

왕(Wang) 타일 지형 합성과 시차 맵핑을 이용한 실시간 렌더링

정재원^o 최민규

광운대학교

jwjeong@cs.kw.ac.kr^o, mgchoi@kw.ac.kr

Wang Tile Terrain Synthesis and Real-Time Rendering using Parallax Mapping

Jae Won Jeong^o Min Gyu Choi

Kwangwoon University

요 약

높이 맵으로부터 지형의 기하구조를 형성하고 그 세부 묘사를 위하여 왕 타일을 이용하여 반복적이지 않는 타일링을 할 수 있다. 타일 내에 색상 정보와 더불어 높이 정보를 추가하여 세부적인 요철의 정보를 담아 시차 맵핑을 통하여 효과적으로 표현 할 수 있다. 본 논문에서는 이를 실시간 렌더링 할 때 문제되는 부분을 해결하였으며, 지형을 위한 시차 맵핑의 보안을 제안한다.

ABSTRACT

The geometric structure constructs terrain from height map and non-periodically tiling with Wang tile for the detail description. It will be able to express detail bump using parallax mapping to add decal data together with height data in tile. We solve the problem when the real-time rendering and propose improving of parallax mapping for the terrain.

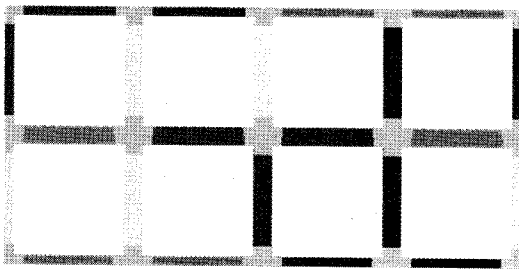
Keyword : Wang Tile, Parallax Mapping, Terrain Rendering , Real-time Rendering

※ (본 연구는 서울시 산학연 협력사업, 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업(IITA-2008-0801-0018)의 연구결과로 수행되었음.)

1. 서론

지형(terrain)은 게임이나 영화, 애니메이션을 포함하여 컴퓨터 그래픽스를 이용하여 생성한 거의 모든 장면(scene)에서 필요하다고 할 수 있으며, 사실적인 지형 생성은 결과 영상의 사실성을 높이는데 있어서 매우 중요하다. 또한, 게임 등의 실시간 대화형 응용 프로그램에서는 지형을 빠르게 렌더링할 수 있어야 한다. 본 논문은 광활한 지형의 세부적인 묘사를 효과적으로 기술하고, 이를 프로그래밍 가능한 하드웨어(programmable graphics hardware)에서 실시간에 사실적으로 렌더링 하는 방법을 다룬다.

높이 맵(height map) 등에 기반하여 생성된 지형의 기하 구조(geometry)를 세부적으로 묘사하기 위해서는 타일(tile) 형태의 컬러 텍스처(texture)를 반복적으로 맵핑(mapping)하는 방식이 널리 사용되고 있다. 하지만, 이러한 방식을 통해 생성된 지형은 원거리에서 바라볼 때 텍스처의 주기적인 반복이 두드러지게 보이는 문제가 있다. Cohen등[4]은 타일 기반의 텍스처 생성에 있어서 주기적인 반복을 해결하기 위해서 컬러 코드(color-code)를 가진 왕 타일들(Wang Tiles)을 사용하여 넓고 복잡한 형태의 색상 텍스처나 패턴, 기하 구조 등을 효과적으로 생성한 바 있다.



[그림 1] 컬러 코드를 가진 여덟 장의 왕 타일

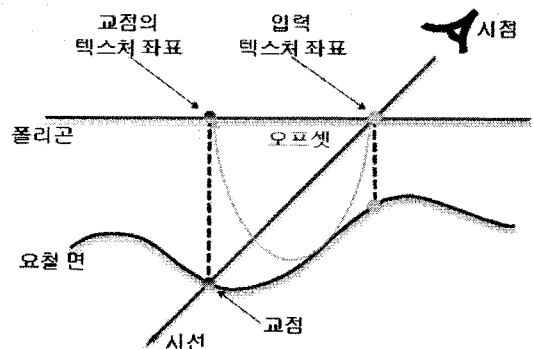
본 논문에서는 반복이 없는 광활한 지형을 세부적으로 묘사하기 위하여 왕 타일에 기반한 지형 생성 방법을 도입한다. 타일 기반의 지형을 보다 사실적으로 또한 효과적으로 생성하기 위해서 타일 내에서의 지형의 색뿐만 아니라 요철 또한 세부적으로 기술할 필요가 있다. 타일 내에서의 지형의

색을 컬러 이미지로 기술할 수 있는 것처럼 요철

또한 높이 맵에 해당하는 그레이 스케일(gray scale) 이미지로 기술할 수 있다. 따라서, 색상 이미지 뿐만 아니라 요철 이미지에 왕 타일을 적용함으로써 사실적이며 광활한 지형을 비교적 간단하게 생성할 수 있다. 하지만, 색상 및 요철 타일로 이루어진 광활한 지형을 실시간에 사실적으로 렌더링하는 것은 보다 복잡한 작업을 요구한다.

높이 맵으로 표현된 요철을 렌더링하기 위해서는 Blinn[1]이 제안한 범프 맵핑(bump mapping)이나 Cook[3]이 제안한 변위 맵핑(displacement mapping) 등이 널리 사용되고 있다. 범프 맵핑은 픽셀의 실제 법선 벡터대신 높이 맵에서 구한 법선 벡터를 사용하여 평면에 요철이 있는 것처럼 셰이딩하는 방법으로써 GPU에서 쉽고 빠르게 구현 가능하다. 반면, GPU 기반 변위 맵핑[5][10]은 광선 추적을 통해 요철 면과의 교점을 높이 맵에서 정확히 계산하고 이를 적용시켜 실제 요철을 표현한다. 따라서 실루엣(silhouette)을 정확히 표현할 수 있지만, 범프 맵핑 보다 구현이 복잡하며 수행 성능이 떨어진다. 본 논문에서는

범프 맵핑이나 변위 맵핑대신 범프 맵핑에서 모션 시차(motion parallax)로 인해 발생하는 오차를 근사적으로 보정하는 급경사 시차 맵핑(step parallax mapping)[8]을 왕 타일로 구성된 높이 맵에 맞도록 변형하여 사용한다. 변형된 급경사 시차 맵핑은 변위를 근사적으로 계산하기 때문에 GPU에서 변위 맵핑보다 빠르게 수행 가능하며 범프 맵핑보다 사실적인 렌더링이 가능하다.

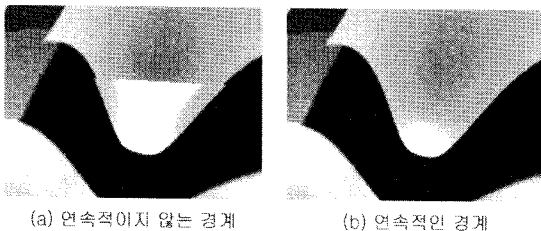


[그림 2] 시차 맵핑

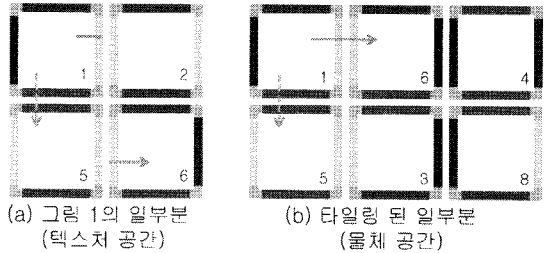
2. 배경 지식

왕 타일. 왕 타일은 모서리에 컬러 코드가 있는 사각형의 타일들이다. 모서리의 컬러 코드가 동일하도록 배치하여 타일 간의 연속성을 보장할 수 있고, 타일들을 확률적으로 배치하여 반복이 없는 무한한 크기의 이미지를 생성할 수 있다. 이때 확률적으로 반복이 없는 선택을 가능하게 하기 위해서는 두 모서리의 컬러 코드가 동일한 타일이 두 장 이상 존재해야 한다. 따라서, 왕 타일은 그림 1과 같이 최소 여덟 장의 타일들로 구성된다. 그림 8(b)는 그림 1의 왕 타일을 사용하여 반복이 없도록 생성한 큰 텍스처를 보여준다.

시차 맵핑. 시차 맵핑에서의 높이 맵은 그림 2에서와 같이 폴리곤의 안쪽 방향으로의 깊이를 나타내므로 깊이를 맵이라 하는 것이 더 정확하지만 역사적인 이유로 높이를 맵이라 불린다. 하드웨어 렌더링을 통해 이미지 평면에 투영된 폴리곤 상의 한 점과 시점(view point)으로부터 그 점을 지나가는 시선(viewing ray)이 실제 요철 면과 교차한 점은 시점이 이동함에 따라 시차가 발생한다. 이미지 평면에 투영된 점을 올바르게 웨이딩하기 위해서는 입력 텍스처 좌표를 시차에 따라 교점의 텍스처 좌표로 보정해야 한다. 이때 두 텍스처 좌표 간의 오차를 오프셋(offset)이라 한다. 시차 맵핑은 정확한 오프셋을 구하는 대신 빠른 시간 내에 근사적으로 오프셋을 구하여 입력 텍스처 좌표를 보정하고, 이를 이용하여 픽셀을 웨이딩함으로써 요철을 표현한다[7][8].



[그림 3] 시차 맵핑에 의한 타일 경계에서의 비연속성



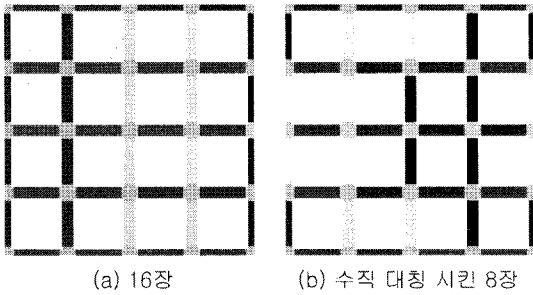
[그림 4] 오프셋에 의한 텍스처 좌표 보정: 텍스처 공간에서 왕 타일의 인접성을 고려하여 텍스처 좌표의 범위를 건너뛴.

3. 방법

3.1 왕 타일에서의 시차 맵핑

타일을 렌더링하기 위하여 단순히 텍스처 맵핑을 하거나 또는 범프 맵핑만을 한다면 참조해야 할 텍스처 좌표 값은 항상 그 타일의 텍스처 좌표 범위 안에 존재 한다. 하지만, 지형을 보다 사실적으로 렌더링하기 위하여 타일 내에서의 기하학적인 세부 묘사를 기술하는 높이 맵을 사용하여 시차 맵핑을 할 경우에는 시차에 따라 발생하는 오프셋만큼 텍스처 좌표를 보정해야 한다. 따라서, 참조해야 할 텍스처 좌표가 해당 타일의 텍스처 좌표 범위 밖으로 나가는 경우가 발생한다. 이로 인하여 왕 타일을 사용하여 연속성을 유지하며 생성한 타일들이라도 시차 맵핑에 의하여 비연속적으로 렌더링되는 현상이 발생한다(그림 3 참조).

그림 1에서와 같은 왕 타일을 사용하여 생성한 반복이 없는 타일들 중에서 그 일부분이 그림 4(b)에서와 같다고 하자. 또한 편의상 그림 4(a)에서와 같이 왕 타일의 일부분만으로 이루어진 텍스처 공간을 고려하자. 물체 공간(object space)상에서 1번 타일에 있는 한 점을 시차 맵핑으로 렌더링할 때 시차에 의한 오프셋으로 인하여 참조할 텍스처 좌표가 1번 타일의 아래쪽으로 넘어가면 텍스처 공간이나 물체 공간 모두에서 5번 타일이라 별다른 문제가 발생하지 않는다. 하지만 참조할 텍스처 좌표가 1번 타일의 오른쪽으로 넘어가면 물체 공간상에서는 6번 타일을 만나는데 반하여 텍스처



[그림 5] 한 장의 텍스처로 패킹된 왕 타일

공간상에서는 2번 타일을 만난다. 따라서, 단순히 텍스처 좌표를 시차에 의한 오프셋으로 보정한다면 텍스처 공간상에서 6번 타일을 참조해야 함에도 불구하고 2번 타일을 참조하게 된다. 이를 해결하기 위해서는 오프셋에 의해 텍스처 좌표를 이동시킬 때 해당 타일의 범위를 벗어나는지를 확인하고, 범위를 벗어나는 경우에는 텍스처 공간상에서 왕 타일의 배치를 참고하여 해당 타일의 텍스처 좌표 범위로 건너뛰어야 한다. 동일한 왕 타일 텍스처를 사용하는 모든 폴리곤들에서는 타일의 인접 관계가 동일하므로 픽셀 셰이더(pixel shader)에 왕 타일 텍스처의 인접 관계 및 텍스처 좌표의 범위를 유니폼(uniform) 파라미터로 넘길 수 있다.

3.2 GPU를 위한 왕 타일 패킹

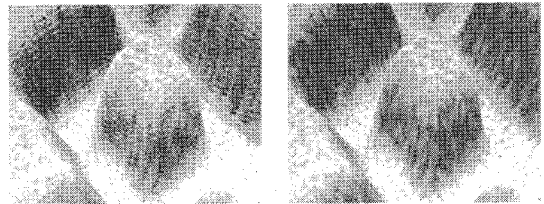
연속적이며 반복이 없는 타일 배치를 위해서는 최소 8장의 타일을 사용하여 왕 타일을 구성해야 한다. 일반적으로, GPU에서 한꺼번에 처리 할 수 있는 텍스처의 개수에는 한계가 있기 때문에 게임 등에서는 텍스처를 패킹(packing)하여 사용하고 있다. 본 논문에서도 왕 타일들을 하나의 텍스처로 패킹하여 사용하고자 한다. 하지만, 그래픽스 하드웨어는 필터링을 통한 확대나 축소 등으로 적재 (loading)된 텍스처의 밍맵(mipmap)을 자동으로 생성한다. 따라서, 이 과정에서 타일 간의 경계에서 원하지 않는 부분과 혼합이 이루어질 수 있다.

그림 1과 같이 8장의 왕 타일을 4장씩 두 줄로 배치할 경우 내부 경계는 같은 컬러 코드로 배치되어 밍맵을 생성할 때 별다른 문제가 없지만, 테두리에서는 두루마리 현상(warp around)으로 인하여

서로 다른 컬러 코드를 가진 타일들이 혼합되는 문제가 발생한다. 또한 직사각형의 텍스처이기 때문에 GPU 내에서 텍스처 좌표를 스케일링해야 하는 불편함이 있다. Wei는 이러한문제를 해결하기 위하여 그림 5(a)에서와 같이 16장의 왕 타일을 한 장의 텍스처에 배치하였다[11].

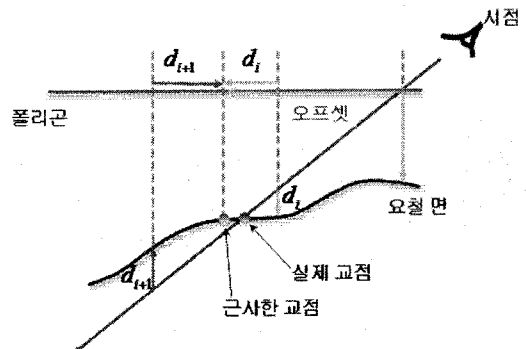
본 논문에서는 16장의 왕 타일을 생성해야 하는 부담을 줄이기 위하여 그림 5(b)에서와 같이 먼저 8장의 왕 타일을 배치하고 이를 수직으로 대칭 시킨 타일들을 포함하여 총 16장의 타일들을 한 장의 텍스처로 패킹한다. 지형을 이루는 폴리곤들의 정점에 텍스처 좌표를 설정할 때에는 수직으로 대칭 시킨 타일들은 사용하지 않고 원래 생성한 8장의 타일만을 이용한다. 수직 대칭 시킨 타일들은 테두리에서 두루마리 현상에 의한 잘못된 혼합을 방지하고 패킹된 타일들의 텍스처가 정사각형이 되게 하는 용도일 뿐이다. 이와 같은 패킹은 미리 준비해야 할 왕 타일의 개수를 증가시키지 않으면서 하드웨어 밍맵핑에 의한 부작용을 방지한다.

3.3 지형을 위한 시차 맵핑의 보안

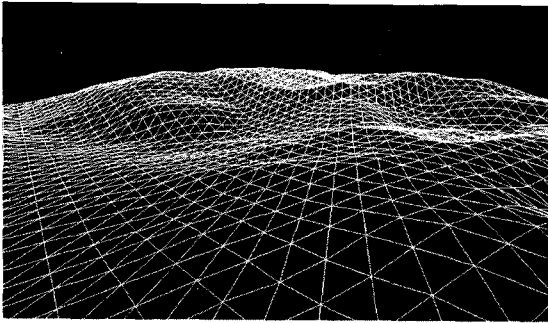


(a) 선형 검색 (b) 선형 검색 후, 선형 보간

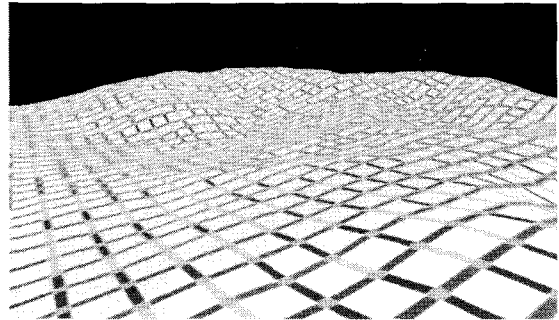
[그림 6] 교점 검색 방법에 따른 차이



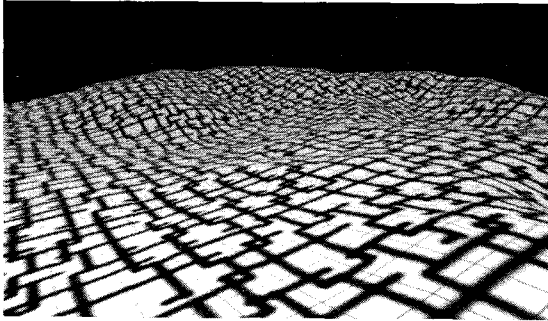
[그림 7] 선형 보간을 이용한 광선 추적



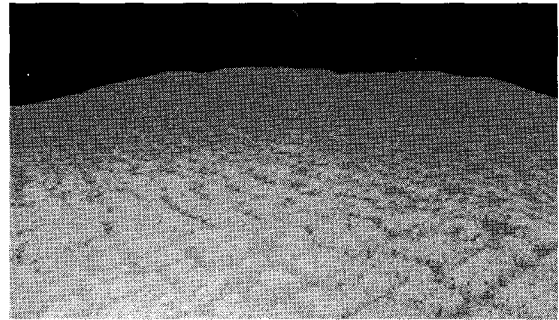
(a) 지형의 기하 구조



(b) 컬러 코드를 가진 왕 타일을 맵핑한 지형



(c) 높이 맵을 담은 타일을 맵핑한 지형



(d) 색상과 시차 맵핑으로 높이 맵을 표현한 지형

[그림 8] 왕 타일을 이용한 타일 기반 지형 생성과 렌더링

시차 맵핑은 그림 6(a)와 같이 경사가 심한 부분에서 꼭면으로 표현되어야 할 부분이 계단처럼 보이는 현상이 있다. 계단 현상이 발생하는 이유는 광선 추적 시에 요철 면과 시점 벡터가 만나는 정확한 교점을 찾지 못하기 때문이다. 이는 교점을 해석적으로 계산하지 않고 포함되어 있는 범위를 찾기 때문이다.

본 논문에서는 시차 차폐 맵핑[6]에서와 같이 교점을 찾기 위하여 선형 검색으로 근사한 후, 선형 보간을 한다. 순차적인 선형 검색을 통하여 교점이 존재하는 작은 범위를 찾을 수 있다[8]. 충분히 작은 범위를 찾았다면 이 범위에서 요철 면은 직선에 가까울 것이다. 그림 6과 같이 범위 시작과 범위 끝에서 광선의 높이와 높이 맵에서 가져온 높이의 차를 구할 수 있다. 이 차이가 적은 쪽으로 교점이 가까이에 있을 것이다. 범위 시작의 차(d_i)와 범위 끝의 차 (d_{i+1})의 비율을 폴리곤 좌표상의 범위

에서 선형 보간하면 교점을 구할 수 있다. 이로 인하여 한번의 연산으로 실제 교점에 매우 근접한 교점을 근사 할 수 있으며 그림 6(b)와 같이 자연스러운 표면을 볼 수 있다.

4. 구현 및 결과

제안된 방법은 마이크로소프트 윈도우 XP 환경에서 DirectX SDK(Feb 2007) 9.0c를 이용하여 픽셀셰이더 3.0으로 작성되었다. 본 논문에서 구현된 모든 결과는 인텔 펜티엄 D 3.2GHz 프로세서, 2GB 메모리와 NVIDIA GeForce 7900 GT 그래픽카드를 장착한 PC에서 실행하였다.

본 논문에서는 그림 8(a)에 보이는 8192장의 삼각형으로 이루어진 지형에 한 장의 텍스처로 패킹한 8장의 왕 타일을 사용하여 실험하였다. 그림 8(b)와 (c)는 각 컬러 코드를 가진 왕 타일과 높이

맵 왕 타일을 색상 텍스처로 맵핑한 지형을 보여 준다. 그림 8(d)는 8개의 높이 맵 왕 타일과 색상 왕 타일을 이용하여 시차 맵핑을 한 결과를 보여 준다. 최소한의 타일을 이용하여 지형의 기본이 되는 바닥 면의 잔디와 땅이 조합된 타일을 합성하고 반복이 없는 지형을 생성하였다. 광선 추적에 의한 오프셋 근사의 선형 검색 횟수를 최대 12회 사용하였을 때, 800 × 480의 해상도에서 50fps 이상을 얻을 수 있었다.

5. 결론

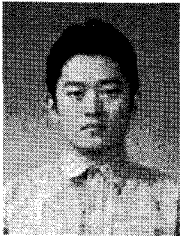
본 논문에서는 왕 타일로 구성된 높이 맵을 이용하여 광활한 지형을 생성하고 실시간에 렌더링하는 기법을 제안하였다. 왕 타일로 구성된 높이 맵을 효과적으로 렌더링하기 위하여 시차 맵핑을 채택하였고, 왕 타일의 인접 관계 및 텍스처 좌표의 범위를 고려하여 시차에 따른 오프셋을 보정하였다.

제안된 기법의 주요 연산은 픽셀 웨이더에서의 광선 추적이므로 해상도가 높은 결과를 원할수록 그에 따른 픽셀의 증가로 수행 성능이 저하된다. 하지만 지형의 크기에 상관없이 화면에 보이는 부분의 최대 픽셀수는 동일하기 때문에, 적절한 수행 성능이 보장되는 해상도를 사용한다면 지형의 크기나 복잡도에 상관없이 안정적인 렌더링이 가능하다.

참고문헌

- [1] Blinn, J. F. Simulation of Wrinkled Surfaces, In Proc. ACM SIGGRAPH 78, pp. 286-292, 1978.
- [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, Engel, W., Ed., Charles River Media, pp. 135-154, 2004.
- [3] Cook, R. L. Shade Trees, In Proc. ACM SIGGRAPH 84, pp. 223-231. 1984.
- [4] Cohen, M. F., Shade, J., Hiller, S., and Deussen, O. Wang Tiles for Image and Texture Generation,

- ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2003), pp. 287-294, 2003.
- [5] Donnelly, W. Per-Pixel Displacement Mapping with Distance Functions. In GPU Gems 2, M. Pharr, Ed., Addison-Wesley, pp. 123-136. 2005.
- [6] Tatarchuk, N. Dynamic Parallax Occlusion Mapping with Approximate Soft Shadows, In Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, 2006
- [7] Kaneko, T., Takahei, T., Inami, M., Kawakami, N., Yanagida, Y., Maeda, T., and Tachi, S. Detailed Shape Representation with Parallax Mapping, In Proc. ICAT 2001, pp. 205-208, 2001.
- [8] McGuire, M. and McGuire, M. Steep Parallax Mapping, I3D 2005 Poster, 2005.
- [9] WANG, L., WANG, X., TONG, X., LIN, S., HU, S., GUO, B., AND SHUM, H.-Y. View-dependent displacement mapping. ACM Trans. Graph. 22, 3, pp. 334-339, 2003.
- [10] Oliveira, M. M., Bishop, G., and Mcallister, D. Relief Texture Mapping. In Proc. ACM SIGGRAPH 2000, pp. 359-368, 2000.
- [11] Wei, L.-Y., Tile-Based Texture Mapping on Graphics Hardware, In Proc. ACM SIGGRAPH/Eurographics Conf. Graphics Hardware, 2004.



정재원 (Jae Won Jeong)

약력 : 2006년 8월 광운대학교 컴퓨터 소프트웨어학과
공학사. 2006년 8월-현재 광운대학교 컴퓨터과학과



최민규 (Min Gyu Choi)

약력 : 1996년 2월 한국과학기술원 전산학과 공학사.
1998년 2월 한국과학기술원 전산학과 공학석사.
2003년 2월 한국과학기술원 전산학과 공학박사.
2003년 3월 -2005년 2월 서울대학교 컴퓨터과
학과 박사 후 연구원.
2005년 3월-현재 광운대학교 컴퓨터과학과 조
교수.

관심분야 : 변형체 애니메이션, 동작 계획, 동작 편집,
영상처리.
