논문 2010-5-3

적분 영상을 이용한 고속 비디오 안정화 기법

Fast Video Stabilization Method Using Integral Image

권영만*, 임명재**, 오병훈***

Young-Man Kwon*, Myung-Jae Lim, Byung-Hun Oh*****

요 약 본 논문에서는 적분 영상(Integral Image)을 이용하여 고속 비디오 안정화를 수행하는 새로운 기법을 제안한다. 제안된 기법에서는 매 프레임마다 적분 영상을 생성하고, 생성된 적분 영상에서 영상의 블록 움직임의 정합을 평가하여 지역 움직임을 추정한다. 이를 사용해서 전역 움직임을 추정하여 보정한다. 제안된 기법의 효율성을 평가하기 실험 데이터를 다양한 패턴으로 직접 제작하였고, 기존 안정화 알고리즘과의 떨림 보정과 수행 시간을 평가하였다. 실험 결과에서 기존 제안한 기법이 기존 방식들에 비해서 수행 시간이 빠르고 떨림을 검출하여 보정함을 확인하였다.

Abstract We proposed a new technique to perform fast video stabilization using integral image in this article. In the proposed technique, it evaluate local and global motion by the block matching using the generated integral image for each frame and compensate the motion like jitter. We made the various experimental jitter patterns to evaluate the effectiveness of the proposed technique and evaluated stabilization capability and execution time with the existing ones. Through the experiment, we found that the execution time of proposed technique was faster than that of existing techniques and the compensation of jitter was well done.

Key Words : Image Stabilization, Integral Image, Jitter, Motion Estimation, Ego-motion.

Ⅰ. 서 론

최근 영상 획득 장치의 발전에 따라 고품질 영상의 획득에 대한 사용자의 요구 사항들이 증가하고 있다. 특히 영상 획득과정에서 캠코더의 불필요한 움직임이나 사용자의 손떨림은 영상의 질을 떨어뜨리는 큰 원인들 중의 하나인데, 이를 해결하기 위해 다양한 영상 안정화 기술들이 제안되고 있다. 이러한 기술들은 블록 정합 방법과 특징 점 추적 방법으로 구별할 수 있다.

블록 정합 방법[1-3]의 대표적인 방법은 그레이코드(Gray-Code) 비트 평면을 이용하는 방법이 있다. 그레이코드 비트 평면 정합 알고리즘[1]에서 사용되는 논리 연산은 간단한 회로로 구현되기 때문에 지역적인 움직임 추정에 필요한 논리를 하드웨어로 용이하게 구현할 수 있으며, 이로 인하여 디지털 영상 안정화의 전체 시스템 구조를 간단하게 구현할 수 있다.

특징 점 추적 알고리즘은[4] 연속된 두 프레임에서 동일한 특징 점들을 추적하여 서로 정합하고 이를 사용해서 모델 매개 변수 값을 구한다. 이는 특징 점들을 찾아서 움직임 벡터를 추출하기 때문에 연산 속도가 느리고, 잘못된 특징점이 선택되었을 경우 오히려 더 성능이 안 좋게 나올 수도 있다.

본 연구에서는 적분 영상을 사용해서 실시간으로도 영상을 안정화 할 수 있는 알고리즘을 제안한다. 제안된 알고리즘은 매 프레임마다 적분 영상을 생성하고, 생성된 적분 영상에서 영상의 일부분 즉 블록 움직임의 정합을 평가하여 지역 움직임을 추정하며, 이를 사용해서 전역 움직임을 추정하여 보정한다. 이 방법이 실시간으로

*종신회원, 을지대학교 의료전산학부
**종신회원, 을지대학교 의료전산학부(교신저자)
***준회원, 을지대학교 의료전산학부
접수일자 2010.9.14 수정일자 2010.10.8
게재확정일자 2010.10.15

사용할 수 있는 이유는 적분 영상을 생성하기 위한 시간은 $O(n)$이기 때문이다.

제안한 알고리즘의 효율성을 평가하는 방법이 필요하다. 이를 위해서 2개의 동영상을 제작하였다. 첫 번째 동영상은 고정된 위치에서 카메라의 떨림 움직임만 있는 경우이고 두 번째 동영상은 패닝(panning)과 떨림 움직임이 같이 있는 경우이다. 즉, 두 개의 시험 동영상에는 회전 움직임은 없다. 이 2개의 동영상을 가지고 실험한 결과 떨림 움직임만 있는 경우와 패닝과 떨림 움직임이 있는 경우 둘 다 정확히 움직임 벡터를 검출하여 보정되어지는 것을 얻을 수 있었다. 아울러 제안된 알고리즘은 기존 알고리즘보다 연산 양이 적어 수행시간이 짧다. 즉, 이 실험은 빠른 수행 속도로 실시간에도 적합한 알고리즘인 것을 검증하였다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 기존의 디지털 영상 안정화와 관련된 기술을 요약하고, 그 특징과 문제점을 서술한다. Ⅲ장에서는 알고리즘을 제안하고, Ⅳ장에서는 실험을 통해 제안한 알고리즘의 성능을 기존 기술과 비교 및 평가한다. 마지막으로 Ⅴ장에서는 결론과 향후 연구 과제를 제시한다.

Ⅱ. 관련 연구

1. 블록 정합(Block Matching) 방법

그레이코드 평면 정합 기법에서는 먼저 그레이 영상을 그레이 코드(Grade-coded) 영상으로 변환한 후에 특정 비트 평면의 이진 영상을 이용해서 정합한다. 이는 그레이 코드 영상은 밝기 값이 연속적인 경우에 오직 하나의 비트 위치에서만 값이 다른 특징을 가지고 있기 때문이다. 따라서 밝기 값이 작게 변화하면 모든 k 번째 비트 평면에 거의 영향을 주지 않는다.

그레이 영상의 밝기 값을 식 1과 같이 이진수로 나타낼 수 있다. 식 1에서 $f^t(i,j)$은 시간 t에서 프레임내의 (i,j)번째 화소를 의미하고, $b_k(0 \le k \le K-1)$는 화소의 밝기 값을 이진수로 표현하였을 때의 k번째 비트의 값이다.

$$f^t(i,j) = b_{K-1}2^{K-1} \cdots + b_k 2^k + \cdots + b_1 2^1 + b_0 2^0 \quad (1)$$

그레이 코드 영상은 그레이 영상의 각 비트를 수식 2와 같이 사용하여 생성할 수 있다. 각 화소를 이루는 K 비트의 데이터들 중에서 동일한 위치의 이진 값 g_k들만을 모아서 하나의 이진 영상을 생성할 수 있는데, 이를 그레이 코드 비트 평면 영상이라고 한다.

$$\begin{aligned} g_i &= b_i \oplus b_{i+1} \qquad 0 \le i \le K-2 \\ g_{K-1} &= b_{K-1} \end{aligned} \quad (2)$$

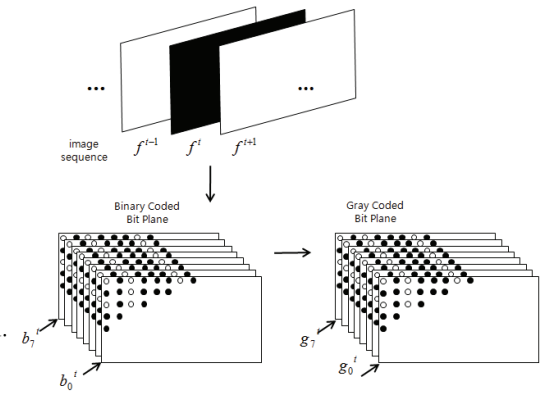


그림 1. 그레이 코드 비트 평면
Fig. 1. Grade Coded Bit-Plane

이 기법에서는 그레이 코드 비트 평면의 크기가 MxN인 블록(Subimage)을 정의하고 이 블록의 상관 계수(Correlation)를 식 3과 같이 정의하여 움직임을 계산한다. 이 값이 가장 적은 경우가 정합이 된 곳이다.

$$\begin{aligned} C_j(m,n) &= \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} g_k^t(x,y) \oplus g_k^{t-1}(x+m, y+n) \\ where,\ -p \le m,n \le p \end{aligned} \quad (3)$$

이 기법에서 정합을 위해서 어떤 특정 비트 평면을 사용해야 하는가에 대한 문제도 다루었다. 그레이 코드 평면 영상을 생성하기 위해서 병렬 처리 구조를 채택했음에도 불구하고 시간이 많이 소요된다는 단점이 있다.

이 알고리즘의 변형된 기법이 존재하는데, 정합 검색 시간을 빨리하기 위한 방법 3-step 그레이 코드 비트 평면 검색 알고리즘이 있다[2].

또한, 이진 영상을 사용해서 이동과 회전을 보정하여 안정화를 이룬 연구도 있다[3]. 이 기법에서 사용된 알고리즘의 전반적인 구조는 그림 3과 같다. 먼저 영상의 밝기 값을 이진화를 하고, 움직임을 추정한다. 그런 후에 떨림(Jitter)을 필터링하고 보정한다.

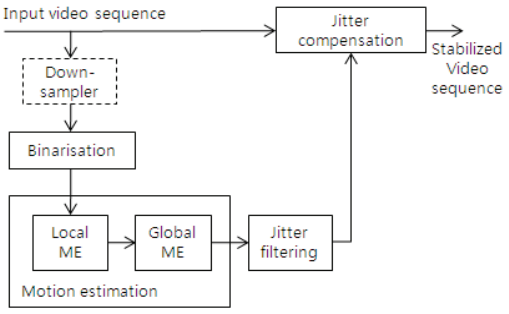


그림 2. 알고리즘 구조
Fig. 2. Algorithm architecture

특히, 이 기법에서 이동과 회전 움직임을 평가하기 위해서 식 4와 같은 모델을 사용하였다. (Xc, Yc)는 현재 영상의 점이고 (Xp, Yp)는 이전 영상에서의 대응하는 점이다. 이동 움직임은 (Tx, Ty)로, 회전 움직임은 회전각 Θ로, 줌(Zoom) 움직임은 크기 요소인 r로 모델링하였다. 그런 후에 LMS(Least Mean Square) 방법을 사용하여 매개 변수의 값을 찾았다.

$$\begin{pmatrix} X_c \\ Y_c \end{pmatrix} = r\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} X_p \\ Y_p \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \end{pmatrix} \quad (4)$$

움직임을 필터링하기 위해서는 이전에 연구된 다른 많은 논문에서는 감쇄(Damping) 상수 δ를 사용한 집적 움직임 벡터(Integrated Motion Vector)를 사용하였으며 이는 식 5와 같다. 이 식에서 IMVt와 IMVt-1은 현재 영상과 이전 영상에서의 집적 움직임 벡터이고 GMVt는 현재 영상에서 전역 움직임 벡터(Global Motion Vector)이다.

$$IMV_t = \delta \times IMV_{t-1} + GMV_t \quad (5)$$

특히 이 연구에서는 적응 감쇄 상수 δ를 식 6과 같이 적응적으로 사용하였고 이 값은 마지막 2개의 전역 움직임 벡터의 합에 의존한다. 만일 마지막 2개의 값의 합이 작으면 정지된 카메라 영상으로 간주하여 감쇄 상수의 값을 크게 설정하였다. 반면에 GMV 합의 값이 크면 패닝과 같은 의도적인 움직임으로 간주하여 감쇄 상수의 값을 작게 설정하였다. 이 필터링은 GMV의 x와 y 성분에 대해서 독립적으로 적용하였다.

$$IMV_t = \delta_{(GMV_t + GMV_{t-1})} \times IMV_{t-1} + GMV_t \quad (6)$$

2. 특징 점 추적(Tracking) 방법

이 기법에서는 영상에서 특징 점을 검출하고 검출된 특징 점의 정합을 사용해서 움직임을 평가한다. 특징 점은 Lowe가 제안한 SIFT(Scale-Invariant Feature Transform) 기법[5]을 사용하며, 이는 영상의 크기, 회전 등에 불변인 특징을 가지고 있다.

SIFT 특징 점을 추출하는 기법은 크게 4가지 단계로 되어 있으며, 첫 번째 영상 크기 공간(Scale-space) 구성하여 크기와 회전에 강인한 극점(Extrema)을 찾는다. 극점들 중에서 지역화(Localization)와 필터링을 통해서 키포인트(Keypoint)를 찾는다. 그런 후에 키포인트의 크기와 방향을 찾아서 이 특성을 포함하는 특징을 키포인트의 서술자(Descriptor)로 사용한다.

이 기법에서도 이동과 회전 움직임을 평가하기 위해서 식 4와 같은 모델을 사용하였고, 특징 점들의 누적 오류를 계산하고 이 값을 사용해서 필터링 문턱치 값을 계산하는 데에 사용하였다. 또한 필터링을 위해서는 식 6을 사용하였으며, 현재 영상에서 보상은 식 7을 사용하였다.

$$C_t = IMV_t - IMV_{t-1} \quad (7)$$

알고리즘의 성능을 평가하기 위해서는 이전에 연구된 다른 많은 논문에서처럼 PSNR 값을 사용하였으며 이는 식 8과 같다.

$$PNSR_t = 10\log_{10}\frac{I_{MAX}^2}{MSE_t} \quad (8)$$

여기서 MSE(Mean Square Error)는 연속적인 프레임 영상 사이의 모든 픽셀들에 대해서 밝기 값 차이의 제곱을 더하고 이를 픽셀 수로 나눈 값이며, 식 9와 같다. H와 W는 영상의 세로 방향과 가로 방향의 크기이고, I_{MAX}은 픽셀이 가질 수 있는 가장 큰 밝기 값이다.

$$MSE_t = \frac{\sum_{y=1,x=1}^{H,W} diff^2(x,y)}{H \times W} \quad (9)$$

또한, 다른 연구에서처럼 알고리즘의 성능을 평가하기 위해서 ITF(Interframe Transformation Fidelity)를 사용하였으며 이는 식 10과 같다. 여기서 N_{frame}은 비디오 파일의 총 프레임 수이다.

$$\mathrm{ITF} = \frac{1}{\mathrm{N_{frame}} - 1} \sum_{\mathrm{k}=1}^{\mathrm{N_{frame}}-1} \mathrm{PSNR(k)} \qquad (10)$$

3. 적분 영상(Integral Image)

적분 영상은 얼굴 특징을 추출하기 위해서 직사각형(블록) 특징을 빨리 계산하기 위해서 영상 표현의 중간 과정으로 사용되었다[6]. 적분 영상에서 영상 좌표 (x, y) 위치에서의 값은 x 왼쪽에 있는 모든 화소들과 y 위쪽에 있는 모든 화소들의 값을 합한 값을 가지고 있다. 즉 이를 수식으로 표현하면 식 11과 같다. 여기서 ii(x, y)는 적분 영상이고 i(x, y)는 원래 영상이다. 영상의 크기가 $H \times W$이면, 적분 영상 크기는 $(H+1)\times(W+1)$ 이어야 한다. 적분 영상의 크기가 더 큰 이유는 연산을 효과적으로 하기 위하여 x축과 y축에 0 값을 갖는 버퍼 영역을 만들기 때문이다.

$$ii(x,y) = \sum_{y'=0}^{y} \sum_{x'=0}^{x} i(x', y') \qquad (11)$$

적분 영상이 주어지면 직사각형(블록) 내에 있는 화소들의 합을 매우 효율적으로 계산할 수 있다. 그림 3을 생각해 보자. 그림에서 D블록 내에 있는 화소 값들의 합은 식 12와 같이 계산될 수 있다. 이 식에서 마지막 항을 더한 이유는 A블록 즉 ii(x1, y1) 영역이 2번 빼기 연산이 되었기 때문이다.

$$S_{block}(x_1,y_1,x_2,y_2) = \sum_{y=y_1}^{y_2} \sum_{x=x_1}^{x2} i(x,y)$$
$$= ii(x_2,y_2) - ii(x_1,y_2) - ii(x_2,y_1) + ii(x_1,y_1) \qquad (12)$$

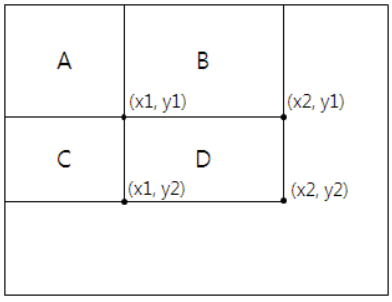


그림 3. 적분 영상에서 블록 내에 있는 화소 값들의 합
Fig. 3. Summation of Pixel Values within Block In Integral Image.

식 12에서 알 수 있듯이 적분 영상을 사용하면 블록 내에 있는 화소 값들의 합을 구하는 시간은 블록 크기와 상관없이 항상 일정하기 때문에 계산 시간이 매우 효율적이다. 즉 집적 영상을 사용하면 블록 정합을 계산할 때에 사용하는 SAD와 같은 값들을 매우 빨리 계산할 수 있다.

Ⅲ. 제안한 알고리즘

본 연구에서 제안한 알고리즘은 4 단계로 구분할 수 있으며 이를 그림 4에 나타낸다.

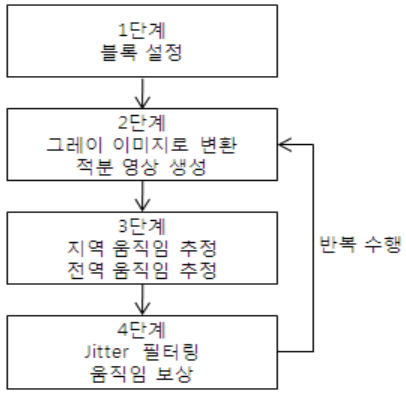


그림 4. 제안한 알고리즘
Fig. 4. Proposed Algorithm

1 단계는 블록을 설정하는 단계이다. 그림 5와 같이 입력된 영상을 5x5로 나눈 후에 내부의 9개의 블록만을 사용한다. 이는 3단계에서 설정된 블록들을 사용해서 지역적인 움직임을 추정할 때에 대응하는 블록을 탐색하게 된다. 이 때 영상의 가장 자리 부근에 있는 블록들은 대응하는 탐색 블록의 일부분이 존재하지 않을 수 있기 때문이다. 따라서 지역적인 움직임을 추정할 때에는 내부에 있는 블록들만 사용한다.

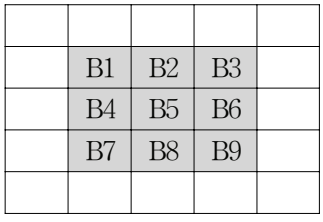


그림 5. 블록 설정
Fig. 5. Setting Block

2단계에서는 먼저 다음 프레임 영상을 읽고, 칼라 영상인 경우에는 추가적으로 그레이 영상으로 변환한다. 그런 후에 3단계의 처리 속도를 빠르게 하기 위해서 적분 영상을 생성하는 단계이다.

3단계에서는 현재 영상의 블록과 이전 영상의 블록 정합 정도를 사용해서 움직임을 추정한다. 이 때 정합되는 정도는 식 13을 사용해서 수치화를 하며 이 값을 블록 SAD(Sum of Absolute Difference)라고 한다. SAD 값은 작은 것이 정합이 잘 된다는 것을 의미한다.

$$SAD_{block} = \sum_{y=1}^{N}\sum_{x=1}^{M} |I^{t-1}(x,y) - I^{t}(x,y)| \tag{13}$$

현재 영상의 블록들이 이전 영상의 어느 위치에서부터 움직였는지를 알기 위해서 이전 영상의 영역을 탐색하게 된다. 이 때 탐색하는 범위를 그림 6에 나타낸다. 즉 $M \times N$ 크기의 블록을 수평 방향으로 $\pm p$ 화소만큼, 수직 방향으로 $\pm q$만큼의 범위에서 탐색한다.

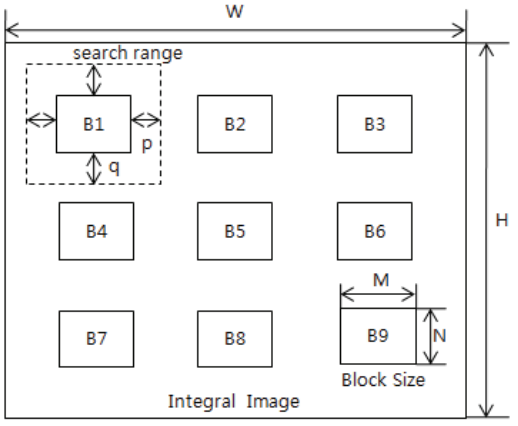


그림 6. 블록의 움직임 탐색 범위
Fig. 6 Search Range For Block Movement

지역적인 움직임은 현재 영상의 블록이 이전 영상의 탐색 범위에 있는 모든 탐색 블록들 중에서 최소 SAD 값을 가지는 위치에서 지금 위치로 움직였다는 것으로 추정한다. 따라서 탐색 범위에 있는 모든 픽셀에 대해서 식 13을 $(2p+1)\times(2q+1)$번 계산해야 한다. 만일 적분 영상의 식 12를 사용하면 식 13의 SAD 계산은 식 14와 같이 변경될 수 있다.

$$\begin{aligned} SAD_{block} &= \sum_{y=1}^{N}\sum_{x=1}^{M} |I^{t-1}(x,y) - I^{t}(x,y)| \\ &= \left| \sum_{y=1}^{N}\sum_{x=1}^{M} I^{t-1}(x,y) - \sum_{y=1}^{N}\sum_{x=1}^{M} I^{t}(x,y) \right| \\ &= \left| \begin{array}{l} ii^{t-1}(x_2,y_2) - ii^{t-1}(x_1,y_2) \\ \quad - ii^{t-1}(x_2,y_1) + ii^{t-1}(x_1,y_1) \\ - \begin{pmatrix} ii^{t} & (x_2,y_2) - ii^{t} & (x_1,y_2) \\ - ii^{t} & (x_2,y_1) + ii^{t} & (x_1,y_1) \end{pmatrix} \end{array} \right| \end{aligned} \tag{14}$$

본 연구에서는 그레이 영상 대신에 적분 영상을 사용해서 SAD를 계산한다. 만일 이전의 그레이 영상을 사용해서 계산하면 $(2p+1)\times(2q+1)\times(M\times N)$의 가감산 연산이 필요하다. 하지만 이전의 적분 영상을 사용해서 계산하면 $(2p+1)\times(2q+1)\times(4)$번의 가감산 연산이 필요하다. 즉 속도를 빨리할 수 있다.

지역적인 블록의 이동 움직임 벡터는 수평 및 수직 방향으로 독립적으로 이동한 양을 픽셀 단위로 추정한다. 전역적인 이동 움직임 값은 추정한 지역적인 블록의 이동 움직임의 평균값을 사용한다. 이를 전역 움직임 벡터(GMV : Global Motion Vector)라고 한다.

4단계에서는 먼저 흔들림을 필터링하고 집적 움직임 벡터를 구하는데 본 연구에서는 식 6를 사용하였다. 즉 이전의 집적 움직임 벡터에 감쇄 상수를 곱하고 현재의 전역 움직임 벡터를 더하여 현재의 집적 움직임 벡터를 구한다. 그런 후에 식 7을 사용하여 보상할 값은 구하고 이를 사용해서 보상한다.

Ⅳ. 실험 및 결과 분석

본 연구에서는 알고리즘이 정확하게 동작하는 것을 분석하기 위해서 실험 데이터를 생성하여 사용하였다. 또한 이 실험 데이터를 사용해서 제안한 알고리즘의 계산 속도도 측정하였다.

1. 실험 데이터 패턴 및 생성

실험 데이터를 생성하기 전에 먼저 손 떨림과 같은 떨림(Jitter)을 표 1과 같은 패턴으로 정의하였다. 그런 후에 먼저 떨림만 있는 실험 데이터를 만들었으며, 실험 데이터의 이름은 Jitter23과 같은 방식으로 명명하였다. 이 이

름에서 앞에 있은 숫자 2는 패턴을 뜻하고 뒤에 있는 숫자 3은 움직인 픽셀 수를 뜻한다.

그리고 의도적인 패닝은 그림 7과 동작하도록 하였다. 그런 후에 패닝만 있는 데이터의 이름은 Pan5와 같이 명명하였다. 이 이름에서 숫자 5는 이동할 때에 프레임마다 5 픽셀을 이동하였다는 것을 의미한다. 추가적으로 떨림과 패닝 동작이 함께 있는 파일은 Pan5Jitter23과 같이 명명하여 사용하였다.

실험 데이터는 모두 256 그레이 레벨을 갖고 480 x 272의 공간 해상도를 갖는다. 떨림만 있는 영상과 떨림과 패닝 동작이 함께 있는 영상은 각각 100 프레임과 240 프레임으로 구성하였다. 떨림만 있는 영상은 표1에서처럼 8종류의 패턴을 움직인 픽셀수를 다르게 정의하여 총 24개의 떨림 패턴을 생성하였다. 떨림과 패닝이 함께 있는 영상은 그림 7에 나타난 패닝 동작과 떨림 패턴을 사용하여 총 9개의 움직임 패턴을 생성 하였다.

표 1. 떨림 데이터 패턴
Table 1. Jitter Data Pattern

Pattern	움직임		설명 (m은 움직인 픽셀수)	
Pattern 1	→	←	x=x+m, y=y	x=x−m, y=y
Pattern 2	↑	↓	x=x, y=y−m	x=x, y=y+m
Pattern 3	↗	↙	x=x+m, y=y−m	x=x−m, y=y+m
Pattern 4	↖	↘	x=x−m, y=y−m	x=x+m, y=y+m
Pattern 5	→	↑	← ↓ ↑ →	
	←	↓		
Pattern 6	↗	↖		
	↙	↘		
Pattern 7	→	←	→ ← ← →	
	←	→		
Pattern 8	↑	↓	↑ ↓ ↓ ↑	
	↓	↑		

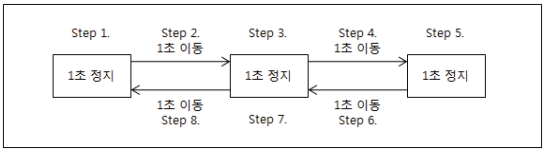


그림 7. 의도적인 패닝 동작
Fig. 7. Intentional Panning Operation

2. 실험 및 평가

먼저 떨림만 있는 데이터를 사용해서 알고리즘의 효율성을 떨림 보정 성능과 수행 시간으로 평가했다. 알고리즘의 떨림 보정 성능 평가는 식 10에서 설명했던 ITF 방법을 사용하였고 그 결과를 표 2에 나타낸다. 또한 Jitter22에 대한 PSNR 그래프를 그림 8에 나타내었다. 그림 8의 결과에서 알 수 있듯이 제안된 알고리즘을 수행해서 일정한 패턴에 의한 떨림을 보정하는 것을 확인할 수 있었다. 그리고 24개의 떨림 패턴을 모두 실험하여 확인하였으며, 모든 떨림 패턴에 대해서 원 데이터 보다 제안된 알고리즘을 수행하고 난 뒤에 떨림이 보정되어 ITF 수치가 높은 것을 확인할 수 있었다. 그 중에서 8개의 떨림 패턴에 대한 ITF 값을 표 2에 나타내었다.

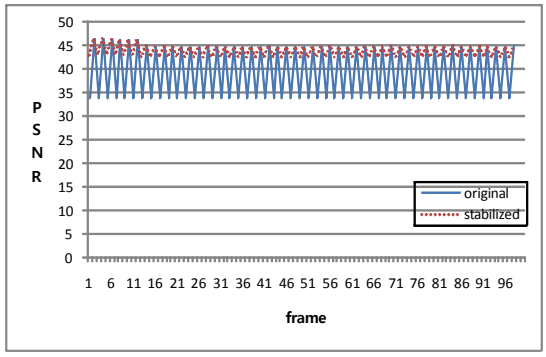


그림 8. PSNR 그래프
Fig. 8. PSNR Graph

표 2. 카메라 떨림의 ITF 평가
Table 2. ITF Evaluation of Camera Jitter

알고리즘 / 데이터	ITF 평가	
	원 데이터	제안한 알고리즘
Jitter13	38.1702	40.6755
Jitter22	39.2647	43.6606
Jitter33	38.6397	41.4560
Jitter41	40.7836	42.0917
Jitter52	36.0520	40.1855
Jitter63	34.3711	38.0053
Jitter71	38.0287	41.3214
Jitter83	38.3254	40.8921

수행 시간 평가는 관련 연구에서 언급한 다른 알고리즘과 비교하여 표 3에 나타내었다. 결과에서 알 수 있듯이 제안한 알고리즘이 다른 2개의 알고리즘과 비교 했을 때, 모든 다양한 떨림 패턴에 대해서 수행 시간이 빠른 것을 볼 수 있었다. 비트 평면을 사용하여 안정화 수행

시간을 빠르게 했던 그레이 코드 비트 평면 알고리즘보다 본 연구에서 제안한 적분 영상을 이용한 안정화 기법의 실행 시간이 더 빠른 것을 알 수 있었다. 즉 적분 영상을 사용하면 수행 시간이 상당히 빠름을 확인할 수 있었다. 결론적으로 제안한 적분 영상을 사용해서 떨림을 보정할 수 있고, 수행 시간도 다른 기법보다 빠르다는 것을 확인하였다.

표 3. 카메라 떨림의 수행시간 평가
Table 3. Running Time Evaluation of Camera Jitter

알고리즘 / 데이터	수행시간 평가(second)		
	그레이 코드 알고리즘	블록 매칭 알고리즘	제안한 알고리즘
Jitter13	12.27686	32.27365	7.94941
Jitter22	12.33111	33.11372	7.86607
Jitter33	12.20691	32.19987	10.93649
Jitter41	11.36069	31.17719	7.53105
Jitter52	12.19982	31.68110	9.86715
Jitter63	12.21117	32.88515	10.67407
Jitter71	12.11441	29.75247	9.05512
Jitter83	12.35116	31.57840	7.60560

표 4와 같이 의도적인 패닝이 존재할 경우에 대해서도 평가를 하였다. 떨림과 패닝이 같이 있을 경우에는 식 6에서 설명된 것과 같이 감쇄 상수를 경우에 따라 다르게 적용하는 적응 기법을 사용하였다. 표 4에서 나타난 것처럼 원 데이터 보다 제안된 알고리즘이 떨림을 보정하는 것을 알 수 있다. 이 경우에도 제안한 알고리즘이 의도적인 패닝을 검출하고, 떨림만을 보정하는 것을 확인할 수 있었다.

표 4. 의도적인 패닝의 ITF 평가
Table 4. ITF Evaluation of Intentional Panning

알고리즘 / 데이터	ITF 평가	
	원 데이터	제안한 알고리즘
Pan5Jitter11	31.9742	33.4267
Pan5Jitter21	32.0212	33.0744
Pan5Jitter22	32.0016	33.6419
Pan5Jitter23	31.9371	32.3717
Pan5Jitter41	31.8562	32.9167
Pan5Jitter42	31.8712	33.3738

V. 결 론

본 논문에서는 적분 영상에 의한 디지털 영상 안정화를 수행하는 새로운 기법을 제안하였다. 제안된 기법에서는 매 프레임마다 적분 영상을 생성하고, 생성된 적분 영상에서 영상의 블록 움직임의 정합을 평가하여 지역 움직임을 추정하며, 이를 사용해서 전역 움직임을 추정하여 보정한다. 제안된 기법의 효율성을 평가하기 실험 데이터를 다양한 떨림 패턴과 의도적인 패닝을 직접 제작하였고, 기존 안정화 알고리즘과의 떨림 보정과 수행 시간 성능을 평가하였다. 실험 결과를 통해서 제안한 기법이 기존 방식들에 비해서 빠른 시간에 떨림을 검출하여 보정할 수 있음을 확인하였다. 또한 의도적인 패닝이 존재할 경우에도 이 기법을 적용할 수 있음을 확인하였다.

참 고 문 헌

[1] Y.M. Yeh, H.C. Chiang, S.J. Wang, “Digital camcorder image stabilizer based on gray-coded bit-plane block matching”, Optical Engineering 40(10), pp.2172-2178, 2001

[2] S.H. Ko, S,.H. Lee, S.W.Jeon, "Fast digital image stabilizer based on gray-coded bit-plane matching", IEEE Trans. on Consumer Electronics, 45(3), 1999

[3] Auberger. S, Miro. C, "Digital video stabilization architecture for low cost device", Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005

[4] Battiato. S, Gallo. G, Puglisi. G. Scellato. S, "SIFT features tracking for video stabilization", 14th International Conference on Image Analysis and Processing, 2007

[5] D. Lowe, “Distinctive image features from scale-invariant keypoints”, International Journal of Computer Vision, Vol. 60(2):91 - -110, 2004.

[6] Viola. P, Jones. M, "Rapid object detection using a boosted cascade of simple features", Accepted Conference on Computer Vision and Pattern Recognition, 2001

저자 소개

권 영 만(종신회원)

- 1985년 2월 : 한국과학기술원 전기 및 전자공학과 석사
- 1998년 8월 : 한국과학기술원 정보 및 통신공학과 박사수료
- 2007년 2월 : 광운대학교 전자공학과 박사
- 1993년 3월~ : 을지대학교 의료IT마케팅학과 교수

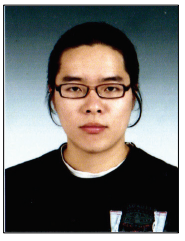
<관심분야> 영상처리, 머신비전, 운영체제

임 명 재(종신회원)

- 1989년 2월 : 중앙대학교 전자계산학과 졸업
- 1991년 2월 : 중앙대학교 컴퓨터공학과 석사
- 1998년 2월 : 중앙대학교 컴퓨터공학과 박사
- 1992년 3월~ : 을지대학교 의료IT마케팅학과 교수

<관심분야: SE 개발방법, HCI, U-Healthcare 등>

오 병 훈 (준회원)



- 2007년 3월~현재 : 을지대학교 의료산업학부 의료전산학전공 학생

<관심분야> 영상처리, 비전, 패턴인식