

# Optical Flow를 사용한 동영상의 흔들림 자동 평가 방법

백상현<sup>†</sup>, 황원준<sup>\*\*</sup>

## Automatic Jitter Evaluation Method from Video using Optical Flow

Sang Hyune Baek<sup>†</sup>, WonJun Hwang<sup>\*\*</sup>

### ABSTRACT

In this paper, we propose a method for evaluating the uncomfortable shaking in the video. When you shoot a video using a handheld device, such as a smartphone, most of the video contains unwanted shake. Most of these fluctuations are caused by hand tremors that occurred during shooting, and many methods for correcting them automatically have been proposed. It is necessary to evaluate the shake correction performance in order to compare the proposed shake correction methods. However, since there is no standardized performance evaluation method, a correction performance evaluation method is proposed for each shake correction method. Therefore, it is difficult to make objective comparison of shake correction method. In this paper, we propose a method for objectively evaluating video shake. Automatically analyze the video to find out how much tremors are included in the video and how much the tremors are concentrated at a specific time. In order to measure the shaking index, we proposed jitter modeling. We applied the algorithm implemented by Optical Flow to the real video to automatically measure shaking frequency. Finally, we analyzed how the shaking indices appeared after applying three different image stabilization methods to nine sample videos.

**Key words:** Optical Flow, Image Stabilization, Video Stabilization Performance

### 1. 서 론

일반적으로 휴대형 장치를 사용하여 동영상을 촬영하는 경우, 원치 않는 흔들림이 동영상 내에 많이 포함된다. 이러한 흔들림은 동영상 시청 시에 불쾌감을 주는 요인이 된다. 흔들림이 발생하는 요인은 여러 가지가 있으나 대부분의 경우 손으로 들고 촬영을 할 때 손 떨림이 발생하게 되면 이 손 떨림이 기기를

흔들어 영상에 흔들림이 나타나게 된다. 이러한 손 떨림이 영상에 나타나는 것을 줄여주기 위해 영상의 흔들림을 보정하는 방법이 연구되었다[1-6]. 손 떨림을 방지하는 방법은 크게 2가지로 나누어지는데, 하나는 하드웨어를 사용해 촬영 시에 실시간으로 손 떨림을 보정하는 방법이 있고, 다른 하나는 촬영이 끝난 후 소프트웨어를 사용해 흔들림을 후보정해주는 방법이다. 하드웨어를 사용해 실시간으로 보정하

\* Corresponding Author: WonJun Hwang, Address: (16499) 206, World cup-ro, Yeongtong-gu, Suwon, Gyeonggi-do, Korea, TEL: +82-31-219-2632, FAX: +82-31-219-1621, E-mail: wjhwang@ajou.ac.kr  
Receipt date: July 4, 2017, Approval date: July 28, 2017  
<sup>†</sup> Dept. of Computer Eng., Graduate School, Ajou University (E-mail: shbaek@gmail.com)

<sup>\*\*</sup> Dept. of Computer Eng., Graduate School, Ajou University  
\* This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2017R1D1A1B03031560).

는 방법은 카메라의 흔들림을 가속도계나 자이로스코프로 측정해 이에 비례하는 만큼 렌즈 또는 이미지 센서를 이동시켜 흔들림을 상쇄시킨다[7]. 소프트웨어로 후처리 하는 방법은 디지털 이미지 안정화(Digital Image Stabilization)라 불리며 촬영된 영상을 프레임 단위로 다시 나눈 후 프레임 간의 Motion Vector를 계산하여, 흔들림이 발생한 곳으로 추정되는 프레임에서 Motion Vector만큼 보정하여 영상의 흔들림을 제거한다[2-6].

영상의 흔들림을 보정하는 하드웨어나 소프트웨어의 보정 성능을 평가하기 위해서는 일반적으로 흔들림이 없는 기준 영상과 동일한 영상에 흔들림을 추가한 영상을 보정한 후 차이를 측정하거나, 흔들림이 포함된 영상의 보정 전, 후에 대한 주관적 평가 또는 프레임 간 Motion Vector 오류를 계산하게 된다. 이러한 성능 측정 방법은 표준화된 방법이 없기 때문에 보정 방법을 연구하는 연구자 별로 필요에 따라 제안하고 있다. 이와 같은 이유로 기준 영상이 없는 경우 성능 평가가 어렵거나, 여러 방법들 간에 객관적인 성능 비교가 불가능하다.

본 연구에서는 영상 자체의 흔들림을 객관적으로 평가할 수 있는 방법을 제시하고자 한다. 이를 위해 영상의 흔들림에 대한 모델링을 하였으며, 모델링을 근거로 Optical Flow를 기반으로 한 흔들림 평가 지수를 계산하는 알고리즘을 제안하였다. 제시한 알고리즘의 타당함을 입증하기 위해 OpenCV를 사용해 알고리즘을 실제 구현하였으며, 구현된 프로그램을 9개의 샘플영상에 대해 적용해 흔들림 지수를 산출해 보았다. 이후 서로 다른 3가지 영상 안정화 소프트웨어를 사용해 흔들림 지수를 계산하여 흔들림 보정 전, 후에 흔들림 지수의 변화를 확인하였다. 흔들림 지수 값을 통해 보정 후에 흔들림이 개선되었음을 확인하였으며, 3가지 안정화 소프트웨어 간에 흔들림 안정화 성능 차이가 있음을 확인하였다.

## 2. 기존 연구

### 2.1 주관적 평가 기법 (Mean Opinion Score, MOS) [2]

음성 통신의 음질을 평가하기 위해 제안된 방법으로서[8], 선별된 여러 명에게 음성을 들려준 후 개인이 느낀 품질을 1(가장 나쁨)~ 5(가장 좋음)으로 표시하게 한 후 이 결과를 가지고 음성의 품질을 평가하

게 된다. 동영상에 적용하는 경우에도 음성과 마찬가지로 선별된 여러 명에게 영상을 보여준 후 1~5까지 품질을 표시하게 한 후 이 값의 평균을 가지고 품질을 평가한다. 주관적 평가의 경우 평가에 걸리는 시간이 길고 비용이 많이 들기 때문에 자주 이용할 수가 없다[9]. 또한 개인의 성향 및 그 날의 심리 상태에 따라서 그 평가 지표가 바뀔 수 있고, 다른 사람들의 평가 결과를 보여줌으로써 평가 결과가 왜곡될 수 있는 단점이 있다.

### 2.2 기준 영상과 안정화 시킨 영상의 Peak Signal-to-Noise Ratio(PSNR)을 사용한 평가[3-4]

기준 동영상의 이미지와 흔들림을 안정화 시킨 이미지 간의 차이를 계산하여 평가한다. 두 영상간의 Mean Square Error(MSE)를 계산한 후 이로부터 PSNR을 계산한다. 안정화가 잘 될수록 PSNR 값이 크게 나온다.

$$MSE(I_C, I_R) = 10 \log \frac{\sum_{x=1, y=1}^{w, h} diff^2(x, y)}{w \times h} \quad (1)$$

$$diff(x, y) = \begin{cases} 1 & |I_C(x, y) - I_R(x, y)| \geq threshold \\ 0 & otherwise \end{cases} \quad (2)$$

$$PSNR(I_C, I_R) = 10 \log \frac{255}{MSE(I_C, I_R)} \quad (3)$$

여기서  $x, y$ 는 영상에서의 좌표를 나타내고  $w$ 는 영상의 폭,  $h$ 는 영상의 높이이다. 수식(2-3)의  $I_C$ 는 안정화 시킨 영상이고  $I_R$ 은 기준 영상이다.

### 2.3 Root Mean-motion Square Error(RMSE) 비교 [5]

동영상의 Motion Vector( $CMV(i)'$ )와 참고문헌 [5]에서 제안한 알고리즘을 적용한 후의  $CMV(i)$ 간의 차이로 부터 RMSE를 계산하여 제안한 알고리즘의 성능을 평가하였다.

$$RMSE = \frac{1}{N} \sqrt{\sum_{i=1}^N (CMV(i) - CMV(i)')^2} \quad (4)$$

여기에서  $CMV(i)$ 는  $i$ 번째 프레임에서 제안한 방법으로 생성된 Motion Vector 이고,  $CMV(i)'$ 는 원본의  $i$ 번째 프레임의 Motion Vector이다.

PSNR이나 MSE를 사용하는 방법은 수학적인 단순함으로 많이 사용되고 있으나, 실제 영상의 품질과

는 큰 관련이 없는 것으로 알려져 있다[9]. 또한 픽셀 기반의 평가이기 때문에 기준 영상과 비교 영상간의 비디오포맷 변환, 이미지 안정화 작업등 후에 밝기 차이, 위치의 변화, 확대/축소 등의 픽셀 변화가 모두 오류로 계산되어 나타나는 문제가 있다[9].

2.4 Motion Vector 그래프 비교 방법[6]

원본 동영상의 프레임 간 Motion Vector(GMV)의 방향성 변화 그래프와 동영상 안정화 후의 프레임 간 Motion Vector(Smooth Motion Vector, SMV)의 방향성 변화 그래프를 비교한다. 비교 그래프를 보면 GMV에 비해 SMV의 변화가 완만해짐을 확인 할 수 있으면, 이러한 Motion Vector 변화의 완만함은 영상의 흔들림이 안정화 되었다는 것을 의미한다. 참고문헌 [6]에서는 GMV와 SMV의 방향성 변화 비교 외에 비디오 압축시의 영상 품질을 비교하였다. 이를 위해 원본 동영상과 안정화 후 동영상의 비디오 압축 후 영상에 대한 PSNR을 측정하였으며, 안정화 후 동영상의 PSNR 값이 원본 동영상에 비해 커짐을 통해 영상 품질이 개선되었음을 보인다.

Motion Vector의 방향성 변화 그래프를 비교하는 방법의 경우 단순히 그래프를 눈으로 보면서 비교하기 때문에 다른 알고리즘 간의 안정화 성능에 대한 객관적인 평가를 할 수 없다.

2.5 연속된 프레임간의 차이와 영상의 흐려짐을 계산해 평가[9]

연속된 프레임간의 차이는 MSE로 계산한다. 계

산된 MSE는 bias와 variance의 두 부분으로 나눌 수 있다.

$$MSE = E\{e\}^2 + Var\{e\} \tag{5}$$

여기에서  $e$ 는 프레임 간의 차이를 나타내며 수식(5)의 앞에 값  $E\{e\}^2$ 는 bias를 의미하며 이는 divergence(발산, 정상적인 움직임)에 해당한다.  $Var\{e\}$ 는 variance를 의미하며 jitter(흔들림)에 해당한다. 이 두 값을 나누기 위해 참고문헌 [9]에서는 low/high pass filter를 적용하였으며, low와 high를 구분하는 cut-off frequency를 설정하였다. 실제 실험에서는 cut-off 주파수에 1Hz를 적용하였다.

$$D_c = \frac{1}{frames} \sum_{i=1}^{frames} \{e_{f \leq c}(i)\}^2 \tag{6}$$

$$J_c = \frac{1}{frames} \sum_{i=1}^{frames} \{e_{f > c}(i)\}^2 \tag{7}$$

$$Attenuation = J_{c\_stabilized} / J_{c\_original} \tag{8}$$

수식(6-7)에서  $c$ 는 cut-off 주파수를 의미한다. 수식(6)에서  $e_{f \leq c}(i)$ 는 MSE의 주파수가 cut-off 주파수 이하인 low frequency(divergence)값을 의미하며, 이 값을 전체 영상에 대해 프레임별 평균을 구한 것이  $D_c$ (divergence)값이다.  $D_c$ 값은 느린 움직임을 나타내기 때문에 카메라의 의도된 움직임으로 해석된다. 수식(7)에서  $e_{f > c}(i)$ 는 MSE의 주파수가 cut-off 주파수 보다 큰 high frequency(jitter)값을 의미하며, 수식(6)과 마찬가지로 전체 영상에 대해 프레임별 평균값을 산출한 것이  $J_c$ (jitter)값이다.  $J_c$ 는 빠

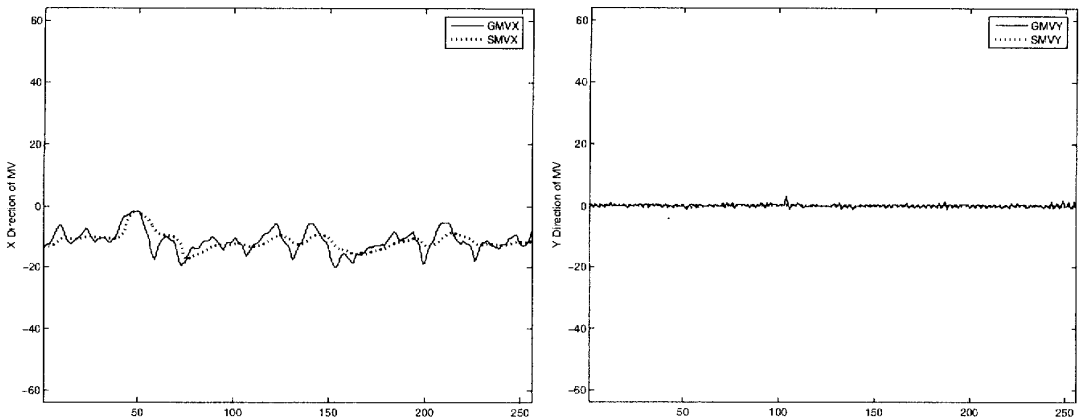


Fig. 1. Example of GMV and SMV graph[6]

른 움직임의 의미하기 때문에 흔들림에 의해 발생한 것으로 해석된다. jitter의 감소 정도는 수식(8)과 같이 원본 영상과 안정화시킨 영상의  $J_c$  비율로 계산한다. 참고문헌 [9]에서는 jitter 외에 영상의 흐림(Blur) 정도를 Point-Spread-Function(PSF)를 사용해 계산하여 평가에 포함시키고 있다. 이 평가 방법은 기존 영상을 사용하기 때문에 기존 영상 없이는 평가가 불가능하고 기존 영상의 선택에 따라 결과가 영향을 받게 되며 연속된 영상에서 실시간 평가가 어렵다.

### 3. 제안 방법

본 연구에서는 기존 연구에서 영상의 품질을 평가하기 위해 사용한 주관적인 방법이나 기존 영상을 사용할 때 생기는 단점을 해결하기 위해 자동으로 흔들림을 판단해 이를 지수화 시키는 방법을 제안하였다. 이 흔들림 지수는 영상을 분석해 자동으로 판정하기 때문에 객관적이며, 기존 영상 없이 영상 자체에 대한 흔들림을 분석하기 때문에 기존 영상에 의한 평가 결과의 영향이 없다. 또한 실시간으로 분석이 가능하여 연속적인 영상에 대해 실시간으로 지수 산출을 할 수 있으며, 서로 다른 이미지 안정화 알고리즘의 결과 영상을 객관적이고 쉽게 상호 비교할 수 있게 도와준다.

#### 3.1 흔들림 모델링

카메라로 동영상을 촬영할 때 발생하는 흔들림은 상하(세로방향), 좌우(가로방향), 회전, 전후의 움직임이 복합적으로 섞여 있다. 카메라가 앞뒤로 흔들리는 경우에는 대상 체까지의 거리 변화에 의한 대상체의 크기 변화 또는 초점의 변화(흐려짐)로 나타나게 되고, 상하좌우의 흔들림은 영상을 상하좌우로 흔들게 된다. 카메라가 회전하는 경우에는 대상체가 돌아간 것으로 표시된다. 이러한 흔들림은 영상 전체에 대해 골고루 영향을 미치게 되며, 영상의 일부만 변경되지는 않는다. 본 연구에서는 흔들림에 의한 영상의 품질을 판단하기 위해 앞뒤 움직임에 의한 초점 변화와 회전에 의한 영향은 고려하지 않았다.

흔들림에 의한 이동 거리는 등가속도 운동[10]을 적용하면 쉽게 계산할 수 있다.

$$a_x, a_y (m/sec^2), t(sec) \quad (9)$$

$$\Delta_x = \frac{1}{2} a_x \times t^2 (m) \quad (10)$$

$$\Delta_y = \frac{1}{2} a_y \times t^2 (m) \quad (11)$$

수식(9)에서  $a_x, a_y$ 는 이미지 센서의 x, y 방향에 대한 가속도이며,  $t$ 는 흔들림이 지속된 시간이다. 수식(10)과 (11)의  $\Delta_x, \Delta_y$ 는 수식(9)의 값으로 계산된 이미지 센서의 x, y 방향의 이동거리이다. 실제 흔들림에 의한 이미지 센서의 이동 거리를 계산하기 위해 흔들림에 대한 가속도 및 지속 시간을 적용코자 하였다. 흔들림의 지속 시간은 0~20Hz 사이인 것으로 알려져 있으나[7,11], 가속도에 대한 측정치를 알기 어려웠다. 참고문헌 [7]의 측정 및 분석에 의하면 흔들림 크기는 대략 0.5도 이내임을 확인할 수 있었으며, 이 값을 가지고 지속시간과 이동 거리를 계산하는 수정된 수식을 만들었다. Fig. 2는 흔들림에 의해 이미지센서가 이동하는 거리를 보여준다.

$$W \times H (pixels), t(sec) = \frac{1}{f(Hz)} \quad (12)$$

$$L = W \times \tan(0.5^\circ) \quad (13)$$

$$F = 30 (frames/sec) \times t(sec) \quad (14)$$

$$P = \frac{L}{F} (pixels) \quad (15)$$

수식(12)에서  $W$ 는 영상의 가로 해상도이고  $H$ 는 영상의 세로 해상도이다.  $t$ 는 흔들림이 지속되는 시간으로 이는 주파수의 역수로 계산한다. 수식(13)에서 영상의 가로, 세로 해상도 중 큰 값과 센서의 회전 0.5도를 곱하여 흔들림에 의해 발생하는 픽셀의 최대 이동 거리를 계산하였다. 수식(14)에서 이 흔들림 지속 시간을 사용해 흔들림이 나타나는 프레임 수를 계산한다. 수식(15)에서 흔들림에 의한 최대 이동 픽

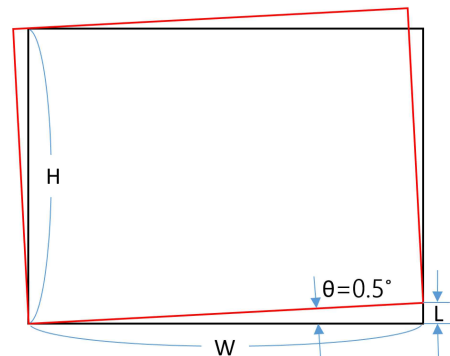


Fig. 2. Drawing of image sensor size and travel distance.

Table 1. Camera travel distance due to shaking

Amplitude degree( $\theta$ )	Frequency	$F(\text{frame})$	$L(\text{pixels})$	$P(\text{pixels})$
0.5°	1 Hz	30	16.76	0.559
0.5°	10 Hz	3	16.76	5.587
0.5°	15 Hz	2	16.76	8.380
0.3°	1 Hz	30	10.01	0.334
0.3°	10 Hz	3	10.01	3.337
0.3°	15 Hz	2	10.01	5.005

셀수를 수식(14)에서 구한 프레임 수로 나누어 프레임 하나 당 나타나는, 흔들림에 의해 이동하는 픽셀의 개수를 구하였다.

실제 영상에서 흔들림에 의해 이동하는 거리가 얼마나 발생하는지 계산하였다. 동영상의 해상도는 본 연구에서 사용한 1920 × 1080 픽셀이며 픽셀 이동거리는 이중 큰 값인 1920을 사용하였다.

Table 1에서 확인 할 수 있듯이 프레임에서 흔들림에 의해 이동하는 거리는 0.3~8.4 픽셀 사이이다. 이 범위를 벗어나는 값은 흔들림이 아니라 의도적인 움직임 등으로 해석될 수 있다.

### 3.2 흔들림 정의

3.1과 같이 흔들림은 짧은 시간 동안, 일정 거리 이상을 움직여야 하며, 이때 카메라 전체가 움직이기 때문에 영상 전체에 동일한 움직임이 기록되어야 한다. 또한 일정 시간 이상 지속되는 움직임은 의도적인 카메라의 움직임으로 판단되어 흔들림에서 제외하여야 한다. 따라서 본 연구에서 흔들림은 다음과 같은 조건을 만족하여야 한다.

- 1) 흔들림 양이 일정 크기 이상 이어야 한다.
  - 2) 프레임 전체가 동일 방향으로 움직여야 한다.
  - 3) 프레임의 이동 방향이 연속적이지 않아야 한다.
- 프레임이 연속적으로 동일 방향으로 움직인 경우 카메라나 대상체가 움직인 영상이다.

### 3.3 알고리즘

3.2의 흔들림 정의에 따라 영상의 흔들림 정도를 알기 위해 영상의 연속된 프레임 간에 Optical Flow를 계산하고 이 양이 일정 크기 이상 프레임 전체에 대해 동일 방향으로 발생하면 영상이 흔들린 것으로 판단하였다. Fig. 3은 알고리즘의 전체 흐름도이다.

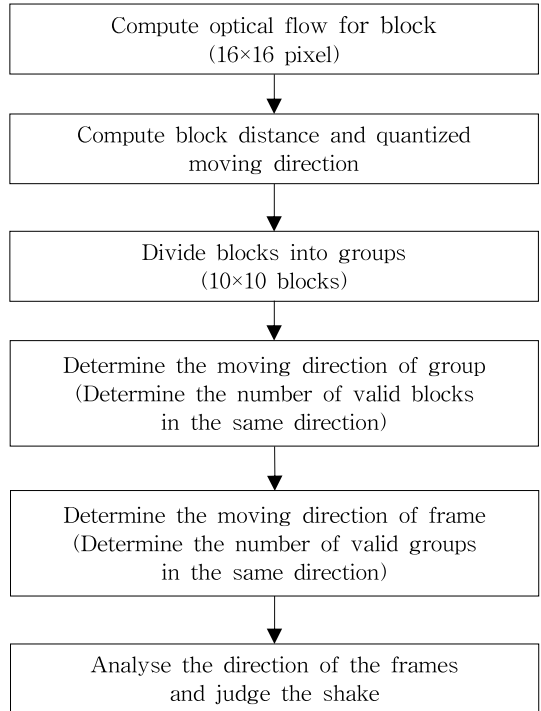


Fig. 3. Flowchart for jitter detection algorithm.

먼저 영상으로부터 이미지 프레임을 가져온다. 가져온 이미지 프레임을 일정 크기 픽셀(16 × 16 pixel) 단위의 블록으로 나눈다. 두 개의 연속된 이미지 프레임  $I_t$ 와  $I_{t-1}$  사이에 개별 블록의 Optical Flow를 계산[12],[13]하고 이로부터 블록의 이동방향과 이동량을 계산한다. 블록의 이동 방향은 n개로 양자화 한다.

$$d_x[i, j], d_y[i, j] \tag{16}$$

$$d[i, j] = \sqrt{d_x^2 + d_y^2} \tag{17}$$

$$a[i, j] = \tan^{-1}\left(\frac{d_y}{d_x}\right) \tag{18}$$

$$A[i, j] = \text{Quantize}(a[i, j]) \tag{19}$$

수식(16)은 Optical Flow로 부터 구한 블록의 이동거리이며, i는 x방향 블록 번호이고 j는 y방향 블록 번호이다. 수식(17)에서 d는 블록의 이동거리이며, 수식(18)에서 a는 블록의 이동방향이다. 수식(19)의 A는 수식(18)의 블록의 이동방향(a)를 n개로 양자화 시킨 값이다. 수식(16-19)에서 계산된 개별 블록을 모아 몇 개의 그룹을 만들고 이 그룹들의 이동

방향을 결정한다.

$$k = A[i, j] \quad (20)$$

$$c[k] = \begin{cases} c[k] + 1, & \text{if } d[i, j] > L_e \\ c[k], & \text{otherwise} \end{cases} \quad (21)$$

$$c[k] = \begin{cases} c[k], & \text{if } c[k] > N_A \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

$$G[I, J] = k, c_k = \text{MAX}(c_0, \dots, c_n) \quad (23)$$

3.2의 흔들림 정의 1에 따라 각 그룹에서 유효 이동 거리를 이동한 블록을 선별한다. 수식(21)에서  $L_e$ 는 유효한 최소 이동 거리이다. 3.2의 흔들림 정의 2에 따라 흔들림인 경우 모두 동일한 방향으로 이동하여야 한다. 수식(22)에서 동일 방향이 최소 개수( $N_A$ ) 이하인 경우는 이동 방향에서 유효하지 않은 방향으로 제외한다. 수식(23)에서 그룹 내의 유효 동일 방향 중 가장 많은 개수를 가진 방향을 그룹의 방향으로 정한다. 수식(20-23)에서 계산된 그룹들의 이동 방향으로부터 전체 프레임의 이동 방향을 결정한다. 프레임의 이동 방향 결정은 수식(20-23)에서 개별 그룹의 이동 방향을 결정한 것과 유사한 과정을 적용한다.

$$\{C_0, \dots, C_n\} = \text{count}(G[I, J]) \quad (24)$$

$$C[k] = \begin{cases} C[k], & \text{if } C[k] > N_G \\ 0, & \text{otherwise} \end{cases} \quad (25)$$

$$D_t = k, C_k = \text{MAX}(C_0, \dots, C_n) \quad (26)$$

먼저 수식(24)에서 그룹의 방향 별 개수를 계수한다. 수식(25)에서 수식(24)로부터 구한 이동 방향 개

수가 프레임의 동일 방향 최소 개수( $N_G$ ) 보다 많은 경우 유효한 방향 개수로 판정한다. 수식(26)에서 유효한 방향 개수 중 가장 많은 개수를 찾아 이 방향을 프레임의 방향  $D_t$ 로 정한다. Fig. 4는 알고리즘을 적용해 생성한 프레임의 방향 표시 예이다.

Fig. 4에서 점은 개별 블록의 Optical Flow를 의미하며, 가로세로로 구분된 선은 블록을 모은 그룹을 의미한다. 중간 크기 화살표는 수식(20-23)을 적용해 블록에서 계산한 그룹의 이동방향을 의미하며, 중앙의 큰 화살표는 수식(24-26)을 적용해 계산한 프레임 전체의 이동방향을 나타낸다.

이미지 프레임의  $D_t$ 는 해당 이미지 프레임이 움직인 방향을 의미한다. 따라서 연속된 프레임에서  $D_t$ 값이 변경되면 프레임의 이동 방향이 변경되었다는 것을 의미한다.  $D_t$ 값이 프레임 간에 크게 변경이 되면 이미지 프레임이 서로 관계없는 다른 방향으로 움직였다는 것을 의미하며 이러한 경우에 이 지점에서 흔들림이 발생한 것으로 판단할 수 있다. 반대로  $D_t$ 값이 동일하거나 인접한 값으로 변경되는 경우에는 동일 방향으로 이동한 것으로 판단한다. 이러한 동일 방향 지속 시간이 일정 시간 이상 나타나면 3.2의 정의 3에 따라 흔들림이 아니라 카메라가 이동한 것으로 판단한다. 그러나 동일 방향이 짧은 시간만 지속되고 없어진다면 흔들림이 몇 개 프레임에 걸쳐 나타난 것으로 판단한다.

$D_t$ 의 변화로부터 흔들림을 판정한 후 흔들림이 발생한 빈도로부터 지수를 산출한다. 일반적으로 사람

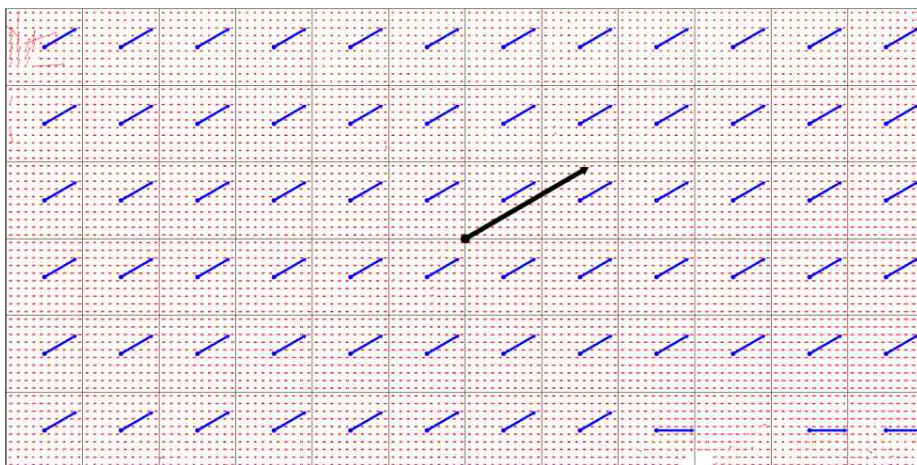


Fig. 4. Optical Flow and group and frame movement directions.

간의 차이가 있기는 하나 동영상에 흔들림이 많아질 수록 시청자는 더 큰 불편감을 느끼게 된다. 따라서 전체 동영상에 얼마나 많은 흔들림이 포함되어 있는지가 흔들림 지수의 한 요소가 될 수 있다. 마찬가지로 전체 횡수는 적더라도 흔들림이 특정 시간대에 집중되어 나타나면 특정 시간대에 불편감을 높게 느끼게 된다. 이와 같은 이유로 다음과 같은 2개의 지수를 산출하였다.

$$m_1 = \frac{\text{Total Jitter Frames}}{\text{No. of Frames}} \quad (27)$$

$$m_2 = \text{MAX} \left( \frac{\text{Jitter Frames for 1 sec}}{31} \right) \quad (28)$$

수식(27)은 전체 프레임 개수 중 흔들림이 발생한

프레임의 비율이다. 수식(28)은 현재 프레임으로부터 전후 0.5초간의 흔들림이 발생한 개수이다. 일반적인 TV의 초당 장면수가 30 프레임이어서 현재 프레임 기준 전후 15프레임을 포함하여 31프레임을 사용하였다. 초당 장면수가 다른 동영상의 경우 다른 값을 사용할 수 있으나, 이 값은 얼마나 집중해서 흔들림이 발생하는가를 알기 위한 것이기 때문에 영상의 초당 장면수와 꼭 일치시킬 필요는 없다.





$$m = \left( \frac{m_1 + m_2}{2} \right) \quad (29)$$

최종 흔들림 지수  $m$ 은 수식(27), 수식(28)에서 구한 2개 지수의 평균으로 수식(29)와 같이 결정하였으며, 이 값이 큰 경우 동영상에 흔들림이 많이 있어

Table 2. list of test conditions

Parameters	Variables	Test value
Block Size		16 × 16 pixel
Optical flow method		PyramidLK
Valid moving length	$L_e$	1 pixel
No. of quantized direction angle	$n$	30° (0 - 11)
Min. no. in the same direction	$N_A, N_G$	50%
Continuous direction recognition		Within the 2 frames, the same direction or adjacent direction
Camera movement recognition		In the same direction for more than 10 frames

Table 3. list of videos (Indoor)

File name	Video	Camera	Object	Remarks
clip19		fix	static	Tapping the desk
clip20		Holding by hand	static	A little shake
clip21		Holding by hand	static	Many shakes
clip22		fix	static	

시청자에게 불쾌감을 많이 주는 영상이라 판단할 수 있다.

#### 4. 실험

3에서 제안한 알고리즘은 Visual Studio 2017 C#과 OpenCV 2.4.10 버전을 사용해 구현하였다. 실험에 사용된 조건은 Table 2와 같다

실험에 사용된 동영상은 APS-C 크기(23.6mm × 15.8mm)의 이미지센서를 가진 카메라와 일반 휴대폰 카메라로 촬영하였다. 동영상의 크기는 1920 × 1080 픽셀이었으며 29.97 frames/sec 속도로 촬영되었다. clip19-22의 동영상은 APS-C 센서를 사용해 촬영된 영상이며, 나머지 영상은 휴대폰 카메라를 사용해 촬영된 것이다. 사용된 각 동영상의 목록은 Table 3, 4와 같다[14].






흔들림 안정화 후의 성능을 비교하기 위해 원본 동영상에 대해 3가지 프로그램을 사용해 안정화를 적용하였다. 사용된 프로그램은 iMovie[15], ffmpeg(open source)[16], Elasty(Creaceed, Demo ver-

sion)[17]이다. Fig. 5는 clip20 동영상의 원본 및 3가지 안정화 방법을 적용한 후에 나타나는 프레임별 이동 방향이다. 각 색깔은 양자화 된 각도를 의미하며, 높이는 동일방향으로 가장 많이 간 그룹의 개수이다. 본 실험에서 그룹의 총 개수는 72(12 × 6)개이며, 이중 절반이상 동일방향인 경우 영상이 움직인 것으로 판단하였다.

구현한 알고리즘을 동영상에 적용하여 구한 흔들림 지수는 Table 5-7과 같다.

Table 7의 결과를 보면 전체적인 이미지 안정화 성능은 ffmpeg, Elasty, iMovie 순으로 좋아짐을 확인할 수 있다. 전반적으로 상용 프로그램들이 이미지 안정화에 좋은 성능을 보임을 알 수 있으며, open source인 ffmpeg의 경우는 좀 더 개선의 여지가 있음을 알 수 있다. 그러나 몇몇 경우에는 다른 결과를 보이고 있는데, 예를 들어 clip21과 IMG\_5943의 경우 iMovie가 오히려 나쁜 결과를 보이고 있다. 이러한 원인은 이미지 안정화 알고리즘의 차이에 기인하는 것으로 판단된다. 전반적으로 iMovie의 경우 이미지

Table 4. list of videos (outdoors)

File name	Video	Camera	Object	Remarks
IMG_5943		Holding by hand	static	Walking
IMG_5955		Holding by hand	moving car	
IMG_5956		Holding by hand	moving car	Many shakes
IMG_5961		Holding by hand	static	Camera movement
IMG_5962		Holding by hand	static	Camera movement Many shakes



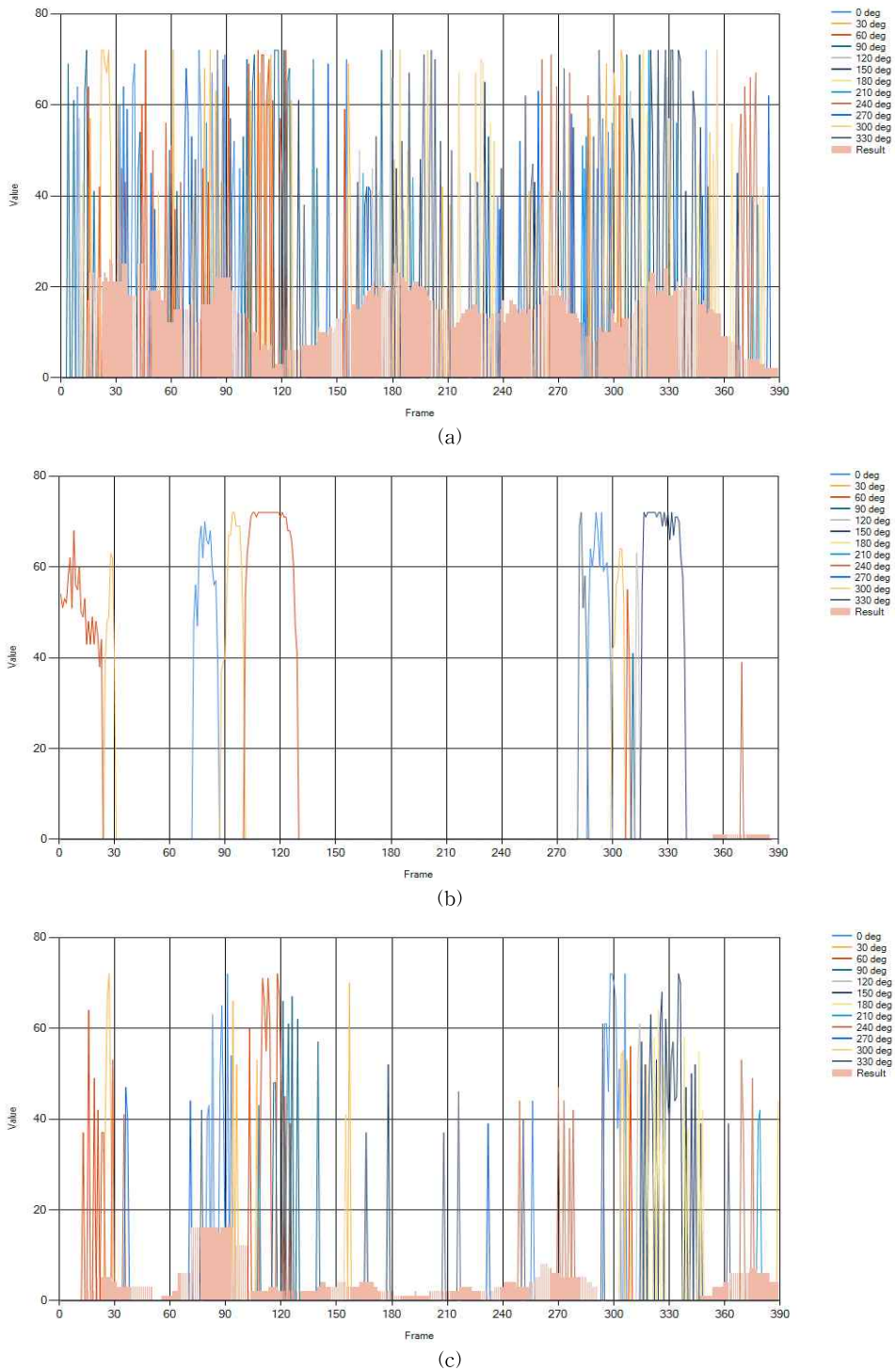
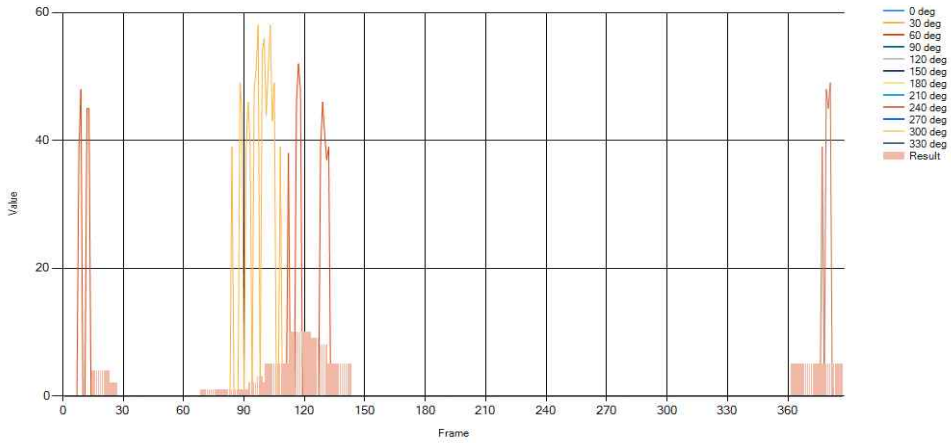


Fig. 5. Clip20. The color of the individual lines means the moving angle. (a) Original, Individual frames are moving randomly in several directions. (b) After iMovie, No frame movement or stabilization in the same direction. (c) After ffmpeg, Move to frame in multiple directions. (d) After Elasty, No frame movement or stabilization in the same direction



(d)

Fig. 5. Continued.

Table 5. Jitter index( $m_1$ =value/frames), Number of frames with shaking in the entire video

File name	Frames(Elasty)	Original	ffmpeg	Elasty	iMovie
clip19	330(328)	22	1	0	0
clip20	390(388)	187	47	20	1
clip21	646(644)	396	60	8	36
clip22	376(374)	3	2	0	0
IMG_5943	357(355)	41	37	23	42
IMG_5955	323(321)	0	5	12	0
IMG_5956	383(381)	107	53	21	7
IMG_5961	330(328)	0	1	0	0
IMG_5962	380(378)	15	2	0	0

Table 6. Jitter index( $m_2$ =value/31), Number of frames when the most shake is measured for 1 second

File name	Frames	Original	ffmpeg	Elasty	iMovie
clip19	31	11	1	0	0
clip20	31	26	17	10	1
clip21	31	29	16	8	12
clip22	31	3	2	0	0
IMG_5943	31	20	15	12	21
IMG_5955	31	0	2	10	0
IMG_5956	31	24	15	6	6
IMG_5961	31	0	1	0	0
IMG_5962	31	15	1	0	0

안정화에 손해가 있어도 영상 전체의 흐름을 안정화 시키려고 하는데 반하여, Elasty는 기본 옵션으로 안정화 시키는 경우 영상의 중심부분을 안정화시키는데 집중하고 한다. 결국 영상의 흔들림 형태에 따라 각 알고리즘이 다른 결과를 보이게 되는 결과를 가져

오게 된다. Fig. 6의 Elasty와 iMovie로 처리한 Clip 21의 512번째 프레임을 비교한 이미지이다. Fig. 6 (a),(b)의 원 안을 비교해보면 Elasty의 경우 회전을 시켜 검은색의 빈 공간이 그대로 있음을 볼 수 있다. 반면 iMovie의 경우 영상을 안정화 하면서 확대를

Table 7. Jitter index( $m$ ), Average of shaking index sum

File name	Original	ffmpeg	Elasty	iMovie
clip19	0.211	0.018	0	0
clip20	0.659	0.334	0.187	0.017
clip21	0.773	0.305	<b>0.135</b>	<b>0.221</b>
clip22	0.052	0.035	0	0
IMG_5943	<b>0.380</b>	<b>0.294</b>	<b>0.226</b>	<b>0.398</b>
IMG_5955	0	<b>0.040</b>	<b>0.180</b>	0
IMG_5956	0.527	0.311	0.124	0.106
IMG_5961	0	0.018	0	0
IMG_5962	0.262	0.019	0	0

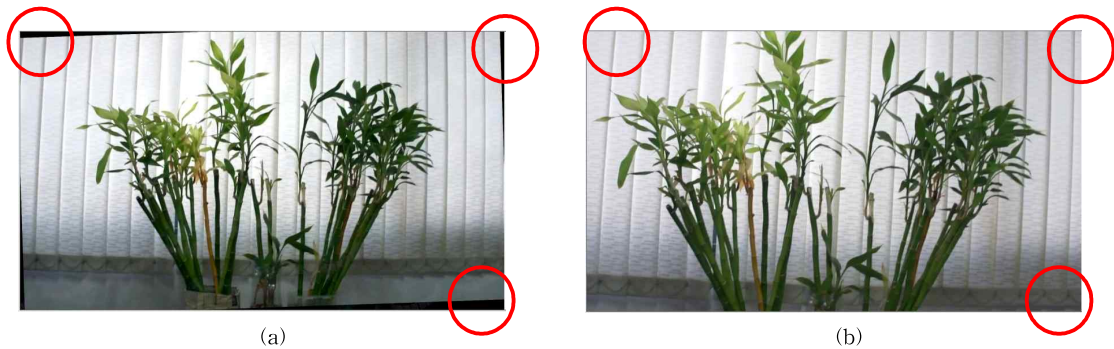


Fig. 6. Clip21. (a) Elasty processing example, Rotation occurs and black gaps appear (b) iMovie processing example, There is no black gap, it is enlarged,

시켜 전체 영상이 자연스러운 상태가 되도록 하였다.

### 5. 결 론

본 연구에서는 영상의 흔들림을 수치화하기 위해 흔들림을 모델링하였으며, 이 모델링으로부터 흔들림을 정의하였다. 흔들림 정의를 적용해 흔들림을 측정하기 위해 Optical Flow를 사용한 알고리즘을 만들었으면, 해당 알고리즘을 OpenCV와 C#을 사용해 구현하였다. 9가지 동영상에 대해 원본 및 3가지 동영상 안정화 알고리즘을 적용한 후 각 영상의 흔들림 지수를 측정해 보았으며, 이로부터 각 이미지 안정화 알고리즘의 성능을 객관적으로 비교 할 수 있었다.

본 연구에서 제안한 알고리즘과 이로부터 만들어진 흔들림 지수가 적절하였음을 확인 할 수 있었다. 향후 제안된 방법을 적용하면 영상의 흔들림을 기준으로 영상의 품질을 객관적으로 평가할 수 있으며, 실시간으로 영상의 품질을 평가 할 수 있을 것으로 기대된다.

### REFERENCE

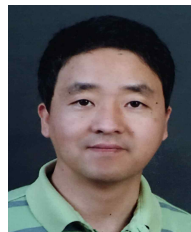
[ 1 ] Image Stabilization, [https://en.wikipedia.org/wiki/Image\\_stabilization](https://en.wikipedia.org/wiki/Image_stabilization) (accessed Jun., 29, 2017).

[ 2 ] N. Ejaz, W.I. Kim, S.I. Kwon, and S.W. Baik, "Video Stabilization by Detecting Intentional and Unintentional Camera Motions," *Proceeding of International Conference on Intelligent Systems, Modeling and Simulation*, pp. 312-316, 2012.

[ 3 ] L. Mercenaro, G. Vernazza, and C. Regazzoni, "Image Stabilization Algorithms for Video-Surveillance Application," *Proceeding of International Conference on Image Processing*, pp. 349-352, 2001.

[ 4 ] Y.G. Ryu and M.J. Chung, "Robust Online Digital Image Stabilization Based on Point-Feature Trajectory Without Accumulative

- Global Motion Estimation,” *Journal of IEEE Signal Processing Letters*, Vol. 19, No. 4. pp. 223–226, 2012.
- [5] J.H. Kim, Y.M. Baek, J.H. Yun, and W.Y. Kim, “In-Car Video Stabilization using Focus of Expansion,” *J. of Korea Multimedia Society* Vol. 14, No. 12, pp. 1536–1542, 2011.
- [6] S.H. Yang and F.M. Jheng, “An Adaptive Image Stabilization Technique,” *Proceeding of IEEE International Conference on Systems, Man and Cybernetics*, pp. 1968–1973, 2006.
- [7] STMicroelectronics, Optical Image Stabilization, [http://www.st.com/content/ccc/resource/technical/document/white\\_paper/c9/a6/fd/e4/e6/4e/48/60/ois\\_white\\_paper.pdf/files/ois\\_white\\_paper.pdf/jcr:content/translations/en.ois\\_white\\_paper.pdf](http://www.st.com/content/ccc/resource/technical/document/white_paper/c9/a6/fd/e4/e6/4e/48/60/ois_white_paper.pdf/files/ois_white_paper.pdf/jcr:content/translations/en.ois_white_paper.pdf) (accessed Jun., 29, 2017).
- [8] International Telecommunication Union, Methods for Subjective Determination of Transmission Quality, ITU-T Recommendation, <http://www.itu.int/rec/T-REC-P.800-199608-I> (accessed Jun., 29, 2017).
- [9] M. Niskanen, O. Silven, and M. Tico, “Video Stabilization Performance Assessment,” *Proceeding of IEEE International Conference on Multimedia and Expo*, pp. 405–408, 2006.
- [10] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of Physics Extended 10<sup>th</sup> Edition*, Wiley, Hoboken, New Jersey, 2014.
- [11] H.W. Baek, Y.J. Hur, M.G. Song, N.C. Park, Y.P. Park, K.S. Park, S.C. Lim, et al., “Development for OIS Actuator for Mobile Phone Camera,” *Journal of The Society Information Storage System*, Vol. 5, No. 1, pp. 8–13, 2009.
- [12] B.D. Lucas and T. Kanade, “An Iterative Image Registration Technique with and Application to Stereo Vision,” *Proceeding of International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.
- [13] J.Y. Bouguet, *Pyramidal Implementation of the Affine Lucas Kanade Feature Tracker Description of the Algorithm*, Intel Corporation, Technical Report, 1999.
- [14] Google Drive, Sample Video, <https://drive.google.com/drive/folders/0B-Ws5b8zpBH-OVJZS2JwSkk3NDA?usp=sharing>, (accessed Jun., 29, 2017).
- [15] iMovie, <https://itunes.apple.com/us/app/imovie/id408981434?mt=12>, (accessed Jun., 29, 2017).
- [16] ffmpeg, <https://www.ffmpeg.org/>, (accessed Jun., 29, 2017).
- [17] Elasty (Video Toolbox), <http://www.creaceed.com/elasty> (accessed Jun., 29, 2017).



#### 백 상 현

1990년 아주대학교 전자공학과  
학사 졸업  
1992년 아주대학교 전자공학과  
석사 졸업  
1999년 아주대학교  
컴퓨터공학과 박사 수료



#### 황 원 준

1999년 고려대학교 전자공학과  
학사 졸업  
2001년 고려대학교 전자공학과  
석사 졸업  
2016년 KAIST 전기 및 전자공  
학부 박사 졸업

2001년 2월~2016년 8월 삼성종합기술원 전문연구원  
2016년 9월 아주대학교 소프트웨어학부 조교수