

# 점묘 기법을 이용한 상호적 비실사 기법

한만준\*, 오세윤, \*임순범, 최윤철

연세대학교 컴퓨터과학과

\* 숙명여자대학교 멀티미디어과학과

{hama153, seyunee, ycchoy}@rainbow.yonsei.ac.kr

sblim@sookmyung.ac.kr

## Interactive Non-Photorealistic Rendering Using Pointillism Techniques

Man-Jun Han\*, Se-Yoon Oh, \*Soon-Bum Lim, Yoon-Chul Choy

Department of Computer Science, Yonsei University

\*Department of Multimedia Science, Sookmyung Woman's University

### 요 약

지금까지의 컴퓨터 그래픽스의 방향은 사진과 유사하게 표현하는 Photorealism이 강세였으나, 최근 각종 기술문서나 의학, 교육등 많은 분야에서 사진과 같은 컴퓨터 그래픽스보다는 사물의 특징을 강조하고, 불필요한 부분을 생략해서 표현하는 비사실적 렌더링(Non-Photorealistic Rendering)분야가 새롭게 떠오르고 있다. 비실사기법을 통해 생성된 이미지는 사물의 특징을 부각시켜 시현함으로써 사진과 같은 사실적인 묘사보다는 사물의 특징을 통한 의미전달에 초점을 둔 렌더링 기법이라 할 수 있다. 비실사기법에는 수채화, 수묵화, 유화, 목탄화, 데생과 같은 미술기법과 만화와 같이 표현하는 Cartoon Rendering등 다양한 분야가 있지만 본 논문에서는 모든 그림의 기본 요소가 되는 점을 이용하여 사물을 표현하는 점묘법을 적용한 비실사기법에 대해서 이미 발표된 논문을 살펴보고, 새로운 기법을 제안하고자 한다.

### 1. 서 론

최근의 컴퓨터 그래픽스의 주류는 고품질의 사진과 같이 표현하는 Photorealism이라 할 수 있다 그러나 최근에는 이러한 주류에 대한 새로운 부류로서 비사실적 렌더링(Non-Photorealistic Rendering) 분야가 각광을 받고 있다. 비실사기법에서는 기존의 컴퓨터 그래픽스가 사진과 같은 이미지를 생성하는 것에 초점을 두었다면, 이와는 달리 각종 미술적 기법을 적용하여 이미지의 의미전달 측면뿐만 아니라 예술적인 측면을 표현하는 기법을 말한다. 컴퓨터 그래픽을 이용하여 사람이 직접 그린 듯한 수채화나 수묵화 뿐만 아니라, 유화, 목탄화, 연필 데생, 제품 설명서에 사용되는 일러스트레이션이나 셀 애니메이션과 같은 효과를 나타내는 카툰 렌더링(Cartoon Rendering)등이 이에 속한다고 할 수 있다. 이러한 많은 분야 중에 모든 그림의 기본요소라 할 수 있는 점을 이용하여 사물을 표현하는 점묘법을 이용한 비실사기법을 제안하고자 한다. 우리가 접하는 모든 사진과 같은 인쇄물들을 확대해서 보게 되면 점으로 구성된 것을 알 수 있다. 즉 모든 그림은 점으로 구성된다고 할 수 있다. 점묘법은 pen-and-ink 일러스트레이션 기법의 하나로 점의 위치에 따라 미묘한 차이가 나기 때문에 적절한 위치에 그려야 한다.

전형적인 점묘법은 많은 시간을 투자해야 하는 작업인데 반하여 컴퓨터가 생성해 내는 이미지에서는 매우 매력적인 면을 가지게 된다. 점은 모든 사물의 기본 요소로서 어떠한 렌더링 환경에서도 적합한 특징을 가진다. 또한 점은 렌더링 하기에 가장 단순하고 빠른 요소이다.[1]

비실사기법은 아무리 복잡한 사물이라 하더라도 간단히 표현하기 때문에 사물에 대한 이해를 위해서는 가장 강력한 도구이다. 제안하는 시스템에서는 기존 방법과는 달리 렌더링 과정을 단순화 하여 속도를 향상시켰다.

### 2. 관련 연구

Non-Photorealistic Rendering에서는 주로 Lines, Curves, Strokes가 주로 사용된다. Praun et al.[2]은 굴곡면의 방향에 따라 texture를 겹쳐서 표현하는 실시간 hatching 시스템을 발표하였다. Adam Lake[3]는 2차원 texture mapping을 이용하여 Pencil sketch Shading기법을 구현하였다. 기존의 비실사기법이 batch processing을 주류를 이루었으나, Lee Markosian[4]은 속도 향상을 위해 정확성과 묘사성을 적절히 조절하여 real-time nonphotorealistic renderer를 구현하였다.

Deussen et al[5]은 전통적인 점묘법을 구현하는데 있어서 pen-and-ink illustration을 위해 점을 사용하였다. 그의 방

본 연구는 한국과학재단 목적기초연구 (R01-2004-000-10117-0(2005)) 지원으로 수행되었음.

법은 Polygon model을 tone image로 변환한 후 다시 점 이미지로 변환하는 방법을 이용하였다. 이렇게 함으로써 복잡한 이미지도 생동감 있게 표현하였으나, 이 방법은 surface rendering으로 입체감을 표현하는데 있어서는 부족한 면이 있었고, 또한 interactive하게 표현하기에는 속도가 느린 단점을 가지고 있었다. Levoy and Whitted[6]는 점이 시각적 기본요소임을 처음으로 보였고, 점으로 구성된 표 형태의array로 평면을 표현하듯이 3차원 표면도 표현할 수 있음을 증명하였다. 이러한 점의 사용은 interactive한 rate로 복잡한 모형과 그림자들을 고수준의 이미지를 생성할 수 있다.[7]

앞에서 살펴본 여러 논문들은 점묘법 표현에 있어서 표현 이미지에 초점이 맞추어져 있는데 반하여 본 논문에서 제시하는 기법은 입체감을 고려하여 보다 이해하기 쉬운 이미지를 생성하는데 있다. 다음 장에서는 점묘법을 구현하기 위한 기본 알고리즘을 살펴보고 이를 단계별로 살펴보고자 한다.

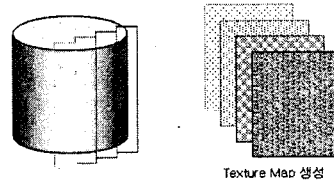
### 3. 점묘 기법

점묘법이란 19세기 쇠라(Georges Seurat, 1859~91), 시냐크(Paul Signac, 1863~1935) 등을 중심으로 행해진 점묘주의 등의 이론과 수법 및 그 운동을 말한다. 이들 신인상주의 학파의 그림들은 캔버스에 색칠을 해나갈 때 순색만을 사용하되 이를 일체 뒤섞지 않으면서 작은 점으로 찍어나가는 방법을 말한다. 이 경우 색조의 순도는 그대로 유지하면서도 보는 이의 망막 위에서 중간색이 형성되는 효과를 낳게 된다. 이러한 병치법을 이용하는 기법은 차후 연구 방향으로 남겨두고, 본 논문에서는 음영과 원근감을 점으로 표현하는 점묘법을 구현하고자 한다. 점묘법의 기본 알고리즘은 다음과 같다. 점의 개수의 많고 적음에 따라 명암을 표시하게 되고, 가깝고 먼 것에 따라 톤이 결정된다. 즉 가까운 거리에 위치한 사물은 어두운 점으로 표현되고, 먼 곳의 사물은 밝은 점으로 표현되게 된다. 본 논문에서 제시하는 기법은 Preprocessing 과정에서 명암과 관련된 2D texture map을 생성하고, 입체감 표현을 위해 깊이버퍼 이미지를 이용하여 손으로 그린 것과 같은 점묘법을 구현하고자 한다.

#### 3.1 전처리기

렌더링 하기 전에 preprocessor는 명암과 관련해서 여러개의 2D texture map을 자동으로 생성한다. 밝기 단계에 따라 점의 개수를 다르게 하여 랜덤하게 분포되는 점 texture map을 생성하게 된다. 이때 같은 밝기에 같은 모양의 texture map을 사용하게 될 경우 패턴의 통일성으로 그림이 부자연스럽게 될 수 있기 때문에, 같은 밝기라도 여러 가지 패턴의 점 분포를 갖는 texture map을

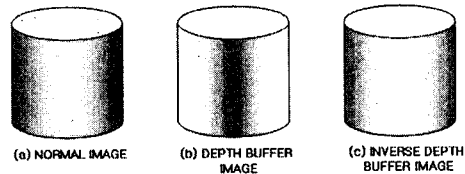
생성하게 된다.



[그림 1] 전처리기 - Texture Map 생성

#### 3.2 입체감 표현

Preprocessing 단계에서 생성된 texture만으로 polygon model에 mapping하는 경우에는 단순히 음영에 따라 점의 개수가 많고 적음이 반영되어 평면상의 점의 분포로만 표현되므로, 부자연스러운 결과물을 얻어내게 된다. 이를 보완하여 보다 입체감 있게 사물을 표현하기 위해서 본 논문에서는 깊이버퍼의 정보를 사용한다. 깊이버퍼에는 사물과 View Port와의 거리 정보가 저장되어 있는데, 이를 이용하게 되면 가까운 거리의 사물은 검게, 먼 곳의 사물은 흰색에 가깝게 표현되어 있다. 이를 통하여 원근감과 사물의 볼륨감을 느끼게 된다. 전 단계에서 생성된 2D texture map을 맵핑한 후에 깊이버퍼 image를 적용함으로써 사물을 보다 볼륨감 있게 표현하게 된다.



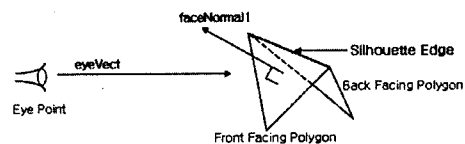
[그림 2] 깊이버퍼 Image

#### 3.3. 외곽선 표현

여러 가지 외곽선 추출기법 중에서 Eye Vector와 Face Normal Vector의 Dot product 연산을 이용한 방법[수식 1]을 적용하였다. Silhouette은 선을 이용하여 rendering되고, 빛에 대한 파라메타에 따라 선의 두께를 조절할 수 있다.[8]

$$(faceNormal_1 \cdot eyeVect) * (faceNormal_2 \cdot eyeVect) \leq 0$$

[수식 1] Silhouette edge detection



[그림 3] Silhouette edge detection

### 3.4. 결과 이미지

전처리기에서 생성된 2D Texture map과 깊이버퍼 이미지를 혼합한 영상에 사물의 명확성을 증가시키기 위해서 외곽선을 추가함으로써 결과 이미지를 생성 하였다. 결과 이미지를 생성하기 위한 공식은 다음과 같다.

$$IM_{total} = IM_{texture} * (1 - IM_{depth}) + IM_{edge}$$

여기서,  $IM_{total}$ : Final Image

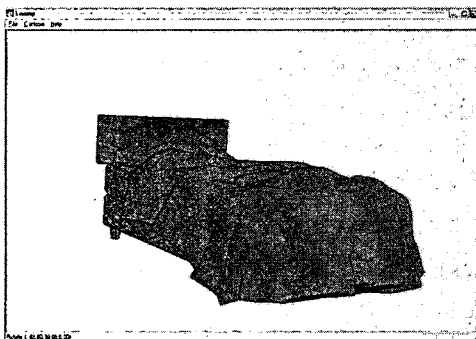
$IM_{texture}$ : 2D Texture Image

$IM_{depth}$ : 깊이버퍼 Image

$IM_{edge}$ : Silhouette Edge

### 4. 구현 결과

구현은 카툰렌더링 Loony 프로그램의 소스를 응용하였고, 계획과는 달리 깊이버퍼 이미지를 사용하지 못하여 볼륨감이 없이 surface rendering한 결과물을 얻게 되었다. 이를 보완하기 위해 멀티 텍스처링을 적용하였을 경우에는 비교적 점묘화에 가까운 이미지를 얻을 수 있었지만 기존에 발표된 기술을 그대로 이용한 것이라 의미가 없다고 생각되었고, 생성된 이미지의 음영을 보다 부각시키기 위해서 피보나치 수열을 적용하여 구현한 결과 의도한 것과 유사한 결과물을 얻을 수 있었다. 처음 의도한대로 처리속도는 상당히 빠르게 나타났고, 대신 짐 밀성은 떨어지게 나타났다.



[그림 4] 구현결과

### 5. 결론 및 향후 과제

지금까지 점묘법을 이용한 실시간 렌더링 기법에 대해서 제안하였다. 과거에도 점묘법을 이용한 기법들이 소개되었지만, Deussen의 기법은 볼륨감을 표현하는데 부족한 면을 보였고, Aidong Lu가 제안한 기법의 경우 생성된 이미지의 퀄리티는 높은 반면 고려요소가 많음으로서 계산량이 증가되었고, 이로 인해 속도가 저하되어 Interactive한 효과를 가져 오지 못하게 되었다. 하지만 Aidong Lu의 기법에 있어서 점

묘법으로 생성된 이미지의 내외부선을 명확하게 하는 알고리즘을 추가하여 보다 정확한 이미지를 생성해 낼 수 있었다. 비록 제안하고자하는 기법이 Aidong Lu의 결과물에 비해 명확한 이미지를 제공하지는 못하지만, 처리요소를 단순화 시키므로 해서 속도를 증가 시킬 수 있었고, 보다 Interactive한 이미지를 생성 할 수 있었다. 현재의 기법은 카툰 렌더링한 이미지에 점묘법을 이용하여 명암 및 톤을 구현하였으나, 차후에는 이를 보다 발전 시켜서 19C 인상파의 그림과 같이 병치법을 적용하여, 모자이크 한 듯한 이미지를 생성하게 하는 것도 좋은 방법이라 생각된다.

### [참고문헌]

- [1] Lu, A., Morris, C., Ebert, D. S., Rheingans, P., Hansen, C., "Non-Photorealistic Volume Rendering Using Stippling Techniques," IEEE Visualization 2002
- [2] E. Praun, H. Hoppe, M. Webb, and A. Finkelstein. Real-time hatching. In Proceedings of ACM SIGGRAPH 2001, Computer Graphics Proceedings, Annual Conference Series, pages 579~584, August 2001.
- [3] Adam Lake , Carl Marshall , Mark Harris , Marc Blackstein, Stylized rendering techniques for scalable real-time 3D animation, Proceedings of the first international symposium on Non-photorealistic animation and rendering, p.13-20, June 05-07, 2000, Annecy, France
- [4] Lee Markosian , Michael A. Kowalski , Daniel Goldstein , Samuel J. Trychin , John F. Hughes , Lubomir D. Bourdev, Real-time nonphotorealistic rendering, Proceedings of the 24th annual conference on Computer graphics and interactive techniques, p.415-420, August 1997
- [5] O. Deussen, S. Hiller, C. van Overveld, and T. Strothotte. Floating points: A method for computing stipple drawings. Computer Graphics Forum, 19(3), August 2000.
- [6] M. Levoy and T. Whitted. The use of points as a display primitive. Technical Report 85-022, University of North Carolina-Chapel Hill Computer Science Department, January 1985.
- [7] H. Pfister, M. Zwicker, J. van Baar, and M. Gross. Surfels: Surface elements as rendering primitives. In Proceedings of ACM SIGGRAPH 2000, Computer Graphics Proceedings, Annual Conference Series, pages 335~342, July 2000.
- [8] Bruce Gooch, Peter-Pike J. Sloam, Amy Gooch, Peter Shirley and Richard Risenfeld. Interactive Technical Illustration, IEEE Symposium on Interactive 3D Graphics, pages. 31~38 April 1999.