

일반논문-08-13-1-13

패치 영상의 효율적 압축을 위한 전처리 방법

이신욱^{a)}, 이선영^{a)}, 장은영^{b)}, 허남호^{b)}, 장의선^{a)‡}

Preprocessing Method for Efficient Compression of Patch-based Image

Sinwook Lee^{a)}, Sunyoung Lee^{a)}, Eunyoung Chang^{b)}, Namho Hur^{b)}, and Euee.S Jang^{a)‡}

요 약

텍스처 영상 압축은 JPEG과 같은 영상 압축 표준을 사용한다. 일반 텍스처 영상에는 JPEG으로도 좋은 압축 성능을 내지만 패치 텍스처 영상은 이러한 영상 압축 기술로 압축했을 때 좋은 압축 성능을 보이지 못하였다. 이를 보완하고 압축 성능을 높이기 위해 기존의 패치 텍스처 영상에 주변 화소 간에 색차를 줄이기 위한 전처리 기술을 소개하고, 제안된 기술과 JPEG 압축을 순서대로 적용하였을 때 압축 성능이 23%에서 45%까지 향상됨을 보이고자 한다.

Abstract

In mapping a texture image into a 3D mesh model for photo-realistic graphic applications, the compression of texture image is as important as geometry of 3D mesh. Typically, the size of the (compressed) texture image of 3D model is comparable to that of the (compressed) 3D mesh geometry. Most 3D model compression techniques are to compress the 3D mesh geometry, rather than to compress the texture image. Well-known image compression standards (i.e., JPEG) have been extensively used for texture image compression. However, such techniques are not so efficient when it comes to compress an image with texture patches, since the patches are little correlated. In this paper, we proposed a preprocessing method to substantially improve the compression efficiency of texture compression. From the experimental results, the proposed method was shown to be efficient in compression with a bit-saving from 23% to 45%.

Key words: patch-based image, 3D mesh coding, texture image compression

1. 서 론

게임이나 애니메이션 영화, 산업용 그래픽 모형 등과 같은 분야의 3차원 그래픽 모델은 좌표 정보, 연결 정보, 측광 정보 등의 3차원 메쉬 정보와 텍스처 영상으로 현실과 같은 모델을 쉽게 표현할 수 있다. 3차원 모델을 더 정교하게 표

현하기 위해서는 메쉬 정보가 증가하거나 텍스처 영상의 질이 높아져야만 하고 그만큼 데이터량이 늘어나게 된다. 이처럼 정교한 3차원 모델 표현을 위해 방대해진 3차원 메쉬 데이터 정보를 압축하기 위해 지난 10여 년 동안 연구가 꾸준히 진행되어 왔다^[1]. 이 중 대부분의 연구는 3차원 메쉬 혹은 좌표 정보 압축에 치중되어 있고, 텍스처 영상에는 일반적인 영상 압축 표준이 주로 이용되고 있다. 예를 들어 2000년에 표준화 된 MPEG-4 mesh coding(3DMC)는 3차원 속성값들을 압축하는 표준을 제공하지만, 여기에 텍스처 영상 압축에 관한 표준은 포함되어 있지 않다.

a) 한양대학교 디지털 미디어 연구실

Hanyang University, Digital Media Lab.

b) 한국전자통신연구원 전파방송연구단 방송시스템연구그룹 3DTV시스템연구팀 ETRI

‡ 교신저자 : 장의선(eueesjang@ihanyang.ac.kr)

텍스처는 3차원 모델의 표면에 입혀질 이차원 비트맵 영상이다. 비트맵 영상을 표현하기 위한 대표적인 객체는 정지 영상이기 때문에 텍스처는 주로 정지 영상으로 표현된다. 텍스처는 크게 두 가지 유형으로 나누어 볼 수 있다. kj; 8 하나는 텍스처 아틀라스(texture atlas)로 조각난 영상의 차트를 하나의 텍스처 영상으로 붙여둔 것이고 다른 하나는 사각형 조각들의 모음으로, 텍스처 매개화(parameterization) 과정을 거쳐 생성된 텍스처 패치(texture patches)가 있다^[2]. 텍스처 영상은 앞에서 말한 바와 같이 일반적인 영상 압축 기술을 이용하여 압축된다. 텍스처 아틀라스의 경우 JPEG과 같은 기존의 영상 압축 기술로도 충분히 좋은 압축 성능을 보이지만 패치 기반 영상의 경우엔 그렇지 않다. 패치 기반 영상은 텍스처로 사용되는 작은 크기의 패치들로 구성되어 있기 때문에 패치와 패치 간에 화소들의 상호 연관성은 다른 일반 영상에 비해 낮다. 게다가 텍스처 매핑 과정에서 패치들이 서로 겹쳐지는 것을 피하기 위해 비사용 화소(unused pixels)를 사용하여 영상 패치들 사이에 필요 없는 공간을 허용하고 있지만 블록 크기를 기반으로 압축을 수행하는 영상 압축 방식의 특성 상 이 또한 압축 효율이 낮다.

저장이나 전송을 위해 3차원 모델을 추가적으로 압축해야 할 때, 메쉬 정보는 압축을 거듭하게 되면 모델의 전체적인 모양을 흐트릴 수 있지만 텍스처는 모델 표면의 화질 이외엔 영향을 끼치지 않는다. 따라서 추가적인 압축을 고려했을 때 3차원 메쉬 정보를 추가로 압축하는 것 보다는 텍스처 정보를 추가로 압축할 수 있는 여지가 많다. 추가적인 3차원 모델의 효과적인 압축을 위해서 메쉬 정보와 텍스처 정보의 압축률의 적절한 구성 비율 결정에 대한 통합적 연구가 필요하다.

이러한 통합적 연구는 시점에 의존적인 다중 해상도 텍스처 압축을 위한 지형 모델 연구 과정에서 이미 시도되었다^[3]. 메쉬와 텍스처의 효율적인 압축 방법은 다중 해상도뿐 아니라 3차원 텍스처 모델의 저장, 전송, 구현 등 모든 측면에서 필요하다. 예를 들어 시스템 자원(e.g., transmission and memory bandwidths)에 의해 현 상태의 3차원 모델의 구현이 제한적이라면 메쉬 정보나 텍스처의 추가적인 압축에 균형을 맞추어주는 것은 충분히 연구해볼 가치

가 있을 것이다. 하지만 메쉬 정보의 추가적 압축보다는 텍스처의 추가적 압축이 모델에 더 적은 영향을 미친다는 것은 자명하다. 우리는 예전에 영상 사이즈, 텍스처 패치의 크기 등의 패치 기반 텍스처 영상의 특징을 이용한 텍스처 좌표 압축을 연구한 바가 있다^[4]. 이 연구는 여전히 메쉬 정보 압축의 한 방법일 뿐 텍스처 영상 압축에는 아무런 연관을 가지고 있지 않다.

이 논문에서 우리는 현재 존재하고 있는 영상 압축 표준을 이용하여 패치 기반의 영상 압축을 향상시킬 수 있는 전처리 방법을 제안한다. 압축 향상은 패치 기반 텍스처 영상 내에 사용되지 않는 화소들과 주변 화소간의 연관성을 높여줌으로써 가능하다. 전처리 방법을 고안함에 있어서 복잡도를 주요 사항으로 두었다. 일반적으로 이러한 전처리 방법은 압축의 필수 과정은 아니기 때문에 처리 과정에서 높은 복잡도로 인해 전체적인 압축 과정에 부정적인 영향을 미쳐서는 안 될 것이다.

이 논문의 구성은 다음과 같다. 2장에서 우리는 패치 기반 텍스처 영상을 이용한 3차원 모델의 압축 과정이 일반적으로 어떻게 이루어지는가를 설명하고, 제안한 전처리 방법의 자세한 설명을 다음 장에서 다루었다. 4장에서는 제안한 방법을 적용, 실험한 결과를 다루었고, 5장에서는 결론을 지었다.

II. 패치 기반 텍스처 3차원 모델의 압축

3차원 텍스처 모델은 위에서 언급했듯이 메쉬 정보와 텍스처 데이터로 구성된다. 메쉬 정보는 연결 정보, 좌표 정보, 색 정보, 법선 정보, 텍스처 좌표 정보들을 포함할 수 있으며 텍스처 데이터는 텍스처 영상 등의 2차원 그래픽 객체로 이루어져 있다. 그림 1(b)와 그림 1(c)에서 보여지는 영상은 실제 텍스처로 사용되고 있는 영상이다. 그림 1(b)에서 보여지는 영상은 텍스처 아틀라스 영상의 한 예이고, 그림 1(c)의 영상은 패치 기반 텍스처 영상의 한 예이다. 그림 1(c)에서 보여지듯이 패치 기반 텍스처 영상은 각기 다른 사이즈의 텍스처 패치들이 모여서 이루어진다. 텍스처 아틀라스 영상에 비해 패치 기반 텍스처 영상의 차이점

은 텍스처 매핑에 의해 사용되는 패치들로 재구성하고 인위적으로 텍스처 매핑에 비사용 화소를 패치들 사이에 배치해 두었다는 것이다. 그림 1(d)에서는 그림 1(c)의 텍스처 영상에서 텍스처 매핑으로 사용되는 사용 화소 부분과 그렇지 않은 비사용 화소 부분으로 나눈 것이다. 그림 1(d)에서 보듯이 실제 비사용 화소 값이 패치들 사이사이에 배치되어 있음을 볼 수 있다. 비사용 화소로 패치들 간에 간격을 둔 것은 영상 공간을 낭비하고 있지만 복호화된 텍스처 좌표를 가지고 텍스처 매핑시 이웃 패치의 영역을 참조하는 것을 피하기 위한 것이다. 3DMC는 손실 압축이기 때문에

메쉬 정보가 3DMC에 의해 부복호화 단계를 거치게 되면 실수로 표현되는 좌표값들은 정밀도 상의 오류가 발생한다. 텍스처 좌표에 발생한 정밀도 오류는 참조하고자 하는 영상의 한 패치에서 오차만큼 다른 패치를 참조하게 될 수도 있다. 이러한 정밀도 오류의 영향을 감소시키기 위해서 비사용 화소로 패치들 간에 간격을 두는 것이 필요하다. 하지만 비사용 화소들을 포함시켜 문제점을 해결하고자 함에도 불구하고 여전히 정밀도 오류로 인한 손상된 렌더링 결과를 육안으로 확인할 수 있다.

메쉬 데이터와 텍스처 데이터의 통합적인 압축 방법이 어

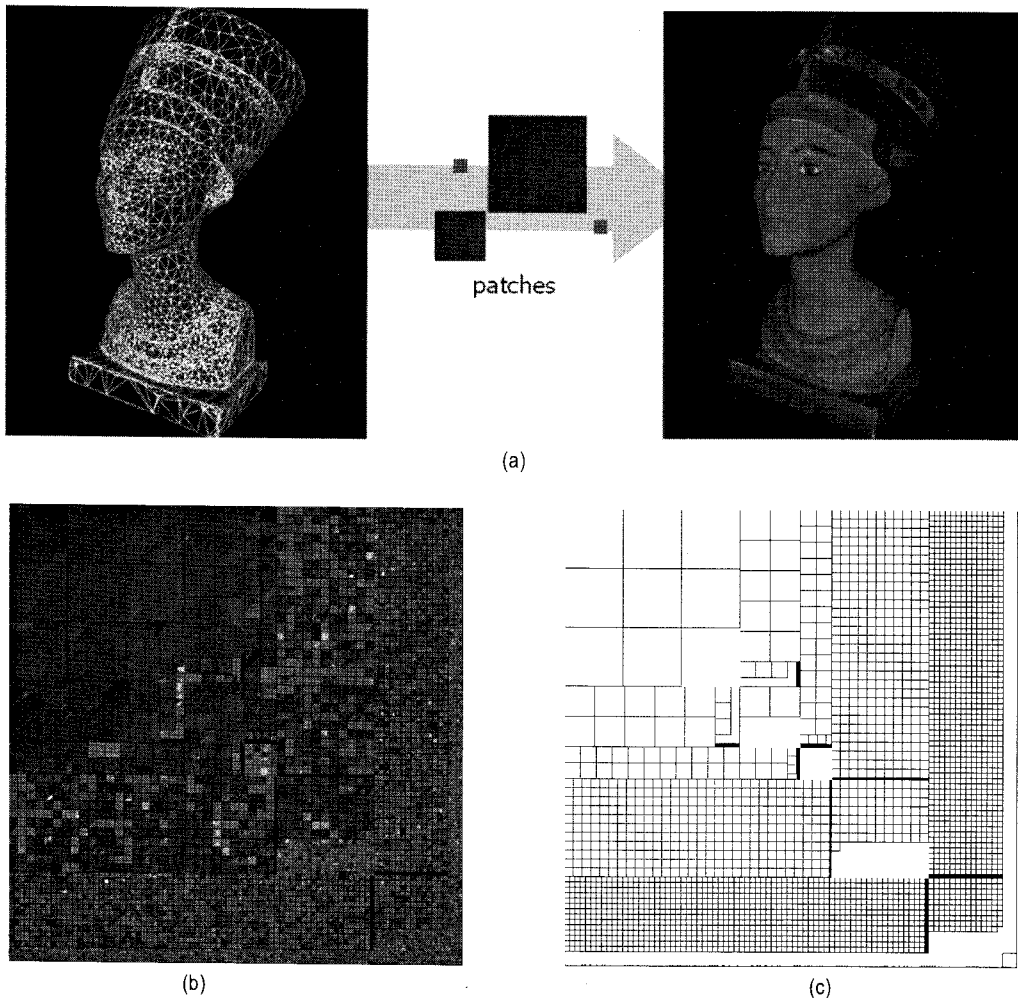


그림 1. 3차원 텍스처 모델의 예
Fig.1. Example of 3D textured model

떻게, 왜 필요한지를 알아내기 위해서 각각의 데이터에 대해 압축 전후의 데이터 용량을 비교하였다. 표 1에서 3차원 모델 데이터의 한 예를 보여주고 있다. 메쉬 데이터의 압축 이전의 3차원 모델(nefert131)은 VRML로 구성된 메쉬 데이터와 JPEG으로 압축된 텍스처 데이터를 가지고 있다. 메쉬 데이터는 텍스처 데이터보다 월등히 큰 용량이지만 MPEG-4 3DMC 표준으로 메쉬 압축을 수행하였을 때 압축된 메쉬 용량은 텍스처 데이터의 용량과 거의 비슷하다.

표 1. Nefert131 모델의 압축 전후로 메쉬와 텍스처 데이터의 용량 비교
Table 1: Size comparison of mesh and texture data before and after compression of nefert131

Mesh data (in VRML)	Texture	Compressed mesh data	Texture size /compressed mesh size
(in bytes)			(in percent)
2,389,478	194,622	198,690	98 %

표 2에서는 메쉬와 텍스처에 대해 추가적인 압축으로 인한 영향을 분석하기 위해 두 가지의 실험적인 압축 및 복원 과정을 거친 뒤 결과를 기술한 것이다. 첫 번째 경우는 메쉬 데이터에만 추가적인 압축을 수행한 것이고, 두 번째 경우는 텍스처 데이터에만 추가적인 압축을 수행한 것이다. 각각의 실험을 거친 후 렌더링 된 영상은 그림 2와 같다. 그림 2에서 보듯이 텍스처 영상에 추가적인 압축을 한 경우는 여전히 그 형태와 질감, 색감 등을 알아볼 수 있지만, 메쉬 데이터의 추가적 압축을 한 경우는 그렇지 못했다. 우리는 다른 텍스처 모델로도 이와 같은 실험을 거쳤지만 같은 결과를 얻을 수 있었다. 다시 말하면 텍스처 영상은 추가적인 압축을 위한 여지가 여전히 존재함을 의미한다. 이는 여러 가지 이유가 있을 수 있겠지만 그 중 하나가 패치 기반 텍스처 영상에는 비사용 화소의 개수가 예상보다 많이 존재하고 있다는 것이다. Nefert131의 경우 비사용 화소의 비율은 무려 17%나 되었고, 다른 모델의 경우 최고 21%까지도 존재하였다. 다른 이유는 메쉬의 압축은 모델의 형체를 일그러뜨리는데 반해 텍스처 영상의 압축은 영상의 질만을 떨어뜨리기 때문이다. 영상 화질의 감소는 메쉬 압축으로 인한 모델의 모양이 망가지는 것에 비했을 때 모델의 전체적인 모습에 영향을 덜 끼친다.

표 2. Nefert131 모델의 메쉬 및 텍스처 데이터에 대한 두 가지 극단적인 경우 실험

Table 2. Two extreme cases for mesh and texture compression (in bytes) of nefert131

Case	Compressed mesh size	Texture image size
1 (compress mesh more)	137,615	194,622
2 (compress texture more)	198,690	141,627



(a) Case 1

(b) Case 2

그림 2. 표 2에 기술된 두 가지 실험 결과로 렌더링된 결과
Fig.2. Rendered result of two extreme cases of Table 2

III. 패치 기반 텍스처 영상의 효율적 압축을 위한 전처리 방법

이전 장에서 우리는 텍스처 데이터의 추가적인 압축 가능성을 제기하였다. 패치 기반 텍스처 영상의 경우 패치 기반 영상에만 적용할 수 있는 압축 알고리즘을 새로 디자인 하여 현재보다 더 효율적인 압축을 기대할 수도 있다. 이러한 압축 방법은 현재 존재하는 영상 압축 방법을 대체하기 위함으로 기존의 영상 압축 방법과 혼용하기 위한 목적은 아니다.

패치 기반 영상의 효율적 압축을 위한 다른 방법은 현존하는 영상 압축 기술이 패치 기반 영상에 효율적으로 기능할 수 있도록 전처리 알고리즘을 디자인하는 것이다. 이 논문에서는 후자의 방법으로서 비사용 화소 값을 인접한 사

용 화소값으로 대체하는 패딩 알고리즘을 제안하였다. 이것은 주변의 화소값이 서로 연관성이 높으면 기존의 영상 압축 기술이 압축 효율과 영상 화질을 같이 높일 수 있다는 것에 착안하였다.

제안하는 방법은 이차원 화소 구분 지도 구성 과정, 비사용 화소의 값 대체 과정으로 나누어진다. 먼저 이차원 화소 구분 도표 구성 과정에서 텍스처 좌표 정보와 텍스처 영상으로부터 사용 화소와 비사용 화소를 구분하여 주는 이차원 도표를 도출해내고, 비사용 화소의 값 대체 과정에서 이 도표를 바탕으로 비사용 화소값을 주변의 사용 화소 값으로 대체하여 텍스처 영상을 재구성한다. 그림 3은 각각 과정의 흐름을 블록도로 나타낸 것이다.

그림 3에서 보는 것처럼 이차원 화소 구분 지도 구성 과정에서 텍스처 좌표 정보와 연결 정보로부터 도출되는 면의 모습을 텍스처 영상에 투영하여 텍스처 영상의 화소를 두 분류로 나눈다. 텍스처 영상에 투영된 면 안쪽으로 포함

되는 화소들을 사용 화소로, 포함되지 못한 화소들을 비사용 화소로 구분하여 저장하고, 이 정보로부터 이차원 화소 구분 지도 PM(i, j)를 구성한다. PM(i, j)를 구성하는 방법은 식(1)과 같다.

$$PM(i, j) = \begin{cases} 0: \text{used pel} & i=0..(Xsize-1), j=0..(Ysize-1) \\ 1: \text{unused pel} \end{cases} \quad (1)$$

단, 여기서 Xsize와 Ysize는 영상의 좌표축을 가리킨다.

이어서 구성된 PM(i, j) 지도를 바탕으로 텍스처 영상의 비사용 화소 T(n, m)를 인접한 사용 화소값으로 대체한다. 비사용 화소를 주변의 사용 화소값으로 대체하는 패딩 방법은 다양하게 존재할 수 있다. 제안된 방법에서는 MPEG-4 형상 부호화(Shape coding)에서 기술하고 있는 방법과 유사한 알고리즘을 사용하였다^[5]. 이외에도 선형 보간법(linear interpolation), 이선형 보간법(bilinear interpolation), 또는 평

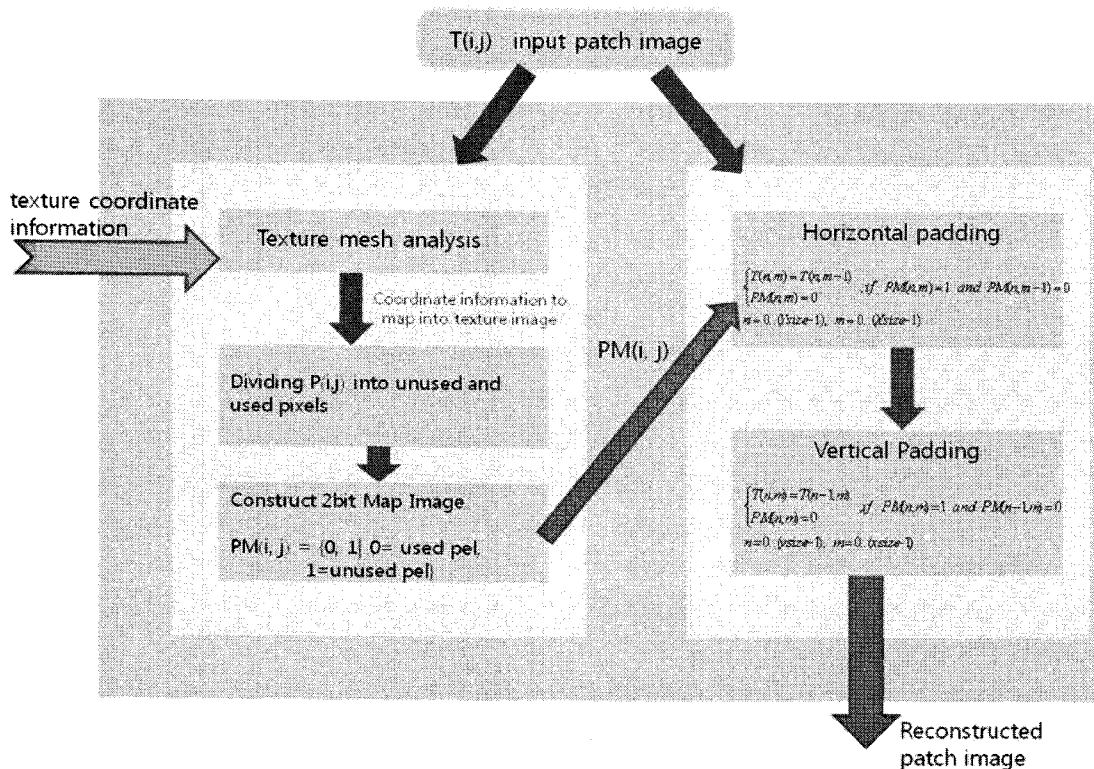


그림 3. 제안된 방법의 블록도
Fig.3. the block diagram of proposed method

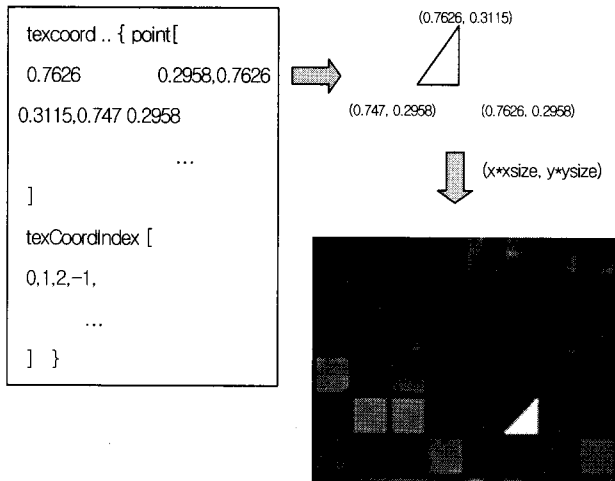


그림 4. 텍스처 영상에 메쉬 정보로부터 구성된 면의 대조 과정의 예
Fig. 4. Example of mapping texture mesh data into texture image

균값 등으로 비사용 화소값을 대체하는 방법이 존재하지만 복잡도 측면에서는 제안된 방법보다 떨어질 수 있고 제안된 방법보다 성능 향상을 보장할 수 없기 때문에 가장 단순하고 빠른 알고리즘을 선택하였다. 선택된 방법에서는 해당 조건을 만족하는 비사용 화소 $T(n, m)$ 에 먼저 수평 패딩(horizontal padding)을 수행한 후 나머지 비사용 화소에 수직 패딩(vertical padding)을 수행하는 두 가지 단계를 포함하고 있다. 각각의 과정이 수행되는 방법은 다음과 같다.

- Horizontal padding

$$\begin{cases} T(n,m) = T(n,m-1) \\ PM(n,m) = 0 \end{cases}, \text{ if } PM(n,m) = 1 \text{ and } PM(n,m-1) = 0 \quad (2)$$

$n = 0..(Ysize-1), m = 0..(Xsize-1)$

- Vertical padding

$$\begin{cases} X(n,m) = X(n-1,m) \\ PM(n,m) = 0 \end{cases}, \text{ if } PM(n,m) = 1 \text{ and } PM(n-1,m) = 0 \quad (3)$$

$n = 0..(ysize-1), m = 0..(xsize-1)$

그림 5는 제안된 방법으로 비사용 화소의 값을 대체하는 예를 보여주고 있다. 이러한 방법은 두 번의 반복 과정으로

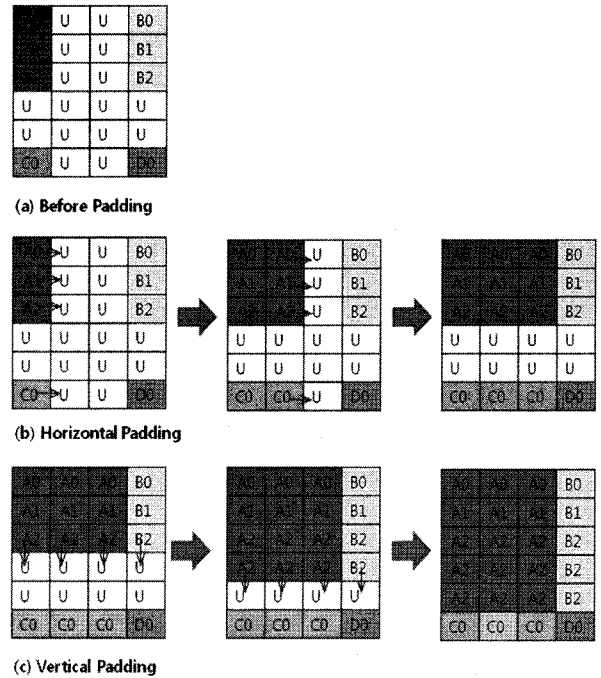


그림 5. 비사용 화소에 패딩 알고리즘을 적용한 예(U: 비사용 화소)
Fig. 5. Example of padding of unused pixels (U: unused pixel)

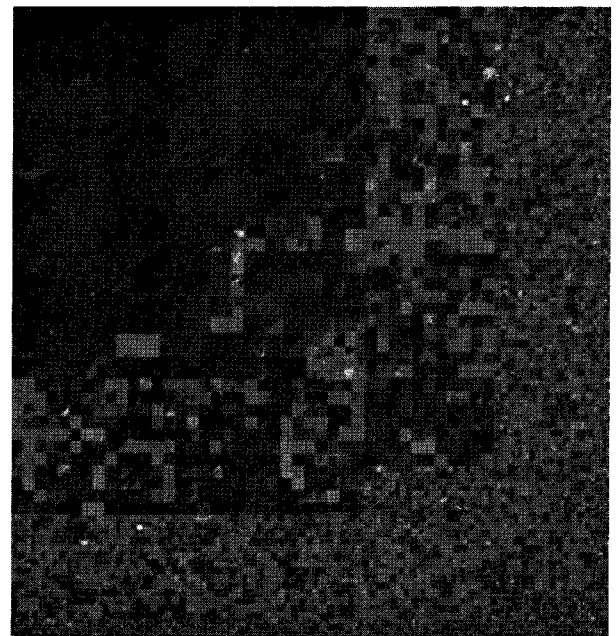


그림 6. 패딩 알고리즘을 적용한 패치 기반 텍스처 영상
Fig. 6. Patch-based image after padding

인한 시간적 낭비가 존재하지만 계산 과정이 필요치 않기 때문에 다양한 대체 방법 중에서 복잡도가 가장 낮다. 그림 6은 제안된 방법으로 처리 과정을 거친 텍스처 영상의 모습이다.

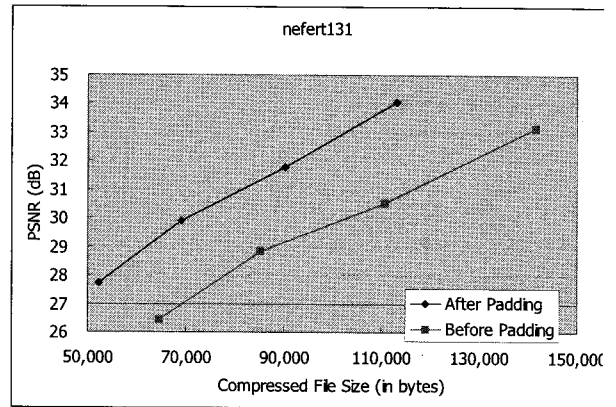
V. 실험 결과

우리는 제안된 전처리 방법을 표 3에 기술된 세 가지의 텍스처 모델에 적용하였다. 압축되기 전의 메쉬 데이터는 VRML 2.0 규격으로 기술되었고, MPEG-4 3DMC로 메쉬 압축을 하였다. 이 논문에서 사용된 모든 실험 모델들은 3DMC 개발 시에 MPEG에서 사용된 것들이고, 압축된 메쉬 데이터의 크기와 텍스처 영상 데이터의 크기가 3개의 실험 모델에서 대등함을 확인하였다.

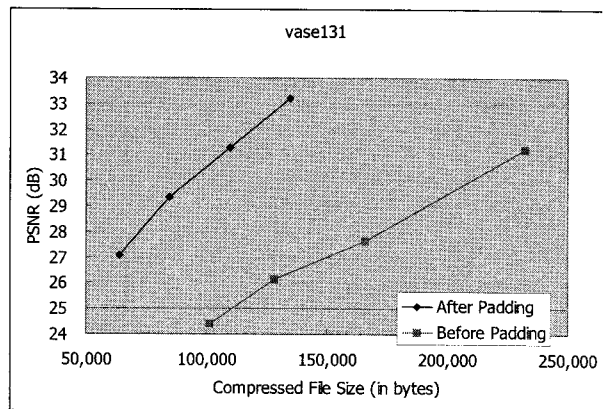
표 3. 실험 모델
Table.3. Test Models

Test models	Mesh size (in VRML)	Mesh size (in 3DMC)	Texture (in JPEG)
Nefert131	2,389,478	198,690	194,622
Vase131	2,404,860	201,811	296,187
Vase212	2,400,516	202,408	301,477

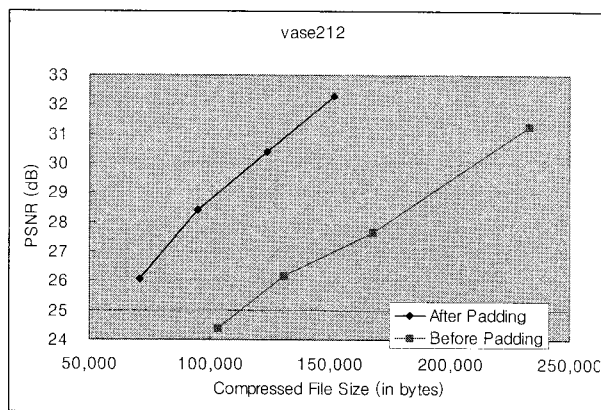
압축 효율을 측정하기 위해서 원본 텍스처 영상을 두 가지 방법으로 재압축하였다. 제안된 전처리 방식을 거친 후 텍스처 영상을 재압축한 영상과 전처리 방식을 거치지 않고 텍스처 영상을 재압축한 영상을 가지고 **rate-distortion performance**로 객관적인 실험을 측정하여 그림 7과 같이 도표로 나타내었다. 비사용 화소 부분은 측정 범위에서 제외하였으며, 오직 사용 화소 부분에 대해서만 원본 영상과 PSNR을 측정하였다. 측정에 사용된 모든 실험 모델에서 두 가지 방법으로부터 생성된 파일의 크기를 비슷하게 조절하였을 때, 제안된 방법이 예전의 압축 방법보다 작게는 2dB 크게는 7dB까지 화질이 향상되었다. 압축 성능 면으로



(a)



(b)



(c)

그림 7. 세 개의 실험 모델의 performance 측정
Fig. 7. Objective performance of three test models (anchor: before padding, proposed: after padding)

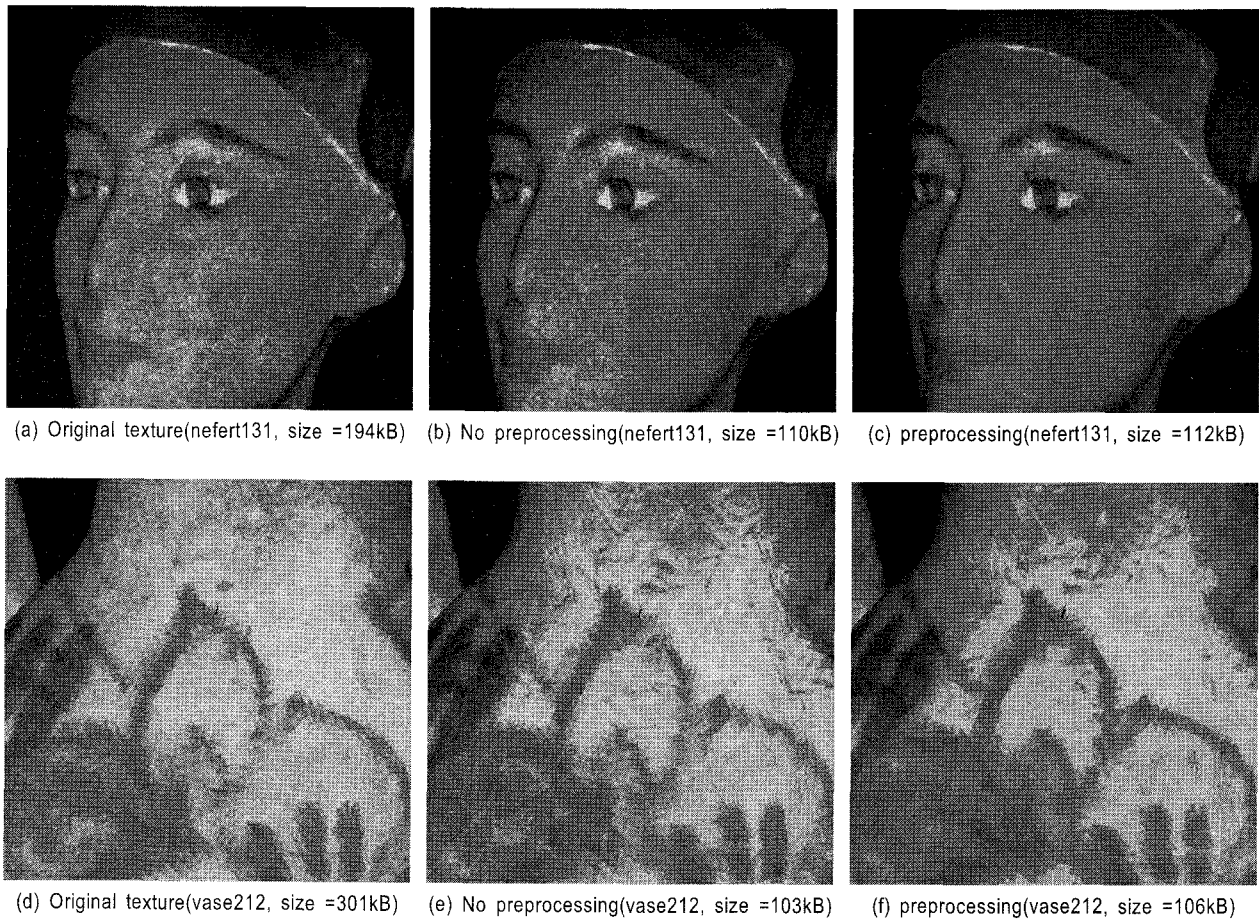


그림 8. 렌더링 후 스냅샷

Fig. 8. Snapshots from rendered models

보았을 때 실험한 모델 중 제안된 처리 방식을 거친 모델과 가장 차이가 적었던 nefert131b에서 23% 이상, 가장 차이가 컸던 vase131b에서 최대 45%까지의 압축 성능 차이를 보여주고 있다.

그림 8에서 보듯이 육안으로도 제안된 방법으로 압축한 텍스처 영상을 사용한 3차원 모델이 화질 면에서 향상되었음을 확인하였다. 이와 같은 실험 결과로 패치 기반 텍스처 영상에서 비사용 화소가 전처리 과정을 거친 뒤 압축하는 것이 모든 면에서 효율적임을 증명하였다.

패치 기반 텍스처 영상의 또 하나의 문제점이었던 메쉬 데이터의 정밀도 오류에 의한 잘못된 화소 참조도 전처리 과정을 통해서 어느 정도 해결할 수 있다. 그림 9(a)에서

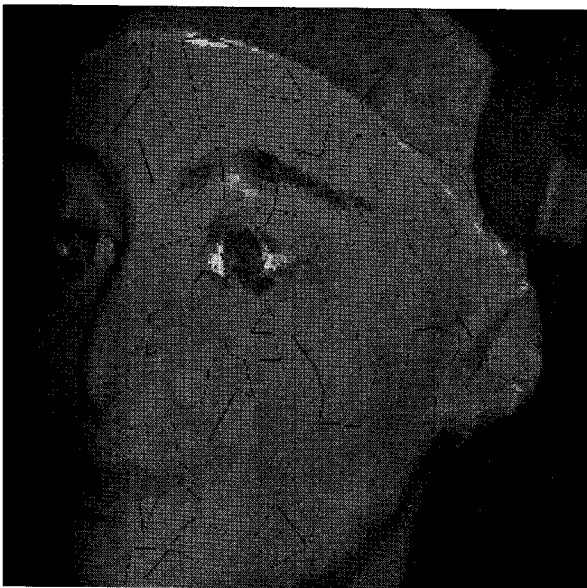
보여지듯이 메쉬 데이터를 MPEG-4 3DMC로 부복호화를 거치면 텍스처 영상을 가리키는 좌표 데이터의 정밀도 오류가 발생한다. 3DMC의 압축 방식이 손실 압축이기 때문에 본래 가리키고 있던 좌표의 위치에서 한 화소 이상 어긋날 수 있다. 전처리 알고리즘을 통해 이러한 정밀도 오류를 해결할 수는 없지만 다른 화소 부분을 참조할 경우 이 부분의 값을 유사한 색으로 대체함으로써 정밀도 오류를 육안으로 확인할 수 없도록 할 수 있다. 그림 9(b)는 패딩 알고리즘을 적용한 후의 패치 기반 텍스처 영상을 가지고 렌더링한 결과로써 그림 10(a)에서 보여지던 문제점을 많이 해결하였다.

VI. 결 론

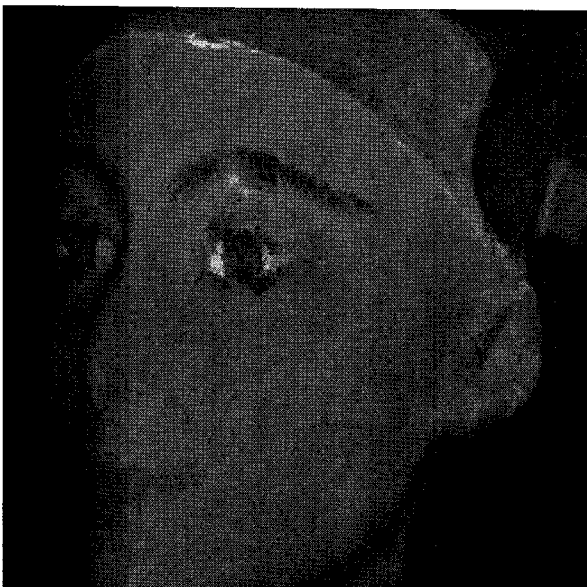
우리는 기존의 패치 기반 텍스처 영상의 문제점이었던 압축 효율 저하 및 렌더링 시 영상 손상 등을 해결하기 위해 패치 기반 텍스처 영상에 전처리 과정을 적용하는 방법을 제안하였다. 제안된 방법으로 기존의 패치 기반 텍스처 영상보다 같은 용량대비 적어도 2dB 이상 화질 개선 효과를 얻을 수 있었고, 압축 효율 면에서는 같은 화질 대비 적어도 23% 이상 개선할 수 있었다. 또한 본 논문에서 제안한 방식을 사용하면 텍스처 좌표의 압축으로 인해 발생하는 렌더링 화질 열화를 최소화시킬 수 있는 이점이 있다. 본 논문의 실험 결과를 통해 3차원 그래픽 모델 데이터의 압축 시 모델의 기하 정보의 압축뿐 아니라 텍스처 정보의 효율적인 압축이 중요함을 보이고자 하였다. 텍스처 기반 3차원 모델표현이 점차 일반화됨에 따라 더 효율적인 기하 및 텍스처 정보의 압축 기술에 대한 연구가 계속되어야 할 것이다.

참 고 문 헌

- [1] G. Taubin and J. Rossignac, "Geometric Compression through Topological Surgery," in ACM Transactions on Graphics, 1998.
- [2] Sung-Yeol Kim and Yo-Sung Ho, "텍스처 영상 재배열을 이용한 삼차원 텍스처 좌표 부호화," 대한전자공학회 논문지, 제 43권 SP편 6호, 2006. 11.
- [3] S. Horbelt, L. Balmelli, and M. Vetterli, "Joint Mesh and Texture Compression using Marginal Analysis," in Proceedings of International Conference Image Processing, Vol. 3, pp.114-117, Oct. 2001.
- [4] Sunyoung Lee, Byeongwook Min, Daiyong Kim, Eun-Young Chang, Namho Hur, Soo Lee, and Euee S. Jang, "An Adaptive Quantization Scheme for Efficient Texture Coordinate Compression in MPEG 3DMC," in PCM 2005, pp. 73-83, 2005.
- [5] Fernando Pereira and Touradj Ebrahimi, The MPEG-4 Book, Prentice Hall, 2002
- [6] Aaron E. Walsh and Mikael Bourges-Sevenier, MPEG-4 Jump-Start, Prentice Hall, 2002



(a) Without padding(decoded mesh + original texture)



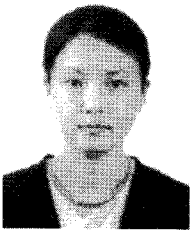
(b) With padding(decoded mesh + preprocessed texture)

그림 9. 정밀도 오류를 가진 텍스처 좌표로 렌더링 화질 비교
Fig. 9. Rendering quality comparison with texture coordinate precision errors

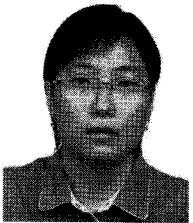
 저 자 소 개

**이 신 욱**

- 2005년 : 동국대학교 컴퓨터공학과 학사
- 2007년 : 한양대학교 전자통신컴퓨터공학과 석사
- 2007년 ~ : 한양대학교 전자통신컴퓨터공학과 박사 과정 중
- 주관심분야 : 영상처리, 3D mesh coding, Reconfigurable Video Coding

**이 선 영**

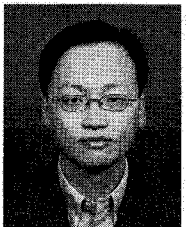
- 1996년 : 숭실대학교 전산학과 학사
- 2005년 : 한양대학교 정보통신공학과 석사
- 2006년 ~ 현재 : 한양대학교 전자컴퓨터통신공학과 박사과정
- 주관심분야 : 멀티미디어, 영상처리, 동영상 부호화, MPEG-4 국제 표준화

**장 은 영**

- 1999년 : 전북대학교 정보통신공학과 학사
- 2001년 : 광주과학기술원 정보통신공학과 공학석사
- 2001년 ~ 현재 : 한국전자통신연구원 전파방송연구단 방송시스템연구그룹 3DTV시스템연구팀 연구원
- 주관심분야 : 3D Graphics Model 및 3D 비디오 압축/처리, 3DTV, Free-viewpoint TV

**허 남 호**

- 1992년 2월 : 포항공과대학교 전기전자공학과 학사
- 1994년 2월 : 포항공과대학교 대학원 전기전자공학과 석사
- 2000년 2월 : 포항공과대학교 대학원 전기전자공학과 박사
- 2000년 4월 ~ 현재 : 한국전자통신연구원 전파방송연구단 방송시스템연구그룹 3DTV시스템연구팀
- 주관심분야 : 3DTV, 3D DMB, Free-viewpoint TV

**장 의 선**

- 1991년 : 전북대학교 전자계산기공학과 학사
- 1994년 : State University of New York at Buffalo, Electrical and Computer Engineering 석사
- 1996년 : State University of New York at Buffalo, Electrical and Computer Engineering 박사
- 1996년 : US Army Research Lab, Adelphi, MD Research Associate
- 2002년 : 삼성종합기술원 책임연구원
- 2002년 ~ 현재 : 한양대학교 전자통신컴퓨터공학과 부교수
- 주관심분야 : 멀티미디어, 3D Graphics, 동영상 부호화, Reconfigurable Media Coding