

# 모바일 기기에서의 비실사 렌더링 기법 적용을 위한

## 실루엣 에지 렌더링

전재웅<sup>0</sup> 장현호 임순범\* 최윤철

연세대학교 컴퓨터과학과

숙명여자대학교 멀티미디어과학과\*

{fyren, kencot, ycchoy}@rainbow.yonsei.ac.kr

sblim@sookmyung.ac.kr\*

### Silhouette Edges Rendering for Non Photorealistic Rendering at Mobile Devices

Jae-Woong Jeon<sup>0</sup> Hyun-Ho Jang Soon-Bum Lim\* Yoon-Chul Choy

Department of Computer Science, Yonsei University

Dept. of Multimedia Science, Sookmyung Women's University\*

#### 요 약

최근 3차원 그래픽 기법 중에서도 비실사(Non-Photorealistic Rendering) 기법은 3차원 그래픽의 변형을 통해 사용자와의 원활한 커뮤니케이션을 이끌어 낼 수 있다는 점에서 많은 주목을 받고 있다. 비실사 기법에서는 실루엣 에지를 찾아내고 디스플레이 하는 것이 중요한 과정이며 여러 가지 연구가 행해지고 있다. 지금까지 실루엣 에지 렌더링 연구는 데스크탑 플랫폼에서는 많이 이루어졌지만 모바일 기기에서의 구현을 위한 노력은 부족하였다. 본 논문은 모바일 기기 환경에서의 비실사 기법 적용을 위한 실루엣 에지 렌더링 기법을 제안한다. 모바일 기기 환경에서의 부족한 리소스와 폴리곤 제한을 극복하기 위하여 기존 연구보다 폴리곤 연산의 숫자를 감소시키는 기법을 제시하였다. 이 기법의 적용은 모바일 기기에서의 렌더링 프레임률을 높여주어 원활한 비실사 기법 적용에 도움을 준다.

#### 1. 서 론

기존의 전통적인 컴퓨터 그래픽은 포토리얼리즘에 기초한 연구들이 많았다. 그러나 사실적인 3차원 그래픽의 범위를 넘어서 사용자와의 커뮤니케이션을 중시하는 비실사 렌더링 기법이 최근 활발히 연구되고 있다. 비실사 렌더링에서는 간략한 선화를 통해서 정보의 전달을 효과적으로 이루어 내며 이러한 과정에서 실루엣 에지 렌더링을 사용하고 있다. 현재 실루엣 에지 렌더링 과정에 관한 연구는 주로 데스크탑 플랫폼을 기준으로 이루어져 있으며 모바일 기기에서의 연구는 부족한 상황이다. 모바일 환경은 데스크탑 플랫폼에 비하여 사용할 수 있는 계산 프로세서와 폴리곤의 숫자가 제한되어 있어서 기존의 실루엣 에지 렌더링 기법을 적용하는데 무리가 있다. 최근 모바일 환경에서의 3차원 그래픽 수요가 높아지는 가운데 본 논문은 모바일 기기에서의 적용을 위해 기존 연구보다 폴리곤 개수 및 폴리곤 계산량을 줄이는 실루엣 에지 렌더링 기법을 제시하고 있다. 본 논문에서는 전면(front facing) 폴리곤과 후면 폴리곤(back-facing)의 경계면을 찾아내어 렌더링 하는 과정 중 후면 폴리곤의 크기를 연산을 통해 일정 비율로 증가시키는 렌더링 기법을 통해서 실루엣 에지를 표현하는 방법을 사용한다. 또한 환경 및 요구하는 효과에 따라서 매개 변수에 변화를 주어 실루엣 에지의 두께 비율을 자유자재로 조절이 가능하여 원하는 효과를 창출할 수 있는 기법을 제안한다. 본 논문에서 제시된 두 가지 기법은 화면의 크기가 데스크탑 환경에 비하여 상대적으로 작고 처리에 사용할 수 있는 리소스가 부족한 모바일 기기에서 데스크탑 플랫폼과 시각적으로 동일한 효과를 보여주며

속도와 계산량면에서는 높은 향상률을 보여준다. 모바일 기기에서의 원활한 3차원 환경 구현을 위하여 본 논문에서의 기법은 OpenGL ES의 Common Lite Profile을 기반으로 제작되었으며 알고리즘의 최적화를 통해서 구현을 간단히 하고 빠른 처리로 실시간 렌더링을 가능하도록 하였다.

#### 2. 관련 연구

##### 2.1 실루엣 에지 렌더링

실루엣 에지를 렌더링 하는 방법에는 실루엣 에지의 가시성을 계산하는 은선 계산 알고리즘[1]을 사용하는 방법이 있다. [1]의 알고리즘은 전체 화면을 탐색하여 계산을 하지 않아도 실루엣 에지 렌더링을 가능하게 해주며 프레임률을 향상시킨다. 은선을 제거하는 기법을 통해서 실루엣 에지 렌더링을 구현한 연구도 있으며 이들은 주로 배치 프로세스를 통해 실루엣 에지 렌더링을 시행하고 실시간으로 렌더링을 구현하지는 못했다 [2][3]. 이후 정적인 다면체 모델에서 근접 정보를 통한 실시간 실루엣 에지 렌더링 방법도 제시되었다[4]. 근접 정보를 통한 실루엣 에지 검출 방식과는 다르게 깊이 버퍼와 두 개의 폴리곤 집합을 사용하여 주어진 시선기준에서 화면에 보이는 실루엣 에지를 실시간으로 계산하는 방법을 사용한 기법도 제시되었다[5][6]. 본 논문에서 제시된 실루엣 에지 렌더링 기법은 [6]의 방법을 기반으로 모바일 기기에서의 적용에 중점을 두어 기존 기법을 개선하고 더불어 새로운 기법을 제안하고 있다.

##### 2.2 OpenGL ES

OpenGL ES는 3차원 그래픽 API인 OpenGL의 임베디드 시스템

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

템 버전으로 모바일 기기에서 향상된 3차원 그래픽 성능을 제공하기 위해 사용되는 로우레벨단의 API이다. OpenGL ES는 OpenGL을 기반으로 하여 모바일 기기에 필요한 부분만을 남기고 불필요한 연산 부분을 제거한 서브셋으로 커먼(POA나 포켓피씨를 위한 프로파일) 프로파일과 커먼-라이트(핸드셋과 같은 최소형 모바일 기기를 위한 프로파일) 프로파일 렌더러 그리고 네이티브 플랫폼 윈도우 시스템과 연결해주는 EGL 등을 포함하고 있다[7].

### 3. 실루엣 에지 렌더링

#### 3.1 다면체 모델의 실루엣 에지 렌더링

실루엣 에지를 렌더링 하기 위하여 주어진 시선 기준에서 폴리곤을 두 종류로 구분한다. 우선 첫 번째 세트는 시선 기준에서 가까운 전면 폴리곤의 가시 부분으로 구성하며[그림 1-a] 두 번째 세트는 후면 폴리곤으로 구성하여 첫 번째 세트와 일치하는 부분들로 구성한다[그림 1-b]. 이 두 종류의 폴리곤 세트의 연산을 통해서 실루엣 에지를 렌더링 한다[그림 1-c]. 첫 번째 전면 폴리곤의 가시 영역 계산은 다양한 가시성 알고리즘을 사용할 수 있다. 전면 폴리곤의 가시 부분은 전체 영역이 모두 가시권에 들어오는 부분과 일부 은면으로 제거된 후 부분적으로 가시권에 들어오는 전면 폴리곤, 모두를 의미한다. 첫 번째 전면 폴리곤 세트의 가시 영역 계산에서와 동일하게 가시성 알고리즘의 결과로 후면 폴리곤으로 이루어진 두 번째 세트를 얻을 수 있다. 후면 폴리곤은 첫 번째 세트의 가시 표면의 후면으로 이루어진다[6].

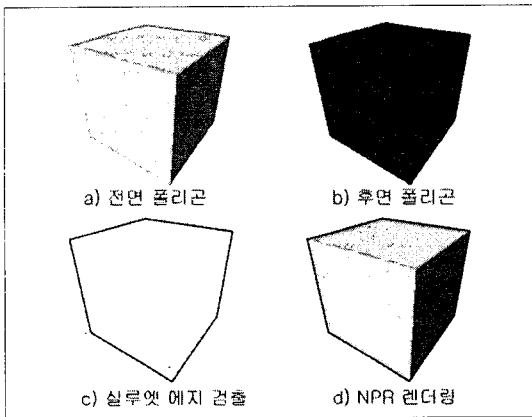


그림1. 실루엣 에지 렌더링

#### 3.2 후면 폴리곤의 확대

3.1에서 처리한 결과를 통해서 후면 폴리곤의 집합을 생성한 후 후면 폴리곤을 다음과 같은 과정을 통해서 확대한다.

우선 후면 폴리곤 삼각형의 중점(Vertex\_center\_of\_Face)을 구한다. 중점과 삼각형 각 정점의 차이를 계산한 후, 방향벡터 E에 비례값 S만큼을 곱하고, 매개변수 T만큼을 더하여 새로운 세 개의 정점의 집합을 생성한다. 확장 변수 S는 중점과 각 점의 거리를 계산하여 비례적으로 확장 거리를 구한다. 세 정점을 사용하여 확장된 새 후면 폴리곤을 생성한다. 새롭게 생성된 후면 폴리곤은 시선 기준상에서 가시영역의 전면 폴리곤의 외곽으로 렌더링됨으로써 모델의 실루엣 에지가 된다. 여기서 매개변수 T는 사용 상황에 따라 조절이 가능하며 실제 전면 폴리곤의 가시영역에 따라서 변화가 가능하고 외곽선 즉, 실루엣 에지의 굵기를 조절하여 다양한 효과를 줄 수 있다.

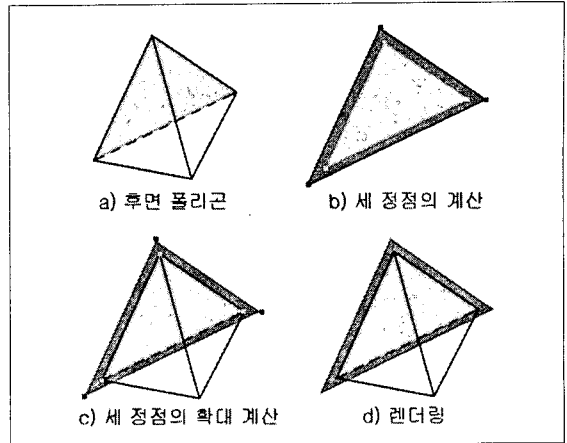


그림2. 후면 폴리곤 확대

```

/* Vertex i, j, k ; each vertex of the Triangle Face*/

/* Vertex_Max is respectively maximum x, y, z
value of the vertex i, j, k */
Vertex_Max := MAX(Vertex I, Vertex j, Vertex k);

/* Vertex_Min is respectively minimum x, y, z
value of the vertex i, j, k */
Vertex_Min := MIN(Vertex i, Vertex j, Vertex k);

Vertex_center_of_Face :=
( Vertex_Max - Vertex_Min ) + Vertex_Min ;
    
```

식1. 후면 폴리곤의 중점 계산

```

for i := 0 to Face n do
begin
if Face i is Back_Face
then
begin
for each Vertex j of Face i do
begin
E := Vertex j - Vertex_center_of_Face i ;
Vertex j := Vertex j + T + ( E * S );
end
end
end
end
    
```

식2. 후면 폴리곤의 확대 연산

후면 폴리곤의 확장을 통한 실루엣 에지의 렌더링 방법은 다음과 같은 면에서 기존의 연구보다 모바일 기기에서 유리하다. [6]의 연구에서 제시된 후면 폴리곤 'Fattening' 기법은 후면 폴리곤의 확장 이후에 한 개의 폴리곤이 (2\*정점 개수)개의 삼각형으로 늘어남으로써 계산량과 폴리곤의 수가 늘어나게 되고 기존 에지와 새로 생성된 에지 사이에 절단된 부분이 나타나게 된다. 그러나 본 논문에서 제시하는 기법은 후면 폴리곤의 면

의 개수를 늘리지 않으며 동시에 각 에지의 방향성을 계산하는 대신 3개의 정점의 연산을 통해서 확장된 정점을 계산하는 연산으로 계산량과 폴리곤 수를 제한하게 된다. 이는 모바일 기기의 표현 가능량과 폴리곤 처리 가능량이 데스크탑 플랫폼에 비해 현저히 낮기 때문에 계산량과 폴리곤 수를 증가시키지 않는 이 기법은 모바일 기기에 적합한 기법이라고 할 수 있다. 즉, 모바일 기기에서 폴리곤 연산의 숫자를 줄여 높은 프레임률을 획득함과 동시에 실제적으로 사용자가 느끼는 가시효율은 유지할 수 있게 된다. 또한 후면 폴리곤의 처리에 있어서 텍스처를 입혀 원하는 효과를 준다거나 또는 다양한 컬러 셰이딩 및 그라데이션 효과 등을 사용하여 실루엣 에지의 변화를 줄 수 있고 이런 변화를 이용하여 발전된 카툰 렌더링 효과, 드로잉 효과 등의 미술적인 효과를 추가적으로 얻을 수 있다는 장점이 있다.

### 3.3 실루엣 에지 렌더링 흐름도

본 논문에서 제시된 기법은 다음 흐름도[그림 3]를 따른다.

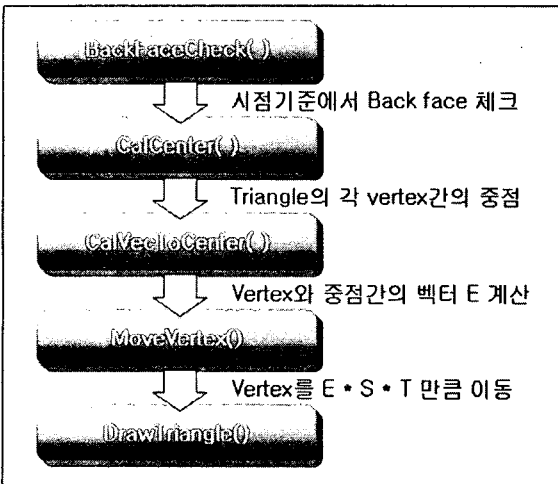


그림3. 흐름도

### 4. 구현 결과

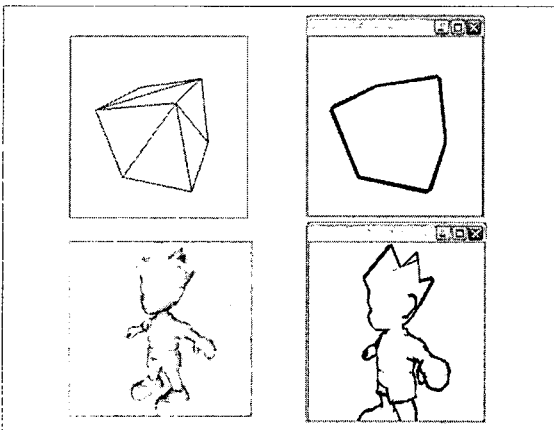


그림4. 구현결과물

본 논문에서 제시된 기법을 적용하여 제작한 프로그래밍 환경

은 Windows XP OS 플랫폼으로 Microsoft Visual C++ 컴파일러와 Hybrid Graphics 사에서 제공하는 Hybrid's OpenGL ES API Implementation[8]을 이용하여 제작하였다. 현재 본 논문에서 제시된 기법을 구현한 프로그램을 통한 결과물은 위의 [그림 4]와 같다. 결과물은 캐릭터와 기본 도형의 두 가지 모델에 제한한 실루엣 에지 렌더링 기법을 적용하여 생성하였다. 향후 결과물의 확장을 통해서 모바일 게임, 인터페이스, 시각 정보 표현 분야 등에 응용될 수 있을 것으로 기대된다.

### 5. 결론 및 향후 과제

본 논문에서는 모바일 기기에서의 비실사 기법 적용을 위한 실루엣 에지 렌더링 기법을 제시하였다. 첫 번째로 실루엣 에지 렌더링 기법은 시선 기준에 가까운 전면 폴리곤의 집합을 계산하고 가시성을 기준으로 후면 폴리곤의 집합을 계산한다. 두 번째로 실루엣 에지를 형성하기 위하여 전면 폴리곤의 가시영역보다 큰 후면 폴리곤을 만들어 낸다. 이 과정에서 폴리곤의 숫자를 유지하며 각 정점의 방향벡터에 따른 비례적 확대를 통해서 계산의 효율성과 동시에 기존과 동일한 시각적 효과를 가지게 된다. 이와 같은 방식은 모바일 기기의 제한된 디스플레이 영역과 폴리곤 계산량을 극복하고 높은 성능과 적용성을 보장하게 된다. 또한 매개 변수를 통해서 개발자와 구현 환경에 맞는 다양한 실루엣 에지를 생성하여 원하는 효과를 얻을 수 있게 된다.

향후 연구 과제로는 모바일 기기에서 좀 더 원활한 렌더링을 위하여 폴리곤 계산량을 줄일 수 있는 진보된 알고리즘의 개발이 요구된다. 현재의 기법으로는 모든 폴리곤에 1차적인 연산이 행해짐으로써 계산량이 크게 줄어들지는 않는데 깊이 버퍼와 객체의 중심으로부터의 거리 등의 계산을 통해서 실제적으로 실루엣 에지로 검출되지 않는 폴리곤을 선별하여 계산에서 제외함으로써 성능의 향상을 얻는 연구가 필요하다. 또한 후면 폴리곤의 단순 확대에서 벗어나 후면 폴리곤의 변형을 통해 실루엣 에지에 미술적 효과를 주는 기법도 연구의 대상이 될 수 있을 것이다.

### 참고문헌

- [1] A. Appel. The notion of quantitative invisibility and the machine rendering of solids. In Proceedings of ACM National Conference, pp.387-393, 1967.
- [2] I.Sutherland, R. Sproull, and R. Schumacker. A characterization of ten hidden-surface algorithms. Computing Surveys 6(1):1-55, March 1974.
- [3] G. Elber and E. Cohen. Hidden curve removal for free form surfaces. In Proceedings of SIGGRAPH '90, pp.95-104, 1990.
- [4] Lee Markosian, Michael A. Kowalski, Samuel J. Trychin, Lubomir D. Bourdev, Daniel Goldstein, and John F. Hughes. Real-Time Nonphotorealistic Rendering. Computer Graphics (Proceedings of SIGGRAPH '97), 1997.
- [5] Jareck Rossignac, Maarten van Emmerik. Hidden Contours on a Framebuffer. Proceedings of the 7th Workshop on Computer Graphics Hardware, Eurographics, 1992.
- [6] Ramesh Raskar, Michael Cohen. Image Precision Silhouette Edges. Symposium on Interactive 3D Graphics, 1999.
- [7] <http://www.opengles.org>
- [8] <http://www.hybrid.fi>