

# 절차적 텍스처 제작을 위한 인터랙티브 디자인 시스템

강재구,<sup>o</sup> 조청운, 홍현기  
중앙대학교 첨단영상대학원 영상공학과  
daemon@orgio.net,<sup>o</sup> blue@cglab.cse.cau.ac.kr, honghk@cau.ac.kr

## Interactive Design System for Procedural Texture Generation

Jaekoo Kang,<sup>o</sup> Cheungwoon Jho, Hyunki Hong  
Dept. of Image Eng., Graduate School of Advanced Imaging Science Multimedia & Film, Chung-Ang Univ.

### 요 약

절차적 텍스처링(procedural texturing) 방법은 프로그래밍 가능하게 설계되어 생산성이 높고 유연(flexible)하다는 장점이 있다. 그러나 설계가 복잡하고 프로그래밍에 익숙하지 않은 일반 사용자들이 사용하기 어렵다는 단점이 있다. 본 논문에서는 사용자가 프로그래밍 과정 없이 대화형 인터페이스를 통해 디자인하고 이를 셰이딩 언어(shading language)로 변환해 주는 시스템을 제안한다. 사용자의 디자인 과정으로 여러 개의 기본 노드(node)와 관련된 연산으로 표현되며, 시스템에서 이를 변환하여 텍스처를 기술하는 셰이딩 언어를 생성한다. 제안된 시스템은 일반 사용자에게 절차적인 텍스처링에 대한 직관적이고 유연한 인터페이스를 제공하며, 기본적인 기하 및 텍스처 요소(element)를 조합하여 다양한 텍스처를 효과적으로 생성하도록 한다.

### 1. 서 론

표현하고자 하는 물체의 모양과 텍스처를 효과적으로 생성하는 연구는 컴퓨터 그래픽스에서 매우 중요한 분야이다. 특히 물체의 재질을 표현하기 위한 방법은, 스캐닝되거나 페인팅에 의해 얻어진 2차원 영상을 이용하는 매핑(image mapping)과 프로그래밍을 활용하는 절차적 텍스처링(procedural texturing)으로 구분된다.

이미지 텍스처 매핑은 대상 모델의 표면에 텍스처 공간의 점들을 매핑하는 방법으로 2차원 영상에 대한 텍스처 공간이 일반적으로 사용된다. 이 방법은 제한된 해상도의 영상을 사용하여 샘플링하기 때문에 텍스처가 왜곡 및 변형될 수 있고, 스캐닝된 영상과 다른 해상도에서는 표면에 의도하지 않은 텍스처의 반박이 나타날 수 있다. 또한 대상 영상의 해상도가 낮은 경우, 최종 렌더링 결과 영상의 질이 저하된다(그림 1). 따라서 임의의 표면 공간에 대해 해상도에 관계없이 텍스처로 완전히 표현할 수 있는 절차적 텍스처 방법에 대한 연구가 1980년대 이후부터 꾸준히 진행되고 있다. 절차적 텍스처 방법은 표현하고자 하는 표면의 재질을 알고리즘적으로 결정하는 방법이다. 사용자가 프로그래밍에 관한 지식이 풍부해야 하고, 절차적 방법에 대해 이해를 충분히 해야 한다. 이러한 제약으로 인해 절차적 텍스처에 대한 다양한 연구 보다는 특정 텍스처를 표현하기 위해 주로 사용되고 있으며, 절차적 방법을 이해하고 있는 프로그래머들만이 제한적으로 활용하고 있다. 따라서 보다 다양한 일반 사용자에게 절차적인 텍스처링에 대한 직관적이고 유연한 제작 인터페이스의 필요성이 강조되고 있다.

본 논문에서는 사용자의 직접적인 디자인 과정에 의해 절차적 텍스처가 자동으로 제작되는 시스템이 제안된다. 구성된 시스템에서 사용자의 디자인 과정은 여러 개의 기본 노드(node)와 관련된 연산으로 표현되며, 최종적으로 텍스처를 기술하는 단일 절차의 코드가 자동으로 생성된다. 제안된 시스템은 기본적인 기하 및 텍스처 요소(element)를 조합하여 다양한 텍스처를 효과적으로 생성하도록 하며, 특히 프로그래밍에 익숙하지 않은 일반 사용자가 쉽게 절차적 텍스처링을 하도록 하는 장점이 있다.

본 논문의 구성은 먼저, 2장에서 기존연구와 절차적 방법을 소개한 다음, 3장에서 텍스처의 특징을 정의하는 암시적인(implicit) 방법에 대한 알고리즘을 제시하고, 다양한 텍스처를 만들 수 있는 노드 및 연산의 효과적인 조합방법을 설명한다. 그리고 유연한 인터페이스를 위한 그래프 기반 노드의 절차적 구성 방법을 4장에서 소개한 다음, 제안된 시스템에 의해 생성된 텍스처 및 이를 적용한 영상을 5장에서 제시한다.

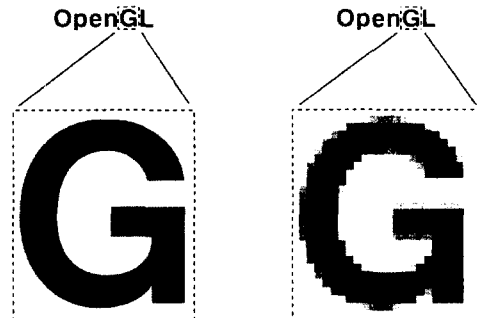


그림 1 절차적 텍스처(왼쪽) 및 이미지 텍스처(오른쪽)

### 2. 관련 연구 및 절차적 방법

대상 모델 표면을 매개변수의 패치(parametric patch)로 표현하여 디지털 텍스처 영상을 매핑하는 방법이 Catmull에 의해 제안된 이후로, 사실적인 텍스처의 표현과 생성에 대한 다양한 연구가 진행되어 왔다[1]. 특히 절차적 텍스처 방법은 표현하고자 하는 표면의 재질을 알고리즘과 수학적 함수를 사용하여 결정한다.

텍스처링과 모델링은 명시적(explicit) 방법과 암시적 방법으로 구분된다. 명시적 방법은 모양을 만드는 점들을 직접 만드는 절차이며, 암시적 방법은 특별한 점에 대한 질문의 대답 절차이다. 절차적 텍스처의 표현에 사용되는 암시적 방법은 그림 2와 같이 대상 모델의 어떤 특정한 위치 (s, t)에서의 색상에

대해 질문을 하고, 절차적으로 이에 대한 해답을 얻을 수 있다. 즉, 렌더러(renderer)가 대상 모델 표면의 색을 결정하기 위해 요구하는 질문에 셰이더(shader)는 절차적 과정으로 해당 색상을 계산한다[2]. 이에 비해 명시적 절차는 2차원 영상내의 고정된 값으로 텍스처 패턴을 만들기 때문에 많은 제약이 존재한다.

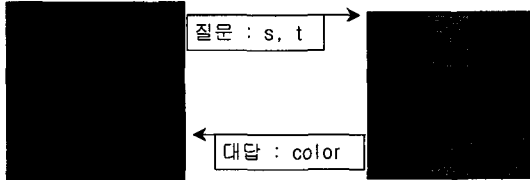


그림 2 텍스처링 대한 셰이딩 과정

### 3. 제안된 시스템

텍스처의 특징은 기본적인 텍스처 요소들의 조합으로 볼 수 있으며, 기본 요소들을 다양하게 조합하여 임의의 텍스처를 생성할 수 있다. 이를 위해 임의의 모양을 암시적 방법으로 정의하는 알고리즘과 다양한 텍스처를 만들 수 있는 노드 및 연산의 효과적인 조합방법이 소개된다.

#### 3.1 암시적 표현을 위한 알고리즘

원, 사각형 등과 같이 폐곡면(closed surface)을 이루는 임의의 모양은 외곽선(boundary), 내부 및 외부 영역 등을 이루는 색상 정보 등에 의해 각 영역이 구분된다. 따라서 임의의 점에 대한 영역을 구분하기 위해 점 분류(classification)하는 알고리즘이 필요하다. 일반적으로 사용되는 Jordan 법칙인 홀수/짝수 교차점(odd/even intersection) 테스트를 사용하여, 특징점의 내부와 외부 판단하여 모양을 정의한다. 또한 반직선/선분의 교점 계산 방법과 반직선/베지어 커브 클리핑 방법[2]을 사용하여, 라인과 커브의 교점과 교차횟수를 계산한다. 2차원 스캐닝 라인과 폐곡면이 교차하는 특징점에서 각 라인이 나 커브의 교점 갯수를 구하면 내부(홀수)와 외부(짝수)를 판단할 수 있고, 이를 이용하여 임의의 단일 모양을 암시적 방법으로 정의한다. 본 논문에서 표현하는 텍스처를 적용할 수 있는 모델의 형태는 라인과 커브로 구성된 폐곡면을 이루는 2차원 물체를 대상으로 한다.

렌더러가 임의의 점에 대한 색을 요구하면, 모양을 구성하는 라인과 커브의 교차점을 계산하여, 영역의 내부/외부를 판단(그림 3의 정선 라인 부분)하고 구성된 모양 정보로부터 결정된 영역의 색을 반환한다.

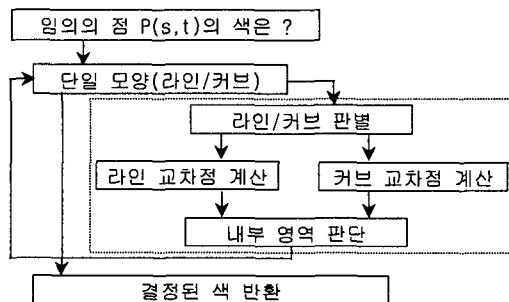


그림 3 임의의 점에 대한 색을 결정하는 절차

#### 3.2 입력된 노드와 연산의 조합 방법

기본적인 기하 및 텍스처 요소를 조합하여 다양한 텍스처를 효과적 생성하는 방법이 제안된다. 표현하고자 하는 특징은 크게 모양과 색으로 구분되며, 임의의 모양을 생성하는 방법과 정해진 형태에 새로운 색을 정하는 방법이 필요하다. 이를 위해 제안된 시스템에서는 사칙연산과 논리 연산 방법 및 2D 변환(이동, 크기, 회전 등)을 사용한다. 또한 구성된 시스템의 기본 연산을 조합하거나 다양한 연산 등을 조합하여 보다 복잡한 특징을 표현할 수도 있다. 연산에 해당하는 요소는 색상 정보를 기본으로 사용하며, 모양을 디자인하기 위해 불린(Boolean) 연산[3]을 이용하고 색상은 논리 연산과 사칙 연산을 사용하여 표현한다.

제안된 방법에서 연산의 조합을 위한 기본 노드는 표 1과 같이 데이터 노드, 연산 노드, 출력 노드로 구성되며, 각 노드에 속성을 설정하여 임의의 모양 및 색상, 연산 등을 정의한다. 여기서 입/출력은 다른 노드와의 연관된 노드간의 데이터 이동을 나타내며, 출력을 제외한 모든 노드는 여러 개의 노드에 의해 레퍼런스(reference)될 수 있다. 입력이 없는 노드는 사용자에 의한 데이터(텍스처 요소)가 저장되고, 입력이 요구되는 노드는 하나 또는 두개의 노드를 레퍼런스하여 연산한다. 출력 노드는 레퍼런스한 노드의 결과를 출력한다.

사용자가 디자인한 대상의 모양 정보를 생성하기 위해 배경과 물체에 각각 검은색 및 흰색의 기본 색을 설정하고, 앞 절의 암시적 표현을 위한 알고리즘을 적용한다. 그리고 사칙 및 논리 연산을 통해 해당 물체의 속성 정보를 표현할 수 있으며, Boolean 연산을 통해 새로운 모양을 생성하기 위한 조합의 기본적인 방법을 그림 4에 보였다.

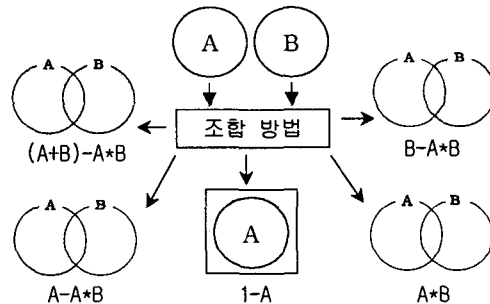


그림 4 모양 생성을 위한 Boolean 연산

표 1 입/출력으로 분류된 그래프 노드.

노드 \ I/O	입력	출력	표현 노드
데이터 노드	없음	다수 가능	텍스처 요소들
연산 노드	1 또는 2	다수 가능	사칙/논리 연산
출력 노드	1	없음	텍스처 저장

### 4. 시스템 설계

텍스처를 효과적으로 제작하기 위해 관련 알고리즘을 적절하게 표현하면서, 동시에 일반 사용자가 절차적 디자인을 하도록 직관적인 인터페이스[4]가 요구된다. 제안된 시스템에서는 단방향성 비순환 그래프 구조를 바탕으로 리스트 자료 구조를 사용하여 표현에 사용된 텍스처를 저장하고, 이진 트리의 깊이 우선 검색 방법을 통해 연산이 절차적으로 수행된다. 텍스처 요소와 연산을 그래프 노드에 정의하고, 절차적 조합을 위해 그래프

노드를 연결하는 직관적인 인터페이스가 제안된다. 시스템은 크게 대화형 디자인의 전반부와 셰이딩 언어 생성[5]의 후반 부분으로 나눌 수 있으며, 대화형 디자인 부분은 사용자의 입력에 의해 텍스처를 디자인하고, 셰이딩 언어 생성의 부분에서는 디자인된 텍스처를 절차적 스크립트로 변환한다.

4.1 대화형 디자인 부분

디자인 부분은 사용자가 데이터 노드에 원하는 노드를 인터랙티브하게 추가하여 텍스처 요소를 정의하게 되며, 데이터 노드를 연산하는 연산 노드를 조합함으로써 프로그래밍에 익숙하지 않은 일반 사용자가 직접 텍스처를 디자인하도록 한다. 근본적인 연결 및 연산을 위해 Base Node(최 상위 클래스) 부분에서 처리가 되어 각 파생된 노드 간에 그래프를 사용하여 표현하게 되고, 각 노드마다의 특성을 부여할 수 있게 된다.

그래프 노드는 표 1과 같이 입력과 출력에 따라 세 가지로 분류할 수 있으며, 각 노드는 절차적 조합으로 그래프를 구성한다. 각 노드는 입/출력에 의해 정의되는데, 텍스처 요소는 데이터 노드에서, 연산은 연산 노드에서 각각 정의되며, 노드로 디자인된 텍스처는 출력 노드에서 코드로 변환(encode)되어 출력된다.

4.2 셰이딩 언어 생성 부분

그림 4에 제시된 것처럼 제안된 알고리즘을 사용하여 임의의 모양을 정의하게 되고, 그림 5과 같은 조합 방법을 통해서 절차적 텍스처가 만들어진다.

이 과정은 텍스처를 디자인한 과정을 그래프 탐색 과정을 통해 절차적 텍스처 코드로 변환하여 절차적 텍스처 파일로 출력하는 과정이다. 출력 노드부터 그래프 탐색을 시작하여, 표현하고자 하는 모양의 결정 알고리즘을 이용하여 그 텍스처 요소에 해당하는 데이터 노드의 코드로 변환하고, 각 관련 노드는 자신의 연산에 해당하는 절차적 코드를 출력하여 조합된 절차로 변환한다. 최종적으로 얻어진 절차적 텍스처를 이후 영상 제작 등에 활용하기 위해 RenderMan Shading Language를 사용하여 출력한다.

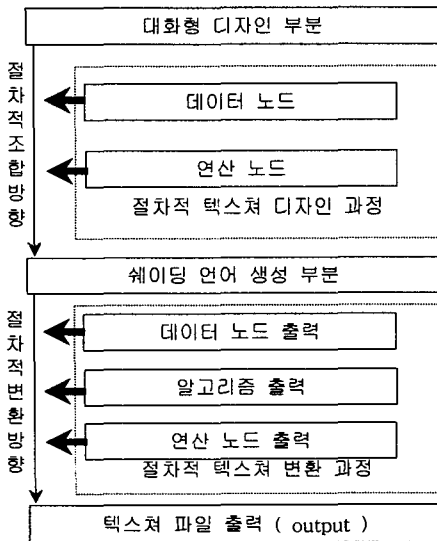


그림 5 전체 시스템 흐름도

5. 결과

[ 제안된 시스템을 사용하여 적용한 예 ]



제작에 사용된 노드 개수  
 데이터 노드 : 페코면 9, 색 2  
 연산 노드 : OR 7, SUB 2, AND 2  
 출력 노드 : 이미지 1



제작에 사용된 노드 개수  
 데이터 노드 : 모양 11, 색 11  
 연산 노드 : OR 9, AND 12, NOT 4  
 출력 노드 :  
 - 이미지 1,  
 셰이더 파일 1

6. 결론

본 논문에서는 사용자의 직접적인 디자인에 의해 절차적 텍스처를 자동으로 제작하는 시스템이 제안된다. 구성된 시스템에서는 기본 노드의 조합을 통해 표현하고자 하는 텍스처를 절차적 과정으로 기술하고, 관련된 코드를 자동으로 생성한다. 제안된 시스템은 절차적인 텍스처링에 대한 직관적이고 유연한 인터페이스를 제공하며, 프로그래밍에 익숙하지 않은 일반 사용자도 쉽게 절차적 텍스처링을 할 수 있다.

7. 감사의 글

본 연구는 교육부 두뇌 한국 21(BK21) 사업과 과학기술부 NRL 사업(2000-N-NL-01-C-285) 지원에 의해 수행되었습니다.

8. 참고 문헌

[1] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, Steven Worley, *Texturing & Modeling A Procedural Approach Second Edition*, AP Professional, July 1998, page 8~11  
 [2] Anthony A. Apodaca, Larry Gritz, *Advanced RenderMan*, July 2002 page 170~194  
 [3] T. Nishita, Thomas W. Sederberg, Masanori Kakimoto, " Ray tracing trimmed rational surface patches," *ACM SIGGRAPH Computer Graphics*, v.24 n.4, p.337~345, Aug. 1990  
 [4] Ken Perlin, An image synthesizer, *ACM SIGGRAPH Computer Graphics*, v.19 n.3, p.287~296, Jul. 1985  
 [5] Nothing Real Shake(ver 5.1080) (design of GUI )  
<http://www.apple.com/shake/>  
 [6] Hanrahan90 Pat Hanrahan, Jim Lawson, A language for shading and lighting calculations, *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, p.289~298, September 1990