

텍스처의 파일 저장형식에 따른 게임 로딩 시간 비교를 통한 효과적인 최적화 기법에 관한 연구

- Virtools 엔진을 기반으로 -

A Study on Effective Optimization by Comparison
with Game Loading Time for File Format of Textures
- Based on Virtools Engine -

채현주, 유석호, 경병표
공주대학교

Chae Heon-Joo, Ryu Seuc-Ho, Kyung Byung-Pyo
Kong-ju National Univ.

요약

게임이 실행될 때 로딩이라는 과정을 거치게 되는데 이때 텍스처는 게임의 실행 파일 외부에 따로 저장되어 있는 이미지 파일을 불러오게 되는 형태를 취하는 경우가 많다. 이에 이미지 파일의 저장 형식에 따라 로딩 시간이 달라지는 현상이 생기는데 본 연구에서는 다양한 저장 형식에 따라 텍스처 파일의 로딩 시간과 저장 용량이 어느 정도 차이가 나는지를 확인해 보고 그 결과를 토대로 텍스처 사용에 대한 최적의 방법을 제시하고자 한다.

Abstract

A game starting, the process of running has some loading steps. In general, textures use corresponding images saved externally and the file format of images affects the loading time of game. We propose some ideas for the texture using method by comparison with loading time according to some file formats.

I. 서론

컴퓨터 게임에 사용되는 텍스처는 생생한 현장감과 사실감을 높여 주는 매우 중요한 요소이다. 하지만 텍스처는 비교적 많은 정보를 담고 있는 이미지 파일을 사용하므로 게임의 성능에 많은 영향을 미치게 된다. 텍스처가 그래픽카드 내에서 어떤 형식을 가지고 있느냐에 따라 게임이 실행되는 동안 실시간 렌더링 성능에 영향을 주기도 하고 그 크기와 색상 깊이에 따라서 처리 능력에 영향을 주기도 한다[2]. 일반적으로 컴퓨터 게임이 실행될 때는 실행 초기에 로딩이라는 과정을 거치게 된다. 이때 텍스처는 실행파일의 내부에 포함되어 있지 않고 외부에 저장되어 있는 이미지 파일을 불러오는 경우가 대부분이다[3]. 텍스처가 실행파일의 외부에 이미지 파일 형태로 저장되어 있으므로 이 이미지 파일이 어떤 파일 형식을 가지고 있느냐에 따라 로딩 시간에 영향을 미치게 된다. 본 연구에서는 서로 다른 이미지 파일의 저장 형식을 비교함으로써 텍스처의 로딩 시간에 대한 최적화 방법을 찾고 그 자료를 제시하고자 하였다.

본 연구를 위하여 5 가지의 형식으로 저장된 이미지 파일을 준비하여 이미지의 크기를 다양하게 변경하면서 로딩 시간을 측정하여 보고 그 결과 분석해 보았다. Virtools 엔진을 기반으로 테스트하였으며, 테스트 결과를 토대로 게임의 로딩 시간

과 이미지 파일의 저장 형식과의 상관관계를 살펴보고 최적의 저장 방법에 대해 고찰해 보았다.

II. 본론

1. 저장 형식별 특징

본 연구에서 사용된 이미지 파일의 저장 형식은 총 5 종이다. 각 종류별 특징을 살펴보면 다음과 같다.

1.1 BMP

압축하지 않은 이미지를 저장하는 포맷으로, 윈도우즈와 OS/2에서 기본으로 사용하고 있는 형식이다. 데이터를 비효율적으로 저장하므로 실제로 필요한 크기보다 큰 파일을 만드는 단점이 있다. 레이어와 알파채널은 지원하지 않는다[4].

1.2 JPG

소프트웨어와 플랫폼에 독립적인 형식이며 파일 용량을 압축하여 이미지 저장에 가장 많이 사용된다. 인터넷상에서 가장 많이 사용되는 형식 중의 하나로 압축률이 가장 뛰어나다[4].

1.3 DDS

DirectDraw Surface의 약자로 Microsoft社 DirectX 게임

개발을 지원하기 위해 만든 형식이다. 단순 이미지 정보뿐만 아니라 환경맵, 밍맵을 함께 저장할 수 있다[5].

1.4 TGA

트루비전社의 타가 비디오 보드를 위하여 고안된 형식으로 32비트 색상은 3D 그래픽에서 유용하게 사용되며, 고휘상도 그래픽 전문작업에 많이 사용되고 있다[4].

1.5 PNG

Portable Network Graphics의 약자로, 인터레이스 기능을 지원하며 JPEG와 같이 압축률을 높이기 위해 등장한 형식이다. GIF처럼 색상수를 줄여 압축하는 형식이며, 256으로 한정된 GIF와는 달리 1600만 컬러모드로 저장이 가능하고 GIF 보다 10~30% 정도의 뛰어난 압축률을 제공한다. 트루컬러를 지원하며 비손실 압축을 사용하며 이미지의 투명성과 관련된 알파 채널에서 향상된 기능을 제공한다[4].

2. 비교 방법

2.1 비교 환경

1) 텍스처 샘플

본 연구에서 사용된 텍스처의 샘플은 그림 1과 같은 3종류의 이미지이다. 각각의 이름을 왼쪽부터 차례대로 TS-1, TS-2, TS-3라고 하였다. 이 이미지들은 투명도가 없는 24bit 색상 깊이를 가지고 있다.



▶▶ 그림 1. 텍스처 샘플

2) 그래픽 API

3차원 그래픽을 표현하는 데에는 크게 OpenGL과 DirectX가 사용되지만, 컴퓨터 게임에서 DirectX가 주로 사용된다. 이에 본 연구에서도 Microsoft 社의 DirectX 9.0c를 주 API로 사용하였다.

3) 시스템 사양

모든 비교는 모두 2개의 하드웨어 시스템에서 이루어졌으며 각각 S-1, S-2라 명명하였다. 시스템 S-1은 Intel Pentium4 2.8GHz의 CPU에 512MB RAM, 64MB의 그래픽카드가 장착되어 있으며, 시스템 S-2는 Intel Pentium4 3.0GHz의 CPU에

1GB RAM, 128MB의 그래픽카드가 장착되어 있다.

2.2 비교 조건

1) 파일 저장 형식(File Format)

이 항목은 본 연구의 주된 비교 대상이다. 비교된 파일 형식은 총 5 종류로서 BMP, JPG, DDS, TGA, PNG 이다. 이 5종의 파일 형식에 따른 로딩시간이 차이가 있는지를 확인해 보고, 차이가 있다면 어떤 형식이 가장 짧은 로딩 시간을 갖는지를 분석해 보았다.

2) 텍스처 크기(Texture Dimension)

3가지의 텍스처 샘플의 크기를 각각 변경시켜 가면서 비교해 보았다. 일반적으로 그래픽카드에서 텍스처의 처리 능력을 극대화 하려면 텍스처의 가로와 세로의 픽셀 단위의 크기가 2의 거듭제곱 형태로 되어야 한다[6]. 그러므로 본 연구에서도 텍스처의 크기를 128, 256, 512, 1024, 2048 등으로 가로와 세로를 동일하게 제작하여 사용하였다.

2.3 비교 도구와 방법

본 연구의 주요 비교 도구로는 Virtools 엔진을 사용하였다. Virtools Dev에서 텍스처 요소를 하나 만들고, 이 텍스처에 저장되어 있는 이미지 파일을 동적으로 로딩하는 Script를 작성한 후 이미지 별로 100번을 수행함으로써 그 평균값을 가지고 비교하여 보았다[7][8].

3. 로딩 시간 비교 분석

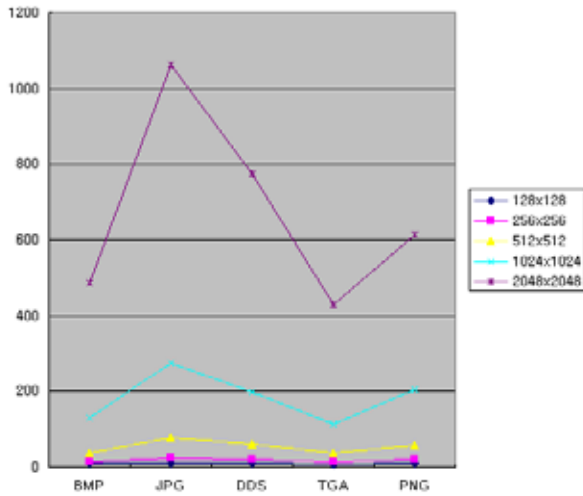
텍스처로 사용되는 이미지의 파일 저장 형식에 따라 로딩 시간을 비교하기 위하여 두 곳의 하드웨어 시스템 즉, S1과 S2에서 테스트를 하였고 세 종류의 텍스처 샘플 즉, TS-1, TS-2, TS-3에 대하여 텍스처의 크기를 변경시켜 가며 그 결과를 비교하여 보았다.

3.1 시스템 S-1에서의 비교

1) 텍스처 샘플 TS-1에 대한 로딩시간 비교

시스템 S-1에서 텍스처 샘플 TS-1을 비교한 결과는 그림2와 같다. 세로축은 로딩시간을 나타내며 단위는 밀리초이다. 모두 100번을 로딩하여 그 시간을 합산한 다음 평균을 구하여 결과치를 도출하였다. 그림에서도 알 수 있듯이 5개의 파일 저장 형식 중에서 TGA 파일 형식이 가장 로딩시간이 짧았고 JPG 파일 형식이 가장 길었다. 또한 텍스처의 크기가 클수록 그 폭이 커지는 것을 확인할 수 있다. 텍스처의 크기가 256x256 보다 작은 경우에는 그 차이가 미미하지만 그 이상으로 될 경우에는 파일 저장 형식 간의 차이가 큰 폭으로 증가한

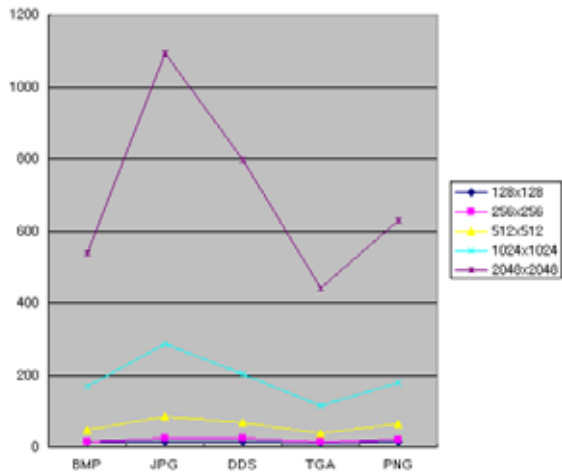
다. 두 번째로 로딩 시간이 짧은 파일 형식은 BMP로서 TGA와 거의 비슷하다.



▶▶ 그림 2. TS-1에 대한 로딩시간 비교

2) 텍스처 샘플 TS-2에 대한 로딩시간 비교

시스템 S-1에서 텍스처 샘플 TS-2을 비교한 결과는 그림3과 같다. TS-1의 경우와 마찬가지로 5개의 파일 저장 형식 중에서 TGA 파일 형식이 가장 로딩시간이 짧았고 JPG 파일 형식이 가장 길었다. 로딩시간이 짧은 순서대로 파일 저장 형식을 나열하면 TGA, BMP, PNG, DDS, JPG 순이다. 이 결과는 TS-1인 경우와 같다.

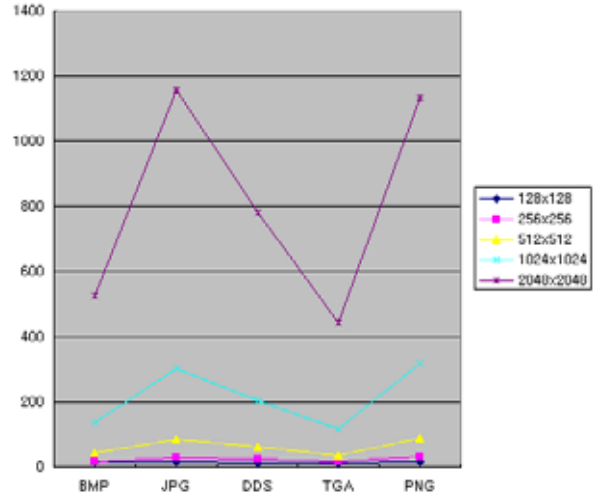


▶▶ 그림 3. TS-2에 대한 로딩시간 비교

3) 텍스처 샘플 TS-3에 대한 로딩시간 비교

시스템 S-1에서 텍스처 샘플 TS-3을 비교한 결과는 그림4와 같다. 앞의 두 경우와 마찬가지로 5개의 파일 저장 형식 중에서 TGA 파일 형식이 가장 로딩시간이 짧았지만 PNG 형식의 경우 앞에서 확인했던 TS-1, TS-2와는 그 결과에 있어서 차이가 있었다. 그림에서 표시하는 바와 같이 앞의 두 경우에

서는 PNG 형식이 로딩시간이 빠른 편에 속해 있었지만, 이번 경우에는 JPG 형식과 거의 비슷한 수준의 낮은 로딩 속도를 보이고 있다.

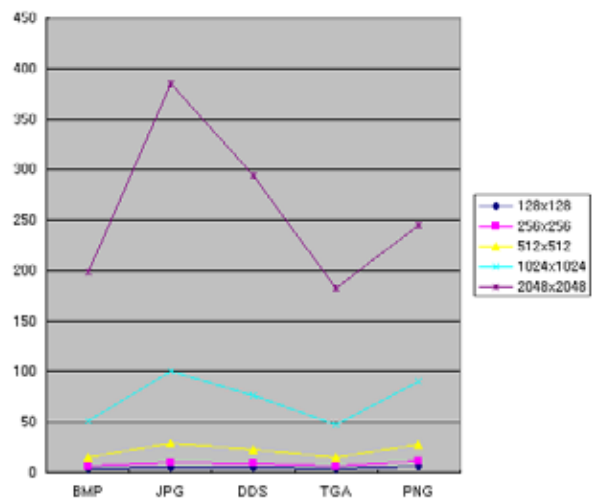


▶▶ 그림 4. TS-3에 대한 로딩시간 비교

3.2 시스템 S-2에서의 비교

1) 텍스처 샘플 TS-1에 대한 로딩시간 비교

시스템 S-2에서 텍스처 샘플 TS-1을 비교한 결과는 그림5와 같다. 그림2와 비교해 보면 로딩시간의 절대값이 더 작아진 것 외에는 큰 차이가 없다는 것을 확인할 수 있다. 5개의 파일 저장 형식 중에서 TGA 파일 형식이 가장 로딩시간이 빨랐고 JPG 파일 형식이 가장 길었다. 특이한 것은 앞의 S-1에서와는 달리 PNG에 있어서는 이미지의 크기에 따라 로딩시간이 DDS보다 빠를 경우도 있고 느릴 경우도 있다는 것이다.

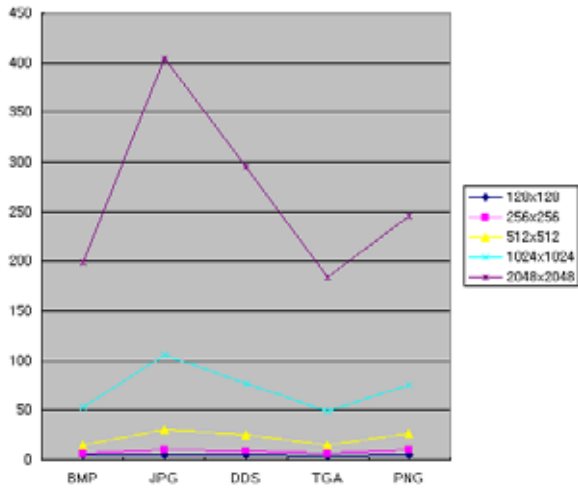


▶▶ 그림 5. TS-1에 대한 로딩시간 비교

2) 텍스처 샘플 TS-2에 대한 로딩시간 비교

시스템 S-2에서 텍스처 샘플 TS-2를 비교한 결과는 그림6

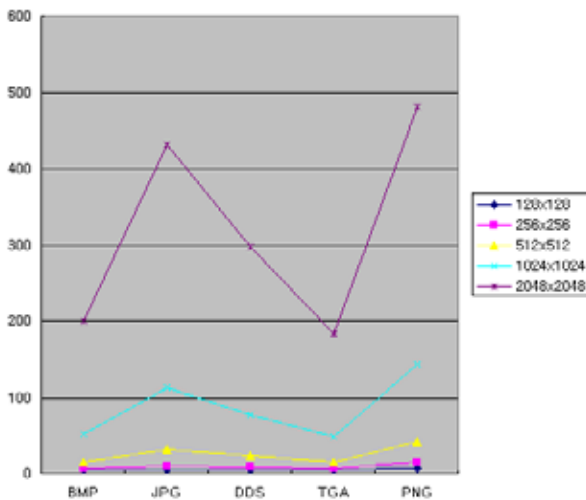
과 같다. 그림 3과 비교해 보면 로딩시간의 절대값이 더 작아진 것 외에는 큰 차이가 없다는 것을 확인할 수 있다.



▶▶ 그림 6. TS-2에 대한 로딩시간 비교

3) 텍스처 샘플 TS-3에 대한 로딩시간 비교

시스템 S-2에서 텍스처 샘플 TS-3을 비교한 결과는 그림7과 같다. 그림 4와 비교하여 보면 그 결과가 로딩시간의 절대값이 전체적으로 짧아진 것 외에는 큰 차이가 없다. 시스템 S-1의 경우와 마찬가지로 PNG 형식이 로딩시간이 빠른 편에 속해 있었지만, 이번 경우에는 JPG 형식과 거의 비슷한 수준의 낮은 로딩 속도를 보이고 있다.



▶▶ 그림 7. TS7-3에 대한 로딩시간 비교

III. 결 론

본 연구는 컴퓨터 게임이 실행될 때, 게임의 로딩 시간이 텍스처의 원본 이미지의 파일 저장 형식에 따라 어떻게 영향을 받는가에 대한 것이다. 대부분의 게임은 텍스처에 사용되는 이

미지의 파일을 실행파일의 외부에 저장해 놓고 게임이 시작될 때 동적으로 불러와서 사용하게 된다. 이에 외부에 저장되어 있는 이미지의 파일 저장 형식에 따라 게임의 로딩 시간이 영향을 받을 것이라는 가정을 하고 적절한 환경과 다양한 형식의 파일을 준비하여 테스트를 진행하였다. 총 2 대의 시스템에서 테스트를 진행하였으며, 3 종류의 텍스처 샘플을 5가지의 이미지 크기로 만들고, 5 종류의 이미지 저장 포맷인 BMP, JPG, DDS, TGA, PNG로 저장한 다음, Virtools 엔진을 이용하여 100번씩 로딩한 후 그 시간을 평균하였다.

이 테스트를 통하여 몇 가지 사실을 알게 되었는데, 첫째는 BMP, JPG, DDS, TGA, PNG 중에서 TGA 파일 형식에 대한 로딩 시간이 상대적으로 가장 짧다는 것이다. 둘째는 로딩 시간은 이미지의 크기가 256x256 이하에서는 저장 형식 간 차이가 그리 크지 않았지만 512x512 이상의 크기에서는 크기가 커질수록 로딩 시간의 차이가 큰 폭으로 커진다는 것이다. 셋째는 PNG 파일에 대한 것인데, 다른 저장 형식에 비해서 이미지의 내용이 어떤 것이냐에 따라 같은 조건이라 하더라도 로딩 속도가 크게 차이 난다는 것이다.

이 세 가지 사실을 통해 도출해 낼 수 있는 결론은 다음과 같다. 이미지의 크기가 256x256 이하인 경우에는 어떤 저장 형식을 사용해도 게임의 로딩 시간에는 크게 영향을 미치지 않으므로 되도록이면 저장 공간이 적게 차지하는 것이 최적화에는 도움이 될 것이다. 즉, BMP, DDS, TGA 파일은 압축이 되지 않은 파일 형식이므로 PNG나 JPG 형식을 사용하는 것이 좋다. 하지만 이미지 크기가 512x512 이상인 경우에는 TGA 파일 형식을 사용하는 것이 최적화에 도움이 될 수 있다. 만약 이 경우에 파일 용량을 고려해야 한다면 JPG 보다는 PNG를 사용하는 것이 좋지만, 이미지의 저장 내용에 따라 JPG 보다 PNG가 더 긴 로딩 시간을 가질 수도 있으므로 반드시 테스트를 거친 후에 사용하는 것이 좋을 듯하다.

본 연구에서는 2대의 시스템에서 24bit 색상 깊이를 갖는 경우만 테스트를 수행하였는데 향후 연구에서는 본 연구를 확장하여 좀더 다양한 시스템에서 여러 가지 색상 깊이를 갖는 이미지를 테스트함으로써 일반적인 결과를 얻어낼 예정이다.

■ 참고 문헌 ■

- [1] 유석호, TLC Rendering을 활용한 효과적인 3D합성에 관한 연구, 한국정보과학회, 2004.
- [2] 채현주, 유석호, 경병표, 다양한 텍스처 형식에 따른 실시간 렌더링의 FMQ 비교를 통한 효과적인 최적화 기법에 관한 연구, 한국콘텐츠학회 2005 추계 종합학술대회, pp.13~18, 2005.
- [3] OpenGL Game Programming, pp.253, 정보문화사, 2001.
- [4] <http://www.dcinside.com/study/grapic.htm>
- [5] DirectX Documentation for C++, Microsoft DirectX 9.0 SDK

- [6] Kelly Dempski, Real-Time Rendering Tricks and Techniques in DirectX, pp.190, Course Technology PTR, 2002.
- [7] Virtools Dev User Guide, Virtools SA, 2004.
- [8] Virtools Online Reference, Virtools SA, 2004.
- [9] Tom Meigs, Ultimate Game Design: Building Game Worlds, p.65, McGraw-Hill, 2003.
- [10] Kenneth C. Finney, Advanced 3D Game Programming All in One, p.25, Course Technology PTR, 2005.