

3D레이저스캐닝을 이용한 옥트리기반 구조물 형상정보 가시화

차기춘¹, 이동환¹, 박승희^{2*}

¹성균관대학교 미래도시융합공학과, ²성균관대학교 건설환경공학부

Visualization of Structural Shape Information based on Octree using Terrestrial Laser Scanning

Gichun Cha¹, Donghwan Lee¹, Seunghee Park^{2*}

¹Department of Convergence Engineering for Future City, Sungkyunkwan University

²School of Civil, Architectural Engineering and Landscape Architecture, Sungkyunkwan University

요약 본 논문은 대형구조물의 형상관리를 위해 3D 레이저 스캐닝을 이용하여 옥트리기반 구조물 형상정보 가시화를 진행하였다. 이를 위해 3D스캐너데이터를 옥트리 데이터 구조로 변환할 수 있는 프로세스를 정립하고, 메쉬기법과 래스터/벡터변환 처리를 통해 점(point) 데이터가 2차원 면 형태를 거쳐 3D객체로 생성되는 프로세스를 진행하였다. 생성된 3D객체는 옥트리 데이터 구조로 전달할 수 있는 형식인 Binary file type로 변환하는 작업이 진행되었고, 변환된 Binary file을 옥트리 최상의 노드인 Root노드로 전달하였다. Root 노드를 시작으로 옥트리 내부에서의 세부분할 작업 후 내부노드 데이터 저장과 비어있는 영역제거를 통해 옥트리기반 구조물 형상정보모델을 구축하였다. 본 연구가 수행된 Test-bed는 성균관대학교 내에 위치한 강교량 구조물로, 구축된 옥트리기반 형상정보는 스캐너데이터를 89.3% 압축하였으며 독일의 사무용빌딩, 대학캠퍼스와 미국 소재 단과대학건물 스캐너데이터와의 비교를 통하여 옥트리 데이터 압축률을 확인하였다. 본 연구를 통해 대형구조물 및 복층터널의 내부형상정보관리를 위한 형상정보 가시화의 기반을 마련하였으며, 형상정보 가시화를 통해 구조물 모니터링 및 유지관리 효율을 높일 수 있을 것이라 기대한다.

Abstract This study presents the visualization of shape information based on Octree using 3D laser scanning. The process of visualization was established to construct the Octree structure from the 3D scan data. The scan data was converted to a 2D surface through the mesh technique and the surface was then converted to a 3D object through the Raster/Vector transformation. The 3D object was transmitted to the Octree Root Node and The shape information was constructed by the recursive partitioning of the Octree Root Node. The test-bed was selected as the steel bridge structure in Sungkyunkwan University. The shape information based on Octree was condensed into 89.3%. In addition, the Octree compressibility was confirmed to compare the shape information of the office building, a computer science campus in Germany and a New College in USA. The basis is created by the visualization of shape information for double-deck tunnel and it will be expected to improve the efficiency of structural health monitoring and maintenance.

Keywords : 3D Scan Data, Double Deck Tunnel, Octree, Structural Shape Management, Terrestrial Laser Scanning

1. 서론

1.1 연구의 배경 및 목적

3D레이저스캐닝 장비가 대형구조물에 사용되어짐에 따라 대용량 스캐너데이터의 가공처리 연구가 진행되고 있다[1-2]. 대형구조물로부터 얻어지는 스캐너데이터는 그

본 논문은 국토교통부의 U-City 석·박사과정 지원사업과 2015년 건설기술연구사업의 ‘대심도 복층터널 설계 및 시공 기술개발 (15SCIP-B089428-02)’ 연구단을 통해 수행되었습니다. 연구지원에 감사드립니다.

*Corresponding Author : Seunghee Park (Sungkyunkwan University)

Tel: +82-31-290-7525 email: shparkpc@skku.edu

Received June 1, 2016

Revised July 19, 2016

Accepted August 11, 2016

Published August 31, 2016

크기가 방대하기에 컴퓨터 메모리용량을 넘어 모든 데이터를 한번에 로드하지 못한다. 때문에 메모리 용량만큼 데이터를 로드하고 해제하는 과정을 반복하게 되며 이 과정에서 많은 시간이 소요된다.[3] 따라서 대용량 스캔 데이터를 신속하고 정확하게 가시화하기 위한 전처리 및 경량화 연구가 이루어지고 있다. 스캔데이터의 경량화를 위한 연구로는 point로부터 삼각형(triangle)과 같은 도형(polygon)을 구성하여 처리하는 방법과 point cloud를 2D 이미지로 렌더링하여 픽셀(pixel)의 형태에 따라 3D 복셀(voxel)의 크기를 결정하는 Splat 방법이 있다. Splat는 3차원 공간에서 구의 형태로 존재하지만 법선 정보와 2차원 화면에 투영되는 크기를 이용해 Splat의 렌더링 크기를 결정하는 이 방법은 point cloud의 일부가 비어 보이는 구멍(hole)현상을 해결하기 좋은 장점이 있지만 시각적결과가 좋지 않아 이론적 개념으로 남아있다 [4-7].

이 외에도 그래픽 지향적 자료구조를 이용하여 데이터 경량화 및 가시화 속도를 향상시키는 연구들도 진행되었다. 관련 연구로 Sequential Point Trees는 Splat의 반지름으로 point를 정렬하여 순차적으로 렌더링 하지만 point들이 그래픽 카드 메모리에 저장되어 있어야 한다 [8]. 이를 external memory algorithms 기법으로 개선한 방법이 Layered Point Clouds, XSplate, Instant Points, 등이다. Layered Point Clouds는 대용량 point cloud의 검색 속도 개선을 위해 정규 샘플링의 결과를 이진트리(binary tree)에 저장하였고, XSplate는 그래픽 카드의 캐시(cache) 낭비 방지, Instant Points는 원본 point 렌더링에 중점을 두었다.[9-12] 또한 Multi-way kd-Trees는 KD트리의 노드가 다수의 자식을 가지도록 하여 최종 노드에 저장되는 point들의 개수를 비슷한 숫자로 조절하였고 그 결과 그래픽 카드가 사용하는 캐시를 일정 크기로 고정시켰다[13]. 그리고 유사한 Tree구조를 갖는 Octree는 최대 8명의 자식을 가지며, 전체형상에 대한 바운딩 박스를 생성하고 반복적으로 세분화함으로써 데이터 경량화 및 가시화연구에 사용되며, 컴퓨터 그래픽스, 로보틱스 등 다양한 연구 분야에서 사용되고 있으며, 메모리 사용량 및 탐색속도 면에서 합리적인 방식임이 증명되었다[14-22].

본 연구에서는 이러한 배경을 바탕으로 스캔데이터 경량화, 효율적인 메모리 사용, 3D 가시화가 가능한 Octree를 이용하여 구조물 형상정보 가시화 연구를 진행

하였다. 본 논문의 구성은 2장에서 Octree 자료구조 및 저장방법, 3장에서 실험환경 및 Test-bed를 대상으로 진행한 옥트리 기반 형상정보 가시화 연구에 대해 소개한다. 4장에서는 결론과 더불어 향후 연구방향을 제시한다.

1.2 연구의 범위 및 방법

본 연구에서는 복층터널과 같은 대형구조물의 형상정보 가시화를 통해 구조물의 형상변화에 따른 유지관리가 가능한 형상정보모델을 구축한다. 이를 위해 3D스캔정보 취득을 위해 Terrestrial Laser Scanning(TLS)장비를 이용하였으며 세부 연구 단계로는 다음과 같다.

- 3D스캔데이터를 이용한 옥트리 구축
- 옥트리 기반 구조물 형상정보 가시화

2. 형상정보 가시화를 위한 옥트리 구현

2.1 Octree Data Structure

옥트리는 Fig. 1처럼 주어진 형상에 대한 바운딩 박스(Bounding Box)를 생성하고 3D 공간 상에서 x, y, z축 방향으로 공간을 각각 이등분하여 분할된 육면체 내에 형상이 포함되면 가시화 대상으로 저장하는 데이터표현 방법이다[23]. 옥트리는 x, y, z축의 각 방향에 대해 이등분한 공간을 다시 8개의 서브 공간으로 분할하여 보다 세밀한 영역으로 표현한다. 분할된 8개의 서브 공간은 노드(node)라고 불리는 요소형태로 트리 구조에 저장 가능하다. 옥트리는 바운딩 박스를 분할하는 수에 따라 단계(Depth level)가 결정된다.[24-25]

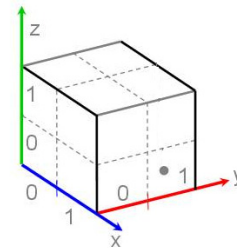


Fig. 1. Octree space partitioning

옥트리는 최대 $N_L = 8^D$ 의 Leaf node를 가지며, $2^D \times 2^D \times 2^D$ 의 해상도를 가진 3D 그리드와 동일하다. Fig. 2는 옥트리 Depth level에 따른 노드 수를 나타

내며, 트리 노드의 총수는 식(1)과 같다.

$$N_T = \sum_{i=0}^D 8^i = \frac{8^{D+1} - 1}{7} \quad (\text{식1})$$

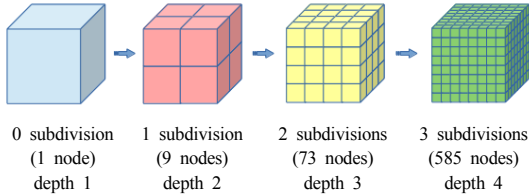


Fig. 2. Octree depth level[26]

Fig. 3은 옥트리 데이터 구조를 나타내며 바운딩박스 중심값, 바운딩박스 크기, 자식노드 주소값 배열, 인근 포인트, 현 노드의 주소 값을 가지고 있다[27-28].

```
struct Octree {
    float center[3];
    float size[3];
    Octree *child[8];
    int nr_points;
    float **point;
};
```

Fig. 3. Definition of an octree

2.2 Octree 인코딩

옥트리를 인코딩하는 방법으로는 Tamminen이 제안한 DF-representation이 가장 많이 쓰인다. 이 방법은 옥트리를 깊이우선(depth-first)으로 탐색하여 만나는 노드의 정보들을 순차적으로 저장하는 방법으로 Fig. 4처럼 각 노드는 Black, White, Gray로 표현된다. Black 노드는 그 영역이 가득 차있음을 나타내고 White노드는 완전히 비어있음을, Gray는 노드는 부분적으로 차있음을 의미한다. 여기서 Gray노드의 경우에는 추가적으로 분할을 계속해야 함을 나타낸다[29]. 옥트리는 분할될 depth의 레벨을 정해주거나 바운딩박스 크기설정을 통해 반복적으로 분할이 가능하다. 본 연구에서는 옥트리를 이용하여 3D스캔 데이터를 샘플링하고 구조물 형상을 가시화할 수 있는 정보모델구축에 사용하였다.

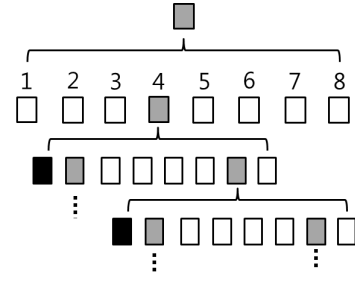


Fig. 4. DF-representation

3. Test-bed 및 형상정보구축

3.1 Test-bed 및 장비현황

3.1.1 Test-bed 현황

Test-bed는 성균관대학교 제1공학관과 제2공학관을 연결하는 강교량 구조물이며 Fig. 5는 Test-bed 전경을 나타낸다. 구조물 크기는 Fig. 6에서 확인 할 수 있듯이 총 길이 34m, 폭 2m이며 사이 경간은 28m이다.



Fig. 5. View of Test-bed

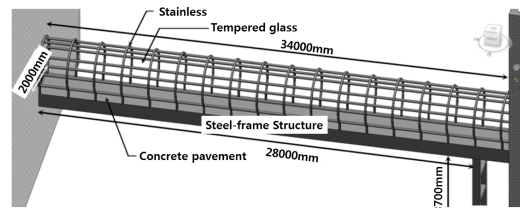


Fig. 6. Condition of Steel-frame Structure

3.1.2 TLS 및 실험환경 현황

본 연구에서는 Test-bed의 레이저스캔을 위해 Table 1과 같이 Leica ScanStation C10/C5 모델을 활용하였으

며, 최대측정거리 300m, 표면정밀도 2mm, 측정속도 50,000 point/sec의 성능을 갖추고 있다.

Table 1. Specification of Terrestrial Laser Scanner

	Model	Leica ScanStation C10/C5
	Measurement distance	300m
	Spot size	From 0-50 m: 4.5 mm (FWHH-based), 7 mm (Gaussian-based)
	Range accuracy	35m at 300m
	Precision	2 mm
	Speed	50,000 point/sec
	Range	Horizontal 360°(max) Vertical 270°(max)
	Laser Class	3R (IEC 60825-1)
	Memory	80 GB

대상 Test-bed의 3D스캐너는 TLS를 통해 습득하였으며 데이터 형태는 각각의 point로 x,y,z 좌표점을 갖는다.

옥트리 구축을 위한 개발 작업환경 및 컴퓨터 환경 사양은 Table 2에서 확인 할 수 있다. 운영체제는 Linux인 ubuntu를 사용하였으며, Programming Language로는 C++를 사용하였다.

Table 2. Development environment in computer

	Specification	Set
CPU	8-Core Intel Xeon E5-2670 (2.6GHz, 20MB L3, 8.0GT/sec QPI)	1
RAM	8GBG DDR3 ECC-Registered / PC3-12800	8
HDD	SATA 1TB 32MB Buffered / 7200rpm	1
POWER	865W AC power supply w/PFC	1
OS	ubuntu 12.04.2 LTS (Linux) / 64bit	-

3.2 옥트리기반 형상정보모델 구축 및 가시화

3.2.1 3D스캐너데이터를 이용한 옥트리 구축

3D레이저스캐닝을 이용하여 취득한 스캔데이터는 Fig. 7과 같다. Fig. 7는 Test-bed의 하부영역을 따로 추

출하여 나타난 것이며, 16,777,216개의 점(Point)으로 구성되어 있다.

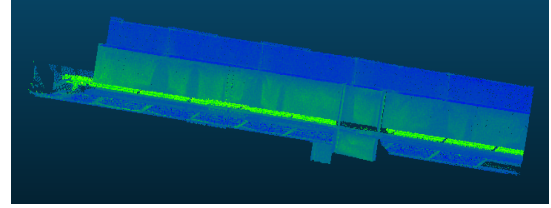


Fig. 7. 3D Scan Data of Test-bed

취득된 스캔데이터는 Fig. 8와 같은 단계를 거쳐 옥트리를 구축하였다. 스캔데이터는 메쉬기법을 적용하여 점(point) 데이터를 2차원 면(surface)형태로 변환하고, 변환된 2차원 면을 래스터/벡터변환 처리방법을 이용하여 3D 객체를 생성하였다. 래스터 데이터는 Fig. 9에서 확인할 수 있듯이 일정 크기의 격자의 집합으로 구성되어 있으며, 특정 위치의 격자 값을 저장하고 있다.

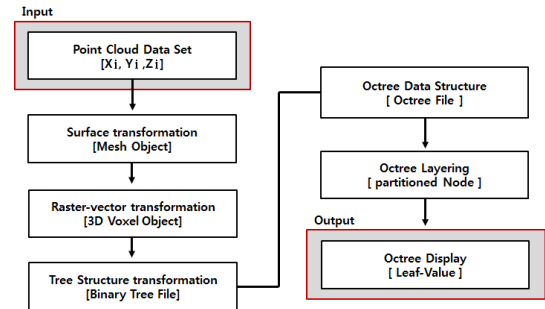


Fig. 8. Process of Octree transformation using Point Cloud Data (3D Scan Data)

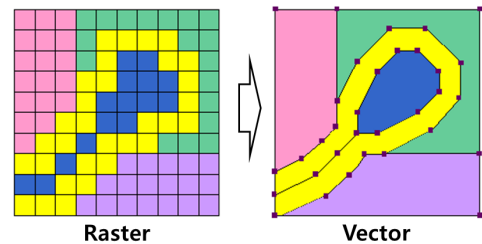


Fig. 9. Example of Raster-Vector Transformation

이러한 래스터 데이터를 공간 정보의 기본 단위인 점, 선, 면을 사용하여 실제계의 위치를 2차원 또는 3차원의

좌표값을 갖는 벡터데이터를 생성하였다. Fig. 10은 2차원 면 형태의 메쉬 오브젝트이며, Fig. 11은 메쉬 오브젝트에 래스터/벡터변환을 적용하여 생성된 3D Voxel을 나타낸다.

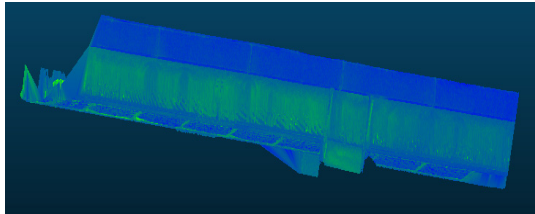


Fig. 10. Surface Transformation (Mesh Object)

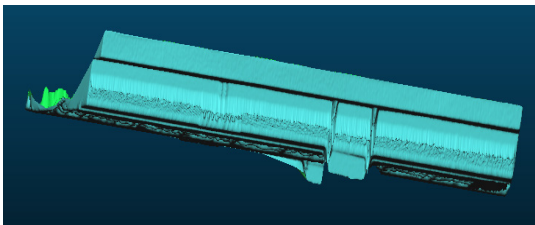


Fig. 11. Raster-Vector Transformation (3D Object)

3D Voxel을 binary type으로 변환하기 위해 binvox 프로그램을 이용하였다. binvox 프로그램은 voxel 파일을 읽어 들여 binary file을 생성해주며 Fig. 12와 같이 각각의 X, Y, Z 방향을 기준으로 voxel 파일을 읽어 들인다. Fig. 13은 생성된 binary file을 가시화 한 것을 나타내며, 이를 통해 옥트리 정보모델을 구축할 수 있는 데이터 구조를 생성하였다.

Binary file은 3D 객체의 x, y, z 좌표 값, 깊이, 높이, 너비 값을 저장하고 있으며 옥트리 정보모델 구축을 위한 입력 값 파라미터로 사용된다. Algorithm. 1은 Binary file을 이용하여 Octree file을 생성하는 수도코드이며, 변수 res 값을 통해 옥트리 해상도를 결정하게 된다. 옥트리를 생성한 후 res 값의 해상도 크기가 될 때까지 옥트리 내부 노드는 반복적으로 분할한다. 분할이 끝난 후 Binary file의 3D 객체의 중심 point(x,y,z) 값과 깊이, 높이, 너비 값을 순차적으로 읽어 들여 옥트리 내부 노드에 저장하게 된다. 모든 point값을 저장하게 되면 비어있는 영역은 제거되고 마지막으로 옥트리 파일을 생성하게 된다. 데이터가 저장된 옥트리 내부노드 간 관계는 Fig. 14

에서 확인 할 수 있다. 총 4개의 클래스가 있으며 각 클래스가 관리하는 정보는 다음과 같다.

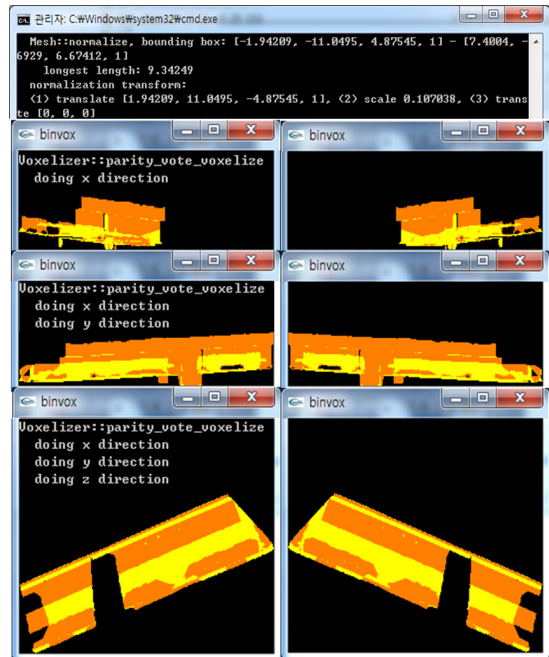


Fig. 12. The binary conversion of the voxel object

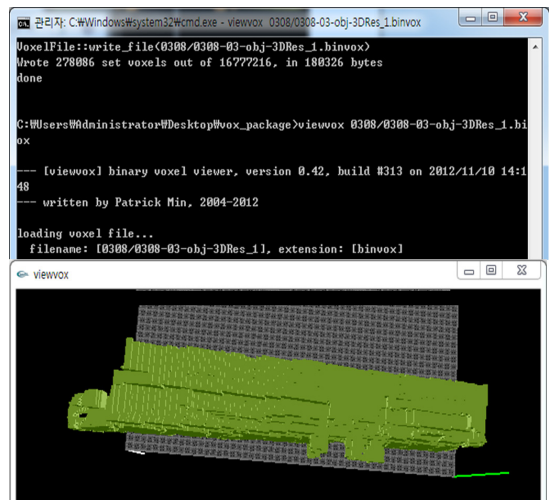


Fig. 13. The view of binary voxel type

Algorithm 1. Octree Data Processing

```

Input Binary file(index, value(point, depth, height, width))

1: resolution res = user input
2: generate new octree (res)
3: while((index != end)
4:   read mesh → value
5:   mark value → free or occupied
6:   if(value == 1)
7:     octree → updateNode(value)
8:   else octree → deleteNode(value)
9:   end if
10:  next index
11: end while
12: write octree → writeBinary()
13: return octree file

```

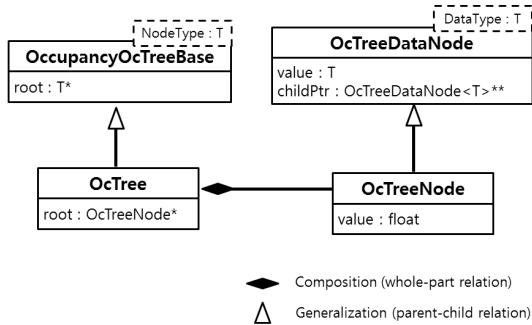


Fig. 14. Class Relation Diagram of Octree

- ① Octree : 최상위 Root 및 모든 OctreeNode 값
- ② OctreeNode : 분할된 모든 Node들의 주소 값
- ③ OctreeDataNode : 더 이상 분할 할 수 없는 최하위 Leaf Node의 바운딩 박스 중심 좌표값
- ④ OccupancyOctreeBase : Octree에서 정보가 담긴 Node들의 값 (값이 없는 빈 Node 제외)

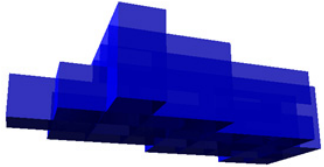
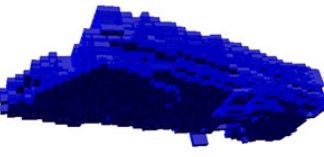
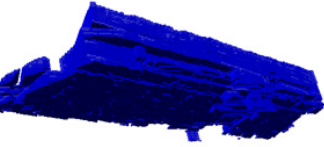
① 클래스는 바이너리 트리 값으로 변환된 3D 객체 전체를 감싸고 있는 바운딩 박스(최상위 Root 노드)로 모든 ② 클래스들을 관리한다. ② 클래스는 각 Depth Level별 분할된 Node 주소 값이며, 하나의 노드는 최대 8명의 자식노드 정보를 저장하고 있으며 ① 클래스와 부모-자식관계를 이루고 있다.

③ 클래스는 자식노드가 없는 가장막내노드(Leaf Node)의 값을 담고 있다. ④ 클래스는 옥트리 내부의 빈 Node를 제외한 실제 형상정보를 저장하고 있는 Node들만 따로 담고 있는 클래스이다. 비어있는 부분은 가시화할 필요가 없기 때문에 ④ 클래스를 통해 형상정보를 표현하게 된다.

3.2.2 옥트리기반 구조물 형상정보 가시화

생성된 옥트리의 형상정보를 확인하기 위해 Octovis를 이용하였다. Octovis는 3D tool kit 프로그램으로 옥트리 데이터를 3D 형상으로 가시화해주는 뷰어 프로그램이다. Table. 3은 구축된 옥트리의 형상정보이며 Depth 11, 14, 16 level에서의 모습을 확인 할 수 있다. 최하위 분할인 16 level의 해상도는 0.036m로 Fig. 7에서 추출된 Test-bed의 스캔데이터를 0.036m 크기의 바운딩박스 값을 가진 형상정보로 구축된 것을 확인 할 수 있다. 구축된 형상정보는 3D 스캔데이터와 비교하여 데이터 크기가 89.32%로 줄어들었으며, Table. 4 정보를 통해 확인할 수 있다.

Table 3. Shape Information Model of Octree

Shape Visualization	Octree depth	Resolution
	11	1.167m
	14	0.144m
	16	0.036m

스캔데이터로부터 취득된 9.3 x 5.4 x 1.8m³ 크기를 가진 구조물 형상정보를 2,355KB의 크기로 경량화하면서 0.036m의 바운딩박스 위치정보를 갖는 정보모델을

Table 4. Data compression rate among different octree shape information

Dataset	Area [m ²]	Resolution [m]	Nodes [Num.]	Depth	Octree size [MB]	Scan size [MB]	Compression [%]
Test-bed, Sungkyunkwan University	9.3 x 5.4 x 1.8	0.036	2,406,992	16	2.3	21.5	89.3
Office building, University of Freiburg[30]	43.8 x 18.2 x 3.3	0.5	87,289,760		24.7	78.8	68.6
Computer science campus, University of Freiburg[30]	292 x 167 x 28	0.5	106,059,088		990.6	5162.9	80.8
New College, Oxford[30]	250 x 161 x 33	0.5	273,130,784		395.4	5058.7	92.1

구축하였다. Test-bed 이외의 Table. 4에 기재되어 있는 데이터는 독일의 Albert Ludwig University of Freiburg의 사무용건물과 컴퓨터공학 캠퍼스, 그리고 미국 Oxford University의 단과대학인 New College의 스캔데이터[30]를 다운받아 옥트리 형상모델의 압축률을 나타낸 것이다. 사무용 빌딩의 경우 압축률이 68.6%로 다른 Case와 차이가 나는 것은 옥트리 내부에 채워진 노드가 적어 압축률이 다른 Case에 비해 작은 결과를 나타내었다.

4. 결론

본 연구에서는 3D레이저스캐닝을 이용하여 옥트리 기반 형상정보 가시화를 진행하였다. 옥트리는 바운딩박스 생성을 통해 3D 공간 상에서 x, y, z축 방향으로 공간을 이등분하며, 분할된 바운딩 박스 내에 물체가 포함되면 데이터를 저장하는 데이터표현방법이다. 스캔데이터를 옥트리구조로 변환하기 위해 메쉬기법을 적용하여 Point 데이터를 2차원 면으로 표현하고 이를 다시 래스터/벡터 변환 처리를 통해 2차원 면을 3D 객체로 생성하였다. 생성된 3D 객체는 Binary file로 변환하여 옥트리를 생성하고 옥트리 분할 과정을 통해 형상정보모델을 구축하였다. 연구가 수행된 Test-bed는 성균관대학교 1공학관과 2공학관을 연결하는 강교량 구조물이었으며, 9.3 x 5.4 x 1.8m 크기를 가진 스캔데이터를 21.5MB에서 2.3MB로 89.3%가 압축되면서 각 영역별로 0.036m의 바운딩 박스 위치정보를 갖는 형상정보모델을 생성하였다. 그리고 Test-bed 이외의 사무용빌딩, 대학 캠퍼스, 단과대학 건물 스캔데이터를 이용하여 옥트리 형상모델과 비교하였고 모두 16 Depth까지 분할하여 그 값을 비교하였다.

그 결과 사무용건물의 압축률은 68.6%로 가장 낮았고, 단과대학건물은 92.1%로 4가지 Case중 가장 높았으며, 옥트리 내부에 채워진 노드 수가 많을수록 압축률이 높은 것으로 나타났다. 따라서 복층터널과 같은 원형 단면의 규칙적인 형태를 띠고 있는 대형구조물의 대용량 스캔데이터를 이용하여 내부형상정보를 구축할 수 있는 방법을 마련할 수 있었다. 옥트리 기반 형상정보는 복층터널 내부형상관리 수행 시 확인하고자 하는 취약부영역의 바운딩박스 위치로 이동 후 바운딩 박스 범위를 확장하여 내부형상변화를 확인 할 수 있는 형상관리모델로써 사용이 가능할 것이다.

향후연구로 옥트리 기반 형상정보의 변화를 탐색하여 구조물의 처짐 및 변형을 측정할 수 있는 알고리즘과 실제 변형된 값과의 비교를 통해 옥트리 기반 형상정보모델의 정확도를 검증하기 위한 연구가 필요할 것이다.

References

- [1] Lee, H. M., "Improved Model for Measurement of Displacements and Deformed Shapes of Beam Structures using Terrestrial Laser Scanning" JOURNAL OF THE ARCHITECTURAL INSTITUTE OF KOREA Structure & Construction, Vol. 24, No. 4, pp. 3-14, 2008.
- [2] Bae, S. W., "A Terrestrial LiDAR Based Method for Detecting Structural Deterioration, and Its Application to Tunnel Maintenance", The Journal of Engineering Geology, Vol. 25, No. 2, pp. 227-235, 2015.
DOI: <http://dx.doi.org/10.9720/kseg.2015.2.227>
- [3] Yang, S. C., Han, S. H., Heo, J., "Massive 3D Point Cloud Visualization by Generating Artificial Center Points from Multi-Resolution Cube Grid Structure", Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography, Vol. 30, No. 4, pp. 335-342, 2012.
DOI: <http://dx.doi.org/10.7848/ksgpc.2012.30.4.335>

- [4] Delaunay, "Sur la sphere vide", Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk, Vol. 7, pp. 793-800, 1934.
- [5] Gustav K., "Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen", Journal für die Reine und Angewandte Mathematik, Vol. 40, pp. 209-227, 1850.
- [6] Westover L., "Interactive Volume Rendering", Proceedings of the 1989 Chapel Hill workshop on Volume visualization, ACM, New York, pp. 9-18, 1989. DOI: <http://dx.doi.org/10.1145/329129.329138>
- [7] Rusinkiewicz S., trimesh2, Princeton University, New Jersey, <http://gfx.cs.princeton.edu/proj/trimesh2> (last date accessed: 13 Apr. 2016).
- [8] Dachsbacher C., Vogelsgang C. and Stamminger M., "Sequential Point Trees", ACM Transactions on Graphics, ACM, Vol. 22, No. 3, pp. 657-662, 2003. DOI: <http://dx.doi.org/10.1145/882262.882321>
- [9] Han, S. H., "Implementation of File-referring Octree for Huge 3D Point Clouds", Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography, Vol. 32, No. 2, pp. 109-115, 2014. DOI: <http://dx.doi.org/10.7848/ksgpc.2014.32.2.109>
- [10] Gobbetti E. and Marton F., "Layered Point Clouds: a simple and efficient multiresolution structure for distributing and rendering gigantic point-sampled models", Computers & Graphics, Pergamon Press, Vol. 28, No. 6, pp. 815-826, 2004. DOI: <http://dx.doi.org/10.1016/j.cag.2004.08.010>
- [11] Pajarola R., Sainz M. and Lario R., "XSplat: External Memory Multiresolution Point Visualization", International Conference on Visualization, Imaging and Image Processing, IEEE Computer Society, Benidorm, pp. 628-633, 2005.
- [12] Wimmer M. and Scheiblauer C., "Instant Points : Fast Rendering of Unprocessed Point Clouds", Proceedings Symposium on Point-Based Graphics 2006, Eurographics Association, Boston, pp. 129-136, 2006.
- [13] Wand M et al., "Interactive Editing of Large Point Clouds", Proceedings of Symposium on Point-Based Graphics 2007, Eurographics Association, Prague, pp. 37-46, 2007.
- [14] Goswami P et al., "High Quality Interactive Rendering of Massive Point Models using Multi-way kd-Trees", Proceedings of the 18th Pacific Conference on Computer Graphics and Applications 2010, IEEE Computer Society, pp. 93-100, 2010. DOI: <http://dx.doi.org/10.1109/PacificGraphics.2010.20>
- [15] Saxena, M., Finnigan, P. M., Graichen, C. M., Hathaway, A.F., and Parthasarathy, V. N., "Octree-based automatic mesh generation for non-manifold domains", Engineering with Computers, Vol. 11, pp. 1-14, 1995. DOI: <http://dx.doi.org/10.1007/BF01230440>
- [16] Woo, H., Kang, E., Wang, S., and Lee, K. H., "A new segmentation method for point cloud data", International Journal of Machine Tools and Manufacture, Vol. 42, pp. 167-178, 2002. DOI: [http://dx.doi.org/10.1016/S0890-6955\(01\)00120-1](http://dx.doi.org/10.1016/S0890-6955(01)00120-1)
- [17] Wang, M. and Tseng, Y.-H., "Lidar data segmentation and classification based on octree structure", XXth ISPRS Congress, ISPRS, Istanbul, Turkey, 2004.
- [18] Schnabel, R., Wahl, R., and Klein, R., "Efficient RANSAC for point-cloud shape detection", Computer Graphics Forum, Vol. 26, pp. 214-226, 2007. DOI: <http://dx.doi.org/10.1111/j.1467-8659.2007.01016.x>
- [19] Cho, H., Cho, W., Park, J., and Song, N., "3D building modeling using aerial LiDAR data", Korean Journal of Remote Sensing, Vol. 24, pp. 141-152, 2008.
- [20] Marechal, L., "Advances in octree-based allhexahedral mesh generation: handling sharp features", 18th International Meshing Roundtable, Salt Lake City, UT, USA, pp. 65-84, 2009.
- [21] Han, S., Lee, S., Kim, S. P., Kim, C., Heo, J., and Lee, H., "A Comparison of 3D R-tree and octree to index large point clouds from a 3D terrestrial laser scanner", Korean Journal of Geomatics, Vol. 29, No. 1, pp. 531-537, 2011. DOI: <http://dx.doi.org/10.7848/ksgpc.2011.29.1.39>
- [22] Han, S., Kim, S., Jung, J. H., Kim, C., Yu, K., and Heo, J., "Development of a hashing-based data structure for the fast retrieval of 3D terrestrial laser scanned data", Computers & Geosciences, Vol. 39, pp. 1-10, 2012. DOI: <http://dx.doi.org/10.1016/j.cageo.2011.05.005>
- [23] Kim, Y. J., Kim, J. J., "Compression of Normal Vectors using Octree Encoding", Society of CAD/CAM Engineers, Vol.12, No.4 pp. 109-117, 2007.
- [24] Ryu, J. H., "Efficient Octree Encoding for Real Time Transmission of 3D Geometry Data Through Internet", Society of CAD/CAM Engineers, Vol. 7, No. 4, pp. 262-268, 2002.
- [25] Kim, Y. J., Kim, J. J., "Compression of Normal Vectors using Octree Encoding", Society of CAD/CAM Engineers, Vol. 12, No. 2, pp. 109-117, 2007.
- [26] CloudCompare, "Octree Structure", 2015[cited 2015 Feb 20], Available From: <http://www.cloudcompare.org/>. (accessed Mar., 16, 2016)
- [27] Elseberg, J., Borrmann D., "One billion points in the cloud - an octree for efficient processing of 3D laser scans", International Society for Photogrammetry and Remote Sensing, Vol. 76, pp. 76-88, 2013. DOI: <http://dx.doi.org/10.1016/j.isprsjprs.2012.10.004>
- [28] Elseberg, J., Borrmann D., "Efficient Processing of Large 3D Point Clouds", Information, Communication and Automation Technologies (ICAT), 2011 XXIII International Symposium, pp. 1-7, Oct, 2011. DOI: <http://dx.doi.org/10.1145/800031.808576>
- [29] Tamminen, M., "Efficient Octree Conversion by Connectivity Labeling", Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques, 1984.
- [30] Autonomy Intelligent Systeme, Albert-Ludwigs-Universität Freiburg, Available From: http://ais.informatik.uni-freiburg.de/projects/datasets/octo_map/ (accessed 26 Apr, 2016)

차 기 춘(GiChun Cha)

[정회원]



- 2015년 2월 : 성균관대학교 일반대학원 미래도시융합공학과 (공학석사)
- 2015년 3월 ~ 현재 : 성균관대학교 일반대학원 미래도시융합공학과 박사과정

<관심분야>

구조물 안전/유지관리, 구조물 형상관리

이 동 환(Donghwan Lee)

[준회원]



- 2013년 2월 : 성균관대학교 일반대학원 u-City공학과 (공학석사)
- 2013년 3월 ~ 현재 : 성균관대학교 일반대학원 미래도시융합공학과 박사과정

<관심분야>

구조물 안전/유지관리, 건물 에너지 관리

박 승 희(Seunghee Park)

[정회원]



- 2004년 2월 : KAIST 건설 및 환경공학 (공학석사)
- 2008년 2월 : KAIST 건설 및 환경공학 (공학박사)
- 2008년 10월 ~ 2009년 2월 : Virginia Tech 박사 후 연구원
- 2009년 9월 ~ 현재 : 성균관대학교 건설환경공학부 교수

<관심분야>

구조물 안전/유지관리, 구조물 건전성 모니터링