

영상 시퀀스의 프레임 지터 고속 정합 알고리즘

이임건 · 우영운 · 한수환

동의대학교

Fast Frame Jitter Matching for Image Sequence

Imgeun Lee · Young Woon Woo · Soowhan Han

Donggeui University

E-mail : iglee@deu.ac.kr

요 약

본 논문에서는 영상 시퀀스 내의 흔들림을 제거하는 고속 알고리즘을 제안한다. 프레임 지터 제거에 관한 연구는 영상 취득과정에서 발생하는 손떨림에 의한 영향을 제거하거나 오래된 영화필름에서 녹화장치의 기구적인 문제로 인해 발생한 흔들림을 제거하기 위해 많이 연구되었다. 제안하는 알고리즘은 영상 프레임을 배경과 전경 영역으로 나누고 움직임 벡터를 이용하여 전역 움직임을 구한다. 전역 움직임을 구하는 과정에서 잘못된 움직임 예측이 발생할 가능성이 높은 블록을 사전에 제거하여 속도를 개선하였으며 알고리즘을 실제 영상에 적용하여 영상 프레임 정합됨을 보였다.

키워드

지터제거, 프레임 안정화, 움직임 예측, 영상정합

1. 서 론

영상복원은 영상 시퀀스에 대해 동일한 2차원 영상처리 알고리즘을 적용하여 동영상의 주관적 화질을 개선하는 것을 목표로 한다. 동영상인 경우 시간축상에 많은 잉여 정보가 포함되어 있으므로 2차원 처리를 단독으로 사용하는 것보다 시간정보를 포함하는 3차원 처리를 행하는 것이 유리하다. 그러나 시공간의 정보를 모두 고려하는 알고리즘은 물체의 움직임, 장면 전환, 패닝, 주밍 등의 비선형적 요소가 존재하므로 알고리즘의 설계가 복잡한 것이 일반적이다. 위성전송 사진의 복원과 같은 움직임 정보를 사용하지 않고 정지영역을 대상으로 하는 복원 알고리즘인 경우 동영상에서 특히 눈에 두드러진 블로치(blotch) 제거 등이 힘들고 특히 움직이는 물체가 많은 경우 그대로 적용하는 것은 불가능하다. 그러므로 동영상의 복원에 있어 물체의 움직임을 발견하는 것이 매우 중요하며 실제 대부분의 실용적인 알고리즘에서 움직임의 추출을 복원부와 분리하여 전처리 단계에서 수행한다[1][2].

훼손영상에 대해 3차원적인 처리를 하기 위해서는 시간축 상으로 각 프레임의 정합되어 있어야 하지만 영상 취득 과정에서 카메라의 흔들림이나 기구적인 한계로 인해 프레임간의 부정합이 발생

하는 경우가 많다. 이러한 프레임 지터(jitter)는 영상복원을 위해 전후 프레임에서 얻을 수 있는 정보의 정확성을 떨어뜨리게 되므로 정확한 후반 처리를 위해서는 각 프레임의 정합이 필수적이다.

최근 영상통신이 가능한 3세대 이동전화기 보급되기 시작하였으며 이동 중에 화상통화를 시도하는 경우 카메라의 흔들림에 의한 프레임 지터는 더욱 심해진다. 특히 실시간 처리가 필요한 모바일 화상통신은 하드웨어 및 소프트웨어적인 자원의 제약으로 고속 알고리즘의 개발이 필요하다.

본 논문은 영상 프레임의 정합을 위해 지터를 제거하는 고속 알고리즘을 제안한다. 제안하는 알고리즘은 영상 영역을 분할하여 통계적으로 전역 움직임이 발생할 확률이 높은 영역의 움직임 벡터만을 계산하여 다수결의 방법으로 전역 움직임을 결정한다. 또한 각 영상 블록에서의 움직임 벡터들의 신뢰성을 판단하여 전역 움직임 벡터의 정확성을 높였다.

II. 프레임 시퀀스 모델과 움직임 예측

영상 시퀀스에 대한 정확한 수학적 모델은 없으나 본 연구에서는 영상 복원을 위해 영상 시퀀스의 모델을 선형이동모델로 가정하였다. 선형이동모델은 현재 영상의 오브젝트가 이전 프레임의

한 부분으로부터 이동되어 나타난 것으로 간주한다. 그러므로 현재 프레임의 화소는 이전 프레임의 화소로부터 예측가능하게 된다. 이러한 모델을 수학적으로 표현하면 다음 식으로 주어진다[3].

$$I_n(\bar{x}) = I_{n-1}(\bar{x} + \bar{d}_{n,n-1}) \quad (1)$$

여기서 $I_n(\bar{x})$ 은 현재 프레임의 \bar{x} 에 위치하는 화소값을 나타내고 $\bar{d}_{n,n-1}$ 은 n 프레임과 $n-1$ 프레임간의 변위를 나타낸다. 이 때 화면간의 움직임 정보는 $\bar{d}_{n,n-1}$ 로 주어지고 이 벡터를 찾아내는 것이 움직임 추출 알고리즘이다.

식 (1)을 움직임을 나타내는 함수를 이용하여 영상관점의 접근방식으로 표현한 수식을 식 (2)에 나타내었다. 여기서 $F(\bar{x})$ 는 영상좌표에 대한 선형변환을 나타내며 확대/축소, 회전, 이동 등의 움직임을 표현한다.

$$I_n(\bar{x}) = I_{n-1}[F(\bar{x})] \quad (2)$$

영상 복원에 있어 정확한 움직임을 예측해 내는 것은 매우 중요하다. 일반적으로 많이 사용되는 움직임 예측 방법은 블록매칭(block matching)을 이용하는 것이다. 블록매칭 알고리즘은 간단하고 효과적이지만 오래된 필름같이 영상이 블러링(blurring) 되어 흐린 경우에는 블록매칭 알고리즘으로는 움직임을 잘 찾아내지 못하는 경우가 많다. 영상이 흐린 경우는 밝기 값을 기반으로 하는 움직임 예측 알고리즘보다 위상상관(phase correlation)을 이용하여 영상의 에지를 기반으로 하는 움직임 예측 알고리즘이 유용하다. 주파수 영역에서 움직임이 있는 블록의 전·후 프레임 간에는 식 (3)이 성립한다.

$$S_n(f_1, f_2) = S_{n-1}(f_1, f_2) \exp[j2\pi(f_1\Delta x + f_2\Delta y)] \quad (3)$$

두 프레임에서 블록간의 위상상관은 공간영역과 주파수 영역에서 각각 식 (4), (5)로 표현되고 식 (5)을 정규화 시키면 식 (6)을 얻는다. 움직임 벡터는 식 (6)을 역 푸리에 변환시켜 얻는다[4].

$$C_{n,n-1}(f_1, f_2) = I_n(x, y) * I_{n-1}(-x, -y) \quad (4)$$

$$C_{n,n-1}(f_1, f_2) = S_n(f_1, f_2) S_{n-1}^*(f_1, f_2) \quad (5)$$

$$\begin{aligned} \Phi[C_{n,n-1}(f_1, f_2)] & \\ &= \frac{S_n(f_1, f_2) S_{n-1}^*(f_1, f_2)}{|S_n(f_1, f_2) S_{n-1}^*(f_1, f_2)|} \\ &= \exp[-j2\pi(f_1\Delta x + f_2\Delta y)] \end{aligned} \quad (6)$$

움직임 벡터는 식 (6)의 값이 최대값이 나오는

경우의 변위로 선택하며 블록의 경계효과에 의한 오류를 줄이기 위해 윈도우 함수를 취한다.

III. 프레임 지터

화면 지터는 단일 프레임에서는 감지되지 않으나 시퀀스를 재생하는 경우 화면의 떨림으로 표현된다. 또한 영상 시퀀스에 있는 지터는 움직임 예측의 오차를 유발시켜 후반부의 영상복원 작업을 어렵게 만들므로 전체 영상 복원 과정의 전체 리 작업으로 우선 해결해야 한다.

현재 영상을 f_n , 이전 영상을 f_{n-1} 이라 할 때 지터에 대한 수학적 모델은 식 (7)과 같고 이때의 보정 벡터 \bar{n} 은 식 (8)이 된다.

$$f_n(x, y) = f_{n-1}(x + \Delta x, y + \Delta y) \quad (7)$$

$$\bar{n} = (\Delta x, \Delta y) \quad (8)$$

그러므로 지터의 보정은 현재 프레임을 보정 벡터만큼을 역 평행 이동하여 수행된다.

위상차를 이용하거나 단순히 블록매칭을 이용한 움직임 추정 알고리즘에서 화면의 지터가 발생한 전후 프레임의 움직임 벡터는 전체적으로 같은 방향의 벡터 성분이 포함되어 있으나 화면 내 물체의 움직임이 존재하는 경우 오류가 있을 수 있다. 즉, 화면전체에 대한 움직임 벡터를 찾는 것은 화면 내에 움직임이 많거나 비교적 큰 물체가 움직이는 경우 영상의 지터를 정확히 계산하지 못하는 경우가 발생한다. 본 연구에서는 이를 해결하기 위해 영상을 배경영역과 전경영역으로 분리하였다. 영상에서의 배경/전경 영역은 다음으로 정의된다.

- 배경영역(Background region)
물체의 움직임은 없고 움직임 벡터는 화면 전체의 떨림에 의한 영향만을 받는 영역
- 전경영역(Foreground region)
화면내의 주제가 되는 물체 혹은 인물이 움직이는 영역으로 움직임 벡터는 물체의 움직임에 의한 것과 지터에 의한 영향을 동시에 받는 영역

대부분의 화면 구성에서 주가 되는 물체는 화면 중앙부에 위치하고 움직이는 물체 역시 화면의 중심부에 위치하고 있으므로 움직임 예측은 전경영역에 대해서는 수행하지 않는다. 또한 영상 시퀀스의 지터는 영상의 가장자리에서 눈에 잘 드러나므로 가장자리를 포함하는 배경영역에 대해서만 움직임 예측을 수행한다. 이러한 과정을 통해 연산량을 더욱 줄일 수 있다. 그림 1에 배경과 전경영역을 보였다.

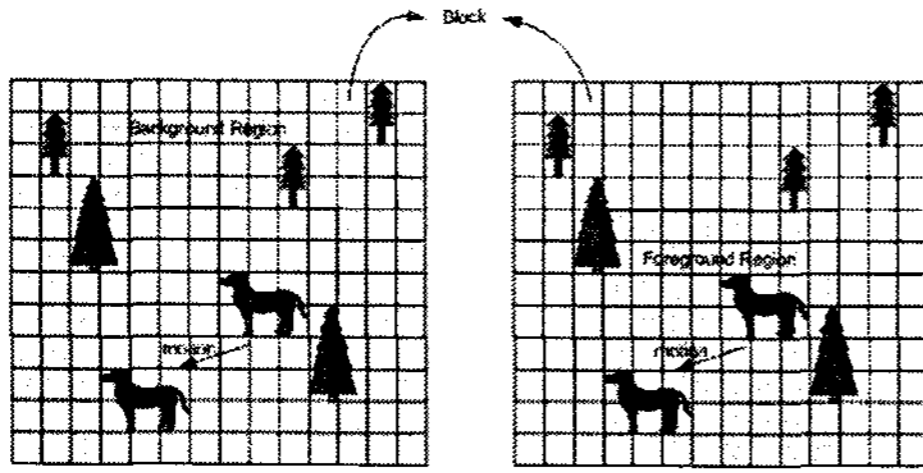


그림 1. 배경영역과 전경영역

위상차를 이용하거나 단순 블록매칭 알고리즘을 이용한 움직임 벡터의 추출은 부영상 블록내의 에지들의 정보를 이용한다. 부영상의 구성이 평이한 계조도 분포를 가지면 정확한 움직임 벡터를 찾지 못해 오류 움직임 벡터들이 많이 생겨나게 된다. 본 연구에서는 이러한 문제점을 해결하기 위해 움직임 벡터를 찾기 위한 부영상 블록에서 주파수 영역상의 고주파 성분이 많은 블록에 대해서만 위상차를 이용한 움직임 추정을 수행하였다.

고주파 성분이 많은 부영상의 판단은 위상차 움직임 추출 알고리즘의 처리 과정에서 위상상관성을 구하기 위해 부영상의 푸리에 변환을 사용하므로 별도의 계산 없이 주파수 영역에서의 고주파 성분의 에너지를 측정함으로써 이루어진다. 본 연구에서는 화면의 복잡도를 계산하기 위해 블록 영상의 에너지에서 중간 주파수 대역의 크기를 구하였으며 DC를 포함한 저주파 대역의 에너지는 고려하지 않는다. 또한 영상의 가장자리와 노이즈에 의해 고주파 성분이 발생할 수 있으므로 고주파 성분의 에너지도 제거한다. 대역통과 필터를 취한 뒤의 에너지가 임계치 T를 넘는 경우는 화면이 움직임 예측을 수행할 수 있을 정도의 복잡성을 가진 것으로 가정한다.

그림 2에 제안한 알고리즘을 이용하여 실제 영상에서 움직임 예측을 수행한 결과를 보였다. 그림에서 'x' 표시는 움직임이 벡터의 값이 0인 것으로 움직임이 없는 블록을 나타내며 '□' 표시는 화면의 복잡도가 높지 않아 움직임 예측을 수행하지 않은 블록을 나타낸다. 실선 표시는 움직임 벡터의 방향과 크기를 표현한다. 좌상단의 문틀 부근은 화면의 복잡도는 움직임 예측을 수행할 만큼 복잡하나 블록내의 패턴의 인접 블록과 유사하여 움직임 예측 오류가 발생한 영역이다.

전체적으로 화면의 지터를 판단하기 위해서는 다수결의 원칙을 이용하며 화면에서 발생한 움직임 벡터의 성분을 발생 빈도로 정렬한 뒤 가장 많은 발생빈도를 갖는 벡터 성분을 지터에 의한 움직임으로 결정한다. 제안하는 알고리즘은 오류 발생확률이 높은 영상 블록에 대해서는 움직임 예측을 수행하지 않아 발생 빈도수에 대한 기여도를 제거함으로써 전역 흔들림에 의한 화면 지터의 정확성을 높이고 계산량을 줄일 수 있다.



그림 2. 제안한 방법으로 추출된 움직임 벡터

IV. 실험 및 고찰

제안하는 알고리즘을 검증하기 위해 정합되지 않은 프레임 시퀀스와 알고리즘을 적용해서 정합된 시퀀스에 대해 블로치(blotch) 제거 실험을 행하였다. 블로치는 필름 베이스의 유제인 젤라틴이 벗겨져 떨어져 나가 발생하며 화면상에는 노이즈와는 달리 얼룩 혹은 백색 계통의 점이 명멸하는 것으로 나타난다.

본 실험에서 블로치 검출기는 SDIa(spike detection index-a)를 이용하였다[5]. SDIa 검출기는 영상 계조도에서 시간축상의 불연속성을 검출하는 휴리스틱한 방법이며 식 (9)로 표현된다. SDIa의 값은 현재 영상의 화소 계조도와 움직임 예측을 통해 얻어진 전후 프레임의 계조도와의 차이에서 적은 값을 취한 것이다. 얻어진 SDIa의 값이 클수록 훼손이 발생했을 가능성이 높아진다. 그러므로 블로치 마스크는 식 (10)과 같이 SDIa의 값에 대한 임계치를 적용한 이진 영상으로 정의된다. 그림 3에 이를 보였다. 여기서 $Z(\bar{i})$ 는 훼손된 영상이며 $d(\bar{i})$ 는 블로치 검출 마스크를 의미한다.

$$SDIa(\bar{i}) = \min [(Z(\bar{i}) - Z_{mc}(\bar{i}, t+1))^2, (Z(\bar{i}) - Z_{mc}(\bar{i}, t-1))^2] \quad (9)$$

$$d_{SDIa}(\bar{i}) = \begin{cases} 1 & \text{if } SDIa(\bar{i}) > T_1 \text{ with } T_1 \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

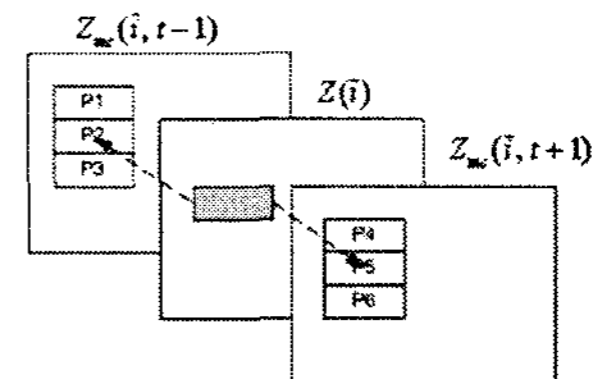


그림 3. SDIa 검출기의 참조 프레임

제안한 알고리즘을 실제 영화 시퀀스에 적용한 결과 영상을 아래에 나타내었다. 그림 4는 영화 시퀀스의 10번째 프레임의 원영상을 나타내고 있으며 9번째 프레임으로부터 지터가 발생한 영상이다. 9번째 프레임과 10번째 프레임의 지터에 의한 전역 움직임 벡터는 행과 열 방향으로 $\vec{n}=(1,2)$ 이며 전체 프레임을 역방향으로 선형 이동시켜 지터를 보정한다. 아래 영상 시퀀스에서 실제 움직임이 있는 부분은 좌 하단의 소년뿐이다.

실험에 사용한 영화 시퀀스는 오래된 필름의 텔레시네된 프레임이라 흐리고 에지가 불분명한 부분이 많이 존재한다. 에지가 불분명하지 않은 블록은 오류 움직임 벡터의 수를 증가시키게 되고 결과적으로 다수결의 원칙으로 지터에 의한 전역 움직임을 결정하는 알고리즘에서 전체적인 오류를 유발시킬 수 있다. 실험결과 제안한 블록내 화면의 복잡도를 사용한 적응적 움직임 예측 알고리즘으로 오류가 발생할 가능성이 높은 블록의 발생을 억제할 수 있었다.



그림 4. 10번째 원영상

그림 5는 지터를 보정하지 않고 블로치를 제거하기 위한 마스크를 얻은 것이다. 검게 표시된 부분이 블로치로 인식되어 앞뒤의 프레임 정보로부터 복원할 부분이다. 실제 필름의 이물질로 인한 블로치는 프레임 중앙부에 약 15개 화소정도이지만 이지만 지터로 인해 원치 않는 부분에서도 블로치 마스크가 생성됨을 알 수 있다.

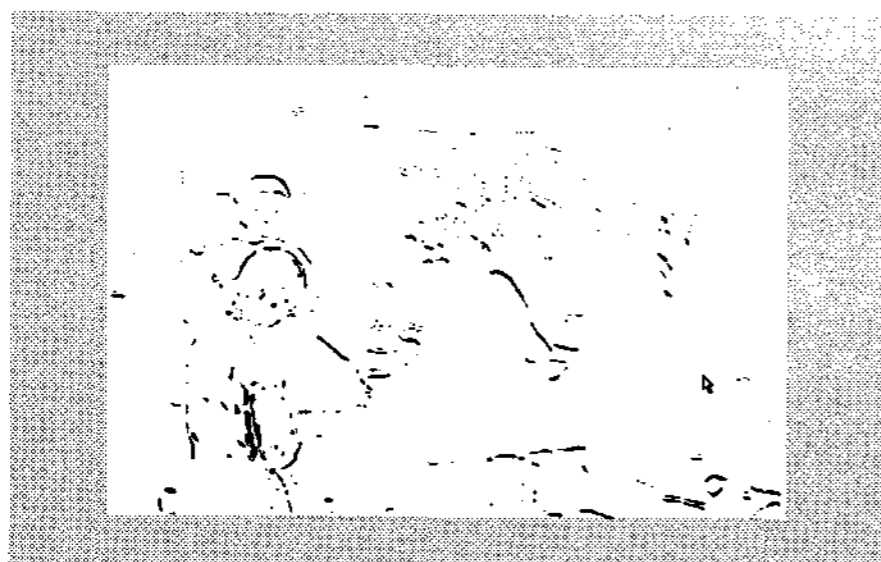


그림 5. 지터 보정하지 않은 시퀀스에서의 블로치 마스크

그림 6은 지터를 보정한 후에 블로치 마스크를 구한 것이다. 프레임 중앙부의 검은 점이 실제 훼손되어 복원되어야 할 부분이며 정확하게 블로치 마스크로 인식됨을 알 수 있다. 소년의 주변부가 마스크로 인식되는 것은 움직임 예측 모듈의 에러 때문이며 좀 더 정교한 움직임 예측 알고리즘이 필요하다.

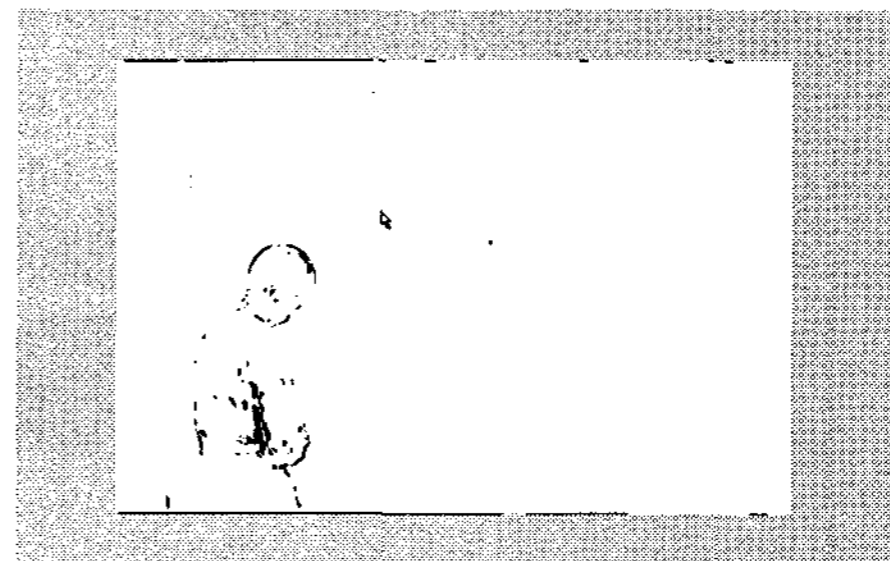


그림 6. 지터 보정 후의 블로치 마스크

V. 결론

본 논문에서는 영상의 흔들림을 제거하는 알고리즘을 제안하였다. 제안한 알고리즘은 프레임을 전경과 배경영역으로 나누고 배경영역에 대해서만 움직임 벡터를 구하여 움직임 벡터의 오류 발생 확률을 줄였다. 움직임 벡터는 위상차 검출방식을 이용하여 블록의 복잡도를 판단하고 복잡도가 단순한 블록에 대해서는 움직임 벡터를 구하는 후보에서 제외하여 정확성을 높이고 계산량을 줄였다.

참고문헌

- [1] R. D. Morris, W. J. Fitzgerald, and A. C. Kokaram, "A sampling Based Approach to Line Scratch Removal from motion Picture Frames", *Proc. of ICIP-96, vol 1, pp.801-804, IEEE, 1996.*
- [2] R. H. Kallenberger and G. D. Cvjetnicar. *Film into Video*. Focal Press, 1994.
- [3] P.M.B van Roosmalen, R.L.Lgendijk, and J.Biemond, "Correction of Intensity Flicker in Old Film Sequences", *Proc. of IEEE Signal Processing Symposium SPS'98, pp.2253-2256, 1998.*
- [4] S. Efstratiadis and A. Katsagellos. "A Model based pel-recursive motion estimation algorithm", *Proceedings IEEE ICASSP, pp.1973-1976, 1990.*
- [5] A. C. Kokaram, *Motion Picture Restoration*, Springer Verlag, 1998.