좌표축에 평행한 직사각형들의 합집합을 구하는 상수시간 RMESH 알고리즘

김수환* · 최진오*

*부산외국어대학교

A Constant Time RMESH Algorithm for the Union of Iso-Oriented Rectangles

Soo-Hwan Kim* · Jinoh Choi*

*Busan University of Foreign Studies

E-mail : {shkim, jochoi}@bufs.ac.kr

요 약

평면 상에 주어진 n개의 직사각형에 대한 합집합과 교집합을 구하는 문제에 대해 많은 연구 결과가 나와 있다. Lipski와 Preparata(1981)는 좌표축에 평행한 직사각형의 합집합을 구하는 문제에 대해서 $O(n\log n)$ 시간과 $O(n\log n)$ 공간의 순차 알고리즘을 제시하였고[1], Alevizos(2013)는 이를 개선한 $O(n\log n)$ 시간과 $O(n)$ 공간의 개선된 순차 알고리즘을 제시하였다[2]. 본 논문에서는 좌표축에 평행한 직사각형들의 교집합이 공집합이 아닌 경우에 대한 직사각형들의 합집합을 구하는 문제를 고려한다. 이 경우, 직사각형들의 합집합은 하나의 연결된 영역, 즉 직교다각형이 된다. 본 논문에서는 재구성가능한 메쉬(RMESH) 모델에서 상수 시간에 이 문제를 해결하는 병렬 알고리즘을 제시한다.

ABSTRACT

There are a lot of research results on the problem of finding the union and intersection of n rectangles on a plane. Lipski와 Preparata(1981) presented a sequential algorithm with $O(n\log n)$ time and $O(n\log n)$ space for the problem of finding the union of rectangles whose sides are parallel to the coordinate axes[1]. Alevizos(2013) presented an improved $O(n\log n)$ time and $O(n)$ space algorithm for the same problem[2]. In this paper, we consider the problem of finding the union of iso-oriented rectangles such that the intersection of them is not an empty set. In this case, the union of the rectangles becomes a connected area, that is, an orthogonal polygon. In this paper, we propose a parallel algorithm that solves this problem in constant time in a reconfigurable mesh(in short, RMESH) model.

키워드

Reconfigurable mesh, Parallel algorithm, Iso-oriented Rectangles, Union

Ⅰ. 서 론

직사각형은 점이나 선분과 마찬가지로 기본적인 기하학 요소이기 때문에, 직사각형 집합에 대한 연구는 VLSI, 지리학, 컴퓨터 그래픽스 등의 다양한 분야에서 이루어지고 있다. 또한 이론적 관점, 즉 알고리즘과 자료구조의 연구에서도 직사각형에 대한 다양한 문제들을 다루고 있다. 예를 들면, 직사각형들의 분리여부 검사, 모든 교차쌍 출력, 합집합의 면적, 질의 직사각형과의 교집합 출력 등이 있다. 본 논문에서는 평면 상에 주어진 n개의 직사각형에 대한 합집합을 구하는 문제를 고려한다. Lipski와 Preparata(1981)는 좌표축에 평행한 직사각형의 합집합을 구하는 문제에 대해서 $O(n\log n)$ 시간과 $O(n\log n)$ 공간의 순차 알고리즘을 제시하였고[1], Alevizos(2013)는 이를 개선한 $O(n\log n)$ 시간과 $O(n)$ 공간의 개선된 순차 알고리즘을 제시하였다[2]. 본 논문에서는 교집합이 공집합이 아닌 경우에 대한 직사각형들의 합집합을 구하는 문제를 재구성가능한 메쉬에서 상수

시간에 해결하는 병렬 알고리즘을 제시한다.

II. 재구성가능한 메쉬

재구성가능한 메쉬(Reconfigurable Mesh; 간단히 RMESH)는 1988년 Miller 등[3]에 의해 처음 소개된 병렬처리 모델이다. 재구성가능한 메쉬의 기본 구조는 프로세서들을 재구성가능한 버스 시스템에 의해 메쉬 형태로 연결한 것이다. 각 프로세서는 동(E), 서(W), 남(S), 북(N)의 4개 포트를 가지며, 알고리즘의 실행 중에 버스 스위치에 의해 각 포트 사이를 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 방송된 자료를 상수 시간에 읽을 수 있다. $n \times n$ RMESH의 각 프로세서는 $O(\log n)$ 비트 크기의 기억공간을 상수 개 저장할 수 있고, 사칙연산을 비롯한 기본 연산을 상수 시간에 수행할 수 있다. 또한, 각 프로세서는 자신이 속한 행과 열을 인지할 수 있다.

다음은 RMESH에서 수행되는 기본 연산과 수행 시간을 보여준다.

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

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

연산 4. n개의 숫자가 $n \times n$ RMESH의 한 열(또는 행)에 배치되어 있을 때, 상수 시간에 이 숫자들 중 최소값(또는 최대값)을 구하여 $R(0,0)$에 배치할 수 있다.

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

연산 6. 1의 개수가 최대 n개인 n개 이진값들이 $n \times n$ RMESH의 한 열(또는 한 행)에 배치되어 있을 때, 이들에 대한 전위합(prefix sum)을 상수 시간에 구할 수 있다.

III. 기본 성질

직교다각형(orthogonal polygon)은 모든 에지가 수평 또는 수직 선분인 다각형을 말한다. 직교볼록다각형(orthogonal convex polygon)은 어떤 수평선(또는 수직선)과의 교집합이 공집합 또는 하나의 선분으로만 구성되는 직교다각형을 말한다.

좌표축에 평행한 직사각형들의 집합을 S_R이라고 하자. S_R의 합집합으로 구성되는 다각형의 에지들은 당연히 수평 또는 수직 에지로만 구성되므로, 이 다각형(들)은 직교다각형이다. 다음 정리는 S_R의 교집합이 공집합이 아닌 경우의 합집합은 하나의 직교볼록다각형임을 증명한다.

정리 1. 좌표축에 평행한 직사각형의 교집합이 공집합이 아닐 때, 이들 직사각형들의 합집합은 하나의 직교볼록다각형이다.

(증명) 직사각형들의 합집합이 분리된 영역으로 구성되어 있다고 하면, 각 영역에 속하는 직사각형들 사이는 교집합이 공집합이어야 한다. 따라서 합집합은 하나의 직교다각형이다.

이제 어떤 수평선(또는 수직선)과 이 직교다각형과의 교집합을 고려하자. 교집합이 두 개 이상의 선분으로 구성된다고 가정하고, 이 들 선분에 교차하는 직사각형들 중 서로 다른 선분에 교차하는 두 직사각형을 각각 r_1과 r_2라고 하자. 직사각형의 성질에 의해 r_1과 $r2$의 교집합은 공집합이다. 따라서 직사각형들의 합집합은 직교볼록다각형이다. □

그림 1은 직사각형들의 합집합인 직교볼록다각형의 예를 보여준다.

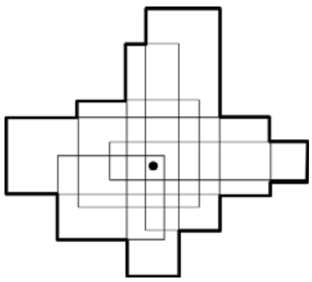

그림 1. 직교볼록다각형

직교볼록다각형의 정점은 직사각형의 꼭지점과 직사각형들 사이의 교차점으로 구성된다. 교차점이 정점이 되는 것은 직사각형마다 최대 8개이고 각 교차점은 두 개의 직사각형에 의해 생성되므로, 정점의 총 개수는 기껏해야 $4n + 4n = 8n$이다.

IV. 알고리즘

$n \times m$ RMESH에서 i번째 행과 j번째 열에 위치한 프로세서를 $R(i,j)$로 나타낸다, 여기서 $0 \le i < n$이고 $0 \le j < m$이다. $R(i,*)$는 R의 i번째 행을 의미하고, $R(*,j)$는 j번째 열을 의미한

다. 이제 $p \times n^2$ RMESH R의 첫 번째 열의 프로세서 $R(i,0)$에 직사각형 정보, 즉, 네 개의 꼭지점 좌표가 배치되어 있다고 하고($p=8n$), $R(i,0)$이 보유한 직사각형을 r_i라고 부르자($0 \le i < n$).

직사각형의 합집합인 직교볼록다각형을 구하는 알고리즘의 개요는 알고리즘 1에 나와 있다. 알고리즘의 종료시 직교볼록다각형의 정점 리스트 $(v_{0,}v_{1,}...,v_{m-1})$이 RMESH R의 첫 번째 열에 배치되고, $R(0,0)$에 정점의 개수인 m이 저장된다. R을 $p \times n$ 크기의 부메쉬로 구분할 때, 각 부메쉬를 R_k라고 표시한다($k=0,1,...,n-1$).

알고리즘 1:

1. $R(i,0)$는 자신이 보유한 직사각형 r_i를 $R(i,*)$의 각 프로세서로 전달한다($0 \le i < n$). {이 작업은 다음과 같이 수행된다. $R(i,0)$는 r_i를 $R(i,*)$에 방송하고 $R(i,*)$의 각 프로세서는 r_i를 저장한다.}
2. R_k는 $R_k(i,0)$에 배치된 각 r_i를 $R_k(0,i)$로 전송한다($0 \le k < n$). 그 다음 $R_k(k,0)$는 자신이 보유한 직사각형 r_k를 $R_k(0,*)$의 각 프로세서로 전달한다.
3. $R_k(0,j)$는 r_k의 각 수평 변 (a,b)에 대해 한 번씩 다음을 수행한다($0 \le j < n$).
   3.1 r_k의 왼쪽 꼭지점 a가 r_j에 포함되면 변수 $flag_a$에 0을 저장한다. 그렇지 않으면 1을 저장한다. $flag_a$가 1일 때, r_j의 왼쪽 변과의 교차점을 구하여 저장한다.
   3.2 r_k의 오른쪽 꼭지점 b가 r_j에 포함되면 변수 $flag_b$에 0을 저장한다. 그렇지 않으면 1을 저장한다. $flag_b$가 1일 때, r_j의 오른쪽 변과의 교차점을 구하여 저장한다.
   3.3 R_k는 0번째 행에 저장된 $flag_a$가 1인 것 중에서 x좌표가 가장 작은 교차점을 $R_k(0,0)$로 이동한다. {이 작업은 $flag_a$가 1이며 교차점을 갖고 있는 $R_k(0,*)$의 각 프로세서들에 대해 연산 4를 이용하여 수행한다.}
   3.4 R_k는 0번째 행에 저장된 $flag_b$가 1인 것 중에서 x좌표가 가장 큰 교차점을 $R_k(0,0)$로 이동한다.
4. R_k는 $R_k(0,0)$가 저장하고 있는 교차점들과 $flag$값이 1인 꼭지점들의 총 개수를 구하여 $R_k(0,0)$에 저장한다. {꼭지점과 교차점들은 직사각형들의 합집합인 직교복록다각형의 정점들을 완전하게 구성한다.}
5. 각 정점 v을 위한 다음의 레코드를 구성한다.
   pid : v가 배치될 프로세서 열 번호
   point : v의 좌표
   adj : v에 인접한 정점의 pid
   adj_point : v에 인접한 정점의 좌표
6. R은 $R_k(0,0)$이 가지고 있는 정점들에 대해 정점의 좌표 point를 설정한다. 그 다음, 연산 6의 전위합을 수행하고, 이 값을 사용하여 각 정점에 pid를 부여한다. 그 다음, 직교볼록다각형의 수평 에지에 대응되는 정점들 사이에 대한 adj와 adj_point를 설정한다. 이 과정에서 구한 정점의 총 개수를 $R(0,0)$에 전달한다.
7. R은 정점 레코드들을 $R(*,0)$으로 이동시킨다. 그 다음, 연산 5를 이용하여 정점들을 y좌표를 기준으로 정렬한다.
8. y좌표가 같은 인접한 정점들 사이에 adj와 adj_point를 설정한다. {이렇게 설정된 정점 레코드들은 직교볼록다각형의 경계선을 유향사이클그래프로 표현하게 된다.}
9. [4]의 알고리즘을 사용하여 직교볼록다각형의 정점 순서를 구한 후 R의 첫 번째 열에 배치한다.

(알고리즘 증명은 지면 관계상 생략함.)

Ⅴ. 결 론

본 논문에서는 n개의 좌표축에 평행한 직사각형들의 교집합이 공집합이 아닌 경우에 대한 직사각형들의 합집합을 $O(n^3)$ 크기의 RMESH에서 상수시간에 구하는 알고리즘을 제시하였다. 앞으로의 연구과제는 프로세서-시간 곱 비용을 개선한 알고리즘을 개발하는 것과 교집합이 공집합인 경우에 대해서도 직사각형들의 합집합을 효율적으로 구하는 RMESH 알고리즘을 개발하는 것이다.

참고문헌

[1] W. Lipski and F.P. Preparata, “Segments, rectangles, contours,” Journal of Algorithms, vol. 2, 63-76, 1981.

[2] P.D. Aleviz os, “An optimal algorithm for computing the non-trivial circuits of a union of iso-oriented rectangles,” Information Processing Letters, vol 113, 55-59, 2013.

[3] R. Miller, V.K. Prasanner Kumar, D. Reisis, and Q. Stout, “Meshes with Reconfigurable Buses,” Proc. 5th MIT Conf. on Adv. Res. in VLSI, 163-178, 1988.

[4] S.H. Kim, J. Choi, "A constant time parallel algorithm for finding a vertex sequence of the directed cycle graph from the individual neighborhood information," Proc. 2013 Conf. on Information and Communication Engineering, KIICE, vol.17, no.2, 326-328, 2013.