
블록매칭을 이용한 비디오 시퀀스의 이미지 모자이크

이지근* · 정성태**

Image Mosaic from a Video Sequence using Block Matching Method

Chi-Geun Lee* · Sung-Tae Jung**

요약

최근 들어 개인용 컴퓨터 성능의 향상과 인터넷 기술의 발전에 따라서 이미지 모자이크는 가상환경 구축, 관광, 광고, 의료 영상 등, 많은 응용분야에서 관심을 모으고 있다. 이미지 모자이크의 주된 문제점은 이미지들 간의 정확한 대응점을 찾는 것이다. 그러나 기존의 대부분의 모자이크 기법들은 정확한 대응점을 찾기 위해서 복잡한 계산과 많은 처리시간을 요구했으며 모자이크 이미지를 생성을 위해 사물이나 배경 주위를 360° 회전하면서 여러 차례 반복 촬영을 해야 하는 어려움을 가지고 있었다. 본 논문에서는 일반 비디오카메라를 이용하여 단 한번의 촬영에 의해 생성된 연속 프레임을 사용하였고 프레임간의 모자이크에 있어서 방향성을 고려한 새로운 방법의 3단계 블록매칭 방법을 적용함으로써 전체적인 모자이크 처리 속도를 단축하는 방법을 제안한다. 실험 결과에 의하면 본 논문에서 제안된 방법이 기존의 블록매칭 방법인 전역탐색이나 K-단계 탐색에 비하여 보다 효과적임을 알 수 있었다.

ABSTRACT

In these days, image mosaic is getting interest in the field of advertisement, tourism, game, medical imaging, and so on with the development of internet technology and the performance of personal computers. The main problem of image mosaic is searching corresponding points correctly in the overlapped area between images. However, previous methods requires a lot of CPU times and data processing for finding corresponding points. And they need repeated recording with a revolution of 360 degree around objects or background. This paper presents a new image mosaic method which generates a panorama image from a video sequence recorded by a general video camera. Our method finds the corresponding points between two successive images by using a new direction oriented 3-step block matching methods. Experimental results show that the suggested method is more efficient than the methods based on existing block matching algorithm, such as full search and K-step search algorithm.

키워드

Block matching, Image Mosaic, Image Processing

1. 서론

최근 들어 컴퓨팅 환경과 웹의 발달에 따라서 이미지 모자이크에 대한 관심이 높아지고 있다. 이미지 모자이크란 사물이나 배경을 연속적으로 촬

영한 이미지를 이음새 없이 하나의 이미지로 연결함으로써 넓은 영역의 시점을 생성해 내는 방법 중의 하나이다. 최근에는 이미지 모자이크를 2차원 정보뿐만 아니라 3차원 정보의 표현과 분석에도

*원광대학교 컴퓨터공학과

**원광대학교 전기전자 및 정보공학부

접수일자 : 2003. 8. 28

이용하고 있으며 웹상에서의 상품광고와 관광산업, 게임, 의료영상 등, 다양한 응용분야에서 활용하고 있다. QuickTime VR[1]이나 Photovista[2]와 같은 상용 프로그램은 이미지 모자이크 기법을 활용한 좋은 예이다.

이미지 모자이크 중에서 360도 전체 방향에서의 이미지들을 하나의 이미지로 만드는 것을 파노라마 이미지 모자이크이라고 한다. 파노라마 이미지 모자이크은 건물 내의 공간이나 주변 경치, 배경 등을 둘러볼 수 있는 환경을 만드는데 널리 이용되고 있다[3]. 그림 1은 파노라마 이미지 모자이크의 개념을 보여주고 있다.

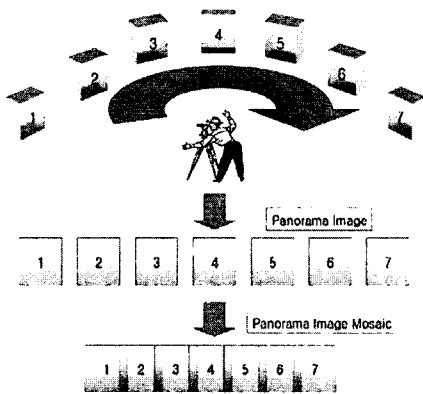


그림 1. 파노라마 이미지 모자이크
Fig. 1 Panorama image mosaic

이미지 모자이크에 있어서 가장 중요한 문제는 이미지들 간의 모자이크를 위한 정확한 대응점을 찾는 문제이다[4][5]. 기존의 대부분의 이미지 모자이크 시스템은 파노라마 카메라나 디지털 카메라, 혹은 어안(fish-eye)렌즈를 장착한 특수한 장비를 이용하여 오브젝트를 360°회전하면서 여러 차례 촬영 하였고[6][7][8][9][10], 생성되는 정지 화상들 간의 모자이크를 위한 대응점을 찾기 위하여 이미지의 어파인 변환과 원근 변환 행렬 계산과 같은 방법을 이용하였다[11][12][13]. 이로 인하여, 모자이크를 처리하기 위해서는 상당히 많은 시간과 계산량이 요구되었다.

따라서 본 논문에서는 일반적인 비디오카메라

를 이용하여 한번의 촬영으로 생성되는 연속 프레임들을 이용하였고, 연속 프레임간의 모자이크에 필요한 대응점을 찾기 위하여 블록매칭 방법을 적용하였다. 블록매칭 방법에는 가장 기본적인 방법인 전역탐색(full-search) 방법과 K-단계 탐색 방법, 그리고 이를 응용한 다양한 방법 등이 있다 [14][15][16]. 본 논문에서는 K-단계 탐색 방법을 수정한 3단계 탐색 방법을 제안한다. 실험을 통하여 본 논문에서 제안한 3단계 탐색 방법이 전체적인 모자이크 처리 과정의 처리시간과 계산량에 있어서 전역탐색 방법이나 K-단계 탐색 방법에 비하여 보다 효과적임을 알 수 있었다.

II. 블록매칭

블록매칭은 블록의 픽셀들이 프레임간에 서로 같은 방향으로 움직인다는 특성을 이용한 모션계측 방법으로 J. R. Jain[17]에 의해 제안 되었으며 계산의 간결성과 결과의 정확성으로 H.261, H.263, MPEG-1, MPEG-2, MPEG-4와 같은 표준 비디오 코딩에서 사용되는 방법이다[18]. 블록매칭 알고리즘은 사물의 이동이나 움직임을 추적하기 위하여 카메라나 비디오카메라로 촬영 후, 생성되는 연속된 이미지 영상들에 대하여 프레임안에 일정한 크기의 블록을 지정하고 블록안의 픽셀들에 대한 밝기 값을 계산하여 움직임에 대한 모션벡터를 구함으로써 모션계측을 하게 된다. 그림 2는 블록매칭 방법의 기본 원리를 설명하고 있다.

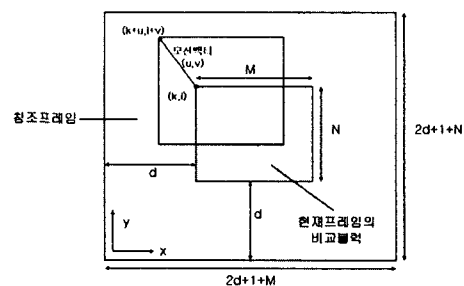


그림 2 블록매칭의 기본 원리
Fig. 2 Principle of Block matching

그림 2에서 현재 프레임의 블록의 크기는 $M \times N$ 이고 모션벡터의 최대 변위는 x축과 y축 모두 $\pm d$ 라고 가정한다면 이미지간의 사물의 움직임을 나타내는 모션벡터 (u, v) 는 $(2d+1) \times (2d+1)$ 크기의 참조 프레임 안에서 대응되는 블록의 위치를 찾음으로써 구할 수 있게 된다.

대응되는 블록의 위치를 찾기 위해서는 식(1)과 같이 정의되는 블록간의 불일치도를 이용한다. 여기에서 $L'(i, j)$ 는 현재 프레임 블록안의 픽셀들에 대한 밝기 값을 나타내고, $L''(i, j)$ 는 참조 프레임 블록안의 픽셀들에 대한 밝기 값을 나타낸다. $D(u, v)$ 를 u, v 가 $-d \leq u \leq d, -d \leq v \leq d$ 인 영역에 대하여 구한 다음, $D(u, v)$ 값이 최소가 되는 점을 대응점으로 선정하게 된다.

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

본 논문에서 사용한 블록매칭 방법은 비디오카메라를 이용하여 촬영된 연속 프레임 이미지들을 나열하고 블록매칭의 3단계 탐색 알고리즘을 적용시켰다. 입력되는 현재 프레임에 일정한 크기의 블록을 지정하고 다음 프레임 안에서 현재 프레임과 대응되는 블록의 좌표를 찾음으로써 두 프레임 사이의 모션벡터를 구할 수 있다. 그림 3과 4는 프레임간의 영상 차이를 이용한 모션 추적과 이에 따른 블록매칭을 위한 블록의 이동을 보여주고 있다.

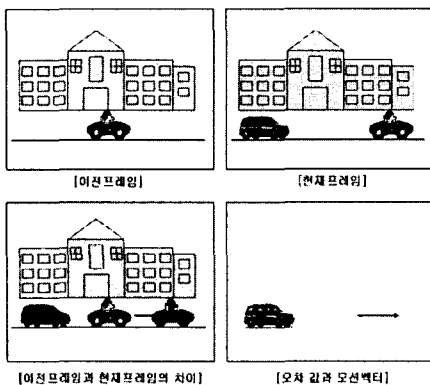


그림 3. 프레임간의 모션 추적
Fig. 3 Motion estimation between frames

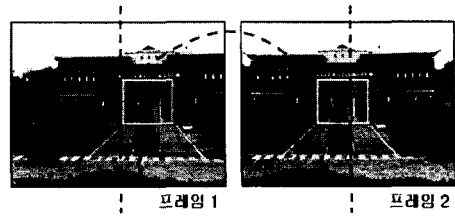


그림 4. 블록의 이동 예
Fig. 4 An example of a block motion

2.1 전역탐색(full-search) 알고리즘

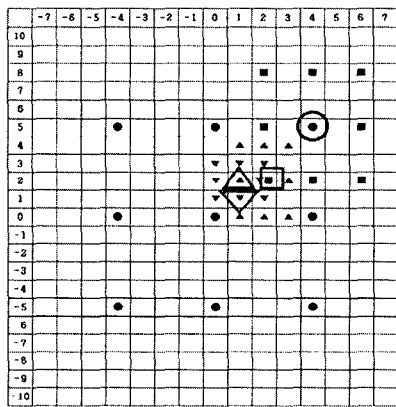
전역탐색 알고리즘은 현재 프레임의 블록에 대응되는 블록을 참조 프레임에서 찾는 방법 중에서 가장 기본이 되는 방법이다. 이 알고리즘은 입력으로 들어오는 프레임들에서 현재 프레임에 대응되는 블록을 참조 프레임에서 찾기 위하여 대응 블록이 존재 할 수 있는 탐색공간 $(-d \leq u \leq d, -d \leq v \leq d)$ 의 모든 위치에서 두 프레임의 블록간의 픽셀에 대한 불일치도 $D(u, v)$ 를 계산하여 그 중에서 $D(u, v)$ 값이 최소가 되는 위치의 블록을 대응 블록으로 선택한다. 이 방법은 탐색 영역의 모든 위치에서 $D(u, v)$ 값을 계산하기 때문에 모자이크의 정확도는 우수하지만 식(2)와 같은 회수의 블록 비교를 하기 때문에 모자이크 처리 속도가 떨어지는 단점을 가지고 있다.

$$\text{비교회수} : N = (2d+1) \times (2d+1) \quad (2)$$

2.2 K-단계 탐색 알고리즘

K-단계 탐색 알고리즘은 전역탐색 알고리즘의 단점을 개선한 방법이다. K-단계 탐색 알고리즘은 블록 간의 대응점을 찾기 위하여 첫 번째 단계에서는 그림 5의 원으로 표시된 점들과 같이 모션 벡터의 이동범위인 d 의 절반의 값으로 거리를 유지하는 점을 탐색 공간 안에 위치시킨 후, 블록 안에 지정된 각 점들의 위치에서 두 프레임 사이에 존재하는 블록 안의 픽셀들에 대한 불일치도 $D(u, v)$ 값들을 계산하여 $D(u, v)$ 값이 최소인 점의 위치를 초기 선택 위치로 선정하게 된다. 그 다음 단계로서, 앞 단계에서 선택된 점들 간의 거리의 절

반인 거리를 유지하는 점들을 위치시킨다. 여기에서 초기 변위의 절반 값이 소수점일 경우에는 반올림 값을 적용하여 점의 거리를 유지하게 된다. 예를 들어 변위가 7일 경우에는 $7/2 = 3.5$ 가 되고 이에 반올림을 적용하여 점들 간의 거리가 4가 된다. 이 점들에 대하여 다시 한번 $D(u, v)$ 값을 계산한다. 이어서 $D(u, v)$ 값이 최소가 되는 위치, 즉, 픽셀 값의 차가 최소인 점을 다시 선택하게 된다. 위와 같은 단계를 반복하여 블록 안에 지정된 점들의 거리가 1이 될 때까지 반복한다. 점들의 거리가 1이 됐을 때 블록 안의 픽셀 값의 차가 최소인 점을 찾게 되고 이 점을 최종 대응점으로 선택하게 된다.



1 단계 : ● 2 단계 : □ 3 단계 : △ 4 단계 : ▼
 그림 5. K-단계 탐색
 Fig. 5 K-step search

K-단계 탐색 알고리즘의 경우, 두 프레임에서의 블록 비교회수는 식(3)과 같다.

$$N = \lceil \log_2(d+1) \rceil \quad (3)$$

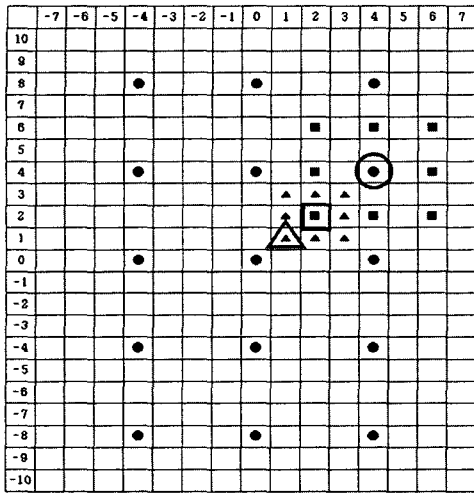
K-단계 탐색 알고리즘은 탐색 공간 내에 일정한 위치를 지정하고 그 위치에서만 K번의 블록 비교를 하기 때문에 전역탐색 알고리즘에 비해 블록 비교 회수가 현저하게 줄어들어 모자이크 처리 속도가 향상되는 장점을 가지고 있는 방법이다.

2.3 새로운 3-단계 블록매칭 알고리즘

K-단계 탐색 알고리즘은 전역탐색 알고리즘에 비하여 대응점을 찾기 위한 계산량은 단축시킬 수 있지만 두 프레임 안에 지정된 블록의 크기에 따라 대응점을 찾기 위한 계산 반복 단계도 변할 수 있다. 이러한 원인으로 인하여 블록의 크기에 따라서 대응점을 찾기 위한 탐색 점의 간격도 변하게 되고, 블록안의 탐색 점들의 간격이 커질 경우에는 초기 대응점으로 선택되는 위치가 실제 정확한 대응점이 아닌 잘못된 위치를 대응점으로 선정하게 되어 모자이크의 오차가 생길 가능성도 높아지게 된다.

따라서 본 논문에서 K-단계 탐색 알고리즘의 단점을 보완하는 새로운 방법의 3-단계 탐색 알고리즘을 제안한다. 제안된 3단계 탐색 알고리즘은 K-단계 탐색 알고리즘의 탐색 반복 단계를 블록의 사이즈와는 상관없이 3단계로 제한하는 대신 블록 안의 대응점을 찾기 위한 탐색점의 간격을 좁혀서 블록 내에서 정밀 탐색하여 K-단계 탐색 방법에서 발생할 수 있는 대응점의 오차를 줄이고자 하는 방법이다. 제안된 3-단계 탐색 알고리즘은 입력되는 두 프레임에 지정된 블록 안의 점들의 거리가 단계별로 4 -> 2 -> 1 이나 3 -> 2 -> 1이 되도록 지정하여 3단계로 탐색을 진행되게 함으로써, 단 3번의 단계를 거치면서 두 블록 간의 픽셀에 대한 최소의 밝기 값의 차이를 갖는 대응점을 찾는 방법이다. 그림 6과 같이 첫 번째 단계에서 입력되는 두 프레임에 존재하는 블록 안에 거리가 4를 유지하는 점들을 위치시키고 이 점들에 대하여 두 블록의 픽셀에 대한 불일치도 $D(u, v)$ 값을 계산하여 $D(u, v)$ 값이 최소가 되는 점을 선정하게 된다. 두 번째 단계로서 앞 단계에서 선정된 점들을 중심으로 점들의 거리가 앞 단계의 절반인 2가 되는 점들의 위치에서 다시 한번 $D(u, v)$ 값을 계산하여 마찬가지로 $D(u, v)$ 값이 최소가 되는 점을 다시 선정하게 된다. 마지막 단계로서 앞 단계에서 선정된 점들을 중심으로 거리가 앞 단계의 절반인 1을 유지하는 점들에 대한 블록 간의 픽셀에 대한 불일치도 $D(u, v)$ 값을 계산하여 $D(u, v)$ 값이 최소가 되는 점의 위치를 최종 대응

블록으로 선정하게 된다.



1 단계 : ● 2 단계 : □ 3 단계 : △
 그림 6. 3-단계 탐색
 Fig. 6 3-step search

이 3-단계 탐색 알고리즘은 K-단계 탐색 알고리즘의 대응점을 찾기 위한 탐색 반복회수를 블록의 크기와는 상관없이 3회로 제한함으로써 두 프레임간의 대응점을 탐색하고 모자이크를 수행하는데 필요한 계산량을 줄임으로써 모자이크 처리시간을 더욱 단축시킬 수 있는 장점을 가지고 있다.

K-단계 탐색방법은 블록의 크기에 따라 대응점을 찾기 위한 초기 탐색 점들 간의 거리가 유동적으로 변할 수 있어서 블록의 크기가 커지면 정확한 초기 대응점을 찾는 데 오차가 발생할 가능성이 높다. 하지만 제안된 3-단계 탐색 알고리즘은 첫 번째 단계에서 블록 안에 지정된 점들의 거리가 블록의 크기와는 상관없이 4 또는 3으로 지정되기 때문에 블록 크기의 변화와는 상관없이 대응점을 찾기 위하여 초기에 더욱 조밀하게 정밀탐색하게 되고 이에 따라 첫 번째 단계에서 상당히 정확한 대응점을 찾을 수 있게 된다. 또한 앞 단계의 정확성에 기반하여 이어지는 2단계와 3단계의 탐색 단계를 거침으로써 더욱 정확한 대응점을 찾을 수 있기 때문에 K-단계 탐색 방법의 대응점에 대한 오차를 줄일 수 있게 된다.

표 1은 블록매칭을 이용한 전역탐색 알고리즘

과 K-단계 탐색 알고리즘, 그리고 본 논문에서 제안한 3-단계 탐색 알고리즘의 픽셀 값의 계산 회수를 비교한 결과이다.

표 1. 블록매칭의 계산회수 비교
 Table. 1 Comparison of computation time between block matching methods

◆ 블록의 크기 : 40 × 40
 ◆ 블록의 최대 움직임 범위 : X방향 = 7, Y방향 = 10

전역탐색	K-단계 탐색	3단계 탐색
21×15 = 315회	9+8+8+8 = 33회	15+8+8 = 31회

III. 블록매칭을 이용한 이미지 모자이크

본 논문에서 제안하는 모자이크 시스템은 특수한 장비의 이용을 배제한 일반적인 비디오카메라를 이용하여 단 한번의 촬영으로 생성되는 연속 프레임 이미지를 사용하였고, 효율적인 블록매칭 알고리즘을 이용하여 처리시간을 줄일 수 있도록 구성되었다. 또한, 카메라의 이동방향에 따른 사물의 이동성, 즉, 블록매칭의 방향성을 고려하여 이전 프레임의 모션벡터 방향으로 블록을 정밀 탐색함으로써 프레임 간의 모자이크 처리의 정확성을 높일 수 있도록 하였다. 본 논문에서 제안하는 모자이크 시스템의 전체적인 흐름도는 그림 7과 같다.

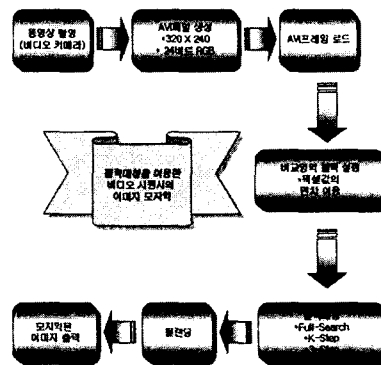


그림 7. 모자이크 시스템의 전체적인 흐름도
 Fig. 7 Flow chart of image mosaic system

3.1 비디오카메라를 이용한 프레임 획득

비디오카메라를 이용하여 생성된 AVI 파일의 연속된 프레임을 획득할 경우 초당 15~30 프레임의 이미지를 획득할 수 있게 된다. 따라서 생성되는 프레임 이미지간의 사물의 이동거리가 매우 작음을 알 수 있고, 이로 인하여 프레임간의 모자이크를 위한 중복영역을 충분히 획득할 수 있게 된다. 중복되는 영역에서 두 프레임간의 대응점을 찾기 위하여 그림 8과 같이 두 프레임의 이미지에 일정한 크기의 블록을 설정하고 그 블록 안에서 픽셀들의 불일치도 값을 계산하여 불일치도 값이 최소가 되는 위치를 대응점으로 선정하는 방법을 이용하였다.

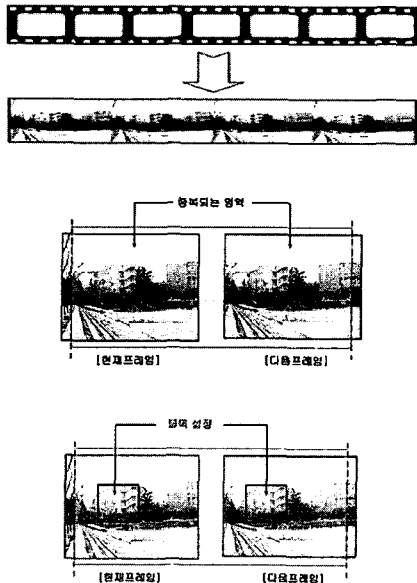


그림 8. 연속된 프레임 이미지 획득과 중복영역
Fig. 8 Capturing panorama video sequence and overlapped area

3.2 프레임간의 비교영역 설정

픽셀 값의 비교에 의하여 대응점 추출을 하기 위해서는 이웃한 프레임간의 픽셀 값은 서로 비교할 수 있을 정도의 다양한 픽셀 값들이 분포되어

있어야 한다. 따라서 본 논문에서는 이웃한 두 프레임에 존재하는 블록 내의 픽셀에 대한 평균 편차를 계산하여 편차의 값이 큰 값을 나타내는 곳이 다양한 픽셀 값을 가진 영역이라고 판단하고 그 곳을 비교 영역으로 선정하도록 하였다.

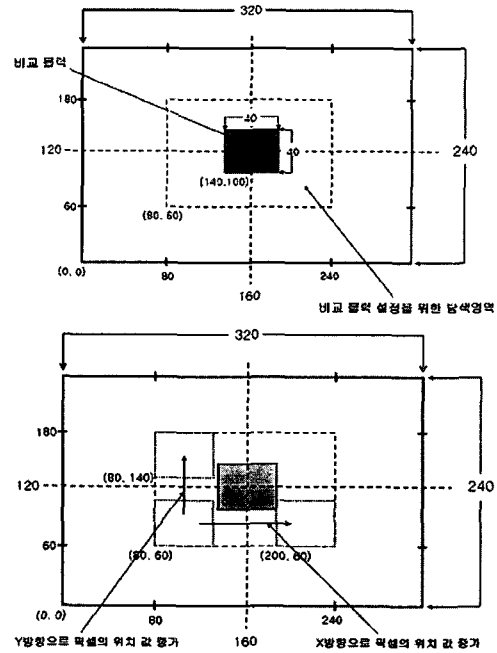


그림 9. 비교 블록 영역 설정을 위한 개념도
Fig. 9 Method for the configuration of search area

그림 9와 같이 입력되는 320 × 240 크기의 프레임에 블록매칭을 위한 블록의 크기는 최적의 크기로 40 × 40 으로 설정하고, 다양한 픽셀 값이 존재하는 영역을 선정하기 위하여 프레임 중심을 기준으로 절반을 차지하는 영역에서 블록 안의 픽셀들에 대한 평균편차를 계산하였다. 탐색영역 안에서 블록은 X방향과 Y방향으로 각각 5픽셀과 10픽셀씩 이동을 하면서 블록 안의 픽셀들에 대한 편차를 계산하고 그 편차 값이 가장 큰 위치를 선정하여 블록매칭을 실행할 위치로 선정하게 하였다.

3.3 모션벡터의 방향성 고려

본 논문에서 구현한 블록매칭 모자이크 시스템은

블록간의 모션벡터의 방향성을 고려하였다. 카메라 이동의 방향성이 오른쪽일 경우 프레임간 사물의 이동은 왼쪽의 방향성을 갖게 되고 현재 프레임에 기준하여 이전 프레임에 대한 모션벡터도 왼쪽의 방향성을 갖게 된다. 따라서, 현재 프레임에 대응되는 블록의 대응점을 찾기 위하여 지정된 블록 안에서 이전의 모션벡터의 방향을 찾아서 더욱 많은 탐색점을 위치시키고 그 영역에서 정밀 탐색하는 방법을 적용하여 모자이크에 필요한 대응점을 찾는 데 존재하는 오차를 줄이고 모자이크의 정확성을 높이고자 하였다. 그림 10은 카메라 이동에 따른 블록매칭의 방향성 고려에 대한 개념도이다.

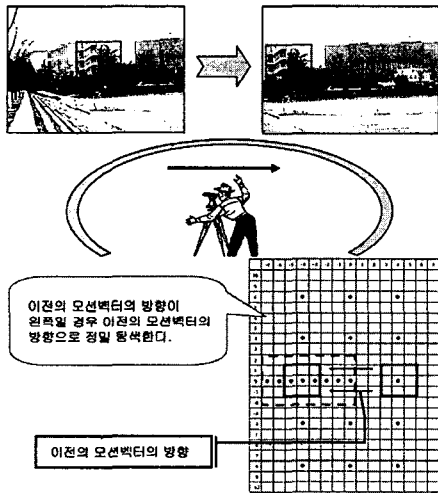


그림 10. 블록매칭의 방향성
Fig. 10 Direction of block matching

IV. 실험 및 성능 평가 분석

본 논문에서는 일반적인 비디오카메라를 이용하여 학교주변 건물과 배경을 촬영하였고 전역탐색 알고리즘과, K-단계 탐색 알고리즘, 본 논문에서 제안한 방향성을 고려한 3-단계 탐색 알고리즘을 이용하여 블록매칭을 수행하였다. 실험 환경은 표 2와 같다.

표 2. 실험 환경
Table. 2 Environment of experimentation

실험 환경		
동영상 촬영환경	촬영장치	일반적인 비디오카메라
	촬영대상	원광대학교 캠퍼스 건물과 주위배경
테스트 프레임	프레임 수	AVI 파일별 100~500 Frames
	크기 및 칼라	320×240 픽셀, 24bit RGB
	비교블록 크기	20×14 픽셀
처리 시스템	운영체제	Window XP Professional
	CPU	Pentium 4. 1.7GHz
	메모리	256 MByte
구현 도구	인터페이스	Open GL
	알고리즘 구현	Visual C++ 6.0

그림 11은 본 논문에서 모자이크 시스템을 구현하는데 있어서 고려한 블록매칭의 방향성에 대한 실험 결과이다.

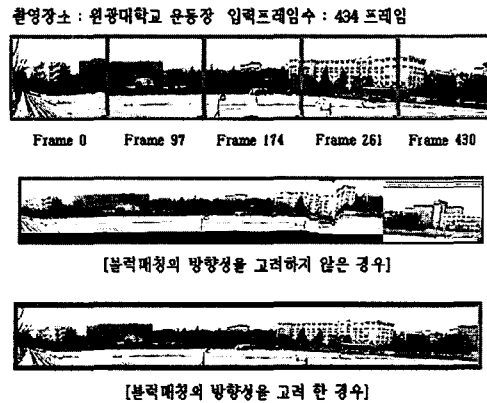


그림 11. 방향성을 고려한 모자이크 실험 결과
Fig. 11 Result images of 3-step block matching using directional search

위의 결과에서도 알 수 있듯이 모션벡터의 방향성을 고려하지 않고 모자이크된 결과이미지는 모자이크 처리의 정확성이 어느정도 유지되다가 마지막 부분에서 정확한 대응점을 찾지 못하여 오차가 발생한 것을 알 수 있다. 이러한 원인은 카메라의 흔들림이나 촬영속도의 변화로 정확한 대응점을 찾지 못하고 잘못된 대응점을 선정했기 때문이다. 이에 반하여 모션벡터의 방향성을 고려하여 모자

익된 결과이미지는 현재 이미지에서 이전의 이미지와 대응되는 위치를 찾기 위하여 이전의 모션벡터의 방향으로 더욱 많은 탐색점을 배치해 정밀탐색 함으로써 오차가 발생할 수 있는 가능성을 줄였기 때문에 같은 조건에서도 정확한 모자이크 처리 결과를 나타내고 있음을 알 수 있다.

그림 12, 13, 14는 전역탐색 방법과 K-단계 탐색 방법, 그리고 본 논문에서 제안한 3-단계 탐색 방법의 블록매칭을 이용하여 모자이크된 출력이미지와 이들의 처리속도를 성능 평가한 그래프이다.

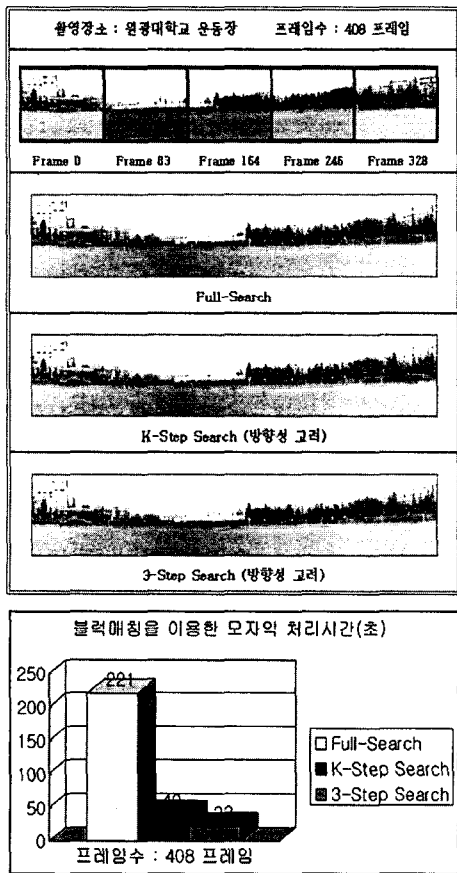


그림 12. 테스트 1 모자이크 결과 및 처리시간
Fig. 12 Test 1 Result images of mosaic and CPU times

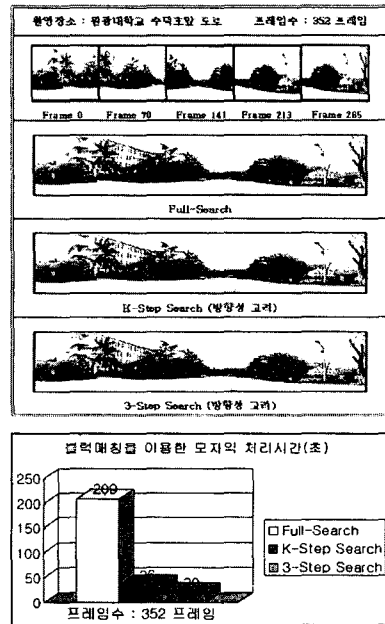


그림 13. 테스트 2 모자이크 결과 및 처리시간
Fig. 13 Test 2 Result images of mosaic and CPU times

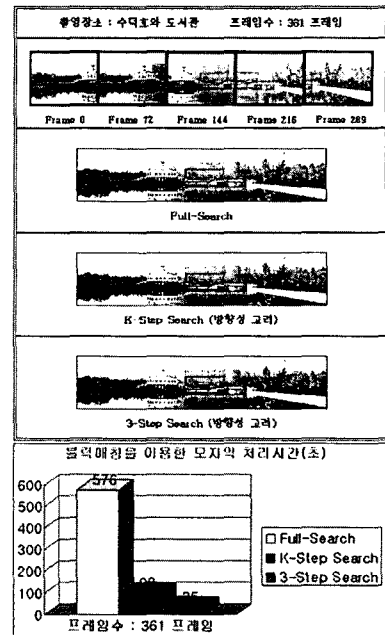


그림 14. 테스트 3 모자이크 결과 및 처리시간
Fig. 14 Test 3 Result images of mosaic and CPU times

테스트 결과에서 알 수 있듯이 블록매칭의 전역탐색 방법과 방향성을 고려한 K-단계 탐색, 그리고 본 논문에서 제안한 방향성을 고려한 3-단계 탐색 방법을 이용하여 비디오 시퀀스의 이미지 모자이크를 수행했을 경우, 모자이크 처리 결과의 정확성은 세 가지 방법 모두 동일한 결과가 나온 것을 알 수 있다. 하지만 모자이크 처리에 소요되는 평균 처리 시간은 전역탐색 방법은 335초, 방향성을 고려한 K-단계 탐색 방법은 58초, 그리고 본 논문에서 제안하는 방향성을 고려한 3-단계 탐색 방법은 25초가 소요된 것을 알 수 있다.

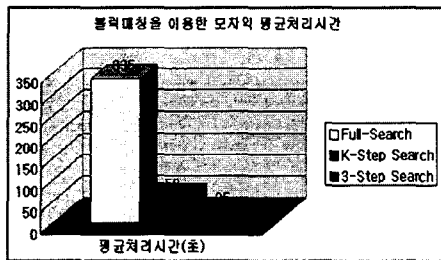


그림 15. 블록매칭을 이용한 모자이크의 평균 처리시간
Fig. 15 Average CPU times of block matching methods

V. 결론

본 논문에서는 블록매칭을 이용한 비디오 시퀀스의 이미지 모자이크 기법에 대하여 설계하고 구현하였다. 모자이크를 위한 이미지는 일반적으로 쉽게 사용할 수 있는 비디오카메라를 이용하여 단 한번의 촬영에 의해 생성된 연속 프레임 이미지를 사용하였고, 프레임 이미지간의 중복된 영역에 일정한 크기의 블록을 지정하고 그 블록 안에서 3단계의 픽셀값 비교를 통하여 대응점을 찾는 3-단계 탐색 알고리즘을 제안하고 구현하였다. 또한 카메라의 이동에 따른 방향성을 고려하여 모션벡터의 방향에 대하여 더욱 많은 탐색점을 정렬시켜 정밀 탐색 함으로써 모자이크의 정확성을 높였다. 전역탐색, K-단계 탐색, 그리고 본 논문에서 제안한 3-단계 탐색 방법을 이용하여 실험한 결과, 방향성을 고려한 3-단계 탐색 블록매칭 방법이 전역탐색 방법과 K-단계 탐색 방법에 비하여 정확한 대

응점을 찾는 데 있어서 불필요한 픽셀 값들의 계산을 피할 수 있었고 전체적인 모자이크 처리 시간을 단축시킬 수 있었다. 근래에 디지털 영상장치의 발달과 대중화로 파노라마 영상의 모자이크 기법에 관한 관심이 증대되고 있다. 본 논문에서 제시한 비디오 시퀀스의 3-단계 탐색 방법은 가상현실과 의료영상, 관광산업, 게임 등, 디지털 영상장치의 활용 영역에서 다양하게 적용이 가능 할 것으로 생각된다. 향후, 좀 더 향상된 모자이크 기법을 위해서 연속적으로 생성되는 프레임 수를 줄여서 처리 속도를 더욱 단축시키는 방법과 카메라의 흔들림에 의한 대응점 탐색의 오류에 더욱 강한 모자이크 방법에 대한 연구를 계속 진행 할 계획이다.

참고 문헌

- [1] S.E, Chen, "QuickTime VR - An image based approach to virtual environment navigation," (SIGGRAPH'95), pp.29-39, August 1995.
- [2] Pohtovista panorama,"http://www.roxio.com"
- [3] Shum and Szeliski, "Panoramic Image Mosaic," Microsoft research, 1997.
- [4] Szeliski, Richard. "Video Mosaics for virtual Environment," IEEE CG&A, March 1996, pp. 22-30.
- [5] R.Szeliski and H.YShum, "Creating full view panoramic image mosaic and environment maps," Proc. SIGGRAPH'97, 1997.
- [6] J. Meehan, "Panoramic photography," Wast-on-Guptill, 1990.
- [7] S.Nayer, "Catadioptric omnidirectional camera," (CVPR'97), pp. 482-488, San Juan, Puerto Rico, June, 1997.
- [8] Tom R. Halfhill. See you around. Byte Magazine, pp. 85-90, May 1995.
- [9] Arun Krishnan and Narendra Ahuja. Panoramic image acquisition. In IEEE Conference on Computer Vision and Pattern Recognition, pp 379-384, San Fransisco, California, June 1996.
- [10] Steve Mann and Rosalind Picard. Virtual bellows: Constructing high quality stills from video. In First IEEE International Conference on Image Processing, Austin, Texas, November 1994.

- [11] Brandon Itkowitz, "<http://cas00.bu.edu/course/cs585/P4/ickey/p4.html>", 1998.
- [12] Szeliski and Sing Bing Kang "Direct Method for scene Reconstruction" In IEEE workshop, June 24, 1995.
- [13] S.Gumustekin R. W. Hall Mosaic image generation on flattened gaussian sphere. In Proc. of IEEE Workshop on Application of Computer Vision, pp 50-55, 1996.
- [14] "<http://www.image.cityu.edu.hk/~ckcheung/thesis/node19.html>"
- [15] S. Kappagantula, and K. R. Rao, "Motion compensated interface image prediction," IEEE Trans. Communication, vol. COM-33, pp. 1011-1015, 1985.
- [16] H. Gharavi, and M. Mills, "Block matching motion estimation algorithm - new results," IEEE Trans. Circuits Syst, pp. 649-651, 1990.
- [17] J. R. Jain and A. K. Jain, Displacement measurement and its application in inter-frame image coding, IEEE Trans. Commun, vol. Com-29, No.12, pp.1799-1808, Dec. 1981.
- [18] K. R. Rao and J. J Hwang, Techniques and Standards for image, Video and Audio Coding. Englewood Cliffs, NJ:Prentice Hall, 1996.



정성태(Sung-Tae Jung)

1989년 2월 서울대학교 공학석사
1994년 8월 서울대학교 공학박사
1995년 3월~현재 원광대학교 전
기전자 및 정보공학부 교수

※ 관심분야 : 영상처리, 컴퓨터 그래픽스

저자 소개



이지근(Chi-Geun Lee)

2000년 원광대학교 컴퓨터공학과
학사
2002년 원광대학교 컴퓨터공학과
석사

2002~현재 원광대학교 컴퓨터공학과 박사과정

※ 관심분야 : 영상처리, 영상인식, 컴퓨터 그래픽스