

정점 응집맵을 이용한 지형 렌더링*

조인우, 이은석, 신병석

인하대학교, 컴퓨터·정보 공학과

joins@inha.edu, elflee77@inha.edu, bsshin@inha.ac.kr

Terrain Rendering Using Vertex Cohesion Map

Inwoo Jo, Eun-Seok Lee, Byeong-Seok Shin

Dept. of Computer Science and Information Engineering, Inha University

요 약

최근 지형렌더링에서 사용되는 DEM(digital elevation map) 데이터들은 일반 컴퓨터에서 처리 가능한 메모리 용량을 초과하기 때문에 밍맵(mipmap)을 이용한 상세단계(LOD : level-of-detail) 기법들을 사용하는 외부 메모리 처리(out-of-core) 기법들이 많이 연구되고 있다. 하지만 밍맵을 이용한 상세단계 기법들은 높은 레벨의 상세단계에서 데이터의 간략화에 따른 기하오차가 발생한다. 이러한 기하오차는 시점이 이동할 때 상세단계가 변화하는 부분에서 기하과핑(geometry popping) 현상을 유발한다. 본 논문에선 기하오차를 줄이기 위해 정점 응집맵을 제안한다. 전처리 단계에서 생성되는 정점 응집맵은 벡터를 저장한 텍스처이다. 이 벡터들은 상대적으로 기울기 변화량이 큰 위치로 주변의 정점들을 응집시켜 지형의 기하오차를 줄이기 때문에 단순히 밍맵을 이용하여 지형을 렌더링 했을 때 나타나는 기하과핑 현상을 효과적으로 줄일 수 있다.

ABSTRACT

Recently in terrain rendering, most researches introduce mipmap-based out-of-core methods for handling large sized DEM data which does not fit in main memory of general computer. However, mipmap-based LOD(level-of-detail) methods occur geometric errors which appear in data simplifying the higher LOD level. These geometric errors cause geometric popping effects where LOD level changes when viewpoint moves. In this paper, we propose vertex cohesion map for reducing geometric error. In preprocessing step, we generate vertex cohesion map, which is a texture that stores the vectors. By these vectors, each vertex will be cohered into the position in which the difference of gradient value is bigger than others. Therefore in terrain rendering, using vertex cohesion map can dramatically reduce the geometry popping effects rather than using mipmap.

Keywords : Terrain(지형), Real-time rendering(실시간 렌더링), Level Of Detail(상세단계)

접수일자 : 2010년 11월 09일 심사완료 : 2010년 12월 08일

교신저자(Corresponding Author) : 신병석

※ 본 연구는 지식경제부 및 한국산업기술평가위원회의 산업원천기술개발사업(정보통신)의 일환으로 수행하였음. [10035243, CPS를 위한 콤포넌트 기반 설계이론 및 제어커널 개발]

1. 서론

컴퓨터 게임, 비행 시뮬레이션, 지리정보 시스템(geographic information system) 등에서 실제 환경 및 가상 환경의 기반이 되는 지형 표현은 중요한 요소이다. 사실감을 더욱 극대화하기 위해서는 빠르고 현실감 있는 지형 렌더링이 필수적이다.

최근의 지형 렌더링 연구 분야에서 가장 큰 이슈는 대용량 지형 데이터를 실시간 처리 하는 것이다. 하지만 일반 컴퓨터에서 제공하는 그래픽 메모리는 한정되어 있기 때문에 대용량의 지형데이터의 실시간 가시화는 쉽지 않다. 이를 해결하기 위한 방법으로 최근에 외부 메모리 처리 방법[1,2,3]들이 많이 연구되고 있다.

외부 메모리 처리 기법은 한정된 그래픽 메모리에 더욱 큰 지형을 렌더링 하기 위해 지형 데이터를 용량이 큰 디스크에서 관리하고, 렌더링시에 갱신되는 부분만 메모리로 업로드하여 사용하는 기법이다. 외부 메모리 처리 기법은 데이터 갱신시에 발생하는 GPU-CPU 메모리간에 데이터 전송량을 줄이기 위해, 지형 데이터에 상세단계 기법을 적용한다. 지형 데이터에 적용하는 상세단계 기법 중 하나인 땀맵은 상세단계가 높아질수록 해상도를 감소시켜 용량을 줄이는 방법이다[2]. 하지만 이 방법은 높은 단계에서 샘플점 개수의 부족으로 기하오차를 발생시킨다.

본 논문에서는 지형 데이터의 해상도가 낮아지면서 발생하는 기하오차를 줄이기 위해 지형의 기울기 변화량이 큰 위치로 주변의 정점들을 응집시켜 샘플점의 간격을 좁게 하는 방법을 제안한다. 정점을 이동시키기 위해 전처리 과정에서 정점 이동벡터와 이동될 지점의 높이값을 텍스처로 저장하는데 이것을 정점 응집맵이라고 한다. 렌더링과정에서는 DEM데이터 대신 정점 응집맵을 정점 셰이더로 업로드한 후 저장된 벡터를 이용하여 정점을 이동 시킨 후 지형을 가시화 한다. 이렇게 하면 지형의 기울기 변화량이 큰 지점으로 주변에 정점들을 응집시켜 기하 오차를 줄일 수 있다.

2절에서는 지형 렌더링에 사용되는 상세단계 표현 기법들과 외부 메모리 처리 기법들에 대해 기술하고, 3절에서는 본 논문에서 제안하는 정점 응집맵에 대해서 자세히 설명한다. 4절에서는 제안된 방법을 사용하여 실험한 결과를 보이고, 마지막 절에서 결론을 맺는다.

2. 관련연구

과거에는 대용량 지형데이터를 빠르게 렌더링 하기 위해 계층구조를 이용하여 상세단계로 표현하는 렌더링 기법들이 연구되었다. 대표적인 방법으로는 Lindstrom이 제안한 사진트리기법[4]과 Duchaineau가 제안한 이진트리구조를 사용하는 ROAM (real-time optimally adaptive meshes) 기법[5]이 있다. Lindstrom은 사진트리의 분할과 병합을 결정하는데 상향식(bottom-up) 접근방법을 이용하였다. ROAM은 시점에 따라 지형을 동적으로 생성하는 이진 트리(binary tree) 구조를 이용한다. 각 패치는 단순한 이등변 삼각형이며, 각 삼각형을 재귀적으로 두 개의 자식노드로 분할하는 구조로 되어 있다. 이 방법은 화면 오차(screen error)를 기반으로 하여 재귀적으로 사용자가 설정한 오차 보다 높은 지형은 분할을 하고, 사용자가 설정한 오차 보다 낮은 지형은 분할을 하지 않는 방법으로 상세단계를 조절한다.

이와 같이 계층구조를 사용하는 지형렌더링 방법들은 화면에 보이는 실제 지형의 오차값을 기반으로 지형을 렌더링 할 수 있다. 하지만 이웃 노드간에 상세단계 차이가 생기면 크랙(crack)이 발생할 수 있다. 이를 해결하기 위해서는 이웃노드에 접근해서 상세단계 차이를 계산해야 하므로 트리탐색이 필요하다. 최근 연구에서는 GPU(graphic process unit)를 이용한 지형 렌더링 방법들이 연구되고 있는데 트리탐색은 일괄적으로 병렬처리를 해야하는 GPU에서는 처리하기 어렵다.

Ulrich는 묶음상세단계(chunked LOD) 방법[6]

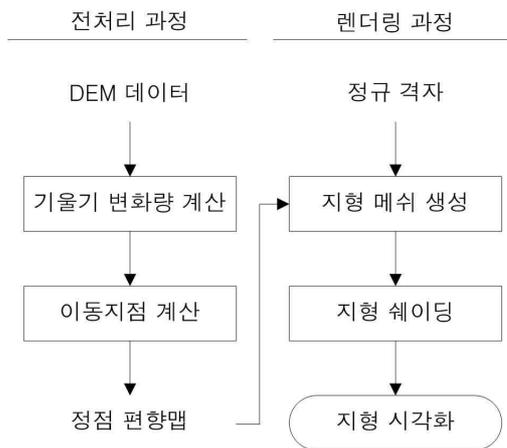
을 제안하였다. 이것은 전처리 단계에서 미리 상세 단계별로 지형 메쉬를 생성하고 CPU상에서 캐쉬로 관리한다. 캐쉬 형태로 관리되는 여러 해상도의 지형 데이터들은 렌더링 단계에서 필요에 의해 사진트리 형태의 묶음으로 GPU메모리로 업로드된다. Dachsbacher는 기하 이미지 와핑(geometry image warping)[7]기법을 제안 하였다. 이 방법은 시점까지 거리와 시야 절두체 선별여부, 표면방향, 차폐 조건 등을 고려하여 중요성 맵(importance map)[8]을 생성하고, GPU에서는 중요성 맵을 참조하여 지형의 중요부분에 기하 정보 주파수를 늘려서 렌더링한다. 최근 지형데이터의 용량이 급격히 커지면서 한정된 크기를 가지는 GPU 메모리에서는 실시간으로 지형을 렌더링 하기 어려워졌다. 그래서 최근에는 대용량의 지형을 가시화하기 외부 메모리 처리 기법에 대한 연구가 활발히 이루어지고 있다.

대표적인 외부 메모리 처리 기법에는 Losasso가 제안한 기하 클립맵(geometry clipmap)[1]이 있다. 기하 클립맵은 정규격자 형태의 정점을 가지는 클립을 생성하고, 시점까지 거리에 따라 클립을 확대하여 피라미드 구조로 렌더링한다. 이때 DEM 데이터는 클립맵[2]에서 제안한 뿔뿔 피라미드 구조를 이용하여 처리하였다. Linvy가 제안한 PGM(persistent grid mapping)기법[3]은 뷰포트 상에서 정규격자의 정점을 생성하여 지형 평면의 투영 시키고, DEM 데이터에서 높이 값을 읽어 지형을 생성한다. PGM도 기하 클립맵과 같이 DEM 데이터의 상세단계를 위해 클립맵을 사용한다. 클립맵 기반의 지형 가시화 기법들은 뿔뿔의 자료구조를 사용하기 때문에 높은 상세단계에서 해상도가 낮아져 기하오차가 발생한다.

3. 정점 응집맵

제안하는 지형 렌더링 방법은 [그림 1]의 순서로 처리된다. 입력된 DEM 데이터를 필터링 하여 각

샘플점의 기울기 변화량을 계산한다. 계산된 기울기 변화량으로 각 정점들의 이동 벡터를 계산한다. 계산된 이동 벡터와 이동된 지점의 높이 값을 텍스처(정점 응집맵)에 저장한다. 정점 응집맵은 렌더링 단계에서 정점 셰이더로 적재 되고, 정규격자의 정점들을 이동시켜 지형을 생성한다. 이것은 일반적인 지형렌더링에서 사용하는 변위매핑방식에 벡터로 정점을 이동시키는 과정만 추가한 것으로 렌더링과정에서 추가적인 시간이 소요되지 않는다.



[그림 1] 정점 응집맵을 이용한 지형 렌더링 과정

3.1 기울기 변화량 계산

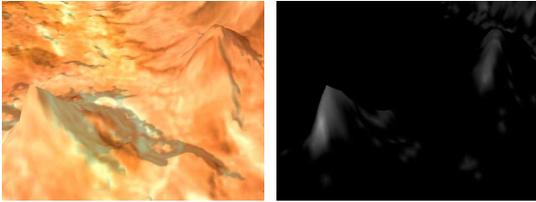
지형의 각 샘플점에서 기울기 변화량을 계산하기 위해 LoG(laplacian of gaussian)필터[9]를 적용한다. 일반적으로 LoG필터는 이미지 모롱화(blurring)를 통해 잡음을 제거하는 가우시안 커널과 주변 픽셀과의 차이를 이용해 윤곽선을 추출하는 라플라시안 커널로 구성된다. 여기서는 라플라시안 커널을 이용하여 각 정점별로 이웃 샘플점들 간의 기울기 변화량을 계산하고, 가우시안 커널을 통해 이웃 샘플점들 간의 기울기 변화량에 샘플점과의 거리에 따른 가중치를 준다. 샘플점 $p(x,y)$ 의 높이값을 $H(x,y)$ 라 할때, $p(x,y)$ 의 기울기 변화량 $D(x,y)$ 은 (식 1)과 같다.

$$D(x,y) = \sum_{h=-\frac{m}{2}}^{\frac{m}{2}} \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} e^{-\frac{r^2}{2\sigma^2}} (H(x,y) - H(x+h,y+k))$$

$$(r = \sqrt{h^2 + k^2})$$

(식 1)

r 은 샘플점이 위치한 커널중앙에서 커널의 다른 지점까지의 거리값으로 가우시안 함수의 인자이다. 계산된 기울기 변화량은 정점 응집맵 생성과정에서 사용하기 위해 정규화 한다. [그림 2]는 정규화한 기울기 변화량을 지형 색상으로 지형을 렌더링한 결과이다.



[그림 2] 실제 지형(왼쪽)의 기울기 변화량(오른쪽)

3.2 정점 응집맵 생성

정점을 기울기 변화량이 큰 위치로 응집시키기 위해, 지형의 각 샘플점에 주변 정점을 끌어 당기는 힘인 “인력” 속성을 부여한다. 인력 G 의 계산식은 (식 2)와 같다.

$$G = V \cdot e^{-\frac{x^2}{2(mD/2)^2}}, (V = p_d - p_s) \quad (식 2)$$

p_d 는 인력이 발생한 지점이고, p_s 는 p_d 의 인력의 영향을 받는 지점이다. 인력은 p_s 에서 p_d 까지의 벡터 V 에 가우시안 함수를 곱하여 계산된다. 가우시안 함수는 인력의 작용으로 p_s 들이 이동되었을 때 서로 겹치지 않게 하기 위한 거리 가중치를 계산하는데 사용된다. m 은 인력이 영향을 미치는 최대 범위이다. D 는 이전 단계에서 계산된 p_d 의 기울기

변화량이다. D 는 정규화된 값으로 m 을 곱하여 D 에 따라 인력의 적용범위가 조절되도록 한다. 가우시안 함수는 표준편차의 2배 이상에서는 0의 근접한 값이 나온다. 표준편차의 2배 값과 m 을 같게 하기 위해 표준편차 값을 인력 범위의 1/2로 한다.

정점 응집맵에서 각 샘플점의 이동벡터는 샘플점의 영향을 미치는 모든 인력의 합으로 계산한다. 인력의 합을 계산할 때 인력이 작용하지 않는 기울기 변화량이 0인 지점은 제외한다. 일반적인 힘의 합은 벡터합으로 계산한다. 하지만 벡터합은 [그림 3]의 (a)와 같은 정점 교차문제가 발생한다. 이것은 벡터합의 평균으로 해결 할 수 있다. 하지만 벡터합의 평균은 [그림 3]의 (b)와 같이 소수의 인력이 큰 지점들의 힘의 크기가 다수의 인력이 작은 지점에 의해 상쇄되어 작아진다. 이것은 샘플점이 인력이 큰 지점으로 적절히 이동하여 기하오차를 줄이는 것을 방해한다. 이것을 해결하기 위해 중요도에 따라 가중치를 곱하여 평균을 내는 가중평균으로 인력의 합을 계산한다. 인력의 합에서는 각 인력의 크기가 중요도 이므로 인력의 크기를 가중치로 하여 가중평균을 낸다. G 를 샘플점의 영향을 미치는 인력, $S = \sum |G_i|$ 이라고 할 때, 본문에서 샘플점의 이동벡터 G 를 계산하는데 사용한 가중평균 공식은 (식 3)과 같다.

$$G = \frac{1}{S} \sum_{i=0}^N |G_i| G \quad (식 3)$$

가중평균은 G 의 크기에 비례한 가중치를 사용하기 때문에 다수의 인력이 작은 지점들로 인해 소수의 인력이 큰 지점들의 인력의 크기가 상쇄되지 않는다. 그리고 가중평균도 벡터합의 평균처럼 벡터의 크기를 평균하기 때문에 정점 교차문제가 발생하지 않는다. [그림 3]의 (c)와 (d)는 문제점이 해결된 예를 보여준다.

계곡 주변의 정점들이 기울기 변화량이 큰 계곡에 응집되어 있는 것을 볼 수 있다. 밭맵은 고도값을 렌더링한 결과에서 계곡 부분의 기하오차가 발생하여 계곡 부분이 중간중간 끊기는 것을 볼 수 있다. 하지만 정점 응집맵은 정점을 기울기 변화량이 큰 곳으로 응집시키기 때문에 고도값을 렌더링한 결과에서 계곡 부분에서 발생한 기하오차가 줄어들어 밭맵에 비해 계곡이 끊기는 부분이 줄어든 것을 볼 수 있다.

[그림 5]는 원본과 깊이맵을 비교하여 나온 오차 수치를 텍스처로 저장한 오차영상을 비교한 것이다. 밭맵과 정점 응집맵 모두 해상도가 낮아지면서 점차 오차가 증가하고 있다. 밭맵은 해상도가 낮아지면서 계곡 부분의 오차가 급격히 증가하는 것을 볼 수 있다. 하지만 정점 응집맵은 전체적으로 오차가 밭맵에 비해 적고, 계곡부분의 오차가 주변의 오차와 큰 차이를 보이지 않는 것을 알 수 있다.

[표 1]은 두 맵의 오차합과 RMSE(root mean square error)를 비교한 표이다. 정점 응집맵의 오차합이 밭맵의 오차합보다 감소한 것을 볼 수 있으며, RMSE 또한 낮아진 것을 볼 수 있다.

[표 1] 밭맵과 정점 응집맵의 오차합과 RMSE 비교

해상도	밭맵		정점 응집맵		
	오차합	RMSE	오차합	RMSE	오차합 감소율
5132	481927	1.64	303309	1.36	0.37
2592	1145087	2.93	713859	2.14	0.38
1292	2301938	4.96	1542820	3.75	0.33
652	3984449	7.30	2967282	6.13	0.26

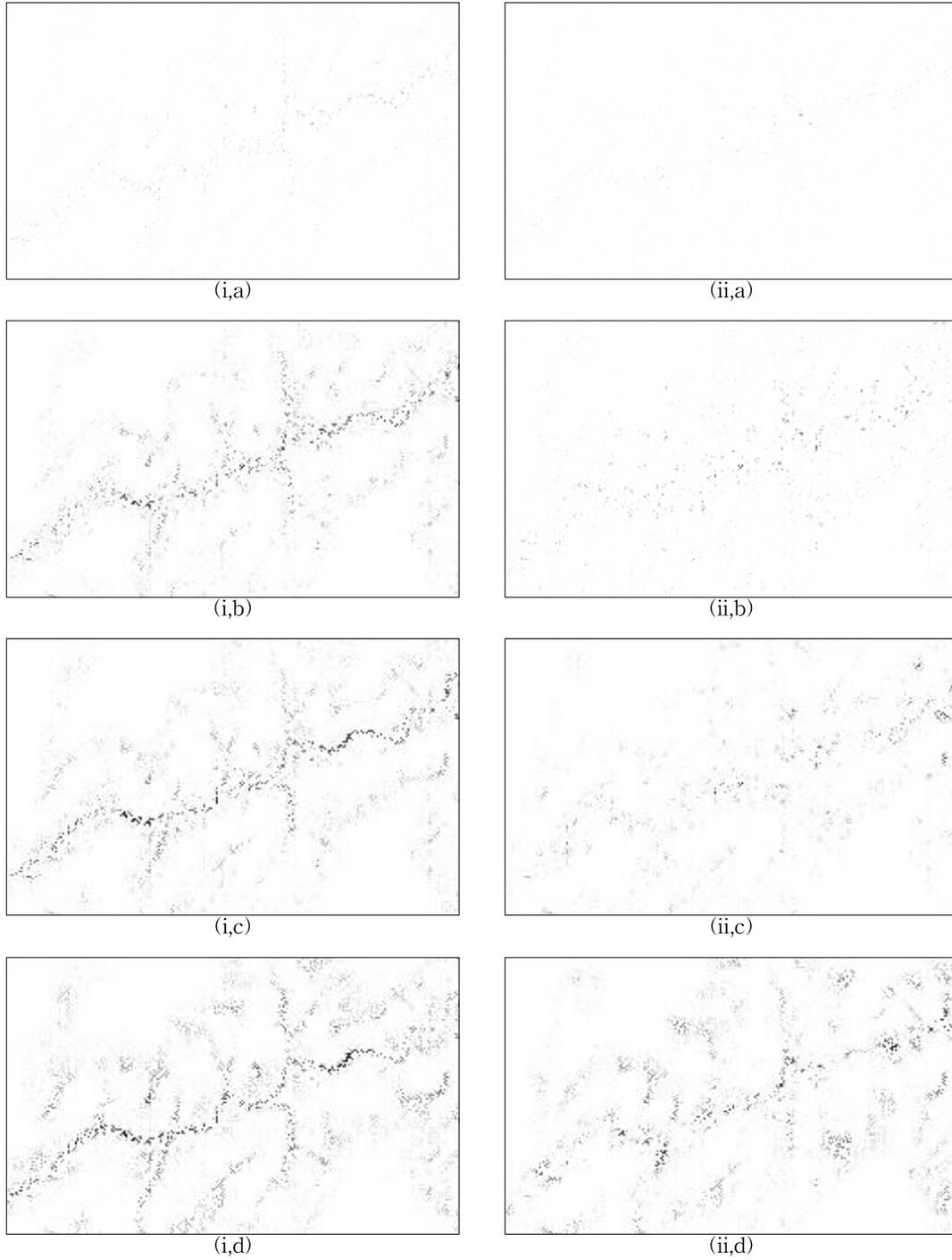
[표 2]는 두 맵의 렌더링 속도를 fps(frame per second) 단위로 비교한 표이다. 밭맵을 사용했을 때와 정점 응집맵을 사용했을 때의 렌더링 속도 차이가 거의 없는 것을 확인할 수 있다.

[표 2] 밭맵과 정점 응집맵의 렌더링 속도 비교

해상도	렌더링 속도 (fps)		성능비교 (B:A %)
	밭맵(A)	정점 응집맵(B)	
513 ²	260	256	98.46
259 ²	850	835	98.24
129 ²	2138	2113	98.83
65 ²	4350	4324	99.40

5. 결 론

본 논문에서는 정규격자를 사용하는 상세대단계방법에서 정점 해상도를 낮춤으로써 발생하는 기하오차를 줄이기 위해 정점 응집맵을 이용한 렌더링 방법을 제안하였다. 전처리 단계에서 생성되는 정점 응집맵은 GPU에서 처리하기 적합하게 텍스처로 저장이 할 수 있으며, DEM 데이터 대신 정점 셰이더에 업로드하여 추가적인 렌더링 시간의 증가 없이 낮아진 해상도로 발생한 기하오차를 효과적으로 줄일 수 있다. 하지만 [그림 5]에 해상도 65²로 렌더링한 정점 응집맵의 오차영상을 보면 근처의 정점들의 이동만으로는 한계가 있다. 앞으로 연구에서는 근처 정점들의 이동만으로 기하오차를 해결하지 못하는 경우에 대한 처리에 대한 연구를 진행할 것이다.



[그림 5] (i)맵맵, (ii)정점 편향맵의 오차맵 비교, (a)(b)(c)(d)는 해상도 5132, 2572, 1292, 652

참고문헌

- [1] F. Losasso, H. Hoppe, “Geometry Clipmaps: Terrain Rendering Using Nested Regular Grids”, Proceedings of SIGGRAPH 2004, Vol. 23, pp. 769-776, 2004
- [2] C. Tanner, C. Migdal, M. Jones, “The clipmap: A Virtual Mipmap.”, Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pp. 151-158, 1998
- [3] Y. Livny, N. Sokolovsky, T. Grinshpoun, J. El-Sana, “A GPU persistent grid mapping for terrain rendering”, The Visual Computer 24, pp. 139-153, 2008
- [4] P. Lindstrom, D. Koller, W. Ribarsky, L. Hodges, N. Faust, G. Turner, “Real-time, Continuous Level-of-Detail Rendering of Height Fields”, Proceedings of ACM SIGGRAPH 96, pp. 109-118, 1996.
- [5] M. Duchaineau, M. Wolinsky, D. Sigeti, M. Miller, C. Aldrich, M. Mineev-Weinstein, “ROAMing terrain: real-time optimally adapting meshes”, Proceedings of Visualization 97, pp. 81-88, 1997.
- [6] T. Ulrich, “Rendering massive terrains using chunked level of detail control”, SIGGRAPH 2002 Course Notes, 2002.
- [7] C. Dachsbacher, M. Stamminger, “Rendering Procedural Terrain by Geometry Image Warping”, Proceedings of Eurographics Symposium on Rendering, pp. 103-110, 2004
- [8] J. Peter-Pike, P. Sloan, D. M. Weinstein, J. D. Brederson, “Importance Driven Texture Coordinate Optimization”, Computer Graphics Forum, Vol. 17, No. 3, pp. 97-104, 1998
- [9] D. Marr, H. Hildreth, “Theory of edge detection”, Proceedings of the Royal Society of London, Vol. B207, pp. 187-217, 1980.



조 인 우 (Jo, In Woo)

2010년 2월 인하대학교 컴퓨터공학부(학사)
2010년-현재 인하대학교 컴퓨터공학부(석사)

관심분야 : 지형 렌더링, 상세단계선별



이 은 석 (Lee, Eun-Seok)

2008년 2월 인하대학교 컴퓨터공학부(학사)
2010년 8월 인하대학교 정보공학부(석사)

관심분야 : 지형 렌더링, 상세단계선별



신 병 석 (Shin, Byeong-Seok)

1990년 2월 서울대학교 컴퓨터공학과(학사)
1992년 2월 서울대학교 컴퓨터공학과(석사)
1997년 2월 서울대학교 컴퓨터공학과(박사)
2000년-현재 인하대학교 컴퓨터공학부 부교수

관심분야 : 볼륨그래픽스, 실시간렌더링
