2014년 12월 전자공학회 논문지 제 51권 제 12 호
Journal of The Institute of Electronics and Information Engineers Vol. 51, NO. 12, December 2014 http://dx.doi.org/10.5573/ieie.2014.51.12.130

논문 2014-51-12-14

카메라 움직임 추정 및 패치 기반 디컨볼루션을 이용한 동영상의 번짐 현상 제거 방법

(Video Deblurring using Camera Motion Estimation and Patch-wise Deconvolution)

정 우 진*, 박 진 욱*, 이 종 민*, 송 태 언**, 최 원 주**, 문 영 식***

(Woojin Jeong, Jin Wook Park, Jong Min Lee, Tae Eun Song, Wonju Choi, and Young Shik Moon©)

요 약

동영상 촬영 시 급격한 카메라의 흔들림은 의도하지 않은 번짐 현상을 발생시켜 동영상의 품질을 낮추는 원인이 된다. 따라서 본 논문에서는 동영상의 품질을 높이기 위해 동영상에서 카메라 흔들림으로 인해 발생한 번짐 현상을 제거하는 방법을 제안한다. 제안하는 방법은 매 프레임 별로 이루어진다. 각 프레임마다 이전 프레임과 현재 프레임, 다음 프레임을 이용하여 카메라 움직임을 계산한다. 그리고 카메라의 움직임을 바탕으로 점 확산 함수를 계산하고 프레임을 패치 단위로 쪼개어 패치별 번짐 현상을 제거한다. 이때 품질을 높이기 위하여 번짐 영상으로부터 외곽선을 예측하는 방법을 사용한다. 번짐 현상이 제거된 패치는 다시 하나의 프레임으로 합한다. 실험 결과를 통해 제안하는 방법이 동영상에서의 카메라 흔들림으로 인한 번짐 현상을 효과적으로 제거함을 확인하였다.

Abstract

Undesired camera shaking can make a blur effect, which causes a degradation of video quality. We propose an efficient method of removing the blur effects on video captured from a single camera. The proposed method has a sequential process that is applied to each frame. The first stage is to estimate the camera motion for each frame. In order to estimate the camera motion, we compute the optical flow using 3 consecutive frames. Then a patch-wise image deconvolution is applied. During the deconvolution, edge prediction is used to improve the quality of image deconvolution. After patch-wise image deconvolution, deblurred patches are integrated into an image to produce a deblurred frame. The above process is performed for each frame. The experimental result shows that the proposed method removes the blur effect efficiently.

Keywords : deblurring, video deblurring, camera motion, motion estimation, patch

Ⅰ. 서 론

동영상을 촬영할 때 급격한 카메라의 흔들림은 동영상에서 번짐(blur) 현상이 발생하는 원인이 된다. 번짐 현상이 발생할 경우 동영상의 품질이 저하되므로 스테디캠(steadicam)과 같이 추가적인 장비를 이용하여 카메라의 흔들림을 방지하는 방법이 이용된다. 그러나 이

* 학생회원, *** 정회원, 한양대학교 컴퓨터공학과
(Dept. of Computer Science & Engineering, Hanyang University)
** 정회원, 삼성탈레스 전자광학체계그룹
(Optronics System Group, Samsung Thales)
© Corresponding Author(E-mail: ysmoon@hanyang.ac.kr)
※ 이 논문은 삼성탈레스의 지원을 받아 연구되었음.
접수일자: 2014년08월12일, 수정일자: 2014년11월20일
게재확정: 2014년11월30일

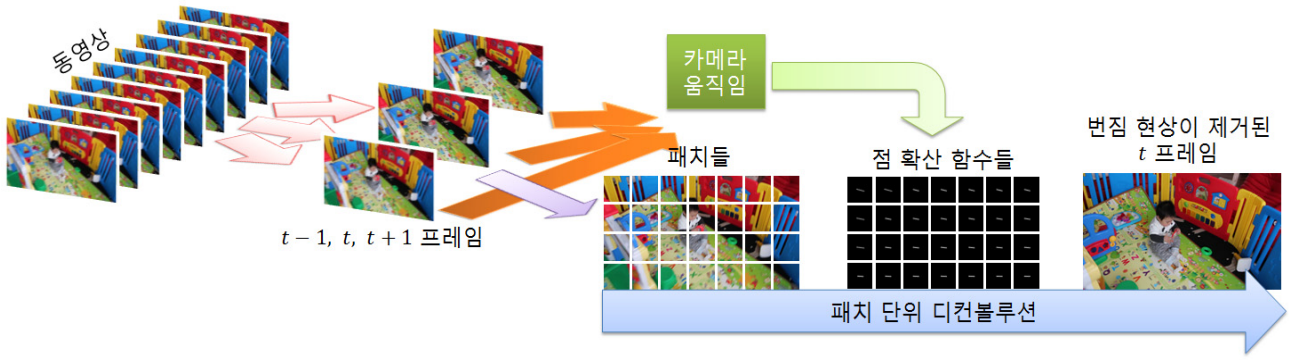

그림 1. 제안하는 알고리즘
Fig. 1. The proposed algorithm.

러한 방법은 추가적인 장비를 사용해야하는 불편함과 가격이 비싸다는 점 등으로 인해 대중적으로 사용할 수 있는 방법은 아니다. 따라서 본 논문에서는 기존의 소프트웨어 기반 번짐 현상 제거(deblurring) 기술과 카메라 움직임 추적 기술을 이용하여 카메라 움직임으로 인한 번짐 현상을 제거하는 기술을 제안한다.

소프트웨어 기반 번짐 현상 제거 기술은 크게 단일 영상을 이용하는 방법과 추가적인 정보를 이용하는 방법으로 나뉜다. 단일 영상을 이용하는 방법은 오래 전부터 연구되어 왔다. Fergus는 번짐 현상이 없는 영상들이 갖는 통계적인 특성을 이용하여 번짐 영상으로부터 선명한 영상을 복원하는 방법을 제시하였으며[1], Shan은 Fergus가 제시한 통계적인 특성을 단순화하고 사후 확률 추정 기반의 방법을 이용한 효과적인 점 확산 함수(point spread function) 추정 방법과 번짐 현상 제거 방법을 제시하였다[2]. 그러나 두 방법은 연산시간이 매우 많이 필요한 단점이 있다. 그래서 Cho는 번짐 영상으로부터 선명한 영상의 외곽선을 예측하고 외곽선 간 비교를 통해 점 확산 함수를 추정하는 방법으로 번짐 현상 제거에 필요한 연산시간을 수십 초 이내로 단축시켰다[3]. Xu는 Cho의 방법이 피사체의 크기가 점 확산 함수의 크기보다 작을 경우 성능이 저하되는 문제점을 발견하고 Cho의 방법을 확장하여 강인한 번짐 현상 제거 방법을 제안하였다[4].

번짐 현상 제거를 위해 추가적인 정보를 이용하는 방법들도 제안되었다. Yuan은 빠르게 촬영된 영상과 번짐 영상을 이용하여 번짐 현상을 제거하였고[5], Son은 Yuan과 비슷한 방법이지만 패치 기반으로 점 확산 함수를 추정하여 실시간으로 번짐 현상을 제거하는 방법을 구현하였다[15]. Joshi는 움직임 센서를 장착한 카메라를 이용하여 촬영 순간 발생하는 카메라의 움직임을 감지하였다. 그리고 카메라의 움직임과 동시에 촬영된 번짐 영상을 선명한 영상으로 복원하였다[6]. Ben-Ezra는 고속 카메라와 일반 카메라를 동시에 사용하는 이종 결합(hybrid) 카메라에 대해 정의하고 이종 결합 카메라를 이용하여 번짐 현상을 제거하는 방법을 제안하였으며[7], Tai는 Ben-Ezra의 방법을 확장하여 카메라의 움직임이 지역적으로 균일하지 않더라도 번짐 현상을 효과적으로 제거하는 방법을 제안하였다[8]. Li는 연속 촬영된 여러 장의 영상으로부터 카메라 움직임, 점 확산 함수, 실제 촬영된 시간 비율을 계산하고 최종적으로 선명한 영상을 추정하는 방법을 제안하였다[9]. Cho는 흔들림이 많은 동영상에서도 선명하게 촬영된 프레임이 있다는 사실에 착안하여 번짐 패치를 선명한 패치들의 가중치 합으로 대체하는 방법을 제안하였다[10].

본 논문에서는 동영상에 발생한 번짐 현상을 소프트웨어만으로 제거하는 방법을 제안한다. 이는 Li와 Cho의 방법과 비슷한 연구 분야이다[9~10]. 그러나 두 방법은 모두 번짐 현상 제거 과정이 복잡하며 구현하기 어렵다는 단점이 있다. 때문에 단순한 연산이 필요한 카메라에서 사용하기에는 적합하지 않다. 본 논문에서는 비교적 단순하면서도 Li와 Cho의 방법만큼 좋은 결과를 갖는 번짐 현상 제거 방법을 제안한다. 제안하는 방법은 크게 두 가지 단계로 구성된다. 첫 번째 단계에서는 카메라 움직임을 계산하여 점 확산 함수를 생성한다. 마지막 단계에서는 외곽선 예측을 이용하여 패치단위 디컨볼루션(deconvolution)을 수행하고 하나의 선명한 프레임을 복원한다.

Ⅱ. 본 론

제안하는 방법은 동영상에서 연속한 프레임들을 이용하여 카메라 움직임을 계산하고 패치 단위로 선명한 영상을 복원한다. 그림 1은 제안하는 방법의 개요를 나타낸다. 우선 연속된 $t-1$, t, $t+1$ 프레임을 이용하여 t 프레임에서의 카메라 움직임을 계산한다. 그리고 t 프레임을 $m \times m$ 크기의 패치로 나눈다. 모든 패치는 외곽선 예측 결과와 패치 각각의 점 확산 함수를 이용하여 디컨볼루션을 수행한 후, 마지막으로 번짐 현상이 제거된 패치들을 하나의 영상으로 합하여 번짐 현상이 제거된 t 번째 프레임을 생성한다.

1. 번짐 모델(Blur Model)

Cho는 동영상에서 프레임 간 카메라의 움직임을 계산하고 각 프레임의 번짐 현상을 추정하기 위한 번짐 모델을 제시하였다[10]. 제안하는 방법은 Cho가 제시한 모델을 일부 수정하여 이용한다.

그림 2는 번짐 모델을 나타낸다. 동영상은 연속된 프레임으로 구성된다. 이때 각 프레임 f는 주기 T 마다 카메라의 조리개가 열려 있는 2τ 동안 촬영 된다. 이때 카메라 움직임은 프레임 간의 어파인 변환(affine transform) $\mathbf{C}$로 모델링 된다. 프레임간의 $\mathbf{C}$는 각 f의 중심부터 중심까지의 움직인 것으로 가정한다. f_t가 촬영되는 2τ 동안의 카메라 움직임 $\mathbf{A}_t$는 어파인 변환이며 $\mathbf{C}_t$와 $\mathbf{C}_{t+1}$의 혼합으로 계산된다.

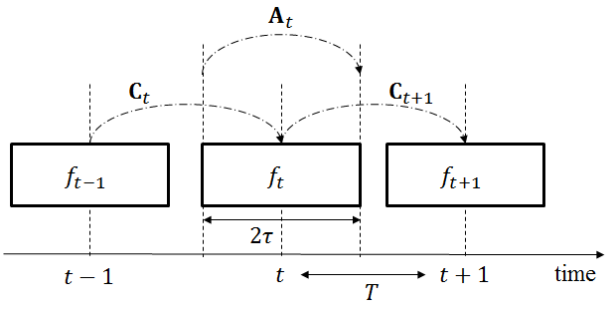

그림 2. 번짐 모델
Fig. 2. Blur model.

2. 움직임 추정(Motion estimation)

이 단계의 목표는 f_t가 촬영될 때의 카메라 움직임 $\mathbf{A}_t$를 계산하는 것이다. $\mathbf{A}_t$는 어파인 변환으로 모델링 되며 $\mathbf{C}_t$와 $\mathbf{C}_{t+1}$를 이용하여 계산한다. 각 $\mathbf{C}$의 계산에는 광류(optical flow)와 RANSAC(RANdom SAmple Consensus)이 이용된다.

가. 광류 계산

광류 계산에는 LK 방법(Lucas-Kanade method)을 이용한다[11]. LK 방법은 광류 계산에 널리 이용되는 방법으로 연속된 2 프레임 간 추적점들의 이동을 추정하는 방법이다. 일반적으로 LK 방법의 추적점 검출에는 FAST(Features from Accelerated Segment Test)나 SIFT(Scale-Invariant Feature Transform)와 같은 특징점 검출 방법이 이용된다[12~13]. 하지만 그림 3과 같이 번짐 영상에서는 특징점 검출에 실패하는 경우가 많다. 또한 동영상에서의 번짐 현상 제거를 위해서는 카메라의 움직임을 계산해야 하는데 이때 영상의 특정 지역을 추적하는 방식의 접근은 적합하지 않다. 따라서 제안하는 방법에서는 영상 전체에 균일한 간격으로 추적점을 설정하고 영상 전체로부터 광류를 계산하였다. 그림 4는 균일하게 설정된 추적점으로부터 광류를 계

(a) (b)

그림 3. FAST의 실패 사례 (a) 선명한 영상에서 FAST 특징점 검출 결과, (b) 번짐 현상 영상에서 FAST 특징점 검출 결과
Fig. 3. A fail case of the FAST. (a) Feature point detection using the FAST on a clear image, (b) Featrue point detection using the FAST on a blur image.

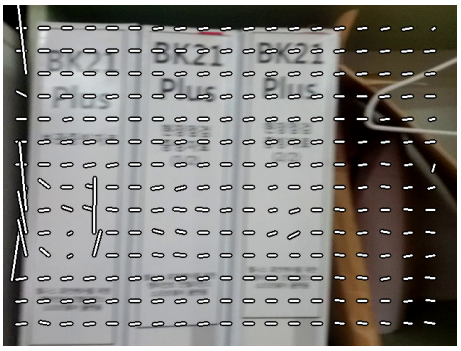

그림 4. 광류 계산 결과
Fig. 4. Computation result of the optical flow.

산한 결과이다. 그림 4의 좌측 부분에 광류 계산에 실패한 결과가 있다. 하지만 다음 절에서 설명하는 RANSAC으로 인해 실패한 추적은 카메라 움직임 추정에서 배제된다.

나. 프레임간 카메라 움직임 계산

RANSAC은 잘못된 값(outlier)이 포함되어 있는 자료에서 올바른 값(inlier)들을 가장 잘 설명하는 모델을 찾아내는 방법이다. 제안하는 방법에서는 RANSAC 알고리즘을 이용하여 광류 쌍들로 부터 어파인 변환으로 모델링된 카메라 움직임 $\mathbf{C}$를 계산하고, 매 단계마다 다음의 과정을 반복하였다. 우선 임의로 3개의 광류 쌍을 선택하여 후보 어파인 모델을 계산한다. 그리고 후보 어파인 모델과 오차가 일정 수준 이하인 광류 쌍을 올바른 값이라고 가정하고 개수를 확인한다. 여러 번의 반복 결과 올바른 값을 가장 많이 갖는 후보 어파인 모델을 $\mathbf{C}$로 선택한다.

다. 프레임 중심에서의 카메라 움직임 계산

t 프레임의 카메라 움직임 $\mathbf{A}_t$는 $\mathbf{C}_t$와 $\mathbf{C}_{t+1}$의 혼합으로 식 (1)과 같이 계산한다.

$$\mathbf{A}_t = \left\{\left(1-\frac{\tau}{T}\right)\mathbf{A}_I + \frac{\tau}{T}\mathbf{C}_{t+1}\right\} \cdot \left\{\left(1-\frac{\tau}{T}\right)\mathbf{A}_I + \frac{\tau}{T}\mathbf{C}_t\right\} \tag{1}$$

$\mathbf{A}_I$는 영상을 변형 시키지 않는 어파인 변환 행렬로 3×3 크기의 단위행렬이다. $\mathbf{A}_t$는 $\mathbf{A}_I$와 $\mathbf{C}_t$의 가중치 합과 $\mathbf{A}_I$와 $\mathbf{C}_{t+1}$의 가중치 합의 곱으로 계산된다. 가중치는 T와 τ의 비율로 계산된다.

3. 패치별 점 확산 함수 생성

번짐 현상 제거 과정은 패치 단위로 이루어진다. 이를 위해 $\mathbf{A}_t$를 이용하여 각각 패치의 점 확산 함수를 생성한다. 패치는 $m\times m$크기로 겹치는 부분 없이 영상 전체를 분할한다.

가. 점 확산 함수 생성

패치 단위 디컨볼루션을 위해서는 각 패치마다 점 확산 함수가 필요하다. 각 패치 중심점의 이동을 추적하여 움직임 벡터를 계산하고, 움직임 벡터를 이용하여 점 확산 함수를 생성한다.

$$\mathbf{d}_p = \mathbf{A}_t\mathbf{c}_p \tag{2}$$

p번째 패치의 중심점 $\mathbf{c}_p$를 $\mathbf{A}_t$로 어파인 변환하면 이동된 중심점 $\mathbf{d}_p$가 계산된다. 두 점을 이용하여 각 패치의 움직임 벡터 $\mathbf{v}_p$를 계산 한다.

$$\mathbf{v}_p = \mathbf{c}_p - \mathbf{d}_p \tag{3}$$

각 패치별로 점 확산 함수을 생성하기 위해 $\mathbf{v}_p$의 길이를 이용하여 1차원 점 확산 함수를 생성하고 $\mathbf{v}_p$의 각도만큼 회전시켜 2차원 점 확산 함수를 생성한다. 식 (4)는 1차원 점 확산 함수이다.

$$k_p(x) = \begin{cases} \dfrac{1}{l_p} & -\dfrac{l_p}{2} < x < \dfrac{l_p}{2} \\ 0 & otherwise \end{cases} \tag{4}$$

l_p는 $\mathbf{v}_p$의 길이를 의미한다. 이후 $k_p()$를 $\mathbf{v}_p$의 방향만큼 회전시켜 2차원 점 확산 함수를 생성한다.

나. 움직임의 속도 가중치 적용

실제 카메라 움직임은 시간에 따라 속도가 변한다. 예를 들어 움직이던 카메라가 멈추는 경우 한 프레임 촬영되는 2τ동안 초반에는 카메라가 빠르게 움직이지만 마지막에는 카메라가 거의 고정될 수 있다. 이로 인해 촬영된 영상의 번짐 현상의 가중치는 달라질 수 있다. 따라서 카메라 움직임의 변화에 따라 점 확산 함수의 가중치를 다르게 적용해야 한다. 그러나 식 (4)와 같이 점 확산 함수를 생성할 경우 실제 카메라 움직임을 올바르게 표현하지 못한다. 식 (5)은 식 (4)를 변형하여 카메라 움직임의 크기에 따라 다른 가중치를 부여한다.

$$k_p(x) = \begin{cases} \dfrac{1}{z_p} w_p(x) & -\dfrac{l_p}{2} < x < \dfrac{l_p}{2} \\ 0 & other \end{cases} \tag{5}$$

$$w_p(x) = \left(1-\frac{x+l_p/2}{l_p}\right)\frac{1}{l_{\mathbf{C}_t}} + \left(\frac{x+l_p/2}{l_p}\right)\frac{1}{l_{\mathbf{C}_{t+1}}} \tag{6}$$

$$z_p = \sum_{x=-l_p/2}^{l_p/2} w_p(x) \tag{7}$$

여기서, $w_p(x)$는 x위치에서의 가중치를 의미하며,

l_{C_t}와 $l_{C_{t+1}}$는 각각 C_t와 C_{t+1}의 이동 성분의 크기이다. 한 위치에서 영상이 촬영되는 시간은 카메라 움직임이 작을수록 길다. 그러므로 점 확산 함수에서 한 점의 가중치는 카메라 움직임의 크기와 반비례 한다. z_p는 정규화를 위한 값으로 가중치의 총 합이다.

4. 패치 RL 디컨볼루션

각각의 패치마다 외곽선 예측을 이용한 가중치 RL 디컨볼루션(Richardson-Lucy deconvolution)을 이용하여 번짐 현상을 제거한다. 기존의 RL 디컨볼루션은 구조가 간단하고 결과의 품질이 우수하기 때문에 많이 쓰이고 있지만, 복원된 영상에 물결 현상(ringing artifact)이 발생하는 문제가 있다. 특히 동영상에 적용할 경우 매 프레임마다 물결 현상이 다르게 나타나므로 RL 디컨볼루션의 단점이 두드러지게 된다. 이 단계에서는 패치 단위 디컨볼루션을 수행한다. 이때 물결 현상을 감소시키기 위해 외곽선 예측과 가중치 RL 디컨볼루션을 이용한다.

가. 가중치 RL 디컨볼루션

RL 디컨볼루션은 번짐 현상 제거 분야에서 오랫동안 사용된 방법으로 점 확산 함수를 알고 있을 경우에 사용된다. RL 디컨볼루션은 식 (8)과 식(9)로 표현된다.

$$\mathbf{I}_{i+1} = \mathbf{I}_i \bullet \mathbf{E}_m \tag{8}$$

$$\mathbf{E}_m = \frac{\mathbf{B}}{\mathbf{I}_i \otimes \mathbf{K}} \otimes \hat{\mathbf{K}} \tag{9}$$

여기서 $\mathbf{I}_i$은 i번째 선명한 영상, $\mathbf{B}$은 번짐 영상 $\mathbf{K}$와 $\hat{\mathbf{K}}$는 점 확산 함수와 뒤집힌 점 확산 함수를 의미하는 행렬이다. $\mathbf{E}_m$은 $\mathbf{I}_i$에 존재하는 오차를 수정하기 위해 사용된다. RL 디컨볼루션은 식 (8)을 반복적으로 적용하여 선명한 영상을 얻는다. 본 논문에서는 기존의 방법을 식 (10)과 같이 수정하여 가중치를 적용하였다.

$$\mathbf{I}_{i+1} = \mathbf{I}_i \bullet \{\mathbf{W} \times (\mathbf{E}_m - 1) + 1\} \tag{10}$$

여기서 $\mathbf{E}_m$의 기준 값, 즉 $\mathbf{I}_i$에 아무런 수정도 가하지 않는 화소의 경우 $\mathbf{E}_m$의 원소의 값은 1이다. $\mathbf{E}_m$에 가중치 $\mathbf{W}$를 곱하기 위해 1을 빼서 기준 값을 0으로 이동시킨다. 가중치 $\mathbf{W}$를 곱한 후 1을 더하여 기준 값을 1로 돌려놓는다.

(a) (b)

그림 5. (a) 기존의 RL 디컨볼루션을 이용한 결과, (b) 가중치 RL 디컨볼루션을 이용한 결과
Fig. 5. (a) Conventional RL deconvolution, (b) Weighted RL deconvolution.

그림 5는 기존의 RL 디컨볼루션과 가중치 RL 디컨볼루션의 차이를 비교 한 것이다. 첫 번째 행은 번짐 현상이 제거된 영상이며 두 번째, 세 번째 행은 확대 영상이다. 세 번째 행은 번짐 현상이 제거된 영상과 번짐 영상의 차 영상이다. 기존의 방법과 가중치 RL 디컨볼루션 방법 모두 번짐 영상을 선명하게 개선하였다. 그러나 확대 영상과 차 영상을 통해 확인할 경우 기존의 방법은 외곽선 근처에 물결 현상이 발생하였으나, 가중치 RL 디컨볼루션의 경우 물결 현상을 억제하였다.

나. 외곽선 예측

가중치 RL 디컨볼루션이 올바르게 작동하기 위해서는 정확한 외곽선 예측이 필요하다. 제안하는 방법에서는 Cho가 제시한 외곽선 예측 방법을 사용하였다[3]. 그림 6은 번짐 영상으로부터 외곽선을 예측하는 방법을 나타낸다. 외곽선 예측의 첫 단계에서는 양방향 필터(bilateral filter)와 쇼크 필터(shock filter)를 순차적으로 적용하여 주요 외곽선을 선명하게 복원한다. 양방향 필터는 밝기 값의 차이와 중심 화소와의 거리에 따른 커널(kernel)을 이용하여 영상의 외곽선은 유지하면서 영상의 잡음을 효과적으로 제거하는 방법이다. 쇼크 필터

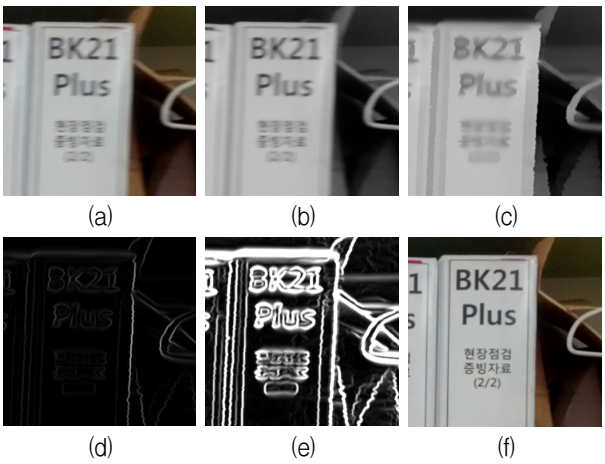

그림 6. a) 번짐 영상, (b) 양방향 필터, (c) 쇼크 필터, (d) 외곽선 추출, (e) 최댓값 제한 정규화, (f) 선명한 영상

Fig. 6. (a) Blur image, (b) Bilateral filter, (c) Shock filter, (d) Edge extraction, (e) Limited maximum normalization, (f) Sharpen image.

는 선명하지 않은 외곽선을 선명한 외곽선으로 바꾸어 주는 필터이다. 두 필터의 적용 결과 번짐 영상으로부터 주요 외곽선이 복원된다. 복원된 외곽선은 가중치 RL 디컨볼루션에서 이용하기 적합한 외곽선 지도로 바꾸어야 한다. 외곽선 지도로 바꾸기 위해 그림 6(d)와 같이 소벨 필터(sobel filter)를 이용하여 외곽선을 검출하고 가우시안 필터(gaussian filter)를 통해 외곽선 영역을 확장한다. 마지막 단계인 최댓값 제한 정규화는 식 (11)으로 정의되는데, λ 이상의 값은 모두 1로 제한하고 1 보다 작은 값은 [0,1] 범위로 정규화 한다. 제안하는 방법에서는 최댓값 제한 정규화를 이용하여 λ 이상의 밝기를 갖는 외곽선이 모두 같은 가중치를 갖게 한다.

$$N_{\max}(v)=\begin{cases}1 & v \geqq \lambda \\ \dfrac{v}{\lambda} & v<\lambda\end{cases} \tag{11}$$

Ⅲ. 실 험

제안하는 방법의 성능을 평가하기 위하여 Li가 제안한 방법 그리고 Cho가 제안한 방법과 비교했다[9,10]. 비교 방법으로는 결과영상을 육안으로 비교하는 주관적 화질 평가와 CPBD(Cumulative Probability of Blur Detection)를 비교하는 객관적 화질 평가가 사용되었다. CPBD는 Narvekar가 제시한 영상의 번짐 정도를 평가하는 척도로 전체 외곽선 중에서 사람이 선명하다고 느끼는 외곽선의 비율을 나타낸다[14]. 정답 영상을 필요로 하는 PSNR(최대 신호 대 잡음비:Peak Signal-to-Noise Ratio), SSIM(Structural SIMilarity)등과는 다르게 CPBD는 정답 영상 없이 영상을 평가한다. 따라서 정답

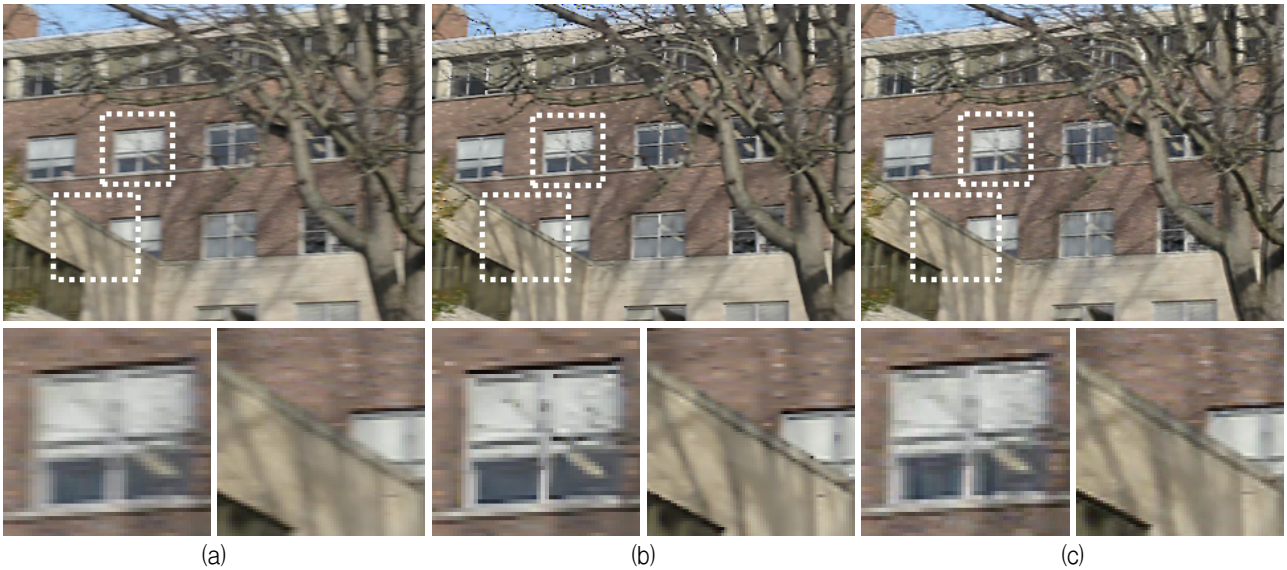

그림 7. 동영상 '2381'에서의 Li의 방법과 제안하는 방법의 비교 (a) 번짐 영상, (b) Li의 방법, (c) 제안하는 방법

Fig. 7. Comparison between Li's method and proposed method using Video '2381'..
(a) Blur image, (b) Li's method, (c) Proposed method.

영상을 확보하기 어려운 동영상에서의 번짐 현상 제거 문제에서 사용하기에 적합한 평가 척도이다.

1. Li의 방법과 비교

Li가 공개한 실험 영상을 이용하여 Li의 방법과 결과를 비교하였다. 그림 7은 동영상 '2381'을 이용하여 Li의 방법과 비교한 결과이다. Li의 방법과 제안하는 방법 모두 번짐 현상을 효과적으로 제거하고 있다. 그림 7의 아래 2개의 영상은 일부분을 확대한 것으로 두 방법 모두 창문의 외곽선을 선명하게 복원하였다. 건물의 일부분을 확대한 영상에서도 두 방법 모두 건물의 외곽선을 복원하였다. 그러나 나뭇가지에서 발생한 번짐 현상의 경우 카메라 포커스가 맞지 않아 발생했기 때문에 두 방법 모두 복원에 실패하였다. 그림 8은 CPBD를 이용한 비교 결과를 그래프로 나타내었다. 그림 8에서 제안하는 방법은 처음과 마지막 한 프레임의 결과가 없는데 이것은 제안하는 방법이 3개 프레임으로부터 점 확산 함수를 추정하기 때문이다. 전체 프레임에서 제안하는 방법이 Li의 방법보다 더 좋은 개선 결과를 보여준다. 그러나 영상의 후반부에 두 방법 모두 개선 결과가 낮아지는데 이는 카메라의 이동에 따라 영상에서 나무가 차지하는 비중이 높아지기 때문이다.

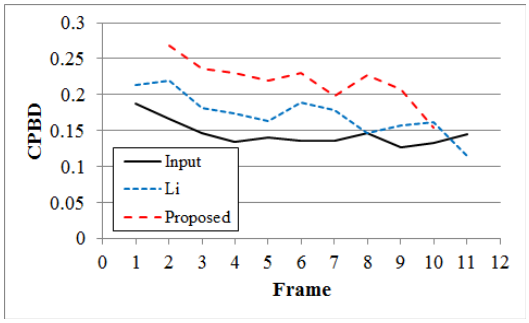

그림 8. 동영상 '2381'에서 Li의 방법과 제안하는 방법의 비교
Fig. 8. Comparison between Li's method and proposed method using Video '2381'.

표 1은 Li가 공개한 동영상을 이용하여 제안하는 방법과 CPBD를 비교한 결과이다. 3개 입력 동영상의 평균 CPBD는 0.1593이며, 제안하는 방법의 평균 CPBD는 0.3578으로 Li의 방법이 가진 0.2499보다 0.1080 더 높았다. 이것은 입력 동영상은 모든 외곽선 중에서 약 15.93%만 선명한 것으로 인식되지만 Li의 방법은 이보다 약 9.06%, 제안하는 방법은 약 19.85% 더 많은 외곽선이 선명하게 인식된다는 의미이다.

표 1. Li의 방법과 제안하는 방법의 비교
Table 1. Comparison between Li's method and proposed method.

	동영상 '2381'	동영상 '1695'	동영상 '1703'	평균	입력영상 대비 개선도
입력 영상 (번짐 영상)	0.1449	0.2192	0.1137	0.1593	-
Li의 방법	0.1724	0.1928	0.3844	0.2499	56.87%
제안하는 방법	0.2189	0.3235	0.5309	0.3578	124.60%

(a) (b) (c)

그림 9. 동영상 'car'에서의 Cho의 방법과 제안하는 방법의 비교 (a) 번짐 영상, (b) Cho의 방법, (c) 제안하는 방법
Fig. 9. Comparison between Cho's method and proposed method using Video 'car'. (a) Blur image, (b) Cho's method, (c) Proposed method.

2. Cho의 방법과 비교

비교에는 Cho가 공개한 데이터 중 'car', 'kid', 'bridge', 'street' 동영상이 사용되었다. 그림 9는 동영상 'car'를 이용하여 Cho의 방법과 제안하는 방법을 비교하였다. 각 영상의 아래 행은 확대 영상이다. 두 방법 모두 주요 외곽선을 선명하게 복원하였다. 확대 영상 중 왼쪽을 보면 두 방법 모두 번짐 현상 제거에 성공하였으나, 제안하는 방법은 약간의 잡음이 발생하는 문제가 있다. 반면에 Cho의 결과는 지나치게 개선되어 원본과는 다른 느낌을 주는 문제가 있다. 확대 영상 중 오른쪽 영상에서 Cho의 방법이 가진 문제가 두드러 진다. Cho의 방법은 제안하는 방에 비해 나무의 모양이 왜곡시켰다. 그러나 두 방법이 갖는 문제점은 확대 영상을 통해서만 확인이 가능한 수준이다. 그림 10은 동영상 'kid'를 이용한 CPBD 비교 결과이다. 동영상 'kid'는 약 20프레임 이후부터 급격하게 흔들리며 따라서 CPBD도 낮아진다. 카메라 움직임이 많지 않은 20 프레임 이전에서의 CPBD의 평균, 즉 0.3670이 이 동영상의 최대 CPBD값이라고 가정할 때, 카메라가 흔들리는 20프레임 이후 Cho의 방법은 평균 0.2592, 제안하는 방법은 0.3423로 최대 CPBD 보다 각각 0.1078, 0.0247 모자란 결과를 보인다.

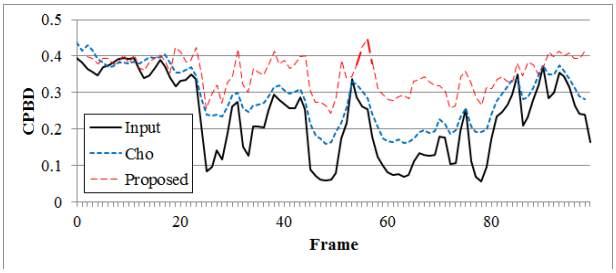

그림 10. 동영상 'kid'에서의 Cho의 방법과 제안하는 방법의 비교
Fig. 10. Comparison between Cho's method and proposed method using Video 'kid'.

표 2. Cho의 방법과 제안하는 방법의 비교
Table 2. Comparison between Cho's method and proposed method.

	동영상 'car'	동영상 'kid'	동영상 'bridge'	동영상 'street'	평균	입력영상 대비 개선도
입력 영상 (번짐 영상)	0.4046	0.2329	0.4127	0.3752	0.3564	-
Cho의 방법	0.4715	0.2870	0.4563	0.3821	0.3992	12.03%
제안하는 방법	0.4817	0.3514	0.4707	0.4230	0.4317	21.14%

표 2는 Cho가 공개한 동영상을 이용하여 제안하는 방법과 비교한 결과이다. 4개 입력 동영상의 평균 CPBD는 0.3564이다. 번짐 현상 제거 결과, Cho의 방법은 평균 CPBD 0.3992로 12.03% 개선시켰고, 제안하는 방법은 평균 CPBD 0.4317로 21.14% 개선시켰다. 이는 두 방법이 거의 비슷한 수준으로 외곽선을 복원하였음을 의미한다.

Ⅳ. 결 론

본 논문에서는 단일 카메라를 이용한 동영상에서의 번짐 현상 제거 방법을 제안하였다. 제안하는 방법은 프레임간 카메라 움직임을 추정하여 번짐 현상에 대한 점 확산 함수를 계산하여, 패치 단위의 디컨볼루션을 수행한 뒤 다시 합하는 방식으로 카메라 이동으로 발생되는 동영상의 번짐 현상을 제거하였다. 제안하는 방법의 성능을 평가하기위해 Li가 공개한 동영상과 Cho가 공개한 동영상을 이용하여 제안하는 방법과 결과를 비교하였다. CPBD를 이용하여 비교한 결과 평균 CPBD가 0.1593인 입력 동영상들을 Li의 방법으로는 0.2499로 56.87% 개선되었고, 제안하는 방법으로는 0.3578으로 124.60% 개선되었다. 다른 비교 대상인 Cho의 방법은 평균 CPBD가 0.3564인 입력 동영상을 평균 CPBD 0.3992으로 12.03% 개선시켰으며, 제안하는 방법은 평균 CPBD 0.4317으로 21.14% 개선시켰다. 실험결과, 제안하는 방법은 비교한 방법보다 더 좋거나 비슷한 수준의 결과를 얻었다.

REFERENCES

[1] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," ACM Transaction on Graphics(TOG), Vol. 25, No. 3, pp. 787-794, 2006
[2] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," ACM Transactions on Graphics (TOG), Vol. 27, No. 3, 2008.
[3] S. H. Cho, and S. Y. Lee, "Fast motion deblurring," ACM Transactions on Graphics (TOG), Vol. 28, No. 5, 2009.

[4] L. Xu, and J. Jia, “Two-phase kernel estimation for robust motion deblurring,” In Computer Vision - ECCV 2010, pp. 157-170, Springer Berlin Heidelberg, 2010.
[5] L. Yuan, J. Sun, L. Quan, and H. Y. Shum, “Image deblurring with blurred/noisy image pairs,” ACM Transactions on Graphics (TOG), Vol. 26, No. 3, 2007.
[6] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski, “Image deblurring using inertial measurement sensors,” ACM Transactions on Graphics (TOG), Vol. 29, No. 4, 2010.
[7] M. Ben-Ezra, and S. K. Nayar, “Motion deblurring using hybrid imaging,” Computer Vision and Pattern Recognition (CVPR), Vol. 1, 2003.
[8] Y. W. Tai, H. Du, M. S. Brown, and S. Lin, “Image/video deblurring using a hybrid camera,” Computer Vision and Pattern Recognition (CVPR), Vol. 1, 2003.
[9] Y. Li, S. B. Kang, N. Joshi, S. and M. Seitz, “Generating sharp panoramas from motion-blurred videos,” Computer Vision and Pattern Recognition (CVPR), pp. 2424-2431, 2010.
[10] S. H. Cho, J. Wang, and S. Y. Lee, “Video deblurring for hand-held cameras using patch-based synthesis,” ACM Transactions on Graphics (TOG) Vol. 31, No. 4, 2012.
[11] B. D. Lucas, and T. Kanade, “An iterative image registration technique with an application to stereo vision,” International Joint Conferences on Artificial Intelligence (IJCAI), Vol. 81, 1981.
[12] E. Mair, G. Hager, D. Burschka, M. Suppa, and G. Hirzinger, “Adaptive and generic corner detection based on the accelerated segment test,” Computer Vision - ECCV 2010. Springer Berlin Heidelberg, pp. 183-196. 2010.
[13] D. G. Lowe, “Object recognition from local scale-invariant features,” The proceedings of the seventh IEEE international conference on computer vision, Vol. 2. 1999.
[14] N. D. Narvekar, and L. J. Karam, “A no-reference image blur metric based on the cumulative probability of blur detection (CPBD),” IEEE Transactions on Image Processing, Vol. 20, No. 9, pp. 2678-2683, 2011.
[15] C. W. Son, and H. M. Park, “Fast Multiple-Image-Based Deblurring Method,” Journal of The Institute of Electronics Engineers of Korea, Vol. 49-SP, No. 4, pp. 49-57, 2012.

저 자 소 개

정 우 진(학생회원)
2012년 2월 한양대학교 전자컴퓨터공학부 학사
2012년 3월~현재 한양대학교 컴퓨터공학과 석박사과정
<주관심분야 : 영상처리, 컴퓨터비전>

박 진 욱(학생회원)
2006년 2월 한양대학교 전자컴퓨터공학부 학사
2008년 3월~현재 한양대학교 컴퓨터공학과 석박사과정
<주관심분야 : 영상처리, 패턴인식 >

이 종 민(학생회원)
2007년 2월 한양대학교 컴퓨터공학과 학사
2009년 2월 한양대학교 컴퓨터공학과 석사
2010년 3월~현재 한양대학교 컴퓨터공학과 박사과정
<주관심분야 : 영상처리, 컴퓨터비전, 패턴인식>

송 태 언(정회원)
2007년 2월 경북대학교 전자전기컴퓨터학부 학사
2009년 2월 경북대학교 전자전기컴퓨터학부 석사
2009년~현재 삼성탈레스 재직
<주관심분야 : 영상처리 알고리즘, 패턴인식>

최 원 주(정회원)
2009년 2월 연세대학교 전기전자공학부 학사
2009년~현재 삼성탈레스 재직
<주관심분야 : FPGA Logic 설계, 영상처리 알고리즘>

문 영 식(정회원)
1980년 2월 서울대학교 전자공학과 공학사
1982년 2월 한국과학기술원 전기 및 전자공학과 공학석사
1990년 Ph. D. in Department of Electrical and computer Engineering University of California at Irvine
1982년~1985년 한국 전자통신연구소 연구원
1989년~1990년 Inno Vision Medical 선임연구원
1990년~1992년 생산기술연구소 선임연구원
1992년~현재 한양대학교 컴퓨터공학과 교수
<주관심분야 : 영상처리, 컴퓨터비전, 패턴인식>