한국 CAD/CAM 학회 논문집
제 7권 제 4 호 2002년 12월 pp. 262-268

학술논문

인터넷을 통한 3D 형상 데이터의 실시간 전송을 위한 효율적인 Octree 인코딩 방법에 관한 연구

류중현*, 김영우**, 김덕수***

Effcient Octree Encoding for Real-Time Transmission of 3D Geometric Data through Internet

Ryu, J. H.**, Kim, Y. W.** and Kim, D. S.*

ABSTRACT

Octree representation has the advantage of being able to represent complex shapes approximately through the repetition of simple primitive shapes. Due to this reason, octree representation together with VRML(Virtual Reality Modelling Language) is usually used for approximating 3D shapes. Since the data size of octree representation increases rapidly as 3D shape to be represented is more and more complicated, its transmission time also increase. In this paper, provided is the new octree representation and encoding/decoding scheme for real-time transmission through the internet in order to visualize 3D geometric data of large size approximately.

Key words : Octree, Encoding, 3D shape, VRML, Transmission time

1. 서 론

정보화 시대로 접어들면서 정보의 공유에 대한 필요성이 대두되고, 이는 네트워의 발전에 힘입어 실현되고 있다. 특히, 인터넷의 사용이 일상화되고 있는 상황에서 3D형상정보를 인터넷을 통해서 공유해야 하는 필요성이 점점 증가하고 있다.

컴퓨터 그래픽스, CAD/CAM/CAE, 가상현실, 컴퓨터 애니메이션을 포함하는 영화/영상산업, 의료산업, GIS 및 시뮬레이션 등 여러 분야에서 3D형상정보가 사용되고 있으며 수학, 통계학, 계산 물리학, 기상학, 천문학, 지리학등과 같은 과학분야에서도 연구대상 및 실험 데이터들의 가시화에 3D형상정보를 사용하는 것이 점점 보편화 되고 있다.

인터넷을 통한 3D형상정보의 실시간 전송 및 공유가 점차 보편화되고 있으며 주고 받을 3D형상정보의 데이터 사이즈가 점점 증가하고 있지만, 인터넷 사용자의 증가에 따른 network traffic이 커지고 최대 전송속도인 대역폭(bandwidth)은 한정되어있다. 이렇게 인터넷의 제한된 환경에서 3D형상정보를 원활히 공유하기 위해서는 무엇보다도 전송시간과 가시화 시간의 단축이라는 문제를 해결해야만 할 것이다.

이러한 요구에 부응하여 인터넷을 통한 대용량의 3D 형상정보 특히 메쉬(mesh)형태의 곡면정보를 실시간에 전송하고 가시화하기 위한 연구들이 활발히 진행중에 있다[1-3].

본 논문에서는 Octree표현법을 이용해서 대용량의 3D형상정보를 실시간 내에 대략적으로 가시화하기 위한 방법을 제안한다. Octree표현법은 3D장면(scene)을 효율적으로 나타내는 방법으로 인체의 뼈나 장기 같은 복잡한 형상도 단순한 형상의 반복을 통하여 근사적으로 표현이 가능하다는 장점이 있다[4-6].

실시간 내에 대용량의 3D형상정보를 인터넷을 통해 대략적으로 가시화하기 위해 기존의 Octree표현법을 수정하고, 새로운 Octree인코딩(encoding)/디코딩(decoding) 방식을 제안하여 전송할 데이터 사이즈와 전송 시간 및 가시화 시간을 단축하고자 한다.

*삼성 SDS
**Bum Han Strapping Tools
***종신회원, 한양대학교 시스템응용공학부 부교수
- 논문투고일: 2001. 06. 09
- 심사완료일: 2002. 09. 05

초기에 Octree는 부피가 있는 복잡한 형태의 솔리드 모델을 표현하기 위해 제안되었으며[7-9], Samet(1984)는 quadtree와 Octree 그리고 이와 관련된 데이터 구조에 관한 폭 넓은 연구를 하였다[10].

이후에는 기존의 Octree표현법을 수정하여 여러 분야에 응용하려는 연구가 시작되었으며 대표적인 예로는, 다면체 형태의 B-rep모델로부터 Octree모델을 구하는 알고리듬에 관한 연구[11]와 Octree생성시 분할된 육면체와 표현하고자 하는 임의의 형상간의 포함관계를 결정하는 classification procedure 알고리듬의 개선에 대한 연구가 있었다[12].

이 밖에 3D *digital images*의 다양한 Octree표현법에 대한 연구[13]와 3D 형상의 가시화와 관련된 연구들이 있었으며, 이는 주로 volume rendering과 관련된 연구들이었다[14-16]. 또한, Octree를 이용해서 VRML데이터의 사이즈를 줄여가면서 LOD(Level of Detail) 생성을 해 나가는 방법에 대한 연구도 진행되었다[17].

본 논문의 구성은 다음과 같다. 2절에서는 기존의 Octree표현법에 대해서 소개하고 3절에서는 제안되는 Octree표현법을 이용한 Octree인코딩과 디코딩 방법을 소개하며, 전송 데이터 사이즈, 전송시간 및 가시화 시간의 단축에 대해서 논의한다. 4절과 5절에서는 본 논문에서 제안하는 인코딩/디코딩 방법을 이용해서 구현된 시스템을 소개하고 실험결과를 보인 후 마지막으로 6절에서 결론을 맺는다.

2. Octree표현법

Octree표현법은 표현하고자 하는 공간을 여덟 개의 육면체들을 이용해서 재귀적으로 분할하는 방법을 말한다. Octree표현법에 의한 Octree생성과정은 다음과 같다. 우선 표현하고자 하는 3D형상을 완전히 둘러싸는 육면체(Bounding Cube)를 생성한다.

Fig. 1에서처럼 타원체(ellipsoid) 모양의 형상을 Octree로 표현하고자 하면, Fig. 2에서 보는 바와 같이 이를 8개의 육면체(octant)로 분할한다. 번호는 분할순서이며 분할 시 8개의 육면체가 생성된다.

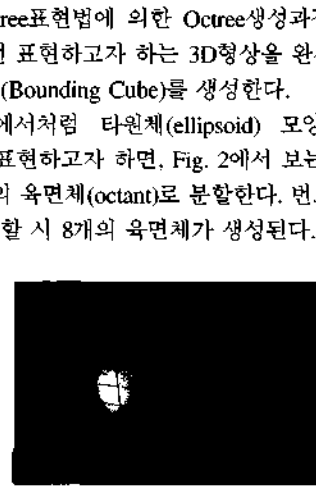


Fig. 1. Ellipsoid and its bounding cube.

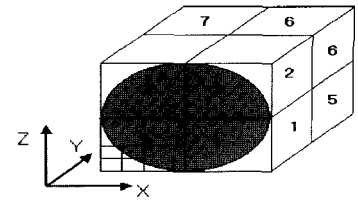


Fig. 2. Octree representation.

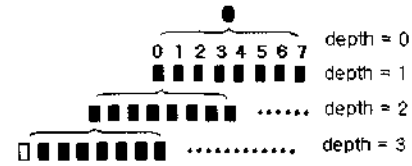


Fig. 3. Data structure of Octree representation.

다음 각각의 육면체가 표현하고자 하는 공간에 완전히 포함되는지, 일부분만 포함이 되는지 혹은 전혀 포함되지않는지에 따라서 각각의 경우를 구별하여 저장한다. Fig. 3은 완전히 포함되면 검은색(black), 전혀 포함되지않으면 흰색(white), 일부분만 포함되어 있다면 회색(gray)으로 표시한 예를 보여주고 있다.

노드가 black인 경우 디스플레이 하고, white인 경우는 디스플레이하지 않으며, 노드가 gray인 경우에는 다시 8개의 육면체로 분할하여 Fig. 3에서 보는 바와 같이 각각이 흰색, 검은색으로 표시될 수 있을 때까지 또는 정해진 분할 값이(depth)까지 분할과정을 반복하며, 마지막 depth의 gray노드는 일반적으로 디스플레이해 준다.

3. 효율적인 Octree 인코딩 방법의 제안

본 논문에서 제시하는 새로운 Octree인코딩 방법에 대해서 논의하기에 앞서 선행 연구들 중 대표적인 linear Octree표현법에 대해서 기술하면 다음과 같다.

3.1 Linear Octree

Octree표현법은 요구되는 기억용량이 크다는 단점을 가지고 있기 때문에, Octree표현법의 데이터를 줄이기 위해서 많은 가능성들이 검토되었으며, 트리 모양의 데이터 구조(explicit tree-structure)대신 포인터가 필요없는 선형의 데이터 구조(linear data structure)가 제안되었다[18-20].

3.1.1 Gargantini의 linear Octree

이 방법은 Octree의 육면체들의 분할순서를 Octree

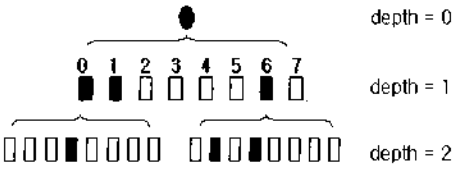


Fig. 4. Octree of depth 3.

의 노드 중에서 root node를 제외한 각각의 node를 찾아가는 주소(path address)로 사용한다, depth가 3일 경우 찾아가는 주소는 0부터 7까지의 숫자로 이루어진 3자리의 일련의 숫자가 된다. 예를 들어, depth가 3인데 그 이전 depth에서 분할을 하지 않아서 3자리를 채우지 못할 경우에는 "end-of-number"를 쓴다. 구성되는 Octree는 'black'인 모든 노드를 위와 같은 방법에 따라서 순서대로 저장한 리스트의 형태가 된다[18,19].

Fig. 4와 같은 Octree를 Gargantini의 linear Octree 방법으로 인코딩하면, **{03, 1X, 61, 63}**과 같이 일련의 리스트로 나타낼 수 있으며, 여기서 **X**는 "end-of-number"를 의미한다.

3.1.2 DF-representation

Linear Octree의 또 다른 형태인 DF-representation은 Octree를 깊이우선(depth-first)으로 탐색하면서 만나는 순서대로 노드의 정보들을 저장한다. 이때 정보를 나타내는 방법으로는 알파벳 "B", "W", 과 "("을 사용하게 된다. "B"는 black 노드를 "W"는 white노드를 "("는 gray노드를 나타내게 된다[20]. Fig. 4와 같은 Octree를 DF-representation의 방식으로 인코딩을 하게 되면,

((WWWBWWWWBWWWW(WBWBWWWWW

과 같이 나타낼 수 있으며, 알파벳이 3가지 경우이기 때문에 각 노드 당 2 bit가 필요하다.

3.2 제안하는 Octree 인코딩 방법

3D형상정보의 대략적인 형태를 빠르게 전송하기 위해서 기존의 Octree표현법을 수정하고, 새로운 인코딩 방법을 제안하고자 한다.

2절에서 언급한 Octree의 정의와 성질에서 알 수 있듯이 Octree표현법은 3D형상의 겉 모습 뿐만 아니라 형상의 내부도 표현하고 있다. 3D 형상의 전송시간과 가시화 시간을 단축하기 위해 형상의 내부를 제외한 겉 모습만을 표현하고자 한다.

이와 같이 내부를 표현하지 않는다면 저장 또는 전송될 데이터의 양이 감소될 뿐만 아니라, 형상을 많은

Fig. 5. Octree representation of outward shapes (1).

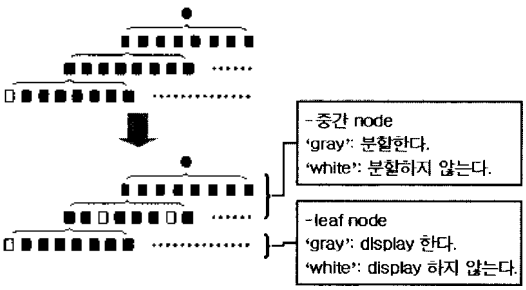


Fig. 6. Octree representation of outward shapes (2).

수의 작은 육면체로 표현하는 Octree표현법에 있어서 디스플레이해야 할 육면체의 수가 줄어들어 최종 가시화 시간이 줄어들게 된다.

Fig. 5과 같이 형상의 겉 모습만을 가시화한다면, 형상의 깊이정보를 표현하던 하나의 정보는 필요가 없어진다. 즉, Fig. 5과 같이 depth가 3인 경우 형상의 내부를 표현하던 'black'노드는 Fig. 6과 같이 전부 'white' 노드로 바뀌게 되고 가시화시간도 줄어들게 된다.

형상의 겉 모습만을 표현하기 때문에 'black', 'white' 'gray'로 표현된 노드들 중 형상의 내부를 표현하는 'black'은 디스플레이 하지 않는 'white'로 그리고 형상의 겉 모습을 표현하는 'black'은 디스플레이 하는 'gray'로 바꿔서 'white'와 'gray'의 2가지 정보만으로 Octree의 각 노드를 표현하는 것이 가능하게 된다.

따라서, 중간 노드에 gray가 할당된 경우에는 분할을 하며, 미리 지정한 depth 혹은 (최종depth)의 노드에서 gray가 할당되는 경우에는 디스플레이 하게 되면 각각의 노드를 1과 0의 1bit로 표현하는 것이 가능하여 종전의 2bit를 사용할 경우보다 데이터 양이 줄어든다.

3.3 각 Octree표현법에 따른 전송데이터 사이즈 비교

기존의 Octree표현법에서 depth가 n일 경우 각각의 depth에서 발생 가능한 최대 노드 개수는

$$2^n \times 2^n \times 2^n = 8^n$$

으로 나타낼 수가 있다. 따라서, 각각의 depth에서 가능한 최대 노드의 개수를 합산한 가능한 전체 노드의 개수는

$$\sum_{i=0}^{n} 8^i = \frac{(8^{n+1}-1)}{(8-1)}$$

이다.

즉 이것이 전송될 가능한 최대 노드개수가 된다. 각각의 노드에 'black', 'white', 'gray'등의 정보들을 2bit로 정한다면 전송될 가능한 최대 데이터 사이즈는 가능한 최대 노드개수에 2bit를 곱한 것이 되지만 본 논문에서 제안하는 방식으로 인코딩하게 된다면

$$\frac{(8^{n+1}-1)}{(8-1)} \quad * \quad 1bit$$

가 된다.

3.4 Octree데이터의 인코딩과 디코딩

먼저, 전송될 정보들이 인코딩되는 방식에 대해서 살펴보면 다음과 같다. 각각 'white'와 'gray'에 해당되는 0과 1의 1bit로 표현된 정보들은 하나의 배열에 저장되며, 인코딩 방식으로는 기존의 전통적인 깊이우선(depth-first)방식과 너비우선(breadth-first)방식 두 가지를 생각해 볼 수 있다.

Fig. 7과 Fig. 8에서 각각 위의 그림은 인코딩될 Octree를 나타내는 것이고, 아래의 그림의 배열은 인코딩되는 순서를 의미한다. 여기서 배열의 인덱스는 Octree의 각 노드번호를 의미하는 것이고, 0과 1은 인코딩된 정보이다.

Octree의 노드번호와 배열의 인덱스들을 비교해보면 확인할 수 있듯이, Octree에 표시된 화살표방향으로 깊이우선방식(Fig. 7) 또는 너비우선방식(Fig. 8)의 순서대로 인코딩할 수 있다. 이 순서대로 저장되어 전송된 데이터를 디코딩 후 가시화하게 된다.

깊이우선과 너비우선방식 모두 전송되는 데이터 사

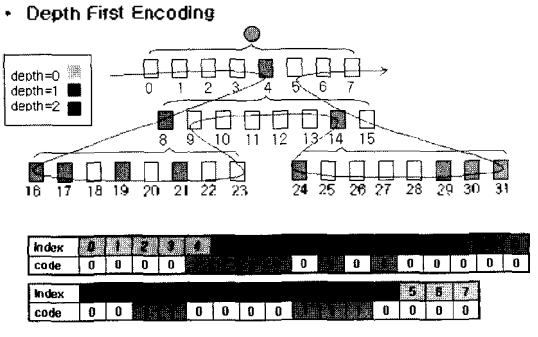


Fig. 7. Depth-first encoding.

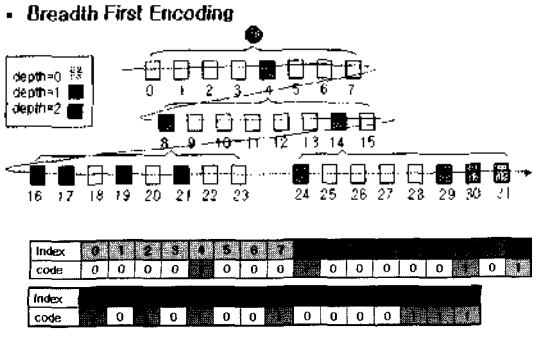


Fig. 8. Breadth-first encoding.

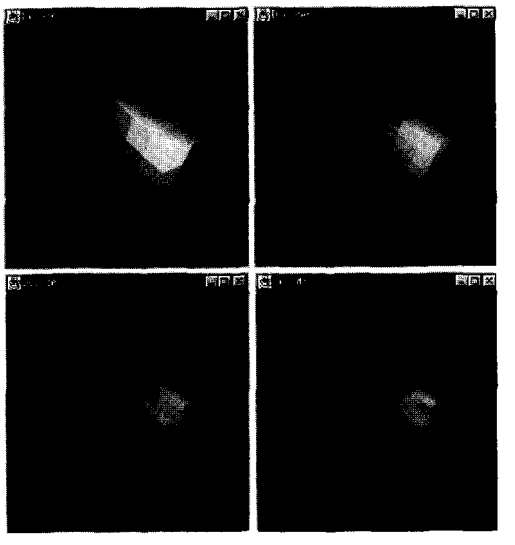


Fig. 9. Breadth-first decoding/visualization.

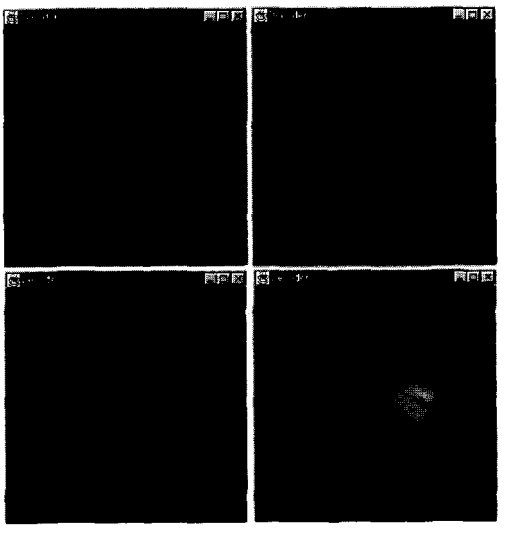


Fig. 10. Depth-first decoding/ visualization.

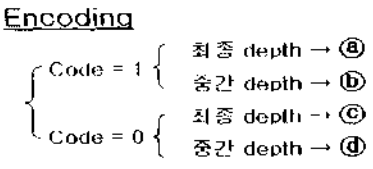


Fig. 11. Encoding/Decoding scheme.

이즈는 같지만 점진적인 전송(progressive transmission)을 구현시 디코딩후 가시화되는 모습에 약간의 차이가 있다. Fig. 9과 Fig. 10은 Fig. 13(a)의 cactus 모델을 각각 너비우선방식과 깊이우선방식으로 전송했을 경우의 모습이다. 너비우선방식일 경우에는 Fig. 9에서와 같이 대략적인 모습에서 점점 자세한 모습으로의 가시화가 가능하며, 깊이우선방식의 경우에는 Fig. 10에서와 같이 국부적인 자세한 모습에서 점점 원래의 모습으로 가시화해 나가는 것이 가능하다.

제안하는 방식으로 인코딩한 Octree정보를 디코딩하고 가시화하는 과정은 다음과 같다. 인코딩되어 전송된 Octree의 최종 depth에 따라서 배열의 각 code를 확인하여 노드의 분할유무와 디스플레이의 유무에 따라서 각 노드의 경우를 4가지로 나누게 되며, 이 4가지 경우에 따라서 가시화를 해 나간다.

예를 들어 Fig. 11에서 중간 depth에 code 1이 부여될 경우(ⓑ)에는 디코딩할 때 분할을 하고 디스플레이하지 않는다.

4. 시스템 구현

제안하는 방법에 따라서 만들어진 프로토타입 시스템에 사용된 언어로는 플랫폼 간에 독립적이고 네트웍프로그래밍시 여러가지 장점을 지니고 있는 Java를 사용하였으며, 그래픽 라이브러리로는 Java3D를 사용하였다. Octree 인코딩을 하게 될 3D형상 데이터로는 VRML(Virtual Reality Modeling Language)파일형식의 모델을 채택하였다.

대략적인 시스템을 살펴보면 Fig. 12과 같이 서버에서 오프라인상의 작업으로 미리 VRML파일을 제안하는 Octree표현법에 따라서 깊이우선 방식 또는 너비우선 방식으로 인코딩하게 되며, 클라이언트의 요청을 기다리게 된다. 클라이언트가 서버측에 요청을 보내면 온

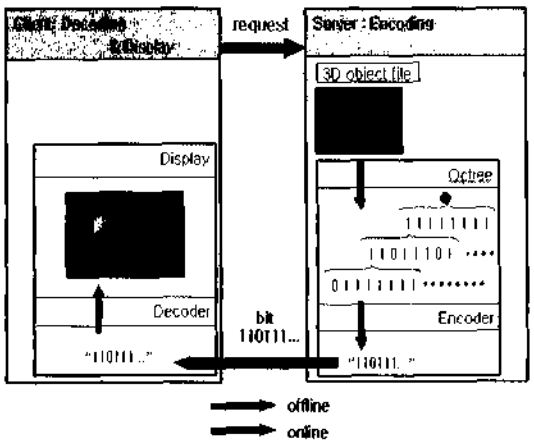
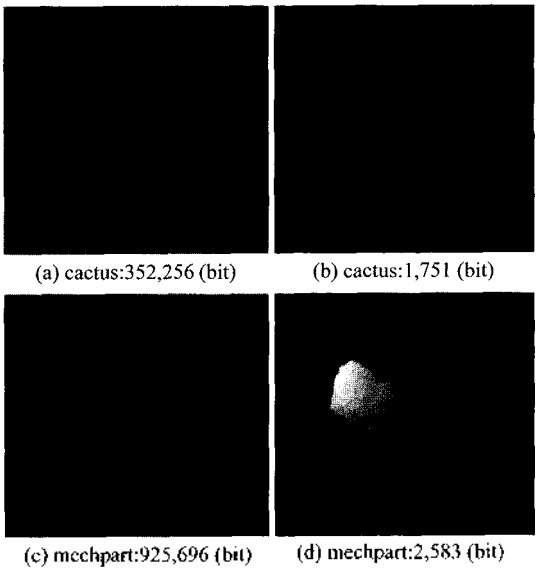


Fig. 12. Architecture of a prototype system.

라인상의 작업으로 서버는 인코딩된 데이터를 클라이언트에 전송하게 되고 이것을 클라이언트쪽에서는 디코딩후 디스플레이를 하게 된다.

5. 실험결과

본 논문의 실험은 몇 가지 VRML포맷의 모델을 실험 데이터로 하여 기존의 Octree표현법과 개선된 Octree 표현법으로 가시화를 해보고, 전송되는 Octree의 노드 개수를 계산하여 각각의 데이터 사이즈를 측정을 한 뒤 실제 VRML파일과의 데이터 사이즈를 비교해 보았다.



(a) cactus:352,256 (bit) (b) cactus:1,751 (bit)

(c) mechpart:925,696 (bit) (d) mechpart:2,583 (bit)

Fig. 13. VRML format vs. proposed Octree representation format.

Table 1. The comparison of data size of each representation

(a) cactus.wrl				
	VRML 파일(bit)	Octree(bit)	개선된 Octree(bit)	개선된 Octree의 압축율(%)
Depth: 3	352,256	398	199	50
Depth: 4		1,134	567	50
Depth: 5		3,502	1,751	50
Depth: 6		11,074	5,537	50
(b) mechpart.wrl				
	VRML 파일(bit)	Octree(bit)	개선된 Octree (bit)	개선된 Octree의 압축율(%)
Depth: 3	925,696	398	199	50
Depth: 4		1,358	679	50
Depth: 5		5,166	2,583	50
Depth: 6		13,874	6,937	50

먼저 VRML 포맷과 제안하는 Octree표현법에 의한 각각의 데이터 사이즈와 가시화 결과를 비교하면 Fig. 13와 같다. Fig. 13의 (a)와 (c)는 실제 VRML모델과 그 데이터 사이즈이며, (b), (d)는 depth를 5로 하여 개선된 Octree표현법으로 가시화한 결과이다. 원래 모델의 데이터에 비해 사이즈가 상당히 줄어 들었음을 알 수가 있으며 원래 모델의 형태를 대략적으로 근사하고 있음을 확인할 수 있다.

Table 1의 두 표는 두 개의 VRML모델 데이터에 대해서 실제 VRML파일 사이즈와 Octree표현법에 의한 데이터 사이즈 그리고 개선된 Octree표현법에 따른 데이터 사이즈를 비교한 것이며, 개선된 Octree의 압축률을 보여주고 있다. 원래의 VRML데이터과 비교해서는 약 2% 내의 매우 적은 사이즈의 데이터로 표현이 가능하며, 표에서 알 수 있는 것과 같이 기존의 Octree 표현법의 50% 정도의 데이터로 원래모델의 대략적인 가시화가 가능하다.

6. 결 론

본 논문에서는 인터넷을 통해 빠른 시간 내에 3D 형상 데이터를 전송하여 가시화하기 위해서 Octree표현법을 도입하였다. 3D형상을 Octree를 이용해서 표현하면 단순한 형상의 반복으로 원래 모양을 근사하는 형태가 가능하지만, 표현하고자 하는 형상이 복잡할 경우 세밀하게 모두 표현하고자 한다면 데이터 사이즈가 매우 커진다.

따라서, 본 논문에서는 기존의 Octree표현법을 수정하여 복잡한 형상을 대략적으로 표현하는데 유리하도록 새로운 Octree인코딩 방식을 제안하였다. 제안하는 방법을 이용하면 데이터 전송시 원래의 3D형상모델 데이터를 보다 적은 데이터로 표현이 가능하므로 전송시간과 가시화 시간이 단축된다. 따라서, 제안하는 방식을 이용하면 원래의 형상을 정확하지는 않지만 대략적으로 빠른 시간내에 가시화는 하는 것이 가능하며, 점진적인 전송 및 가시화도 가능하게 된다.

Octree표현법이 실제 VRML파일의 데이터 사이즈를 상당히 줄이고 있다는 것은 수치적으로 알 수 있지만, 실제 모델을 얼마나 비슷하게 가시화하고 있는가에 대한 구체적인 측정기준이 없으므로, 이에 대한 추후 연구가 필요할 것이다.

감사의 글

본 연구는 한양대학교 세라믹공정연구센터의 지원으로 수행되었습니다.

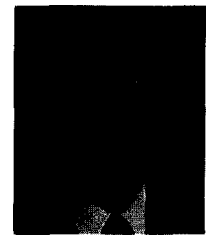
참고문헌

1. Chandrajit L. Bajaj, Valerio Pascucci and Guozhong Zhuang, "Progressive compression and transmission of arbitrary triangular meshes," *Proceedings of the conference on Visualization '99: Celebrating ten years*, pp. 307-316, 1999.
2. Pierre Alliez and Mathieu Desbrun, "Progressive compression for lossless transmission of triangle meshes," *Proceedings of the 2001 conference on Computer Graphics*, pp. 195-202, 2001.
3. 김덕수, 정재열, 김 현, "삼차원 메쉬 모델의 압축 및 점진적 전송을 위한 가수부 분할 기법," 한국 CAD/CAM학회 논문집, 제7권, 제2호, pp. 81-88, 2002년.
4. H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison Wesley, Reading, MA, 1990.

5. Kunwoo Lee, *Principles of CAD/CAM/CAE Systems*, Addison Wesley, 1999.
6. Donald Meagher, "Geometric Modeling Using Octree Encoding," *Computer Graphics and Image Processing*, Vol. 19, pp. 129-147, 1982.
7. G.M. Hunter, *Efficient Computation and Data Structures for Graphics PhD thesis*, Department of Electrical Engineering and Computer Science, Princeton University, Princeton, N. J., 1978.
8. D. J. Meagher "Octree Encoding: A New Technique for the Representation, Manipulation, and Display of Arbitrary Three-Dimensional Objects by Computer," *Technical Report IPL-TR-80, 111*, Image Processing Laboratory, Rensselaer Polytechnic Institute, Troy, New York, Oct. 1980.
9. C. L. Jackins and S. L. Tanimoto, "Octrees and Their Use in the Representation of Three-Dimensional Objects," *Computer Graphics and Image Processing*, pp. 249-270, 1980.
10. H. Samet, "The quadtree and related hierarchical data structure," *ACM Computing Surveys*, Vol. 16, No. 2, pp. 187-260, 1984.
11. R. Krishnan, A. Das and B. Gurumoorthy, "Octree Encoding B-rep based Objects," *Computers & Graphics*, Vol. 20, No. 1, pp. 107-114, 1996.
12. Yoshifumi Kitamura and Fumio Kishino, "A Parallel Algorithm for Octree Generation from Polyhedral Shape Representaion," *IEEE Proceedings of ICPR*, pp. 303-309, 1996.
13. Sargurn. Srihari, "Representation of Three-Dimensional Digital Images," *ACM Computing Surveys*, Vol. 13, No. 4, pp. 399-424, 1981.
14. Jane Wilhelms and Allen Van Gelder, "Octrees for Faster Isosurface Generation," *ACM Transactions on Graphics*, Vol. 11, No. 3, pp. 201-227, 1992.
15. R. Westermann and L. Kobbelt, and T. Ertl "Real-time exploration of regular volume data by adaptive reconstruction of isosurfaces," *The Visual Computer*, Vol. 15, pp. 100-111, 1999.
16. Shiaofen Fang, Su Huang, Rajagopalan Srinivasan and Raghu Raghavan, "Deformable Volume Rendering by 3D Texture Mapping and Octree Encoding," *Visualization '96 Proceedings*, pp. 73-80, 1996.
17. Dieter Schmalstieg, "Lodestar: an octree-based level of detail generator for VRML," *Proceedings of the Second Symposium on Virtual Reality Modeling Language*, pp. 125-132, 1997.
18. Martti Mantyla, *An Introduction to Solid Modeling*, Computer Science Press, 1988.
19. I. Gargantini, "Linear octrees for fast processing of three-dimensional objects," *Computer Graphics and Image Processing*, Vol. 20, pp. 365-374, 1982.
20. E. Kawaguchi and T. Endo, "On a method of binary picture representation and its application to picture compression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 2, No. 1, pp. 27-35, 1980.

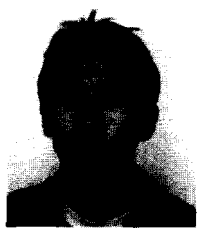
류 중 현

1997년 한양대학교 산업공학과 학사
1999년 한양대학교 산업공학과 석사
2002년 한양대학교 산업공학과 박사
2002년~현재 삼성SDS
관심분야: 기하모델링, 계산기하학, 3차원 기하 모델압축 및 인터넷 응용



김 덕 수

1982년 한양대학교 산업공학과 학사
1985년 New Jersey Institute of Technology 산업공학과 석사
1990년 The University of Michigan 산업공학과 박사
1989년~1991년 Schlumberger Technology CAD/CAM Co. Senior Software Engineer
1991년~1995년 삼성종합기술원 선임연구원
1995년~현재 한양대학교 산업공학과 부교수
관심분야: CAD/CAM, 계산기하학, 기하모델압축



김 영 우

1998년 한양대학교 산업공학과 학사
2001년 한양대학교 산업공학과 석사
2001년~현재 Bum Han Strapping Tools
관심분야: 기하모델링, NC 공구경로생성, 3차원 컴퓨터그래픽스응용