

# 관계검색을 지원하는 간결한 3D 삼각 메시 데이터 구조 A Compact 3D mesh data structure supporting associative search

\*양상욱<sup>1</sup>, #최영<sup>2</sup>, 박진표<sup>3</sup>, 조성욱<sup>2</sup>

\*Sang-Wook Yang<sup>1</sup>, #Young Choi(yychoi@cau.ac.kr)<sup>2</sup>, Jinp-Pyo Park<sup>3</sup>, Seong-Wook Cho<sup>2</sup>

<sup>1</sup> 중앙대학교 미래신기술연구소, <sup>2</sup> 중앙대학교 기계공학부, <sup>3</sup> ㈜비투겐

Key words : Triangle mesh, Compact data structure, Associative search

## 1. 서론

설계부터 생산기술, 대량생산, 유지보수 등에 이르는 전체 제조프로세스에 관여하는 작업자들은 설계 데이터에 대한 검증과 피드백 과정에도 참여하게 된다. 이러한 경우 일반적으로 원래의 설계 데이터가 아닌 경량 가시화 파일을 이용하여 설계 검토가 이루어진다. 현재 널리 사용되고 있는 대표적인 경량 가시화 데이터 형식으로는 Siemens사의 JT 형식[1]과 Dassault의 3D XML 형식[2]을 들 수 있다. JT는 1996년에 처음 소개된 최초의 경량화된 CAD 데이터 형식이며 B-rep 구조의 실제 형상 정보와 모델에 대한 다양한 속성정보를 가지고 있어 대용량 CAD 파일 없이 설계된 모델을 효율적으로 공유할 수 있다. Dassault에서 채택한 3D XML의 경우 Lattice사의 XVL [3]에 기반을 두고 동적 LOD(Level-Of-Detail)을 지원하는 경량화된 형식이다. 또한 Intel을 중심으로 Adobe 등 24개 회사가 참여한 3D 산업포럼(3DIF:3D Industry Forum)에서 제안한 U3D 형식은[4] Microsoft Office나 PDF와 같은 문서에 삽차될 CAD 정보를 포함시켜 쉽고, 빠르게 3D 제품정보를 공유하기 위한 기술을 제공한다. 현재 널리 사용되고 있는 대부분의 경량 가시화 형식들은 기본적으로 메시 형식의 데이터를 기반으로 하고 있는데 메시 데이터 만으로는 치수 측정이나 경계 곡선의 선택 등 DMU(Digital Mock-Up)의 기능을 모두 지원할 수 없기 때문에 경계 에지 정보나 특징 형상 정보 등 부가적인 데이터를 추가하여 이러한 기능들을 지원하고 있다. 즉 경량 가시화 데이터 형식은 경량화와 기능 사이의 절충(trade-off)에 의해 결정된다고 할 수 있다.

본 논문에서는 가시화 및 DMU 목적의 3D 메시지를 표현하기 위한 배열 기반의 간략한 데이터 형식을 제안한다. 제안 데이터 형식은 삼각망 메시의 꼭지점들과 삼각형들의 배열로만 이루어지는 최소한의 데이터로 표현된다. 또한 별도의 데이터 추가 없이 꼭지점 배열의 순서에 공간 검색을 위한 정보를 내포시킴으로써 최적화된 이진 트리와 같은 속도의 검색을 가능하게 한다. 이로써  $n$  개의 꼭지점을 갖는 삼각망에 대해서 최단 거리 검색과 바운딩 박스 검색을  $O(\log n)$ 의 시간 복잡도로 수행할 수 있다. 제안하는 메시 표현 방법은 삼각형 메시지를 에지 기반의 데이터 형식에 비해 더욱 간결하게 표현하면서도 빠른 형상 검색 방법을 제공하기 위한 새로운 데이터 표현 방법이다.

## 2. 경량 메시 데이터 형식

### 2.1 메시 데이터 형식

모델의 가시화 및 렌더링 목적의 폴리곤 메시에 대해서 모델을 데이터 크기와 속도의 측면에서 효율을 높이기 위한 단순화(simplification)나 반복분할(subdivision)을 적용하기 위한 측면의 연구가 많이 수행되었다. 이러한 측면에서 Ketner와 Zürich는 half-edge에 기반한 데이터 구조를 제안하고 CGAL[5]에 적용하였으며[6] Compagna 등은 winged-edge 구조를 개선하여 용도에 따라 속도와 메모리 사용량을 절충할 수 있는 Directed Edges 데이터 구조를 제안하였다[7]. Zorino 등은 반복분할 과정의 절점과 에지의 인접 관계를 quad-tree로 정리하여 분할을 다루기 편리한 데이터 구조를 제안하였다[8]. 이들은 모두 메시의 압축과 확장을

위한 데이터 구조를 지원하기 위한 에지 기반의 데이터 구조를 사용하고 있다. Blandford 등은 2D의 경우 삼각형 당 5 바이트, 3D의 경우 사면체(tetrahedron) 당 7.5 바이트로 표시되는 단체(simplex) 기반의 경량 데이터 구조를 제안한 바 있다[9]. 이 데이터 구조는 에지나 삼각형을 공유하는 절점들의 인덱스가 근처에 있도록 재배열하고 하나의 절점이 그 주위의 절점에 대한 상대 위치의 원형 링크를 가지도록 하는 방법을 이용한다.

기존의 자료구조들은 메시의 재생성과 간소화를 위해서 널리 사용되는 것들이며 대부분이 이러한 오퍼레이션들의 효율을 추구하는 측면에서 연구되었다 할 수 있다.

본 논문에서 제안하는 메시 표현 방법은 2D 메시와 3D 셀(shell)메시를 위한 데이터 구조이며 삼각형 메시지를 간결하게 표현하면서 부가적인 데이터 구조 없이 공간 검색 방법을 제공하기 위한 방법이다. 제안하는 데이터 구조는 원래 2D에서 경계기반의 삼각화 알고리즘을 구현하기 위한 데이터 구조인 CET(Constrained Embedded Triangles) 데이터 구조[10]를 기반으로 검색을 위한 데이터 정렬 방법을 명시함으로써 정의된다.

### 2.2 제안 메시 데이터 구조

$n$  개의 절점과  $m$  개의 삼각형으로 이루어진 2D 삼각형 메시와 3D 삼각형 셀 메시는 다음의 배열  $V$  와  $T$  로 나타낼 수 있다.

$$V = \{v / kd\text{-array of } v_0, v_1, \dots, v_{n-1}\}$$

$$T(V) = \{t\} = \{t_0, t_1, \dots, t_{m-1}\}$$

배열  $V$  는 공간 검색을 위한 분할 정보를 나타내기 위하여 kd-tree[11]의 배열 형태인 순열벡터(permutation vector) [12]로 표현된다. 배열의 순서를 결정하는 방법은 2.3 절에서 설명한다.

각각의 절점  $v$  와 삼각형  $t$  는 다음과 같이 나타낼 수 있다.

$$v = (c, L), \text{ where } c=(s_0, s_1) \text{ or } c=(s_0, s_1, s_2)$$

$$t = (M_0, M_1, M_2, N_0, N_1, N_2)$$

$c$  는 해당 절점의 좌표 값(coordinates)이며 2D의 경우 두 개의 스칼라 값, 3D의 경우 세 개의 스칼라 값으로 표현된다.  $L$  은  $v$  에 연결된 여러 삼각형들 중 임의의 한 삼각형의 인덱스를 나타낸다.  $v$  를 참조하고 있는 삼각형은 동시에 여러 개가 될 수 있지만  $v$  의 측면에서는 최소한 하나의 인접한 삼각형만 알 수 있다면  $v$  를 참조하는 모든 삼각형을 순회할 수 있다.  $M_i$  는 삼각형  $t$  를 구성하는 세 절점의 인덱스를 나타내며 그 절점들은 배열  $V$  의 요소이다.  $M_0$  와  $M_1, M_2$  에 의해 참조되는 점들의 좌표는 이차원 공간에서 반시계 방향(CCW)으로 삼각형을 이룬다.  $N_i$  는 삼각형  $t$  에 인접한 세 삼각형의 인덱스를 나타낸다.  $N_i$  에 의해 참조되는 삼각형은  $t$  와 에지를 공유하는 인접 삼각형이면서  $M_i$  에 의해 참조되는 절점과 연결되지 않는 삼각형이다. 즉, 하나의 삼각형은 세 개의 절점 인덱스와 세 개의 인접 삼각형 인덱스로 이루어지는데,  $M_i$  와  $N_i$  는  $t$  를 중심으로 마주보는 위치에 있다.

Fig.1의 예에서 삼각형  $t_0$  는 절점  $v_0, v_3, v_2$  에 의해 이루어지며 삼각형  $t_1, t_2, t_3$  와 이웃하고 있다. 또한  $v_0$  는  $t_0$  를

중심으로  $t_1$ 의 반대편에 위치하며,  $v_3$ 와  $t_3$ ,  $v_2$ 와  $t_2$ 도 마찬가지로 서로 반대편에 위치한다. 따라서  $t_0=(0, 3, 2, 1, 3, 2)$ 로 표현된다. 앞의 0, 3, 2는 세 절점의 인덱스이며 뒤의 1, 3, 2는 각각 세 절점의 반대편에 위치하는 인접 삼각형의 인덱스를 의미한다.

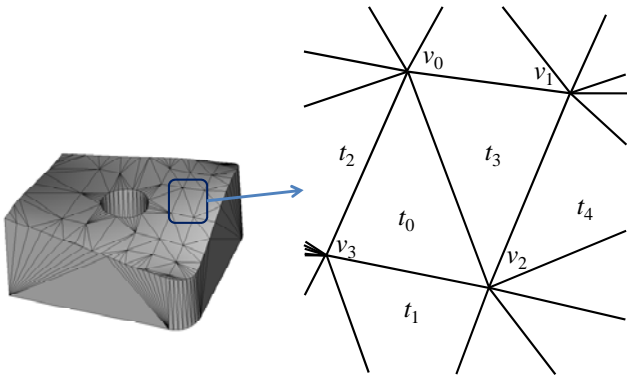


Fig. 1 An example of proposed data structure

$n$  개의 절점과  $m$  개의 삼각형에 대해 절점 좌표를 제외한 토폴로지의 표현을 위한 인덱스의 개수는  $6m+n$  이 되며 2D 공간과 3D 셀 메시에 대해  $m, n \gg 0$  일 때  $m$  개의 삼각형 메시에 대해  $7m$  개의 인덱스로 토폴로지의 표현이 가능하다.

데이터 형식은 매우 간단한 인덱스 삼각형 구조로 표현되지만 삼각형의 인덱스간 규칙을 이용해서 하나의 절점이나 삼각형으로부터 이웃하는 모든 삼각형이나 절점의 정보를 구해낼 수 있다[10,13].

2.3 관계 검색을 위한 절점 배열 순서 정렬

2.2 절에서 설명된 삼각형 정보 중 절점 배열  $V$ 는 배열 요소의 순서에 의해서 공간 분할 정보를 표현하는 순열벡터이다. 순열벡터는 kd-tree의 배열 표현이라고 할 수 있는데, 실제로 이진 검색트리를 위한 데이터 공간을 필요로 하지 않으면서 kd-tree의 최단 거리 검색과 바운딩 박스 검색과 같은 관계 검색을 지원할 수 있다. Fig. 2 (a)는 2D 공간에 분포된 절점들이 최적화된(balanced) kd-tree에 의해 분할된 공간을 보여주며, Fig. 2 (b)는 이 때의 노드의 구성을 보여주고 있다. Fig. 2 (b)와 같은 트리를 직접 구성한다면 각 노드는 좌,우 자손 노드를 위한 레퍼런스를 가져야 하며 구현 목적에 따라 kd-tree의 판별자(discriminator)를 위한 인덱스를 가져야 할 경우도 있고, 빠른 순회를 위해 부모 노드를 위한 인덱스를 가져야 하므로 하나의 노드에 대해 세 개의 인덱스와  $2^k$  bit ( $k$ 는 차원)가 필요하게 된다[12].

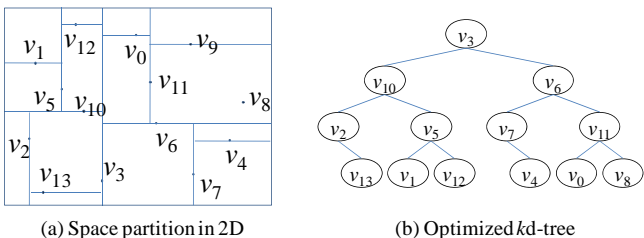


Fig. 2 Space partitioning of an optimized kd-tree

제안하는 데이터구조에서 Fig. 2 (a)와 같은 절점 분포에 대한 절점 배열 순서는  $V=\{v_2, v_{13}, v_{10}, v_1, v_5, v_{12}, v_3, v_7, v_4, v_6, v_0, v_{11}, v_8\}$  이 되어야 하며 이는 Fig. 2(b)의 트리를 순회할 때 중위순회(in-order traversal)하는 순서와 동일하다. 임의 순서의 절점 배열에 대해서 순열벡터로 정렬하는 알고리즘은 최적화된 kd-tree를 구성하는 방법과 같으며  $n$  개의 절점에 대해서  $O(n \log n)$ 의 시간이 소요된다. 정렬된 순열  $V$ 를 중간요소(median)를 중심으로 나누어 보면  $V_{IL}=\{v_2, v_{13}, v_{10}, v_1, v_5, v_{12}\}$ ,  $V_{IR}=\{v_7, v_4, v_6, v_0, v_{11}, v_8\}$ 이며 이는 Fig. 2(b)의

루트 노드에 의해 나누어진 좌측과 우측 서브트리 집합과 동일하다.  $V_{IL}$ 을 중간요소에 의해 나누면  $V_{IL2L}=\{v_2, v_{13}\}$ 와  $V_{IL2R}=\{v_1, v_5, v_{12}\}$ 이며 이들은 Fig. 2(b)의  $v_{10}$ 에 의해서 나누어지는 좌,우측 서브트리 집합과 동일하다. 이렇게 순열벡터는 배열의 순서만으로 인덱스를 반분하며 재귀적으로 kd-tree의 노드에 접근할 수 있으며 배열만을 이용하여 kd-tree의 관계 검색 기능을 제공할 수 있다.

3. 결론

본 논문에서는 간결한 인덱스 삼각형 구조를 가지면서 최단거리 검색 등의 관계 검색을 지원할 수 있는 데이터 구조를 소개하였다. 제안하는 데이터 구조는 2D 삼각화와 3D 셀 메시를 효율적으로 표현하고 검색하기 위한 목적으로 개발되었으며 데이터 변환 없이 빠르게 렌더링될 수 있다. 또한 경량 가시화 데이터 형식에 적용되어 형상의 비교나 충돌 검사 등에 응용될 수 있다.

후기

본 연구는 한국과학재단 우수연구센터 지원과제(No. R11-2007-028-01001-0)와 수중운동체특화연구센터(UVRC)의 지원에 의해서 수행되었으며 이에 감사 드립니다.

참고문헌

1. UGS, "UGS: JT File Format Reference Version 8.1," 2006.
2. K. Versprille, "Dassault Systemes' Strategic Initiative: 3D XML for Sharing Product Information," Collaborative Product Development Associates, 2005.
3. Wakita, A., Yajima, M., Harada, T., Toriya H., and Chiyokura, H., "XVL : A Compact And Qualified 3D Representation With Lattice Mesh and Surface for the Internet", Proceedings of the 50<sup>th</sup> symposium Web3D-VRML, 45-51, 2000.
4. ECMA, Ultimate final draft U3D File Format 3rd Edition, <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-363%203rd%20edition.pdf>
5. CGAL-Computational Geometry Algorithms Library, <http://www.cgal.org/>
6. Kettner, L. and Zürich, E., "Designing a Data Structure for Polyhedral Surfaces", Proceedings of the 14<sup>th</sup> ACM Symposium on Computational Geometry, 146-154, 1998.
7. Compagna, S., Kobbelt, L., and Seidel H.P., "Directed Edges-A Scalable Representation for Triangle Meshes", J. Graph Tools, 3(4), 1-12, 1998.
8. Zorin, D., Schroder, P., DeRose, T., Kobbelt, L., Levin, A., and Sweldens, W., "Subdivision for modeling and animation", SIGGRAPH00 Course Notes, 2000.
9. Blandford, D., Blemloch, G., Cardoze, D., and Kadow, C., "Compact representations of simplicial meshes in two and three dimensions", International Journal of Computational Geometry and Applications 15(1), 3-24, 2005.
10. Yang, S., Choi, Y., and Lee, H., "CAD data visualization on mobile devices using sequential constrained Delaunay triangulation." Computer-Aided Design(2009), doi:10.1016/j.cad.2008.08.005.
11. Bentley, J., "Multidimensional binary search trees used for associative Searching", Communications of ACM, 18, 509-517, 1975.
12. Bentley, J., "Multidimensional Binary Search Tree in Database Applications", IEEE Transactions on Software Engineering, 5(4),333-340, 1979.
13. 양상욱, "들로네 삼각화를 위한 간결한 알고리즘 및 CAD 가시화 적용 연구", 박사학위논문, 중앙대학교, 2008.2.