

프랙탈과 셰이더 인스턴싱 기법을 이용한 자연스러운 잔디의 실시간 렌더링

김진모[†], 조형제^{**}

요 약

게임이나 실시간 시뮬레이션 시스템과 같은 분야에서 잔디는 광범위한 지표면을 구성하는 중요한 요소 중 하나이다. 실제 넓은 지형에 자라나는 많은 수의 잔디를 효율적으로 렌더링 하는 작업은 결코 쉬운 일이 아니다. 이러한 문제를 해결하려면 모델의 질과 계산 비용 절약의 두 가지 문제에 늘 직면하게 된다. 본 논문에서는 이러한 두 가지 문제를 해결하기 위한 방안으로 프랙탈 이론과 인스턴싱 기법을 도입하여 자연스러운 잔디의 효율적인 표현 방법을 제안한다. 기존의 잔디 표현의 경우 동일한 모델을 반복적으로 적용하였지만 본 연구에서는 프랙탈의 자기 유사성의 기본 이론을 활용하되 지형의 성질과 빛의 양을 인자로 도입하여 조건에 맞는 다양한 잔디의 자연스러운 표현 방법을 설계하고 또한 많은 수의 잔디 모델을 실시간으로 렌더링하기 위하여 GPU를 활용한 셰이더 인스턴싱 기법을 적용한다.

Real-time Rendering of Realistic Grasses Using Fractal and Shader-Instancing

Jin-Mo Kim[†], Hyung-Je Cho^{**}

ABSTRACT

The grass is one of important components that cover the wide surfaces in the application such as game or real time simulation. Actually, it not easy to render effectively numerous grasses that grow over the wide terrain. To solve the difficulty, we must find a solution to the two contradictions in terms : quality and calculation cost. As a solution to the above-mentioned task, in this paper, we propose an efficient method to represent the natural grasses by introducing fractal theory and instancing technique. Although the existing grass representation methods make use of a simple rule of applying a basic grass model repeatedly in rendering process, on the contrary we take advantage of the basic property of fractal's self-similarity and we devise a natural representation method suited to the given environment by introducing two important growth factors such as nature of terrain and quantity of light, and finally we apply a GPU-based shader instancing technique to rendering numerous grass models in real-time.

Key words: Grass Representation(잔디 표현), Real-time Rendering(실시간 렌더링), Fractal(프랙탈), GPU(그래픽 처리장치), Shader-Instancing(셰이더 인스턴싱)

※ 교신저자(Corresponding Author): 조형제, 주소: 서울특별시 중구 필동 3가 26번지(100-715), 전화: 02)2260-3344, E-mail: chojh@dgu.edu
접수일: 2009년 5월 26일, 수정일: 2009년 8월 6일

완료일: 2009년 9월 29일

[†] 준회원, 동국대학교 멀티미디어학과 박사과정
(E-mail: mythsjin@dongguk.edu)

^{**} 정회원, 동국대학교 멀티미디어학과 교수

1. 서론

잔디는 풀밭, 숲속, 운동장과 같은 일상생활에서 흔히 볼 수 있는 실외 지형을 자연스럽게 표현하는 중요한 요소 중 하나이다. 하지만 넓은 지형위에 각각 다른 형태를 지닌 잔디를 표현하는 일은 결코 쉬운 일이 아니다. 많은 수의 줄기들로 구성된 잔디 무리의 표현을 위해서는 많은 메모리 공간을 필요로 하기 때문이다.

현재까지 이런 복잡한 형태를 지닌 잔디를 자연스럽게 효율적으로 표현하기 위한 다양한 연구들이 이루어져 왔다. 기존의 연구들은 자연스러운 잔디의 표현과 속도 개선을 위한 효율적 처리 공정 사이의 문제를 해결하기 위한 여러 가지 방법을 제안하고 있다. 일반적으로 모델의 외형적 질을 향상시키는 데에 초점을 맞추게 되면 광범위한 잔디 표현에 따른 연산의 부하가 발생하고 계산 속도를 개선시키기 위한 방법들은 상대적으로 잔디의 질이 떨어지게 된다[1]. 이러한 문제를 효율적으로 처리하기 위하여 본 논문에서는 두 가지 알고리즘을 제안한다. 먼저 잔디 모델링에 있어 기존 연구들의 공통적인 특징은 기본 외형이 동일하다는 점이다. 일반적으로 동일한 형태를 가지는 잔디 줄기를 반복적으로 렌더링하여 지형위의 잔디 무리를 표현하는 방식이다. 이는 조명과 그림자를 통해 최종적인 잔디 색에는 변화를 주지만 기본적인 잔디의 형태는 변하지 않는 한계를 가지고 있다.

이러한 한계를 극복하기 위한 방법으로 본 연구에서 제안하는 첫 번째 알고리즘은 프랙탈의 기본 이론을 바탕으로 동일한 외형의 잔디 모델을 몇 가지 환경 요인에 따라 다양하게 변화시키는 방법이다. 프랙탈은 자기 유사성이라는 특징을 지니고 있어 유클리드 기하학보다 자연현상을 표현하는데 적합하다. 따라서 자기 유사성의 성질을 유지하면서 반복적으로 분할해나가는 프랙탈의 성질을 잔디의 성장에 적용하여 잔디를 더욱 자연스럽게 표현한다[2]. 여기에 기본적인 잔디의 성장 모델을 설정하고 잔디의 성장에 영향을 미치는 토양의 비옥도 및 빛의 양과 같은 요인들을 정의하여 주변 환경에 맞게 잔디를 실시간으로 변형한다. 이를 통하여 반복적인 잔디 표현의 한계에서 벗어나 넓은 지형에 다양한 종류의 잔디 생성을 가능하게 한다.

프랙탈 이론을 통한 잔디 성장 모델은 최종적으로 성장하는 잔디의 형태를 조건에 따라 다르게 표현하지만 잔디를 구성하는 정점과 인덱스의 수는 단계마다 동일하다. 따라서 메모리 구조를 효율적으로 설계한다면 많은 수의 잔디를 좀 더 효율적으로 표현할 수 있을 것이다. 이러한 원리를 바탕으로 제안하는 두 번째 알고리즘이 GPU를 활용한 셰이더 인스턴싱 기법이다. 인스턴싱 기법은 동일한 객체를 반복적으로 렌더링 할 경우 드로잉 호출을 줄이고 메모리 사용을 효율적으로 하는 방법이다. 객체를 그리는 호출은 최소로 하고 반복적으로 그려지는 객체는 저장된 정점, 인덱스, 노말, 텍스처 좌표 등을 복사하여 환경 요인에 따라 기하학적으로 변화된 잔디를 화면에 그리게 된다. 셰이더 인스턴싱의 경우 이러한 처리를 CPU가 아닌 GPU에서 처리하게 된다. 그래픽 카드의 성능이 발전하면서 모든 연산 처리를 CPU에만 의존하지 않고 GPU로 분할 처리하면서 연산속도의 효율을 높이는데 활용하게 되는 것이다. 본 논문에서 제안하는 두 가지 방법을 통하여 좀 더 자연스러운 잔디를 실시간으로 렌더링 할 수 있도록 한다.

다음의 2장에서는 게임과 실시간 시뮬레이션 시스템에서의 잔디 표현에 대한 기존 연구들을 고찰하고 3장에서는 프랙탈 이론을 바탕으로 다양한 환경 요인에 반응하는 잔디 성장 모델에 대해 설명한다. 4장에서는 셰이더 인스턴싱의 개념과 이를 잔디 성장 모델에 효율적으로 적용하는 방법을 설명하고 5장에서는 설계한 두 가지 단계의 방법이 실시간으로 작동하는지 여부를 실험을 통해 증명한다. 또한 제안한 성장 모델이 게임이나 실시간 시뮬레이션 시스템에서 자연스러우면서 효율적으로 적용되었는지를 간단한 테스트 프로그램을 통해 분석한다. 그리고 마지막으로 6장에서는 본 논문의 결론을 내린다.

2. 관련 연구

넓은 실외 지형을 사실적으로 표현하는 중요한 요소인 잔디를 표현하기 위한 다양한 연구들이 이루어져 왔다. 기존의 연구들은 항상 실제와 같은 잔디 표현과 빠른 렌더링 처리 사이에서 고민해 왔다. 다음은 이러한 문제들을 바탕으로 다양한 접근을 시도한 기존 잔디 표현에 관한 연구 내용이다.

먼저 3ds Max, Maya와 같은 3차원 저작도구를

사용하여 직접 잔디를 모델링하고 이를 렌더링 하는 방식이다. 전체 지형에 대한 잔디를 직접 다 제작하여 화면에 렌더링하기 때문에 사실적인 표현에는 적합하나 게임과 같은 실시간으로 렌더링 되어야 하는 조건에서는 계산의 부하로 인하여 적용하기 힘든 방식이다[1].

다음은 게임에서 자주 사용되는 빌보드 기법이다. 4개의 정점으로 구성된 사각형 폴리곤에 그래픽 디자이너가 실제 잔디와 같이 제작한 텍스처를 매핑하여 잔디를 표현하는 방법이다. 이는 일정 수의 잔디를 4개의 정점으로 표현하기 때문에 메모리 사용이나 계산 시간을 상당히 절약한다. 많은 수의 잔디를 적은 비용으로 처리하기에 적합한 방식으로 하드웨어의 성능이 떨어지면 예전 게임들부터 현재 출시되는 게임까지 많이 사용되고 있다. 하지만 실제와 같은 잔디를 표현함에 있어서는 많은 한계를 드러내고 있다. 또한 빌보드에 사용되는 텍스처의 경우 조명에 상관없이 하나의 고정된 텍스처를 사용하기 때문에 사실적인 잔디 묘사를 위한 그림자 표현에 한계가 있다[1,3].

빌보드 기법과 다르게 하나의 텍스처가 아닌 빛과 카메라의 방향을 고려하여 생성한 여러장의 텍스처를 사용하는 방법으로 BTFs (Bidirectional Texture Function) 알고리즘이 있다[4]. 이는 미리 제작된 3차원 잔디 객체와 빛과 카메라의 방향에 따라 다른 재질과 형태를 띄는 잔디 텍스처를 사전에 다수 제작하여 저장해 둔다. 실제 렌더링 단계에서는 앞서 설명한 빌보드 기법과 동일하게 4개의 정점을 통하여 사각형을 만들고 텍스처를 월드에서의 빛과 카메라 방향에 따라 유동적으로 변경하여 렌더링한다. 전처리 단계에서 제작된 텍스처는 빛의 방향을 고려하고 있기 때문에 사실적인 그림자 표현도 가능하다[5].

다음으로는 볼륨렌더링 기법으로 의학 분야에서 많이 사용되던 방법이다. 미리 제작된 3차원 잔디 객체를 카메라와 빛의 방향을 고려하여 아주 얇은 여러장의 2차원 슬라이스 이미지로 저장한다. 실제 화면에 잔디를 렌더링할 때 3차원 잔디 객체를 렌더링하는 것이 아니라 미리 제작된 2차원 슬라이스 이미지들을 포개어 3차원 볼륨을 형성하여 렌더링한다. 이는 3차원 객체를 직접 렌더링 하는 것보다 계산 속도를 향상시키며 앞서 설명한 빌보드 방법보다 사실적인 표현이 가능하게 한다. 볼륨 렌더링 방법의

경우 조명을 고려하여 이미지를 제작하기 때문에 동적인 조명에서의 그림자 표현에도 적합하다[1,6].

이밖에도 앞서 설명한 여러 가지 잔디 표현 방법을 종합하여 단계별로 잔디 객체의 세밀함을 분류하여 표현하는 LOD(Level of Detail)기법이 있다. 한 예로 카메라의 시점과 잔디의 위치가 가까울 경우 3차원 객체를 직접 렌더링하고 일정 거리 이상이 되면 볼륨렌더링 기법이나 BTFs기법으로 세밀함을 낮추는 것이다. 이러한 LOD기법은 계산 비용을 효과적으로 줄이고 화면에 보이는 잔디도 사실적으로 표현가능하게 한다[7].

이와 같이 잔디의 사실적인 표현과 그로 인해 발생할 수 있는 연산의 문제를 해결하기 위한 다양한 연구들이 이루어져 왔다. 하지만 앞서 언급된 방법들의 경우 동일한 잔디를 반복적으로 렌더링 함으로써 실제 잔디 무리가 가지는 불규칙성이 존재하지 않는 문제점이 있다. 그렇다고 해서 잔디 모델을 모두 다르게 제작하여 표현하는 것은 렌더링에 큰 문제를 발생시킨다. 따라서 본 논문에서는 기본적인 잔디 객체의 형태는 유지하면서 변화 요소를 통해 다양한 잔디 생성을 가능하게 하도록 프랙탈 기반 잔디 성장 모델을 설계한다. 그리고 설계한 잔디 성장 모델을 효율적으로 렌더링하기 위해 GPU기반에서의 셰이더 인스턴싱 기법을 도입한다.

3. 프랙탈 기반 잔디 모델링

프랙탈은 1975년 만델브로트에 의해 자연을 모델링하는 새로운 도구로서 소개되었다. 프랙탈은 전체를 여러 부분으로 나누었을 때 부분 안에 전체의 모습을 갖는 무한단계에서의 기하학적인 도형을 갖는 성질을 가지고 있으며 유사성과 순환성을 그 특징으로 한다. 프랙탈 이론은 자기유사성과 자기 순환성을 가지는 도형을 밝히는 학문으로 자기 닮음이라는 고유의 특징을 통해 고전적인 유클리드 기하학보다 자연현상을 더 잘 표현할 수 있다는 장점을 지니고 있다[2].

본 논문에서는 이러한 프랙탈의 기본 이론을 바탕으로 자연스러운 잔디 성장 모델을 설계한다. 단 무한단계로의 확장이 아닌 4단계로 한정하여 설정하고 사실적인 잔디를 표현할 수 있는 다양한 변화 요소를 도입한다.

3.1 기본 성장 모델

잔디 모델링의 기본 형태는 4개의 정점으로 구성된 사각형 폴리곤이다. 잔디의 성장은 총 4단계로 단계가 올라감에 따라 잔디의 너비, 높이, 회전 각도가 변화한다. 설계한 잔디 성장 모델은 다음과 같다.

먼저 잔디는 6개 줄기가 하나의 집합(set)으로 구성된다. 따라서 총 6줄기에 관한 초기 위치 값 set_0 , $\{(x_{00}, y_{00}, z_{00}), (x_{10}, y_{10}, z_{10}), (x_{20}, y_{20}, z_{20}), (x_{30}, y_{30}, z_{30}), (x_{40}, y_{40}, z_{40}), (x_{50}, y_{50}, z_{50})\}$ 을 설정한다. 이를 간단히 (x_{si}, y_{si}, z_{si}) 로 표기한다. 여기서 x_{si} 의 s 는 set의 개수(현재 6)로 잔디 성장에 영향을 주는 변수는 아니므로 성장 모델에서는 제외한다. 따라서 초기 설정 값을 기준으로 하여 새로 생성하는 잔디의 정점을 (x_i, y_i, z_i) 라 하고 잔디 성장 모델은 식(1)과 같이 정의한다. 식(1)에서 도입된 다양한 변화 요소를 표현한 것이 그림 1이다.

$$\begin{aligned}
 x_i &= \pm \alpha/2 \pm x_{\text{축 Spread Value}} \\
 y_i &= (y_{i-1} + \beta^* \beta^* \dots \beta^*) \cos \theta - z_{i-1} \sin \theta \\
 z_i &= (y_{i-1} + \beta^* \beta^* \dots \beta^*) \sin \theta + z_{i-1} \cos \theta \pm z_{\text{축 Spread Value}}
 \end{aligned}
 \tag{1}$$

여기서 i 는 성장단계($1 \leq i \leq \text{Maximum Level}(4)$)이고, α 는 너비 제어 값이며, β 는 높이 제어 값이다. 또한 θ 는 회전각 제어 값이며, Spread Value는 집합 내 잔디의 흩어짐 정도를 의미한다. 초기 값 (x_0, y_0, z_0) 이 주어지면 잔디가 성장하면서 다음 좌표 값을 식(1)을 이용하여 계산한다. 먼저 주어진 환경요인에 의해 너비 제어 값 α 가 계산되면 잔디의 x좌표는 $+\alpha/2$, $-\alpha/2$ 로 결정되어 성장단계에서 모든 x좌표는 α 의 너비를 갖게 된다. 다음으로 성장하면서 각 단계마다 새롭게 결정되는 y좌표는 이전 좌표에 누적되는 높이 변

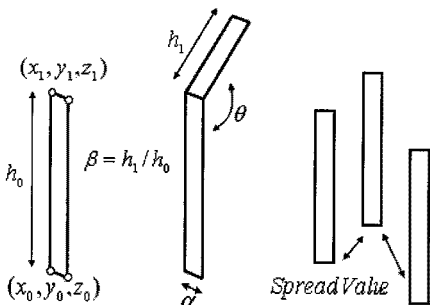


그림 1. 잔디의 변화 요소

화율의 합으로 결정된다. 가령 초기 y좌표가 0이고 β 값이 0.7로 결정되면 $y_0 : 0, y_1 : 0.7, y_2 : 1.19$ 로 계산된다. 여기에 잔디의 회전각 θ 가 정해지면 x축을 중심으로 θ 만큼 회전하여 최종 y, z값이 산출된다.

새로 생성되는 정점의 위치는 이전 좌표 값에서 환경 요인으로부터 결정된 α, β, θ 에 의해 다른 좌표가 결정된다. 여기서 Spread Value는 6개의 줄기 무리가 동일한 간격으로 위치하지 않도록 설정한 값으로 후에 잔디의 밀집도와 병행하여 사용된다.

설계한 잔디의 기본 성장 모델을 기반으로 하여 잔디의 너비, 높이, 그리고 회전각이 변화할 수 있는 다양한 환경 요인을 설정하고 여기에 잔디의 색까지 자연스럽게 변화시켜 사실적인 잔디 객체로 발전시킨다[8-10].

3.2 잔디 성장 모델의 변화 요인

앞서 주변 환경 요인에 반응하는 잔디의 성장 모델을 설계하였다. 여기서 환경 요인이란 실생활에서 볼 수 있는 것과 같이 잔디의 성장에 영향을 미치는 것으로 자연스러운 잔디 모델 설계에 필수 요소이다. 이러한 환경 요인을 본 논문에서는 잔디가 자라나는 토양의 성질과 잔디가 자라는 지형에 비치는 빛의 양, 두 가지로 한정한다.

먼저 토양의 성질 요인으로 범위를 0부터 1사이의 실수 값으로 표현하여 0에 가까울수록 토양의 질이 나쁜 것으로 1에 가까울수록 좋은 환경으로 설정한다. 이는 최종적으로 잔디 잎의 너비, 높이, 회전각, 그리고 색의 변화에 영향을 미치므로 잔디의 최종 외형이 확연히 차이가 나게 될 것이다. 그림 2는 이러한 가정을 바탕으로 토양의 성질을 그래프로 나타낸 것이다.

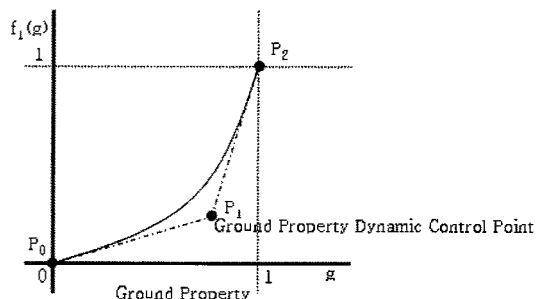


그림 2. 토양의 성질에 따른 변화율

위 그림의 곡선은 아래 식(2)으로부터 유도된 것이다.

$$C(t) = \sum_{i=0}^2 P_i B_i^2(t), 0 \leq t \leq 1$$

P_i : Control Points

여기서 $P_0 = \begin{bmatrix} gx_0 \\ gy_0 \end{bmatrix}, P_1 = \begin{bmatrix} gx_1 \\ gy_1 \end{bmatrix}, P_2 = \begin{bmatrix} gx_2 \\ gy_2 \end{bmatrix}$. (2)

Ground Function $f_i(g) = [gy_0](1-g)^2 + [gy_1]2(1-g)g + [gy_2]g^2$

여기서 제어점 P_i 는 곡선의 형태를 사용자가 조정할 수 있게 하기 위한 것이다. 만약 $gy_0 = gy_1 = gy_2$ 로 설정된다면 토양의 성질과 무관하게 빛의 성질만으로 잔디의 성장이 결정될 것이다.

다음은 빛의 양에 관한 것으로 토양의 성질과 더불어 잔디의 성장에 많은 영향을 미치는 요인 중 하나이다. 빛은 잔디가 자라기에 부족하지도 과하지도 않은 적당한 양을 받아야 잘 성장하게 된다. 이러한 관계를 고려하여 빛의 양을 입력 값으로 하여 잎의 너비에 관한 식(3)과 잎의 높이에 관한 식(4)을 설정하고 이를 그래프로 나타낸 것이 그림 3이다.

$$P_{00} = \begin{bmatrix} wx_0 \\ wy_0 \end{bmatrix}, P_{01} = \begin{bmatrix} wx_1 \\ wy_1 \end{bmatrix}, P_{02} = P_{03} = \begin{bmatrix} wx_2 \\ wy_2 \end{bmatrix}, P_{04} = \begin{bmatrix} wx_3 \\ wy_3 \end{bmatrix}, P_{05} = \begin{bmatrix} wx_4 \\ wy_4 \end{bmatrix}$$
 (3)

Light Function - 1:

$$f_2(l) = [wy_0](1-l)^2 + [wy_1]2(1-l)l + [wy_2]l^2 \quad (0 \leq l \leq 0.3)$$

$$f_2(l) = [wy_2](1-l)^2 + [wy_3]2(1-l)l + [wy_4]l^2 \quad (0.3 < l \leq 1)$$

$$P_{10} = \begin{bmatrix} hx_0 \\ hy_0 \end{bmatrix}, P_{11} = \begin{bmatrix} hx_1 \\ hy_1 \end{bmatrix}, P_{12} = P_{13} = \begin{bmatrix} hx_2 \\ hy_2 \end{bmatrix}, P_{14} = \begin{bmatrix} hx_3 \\ hy_3 \end{bmatrix}, P_{15} = \begin{bmatrix} hx_4 \\ hy_4 \end{bmatrix}$$
 (4)

Light Function - 2:

$$f_3(l) = [hy_0](1-l)^2 + [hy_1]2(1-l)l + [hy_2]l^2 \quad (0 \leq l \leq 0.3)$$

$$f_3(l) = [hy_2](1-l)^2 + [hy_3]2(1-l)l + [hy_4]l^2 \quad (0.3 < l \leq 1)$$

잔디의 경우 빛의 양이 적으면 너비가 좁아지면서 상대적으로 높이는 커지는 걸자라는 특징이 있다. 반대로 빛의 양이 과하면 너비가 넓어지면서 높이는 작아지는 성질로 인하여 빛의 양에 반응하는 너비와 높이의 값을 둘로 나누어 설정한다. 여기서 제어점 또한 토양 성질에서의 제어점과 같은 역할을 한다.

위의 두 가지 환경 요인은 설계한 잔디의 기본 성장 모델에서 잔디의 너비, 높이, 회전각의 변화는

물론 색의 변화에도 영향을 미치게 된다.

3.3 변화 요소 결정

잔디의 기본 성장 모델을 바탕으로 토양의 성질과 빛의 양 두 가지 환경 요인을 통해 변화하는 변화 요소를 정의한다. 여기서 변화 요소는 크게 네가지로 분류된다. 우선 첫 번째 요소인 너비 변화율이다. 이는 잔디 성장 모델의 α 값으로 토양의 성질과 빛의 양 모두에 영향을 받는다. 따라서 설계한 식(2)과 식(3)을 선형 결합하여 최종 값을 얻는다. 여기서 잔디의 높이와 너비와의 비율을 4:1로 설정하여 너비 변화율의 최대값을 1이 아닌 0.25로 설정한다. 다음은 너비 변화율로 이 역시 토양의 성질과 빛의 양에 영향을 받으므로 식(2)과 식(3)을 선형 결합하여 최종 높이 변화율(β)을 계산한다.

세 번째로 회전각 성분의 변화이다. 회전각의 경우 빛의 양과 밀접한 관련이 있으며 잔디의 너비, 높이와도 관계를 갖는다. 상대적으로 너비가 넓은 잔디가 좁은 잔디보다 덜 굽을 것이고 긴 잔디가 짧은 것보다 더 굽는 성질이 있다. 그리고 빛의 양이 부족하거나 과함에 따라 회전각의 변화폭이 결정되므로 이를 반영하여 앞서 결정된 너비 변화율과 높이 변화율 그리고 식(4)을 선형 결합하여 회전각을 구한다. 여기서 식(3)이 아닌 식(4)을 사용하는 것은 빛의 양에 따른 회전각의 변화의 형태가 식(4)과 유사하기 때문이다. 이러한 관계를 고려하여 식(5)을 정의한다.

$$\alpha = (f_1(g) + f_2(l)) / 8, 0 \leq \alpha \leq 0.25$$

$$\beta = (f_1(g) + f_3(l)) / 2, 0 \leq \beta \leq 1$$

$$\gamma = \beta - \alpha + f_3(l)$$

$$\theta = \frac{\pi\gamma}{1.75} * \frac{1}{3}, 0 \leq \theta \leq \frac{\pi}{3}$$

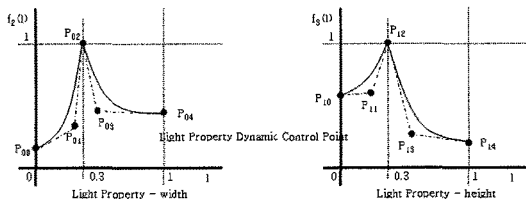
마지막으로 잔디의 색을 고려해보자. 이 역시 토양의 성질과 빛의 양에 의해 변화한다. 식(6)은 잔디의 색을 결정하는 관계를 나타낸 것이다.

$$b = (f_1(g) + (f_2(l) + f_3(l)) / 2) / 2, 0 \leq b \leq 1$$
 (6)

잔디에 입힐 텍스처로는 초기에 잔디가 가장 잘 성장하였을 때의 텍스처 이미지와 가장 못 성장하였을 때의 텍스처 이미지를 저장하고 이 두 텍스처를 구해진 b값을 통해 블렌딩하여 최종색을 결정한다.

$$(1-b)*GT + b*BT =, 0 \leq b \leq 1$$
 (7)

여기서 GT : Good Texture, BT : Bad Texture



(a) 잎의 너비와 관련된 함수 (b) 잎의 높이에 관련된 함수

그림 3. 빛의 양에 따른 변화율

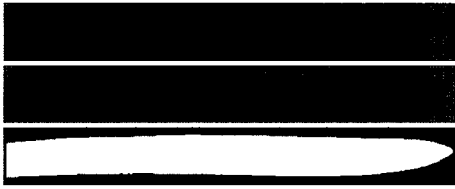


그림 4. 텍스처 이미지

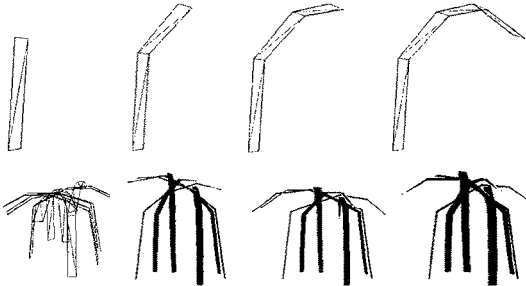


그림 5. 환경 요인을 통한 변화 요소가 적용된 잔디 성장의 예

그림 4는 초기에 입력하는 두 텍스처의 이미지를 나타낸 것이다.

두 가지 환경 요인들로 인한 너비, 높이, 각, 색의 변화를 통해 다양한 잔디의 표현이 가능해진다. 그러면서 동시에 전체적인 잔디의 형태는 유지하면서 변화해 나가는 프랙탈의 자기 유사성을 지닌다. 그림 5는 잔디 성장 모델을 통한 단계별 성장 모습과 변화 요소로 인한 잔디 성장의 예이다.

3.4 B-spline 곡선을 이용한 최종 잔디 설정

앞서 다양한 환경 요인에 반응하는 변화 요소를 통해 4단계로 성장하는 잔디 성장 모델을 설계하였다. 최종적으로 잔디 성장 모델을 통해 결정된 잔디의 정점 정보와 B-spline 곡선을 활용하여 자연스러운 잔디를 설정한다. 잔디 성장 모델의 각 단계별로 생성되는 정점을 제어점으로 하여 B-spline 곡선을 계산하고 곡선의 간격을 일정하게 나누어 자연스럽게 성장하는 잔디를 표현한다. 식(8)은 잔디 성장 모델을 통해 생성된 정점을 바탕으로 B-spline 곡선을 계산하는 과정이다.

$$C(u) = \sum_{i=0}^n P_i N_i^p(u) \tag{8}$$

$$m = n + p + 1,$$

$$U = \{u_0 = u_1 = \dots = u_p, u_{p+1}, \dots, u_{m-p} = \dots = u_{m-1} = u_m\}$$

여기서 P_i 는 잔디 성장 모델을 통해 생성된 정점의



그림 6. B-spline 곡선을 활용한 최종 잔디의 렌더링 예

위치이고 n 은 정점의 마지막 인덱스 정보를 p 는 현재 차수(3)로 이를 통해 knot 벡터의 마지막 인덱스 m 을 계산한다. 이를 통해 knot 벡터 U 를 계산하여 B-spline 곡선을 생성한다. 그림 6은 잔디 성장 모델을 통해 구해진 각 단계별 정점을 바탕으로 계산된 B-spline 곡선으로 최종 잔디를 렌더링한 예이다.

4. GPU 기반의 셰이더 인스턴싱

앞서 설계한 잔디 모델의 경우 그 형상이 환경 요인에 따라 다르게 변화하지만 기본적으로 잔디를 구성하는 정점과 인덱스 정보는 동일하다. 따라서 넓은 지형에 많은 수의 잔디를 여러 번 렌더링 하는 일은 CPU에 부하를 가중하며 불필요한 연산을 반복적으로 수행한다. 이러한 문제를 해결하기 위해 인스턴싱이라는 기법을 사용하여 효율적으로 잔디를 표현할 필요가 있고 이러한 인스턴싱 연산을 CPU가 아닌 GPU에서 처리하게 함으로써 계산 속도를 향상시킬 수 있다[11,12].

인스턴싱이란 동일한 객체를 여러 번 반복적으로 화면에 렌더링 할 경우 드로우 호출을 줄이고 메모리 사용을 효율적으로 하는 방법이다.

렌더링 시 잔디의 정점과 인덱스 등의 정보를 메모리에 넘겨주고 반복적으로 잔디를 렌더링 할 경우 저장해놓은 잔디의 정보를 복사해서 사용한다[13]. 그림 7은 설명한 인스턴싱 정보 흐름의 구조를 나타낸 것이다.

실제 GPU에서 처리하는 셰이더는 크게 버텍스 셰

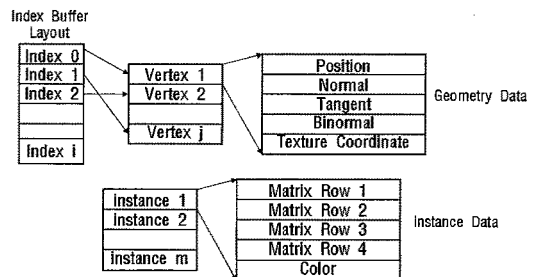


그림 7. 인스턴싱 정보 흐름 구조

이더와 픽셀 셰이더로 두 가지로 나누어 진다. 잔디의 너비, 높이 변화를, 각의 변화율을 통한 잔디 모델의 변화는 버텍스 셰이더에서 앞서 설계한 행렬식을 적용하여 처리하고, 초기에 저장한 텍스처를 구해진 블렌딩 값을 통해 최종 잔디의 색을 결정하는 과정은 픽셀 셰이더에서 처리하게 된다. 이러한 과정을 통하여 광범위한 지형에 다양한 잔디를 효율적으로 처리하게 된다[13].

5. 실험 및 분석

제작한 잔디의 시뮬레이션 프로그램은 Microsoft Visual Studio 2005와, DirectX SDK 9.0c 를 사용하여 개발 되었으며, 실험에 사용된 PC는 Intel Core2Duo CPU E8500, 2GB RAM, Geforce 9600GT GPU를 탑재하고 있다.

5.1 성능 실험

잔디 줄기를 6개씩 한 블록으로 설정하였으며 최소 한 블록(무리)에서 최대 10만 블록까지 처리하였다. 실험은 초당 프레임 수(FPS, frame per second)를 확인하여 실시간 처리가 가능한지 여부를 판단하였고 총 두 단계로 나누어 진행하였다. 첫 번째 단계에서는 4단계까지 성장한 잔디 블록을 하나에서부터 10만 블록까지 렌더링하면서 초당 프레임 수를 기록하였고(표 1), 두 번째 단계에서는 5만 블록과 10만 블록의 잔디를 앞서 설계한 기본 성장 모델의 1단계에서 4단계까지 변경하면서 초당 프레임 수를 확인하였다(표 2). 4단계로 성장한 하나의 줄기는 총 40개의 정점을 가지며, 1블록은 6줄기의 무리로 구성된

표 1. 최종 성장단계에서의 잔디 블록 수 증가에 따른 시뮬레이션 속도

잔디 블록의 수(정점의 개수)	초당 프레임 수(fps)
1(240)	693.05
100(24,000)	548.28
1,000(240,000)	481.10
5,000(1,200,000)	223.76
10,000(2,400,000)	117.30
30,000(7,200,000)	38.47
50,000(12,000,000)	24.70
100,000(24,000,000)	12.46

표 2. 잔디의 성장단계에 따른 시뮬레이션 속도

잔디의 성장단계	초당 프레임 수(fps)	
	5만 블록	10만 블록
1	73.40	39.16
2	45.29	22.01
3	30.88	15.27
4	24.47	12.02

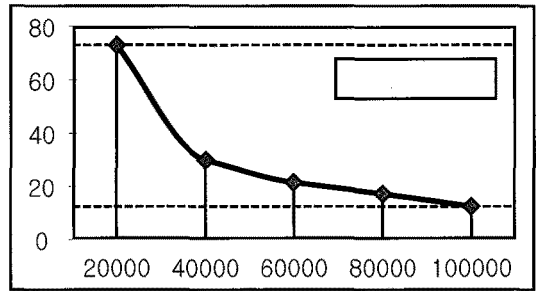


그림 8. 잔디 20000블록 단위 시뮬레이션 속도

다. 또한 그림 8은 잔디를 2만 블록씩 나누어 시뮬레이션한 결과를 나타낸 것이다.

표 1과 그림 8에서 볼 수 있듯이 5만 개의 잔디블록을 렌더링 함에도 실시간 처리가 가능함을 알 수 있고 10만 개의 잔디블록의 경우 인터랙티브한 처리가 가능함을 알 수 있다.

위의 실험을 통하여 셰이더 인스턴싱을 사용하여 수많은 잔디의 실시간 처리가 가능함을 확인하였다.

5.2 실험 시뮬레이션

제안한 두 알고리즘을 적용하여 다양하게 성장하는 많은 수의 잔디 모델을 화면에 실시간으로 렌더링한 예가 그림 9이다. 그림 9는 최대 십만 블록의 잔디를 시뮬레이션한 화면으로 좌측 상단 이미지는 2단계 성장 후 잔디 객체이고 우측 상단은 3단계 성장 후의 이미지이며 우측하단은 4단계 성장 후 이미지이다. 또한 성장한 잔디들의 너비, 높이, 회전각, 색이 환경 요인에 따라 다르게 변화하는 모습도 확인할 수 있다.

5.3 기존 표현방법과의 비교

본 논문에서 제안하는 프랙탈 기반 잔디 성장 모델은 기존의 단순한 패틴의 반복적 잔디 렌더링 방법



그림 9. 실험을 위해 시뮬레이션한 10만 블록의 잔디 성장의 예

을 간단하면서 효율적으로 처리하여 해결하는 것을 목적으로 한다. 너비, 높이, 색 등의 잔디 변화 요소를 설정하고 토양, 빛의 환경 요인을 선택하여 간단한 성장 규칙을 적용함으로써 과거 잔디 표현의 문제점을 손쉽게 극복함으로써 자연스러운 잔디 표현이 가능해진다. 다음 그림 10은 기존의 잔디 표현의 한계와 본 논문에서 해결한 잔디 표현의 차이를 나타낸 것이다.

그림 10-(a)의 경우 기존 잔디의 표현방법으로 조명으로 인한 잔디의 색의 차이는 발생하지만 기본적인 길이, 너비의 외형이 동일하여[7] 조건에 따라 다양한 형태로 성장하는 잔디를 표현하는데 한계가 존재한다. 하지만 그림 10-(b)의 경우 본 논문에서 제안하는 잔디 표현방법으로 성장차이를 보이기 위해 다소 극단적으로 환경 조건을 변화시킴으로써 잔디의 색, 길이, 너비 등의 변화가 확연히 달라졌음을 확인할 수 있다. 이를 통해 사용자가 원하는 환경을 설정하여 조건에 맞는 여러 형태의 잔디를 성장시켜 표현할 수 있음을 확인하였다.



(a) 기존 잔디 표현방법 (b) 제안하는 잔디 표현방법

그림 10. 기존 잔디 표현방법과 제안하는 잔디 표현방법 비교

5.4 응용

본 논문에서는 주변 환경에 반응하며 성장하는 잔디를 설계하였다. 하지만 게임이나 실시간 시뮬레이션 시스템에서 환경 요인은 시간에 따라 끊임없이 변화한다. 다음은 이러한 실시간 시스템에서 효율적으로 잔디 성장 모델을 응용하는 방법의 예이다.

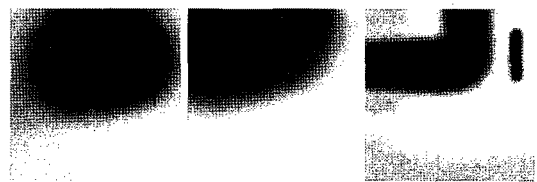
먼저 앞서 전제한 환경 요인인 토양의 성질과 빛의 양 그리고 추가로 잔디의 밀집도 정보를 포함하는 맵 이미지를 제작하여 불러온다. 이는 게임에서 자주 사용하는 높이맵과 유사한 것으로 해당 지역의 환경 정보를 담고 있는 것으로 그림 11이 그 한 예로 밝을수록 토양의 성질이 좋고, 빛의 양은 많으며, 밀집도가 높음을 의미한다.

불러온 환경 요인을 바탕으로 실제 가상 환경에 각기 다른 잔디 무리를 성장시킴으로써 넓은 실외지형에 자연스러운 잔디를 효율적으로 표현할 수 있으며, 그림 12는 이 모든 방법을 종합하여 최종적으로 응용한 잔디 성장 모델의 예이다.

여기에 다양한 3차원 객체를 배치하여 실제 게임과 같은 가상 환경을 구축함으로써 설계한 잔디 성장 모델이 효과적으로 적용됨을 확인해 보았다 그림 13은 3차원 가상공간에 나무, 꽃, 의자 등의 3차원 객체를 배치하여 게임 환경과 유사하게 설정하고 여기에 본 논문에서 제안한 잔디 성장 모델을 앞서 응용한 환경 요인 맵을 기반으로 적용하여 나타낸 것이다.

또한 실제 가상 환경에서 잔디의 실시간 표현이 가능함을 확인하기 위해 잔디의 성장단계와 카메라 위치에 따른 잔디의 초당 프레임수를 확인하였다. 표 3은 이를 나타낸 것으로 카메라와 잔디의 거리와 성장단계가 변하여도 실시간 처리에 큰 무리가 없음을 확인하였다.

실제 본 논문에서는 특별한 LOD 처리를 제안하지 않기 때문에 카메라와 잔디와의 거리가 가까울수록



(a) 토양의 성질 (b) 빛의 양 (c) 밀집도

그림 11. 환경 요인을 표현하는 맵 이미지의 예

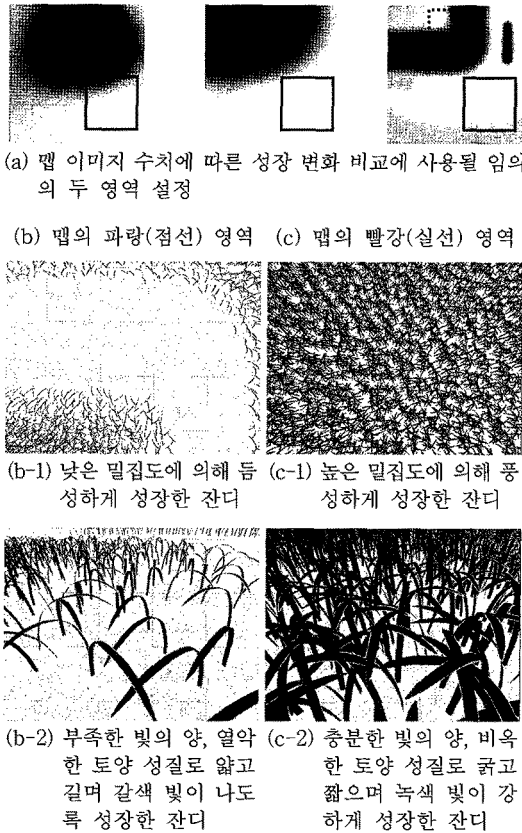


그림 12. 환경 요인에 따라 다르게 성장하는 잔디의 예



그림 13. 가상 환경에 적용한 잔디 성장 모델

표 3. 가상 환경에서의 잔디 시뮬레이션 속도

카메라 위치	성장단계별 초당 프레임 수(fps)			
	1단계	2단계	3단계	4단계
먼 거리	110.46	80.52	41.22	28.70
사람 눈높이	212.88	109.58	85.26	56.25
가까운 거리	301.58	229.46	191.62	150.25

(그림 13의 우측) 렌더링 되는 잔디의 수가 적어 초당 프레임 수가 높아짐을 확인할 수 있다. 또한 카메라 위치를 먼 거리로 잡았을 때에는 잔디의 수가

4만~5만 블록 내외로 제한을 둔 것으로 그 이상의 먼 거리가 될 경우에는 실시간 처리에 다소 어려움이 따를 수 있다. 하지만 이 경우 잔디 각각의 외형이 자세히 보이지 않음으로써 세밀한 표현이 불필요하기 때문에 기존의 빌보드 기법 등을 통해 디테일을 줄임으로써 해결할 수 있을 것이다. 위 실험을 통해 가상 환경에서 성장하는 다양한 잔디의 실시간 처리가 가능함을 확인하였다.

6. 결 론

본 논문에서는 넓은 실외 지형을 사실적으로 표현하기 위한 중요한 요소 중 하나인 잔디를 자연스럽게 시시각각으로 표현하기 위한 하나의 방법을 제안하였다. 동일 형태의 잔디를 반복적으로 렌더링하는 기존 연구들의 한계를 극복하기 위하여 자기유사성을 유지하면서 부분 안에 본래의 모습을 유지하는 프랙탈의 특징을 사용하여 다양한 환경 요인의 영향을 받으며 성장하는 자연스러운 잔디 성장 모델을 설계하였다. 그리고 광범위한 지형을 둘러싸는 많은 수의 잔디를 표현함으로써 발생하는 계산 부하를 최소화하는 방법으로 GPU기반의 셰이더 인스턴싱 기법을 도입하였다. 이 두 가지 알고리즘을 사용하여 계산속도의 문제를 보완하면서 기존의 잔디 모델보다 더욱 사실적인 표현이 가능함을 보였다.

본 연구에서 제안한 방식에 효율적인 LOD를 적용함으로써 더욱 많은 수의 잔디를 효과적으로 관리할 수 있을 것이다. 가령 카메라와 가까운 거리의 잔디 객체는 본 논문에서 제안한 3차원 잔디를 직접 렌더링하고 일정 거리 이상이 되면 볼륨렌더링과 BTFs 알고리즘을 사용하여 메모리 관리나 속도 개선의 방법으로 활용이 가능할 것이다.

그리고 현재 논문에서 제안한 프랙탈 배경의 잔디 성장 모델의 경우 잔디의 종류에 상관없이 단일한 기본 모델만을 설정하였지만 다양한 종의 잔디를 고려하여 설계한다면 좀 더 자연스러운 실외 지형에서의 잔디 표현이 가능할 것이다. 뿐만 아니라 잔디 성장에 영향을 미치는 외부 요인을 본 논문에서 제안하는 두 가지 이외에 다양하게 확장한다면 더욱 사실적인 잔디표현이 가능할 것으로 예상된다.

참 고 문 헌

[1] M.A. Shah, J. Konttinen, and S. Pattanaik
 "real-time rendering of realistic-looking
 grass," Proceedings of GRAPHITE'05, pp.
 77-82, 2005.

[2] K. Falconer, *Fractal Geometry - second edi-
 tion*, Wiley, Hoboken, New Jersey 2003.

[3] K. Pelzer, *GPU Gems, chapter 7 - Rendering
 Countless Blades of Waving Grass*, Addison-
 Wesley, Berkeley, California, 2004.

[4] A. Meyer, F. Neyret, and P. Poulin,
 "Interactive Rendering of Trees with Shading
 and Shadow," Proceedings of the 12th
 Eurographics Workshop on Rendering
 Techniques, pp. 183-196, 2001.

[5] F. Suykens, K. Vom, and A. Lagae,
 "Interactive Rendering with Bidirectional
 Texture Functions," Computer Graphics
 Fourm 22, pp. 463-472, 2003.

[6] P. Decaudin and F. Neyret, "Rendering Forest
 Scenes in Real-time," *Rendering Techniques'
 04*(Eurographics Symposium on Rendering),
 pp. 93-102, 2004.

[7] K. Boulanger, S. Pattanaik, and K. Bouatouch,
 "Rendering Grass in Real-time with Dynamic
 Light Sources and Shadows," SIGGRAPH
 2006(sketch), 2006.

[8] K.J. Vinoy, K.A. Jose, and V.K. Varadan,
 "Fractal Dimension and Frequency Response
 of Fractal Shaped Antennas," IEEE-APS
 International Symposium, pp. 222-225, 2003.

[9] B. Bakay, P. Lalonde, and W. Heidrich, "Real
 Time Animated Grass," Eurographics 2002,

2002.

[10] J. angel and G. Rodriguez, "A Tutorial and
 Recipe for Moving Fractal Trees," *Comput. &
 Graphics*, Vol.22, No.2-3, pp. 301-305, 1998.

[11] E. Millan and L. Rudomin, "Impostors,
 Pseudo-instancing and Image Maps for GPU
 Crowd Rendering," *The International Journal
 of Virtual Reality*, Vol.6, No.1, pp. 35-44, 2007

[12] nVIDIA, <http://nvdeveloper.nvidia.com>, 2007.

[13] DirectX Developer Center, [http://msdn.microsoft.com/ko-kr/directx/default\(en-us\).aspx](http://msdn.microsoft.com/ko-kr/directx/default(en-us).aspx), 2007.



김진모

2006년 2월 동국대학교 멀티미디어
 공학과 졸업(공학사)
 2008년 2월 동국대학교 멀티미디어
 공학과 졸업(예술공학석사)
 2008년 3월~현재 동국대학교 멀티
 미디어학과 박사과정
 관심분야 : 게임 프로그래밍, 게임
 알고리즘, 영상처리, 실시간 렌더링



조형제

1973년 부산대학교 전자공학과
 (학사)
 1975년 한국과학기술원 전기·
 전자공학과(공학석사)
 1986년 한국과학기술원 전기·
 전자공학과(공학박사)
 1986년~현재 동국대학교 멀티미
 디어학과 교수
 관심분야 : 컴퓨터 그래픽스, 게임공학, 컴퓨터비전, 디지
 털사운드처리