다중 블록을 이용한 적응적 블록 정합 알고리즘

*조윤섭, **한윤상, ***이상근
중앙대학교
*jo258kr@naver.com, **imageofdream@mmc.cau.ac.kr, ***sangkny@cau.ac.kr

Adaptive Block Matching Algorithm Using Multi Block

*Yunsub Cho **Yunsang Han ***Sangkeun Lee
Chung-Ang University

요 약

본 논문은 기존의 고정블록 알고리즘에서 발생할 수 있는 객체 추적의 문제점을 보안한 다중 블록을 이용한 적응적 블록 정합 알고리즘을 제안한다. 기존의 고정블록을 이용한 정합 알고리즘은 추적 대상 객체의 크기와 블록의 크기가 추적 성능에 미치는 영향이 크기 때문에 객체 추적에 실패하는 경우가 발생한다. 본 논문에서는 고정된 블록 정합 알고리즘의 문제점을 해결하기 위해 다중 블록을 이용하여 객체 내의 포함된 블록들을 효과적으로 선택하고 선택된 블록들의 특성으로부터 효과적인 가중치를 부여하여 추적 성능을 향상시킨다. 제안된 알고리즘은 블록 정합 알고리즘 중 가장 정확도가 높다고 알려진 전역 탐색 방법을 이용하여 정확도를 평가 하였다.

1. 서론

최근 고성능 컴퓨터와 디지털 카메라의 보급으로 컴퓨터를 이용한 영상분석에 대해 관심이 많아 졌으며, 특히 지능형 영상감시 분야와 자동화된 영상분석 분야에 활발한 연구가 진행됨에 따라 객체 추적 분야 또한 각광받고 있다[1].

객체 추적 알고리즘은 연속된 영상에서 객체의 움직임을 검출하는 것으로, 대표적인 객체 추적 방법으로는 차영상기법(Background subtraction), 블록 정합 기법(Block matching) 등이 있다.

차영상 기법은 연속된 두 영상간의 밝기 차이를 이용한 방법으로 배경과 객체를 구별한다. 하지만 이 방법은 잡음과 조명변화에 민감하고, 객체가 정지한 경우 배경으로 인식 될 수 있다[2].

블록 정합 알고리즘은 연속된 영상에서 이전 영상의 블록정보를 이용하여 현재 영상에서의 블록위치를 추정 한다. 일반적으로 블록 정합 알고리즘은 블록 내에 모든 픽셀을 하나의 움직임 벡터로 표현 하기 때문에 객체가 블록의 크기보다 작으면 블록 내에 시시각각 변하는 배경 영역이 블록 내에 포함되어 추적에 실패 하는 경우를 볼 수 있다. 또한 블록의 크기가 객체의 크기보다 작으면 계산량이 많아지고 블록은 객체 성분을 충분히 포함하지 않게 되어 객체 추적에 실패하는 경우가 발생한다[3].

본 논문은 적응적 정합블록[4]을 이용하였으며, 이전 연구의 객체 추적 성능의 저하의 원인이 되었던, 객체블록이 아닌 후보 블록을 객체블록으로 포함시키는 문제점을 블록들의 움직임 정보를 효과적으로 이용하는 방법을 통해 추적의 성능을 향상 시켰다.

2. 제안하는 블록 정합 알고리즘

제안된 알고리즘은 기존 블록 정합 알고리즘의 문제점인 고정된 블록 크기에 의한 성능 저하를 개선시키기 위해 주 블록을 포함한 9 개의 블록을 생성한다. 생성된 9 개의 블록으로부터 움직임 벡터를 추정하고, 주/부 블록의 벡터의 사잇각과 크기의 차를 이용하여 블록들에 대한 가중치를 계산한다. 이전 영상에 대한 가중치를 고려하여 계산된 가중치를 갱신하도록 하였으며, 전체 흐름도는 다음 그림 1 과 같다.

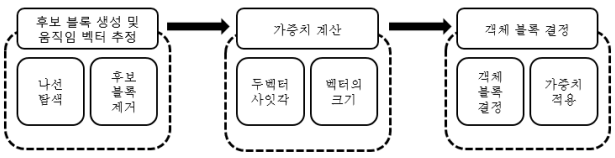


그림 1. 제안된 방법의 흐름도

먼저, 후보 블록은 그림 2 와 같이 한 개의 주 블록 및 8 개의 부 블록으로 구성된다.

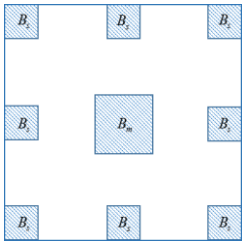


그림 2. 제안된 블록

각각의 블록들의 움직임 벡터를 빠르고 효율적으로 추정하기 위해 나선탐색방법(Spiral search)[5]과 부분왜곡제거(PDE: Partial distortion elimination)[6] 알고리즘 및 MAD(Mean absolute difference)를 이용하였다.

일반적으로 객체에 포함된 블록들은 대부분 같은 방향의 방향벡터를 가지기 때문에 주/부 블록들의 방향 벡터는 서로 비슷한 방향을 가리킨다. 이러한 정보들로부터 가중치를 수식(1)과 같이 정의 하였다.

$$\angle B_m B_s = \left(\overrightarrow{um} \cdot \overrightarrow{us}\right) \times \frac{180}{\pi}$$
$$w_{\deg} = e^{\left(-\frac{\angle B_m B_s}{\lambda_1}\right)} \quad (1)$$

여기서, $\angle B_m B_s$는 주/부 블록의 사잇각, $\overrightarrow{um}$와 $\overrightarrow{us}$는 각각 주/부 블록의 방향벡터들을 나타낸다. 또한, $w_{\deg}$는 두 벡터에 의해 계산된 가중치를 의미한다.

다음으로, 주/부 블록 벡터들의 거리를 이용한 가중치는 식(2)와 같이 정의 하였다.

$$\alpha = abs\left(\left|\overrightarrow{mv}\right| - \left|\overrightarrow{sv}\right|\right)$$
$$w_{dist} = e^{-\frac{\alpha}{\lambda_2}} \quad (2)$$

식 (2)의 $\left|\overrightarrow{mv}\right|$, $\left|\overrightarrow{sv}\right|$는 각각 주/부 블록 벡터의 크기, w_{dist} 는 두 벡터의 차에 대한 가중치를 나타낸다.

식(1), 식(2)으로부터 구해진 가중치는 식(3)과 같이 부 블록들의 가중치를 계산한다.

$$\overline{w}_{total} = \delta w_{\text{dist}} + (1-\delta) w_{\deg} \quad (3)$$

식(3)에서 계산된 가중치는 현재 프레임의 대한 가중치로 이전 정보에 대해 독립적이다. 하지만 일반적인 객체의 움직임은 각 프레임에 의존적인 특성을 갖고 있으므로 이전 프레임의 가중치를 이용하여 다음 식(4)과 같이 최종의 가중치를 계산한다.

$$\hat{w}_{total}^{f} = \gamma \hat{w}_{total}^{f-1} + (1-\gamma) \overline{w}_{total} \quad (4)$$

식(4)에서 f는 현재 프레임, $\hat{w}$는 계산된 최종 가중치, $\overline{w}$ 식(3)에서 계산된 두 벡터의 방향성과 크기의 관계를 고려한 가중치를 의미한다. 이렇게 구해진 가중치는 일정 임계값을 이용하여 객체 블록으로 포함 여부를 판단하게 되며, 구해진 객체 블록과 가중치를 고려하여 객체의 움직임을 판단하게 된다.

3. 실험

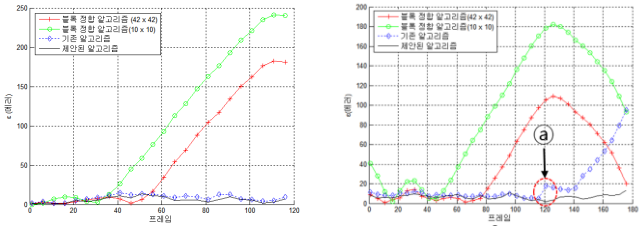
실험을 위해 일반 실내환경에서 웹 캠을 통하여 취득된 640×480 의 해상도의 영상을 이용하였고, 블록크기를 각각 10, 42 로 고정된 전역 탐색 알고리즘과 이전에 제안된 알고리즘과 성능 평가를 하였다.

λ_1, λ_2, δ 및 γ는 각각 20, 2, 0.5, 및 0.3으로 설정하였다.

제안하는 알고리즘의 성능을 평가하기 위해 식 (5)와 같이 실제 좌표와 구해진 좌표와의 유클리디언 거리(Euclidean Distance)를 계산하였고 그림 3 과 같은 결과를 얻었다.

$$\varepsilon = \sqrt{\left(\hat{x} - \bar{x}\right)^2 + \left(\hat{y} - \bar{y}\right)^2} \quad (5)$$

여기서, $\hat{x}$, $\hat{y}$ 은 구해진 좌표를 의미 하며, $\bar{x}$, $\bar{y}$ 은 실제 좌표를 의미한다.



(a) (b)

그림 3. 결과 비교

그림 3-(a)에서와 같이 제안된 알고리즘이 비교 알고리즘 보다 뛰어난 추적성능을 보였으며, 서로 다른 크기의 블록을 이용한 전역 탐색 알고리즘 및 기존에 연구되었던[4] 알고리즘은 객체 ⓐ위치에서와 같이 추적에 실패하였다.

4. 결론

본 논문에서는 객체 추적을 위한 다중 블록 정합 알고리즘을 제안하였다. 제안된 알고리즘은 나선 탐색 방법과 부분 왜곡 제거 방법을 이용하여 움직임 벡터를 추정하였고, 추정된 움직임 벡터들의 정보를 이용하여 가중치를 계산하였다. 계산된 가중치는 이전 영상의 가중치를 고려하여 가중치를 갱신시켜 적용함으로써 객체 추적 성능을 향상시켰다. 제안된 알고리즘은 고정된 블록을 사용한 전역 탐색방법과 기존에 연구되었던 알고리즘과의 비교를 통해 객체 추적 성능 개선 결과를 확인 할 수 있었다.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Association for Computing Machinery*, Vol. 38, No. 4, pp. 1- 45, 2006.

[2] A. Barjatya, "Block matching algorithms for motion estimation." *IEEE Transactions Evolution Computation*, pp. 225-239, Aug 2004.

[3] A. Barjatya, "Block matching algorithms for motion estimation," *IEEE Transactions Evolution Computation*, pp. 225-239, Aug. 2004.

[4] 김진태, 안수홍, 오정수 "객체추적을 위한 적응적 정합블록을 이용한 블록 정합 알고리즘," *한국해양정보통신학회논문지*, Vol. 2 no.15, pp. 455-461, Feb. 2011.

[5] C. H. Cheung, and L. M. Po "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Transactions on Circuits And Systems For Video Technology*, Vol. 12, No. 12, pp. 1168- 1177, Dec. 2002.

[6] H. S. Wang and R. M. Mersereau, "Fast algorithms for the estimation of motion vectors," *IEEE Transactions on Image Processing*, pp. 435-438, Aug. 1999.