

Edge Detection과 Lucas-Kanade Optical Flow 방식에 기반한 디지털 영상 안정화 기법

Digital Image Stabilization Based on Edge Detection and Lucas-Kanade Optical Flow

이 혜 정¹, 최 윤 원², 강 태 훈³, 이 석 규[†]

Hye Jung Lee¹, Yun Won Choi², Tae Hun Kang³, Suk Gyu Lee[†]

Abstract In this paper, we propose a digital image stabilization technique using edge detection and Lucas-Kanade optical flow in order to minimize the motion of the shaken image. The accuracy of motion estimation based on block matching technique depends on the size of search window, which results in long calculation time. Therefore it is not applicable to real-time system. In addition, since the size of vector depends on that of block, it is difficult to estimate the motion which is bigger than the block size. The proposed method extracts the trust region using edge detection, to estimate the motion of some critical points in trust region based on Lucas-Kanade optical flow algorithm. The experimental results show that the proposed method stabilizes the shaking of motion image effectively in real time.

Keywords : Image Stabilization, Motion Estimation, Lucas-Kanade Optical Flow, Edge Detection

1. 서론

Digital Image Stabilization(DIS) 기법은 영상을 촬영할 때 이동을 하거나 손 떨림이 발생할 경우 영상의 흔들림을 디지털 영상 처리를 통해 제거하는 기술이다. DIS 기법은 흔들림이 있으나 안정된 영상을 필요로 할 경우(무인차량시스템제어, 자율주행로봇, 고배율의 카메라 등) 필수적으로 선행되어야 한다. 이러한 기술이 임베디드 시스템 등에 적용되기 위해서는 정확도 뿐 아니라 실시간 동작을 만족 시켜야 하는 경우가 많다. 기존의 디지털 영상 안정화 알고리즘들은 Block Matching^[7]을 이용하는 방법이 대부분이고, Diamond Search^[8]를 이용하는 방법, Optical flow estimation^[9] 방법 등이 있다. 이들 방법들은 영상의 전역 움직임 벡터를 구하는데 있어서 각기 제약 사항이 따른다.

본 논문에서는 기존의 Block Matching과 Block Matching의 속도 문제를 개선한 Diamond Matching Algorithm을 살펴보고, Edge detection을 통해 신뢰영역을 구하여 Lucas-Kanade Optical Flow Estimation 방법을 적용한 정확도가 높고 기존 방법에 비해 고속 동작이 가능한 새로운 디지털 영상 안정화 기법을 제안한다. 즉, 2장에서는 기존의 Block Matching 방식을 이용한 영상 안정화 기법에 대하여 조사하고, 3장에서는 본 논문에서 제안하는 Lucas-Kanade Optical flow 방식에 대하여 기술하고 이를 증명하기 위하여 4장과 5장에서는 각각 simulation과 실험을 수행하였다.

2. 기존의 Block Matching 방식을 이용한 영상 안정화 기법

영상 안정화 기법은 크게 두 부분으로 나눌 수 있는데 첫째는 연속된 영상에서 두 프레임 간의 움직임 벡터를 구하는 것이고, 둘째로는 앞에서 구한 움직임 벡터를 이용해 뒤쪽 프레임을 보상하여 영상을 안정화 하는 것이다.

Received: Jan. 21, 2010; Reviewed: Mar. 11, 2010; Accepted: Mar. 17, 2010

[†] 교신저자: 영남대학교 전기공학과 교수(Email : sglee@ynu.ac.kr)

¹ 삼성전자

² 영남대학교 전기공학과 석사과정

³ (재)포항지능로봇연구소 연구팀장

2.1 Block Matching 방식을 이용한 움직임 벡터추정

Block matching 방식은 t번째 프레임의 특정 위치에 놓인 블록이 t-1번째 프레임의 어느 위치에서 이동해 온 블록인지를 검색하는 방법이다.(단, block matching 방식에서 움직임 추정은 연속하는 두 프레임 상에서 2차원적 움직임만이 존재한다고 가정한다.) t번째 프레임인 I_t 의 좌표 (a, b)에서 $M \times N$ 크기의 블록이 있다고 가정하면, 이 블록과 t-1번째 프레임 I_{t-1} 에서 좌표 (a+x, b+y)에 위치한 블록 사이의 유사도를 평가하기 위해서 평균 절대값 차이(MAD:Mean Absolute Difference)를 이용하여 다음과 같이 계산한다.

$$MAD_{(a,b)}(x,y) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |I_t(a+i,b+j) - I_{t-1}(a+x+i,b+y+j)| \quad (1)$$

(1)의 식에서 평균 절대값 차이를 최소로 하는 변위(x, y) 값을 검사하는 블록 (a, b)의 움직임 벡터로 생각할 수 있으며, 블록 (a, b)에 대한 움직임 벡터를 $v(a, b)$ 로 나타내면 (2)와 같다.

$$v(a,b) = \operatorname{argmin} MAD_{(a,b)}(x,y) \quad (2)$$

그림 1은 위에서 설명한 Block Matching 방식의 움직임 벡터의 추정 과정을 순서대로 보여주고 있다. (a)는 t번째 프레임 영상에서 $M \times N$ 크기의 블록을 보여주고, (b)는 t-1번째 프레임 영상에서 (a)에서 나타난 $M \times N$ 크기의 블록과 가장 유사한 블록의 위치를 나타낸다. 그리고 (c)는 검색 윈도우 범위 안에서 해당하는 블록과 평균 절대

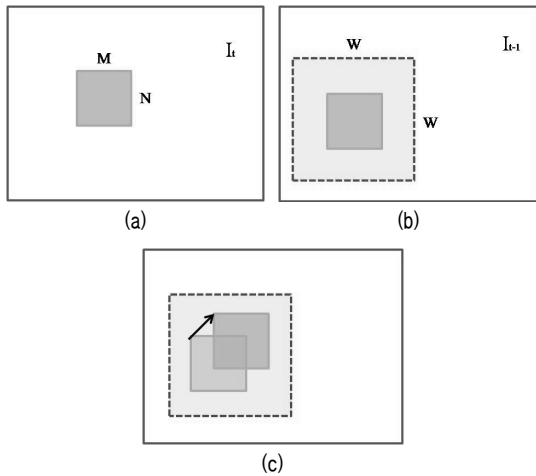


그림 1. 블록 매칭 알고리즘을 이용한 움직임 추정 방법

값 차이 값이 최소가 되는 위치의 블록을 검색하여 추정된 움직임 벡터의 위치를 보여준다.

그림 1에서 보듯이 Block matching 방법의 움직임 벡터 추정 과정에서 가장 정확한 값을 알아내기 위해서는 블록 주변의 모든 변위에 대해 평균 절대값 차이를 계산하는 전역 검색(Full Search) 방법을 사용해야 하는데, 이는 전역 검색을 할 때에 윈도우 크기가 커질수록 많은 연산 시간을 요구하므로 실시간 움직임을 구하는데 있어서 치명적인 단점으로 작용한다. 또한 블록 크기에 따라 벡터의 크기가 한정적이 되므로 움직임을 찾는 데 있어서 제약이 따른다.

2.2 Diamond Search Algorithm

Diamond Search Algorithm은 Block Matching이 윈도우 크기가 커질수록 연산 속도가 느려지는 것을 해결하기 위해서 국부적인 위치에 대해 평균 절대값 차이를 계산하는 방법이다. 이는 Large Diamond Search Pattern과 Small Diamond Search Pattern, 두 개의 검색 패턴을 가지며 그림 2에서 두 검색 패턴의 모습을 보여준다.

Diamond Search 방법은 먼저 LDSP 방법으로 최소 평균 절대값 차이를 구하여 이동한 후에 SDSP 방법을 이용하여 정확한 위치를 찾아낸다. Diamond Search Algorithm의 단계는 다음과 같다.

- (1) 처음 LDSP의 중심을 (0,0)으로 가정한다.
- (2) LDSP 9개의 위치에서 평균 절대값 차이를 계산한다. 만일 이 9개의 점 중에서 가운데 점이 최소 평균 절대값 차이로 나오게 되면 (4)번 단계로 이동하고, 그렇지 않으면 (3)번 단계로 이동한다.
- (3) (2)번 단계에서 구한 최소 평균 절대값 차이의 위치를 다시 LDSP의 중심으로 설정한 뒤, 또 다시 9개의 위치에서 평균 절대값 차이를 구한다. 만일 LDSP의 가운데 위치에서 최소 평균 절대값 차이가 발견되면 (4)번 단계로 이동한다. 그렇지 않으면, 이 단계를 반복한다.

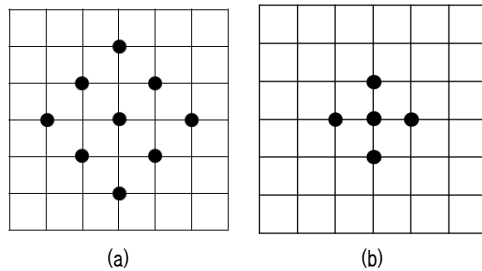


그림 2. 다이아몬드 검색 패턴 (a)LDSP, (b)SDSP

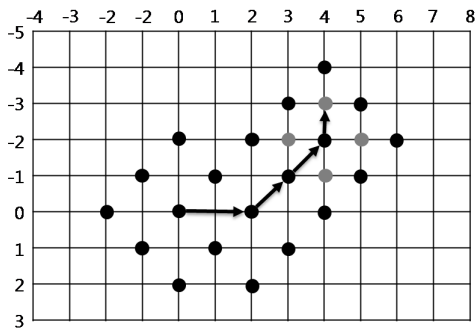


그림 3. Diamond Search 알고리즘을 이용한 움직임 추정

- (4) 5개의 점을 가지는 SDSP 방법으로 그 점들을 중심으로 한 평균 절대값 차이를 계산한다. 5개의 점들 중 최소 평균 절대값 차이 값을 나타내는 위치가 최종적으로 이동한 위치이다.

위의 Diamond Search Algorithm 단계를 그림으로 나타내면 그림 3과 같다. (0,0)지점에서 출발하여 LDSP 방법으로 9개의 점 중에서 가운데 위치의 점이 최소 평균 절대값 차이가 되는 지점을 찾은 후, SDSP 방법으로 정확한 위치를 찾는다. 그림 3에서는 이전 프레임의 (4, -3) 위치에서 이동해 왔음을 알 수 있다.

3. 제안하는 Lucas-Kanade Optical Flow 방식

본 논문에서는 Edge Detection을 통해 신뢰 영역을 선택한 후 Optical Flow Analysis를 적용하여 움직임 벡터를 추정하고 영상을 안정화 하는 방법을 제안한다.

3.1 움직임 벡터 추정

3.1.1 Edge detection을 이용한 신뢰 영역 선택

전역 움직임 벡터를 추정할 때에 영상의 전체 영역에서 지역 움직임 벡터를 검출하여 이용하면 시간이 증가할 뿐만 아니라 실제 움직임과는 관계없는 부분까지 고려되어 전역 움직임 벡터에 대한 신뢰성이 떨어지게 된다. 따라서 처리 속도를 향상시키고 특징이 뚜렷한 부분의 움직임 벡터를 추정하기 위해 전체 영상에서 Edge detection을 수행하여 Edge가 다수 포함되어 있는 영역만 골라서 검출하면 신뢰도가 향상된다고 할 수 있다. 본 논문에서는 Sobel 연산자와 Canny 연산자를 이용하여 영상의 경계를 검출하고 그에 따른 신뢰가 높은 영역을 결정하는 방법을 제안한다. 그림 4에서는 320x240 영상에서 영역을 가로 세로 각 40 pixels 크기를 가지도록 8x6개로 나누고 Sobel 연산자와 Canny 연산자의 임계치를 결정하였

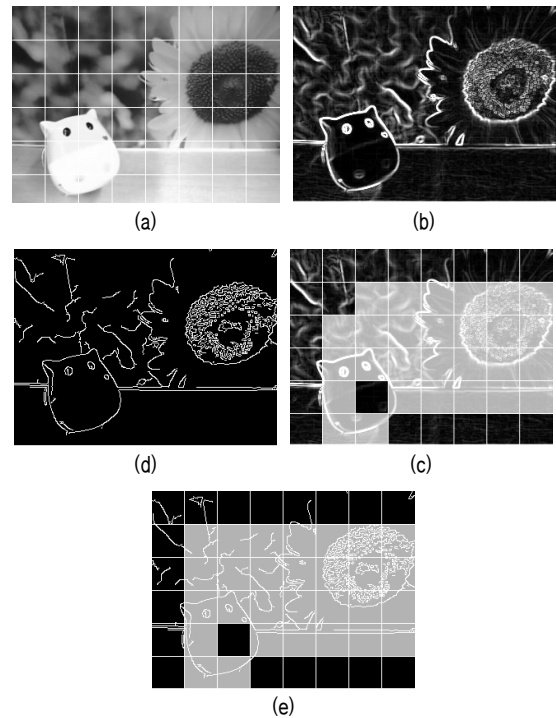


그림 4. Edge Detection을 통한 신뢰영역 추출

- (a) 원본영상 (b) 소벨 엣지 검출 (c) 소벨 엣지 신뢰 영역
(d) 캐니 엣지 검출 (e) 캐니 엣지 신뢰 영역

다. 이를 이진화를 통해서 흑백영상의 상태에서 각 영역 마다의 흰색 경계 빈도수를 구한 다음 특정 임계치 이상일 때 신뢰영역으로 선택했다. 본 논문에서는 흰색 영역의 빈도수를 450 pixels 기준으로 설정했다.

3.1.2 Lucas-Kanade Optical Flow를 이용한 특징점 추출

Optical Flow는 연속하는 두 프레임 간 차이를 통해 움직임을 추정하는 방법 중 하나로서 명도를 서서히 변화시켜 영상에서 발생하는 움직임의 방향과 속도를 벡터적으로 나타내는 방식이다.

이 때 모든 픽셀에 대하여 움직임 벡터를 구하면 엄청난 연산을 요구하므로 corner 지점에 해당하는 특징점에 대해서만 계산한다. 그리고 특징점이 많아지면 적절한 개수의 특징점을 선택하여 그 점들에 대해서만 Optical flow를 적용하면 되는데 일반적인 Optical flow 방식으로는 특징점의 개수를 지정할 방법이 모호하다. 그러므로 Gaussian Filter를 적용하여 특징점의 개수를 조절할 수 있는 Lucas-Kanade optical flow 방식을 채택하여 움직임을 추정하였다.

Lucas-Kanade optical flow 방식은 언급한 대로 Gaussian Filter를 적용해 특징점의 수를 줄이는 역할을

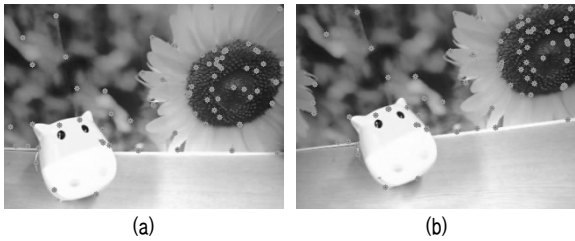


그림 5. LK Optical flow 방식을 적용하여 특징점을 검출한 영상 (a) t-1번째 프레임 (b) t번째 프레임

하며, 만일 Gaussian Filter를 N번 수행하여 N개의 영상이 있다고 한다면 원하는 개수의 특징점을 찾아내기 위해서 N, N-1 ..., 0번 수행한 영상을 순서대로 검색 한다. 이 때 특정 개수의 특징점을 찾으면 검색을 종료하고 특징점을 찾는 과정을 끝낸다. 그리고 Gaussian Filter를 적용한 횟수가 클수록 영상에서 검출되는 특징점의 수가 적으므로 움직임 추정 속도를 높이려면 Gaussian Filter를 최대한으로 적용하는 것이 좋다.

그림 5와 같이 실시간 영상에 적용하기 위해 모든 프레임에 대해서 특징점을 구하는 연산을 수행하지 않고 처음 프레임에 대해서만 특징점을 구한 뒤 다음 프레임부터는 검출된 특징점에 대해서만 LK Optical flow를 적용하여 움직임을 추정하였다.

3.2 영상 보정 알고리즘

앞에서 추정된 신뢰성이 높은 영역에 위치한 움직임 벡터를 이용하여 영상을 보정한다. 추정된 움직임 벡터를 이용해서 기본적으로 2차원적인 이동을 수행한 다음, 추정된 움직임 벡터에서 움직임이 발생하기 전의 좌표를 아래의 식을 이용하여 추정하였다.

$$\overrightarrow{movV} = \overrightarrow{curV} - \overrightarrow{preV} \quad (3)$$

$$\overrightarrow{preV} = \overrightarrow{curV} + (-\overrightarrow{movV}) \quad (4)$$

$$\theta = \arctan\left(-\frac{x_{cur} - x_{pre}}{y_{cur} - y_{pre}}\right) \quad (5)$$

(x_{pre}, y_{pre}) 과 (x_{mov}, y_{mov}) 는 각각 흔들림이 발생하기 전의 좌표 및 움직임 벡터이며, 영상의 중심을 원점으로 두고 각도를 추정한다. 움직임이 발생하기 전의 좌표를 구하게 되면 이를 바탕으로 영상이 회전한 각도를 구할 수 있는데 본 논문에서는 이동한 좌표를 아크 탄젠트에 대입하여 영상의 회전 각도를 추정하였다.

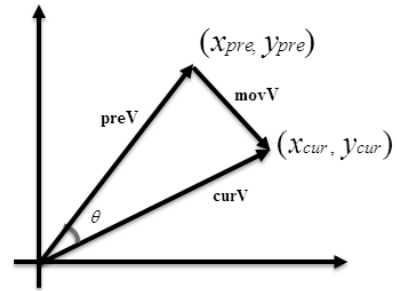


그림 6. 움직임벡터와 흔들린 각도 추정

추정된 회전 각도를 이용해 영상을 회전시키는데, 이 때 영상의 회전은 Warming을 이용하여 수행한다. Warming이란 영상을 늘리거나 크기를 조절하는 기하학적인 처리로서 순수한 스케일링과 달리 크기 변화의 정도가 영상 전체에 대해 균일하지 않은 알고리즘이다.

4. 시뮬레이션 결과

기존의 알고리즘과 제안된 알고리즘을 카메라를 한쪽으로 임의로 기울였을 때 안정화 정도를 시뮬레이션을 통하여 비교하였다.

그림 7 (a)는 원 영상을 나타내고 (b) 흔들림이 발생한 영상을 나타내며 전체적인 영상이 회전한 것을 알 수 있다. 그림 7 (c), (d)에서 붉은색 선은 이전 영상에서 현재 영상으로의 움직임 벡터를 나타내고 초록색 점은 이전 영상에서 현재 영상으로 이동한 위치 좌표를 나타낸다. (c)

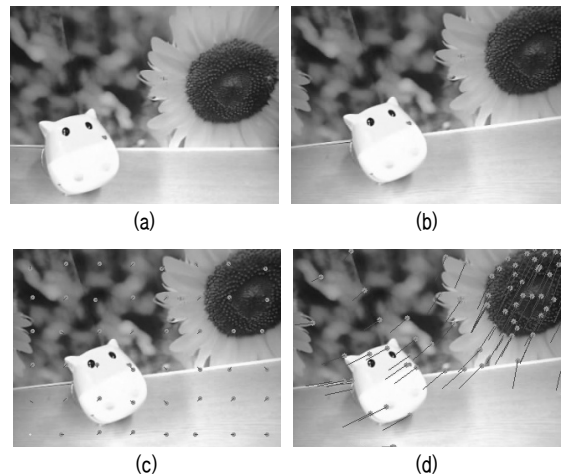


그림 7. 움직임 벡터

- (a) 이전 영상 (b) 흔들림이 발생한 현재 영상
- (c) Block matching 방식으로 움직임 벡터를 구한 영상
- (d) LK Optical flow를 이용해 구한 움직임 벡터 영상

는 Block matching 방법을 이용하여 추정된 움직임 벡터를 나타내고 (d)는 LK Optical flow estimation을 통해 추정된 움직임 벡터를 나타낸다. 그림에서 보듯이 Block matching 방법과 LK Optical flow 방식은 같은 영상에서도 움직임 벡터가 확연한 차이를 보인다. (c)에서는 추정된 움직임 벡터의 방향이 전체적으로 일정하지 않고 이동된 좌표도 거의 제한적이다. 반면 LK Optical flow estimation은 움직임 벡터의 방향이 전체적으로 비교적 일관성 있게 나타나는 것을 볼 수 있다.

그림 9는 이전 영상과 그림 8에서 안정화된 영상 사이의 차 영상을 나타낸 것이다. 그림 9 (a)는 이전영상과 현재영상의 차 영상을 나타낸 것이고 그림 9 (b)는 이전영상과 제안된 LK Optical flow 방법으로 보정한 영상의 차 영상이다. 그림 9 (b)에서 차 영상이 일부만 보이는 것은 흔들림이 발생한 영상 자체를 보정하게 되면 흔들림이 발생한 영상에서 원 영상과 공통 된 부분은 작아지기 때문에 비교하는 부분이 작아지게 되기 때문이다. 두 영상을 통해서 흔들림을 보정했을 경우 이전 영상과 유사한 영상을 얻을 수 있었다.



그림 8. 영상 안정화가 수행된 결과

- (a) 일반적인방법(Block matching)으로 움직임벡터를 구하여 안정화가 수행한 결과
- (b) 제안하는 방법(LK Optical flow)으로 움직임 벡터를 구하여 안정화를 수행한 결과



그림 9. 이전영상과 보정된 영상의 차영상

- (a) 이전영상과 현재영상의 차영상
- (b) 이전영상과 LK Optical flow 방법으로 움직임 벡터를 구해 현재 영상을 보정한 영상의 차영상

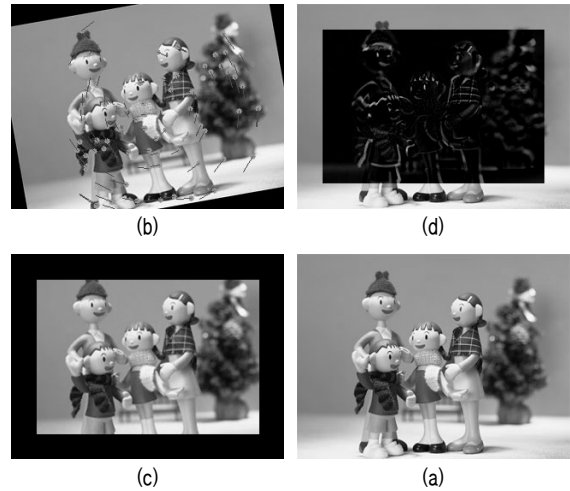


그림 10. 다른 이미지의 시뮬레이션

- (a) 원 영상 (b) 움직임 벡터를 표시한 흔들린 현재 영상
- (c) 각도 보정한 영상 (d) 원 영상과 차영상

5. 실험 결과

제안된 알고리즘의 효율성을 증명하기 위하여 본 논문에서는 스텝 모터를 이용하여 카메라를 일정한 주기로 회전시키는 실험 장비를 이용하였다. 그림 12와 같은 시스템으로 모터 End-point에 카메라가 직결되어서 모터의 회전력을 통하여 회전하게 되며 스텝 모터 1각당 1.8도이기 때문에 정확한 각도제어와 타이머를 통한 정확한 진동 주파수를 주게 된다.

사용된 그림은 그림 11과 같으며 그림이 회전하게 되



그림 11. 실험에 사용된 영상

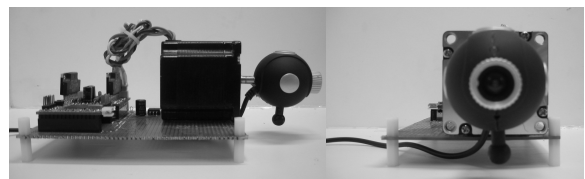


그림 12. 진동 실험 장치

면 그림의 이외 부분이 카메라로 들어오게 되는데 이를 방지하기 위하여 이미지 외곽을 검은색으로 처리 하였으며 회전시 기준 영상이 달라지는 것을 방지하기 위해서 카메라를 모터 축과 일렬로 연결하여 이미지 센서의 중심과 일치하도록 하여 테스트를 진행하였다.

그림 13 (a)은 스텝 모터를 통해서 카메라에 입력한 10Hz의 주파수, $\pm 18^\circ$ 의 진폭을 가진 입력 진동과 Lucas-Kanade Optical flow Estimation 알고리즘 상에서 움직임 보정하기 위해서 복원하는 각도를 나타내고 있다. 그림 13 (b)는 (a) 그래프에서의 오차를 표시한 그래프이며, 그림 13 (c)는 (a)의 같은 입력 진동과 Block motion algorithm 으로 보정한 각도를 보여주고 있다. 그림 13 (d)는 (c) 그래프에서의 오차를 표시한 그래프이며, 그림 13 (b)의 실험결과 그래프를 통하여 볼 때 입력 진동을 약 70% 이상 유사한 복원 각도를 얻었다. 또한 그림 13 (b)와 (d)를 비교하면 제안한 알고리즘을 사용하였을 때 오차가 대폭 감소하였음을 알 수 있다. 특히 약 9 step 마다 영상이 복원이 되는 것이 비슷하지만 그림 14에 처럼 초당 프레임 수의 차이가 많이 난다는 것을 실험을 통해 알 수 있다. fps의 차이가 많이 나면 이미지 안정화된 영상을 인식하여 정보를 획득하는데 상당한 문제를 가지고 있기 때문에 제안된 알고리즘을 사용한 시스템이 좀 더

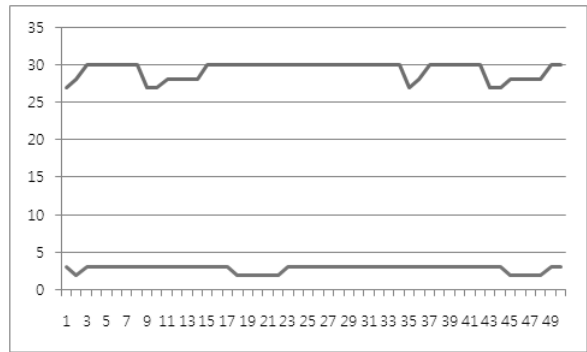


그림 14. FPS 실험 결과 그래프

윗선 - 제한한 Optical flow, 아래선 - Block motion Algorithm

정보획득에 유리함을 알 수 있다.

6. 결론

본 논문에서는 신뢰영역을 선택하기 위한 Edge Detection 과 움직임을 추정하기 위한 Lucas-Kanade Optical flow Estimation을 이용하여 디지털 영상을 안정화하는 기법을 제안하였다. 기존의 block matching 방법은 검색 윈도우의 크기에 따라 속도에 치명적인 영향이 있고, 제한적인 움직임 벡터를 추정할 수 있다. 반면, 제안한 알고리즘은

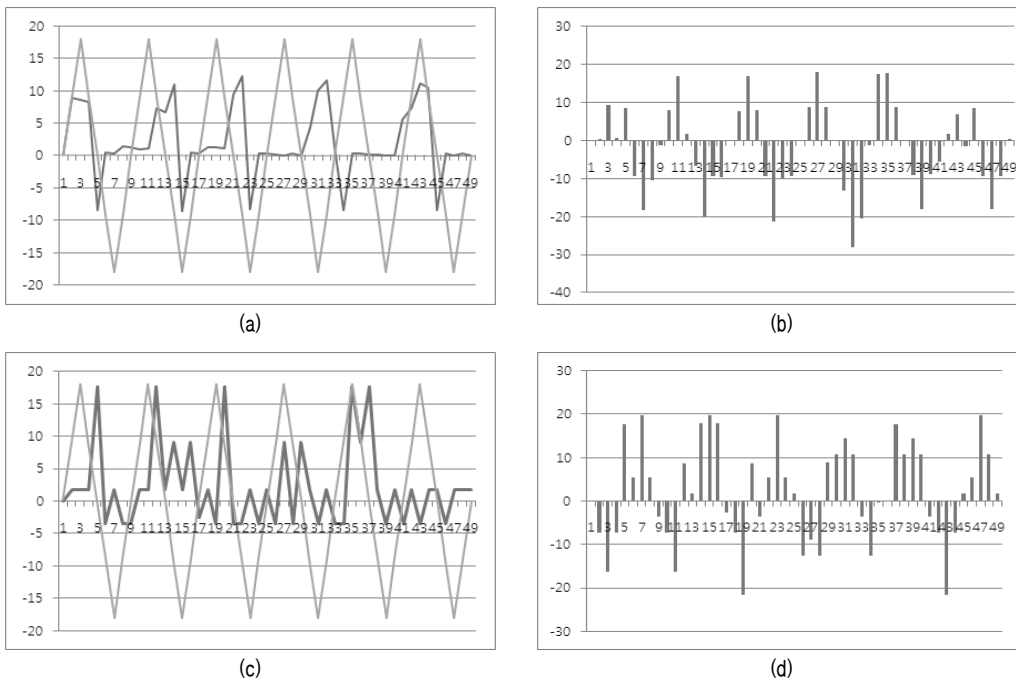


그림 13. 실험 결과 그래프

(a) optical flow에서 입력과 출력 그래프, (b) optical flow에서 입력과 출력 오차 그래프, (c) Block motion에서 입력과 출력 그래프
(d) Block motion에서 입력과 출력 오차 그래프

기존의 block matching 방법보다 전체적인 움직임 벡터가 일관성이 있으며 일반적인 optical flow 방법과는 달리 특징점의 개수를 조절하여 원하는 개수를 선택할 수 있다. 단, 영상을 보정하면 기존의 영상보다는 화질이 떨어진다는 단점이 있다.

본 논문에서는 320x240 영상에서 실시간으로 영상 안정화를 수행하였으나 향후 연구에서는 영상 사이즈가 더 큰 영상에서 이미지 안정화가 수행되어야 할 것이며, 좀 더 넓은 범위의 진동에서 이미지 안정화 실험이 수행되어 질 것이다.

참고문헌

- [1] Alan C, Brooks. "Real-Time Digital Image Stabilization," EE 420 Image Processing Computer Project Final Paper, March, 2003.
- [2] J. Kim, "Fast Digital Image Stabilization based on Edge Detection," 대한전자공학회 하계종합 학술대회 제31권 제1호, 2008.
- [3] D. Kim, "Improving Performance of Digital Image Stabilization using Adoptive motion estimation Area selection," Paper, 45SP-5-3, 2008.
- [4] K. Yang, "The ball position and Path plain for the robot-soccer using Optical flow," 대한전기학회 하계학술대회논문집, 1999.
- [5] L. Joon, "Estimation of Rotational Center and Angle for Image Stabilization," Journal of Control, Automation, and Systems Engineering Vol.10, No.7, July, 2004.
- [6] M. Chang, "Information extraction of the moving objects based on edge detection and optical flow," 한국통신학회논문지, '02-8 Vol.27 No.8A, August, 2002.
- [7] Lidong Xu, "Digital Image Stabilization Based on Circular Block Matching," IEEE Transactions on Consumer Electronics, Vol.52, No.2, pp.566-574, May, 2006.
- [8] S. Zhu, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE IEEE Transactions on Image Processing, Vol.9, No.2, pp.287-290, February, 2000.
- [9] J. Chang, "Digital Image Translational and Rotational Motion Stabilization Using Optical Flow Technique," IEEE Transactions on Consumer Electronics, Vol.48, No.1, pp.108-115, February, 2002.
- [10] A. Sibiryakov, "Estimating Inter-Frame Parametric Dominant Motion at 1000 fps Rate", Visual Information Laboratory, Mitsubishi Electric ITE-BV, Guildford, UK.
- [11] J. Jin "A Stable Vision System for Moving Vehicles", IEEE Transactions on intelligent transportation systems, Vol.1, No.1, pp.32-39, March, 2000.
- [12] S. Hwang, "IT EXPERT 영상처리 프로그래밍 by Visual C++," ("IT EXPERT Image Processing Programming by Visual C++,"), 한빛 미디어(주), 2007.
- [13] Optical Flow using OpenCV <http://altruisticrobot.tistory.com/221>
- [14] OpenCV OpticalFlow <http://deadwi.jaram.org/wiki/wikka.php?wakka=OpenCVOpticalFlow>
- [15] Lucas-Kanade Optical Flow Method http://en.wikipedia.org/wiki/Lucas_Kanade_Method



이혜정

2010 영남대학교 전자공학과
(공학사)
2010~현재 삼성전자
관심분야: 영상처리



최윤원

2009 영남대학교 전기공학과
(공학사)
2010~현재 영남대학교 전기공
학과 석사과정
관심분야: Robot Vision, Omni-
Directional Image
Processing



강 태 훈

- 2000 성균관대학교 기계설계학과(공학사)
- 2002 성균관대학교 기계설계학과(공학석사)
- 2006 성균관대학교 기계설계학과(공학박사)
- 2006~현재 (재)포항지능로봇연구소 연구팀장

관심분야 : Biomimetics, Quadruped Walking Robot



이 석 규

- 1979 서울대학교 전기공학과(공학사)
- 1981 서울대학교 전기공학과(공학석사)
- 1990 UCLA 제어공학과(공학박사)
- 1982~현재 영남대학교 전기공학과 교수

관심분야 : 제어공학, 로봇공학