실시간 비디오 포토 모자이크를 위한 이미지 집합 최적화

Image Set Optimization for Real-Time Video Photomosaics

최윤석, Yoon-Seok Choi*, 구본기, Bon-Ki Koo**

요약 본 논문에서는 실시간 포토 모자이크 생성을 위한 이미지 집합 최적화 기법을 소개한다. 포토 모자이크 기법은 작은 포토 이미지를 사용하여 한 장의 큰 사진을 완성하는 기법으로 이미지를 일정한 영역의 셀로 분할한 후 각 셀 영역을 색상, 모양, 무늬 등에서 적합한 이미지로 대체한다. 큰 사진을 구성하는 이미지에 필요한 다양한 패턴을 얻기 위해서는 많은 양의 포토 타일 이미지가 필요하게 된다. 많은 양의 포토 이미지는 이미지 패턴 검색 시간을 오래 걸리게 하고, 이미지 저장을 위해 많은 리소스를 필요로 한다. 이러한 제약은 실시간 처리나 리소스의 제한이 있는 휴대용 기기에서의 포토 모자이크의 적용을 힘들게 한다. 본 논문에서는 패턴 검색 시간 향상과 메모리 요구 최소화를 위해 유전 알고리즘 활용하여 전체 이미지 데이터베이스에서 가장 특징이 있는 이미지를 선별하여 작은 이미지 집합을 구축한다.

Abstract We present a real-time photomosaics method for small image set optimized by feature selection method. Photomosaics is an image that is divided into cells (usually rectangular grids), each of which is replaced with another image of appropriate color, shape and texture pattern. This method needs large set of tile images which have various types of image pattern. But large amount of photo images requires high cost for pattern searching and large space for saving the images. These requirements can cause problems in the application to a real-time domain or mobile devices with limited resources. Our approach is a genetic feature selection method for building an optimized image set to accelerate pattern searching speed and minimize the memory cost.

핵심어 : 포토모자이크, 실시간 처리, 유전 알고리즘, 특징 추출
Keywords: Photomosaic, Real-time, Genetic Algorithm, Feature selection

본 연구는 지식경제부 및 정보통신연구진흥원의 IT핵심기술개발사업의 일환으로 수행하였음. [2008-F-030-01, 방통융합형 Full 3D 복원 기술 개발]

*주저자 : 한국전자통신연구원 연구원 e-mail: ys-choi@etri.re.kr
**공동저자 : 한국전자통신연구원 연구원; e-mail: bkkoo@etri.re.kr

1. 서론

모자이크 기법은 인류가 아주 오래전부터 사용한 이미치 처리 예술 기법이다. 초기에는 이미지를 사각형 등의 작은 공간으로 세분하여 타일이나 색종이, 색유리 등으로 대체하는 기법에서 산업의 발달과 더불어 다양한 매체를 활용하는 기법으로 발전되었으며, 그 중 많이 활용되는 기법 중의 하나가 포토 모자이크이다.

Silvers[1]가 작은 크기의 온전한 이미지를 조합하여 새로운 그림을 만들어 내는 포토 모자이크 방법을 제안한 이후로 여러 형태의 포토 모자이크가 제안되었다. Finkelstein과 Range [2]는 타일 이미지를 다양하게 나열함으로써 다양한 효과를 생성하였으며, 또한 타일 이미지의 색상 보정 기법을 적용하여 원본 이미지의 형태를 강조한 포토 모자이크 기법을 제안하였다. 타일 이미지에 동전 이미지를 사용하고, 다층 개념을 적용하여 여러 개의 타일을 겹쳐서 표현하는 스태커블 [3] 형태도 제안되었다. 보다 빠른 이미지 검색 속도 가속을 위하여 GPU를 사용한 방법을 사용하였다 [4]. 또한 Klein은 단순한 2차원 이미지뿐만 아니라, 비디오를 이용한 비디오 모자이크를 제안하였다 [5]. 위의 방법들은 많은 포토 이미지를 활용하여 좋은 품질의 포토 모자이크를 생성하는데 목적을 두고 있다.

포토 모자이크의 활용이 다양해짐에 따라 웹이나 모바일 기기 등으로 적용이 확대됨과 동시에 사용 환경에 따른 제약도 발생하였다. 특히 PDA와 같은 휴대 장치에서 기억 공간의 제약은 이미지 집합의 소형화가 필요하며, 백화점 등의 광고관 등에 포토 모자이크를 이용한 실시간 광고 효과 등에 적용하기 위해서 실시간 구현도 필요하게 되었다.

본 연구에서는 이와 같이 메모리와 같은 제약에도 포토 모자이크를 적용할 수 있도록 포토 모자이크에 많이 사용되는 패턴을 갖는 이미지를 추출하여 최적화된 타일 이미지 집합을 생성하는 방법을 제안한다. 이는 타일 이미지 집합의 크기를 줄임과 동시에 이미지 검색 시간을 최소화함으로써 비디오 영상에도 실시간으로 포토 모자이크를 생성할 수 있는 성능을 제공할 수 있다.

본 논문은 2장에서는 포토 모자이크와 유전 알고리즘 기반의 특징 추출 기법에 대해서 설명하고, 3장에서는 연구에서 적용한 각 기법을 자세하게 설명한다. 4장에서는 실제 구현 방법과 그 결과를 설명하고, 5장에서는 결론을 맺는다.

2. 본문

2.1 포토 모자이크

포토 모자이크는 대상 사진을 균등한 크기의 사각형 셀로 분할하여 이 공간에 적당한 패턴을 가진 다른 사진으로 대체하여 이미지를 생성하는 기법이다. 포토 모자이크의 큰 특징이라면 거리에 따라서 다른 그림으로 보이는 것으로 근거리에서는 타일을 구성하는 타일 이미지가, 원 거리에서는 대상 이미지가 하나의 완전한 그림으로 보이게 된다.

먼저 포토 모자이크를 적용할 대상 이미지를 $M \times N$ 그리드로 분할 한 후, 각 그리드를 구성한 셀을 대체할 적당한 이미지를 타일 이미지 데이터베이스에서 찾아 이미지로 대체한다.

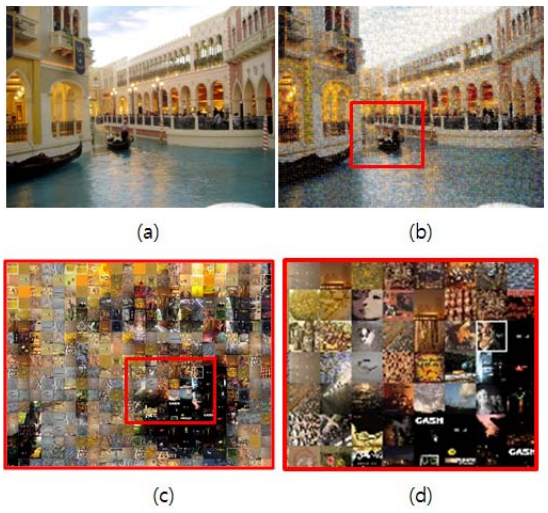

그림 1 대상 이미지 (a), 포토 모자이크 (b), (c)는 (b)의 일부를, (d)는 (c)의 일부를 확대한 이미지

Finkelstein과 Range는 이미지 모자이크를 생성하는 단계를 다음의 4단계로 지정하였다 [2].

1. 대상 이미지를 선택한다.
2. 포토 모자이크에 적용할 타일 패턴을 지정한다. 규칙적인 사각 그리드로 사용하였으며, 6각형 그리드 [6]를 사용할 수 도 있다.
3. 각 그리드의 대상 이미지와 타일 이미지간의 모양과 색상 비교를 통해 타일 이미지를 선택한다.
4. 대상 이미지와 선택된 타일 이미지간의 색상 차이를 계산하여 색상을 교정하여 최종적으로 대상 이미지의 셀 영역을 채우게 된다.

먼저 대상 이미지를 선택 한 다음, 대상 이미지의 영역을 작은 셀 단위로 분할한다. 바둑판 형태의 규칙적인 사각 그리드로 분할할 수 있으며, 서로 다른 크기로 분할 할 수도 있다.

그림 2 좌측부터 사각형, 마름모, 육각형으로 구성된 그리드

그 다음 대상 이미지의 분할 된 셀 영역을 타일 이미지를 이용하여 채우게 된다. 이 때 어떠한 타일 이미지를 사용하는가에 따라서 다양한 효과를 발휘하게 된다. 오로지 하나의 이미지만을 사용할 수도 있으며, 비디오 프레임을 타일 이미지로 사용할 수도 있다 [5]. 가장 많이 사용되는 방법은 아주 많은 이미지를 모은 타일 이미지 집합을 구성하여 이중에서 타일을 선택하여 셀 영역을 채우도록 하는 방법이다.

다음 작업은 셀 영역과 유사한 타일 이미지를 찾는 것이다. 이미지가 유사하다는 것은 이미지의 색상(color), 모양(shape) 그리고 무늬(texture)가 서로 비슷하다는 것을 말한다. 이러한 부분은 내용 기반 이미지 검색 분야에서 다루는 문제로 색상 히스토그램 분석이나, 텍스처 (texture analysis)분석, 모양 분석(edge analysis) 등의 방법이 사용된다 [7]. 포토 모자이크 작업에서 가장 많은 시간이 소요되는 단계로 타일 이미지 집합의 크기가 커질수록 많은 시간이 소요되게 된다.

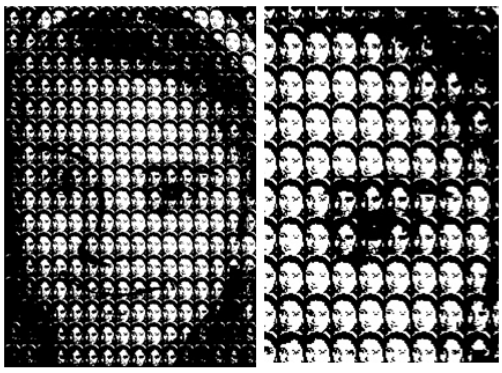

그림 3 하나의 이미지만을 이용한 모자이크 (이중 그림)

다양한 모양과 색상 패턴을 지닌 타일 이미지를 아무리 많이 준비한다 하더라도, 대상 이미지와 완벽하게 유사한 타일 이미지를 찾는 것은 힘들기 때문에 대상 이미지의 색상과 모양 정보를 혼합하는 색상 혼합 기능이 많이 활용이 된다. 그림 4에서 색상 혼합을 적용하였을 때와 적용하지 않았을 때의 차이점을 볼 수 있다.

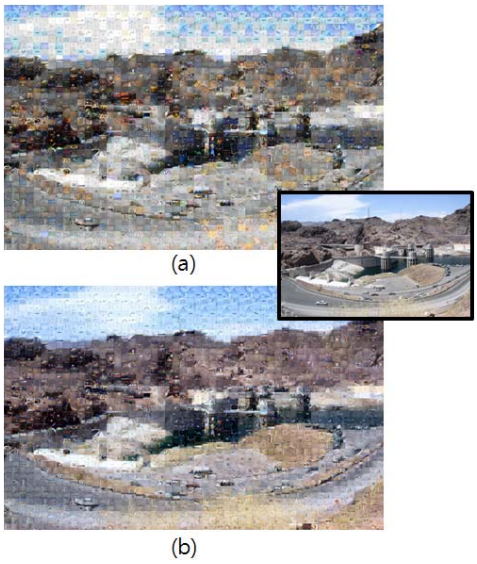

그림 4 원본에 대해 색상 혼합을 사용하지 않은 경우 (a)와 사용한 경우 (b)

2.2 특징 이미지 추출을 위한 유전 알고리즘

특징 추출(Feature Selection)은 전체 D개의 특징 중에서 d개의 하위집합을 선별하는 작업이다. 단순하게 d개의 하위집합을 만드는 것이 아니라, 집합의 크기가 작아짐에 따른 원본 정보의 손실과 추출된 특징의 적용함에 있어서의 성능 및 품질의 감소를 최소화 한다 [8].

본 연구에서는 전체 이미지 집합(D)에서 비디오 포토 모자이크를 위한 d개로 구성된 최적의 이미지 집합을 추출하는데 있어서 유전 알고리즘의 해 공간에 대한 검색 기법을 활용하여 전체 타일 이미지 집합에서 최적의 하위 집합을 생성하고자 한다.

3. 실시간 비디오 모자이크

3.1 이미지 집합 최적화를 위한 유전 알고리즘

본 연구에서는 전체 이미지 집합(D)에서 최적의 이미지 하위 집합을 추출하기 위한 방법으로 유전 알고리즘 기반의 특징 선택(genetic feature selection)을 적용한다.

3.1.1 인코딩 (Encoding)

염색체(chromosome)는 전체 2,800개의 이미지와 각 유전자(gene)를 1:1로 대응시켜 2,800개의 유전자로 구성된다. 즉 n번째 유전자가 1의 값을 가지면, n번째 타일 이미지도 추출 이미지 집합에 포함된다. 각 유전자는 1 또는 0의 값을 가지는데 이미지 최적 집합의 크기를 d로 지정하면, 각 염색체는 총 d개의 1만을 갖게 된다.

```
크기 n의 해 집합 초기화;
repeat {
    for i=1 to k {
        해집합에서 parent1과 parent2 선택;
        offspringi ← crossover(parent1,parent2);
        offspringi ← motation(offspringi );
        Repair(offspringi);
    }
    replace(해집합, [offspring1,offspring2, · · · ,offspringk]);
} until (정지 조건);
return 최적해;
```

3.1.2 초기화

초기 해집합은 무작위로 구성된다. 염색제의 각 유전자는 0과 1의 값을 가지며, 1을 가진 유전자의 수가 모두 d가 되도록 설정되었다. 해집합의 크기는 1,000 으로 설정하였다.

3.1.3. 기본 알고리즘

본 연구에서의 유전 알고리즘의 기본 연산으로 선택(selection) 연산은 토너먼트 선택(tournament selection) [9], 교배 연산(crossover)은 5개의 절단점(cut point)을 갖는 다점 교배(multi-point crossover) [10]를 사용하였다. 변이(mutation)는 각 유전자를 0.1의 확률을 적용하여 0 또는 1의 값으로 토글하였다.

치환(replacement) 작업은 휴리스틱 치환 기법 [11]을 적용하였다 . 생성된 자식 해(offspring)가 부모 해 중에 하나보다 좋을 경우 나쁜 부모 해와 치환된다. 만약 부모 해보다 못할 경우에는 해집합에서 가장 나쁜 해와 치환된다. 해집합은 한 세대에 1개의 해가 치환되는 안정 상태 유전 알고리즘(steady-state GA)로 구성되었으며, 전체 실행 세대는 10만 세대로 지정하였으며, 5,000 세대가 지날 때 새로운 최적해가 나타나지 않으면 수행이 멈추도록 하였다.

3.1.4 교정(Repair)

두 부모해에 대한 교차 연산 및 변이 연산에 대한 결과로 새로이 생성되어진 염색체가 갖는 1의 숫자는 d개보다 많을 수도 있으며, 적을 수도 있다. 이 경우 염색체에 존재하는 1의 수를 d개로 맞추는 작업을 수행하여야 한다. 즉 많을 경우에는 1로 설정된 유전자를 0으로 설정하며, 반대의 경우에는 0을 1로 설정한다. 유전 알고리즘의 수행시 유전자의 수가 전체 이미지 집합의 크기로 지정되어 크기가 매우 크기 때문에 해 공간의 검색이 오래 걸리기 때문에 수렴 속도를 빠르게 하기 위해 유전자의 통계 정보를 활용하였다.

유전자와 타일 이미지가 1:1 대응 되었으므로, 각 유전자에 대응되는 타일 이미지가 훈련 이미지에 대한 포토 모자이크 변환에 사용되었을 때, 해당 훈련 이미지와 변환된 포토 모자이크 이미지와의 오차 정보를 각 타일 이미지에 누적하여 이를 활용하여 염색체 교정시 삭제 혹은 추가하여야 할 유전자를 선택하였다.

3.1.5 적합도 평가 (Fitness Evaluation)

염색체의 평가는 대상 이미지와 포토 모자이크로 생성된 이미지의 오차를 계산하였다. 대상 이미지를 분할하였던 그리드의 각 셀을 기준으로 대상 이미지와 포토 모자이크 이미지 간의 각 셀 간의 색상 거리가 임계값 이상인 셀의 수를 모두 합하였다.

$$Eval_i = \sum_{t=1}^{d} Count_t,$$

$$Count_t = \sum_{c=1}^{n} DistThres(c,h)$$

$$DistThres(c,h) = \begin{cases} 1, Dist(\mathrm{or}igin_c, mosaic_c) > h, \\ 0, otherwise. \end{cases}$$

$Eval_i$ 함수는 염색체 i에 대해서 1로 설정된 d개의 선택된 이미지를 이용하여 포토 모자이크를 생성하였을 때, 대상 이미지와 포토 모자이크 간의 오차 셀 함수($Count_t$)의 합을 의미한다. $Count_t$함수는 이미지를 구성하는 n개의 셀 중에서 임계 오차를 범위를 벗어난 셀의 수를 세는 함수로써, 대상 이미지의 셀($\mathrm{or}igin_c$)과 포토 모자이크의 대응 셀($mosaic_c$)의 색상 거리($Dist(Origin_c, mosaic_c)$)를 비교하여 그 거리가 임계값(h) 이상인 셀의 수를 구하는 함수이다.

3.2 실시간 비디오 모자이크

비디오에 대한 실시간 비디오 모자이크는 기본적으로 이미지 포토 모자이크 방법으로 접근하였다. 비디오 스트림에서 각 프레임을 추출하여 프레임 단위의 포토 모자이크변환을 적용하였다. 이때, 각 프레임 이미지는 사각 그리드에 의해서 지정한 크기의 사각 셀로 분할 되었으며, 각 셀은 타일 이미지 집합의 이미지와 비교하여 가장 적합한 타일 이미지를 선택하여 셀을 대체하였다.

연속된 이미지 프레임이 프레임별로 독립적으로 변환되면, 각 프레임간의 영상의 일관성이 없어지기 된다. 이를 해결하기 위해 연속 프레임별로 같은 위치에 있는 대응셀 간의 색상 변화를 계산하여 변화 정도가 주어진 임계값이하이면, 다음 프레임에서는 대응 셀은 이전 프레임 셀의 타일 이미지를 그대로 사용하였다.

4. 구현 및 결과

본 연구에서는 640×480 픽셀 크기를 가진 이미지를 기준으로 셀의 크기를 20×20로 지정하여 그리드로 분할하였다. 이후 이미지 크기는 별도로 언급하지 않은 경우 640×480로 가정한다. 시스템은 인텔 코어2듀오 CPU 2.83Ghz을 사용하였으며 구현 언어는 C++와 이미지 라이브러리로 OpenCV를 사용하였다.

전체 이미지 집합(D)을 사용한 경우에는 프레임 당 375 ms의 시간이 걸렸다 이는 초당 3 프레임 미만의 성능을 의미하는데, 비디오에 적용하기 위해 본 연구에서는 초당 10 프레임 이상의 성능을 내기 위해서 추출할 최적 이미지 집합의 크기를 200으로 설정하였다. 추출한 이미지의 품질 측정 정확도를 높이기 위해 색상 혼합은 사용하지 않았지만, 특정 패턴의 반복을 피하기 위하여 중복 타일 회피를 지정하였다.

본 연구에서는 최적화 이미지 집합을 구하기 위해 16개 이미지를 특징 이미지 추출을 위한 훈련 집합으로 사용하고, 추출된 이미지 집합의 검증을 15개 이미지를 사용하였다. 각 집합에는 인물, 풍경, 정물 사진 등의 다양한 분류 형태의 이미지를 사용하였다.

그림 6 훈련 집합으로 사용된 16개의 이미지(좌)와 검증 집합으로 사용된 20개의 이미지(우)

그림 7 선택된 최적 타일 이미지 집합의 일부

본 연구에서 사용한 이미지 최적화의 성능 비교를 위해 전체 이미지를 사용하는 경우, 훈련 집합에서 가장 많이 사용된 이미지를 사용하는 경우, 최적화 집합, 무작위 추출 등의 4가지 방법을 비교하였다.

표 1. 각 집합에 대한 오차 차이 분석, (무작위 추출은 1,000번 반복 수행하여 최소 오차 값을 선택하였다.)

이미지 집합	훈련오차	검증오차
전체 이미지	**5,145**	**5,294**
통계적 이미지	5,653	5,886
최적화 이미지	**5,581**	**5,868**
무작위 이미지	–	7,194

최적화 이미지(d_{200})를 적용한 경우에는 프레임당 $95ms$의 시간이 소요되어 초당 10프레임의 성능을 지원하였다. 또한 타일이미지 집합의 각 타일 이미지의 크기를 가로 ,세로를 150×150 픽셀로 정규화 하였는데, 이 경우 전체 타일 이미지 집합에서는 189MB의 메모리가 소요되지만, 최적화 타일 이미지 집합은 13.5MB 만이 필요하기 때문에 리소스의 제한이 심한 모바일 기기에도 적용할 수 있게 된다.

5. 결론

본 논문에서는 유전 특징 추출 알고리즘을 적용하여 원하는 크기의 최적화 이미지 집합을 생성하였다. 물론 사용자의 선택에 따라서 다양한 크기의 이미지 최적 집합을 생성할 수 있다. 이는 다양한 환경에 맞추어 이미지 집합 크기를 최적화 할 수 있으며, 이미지 집합의 크기를 줄임으로써 640×480 크기의 비디오 스트림을 기준으로 약 초당 10 프레임의 실시간 포토 모자이크 생성 성능도 갖추었다. 또한 이미지 크기 최적화와 더불어 이미지 검색 기법을 그래픽 하드웨어 기반의 병렬 계산 기법인 CUDA를 이용하여 가속화한다면 640×480보다 큰 HD급 비디오에 대해서 실시간 포토 모자이크를 지원할 수 있을 것이다.

참고문헌

[1] R. Silvers and M. Hawley, "Photomosaics", New York, NY, USA: Henry Holt and Co., Inc., 1997.

[2] A. Finkelstein and M. Range, "Image mosaics," Proceedings of the 7th International Conference on Electronic Publishing, Held Jointly with the 4th International Conference on Raster Imaging and Digital Typography. pp. 11-22, 1998.

[3] J. Park, K. Yoon, and S. Ryoo, "Multi-layered stack mosaic with rotatable objects", Computer Graphics International, pp. 12-23, 2006.

[4] 강동완, 윤경현, "프로그래밍 가능한 GPU를 이용한 포토 모자이크", 컴퓨터 그래픽스학회 논문지, 14권 1호, pp.17-25, 2008.

[5] A. W. Klein, T. Grant, A. Finkelstein, and M. F. Cohen, "Video mosaics", in NPAR '02: Proceedings of

the 2nd international symposium on Non-photorealistic animation and rendering, pp. 21-28, 2002.

[6] R. Ulichney, "Digital Halftoning", MIT Press, Cambridge, Massachusetts, 1987.

[7] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, G. Taubin. "The QBIC Project: Querying Images By Content Using Color, Texture, and Shape", In Storage and Retrieval for Image and Video Databases, pp.173-187, 1993.

[8] M. Martin-Bautista and M. A. Vila. "A survey of genetic feature selection in mining issues". In Proc. of the Congress of Evolutionary Computation, pp. 13-23, 1999.

[9] B. L. Miller and D. E. Goldberg, "Genetic Algorithms, Tournament Selection, and the Effects of Noise," Complex Systems, pp. 193-212, June 1995.

[10] Davis, L. D., "Genetic Algorithms and Simulated Annealing". San Mateo, California, USA: Morgan Kaufmann Publishers, 1987.

[11] T. N. Bui and B. R. Moon, "Genetic algorithm and graph partitioning", IEEE Trans. on Computers, Vol. 45 No.7 pp. 841-855, 1996.

그림 8 좌측은 전체 집합을 사용한 경우이며 속도는 3,930ms 걸렸으며, 우측은 최적 이미지 집합을 사용하였으며, 속도는 953ms가 소요되었다. 품질 비표를 위하여 색상 혼합은 사용하지 않았다. (이미지 크기는 2060 x 1540).