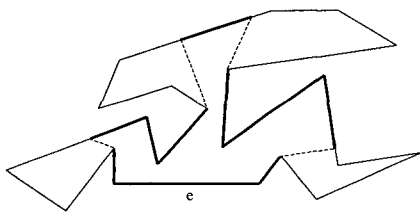


그림 8 $V(P, e)$

단계 1에서 N개의 에지들에 대해 동시에 알고리즘 1을 수행하기 위해서는 $N \times N^2$ 개의 프로세서가 필요하다. 단계 2에서 $vp(e_i, e)$ 들의 양 끝점들을 중복없이 차례대

로 나열하는 것은 $N \times N$ RMESH에서 상수 시간에 해결할 수 있다[9].

첫 번째 문제는 $V(P, e) = P$ 인가를 검사하면 해결되니까 역시 $O(N^2)$ 개의 프로세서만 있으면 충분하다. 두 번째 문제를 상수 시간에 해결하는 것은 P의 모든 에지에 대해 동시에 첫 번째 문제를 풀면 되므로 $O(N^4)$ 개의 프로세서만 있으면 충분하다.

에지 가시성에 관한 세 문제들은 순차 알고리즘으로 모두 선형 시간에 해결된다. 시간복잡도는 동일하지만 알고리즘의 복잡성이나 난이도를 고려할 때 첫 번째 문제가 가장 쉽고, 세 번째 문제가 가장 어렵다고 볼 수 있다. 특히, 세 번째 문제인 $V(P, e)$ 를 구하는 문제는 P에 대한 삼각분할이 선행되어야만 선형 시간에 해결될 수 있다. 선형 시간에 다각형을 삼각 분할하는 알고리즘은 매우 복잡한 것으로 알려져 있다[13]. 이 들 세 문제를 상수 시간에 해결하는데 있어 프로세서의 수를 얼마나 줄일 수 있는가를 탐구하는 것은 흥미로운 연구 과제이다. 특히, 첫 번째와 두 번째 문제는 세 번째 문제에 비해 비교적 쉬운 문제로 볼 수 있기 때문에 프로세서의 수를 줄이는 것이 가능하리라 기대된다.

참 고 문 헌

- [1] Y. Ben-Asher, D. Peleg, R. Ramaswami, and A. Schuster, "The power of reconfiguration," *J. Parallel and Distrib. Comput.*, Vol 13, pp. 139-153, 1991.
- [2] R. Miller, V. K. Prasanna Kumar, D. Reisis, and Q. Stout, "Meshes with reconfigurable buses," *Proc. 5th MIT Conf. on Adv. Res. in VLSI*, pp. 163-178, 1988.
- [3] B. Wang and G. Chen, "Constant time algorithms for the transitive closure problem and some related graph problems with reconfigurable bus systems," *IEEE Trans. Parallel and Dist. System*, Vol. 1, pp. 500-507, 1990.
- [4] D. Avis, T. Gum, G. Toussaint, "Visibility between two edges of a simple polygon," *Visual Computer*, Vol. 2, pp. 342-357, 1986.
- [5] D. Avis and G. Toussaint, "An optimal algorithm for determining the visibility of a polygon from an edge," *IEEE Trans. Comput.*, Vol. C-30, pp. 910-914, 1981.
- [6] 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, pp. 209-233, 1987.
- [7] S. Y. Shin and T. C. Woo, "An optimal algorithm for

- finding all visible edges in a simple polygon," *IEEE Trans. on Robotics and Automation*, Vol. 5, pp. 202-207, 1989.
- [8] G. Toussaint, "A linear time algorithm for solving the strong hidden line problem in a simple polygon," *Pattern Recog. Letters*, Vol. 4, pp. 449-451, 1986.
- [9] 김수환, "구멍이 있는 다각형에서 가시성 다각형을 구하는 상수 시간 RMESH 알고리즘," 한국정보과학회, 제27회 추계학술발표논문집, 제27권 2호, 2000년 10월.
- [10] J.-W. Jang, M. Nigam, V. K. Prasanna, and S. Sahni, "Constant time algorithms for computational geometry on the reconfigurable Mesh," *IEEE Trans. on Parallel and Dist. System*, Vol. 8, pp. 1-12, 1997.
- [11] 김수환, "다각형 교차 문제를 위한 상수 시간 재구성메쉬 알고리즘," 한국정보과학회, 정보과학회 논문지, 제26권 11호, pp. 1344-1352, 1999.
- [12] R. Miller and Q. Stout, "Mesh computer algorithms for computational geometry," *IEEE Trans. Computers*, Vol. 38, pp. 321-340, 1989.
- [13] B. Chazelle, "Triangulating a simple polygon in linear time," *Proc. 31st Annual Symp. on FOCS*, Vol. I, pp. 220-230, 1990.



김 수 환

1987년 서울대학교 계산통계학과 학사.
 1989년 한국과학기술원 전산학과 석사.
 1995년 한국과학기술원 전산학과 박사.
 1992년 ~ 현재 부산외국어대학교 컴퓨터공학부 부교수. 관심분야는 알고리즘 설계와 분석, 계산기하학, 병렬 알고리즘.