

동영상의 다중 참조 빗줄기 제거 기법

김진환, 김창수
고려대학교

{jinhwankim, cskim}@mcl.korea.ac.kr

Multi Hypothesis Rain Streak Removal for Video Sequences

Jin-Hwan Kim, Chang-Su Kim
Korea University

요 약

본 논문은 비가 오는 장면을 촬영한 동영상에서 빗줄기를 효과적으로 제거하는 기법을 제안한다. 제안하는 기법에서는 광흐름 검출 기법을 이용하여 인접한 프레임에서 현재 프레임의 픽셀에 대응하는 픽셀을 검출하고, 확률에 따라 해당 프레임의 픽셀 적용 유무를 결정한다. 빗줄기로 검출된 픽셀을 인접한 프레임의 픽셀 값으로 대체함으로써 영상 내 빗줄기를 제거한다. 컴퓨터 모의실험을 통해 제안하는 알고리즘이 동영상에서 효과적으로 빗줄기를 제거할 수 있음을 확인한다.

1. 서론

컴퓨터 비전 알고리즘의 성능은 영상 품질에 의존적이다. 특히 차선 검출, 객체 추적 및 감시 기법 등의 외부에서 촬영된 영상에 기반한 비전 알고리즘은 기상상태에 따라 성능에 영향을 받는다. 예를 들어 비가 오거나 안개가 낀 영상에서는 객체 추적 또는 차선 검출의 성능이 저하된다. 특히 비가 내리는 영상에서는 빗줄기의 구조 때문에 영상 품질이 저하되고 엣지 검출 또는 색상 정보 검출을 사용하는 컴퓨터 비전 알고리즘의 성능이 저하될 수 있다. 따라서 빗줄기를 제거할 경우 비전 알고리즘의 성능을 크게 높일 수 있기 때문에 최근 빗줄기 제거 기법에 대한 연구가 활발히 진행되고 있다.

빗줄기 제거 기법은 동영상의 인접 프레임간 차이를 이용하는 방식이 대표적이다. 빗줄기의 경우 매 프레임마다 이동거리가 클 뿐만 아니라 빗줄기에 영향을 받은 픽셀이 밝아지기 때문에, 밝기 및 이동방향에 따른 제한 조건을 이용하여 빗줄기를 제거하는 기법이 제안되었다[1]. 또한 Barnum 등은 주파수 도메인에서 나타나는 빗줄기의 특징을 이용하여 빗줄기를 효과적으로 검출하는 기법을 제안하였다. [2]. 위와 같은 소프트웨어 방식의 접근 뿐만 아니라 카메라로 촬영 시 빗줄기가 촬영되지 않도록 셔터 속도 및 조리개 값을 조절하는 하드웨어의 조절을 통한 방법도 제안되었다 [3]. 하지만 기존 기법의 경우 알고리즘의 복잡도가 매우 높거나 빗줄기를 효과적으로 제거하지 못하여 영상에 빗줄기가 남는 등의 문제가 발생한다.

본 논문에서는 다중 프레임 참조를 이용하여 동영상 내의

빗줄기를 제거함으로써 빗방울이 없는 깨끗한 영상을 복원하는 알고리즘을 제안한다. 또한 제안하는 알고리즘은 기존 기법에 비해 향상된 성능을 제공할 뿐만 아니라 낮은 복잡도를 가진다. 컴퓨터 모의 실험을 통하여 제안하는 알고리즘이 효과적으로 빗줄기를 제거함을 확인한다.

2. 제안하는 알고리즘

본 논문에서는 우선 빗줄기가 가지는 특성을 분석한 후 빗줄기의 특성을 이용하여 빗줄기를 제거한다.

영상에서 빗줄기는 프레임이 변화함에 따라 선형적으로 움직이고, 영상의 배경을 밝게 변화시킨다. 따라서 움직임이 없는 영상의 경우 연속된 프레임의 동일 위치에서의 픽셀 값은 빗줄기에 영향을 받았을 경우에 밝아진다 [1]. 또한 빗줄기의 영향을 받지 않았을 경우 프레임 f 에서의 픽셀 값 $I^f(p)$ 는 인접 프레임의 픽셀 값 $I^{f-1}(p)$ 및 $I^{f+1}(p)$ 과 동일하다.

$$I^f(p) - I^{f-1}(p) = I^f(p) - I^{f+1}(p) > c, \quad (1)$$

하지만 실제 영상의 경우 움직임이 계속해서 발생하므로 (1)의 조건을 이용하여 빗줄기를 검출하면 객체의 움직임이나 영상의 변화에 따른 오검출의 발생이 증가하고, 이는 빗줄기를 제거한 결과영상의 품질 저하를 야기한다. 기존 기법[1]의 경우 이를 제거하기 위해 다양한 제한조건을 사용하였으나 오검출을 충분히 제거하지 못했다.

동영상의 움직임으로 인한 오검출을 보완하기 위해 본 논문에서는 움직임 벡터를 이용하여 빗줄기를 검출한다. 즉 (1)의 식을 이용하여 빗줄기를 검출하되 동일한 픽셀 위치가 아닌 움직임 벡터만큼 보상된 픽셀 위치로부터 빗줄기를 검출한다. 빗줄기는 크기가 작고, 매 프레임마다 다른 픽셀 위치로 이동할 뿐만 아니라 이동하는 거리 또한 매우 멀기

때문에 블록 매칭 등을 이용하여 움직임 벡터를 검출할 때 영향을 미치지 못한다. 즉 블록 매칭 등의 움직임 검출 시 객체나 배경이 가지는 기하 구조 차이에 비해 빗줄기의 유무로 발생하는 차이가 매우 작기 때문에, 움직임 검출에 영향을 미치지 않는다.

하지만 빗줄기 구조성분은 변화가 없는 평면 영역에서는 큰 값으로 검출 될 수 있기 때문에, 블록 매칭 기법만을 이용하여 움직임 벡터를 검출할 경우 빗줄기의 이동에 따라 움직임 벡터가 결정되는 문제가 발생 수 있다. 따라서 본 논문에서는 해당 픽셀의 움직임 벡터뿐만 아니라 주위 움직임 벡터 값을 고려하는 광흐름(optical flow) 검출 기법을 이용한다. 광흐름 검출 기법은 아래의 에너지 함수를 최소화 하는 움직임 벡터를 추출하는 기법으로 본 논문에서는 [4]의 광흐름 검출 기법을 사용한다.

$$E(\mathbf{u}) = \sum_p \psi \left(\left(I^{f-1}(\mathbf{p} + \mathbf{u}_p) - I^f(\mathbf{p}) \right)^2 \right) + \lambda \psi \left(\|\nabla \mathbf{u}_p\|^2 \right), \quad (2)$$

여기에서 \mathbf{u} 는 움직임 벡터를 나타내고 ψ 는 페널티 함수를 의미한다. (2)에서 $\psi \left(\|\nabla \mathbf{u}_p\|^2 \right)$ 는 평활도(smoothness) 향으로 픽셀 값의 변화가 거의 없는 영역에서 주위 움직임 벡터를 활용할 수 있도록 제한하는 항이다. (2)를 이용하여 계산된 움직임 벡터를 이용하여 (1)을 수정하면 아래와 같이 움직임이 있을 때에도 빗줄기를 검출 할 수 있는 조건을 만들 수 있다.

$$I^f(p) - I^{f-1}(p + \mathbf{u}_p^{f-1}) = I^f(p) - I^{f+1}(p + \mathbf{u}_p^{f+1}) > c. \quad (3)$$

(3)을 이용하여 검출된 빗줄기 픽셀은 인접한 프레임의 일치하는 픽셀 값으로 대체함으로써 빗줄기를 제거할 수 있다. 본 논문에서는 인접한 프레임의 픽셀 중 가장 지배적인 픽셀 값을 선택하였고 이를 위해 실험에서는 중간 값 필터를 통해 값을 획득하였다. 이 때 각 프레임에서 잘못된 움직임 벡터 검출로 인한 오류가 발생할 수 있기 때문에, 확률 정보를 이용하여 프레임의 픽셀 값 이용 여부를 결정한다. 우선, 픽셀 주위의 블록을 결정하고 각 프레임에서 현재 프레임과의 블록간 SAD 값 비교를 통한 확률을 추정한다. 픽셀 p 에 대응하는 $f+k \in \{-2, -1, +1, +2 \dots\}$ 프레임에서의 픽셀 $p + \mathbf{u}^{f+k}$ 의 확률이 낮을 경우 빗줄기를 제거할 때 해당 프레임 정보를 사용하지 않음으로써 오검출로 인한 결과영상의 품질 저하를 보완할 수 있다.

3. 실험 결과

본 논문에서는 인접 프레임 정보를 사용할 때 $f+k$, $k \in \{-2, -1, +1, +2\}$ 의 총 4 개의 프레임을 사용하여 빗줄기를 제거한다. 더 많은 프레임을 사용할 경우 현재 프레임과의 움직임 벡터 추출 시 오류가 발생할 가능성이 높아지므로 실제 빗줄기 제거 시 해당 프레임이 계산에 포함되지 않기 때문에, 실제 결과에는 거의 영향을 미치지 않는다. 따라서 본 논문에서는 인접한 4개의 프레임으로 제한한다.

본 논문에서는 제안하는 알고리즘의 성능을 기존 기법 중 가장 높은 성능을 보이는 기법[3] 과 비교하여 그림 1 에 도시한다. 그림 1.(a)의 n 번째 입력 프레임과 비교하여 그림 1.(b)의 기존기법[3]의 결과는 대부분의 빗줄기를 제거하였으나 빗줄기의 크기가 클 경우 빗줄기가 여전히 존재하는 것을 확인할 수 있다. 하지만 그림 1.(c)에서 도시하는 바와 같이 제안하는 알고리즘은 기존 기법에 비해

효과적으로 빗 줄기를 제거함을 확인할 수 있다.

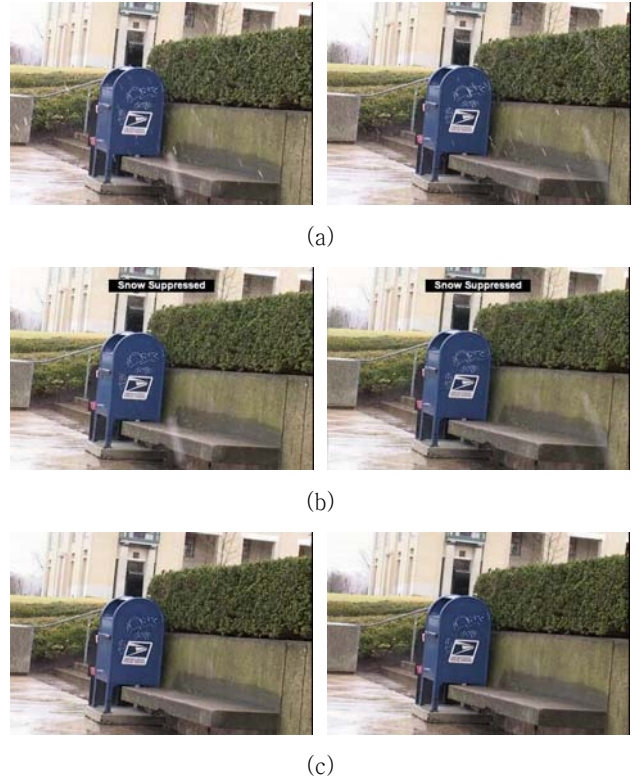


그림 1. 빗줄기 제거 결과. 좌측 영상-17 번째 프레임, 우측 영상-25 번째 프레임. (a) 원본 영상, (b) 기존 기법 [3], (c) 제안하는 기법.

4. 결론

본 논문에서는 동영상에서 발생하는 빗줄기를 제거하는 기법을 제안하였다. 제안하는 방법에서는 현재 프레임에 인접한 다중 프레임으로부터 동일한 구조를 가지는 픽셀 위치를 추정한 후 오류가 최소가 되며 동시에 가장 지배적인 픽셀 값을 선택하는 방법으로 빗줄기를 제거하였다. 실험을 통하여 제안하는 알고리즘이 효과적으로 빗줄기를 제거하는 것을 확인하였다.

5. 참고 문헌

[1] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proc. IEEE CVPR*, June 2004, pp. 528-535.
 [2] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *Int. J. Comput., Vis.*, vol. 86, no. 2/3, pp. 256-274, Jan. 2010.
 [3] K. Garg and S. K. Nayar, "When does a camera see rain?," in *Proc. IEEE ICCV*, Oct. 2005, pp. 1067-1074.
 [4] C. Liu. Beyond Pixels: Exploring New Representations and Applications for Motion Analysis. PhD thesis.