

논문 2006-43SP-6-5

텍스처 영상 재배열을 이용한 삼차원 텍스처 좌표 부호화

(Three-dimensional Texture Coordinate Coding Using Texture Image Rearrangement)

김 성 열* , 호 요 성**

(Sung-Yeol Kim and Yo-Sung Ho)

요 약

삼차원 텍스처 좌표(texture coordinates)란 삼차원 표면을 텍스처와 텍스처 매핑(texture mapping) 기술을 이용하여 렌더링 할 때, 폴리곤(polygon)에 사상되는 텍스처 조각의 위치 정보를 의미한다. 삼차원 텍스처 좌표를 압축하는 기존의 방법은 기하학 정보를 압축하기 위해 사용한 선형 예측기(linear predictor)를 그대로 다시 이용하였다. 그러나 기존의 방법은 부호화 순서에 따라 텍스처 좌표가 불연속적으로 위치하기 때문에, 선형 예측을 효과적으로 수행할 수가 없었다. 특히, 이러한 텍스처 좌표의 불연속성(discontinuity)은 non-atlas 텍스처를 가지고 있는 삼차원 메쉬 모델에서 더 심각하게 나타난다. 본 논문에서는 부호화 순서에 따라 텍스처 영상을 재배열하여 삼차원 텍스처 좌표를 부호화하는 새로운 방법을 제안한다. 제안한 텍스처 좌표 부호화는 텍스처로부터 텍스처 조각을 추출하고, 이를 부호화 순서에 맞게 연속적으로 재배치한 후, 텍스처 좌표를 선형 예측한다. 제안한 방법은 텍스처 좌표의 불연속성을 효과적으로 최소화할 수 있기 때문에, 텍스처 좌표의 부호화 효율을 향상시킬 수 있다. 실험 결과는 제안한 방법이 MPEG-4 3DMC 표준보다 부호화 효율성 측면에서 성능이 우수함을 보여준다.

Abstract

Three-dimensional (3-D) texture coordinates mean the position information of texture segments that are mapped into polygons in a 3-D mesh model. In order to compress texture coordinates, previous works reused the same linear predictor that had already been employed to code geometry data. However, the previous approaches could not carry out linear prediction efficiently since texture coordinates were discontinuous along a coding order. Especially, discontinuities of texture coordinates became more serious in the 3-D mesh model including a non-atlas texture. In this paper, we propose a new scheme to code 3-D texture coordinates using as a texture image rearrangement. The proposed coding scheme first extracts texture segments from a texture. Then, we rearrange the texture segments consecutively along the coding order, and apply a linear prediction to compress texture coordinates. Since the proposed scheme minimizes discontinuities of texture coordinates, we can improve coding efficiency of texture coordinates. Experiment results show that the proposed scheme outperforms the MPEG-4 3DMC standard in terms of coding efficiency.

Keywords : 텍스처 영상 재배열, 삼차원 텍스처 좌표 부호화, 삼차원 메쉬 모델

I. 서 론

우리는 정보 혁명과 디지털 시대로 대변되는 21세기

에 살고 있다. 정보고속도로가 가정에 있는 고성능 PC에 연결되어 네트워크로 필요한 정보와 지식을 습득한다. 집안에서는 대형 스크린과 입체 음향기기를 갖는 고성능 텔레비전을 통해 영화관에서의 감동을 그대로 느낄 수 있으며, 영상 전화를 이용하여 멀리 떨어져 있는 사람들과 통화도 할 수 있다.

이러한 디지털 정보화 시대를 맞이하여, 대용량의 정보를 초고속 네트워크에 실어 전송하는 기술이 개발되고 있고 인터넷을 이용하는 사용자가 기하급수적으로 증가하고 있다. 또한 고품질의 멀티미디어를 요구하는

* 학생회원, ** 평생회원, 광주과학기술원 정보통신공학과 (Gwangju Institute of Science and Technology, Department of Information and Communications)

※ 본 논문은 광주과학기술원(GIST) 실감방송연구센터(RBRC)를 통한 정보통신부 대학 IT연구센터(ITRC)와 교육인적자원부 두뇌한국21(BK21) 정보기술사업의 지원으로 수행되었다.

접수일자: 2006년2월14일, 수정완료일:2006년10월11일

사용자들이 늘고 있어, 실감방송^[1], 삼차원 게임, 삼차원 교육프로그램과 같이, 삼차원 오디오 비디오 정보를 이용한 멀티미디어 응용 프로그램들이 출현하고 있다.

삼차원 정보를 이용한 멀티미디어 응용은 사용자에게 고품질의 영상 서비스를 제공할 뿐만 아니라, 이차원 정보만으로 제공할 수 없는 다양한 사용자 친화적 상호작용(user-friendly interaction)도 제공한다. 이러한 멀티미디어 응용을 실현하기 위해서는 기본적으로 삼차원 정보를 효율적으로 표현하고 처리해야 한다. 삼차원 메쉬 (three-dimensional mesh)^[2] 표현방법은 기하학, 연결성, 광학적 정보로 삼차원 정보를 표현하는 가장 널리 알려진 기법이다. 이와 관련하여 삼차원 메쉬 모델을 효과적으로 처리하는 기술들이 개발되어왔다.

메쉬 변형기법^[3]은 삼차원 메쉬 모델의 표면을 임의의 형태의 삼차원 표면으로 변경한다. 메쉬 정제기법^[4]은 서브디비전(subdivision) 알고리즘을 사용하여 삼차원 표면을 부드럽게 한다. 또한, 삼차원 메쉬 단순화기법^[5]을 이용하여 우리는 삼차원 장면을 다해상도로 표현하고 삼차원 모델을 점진적으로 전송한다. 삼차원 메쉬 부호화기법^[6,7]은 삼차원 모델을 효과적으로 저장하고 전송하기 위해 삼차원 정보를 압축하는 것이다.

일반적으로, 삼차원 메쉬 모델은 엄청난 양의 데이터를 포함하기 때문에, 삼차원 정보를 효율적으로 압축하는 것은 필수적이다. 삼차원 메쉬 부호화는 크게 기하학 정보 부호화, 연결성 정보 부호화, 광학적 정보 부호화로 나눌 수 있다. 기하학 정보는 예측 및 변환 부호화 기법을 사용하여 부호화하고, 연결성 정보는 꼭지점, 폴리곤, 에지 기반의 부호화 기법을 사용한다. 그림 1은 메쉬 부호화의 대표적인 방법을 도식화한 것이다.

기존의 삼차원 메쉬 부호화에 관련된 연구는 기하학 정보와 연결성 정보를 압축하는 기술을 개발하는데 치중되어 있었다. 하지만, 삼차원 메쉬 모델을 나타내는 정보의 60% 이상이 광학적 정보인 것을 감안한다면, 광학적 정보 부호화에 좀 더 관심을 가져야 한다. 특히,

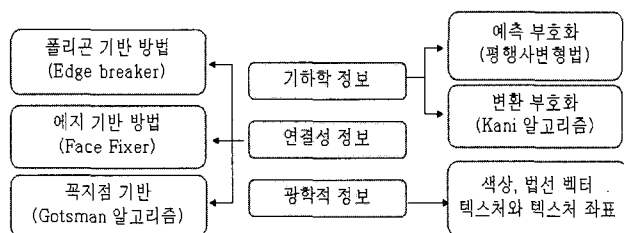


그림 1. 삼차원 메쉬 부호화 방법
Fig. 1. Methods of 3-D mesh compression.

실제와 같은 삼차원 모델을 만들 때, 삼차원 스캐너 같은 삼차원 정보획득 장치를 사용한다. 이 때, 생성된 대부분의 삼차원 메쉬 모델은 색상 정보 대신 텍스처(texture)와 텍스처 좌표(texture coordinates) 정보를 포함하고 있다. 따라서 텍스처와 텍스처 좌표를 효율적으로 부호화하는 연구가 필요하다. 일반적으로, 텍스처는 JPEG 표준을 이용하여 부호화하고, 텍스처 좌표 정보는 기존의 기하학 정보를 부호화하는 방법과 같은 방식으로 부호화한다. 하지만, 기하학 정보를 부호화하는 기법을 그대로 이용하여 텍스처 좌표를 부호화하면, 텍스처 좌표의 불연속성(discontinuity) 때문에 부호화 효율이 떨어지게 된다.

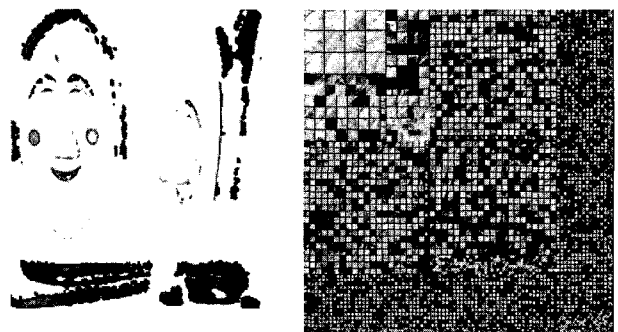
본 논문에서는 삼차원 메쉬 모델의 텍스처 좌표 정보를 효율적으로 부호화하는 텍스처 영상 재배열 방법을 소개한다. 텍스처 영상 재배열은 삼차원 메쉬 정보로부터 부호화 순서와 텍스처 조각을 추출한 후, 그 부호화 순서에 따라 텍스처 조각을 재배열하여 텍스처 좌표의 불연속성을 최소화한다.

본 논문의 II장에서는 텍스처의 종류와 기존의 텍스처 좌표 부호화 방법에 대해서 설명한다. III장에서는 제안한 non-atlas 텍스처 재배열 과정을 설명한다. IV장에서는 제안한 방법을 실험한 통해 검증한 결과를 보여 주고, V장에서 본 논문의 결론을 맺는다.

II. 기존의 텍스처 좌표 부호화와 불연속성

1. 텍스처와 텍스처 매핑

텍스처는 삼차원 모델에 입혀질 이차원 사진 영상이고, 텍스처 좌표는 각 꼭지점에 할당되는 텍스처 위치 정보이다. 일반적으로, 텍스처는 직접 사물이나 배경을 사진기로 찍어 얻어낸다. 텍스처 좌표는 다른 광학적



(a) atlas 텍스처

(b) non-atlas 텍스처

그림 2. 텍스처의 종류

Fig. 2. Types of Textures.

정보인 색상 정보와 법선 벡터 정보와 같이, 텍스처 좌표 리스트와 텍스처 좌표 인덱스로 표현한다. 그림 2는 텍스처의 종류를 보여준다.

텍스처는 atlas 텍스처와 non-atlas 텍스처로 나눈다. 그림 2(a)과 같이, atlas 텍스처는 조각난 영상의 차트를 하나의 텍스처 영상으로 모아놓은 것이다. 반면에 그림 2(b)에서 보는 것과 같이, non-atlas 텍스처는 사각형 조각들의 모음으로, 텍스처 매개화(parameterization)과정을 거쳐 만든 영상이다.

텍스처를 이용하여 삼차원 모델에 적용하는 것을 텍스처 매핑(texture mapping)^[8,9]이라 한다. 텍스처 매핑은 텍스처 좌표 정보를 이용하여 텍스처로부터 추출한 이차원의 텍스처 조각(texture segment)을 삼차원 메쉬 모델의 폴리곤에 위치시켜 이루어진다.

2. 텍스처 좌표 부호화

텍스처 좌표는 꼭지점마다 하나 또는 여러 개가 할당되기 때문에 엄청난 양의 정보를 포함한다. 결과적으로 이를 효율적으로 부호화하는 방법을 개발해야 한다. 텍스처 좌표를 부호화하기 위해서 기존의 방법은 기하학 정보를 부호화하기 위해 사용하였던 선형 예측기(linear predictor)를 그대로 재사용하였다. 물론 다른 방법을 사용하여 텍스처 좌표를 부호화하려는 시도도 있었지만 만족할만한 성과를 내지 못했다.

Deering^[10]와 Taubin^[11]은 연결성 정보에 따라 모든 꼭지점을 탐색한 후, 토폴로지 분해과정을 수행하고 선형 예측기를 사용하여 텍스처 좌표를 부호화 하였다. Touma^[12]는 그림 3에서 보는 것과 같이 꼭지점 탐색(vertex traversal) 순서에 따라 부호화 순서를 정하고, 평행사변형법을 이용하여 텍스처 좌표를 예측 부호화하였다. 이 때 발생한 텍스처 좌표의 잔여 오류(residual

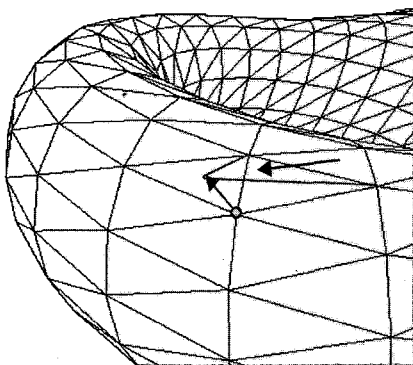


그림 3. 텍스처 좌표 예측 부호화
Fig. 3. Predictive Coding of Texture Coordinates.

error)는 엔트로피 부호화하였다. Isenburg^[13]는 텍스처 좌표의 불연속성에 대한 문제를 제기하였고, 선택적인 선형 예측기를 이용하여 텍스처 좌표를 부호화하였다.

기존의 텍스처 좌표 부호화방법은 atlas 텍스처를 포함하는 삼차원 메쉬 모델에 대해서 좋은 성능을 보여주었다. 하지만, non-atlas 텍스처를 포함하는 삼차원 메쉬 모델에 대해서는 텍스처 좌표가 부호화 순서에 따라 불연속적으로 위치하기 때문에, 좋은 성능을 보여주지 못했다. 차세대 멀티미디어 응용에서 실감나는 삼차원 메쉬 모델을 이용하기 위해서, 우리는 삼차원 스캐너와 같은 삼차원 획득 장치로 삼차원 모델을 얻어낸다. 이러한 경우, 대부분의 삼차원 메쉬 모델들이 non-atlas 텍스처를 포함하고 있다. 따라서 non-atlas 텍스처를 포함한 삼차원 메쉬 모델에 대해서 효율적으로 텍스처 좌표를 부호화하는 방법을 개발해야 한다.

3. 텍스처 좌표의 불연속성

텍스처 매핑은 이차원 텍스처 조각을 텍스처 좌표에 따라 삼차원 메쉬 모델의 폴리곤에 사상하는 과정이다. 텍스처 매핑은 폴리곤마다 독립적으로 이루어지지만, 이웃한 폴리곤이 이웃한 텍스처 조각으로 사상되는 것이 렌더링에 더 효과적이다. 텍스처 매개화는 폴리곤과 사상되는 텍스처가 연속적으로 일어나도록 텍스처 좌표를 찾아내는 과정이다. 또한, 텍스처 매개화는 꼭지점 정보와 텍스처 사이의 관계를 나타낸다.

Non-atlas 텍스처를 포함하는 삼차원 메쉬 모델인 경우, 텍스처 매개화가 가끔씩 어긋나게 된다. 그리고 꼭지점 부호화 순서에 따라 텍스처 좌표를 부호화하면 텍스처 좌표가 불연속적으로 정렬된다. 이와 같이, 텍스처 좌표가 불연속적으로 정렬되는 현상을 텍스처 좌표의 불연속성이라고 한다^[14]. 텍스처 좌표의 불연속성으로 인하여, 기존의 선형 예측기를 이용하여 텍스처 좌표를 부호화하면 잔여 오류가 증가하여 텍스처 좌표의

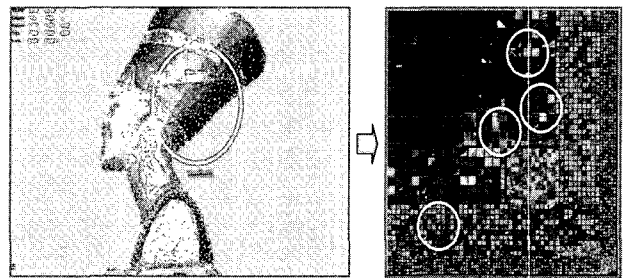


그림 4. 텍스처 좌표의 불연속성
Fig. 4. Discontinuities of Texture Coordinates.

부호화 효율이 떨어지게 된다. 그림 4는 텍스처 좌표의 불연속성을 보여준다.

III. 텍스처 재배열을 이용한 텍스처 좌표 부호화

본 논문에서는 non-atlas 텍스처를 재배열하여 텍스처 좌표를 부호화하는 새로운 방법을 제안한다. 제안하는 방법은 삼차원 메쉬 모델 분석과정, 텍스처 좌표 분석과정, non-atlas 텍스처 재배열과정, 텍스처 좌표 예측 부호화과정으로 나뉘어진다. 그림 5는 제안한 부호화 시스템의 전체적인 흐름도를 도식화한 것이다.

제안한 방법은 우선 삼차원 메쉬 모델을 입력으로 받아 메쉬 모델 분석기를 통해 기하학, 연결성, 광학적 정보를 얻어낸다. 그런 다음, 꼭지점 탐색 기술을 이용하여 부호화 순서를 결정하고, 이에 따라 텍스처 조각을 non-atlas 텍스처로부터 추출한다. 추출된 텍스처 조각은 부호화 순서에 따라 재배열되고 새로운 텍스처로 만들어진다. 마지막으로 평행사변형법을 사용하여 텍스처 좌표를 예측 부호화한다.

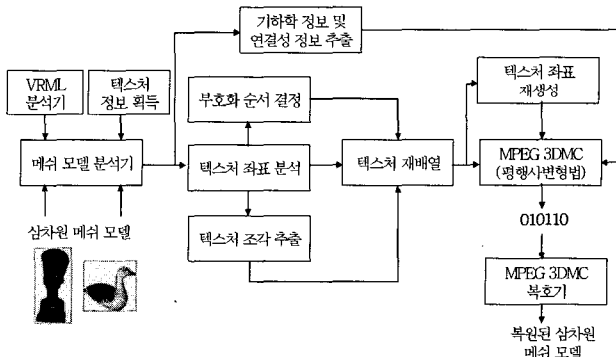


그림 5. 제안한 텍스처 좌표 부호화의 흐름도
Fig. 5. Overall Flow of the Proposed Scheme.

1. 삼차원 메쉬 모델 분석

삼차원 메쉬 모델 분석은 삼차원 메쉬 정보로부터 텍스처와 텍스처 좌표를 분리해 내는 과정이다. 입력 받는 삼차원 모델의 형식은 VRML(virtual reality modeling language)이다^[15]. VRML은 국제 표준 기구인 ISO와 IEC에서 공인한 인터넷상의 삼차원 그래픽을 표현하는 표준이다.

VRML 분석기는 VRML 형식의 삼차원 모델을 파싱(parsing)하여 삼차원 정보를 분리한다. 그림 6은 Nefertiti 모델을 표현하는 VRML 파일의 일부분으로, 텍스처와 텍스처 좌표 정보를 보여준다. 텍스처는 ImageTexture라는 노드(node)를 통해 정의하고, 해당

```
texture ImageTexture {
    url "nefert5_T5k1.bmp"
}
:
texCoord TextureCoordinate { point [
0.333008 1.13184, 0.333008 1.11621, 0.348633 1.13184,
0.192383 1.19434, 0.192383 1.20996, 0.176758 1.19434,
0.750977 1.53027,
:
texCoordIndex [
0, 1, 2, -1, 3, 4, 5, -1, 6, 7, 8, -1, 9, 10, 11, -1, 12, 13, 14, -1,
15, 16, 17, -1, 18, 19, 20, -1, 21, 22, 23, -1, 24, 25, 26, -1
:
}
```

그림 6. VRML에서의 텍스처 좌표
Fig. 6. Texture Coordinates in VRML.

되는 파일 위치는 url로 나타낸다. 위의 예인 Nefertiti 모델의 경우, 텍스처는 nefert5_t5k1.bmp 파일임을 알 수 있다. nefert5_5k1.bmp는 512×512의 해상도를 갖는 이차원 영상이다. 일반적으로 텍스처는 가로와 세로의 길이가 2ⁿ의 해상도를 갖는다.

텍스처 좌표는 TextureCoordinate라는 노드로 정의되고 point 노드 안에 실제 좌표들이 있다. Nefertiti 모델은 14993개의 텍스처 좌표를 가지고 있다. 위 예에서 보듯이, 텍스처 좌표는 텍스처의 한 위치를 나타내기 때문에 x축과 y축의 좌표 쌍으로 이루어져 있다.

텍스처 좌표는 색상이나 법선 벡터와 같이 인덱스를 통해 실제 폴리곤과 사상한다. 텍스처 좌표 인덱스는 texCoordIndex라는 노드로 정의하고 -1을 구별자로 하여 텍스처 좌표가 사상될 폴리곤을 가리킨다. 일반적으로, 삼차원 메쉬 모델은 삼각형으로 이루어져 있기 때문에, 구별자인 -1은 세 개의 텍스처 좌표 인덱스 후에 대부분 나타나게 된다.

텍스처 좌표 인덱스 정보가 t_{1_index}, t_{2_index}, t_{3_index}, -1 이고 그 때의 연결성 정보 인덱스가 v_{1_index}, v_{2_index}, v_{3_index}, -1이면, 구별자인 -1 이전에 세 개의 인덱스로 구성되었기 때문에 사상할 폴리곤이 삼각형을 알 수 있다. 그리고 t_{1_index} 번째, t_{2_index} 번째, t_{3_index} 번째의 텍스처 좌표를 찾아 v_{1_index} 번째, v_{2_index} 번째, v_{3_index} 번째의 꼭지점에 할당하여 텍스처 매핑을 수행한다. 그림 7은 Nefertiti 모델과 텍스처를 보여준다.

삼차원 메쉬 모델 분석에서는 텍스처와 텍스처 좌표 뿐 만 아니라, 삼차원 메쉬 모델의 정보인 기하학 정보

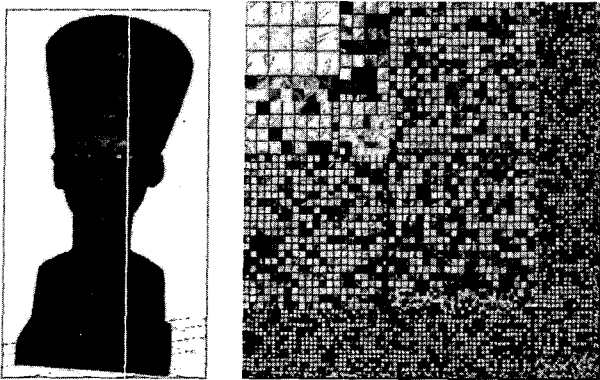


그림 7. 삼차원 메쉬 모델과 그 텍스처
Fig. 7. A 3-D Mesh Model and its Texture.

와 연결성 정보도 함께 찾는다. 또한, 이 과정에서 법선 벡터 정보도 텍스처 좌표와 같은 방식으로 얻어낸다.

2. 텍스처 좌표 분석

텍스처 좌표 분석 과정에서는 꼭지점 탐색 순서에 따라 부호화 순서를 결정하고 텍스처 조각을 추출한다. 또한 텍스처 좌표 분석과정에서는 정규화된 텍스처 좌표를 텍스처의 영상 좌표로 변환한다. 일반적으로, 텍스처 좌표는 0과 1 사이로 정규화되어 있다. 따라서 정규화된 텍스처 좌표를 텍스처의 영상 좌표로 변환해야 한다. 텍스처의 영상 좌표는 텍스처 좌표에 텍스처의 가로와 세로의 크기를 곱하여 식 (1)과 식 (2)와 같이 얻어낼 수 있다.

$$X_{pos} = T_{width} * X_{norm} \tag{1}$$

$$Y_{pos} = T_{height} * Y_{norm} \tag{2}$$

여기서, X_{pos} 와 Y_{pos} 는 텍스처의 x축과 y축의 영상 좌표 값을 의미하며, T_{width} 와 T_{height} 는 텍스처의 가로와 세로 크기를 나타낸다. X_{norm} 과 Y_{norm} 은 각각 정규화된 텍스처 좌표의 x축과 y축 좌표 값을 의미한다. 한 가지 주의할 점은, 어떠한 삼차원 메쉬 모델은 텍스처 좌표의 y값이 뒤집어져 있다. 이런 경우에는 식 (3)과 같이 텍스처의 y축 영상 좌표를 구한다.

$$Y_{pos} = T_{Height} * (1 - Y_{norm}) \tag{3}$$

그리고 X_{pos} 와 Y_{pos} 는 영상 좌표 값이기 때문에 정수형의 값을 갖는다. 따라서 실수 값인 X_{norm} 과 Y_{norm} 을 영상 좌표로 변환한 후, 반올림 과정을 통해서 X_{pos} 와 Y_{pos} 을 정수형으로 보정한다.

텍스처 좌표 분석에서는 꼭지점 탐색을 통해서 부호

화 순서를 정한다. 본 논문에서는 삼차원 메쉬 모델을 부호화하는 MPEG-4 표준인 3DMC^[16]에서 제공하는 꼭지점 탐색 알고리즘을 사용한다. 꼭지점 탐색 과정은 다음과 같다.

- i) 탐색을 위한 시작 꼭지점 v_0 과 그와 이웃하고 반시계 방향의 꼭지점인 v_1 을 정한다. v_0 와 v_1 을 잇는 에지를 기본 에지라 한다.
- ii) v_0 를 회전축(pivot)으로 하여, v_1 과 이웃하고 반시계 방향의 꼭지점인 v_2 를 찾는다. 그리고 v_1 과 v_2 를 잇는 에지를 큐(queue)에 저장한다.
- iii) v_0 를 회전축으로 하여 다른 이웃한 꼭지점을 모두 찾으면 큐로부터 v_1 과 v_2 를 꺼내어 기본 에지로 한다.
- iv) 큐에 에지가 없을 때까지 (i)부터 (iii)를 반복한다.

위 과정에서 부호화 순서는 처음 선정한 기본 에지인 v_0 와 v_1 을 시작으로 큐로부터 꺼내어진 꼭지점의 순서이다. 그림 8은 꼭지점 탐색 과정을 보여준다.

꼭지점 탐색 과정을 통해 삼차원 메쉬 모델의 폴리곤을 추출한다. 그리고 폴리곤을 텍스처에 있는 조각을 사상하여 렌더링한다. 여기서, 폴리곤을 이루는 꼭지점마다 텍스처 좌표가 존재하기 때문에, 해당 폴리곤에 사상되는 텍스처 조각을 추출할 수 있다.

꼭지점 탐색 과정에서 현재 큐에 저장할 에지를 (v_1 ,

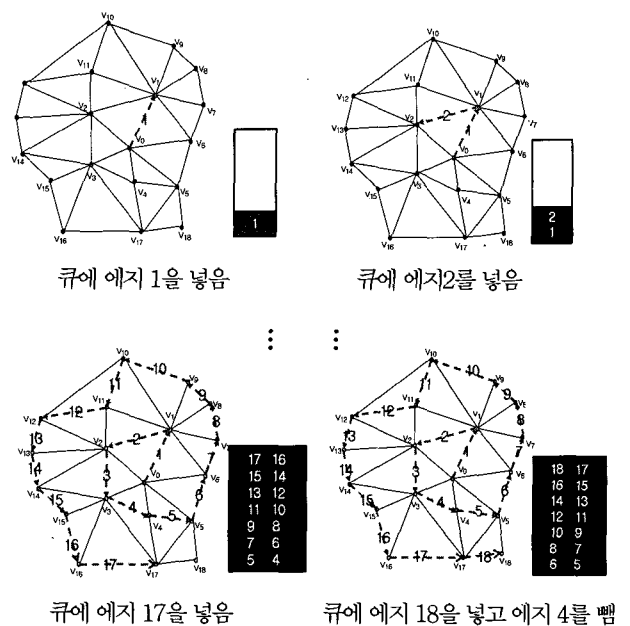


그림 8. 꼭지점 탐색
Fig. 8. Vertex Traversal.

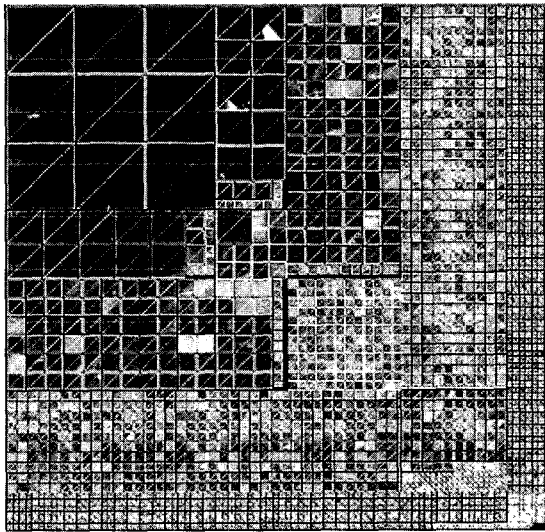


그림 9. 텍스처 조각 추출
Fig. 9. Extraction of Texture Segments.

v_2)라고 하고, 이 때의 회전축 꼭지점을 v_0 라 하자. 다음에 탐색되는 꼭지점이 v_3 인 경우, 이 때 만들어지는 삼각형은 (v_0, v_1, v_2) 와 (v_0, v_2, v_3) 이다. 그리고 v_0, v_1, v_2, v_3 에 할당된 텍스처 좌표를 각각 $(tx_0, ty_0), (tx_1, ty_1), (tx_2, ty_2), (tx_3, ty_3)$ 로 가정하자. 또한 각 텍스처 좌표를 텍스처 영상 좌표로 변환한 결과를 $(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)$ 라 하자. 결과적으로 텍스처의 삼각형 영역인 $\{(x_0, y_0), (x_1, y_1), (x_2, y_2)\}$ 와 $\{(x_0, y_0), (x_2, y_2), (x_3, y_3)\}$ 이 추출한 텍스처 조각의 위치 정보이다. 우리는 텍스처 조각의 위치 정보를 연결리스트 구조를 사용하여 정보를 저장한다.

주의할 점은 회전축 꼭지점을 중심으로 이웃한 모든 꼭지점을 탐색하여, 큐에서 새로운 꼭지점을 얻을 때, 폴리곤이 제대로 추출되지 않을 수 있다. 이와 같은 문제를 해결하기 위해, 추출한 폴리곤이 연결성 정보에 포함되는지 확인해야 한다. 삼차원 메쉬 모델 분석과정에서 연결성 정보를 추출할 때, 이웃한 폴리곤들의 집합을 구하여 비교하면 쉽게 확인할 수 있다. 그림 9는 삼차원 메쉬 모델의 텍스처 조각을 보여준다.

3. 텍스처 재배열

텍스처 재배열은 텍스처 조각을 부호화 순서에 따라 재배열하여 새로운 텍스처를 만드는 과정이다. 또한, 텍스처 재배열 과정에서는 만들어진 텍스처로부터 텍스처 좌표를 다시 생성한다. 텍스처 재배열을 통해 텍스처 좌표를 부호화할 때 생기는 불연속성을 최소화한다.

Non-atlas 텍스처로부터 추출한 텍스처 조각은 크게 세 가지 유형으로 나눈다. 하나는 위 방향의 직각 삼각

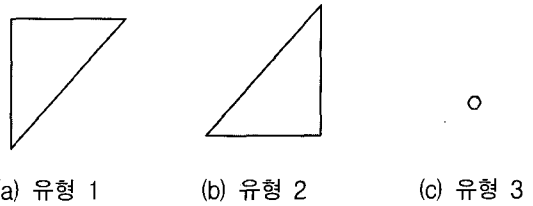


그림 10. 텍스처 조각의 유형
Fig. 10. Types of Texture Segments.

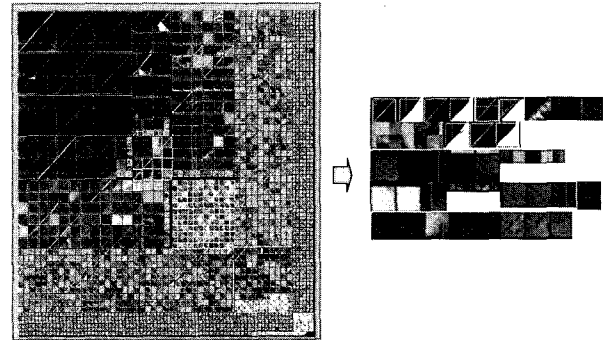


그림 11. 텍스처 재배열
Fig. 11. Texture Rearrangement.

형이고, 다른 하나는 아래 방향의 직각 삼각형이며, 나머지 하나는 점이다. 텍스처 조각이 점인 경우는 세 꼭지점의 텍스처 좌표가 같다. VRML에서 텍스처 좌표를 중복하여 사용하였을 경우, 꼭지점의 텍스처 좌표가 같을 수 있다. 그림 10은 텍스처 좌표의 유형을 보여준다.

Non-atlas 텍스처를 재배열을 하기 전에, 생성할 새로운 텍스처를 위해 영상 버퍼를 메모리에 할당한다. 이 때, 가로의 크기는 원 텍스처의 가로의 크기와 같게 하고, 세로의 크기는 약 두 배가량 크게 한다.

우리는 텍스처 좌표분석과정에 생성한 연결리스트로부터 차례로 텍스처 조각을 찾아낸다. 연결리스트는 부호화 순서에 따라 저장되어 있기 때문에, 연결리스트 머리(head)부분부터 꼬리(tail)까지 순서대로 영상 버퍼에 할당한다. 텍스처 조각은 영상 버퍼에 zig-zag 순으로 재배열된다. zig-zag 순으로 재배열하여 텍스처 좌표의 연속성을 유지할 수 있다. 그림 11은 텍스처를 재배열하는 과정을 보여준다.

텍스처 조각의 텍스처 영상 좌표가 $\{(x_0, y_0), (x_1, y_1), (x_2, y_2)\}$ 라 가정하자. 그리고 영상 버퍼의 위치 좌표를 (x_{pos}, y_{pos}) 라고 하고 현재의 이동 방향이 왼쪽에서 오른쪽 쪽임을 나타내기 위해 flag에 0을 할당하자. 오른쪽에서 왼쪽 방향인 경우 flag는 1이 된다. 또한, 현재 방향에서 제일 크기가 큰 텍스처 조각의 높이를 y_{max} 라 하여 유지하자. non-atlas 텍스처의 텍스처 조각이 직각 삼각형이거나 점이기 때문에, 현재 텍스처 조각을 삽입한 후

의 영상 좌표는 식 (4), 식 (5)와 같다.

$$X_{pos} = X_{pos} + (X_2 - X_0) \tag{4}$$

$$Y_{pos} = Y_{pos} + (y_2 - y_0) \tag{5}$$

만약 X_{pos} 가 텍스처의 가로 크기보다 크면, flag를 1로 반전시키고, Y_{pos} 를 y_{max} 만큼 이동하여 텍스처 조각을 버퍼에 할당한다. y_{max} 는 $(y_2 - y_0)$ 값으로 초기화한다.

텍스처 재배열 후에, 텍스처 좌표를 새로운 텍스처에 맞게 재생성 한다. 텍스처 조각을 영상 버퍼에 할당하면서, 해당 영상 위치 정보를 버퍼에 저장한다. 기본적으로 텍스처 좌표는 0과 1사이의 값으로 정규화되어 있기 때문에, 식 (6)과 식 (7)을 통하여 얻는다.

$$X_{new_norm} = X_{new_pos} \div T_{new_width} \tag{6}$$

$$Y_{new_norm} = Y_{new_pos} \div T_{new_height} \tag{7}$$

여기서 X_{new_norm} 과 Y_{new_norm} 은 새로운 텍스처 좌표이고 X_{new_pos} 와 Y_{new_pos} 는 영상버퍼의 위치 좌표를 의미하며, T_{new_width} 와 T_{new_height} 는 새로운 텍스처의 가로 세로 크기를 각각 나타낸다.

4. 텍스처 좌표의 예측 부호화

텍스처 좌표 부호화는 MPEG-4 3DMC를 이용하여 예측 부호화한다. MPEG-4 3DMC는 꼭지점 탐색 과정을 거쳐 평행사변형법을 이용한 예측기를 사용한다. 텍스처 좌표의 부호화 순서로 텍스처 영상을 재배열하여 잔여 오류의 분산과 엔트로피를 줄인다. 그림 12는 평행사변형법을 이용한 예측부호화 과정을 보여준다.

텍스처 좌표 (tx_4, ty_4) 를 부호화하기 위해, 식 (8)로부터 참조 텍스처 좌표인 (tx_4', ty_4') 를 얻는다.

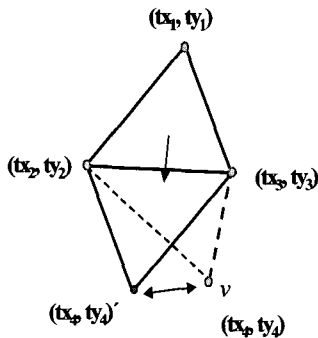


그림 12. 평행사변형법
Fig. 12. Parallelogram Predictor.

$$(tx_4', ty_4') = (tx_1, ty_1) + (tx_2, ty_2) - (tx_3, ty_3) \tag{8}$$

그리고 식 (9)로부터 텍스처 좌표의 잔여 오류 (ex_4, ey_4) 를 얻는다.

$$(ex_4, ey_4) = (tx_4, ty_4) - (tx_4', ty_4') \tag{9}$$

잔여 오류는 양자화 과정과 가변길이 부호기 (variable length coder)를 거쳐 엔트로피 부호화한다. 텍스처 재배열 과정을 통해 불연속성을 제거하여 잔여 오류를 최소화하였기 때문에, 제안한 방법은 텍스처 좌표에 대한 부호화 효율을 높일 수 있다.

IV. 실험 결과

본 논문에서는 non-atlas 텍스처 재배열을 이용한 텍스처 좌표의 부호화방법을 제안하였다. 제안한 방법의 성능 검증을 위해 non-atlas 텍스처를 포함하는 삼차원 메쉬 모델인 Coin모델과 Nefertiti 모델을 실험하였다.

Coin 모델과 Nefertiti 모델은 각각 1468개의 꼭지점과 2501개의 꼭지점으로 이루어져 있으며, 2391개의 폴리곤과 4997개의 폴리곤으로 구성되어 있다. 또한, 8796개와 14993개의 텍스처 좌표로 이루어져 있으며, 512x512 크기의 non-atlas 텍스처 영상을 포함한다. 두 실험 모델 모두 VRML 형식으로 표현되어 있다. 그림 13은 실험 모델과 해당 텍스처를 보여준다. 그리고 표 1은 실험 모델의 정보를 나타낸다.

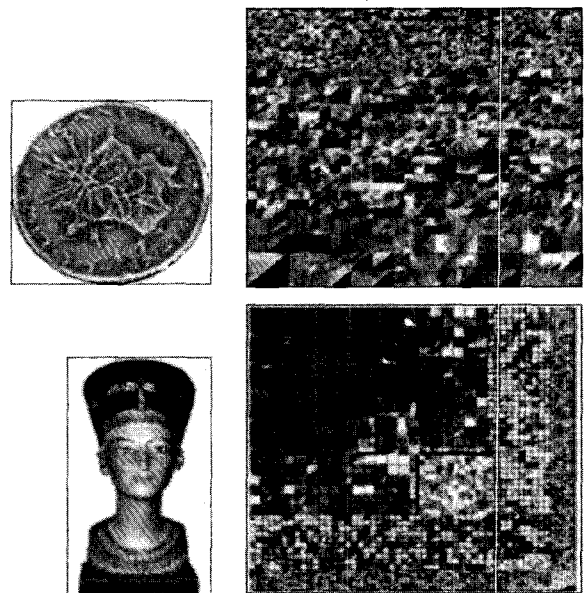


그림 13. 실험 모델
Fig. 13. Test Models.

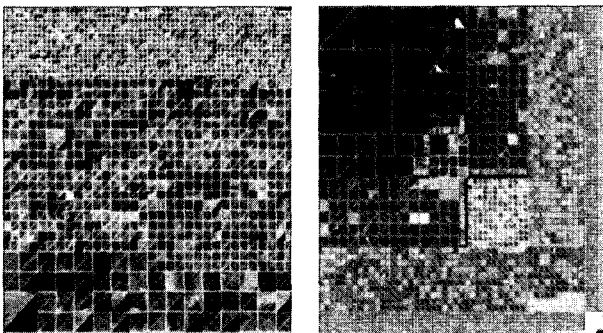
1. 텍스처 재배열

텍스처 재배열을 수행하기 위해 텍스처 조각을 추출하였다. 꼭지점 탐색 알고리즘을 사용하여 부호화 순서를 찾았고, 부호화 순서에 따라 텍스처 조각을 연결리스트에 저장하였다. 그림 14는 Coin 모델과 Nefertiti 모델의 텍스처로부터 텍스처 조각을 추출한 결과이다.

그런 다음, 추출한 텍스처 조각을 텍스처 좌표의 부호화 순서에 따라 재배열하여 새로운 텍스처를 생성하였다. 그림 15에서 보듯이, 원 텍스처보다 영상의 크기가 커졌다. Coin 모델인 경우 약 1.2배가 증가하였고, Nefertiti 모델인 경우 1.4배가 증가하였다. 텍스처 좌표

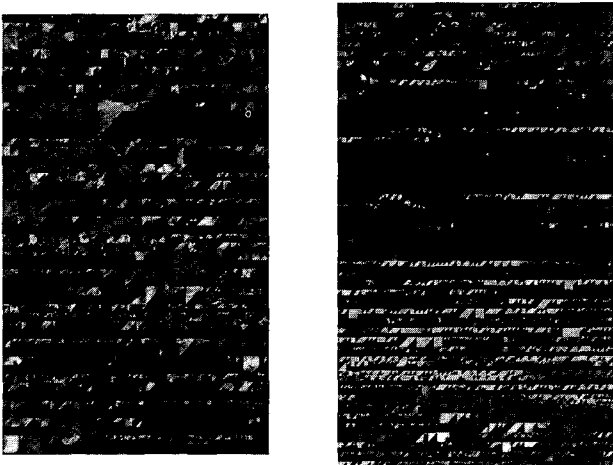
표 1. 실험 모델 정보
Table 1. Test Model Information.

모델	꼭지점 수	폴리곤 수	텍스처 좌표 수
Nefertiti	1468	2391	8796
Coin	2501	4997	14993



(a) Coin 모델 (b)Nefertiti 모델

그림 14. 텍스처 조각 결과
Fig. 14. Results of Texture Segments.



(a) Coin 모델 (b) Nefertiti 모델

그림 15. 텍스처 재배열 결과
Fig. 15. Results of Texture Rearrangement.

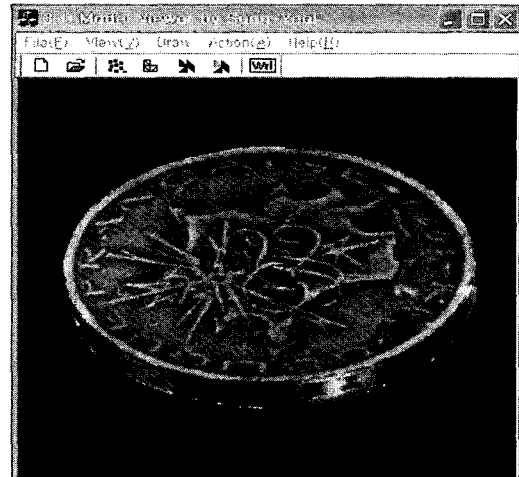


그림 16. 텍스처 재배열 후의 Coin 모델
Fig. 16 Coin Model after Texture Rearrangement.

의 불연속성을 최소화하기 위해 zig-zag 탐색 순서로 재배열하였다.

텍스처 재배열 과정을 거쳐 다시 만들어진 삼차원 메쉬 모델은 원 모델과 같다. 텍스처 좌표를 새로운 텍스처에 맞게 재조정을 하여, 해당 폴리곤에 같은 텍스처 조각을 사상하였기 때문에 왜곡이 없다고 가정했다. 다만, 새로 만들어진 텍스처 좌표를 텍스처 영상 좌표로부터 정규화하고 이를 다시 텍스처 좌표를 변환하여 폴리곤에 사상시킬 때, 반올림에 따른 오류는 있을 수 있다. 그림 16은 새로운 텍스처를 사용하여 텍스처 매핑을 수행한 Coin 모델의 결과를 보여준다.

2. 텍스처 좌표 부호화

텍스처 좌표의 부호화 효율은 제안한 방법과 기존의 MPEG-4 3DMC와 비교하였다. 두 방법 모두 평행사변형법을 이용하여 예측 부호화하였고, 텍스처 좌표의 할당 비트는 10bit로 고정하였다. 본 실험은 Windows XP 운영체제에 512MB 메모리를 포함한 Pentium-4 (CPU 1.5Ghz) PC에서 이루어졌다. 사용 프로그램 언어는 Visual C++ 6.0이다.

표 2는 제안한 방법과 3DMC를 비교한 결과를 보여준다. 잔여 오류 분산은 평행사변형법을 이용하여 예측한 텍스처 좌표의 잔여 오류에 대한 분산을 의미한다. Coin 모델의 경우는 3DMC를 이용하였을 때, 잔여 오류 분산은 25.44였고, 제안한 방법의 잔여 오류 분산은 13.22였다. Nefertiti 모델인 경우에서도 잔여 오류 분산이 제안한 방법에서 더 작게 나타났다.

텍스처 재배열 과정을 통해 텍스처 좌표의 불연속성을 줄였기 때문에, 제안한 방법이 3DMC보다 잔여 오류

표 2. 텍스처 좌표 부호화 결과 비교
Table 2. Comparison of Texture Coordinate Coding.

실험모델	3DMC		제안한 방법	
	잔여오류 분산	텍스처좌표 압축	잔여오류 분산	텍스처좌표 압축
Coin	25.44	168.4 KB	13.22	106.3 KB
Nefertiti	27.64	287.5 KB	14.27	184.3 KB

의 분산이 작았다. 최종적으로 우리는 이 잔여 오류를 엔트로피 부호화하였다. 결과적으로 제안한 방법이 부호화 효율면에서도 3DMC보다 성능이 좋았다.

3. 텍스처 부호화

텍스처는 JPEG을 이용하여 부호화했다. 양자화는 JPEG 표준인 Rao가 제안한 양자화 행렬(quantization matrix)을 사용하였다^[17]. 표 3은 원 텍스처와 텍스처 재배열하여 생성한 텍스처의 부호화 결과를 보여준다.

표 3에서 보듯이, zig-zag 순서로 텍스처 조각을 재배열하였기 때문에, 제안한 텍스처의 영상 크기가 원 텍스처 영상의 크기보다 커졌다. Coin 모델의 경우 원 텍스처의 해상도가 512×152였는데, 텍스처 재배열 후 512×612로 커졌다. Nefertiti 모델의 경우도 텍스처 재배열 후 텍스처의 크기가 원 텍스처보다 증가하였다. 결과적으로 텍스처를 부호화하기 위해 제안한 텍스처는 원 텍스처보다 더 많은 비트가 필요하였다.

표 3. 텍스처 부호화 결과 비교
Table 3. Comparison of Texture Coding

실험모델	3DMC		제안한 방법	
	텍스처크기	JPEG압축	텍스처크기	JPEG압축
Coin	512×512	79 KB	512×612	91 KB
Neffertiti	512×512	96 KB	512×680	112 KB

4. 실험 평가

실험에서 알 수 있듯이, non-atlas 텍스처 재배열 과정을 통해 텍스처의 영상 크기가 증가하였지만, 텍스처 좌표의 불연속성을 효과적으로 줄일 수 있었다.

표 1에서 보듯이, 대부분의 삼각형 메쉬 모델은 하나의 꼭지점에 여러 개의 텍스처 좌표를 갖는다. Coin 모델인 경우는 꼭지점의 수가 1468에 반해 텍스처 좌표의 수는 8796이었다. 결과적으로 텍스처를 부호화하는데

필요한 비트보다 텍스처 좌표를 부호화하는데 더 많은 비트가 소모되었다. 표 2와 표 3의 결과에서 보듯이, 텍스처 좌표와 텍스처를 포함한 부호화 효율은 전체적으로 약 20~30% 증가하였다.

V. 결 론

본 논문에서는 텍스처 좌표와 non-atlas 텍스처 사이의 불연속성을 최소화하여 텍스처 좌표를 부호화하기 위해, non-atlas 텍스처의 재배열 기법을 제안하였다. 텍스처 좌표 부호화에 관련된 기존의 방법들은 텍스처 좌표의 불연속성을 고려하지 않았다. 특히, non-atlas 텍스처를 포함한 삼차원 모델인 경우, 부호화 순서에 따라 텍스처 좌표를 부호화하면 불연속성 문제가 더욱 심각해진다. 삼차원 메쉬 모델의 정보를 분석하여 텍스처와 텍스처 좌표를 얻었고, 텍스처로부터 부호화 순서에 따라 폴리곤에 맞는 텍스처 조각을 추출하였다. 그리고 텍스처 조각을 연속적으로 재배열하여 텍스처 좌표의 불연속성을 최소화하였다. 제안한 방법은 텍스처 좌표의 연속적으로 재배치하여 텍스처 좌표를 부호화하였기 때문에, 기존의 MPEG-4 3DMC보다 부호화 효율성면에서 성능이 더 우수하였다.

참 고 문 헌

[1] S. Y Kim, S. U. Yoon, and Y.S. Ho, "Realistic Broadcasting Using Multi-modal Immersive Media," Lecture Notes in Computer Science. vol. 3767, pp. 687-697, 2005.

[2] H. Hoppe, "Progressive Meshes", SIGGRAPH, pp. 99-108, 1996.

[3] M. Desbrum, M. Meyer, and P. Alliez, "Intrinsic Parameterizations of Surface Meshes," Proc. of EUROGRAPHICS, pp. 209-218, 2005.

[4] T. Derose, M. Kass, and T. Truong, "Subdivision Surfaces in Character Animation," SIGGRAPH. pp. 85-94, 1998.

[5] M. Garland and P.S. Heckbert, "Surface Simplification Using Quadric Error Metrics," SIGGRAPH, pp. 209-216. 1997.

[6] J. Rossignac, "Geometry Simplification and Compression," Course Notes of SIGGRAPH, 1997.

[7] 안정환, 호요성, "통합예측을 이용한 삼차원 메쉬의 기하 정보 부호화 알고리즘," 전자공학회논문지, 제 40권 SP편, 제 3호, pp. 185-193, 2003..

- [8] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe, "Texture Mapping Progressive Meshes," SIGGRAPH, pp. 409-416, 2001.
- [9] <http://www.cs.cmu.edu/~ph/>.
- [10] M. Deering, "Geometry Compression," SIGGRAPH, pp. 13-20, 1995.
- [11] G. Taubin and J. Rossignac, "Geometric Compression through Topological Surgery," ACM Trans. on Graphics, vol. 17-2, pp. 84-115, 1998.
- [12] C. Touma and C. Gotsman, "Triangle Mesh Compression," Proc. of Graphics Interface, pp. 26-34, 1998.
- [13] M. Isenburg and J. Snoeyink, "Compressing Texture Coordinates with Selective Linear Predictions," Proc. of Graphics Interface, pp. 126-131, 2003.
- [14] S. Y Kim, Y.S. Yoon, S.M Kim, K.H. Lee, and Y.S. Ho, "Texture Coordinate Compression for 3-D Mesh Models Using Texture Image Rearrangement," Lecture Notes in Computer Science, vol. 3767, pp. 687-697, 2005.
- [15] A. L Ames, D. P. Nadeau, and J. L. Moreland, VRML 2.0 Source Book, John Wiley, 1997.
- [16] G. Taubin, W. Horn, and F. Lazarus, The VRML Compressed Binary Format, ISO/IEC JTC1/SC29 WG11 m3062, 1998.
- [17] W. B. Pennebaker and J. L. Mitchell, The JPEG Still Image Data Compression Standard, Van Nostrand Reinhold, 1993.

저 자 소 개



김 성 열(학생회원)
 2001년 2월 강원대학교 정보통신 공학과 졸업(학사).
 2003년 2월 광주과학기술원 정보통신공학과 졸업(석사).
 2003년 3월~현재 광주과학기술원 정보통신공학과 박사과정.

<주관심분야: 디지털 신호처리, 삼차원 메쉬 모델링 및 압축, 삼차원 비디오 표현, 3차원 방송 및 실감방송>



호 요 성(평생회원)
 1981년 2월 서울대학교 전자공학과 졸업(학사).
 1983년 2월 서울대학교 대학원 전자공학과 졸업(석사).
 1983년 3월~1995년 9월 한국전자통신연구원 선임연구원.

1989년 12월 미국 University of California, Santa Barbara Department of Electrical and Computer Engineering(박사),
 1990년 1월~1993년 5월 미국 Philips 연구소 Senior Research Member.
 1995년 9월~현재 광주과학기술원 정보통신공학과 교수.

<주관심분야: 디지털 신호처리, 영상신호 처리 및 압축, 초저속 영상통신, 디지털 TV와 고선명 TV, MPEG 표준, 다시점 비디오 부호화, 실감방송>