

# 실시간 클레이메이션 스타일 렌더링

신호석, 한정현  
성균관대학교 정보통신공학부  
e-mail:shs55@nate.com

## Real-Time Claymation Style Rendering

HoSuk Shin, JungHyun Han  
School of Information and Communications Engineering  
Sung Kyun Kwan University

### 요 약

본 논문에서는 실시간 클레이메이션 스타일 렌더링 기법을 구현하였다. 클레이메이션의 특징은 완만한 모서리와 표면 자국 효과 그리고 색상별로 점토를 덧붙여 만들었다는 데 있다. 색상영역 별로 메쉬의 정보를 변화시키기 위하여 색상영역을 계산하여 디스플레이스먼트 맵을 생성하고 적용하였으며, 모서리 부분을 굴곡화시키고 디스플레이스먼트 맵을 적용하기 위하여 텍셀레이션을 적용하였다. 표면 자국 효과를 위하여 Perlin 노이즈 기법을 사용하여 범프 맵을 생성하고 적용하였다. 다른 비사실적 렌더링 기법과 마찬가지로 클레이메이션 스타일 렌더링 기법은 게임이나 광고 등에 적용가능하다.

### 1. 서론

비사실적 렌더링 기법(Non-Photorealistic Rendering)은 기존의 사실적인 렌더링 기법과는 다르게 표현의 목적이 시뮬레이션이 아닌 예술적인 부분에 초점을 맞추고 있다. 이 분야는 최근 하드웨어 구조의 획기적인 발전과 함께 실시간 구현이 가능해지면서 연구가 활발해지고 있다. 유화, 수묵화, 스케치 등의 예술적인 미술 기법이나 만화 등과 같은 분위기의 렌더링 기법들이 비사실적 렌더링 기법에 포함된다. 본 논문에서는 기존의 메쉬를 사용하여 실시간으로 클레이메이션 스타일의 장면을 렌더링 하는 알고리즘을 구현하고자 한다. 클레이메이션(Claymation)이란 Clay와 Animation의 합성어로 점토를 사용하여 만들어진 오브젝트들을 카메라로 촬영하고 이 장면들을 연결하여 애니메이션을 만드는 것을 말한다. 클레이메이션을 사용한 대표적인 사례는 Nick Park의 “웰레스와 그로밋”, “치킨런”, Tim Burton의 “크리스마스의 악몽” 등과 같은 영화들이 있다. 클레이메이션은 각 장면마다 수작업으로 캐릭터들을 조정하고 카메라로 촬영하는 반복 작업을 하

기 때문에 적은 분량의 애니메이션을 만드는 데에도 많은 기간과 작업이 필요하므로 게임 등과 같은 분야에서는 적용하기 힘들다. 기존의 메쉬를 사용하여 이와 비슷하게 렌더링한다면 많은 분야에서 적용가능할 것이다.

앞으로 본 논문은 클레이메이션의 특징을 알아보고, 이를 구현하기 위한 알고리즘에 대해 말하고자 한다.

### 2. 클레이메이션의 특징

[그림 1]은 클레이메이션으로 구현된 “웰레스와 그로밋”의 한 장면이다.



[그림 1] Nick Park의 “웰레스와 그로밋”

## 2.1 표면 자국

클레이메이션은 점토로 모양을 만들기 위하여 손이나 도구를 사용하여 주무르고 뭉쳐서 캐릭터를 제작한다. 그렇기 때문에 캐릭터의 표면에 손이나 도구에 의한 울퉁불퉁한 표면 자국들이 생성된다.

## 2.2 완만한 모서리

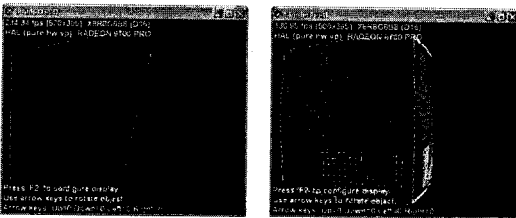
점토로 만들어진 캐릭터의 커다란 특징은 전체적인 느낌이 둥글둥글하다는 것이다. 모서리 부분 역시 완만한 곡선 형태를 취하고 있다.

## 2.3 색상이 다른 점토를 서로 덧붙임

점토는 기본 바탕이 되는 부분을 만들고 그 표면 위로 다른 색상의 점토로 세부적인 부분을 만들어 붙인다. 예를 들어, [그림 1]과 같은 장면을 기존의 렌더링 방식으로 만든다면 가슴에 해당하는 메쉬에 넥타이와 조끼 이미지를 가진 텍스처를 맵핑한다. 하지만, 클레이메이션으로 표현한다면 넥타이와 조끼에 해당하는 부분은 위로 튀어나와서 마치 위로 덧붙인 듯한 장면을 만들어야 한다.

## 3. 텍셀레이션

모서리를 완만한 곡선면으로 변형하려면 정점의 추가 생성이 필요하다. 본 논문에서는 Alex Vlachos가 제시한 Curved PN Triangles[10]라는 텍셀레이션 기법을 사용하여 구현하였다. 이 기법은 하나의 삼각형에서 각 정점의 노말 벡터를 사용한 탄젠트 계수를 고려하여 텍셀레이션을 수행하기 때문에 높은 레벨의 텍셀레이션을 수행할 경우 하나의 삼각형을 곡선면으로 만들어준다. [그림 2]는 본래의 메쉬 모델과 이를 Curved PN Triangles 기법을 사용하여 텍셀레이션을 수행한 결과이다.



[그림 2] 본래 메쉬(좌), 텍셀레이션 적용한 메쉬(우)

## 4. 색상 영역에 따른 디스플레이스먼트 맵 생성

점토는 색상별로 뭉쳐 서로 붙여나가는 방식이므로 텍스처가 맵핑되는 메쉬를 색상별로 나누어 정점의 위치 정보를 변화시켜야 하며 이를 위하여 디스플레이

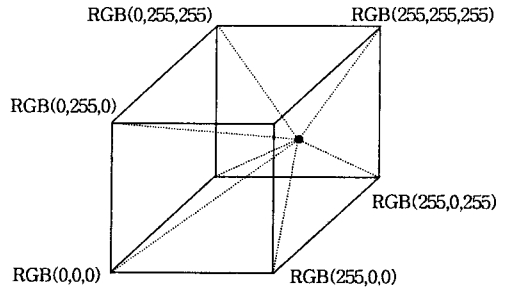
이스먼트 맵을 생성한다.

다음은 디스플레이스먼트 맵을 생성하는 과정이다.

4.1. 메쉬에 맵핑될 텍스처에 대해서 색상별로 영역을 구분한다.

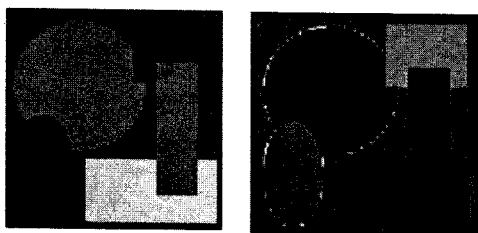
4.1.1. 먼저 몇 개의 대표적인 색상을 정한다. 이는 대부분의 텍스처가 수많은 색상을 내포하므로 그것들을 모두 구분하여 디스플레이스먼트 맵을 생성하는 것은 수많은 색상 영역을 만들어내어 효율적이지 못하기 때문이다.

4.1.2. 텍셀을 차례대로 참조하여 얻는 색상과 가장 비슷한 대표 색상을 선택한다. 색상을 3차원적으로 해석하여 텍셀의 색상과 대표 색상들의 거리를 구하여 가장 가까운 거리에 있는 대표 색상을 선택한다.



[그림 3] 텍셀과 대표색상들과의 거리 계산

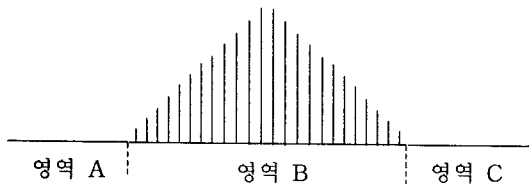
4.1.3. 텍셀을 색상영역 별로 구분한다. 첫 번째 텍셀이 해당하는 영역의 텍셀들을 먼저 찾아내어 영역 아이디를 부여한다. 하나의 영역에 대해 검사가 끝나면 영역 아이디가 부여되지 않은 텍셀에서부터 다시 새로운 영역에 대한 검사를 시작한다. 영역검사는 현재의 텍셀에서 주위 4방향(상하좌우)을 일정한 규칙(예. 시계방향)에 따라 검사하여 영역 아이디가 부여되지 않았고 현재 텍셀과 같은 색상 영역이라고 판단되면 해당 텍셀로 이동하여 다시 검사하는 방식으로 수행한다. 이렇게 인접 영역을 따라 검사하는 것은 무조건적인 색상 검사를 통해 잘못된 영역 판단을 내리는 것을 방지하기 위해서이다. [그림 4]은 본래의 텍스처와 이를 사용하여 색상 영역을 판단한 결과이다. 서로 다른 색상은 영역이 다름을 보여주고 있다.



[그림 4] 본래 텍스처(좌), 색상 영역 판단(우)

4.2. 앞에서 생성한 영역 정보를 사용하여 디스플레이스먼트 맵을 생성한다.

4.2.1. 첫 번째 텍셀부터 상하좌우 4방향으로 체크를 하며 한 방향에 대해서 동일한 영역 아이디를 가진 텍셀의 수를 체크한다. 4방향에 대해서 모두 체크가 끝나면 4방향 중 가장 적은 텍셀 수를 현재 텍셀의 최소 이웃 텍셀 수로 설정한다. 각 영역 별로 설정된 최소 이웃 텍셀 수는 [그림 5]와 같이 영역의 중심으로 갈수록 선형적으로 증가하고, 반대로 테두리 부분으로 갈수록 감소된다.



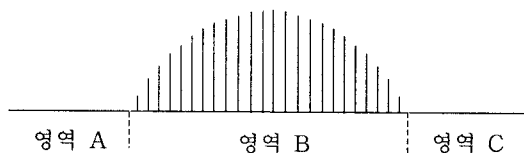
[그림 5] 영역에 따른 텍셀 당 최소 이웃 텍셀 수의 변화

4.2.2. 클레이메이션의 오브젝트들은 튀어나온 부분들이 완만한 곡선을 이루고 있기 때문에 [식 1]과 같이 log 연산을 적용하여 디스플레이스먼트 팩터를 설정한다. [그림 6]은 [식 1]을 적용하여 영역에 따른 디스플레이스먼트 팩터들이 곡선을 이루며 부드럽게 변화하고 있음을 보여주고 있다.

$$\text{최소이웃텍셀수} \leq 1 : 0$$

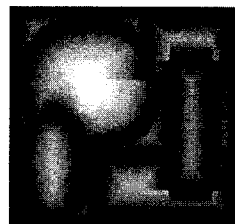
$$\text{최소이웃텍셀수} > 1 : \log(\text{텍셀의 최소이웃텍셀수})$$

[식 1] 완만한 디스플레이스먼트 팩터의 계산



[그림 6] 영역에 따른 디스플레이스먼트 팩터들의 변화

[그림 7]은 디스플레이스먼트 팩터들을 사용하여 생성한 디스플레이스먼트 맵이다. 밝은 부분은 디스플레이스먼트 팩터가 크고, 반대로 어두운 부분은 팩터가 작다는 것을 보여준다. 현재 텍셀의 주위 4방향만을 고려하였기 때문에 전체적으로 디스플레이스먼트 팩터가 부드럽게 변하지 못하고 있다.

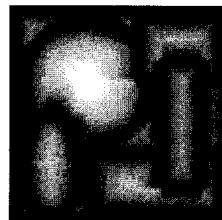


[그림 7] 디스플레이스먼트 맵

4.2.3. 디스플레이스먼트 맵을 전체적으로 부드럽게 변화시키기 위하여 본 논문에서는 가우시안 필터링 기법을 사용하였다. [그림 8]은 적용된 5 x 5 가우시안 필터이고 [그림 9]은 가우시안 필터링을 사용하여 전체적으로 부드러운 팩터의 변화를 보여주고 있는 디스플레이스먼트 맵이다.

1	1	2	1	1
1	2	3	2	1
2	3	4	3	2
1	2	3	2	1
1	1	2	1	1

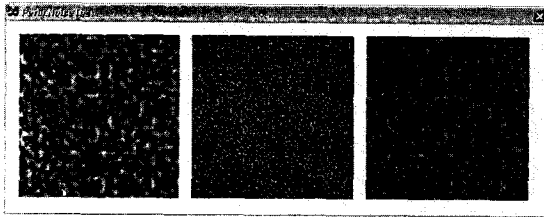
[그림 8] 5 x 5 가우시안 필터



[그림 9] 가우시안 필터링을 적용한 디스플레이스먼트 맵

### 5. 노이즈를 사용한 범프 매핑

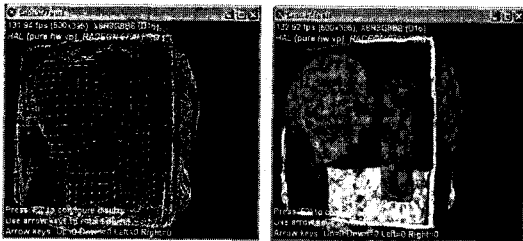
본 논문에서는 표면 자국 효과를 나타낼 수 있는 노이즈 함수를 구현하여 질차적 기법으로 범프 맵을 생성한다. [그림 10]은 Perlin 노이즈를 사용하여 표면 자국 효과를 구현한 것이다. 여기에서는 서로 다른 파형을 갖는 두 개의 노이즈들을 생성하고 이들을 혼합하여 최종적으로 부드러운 노이즈를 만들어 낸다.



[그림 10] 노이즈를 사용한 표면 자국 효과

### 6. 최종 렌더링 결과

텍셀레이션과 디스플레이스먼트 맵 생성, 범프 맵 생성은 실제 렌더링 초기에 수행되고, 렌더링 시에는 위에서 생성된 디스플레이스먼트 맵과 범프 맵만을 사용하여 렌더링 되기 때문에 실시간 클레이메이션 렌더링이 가능하다. [그림 11]은 최종적인 클레이메이션 스타일 렌더링 결과이다.



[그림 11] 최종적인 렌더링 결과

### 7. 결론

게임이나 영화 등에서 카툰 렌더링과 같은 비사실적 렌더링 기법이 점점 널리 쓰이고 있다. 이는 만화나 그림과 같은 비사실적인 이미지가 사람들에게 사실적인 이미지에서 느끼지 못하는 즐거움을 주기 때문이다. 따라서 비사실적 렌더링 분야는 앞으로 많은 가능성을 가지고 발전할 것으로 보여 진다. 본 논문에서 소개한 클레이메이션 스타일 렌더링은 아직까지는 미흡하지만 연구 가치가 있는 기법이고 머지않아 카툰 렌더링과 같이 게임 등에서 클레이 스타일 렌더링을 사용할 가능성이 있다고 보여 진다.

### 참고문헌

[1] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A Non-photorealistic Lighting Model for Automatic Technical Illustration. In *SIGGRAPH 98 Conference Proceedings*, pp. 447-452

[2] Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Loring S. Holden, J. D. Northrup, and F. Hughes. Art-based Rendering with Continuous Levels of Detail. In *Proceedings of the First International Symposium on Non Photorealistic Animation and Rendering (NPAR) for Art and Entertainment*, June 2000. Held in Annecy, France.

[3] Robert L. Cook. Shader Trees. *Computer Graphics (Proceedings of SIGGRAPH 84)*, 18(3):223-231, July 1984, Held in Minneapolis, Minnesota.

[4] Lee Markosian, Michael A. Kowalski, Samuel J. Trychin, Lubomir D. Bourdev, Daniel Goldstein, and John F. Hughes. Real-Time Nonphotorealistic Rendering. *Proceedings of SIGGRAPH 97*, pp. 415-420

[5] Aditi Majumder, M. Gopi, Hardware Accelerated Real Time Charcoal Rendering, *Proceeding of NAPR 2002*, pp. 59-66.

[6] Emil Praun, Hugues Hoppe, Matthew Webb, Adam Finkelstein, Real-Time Hatching, *Proceeding of SIGGRAPH 2001*, pp.581-586.

[9] B. J. Meier, Painterly Rendering for Animation, In *Computer Graphics (SIGGRAPH 96)*, pp. 477-484.

[10] Alex Vlachos, Jörg Peters, Chas Boyd, Jason L. Mitchell, Curved PN Triangles, In *ACM Symposium on Interactive 3D Graphics 2001*, pp. 159-166.