(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0080147
(43) 공개일자 2025년06월05일

(51) 국제특허분류(Int. Cl.)
G06F 18/2323 (2023.01) *G06F 18/2413* (2023.01)
(52) CPC특허분류
G06F 18/2323 (2023.01)
G06F 18/24147 (2023.01)
(21) 출원번호 10-2023-0167571
(22) 출원일자 2023년11월28일
심사청구일자 2023년11월28일

(71) 출원인
세종대학교산학협력단
서울특별시 광진구 능동로 209 (군자동, 세종대학교)
(72) 발명자
신지선
서울특별시 강남구 선릉로126길 22, 103동 1202호 (삼성동, 롯데캐슬프레미어아파트)
아지즈 카유뮵
서울특별시 광진구 아차산로30길 20-8, 502호 (자양동, 로데빌)
(74) 대리인
특허법인린

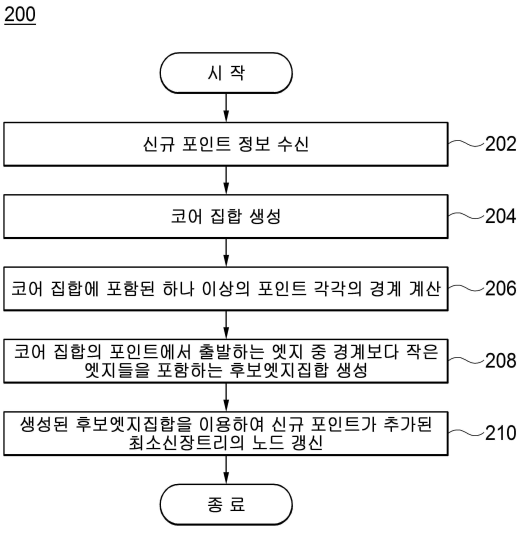
전체 청구항 수 : 총 14 항

(54) 발명의 명칭 **최소신장트리의 업데이트 방법 및 이를 수행하기 위한 컴퓨팅 장치**

(57) 요 약

최소신장트리의 업데이트 방법 및 이를 수행하기 위한 컴퓨팅 장치가 개시된다. 일 실시예에 따른 최소신장트리의 업데이트 방법은, 최소신장트리(Minimum Spanning Tree)에 추가될 신규 포인트의 정보를 수신하는 단계; 최소신장트리에 포함된 복수의 포인트 중 신규 포인트로 인해 코어 거리가 감소되는 하나 이상의 포인트를 포함하는 코어 집합을 생성하는 단계; 코어 집합에 포함된 하나 이상의 포인트 각각의 경계를 계산하는 단계; 하나 이상의 포인트 각각에서 출발하는 엣지 중 해당 포인트의 경계보다 작은 엣지들을 포함하는 후보엣지집합을 생성하는 단계; 및 후보엣지집합을 이용하여 신규 포인트가 추가된 최소신장트리를 갱신하는 단계를 포함한다.

대 표 도 - 도2



이 발명을 지원한 국가연구개발사업

과제고유번호	1711168648
과제번호	2020R1F1A1072275
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	개인기초연구(과기정통부)(R&D)
연구과제명	무인항공기를 위한 블록체인 기반 보안 강화 기술 개발
기 여 율	1/2
과제수행기관명	세종대학교
연구기간	2022.03.01 ~ 2024.02.29

이 발명을 지원한 국가연구개발사업

과제고유번호	1345366486
과제번호	2020R1A6A1A03038540
부처명	교육부
과제관리(전문)기관명	한국연구재단
연구사업명	이공학학술연구기반구축(R&D)
연구과제명	자율지능무인비행체연구소
기 여 율	1/2
과제수행기관명	세종대학교
연구기간	2023.03.01 ~ 2024.02.29

명 세 서

청구범위

청구항 1

하나 이상의 프로세서들, 및

상기 하나 이상의 프로세서들에 의해 실행되는 하나 이상의 프로그램들을 저장하는 메모리를 구비한 컴퓨팅 장치에서 수행되는 방법으로서,

최소신장트리(Minimum Spanning Tree)에 추가될 신규 포인트의 정보를 수신하는 단계;

상기 최소신장트리에 포함된 복수의 포인트 중 상기 신규 포인트로 인해 코어 거리가 감소되는 하나 이상의 포인트를 포함하는 코어 집합을 생성하는 단계;

상기 코어 집합에 포함된 상기 하나 이상의 포인트 각각의 경계를 계산하는 단계;

상기 하나 이상의 포인트 각각에서 출발하는 엣지 중 해당 포인트의 경계보다 작은 엣지들을 포함하는 후보엣지집합을 생성하는 단계; 및

상기 후보엣지집합을 이용하여 상기 신규 포인트가 추가된 상기 최소신장트리를 갱신하는 단계를 포함하는, 최소신장트리의 업데이트 방법.

청구항 2

청구항 1에 있어서,

상기 최소신장트리에 포함된 특정 포인트의 상기 코어 거리는, 상기 특정 포인트에 대한 k번째 최근접 이웃의 거리(k는 자연수)인, 최소신장트리의 업데이트 방법.

청구항 3

청구항 2에 있어서,

상기 경계는, 상기 최소신장트리에 포함된 각 엣지의 무게(weight), 상기 신규 포인트의 코어 거리 및 상기 코어 집합에 포함된 상기 하나 이상의 포인트와 상기 신규 포인트 간의 HDBSCAN 거리 중 하나 이상을 고려하여 계산되는, 최소신장트리의 업데이트 방법.

청구항 4

청구항 3에 있어서,

상기 코어 집합에 포함된 특정 포인트의 경계는,

상기 특정 포인트와 연결된 엣지 중 가장 무거운 엣지의 무게 및 상기 특정 포인트와 상기 신규 포인트 간의 HDBSCAN 거리 중 큰 값으로 설정되는, 최소신장트리의 업데이트 방법.

청구항 5

청구항 4에 있어서,

상기 HDBSCAN 거리는, 다음의 수학식

$d_{mreach}(u, v) = max(cd(u), cd(v), d(u,v))$

(여기서, $d_{mreach}(u, v)$는 점 u와 v 아이의 HDBSCAN 거리, cd(u)는 u의 코어 거리, d(u, v)는 u와 v 사이의 유클리디언 거리)

에 의하여 계산되는, 최소신장트리의 업데이트 방법.

청구항 6

청구항 4에 있어서,

상기 특정 포인트가 상기 신규 포인트인 경우,

상기 신규 포인트의 경계는 상기 최소신장트리에서 가장 무거운 엣지의 무게 및 상기 신규 포인트의 코어 거리 중 큰 값으로 설정되는, 최소신장트리의 업데이트 방법.

청구항 7

청구항 1에 있어서,

상기 최소신장트리를 업데이트하는 단계는,

(a) 상기 후보엣지집합에 포함된 엣지 중 가장 가벼운 엣지를 추출하는 단계;

(b) 상기 추출된 엣지에 대하여 링크-컷 트리 연산을 적용하는 단계; 및

(c) 상기 후보엣지집합에 포함된 나머지 엣지에 대하여 상기 (a) 단계 및 상기 (b) 단계를 반복 수행하는 단계를 포함하는, 최소신장트리의 업데이트 방법.

청구항 8

하나 이상의 프로세서;

메모리; 및

하나 이상의 프로그램을 포함하는 장치로서,

상기 하나 이상의 프로그램은 상기 메모리에 저장되고 상기 하나 이상의 프로세서에 의해 실행되도록 구성되며,

상기 프로그램은,

최소신장트리(Minimum Spanning Tree)에 추가될 신규 포인트의 정보를 수신하는 단계;

상기 최소신장트리에 포함된 복수의 포인트 중 상기 신규 포인트로 인해 코어 거리가 감소되는 하나 이상의 포인트를 포함하는 코어 집합을 생성하는 단계;

상기 코어 집합에 포함된 상기 하나 이상의 포인트 각각의 경계를 계산하는 단계;

상기 하나 이상의 포인트 각각에서 출발하는 엣지 중 해당 포인트의 경계보다 작은 엣지들을 포함하는 후보엣지집합을 생성하는 단계; 및

상기 후보엣지집합을 이용하여 상기 신규 포인트가 추가된 상기 최소신장트리를 갱신하는 단계를 실행시키기 위한 명령어들을 포함하는, 컴퓨팅 장치.

청구항 9

청구항 8에 있어서,

상기 최소신장트리에 포함된 특정 포인트의 상기 코어 거리는, 상기 특정 포인트에 대한 k번째 최근접 이웃의 거리(k는 자연수)인, 컴퓨팅 장치.

청구항 10

청구항 9에 있어서,

상기 경계는, 상기 최소신장트리에 포함된 각 엣지의 무게(weight), 상기 신규 포인트의 코어 거리 및 상기 코어 집합에 포함된 상기 하나 이상의 포인트와 상기 신규 포인트 간의 HDBSCAN 거리 중 하나 이상을 고려하여 계산되는, 컴퓨팅 장치.

청구항 11

청구항 10에 있어서,

상기 코어 집합에 포함된 특정 포인트의 경계는,

상기 특정 포인트와 연결된 엣지 중 가장 무거운 엣지의 무게 및 상기 특정 포인트와 상기 신규 포인트 간의 HDBSCAN 거리 중 큰 값으로 설정되는, 컴퓨팅 장치.

청구항 12

청구항 11에 있어서,

상기 HDBSCAN 거리는, 다음의 수학식

$d_{mreach}(u, v) = max(cd(u), cd(v), d(u,v))$

(여기서, $d_{mreach}(u, v)$는 점 u와 v 아이의 HDBSCAN 거리, cd(u)는 u의 코어 거리, d(u, v)는 u와 v 사이의 유클리디언 거리)

에 의하여 계산되는, 컴퓨팅 장치.

청구항 13

청구항 11에 있어서,

상기 특정 포인트가 상기 신규 포인트인 경우,

상기 신규 포인트의 경계는 상기 최소신장트리에서 가장 무거운 엣지의 무게 및 상기 신규 포인트의 코어 거리 중 큰 값으로 설정되는, 컴퓨팅 장치.

청구항 14

청구항 8에 있어서,

상기 최소신장트리를 업데이트하는 단계는,

(a) 상기 후보엣지집합에 포함된 엣지 중 가장 가벼운 엣지를 추출하는 단계;

(b) 상기 추출된 엣지에 대하여 링크-컷 트리 연산을 적용하는 단계; 및

(c) 상기 후보엣지집합에 포함된 나머지 엣지에 대하여 상기 (a) 단계 및 상기 (b) 단계를 반복 수행하는 단계를 포함하는, 컴퓨팅 장치.

발명의 설명

기 술 분 야

[0001] 개시되는 실시예들은 최소신장트리(MST; Minimum Spanning Tree)를 효율적으로 업데이트하기 위한 기술과 관련된다.

배 경 기 술

[0003] HDBSCAN(Hierarchical density-based spatial clustering applications with noise)은 밀도와 공간 근접성을 고려하여 공간 데이터의 의미 있는 패턴을 탐색하기 위한 알고리즘이다. HDBSCAN은 클러스터링 계층 구조를 구축하고 데이터의 플랫 파티션을 추출하기 위한 기본 구조로 최소신장트리(MST)를 이용한다.

[0004] 최소신장트리의 계산은 클러스터링 알고리즘에서 가장 큰 부분을 차지하며, 이에 따라 최소신장트리의 계산을 가속화하기 위한 많은 방법들이 제안되었다. 그러나 기존의 연구들은 기 생성된 최소신장트리에 새로운 점(vertex)이 추가되어 최소신장트리를 업데이트하는 상황에 대해서는 고려하지 않았다. 이에 따라 새 점을 삽입할 때 마다 전체 트리를 처음부터 다시 계산해야 하는 문제가 있었다.

발명의 내용

해결하려는 과제

[0006] 개시되는 실시예들은 최소신장트리에 새로운 점이 삽입될 경우 최소신장트리의 업데이트를 효율적으로 수행하기 위한 기술적인 수단을 제공하기 위한 것이다.

과제의 해결 수단

[0008] 예시적인 실시예에 따르면, 하나 이상의 프로세서들, 및 하나 이상의 프로세서들에 의해 실행되는 하나 이상의 프로그램들을 저장하는 메모리를 구비한 컴퓨팅 장치에서 수행되는 방법으로서, 최소신장트리(Minimum Spanning Tree)에 추가될 신규 포인트의 정보를 수신하는 단계; 최소신장트리에 포함된 복수의 포인트 중 신규 포인트로 인해 코어 거리가 감소되는 하나 이상의 포인트를 포함하는 코어 집합을 생성하는 단계; 코어 집합에 포함된 하나 이상의 포인트 각각의 경계를 계산하는 단계; 하나 이상의 포인트 각각에서 출발하는 엣지 중 해당 포인트의 경계보다 작은 엣지들을 포함하는 후보엣지집합을 생성하는 단계; 및 후보엣지집합을 이용하여 신규 포인트가 추가된 최소신장트리를 갱신하는 단계를 포함하는, 최소신장트리의 업데이트 방법이 제공된다.

[0009] 일 실시예에서, 최소신장트리에 포함된 특정 포인트의 코어 거리는, 특정 포인트에 대한 k번째 최근접 이웃의 거리(k는 자연수)일 수 있다.

[0010] 일 실시예에서, 경계는 최소신장트리에 포함된 각 엣지의 무게(weight), 신규 포인트의 코어 거리 및 코어 집합에 포함된 하나 이상의 포인트와 신규 포인트 간의 HDBSCAN 거리 중 하나 이상을 고려하여 계산될 수 있다.

[0011] 일 실시예에서, 코어 집합에 포함된 특정 포인트의 경계는 특정 포인트와 연결된 엣지 중 가장 무거운 엣지의 무게 및 특정 포인트와 신규 포인트 간의 HDBSCAN 거리 중 큰 값으로 설정될 수 있다.

[0012] 일 실시예에서, HDBSCAN 거리는 다음의 수학식

[0013] $d_{mreach}(u, v) = \max(cd(u), cd(v), d(u,v))$

[0014] (여기서, $d_{mreach}(u, v)$는 점 u와 v 아이의 HDBSCAN 거리, cd(u)는 u의 코어 거리, d(u, v)는 u와 v 사이의 유클리디언 거리)

[0015] 에 의하여 계산될 수 있다.

[0016] 일 실시예에서, 특정 포인트가 신규 포인트인 경우, 신규 포인트의 경계는 최소신장트리에서 가장 무거운 엣지

의 무게 및 신규 포인트의 코어 거리 중 큰 값으로 설정될 수 있다.

[0017] 일 실시예에서, 최소신장트리를 업데이트하는 단계는, (a) 후보엣지집합에 포함된 엣지 중 가장 가벼운 엣지를 추출하는 단계; (b) 추출된 엣지에 대하여 링크-컷 트리 연산을 적용하는 단계; 및 (c) 후보엣지집합에 포함된 나머지 엣지에 대하여 (a) 단계 및 (b) 단계를 반복 수행하는 단계를 포함할 수 있다.

[0018] 다른 예시적인 실시예에 따르면, 하나 이상의 프로세서; 메모리; 및 하나 이상의 프로그램을 포함하는 장치로서, 하나 이상의 프로그램은 메모리에 저장되고 하나 이상의 프로세서에 의해 실행되도록 구성되며, 프로그램은, 최소신장트리(Minimum Spanning Tree)에 추가될 신규 포인트의 정보를 수신하는 단계; 최소신장트리에 포함된 복수의 포인트 중 신규 포인트로 인해 코어 거리가 감소되는 하나 이상의 포인트를 포함하는 코어 집합을 생성하는 단계; 코어 집합에 포함된 하나 이상의 포인트 각각의 경계를 계산하는 단계; 하나 이상의 포인트 각각에서 출발하는 엣지 중 해당 포인트의 경계보다 작은 엣지들을 포함하는 후보엣지집합을 생성하는 단계; 및 후보엣지집합을 이용하여 신규 포인트가 추가된 최소신장트리를 갱신하는 단계를 실행시키기 위한 명령어들을 포함하는, 컴퓨팅 장치가 제공된다.

[0019] 일 실시예에서, 최소신장트리에 포함된 특정 포인트의 코어 거리는, 특정 포인트에 대한 k번째 최근접 이웃의 거리(k는 자연수)일 수 있다.

[0020] 일 실시예에서, 경계는 최소신장트리에 포함된 각 엣지의 무게(weight), 신규 포인트의 코어 거리 및 코어 집합에 포함된 하나 이상의 포인트와 신규 포인트 간의 HDBSCAN 거리 중 하나 이상을 고려하여 계산될 수 있다.

[0021] 일 실시예에서, 코어 집합에 포함된 특정 포인트의 경계는 특정 포인트와 연결된 엣지 중 가장 무거운 엣지의 무게 및 특정 포인트와 신규 포인트 간의 HDBSCAN 거리 중 큰 값으로 설정될 수 있다.

[0022] 일 실시예에서, HDBSCAN 거리는 다음의 수학식

[0023] $d_{mreach}(u, v) = \max(cd(u), cd(v), d(u,v))$

[0024] (여기서, $d_{mreach}(u, v)$는 점 u와 v 아이의 HDBSCAN 거리, cd(u)는 u의 코어 거리, d(u, v)는 u와 v 사이의 유클리디언 거리)

[0025] 에 의하여 계산될 수 있다.

[0026] 일 실시예에서, 특정 포인트가 신규 포인트인 경우, 신규 포인트의 경계는 최소신장트리에서 가장 무거운 엣지의 무게 및 신규 포인트의 코어 거리 중 큰 값으로 설정될 수 있다.

[0027] 일 실시예에서, 최소신장트리를 업데이트하는 단계는, (a) 후보엣지집합에 포함된 엣지 중 가장 가벼운 엣지를 추출하는 단계; (b) 추출된 엣지에 대하여 링크-컷 트리 연산을 적용하는 단계; 및 (c) 후보엣지집합에 포함된 나머지 엣지에 대하여 (a) 단계 및 (b) 단계를 반복 수행하는 단계를 포함할 수 있다.

발명의 효과

[0029] 개시되는 실시예들에 따를 경우 최소신장트리에 새로운 점이 삽입될 경우 최소신장트리의 업데이트를 위한 계산량을 감소시켜 최소신장트리의 업데이트를 효율적으로 수행할 수 있다.

도면의 간단한 설명

[0031] 도 1은 예시적인 실시예들에서 사용되기에 적합한 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 예시하여 설명하기 위한 블록도

도 2는 일 실시예에 따른 최소신장트리의 업데이트 방법(200)을 설명하기 위한 순서도

도 3은 일 실시예에 따른 최소신장트리의 노드 갱신 단계(210)를 좀 더 상세히 설명하기 위한 순서도

발명을 실시하기 위한 구체적인 내용

[0032] 이하, 도면을 참조하여 본 발명의 구체적인 실시형태를 설명하기로 한다. 이하의 상세한 설명은 본 명세서에서

기술된 방법, 장치 및/또는 시스템에 대한 포괄적인 이해를 돕기 위해 제공된다. 그러나 이는 예시에 불과하며 본 발명은 이에 제한되지 않는다.

[0033] 본 발명의 실시예들을 설명함에 있어서, 본 발명과 관련된 공지기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하기로 한다. 그리고, 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다. 상세한 설명에서 사용되는 용어는 단지 본 발명의 실시예들을 기술하기 위한 것이며, 결코 제한적이어서는 안 된다. 명확하게 달리 사용되지 않는 한, 단수 형태의 표현은 복수 형태의 의미를 포함한다. 본 설명에서, "포함" 또는 "구비"와 같은 표현은 어떤 특성들, 숫자들, 단계들, 동작들, 요소들, 이들의 일부 또는 조합을 가리키기 위한 것이며, 기술된 것 이외에 하나 또는 그 이상의 다른 특성, 숫자, 단계, 동작, 요소, 이들의 일부 또는 조합의 존재 또는 가능성을 배제하도록 해석되어서는 안 된다.

[0035] 도 1은 예시적인 실시예들에서 사용되기에 적합한 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 예시하여 설명하기 위한 블록도이다. 도시된 실시예에서, 각 컴포넌트들은 이하에 기술된 것 이외에 상이한 기능 및 능력을 가질 수 있고, 이하에 기술되지 않은 것 이외에도 추가적인 컴포넌트를 포함할 수 있다.

[0036] 도시된 컴퓨팅 환경(10)은 컴퓨팅 장치(12)를 포함한다. 컴퓨팅 장치(12)는 적어도 하나의 프로세서(14), 컴퓨터 판독 가능 저장 매체(16) 및 통신 버스(18)를 포함한다. 프로세서(14)는 컴퓨팅 장치(12)로 하여금 앞서 언급된 예시적인 실시예에 따라 동작하도록 할 수 있다. 예컨대, 프로세서(14)는 컴퓨터 판독 가능 저장 매체(16)에 저장된 하나 이상의 프로그램들을 실행할 수 있다. 상기 하나 이상의 프로그램들은 하나 이상의 컴퓨터 실행 가능 명령어를 포함할 수 있으며, 상기 컴퓨터 실행 가능 명령어는 프로세서(14)에 의해 실행되는 경우 컴퓨팅 장치(12)로 하여금 예시적인 실시예에 따른 동작들을 수행하도록 구성될 수 있다.

[0037] 컴퓨터 판독 가능 저장 매체(16)는 컴퓨터 실행 가능 명령어 내지 프로그램 코드, 프로그램 데이터 및/또는 다른 적합한 형태의 정보를 저장하도록 구성된다. 컴퓨터 판독 가능 저장 매체(16)에 저장된 프로그램(20)은 프로세서(14)에 의해 실행 가능한 명령어의 집합을 포함한다. 일 실시예에서, 컴퓨터 판독 가능 저장 매체(16)는 메모리(랜덤 액세스 메모리와 같은 휘발성 메모리, 비휘발성 메모리, 또는 이들의 적절한 조합), 하나 이상의 자기 디스크 저장 디바이스들, 광학 디스크 저장 디바이스들, 플래시 메모리 디바이스들, 그 밖에 컴퓨팅 장치(12)에 의해 액세스되고 원하는 정보를 저장할 수 있는 다른 형태의 저장 매체, 또는 이들의 적합한 조합일 수 있다.

[0038] 통신 버스(18)는 프로세서(14), 컴퓨터 판독 가능 저장 매체(16)를 포함하여 컴퓨팅 장치(12)의 다른 다양한 컴포넌트들을 상호 연결한다.

[0039] 컴퓨팅 장치(12)는 또한 하나 이상의 입출력 장치(24)를 위한 인터페이스를 제공하는 하나 이상의 입출력 인터페이스(22) 및 하나 이상의 네트워크 통신 인터페이스(26)를 포함할 수 있다. 입출력 인터페이스(22) 및 네트워크 통신 인터페이스(26)는 통신 버스(18)에 연결된다. 입출력 장치(24)는 입출력 인터페이스(22)를 통해 컴퓨팅 장치(12)의 다른 컴포넌트들에 연결될 수 있다. 예시적인 입출력 장치(24)는 포인팅 장치(마우스 또는 트랙패드 등), 키보드, 터치 입력 장치(터치패드 또는 터치스크린 등), 음성 또는 소리 입력 장치, 다양한 종류의 센서 장치 및/또는 촬영 장치와 같은 입력 장치, 및/또는 디스플레이 장치, 프린터, 스피커 및/또는 네트워크 카드와 같은 출력 장치를 포함할 수 있다. 예시적인 입출력 장치(24)는 컴퓨팅 장치(12)를 구성하는 일 컴포넌트로서 컴퓨팅 장치(12)의 내부에 포함될 수도 있고, 컴퓨팅 장치(12)와는 구별되는 별개의 장치로 컴퓨팅 장치(12)와 연결될 수도 있다.

[0041] 도 2는 일 실시예에 따른 최소신장트리의 업데이트 방법(200)을 설명하기 위한 순서도이다. 일 실시예에 따른 최소신장트리(MST)의 업데이트 방법(200)은 기 구성된 최소신장트리에 새로운 포인트(vertex)가 추가되었을 때, 최소신장트리 전체를 재구성하지 않고 최소의 계산 만으로 최소신장트리를 업데이트하기 위한 방법을 의미한다. 도시된 방법은 하나 이상의 프로세서들, 및 상기 하나 이상의 프로세서들에 의해 실행되는 하나 이상의 프로그램들을 저장하는 메모리를 구비한 컴퓨팅 장치, 예컨대 전술한 컴퓨팅 장치(12)에서 수행될 수 있다. 도시된 흐름도에서는 상기 방법 또는 과정을 복수 개의 단계로 나누어 기재하였으나, 적어도 일부의 단계들은 순서를 바꾸어 수행되거나, 다른 단계와 결합되어 함께 수행되거나, 생략되거나, 세부 단계들로 나뉘어 수행되거나, 또는

도시되지 않은 하나 이상의 단계가 부가되어 수행될 수 있다.

[0042] 일 실시예에 따른 최소신장트리의 업데이트 방법(200)을 설명하기에 앞서, 컴퓨팅 장치(12)는 동적 공간 인덱스 구조(dynamic spatial index structure)를 이용하여 최소신장트리를 구성하는 각 포인트들의 정보를 저장 및 관리하는 것으로 가정한다. 예를 들어, 컴퓨팅 장치(12)는 R트리, SS트리, 또는 SR트리 등의 구조를 이용하여 각 포인트들의 정보를 저장할 수 있다. SR-트리와 같은 동적 공간 인덱스 구조는 새로운 포인트가 추가될 경우 자체 밸런싱(self-balancing)을 이용하여 효율적 구조를 스스로 유지하는 특징이 있다.

[0043] 단계 202에서, 컴퓨팅 장치(12)는 최소신장트리(Minimum Spanning Tree)에 추가될 신규 포인트의 정보를 수신한다. 일 실시예에서, 컴퓨팅 장치(12)는 상기 신규 포인트를 전술한 SR트리와 같은 동적 인덱스 구조에 추가할 수 있다.

[0044] 단계 204에서, 컴퓨팅 장치(12)는 코어 집합을 생성한다. 여기에서 코어 집합은 최소신장트리에 포함된 복수의 포인트 중 신규 포인트로 인해 코어 거리가 감소되는 하나 이상의 포인트로 구성된 집합을 의미한다.

[0045] 일 실시예에서, 최소신장트리에 포함된 포인트 u의 코어 거리 cd(u)는, u에 대한 k번째 최근접 이웃(kth nearest neighbor, 또는 knn)의 거리(이때, k는 자연수)로 정의될 수 있다.

[0046] 예를 들어 신규 포인트 x가 추가되기 전, 최소신장트리 내의 포인트 u의 코어 거리를 cd(u)라고 가정하자. 만약 cd(u)가 x와 u 사이의 유클리디언 거리 d(u, v)보다 크면 u의 코어 거리는 신규 포인트 x로 인해 줄어들게 되므로, u는 코어 집합에 포함된다.

[0047] 일 실시예에서, 컴퓨팅 장치(12)는 코어 집합을 효율적으로 탐색하기 위해 동적 공간 인덱스 구조의 각 노드의 경계를 이용할 수 있다. 이를 좀 더 상세하게 설명하면 다음과 같다.

[0048] 최소신장트리의 각 포인트의 정보가 포함되는 동적 공간 인덱스 구조는 루트 노드 및 복수의 리프 노드를 포함하는 트리 구조를 가진다. 만약 상기 동적 공간 인덱스 구조가 R트리인 경우, 각 노드는 사각형 영역을, S트리인 경우는 원 모양의 영역을, SR 트리인 경우 사각형과 원이 중첩된 형태의 영역을 가진다. 이와 같은 구조에서 컴퓨팅 장치(12)는 다음과 같은 방식으로 코어 집합을 구성한다.

[0049] 1) 만약 동적 공간 인덱스 구조의 특정 노드 A의 경계(bound) 내에 신규 포인트 x가 위치하지 않을 경우, 컴퓨팅 장치(12)는 노드 A를 프룬(prune)한다. 즉, 컴퓨팅 장치(12)는 노드 A와 해당 노드의 하위 노드에 대한 탐색을 중단한다.

[0050] 2) 만약 특정 노드 A가 리프 노드라면, 컴퓨팅 장치(12)는 노드 A에 포함된 모든 점 y에 대하여 cd(y)와 d(x, y)를 비교하고, 만약 cd(y) > d(x, y)인 경우 y를 코어 집합에 추가한다.

[0051] 3) 특정 노드 A가 리프 노드가 아니라면, A의 자식 노드들에 대하여 1) 및 2) 단계를 재귀적으로 수행한다.

[0052] 다시 도 2로 돌아와서, 단계 206에서 컴퓨팅 장치(12)는 코어 집합에 포함된 하나 이상의 포인트 각각의 경계(bound)를 계산한다.

[0053] 일 일시예에서, 각 포인트의 경계는, 최소신장트리에 포함된 각 엣지의 무게(weight), 신규 포인트의 코어 거리 및 코어 집합에 포함된 하나 이상의 포인트와 신규 포인트 간의 HDBSCAN 거리 중 하나 이상을 고려하여 계산될 수 있다. 이를 좀 더 상세하게 설명하면 다음과 같다.

[0054] 코어 집합에 속하는 각 포인트 u의 경계는 다음과 같이 계산된다.

[0055] 먼저 u가 신규 포인트 x인 경우, 포인트 u의 경계(B)는 신규 포인트 x가 추가되기 전 최소신장트리에서 가장 무거운 엣지의 무게 및 신규 포인트의 코어 거리 중 큰 값으로 설정된다. 이를 수학식으로 나타내면 다음과 같다.

[0057] [수학식 1]

[0058] B = max (기존 최소신장트리에서 가장 무거운 엣지의 무게, cd(x))

[0060] 이와 달리 u가 신규 포인트가 아닌 경우, 포인트 u의 경계(B)는 포인트 u와 연결된 엣지 중 가장 무거운 엣지의 무게 및 포인트 u와 신규 포인트 x 간의 HDBSCAN 거리 중 큰 값으로 설정된다. 이를 수학식으로 나타내면 다음과 같다.

[0062] [수학식 2]

[0063] $B = \max$ (기존 최소신장트리에서 u와 연결된 엣지 중 가장 무거운 엣지의 무게, $d_{mreach}(u, x)$)

[0065] 이때, 점 u와 v 아이의 HDBSCAN 거리 $d_{mreach}(u, v)$는 다음과 같이 계산될 수 있다.

[0066] [수학식 3]

[0067] $d_{mreach}(u, v) = \max(cd(u), cd(v), d(u,v))$

[0069] 단계 208에서, 컴퓨팅 장치(12)는 코어 집합에 포함된 각 포인트에서 출발하는 엣지(edge) 중, 해당 포인트의 경계보다 작은 엣지들을 포함하는 후보엣지집합을 생성한다. 일 실시예에서, 컴퓨팅 장치(12)는 앞서 계산된 각 포인트의 경계를 이용하여 동적 공간 인덱스 구조에서 레인지 쿼리(rqnge query)를 통해 상기 후보엣지집합을 생성할 수 있다. 이를 좀 더 상세히 설명하면 다음과 같다.

[0070] 코어 집합에 포함된 포인트 y와 206 단계에서 계산된 경계 B_y가 주어졌을 때, 동적 공간 인덱스 구조에서 y로부터 B_y 거리 안에 있는 엣지(y, w)는 다음과 같이 구할 수 있다.

[0071] 1) 동적 공간 인덱스 구조 내에서 y로부터 B_y 거리를 가지는 가상의 영역을 설정한다.

[0072] 2) 만약 동적 공간 인덱스 노드의 특정 노드 A가 상기 가상의 영역과 오버랩되지 않을 경우, 컴퓨팅 장치(12)는 노드 A를 프룬(prune)한다. 즉, 컴퓨팅 장치(12)는 노드 A와 해당 노드의 하위 노드에 대한 탐색을 중단한다.

[0073] 3) 만약 특정 노드 A가 리프 노드이고 상기 가상의 영역과 오버랩되는 경우, 컴퓨팅 장치는 노드 A에 포함된 모든 점 w에 대하여 엣지 (y, w)를 후보엣지집합에 추가할지의 여부를 판단한다.

[0074] 4) 특정 노드 A가 리프 노드가 아니라면, A의 자식 노드들에 대하여 2) 및 3) 단계를 재귀적으로 수행한다.

[0075] 원형으로 노드의 범위가 정해진 SS트리로 예를 들면, 각 노드의 원의 중점과 y와의 거리에 원의 반지름을 뺀 길이가 B_y보다 크면 해당 노드는 프룬, 아니면 해당 노드와 오버랩이 존재하므로 해당 노드에 대해서는 탐색을 진행한다.

[0076] 단계 210에서, 컴퓨팅 장치(12)는 생성된 후보엣지집합을 이용하여 신규 포인트가 추가된 최소신장트리를 갱신한다. 일 실시예에서, 컴퓨팅 장치(12)는 link-cut 트리 연산을 이용하여 최소신장트리의 노드를 갱신할 수 있다. 이를 좀 더 상세히 설명하면 다음과 같다.

[0077] 도 3은 일 실시예에 따른 최소신장트리의 노드 갱신 단계(210)를 좀 더 상세히 설명하기 위한 순서도이다.

[0078] 단계 302에서, 컴퓨팅 장치(12)는 후보엣지집합을 구성하는 각 엣지를 크기가 작은 순서로 정렬한다.

[0079] 단계 304에서, 컴퓨팅 장치(12)는 후보엣지집합에 포함된 엣지 중 가장 가벼운 엣지 (p, q)를 추출한다.

[0080] 단계 306에서, 컴퓨팅 장치(12)는 p, q가 서로 연결되어 있는지의 여부를 판단한다.

[0081] 상기 306 단계의 판단 결과 p, q가 서로 연결되어 있는 경우, 단계 308에서 컴퓨팅 장치(12)는 p, q 사이의 경로 중 가장 무거운 엣지 (u, v)를 탐색한다.

[0082] 단계 310에서, 컴퓨팅 장치(12)는 u,v 사이의 HDBSCAN 거리 $d_{mreach}(u, v)$가 p,q 사이의 HDBSCAN 거리 $d_{mreach}(p, q)$보다 큰지 여부를 판단한다.

[0083] 상기 310 단계의 판단 결과 $d_{mreach}(u, v)$가 $d_{mreach}(p, q)$보다 큰 경우, 단계 312에서 컴퓨팅 장치(12)는 (u, v)를 제거(cut (u, v))하고, (p, q)를 연결(link (p, q))한다.

[0084] 한편, 만약 306 단계의 판단 결과 p, q가 연결되어 있지 않거나, 310 단계의 판단 결과 $d_{mreach}(u, v)$가 $d_{mreach}(p, q)$보다 작은 경우, 컴퓨팅 장치(12)는 엣지 (p, q)를 삭제한다.

[0085] 단계 316에서 컴퓨팅 장치(316)는 후보엣지집합에 남은 엣지가 존재하는지 여부를 판단하고, 만약 존재하는 경우 304 단계로 돌아가 나머지 엣지들에 대하여 304 단계 이하를 반복 수행한다. 만약 후보엣지집합에 남은 엣지가 없는 경우 과정은 종료된다.

[0087] 이상에서 대표적인 실시예를 통하여 본 발명에 대하여 상세하게 설명하였으나, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 전술한 실시예에 대하여 본 발명의 범주에서 벗어나지 않는 한도 내에서 다양한 변형이 가능함을 이해할 것이다. 그러므로 본 발명의 권리범위는 설명된 실시예에 국한되어 정해져서는 안 되며, 후술하는 특허청구범위뿐만 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

부호의 설명

[0089] 10: 컴퓨팅 환경

12: 컴퓨팅 장치

14: 프로세서

16: 컴퓨터 판독 가능 저장 매체

18: 통신 버스

20: 프로그램

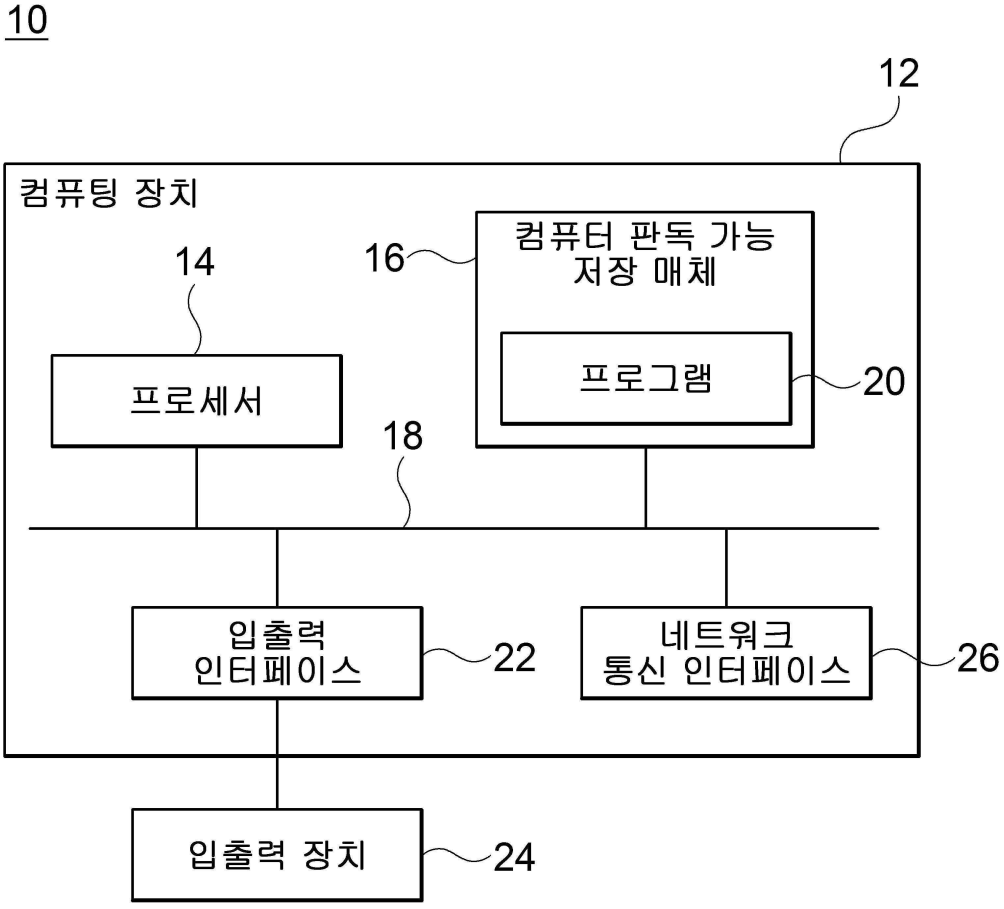
22: 입출력 인터페이스

24: 입출력 장치

26: 네트워크 통신 인터페이스

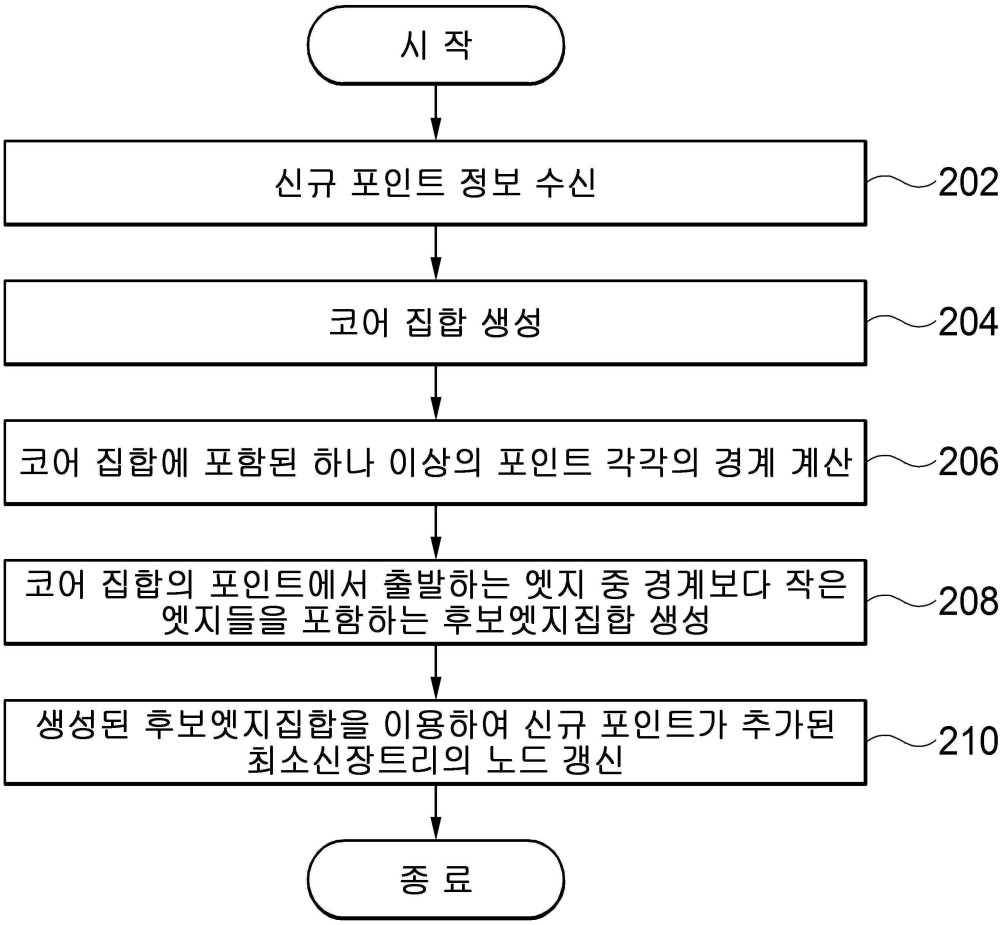
도면

도면1

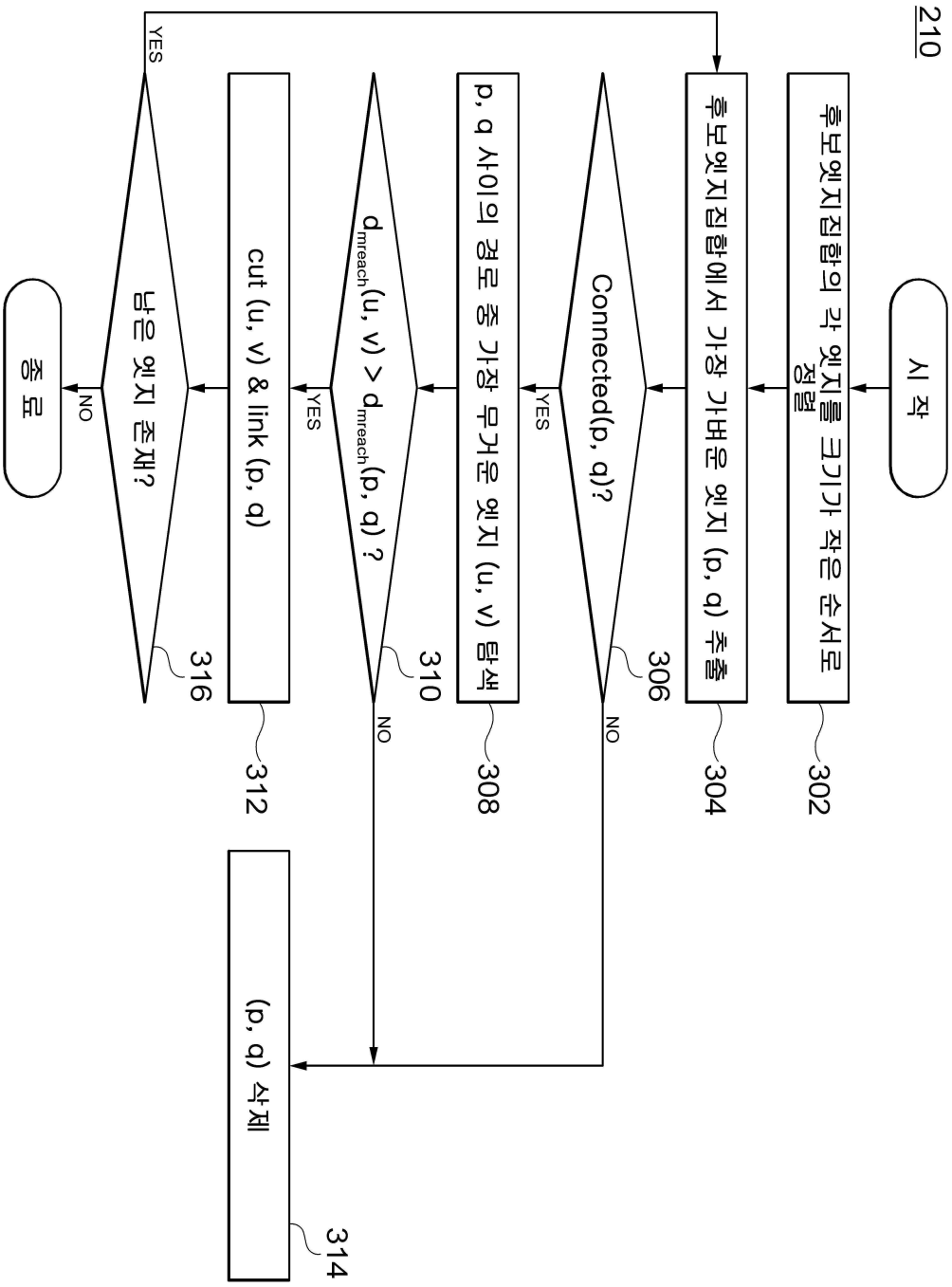

10
12
컴퓨팅 장치
14
프로세서
16
컴퓨터 판독 가능
저장 매체
프로그램
20
18
입출력
인터페이스
22
네트워크
통신 인터페이스
26
입출력 장치
24

도면2

200


시 작
신규 포인트 정보 수신
202
코어 집합 생성
204
코어 집합에 포함된 하나 이상의 포인트 각각의 경계 계산
206
코어 집합의 포인트에서 출발하는 엣지 중 경계보다 작은 엣지들을 포함하는 후보엣지집합 생성
208
생성된 후보엣지집합을 이용하여 신규 포인트가 추가된 최소신장트리의 노드 갱신
210
종 료

도면3


210
시 작
후보엣지집합의 각 엣지를 크기가 작은 순서로 정렬
302
후보엣지집합에서 가장 가벼운 엣지 (p, q) 추출
304
Connected(p, q)?
306
YES
NO
p, q 사이의 경로 중 가장 무거운 엣지 (u, v) 탐색
308
$d_{mreach}(u, v) > d_{mreach}(p, q)$?
310
YES
NO
cut (u, v) & link (p, q)
312
남은 엣지 존재?
316
YES
NO
(p, q) 삭제
314
종 료