

특징값 기반 블록 매칭을 이용한 자동 포토 모자이크 알고리즘

서성진, 김기웅, 조현우, 이해연
금오공과대학교 컴퓨터소프트웨어공학과
e-mail: urloverme@hotmail.com

Automatic Photo Mosaic Algorithm using Feature-Based Block Matching

Sung-Jin Seo, Gi-Woong Kim, Hyun-Woo Jo, Hae-Yeoun Lee
Department of Computer Software Engineering
Kumoh National Institute of Technology

요 약

모자이크는 여러 개의 작은 영상들을 모아서 하나의 큰 영상을 만드는 것을 말한다. 본 논문에서는 모자이크 방법 중 하나인 사진을 이용하여 영상을 만드는 포토 모자이크 방법을 컴퓨터 알고리즘으로 구현하는 기술 제안을 한다. 이미지를 원하는 사이즈의 타일로 나눈 다음, 나뉜 타일을 16등분을 한다. 16등분된 이미지 각 요소들에 대하여 RGB 평균값을 계산하여 총 48개 특징값을 추출하여 데이터베이스에 저장해둔다. 그리고 타일과 비교가 될 이미지들은 이미 똑같은 작업을 통하여 데이터베이스에 저장되어 있다. 이렇게 저장된 값들을 통하여 유클리드 거리를 통하여 두 이미지의 유사도를 측정하게 된다. 최적의 값을 찾으면 바로 대입하는 것이 아니라 이전에 있던 타일 이미지의 명암값을 새로 삽입되는 타일에 부여를 하여, 부드러운 영상을 만들게 된다. 그리고 타일을 삽입할 때 이전에 사용된 이미지는 배열의 마지막으로 옮겨지게 되며 사용횟수를 체크하여 반복적 사용을 제한하였다.

키워드 : 포토 모자이크, 유클리드 거리, 블록 매칭, 명암값 교체

1. 서론

모자이크는 기본적인 밑그림에다가 작은 조각들을 집착제를 바른 곳에다 하나씩 붙여 그림이나 모양을 나타내는 장식미술의 하나이다[1]. 그리하여 가까이서 보기에는 작은 조각들을 볼 수 있으며 멀리서 보았을 때는 그 조각들이 만들어 내는 하나의 큰 형상을 알아볼 수 있게 된다. 이러한 모자이크 작업은 과거에 예술가들에 의해서 대부분 진행이 되어 왔으며 현대에 와서 컴퓨터 그래픽스 연구자들에 의해서 비사실적 렌더링 기법중의 하나로 자리잡게 되었다. 이러한 기법으로 만들어진 영상들은 이미지의 흥미성이나 예술성을 높여준다.

예전에는 이러한 모자이크를 위해서는 오랜 시간동안 많은 자료를 수집해야 했다. 하지만 현대에는 저가 디지털 디바이스가 많이 출시되면서 이러한 문제는 많이 해소되고, 일반 사용자들도 손쉽게 사용할 수 있는 소프트웨어들의 등장으로 모자이크 처리에 대한 관심이 높아졌다. 따라서, 최근에는 비사실적 렌더링 기법은 일부 연구자들의 기술이 아니라, 많은 그래픽 디자이너들에게도 구현의 대상이 되고 있다.

본 논문에서는 이러한 모자이크 방식 중 하나인 포토 모자이크를 생성하는 기술에 대해서 소개하고, 컴퓨터 알고리즘을 사용해 자동 생성하는 기술을 제안한다.

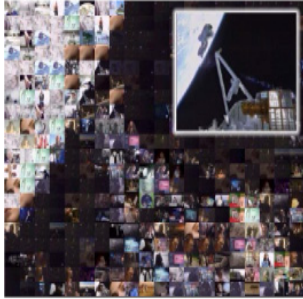
본 논문의 구성은 다음과 같다. 2절에서는 모자이크와 관련된 기술에 대하여 설명하고, 3절에서는 제안하는 포토 모자이크 알고리즘을 설명한다. 4절에서는 이에 대한 실험 결과를 제시하고, 5절에서 결론을 짓겠다.

2. 관련 연구

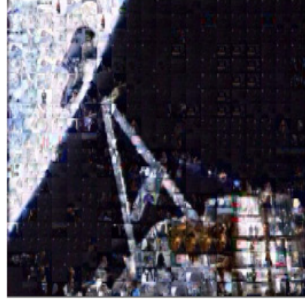
포토모자이크가 Silvers and Hawley[2]에 소개되고 나서 몇몇의 프로그래머에 의해서 프로그램이 개발되기는 하였지만, 속도개선이나 품질향상에 관한 연구에는 아직은 미약하다. 그리고 배포되는 포토모자이크는 대부분 수작업을 하는 것이 현실이다.

비디오 모자이크는 소스영상들로부터 이미지를 추출하여 타겟이 되는 영상에 타일로 배치할 하여 모자이크 동영상 만드는 것이다[3]. 이 기술은 동영상물을 만드는 것이기 때문에 Y, I, Q 모델의 값을 이용하여 Color Matching을 하게 된다. (그림 1)은 소스영상들에서 결과물을 만든 것을 보여주는 예이다. 또한 Color Correction에서는 오버레이를 통하여 원본의 영상을 알아 볼 수 있게 하였다. (그림 2)는 Color Correction후의 결과물을 보여주는데 이는 Color Matching만 통한 결과물 보다는 보다 부드러운 영상을 제공한다. 이렇게 만들어진 영상은 시청자로 하여금 흥미를 이끌 수가 있다. 하지만 10초의 길

이를 가지는 동영상 제작하기 위해서는 3시간 47분이라는 시간이 걸린다. 또는 초당 30프레임인 동영상에서 포토 모자이크를 통한 영상은 세부적인 타일에 대한 이미지를 보여줄 수 없고 단순한 모자이크 효과만 전달될 뿐이다.

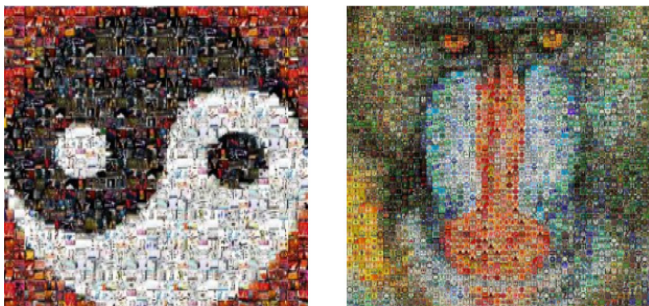


(그림 1) Color Matching 후 모자이크 결과



(그림 2) Color Correction 후 모자이크 결과

Fast Photomosaic[4]은 기존의 포토 모자이크에 비해서 속도를 개선한 것으로 보여진다. RGB를 통하여 Color Matching을 사용하는데 어떠한 방법으로 사용하였는지는 자세히 정보가 제공되지 않는 것이다. 하지만 여기서 속도를 개선하고자 제안하였던 사항은 타일 이미지 정보를 수집할 때 3×3으로 이미지를 나누고 RGB 각각의 채널별로 데이터베이스에 저장을 하면 27개의 요소들로 저장이 되게 된다. 이 요소들은 각 영역의 평균값들로서 타겟 이미지의 타일에 비교되어 대입되게 된다. 사실 이 방법은 Fast Photomosaic이라고 하지만 기존의 개발 기술들과 대조된 실험이 없을 뿐만 아니라 시간도 오래 걸린다. 320×240 크기의 포토 모자이크를 생성하는데 걸리는 시간은 5.980초이다. 1024×768은 32.487초가 필요하다. 또한 이 영상에 쓰인 타일 같은 경우는 알아보기도 힘들며, 색상 조정의 작업도 이뤄지지 않은 것으로 보인다. (그림 3)은 Fast Photomosaic을 통해서 생성된 결과 이미지들이다.



(그림 3) Fast Photomosaic 결과 이미지

3. 제안하는 포토 모자이크 기술

본 논문에서 제안하는 포토 모자이크 기술은 크게 4 단계로 구성된다. 즉, 이미지 분할, 이미지 비교, 이미지 중복사용제한, 컬러튜닝의 네가지 단계를 거치게 되며, 이미지 분할과정에서는 입력받은 이미지를 타일 사이즈에

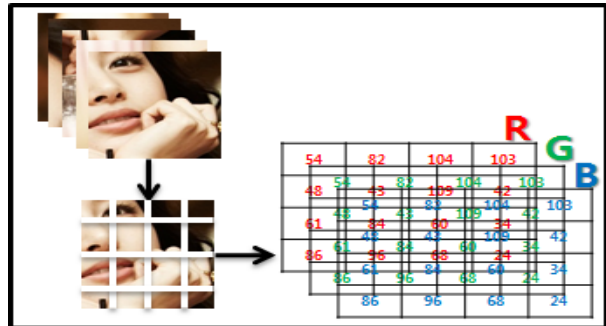
맞게 분할한다. 이미지 비교에서는 분할된 타일 이미지를 불러진 타일 이미지랑 비교를 하게 된다. 이렇게 비교된 타일 이미지는 사용된 횟수에 따라 이미지 최소 중복 작업 처리를 거치게 된다. 마지막으로 배치될 타일 이미지는 원래의 타일 이미지의 명암값을 수정하여 대입되게 된다. 본 논문에서는 타일의 크기로 64×64를 사용하였다.

3.1 이미지 분할

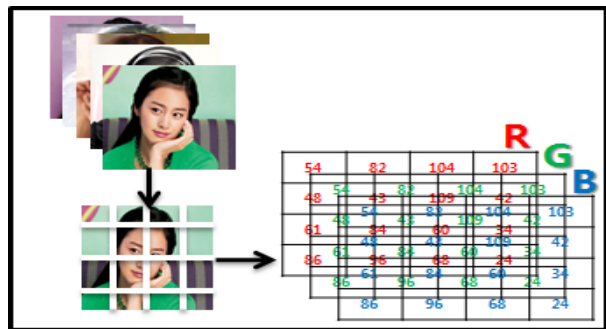
이미지를 (그림 4)와 같이 정해진 타일사이즈로 분할을 한다. 그리고 분할된 각 타일들에 대하여 (그림 5)와 같이 16등분이 되어 나뉘진 각 영역의 RGB 평균값을 계산하여 특징값으로 데이터베이스에 저장한다. 비교될 배경 이미지도 (그림 6)과 같이 똑같은 연산을 수행하게 된다. 원본 이미지의 각각 타일들과 배경으로 쓰일 타일들은 각각 48개의 특징값들을 가지고 비교를 하게 되기 때문에 최적의 타일 이미지를 찾을 수 있다.



(그림 4) 원형영상을 타일크기로 분할



(그림 5) 타일을 16등분하여 블록당 RGB 평균값 계산



(그림 6) 비교할 배경이미지의 16등분

3.2 이미지 유사도 비교

알고리즘은 아래 식과 같이 유클리드 거리(Euclid Distance)를 이용하여 두 이미지의 유사도를 측정하였다. 유클리드거리(Euclidean distance)는 두 점 사이의 거리를 계산할 때 흔히 쓰는 방법이다.

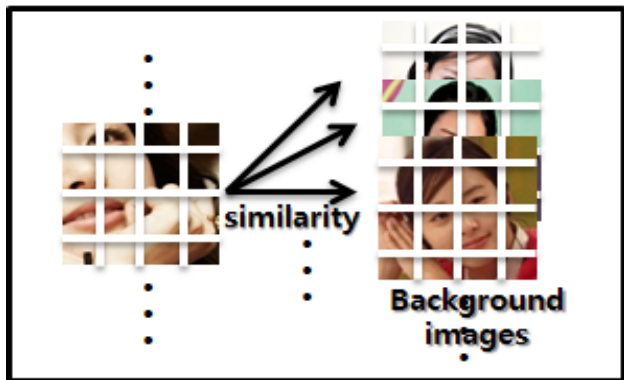
$$d(p, q) = \sqrt{(p_1 - q_1)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

이를 유사도 측정에 이용하기 위하여 식을 다시 표현하면 $D = \sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2}$ 와 같고, D는 원본 이미지의 한 점과 비교 이미지의 한 점에서 RGB 각각의 채널의 차이에 의한 거리값을 나타낸다. 이 값은 낮을수록 두 영상의 유사도는 높게 되며 모든 점의 비교값을 계산하여서 가장 낮은 수치의 값을 가진 비교 이미지를 그 위치에 배치하게 하는 방식이다. 보통 픽셀 단위로 비교를 하게 되며 이미지 크기에 따라서 걸리는 시간이 동적이며, 큰 사이즈의 이미지에는 유사도를 비교하는데 오랜 시간이 걸리게 된다.

제안하는 알고리즘은 48개의 특징점을 가지고 있기 때문에 다음과 같은 식을 사용하였다.

$$D = \sum_{k=1}^{48} (Original B(k) - Compared B(k))^2$$

따라서 선행 작업이 수행되고 나면 (그림 7)과 같이 저장된 값들로 이미지 유사도를 비교하게 된다. 이는 기존에 유클리드 거리로만 비교되던 방법에 비해 비약적인 속도 향상을 보였으며 16등분된 영역의 평균값으로 제공되는 결과 또한 높은 수준을 보여주었다.



(그림 7) 유클리드 거리로 타일 이미지 비교

3.3 이미지 중복사용제한

타일 이미지를 대입시에 주변 색깔이 비슷하여 똑같은 이미지가 너무 많은 횟수로 대입되게 되면 준비된 배경타일도 모두 사용하지 않고 결과물도 똑같은 타일을 이용한 결과물을 만들게 된다. 그리하여 제안하는 알고리즘에서는 특정 타일 이미지를 사용하게 되면 횟수를 체크를 하고 제한한도(default 5)를 넘어가면 그 타일 이미지는 더 이상 사용할 수가 없다. 횟수를 제한하더라도 제한한도내의

배경타일이 한번에 나오는 것을 막기 위해서 사용된 이미지는 배열의 가장 마지막에 위치하게 된다.

3.4 컬러튜닝

컬러튜닝은 이미지들의 비교만으로 타일 이미지를 배치하게 된다면, 최적의 값으로 배치된 값이지 원본 타일 이미지의 값을 동일하게 가지지 않게 된다. 그리하여 원본과 결과물을 대조를 하게 된다면 원본과 색감이 많이 다르다는 것을 찾아 낼 수 있게 된다. 이에 알고리즘은 HSI 값중 Intensity(명암)을 원본타일과 교체될 타일의 값으로 바꾸어서 대입하여 컬러튜닝을 하였다 (그림 8).



(그림 8) 타일의 명암값 교체 전과 교체 후의 영상

4. 실험 결과

제안한 알고리즘은 Intel(R) Core i5 650 프로세서, nVidia GeForce 310, 메모리 3GB DDR3의 사용 환경에서 테스트 하였다.

(그림 9) 와 (그림 10)은 실험에 사용한 원본 이미지이며 (그림 11)과 (그림 12)는 제안한 알고리즘으로 수행한 포토 모자이크 결과를 보여준 영상이다. 그림에 나타난 것과 같이 원본 이미지에 대하여 제안한 알고리즘을 통하여 생성한 포토 모자이크 영상의 품질이 정성적으로 우수한 것을 확인할 수 있다.

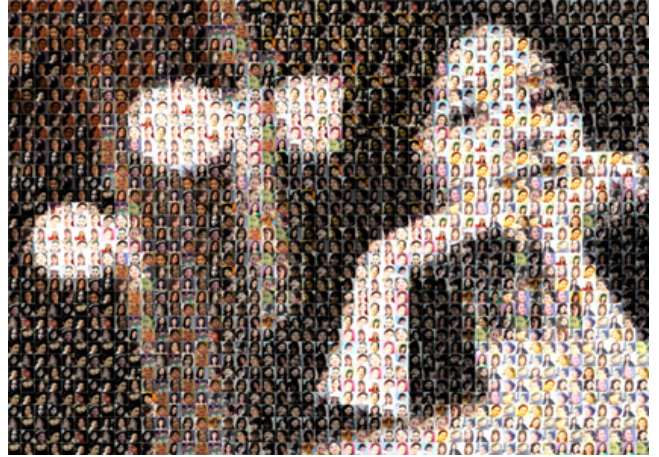
100개의 타일 이미지를 사용하였고, 64x64픽셀크기로 바탕화면으로 사용하였다. 이와 같은 실험결과와는 <표 1>에 요약되어 있으며, 3072x2304의 이미지를 계산하는데 평균 5초의 시간이 소모되었다.

<표 1> 알고리즘 성능분석

| | 제안한 포토 모자이크 |
|--------|-------------|
| 사이즈 | 3072 x 2304 |
| 타일 개수 | 1728 |
| 타일 크기 | 64 x 64 |
| CPU사용율 | 30% |
| 수행시간 | 5초 |



(그림 9) 원본이미지1



(그림 11) 모자이크 결과이미지1



(그림 10) 원본이미지2



(그림 12) 모자이크 결과이미지1

5. 결론 및 향후 과제

본 논문에서는 자동 포토 모자이크 생성 방법을 제안하였다. 특히, 타일의 16등분하여 48개의 특징점을 계산하여 활용함으로써 수행시간을 단축하고 성능의 향상을 얻을 수 있었다. 향후에는 RGB 평균값을 단순 비교가 아닌 최적의 유사 이미지를 찾았을 경우 그 이미지와 같은 값으로 교체하는 방안을 진행할 것이다. 그리고 비교가 될 대상 이미지들의 특징들을 미리 뽑아서 저장할 필요가 있는데, 이에 대해서 양자화와 같은 방법을 쓸 것인지에 대한 연구가 필요할 것으로 보인다.

Acknowledgement

본 연구는 문화체육관광부 및 한국저작권위원회의 2011년도 저작권 기술개발사업의 연구결과로 수행되었음.

참고문헌

- [1] Wikipedia, *Mosaic*, <http://en.wikipedia.org/wiki/Mosaic>
- [2] R. Silvers and M. Hawley, "*Photomosaics*," Henry Holt and Co., Inc. New York, 1997.
- [3] A. W. Klein, T. Grant, A. Finkelstein, and M. F. Cohen, "*Video Mosaics*," Proc. of International Symposium on Non Photorealistic Rendering, 2002, pp. 21-28.
- [4] G. Di Blasi and M. Petralia, "*Fast Photomosaic*," Proc. of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, 2005.