

게임을 위한 코믹북 스타일 렌더링

김태규⁰, 오규환^{*}, 이창신^{**}

아주대학교 미디어학과 인터랙티브 엔터테인먼트랩^{0*}, 아주대학교 미디어학부^{**}
famous-kim@hanmail.net⁰, drghoh@ajou.ac.kr^{*}, seohwa@gmail.com^{**}

Comic-Book Style Rendering for Game

Tae-Gyu Kim⁰, Gyuhan Oh^{*}, Chang-Shin Lee^{**}

Interactive Entertainment Lab, Division of Media, Ajou University^{0*}
Division of Media, Ajou University^{**}

요 약

최근 들어 NPR(Non-Photorealistic Rendering) 기법을 이용한 게임들이 많아지고 있다. 그렇지만 게임에서 적용하고 있는 NPR 기법들은 몇 가지 종류로 한정되며, 특히 카툰렌더링에 집중되고 있다. 본 논문에서는 그간 연구가 거의 이루어지지 않았던 게임에 적용 가능한 코믹북 스타일의 NPR 기법을 제안한다. 이를 위해 먼저 코믹북의 비주얼과 셀 애니메이션 비주얼의 차이점을 비교, 분석하여 코믹북 스타일의 비주얼에 대한 표현 요소를 도출한다. 다음으로, 도출된 세 가지 표현 요소, 외곽선 스케치, 톤, 해칭을 효과적으로 구현하기 위한 렌더링 알고리즘들을 제시한다. 마지막으로 이 알고리즘이 적용된 게임을 제작함으로써 게임을 위한 알고리즘으로 적용이 가능한지를 검증한다.

ABSTRACT

Nowadays, many computer games based on NPR(Non-photorealistic Rendering) techniques have been developed due to their distinctive visual properties. However, only limited methods of NPR techniques have been exploited in producing computer games and amongst them cartoon-style rendering techniques especially has had the special interest. In the paper, we suggest an effective rendering method of comic-book style that will be applicable to computer game. In order to do, we first characterize the properties of comic-book from comparing two visuals: celluloid animation and comic-book. We then suggest a real-time rendering method of comic-book style represented by outline sketch, tone, and hatching. We finally examine its effectiveness by observing the game developed using the method.

Keyword : NPR, Real Time Hatching, Silhouette, Comic-book style

1. 서론

NPR(Non-Photorealistic Rendering)은 1990년대 초부터 컴퓨터 그래픽스 분야에서 주목받기 시작한 기술로서, 대상을 사실적으로 표현하기보다 대상의 특징을 포착해 의미와 느낌을 전달하는 것을 목표로 삼고 있다.

NPR 기법에는 여러 가지가 있으며 분류 방식도 다양한데, 결과물의 비주얼적인 표현 방식에 따라 선, 펜&잉크, 카툰, 회화, 수묵화, 수채화, 이미지 모자이크, 일러스트레이션 등으로 나눌 수 있다[1].

표현 방식에 의한 분류에서 확인할 수 있듯이 NPR 기법은 다른 예술, 미디어 장르에서 보여주었던 비주얼을 구현할 수 있기 때문에 NPR을 사용한 게임들은 기존의 사실적인 그래픽으로 제작된 게임들과 비교했을 때 차별성과 개성을 가질 수 있으며, 그로인해 최근 NPR 게임의 수가 증가하고 있는 추세이다. 셀 애니메이션(Celuloid Animation)의 느낌을 보여주는 「마비노기」[2]나 한 폭의 수묵화 같은 「오오카미(OKAMI)」[3], 손으로 그린 수채화를 보는 듯한 「전장의 발큐리아(Battle Field Valkyrie)」[4]등 다양한 플랫폼, 장르에서 여러 게임들이 NPR 기법을 적용하고 있다.

그러나 다양한 NPR 기법들이 있고 NPR 게임들이 증가하고 있는 추세이긴 하지만, 실제 게임들에 적용되는 NPR 기법은 몇 가지로 한정되어 있다. 현재 게임에서 가장 많이 사용되고 있는 NPR 기법은 셀 애니메이션의 비주얼을 보여주는 카툰렌더링(Cartoon Rendering)이다. 수묵화, 수채화의 느낌을 표현하는 NPR 게임들이 있지만 그 수가 적은 편이다.

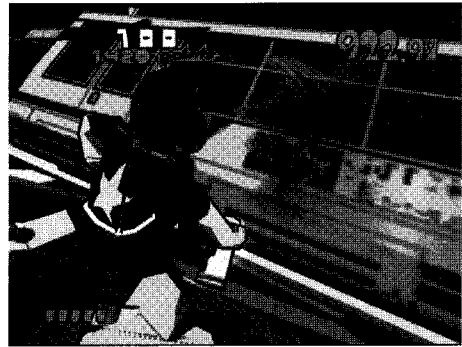
본 연구에서는 코믹북 스타일의 비주얼을 구현하기 위한 효과적인 알고리즘을 제시한다. 이를 위해 셀 애니메이션과 코믹북에서 비주얼의 표현방식을 비교하여 코믹북 비주얼이 가지는 특징을 도출한다. 도출된 세 가지 특징인 외곽선, 톤, 해칭을 게임에 적용하기 위해 이를 효과적으로 표현하기 위한 렌더링 알고리즘들을 제시한다. 또한, 본 연구의 렌더링 기법을 적용한 게임을 제작하여 알고리즘의 성능이 게임에 적합한지의 여부와 코믹북 스타일의 비주얼을 효과적으로 표현하는지를 검증하도록 한다. 2장에서는 기존의 게임들에 적용된 NPR 기법에 대해 살펴보고, 3장에서 본 논문이 제안하는 알고리즘에 대한 설명이 이루어진다. 4장에서는 연구 결과를 적용하여 개발한 게임과 결과를 제시한다.

2.NPR 기법에 따른 게임 분류

2.1 게임에 적용된 NPR 기법들

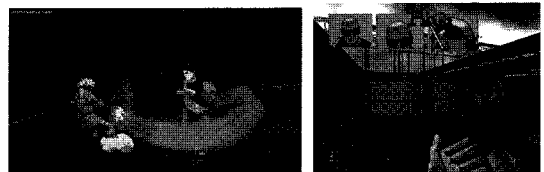
게임에 적용된 NPR 게임들을 분류해보면 카툰렌더링, 수묵화 렌더링, 수채화 렌더링의 세 종류로 나눌 수 있다.

2.1.1 카툰렌더링



[그림 1] 「젯 셋 라디오」플레이 스크린 샷

카툰렌더링은 셀 애니메이션의 비주얼을 게임 화면에 구현하는 기술로서, 「젯 셋 라디오(dera Jet Set Radio)」[5]를 통해 상용 게임에 최초로 도입되었다. 2000년 출시된 「젯 셋 라디오」는 그래피티(Graffiti)라는 게임의 소재와 어울리는 비주얼을 위해 카툰렌더링을 사용하였다(그림 1 참조).



(a)

(b)

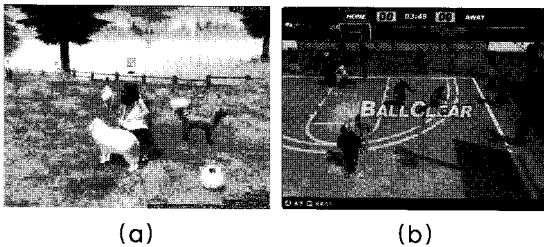
[그림 2] (a)「나루토-rise of ninja」, (b)「써틴」의 플레이 스크린 샷

그리고 카툰렌더링은 「드래곤 볼(Dragon Ball)」시리즈[6], 「강철의 연금술사(Full Metal Alchemist)」[7], 「나루토(NARUTO)」시리즈[8], 「써틴(XIII)」[9] 등 만화, 특히 애니메이션을 바탕으로 하는 게임들에서 많이 사용되고 있다. [그림 2]에서 일본 만화를 원작으로 하는 「나루토-rise of ninja」와 프랑스 만화를 원작으로 하는 「써틴」에서 카툰렌더링이 적용된 모습을 확인할 수 있다.



[그림 3] 「젤다의 전설-바람의 택트」의 플레이 스크린 샷

카툰렌더링의 사용이 많아지면서 카툰렌더링 내에서 다양한 시도들이 이루어졌는데, 외곽선을 살렸던 기존의 카툰렌더링과는 다르게 외곽선을 제거한 형태로 구현이 된 「젤다의 전설-바람의 택트(The Legend of Zelda-The Wind Waker)」[10]나 외곽선을 극단적으로 강조하여 서양만화의 느낌을 풍기는 「뷰티풀 조(Viewtiful Joe)」시리즈[11], 카툰렌더링이면서 텍스처는 실사화한 「크랙다운(Crackdown)」[12] 등을 예로 들 수 있다. [그림 3]은 「젤다의 전설-바람의 택트」의 플레이 스크린 샷을 보여준다.



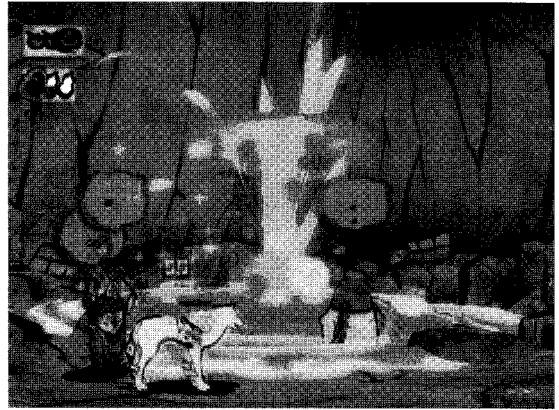
(a)

(b)

[그림 4] (a) 「마비노기」와 (b) 「프리스타일」의 플레이 스크린 샷

한국에서는 「천랑열전」[13] 등 패키지 게임에서 카툰렌더링이 적용된 경우가 있고, 온라인 게임에서 카툰렌더링의 사용이 활발하게 이루어지고 있다. 2003년 “그리곤 엔터테인먼트”에서 서비스를 시작한 「썰 온라인」[14]은 국내 최초로 카툰렌더링을 적용한 온라인 게임으로 알려져 있으며, 카툰렌더링을 적용하고 상업적으로 성공한 온라인게임으로는 「마비노기」와 「프리스타일」[15]을 예로 들 수 있다([그림 4] 참조). 이 후에도 「풍류공작소」[16], 「스키드 러쉬」[17], 「페이퍼 맨」[18], 「마구마구」[19] 등 다양한 게임에서 카툰렌더링 기법을 사용하고 있다.

2.1.2 수묵화 렌더링



[그림 5] 「오오카미」플레이 스크린 샷

수묵화 렌더링은 동양화에서 나타나는 농, 담, 묵이라는 독특한 표현법을 재현하기 위해 등장한 기술로서, 수묵화 렌더링이 본격적으로 게임에 적용된 예로는 “클로버 스튜디오(Clover Studio)”에서 개발하고 “캡콤(CAPCOM)”에서 배급한 「오오카미」를 들 수 있다. [그림 5]는 「오오카미」의 플레이 스크린 샷을 보여준다. 일본의 전통 설화와 신화를 바탕으로 한 이 게임은 게임 전체를 수묵화 형식으로 표현하여 신화적인 게임의 분위기를 강조한다. 그리고 수묵화 그래픽 분위기의 연장선으로서 붓놀림을 이용한 게임 플레이를 보여준다.



[그림 6] 「파천일검2」의 플레이 스크린 샷

한국에서도 (주)매직스에서 개발한 무협 MMORPG 「파천 일검2」[20]가 수묵화 렌더링을 사용하고 있다(그림 6 참조). 동일한 수묵화 렌더링을 사용하였지만 「파천일검2」에서는 「오오카미」보다 좀 더 ‘수묵담채화’에 가까운 결과물을 보여주고 있다.

2.1.3 수채화 렌더링

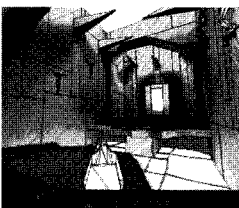
수채화 렌더링은 화가들에 의해 그려진 일러스트와 같이 물감으로 그려진 비주얼을 표현해내는 기술이다. 다른 NPR 기법과 달리 수채화 렌더링은 최근 들어 게임에 적용되고 있는데, 2007년 10월 PS2(PlayStation2)용으로 발매되었던 “세가(SEGA)의 「샤이닝 포스 EXA(Shinning Force EXA)」[21], PS3용으로 개발 중인 “레드컴퍼니(Red Company)의 「전장의 발큐리아」(그림 7 참조)가 있다.



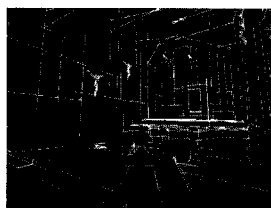
〔그림 7〕「전장의 발큐리아」플레이 스크린 샷

이와 같이 다수의 게임들이 NPR 기법을 사용하고 있으며 그 수가 점차 늘어나고 있지만, 실제 게임에 적용되고 있는 NPR 기법들의 종류는 한정적이며 대부분 카툰렌더링에 치우쳐져 있다. 그러나 NPR 기법을 이용하면 수채화 렌더링과 같이 다양한 분야의 비주얼들을 게임에 접목시킬 수 있는 가능성이 충분하기 때문에, 본 연구에서는 기존에는 없었던 코믹북 스타일의 새로운 렌더링 기법을 제시하고자 한다.

2.2 연구 동기



(a)



(b)



(c)

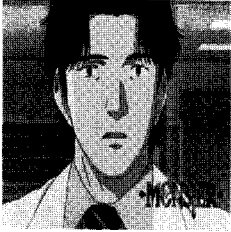
〔그림 8〕“NPR Quake”의 결과 화면. (a)스케치 렌더링, (b)블루프린트 렌더링, (c)붓터치 렌더링

코믹북 스타일 비주얼을 게임 화면에서 담기 위해서는 손으로 그린 것과 같은 외곽선의 표현법과 선의 반복으로 명암을 나타낼 수 있는 방법을 찾아야 한다. 기존의 유사한 연구로는 2000년 발표되었던 “NPR Quake”[22]로서 이는 기존의 Quake 게임에 다양한 NPR 기법을 적용하여 비주얼을 변형시켰던 프로젝트였다. 여기서 구현되었던 NPR 기법은 스케치 렌더링, 블루프린트 렌더링, 붓터치 렌더링의 세 종류이다(그림 8 참조).

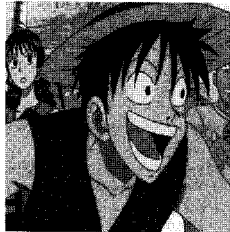
세 가지 기법 중 스케치 렌더링이 손으로 그려진 코믹북의 스타일을 유사하게 표현하고 있으며, 본 논문에서 제시하는 렌더링 기법에 근접해 있다(그림 8(a) 참조). 그렇지만 스케치 렌더링 기법을 코믹북 스타일의 비주얼이라고 하기에는 부족한 점이 있다. 빛을 받지 못하는 부분에 해칭(Hatching)이 들어가 있기는 하지만, 해칭이 명암을 표현하는 것이 아니라 단순히 어두운 부분을 맵핑하는 텍스처로만 사용되었다. 코믹북 스타일의 비주얼을 구현하기 위해서는 어두운 정도에 따라 해칭이 다르게 그려져야 하지만, “NPR Quake”의 스케치 렌더링에서는 이런 부분을 효과적으로 구현하지 못하였다. 본 연구에서는 “NPR Quake”에서 구현하지 못하였던 부분들을 보완하고 코믹북 스타일 같은 그래픽을 실시간으로 보여줄 수 있는 효율적인 렌더링 기법을 제안한다.

3. 연구 내용

3.1 접근 방법



(a)



(b)

[그림 9] 셀 애니메이션의 한 장면 (a)「몬스터」 (b)「원피스」



(a)



(b)

[그림 10] 출판 코믹북의 한 장면 (a)「몬스터」 (b)「원피스」

[그림 9]와 [그림 10]의 (a), (b)는 각각 「몬스터(Monster)」, 「원피스(One-piece)」이라는 작품들이다. [그림 9]는 셀 애니메이션으로 제작된 각 작품들의 스크린 샷이고, [그림 10]은 코믹북으로 출판된 동일한 작품들의 한 장면이다. 셀 애니메이션과 코믹북은 모두 2D의 그림으로 제작되는 콘텐츠(Contents)이지만, [그림 9]와 [그림 10]에서 보듯이 동일한 인물이 나오는 같은 작품이더라도 비주얼의 느낌은 다소 차이가 있다는 것을 알 수 있다.

셀 애니메이션인 [그림 9]에서의 인물들은 선명하고 단정하며, 경계가 분절되는 듯한 느낌을, 코믹북인 [그림 10]에서의 인물들은 거칠고 복잡하지만, 역동적이면서 생동감이 넘치는 느낌을 준다. 이처럼 셀 애니메이션과 코믹북이 서로 다른 느낌을 주는 이유는 각 미디어에서 그림을 그리는 표현법의 차이에서 찾을 수 있다.

표현법	특징	역할
외곽선	일정한 굵기의 가 는 선 이용	선명하고 단정한 느낌 표현
톤	명암 단계의 변화 가 급격한	분절된 느낌 전달

[표 1] 셀 애니메이션의 표현 기법

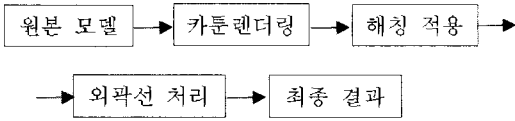
표현법	특징	역할
외곽선	두꺼우면서 굵기가 불규칙한 선 이용	거칠고 복잡한 느 낌 표현
톤	빛의 경계가 뚜렷하 지 않음	자연스러운 느낌 전달
해칭	평행선들이 여러 번 교차되어 그려짐	거친 느낌의 강조 및 역동감과 생동 성을 불어넣음

[표 2] 코믹북에서의 표현 기법

셀 애니메이션에서는 형태를 나타내기 위한 외곽선으로 가늘면서 굵기가 일정한 선을 이용하고, 내부의 음영을 표현할 때에는 톤(Tone)만을 사용한다(톤이란 빛을 약하게 받는 부분을 명암 단계가 낮은 색상으로 칠하여 음영을 표현하는 묘사법을 의미한다). 일정하게 가는 굵기로 표현된 외곽선은 인물과 배경들이 선명하고 단정하게 보이는 역할을 하며, 급격하게 명암 단계가 달라지는 톤의 묘사법은 빛의 경계를 뚜렷하게 나타내어 분절된 느낌을 준다([표 1] 참조).

반면, 코믹북의 경우는 셀 애니메이션과 동일하게 외곽선과 톤을 사용하지만, 외곽선의 경우 두껍고 굵기가 불규칙한 선을 사용하며, 음영의 경우에도 빛의 경계가 뚜렷하게 보이지 않는 톤을 이용한다. 이와 같은 코믹북의 특징은 거칠고 복잡하면서도 특유의 자연스러운 느낌을 만들어낸다. 특히 코믹북의 경우 셀 애니메이션에서는 찾아보기 힘든 해칭 묘사법을 이용하는데, 이것이 코믹북에서 셀 애니메이션과 다른 느낌을 만들어내는 핵심적인 역할을 한다. 해칭은 평행한 선을 여러 번 교차되도록 겹쳐 그려 어두운 부분이나 움직임의 방향 등을 표현하는 묘사법을 의미하는 것으로, 코믹북의 거친 느낌을 강조하면서 역동적이고 생동적인 느낌을 만들어낸다([표 2] 참조).

기존의 NPR 게임들은 2.1. 절에서 보는 것처럼 대부분이 [그림 9]의 셀 애니메이션과 같은 느낌을 표현하고 있다. 수록화나 수채화와 같은 묘사법을 사용한 게임도 있었지만, [그림 10]과 같은 코믹북의 느낌을 표현한 게임을 찾아보기는 힘들다. [그림 8]의 「NPR Quake」에서 이와 같은 시도는 있었으나 해칭으로 사용된 선의 방향과 겹쳐짐이 단조롭다는 단점을 가지고 있었다. 이에 본 논문은 코믹북의 특징적인 표현법인 굵기가 불규칙한 외곽선, 해칭, 톤을 이용하여 [그림 9]와 같은 느낌을 낼 수 있는 게임의 렌더링 기법을 제시한다.



[그림 11] 렌더링 프로세스 다이어그램

본 연구에서 제안하는 렌더링 과정은 3단계를 거쳐 이루어지며, 톤, 해칭, 외곽선의 순서로 표현법이 구현된다. [그림 11]에서 렌더링 프로세스의 다이어그램이 보여진다. 톤 묘사는 기존의 카툰 렌더링(Cartoon rendering), 셀 셰이딩(Celluloid shading)이라는 이름을 갖는 NPR 기법에 의해 구현된다. 일반적으로 카툰 렌더링 알고리즘은 음영 텍스처와 스텝 함수를 이용하는데, 본 연구에서는 이들 중 스텝 함수를 적용시켜 톤을 표현한다. 다음으로 코믹북과 같은 비주얼을 생성하는데 중요한 역할을 하는 해칭 기법의 경우에는 Praun[25]이 제시한 실시간 해칭(Real-time Hatching)이라는 NPR 알고리즘을 수정하여 구현한다. 본 연구에서는 Praun의 알고리즘을 게임에서 사용할 수 있도록 블렌딩 과정을 단순화시키고 투박한 느낌을 제거하였다. 외곽선의 구현에는 일반적으로 오브젝트 공간을 이용하는 알고리즘과 이미지 공간을 이용하는 알고리즘 두 가지가 사용된다. 오브젝트 공간을 이용하는 알고리즘들은 기존의 NPR 게임들에서 사용되었던 방식으로, 두께가 일정한 외곽선을 그리게 된다. 하지만 이와 같은 외곽선은 깔끔하면서 단정한 느낌을 주기 때문에 코믹북의 표현에는 어울리지 않는다. 따라서 본 알고리즘에서는 이미지 공간상에서 소벨 필터(Sobel filter)를 이용하여 연필이나 펜으로 그린 듯한 거칠고 두께가 일정하지 않은 외곽선을 생성해낸다.

이번 연구에서는 알고리즘의 품질과 성능을 입증하기 위하여 코믹북을 컨셉으로 하는 게임을 제작하였다. 게임은 코믹북 속으로 빨려 들어간 주인공이 겪는 모험을 그리고 있으며, 본 논문의 알고리즘을 이용하여 코믹북과 같은 비주얼을 만들어낸다. 게임은 Direct3D를 그래픽 라이브러리로 사용하며, 렌더링 알고리즘의 모든 과정은 버텍스 셰이더와 픽셀 셰이더를 이용하여 GPU의 프로그래머블 렌더링 파이프라인(Programmable Rendering Pipeline)에서 실행된다. 버텍스, 픽셀 셰이더에 사용되는 코드들은 HLSL(High-Level Shader Language) 문법에 맞게 작성되었으며, 셰이더 모델은 3.0을 이용한다.

3.2 코믹북과 같은 그래픽 구현을 위한 실시간 NPR 기법의 제안

3.2.1 카툰 렌더링의 적용

카툰 렌더링은 셀 애니메이션과 같은 형태로 색상의 명암을 표현하기 때문에 셀 셰이딩(Cel-shading)이라는 이름으로 불리기도 하며, 만화적인 느낌을 표현하기 위한 대표적인 NPR 기법이다. 카툰 렌더링은 단순화된 리플렉션 모델(Reflection model)에 의해 몇 단계의 불연속적인 셰이딩 영역을 도출하게 되고, 각 영역에 명암이 다른 색상을 나타냄으로써 구현한다[23]. 일반적으로 게임에서는 음영 정보를 담은 텍스처 맵을 이용하는 방식과 픽셀 단위로 스텝 함수(Step function)를 적용시켜 명암 단계를 결정짓는 2가지 방법이 널리 사용되고 있다[23, 24].

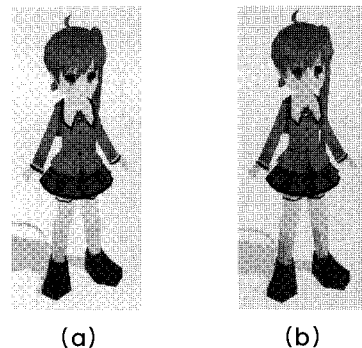
```

color StepFunction() {
    vector Ldir; //픽셀에서 광원으로 향하는 방향 벡터
    vector Normal; //픽셀의 노멀 벡터

    if(angle(Ldir, Normal) > a_threshold)
        return texture.rgb; //밝은 영역
    else
        return texture.rgb * s_threshold; //어두운 영역
}
  
```

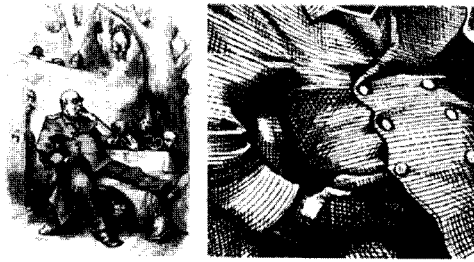
[그림 12] 카툰 셰이딩을 표현하는 스텝 함수

본 알고리즘에서는 픽셀 셰이더를 이용하여 픽셀 단위로 광원의 세기를 계산하며, [그림 12]의 스텝 함수를 이용하여 2단계의 셰이딩 영역을 도출한다. 그 다음 어두운 부분은 원래 텍스처에 $s_threshold$ 를 곱하여 명암 단계를 낮춘 색상을 맵핑한다. 셰이딩 영역을 구별할 때는 $a_threshold$ 의 값으로 0.3을 사용하였으며, 명암을 낮추기 위한 $s_threshold$ 값은 0.92를 사용하였다. [그림 13]에서 카툰 렌더링이 적용된 결과를 확인할 수 있다.



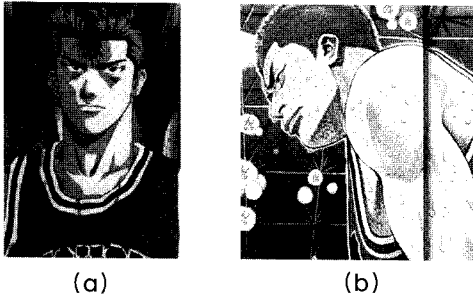
[그림 13] (a) 일반 텍스처만 적용된 상태 (b) 카툰 셰이딩이 적용된 상태

3.1.2 실시간 해칭 구현



[그림 14] 해칭 스타일로 그려진 만화. Thomas Nest 作

해칭은 [그림 14]와 같이 이미지의 음영을 평행한 선으로 교차시켜서 그리는 표현법을 의미한다.

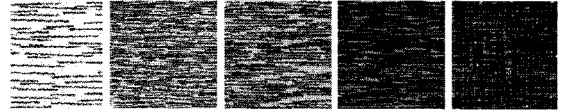


[그림 15] (a) 셀 애니메이션에서의 캐릭터 (b) 코믹북에서의 캐릭터

해칭 스타일은 손으로 그려진 만화의 느낌을 표현하는데 핵심적인 역할을 하는데, [그림 15]와 같이 해칭의 유무에 따라 동일한 캐릭터가 서로 다른 느낌을 가지게 되며, [그림 15(b)]가 펜으로 그려진 코믹북의 한 장면이라는 것을 직관적으로 알 수 있게 해준다.

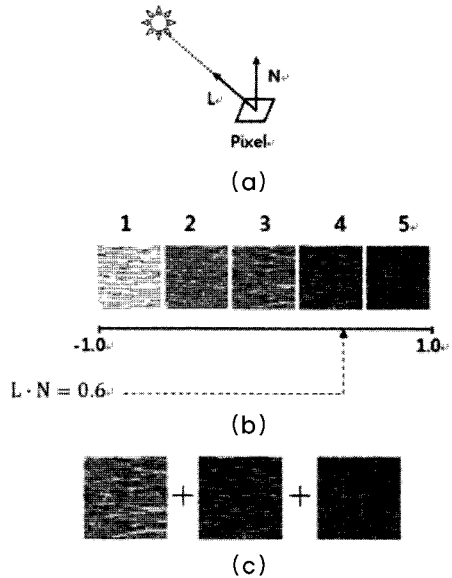
컴퓨터 그래픽스 분야에서는 3D 오브젝트를 해칭 스타일로 렌더링하는 방법에 대한 연구가 다양하게 진행되어 왔는데, 그 중 Hertzman, Zorin과 Praun에 의한 알고리즘이 주목할 만하다.

Hertzman과 Zorin[26]은 오브젝트의 내부를 C2 연속성의 방향을 갖는 영역-디렉션 필드(Direction Field)로 구분하고, 각 영역의 방향에 맞춰 한 장의 해칭 텍스처를 반복적으로 맵핑하는 방법으로 스타일을 구현한다. 이들 알고리즘은 명암의 표현과 선의 방향 등에서 높은 품질의 결과를 낼 수 있지만, 계산 속도가 느려[26] 실시간 렌더링이 필요한 게임에는 적용하기 힘들다.



[그림 16] 밝기 단계별로 미리 제작한 해칭 텍스처들

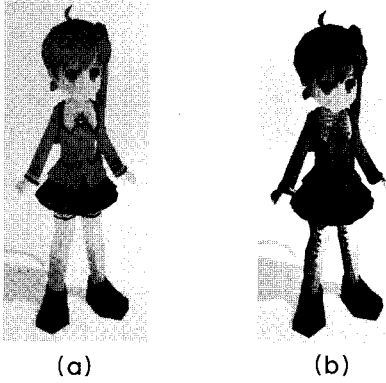
반면, Praun[25]이 제시한 해칭 알고리즘은 실시간으로 처리할 수 있다는 장점을 가지고 있기 때문에, 본 기법에서는 [25]의 알고리즘을 응용하여 해칭을 구현한다. 실시간 해칭에서는 [그림 16]과 같이 밝기 단계별로 미리 준비되어 있는 여러 개의 텍스처를 이용하는데, 이를 Tonal Art Map(TAM)이라고 한다. TAM에서의 가로 축은 명암 단계를 의미하며, 오른쪽 텍스처일 수록 어두운 부분에 많이 사용된다.



[그림 17] 해칭 렌더링 과정. (a) 픽셀 단위로 노멀 벡터(N)와 광원로의 벡터(L)를 구함. (b) N과 L의 내적 값을 이용하여 해당 픽셀이 TAM의 어느 단계에 포함되는지 확인. (c) 인접에 있는 TAM들의 블렌딩 결과 값을 픽셀에 맵핑.

실시간 해칭은 픽셀 셰이더로 구현하는데, 우선 각 픽셀별로 광원의 방향(L)과 픽셀의 노멀(N)을 곱하여 해당 픽셀이 TAM에서 몇 번째 명암 단계에 해당하는지 확인한다. 그리고 해당 단계에 있는 해칭 텍스처와 좌우 인접해 있는 텍스처들을 블렌딩한 후 그 결과 값을 픽셀에 맵핑한다. 블렌딩 과정에서는 값이 -1.0에서 1.0 사이에서 위치해 있는 비율을 가중치로 사용한다(그림 17 참조). 실시간 해칭 알고리즘에서는 TAM을 어떤 형태로 제작하느냐에 따라 다양한 효과를 나타낼 수 있다[27]. 여기에서는 연필의 거친 느낌을

표현하기 위해 [그림 16]과 같이 굵기가 불규칙하면서 거친 느낌의 선들을 이용하여 TAM을 제작하였다.



[그림 18] (a)카툰렌더링의 결과 이미지 (b)실시간 예정이 적용된 결과 이미지

그리고 [25]의 알고리즘에서는 TAM의 모든 텍스처들을 블렌딩하여 하나의 픽셀에 맵핑하는데 반해, 본 알고리즘에서는 이를 수정하여 3개의 텍스처만을 블렌딩하는 방식을 사용한다. 게임에서는 결과 품질을 약간 낮추는 대신 처리 속도를 높이는 것이 중요하기 때문이다. [그림 18]에서 해칭이 적용된 결과 이미지를 확인할 수 있다.

3.1.3 외곽선 그리기

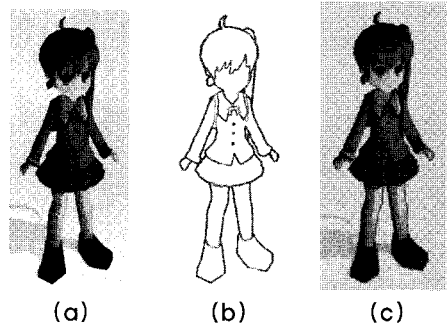
외곽선은 NPR에서 중요한 역할을 한다. 해칭이 오브젝트의 내부를 만화적인 느낌으로 표현하는 기법이라면, 외곽선은 오브젝트의 형태를 펜 선으로 그린 느낌을 강조하는 역할을 한다.

3D 오브젝트의 외곽선을 검출하는 일은 수행시간과 시각적 효과를 모두 고려해야 하는 까다로운 문제이며, 오브젝트 공간(Object space)과 이미지 공간(Image space)에서 처리하는 두 가지 방법이 있다[28].



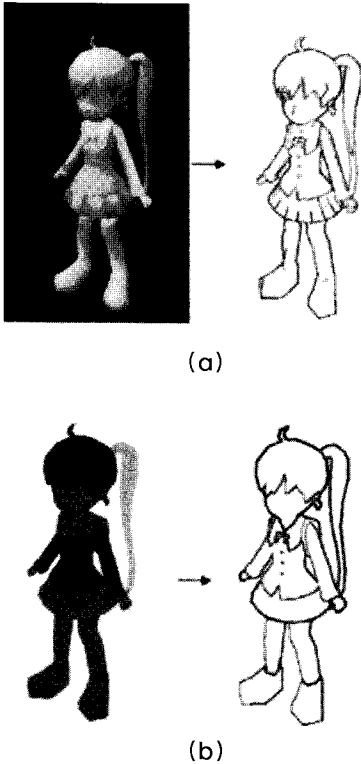
[그림 19] 선의 굵기가 일정하여 깔끔한 느낌을 내는 외곽선

오브젝트 공간에서의 검출 알고리즘은 오브젝트를 구성하는 폴리곤들을 대상으로 하며, 카메라에 보이는 면과 보이지 않는 면 사이에 끼어있는 경계선을 외곽선으로 처리한다. 즉, 카메라의 시선 방향 벡터(L)와 면의 노멀 벡터(N)가 이루는 각의 크기, 의 부호가 +에서 -로 바뀌는 구간에서 외곽선이 발생한다는 점을 이용하여 알고리즘을 구현한다. 오브젝트 공간에서 외곽선을 검출하는 알고리즘은 실행 속도가 빠르다는 장점이 있지만, 선 자체가 동일한 굵기를 가지고 깔끔한 느낌이 들기 때문에 코믹북 같은 거친 느낌에는 어울리지 않는다([그림19] 참조).



[그림 20] 외곽선이 추가된 이미지. (a)예정이 적용된 이미지. (b)예정 텍스처가 적용된 외곽선 결과. (c)외곽선이 합쳐진 결과물

그에 반해 이미지 공간에서 실행되는 알고리즘은 렌더링된 이미지를 분석하여 외곽선을 찾아낸다. 이미지 상에서의 색상 변화 값을 이용하여 변화 폭이 큰 곳을 외곽선 부분으로 판별하는 것으로, 로버트 크로스(Roberts' Cross), 소벨 연산(Sobel operator)[29] 등의 알고리즘들이 있다. 이미지 공간에서 검출된 외곽선은 부정확하면서 거칠게 그려지는 특징이 있는데, 코믹북 느낌에는 오히려 그런 불규칙성이 잘 어울린다. 이는 셀 애니메이션들과 달리 코믹북의 스케치들은 손으로 그린 거친 느낌을 갖고 있기 때문이다. 따라서 여기에서는 렌더링된 이미지를 텍스처에 저장한 다음, 소벨 연산으로 외곽선을 검출하고 이를 다시 원본 이미지와 합치는 과정을 통해 결과물을 만들어낸다. [그림 20]에서와 같이 (a)와 (b)가 합쳐져 (c)의 결과 이미지가 생성된다.

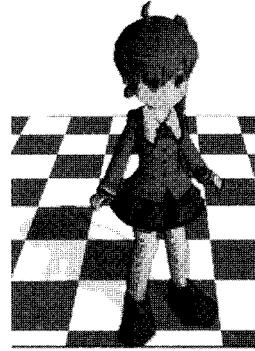


[그림 21] (a) 노멀 맵을 이용하여 추출한 외곽선 (b) 색상 맵을 텍스처로 제작하여 렌더링

그런데 소벨 연산은 분석의 대상인 소스(Source) 이미지로 어떤 것을 사용하느냐에 따라 결과가 많이 달라진다. 오브젝트의 노멀 맵을 추출하여 소벨 연산의 소스 이미지로 사용하였을 경우, [그림 21(a)]에서 보듯이 의도하지 않은 부분에 외곽선이 생길 뿐만 아니라 과도하게 지저분해 보이는 단점을 가지고 있다. 따라서 원하는 부분에 외곽선이 형성되도록 하기 위해서는 외곽선이 나타나기를 기대하는 부분을 경계로 색상이 급격하게 변하는 색상 맵(Color map)을 따로 제작해야 한다([그림 21(b)] 참조). 최종적으로는 검출된 외곽선에 해칭 텍스처를 맵핑하여 연필 선의 느낌을 강조하였다 ([그림 20(b)] 참조).

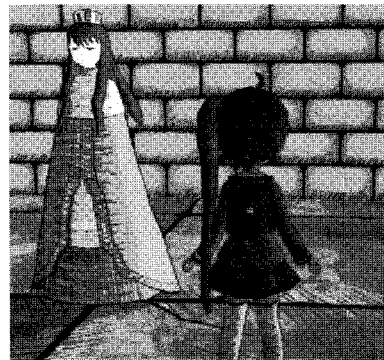
4. 결론

4.1 구현 결과



[그림 22] 최종 결과물

[그림 22]은 3장에서 기술한 알고리즘이 적용된 결과 이미지를 보여준다. 손으로 스케치를 한 느낌을 주는 외곽선, 광원에 의한 명암의 변화가 반영된 톤, 어두운 정도에 따라 다른 빈도로 그려진 해칭들을 통해 코믹북 스타일의 비주얼 보여주고 있다.

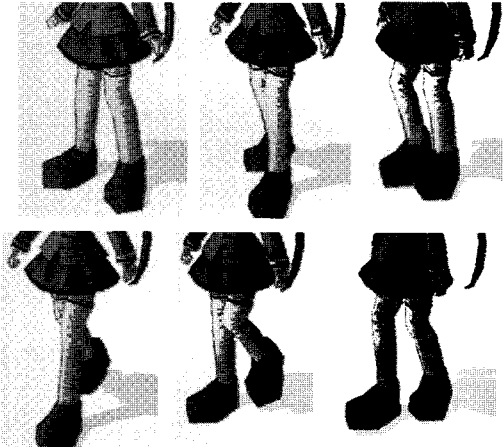


[그림 23] 칼라 캐릭터와 흑백 캐릭터 모두 적용된 결과

그리고 단순히 렌더링으로 그치는 것이 아니라 게임 환경에서 실시간으로 사용할 수 있음을 보여주는 것이 [그림 23]이다. 이는 본 연구의 알고리즘을 적용하여 제작한 게임의 스크린샷이다.

[그림 23]에서 볼 수 있듯이 칼라 캐릭터, 흑백 캐릭터 모두에서 코믹북 스타일의 비주얼이 나타난다. 이는 본 알고리즘이 캐릭터에 사용되는 텍스처에 큰 영향을 받지 않으면서 코믹북 스타일을 표현할 수 있음을 보여준다. 그리고 캐릭터가 애니메이션 되면서도 [그림 24]와 같이 시간적 동

일성(temporal coherence)를 유지한 상태로 코믹북 스타일이 효과적으로 표현된다.



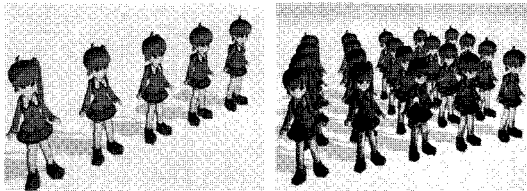
[그림 24] 애니메이션 되는 캐릭터의 결과

4.2 성능 측정

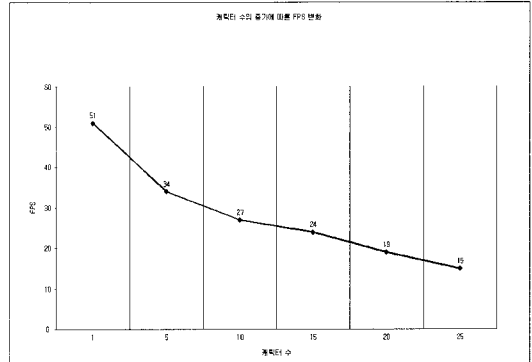
CPU	Graphics Card	fps(전/후)	
Athlon64 X2 Dual 4400+ 2.22Ghz	GeForce 7800 GTX	129	190
Athlon64 4400 1.98Ghz	GeForce 6800 GT	49	97
Core 2 Duo T7200 2.0Ghz	GeForce Go 7600	52	108

[표 3] 컴퓨터 사양별 fps(frames per second)

본 알고리즘이 게임에 적합한지를 검증하기 위해 성능이 다른 세 대의 컴퓨터에서 1024 x 768의 해상도로 게임을 실행시킨 뒤 fps(Frame/sec)를 확인하였으며, 그 결과는 [표 3]과 같다. 측정에 사용된 캐릭터의 폴리곤 개수는 약 3400여 개이며, fps는 본 알고리즘이 적용되기 전과 후를 모두 측정하여 표기하였다.



[그림 24] 다수 캐릭터의 렌더링 이미지



[그림 25] 캐릭터 증가에 따른 fps 변화 측정 결과

[그림 24]와 [그림 25]는 각각 본 연구의 결과가 적용된 캐릭터의 수를 증가시켜서 얻은 이미지와, 캐릭터 수의 증가에 따른 fps의 변화량을 보여준다. 측정에는 [표 3]의 세 번째 컴퓨터 환경이 사용되었다. 그래프에서 확인할 수 있는 것과 같이 fps는 폴리곤의 수가 증가함에 따라 선형적으로 감소하게 되며, 자연스러운 움직임을 주는 기준으로 24fps으로 할 경우, 최대 약 15개의 캐릭터가 등장하는 화면에서 원활한 게임 플레이가 가능하다.

4.3 맺음말

본 연구에서는 코믹북 스타일의 특징을 분석하여 표현 요소들을 도출하고 이를 게임에 활용할 수 있는 렌더링 알고리즘들을 제시하였다. [그림 22]와 [그림 23]에서 확인할 수 있듯이 본 연구의 기법을 적용한 게임은 코믹북 스타일의 비주얼을 표현해 내었으며, [표 3]의 성능 측정 결과를 통해 GeForce 6800이상의 그래픽 하드웨어에서 실시간으로 구현이 가능하다는 것을 확인할 수 있었다. 또한 [그림 24]의 스크린 샷을 통해 애니메이션 되는 캐릭터에서 시간적 동일성을 유지하며 코믹북 스타일의 비주얼이 렌더링 되는 것을 검증할 수 있었다.

본 연구의 알고리즘을 적용한 게임(게임명 : ‘쓱쓱’, 팀명: ‘Studio Type B’)은 한국게임개발자협회가 주관하는 ‘2007 게임 아이디어 및 인디게임 공모전’에서 대상(문화관광부 장관상)과 한국게임산업협회와 한국게임산업진흥원이 주관하는 ‘2007 대한민국 게임대상’에서 아마추어상(게임산업진흥원장상)을 수상하여 게임 비주얼의 차별성을 인정받았다[30, 31].

본 알고리즘에서는 TAM이라는 텍스처들을 이용하여 실

시간 해칭을 구현한다. 이 기술은 구현 결과에서 보듯이 게임에 적합한 비주얼을 생성해 내며, 플레이에 불편함이 없을 정도의 실시간 처리를 보장한다. 그러나 해칭 효과의 품질을 높이기 위해서는 고해상도의 TAM가 필요하고, 1024 × 1024의 해상도(약 5메가)를 가지는 여섯 개의 텍스처가 픽셀 셰이더의 샘플러(Sampler)로 이동함에 따라, 노트북이나 AGP 인터페이스의 비디오 카드와 같이 메모리 간의 전송 대역폭이 낮은 환경에서는 프레임 저하의 문제가 발생하기도 한다. 그리고 실시간 처리를 위해 해칭의 방향을 결정하는 디렉션 필드 계산을 생략하고 UV 맵핑으로 이를 대체했는데, 좀 더 높은 품질의 해칭 결과를 얻기 위해서는 계산 속도가 개선된 디렉션 필드 알고리즘의 개발이 필요하다. 이와 같이 해칭 결과를 발전시키기 위해서는 알고리즘의 처리 성능의 개선이 필요하다.

참고 문헌

[1] 김성수, "GPU/HLSL을 이용한 3D 게임에 최적화 된 수목화 셰이더", 2005

[2] <http://www.mabinogi.com>

[3] <http://www.capcom.com/okami/>

[4] <http://valkyria.jp/>

[5] <http://www.jsrf.com>

[6] <http://www.atari.com/dragonballz/>

[7] <http://www.bandainames.com/channel.or.jp/list/hagane/>

[8] <http://naruto.namcobandainames.com/>

[9] <http://www.xiii-thegame.com/>

[10] <http://www.zelda.com/gcn/>

[11] <http://www.capcom.com/vj/>

[12] <http://crackdownoncrime.com/>

[13] <http://www.grigon.co.kr/game.html?mode=package>

[14] <http://www.sealonline.co.kr/>

[15] <http://fs.joycity.com/>

[16] <http://p.paran.com/>

[17] <http://skidrush.hangame.com/>

[18] <http://paperman.hangame.com/>

[19] <http://game4.netmarble.net/ma9/>

[20] <http://www.pcik2.co.kr/>

[21] <http://shining-force.jp/exa/>

[22] <http://www.cs.wisc.edu/graphics/Gallery/NPRQuake/>

[23] Jung Seo Kim, Kang Soo You and Hoon Sung Kwak, "An Efficient Expression on Cartoon Rendering Scheme in Game Characters", 5th ACIS International Conference on Software Engineering Research, Management & Applications (SERA 2007), pp. 924-928, Aug. 2007.

[24] Lee Markosian, Michael A. Kowalski, Samuel J. Trychin, Lubomir D. Bourdev, Daniel Goldstein and John F. Hughes, "Real-Time Nonphotorealistic Rendering", Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp. 415-420, 1997.

[25] Emil Praun, Hugues Hoppe, Matthew Webb and Adam Finkelstein, "Real-Time Hatching", Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 581, 2001.

[26] Aaron Hertzman and Denis Zorin, "Illustrating smooth surfaces", Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp. 517-526, 2000.

[27] Matthew Webb, Emil Praun, Adam Finkelstein and Hugues Hoppe, "Fine Tone Control in Hardware Hatching", Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering, pp 53 ff, 2002.

[28] "A Survey of silhouette detection techniques for non-photorealistic rendering", Image and Graphics, 2004. Proceedings, Third International Conference on, pp. 434-437, Dec. 2004

[29] Sobel operator, http://en.wikipedia.org/wiki/Sobel_operator.

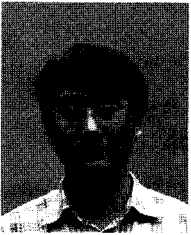
[30] <http://www.gamecompo.or.kr/2007/company/company4.asp>

[31] <http://www.gamek.or.kr/Mboard.asp?exec=view&strBoardID=main&intSeq=2835>



김 태 규 (Tae-Gyu Kim)

2006년 아주대학교 미디어학부 (학사)
2006년 ~ 현재 아주대학교 미디어학과 (석사과정)



오 규 환 (Gyuhwan Oh)

1998년 한국과학기술원 전산학과 박사
2000년 ~ 2005년 (주)넥슨 게임 개발 실장
2005년 ~ 현재 아주대학교 미디어학부 조교수
2005년 ~ 현재 (주)넥슨 기술고문



이 창 신 (Chang-Shin Lee)

2002년 ~ 2003년 (주)나스카 게임 기획
2003년 ~ 2006년 (주)토이소프트 게임 기획
2001년 ~ 현재 아주대학교 미디어학부 학사