

무인 항공기 촬영 동영상을 위한 실시간 안정화 기법

조현태 · 배효철 · 김민욱 · 윤경로[†]

[†]건국대학교 컴퓨터 공학부

Real-time Stabilization Method for Video acquired by Unmanned Aerial Vehicle

Hyun-Tae Cho, Hyo-Chul Bae, Min-Uk Kim and Kyoungro Yoon[†]

[†]Department of Computer Science and Engineering, Konkuk University

Abstract

Video from unmanned aerial vehicle (UAV) is influenced by natural environments due to the light-weight UAV, specifically by winds. Thus UAV's shaking movements make the video shaking. Objective of this paper is making a stabilized video by removing shakiness of video acquired by UAV. Stabilizer estimates camera's motion from calculation of optical flow between two successive frames. Estimated camera's movements have intended movements as well as unintended movements of shaking. Unintended movements are eliminated by smoothing process. Experimental results showed that our proposed method performs almost as good as the other off-line based stabilizer. However estimation of camera's movements, i.e., calculation of optical flow, becomes a bottleneck to the real-time stabilization. To solve this problem, we make parallel stabilizer making average 30 frames per second of stabilized video. Our proposed method can be used for the video acquired by UAV and also for the shaking video from non-professional users. The proposed method can also be used in any other fields which require object tracking, or accurate image analysis/representation.

Key Words : Video stabilization, Parallel processing, Savitzky-golay smoothing Filter, Affine motion model

1. 서 론

과학 기술의 발전으로 군사적 목적, 화재 방지, 재난 방지 등 다양한 목적을 위한 다양한 형태의 무인 비행체가 널리 연구되고 있다.

무인 비행체에 카메라를 탑재함으로써 더욱 많은 정보를 수집할 수 있지만, 소형 무인 항공기는 엔진 자체의 진동으로 인해 영상의 흔들림이 발생한다. 또한 기체 자체의 적은 무게로 인하여 환경적 요인의 영향을 많이 받고, 특히 바람에 의한 흔들림이 발생한다. 그러므로 무인 항공기에 탑재된 카메라 역시 같은 흔들림이 발생하고, 촬영된 영상도 흔들림이 발생하게 된다. 이러한 흔들림은 동영상의 품질을 저해하는 요소로 작용한다.

흔들림이 포함된 영상의 문제를 해결하기 위해 영상 안정화에 관한 많은 연구들이 있었다. 일반적인 영상 안정화 기법에는 기계적 영상 안정화 방법과 디지털 영상 안정화 방법이 있다.

기계적인 방식은 물체의 각속도를 측정할 수 있는 자이로 센서 등을 이용하며, 카메라 움직임의 각속도를 측정하여 카메라 렌즈를 반대방향으로 보정하여 영상의 흔들림을 보상하게 된다. 보상 방법은 렌즈의 위치를 보정 시켜주는 방법과 CCD (charge-coupled device) 센서의 위치를 보정 시켜주는 방법이 사용되고 있다.

디지털 영상 안정화 기법은 입력되는 영상 신호를 분석하여 인접한 두 프레임간의 움직임을 추정하고, 이를 이용하여 CCD 센서로부터 얻어진 영상 데이터를 보상하여 준다. 이러한 소프트웨어를 이용한 디지털 영상 안정화 기법에서 전역 움직임 추정 방식은 핵심기술로 자리 잡고 있다[1]. 하지만 아직 완벽한 안정화기

[†]E-mail : yoonk@konkuk.ac.kr

는 발표되지 않고 있다.

영상 안정화에서 가장 중요하고, 안정화의 품질을 결정하는 것은 카메라의 움직임 예측이다.

일반적으로 쓰이는 방법은 연속된 두 프레임간의 옵티컬 플로우를 계산하여 카메라의 움직임을 예측하는 방법이다.

옵티컬 플로우는 프레임에서 임의의 특징 점들의 움직임만을 계산하는 희소 옵티컬 플로우와 프레임의 모든 픽셀의 움직임을 계산하는 밀집 옵티컬 플로우가 있다.

희소 옵티컬 플로우는 특징 점들만의 움직임을 계산하기 때문에 계산속도가 빠르다. 하지만 카메라의 움직임을 예측하는데 있어서 임의의 점들의 움직임만으로 예측하기는 부족한 면이 있다.

본 논문에서는 무인 항공기에서 촬영한 동영상을 안정화하여 부드러운 영상을 얻는 것을 목표로 하고 있으며, 특히 옵티컬 플로우 중에서 밀집 옵티컬 플로우를 사용하여 카메라의 움직임을 더 정확하게 예측한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를, 3장에서는 실시간 영상안정화기의 구현을, 4장에서는 실험 및 결과, 5장에서는 결론 및 향후 과제를 다룬다.

2. 관련 연구

Fig. 1은 일반적인 영상 안정화기의 구조를 보여준다. 일반적인 영상 안정화기는 두 개의 연속한 프레임간의 로컬 모션을 예측한다. 예측된 로컬 모션을 이용하여 프레임의 전역 모션 벡터를 계산한다. 그 후 떨림을 제거하기 위해 모션 벡터를 부드럽게 한다. 안정화된 영상은 떨림이 제거된 모션 벡터만큼을 보상하여 얻을 수 있다.

Chang이 제안한 안정화기[2]는 희소 옵티컬 플로우를 이용하여 인접한 두 프레임간의 움직임을 예측하였

다. 또한 카메라의 움직임을 단순화된 어파인 모션 (simplified affine motion) 모델로 모델링 하였다.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ b & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix} \quad (1)$$

식 (1)은 단순화된 어파인 모션 모델을 나타낸다. 기존의 어파인 모델과의 차이점은 x축과 y축의 크기 (scale) 파라미터를 하나의 값으로 나타내고 (a), x, y의 변형(shear) 파라미터도 하나의 값으로 (b) 표현한다. Trimmed least square 메서드를 사용하여 이상치를 제거하고, 정규화 방법을 이용하여 어파인 행렬 값들을 부드럽게 해서 떨림을 제거하였다. 320×240 해상도의 총 76 프레임의 동영상으로 실험을 했고, 그 결과 펜티엄4 2.4GHz의 PC에서 초당 30프레임을 처리할 수 있는 성능을 보였다. 강건하고 효율적인 비디오 안정화 알고리즘을 제안하였으나, 카메라의 움직임을 예측할 때 희소 옵티컬 플로우를 사용하므로 처리속도는 빠르지만 밀집 옵티컬 플로우보다는 정확도가 떨어질 수 있다. 또한 단순화된 어파인 모션 모델을 기반으로 하기 때문에 x, y 축의 변화가 크게 오는 소형 무인 항공기 촬영 영상에서의 떨림 현상을 제거하기 힘들다는 단점이 있다.

Shen이 제안한 안정화기[3]는 무인 비행체에서 촬영한 영상의 원하지 않은 움직임을 제거 하고 안정화된 영상을 고속으로 만드는 실시간 알고리즘을 제안했다. Circular block matching 알고리즘을 사용하여 카메라의 움직임을 예측하고, 카메라의 움직임을 어파인 모델로 모델링 하였다. Shen의 논문에서는 프레임과 프레임간의 시간 간격은 40 ms정도로 작기 때문에 두 영상간의 움직임이 작다고 판단을 하였다. 그러므로 어파인 모델에서 크기 파라미터(a)는 1로 고정을 시키고, 카메라의 정상적인 움직임, 즉 의도된 움직임은 천천히 일정한 률을 가지고 이동을 하고, 반면 의도하지 않은 움직임은 빠르고 랜덤 하게 이동한다고 가정했다. Shen이 제안한 안정화기는 크기 파라미터를 1로 고정시키기 때문에 확대, 축소가 빈번하게 발생하는 소형 무인 항공기 촬영 영상의 떨림은 보정할 수 없는 단점이 있다.

3. 영상 안정화기의 구현

3.1. 영상 안정화기의 구조

Fig. 2는 실시간 영상 안정화기의 흐름도이다. 현재 프레임과 이전 프레임의 안정화된 영상간의 옵티컬 플로우를 계산하고, 계산된 각 블록들의 옵티컬 플로우를

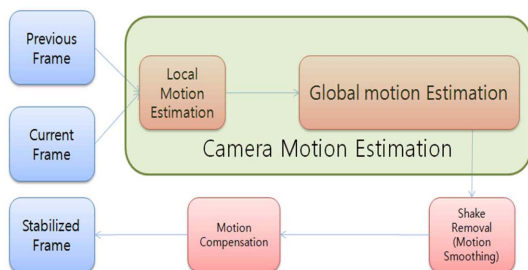


Fig. 1. Structure of the general video stabilizer.

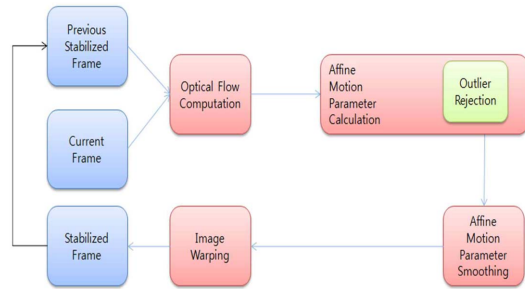


Fig. 2. Flowchart of video stabilizer.

최소 제곱법을 이용하여 어파인 파라미터를 결정한다. 이렇게 각각의 프레임에서 결정된 어파인 파라미터를 스무딩 필터를 이용하여 파라미터 값들을 부드럽게 만든다. 그 후에 스무딩된 어파인 파라미터를 이용하여 어파인 변환을 하여 안정화된 영상을 얻을 수 있다.

3.2. 옵티컬 플로우 계산

본 논문에서는 Gunnar Farnéback이 제안한 옵티컬 플로우 계산 알고리즘을 사용하였다[4]. 이 알고리즘은 2차원 밀집 옵티컬 플로우이며, 모든 픽셀간의 옵티컬 플로우를 계산하여 계산비용이 많이 소비되기는 하지만, 최소 옵티컬 플로우보다는 더 좋은 옵티컬 플로우를 얻을 수 있으며, 계산된 옵티컬 플로우는 카메라의 움직임을 예측하기 위하여 사용되므로, 정확한 옵티컬 플로우의 계산은 향상된 안정화 결과를 얻을 수 있다.

3.3. 어파인 모션 파라미터 계산

2절에서 계산된 옵티컬 플로우 값으로 어파인 모션 모델을 구할 수 있고, 본 논문에서는 카메라의 움직임을 어파인 모델로 모델링 하였다.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \quad (2)$$

식 (2)는 어파인 모션 모델을 표현한다. 식 (2)를 n개의 픽셀에 대해 풀어 쓰면 식 (3)과 같이 표현 할 수 있다.

$$\begin{bmatrix} x_1 & y_1 & 0 & 0 & 1 & 0 \\ & \vdots & & & & \\ x_n & y_n & 0 & 0 & 1 & 0 \\ 0 & 0 & x_1 & y_1 & 0 & 1 \\ & \vdots & & & & \\ 0 & 0 & x_n & y_n & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} x_1' \\ \vdots \\ x_n' \\ y_1' \\ \vdots \\ y_n' \end{bmatrix} \quad (3)$$

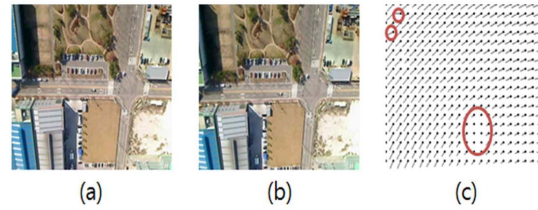


Fig. 3. Outlier of optical flow.

식 (2)에서 $\begin{bmatrix} x \\ y \end{bmatrix}$ 을 오른쪽 항으로 넘겨 (x, y) 의 좌표와 (x', y') 의 좌표의 오차를 함수로 표현을 하면 식 (4)와 같은 에러 함수를 정의 할 수 있다.

$$Error_i = |x'_i - (ax_i + by_i + e) + y'_i - (cx_i + dy_i + f)|, \quad i = 1, \dots, n \quad (4)$$

식 (4)에서 에러가 최소가 되는 어파인 모션 파라미터를 결정할 수 있다. 어파인 모션 파라미터는 (3)식에서 계산된 옵티컬 플로우 세트를 최소 제곱법을 이용하여 계산할 수 있다. 물론 식 (3)에서 계산된 값들 중에 이상치가 포함될 수 있다. 하지만 이러한 이상치는 3.4에 설명한 이상치 제거 과정을 통해 해결할 수 있다.

3.4. 이상치 제거

Fig. 3(c)는 인접한 두 개의 프레임, Fig. 3(a)와 Fig. 3(b)를 대상으로 옵티컬 플로우를 계산한 결과다. Fig. 3의 (c) 붉은색 원은 다른 위치의 옵티컬 플로우들과 다른 값을 가지고 있다. 이러한 값들을 이상치(outlier)라 정의하며, 이상치들은 3절에서 어파인 파라미터를 계산하는데 있어 민감한 요소로 작용을 한다. 더 정확한 어파인 파라미터를 계산하기 위해서는 이러한 이상치를 제거해야만 한다.

Fig. 3에서와 같이 옵티컬 플로우의 목적좌표와 3절에서 계산된 어파인 파라미터 값을 이용하여 다시 계산하였을 때 오차거리가 어느 정도 값보다 크다면 이상치로 판단을 한다. 이러한 이상치 판단 기준을 Fig. 5에서 보여주고 있다.

이상치의 제거 절차는 다음과 같다.

- 3절에서 계산된 옵티컬 플로우 세트로 어파인 파라미터를 계산한다.
- 옵티컬 플로우 세트의 destination좌표와 계산된 어파인 파라미터를 이용하여 재계산된 옵티컬 플로우 세트의 destination좌표와의 오차를 계산하여 두 점간의 오차를 계산한다.
- 평균 오차거리에 허용오차 값을 더한 값보다 큰

값들이 발견되면, 그 값들을 제외한 값들의 유티컬 플로우 셋을 이용하여 다시 계산을 한다.

4. 이상치가 발견되지 않을 때 까지 2와 3의 과정을 반복한다.

3.5. 어파인 모션 파라미터 스무딩

본 논문에서는 Savitzky-Golay 가 제안한 스무딩 필터[5]를 사용하여 어파인 모션 파라미터를 스무딩 한다. Savitzky-Golay 가 제안한 스무딩 필터는 이동평균의 윈도우 크기, 데이터의 위치에 따른 미리 계산된 가중치 값을 이용한 이동 평균의 한 종류이다. 본 논문에서는 윈도우 크기를 25로 설정하여 그에 해당하는 계수 값을 사용한다. 어파인 파라미터의 각 파라미터에 대해 스무딩을 하여 잡음을 제거하고 카메라의 의도된 움직임만을 얻어낼 수 있다.

3.6. 어파인 변환

스무딩된 어파인 파라미터 값을 이용하여 입력영상을 어파인 변환 하면 안정화된 영상을 얻을 수 있다. 어파인 변환은 회전, 확대 축소와 같은 기하학적 변환을 단순하게 모델링한다. 아래 식은 의 좌표를 공간으로의 어파인 변환 식을 나타낸다. 이차원 평면에서는 6개의 파라미터를 가진다. 임의의 영상의 모든 픽셀좌표가 식 (5)을 이용한 계산과정을 통해 대상 이미지의 픽셀좌표로 변환된다.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (5)$$

식 (5)의 각 파라미터 마다 기하학적 변환에 기여하는 특성이 있다. 파라미터 a, e는 확대, 축소와 같은 기하학적 변화에 영향을 미치며, 파라미터 b, d는 평면의 회전에 영향을 미친다. 어파인 파라미터 c, f는 각각 수평 및 수직방향으로의 이동을 나타낸다.

4. 실험 및 결과

본 장은 논문에서 제안한 알고리즘의 실험 결과를 보여준다. 실험환경은 Intel Core2 Quad(2.4GHz) CPU와 RAM 4GB가 장착된 Windows 7 64bit 환경에서 실험을 하였다. 또한 OpenCV 2.1 라이브러리를 사용하였고, 실험 동영상은 무인항공기로 촬영한 352×288의 해상도 총 300프레임 영상으로 실험 하였다.

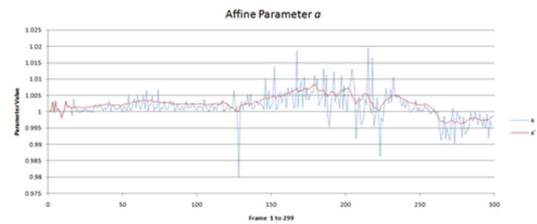


Fig. 4. Smoothing results of affine parameter.

4.1. 어파인 모션 스무딩

Fig. 4는 인접한 두 프레임간의 유티컬 플로우를 계산하고, 계산된 값을 이용하여 어파인 파라미터를 구한 결과를 그래프로 표현했다. 본 논문은 Savitzky-Golay 가 제안한 가중 이동 평균 필터를 사용한다. 붉은색 실선은 어파인 파라미터를 스무딩하기 이전의 값들이고 푸른색 점선은 스무딩한 결과를 보여준다. 푸른색 점선이 12프레임 이전에 표시되지 않은 이유는 스무딩 알고리즘에서 현재 프레임에서 뒤의 12프레임을 참조하여 스무딩하기 때문에 12프레임의 딜레이가 발생한 후 스무딩 과정을 거치기 때문이다. 실험결과 푸른색 점선의 고주파 부분은 필터링 되고 저주파 영역만 통과시켜 스무딩된 파라미터 값들을 구할 수 있다.

4.2. 처리속도의 측정

이와 같이 구성된 안정화기의 성능은 한 프레임을 처리하는데 평균적으로 0.094초의 시간이 소비되었다. 초당 10프레임의 프레임정도의 성능을 나타내고, 초당 10프레임은 실시간 영상 안정화에 부적합한 성능이다. 이러한 문제를 해결하기 위해 병목현상이 발생하는 부분을 병렬 처리해야 한다. 병목현상이 발생하는 부분을 찾고, 데이터의 의존성을 제거하여 독립적인 처리를 가능하게 영상 안정화기의 구조를 재설계 했다. 실험결과 병목현상이 발생하는 부분은 유티컬 플로우 계산을 하는 부분이다. 이 부분은 한 프레임을 처리하는 총 소비 시간 중 96%정도를 차지했다.

4.3. 실시간성을 위한 개선

병목현상은 3절의 실험결과를 통해 유티컬 플로우를 계산하는 과정에서 발생한다는 것을 알 수 있었다. 실시간 영상 안정화를 위해 유티컬 플로우를 계산하는 과정을 병렬 처리를 통해서 해결을 할 수 있다. 기존의 영상 안정화기는 이전 프레임과 그 이전의 안정화된 프레임간의 유티컬 플로우를 계산하는 구조로 되어 있다. 때문에 이전 프레임의 안정화된 영상이 계산되어야만 다음 처리가 가능한 구조로 되어 있다. 이를 해결

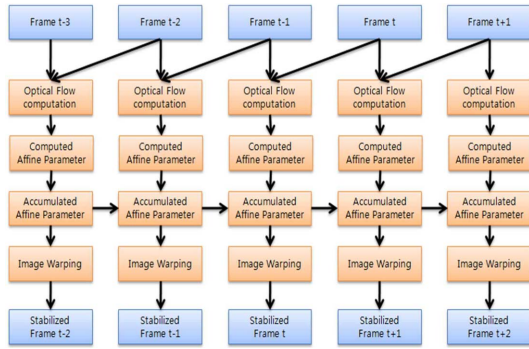


Fig. 5. Structure of the video stabilizer for real-time.

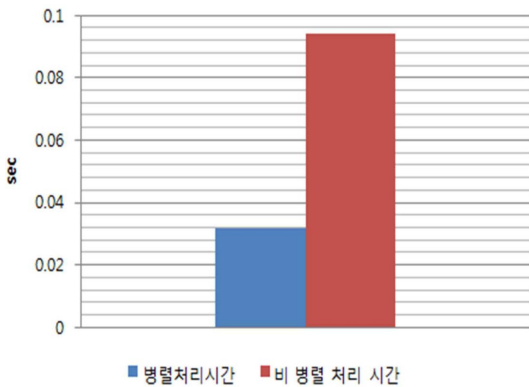


Fig. 6. Parallel Processing performance measurements.

하기 위해 본 논문에서는 Fig. 5와 같이 어파인 파라미터 값들을 누적시켜 전전 프레임을 안정화하는 과정에서 누적된 값을 반영하여 어파인 파라미터를 스무딩한다. 어파인 파라미터를 스무딩하는 과정에서 현재 프레임에서 +12프레임부터 -12프레임들의 어파인 파라미터가 필요하므로, 어파인 파라미터 스무딩 이전의 처리과정까지를 병렬처리의 범위로 설정을 한다.

Fig. 6은 병렬처리 이전과 이후의 성능 나타낸다. 처리성능의 측면에서 약 3배정도의 성능 향상을 보이고 있다. 병렬처리 이전에는 초당 10개의 프레임을 처리하였지만, 병렬처리 결과 실시간 영상 안정화에 적합한 성능이라고 할 수 있는 초당 30프레임을 처리하는 성능을 나타낸다.

4.4. 영상 안정화의 결과

본 논문에서는 본 논문이 제안한 병렬화된 영상안정화기의 성능을 측정하기 위해 vStab[6] 과 deshaker[7] 영상 안정화기와의 성능을 비교했다. vStab은 카나데-루카스-토마시 특징 추적기 기반의 비디오 안정화기이

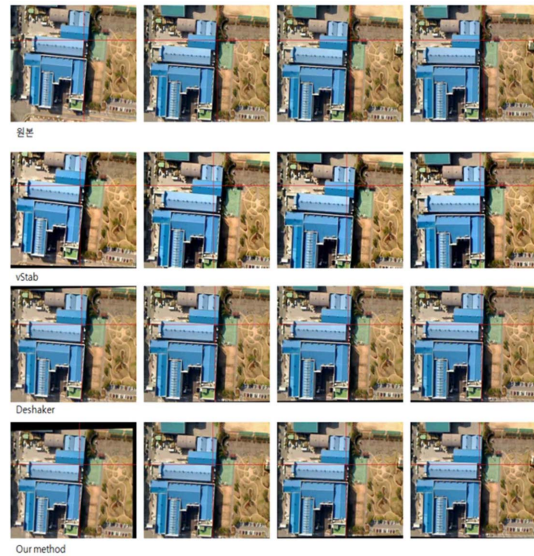


Fig. 7. Video stabilization results.

다. 성능을 비교하기위해 실제 무인항공기에서 촬영한 영상 2개와 임의로 제작한 영상 1개(30 frame @ 15 fps), 총 3개의 흔들린 영상으로 안정화 성능을 측정 하였다. 그 중 Fig. 7의 영상은 바람이 많이 부는 날 촬영한 영상이고, Fig. 8는 바람이 많이 불지 않은 날 촬영한 영상이다.

우선 무인항공기에서 촬영한 두 개의 영상의 결과 영상 중 임의의 연속된 4개의 프레임을 선택하여 비교해 보았다. Fig. 7과 Fig. 8은 왼쪽에서 오른쪽으로 시간 순서대로 연속된 4개의 프레임은 보이며, 위에서 아래의 순서로 원본 영상, vStab의 결과영상, deshaker의 결과 영상, 본 논문이 제안하는 결과 영상이다.

Fig. 7은 촬영 영상의 31프레임부터 34프레임까지 총 4개 프레임이다. 촬영 영상의 카메라 움직임이 오른쪽으로 발생하는 것을 알 수 있다. 즉, 붉은 색 선의 교차점이 건물 지붕의 모서리 한 점에서 이동하는 것을 알 수 있다. 원본 영상에서 두 번째 프레임, 세 번째 프레임과 네 번째 프레임의 지붕 위의 한 점과 붉은색 교차점과의 거리가 각각 31픽셀, 30픽셀 25픽셀이다. vStab의 두 점과의 거리는 각각 26픽셀, 29픽셀, 24픽셀이다. Deshaker의 두 점과의 거리는 각각 3픽셀, 6픽셀, 8픽셀이다. 마지막으로 본 논문이 제안하는 안정화기의 두 점과의 거리는 각각 3픽셀, 4픽셀, 6픽셀이다. 실험결과 바람이 부는 영상에서 본 논문이 제안하는 안정화기가 가장 좋은 성능을 보이고 있다.

Fig. 15는 촬영 영상의 178프레임부터 181프레임까

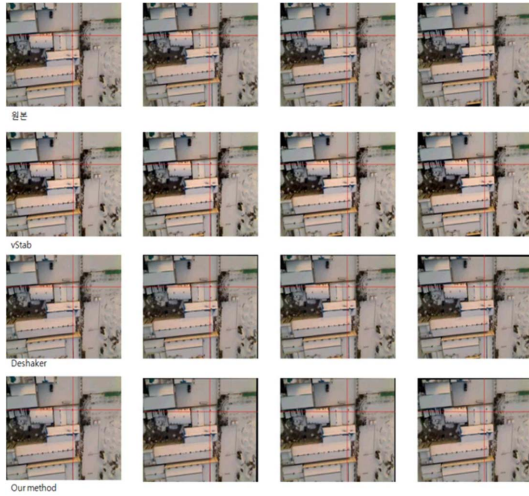


Fig. 8. Video stabilization results.

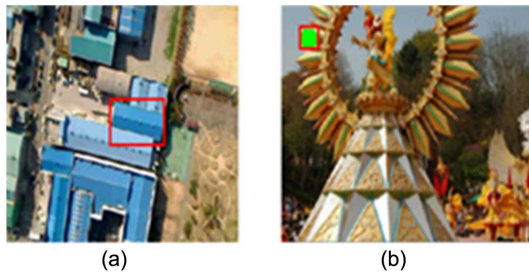


Fig. 9. Video stabilization results.

지 총 4개 프레임이다. 촬영 영상의 카메라 움직임이 왼쪽 아래로 움직이는 것을 알 수 있다. 즉, 붉은 색 선의 교차점이 건물 지붕의 한 점에서 이동하는 것을 알 수 있다. 원본 영상에서 두 번째 프레임, 세 번째 프레임과 네 번째 프레임의 지붕위의 한 점과 붉은색 교차점과의 거리가 각각 11픽셀, 10픽셀 15픽셀이다. vStab의 두 점과의 거리는 각각 3픽셀, 7픽셀, 11픽셀이다. Deshaker의 두 점과의 거리는 각각 2픽셀, 6픽셀, 8픽셀이다. 마지막으로 본 논문이 제안하는 안정화기의 두 점과의 거리는 각각 4픽셀, 6픽셀, 10픽셀이다.

두 번째로는 Fig. 9와 같이 Fig. 7에서 사용된 영상 (a)와 임의로 제작한 영상 (b)에서 각각 빨간색 네모박스 안에 있는 하나의 객체를 선택하고, 원본 영상, vStab의 결과영상, Deshaker의 결과영상, 그리고 본 논문에서 제안한 안정화기법의 결과영상에서 선택한 객체를 추적한 뒤, 객체의 흔들림을 체크하여 성능을 측정 했다.

두 번째로는 Fig. 9와 같이 Fig. 7에서 사용된 영상

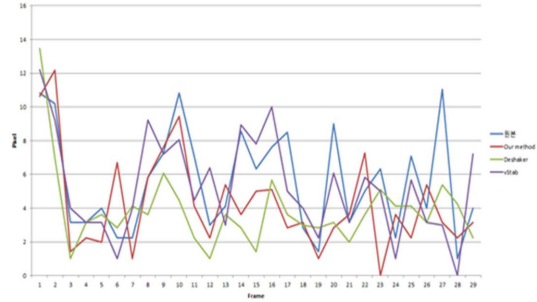


Fig. 10. Object moving distance of (a) video.

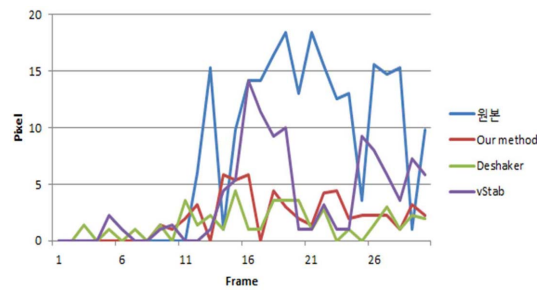


Fig. 11. Object moving distance of (b) video.

Table 1. Average and standard deviation of object moving distance

		원본	Our method	Deshaker	vStab
(A)	평균	5.5857	4.3095	3.8844	5.2833
	표준편차	4.2030	3.1653	2.0426	2.8504
(B)	평균	7.1250	1.8541	1.5027	3.3835
	표준편차	5.3578	1.7814	1.0085	4.1833

(a)와 임의로 제작한 영상 (b)에서 각각 빨간색 네모박스 안에 있는 하나의 객체를 선택하고, 원본 영상, vStab의 결과영상, Deshaker의 결과영상, 그리고 본 논문에서 제안한 안정화기법의 결과영상에서 선택한 객체를 추적한 뒤, 객체의 흔들림을 체크하여 성능을 측정 했다.

Table 1은 두 영상에 대해 원본과 각 안정화 알고리즘의 객체 움직임 픽셀의 평균값과 표준 편차를 나타내고 있으며, Deshaker가 가장 뛰어난 성능을 보여주었고, vStab이 가장 안 좋은 결과를 보여주었으며, 본 논문에서 제시하는 안정화기는 Deshaker에 근접하는 결과를 볼 수 있었다.

이러한 두 가지의 실험을 통하여 Deshaker의 안정화기와 본 논문이 제안하는 안정화기의 성능이 비슷하다는 것을 확인 할 수 있다. 하지만 Deshaker는 오프라인

에서 영상을 안정화를 하고 2Pass를 거쳐 안정화를 하지만, 본 논문이 제안하는 안정화기는 온라인에서 처리를 하며 한 번에 걸쳐 실시간 영상 안정화를 하는 시스템에는 본 논문이 제안하는 안정화기가 더 적합하다.

5. 결론 및 향후 과제

본 논문은 항공에서 촬영한 영상의 떨림을 보정하는 안정화기를 제안 했다. 밀집 옵티컬 플로우 계산을 통하여 계산비용은 많이 소비되지만 정확도를 높여 좀 더 안정화된 영상을 얻을 수 있었고, 병목부분을 병렬 처리하여 실시간성을 달성하였다. 어파인 파라미터를 스무딩 하는데 12프레임의 딜레이가 발생하고, 실험결과 미세한 떨림은 여전히 남아있다. 병렬처리를 위해 어파인 파라미터값을 계속 누적시키기 때문에 어파인 모델에서 스케일 파라미터의 값이 계속적으로 증가하여 일정 시간이 지난 후의 영상은 원 영상과 대비하여 일부 왜곡되는 현상이 남아있다. 영상 안정화 연구는 계속적으로 연구는 진행 되고 있으나, 완벽에 가까운 안정화기는 아직까지 발표되지 않고 있다. 향후 과제는 떨림 보정과 어파인 변환에 의해 정의 되지 않은 영역을 복원하는(inpainting) 방법을 적용한 안정화기의 구현이다. 또한 카메라의 움직임을 모델링 할 때 6개의 파라미터를 사용하여 x축과 y축의 스케일 값이 다르게 적용되어 왜곡 결과가 더욱 부각되는 경향이 있다. 이러한 문제는 단순화된 어파인 모델을 사용하여 해결할 수 있을 것이다.

Acknowledgements

This research was supported by the Basic Science Research Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (Grant number: 2012006817).

참고문헌

1. Kim, J.H., "Fast digital image stabilization based on edge detection", Korea University, Master Thesis, 2009.
2. Chang, H.C., Lai, S.H., and K.R. Lu. "A robust real-time video stabilization algorithm", Journal on Visual Communications and Image Representation, 17(3):659-673, 2006.
3. Shen, H., Pan, Q., Cheng, Y., and Yu, Y., "Fast Video Stabilization Algorithm for UAV", in proceeding of the IEEE International Conference on Intelligent Computing and Intelligent Systems, Vol. 4, pp. 542-546, 2009.
4. Gunnar Farnebäck, "Two-frame motion estimation based on polynomial expansion,"in Proc. of the 13th Scandinavian Conf. on Image Analysis, June-July 2003, vol. 2749 of LNCS, pp. 363-370.
5. Savitsky, A., Golay, M.J.E., Smoothing and differentiation of data by simplified least squares procedures, Analytical Chemistry, 1964.
6. vStab - KLT Video Stabilizer: <http://vstab.sourceforge.net/>
7. deshaker: <http://www.guthspot.se/video/deshaker.htm>

접수일: 2014년 2월 12일, 심사일: 2014년 2월 27일,
게재확정일: 2014년 3월 20일