다층 변위 맵의 비손실 압축과 렌더링※

전영재°, 김해동*, 조성현**
숭실대학교 미디어학부°, 한국전자통신연구원*, 홍익대학교 게임학과**
dkreformer@ssu.ac.kr, hdkim@etri.re.kr, scho@hongik.ac.kr

Lossless Compression and Rendering of Multiple Layer Displacement Map

Young-Jae Chun°, Hae-Dong Kim*, Sung-Hyun Cho**
Dept. of Media, Soong-sil University, Electronics and Telecommunications Research Institute, School of Games, Hongik University

요 약

다층 변위 매핑 기법은 여러 층으로 구성된 변위 맵들을 사용하여 단층의 변위 맵으로 표현할 수 없는 복잡한 형태를 재현할 수 있으며, 영화와 게임 등의 디지털 콘텐츠에서 보다 사실적인 표현을 저비용으로 제공할 수 있다. 하지만 다층 변위 맵 자료 구조의 특성상 표현을 자세히 할수록 여러 층의 변위 맵이 필요해지고 하단부의 변위 맵일수록 변위 정보를 저장하고 있지 않은 공간이 많이 생긴다. 본 논문에서는 저장 공간을 비효율적으로 사용하는 다층 변위 맵을 비손실 압축하고 렌더링하는 방법을 제안한다. 제안하는 렌더링 방법은 비손실 압축 변위 맵 정보를 사용하기 때문에 원본 변위 맵의 렌더링 결과와 동일한 화질을 보장한다.

ABSTRACT

Multiple layer displacement mapping methods are able to represent more complex and general geometries which cannot be presented by single layer displacement mapping methods, and provide a realistic scene to digital contents such as 3D games and movies with relatively low costs. However, as we use more layers for details, data space is wasted more because lower layers have less displacement data than higher layers. In this paper, we suggest a lossless compression and rendering method of a multiple layer displacement map. Since we compress the map without data loss, the proposed method provides the same quality as the rendering result that uses an original multiple layer displacement map.

Keyword : Displacement mapping, Lossless image compression, Real-time rendering

접수일자 : 2009년 11월 01일
심사완료 : 2009년 12월 02일
교신저자 : 조성현
※ 본 연구는 문화체육관광부 및 한국기술평가관리원(MCST/KEIT)의 IT 연구개발 프로그램의 일환으로 수행하였음. [2006-S-045-1, 기능 확장형 초고속 렌더러 개발]
※ 이 논문은 2007년도 홍익대학교 학술연구진흥비에 의하여 지원되었음.

1. 서 론

변위 매핑 기법은 단순한 평면 등의 기하에 변위 맵이라는 특정 자료 구조를 사용하여 복잡한 표면을 표현해주는 기법이며 많은 수의 정점을 사용하지 않고도 현실과 유사한 차폐, 음영 등의 효과를 제공해준다.

게임 등에서 많이 사용되고 있는 범프 매핑(Bump Mapping)은 법선정보를 사용하여 음영 계산을 보다 사실적으로 수행해주는 방식이다. 이와 달리 릴리프 매핑(Relief Mapping)[1], 시차 폐색 매핑(Parallax Occlusion Mapping)[2] 그리고 이미지 피라미드 변위 매핑[3]등은 변위 맵의 텍스처 공간에서 시점 광선을 진행하여 변위 맵과의 교차점을 구하는 방식으로 변위의 모습을 표현하였으며 음영 뿐 아니라 차폐까지 표현하였다. 각 변위 매핑 방식에서 교차점을 구하는 방법은 선형 탐색과 이진 탐색 그리고 계층적 탐색 등 여러 가지 방법이 있고 탐색 방법에 따라 정확성, 속도 등이 각각 다르다.

변위 매핑에서는 변위 맵의 자료 구조에 따라 표현 가능한 범위가 달라지는데 이는 변위 맵으로 표현하고자 하는 모양의 수직적, 수평적 복잡도에 따라 변한다. 수평적으로 보다 자세한 표면을 표현하기 위해서는 변위 맵의 해상도를 보다 높게 사용해야 하며 수직적으로 보다 자세한 표현을 하기 위해서는 더 많은 층의 변위 맵이 필요하다. 일반적으로 단층의 변위 맵은 기본 표면으로부터 솟아나온 형태를 표현하기 위해 사용되므로 모든 지점에 변위 정보를 저장해야 한다. 하지만 다층의 변위 맵은 공중에 떠 있는 즉, 기본 표면과 붙어 있지 않은 형태를 표현하기 위한 자료 구조이므로 변위 맵의 각 지점의 모든 층에 변위 정보를 가지고 있을 필요가 없다. 따라서 실제 변위 맵 렌더링 시에 사용되지 않는 부분이 많이 생기게 되는데 이미지 비손실 압축을 통한 저장 공간 확보의 여지가 있다.

우리는 이 논문에서 다층 변위 맵의 비손실 압축방식과 압축된 결과를 GPU에서 해제하여 변위 맵을 렌더링 하는 방법을 제안한다. 다층의 변위를 표현할 수 있는 기존의 연구 중에는 수많은 방향에서의 깊이정보를 측정 사용하는 방법[4,5]과 다층 변위 맵을 사용하여 렌더링하는 방법[6,7]이 있었다. [4,5]에서는 방대한 자료를 압축하기 위하여 손실 압축을 사용하였고 매우 복잡한 압축 해제 연산이 요구되었다. [7]에서는 비교적 높은 화질을 보이나 층이 늘어날 경우 층의 수에 비례해서 메모리 사용량이 증가하는 단점이 있다.

본 논문에서는 기존의 연구와 달리 다층 변위 맵을 비손실 압축하고 렌더링한다. 다층 변위 맵의 특성 상 변위 정보가 저장되지 않은 부분이 많기 때문에 우리는 실제로 사용되는 변위 정보만 변위 맵의 각 위치 단위로 패킹(packing)하고 이렇게 압축된 변위 정보에 접근하기 위한 인덱스(index)를 저장한 주소 맵을 생성한다. 주소 맵은 원본 다층 변위 맵과 동일한 크기로 각 지점은 압축된 변위 정보가 저장된 위치를 가리킨다. 우리가 제안하는 방법은 실제 변위 맵을 사용한 경우와 비교하여 화질에 차이가 없으며 압축된 정보를 GPU상에서 해제하는 과정은 주소 맵을 통한 간접적인 접근 1회로 수행 과정에 큰 부담을 주지 않는다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해서 소개하고 3장에서는 다층 높이 맵의 필요성과 이 논문에서 제안하는 압축 방법에 대해 설명한다. 4장에서는 압축된 변위 맵 정보를 사용한 광선 탐색 과정에 대해 설명하며 5장에서 실험 결과를 서술한다. 마지막으로 6장에서 결론과 향후 과제에 대하여 논의한다.

2. 관련연구

GPU를 사용한 실시간 변위 매핑은 다양하게 연구가 진행되었다[1,2,3,4,5,6,7]. 릴리프 매핑[1]과 시차 폐색 매핑[2]은 광선 탐색 방식이 간단하여 빠른 렌더링 속도를 보장하지만 카메라의 시점이 지

표각(Grazing angle)에 가까워짐에 따라 심각한 시각적 오류가 발생한다. 선형 탐색과 이진 탐색으로 이루어진 광선 탐색은 변위와의 올바른 교차점을 찾지 못하는 경우가 존재하기 때문이다. 이런 문제를 해결하기 위해 이미지 피라미드 변위 매핑[3]은 변위 맵을 사진 트리(Quad Tree)로 재구성하고 부모 노드는 4개의 자식 노드들 중 가장 큰 변위 값을 저장하게 하였다. 광선 탐색 시 사진 트리를 탐색할 경우 시점 광선은 해당 레벨의 자식 노드 중 가장 높은 변위와 만나게 되므로 광선이 교차점을 지나치거나 찾지 못하는 문제는 해결된다. 변위 맵을 사용하지 않고 기하의 모양 자체를 3차원 텍스쳐에 저장한 다음에 그리는 연구도 많이 이루어졌다[4,5,8,9]. 이 방법들은 단층 높이 맵으로 구성되는 표면보다 더 복잡한 형태를 표현할 수 있으나 저장 공간 또는 처리 속도에 문제가 있다. 특히 [4]와 [5]는 SVD(Singular Value Decomposition)를 사용하여 방대한 변위 맵 정보를 압축하였으나 손실 압축 방식인 SVD의 특성상 최종 렌더링 결과는 실제 변위 정보를 전부 사용했을 때와는 달라진다. 또한 SVD로 압축된 변위 정보를 GPU에서 해제하기 위해서는 복잡한 해제 과정을 거쳐야 한다.

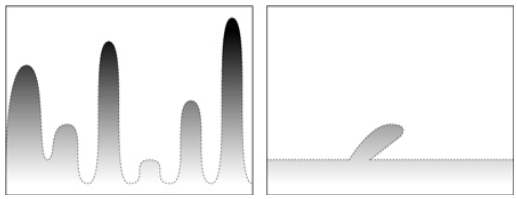
다층 변위 맵은 단층의 변위 맵을 사용할 경우 표현할 수 없는 복잡하고 오목한 물체를 표현하기 위해 동일한 크기의 변위 맵을 여러 층으로 사용하는 자료 구조이다[6,7]. Policarpo 등[6]이 제안한 다층 변위 맵은 4개의 층으로 이루어져 있으며 한 지점에 존재하는 변위의 크기 순서대로 변위 정보를 각 층에 저장한다. 그리고 광선 탐색 시에 현재 광선이 변위 정보의 내부에 존재하는지 검사하여 외부에 위치한 경우에만 광선을 진행시키는 방식으로 탐색한다.

보다 범용적인 다층 변위 렌더링을 하기 위해서는 더 많은 층수의 변위 맵을 사용할 수 있으나 추가적인 저장 공간을 필요로 하고 광선 탐색 시 참조해야 하는 변위 정보가 늘어나므로 속도는 저하된다. 또한 다층 변위 맵의 특성상 변위의 크기 순서대로 저장을 하기 때문에 변위 맵 층의 개수가 증가할수록 사용되지 않는 공간이 늘어날 수밖에 없다. 본 논문에서는 실제 사용되는 정보에 비해 많은 저장 공간을 사용하는 다층 변위 맵의 특성에 따른 저장 공간의 낭비를 막기 위해 다층 변위 맵 압축 방법을 제안하며 압축된 변위 맵 자료 구조를 사용한 광선 탐색 방식을 설명한다.

3. 다층 변위 맵

3.1 다층 변위 맵의 필요성

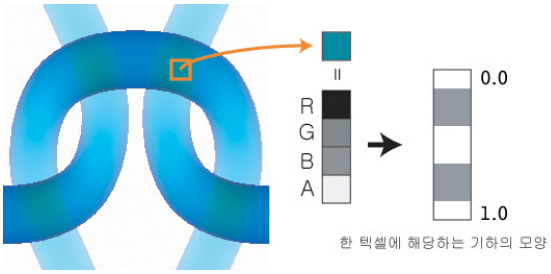
최근의 실시간 변위 매핑 기법은 다수의 정점으로 이루어진 가상의 물체를 그리는 대신에 회색조의 변위 맵을 사용, 텍스쳐 공간에서의 시점 광선 탐색을 통해 복잡한 기하를 표현할 수 있는 유용한 기법이다. 하지만 회색조로 제작된 단층의 변위 맵은 한 지점에 대해서 한 개의 변위 값만을 저장하기 때문에 더욱 복잡한 형태 혹은 특수한 형태의 기하를 표현 할 수 없다는 것이 단점이다. [그림 1]의 왼쪽 그림은 복잡해 보이지만 단층의 변위 맵으로 표현이 가능하다. 하지만 오른쪽 그림의 경우 돌출된 기하를 표현하기 위해서는 기하가 존재하는 변위 맵 지점에 대해 최소 3개의 변위 정보가 저장해야 한다. 따라서 단층의 변위 맵은 오른쪽 그림과 같은 기하를 변위 매핑할 수 없다.



[그림 1] 변위 모양의 종류

앞서 소개된 Policarpo 등[6]은 [그림 2]와 같이 4개의 층으로 구성된 변위 맵을 사용하여 단층의 변위 맵을 사용한 경우보다 더 복잡한 물체를 그

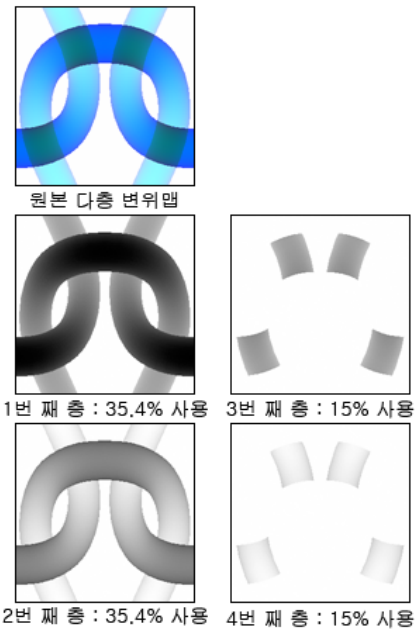
릴 수 있었지만 광선 탐색 방식이 기본적으로 릴리프 매핑[1]의 방식을 따르므로 시각적인 오류를 유발할 수 있다. 이런 시각적 오류를 개선하기 위해 다층 변위 맵을 계층적으로 재구성하여 밉맵(Mip-Map)과 유사한 방식으로 정리한 연구도 있었다[7]. 다층 변위 맵은 표면에서부터 임의의 기하까지의 변위 정보를 앞뒷면의 상태는 가리지 않고 순서대로 저장한 자료구조다. 따라서 다층 변위 맵의 각 높이 정보는 한 쌍씩 임의의 기하의 앞면과 뒷면의 변위 정보를 저장하여 기하 내부를 표현할 수 있다. [그림 2]와 같이 한 지점의 4개의 채널에 저장된 변위 정보는 R채널과 G채널이 그리고 B채널과 A채널이 각각 기하의 덩어리를 이룬다. 한 지점의 변위 정보에 따른 기하의 측면은 [그림 2]의 오른쪽과 같다.



[그림 2] 다층 변위 맵의 구조

3.2 다층 변위 맵의 압축

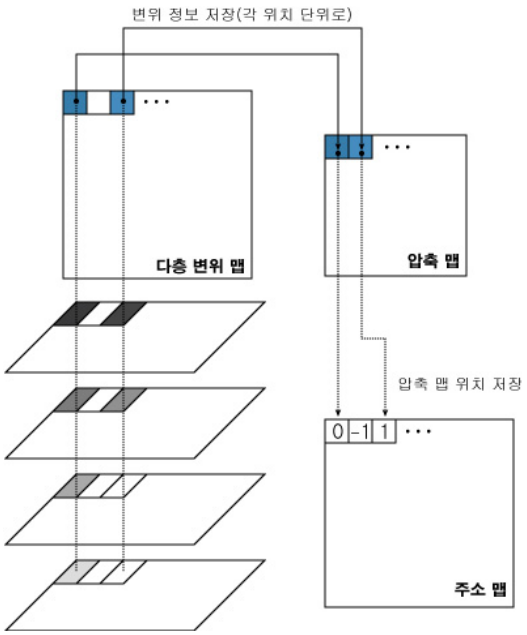
다층 변위 맵은 표현하고자 하는 모양의 수직적인 정밀도에 따라 많은 층을 필요로 한다. 또한 다층 변위 맵은 각 층이 고정된 동일한 크기로 구성되어 있으므로 한 층에 매우 적은 변위 정보가 존재할지라도 정해진 자료 공간을 제공하여야 한다. 이 경우 변위 정보가 없는 변위 맵의 많은 부분에 대해서는 저장 공간을 제공할 필요가 없다([그림 3]). 이러한 이미지 상의 자료 공간의 낭비를 개선하기 위한 비손실 이미지 압축 방식은 많이 연구되어왔다. Lefebvre 등[10]은 완전 해싱을 사용한 비손실 이미지 압축 방식을 소개하였다. 이 방식은 높은 압축률을 제공하지만 다층 변위 맵으로의 적용은 비효율적이다. 다층 변위 맵의 각 층 별로 압축을 수행해야 하며 압축되는 변위 정보의 개수에 따라 자료 구조의 크기와 해싱을 통한 접근법이 각각 달라진다. 이러한 방식으로 압축한 정보를 GPU에서 해제하는 경우 해싱을 위해 각각의 인자들을 저장하는 공간을 할당해야 한다. 또한 GPU에서 수행되는 텍스쳐 참조는 고비용의 작업에 속하는데 기존의 변위 매핑에서 수행되는 텍스쳐 참조 횟수와 비교하여 더욱 많이 참조를 하게 된다.



[그림 3] 저장 공간이 낭비되는 다층 변위 맵의 예

우리는 텍스쳐 참조 횟수를 최대한 줄이면서도 다층의 변위 정보를 압축하는 방법을 제안한다. 우리는 다층 변위 맵의 변위 정보를 저장하기 위한 압축 맵과 압축된 변위 맵의 각 지점에 접근하기 위한 주소 맵을 사용한다. 각 맵은 GPU에서 사용하기 위해 2차원 텍스쳐의 형태로 구성된다. 압축 맵의 크기는 입력 다층 변위 맵의 최상위층에 존재하는 변위 개수를 고려하여 최적화하며 다층 변

위 맵과 층수는 동일하게 생성한다. 압축된 변위 정보를 찾아가기 위한 주소 맵은 입력 다층 변위 맵과 동일한 크기를 가지며 압축 맵의 정보를 찾아가기 위한 주소정보만을 저장하므로 단층으로 되어있다. 우리의 변위 맵 압축 과정은 매우 간단하다. [그림 4]와 같이 다층 변위 맵의 각 지점에는 변위 정보가 있는 층과 없는 층이 존재한다. 우리는 압축을 위해 입력 다층 변위 맵의 각 위치를 차례대로 조사하면서 현재 위치의 각 층에 변위 정보가 존재하는지 확인한다. 만약 변위 정보가 존재한다면 해당 위치의 각 층의 변위 정보를 압축 맵에 전부 저장해나간다. 그리고 압축 맵에 저장되는 순서를 인덱스의 형태로 주소 맵에 저장한다. [그림 4]에서 입력 다층 변위 맵의 두 번째 위치에는 변위 정보가 하나도 존재하지 않으므로 압축 맵에 저장하지 않았으며 해당 위치의 주소 또한 -1로 저장하여 변위가 존재하지 않음을 복원 시 알 수 있게 한다.



[그림 4] 다층 변위 맵 압축 방식

여기서 우리는 변위 맵의 각 지점의 모든 층을 조사하여 단 한 개의 변위 정보라도 존재하면 해당 지점의 모든 층의 정보를 전부 저장한다. 만약 비어있는 층의 정보를 저장하지 않고 렌더링에 사용 될 변위 정보만 저장한다면 압축률이 비교적 좋아질 수 있다. 하지만 이런 경우 압축 맵의 한 지점에 여러 지점의 변위 정보가 저장될 수 있고 복원을 위해서는 한 지점의 각 층에서 저장된 변위의 개수를 추가적으로 저장해야 한다. 또한 GPU에서 압축을 해제하는 경우에 변위의 개수를 검사하고 변위 정보 획득을 위해 다시 텍스쳐를 참조하는 등, GPU에서 고비용의 작업에 속하는 텍스쳐 참조 작업이 증가하여 렌더링 속도를 저하시킬 수 있다. 따라서 우리는 압축률을 다소 낮추는 대신에 해제 속도를 증가시켜 최종 렌더링 속도가 저하되지 않도록 하였다.

압축을 완료한 다층 변위 맵의 모양은 [그림 5]의 왼쪽 그림과 같다. [그림 5]의 오른쪽 그림은 주소 맵(가운데 그림)을 사용하여 압축을 해제한 다층 변위 맵을 보여주고 있지만 실제로는 광선 탐색 시에는 압축을 전부 해제하지 않고 필요에 따라 임의의 위치에 대한 변위 정보만 해제하는 방식으로 사용한다.

[그림 5] 압축 맵, 주소 맵, 압축 해제된 다층 변위 맵

4. 압축된 다층 변위 맵에서의 광선탐색

4.1 압축된 다층 변위 맵 접근

우리는 압축 맵과 주소 맵을 텍스쳐의 형태로 GPU에 전달하여 광선 탐색 시 참조할 수 있도록 하였다. 언급한 바와 같이 GPU에서는 압축 맵을 해제하지 않고 광선 탐색 중 변위 정보를 필요로

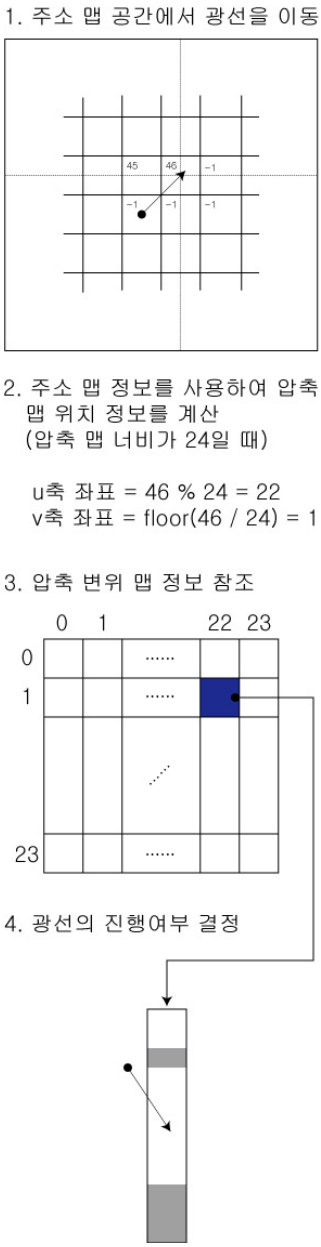
할 때 마다 주소 맵을 통해 간접적으로 접근한다. 변위 정보 접근 방법은 [표 1]의 쉐이더 코드와 같다. 여기서 주소 맵과 원본 변위 맵의 각 위치가 일대일로 대응하므로 광선이 탐색하는 텍스쳐 공간 역시 동일하다. 우리는 변위 매핑을 위해 광선 탐색을 할 때에 먼저 광선의 위치에서 주소 맵을 읽은 후 높이를 비교할 수 있는 변위가 존재하는지 확인한다. 만약 주소 맵에서 읽어온 주소가 -1이라면 변위 정보가 없는 것으로 간주한다. 만약 0 이상의 주소를 가지고 있다면 변위가 존재하는 것이고 저장된 주소를 사용하여 압축 맵에서 변위가 저장된 위치를 계산한다. 압축 과정에서 변위를 저장한 순서대로 주소 값을 늘려나가므로 읽고자 하는 변위의 u축의 값은 '읽어온 주소 값'과 '압축 맵 너비'의 나머지 연산을, v축의 값은 나누기 연산을 한 결과로 각각 구할 수 있다.

[표 1] 압축된 다층 변위 맵 정보 접근을 위한 쉐이더 코드

```
...
float4 depth = 1.0f;
float2 tex;
float addr = tex2D(addressSampler, ray.xy);
if(address >= 0)
{
    tex.x = addr % WidthOfCompressMap;
    tex.y = addr / WidthOfCompressMap;
    tex /= WidthOfCompressMap;
    depth = tex2D(compressSampler, tex);
}
...
```

4.2 광선 탐색

기본적으로 광선의 높이와 변위 값들과의 비교, 그리고 광선의 이동 방식은 이미지 피라미드 방식에 기반 한 다층 높이 맵의 렌더링[7]과 유사하다. 다만, 이전의 방식과 달리 다층 변위 맵을 압축함에 따라 계층적인 재구성이 불가능해졌기 때문에 계층적 광선 탐색이 아닌 선형적 광선 탐색을 수행한다. 우리는 광선의 위치와 광선의 방향 정보를 입력으로 받아 변위 맵의 텍스쳐 공간에서 광선 탐색을 수행한다. 여기서 사용되는 광선 탐색 방식은 광선의 시작 지점에서부터 광선의 방향으로 광선의 위치를 이동시켜 나가면서 만나는 모든 변위와의 높이 비교를 통해 광선을 전진시킨다. 기존의

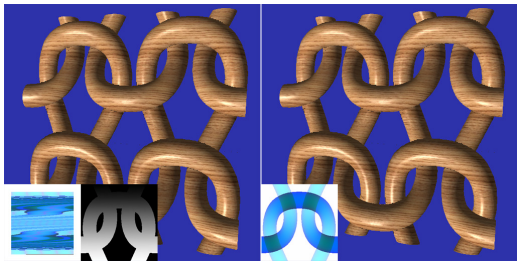


[그림 6] 광선 탐색 및 압축 해제 과정

방법과 다른 점은 광선이 이동을 위해 현재 지점의 변위 값을 참조할 때 주소 맵을 통해 간접적으로 접근한다는 점과 사진 트리 형식의 탐색을 하지 않는다는 점이다. 이 광선 탐색에서 가장 많은 작업을 수행하는 부분은 광선의 현재 위치 P가 변위 정보로 구성되는 기하의 내부에 존재하는지 판단하는 부분이다. 이 부분에서 압축된 변위 맵의 해제가 이루어지며 [그림 6]과 같은 과정을 거쳐 현재 위치의 변위 정보를 획득할 수 있다. 순서는 다음과 같다.

1. 주소 맵 공간에서 광선을 이동시킨다.
2. 이동한 지점의 주소 맵 정보를 참조하여 압축된 변위 맵에서의 좌표를 계산한다.
3. 압축 변위 맵의 좌표에서 변위 정보를 얻어 온다.
4. 현재 광선의 위치가 변위의 내부인지 판단하여 광선의 진행 여부를 결정한다.

[그림 7]의 왼쪽 그림은 압축된 변위 정보를 사용하여 렌더링 한 다층 변위 매핑 결과를 보여주고, 우측 그림은 원본 다층 변위 맵을 사용한 렌더링 결과를 보여 준다. 다층 변위 맵의 압축이 비손실로 이루어졌기 때문에 원본 다층 변위 맵을 사용한 경우와 시각적으로 전혀 차이가 없다.



[그림 7] 압축 맵과 주소 맵을 사용한 렌더링 결과

5. 실험 결과

본 논문에서는 DirectX 9.0을 이용하였으며, 시스템은 Nvidia GeForce 9800 GX2 그래픽 카드와 Intel CPU 2.13GHz를 사용하였다. 이 논문에서 사용한 [그림 2]의 다층 변위 맵을 기준으로 렌더링을 위한 자료의 크기는 압축 맵이 35.4%, 주소 맵이 50%로 원본 데이터의 85.4%이므로 총 14.6%의 자료공간을 줄일 수 있었다. 변위의 위치 단위로 저장하지 않고 실제 사용될 변위 정보만 저장할 경우 해당 지점에서 저장할 변위의 개수 정보가 추가적으로 저장되어야하며 주소 맵은 여전히 텍셀 당 최소 16비트를 할당해야 하므로 압축률이 획기적으로 좋아지지 않는다. 본 논문에서 제안한 방법은 압축으로 인해 변위 맵의 위치 특성이 사라지므로 계층적이 아닌 선형적인 광선 탐색을 하였기 때문에 기존 방식[7]의 실행속도는 약 57fps인데 비하여 실행 속도가 약 27fps로 약 47% 저하되었다.

6. 결론 및 향후 과제

본 논문은 저장 공간의 낭비가 심한 다층 변위 맵을 비손실 방식으로 압축하고 GPU상에서 해제하여 사용하는 변위 매핑 방식을 제안하였다. 본 논문에서 제안하는 방법은 다층 변위 맵을 보다 범용적으로 사용할 수 있으며 지표각에서 오류가 발생하지 않는 올바른 탐색을 보장한다. 현재는 GPU의 특성상 많은 비용을 필요로 하는 텍스쳐 참조를 줄이기 위해서 최적화되지 않은 압축을 하였다. 하지만 다중 GPU를 사용한 탐색의 병렬 처리로 텍스쳐 참조 작업을 분담할 수 있다면 최적화된 압축 맵 정보를 사용하더라도 현재와 비슷한 수준의 실행 속도를 보장할 수 있을 것으로 예상된다. 이를 위해서는 다중 GPU에서 광선의 탐색을 나누어 수행하고 그 결과물을 조합하는 방법에 대한 추가적인 연구가 필요하다.

참고문헌

[1] Policarpo, F., Oliveira, M, M., and Comba, J., “Real-time relief mapping on arbitrary polygonal surfaces”, In Proceeding of ACM Symposium on Interactive 3D Graphics and Games 2005, pp.155-162, 2005.

[2] Brawley, Z., and Tatarchuk, N., “Parallax Occlusion Mapping: Self-Shadowing, Perspective-Correct Bump Mapping Using Reverse Height Map Tracing”, In ShaderX3: Advanced Rendering with DirectX and OpenGL, pp.135-154, 2004.

[3] 오경수, 기현우, “이미지 피라미드를 이용한 변위 맵의 실시간 렌더링”, 하계 한국게임학회 학술발표대회, pp. 209-215, 2006.

[4] Wang, L., Wang, X., Tong, X., Hu, S., Guo, B., and Shum, H.-Y., “View-dependent displacement mapping”, Proceeding of SIGGRAPH 2003, pp.334-339, 2003.

[5] Wang, X., Tong, X., Lin, S., Hu, S., Guo, B., and Shum, H.-Y., “Generalized displacement maps”, In Eurographics Symposium on Rendering 2004, EUROGRAPHICS, Keller and Jensen, Eds., EUROGRAPHICS, pp.227-233, 2004.

[6] Policarpo, F., and Oliveira, M, M., “Relief mapping of Non-height-field surface details”, I3D ′06: Proceedings of the 2006 symposium on Interactive 3D graphics and games, pp.55-62, 2006.

[7] 전영재, 오경수, “이미지 피라미드 기반 다층 높이 맵을 사용한 변위 매핑 기법”, 한국컴퓨터그래픽스학회논문지, 14권, 3호, pp.11-19, 2008.

[8] Porumbescu, S, D., Burge, B., Feng, L., Joy, K, I., “Shell maps”, In Proceeding of ACM SIGGRAPH 2005, pp.626-633, 2005.

[9] Chen, Y., Tong. X., Wang, J., Lin, S., Guo, B., and Shum, H. Y., “Shell texture functions”, ACM Transaction of Graphics - Proceedings of SIGGRAPH 2004, pp.343-352, 2004.

[10] Lefebvre, S. and Hoppe, H., “Perfect spatial hashing”. In ACM SIGGRAPH 2006. SIGGRAPH ′06. ACM, New York, pp.579-588, 2006.

전 영 재(Chun, Young Jae)

2007년～현재 숭실대학교 미디어학부 석박사통합과정

관심분야 : 실시간 컴퓨터 그래픽스, 게임

김 해 동(Kim, Hae Dong)

1994년 전남대학교 전사학과 학사
1996년 전남대학교 이학석사(컴퓨터그래픽스)
1996년 시스템공학연구소 입소
～현재 한국전자통신연구원 선임연구원

관심분야 : 실시간 렌더링, 전역조명, 컴퓨터비전

조 성 현(Cho, Sung Hyun)

1978년 서울대학교 계산통계학과 이학사
1980년 서울대학교 계산통계학과 이학석사
1995년 UCLA 컴퓨터과학과 이학박사
1996년～현재 홍익대학교 게임학부 교수

관심분야 : 게임 프로그래밍, 게임 AI, 게임 물리