

블럭매칭을 이용한 비디오 시퀀스의 이미지 모자이크

이지근[○] 김희숙 정성태

원광대학교 컴퓨터공학과

lchg74@gaebyok.wonkwang.ac.kr

hskim@gaebyok.wonkwang.ac.kr

stjung@wonkwang.ac.kr

Image Mosaic from a Video Sequence using Block Matching Methods

Chi-Geun Lee[○] Hee-Sook Kim Sung-Tae Jung
Dept. of Computer Engineering, Wonkwang University

요 약

최근 들어 이미지 모자이크는 가상현실과 웹 브라우저 상에서의 가상환경 구축, 관광, 광고 등, 많은 응용분야에서 관심을 모으고 있다. 이미지 모자이크를 위해 해결해야 할 주된 문제는 이미지들 간의 중복되는 영역에서 정확한 대응점을 찾는 것이다. 지금까지 이 대응점을 찾기 위한 노력은 시스템의 많은 계산량과 시간을 투자해야만 했다. 또한 대부분의 모자이크 여러 차례의 촬영에 의한 정지 화상들 간의 모자이크에 그치고 있다. 따라서 본 논문에서는 비디오 카메라를 이용하여 프레임들 간의 중복되는 영역에서 일정한 비교 영역을 지정하고 그 비교 영역 안에서 대응점을 찾는 블럭 매칭 방법을 적용함으로써 전체적인 모자이크 시간을 줄이는 방법을 구현하였다.

1. 서론

전통적으로 가상현실의 구현은 사물이나 배경을 구현하는데 있어서 3D의 복잡한 모델링과 렌더링을 이용하였고 고가의 특별한 장비와 기술을 요구했기 때문에 많은 제약점을 가지고 있었다. 최근 들어 가상현실에 대한 관심이 높아짐에 따라서 이러한 가상환경을 쉽게 구축할 수 있게 하는 이미지 기반의 기술에 대한 연구가 활발히 이루어지고 있다. 대표적인 예로써 2D이미지를 기반으로 하여 3D의 효과를 내도록 하는 이미지 모자이크 방법이 대표적이라고 할 수 있다. 이미지 모자이크이란 사물이나 배경을 연속적으로 촬영한 이미지를 이음새 없이 하나의 이미지로 연결하는 방법으로써 고가의 장비나 복잡한 모델링 작업 없이도 가상환경을 구축할 수 있다는 장점 때문에 현재 관광이나 광고, 웹 브라우저 상에서 가상현실 구현에 널리 사용되고 있다. 기존의 대부분의 이미지 모자이크 시스템은 디지털 카메라나 특수한 장비를 이용하여 오브젝트를 360° 회전하면서 여러 차례 촬영을 하였고 이 이미지들의 연속된 프레임들간에 모자이크를 위한 대응점을 찾기 위하여 많은 시간과 계산량을 요구하였다.

따라서 본 논문에서는 비디오 카메라를 이용하여 한번의 촬영으로 생성되는 프레임들을 이용하였고 프레임간의 대응점을 찾기 위하여 중복되는 영역에서 일정한 블럭을 지정하여 비교하고 대응점을 찾는 블럭 매칭 방법을 구현 함으로써 전체적인 모자이크 과정의 시간과 계산량을 줄이고자 하였다.

2. 블럭매칭

2.1 프레임간의 비교영역 설정

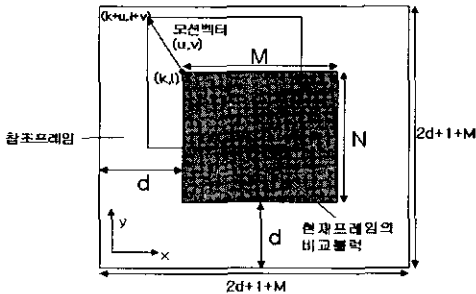
디지털 카메라와 같은 장비를 이용하여 일정한 각도의 움직임을 유지하면서 정지 화상을 여러 차례 촬영을 하는 모자이크 시스템에서는 이미지들간의 중복되는 영역을 인위적으로 조정하는 작업을 고려해야 했다.

본 논문에서는 이미지들간의 중복되는 영역을 고려하지 않고 단 한번의 촬영으로 여러장의 연속된 프레임 이미지를 획득한다. 비디오 카메라를 이용한 촬영으로 여러 장의 프레임 이미지를 획득할 경우 초당 15-30 프레임의 이미지가 생성되기 때문에 이웃해 있는 프레임간에 사물의 이동거리가 매우 작음을 알 수 있고 이미지간에 충분히 중복되는 영역을 얻을 수 있게 된다. 이런 상황에서 이웃해 있는 두 프레임간의 대응점은 각 프레임의 일정한 영역 안에 존재하게 된다. 따라서 이웃해 있는 두 프레임들간의 대응점을 찾기 위하여 두 프레임의 이미지에 일정한 크기의 블럭을 설정하고 그 블럭 안에서 픽셀들의 값에 의하여 대응점을 찾게 된다. 프레임간의 대응점을 찾기 위하여 픽셀값을 이용하기 때문에 비교 영역의 위치 설정은 프레임 이미지의 픽셀값이 유사한 부분이나 가장자리인 피해야 하고 이를 위하여 프레임내에 있는 블럭안의 픽셀값의 평균편차를 이용하여 픽셀들의 값이 다양하게 분포된 곳을 비교영역으로 설정하게 된다.

2.2 블럭 매칭의 원리

블럭매칭 방법은 요즘 각광 받는 모션 계속 방법으로써 다양한 표준 비디오 코딩에 적용되고 있는 방법이다. 블럭 매칭은 블럭의 픽셀들이 프레임간에 서로 같은 방향으로 움직인다는 특성을 이용한다. 다음의 <그림1>은 블럭 매칭 방법의 기본 원리를 설명하고

있다. 현재 프레임에 일정한 크기의 블럭을 지정하고 다음 프레임 안에서 현재 프레임과 대응되는 블럭의 좌표를 찾음으로써 두 프레임 사이의 모션벡터를 구할 수 있다.



<그림1.블럭매칭>

블럭의 크기는 $M \times N$ 이고 모션벡터의 최대 변위는 x 축과 y 축 모두 $\pm d$ 라고 가정한다면 모션벡터 (u, v) 는 $(2d+1+M) \times (2d+1+M)$ 크기의 참조프레임 안에서 대응되는 블럭을 찾음으로써 구할 수 있게 된다. 대응되는 블럭을 찾기 위해서는 (식1)과 같이 정의되는 블럭간의 픽셀에 대한 불일치도를 이용한다. 여기에서 $L'(i, j)$ 는 현재 프레임 블럭안의 픽셀들에 대한 luminance값을 나타내고, $L''(i, j)$ 는 참조프레임 블럭안의 픽셀들에 대한 luminance값을 나타낸다.

$$D(u, v) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |L''(i+u, j+v) - L'(i, j)| \quad (\text{식1})$$

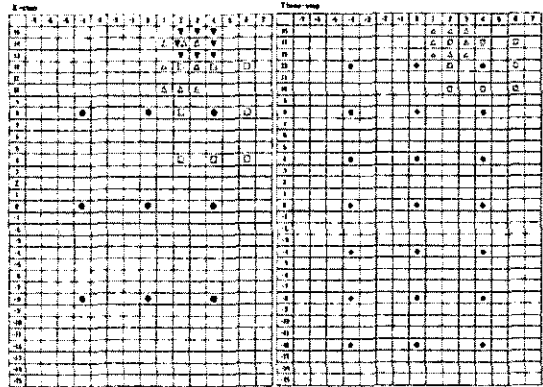
2.3 Full-search

Full-search 알고리즘은 현재 프레임의 블럭에 대응되는 블럭을 참조 프레임에서 찾는 방법중에서 가장 기본적인 방법이다. 이 알고리즘에서는 모션벡터가 존재할 수 있는 모든 탐색공간 $(-d \leq u, v \leq d)$ 에서 $D(u, v)$ 을 계산하여 그중에서 $D(u, v)$ 값이 최소가 되는 위치의 블럭을 대응블럭으로 선택한다. 이 방법은 탐색 영역의 모든 위치에서 두 블럭 사이의 불일치도 $D(u, v)$ 값을 계산 하기 때문에 모자이크의 정확도는 우수하지만 $(2d+1) \times (2d+1)$ 번의 블럭 비교를 하기 때문에 모자이크 처리 속도가 떨어지는 단점을 가지고 있다.

2.4 K-step search

K-step 알고리즘은 Full-search 알고리즘의 단점을 개선한 방법이다. 대응점을 찾기 위하여 각 프레임의 $(-d \leq u, v \leq d)$ 구간내에서 모든 위치에 대하여 불일치도 D 값을 계산하는 대신 아래 <그림2>와 같이 첫 번째 단계에서 블럭에 대한 모션벡터의 이동범위인 d 의 절반의 값으로 거리를 유지하는 점을 탐색공간 안에 위치 시킨 후 그 점의 $D(u, v)$ 값 들을 계산하여 $D(u, v)$ 값이 최소인 점의 위치를 선택하게 된다. 그 다음 단계로서 선택된 점에서 초기 d 값의 절반인 거리에 있는 점들에 대하여 $D(u, v)$ 을 계산하고 픽셀값의 차가 최소인 점을 다시 한번 선택하게 된다. 이러한 단계를 반복하여 블럭안에 지정되어진 점

들의 거리가 1이 될 때까지 반복하여 픽셀값의 차가 최소인 점을 찾게 되고 이 점을 최종 대응점으로 선택하게 된다.



<그림2 K-step과 Three-step >

여기에서 초기 점들간의 거리는 최대 움직임 변위의 절반으로 한다. 변위의 절반의 값이 소수점일때는 반올림값을 적용하게 된다. 예를 들어 변위가 7일 경우, $7/2 = 3.5$ 가 되고 반올림 하여 점들간의 거리가 4가 된다. 이러한 경우 계산 반복 회수는 다음 (식2)와 같다.

$$K = \lceil 1 + 8 \lceil \log_2 (d + 1) \rceil \rceil \quad (\text{식2})$$

2.5 3-단계 search

3-단계 알고리즘은 위의 <그림2>와 같이 K-step 알고리즘의 반복 회수를 3회로 하여 참조 프레임 안에서 대응 블럭을 찾는 방법으로써 블럭 안에 지정된 점들간의 거리가 $4 \rightarrow 2 \rightarrow 1$ 이나 $3 \rightarrow 2 \rightarrow 1$ 과 같이 3번의 단계로 1이 될 수 있도록 하여 반복 회수를 줄인 방법이다. 이렇게 3회에 걸치는 오차도 $D(u, v)$ 값의 계산에 의하여 블럭간의 위치에 대하여 $D(u, v)$ 값이 최소인 위치를 찾아 냄으로써 적합한 대응점을 얻을 수 있게 된다.

대응되는 블럭을 찾기위한 계산 회수를 비교해 보면, 위의 <그림2>와같이 최대 움직임의 범위가 x 축으로 7이고 y 축으로 15일 때 계산 회수는 다음 <표>과 같다.

Full-search	K-step	Three-step
31× 15=465	9+8+8+8=33회	21+8+8=37회

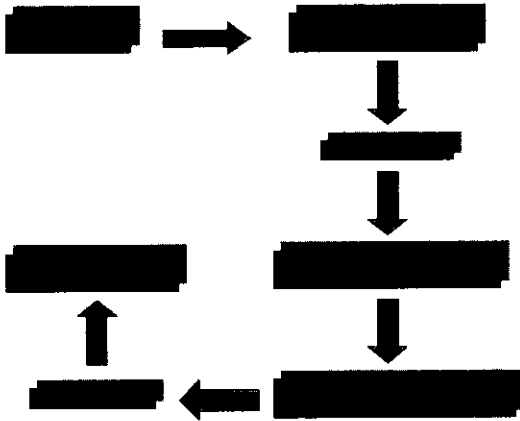
<표1. 픽셀값의 계산회수 비교>

3. 블럭 매칭을 이용한

비디오 시퀀스의 이미지 모자이크

본 논문에서 제안 하는 모자이크 시스템의 전체적인 구성은 <그림3>과 같다. 비디오 카메라를 이용하여 사물이나 배경을 촬영한 후, 320×240 크기의 24비트 RGB 값을 갖는 AVI 동영상 파일을 생성한다. 생성된

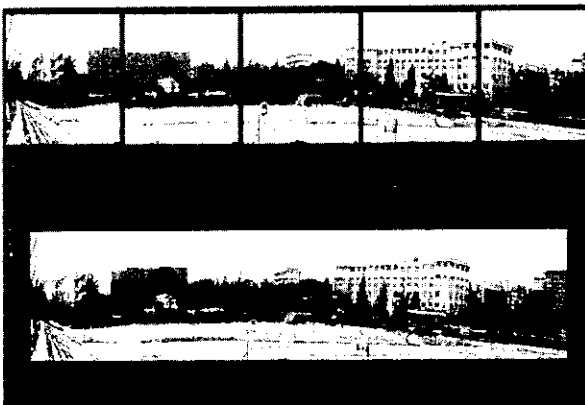
AVI파일을 프레임별로 로드하여 나열시키고 비교 블럭 영역 설정을 위하여 프레임에 존재하는 블럭내의 픽셀에 대한 평균 편차를 구한다. 편차의 값이 큰 값을 가지고 있는 영역을 비교 영역으로 설정하고 그 영역에서 블럭매칭방법을 적용하게 된다. 블럭 안에서 Full search, K-step, Three-step의 세 가지 방법을 적용할 수 있게 하였고 모자이크되어 완성된 이미지를 출력하여 비교할 수 있게 하였다.



<그림3 모자이크 시스템 구성도>

4. 실험 결과

블럭 매칭을 이용한 비디오 시퀀스의 이미지 모자이크 실험은 주변 배경을 중심으로 하여 삼발이에 장치된 비디오 카메라를 사용하여 촬영을 하였다. 블럭 매칭을 이용하여 블럭 안에서 Three-step에 대한 입력 프레임과 모자이크된 출력 이미지는 <그림4>와 같다.



<그림 4. 실험 결과 이미지>

테스트에 사용된 AVI파일의 프레임 수는 434프레임 이었고 사용된 시스템의 OS는 Windows 2000 Advanced Server를 사용했으며 펜티엄III 667Mhz의 128메가 바이트 메모리에서 테스트되었다. 434개의 프레임들을 블럭매칭 방법을 이용하여 Full search, K-step, Three-step 모자이크를 하는데 각각 소요된 시간은 <표 2>와 같다.

* Frame 수 : 434 프레임

	Full search	K-step	Three-step
처리시간(초)	306초	80초	57초

<표2. 블럭매칭을 이용한 모자이크에 소요된 시간>

실험 결과를 살펴보게 되면 같은 조건하에서 블럭 매칭을 이용하여 Full search, K-step, Three-step 방법으로 모자이크 했을 경우, 모자이크가 완성된 출력 이미지는 서로 차이가 거의 없었다. 하지만 모자이크 처리 시간을 살펴보면 Three-step방법을 이용하여 모자이크 했을 경우가 Full search 방법을 이용했을 때 보다 약 5.5배의 속도가 향상 됐으며 가장 짧은 처리 시간을 나타내고 있다.

5. 결 론

위의 실험 결과와 같이 블럭매칭 방법을 이용하여 불필요한 픽셀값들의 계산을 피할 수 있고 블럭 안에서 Three-step알고리즘을 적용하여 전체적으로 모자이크 처리에 걸리는 처리시간을 단축시킬 수 있음을 알 수 있다. 최근에 웹상에서의 가상현실 구축과 관련하여 관광, 상품광고등에서 이미지 모자이크 기법이 선호되고 있는 시점에서 블럭 매칭 모자이크 처리 시간은 매우 중요한 문제임에 틀림없다. 본 논문에서 제시한 블럭 매칭을 이용한 Three-step 알고리즘의 모자이크 기법은 영상통신시대에 여러 응용분야에 널리 적용될 수 있을 것이다.

6. 향후 연구과제

위의 실험을 하면서 몇 가지 문제점이 발견되었는데 비디오 카메라로 촬영시 생성되는 프레임의 수가 너무 많다는 것과 카메라의 이동 방향이 한쪽 방향으로 일정하게 움직여야 오차가 적게 나온다는 것이었다. 동영상 로드시 프레임을 비교하여 사물의 이동이 적으면 프레임을 건너뛰어 적은 수의 프레임들을 로드하는 문제를 연구할 것이고 카메라의 움직임 방향에 상관없이 오차를 적게 하는 방법에 관한 연구를 진행 할 것이다. 또한 현재는 프레임간 비교를 하였으나 mosaic-to-frame 기법과 Global 모자이크 기법에 대하여 연구를 할 것이다.

7. 참고문헌

- [1] <http://www.image.cityu.edu.hk/> cityu image processing Lab
- [2] L.G. Brown, "A survey of image registration techniques," ACM Computing Surveys, Vol.24, No.4, pp.325-376, 1992.
- [3] H. S. Sawhney, R. Kumar, G. Gendel, J. Bergen, D. Dixon, and V. Paragano, "VideoBrush: Experience with Consumer Video Mosaicing," Proc. of Fourth IEEE Workshop on Application of Computer Vision, pp. 56-62, 1999.
- [4] R.Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp.438-442, Aug. 1994
- [5] R. Szeliski, H. Y. Shum, "Video mosaics for virtual environment," IEEE Computer Graphics and Application, 16(2):22-30, 1996.