

# 사진트리 구조를 이용한 포토 모자이크의 GPU구현

양진석<sup>0</sup>                      주충재                      오경수

승실대학교 미디어학과

{ispio, goodbye302, oks}@ssu.ac.kr

## Photo Mosaics using Quad-tree structure on GPU

Jinsuk Yang<sup>0</sup>                      Choongjae Joo                      Kyoungsu Oh

Dept. of Media, Soongsil University

### 요 약

포토 모자이크는 입력 이미지를 작은 타일 이미지들을 조합하여 모자이크로 나타내는 방법이다. 일반적으로 포토 모자이크 방법은 균일한 사각형 모양의 타일을 이용한다. 그렇기 때문에 이미지 특성에 따른 모자이크 표현 방법에 제한이 있다. 본 논문에서는 다양한 크기의 타일을 생성하기 위해 사진트리(Quad-tree) 구조를 이용한 포토 모자이크 방법을 제안한다. 먼저 그래픽 하드웨어의 밍맵(mipmap)을 이용해 입력 이미지의 레벨별 색 분산을 구한다. 분산의 크기에 따라 입력 이미지는 사진트리 구조를 가지는 다양한 크기의 타일로 나뉜다. 각각의 타일은 후보 이미지들 중 가장 유사한 이미지 찾아 대체한다. 결과적으로 우리가 제안하는 방법은 입력 이미지의 색에 따라 다양한 크기의 타일로 나뉘므로써 또 다른 모자이크 효과를 제공한다. 또한 모든 과정은 그래픽 하드웨어를 통해 계산됨으로 빠른 수행속도를 갖는다.

### Abstract

Photomosaic is a method of representation an input image in forms of mosaics by a set of small tile images. Generally, equal rectangular tiles are used in photo mosaics. Therefore, there are limitations to expression of mosaic according to characteristic of images. In this paper, photo mosaics using quad-tree structure is proposed to create tiles in varies sizes. Initially, color variance per each level of an input image is computed using the mipmap of graphic hardware. Depending on the value of variance, the input image is divided into tiles in varies sizes. Each tile finds the most similar reference image and replaces with it. As a result, the method provides another pictorial effectiveness by dividing the input image into tiles in varies sizes depending on color held by the input image. In addition, whole processing is done on the graphic hardware and thereby we achieve faster performance.

키워드: 포토 모자이크, 사진트리, 밍맵, 실시간 렌더링

**Keywords:** Photo Mosaics, Quad-Tree, Mip-Map, Real-time Rendering

# 1. 서론

포토 모자이크는 입력 이미지를 작은 타일 이미지들을 조합하여 모자이크로 나타내는 방법이다. 모자이크는 고대에서부터 회화효과나 장식적 가치 때문에 건축의 바닥이나 벽면, 공예품 등을 장식하는데 많이 사용되어왔으며 과학자들은 모자이크 표현방법에 대해 많은 관심과 연구를 진행하였다. 컴퓨터 그래픽스에서는 타일들의 크기와 위치, 모양에 따라 그림 1에서처럼 다양한 형태의 모자이크를 생성하는 방법이 많이 연구되었다([1]~[4]). 그 중에서 Silver et al.[1]는 입력 이미지를 균일한 사각형의 타일로 나누는 후 이미지 데이터베이스에서 가장 유사한 이미지를 찾아 각 타일을 대체했다. Hausner[2]와 Dobashi et al.[3]는 입력 이미지의 외각선을 따라 타일을 배치시키고 타일들 간에 적당한 모양을 갖게 하기 위해 보로노이 다이어그램(Voronoi Diagram)방법을 사용했다. 하지만 위의 연구들은 이미지의 특성과는 관계없이 모두 균일한 크기의 타일을 사용한다. 따라서 이미지의 외각선이나 색에 따른 모자이크 표현 방법에 제한이 있다[5]. Gianpiero et al.[4]는 이미지 특성에 따른 모자이크 효과를 주기 위해 입력 이미지를 사진트리 구조로 타일을 나누었다. 색의 복잡도가 낮은 영역은 큰 타일로 나누고 색의 복잡도가 높은 영역은 작은 타일로 나눔으로써 또 다른 모자이크 효과를 만들었다. 하지만 타일을 나누는 과정이 모두 CPU에서 처리되기 때문에 실행속도가 느린 단점이 있다.

본 논문에서는 Gianpiero et al.[4]의 방법처럼 입력 이미지의 특성에 따라 다양한 크기의 타일로 나누기 위해 사진트

리 구조를 이용한다. 하지만 우리는 모든 계산과정을 그래픽 하드웨어를 통해 처리함으로써 빠른 수행속도를 갖게 한다.

알고리즘은 다음과 같이 3단계 과정으로 나누어진다.

- 1) 입력 이미지의 각 영역별 분산을 이용해 사진트리 구조를 생성한다.
- 2) 사진트리 구조를 이용해 입력 이미지를 여러 크기의 타일로 나눈다.
- 3) 나누어진 타일은 이미지 데이터베이스에 저장된 후보 이미지들 중에서 가장 유사한 이미지를 찾아 대체한다.

이 논문의 2장에서는 주요 알고리즘들을 좀 더 자세히 기술하고 3장에서는 실험결과를, 마지막으로 4장에서는 결론 및 향후과제에 대해 기술한다.

## 2. 사진트리 구조를 이용한 포토 모자이크

사진트리[6], [7]는 각 노드마다 4개의 자식 노드를 가지는 트리 구조이다. 사진트리는 사각형 모양의 2차원 공간을 4등분 한 후, 필요에 따라 나누어진 사각형을 다시 4등분한다. 이를 이용하면 우리는 정해진 기준에 따라 필요한 만큼 사각형을 나눌 수 있고, 나누어진 사각형 크기에 따라 포토 모자이크에 사용할 타일 크기를 정할 수 있다. 예를 들면 사진트리의 하위노드 영역은 작은 크기의 타일을 갖고, 상위노드 영역은 큰 크기의 타일을 갖는다.

이번 장에서는 다양한 크기로 타일을 나누기 위해 사진트리 구조를 이용한 포토 모자이크의 GPU 구현 방법을 설명한다. 먼저 2.1절에서는 입력 이미지를 밍맵화하여 사진트리 구조로 만든 후, 각 레벨별 색 분산을 계산한다. 2.2절에서는 2.1절에서 계산된 분산을 레벨별로 탐색해 가며, 입력 이미지를 사진트리 구조의 여러 크기 타일로 나눈다. 마지막으로 2.3절에서는 이미지 데이터베이스에 저장된 후보 이미지 중 해당 타일과 가장 유사한 이미지를 찾아 타일을 대체하는 과정을 설명한다. 그림 2는 우리가 제안한 방법의 개요를 나타낸다.

### 2.1 압력 이미지의 색 분산 계산

William et al.[8]은 그림자 맵(depth map)에 저장된 깊이와 셰이딩 할 지점의 깊이에 대한 분산을 계산하기 위해 그림자 맵의 한 채널에는 광원으로부터 장면까지의 최단거리를, 다른 한 채널에는 최단거리의 제곱을 저장했다. 분산은 식 (1)과 같이 주어진 필터 크기 안에서 최단거리 제곱의 평균

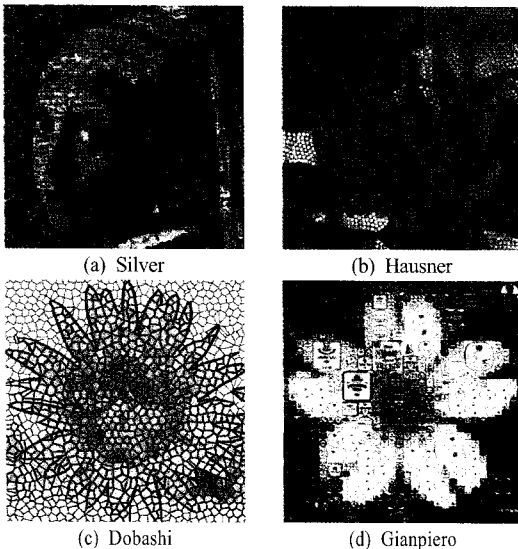


그림 1. 기존의 방법을 이용한 포토 모자이크의 예

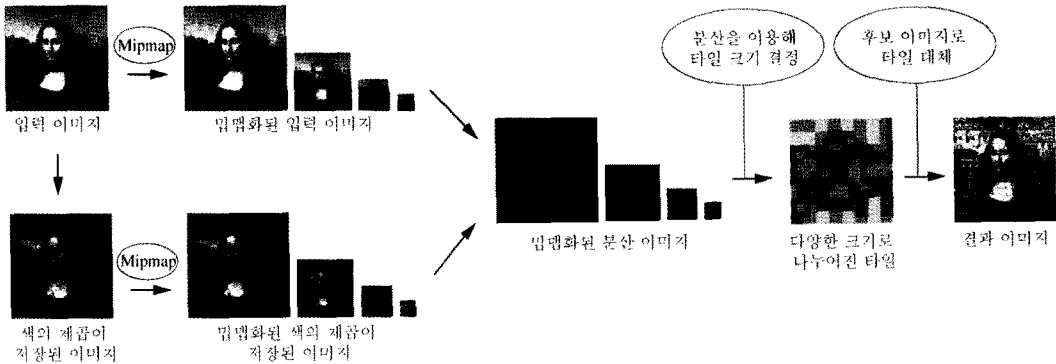


그림 2. 본 논문의 알고리즘 개요

$(E(x^2))$ 과 최단거리 평균의 제곱( $E(x)^2$ )의 차에 의해 계산된다.

$$V(x) = E(x^2) - E(x)^2 \quad (\text{식 1})$$

본 논문에서는 입력 이미지의 분산을 이용하여 사진트리의 영역을 나눈다. 우리는 분산을 계산하기 위해 [8]의 방법을 이용했다. 먼저 입력 이미지를 mipmapping하여 사진트리 구조로 만든다. mipmapping된 이미지는  $2^n \times 2^n$ 개의 텍셀을 가진 0번째 레벨부터  $2^0 \times 2^0$ 개의 텍셀을 가진 n번째 레벨까지의 이미지 집합이다. 그림 3과 같이 각 레벨의 텍셀(i, j)에는  $(2i, 2j)$ ,  $(2i, 2j+1)$ ,  $(2i+1, 2j)$ ,  $(2i+1, 2j+1)$ 와 같은 4개 자식노드를 갖는다. 또한 상위 레벨의 이미지 텍셀에는 4개 자식노드의 색 평균이 저장된다. 이 과정은 그래픽 하드웨어의 mipmapping을 통해 쉽고 빠르게 계산된다. 마찬가지로 색의 제공이 저장된 이미지를 만들어 입력 이미지처럼 mipmapping한다.

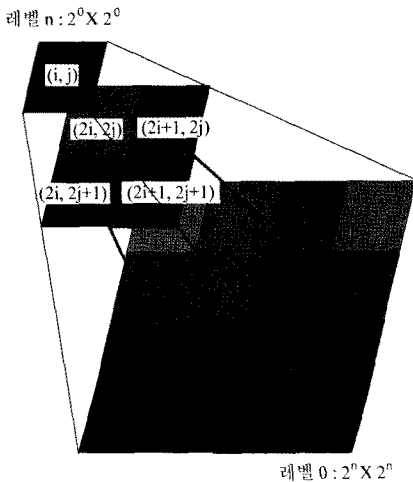


그림 3. mipmapping된 이미지의 예

mipmapping된 입력 이미지와 mipmapping된 색의 제공이 저장된 이미지는 식 (1)을 이용해 레벨별 분산을 계산하는데 이용된다.

## 2.2 분산을 이용한 타일 크기 결정

이번 절에서는 분산을 이용하여 사진트리의 영역을 나누는 방법에 대해 설명한다. 먼저 레벨별로 계산된 분산을 최상위 레벨(n번째 레벨)부터 최하위 레벨(0번째 레벨)까지 하향식(Top-Down)으로 탐색한다. 그림 4와 같이 현재 레벨에 해당하는 분산이 사용자가 정한 기준값(threshold)보다 크면 다음 하위 레벨로 탐색하고, 작으면 탐색을 멈춘다. 탐색을 멈춘 지점의 레벨을 L이라고 했을 때 타일 크기는  $2^{n-L} \times 2^{n-L}$ 이다. 실제적으로 우리는 입력 이미지가 너무 작거나 큰 타일로 나누어지는 것을 막기 위해 타일 크기를 최소

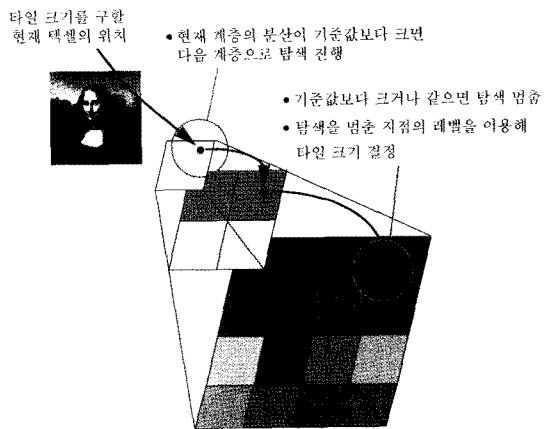


그림 4. 레벨별 분산을 이용한 탐색 방법

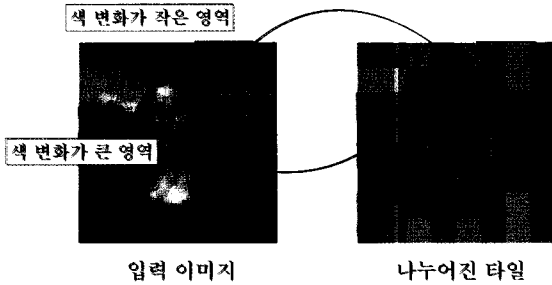


그림 5. 나누어진 타일의 결과. 색 변화가 작은 영역(분산이 작음)은 큰 크기의 타일로 나누어졌고, 색 변화가 큰 영역(분산이 큼)은 작은 크기의 타일로 나누어졌다. 나누어진 타일 결과에서 색이 진할수록 작은 크기의 타일이다.

$2^2 \times 2^2$ 부터 최대  $2^6 \times 2^6$ 으로 고정하였다.(입력 이미지의 크기가  $2^9 \times 2^9$ 일 경우)

그림 5는 분산에 따라 나누어진 타일 크기를 나타냈다. 입력 이미지의 색 변화가 큰(분산이 큼) 영역은 큰 크기의 타일을 가지며 색 변화가 작은(분산이 작음) 영역은 작은 크기의 타일을 가진다.

### 2.3 타일을 대체할 후보 이미지 선정

입력 이미지에 근접한 포토 모자이크를 얻기 위해서는 타일을 대체할 후보 이미지의 선정이 매우 중요하다. Gianpiero et al.[4]는 이미지 데이터베이스에서 빠른 속도로 입력 이미지의 대상 타일과 가장 유사한 이미지를 찾기 위해 Antipole 트리구조[9]를 이용했다. 하지만 이 방법은 이미지 데이터베이스를 전처리 과정을 통해 Antipole 트리구조로 만들어야 한다. 또한 이미지와 같은 다차원 데이터의 정확한 검색은 어렵기 때문에 입력 이미지와 유사도가 낮은 단점이 있다.

본 논문에서는 가장 유사한 이미지를 찾기 위해 전역 검색(exhaustive search) 방법을 이용했다. 또한 힙맵을 통해 색 평균을 쉽고 빠르게 구할 수 있기 때문에 Adam et al.[10]의

방법처럼 색 평균을 이용하여 가장 유사한 후보 이미지를 선정한다. 이를 위해 먼저 후보 이미지들을 그림 6과 같이 타일화하여 여러 장의 후보 이미지 데이터베이스로 만든다. 그런 다음 힙맵을 이용해 후보 이미지의 색 평균을 구한다. 가장 유사한 후보 이미지 선정은 해당 타일에 위치한 입력 이미지 색 평균과 이미지 데이터베이스에 있는 모든 후보 이미지의 색 평균을 비교하여 이루어진다. 최종적으로 색 평균이 가장 비슷한 후보 이미지를 선정하여 해당 타일을 대체한다.

### 3. 실험결과

본 논문의 구현에 있어 그래픽 API는 DirectX 9.0, 하드웨어는 nvidia Geforce GTX 480 그래픽 카드와 Intel CPU 3.0GHz를 사용하였다. 그리고 무작위로 256개의 이미지를 선정하여 후보 이미지로 사용했다.

그림 7은 하나의 입력 이미지에 대해 서로 다른 사용자의 기준값으로 분산 탐색을 하여 타일을 나눈 실험결과이다. 기준값이 클수록 상위 레벨에서 탐색이 멈추는 경우가 많기 때문에 큰 타일이 많이 생성되는 것을 볼 수 있다. 그림 8은 여러 가지 입력 이미지에 대한 실험결과이다. 색의 변화가 큰 영역은 작은 크기의 타일로 이루어져있고, 색의 변화가 작은 영역은 큰 크기의 타일로 이루어진 것을 확인할 수 있다. 하지만 색 평균이 가장 유사한 후보 이미지로 타일을 대체하기 때문에 색 변화가 큰 후보 이미지를 사용했을 때에는 입력 이미지와 유사성이 떨어지는 경우가 발생한다.

표 1은 Gianpiero et al.[4]의 방법처럼 CPU를 통해 타일을 나눈 결과와 본 논문에서 제안한 GPU를 통해 타일을 나눈 실험결과의 성능(performance)을 비교한 것이다. CPU에 비해 GPU로 구현한 결과가 약 14배 빠른 것을 알 수 있다.

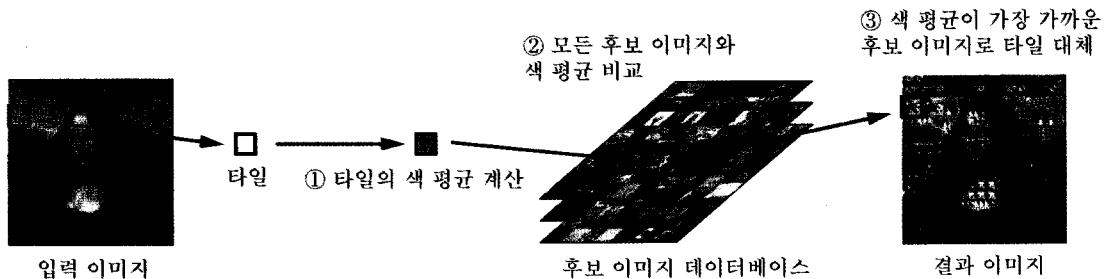


그림 6. 타일을 대체할 후보 이미지 선정

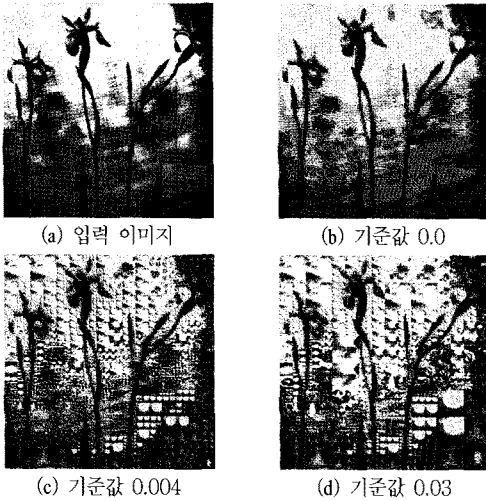


그림 7. 사용자의 기준값 변화에 따른 타일 크기 변화. (b)에서 (d)로 갈수록 기준값의 크기가 커지며, 이에 따라 타일의 크기도 커진다.

표 1. CPU와 GPU 구현에 따른 성능 비교

구현방식	CPU	GPU
프레임율	18~19	257~258

#### 4. 결론 및 향후과제

본 논문은 사진트리를 이용한 포토 모자이크의 GPU 구현 방법을 설명하고 있다. 논문에서 제안한 방법은 입력 이미지의 색 분산에 따라 다른 크기의 타일을 가지도록 하였으며 모든 과정이 그래픽 하드웨어를 통해 계산되므로 빠른 수행속도를 갖는다. 또한 힙맵을 통해 사진트리 구조를 만들기 때문에 복잡한 자료구조를 필요로 하지 않아 쉽게 구현할 수 있다.

하지만 본 논문에서 제안한 방법은 인접한 타일들 간에 중복되는 후보 이미지 문제에 대해서는 고려하지 않았다. 따라서 타일들 간에 중복되는 이미지들이 나타나는 현상이 야기된다. 또한 후보 이미지의 개수가 많아질 경우에는 전역 검색을 통한 후보 이미지 선정은 매우 느리다. 후보 이미지 개수가 많아질 경우에는 다른 유사 이미지 검색 방법 등이 추가적으로 연구되어야 할 것이다.

#### 감사의 글

본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2010년도 콘텐츠 산업기술지원사업의 연구결과로 수행되었음

#### 참고 문헌

- [1] R.Silvers and M. Hwaley, "Photomosaics", New York, NY, USA: Henry Holt and Co., Inc., 1997.
- [2] HAUSNER, A. "Simulating Decorative Mosaics", *In Proceedings of SIGGRAPH*, pp. 573-580, 2001.
- [3] DOBASHI J., HAGA T., JOHAN H., NISHITA T. "A Method for Creating Mosaic Images Using Voronoi Diagrams", *In Proc Eurographics*, pp. 341-348, 2002.
- [4] Gianpiero Di Blasi, Giovanni Gallo, Maria Pia Petralia, "Smart Ideas for Photomosaic Rendering", *In Proceeding of Eurographics Italian Chapter Conference*, pp. 267-272, 2006.
- [5] Dongwann Kang, Young-Sup Park, Sang-Hyun Seo, Kyung-Hyun Yoon, "Tow Layer Image Tile Mosaics", *In Proceedings of the Eurographics2006 short paper*, pp. 145-148, 2006.
- [6] FINKEL R.A., BENTLEY J.L., "Quad trees: A data structure for retrieval on composite keys", *Acta Informatica 4*, pp. 1-9, 1974.
- [7] John R. Smith, Shih-Fu Chang, "Quad-tree segmentation for texture-based image query", *MULTIMEDIA*, 1994.
- [8] William Donnelly, Andrew Lauritzen, "Variance shadow maps", *Proceedings of the symposium on Interactive 3D graphics and games*, pp. 161-165, 2006
- [9] D. Cantone, A. Ferro, A. Pulvirenti, D. R. Recupero, and D. Shasha, "Antipole tree indexing to support range search and k-nearest neighbor search in metric spaces," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 553-550, 2005.
- [10] Adam Finkelstein and Marisa Range. "Image Mosaics", *Lecture Notes in Computer Science*, Vol. 1375, pp. 11-22, 1998.

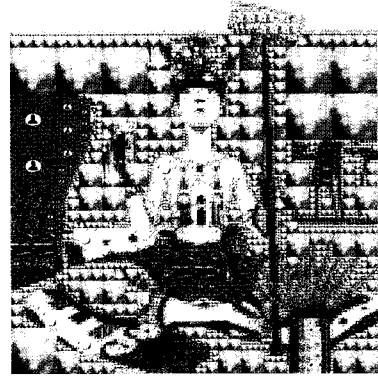
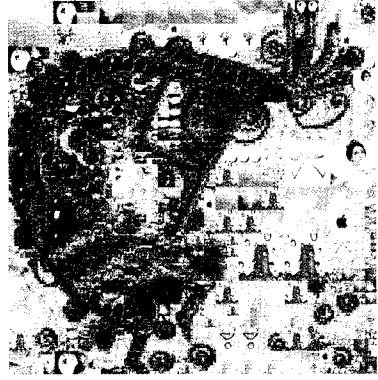


그림 7. 입력 이미지(좌)에 따른 포토 모자이크 결과(우)

## 〈저자 소개〉



양진석

- 2009년 송실대학교 미디어학부 학사
- 2009년~현재 송실대학교 미디어학과 석사통합과정
- 관심분야: 실시간 컴퓨터 그래픽스, 게임 프로그래밍



주충재

- 2010년 송실대학교 미디어학부 학사
- 2010년~현재 송실대학교 미디어학과 석사과정
- 관심분야: 실시간 컴퓨터 그래픽스, 게임 프로그래밍



오경수

- 1994년 서울대학교 계산통계학과 학사
- 1996년 서울대학교 전산학과 석사
- 2001년 서울대학교 전기컴퓨터공학부 박사
- 2001년~2002년 (주)조이멘트 개발팀장
- 2003년~현재 송실대학교 글로벌미디어학부 부교수
- 관심분야: 실시간 컴퓨터 그래픽스, 게임 프로그래밍