

3D 그래픽 게임에서 모션블러 효과 개발을 위한 연구

최학현⁰ 이명학 김정희
게임솔루션 부설연구소

woono7@yahoo.co.kr⁰, lemai@hanmail.net, medherbs@empal.com

A Study on The Development of Motion Blur Effects based for 3D Graphic Game

Hakhyun Choi⁰ Myunghak Lee Junghee Kim
Dept. of Game Lab, Game Solution

요약

오늘날 2D 그래픽 게임 개발에서 점차적으로 3D 그래픽 게임을 만들게 되는 추세이다. 3D 그래픽 게임의 문제는 역시 그래픽 처리 속도이다. 많은 3D 그래픽 게임 개발사들이 제한된 하드웨어의 성능으로 인하여 표현에 한계를 받고 있는 것이 사실이다. 또한 하드웨어의 발달의 3D 그래픽 게임을 즐기는 게임 유저들에게 보다 나은 게임 환경에 대한 욕구가 늘어나게 된 것이다. 3D 그래픽 게임에 화려하고 멋진 그래픽 기술들이 존재한다. 그러한 기술들은 고성능의 하드웨어를 요구하는 것들이 많고 실제적으로 게임에서 사용되는 데는 많은 어려움을 가지고 있다. 본 논문에서는 이러한 어려운 문제점의 하나인 Motion Blur Effect의 문제점을 파악하고 이를 해결 할 수 있는 방법들을 모색하고자 하였으며 3D 그래픽 게임에서의 Motion Blur Effect의 가장 큰 문제점인 게임 속도의 전반적인 속도 저하를 해결하는 방법과 프로그램 디자인과 프로그램 구현을 Offscreen Buffer를 이용하는 데에 초점을 두었다.

1. 서론

3D Graphic Game에서의 Motion Blur 효과 영상은 물체에 시간차를 두고 생기는 잔상들을 하나의 속도감 있는 Image로 얻는 것을 말한다. 많은 3D 그래픽 게임개발자나 연구원들이 이러한 Texture Image Data를 얻기 위해서 아직까지는 하나의 물체를 여러번 Rendering을 하여 마치 속도감이 있는 물체인 것처럼 표현하고 있다. 그러나 얻어내는 결과치에 비해 처리 속도가 현저하게 떨어지므로 3D 그래픽 게임에서 잘 활용하지 못하는 실정이다. 결국 빠른 그래픽 연산 처리가 탑재된 컴퓨터가 아니고서는 3D 그래픽 게임내에서의 모션 블러 영상 처리 효과는 Game User들의 선택 사항이 되어 버렸다. 본 논문은 게임에서 표현하고자 하는 모션블러 영상 효과를 속도 저하라는

제약에서 벗어나서 3D 그래픽 게임의 표현의 자유를 얻고자 함에 있다. 기존의 모션블러 효과의 구현 연구는 주로 게임개발사 프로그래머들이나 그래픽카드 제조회사의 연구원들이 일반 렌더링 기술의 기본 원리를 이용하여 구현 한 것에 불과하다. 그러나 현재 사용되는 모션블러 효과 구현의 단점과 문제점을 파악하여 그에 대한 해결책을 제시하고 그것을 Offscreen Buffer를 통해서 프로그램을 제작하고 구현하여 3D 그래픽 게임에서의 처리속도의 향상 여부를 점검하는데 초점을 두고 있다.

2. 문제점 및 해결 방안

2-1. 기존 모션블러 효과의 문제점

2.1.1 구현의 복잡성

누적버퍼를 사용하여 모션블러 효과를 처리하기 위해서는 먼저 현재 프레임을 그리고 50%만 누적버퍼에 로드한다.

2.1.2 누적버퍼의 사용 제약

DirectX 라이브러리에서 누적버퍼를 이용한 모션블러 효과를 사용하기 위해선 누적버퍼를 임시로 만들어야 하는 점은 멀티 플랫폼의 게임제작에 어려움을 발생시키는 문제점이 된다.

2.1.3 3D 그래픽카드 가속 기능의 미활용

3D 그래픽 카드의 최대의 장점인 3D 그래픽 하드웨어 가속을 사용할 없는 누적버퍼를 사용한 모션블러 효과는 3D 그래픽 게임 프레임에 현저한 속도 저하의 문제를 안게 된다는 점이다.

2.1.4 반복 렌더링 연산처리와 버퍼의 낭비

잔상이 많아지면 많아 질수록 렌더링 연산은 길어져 프레임 속도가 저하되고 버퍼의 사용도 많아짐으로 인해 많은 양의 메모리를 낭비하게 되는 문제점이 발생한다.

2.1.5 모션블러 효과의 비가속 처리

모션블러 효과도 3D 그래픽 하드웨어 가속 기능을 활용 할 수 있다면 빠른 처리가 가능하다.

2-2. 연구 목표 및 해결 방안

2.2.1 누적버퍼 사용의 대체 방안 연구

누적버퍼를 사용한 모션블러 효과 기법은 일반적인 모션블러 효과 기법에서처럼 많은 렌더링과 많은 버퍼량을 소비 하지 않는다는 장점이 있다.

2.2.2 텍스처 이미지의 가속 처리 연구

3D 그래픽 게임에서의 모션블러 효과를 구현함에 있어 HAL 처리가 가능한 텍스처 이미지 데이터의 가속 처리 기법을 이용하여 처리하도록 하며, 이러한 기법을 사용하도록 하는 해결 방법을 제시하는데 비중을 두고 있다.

2.2.3 하드웨어 가속 지원

최근에 오프스크린도 렌더링에 가속을 지원받을 수 있게 되었다. 그러나 아직까지도 오프스크린의 이용은 버려진 상태이다. 본 연구는 이 새로운 하드웨어

가속 지원을 본격적인 해결 방안으로 이용하고자 한다.

3. 오프스크린버퍼를 이용한 모션블러 효과

텍스처서피스(Texture surface)를 이용하면 위의 문제점들이 한 번에 해결이 되고, 이 기능을 잘 활용하여 모션블러효과 데이터를 3D 그래픽 카드의 하드웨어 가속 도움을 받게 된다면 연구의 목적을 이룰 수 있다.

3-1. 누적버퍼 사용의 대체 방안

프레임 버퍼가 렌더링 타겟이나 새로 만든 텍스처 서피스를 새로운 렌더링 타겟으로 지정하게 되면 그 이후에 렌더링 오퍼레이션 들은 그 새로운 버퍼로 발생하게 된다. 그 새로운 버퍼는 텍스처 서피스로 사용할수 있고 렌더링 버퍼로도 쓸수 있다. 이 새로운 버퍼를 오프스크린버퍼로 이용하고자 한다.

3-2. 모션블러 효과의 가속 구현방법

- 현재 장면의 모션블러 효과가 작동되지 않는 객체들만을 그린다.
- 이전 누적된 오프스크린버퍼 이미지를 블러링 하여 약하게 한다.
- 오프스크린버퍼의 누적된 블러 이미지를 렌더링 한다.
- 현재 장면의 모션블러 효과 객체들을 그린다.
- 현재 장면의 모션블러 효과 객체들을 오프스크린버퍼에 렌더링 한다.

4. 모션블러 효과 설계 및 구현

오프스크린버퍼를 이용한 모션블러 효과처리 테스트기는 메인 윈도우, 메뉴, 로더, 렌더, 허드, 모션블러 효과처리로 설계되어 있다.

4-1. 설계

4.1.1 메인 윈도우

전체화면 윈도우 혹은 창모드 윈도우등으로 구성되며 이 소프트웨어를 시작하기 위한 처음 생성 부분이다. 메인 윈도우의 초기 크기에 따라서 모든 비주얼썬은 그에 맞게 조정이 되게 제작되어 있다.

4.1.2 메뉴

메뉴에선 테스트 하고 싶은 형태를 고르거나 현재 자신의 컴퓨터 사양에 대한 정보, 혹은 실험 환경 시스템 환경 설정, 각종 기타 설정 및 데이터의 저장 및 세이브등을 담당하고, 프로그램 종료와 테스트 진행을 할 수 있도록 구성되어 있다.

4.1.3 로더

테스터기의 특징상 3D Graphic 이기 때문에 월드를 구성하기 위한 데이터, 캐릭터 데이터, 각종 텍스처 데이터와 음향 데이터등을 로딩한다.

4.1.4 3D 그래픽 엔진 및 렌더

모든 장면은 렌더링 과정을 거쳐서 최종 이미지로 볼 수 있게 해주는 역할을 담당하고 있으며 이부분은 미리 제작해 둔 3D 그래픽 엔진을 이용하여 표현할 수 있다.

4.1.5 모션블러 효과처리

기본 누적버퍼를 이용한 모션블러 효과처리와 오프스크린버퍼를 이용한 모션블러 효과처리를 구현하여 실제로 비교해 볼 수 있도록 선택적으로 처리를 해준다.

4.1.6 HUD

렌더링 처리 속도와 이미지 데이터의 폴리곤 수, 텍스처 데이터의 양 혹은 기타 정보들에 대해서 정확하고 자세히 실시간으로 처리하여 보여 준다.

4-2. 구현

4.2.1 방법

- 테스트를 위해 필요한 100m * 100m * 50m 의 공간을 생성한다.
- 마련된 공간에 10 명의 캐릭터를 배치한다.
- 누적버퍼를 이용하여 모션블러 효과를 처리했을 경우 처리속도를 측정한다.
- 오프스크린버퍼를 이용한 모션블러 효과를 처리했을 경우 처리속도를 측정한다.
- 위와 같은 방법으로 처음부터 다시 하되 각각 20 명, 30 명으로 늘려 테스트를 한다.

4.2.2 결과

다음은 누적버퍼를 이용한 모션블러 효과처리와 오프스크린버퍼를 이용한 모션블러 효과처리로 얻은 결과 데이터이다.

[표 1] 실험결과 데이터

누적버퍼	이용시	오프스크린	이용시
------	-----	-------	-----

	프레임	프레임
10 명	50	62
20 명	30	38
30 명	20	28

[표 1]은 오프스크린버퍼를 이용한 모션블러 효과를 구현하면 누적버퍼를 이용했던 것 보다 훨씬 빠른 속도의 처리 결과를 얻게 된다는 것을 볼 수 있다.

5. 결론

본 논문에서 오프스크린버퍼를 이용하여 모션블러 효과를 구현하여 처리속도가 향상 되었음을 구현하기 위한 설계와 구현을 해보았다. 게임에서 모션블러 효과를 이용 하면 화려한 게임장면의 연출이 가능하다. 그러나 게임의 전체적인 속도저하를 심하게 일으키는 문제점을 해결하고자 하였다. 본 논문에서는 가장 효과적이고 최신의 기법인 누적버퍼를 이용한 모션블러 효과구현 방법을 알아 보았고, 그의 문제점을 파악하여 보완한 오프스크린버퍼를 이용하는 방법을 제안하였다. 이러한 처리의 향상은 한정된 시간에 보다 많은 3D 그래픽 데이터 처리가 가능해 지고, 앞으로 실시간에서 보다 실사와 가깝게 영상을 출력할 수 있게 될 것이며, 좀 더 발전된 기술들을 이용한 실시간 렌더링 기법으로 영화와 똑같은 3D 그래픽 게임을 즐길 수 있으리라 생각된다.

6. 참고 문헌

[1] Roger T.Stevens, Computer Graphics Dictionary CHARLES RIVER MEDIA, 2001

[2] Richard S. Wright, Jr. Michael Sweet, /남기혁 역, OpenGL Super Bible Second Edition, 인포북, 2001

[3] Mason Woo, Jackie Neider, Tom Davis, Dave Shreiner, OpenGL Programming Guide Third Edition ADDISON-WESLEY, 2001

[4] Adrain Perez with Dan Royer, /이주리 역, Advanced 3D Game Programmin using DirectX 민프레스, 2001