

## 공간 노드들의 최단연결을 위한 3차원 유클리드 최소신장트리

김재각\*, 김인범\*\*

### Three Dimensional Euclidean Minimum Spanning Tree for Connecting Nodes of Space with the Shortest Length

Chae-kak Kim\*, Inbum Kim\*\*

#### 요약

일반적으로 유클리드 최소신장트리는 2차원 평면상에 존재하는 입력노드들이 최소 비용으로 연결된 것이다. 그러나 생성된 유클리드 최소신장트리는 3차원의 현실세계에 적용할 경우 그 연결비용은 최소비용이 아닐 수 있다. 본 논문에서는 3차원 공간상에 존재하는 입력노드를 최단 길이로 연결하는 3차원 유클리드 최소신장트리를 제안한다. 100%의 공간비율의 3차원 공간상에 존재하는 30,000개의 입력 노드에 대한 실험에서, 본 논문에서 제안된 방법에 생성된 트리는, Prim의 2차원 최소신장트리 알고리즘에 의해 생성된 유클리드 최소신장트리에 비해, 2차원 평면에서만 고려했을 때 251.2%의 연결 비용의 증가를 보이지만 이것은 3차원 실세계에서는 의미가 없다. 본 논문에서 제안된 방법에 의해 생성된 트리는 3차원 공간에서는 90.0%의 비용의 절감율을 보인다. 이는 제안된 방법이 3차원적 연결에 관한 많은 현실적인 문제에 잘 적용될 수 있음을 나타낸다.

▶ Keyword : 유클리드 최소신장트리, 공간비율, 3차원 공간, Prim의 최소신장트리 알고리즘, 연결 비용

#### Abstract

In general, Euclidean minimum spanning tree is a tree connecting input nodes with minimum connecting cost. But the tree may not be optimal when applied to real world problems of three dimension. In this paper, three dimension Euclidean minimum spanning tree is proposed, connecting all input nodes of 3-dimensional space with minimum cost. In experiments for 30,000 input nodes with 100% space ratio, the tree produced by the proposed method can reduce 90.0%

- 제1저자 : 김재각 • 교신저자 : 김인범
- 투고일 : 2011. 11. 28, 심사일 : 2011. 12. 18, 게재확정일 : 2011. 12. 29.
- \* 김포대학 사회복지학과(Dept. of Social Welfare, Kimpo College)
- \*\* 김포대학 인터넷정보과(Dept. of Internet Information, Kimpo College)
- ※ 본 논문은 2011학년도 김포대학의 연구비 지원에 의하여 연구되었음

connection cost tree, compared with the tree by two dimension Prim's minimum spanning tree. In two dimension plane, the proposed tree increases 251.2% connecting cost, which is pointless in 3-dimensional real world. Therefore, the proposed method can work well for many connecting problems in real world space of three dimensions.

▶ Keyword : Euclidean Minimum Spanning Tree, Space Ratio, 3-Dimensional Space, Prim's MST Algorithm, Connecting Cost

## I. 서 론

과거 비록 3차원적인 문제이지만 생활이나 활동 범위, 문제 도메인이 작은 공간에서의 발생하는 문제나 제공하는 서비스에 대한 2차원 평면적인 해결 시도는 비교적 우수한 결과를 보였다. 그 이유는 그러한 문제들에서는 대부분 3차원 공간에서의 높이, 즉 z축의 좌표 값이 평면적 좌표, 즉 x, y축의 좌표 값에 비해 상대적으로 매우 작았기 때문이다. 그러나 최근에는 교통, 통신의 발달로 인해 전 지구적인 또는 심해나 우주 공간 등에서 고려해야 하는 문제나 제공 서비스가 증가하고 있다. 이러한 문제들의 경우 공간에서 z축 좌표 값이 상대적으로 매우 크므로, 과거와 같은 2차원 평면적 문제 해결은 많은 오차를 발생한다. 더욱이 사람들의 서비스에 대한 기대 수준이 과거에 비해 크게 향상되었으므로, 현실성을 이용한 다양한 분야에서의 다양한 3D 응용이 제공되고 있다. 특히 엔터테인먼트에서의 3D TV나 영화, 3D 지리정보시스템 및 3D 내비게이션, 3D를 이용한 의학 이미지 정보시스템 등이 대표적인 응용이다. 따라서 3차원 현실성을 표현하는 기술 및 3차원 처리에 관한 연구 및 기술 개발이 점차 확대되고 있는 추세이다.

현재 통신 네트워크의 구축 및 라우팅 등에 많이 적용되어 활용 중인 최소신장트리(Minimum Spanning Tree)는 주로 2차원 평면에서 고려되어 왔다. 그러나 현실 세계는 평면이 아닌 3차원 공간이므로, 2차원에서 산출된 결과는 대규모 공간이나 높은 z축의 좌표 값을 갖는 환경의 3차원 공간에 적용했을 때 많은 오차를 보일 수 있다.

본 논문에서는 3차원 공간에서 좌표를 가지는 입력 노드들에 대하여 최소 비용으로 입력노드들을 모두 연결하는 효과적인 3차원 최소신장트리 생성 방법을 제안한다. 또한 Prim의 2차원 최소신장트리와 비교하여 그 오차를 분석한다. 제안된 방법은 3차원 공간에서의 네트워크 구축 및 라우팅, 비행 및 항 후 우주항로, 심해 항로의 개발, 가시거리통신 및 위성통신 등에 잘 활용될 수 있다.

본 논문은 2장에서 관련된 연구들의 기술, 3장에서 제안하는 3차원 최소신장트리 생성에 관한 기술, 4장에서 제안 방법의 실제 구현 및 실험분석을 기술하고, 5장의 결론으로 맺는다.

## II. 관련 연구

최소신장트리는 모든 입력 노드들을 일부 입력 선분들을 이용하여 연결할 때, 최소의 가중치의 합을 가지는 트리이다. 이러한 트리를 찾는 문제의 해결 알고리즘으로 Kruskal과 Prim의 알고리즘이 유명하다[1]. 일반적인 최소신장트리 문제를 확대한 유클리드 최소신장트리 문제는 입력 선분이 존재하지 않는 것이 큰 차이점이다. 유클리드 최소신장트리를 구축하기 위한 단순한 방법은, 주어진 모든 입력 노드  $n$ 개에 대해서, 각각 다른 노드  $(n-1)$ 개를 모두 연결하는  $1/2 \times n \times (n-1)$ 개의 연결 선분들을 생성하고, 이들 노드들과 새로이 생성된 연결 선분들을 기존의 최소신장트리 알고리즘에 도입하는 것이다.

주어진 입력 선분들을 직선 연결 선분을 이용해서 최소 비용으로 연결하는 방법에 대한 연구가 있었다[2]. 이 연구에서 입력 선분들 상에 위치하며, 이들을 일정한 길이로 분할하는 가상 노드인 포탈(Portal)을 이용하였다. 입력 선분을 모두 최소 비용으로 연결하는 트리의 생성 방법은 각 선분의 양 끝점에서 다른 선분들과의 최단 거리의 연결선들을 모두 구하고, 이 중 최소 길이의 연결선을 선택하여 최소신장트리를 구하는 것이다. 이를 위해 각 입력 선분 위에 포탈이라는 가상 노드를 도입하여 이용하였다. 제안된 방법의 경우는, 외부 인스턴 포탈 간격과 포탈 포기 비율에 따라 계산식의 상수항 조정이 가능하므로 적용하려는 응용의 목적에 따라 실행시간과 트리 길이 증가분의 조절이 가능하다고 주장하였다.

최소신장트리 문제는 네트워크 토폴로지 구성, 라우팅뿐만 아니라 3차원 공간에서의 다양한 적용에 대한 연구가 진행되었다. 최소신장트리를 이용하여 비디오 영상에서 문자 영역을 추출하는 연구가 그중의 하나이다[3]. 일반적인 문자 영역 추

출 방법은 전체 영상으로부터 컬러 영역 분할이나 프레임 차 방법을 이용하는데, 이 경우 추출하려는 문자의 사전 정보를 가지고 있어야 하며 실제 구현에 많은 어려움이 따른다. 이 연구에서는 문자의 지형학적 특징 점을 추출하고 이 점들을 이용한 최소신장트리를 구성한 후, 문자의 후보 영역을 추출한다. 최종 문자 영역은 이러한 후보 영역의 검증을 통하여 추출하게 하였다.

보안 전송(confidential transmission), 비디오 감시(video surveillance), 군사 및 의학 분야와 같은 많은 응용에서 데이터 은폐(data hiding)는 활용될 수 있다. 이러한 정보 은폐 처리 분야에 최소신장트리를 적용한 연구가 있었다[4]. 이 연구에서, 3차원 공간에서 정점(vertex) 위치의 변화 없는 3D 객체 데이터 은폐에 대한 새로운 방법을 제안하였다. 또한 이 연구에서는 메시지를 삽입하는데 사용하기 위해 3D 객체의 특정 영역을 찾고 동기화하는 방법을 제안하였다. 삽입은 4부분으로 구성된 선택된 영역의 선분 연결을 변경함으로써 수행된다. 이 연구에서는 3D 객체의 데이터 은폐를 위해 최소신장트리를 사용하였다. 이 연구에서 정점의 위치가 삽입 전이나 삽입 후가 동일하다는 측면에서 데이터의 손실이 없고, 파일 내의 데이터의 순서에 영향을 받지 않는다고 주장하였다. 또한 제안된 방법은 3D 객체들이 높은 정밀도를 가지고 디지털화될 때 의미 있다고 주장하였다.

카메라로부터 얻은 영상의 해석결과를 이용하여 특정 영역에 대한 감시를 목적으로 하는 시스템 개발을 위해, 최소신장트리를 적용한 연구가 있었다[5]. 이 연구에서는 그래프 이론 기반의 클러스터링을 이용한 감시 카메라 시야 내의 출입 영역 검출 방법을 제안하였다. 입력 데이터 포인트를 이용하여 최소신장트리를 구성 한 후, 이 트리의 선분 중에서 이웃하는 선분들에 비해 특이성을 갖는 것을 삭제하여 최소신장트리의 부분 트리를 얻는다. 최소신장트리의 각 부분 트리가 한 개의 클러스터가 되고, 모든 클러스터가 well-formed 될 때까지 분할을 반복하여 출입 영역을 효과적으로 검출하는 방법을 제안하였다.

일반적인 2차원 비디오 프레임을 3차원 입체 비디오 프레임으로 변환하는 방법에 대한 연구가 있었다[6]. 우선 연속적인 프레임 간 픽셀 변위를 구하고 이 변위 정보를 기반으로 스테레오 카메라로 촬영한 것과 같은 입체영상을 합성하는 방법을 채택하였다. 이 방법에서 픽셀 변위를 구하기 위하여 기존 수렴 반복법을 개선 보완하여 사용하였으며 픽셀 변위 정보로부터 입체영상을 합성하였다. 이 연구에서 제안된 알고리즘은 유형 분류 착오에 의한 문제점이 없는 입체 영상을 재현할 수 있고, 적절한 전용 프로세서를 활용한다면 실시간 변환

도 가능할 것이라고 주장하였다.

도시 환경의 모델링을 위해 3D 기법과 최소신장트리를 적용한 연구가 있었다[7]. 이 연구의 주요 목적은 도시 환경의 실제 3D 모델을 생성하기 위한 레이저 거리측정 데이터와 시각 이미지 데이터를 동시에 습득할 수 있는 시스템의 개발이다. 이러한 시스템 개발 과정에서의 주요 어려움은 모든 측정 데이터를 일반 좌표계로 자동 조정하는 것이다. 이를 해결하기 위해 이 연구에서 다중 3D 데이터 집합을 특성 단위(Feature Unit)를 이용하여 자동 등록해주는 방법을 제안하였다. 측정 데이터로부터 3D 특성을 추출한 후, 가상 특성(Virtual Feature)을 정의하고, 특성 단위를 구축한다. 대규모 데이터에 대한 등록 작업을 수월하게 수행하기 위해 가상 특성은 막힘 등으로 발생된 손실 정보를 보완하고, 특성 단위는 특성간의 관계를 기술한다. 다음 단계로 자동적으로 특성 단위를 이용해서 개별 검색자료를 사이의 부분적 등록을 수행한 후에 최소신장트리를 이용해서 토폴로지 그래프를 구축한다. 모든 관찰된 자료들은 같은 좌표계에 위치시킨다. 이 연구의 실험 결과는 제안된 알고리즘이 심각한 막힘에서도 도시 스캔을 건설하는데 상당히 효과적이라는 것이다.

항공사진을 이용한 효율적인 3차원 공간정보의 구축 방안 및 이에 대한 분석에 대한 연구가 있었다[8]. 이 연구에서는 항공기에 장착되어진 다각 촬영카메라로부터 수직 및 경사영상을 촬영하고 이를 통해 3차원 공간 정보를 효과적으로 구축하기 위해 Pictometry 시스템을 활용하였다. 이 연구에서는 또한 수집된 수직 영상과 경사 영상을 사용한 3차원 국토공간 정보 구축의 정확도 확보를 위한 절대위치결정의 기준점 배치 및 수량이 최종 정확도에 미치는 영향을 평가하고 가장 효율적인 기준점 운영 방안을 제시하였다.

3차원 세계를 모델링하려는 연구가 있었다[9]. 이 연구에서는 무인 차량의 주변 환경 인식을 위한 센서 시스템을 설계하였고, 이를 통해 3차원 세계 모델을 구축하고 확장하였다. 제안된 센서 시스템은 2대의 2차원 레이저 스캐너와 2대의 카메라를 통해 주변 환경 정보를 획득하고, DGPS와 IMU를 통해 차량의 위치를 추정한다. 전방 스테레오 카메라를 통해 비전 센서 기반 3차원 세계 모델을 생성하고, 이를 레이저 센서 기반 3차원 세계 모델과 결합하여, 보다 나은 3차원 세계 모델을 생성하고자 하였다. 또한 이 연구에서는 측정 범위가 넓은 센서를 활용하여 원거리 정보와 전방의 추가적인 레이저 스캐너를 통한 성능 개선이 가능하다고 주장하였다.

의학 분야에 3D와 최소신장트리를 적용한 연구 사례가 있었다[10]. 이 연구에서는 3D 이미지 입체 화소(Voxel)와 이와 연관된 반경 정보를 이용하여 생성된 이산 4D 그래프에서

의 평균 경로 선분 비용을 최소화하는 평균 비용 경로 (Minimum Average Cost Path, MACP) 모델을 제안하였다. Prim의 최소신장트리 방법은 MACP의 효과적인 최적화를 위해 이용되었다. 관상동맥(Coronary Arteries) 단면의 중심선과 반경은 최적화 작업 동안 동시에 추출된다. 이 방법은 다른 이미지 처리 과정이 필요 없는 장점이 있고, 실험을 통해 처리 비용 및 계산 량이 상당히 낮았음을 보였는데, 이것은 관상동맥의 실시간 분할 처리가 가능하다고 이 연구에서 주장하였다.

### III. 3차원 유클리드 최소신장트리

#### 1. 제안방법

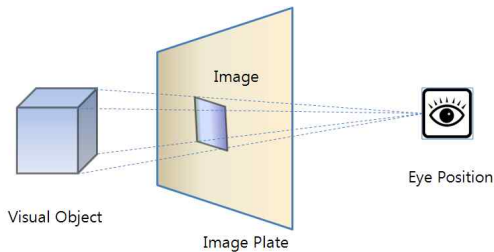


그림 1. 3차원 객체의 형태 인식 과정  
Fig. 1. Cognition Process of Object Shape in Space of Three Dimensions

그림 1은 일반적으로 사람의 눈을 통해 3차원 객체 즉 Visual Object를 인식하는 형태를 보인다. 공간상에 위치한 Visual Object는 사람의 Eye Position에 따라 Image Plate에 투영되는 2차원 객체 즉 Image로 보인다. 따라서 사람이 인식하는 Image의 길이나 위치는 공간상의 실제 값과는 오차가 발생한다. 그림 2는 표 1과 같은 좌표를 가지는 세 개의 노드 A, B, C의 3차원 공간에서의 위치를 나타낸다. 여기서  $x^+$ ,  $y^+$ ,  $z^+$ 는 각각 x 좌표축의 양의 방향, y 좌표축의 양의 방향, z 좌표축의 양의 방향을 나타낸다. 그림 3은 이 세 노드를 완전 연결한 공간상의 삼각형을 표현하고 있다. 그림 4는 공간상의 세 노드를  $z=0$ 인 평면상에 투영하였을 때 생성되는 세 노드 A', B', C'과 이들을 원소로 평면상에 생성되는 삼각형 'A'B'C'을 표현하고 있다. 평면상의 가상 노드인 A', B', C'의 좌표는 표 2에 나타나 있고, 이들의 각각의 거리로 생성되는 선분의 길이는 표 3에 나타나 있다. 그림 5는 공간상의 세 노드와 이들의 연결선으로 구성된 삼각형과, 이를 평면에 투영했을 때 생성되는 노드들과 삼각형의 모습을

나타낸다. 일반적인 2차원 최소신장트리는 평면상의 삼각형을 구성하는 세 선분 중에서 길이가 작은 두 선분, 즉 선분 A'B'과 선분 C'A'이 최소신장트리의 구성 원소가 된다. 이것을 바탕으로 공간상의 최소신장트리를 생성하면 그림 6과 같이 노드 A와 B, 그리고 노드 C와 A를 연결하는 선분이 생성된다. 그러나 이 두 연결은 그림 7과 같이 노드 B와 C, 노드 C와 A를 연결하는 선분의 길이 합이 더 크다. 표 4에는 세 노드 A, B, C의 연결 선분의 공간상의 길이를 나타내고 있고, 이 세 선분 중에서 길이의 합이 가장 작은 두 선분, 즉 선분 BC와 선분 CA가 공간상의 최소신장트리의 구성원소가 된다. 이는 평면 최소신장트리 방법에서 유추된 공간 최소신장트리와 다른 결과가 발생한다. 표 5는 평면 기반의 최소신장트리와 3차원 공간에서 생성된 실제 최소신장트리의 길이를 비교한 것이다. 이 표에서, 공간의 실제 최소신장트리는 평면에서 추정한 최소신장트리보다 5.57%의 길이가 감소함을 확인할 수 있다. 반면에 3차원 공간을 평면에 투영시켜 최소신장트리를 생성한다면 실제 길이는 5.90% 증가함을 알 수 있다. 그러나 3차원 공간에서, 2차원 평면 기반 최소신장트리는 착시에 의해 생성된 것으로 이 수치는 의미가 없다고 할 수 있다.

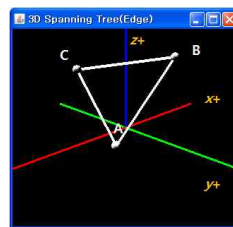


그림 2. 3차원 입력노드의 완전 연결 선분  
Fig. 3. Complete Connection Edges in Space of Three Dimensions

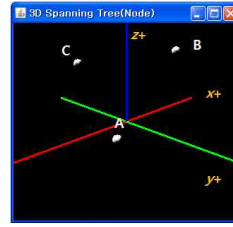


그림 3. 공간에서의 3개 입력노드  
Fig. 2. Three Input Nodes in Space of Three Dimensions

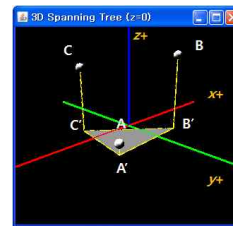


그림 4. 공간 입력 노드들의 평면으로의 투영  
Fig. 4. Projected Input Nodes of Space onto Plane

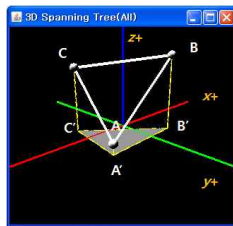


그림 5. 공간 입력 노드와 연결선들의 평면으로의 투영  
Fig. 5. Projected Input Nodes and Edges of Space onto Plane

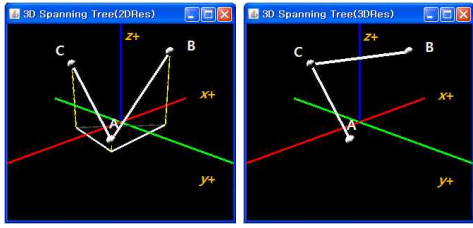


그림 6. 2D MST를 바탕으로 공간 연결 선분  
Fig. 6. Edge Elements of 2-dimensional Minimum Spanning Tree

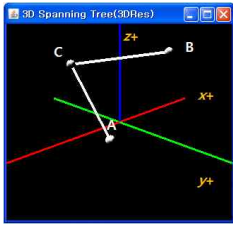


그림 7. 3D MST를 구성하는 선분  
Fig. 7. Edge Elements of 3-dimensional Minimum Spanning Tree

표 1. 세 입력 노드의 3차원 좌표  
Table 1. Coordinates of Three Input Nodes in Space

노드	A	B	C
좌표	(-2.5, -2.0, 0.5)	(1.3, -1.8, 3.5)	(-2.0, 1.0, 3.0)

표 2. 세 입력 노드의 Z 평면상에서의 좌표  
Table 2. Coordinates of Three Input Nodes on Z-Plane

노드	A'	B'	C'
좌표	(-2.5, -2.0, 0.0)	(1.3, -1.8, 0.0)	(-2.0, 1.0, 0.0)

표 3. 세 입력 노드의 Z 평면상에서의 연결 길이 및 최소신장트리의 구성 선분 여부  
Table 3. Connecting Lengths of Three Input Nodes on Z-Plane and Member Edges of Built MST

선분	A'B'	B'C'	CA'
길이	3.81	4.33	3.04
MST 선분	○	×	○

표 4. 세 입력 노드의 3차원 공간에서의 연결 길이 및 최소신장트리의 구성 선분 여부  
Table 3. Connecting Lengths of Three Input Nodes in Space and Member Edges of Built MST

선분	AB	BC	CA
길이	4.85	4.36	3.94
MST 선분	×	○	○

표 5. 세 입력 노드로 구성된 2차원, 3차원 MST 연결 길이 및 비교  
Table 5. Connecting Lengths of 2D/3D MST for Three Input Nodes and Comparison

구분	연결길이	상대 비교
2D MST	AB+CA= 4.85+3.94= 8.79	5.90% 증가
3D MST	BC+CA= 4.36+3.94= 8.30	5.57% 감소

표 6은 본 논문에서 제안하는 3차원 최소신장트리 생성 알고리즘이다. 이는 대표적 2차원 최소신장트리 알고리즘인 Prim의 알고리즘을 기반으로 하였다[1]. 단계 1에서 3차원 공간상에 위치하는 입력 노드들의 좌표 정보를 활용하여 모든 노드들을 완전 연결시킨다. 이 연결들은  $n \times (n-1)$ 개의 3차원 공간 선분으로 표시된다. 3차원 입력 노드들은 미 선택 정점 집합에 포함시키고, 생성된 연결 선분들은 미 선택 선분 집합에 포함시킨다. 또한 선택 정점 집합과 선택 선분 집합은 공집합으로 생성한다. 단계 2에서는 단계 1에서 생성된 공간 선분들의 길이를 구한다. 예를 들어 공간상의 선분을 생성하는 입력 노드의 두 좌표가 각각  $(x_1, y_1, z_1), (x_2, y_2, z_2)$ 이라면 그 선분의 길이는 식 (1)을 이용하여 계산되어 저장된다.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \dots\dots\dots (1)$$

단계 3에서는 미 선택 정점 집합의 임의의 원소인 노드 S를 선택하고, S를 제외한 다른 원소인 노드들 중에서 가장 가까운 노드인 T를 선택하고 이들을 연결하는 선분 ST를 선택한다. 실제 선택을 위한 구현 과정은 자료 구조인 우선순위 큐(Priority Queue)를 이용한다. 정점 S와 T는 선택 정점 집합에 포함시키고, 선분 ST는 선택 선분 집합에 포함시킨다. 단계 4에서는 선택 정점 집합의 한 원소와 미 선택 정점 집합의 한 원소로 구성된 선분들 중에서 가장 작은 선분길이를 가지는 선분 S'T'를 선택한다. 이 선분에서 미 선택 정점의 원소인 정점 T'를 선택 정점 집합에 포함시킨다. 동시에 선분 S'T'은 선택 선분 집합에 포함시킨다. 단계 5에서는 미 선택 정점 집합이 공집합이 될 때까지 단계 4를 반복한다. 단계 6에서는 선택 선분 집합의 원소인 선분들을 구성원으로 최종 결과인 3차원 최소신장트리를 생성한다.

표 6의 3차원 최소신장트리 생성 알고리즘의 실행 시간을 분석하면 입력 노드의 수를  $N$ 이라 했을 때, 단계 1과 2에서 유클리드 최소신장트리를 위한 완전 연결 선분의 생성과 각 선분의 길이 계산을 위해서 식 (2)와 같은 실행시간이 필요하다. 여기서 생성된 연결 선분 수  $E$ 는  $O(N^2)$ 이다. 단계 3, 4, 5의 처리 과정은 우선순위 큐를 사용하여 구현한다. 즉 단계 2에서 계산된 각 선분의 길이를 키 값으로 하여 우선순위 큐에 입력한 후에 처리 노드를 포함한 선분들 중에서 가장 작은 길이를 가지는 선분을 선택하면서 3차원 최소신장트리의 구성 선분들을 찾는다. 이 과정은 우선순위 큐가 어떤 구조로 구현됨에 따라 성능이 달라지는데, 일반적인 이진 힙(Binary Heap)인 경우에는 식 (3)과 같은 시간이고, 노드의 수가 선분

의 수보다 매우 작은 환경에서 우선순위 큐를 피보나치 힙(Fibonacci Heap)으로 구현할 경우에는 식(4)와 같은 시간 안에 실행될 수 있다. 단계 6에서는 단계 5가 종료된 후 선택 선분 집합의 원소들에 대한 처리 과정이 필요하므로  $O(N-1) = O(N)$ 의 시간이 필요하다. 따라서 표 6의 전체적인 실행시간은  $O(N^2)$ 이라고 할 수 있다.

표 6. 3차원 최소신장트리 생성 알고리즘  
Table 1. Algorithm for 3-D Minimum Spanning Tree

단계	작업 내용
1	공간상의 모든 입력 노드들을 완전 연결하는 선분을 구한다. n개의 입력 노드에 대해 총 $n \times (n - 1)$ 개의 공간상의 선분이 생성된다. 입력 노드들은 미 선택 정점 집합에 포함시킨다. 선택 선분 집합은 공집합으로 생성한다.
2	임의의 두 노드 $A=(x_1,y_1,z_1)$ , $B=(x_2,y_2,z_2)$ 을 연결하는 선분 AB의 길이 $\ell$ 을 다음과 같이 구한다. $\ell = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$
3	임의의 한 정점인 노드 S를 선정하고 이 정점과 연결된 선분 중에서 가장 짧은 길이를 가지는 선분 ST를 선택한다. 선분 ST를 구성하는 S와 또 다른 노드 T를 선택 정점 집합에 포함시킨다. 선분 ST는 선택 선분 집합에 포함시킨다.
4	선택 정점 집합과 미 선택 정점 집합의 원소들 중에서 각각 하나씩 선택하여 그들의 길이를 구했을 때 최소의 길이를 가지는 선분ST를 선택한다. 여기서 S는 선택 정점 집합의 원소이고, T은 미 선택 정점 집합의 원소이다. T을 미 선택 정점 집합에서 제거하고 선택 정점집합에 포함시킨다. 선분 ST는 선택 선분 집합에 포함시킨다.
5	미 선택 정점 집합이 공집합이 될 때까지 단계 4를 반복한다.
6	선택 선분 집합의 원소들로 최종 결과(3차원 최소신장트리)를 생성 한다.

$$O(N \times (N - 1)) = O(N^2) \dots\dots\dots (2)$$

$$O(E \times \lg N) \dots\dots\dots (3)$$

$$O(E + N \times \lg N) \dots\dots\dots (4)$$

## 2. 실험 및 결과분석

### 2.1 실험방법

실험에 사용된 인자는 입력 노드 수와 공간비율이다. 공간 비율은 임의의 노드의 z축 좌표가 가질 수 있는 값이 x, y축의 최댓값에 대한 어느 정도 비율인가를 의미한다. 예를 들어 x축의 양의 방향의 최댓값이 6일 때, 공간비율 50%의 의미는 공간상의 임의의 노드의 z축 좌표 값이 최소 0부터 최대 3까지 값을 가질 수 있다. 본 논문을 위한 실험에서, 입력노드의 크기는 5000, 10000, 15000, 20000, 25000, 30000이고, 공간비율은 20%, 40%, 60%, 80%, 100%이다. 관찰 결과는 2차원 평면 좌표를 대상으로 한 Prim의 최소신장트리를 이용한 방법[1]과 본 논문에서 제안하는 표 6에서 기술된 3차원 최소신장트리 생성 방법에 의한 트리의 연결 비용, 즉 트리의 길이이다. 실험의 목적은 본 논문에서 제안하는 방법이, 평면상의 좌표를 근거로 생성된 3차원 최소신장트리 방법보다 더 길이가 감소됨을 검증하기 위함이다. 이를 위해 본 논문에서 제안하는 3차원 최소신장트리와 Prim의 최소신장트리 생성 알고리즘에 의해 생성된 3차원 트리의 선분 길이가 비교된다. 비교 방법들은 Intel 프로세서와 4 Giga 램의 Microsoft 윈도우즈 환경에서 Java로 구현하여 분석하였다.

### 2.2 실험 결과 및 분석

그림 8은 입력 노드의 수에 따라 Prim의 최소신장트리 알고리즘에 의해 생성된 트리와 본 논문에서 제안한 방법에 의해 생성된 트리의 2차원 평면 투영 트리의 연결 길이의 합을 분석한 결과이다. 2차원 평면에서만 고려했을 때, 당연히 Prim의 알고리즘에 의한 트리는 최단의 연결 길이를 보인다. 공간 입력 노드의 수가 5,000인 경우에 본 논문에서 제안된 방법은 Prim의 최소신장트리의 연결 길이에 비해 163.9%가 증가하였으나, 공간 입력 노드의 수가 30,000의 경우에는 251.2% 증가 하였다.

표 7과 그림 9는 입력 노드의 수에 따라 Prim의 최소신장트리 알고리즘에 의해 생성된 트리와 본 논문에서 제안한 방법에 의해 생성된 트리의 실제 3차원 연결 길이의 합을 분석한 결과이다. 공간 입력 노드의 수가 5,000인 경우에 본 논문에서 제안된 방법은 Prim의 최소신장트리의 3차원 연결 길이에 비해 81.7%가 감소하였고, 공간 입력 노드의수가 30,000의 경우에는 90.0% 감소하였다. 따라서 실험을 통해 제안된 방법은 공간에서의 입력 노드 수가 클수록 Prim의 2차원 최소신장트리 알고리즘보다 좋은 성능을 보임을 확인할 수 있다.

그림 10은 공간비율, 즉 x 또는 y 의 최댓값에 대한 z축 좌표의 최댓값 비율에 따라 Prim의 최소신장트리 알고리즘에 의해 생성된 트리와 본 논문에서 제안한 방법에 의해 생성된 트리의 2차원 평면 투영 트리의 연결 길이의 합을 구한 것이다. 2차원 평면에서만 고려했을 때, 당연히 Prim의 알고리즘에 의한 트리는 최단의 연결 길이를 보인다. 공간 입력 노드의 수가 30,000일 때, 공간비율이 20%인 경우 본 논문에서 제안된 방법은 Prim의 최소신장트리의 연결 길이에 비해 2차원 평면에서의 길이는 108.6%가 증가하였으나, 공간비율이 100%인 경우에는 251.2% 증가 하였다.

표 8과 그림 11은 공간 비율에 따라 Prim의 최소신장트리 알고리즘에 의해 생성된 트리와 본 논문에서 제안한 방법에 의해 생성된 트리의 실제 3차원 연결 길이의 합을 분석한 결과이다. 공간 입력 노드의 수가 30,000일 때, 공간비율이 20%인 경우 제안된 방법은 Prim의 최소신장트리의 연결 길이에 비해 3차원 공간에서의 길이가 70.7% 감소하였으나, 공간비율이 100%인 경우에는 90.0% 감소하였다. 따라서 실험을 통해 제안된 방법은 Prim의 2차원 최소신장트리 알고리즘보다 공간비율이 클수록, 즉 최대 허용 또는 처리 높이가 클수록 상대적으로 좋은 성능을 보임을 확인할 수 있다.

표 7. 입력 노드 수에 따른 3D 연결 길이 합 비교  
Table 7. Comparison of Three Dimensional Connecting Length Sum for Input Node Numbers

노드 수	5,000	10,000	15,000	20,000	25,000	30,000
Prim 알고리즘	1006	2025	3029	4021	5024	5987
제안방법	1841	2915	3805	4594	5328	6017
절감율	81.7%	85.6%	87.4%	88.6%	89.4%	90.0%

표 8. 공간비율에 따른 3D 연결 길이 합 비교  
Table 8. Comparison of Three Dimensional Connecting Length Sum for Space Ratios

공간비율	20%	40%	60%	80%	100%
Prim 알고리즘	1216	2405	3598	4792	5987
제안방법	3560	4457	5088	5593	6017
절감율	70.7%	81.5%	85.9%	88.3%	90.0%

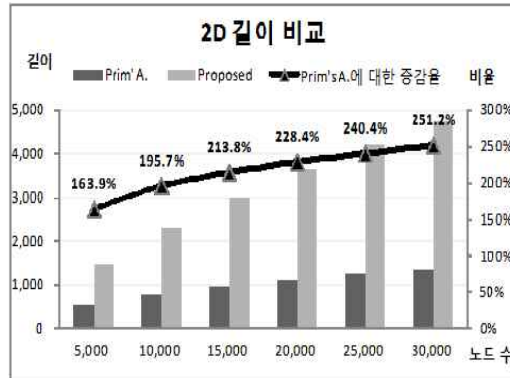


그림 8. 입력 노드 수에 따른 평면에서의 연결 길이의 합  
Fig. 8. Connecting Length Sum in Two Dimensions for 6 Kinds of Input Node Numbers

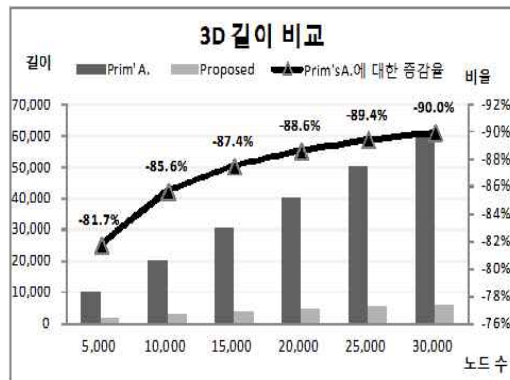


그림 9. 입력 노드 수에 따른 공간에서의 연결 길이의 합  
Fig. 9. Connecting Length Sum in Three Dimensions for 6 Kinds of Input Node Numbers

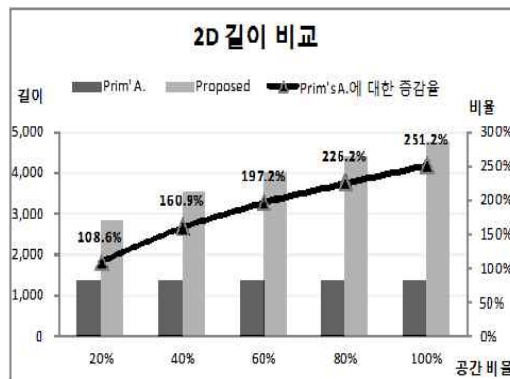


그림 10. 공간비율에 따른 평면에서의 연결 길이의 합  
Fig. 10. Connecting Length Sum in Two Dimensions for 5 Kinds of Space Ratios



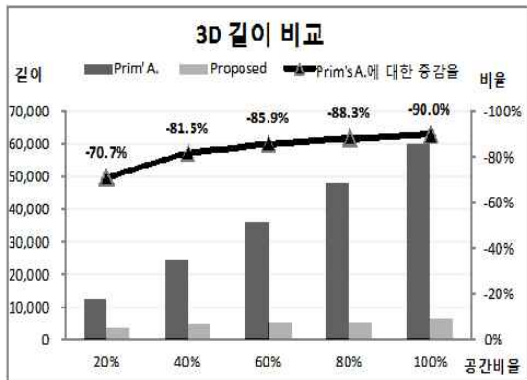


그림 11. 공간비율에 따른 공간에서의 연결 길이의 합  
Fig. 11. Connecting Length Sum in Three Dimensions for 5 Kinds of Space Ratios

#### IV. 결론

본 논문은 네트워크 구성의 최적화 문제에서 주로 사용되는 2차원 최소신장트리 문제를 3차원 문제에 확대 적용한 것이다. 실세계의 많은 문제들은 2차원 평면 보다는 3차원 공간에 관한 문제가 더 많다. 특히 산업시대가 많은 우리나라에서는 이러한 고려를 더 해야 할 필요가 있다. 또한 가까운 장래에 우주나 심해 자원의 활용이 중요하게 될 것이다. 공간상의 자유로운 이동이나 필요한 설비의 설치 및 관련 응용에서 본 논문에서 제안하는 3차원 최소신장트리는 유용한 해결책을 제시할 것이다.

100% 공간비율 환경에서의 30,000개의 공간상의 노드들을 연결함에 있어, 이를 이차원 평면으로 투영된 평면에서의 최소신장트리는 본 논문에서 제안된 방법으로 3차원 공간에서 생성된 최소신장트리에 비해 길이는 비록 25.2% 감소되었지만, 공간을 고려한 여건에서 이것은 의미가 없다고 할 수 있다. 오히려 시간적인 차이에 의해 공간에서 90.0%의 연결 길이의 증가를 가져왔다. 본 논문에서 제안한 방법은 공간 비율이 높을수록, 그리고 입력 노드의 수가 많을수록 2차원 최소신장트리에 비해 좋은 길이 절감율을 보인다. 이는 제안된 방법이 공간상에 위치한 많은 노드들을 대상으로 높이나 깊이의 편차가 많은 즉 공간비율이 높은 환경에서 좋은 성능을 보일 수 있음을 나타낸다.

향후 연구는, 최소 신장트리와 같은 P(Polynomial) 문제 영역에서의 해보다 좋은 연결 비용의 트리를 얻을 수 있는 NP(Non-Polynomial) 문제 영역인 3차원 스타이너 트리(3D Steiner Tree)에 관한 것이다. 이미 2차원 영역에서의

스타이너 트리에 관한 많은 연구가 실행되었으나 3차원 공간에서의 연구는 그리 활발하지 않다. 2차원 문제에 대해서도 최소신장트리보다 스타이너 최소 트리가 좋은 성능을 보인 것처럼 3차원에서도 유사한 결과 얻을 수 있을 것으로 추정된다. 그러나 스타이너 트리문제는 NP 문제이므로 적절한 휴리스틱의 개발 및 검증이 필요하다. 향후 이러한 연구를 통해 본 논문에서 제안된 방법보다 개선된 방법을 제시할 수 있을 것이다.

#### 참고문헌

- [1] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, "Introduction to Algorithms," 2nd Ed., THE MIT Press, pp.561-579, 2001.
- [2] I. Kim, S. Kim, "Mechanism for Building Approximation Edge Minimum Spanning Tree Using Portals on Input Edges," The KIPS Transactions: PartA, Vol.16A, No.6, pp.509-518, 2009.
- [3] B. Chun, Y. Kim, "A Method for Character Segmentation using MST," Journal of the Korea Society of Computer and Information, Vol.11, No.3, pp.73-78, 2006.
- [4] P. Amat, W. Puech, S. Druon, J. P. Pedebay, "Lossless 3D Steganography Based on MST and Connectivity Modification," Image Communication, Vol.25, No.6, pp.400-412, 2010.
- [5] H. Woo, G. Kim, "Detection of Entry/Exit Zones for Visual Surveillance System using Graph Theoretic Clustering," Journal of the Institute of Electronics Engineers of Korea SC, Vol.46, No.6, pp.1-8, 2009.
- [6] H. Lee, "Convert 2D Video Frames into 3D Video Frames," Journal of the Korea Society of Computer and Information, Vol.14, No.6, pp.117-123, 2009.
- [7] A. Zhang, W. Sun, S. Hu, C. Qian, "Automatic Global Registration of Multiple 3D Data Sets from Outdoor Urban Environments Based on Feature Units," SCCG'04 Proceedings of the 20th Spring Conference on Computer Graphics, pp.193-199, 2004.
- [8] J. Go, Y. Choi, S. Jang, K. Lee, "The Analysis of 3D



- Position Accuracy of Multi-Looking Camera," Journal of Korea Spatial Information Society, Vol.19, No.3, pp.33-42, 2011.
- [9] S. Kim, J. Kang, Y. Choe, S. Park, I. Shim, S. Ahn, M. Chung, "The Development of Sensor System and 3D World Modeling for Autonomous Vehicle," Journal of Institute of Control, Robotics and Systems, Vol.17, No.6, pp.531-538, 2011.
- [10] N. Zhu, A.C.S. Chung, "Minimum Average-Cost Path for Real Time 3D Coronary Artery Segmentation of CT Images," MICCAI'11 Proc. of the 14th international conference on Medical Image Computing and Computer Assisted Intervention, Vol.III, pp.436-444, 2011.

## 저 자 소 개



### 김 재 각

1981 : 숭실대학교 전자계산학과  
공학사.

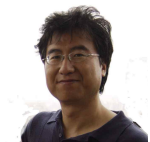
1985 : 연세대학교 산업대학원  
전자계산 전공 석사.

2002 : 숭실대학교 컴퓨터과학과  
공학박사.

현 재 : 김포대학 사회복지과 교수

관심분야 : 암호이론, 시스템보안, 알  
고리즘

Email : ckkim@kimpo.ac.kr



### 김 인 범

1989 : 서울대학교 컴퓨터공학과  
공학사.

1991 : 서울대학교 컴퓨터공학과  
공학석사.

2007 : 위스콘신 주립대학-밀워키  
컴퓨터과학과 공학박사.

현 재 : 김포대학 인터넷정보과 부교수

관심분야 : 네트워크, 데이터베이스

Email : ibkim@kimpo.ac.kr

