

# Rolling Shutter 환경에서의 동영상 모션벡터 예측

조재진 \*이성원

광운대학교 컴퓨터 공학과

[ragaza@kw.ac.kr](mailto:ragaza@kw.ac.kr) \*[swlee@kw.ac.kr](mailto:swlee@kw.ac.kr)

## Moving Picture Motion Vector Estimation in the Rolling Shutter Environment

Cho, Jae-Jin \*Lee, Seong-Won

Kwangwoon University

### 요약

모바일 영상 기기들에 사용되는 저전력, 고품질의 이미지 센서는 Rolling Shutter 방식을 사용하는 CIS(CMOS Image Sensor)이다. 구조적 특성에 의해 Rolling Shutter 방식을 사용하는 CIS는 촬영 환경에 의해 결과물에 왜곡을 일으키게 된다. 본 논문은 Rolling Shutter에 의한 왜곡을 분석하고 촬영된 영상의 Rolling Shutter 왜곡 모션벡터를 예측하는 알고리즘을 제안하였다. 또한 제안 알고리즘의 정확성을 높이기 위해 Sub-block과 Kalman-filter를 적용하여 Sub-pixel 단위로 계산하여 기존의 Lucas-Kanade 알고리즘 보다 효율적인 모션벡터 예측 알고리즘을 제안한다. 또한 보다 정확한 성능 변화를 추적하기 위하여 MSE 방식을 사용하여 비교 분석 하였다.

## 1. 서론

영상센서는 CCD(Charge Coupled Device)와 CIS(CMOS Image Sensor)로 구분된다. 대부분의 모바일 영상 기기들은 전력소모와 반응 속도를 고려하여 상대적으로 좋은 CIS를 사용한다. CCD는 CIS에 비해 다양한 환경에서 좋은 화질을 얻을 수 있지만 최근 기술의 발전으로 CIS도 뛰어난 화질을 구현하고 있다. 그렇지만 CIS는 구조적 특성 때문에 Rolling Shutter 방식을 사용한다. Rolling Shutter 방식은 좌측 상단부터 순차적으로 행을 읽어 들이는 방식이다. 이런 셔터 방식은 각 행의 시간 차이를 발생시키게 되어 촬영 영상이 왜곡을 일으킬 수 있는 환경이 만들어 진다.

CIS의 Rolling Shutter 방식으로 촬영된 영상은 카메라의 움직임, 촬영자의 손떨림, 물체의 움직임 그리고 각 행의 시간차와 결합하여 영상 전체의 왜곡을 일으킨다. Rolling shutter 왜곡 보정에 대한 기존의 이론적 연구는 Rolling shutter에 대한 모델링과 Lucas-Kanade (L-K) 알고리즘을 이용하여 X, Y축 모션 벡터를 구하는 방법이 있다. 하지만 이 방법은 Rolling shutter에 의한 왜곡을 찾기 위해 많은 연산 과정이 필요하다.

본 논문에서는 Rolling shutter로 촬영된 영상의 행간의 상관도를 구하고 그 값을 통해 각 행의 시간적 차이에 의한 영상의 좌우 움직임의 크기를 계산하여 라인 Projection을 이용하여 상하의 움직임을 계산하는 기존의 방법<sup>[4]</sup>을 Sub-pixel로 확장 적용 하였다. 또한 각 라인의 Block 화를 통한 연산량 감소와 연산과정의 오차를 줄이기 위한 Kalman-Filter에 대한 실험을 추가하여 보다 적합한 임계값을 구하였다. 적용해 보았다. 이 알고리즘은 기존의 알고리즘에 비해 연산량을 크게 줄일 수 있고 왜곡보정의 정확성을 유지할 수 있다. 이에 대한 기존 알고리즘과의 보다 정확한 비교를 위하여 보다 많은 실험 데이터에 대한 MSE를 비교 분석 하여 제시하였다.

## 2. 본론

### 가. Rolling Shutter Distortion

모바일 장치에 사용되는 CIS는 저전력과 낮은 단가 등의 이유로 3개의 트랜지스터를 가지는 3T-CIS를 사용한다. 3T-CIS는 구조상 축적된 전하의 위치 저장 공간이 거의 없기 때문에 센서의 일정 영역이 노출되는 동안 이미지를 저장하고 노출이 끝난 상위 영역을 읽어 들이는 Rolling shutter 방식을 사용한다.

Rolling shutter는 좌측상단에서부터 시작하여 각 라인별로 노출 하며 촬영물은 각 라인이 각각 다른 시간에 촬영 된다. 각 라인의 노출 시간은 QVGA(320x240), 30프레임의 영상을 촬영할 때 “1/30x240 Line per Sec”가 된다. 각 라인의 촬영 시간 차이는 카메라의 움직임, 촬영자의 손 떨림 그리고 촬영 물체의 움직임과 합쳐져 왜곡을 일으킨다. 이때 나타나는 왜곡현상이 Rolling shutter distortion<sup>[1]-[4]</sup> 혹은 Rolling shutter effect 이다.

### 나. Lucas-Kanade 알고리즘

Rolling shutter distortion 보정을 위한 기존알 고리즘은 Lucas-Kanade (L-K) 알고리즘<sup>[3]</sup>이다. Lucas-Kanade 알고리즘의 수행 과정은 RGB의 Color 영상을 Gray-scale로 바꾸고, 영상의 고정된 16개의 영역에 대해서만 연산을 수행한다. 이후 검출된 16개의 모션 벡터는 Motion vector selector를 거쳐 하나의 Motion vector를 선택 하게 된다.

$$E(h) = \sum_X [F(X+h) - G(X)]^2 \quad (1)$$

식 (1)에서는  $F(X)$ 와  $G(X)$ 로 정의된 두 이미지를 LK(Lucas-Kanade) 알고리즘을 이용하여 error를 검출하는 식이다. 여기에서  $X = (x, y)$ 로 정의되고  $h$ 는 shift된 크기이다.

## 다.. 제안 알고리즘

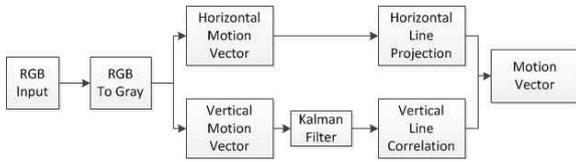


그림 1. 제안알고리즘의 순서도

Fig. 1. Block diagram of proposed algorithm

Lucas-Kanade (L-K) 알고리즘의 문제점은 많은 연산량이다. 제안하는 알고리즘은 연산 속도를 개선하기 위해 라인별 상관성(Correlation)과 투영기법(Projection)을 이용하였다.

제안 알고리즘은 원본 영상을 Gray-scale로 변환하고 변환된 영상은 각각 Vertical line correlation과 Horizontal line projection의 연산을 수행하여 Motion vector를 결정한다. 이때 Kalman filter를 이용하여 Vertical line correlation의 더 정확한 결과를 얻을 수 있게 한다.

Rolling shutter 왜곡 영상은 좌우 움직임에 대한 왜곡이 가장 심하기 때문에 Vertical line correlation 과정에서 많은 연산이 필요하다. 그래서 Vertical line correlation 과정은 Pixel 단위의 연산과 Sub-block 단위의 연산 방식을 사용해 각각의 연산 속도를 비교 하였다.

식 (2),(3)은 각각 Horizontal motion vector와 Vertical motion vector를 구하기 위해 상관성을 계산하는 수식이다.<sup>[4]</sup> 연산량을 줄이기 위해 상관성은 Absolute of subtraction의 형태로 계산하였다. 계산 범위는 상하좌우 20pixel범위 이며 이는 사람의 손 떨림의 범위가 0 ~ 30 Hz 의 대역이기 때문이다.

$$\Delta x = \min \left\{ d_i \left| \sum_{x=1}^h |a(x,y) - a(x+i,y+1)| \right| \right\} \quad (2)$$

, for  $i = -10, \dots, 10$

$$\Delta y = \min \left\{ d_i \left| d_i = \sum_{y=1}^v \left| \sum_{x=1}^h a(x,y) - \sum_{x=1}^h b(x,y+i) \right| \right| \right\} \quad (3)$$

, for  $i = -10, \dots, 10$

위의 Pixel을 이용한 계산 방법은 영상에 물체와 배경의 움직임이 다를 경우 여러 움직임들에 의한 오차가 생긴다. 이런 오차를 줄이기 위해 Sub-block 의 연산을 수행 하였다. Sub-block 연산은 영상내의 물체와 배경을 분리 시켜 물체의 움직임에 의한 흔들림을 제거 하여 오차를 줄일 수 있게 한다.

Kalman-filter<sup>[4]</sup>는 바로 이전 시간에 측정된 값을 토대로 현재의 값을 추정한다. 제안된 알고리즘에서는 모션벡터 예측의 정확성을 높이기 위하여 Vertical line correlation 연산 결과들 중 예측 값과의 오차가 큰 데이터를 제거 하였다. 식 (5)는 Kalman-filter 함수 이다. 오차의 판단 및 제외 기준은 추정된 값과 계산된 값의 차이가 3 pixel 이상 크게 벌어지는 경우이다.

## 3. 실험 및 실험 결과

실험에 이용한 영상은 Rolling shutter에 의한 왜곡 현상이 나타나 는 환경을 촬영하여 실험하였다. 촬영환경은 QVGA(320×240), 60fps 이며, 사용된 카메라는 Casio exlim EX-F1이다.

본 실험은 각 영상 마다 10 프레임씩을 사용하였다. 실험은 픽셀 단위의 모션벡터 검출, Sub-block (각 8, 16 pixel)을 이용한 모션벡터 검출, 그리고 각 알고리즘에 Kalman-filter를 적용하여 실험으로 구분 하여 진행 하였다. 또한 각 알고리즘의 연산 시간을 측정하여 비교하였다.



영상 (a)

영상 (b)

그림 2. 실험 원본 영상

Fig. 2. The original test scenes.

표 1과 2는 그림2. 영상 (a),(b)에 대한 각 알고리즘의 실 측정 결과이다. 대체로 측정치(Measurement)의 경향을 따라 가는 것을 확인할 수 있었다. Kalman-filter가 적용된 알고리즘은 각 표에서 확인해 볼 수 있듯이 프레임이 넘어 갈수록 실측정치와 근접해 지는 것을 볼 수 있었다. 각 프레임의 모션벡터는 MSE (Mean Square Error)를 이용하여 비교 하였다. 실측값을 기준으로 각 알고리즘의 결과 값의 차를 이용하여 계산하였다. MSE가 작을수록 더 좋은 성능을 보이는 것으로 볼 수 있다.

표 1. 모션벡터 검출 결과 - 영상 (a)

Table 1. Result of motion vector detection - scene (a)

|                       |          | Score    |          |          |          |          |          |
|-----------------------|----------|----------|----------|----------|----------|----------|----------|
|                       |          | (1)      | (2)      | (3)      | (4)      | (5)      | (6)      |
| F<br>r<br>a<br>m<br>e | 0        | (2510)   | (2312)   | (2311)   | (2111)   | (2111)   | (2313)   |
|                       | 1        | (-1224)  | (-1423)  | (-1322)  | (-1122)  | (-1323)  | (-1222)  |
|                       | 2        | (2235)   | (2333)   | (2133)   | (2233)   | (2235)   | (2635)   |
|                       | 3        | (3030)   | (3231)   | (3130)   | (3031)   | (3234)   | (3133)   |
|                       | 4        | (4015)   | (4113)   | (3813)   | (3613)   | (3413)   | (3520)   |
|                       | 5        | (2312)   | (2111)   | (1912)   | (1712)   | (1811)   | (1508)   |
|                       | 6        | (-25-13) | (-27-12) | (-25-12) | (-26-12) | (-23-15) | (-21-11) |
|                       | 7        | (-40-10) | (-33-07) | (-37-06) | (-35-06) | (-34-06) | (-33-08) |
| 8                     | (-50-30) | (-51-31) | (-48-31) | (-44-31) | (-52-33) | (-54-35) |          |
| MSE                   |          | .        | (014015) | (016014) | (028015) | (0302)   | (03028)  |

표 2. 모션벡터 검출 결과 - 영상 (b)

Table 2. Result of Pixel motion vector detection - scene (b)

|                       |         | Score   |         |         |          |          |         |
|-----------------------|---------|---------|---------|---------|----------|----------|---------|
|                       |         | (1)     | (2)     | (3)     | (4)      | (5)      | (6)     |
| F<br>r<br>a<br>m<br>e | 0       | (3020)  | (2313)  | (2113)  | (2013)   | (2013)   | (3021)  |
|                       | 1       | (15-15) | (22-19) | (20-18) | (13-18)  | (16-19)  | (15-14) |
|                       | 2       | (33-23) | (31-28) | (32-28) | (31-28)  | (33-28)  | (35-24) |
|                       | 3       | (22-14) | (26-13) | (25-13) | (24-13)  | (23-13)  | (21-16) |
|                       | 4       | (32-15) | (34-12) | (34-12) | (32-12)  | (33-11)  | (31-14) |
|                       | 5       | (21-41) | (23-46) | (23-45) | (25-43)  | (24-44)  | (22-42) |
|                       | 6       | (4528)  | (4426)  | (4226)  | (4126)   | (4026)   | (4530)  |
|                       | 7       | (2023)  | (2323)  | (2322)  | (2322)   | (2122)   | (2024)  |
| 8                     | (28-21) | (36-21) | (36-21) | (34-21) | (33-22)  | (29-20)  |         |
| MSE                   |         | .       | (03803) | (04028) | (037023) | (028031) | (05012) |

## 4. 결론

본 논문에서는 기존의 Lucas-Kande(L-K) 알고리즘에 비해 단순한 구조를 가지면서 비슷한 성능을 가지는 알고리즘을 제안하였다. 실험 결과에서 대부분 1~2 Pixel 이내에서 측정치와 비슷한 결과가 나타났으며, MSE를 기준으로 보면 제안 알고리즘이 L-K 알고리즘에 비해 대체로 좋은 결과를 보이고 있었다.

## Acknowledgement

본 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국 연구재단의 기초연구사업 지원을 받아 수행된 것임 (2010-0028178 2010-0015441)

## 참 고 문 헌

- [1] C.-K. Liang and H. H. Chen, "Rolling shutter distortion correction," SPIE Proc. Visual Communication and Image Processing, Beijing, China, Jul. 2005.
- [2] Christopher Geyer, Marci Meingast, and Shankar Sastry, "Geometric models of rolling-shutter cameras," IEEE Workshop on Omnidirectional Vision 2005.
- [3] Jung-Bum Chun, Hunjoon Jung, and Chong-Min Kyung, "Suppressing Rolling-Shutter Distortion of CMOS Image Sensors by Motion Vector Detection", IEEE Transactions on Consumer Electronics, Vol. 54, No. 4, NOVEMBER 2008.
- [4] Jae-jin Cho and Seong-won Lee, "Rolling shutter 환경에서의 모션벡터 예측에 대한 연구", 제 23회 신호처리합동학술대회, 2010.10.