

# Java언어를 이용한 MPEG-4기반 3차원 플레이어의 구현에 관한 연구

준회원 박 영 경<sup>\*</sup>, 정회원 김 용 호<sup>\*</sup>, 정 종 진<sup>\*\*</sup>, 김 중 규<sup>\*</sup>, 안 상 우<sup>\*\*\*</sup>,  
최 진 수<sup>\*\*\*</sup>, 김 진 웅<sup>\*\*\*</sup>, 안 치 득<sup>\*\*\*\*</sup>

## A Study on Implementation of 3D Player based on MPEG-4 Using Java Language

Young-Kyung Park\* Associate Member, Yong-Ho Kim\*, Jong-Jin Jung\*\*, Joong-Kyu Kim\*, Sang-Woo Ahn\*\*\*, Jin-Soo Choi\*\*\*, Jin-Woong Kim\*\*\*, Chie-Teuk Ahn\*\*\*\* Regular Members

### 요 약

MPEG-4 3D mesh coding(3DMC)은 높은 압축률을 가지므로 3차원 정보를 저장, 전송하는 데 요구되는 큰 대역폭의 문제를 해결할 수 있다. 그리고 방송이나 인터넷을 통한 3차원 정보 서비스의 실현을 위해서는 3차원 정보와 함께 정보를 출력할 수 있는 플레이어를 동시에 전송하는 것이 요구된다. 따라서 본 논문에서는 java언어를 이용하여 MPEG-4 기반의 3차원 플레이어를 구현하였다. 다른 언어와 차별화된 java의 특징인 scalability, universality는 O/S나 플랫폼 성격이 다를지라도 플레이어가 동작할 수 있도록 해준다. 플레이어는 MPEG-4 3D mesh decoder의 기능을 가지며 rendering을 위해 java 3D API를 이용하였다. 또한 이동, 회전 등 콘텐츠의 조작을 위한 사용자 인터페이스를 제공하며, 서버로부터 콘텐츠 파일을 전송받는 네트워크 기능이 구현되었다. 플레이어를 이용하여 MPEG-4 3DMC에 의해 생성된 다양한 콘텐츠를 출력하고 성능을 테스트하였다.

Key Words : MPEG-4, 3D mesh decoding, java, 3D player, transformation

### ABSTRACT

With MPEG-4 3D mesh coding(3DMC), the problem of the need of a wide bandwidth can be solved to store and transmit 3D information because of its high compression rate. And to realize the 3D information service with broadcasting or internet, one needs to transmit not only the 3D contents but also the 3D player. Therefore, in this paper we implement a 3D player based on MPEG-4 using a java language. A well-known java language employed in this paper provides the player with a wider range of applications, for example, when the O/S or the platform are different, due to its properties of scalability and universality. The implemented player which has functions (translation, rotation, etc) that can manipulate contents decodes the 3D contents and displays them. In addition, the player has a network function that receives a 3D content from the server. This paper explains the architecture and characteristics of the player and shows its simulation results.

\*성균관대학교 정보통신공학부 디지털신호처리연구실(multipym@ece.skku.ac.kr, solson@ece.skku.ac.kr, jkkim@yurim.skku.ac.kr)

\*\*전자부품연구원, 인터넷미디어연구센터

\*\*\*한국전자통신연구원 방송미디어연구부

\*\*\*\*한국전자통신연구원 전파방송연구소

논문번호:030136-0326, 접수일자:2002년 03월 26일

※본 논문은 한국전자통신연구원(ETRI)의 "MPEG-4 기반 3차원 콘텐츠 및 플레이어 개발"의 수행 일부로서 얻어진 결과임을 밝히고 지원에 감사드립니다.

## I. 서 론

현대 사회에서 3차원 모델링은 방송, 게임, 애니메이션 등의 여러 분야에서 광범위하게 사용되고 있다. 이는 3차원 모델이 사용자가 보다 쉽고 입체적으로 정보를 접할 수 있다는 장점을 가지고 있기 때문이다. 여러 가지의 3차원 모델링 방법이 있지만 그 중 3D mesh는 가장 많이 이용됨과 동시에 하드웨어나 소프트웨어로 3차원 모델을 표현할 수 있는 가장 저렴한 방법이다[1]. 그렇지만 3D mesh를 압축 없이 단순히 IndexedFaceSet으로 표현할 경우 전송과 저장에 요구되는 현저히 큰 대역폭은 방송이나 인터넷상에서 응용되는데 제약조건이 된다[2]. MPEG-4의 3D mesh coding(3DMC)을 이용하면 IndexedFaceSet으로 표현된 3D mesh 모델에 대해 약 50:1의 높은 압축율을 제공함으로써 요구되는 대역폭을 감소시킬 수 있다[2][3].

본 논문에서는 MPEG-4 3DMC에 의해 생성된 3차원 콘텐츠를 decoding 후 rendering할 수 있는 MPEG-4기반의 3차원 플레이어를 구현했다. 플레이어는 MPEG-4규격을 바탕으로 하여 설계되었으며 모두 java언어를 이용하여 구현되었다. 구현언어인 java는 다른 언어와 차별화 된 특징인 scalability와 universality를 가진다. Scalability는 java로 제작한 프로그램은 기업의 웹 시스템에서부터 작은 핸드폰에 이르기까지 어디에든 사용할 수 있다는 것이다. 즉, java를 이용하면 어느 곳에나 적용 가능한 프로그램 개발할 수 있다. 또한 universality라는 것은 java로 개발된 프로그램은 Windows, Linux, Macintosh 등의 O/S에 종속되지 않는다는 것이다 [4]. 이 두 가지 특징은 플레이어에게 보다 폭 넓은 응용범위를 제공한다. 비록 O/S가 다르거나 플랫폼 성격이 다를지라도 플레이어가 동작할 수 있기 때문에 방송 서비스 혹은 인터넷 전자상거래 등에서 단말기가 3차원 플레이어를 가지고 있지 않을 경우 플레이어를 전송해 주지만 하면 구동될 수 있는 것이다.

플레이어는 3차원 모델을 rendering하기 위해 네트워크 환경에 기반을 두고 설계된 java 3D API를 이용하였다[5]. 그리고 rendering 후 버튼이나 마우스를 통해 모델의 회전이나 이동등과 같은 조작이 가능함은 물론 콘텐츠의 목록을 전송받아 보여 줄 수 있는 기능을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 플레이

어의 구조를 설명했고, 3장에서는 구현된 주요 클래스들의 구성을 설명하였다. 4장에서는 플레이어의 인터페이스 및 기능을 설명하고 다양한 콘텐츠의 렌더링 결과를 보였으며 5장에서 결론을 맺었다.

## II. 플레이어의 구조

플레이어의 구조는 크게 콘텐츠를 입력받아 3D mesh정보를 출력하는 decoder와 decoding된 정보를 이용하여 3차원 모델을 화면에 출력하는 rendering 처리부로 나누어진다.

### 1. Decoder

플레이어는 MPEG-4 3d mesh decoder[2]의 기능을 가지며 decoder는 그림 1에서와 같이 주로 4가지 모듈로 구성된다. Arithmetic decoder는 입력 스트림을 읽어서, 비트 스트림으로 출력을 해준다. Vertex graph decoder는 bounding loop look-up table을 만들어 주는 부분이다. 여기에서 has\_stitches 필드가 on되어 있으면 stitches가 decoding되게 된다. Triangle tree decoder는 각 run의 길이와 triangle tree에서 각 sub-tree의 크기를 decoding해 준다. 그리고 마지막으로 triangle data decoder에서는 triangle data 스트림을 출력해준다 [2]. Triangle data 스트림은 geometry 및 각 삼각형과 관련된 property를 포함하며 3차원 모델은 이 정보를 이용하여 rendering된다.

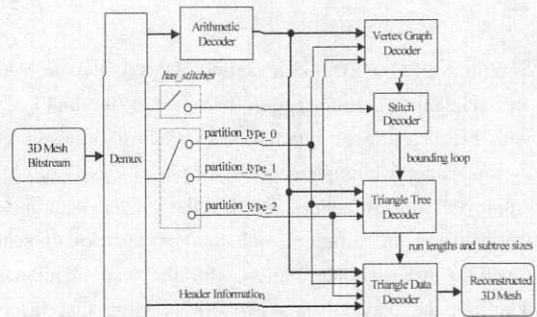


그림 1. MPEG-4 3d mesh decoder의 구조[2]

#### 1.1 Vertex graph decoder

MPEG-4 기반의 3D mesh coding에서 vertex들의 좌표와 connectivity는 각기 다르게 coding되는데 이때 connectivity에 대한 encoding을 위해 vertex graph가 적용된다.

Vertex graph는 vertex graph edge와 vertex들로

구성되며 vertex graph edge는 original 3d mesh의 graph를 spanning하는 edge이다. Boundary가 없는 simple 3d mesh의 경우 vertex graph는 vertex tree를 형성한다(그림 2). 만일 boundary를 가지는 non-simple 3d mesh의 경우에 vertex graph는 loop를 가지게 된다. 그림 2와 같이 vertex graph는 run들의 sequence로 이루어진다. 그리고 이들 run에 대한 정보만으로 충분히 vertex graph의 connectivity를 표현할 수 있다.

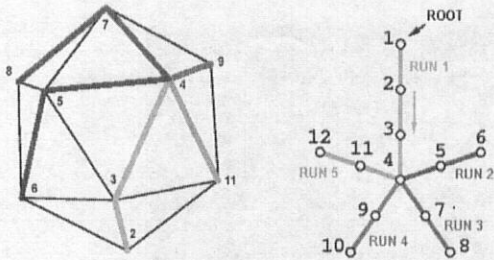


그림 2. Simple mesh의 vertex graph[2]

3D mesh encoder는 vertex graph에 대해 vertex run의 sequence를 encoding한다. 각각의 run은 run record field를 가지고 있고 이 field들은 각 run에 대한 data(run의 길이, loop의 유무 등)를 가진다. 그러므로 vertex graph decoder는 비트 스트림을 읽어 들여 vertex run의 sequence를 재구성하고 bounding loop look-up table을 생성한다. 이 때 has\_stitches필드가 on이면 stitch가 decoding된다[2].

1.2 Stitch decoder

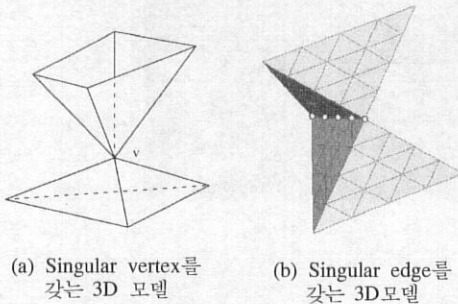


그림 3. Simple 3D mesh에 대한 triangle tree[2]

3D mesh model의 대부분이 다각형의 수가 많아짐에 따라 topological singularity를 갖는 non-manifold model이 된다(그림 3). 그러나 이런

특성은 3차원 모델에 대한 simplification, compression, smoothing작업을 어렵게 한다[6]. 그러므로 encoding시 non-manifold 3D모델의 경우 cutting과 stitching을 통해 manifold 3d model로 변환되게 된다. Stitch decoder는 stitch를 decoding함으로써 vertex graph상의 vertex들의 관계를 정의한 vertex cluster를 재구성한다[2].

1.3 Triangle tree decoder

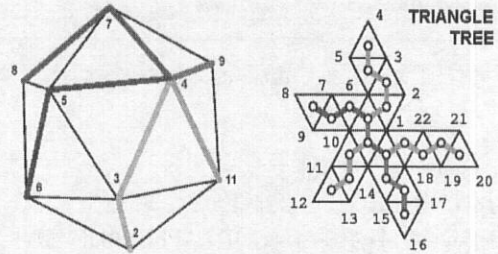


그림 4. Simple 3D mesh에 대한 triangle tree[2]

3D mesh를 vertex graph edge로 잘라냄으로써 나온 결과를 simple polygon이라 한다. 3D triangular mesh의 connectivity encoding과정에서 vertex graph는 simple polygon의 boundary에 대한 정보만을 encoding할 뿐이다. 이 simple polygon을 이루고 있는 triangle간의 connectivity를 encoding하기 위해 triangle tree가 적용된다(그림 4). Vertex graph decoder가 vertex run의 sequence를 decoding한다면 triangle tree decoder에서는 triangle tree의 traversal 순서에 따라 triangle run의 정보를 decoding한다[2]. 그림 5는 triangle tree decoder의 블록도를 보여준다.



그림 5. Triangle tree decoder의 블록도

1.4 Triangle data decoder

Vertex graph decoder, triangle tree decoder의 출력은 최초에 decoding된 property header 정보와 함께 triangle data decoder의 입력이 된다. Triangle data decoder는 입력받은 값을 적절히 처리한 후 최종적으로 각 triangle에 대한 정보를 출력한다. 이것은 triangle을 구성하는 vertex의 좌표 및 각 triangle에 대한 property정보이다[2]. 결국 이 정보

들은 3D mesh를 재구성하는데 필요한 최종 결과가 된다.

Triangle data decoder의 decoding 과정은 header decoding, root triangle decoding, non-root triangle decoding의 세 단계로 나뉜다. 그림 6은 Triangle data decoder의 블록도를 보여준다.

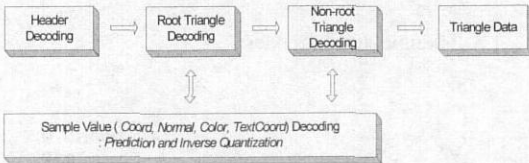


그림 6. Triangle data decoder의 블록도

### 2. Rendering 처리부

앞서 언급한대로 플레이어는 3차원 모델을 rendering 하기 위해 java 3D API를 이용하였다. Java 3D API는 객체 기반의 API이며 네트워크 환경에 기반을 두고 설계되었다는 점에서 플레이어가 방송이나 인터넷 환경에서 응용되기 가장 적합하도록 만들어 준다. java 3D API는 장면 그래프(scene graph)구조를 가진다. 장면 그래프 구조는 3차원 오브젝트를 화면에 그릴 때 그 속성을 계층화된 장면 그래프를 따라 가며 그려주게 된다[5]. 장면 그래프는 다양한 노드들로 구성될 수 있다. 본 논문의 플레이어가 rendering을 위해 취하고 있는 장면 그래프 구조가 그림 7에 나와 있다[7].

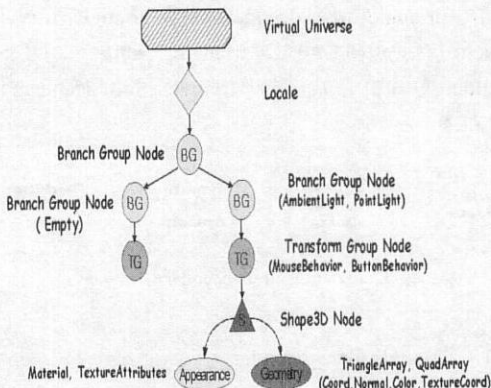


그림 7. 3차원 콘텐츠 rendering을 위한 장면 그래프

### III. 플레이어 인터페이스

기본적인 인터페이스의 구성은 프레임을 크게 두 개로 나누어 우측에 3차원 모델에 대한 콘트롤 패

널들을 위치시켰고 좌측에 3차원 오브젝트를 rendering하도록 했다. 그리고 우측 프레임의 하단에는 사용자가 볼 수 있는 3차원 콘텐츠의 리스트를 출력할 수 있도록 해 사용자가 리스트를 보고 원하는 콘텐츠의 이름을 입력할 수 있도록 하였다. 플레이어의 인터페이스와 관련된 기능들을 살펴보면 다음과 같다.

#### 1. 콘텐츠 파일 수신 및 입력

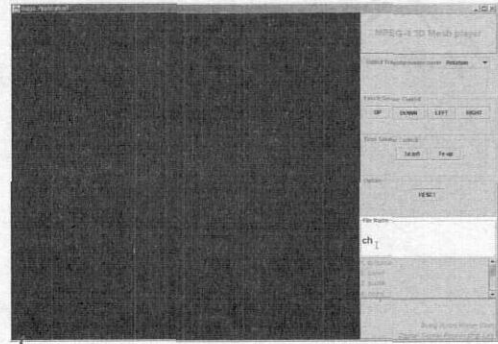


그림 8. File Name 창에 콘텐츠 파일이름 입력하기

플레이어는 직접 제작된 서버와의 통신을 통해 콘텐츠 파일의 리스트 및 콘텐츠 파일을 수신할 수 있도록 했다. 이것은 플레이어가 가지고 있는 네트워크 기능으로 플레이어는 다음과 같이 동작한다.

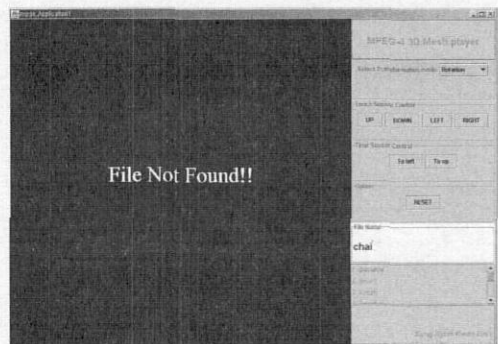


그림 9. 리스트에 없는 콘텐츠 파일 입력 시 에러 출력

플레이어는 실행됨과 동시에 서버로부터 업데이트된 최신의 리스트 파일을 다운 받는다. 그러므로 사용자는 플레이어를 실행하자마자 이용 가능한 콘텐츠 파일의 목록을 볼 수 있다. 사용자가 콘텐츠 파일 이름을 입력했을 시 먼저 콘텐츠를 파일을 다운

받는 디렉토리를 검색해 요청된 콘텐츠가 이미 다운로드 받은 적이 있는 지를 알아본다. 만일 이 때 입력한 파일이 이 디렉토리에 존재한다면 이 파일을 열게 되고 그렇지 않다면 서버에게 콘텐츠 파일을 요청해 콘텐츠 파일을 해당 디렉토리로 다운로드 실행한다. 하지만 이 때 서버에도 존재하지 않는 콘텐츠 파일이 요청되었다면 그림 9와 같이 "FILE NOT FOUND"라는 ERROR 메시지를 보여준다.

2. 사용자의 콘텐츠 조작

사용자는 버튼을 이용하거나 직접 마우스를 이용해 콘텐츠를 조작할 수 있다. 그림 10과 같이 Transformation mode버튼을 눌러 회전, 이동, 확대/축소 기능을 수행할 수 있다.

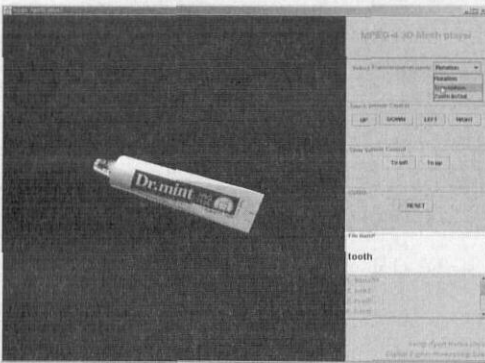


그림 10. Transformation mode 선택

그림 11, 12, 13은 선택된 transformation mode에 대해서 마우스 커서가 가리키는 touch sensor control버튼에 대한 동작을 보여준다.

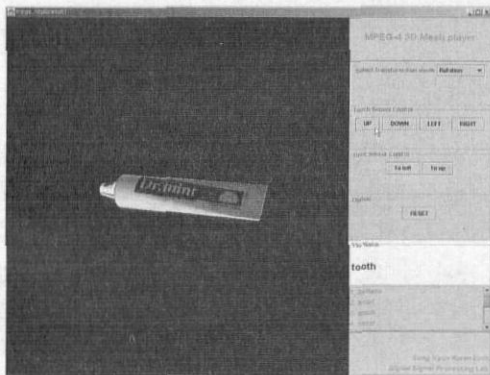


그림 11. 콘텐츠를 위 방향으로 회전

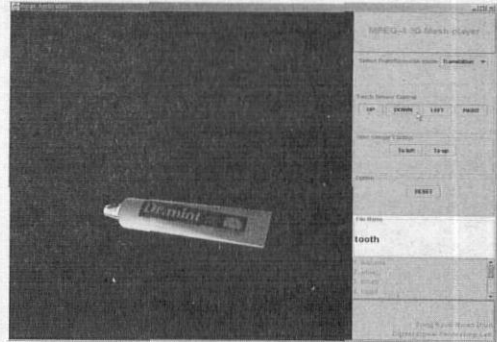


그림 12. 콘텐츠를 아래 방향으로 이동



그림 13. 콘텐츠를 확대

플레이어는 touch sensor control외에도 time sensor control을 지원한다. time sensor control의 버튼을 선택했을 경우 콘텐츠는 해당 방향으로 일 정속도로 회전하게 된다. touch sensor control버튼이나 "RESET"버튼을 누르면 회전을 멈춘다. 그림 14는 time sensor control의 "To left"버튼을 선택한 경우를 보여준다.

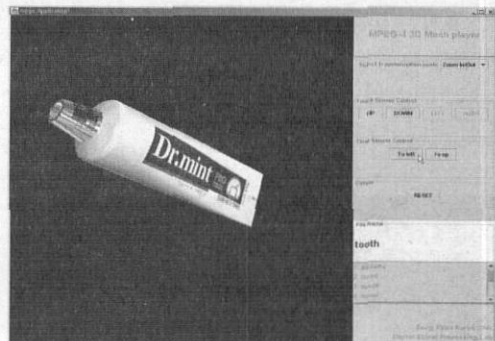


그림 14. 좌측 방향으로 회전

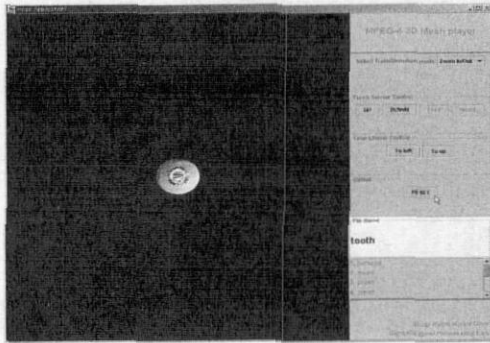
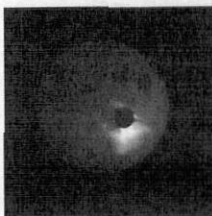


그림 15. 초기 위치 설정 기능

#### IV. 플레이어의 구현 및 실험결과

본 플레이어는 모두 java언어를 이용하여 구현되었으며 3차원 오브젝트를 rendering하기 위한 방법으로 java 3D API를 이용 하였다. 개발 환경은 O/S가 Window 2000인 펜티엄 III PC에서 JBuilder 7으로 개발하였다. 구현된 클래스들은 모두 25개로 각 특성에 따라 크게 main, decoding, rendering, networking, data처리의 다섯 가지로 분류할 수 있다. 각 클래스들은 플레이어의 동작 단위에서 독립적으로 수행되고 수행된 결과를 다음 단계로 넘겨주게 된다.

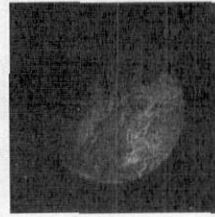
구현된 플레이어는 MPEG-4 기반의 3차원 콘텐츠인 m3d파일과 함께 텍스처가 적용된 3차원 콘텐츠의 경우에는 텍스처 파일(.gif 파일)을 함께 열어 텍스처 맵핑된 3차원 콘텐츠를 rendering한다. 플레이어가 요청한 3차원 콘텐츠가 텍스처가 적용된 콘텐츠인지에 대한 여부는 서버가 알려주게 되고 만일 텍스처가 적용된 콘텐츠라면 서버가 텍스처 파일도 함께 전송한다. 본 논문에서는 실험을 위해 모두 8개의 3차원 모델을 제작해 MPEG-4 3DMC을 적용하여 콘텐츠를 생성한 후 구현된 플레이어 로 rendering하였다. rendering된 결과는 그림 16과 같다.



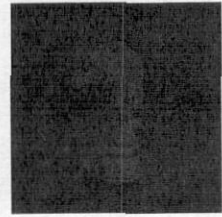
(a) bowl.m3d



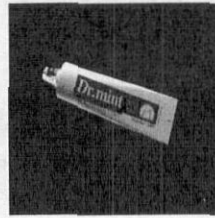
(b) umbrella.m3d



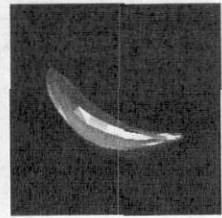
(c) earth.m3d



(d) knight.m3d



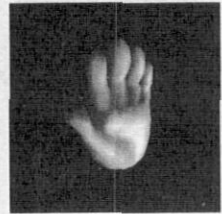
(e) tooth.m3d



(f) banana.m3d



(g) chair.m3d



(h) hand.m3d

그림 16. 각종 콘텐츠의 rendering결과

#### V. 결론


본 논문에서는 MPEG-4 표준으로 채택된 3d mesh decoding 알고리즘을 적용한 3d mesh 플레이어를 java 언어 및 java 3D API를 이용하여 구현했다. 본 플레이어는 MPEG-4기반의 3차원 콘텐츠를 decoding 후 rendering함은 물론 네트워크기능 및 콘텐츠의 다양한 조작기능을 제공한다. MPEG-4 기반의 3차원 콘텐츠는 높은 압축률로 일반 3차원 모델과 비교해 저장 및 전송에 요구되는 대역폭이 현저히 줄어든다. 특히 java언어 및 java3D API를 이용해 구현된 플레이어는 데이터 방송 서비스에 응용되기에 적합할 것으로 기대된다. 시청자가 방송 프로그램을 시청하면서 부가 데이터 서비스를 이용할 수 있는 데이터 방송 서비스의 일례로서, 부가 데이터로 MPEG-4기반의 3차원 콘텐츠 및 플레이어를 전송하여 수신자가 단말기를 통해 3차원 콘텐츠를 조작할 수 있을 것이다. 현재 플레이어는 단일 3차원 오브젝트만을 rendering하는데 이러한 부분은 향후 MPEG-4 BIFS기반의 장면 묘사 정보를 적용함으로써 여러 개의 3차원 오브젝트를 동시에 rendering할 수 있음은 물론 각각의 오브젝트에 대해 보다 다양한 이

멘트를 제공할 수 있을 것으로 기대된다.

### 참 고 문 헌

- [1] Fernando Pereira, Touradj Ebrahimi, "The MPEG-4 book", Prentice Hall, IMSC, 2002
- [2] "Information technology - Coding of audio-visual objects - Part 2 : Visual", ISO/IEC 14496-2, 1999
- [3] Gabriel Taubin and Jarec Rossignac, "Geometric Compression Through Topological Surgery", ACM Transactions on Graphics, Vol.17, No.2, pp. 84-115
- [4] 김철희, "New 알기 쉬운 Java2", 정보문화사, 2002
- [5] "http://www.java3d.co.kr"
- [6] Andre Guezic, Gabriel Taubin, Francis Lazarus and Bill Horn, "Cutting and Stitching: Converting Sets of Polygons to Manifold Surfaces", IEEE Transaction on Visualization and Computer Graphics, Vol.7, No.2, pp.136-151, 2001
- [7] 박동규, "자바 3D 프로그래밍", 도서출판 대림

박 영 경(Young-Kyung Park)      준회원




2002년 2월 : 성균관대학교  
전자공학과 졸업

2002년 3월~현재 :  
성균관대학교 전기전자 및 컴퓨터 공학과 석사과정

<관심분야> 데이터 방송, 컴퓨터 그래픽스, 디지털 영상신호처리

김 용 호(Yong-Ho Kim)      정회원




1998년 2월: 성균관대학교 전자공학과 졸업

2000년 2월: 성균관대학교 전기전자 및 컴퓨터공학부 석사

2000년 3월~현재: 성균관대학교 전기전자 및 컴퓨터공학부 박사과정

<주관심분야> 영상처리, 컴퓨터비전, 컴퓨터그래픽스, 응용공학, 데이터방송

정 종 진(Jong-Jin Jung)      정회원




1997년 2월: 성균관대학교 전자공학과 졸업

2002년 8월: 성균관대학교 전기전자 및 컴퓨터공학부 석사

2002년 9월~현재: 전자부품연구원 인터넷미디어연구센터 선임연구원

<주관심분야> 영상처리, 컴퓨터비전, 데이터방송

김 중 규(Joong-Kyu Kim)      정회원



1980년: 서울대학교 전자공학과 졸업

1982년: 서울대학교 전자공학과 석사

1989년: The University of Michigan, Ann Arbor; Department of Electrical and Computer Engineering

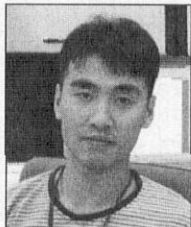
1980년~1981년: 한국전자통신연구소 위촉연구원

1989년~1990년: University of Michigan, Post Doctoral

1990년~현재: 성균관대학교 전기전자 및 컴퓨터공학부 교수

<주관심분야> 적응신호처리, 레이더신호처리, 의학영상신호처리, 음향신호처리, 데이터방송

안 상 우(Sang-Woo Ahn)      정회원



1997년 2월 : 경희대학교 공과대학 전자공학과 졸업

1999년 2월 : 경희대학교 대학원 전자공학과 석사

1999년~현재 : 한국전자통신연구원 연구원

<주관심분야> 영상처리, 대화형 방송 시스템

최 진 수(Jin-Soo Choi)

정회원



1990년 2월 : 경북대학교 공과대  
학 전자공학과 졸업

1992년 2월 : 경북대학교 대학원  
전자공학과 석사

1996년 2월 : 경북대학교 대학원  
전자공학과 박사

1996년~현재 : 한국전자통신연  
구원 선임연구원/영상미디어연구팀장

<주관심분야> 영상통신, 멀티미디어 데이터 방송

김 진 응(Jin-Woong Kim)

정회원



1981년 2월 : 서울대학교 공과대  
학 전자공학과 졸업

1983년 2월 : 서울대학교 대학원  
전자공학과 석사

1993년 8월 : 미국 Texas  
A&M University 전  
기공학과 박사

1983년~현재 : 한국전자통신연구원 책임연구원/방송미  
디어연구부장

<주관심분야> 디지털 VLSI 신호처리, 영상 압축, 영  
상 통신, 멀티미디어 데이터 방송,  
MPEG-4/7, 콘텐츠 보호

안 치 득(Chie-Teuk Ahn)

정회원

1980년: 서울대학교 전자공학과 졸업

1982년: 서울대학교 전자공학과 석사

1991년: The University of Florida, Gainesville

1982년~현재: 한국전자통신연구원 책임연구원(현 전파  
방송연구소장)

<주관심분야> 신호처리, 영상통신