Journal of Korea Game Society 2013 Dec; 13(6): 35-42
http://dx.doi.org/10.7583/JKGS.2013.13.6.35

게임 프로그래밍

모바일 단말기에서 이미지 처리에 필요한 메모리 사용량을 줄이기 위한 타일화 이미지 압축 기법

오황석
한국산업기술대학교 게임공학부
hsoh@kpu.ac.kr

Tiled Image Compression Method to Reduce the Amount of Memory Needed for Image Processing in Mobile Devices

Hwang-Seok Oh
Dept. of Game and Multimedia Eng., Korea Polytechnic Univ.

요 약

본 논문에서는 모바일 기기의 프로세서 처리 능력과 메모리 용량의 제한에 관계없이 거대 크기의 이미지를 게임의 배경으로 사용하기 위한 타일화된 이미지 압축 기법을 제안한다. 실험을 통하여 제안한 기법이 압축 효율 측면에서는 하나의 PNG 파일로 압축하는 방법과 유사하나 모바일 단말기의 메모리 제한으로 인한 디코딩 가능한 이미지의 크기 제한 문제를 해결할 수 있음을 보였다. 또한 실행 시 부분 디코딩 기법을 적용하여 단말기 화면에 출력되는 이미지 부분만 디코딩하여 출력함으로 초기의 대규모 이미지의 로딩 시간을 줄일 수 있음을 보였다.

ABSTRACT

A new compressed image format is proposed to use a large size of image in mobile games without the constraints of hardware specifications such as memory amount, processing power, which encodes each block of a large size image in scan line order. Using the experiments, we show the effectiveness of proposed method compared with a general PNG in terms of compression ratios and required memory in decoding processes. Also, the loading delay can be reduced by decoding only the displaying area of a large image in run-time.

Keywords : Large Image Encoding(큰 이미지 압축), Tiled Image(타일화된 이미지), Memory Constraints(메모리 제약), PNG(Portable Network Graphics), Run-time decoding(실행 시간 디코딩)

Received: Nov. 15, 2013 Accepted: Dec. 11, 2013
Corresponding Author: Hwang-Seok Oh(Korea Polytechnic Univ.)
E-mail: hsoh@kpu.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

Ⓒ The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.otg/licenses/by-nc/3.0), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서 론

2009년 이후 스마트 휴대 단말기가 보급되기 시작한 이후, 2012년 6월 기준 국내 스마트 단말기의 보급률이 67.6%로 세계에서 가장 높은 비율을 보이고 있다[1]. 스마트 휴대 단말기 보급의 확산에 가장 큰 기여를 한 부분은 사용자들에게 다양한 응용 프로그램을 제공할 수 있는 앱 스토어(App. store)의 활성화를 들 수 있다. 특히 다양한 분야의 앱들 중 게임 분야의 성장세가 매우 두드러지게 나타나고 있으며, 2012년 세계 모바일 게임 시장의 규모는 약 8,757억달러 규모이며 매년 10%의 성장률을 보이고 있다[2].

최근 스마트 휴대 단말기의 프로세서 속도나 메모리 용량 등 하드웨어적인 측면은 크게 발전하였지만 단말기에서 동작하는 콘텐츠의 복잡성 및 규모는 단말기 발전 속도보다 훨씬 빠른 속도로 진화하고 있다. 모바일 게임의 경우 초기의 2D 게임으로부터 현재는 3D 게임이 개발되어 서비스 되고 있다[3].

모바일 휴대 단말기에서 게임의 복잡도 및 규모가 커짐에 따라 게임에 사용되는 리소스의 용량도 매우 크게 증가하고 있으며, 이로 인해 단말기의 프로세서 처리능력 및 가용 메모리가 더욱 많이 요구되고 있다. 특히 많은 리소스 중 배경으로 사용되는 이미지는 동적인 게임을 구성하기 위해 크기와 복잡성이 높아지고 있으나 단말기 하드웨어 사양 제한으로 인해 사용가능한 크기가 제한된다. 예를 들어 갤럭시 탭 7.0의 경우 단순히 하나의 이미지만 단말기에 올려 시험할 경우 3072x3072 크기의 이미지는 디코딩 되나 4096x3072 크기의 이미지는 사용할 수 없다.

대규모의 이미지를 리소스로 사용할 경우 크게 두 가지 문제가 발생한다. 첫째, 리소스들은 압축한 형태로 콘텐츠에 포함되어 배포되기 때문에 게임에 사용되기 위해 디코딩된 후 메모리에 적재되어야 하는데 이미지의 크기가 클 경우 메모리 사용량이 매우 크게 된다. 이로 인해 사용 가능한 이미지의 크기가 제한된다. 이미지 리소스를 사용하기 위해 필요한 메모리는 압축 스트림 저장 공간, 디코딩된 이미지 저장 공간, 그리고 디코딩 시에 필요한 임시 저장 공간이다. 이들 중 디코딩된 이미지를 보관하기 위한 메모리 용량이 다른 것에 비하여 매우 크기 때문에 게임에서 큰 크기의 이미지 사용 제한을 받게 된다. 둘째, 이미지의 크기가 클 경우, 실행 시에 이미지를 디코딩하여 사용하면 처리 속도가 느려 게임 진행을 방해할 수 있다. 대부분의 게임에서는 리소스를 게임 시작 전에 미리 메모리에 디코딩하여 적재하는 기법[3]을 사용하고 있다. 이로 인해 초기 로딩 지연 시간이 존재한다.

본 논문에서는 단말기에 적정량의 가용 메모리가 있을 경우 모바일 게임에서 큰 이미지를 사용할 수 있는 이미지 압축 형식을 제안한다. 제안한 기법은 큰 이미지를 타일화하여 타일별로 압축 스트림을 만들어 하나의 파일에 저장한다. 제안한 기법은 게임 실행 시 화면 출력에 필요한 영역만 디코딩하여 사용하기 때문에 실행 시 디코딩 및 출력이 가능하므로 게임의 초기 로딩 시간을 줄일 수 있고, 메모리 용량 제한 문제를 해결할 수 있다.

본 논문의 구성은 다음과 같다. 2절에서 모바일 단말기에서 큰 이미지를 메모리 제한 없이 사용가능하도록 하는 타일드 이미지 인코딩 기법에 대하여 기술하고, 3절에서는 실험을 통하여 압축률, 메모리 사용 측면, 실행 시 디코딩 시간 측면에서 제안된 기법의 효율성을 보인다. 그리고 4절에서 결론을 내린다.

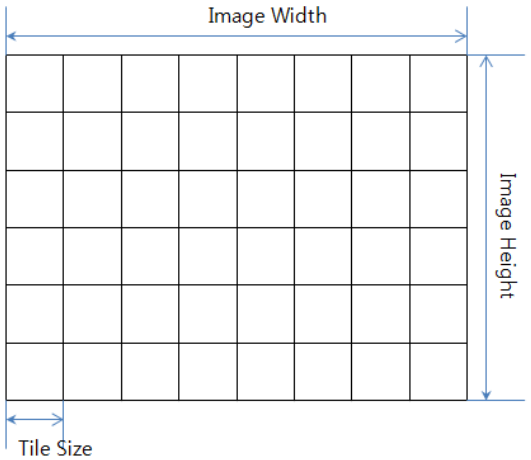
2. 타일드 이미지 인코딩

타일드 이미지 인코딩 기법은 모바일 단말기의 메모리 용량 제한으로 인해 모바일 게임에서 배경으로 사용하는 2D 이미지의 크기 제한 문제로 인해 대규모, 복잡한 이미지를 사용하지 못하고, 게

임 시작 시 로딩 시간이 많이 소요되는 문제점을 해결하기 위해 화면에 출력될 최소의 이미지 부분만 실행 시 디코딩하여 사용할 수 있도록 지원하는 이미지 인코딩 및 디코딩하는 방법이다.

타일드 이미지는 내부적으로 사용하는 압축 기법은 현재 많이 사용하고 있는 압축 기법, 즉 PNG나 GIF, 손실이 발생하여도 무방한 경우 JPEG 포맷을 사용한다. 본 논문에서는 비손실 압축 기법으로 많이 사용되는 PNG(Portable Network Graphics) 포맷[4]을 대상으로 기술한다.

타일드 이미지 압축 기법은 먼저 매우 큰 이미지를 입력받아 [Fig. 1]과 같이 먼저 정의된 블록의 크기(타일 크기)로 겹치지 않게 나누고, 각 블록 단위로 스캔 라인 순서대로 기존 PNG 압축 기법을 적용하여 압축한다.



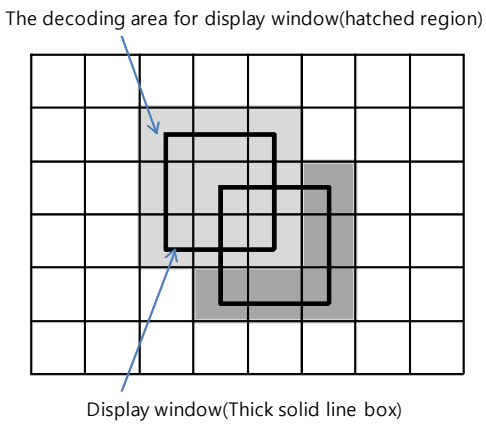
[Fig. 1] Image partitioning for tiled image compression

각 블록별 압축된 압축 스트림은 [Fig. 2]와 같이 타일화된 이미지 크기 및 타일의 크기 정보와 각 블록이 압축되어 스트림화되어 순서대로 파일에 저장되어 있을 경우 각 블록들의 스트림에 대한 인덱스 정보들, 그리고 블록별 압축 스트림들이 차례로 저장된다.

위와 같이 타일별로 분할 압축되어 하나의 파일 스트림으로 저장된 이미지 파일은 모바일 게임에 하나의 리소스로 사용된다. 게임에서 제안된 구조의 파일을 사용할 경우 추가적인 이미지 디코더 라이브러리가 필요하지 않으며, 단지 파일 포맷을 재해석하여 블록 단위로 동일한 라이브러리를 이용해서 디코딩할 수 있다. 타일화된 이미지는 [Fig. 3]과 같은 방법으로 디코딩되어 게임에 사용된다.

ImageWidth ImageHeight
TileSize
TiledImage_1_StreamIndex
...
TiledImage_N_StreamIndex
TiledImage_1_Stream
...
TiledImage_1_Stream

[Fig. 2] File format for tiled image compressed stream



[Fig. 3] Decoding process for a display window within tiled image

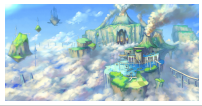
[Fig. 3]에서 화면에 출력될 이미지 영역을 굵은 실선으로 표시하였으며, 실행 시에 출력하기 위한 이미지 데이터를 요청할 경우, 출력 영역 범위에 걸쳐 있는 타일 이미지 블록들을 디코딩하여 전달한다. 디코딩된 이미지를 저장하기 위한 메모리 공간은 디코딩 범위에 해당하는 이미지 크기를 저장

할 수 있는 크기만 필요하다. 디코딩 시 출력할 영역의 시작 위치와 크기 정보를 전달 받아 타일드 이미지 포맷의 이미지 크기와 타일 크기 정보를 이용해서 디코딩 될 블록을 계산하고, 타일 이미지 스트림의 위치를 이용하여 해당 블록 압축 스트림을 디코딩하여 메모리에 저장한다. 디코딩 영역에 포함된 모든 블록을 디코딩 한 후, 화면에 출력될 이미지 부분을 넘겨준다. 게임에서 배경으로 사용되는 이미지는 점진적으로 스크롤링 되면서 화면에 출력된다. 이 경우, 제안한 압축 스트림 양식을 이용할 경우 디코딩 범위에 새로 추가되는 부분만 디코딩하여 추가하고, 범위를 벗어나는 영역에 대한 이미지는 버림으로써 큰 이미지를 사용하더라도 메모리 요구량이 크게 증가하지 않는다. [Fig. 3]에서 화면에 출력될 위치가 굵은 실선 왼쪽 위에서 오른쪽 아래로 이동될 경우 오른쪽 아래의 출력 화면에 걸쳐 있는 블록들 중 새롭게 추가되는 블록들만 디코딩하여 디코딩 된 데이터를 저장하는 메모리에 보관하면 되기 때문에 추가적인 디코딩 시간이 많이 소요되지 않는다. 출력 화면이 현재 디코딩되어 있는 범위 내에서 이동할 경우 추가적인 디코딩 없이 디코딩된 이미지를 저장하고 있는 메모리에서 요구되는 이미지를 가져와 사용할 수 있다.

배경 이미지의 크기는 ImageWidth * ImageHeight, 타일의 크기를 TileSize로, 그리고 디스플레이 될 화면의 크기를 DispWidth * DispHeight라고 할 경우 [Fig. 3]에서와 같이 디스플레이에 필요한 디코딩 범위 영역의 이미지만 디코딩하여 메모리에 가지고 있으면 된다. 이 경우 이미지를 화면에 출력하기 위해 최대로 필요한 메모리 요구량은 *{DispWidth + 2 * (TileSize - 1)} * {DispHeight + 2 * (TileSize - 1)} * 화소당 Byte수* 이다. 이는 최종 디코딩 된 이미지를 내부적으로 보관하기 위한 메모리양이며, 디코딩 되는 과정에 필요한 메모리양 즉 압축 스트림 전체를 읽은 파일을 저장하기 위한 부분과 디코딩 과정에 필요한 동적 메모리양은 하나의 PNG 파일로 저장하는 방법과 유사하기 때문에 메모리 요구량 비교 계산에서 제외한다.

3. 실험 및 결과 분석

제안된 타일드 이미지 압축 기법의 성능을 분석하기 위하여 [Fig. 4]와 같은 PNG 타입의 크기가 다른 3개의 이미지를 사용하였다. 제안한 기법의 성능을 평가하기 위해 출력할 화면의 크기를 480x800로 가정하고 타일 크기를 32x32, 64x64, 128x128, 256x256, 512x512로 설정하였을 때 압축률, 최대 메모리 요구량, 디코딩 속도 면에서 실험을 통한 결과를 [Table 1], [Table 2], [Table 3]에 각각 제시하였다. 응용 프로그램에서 실행 시 디코딩 속도를 측정하기 위하여 삼성 갤럭시 탭 7.0 모델(A8 1GHz processor, 512MB DDR SDRAM)을 사용하였다.

Images	Image Thumbnail	Image Size
Test Image 1		2048x1024
Test Image 2		1536x2048
Test Image 3		3072x2048

[Fig. 4] Test images for experiments

실험에 사용된 데이터는 게임 배경 화면으로 사용되는 대용량 크기의 이미지로 출력되는 화면보다 커 이동 방향에 따라 스크롤링 되면서 화면에 출력된다. [Table 1]에는 실험에 사용된 이미지를 기존의 하나의 PNG 파일로 압축하는 기법과 제안된 기법으로 압축하였을 경우 압축 스트림의 크기를

KB(Kilo Byte) 단위로 표시하였다. 기본적으로 제안한 방법이나 기존 PNG 압축 기법 모두 PNG 압축 기법을 사용하기 때문에 압축 스트림의 크기는 크게 차이나지 않는다. 블록의 크기가 32x32와 같이 블록의 크기가 작을 경우에는 PNG 압축 대상 블록 수가 많아짐에 따라 헤드 정보와 청크 태크 정보가 많이 포함되고 블록간의 중복성이 많으나 이를 잘 활용하지 못하는 특성으로 인해 하나의 파일로 압축하는 기법에 비해 압축 스트림의 크기가 크게 된다. 그러나 블록의 크기가 128x128이나 256x256 정도의 크기일 경우 하나의 PNG 파일로 압축하는 것보다 오히려 압축 효율이 좋게 나타나기도 하는데 이는 이미지들의 지역적 특성(각 블록으로 나누어진 이미지 블록들의 화소 값들 변화량이 전체 이미지의 화소 값들의 변화량보다 작음)이 반영되었기 때문이다.

[Table 1] Amount of compressed image data (Unit : KB)

Compression Format	Test Image 1	Test Image 2	Test Image 3
Single PNG Format	2,011	2,478	6,139
Tiled PNG Format with 32x32 tile size	2,343	2,974	7,043
Tiled PNG Format with 64x64 tile size	2,096	2,620	6,256
Tiled PNG Format with 128x128 tile size	2,018	2,510	6,007
Tiled PNG Format with 256x256 tile size	1,985	2,467	5,908
Tiled PNG Format with 512x512 tile size	1,983	2,460	5,912

[Table 2] Required memory amount for decoded image data(Unit : KB)

Compression Format	Test Image 1	Test Image 2	Test Image 3
Single PNG Format	8,388	12,582	25,165
Tiled PNG Format with 32x32 tile size	1,868	1,868	1,868
Tiled PNG Format with 64x64 tile size	2,244	2,244	2,244
Tiled PNG Format with 128x128 tile size	3,006	3,094	3,094
Tiled PNG Format with 256x256 tile size	4,055	5,187	5,187
Tiled PNG Format with 512x512 tile size	6,152	10,946	10,946

[Table 2]에는 한 출력 화면의 크기를 요청하였을 경우 디코딩에 필요한 메모리 요구량을 제시하였다. 전체 이미지를 디코딩하여 메모리에 보관할 경우 메모리 용량의 제한으로 리소스로 활용할 수 있는 이미지 개수와 이미지 크기가 제한된다. 본 논문에서 제시한 기법은 큰 이미지에서 출력될 부분을 지정할 경우 출력 화면에 속하는 블록들을 실행 시 디코딩하여 출력될 이미지를 전달해 줌으로 실행 시 사용되는 메모리양을 크게 줄일 수 있다. 실험 결과에서 보는 바와 같이 제안된 기법은 이미지를 블록으로 나누는 크기에 따라 디코딩 시 결과를 저장하는 최대 메모리 요구량이 이미지의 크기와 관계없이 일정하게 유지되나 하나의 PNG로 압축된 경우 전체 디코딩된 이미지를 보관해야 하기 때문에 이미지 크기에 종속된다. 즉 기존의 PNG 기법은 모바일 단말기에 사용할 경우 이미지

의 크기가 제한되는 반면 제안한 기법은 이미지의 크기에 제한을 받지 않는다. 또한 [Fig. 3]에서와 본 바와 같이 제시한 기법은 화면에 출력할 이미지를 생성하기 위해 디코딩해야 할 영역이 매우 적기 때문에 실행 시 디코딩해야 할 블록을 메모리에서 압축 스트림을 읽어 디코딩하더라도 게임 진행에 크게 문제가 되지 않는다. [Table 3]에는 갤럭시 탭 7.0에서 각 실험 영상을 디코딩하여 화면에 출력하는 디코딩 속도를 측정하여 제시하였다. 프로세서의 속도가 매우 빨라 10만번 반복한 측정 결과를 msec 단위로 제시하였다. 타일화된 이미지는 스크롤링을 가정하여 일부만 디코딩한 것이 아니고 출력 화면에 걸쳐 있는 영역 전체를 디코딩할 때의 소요 시간을 제시하였다. 일반적으로 PNG 이미지의 디코딩 속도는 디코딩할 이미지의 크기에 비례하는 시간이 소요되며, 제안된 방법에서 각 실험 영상에 따라 디코딩 속도가 조금씩 차이는 있으나 블록 크기가 작은 경우가 블록 크기가 큰 경우에 비해 디코딩 속도가 빠름을 알 수 있다. 이는 [Table 3]에 제시한 바와 같이 디코딩될 영역의 크기가 작을수록 디코딩 속도가 빠름을 보여준다. 실제 게임에서 큰 이미지가 배경 이미지로 사용될 경우 게임이 진행됨에 따라 배경이 스크롤링된다. 이 때 새롭게 출력 화면에 추가되는 이미지 부분과 출력 화면에서 사라지는 이미지 부분이 있을 경우, 추가되는 부분만 디코딩할 경우는 디코딩 소요 시간은 [Table 3]에 제시한 소요 시간보다 훨씬 작게 요구된다. 즉, 제안된 기법을 게임 실행 시 실시간 디코딩을 통하여 게임에 활용할 수 있음을 알 수 있다.

실험을 통하여 제안한 기법을 모바일 게임에 적용할 경우 하나의 PNG 파일로 압축된 이미지를 사용하는 경우의 크기 제한 문제와 프로그램 시작 시 초기 로딩 시간이 많이 걸리는 문제를 해결할 수 있음을 보였다.

[Table 3] Processing time for decoding at Galaxy Tap 7.0 (Unit:msec)

Compression Format	Test Image 1	Test Image 2	Test Image 3
Single PNG Format	1,482	2,973	4,127
Tiled PNG Format with 32x32 tile size	380	471	336
Tiled PNG Format with 64x64 tile size	412	550	388
Tiled PNG Format with 128x128 tile size	531	741	517
Tiled PNG Format with 256x256 tile size	716	1225	850
Tiled PNG Format with 512x512 tile size	1,086	2586	1795

4. 결 론

본 논문에서는 모바일 기기의 프로세서 처리능력과 메모리 용량의 제한으로 인한 거대 크기의 이미지를 게임의 배경으로 사용할 수 없는 문제를 해결하기 위해 타일화된 이미지 압축 기법을 제안하였다. 제안한 기법은 압축 효율 측면에서는 하나의 PNG 파일로 압축하는 방법에 비하여 유사하나 메모리 사용을 줄여 모바일 단말기의 메모리 제한으로 인한 디코딩 가능한 이미지의 크기 제한 문제를 해결하였다. 또한 런타임 부분 디코딩 기법을 적용하여 단말기 화면에 출력되는 이미지 부분만 디코딩하여 출력함으로 초기의 대규모 이미지의 로딩 시간을 줄일 수 있음을 보였다.

본 논문에서는 큰 이미지를 타일화하여 각 블록

별로 기존 압축 기법 PNG를 적용하여 압축 스트림을 생성하였으나 향후 압축 스트림의 크기를 줄이기 위해서는 게임 그래픽 데이터를 효과적으로 압축할 수 있는 방법에 대한 연구가 추가적으로 필요하다. 특히, 타일의 크기에 따라 인접한 타일 간의 유사성 정보를 이용할 경우 압축 효율을 높일 수 있을 것이다.

REFERENCES

[1] Jong-Il Park, "Worldwide Smartphone Penetration in South Korea. The Challenge for the Future?", ConnectingLab, http://ww w.connectinglab.net/wordpress/?p=6411.

[2] KOCCA, "World Game Market Size and Forecast (2008-2017)", KOCCA Global Information Statistics Report, http://www.kocca.kr/knowledge/abroad/ stat/ index.html, 2013.

[3] Dong-Youn Hwang, "Making and Learning Android Game Programming", Hanbit Media, 2011.

[4] IOS/IEC 15948:2003(E) International Standard, "Protable Newtork Graphics(PNG) Specification (Second Edition)", ISO/IEC, 2003.

[5] WonJun Kim, SangHyeok Jeong, "Image Loading and Its Library for Android", LiveMeida Soft, http://livemedia-soft.com/15847

[6] SangHyeong Kim, "Android Programming Complete Guide(New Released)", Hanbit Media, pp. 256-261, 2012.

[7] David Salomon, "Data Compression The Complete Reference", 3rd Edition, Springer, 2004.

[8] Jeong-Hoon Kim, "A study for efficient image use in the mobile contents development", Journal of Korea Game Society, Vol.5, No.1, pp. 53-60, 2005.

[9] Young-jae Chun, Hae-Dong Kim, Sung-Hyun Cho, "Lossless Compression and Rendering of Multiple Layer Displacement Map", Journal of Korea Game Society, vol. 9, no. 6, pp. 171-178, 2009.

[10] Eun-sun Jeon, Bo-koan Kim, "New Image Compression Technique for Portable Device Display", M.S. Thesis, ChungNam National Univ., 2009.

[11] Mark Nelson, "The Data Compression Book", 2nd ed., M&T Books, 1995.

[12] ISO/IES 14495-1 International Standard, "Information technology - Lossless and near-lossless compression of continuous-tone still images: Baseline", ISO/IEC, 1999.

오 황 석 (Hwang-Seok Oh)

1999-2001 한국전자통신연구원 가상현실연구부 선임연구원
2001-2004 (주)디지탈아리아 부설연구소 수석연구원
2004-현재 한국산업기술대학교 게임공학부 부교수

관심분야 : 모바일 게임, 모바일 멀티미디어 소프트웨어

— 모바일 단말기에서 이미지 처리에 필요한 메모리 사용량을 줄이기 위한 타일화 이미지 압축 기법 —