

두 가시성 다각형의 교집합과 합집합을 구하는 효율적인 RMESH 알고리즘

김수환*

Efficient RMESH Algorithms for Computing the Intersection and the Union of Two Visibility Polygons

Soo-Hwan Kim*

Department of Embedded Software, Busan University of Foreign Studies, Busan 46234, Korea

요 약

단순 다각형 내부의 두 점 p 와 q 가 주어질 때 다음의 문제를 고려할 수 있다. (1) 다각형에서 p 와 q 둘 다로부터 가시적인 점들의 집합을 구하라. (2) p 와 q 둘 중의 적어도 하나로부터 가시적인 점들의 집합을 구하라. 이 문제들은 두 가시성 다각형 사이의 교집합과 합집합을 구하는 문제에 해당한다. 본 논문에서는 재구성가능한 메쉬(RMESH)에서 이 문제들을 해결하는 알고리즘을 고려한다. 일반적인 두 다각형의 교차 영역을 구하는 알고리즘[1]을 이용하면, 두 가시성 다각형의 교집합을 구하는 문제를 $O(n^3)$ 크기의 RMESH에서 상수 시간에 해결할 수 있다. 여기서 n 은 두 가시성 다각형의 꼭짓점 개수의 합이다. 본 논문에서는 가시성 다각형의 특성을 이용하여 평면 분할 그래프를 $O(n^2)$ 크기의 RMESH에서 상수 시간에 구축하고, 이를 통해 두 가시성 다각형의 교집합뿐만 아니라 합집합도 $O(\log^2 n)$ 시간에 구하는 알고리즘을 제시한다. 이 알고리즘은 기존의 결과에 비해 병렬 알고리즘의 비용을 나타내는 프로세서-시간 곱 지표를 $O(n^3)$ 에서 $O(n^2 \log^2 n)$ 으로 개선한다.

ABSTRACT

We can consider the following problems for two given points p and q in a simple polygon P . (1) Compute the set of points of P which are visible from both p and q . (2) Compute the set of points of P which are visible from either p or q . They are corresponding to the problems which are to compute the intersection and the union of two visibility polygons. In this paper, we consider algorithms for solving these problems on a reconfigurable mesh (in short, RMESH). The algorithm in [1] can compute the intersection of two general polygons in constant time on an RMESH with size $O(n^3)$, where n is the total number of vertices of two polygons. In this paper, we construct the planar subdivision graph in constant time on an RMESH with size $O(n^2)$ using the properties of the visibility polygon for preprocessing. Then we present $O(\log^2 n)$ time algorithms for computing the union as well as the intersection of two visibility polygons, which improve the processor-time product from $O(n^3)$ to $O(n^2 \log^2 n)$.

키워드 : 병렬 알고리즘, 재구성가능한 메쉬, 가시성 다각형, 평면 분할

Key word : parallel algorithm, reconfigurable mesh, visibility polygon, planar subdivision

Received 14 December 2015, Revised 31 December 2015, Accepted 14 January 2016

* Corresponding Author Soo-Hwan Kim(E-mail:shkim@bufs.ac.kr, Tel:+82-51-509-6243)

Department of Embedded Software, Busan University of Foreign Studies, Busan 46234, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2016.20.2.401>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

재구성가능한 메쉬(Reconfigurable Mesh; 줄여서 RMESH)는 1988년 Miller 등[1]에 의해 처음 소개된 병렬 계산 모델로서, 그래프 이론, 계산기하학, 영상처리 등의 분야에서 RMESH 모델을 대상으로 한 병렬 알고리즘이 많이 발표되고 있다[2-4]. RMESH의 기본 구조는 격자 형태로 배치된 프로세서들을 재구성가능한 버스 시스템으로 연결한 것이다. 각 프로세서는 동(E), 서(W), 남(S), 북(N)의 4개 포트를 가지며, 알고리즘의 실행 중에 버스 스위치에 의해 각 포트를 연결하거나 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 방송 자료를 상수 시간에 읽을 수 있다. $N \times N$ 크기를 가진 RMESH의 각 프로세서는 $O(\log N)$ 비트 크기의 기억공간을 상수 개 가질 수 있고, 사칙 연산을 비롯한 기본 연산을 상수 시간에 수행할 수 있다. 또한 각 프로세서는 자신이 속한 행과 열을 인지할 수 있다. 그림 1은 4×4 RMESH와 프로세서의 스위치 구조를 보여준다.

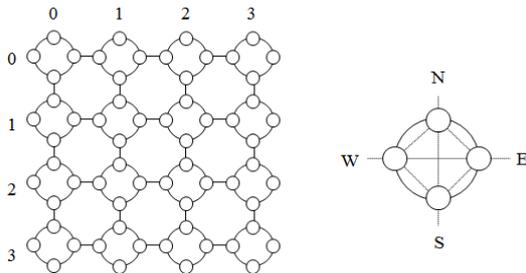


Fig. 1 An 4×4 RMESH and a processor

다각형 내부의 주어진 두 점을 연결하는 선분이 다각형 외부와 교차하지 않으면, 그 두 점은 서로 가시적이라고 말한다. 물체들 사이의 가시성에 관한 문제는 그래픽스, 로보틱스, 영상처리 등의 분야에서 많이 발생한다. 다각형 내부의 한 점 p 로부터 가시적인 점들의 집합을 p 로부터의 가시성 다각형이라고 부른다. 기존 결과로서 n 개의 꼭짓점을 가진 다각형에서 주어진 한 점으로부터의 가시성 다각형을 구하는 선형 시간 순차 알

고리즘[5]과 상수시간 $n \times n$ RMESH 알고리즘이 나와 있다[6]. 다각형 P 내부의 두 점 p 와 q 가 주어질 때, p 와 q 둘 다로부터 가시적인 P 의 영역은 p 와 q 로부터의 가시성 다각형들의 교집합에 해당한다. 또한 p 와 q 둘 중의 적어도 하나로부터 가시적인 P 의 영역은 두 가시성 다각형의 합집합에 해당한다(그림 2 참조).

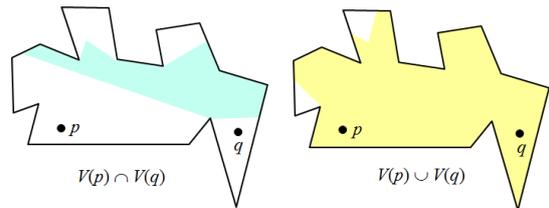


Fig. 2 Examples of the intersection and the union of two visibility polygons $V(p)$ and $V(q)$

본 논문에서는 RMESH 병렬 계산 모델에서 두 가시성 다각형의 교집합과 합집합을 구하는 문제를 고려한다. 이 문제에 대해 선형 시간에 수행되는 순차 알고리즘이 나와 있다[7]. 또한 일반적인 두 다각형의 교차 영역을 구하는 문제를 $(n+t) \times (n+t)^2$ RMESH에서 상수 시간에 해결하는 알고리즘이 나와 있다[8]. 여기서 n 은 두 다각형의 꼭짓점 개수의 합이고, t 는 두 경계선상의 교차점의 개수이다. t 는 $O(n^2)$ 이므로, 최악의 경우 $O(n^6)$ 크기의 RMESH가 요구된다. 가시성 다각형에서는 $t = O(n)$ 이므로[7], 이 알고리즘을 이용하는 경우 $O(n^3)$ 크기의 RMESH가 요구된다. 본 논문에서는 가시성 다각형의 특성을 이용하여 $O(n^2)$ 크기의 RMESH에서 $O(\log^2 n)$ 시간에 두 가시성 다각형의 교집합뿐만 아니라 합집합을 구하는 알고리즘을 제시한다. 이 알고리즘은 [8]의 알고리즘에 비해 수행 시간은 더 길리지만, 프로세서의 개수를 훨씬 적게 사용하며, 병렬 알고리즘의 비용을 나타내는 프로세서-시간 곱 지표[9]를 $O(n^3)$ 에서 $O(n^2 \log^2 n)$ 으로 개선한다.

II. 용어 정의 및 기본 성질

다각형 P 를 평면 상에 놓인 단순 다각형(simple polygon)이라고 하자. P 는 꼭짓점(vertex)들이 반시계 방향으로 주어진 순서 리스트 $(v_0, v_1, \dots, v_{n-1})$ 로 표현된

다. 두 꼭짓점 v_i 와 v_{i+1} 을 연결하는 선분을 변(edge)이라고 부르고 (v_i, v_{i+1}) 로 표시한다. 편의상 각 변을 v_i 에서 v_{i+1} 방향의 유향 선분(directed line segment)으로 간주한다. 이 때, v_i 를 이 유향 선분의 꼬리, v_{i+1} 을 머리라고 부른다. P 의 내부와 외부를 구분하는 점들의 집합을 P 의 경계선이라고 부른다. P 의 경계선을 반시계방향으로 따라갈 때 다각형의 내부는 항상 왼쪽에 놓인다.

Kim[7]은 두 가시성 다각형에 대한 다음의 성질을 증명하였다.

성질 1. 다각형 P 의 두 점 s 와 t 가 점 p 로부터 가시적이면, P 의 내부에서 정의되는 s 와 t 사이의 최단 경로 상의 모든 점도 p 로부터 가시적이다.

성질 2. 두 가시성 다각형의 교차 영역은 항상 하나의 연결된 영역이다.

성질 3. 두 가시성 다각형 P 와 Q 에 대해, P 의 어떤 한 변과 Q 와의 교차 영역은 공집합이 아니면 하나의 선분이다.

성질 2와 3에 의해 두 가시성 다각형의 교집합에 해당하는 다각형의 꼭짓점의 개수는 $O(n)$ 임을 알 수 있다. 또한 두 가시성 다각형의 교집합이 공집합이 아니라면 합집합에 해당하는 영역도 한 개의 다각형으로 나타나고, 그 크기도 역시 $O(n)$ 임을 알 수 있다.

$m \times n$ RMESH R 에서 i 번째 행과 j 번째 열에 위치한 프로세서를 $R(i, j)$ 로 나타낸다, 여기서 $0 \leq i < m$ 이고 $0 \leq j < n$ 이다. R 의 i 번째 행과 j 번째 열은 각각 $R(i, *)$ 와 $R(*, j)$ 로 표시한다. 다음은 RMESH에서 수행되는 기본 연산과 수행 시간을 보여준다[1, 2, 8].

연산 1. RMESH의 연결된 버스 조각에서 한 프로세서가 방송한 $O(1)$ 크기의 데이터를 버스 조각에 연결된 모든 프로세서가 읽는 작업은 상수 시간에 수행된다.

연산 2. RMESH에서 임의의 두 프로세서 사이에 $O(1)$ 크기의 데이터를 상수 시간에 전송할 수 있다.

연산 3. $n \times n$ RMESH R 의 각 $R(0, i)$ 에 저장된 $O(1)$ 크기의 데이터를 상수 시간에 $R(i, 0)$ 에 전송할 수 있다, $i = 0, 1, \dots, n-1$.

연산 4. n 개의 원소가 $n \times n$ RMESH의 한 열(또는 행)에 배치되어 있을 때, 이 원소들을 임의의 순열

에 따라 상수 시간에 특정 행(또는 열)에 재배치할 수 있다.

연산 5. n 개의 숫자가 $n \times n$ RMESH의 한 열(또는 행)에 배치되어 있을 때, 상수 시간에 이 숫자들을 정렬하여 특정 행(또는 열)에 배치할 수 있다.

연산 6. 1의 개수가 총 k 개인 n 개의 이진값이 $k \times n$ RMESH의 한 행에 배치되어 있을 때, 이 이진값들의 전위합(prefix sum)을 구해 특정 행에 배치하는 작업을 상수 시간에 수행할 수 있다.

연산 7. $1 \times n$ RMESH R 에서 특정한 k 개의 프로세서가 가진 데이터를 모두 $R(0, 0)$ 으로 전송하는 작업은 $O(k)$ 시간에 수행된다.

III. 평면 분할 그래프

본 절에서는 두 가시성 다각형의 교집합과 합집합을 구하기 위한 전처리 작업으로 구축하게 될 평면 분할 그래프(planar subdivision graph)에 대해서 설명한다. 입력으로 주어지는 두 가시성 다각형 P 와 Q 를 단순 다각형이라고 가정한다. 두 다각형의 경계선은 평면을 분할한다. 다각형의 꼭짓점과 경계선 상의 교차점들을 그래프의 정점으로 두고, 두 꼭짓점 사이의 변을 그래프의 에지로 둔다면, 분할된 평면은 평면 그래프로 표현된다. Kim[8]은 두 다각형의 교차 영역을 구하기 위한 다각형 교차 그래프를 제안했지만, 본 논문에서는 이를 확장하여 교집합 영역뿐만 아니라 합집합 영역도 쉽게 구할 수 있도록 평면 분할 그래프를 도입한다. 다각형의 변을 유향 선분으로 간주하기 때문에 평면 분할 그래프의 에지도 유향 에지로 간주한다. 따라서 평면 분할 그래프는 유향 그래프이다(그림 3 참조).

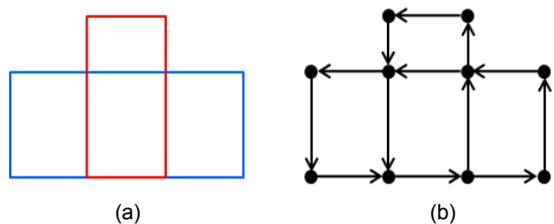


Fig. 3 An example of the planar subdivision graph of two polygons (a) Two polygons (b) A planar subdivision graph

평면 분할 그래프의 각 정점의 차수는 2, 3, 4 중의 하나이다. 그래프에서 정점 v 가 꼬리인 유향 에지가 두 개 존재하고 이를 각각 (v,u) 와 (v,w) 라고 하자. 이 때, 각도 $\angle(u,v,w)$ 가 180도 보다 작으면, (v,u) 는 (v,w) 의 오른쪽에 있는 에지라고 말한다. 두 가시성 다각형의 교차 영역은 하나의 연결된 영역이므로(성질 2), 평면 분할 그래프의 한 정점에서 시작하여 유향 에지를 따라 가면서, 진출 차수(out-degree)가 2인 정점을 만날 때 항상 오른쪽 에지를 따라가면 두 다각형의 합집합 영역을 구할 수 있다. 진입 차수(in-degree)가 2인 정점에서 보더라도 합집합 영역의 에지는 항상 오른쪽에 있는 에지로 구성된다. 또한 두 다각형의 교차점에 해당하는 한 정점에서 시작하여 항상 왼쪽 에지를 따라가면 두 다각형의 교집합 영역이 구해진다.

평면 분할 그래프는 RMESH의 첫 번째 열에 에지들을 배치하는 것으로 표현된다. 성질 3에 의해 P 와 Q 의 변 하나는 최대 3개의 구간으로 분할되고, 각 구간은 평면 분할 그래프의 에지가 된다는 것을 알 수 있다. 따라서 평면 분할 그래프의 에지의 개수는 기껏해야 $3n$ 이다, 여기서 n 은 두 다각형의 꼭짓점의 총 개수이다.

평면 분할 그래프를 이용하여 두 가시성 다각형의 교집합을 구하는 알고리즘이 알고리즘 1에 나와 있다. 알고리즘이 시작하기 전에 P 와 Q 의 변(유향 선분)들이 $3n \times 3n$ RMESH R 의 첫 번째 열에 배치된다. 알고리즘이 종료되면 교집합에 해당하는 다각형의 변들이 R 의 첫 번째 열에 배치된다. 알고리즘은 P 와 Q 의 경계선이 서로 교차하는 경우만 고려한다. 경계선이 교차하지 않으면, 두 다각형의 교집합은 공집합이고, 합집합은 $P \cup Q$ 이다. 경계선 교차 여부는 상수 시간에 간단하게 판별된다[8].

알고리즘 1: 교집합을 구하는 알고리즘

1. 평면 분할 그래프 G 를 구축한다. 그 결과, 그래프 G 의 에지들이 $R(*,0)$ 에 배치된다.
2. G 의 에지들을 각 에지의 꼬리 좌표를 기준으로 정렬한다. 그 결과, 꼬리가 같은 에지들은 $R(*,0)$ 의 인접한 열에 위치한다.
3. $R(*,0)$ 의 각 프로세서는 인접한 열에 있는 프로세서로부터 에지 정보를 전송받아 자신이 가진 에지와 꼬리가 같으면서 더 왼쪽에 위치한 에지가 존재하면, 자신이 가진 에지를 삭제한다.

4. G 의 에지들을 각 에지의 머리 좌표를 기준으로 정렬한다. 그 다음, $R(*,0)$ 의 각 프로세서는 인접한 열에 있는 프로세서로부터 에지 정보를 전송받아 자신이 가진 에지와 머리가 같으면서 더 왼쪽에 위치한 에지가 존재하면, 자신이 가진 에지를 삭제한다.
5. 남아 있는 에지들에서 사이클을 찾아 RMESH의 첫 번째 열에 배치한다.

알고리즘 1의 단계 1이 상수 시간에 수행된다는 것은 4절에서 증명한다. 단계 2의 정렬은 2열의 연산 5에 의해 상수 시간에 수행됨을 알 수 있다. 단계 3을 위해 인접한 홀수-짝수 번째와 짝수-홀수 번째 프로세서 쌍 사이에 한 번씩 독립된 버스 조각을 구성하면 에지 정보를 상수 시간에 전송할 수 있다. 단계 4는 단계 2, 3과 같은 방식으로 수행된다. 단계 4가 수행된 후에는 P 와 Q 의 교집합에 해당되는 다각형의 변들이 평면 분할 그래프의 사이클로 표현되고, 이 사이클은 그래프에서 연결 요소로 독립되어 있다. 따라서 논문 [8]의 방식으로 에지들을 RMESH에 임베딩하면, 사이클에 해당하는 부분은 독립된 버스 조각으로 구성되어 이 사이클에 속한 에지들을 검색할 수 있다. 그 다음, 이 에지들로부터 사이클의 정점 순서를 찾으면 되므로, 단계 5는 $O(\log^2 n)$ 시간에 수행된다[10].

두 가시성 다각형의 합집합을 구하는 알고리즘은 알고리즘 1의 단계 3과 4에서 오른쪽 에지 대신에 왼쪽 에지를 삭제하는 것으로 대체하면 된다.

IV. 알고리즘

본 절에서는 두 가시성 다각형 P 와 Q 에 대한 평면 분할 그래프를 구축하는 알고리즘을 소개한다. 알고리즘의 시작 전에 P 와 Q 의 변들이 $3n \times 3n$ RMESH R 의 첫 번째 열에 배치되어 있다. 즉, 각 프로세서 $R(i,0)$, $i = 0, 1, \dots, r-1$,에는 P 의 변들이 하나씩 저장되고, $R(r+j,0)$, $j = 0, 1, \dots, s-1$,에는 Q 의 변들이 하나씩 저장된다. 여기서 r 과 s 는 각각 P 와 Q 의 변의 개수이고 $n = r + s$ 이다. $R(0,0)$ 는 n 에 대한 정보를 가지고 있고, 필요시 모든 프로세서에게 이 정보를 전달한다. 알고리즘이 종료되면 평면 분할 그래프의 에지들이 RMESH의 첫 번째 열에 배치된다. 알고리즘의 개요는

다음과 같다. 알고리즘에서 다각형의 변과 그래프의 에지를 구분하지 않고 그냥 에지라고 부른다.

알고리즘 2: 평면 분할 그래프를 구축하는 알고리즘

1. $R(*,0)$ 에 저장된 에지들 중에서 동일한 에지를 찾아 제거한다.
2. $R(*,i)$ 에 저장된 에지들 중에서 서로 겹쳐있는, 즉, 일직선 상에 위치하면서 서로 교차하는 에지 쌍들을 찾아 기존 에지의 정보 갱신과 새로 추가되는 에지 정보를 만든다.
3. $R(*,i)$ 에 저장된 에지들 중에서 일직선 상에 위치하지 않으면서 기존 정점이 아닌 곳에서 교차하는 에지 쌍들을 찾아 기존 에지의 정보 갱신과 새로 추가되는 에지 정보를 만든다.

알고리즘 2의 수행 과정을 예를 들어 설명하기 위해 그림 4의 두 가시성 다각형이 주어진다고 하자. 그림 4의 다각형에 대해 RMESH의 첫 번째 열에 다음과 같은 에지 집합 E 가 배정된다.

$$E = \{(0,1), (1,2), (2,3), (3,4), (4,0), (5,6), (6,7), (7,8), (8,9), (9,5)\}$$

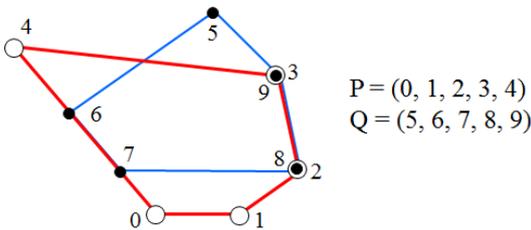


Fig. 4 An example of two polygons

알고리즘 2의 단계 1을 거치고 나면 $R(*,0)$ 에 저장된 에지 집합 E 에서 에지 (2,3)과 중복인 에지 (8,9)가 삭제된다. 단계 2에서는 서로 겹쳐져 있는 에지 쌍 (4,0)과 (6,7)을 찾아 에지 (4,0)을 (4,6)으로 바꾸고 새 에지 (7,0)을 추가한다. 단계 3에서는 서로 교차하는 에지 쌍 (3,4)와 (5,6)을 찾아 교차점에 해당하는 새로운 정점 10을 만들고, 기존 두 에지 (3,4)와 (5,6)을 네 개의 새 에지 (3,10), (10,4), (5,10), (10,6)으로 대체한다.

이제 알고리즘 2의 각 단계별 상세 설계와 수행 시간에 대해서 설명한다. 앞으로 기술하는 알고리즘에서 필요시 $R(i,0)$ 에 저장된 에지 e_i 를 (s_i, t_i) 로 표시한다. 여기서 s_i, t_i 는 다각형의 꼭짓점 또는 교차점의 좌표이다.

단계 1의 상세 설계:

- ① $R(*,0)$ 에 저장된 에지들을 각 에지의 꼬리 좌표를 기준으로 정렬한다.
- ② $R(i,0)$ 는 $R(i-1,0)$ 로부터 에지 e_{i-1} 을 전송받아 e_i 와 동일한 지 검사한다. 만일 두 에지가 동일하면 e_i 를 삭제한다, $i = 1, 2, \dots, n-1$.
- ③ 남아있는 에지의 수를 n 에 배정하고, $R(0,0)$ 부터 $R(n-1,0)$ 까지 구간에 에지들을 배치한다,

2절의 기본 연산들을 통해 소단계 1의 정렬 과정과 소단계 2의 인접한 프로세서 사이의 데이터 전송은 상수 시간에 수행됨을 알 수 있다. 소단계 3은 남아 있는 에지들을 정렬하여 재배치하고, 맨 마지막 에지를 가진 프로세서가 $R(0,0)$ 에 자신의 행 번호를 전달하면 된다. 따라서 단계 1은 상수 시간에 수행된다.

단계 2의 상세 설계를 기술하기 전에 다음을 고려하자. 가시성 다각형의 한 변 e 의 내부에 다른 가시성 다각형의 꼭짓점이 놓여 있을 때, 이 꼭짓점에 인접한 두 변 중 하나는 반드시 e 와 겹쳐야 한다. 왜냐하면 이 경우 e 은 항상 가시 대상인 다각형의 변이고 주어진 두 가시성 다각형은 단순 다각형이기 때문이다. 가시성 다각형의 두 변이 일직선 위에 놓여 있는 상황은 그림 5에서 볼 수 있다.

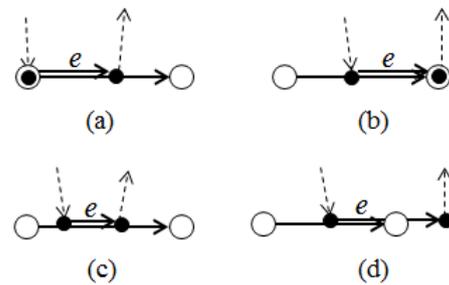


Fig. 5 Cases where two edges are overlapped

단계 2의 상세 설계:

- ① $\{R(i, j)$ 에 에지 e_j 를 배치한다, $i, j = 0, 1, \dots, n-1$.
먼저, $R(i, 0)$ 는 에지 e_i 를 $R(i, i)$ 로 전송한다. 그 다음, $R(i, i)$ 는 받은 에지를 $R(*, i)$ 로 방송한다. 각 $R(i, j)$ 는 방송된 에지를 읽어 저장한다.
- ② 각 $R(i, 0)$ 는 에지 e_i 를 $R(i, *)$ 에 방송하고, 각 프로세서는 방송된 에지를 읽어 저장한다.
- ③ $e_j \subset e_i$ 이고 $s_i = s_j$ 이며 t_j 가 e_i 의 내부에 있는 경우(그림 5의 (a)), $R(i, j)$ 는 $R(i, 0)$ 에게 t_j 를 전송하고, $R(i, 0)$ 는 자신이 갖고 있는 에지 e_i 에서 s_i 를 t_j 로 대체한다.
- ④ e_i 와 e_j 가 일직선 상에 있고 s_j 가 e_i 의 내부에 있는 경우(그림 5의 (b), (c), (d)), $R(i, j)$ 는 $R(i, 0)$ 에게 s_j 를 전송하고, $R(i, 0)$ 는 자신이 갖고 있는 에지 e_i 에서 t_i 를 s_j 로 대체한다.
- ⑤ $e_j \subset e_i$ 이고 s_j 와 t_j 모두 e_i 의 내부에 있는 경우(그림 5의 (c)), $R(i, j)$ 는 새로운 에지 (t_j, t_i) 를 만든다.
- ⑥ e_i 와 e_j 가 일직선 상에 있고 $s_i \in e_j$ 이고 $t_j \in e_i$ 이면(그림 5의 (d)), $R(i, j)$ 는 $R(i, 0)$ 에게 t_j 를 전송하고, $R(i, 0)$ 는 자신이 갖고 있는 에지 e_i 에서 t_i 를 t_j 로 대체한다. 그 다음, $R(i, j)$ 는 새로운 에지 (t_j, t_i) 를 만든다.
- ⑦ {추가된 에지의 개수 h 를 구하고, 추가된 에지들을 $R(n, 0)$ 부터 $R(n+h-1, 0)$ 까지의 구간에 배치한다.} 새로운 에지를 만든 $R(i, j)$ 는 이 에지를 $R(i, 0)$ 에 전송한다. 그 다음, $R(i, 0)$ 는 변수 b 를 생성하여 새 에지를 받은 경우는 1, 아니면 0을 b 에 배정한다. 그 다음, RMESH R 은 $R(*, 0)$ 에 저장된 이진값 b 들에 대한 전위합을 구한다. 그 다음, 각 $R(i, 0)$ 는 추가된 에지를 $R(h_i+n, 0)$ 으로 전송한다, 여기서 h_i 는 $R(i, 0)$ 가 가진 전위합이다. 마지막으로, 에지의 총 수 n 을 갱신한다.

단계 2의 소단계 3~6에서 서로 교차하는 에지 쌍은 각 행에 기껏해야 하나만 존재한다. 따라서 모든 i 에 대해 $R(i, j)$ 에서 $R(i, 0)$ 으로 데이터를 한꺼번에 전송하는 것이 2절의 연산 3에 의해 상수 시간에 수행된다. 소단계 7에서 전위합을 구하는 것은 상수 시간에 수행된다(연산 6). 따라서 단계 2는 상수 시간에 수행된다.

단계 3의 상세 설계:

- ① 단계 2의 소단계 1과 2를 차례로 수행한 다음, 각 $R(i, j)$ 는 e_i 와 e_j 가 에지 내부의 한 점에서만 교차하는지 여부를 검사한다.
- ② $R(i, *)$ 에서 교차점을 가진 프로세서의 수를 구한다. (2절의 성질 3에 의해 $R(i, *)$ 에서 교차점을 가진 프로세서의 수는 둘 이하이다.)
- ③ $R(i, *)$ 에서 교차점을 가진 프로세서가 하나인 경우, 이 교차점 z 를 가진 $R(i, j)$ 는 $R(i, 0)$ 에게 z 를 전송한다. 그 다음, $R(i, 0)$ 는 자신이 갖고 있는 에지 e_i 에서 t_i 를 z 로 대체한다. 그 다음, 새로운 에지 (z, t_i) 를 만든다.
- ④ $R(i, *)$ 에서 교차점을 가진 프로세서가 두 개인 경우, $R(i, *)$ 는 두 교차점을 $R(i, 0)$ 로 전송한다. 그 다음, $R(i, 0)$ 는 두 교차점 중에서 s_i 에 가까운 점을 a , t_i 에 가까운 점을 b 로 둔다. 그 다음, $R(i, 0)$ 는 자신이 가지고 있는 에지 e_i 에서 t_i 를 a 로 대체한다. 그 다음, 새로운 에지 (a, b) 와 (b, t_i) 를 만든다.
- ⑤ {추가된 에지의 개수 h 를 구하고, 추가된 에지들을 $R(n, 0)$ 부터 $R(n+h-1, 0)$ 까지 구간에 배치한다.} $R(*, 0)$ 의 각 프로세서가 갖고 있는 추가된 에지 수는 0, 1, 또는 2이다. 먼저, 이 값들에 대한 전위합을 구한 다음, 각 $R(i, 0)$ 는 $R(h_i+n, 0)$ 로 추가된 에지를 전송한다, 여기서 h_i 는 $R(i, 0)$ 가 가진 전위합이다. 마지막으로 에지의 총 수 n 을 갱신한다.

소단계 4에서 두 프로세서가 가진 교차점 정보를 $R(i, 0)$ 에게 전송하는 것은 2절의 연산 7에 의해 상수 시간에 수행된다. 소단계 5에서 삼진값 $\{0, 1, 2\}$ 에 대한 전위합을 구할 때, $0 = 00, 1 = 10, 2 = 11$ 로 표현하여 각 자릿수(이진값)에 대한 전위합을 구하는 과정을 두 번 수행하면 상수 시간에 수행된다. 나머지 소단계들도 RMESH의 기본 연산만으로 수행되기 때문에, 단계 3은 상수 시간에 수행된다. 각 단계가 상수 시간에 수행되므로 알고리즘 2는 상수 시간에 수행된다.

V. 결 론

본 논문에서는 현실적인 병렬 계산 모델로 주목을 받

고 있는 재구성가능한 메쉬에서 두 가시성 다각형의 교집합과 합집합을 구하는 알고리즘을 제시했다. 이 알고리즘은 전처리 작업으로 두 가시성 다각형에 대한 평면 분할 그래프를 구축한다. 평면 분할 그래프의 구축은 $O(n^2)$ 크기의 RMESH에서 상수 시간에 수행된다. 평면 분할 그래프의 구축 후에 교집합과 합집합을 구하는 과정은 Kim[10]의 사이클의 정점 순서를 찾는 알고리즘을 사용하여 $O(\log^2 n)$ 시간에 수행된다. 따라서 제시된 알고리즘은 가시성 다각형에 관해서는 교집합을 구하는 문제를 기존 알고리즘에 비해 프로세서-시간 곱 지표를 $O(n^3)$ 에서 $O(n^2 \log^2 n)$ 으로 개선하였다. 앞으로의 연구 과제는 프로세서-시간 곱 지표를 더욱 개선하는 RMESH 알고리즘의 개발과 평면 분할 그래프를 활용하여 다각형을 대상으로 하는 응용 문제를 효율적으로 해결하는 알고리즘을 찾는 것이다.

ACKNOWLEDGMENTS

This work was supported by the research grant of the Busan University of Foreign Studies in 2015.

REFERENCES

[1] R. Miller, V. K. Prasanna Kumar, D. Reisis, and Q. Stout, "Meshes with Reconfigurable Buses," in *Proceeding of 5th MIT Conf. on Adv. Res. in VLSI*, pp. 163-178, March 1988.

[2] R. Wankar and R. Akerkar, "Reconfigurable architectures and algorithms: a research survey," *International Journal of Computer Science and Applications*, vol. 6, no. 1, pp. 108-123, Jan. 2009.

[3] T. Pan, "The transitive closure and related algorithms of digraph on the reconfigurable architecture," *Parallel Processing Letters*, vol. 21, no. 1, pp. 27-43, March 2011.

[4] O. Bouattane, B. Cherradi, M. Youssfi, M. O. Bensalah, "Paralle c-means algorithm for image segmentation on a reconfigurable mesh computer," *Parallel Computing*, vol. 37, pp. 230-243, April-May 2011.

[5] D. T. Lee, "Visibility of a simple polygon," *Computer Vision, Graphics, and Image Processing*, vol. 22, pp. 207-221, May 1983.

[6] H.-G. Kim and Y.-K. Cho, "Point Visibility of a Simple Polygon on Reconfigurable Mesh," in *Proceeding of 5th IEEE Symposium on Parallel and Distributed Processing*, pp. 748-751, Dec. 1993.

[7] S. H. Kim, "Optimal algorithms for the Set operations of two visibility polygons in a simple polygon," *Journal of KISS: Computer Systems and Theory*, vol. 31, no. 1-2, pp. 102-111, Feb. 2004.

[8] S. H. Kim, "Constant time RMESH algorithms for polygon intersection problems," *Journal of KISS: Computer Systems and Theory*, vol. 26, no. 11, pp. 1344-1352, Nov. 1999.

[9] S. G. Akl, *The Design and Analysis of Parallel Algorithms*, Prentice-Hall, Inc., Upper Saddle River, NJ, 1989.

[10] S. H. Kim, "An efficient RMESH algorithms for finding a cycle vertex sequence from the edge set of a directed cycle graph," *Journal of KIISE: Computer Systems and Theory*, vol. 41, no. 3, pp. 146-150, Jun 2014.



김수환(Soo-Hwan Kim)

1987년 서울대학교 계산통계학과 학사.
 1989년 한국과학기술원 전산학과 석사.
 1995년 한국과학기술원 전산학과 박사.
 1992년~현재 부산외국어대학교 임베디드소프트웨어학과 교수
 ※관심분야 : 순차 및 병렬 알고리즘, 계산기하학, 그래프 이론 등