

시공간 정보를 이용한 움직이는 물체의 분할

장 재식, 김 종배, 이 창우*, 김 항준
 경북대학교 컴퓨터공학과 인공지능연구실, *포항 1대학 전산정보처리과

Moving Object Segmentation Using Spatio-Temporal Information

Jea Sig Chang, Jong Bae Kim, Chang Woo Lee*, Hang Joog Kim
 Dept.of Computer Engineering, Kyungpook National University
 Dept.of Computer Information Processing, Pohang College

요약

본 논문에서는 시공간정보를 이용하여 연속된 영상에서 움직이는 물체를 분할하는 방법을 제안한다. 제안된 방법은 차영상(difference image)을 이용한 움직임 추출단계, k -means 클러스터링 알고리즘을 이용한 영역 분할단계, 그리고 영역의 밝기값과 움직임 정보를 움직임 추정 및 분할단계로 구성되어 있다. 제안된 방법을 실험해본 결과 연속영상 내에서 다양한 움직임을 가진 물체를 효과적으로 분할 할 수 있는 결과를 얻을 수 있다.

Abstract

In this paper, we propose a segmentation method of moving object using the spatio-temporal information in image sequences. Proposed method consists of motion detection step using difference image, region segmentation step using k -means algorithm, motion estimation step and segmenting step using intensity and motion information. Experimental results show that the method is capable of segmenting variously moving objects in image sequences.

I. 서론

연속 영상에서 움직이는 물체를 분할하고 추적하는 일은 비디오 감시 시스템, 침입감시 시스템, 항공 안전 시스템 등의 컴퓨터 비전 응용프로그램에 기본적인 작업이다. 전통적으로, 대부분의 감시 시스템들은 사람의 시각에 기반을 두고 있다. 이러한 자동 시스템은 완전히 사람의 감각을 대신할 수는 없지만, 예상치 못하거나 위험한 상황을 감지하는 데는 많은 도움을 줄 수 있다.

감시 시스템을 구현하기 위해서는 효과적인 이미지의 검출 및 분할 알고리즘이 사용되어야 한다. 일반적으로 이미지 분할은 이미지를 각각의 물체들로 간주되는 의

미가 있는 영역들로 분할한다. 감시 시스템에서 중요한 요소중 하나는 연속영상에서 움직이는 물체를 추출하고 모션들을 추정하기 위한 모션 정보이다. 이 작업은 여러 개의 움직이는 물체들의 겹침, 관심있는 물체의 크기가 매우 작거나, 빈약하고 빨리 변하는 조명등의 문제로 인해 어려움이 있다. 또한 주위 환경이 매우 복잡할 수도 있다. 하지만, 이 작업들은 비교적 가격이 낮은 하드웨어 환경에서 실시간 동작을 위해 필요한 작업들이다. 이런 어려움들과 제약들을 해결 하기 위해서는 특별한 알고리즘들이 사용되어야 한다.

본 논문에서는 연속 이미지내의 시공간 정보를 이용하여 비디오 감시 시스템을 위한 움직이는 물체를 자동적으로 분할하는 방법을 제안한다. 제안한 방법은 그레이 레벨과 모션 정보에 기반을 두고 있다. 제안한 방법의 첫 번째인 연속이미지의 그레이 레벨 정보는 움직임이 있는 영역과 배경 영역들로 구성된 변화된 영역들을 추출하기 위해 사용된다. 움직임 추출을 위해서 차 영상(difference image) 분석 방법에 의해 배경 차이 영상과 시간 차이영상을 구한다 [1]. 이때 차이 영상에서 정확한 움직임이 있는 영역을 추출하기 위한 임계치는 수동적으로 결정하는 대신, 주위 환경 변화에 따라 적응적으로 임계치를 선택하는 적응적 임계치(adaptive thresholding) 방법을 사용한다 [2]. 그리고 추출된 움직임이 있는 영역들에서만 k -means 클러스터링 알고리즘

을 적용함으로써 영상을 분할한다. 여기서 계산 시간을 줄이기 위해 전체 이미지에 대해서 적용하지는 않고 오직 움직임이 추출된 영역에 대해서만 적용한다. *K*-means 클러스터링 알고리즘에 의해 클러스터된 영역들은 과분할(oversegment)된 결과를 초래한다. 이를 움직임 추출된 영역의 모션 정보를 정보로부터 동일한 모션 벡터를 가진 인접한 영역들을 병합(merge)함으로써 과분할 및 다른 모션을 가진 겹친(occlusion) 물체를 분리 할 수 있다. 그림 1에서 제안된 방법의 개요를 보여준다.

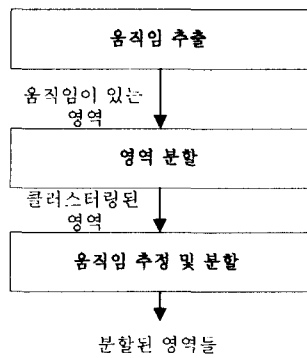


그림 1. 제안된 방법의 개요

II. 움직임 추출

움직임 추출 단계에서는 현재 프레임에서 움직이는 물체의 위치를 찾는 일을 한다. 일반적으로 움직이는 물체는 이미지 상에서 큰 밝기변화를 수반한다. 이 밝기 변화는 연속된 두 프레임간에서 뿐만 아니라 현재 프레임과 배경영상간의 차 영상(difference image)으로도 표현할 수 있다. 대부분의 시각 기반의 감시 시스템과 같이 임계값을 수동적으로 선택하는 대신, 제안된 방법에서는 변화된 영역을 추출하기 위한 임계값을 자동적으로 선택하는 적응적 임계치(adaptive thresholding)방법을 사용한다. 그림 2는 움직임 추출 단계를 보여준다.

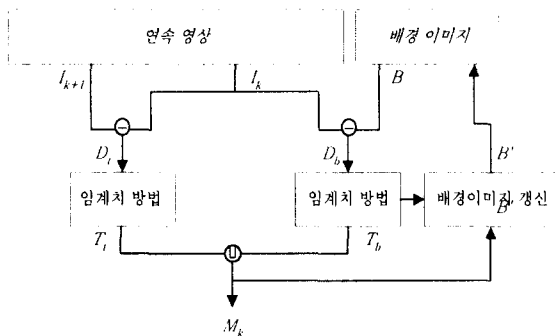


그림 2. 움직임 분할 단계

본 단계에서는 연속된 그레이 레벨 영상, $I_k(x,y)$, $I_{k+1}(x,y)$, 그리고 미리 정의된 물체가 없는 배경영상, $B(x,y)$ 을 입력으로 받아서 움직임을 가진 영역들을 출력한다. 우선 *k*번째 프레임(I_k)와 *k+1*번째 프레임(I_{k+1}) 간의 시간 차이 영상(D_t), $I_k - I_{k+1}$, 과 *k*번째 프레임(I_k)과 배경영상(B)간의 배경 차이 영상(D_b), $I_k - B$, 을 계산한다. 두 차이 영상에서 노이즈나 주위 밝기 변화에 의해 발생한 영역을 제외하고 움직임에 의해 발생한 영역을 추출하기 위해 적응적 임계치 방법을 사용하여 차영상(difference image)를 구한다.

$$T_t(x,y) = \begin{cases} 1, & \text{if } D_t(x,y) > t_t \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$T_b(x,y) = \begin{cases} 1, & \text{if } D_b(x,y) > t_b \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

차영상은 식(1),(2)에 따라서 구하고 식(3),(4)로 움직임이 있는 영역의 위치를 나타내는 모션 마스크 이진 영상(M_k)과 갱신된 배경영상(B')를 계산한다. 그림 3에서 움직임 추출단계의 결과를 보여준다.

$$M_k(x,y) = \begin{cases} 1 & \text{if } T_t(x,y) \cap T_b(x,y) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$B'(x,y) = M_k(x,y) - T_b(x,y) \quad (4)$$

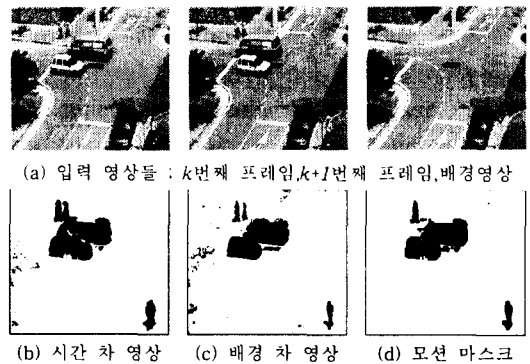


그림 3. 움직임 추출 단계의 결과

III. 영역 분할

영역 분할단계에서는 움직임을 가진 영역들을 균일한 밝기값을 가진 영역들로 분할한다. 제안된 방법에서 사용되는 영역 분할과정은 *k*-mean 클러스터링 알고리즘을 기반으로 하고있다. 움직임을 가진 영역들은 *k*개의 클러스터들로 분할한다. 영역분할을 쉽게하기 위해서

우선 5×5의 모폴로지 필터를 사용하여 영역들을 단순화시킨다 [3]. 그림 4 (b)에서 opening/closing 모폴로지 연산을 이용해서 단순화된 결과를 보여준다. 단순화된 영상을 k-means 알고리즘을 사용하여 클러스터링한다. 영역내의 각 화소들에 대해서 식 (5)의 가중치를 가진 squared 유클리디안 디스턴스가 최소화되게 클러스터링하였다 [4].

$$e'' = (p' - f^j)W^j(p' - f^j) \quad (5)$$

p^j : 현재 화소, $(x,y,I(x,y))$, $I(x,y)$: 화소의 밝기
 f^j : cluster j 의 mean vector, (m_x^j, m_y^j, m_i^j)
 W^j : cluster j 의 가중치, $(\omega_x^j, \omega_y^j, \omega_i^j)$

가중치는 다음과 같이 주어진다.

$$\omega_x^j = \frac{c^j}{\sigma_x^j}, \omega_y^j = \frac{c^j}{\sigma_y^j}, \omega_i^j = \frac{c^j}{\sigma_i^j} \quad (6)$$

$$c^j = (\sigma_x^j \sigma_y^j \sigma_i^j)^{\frac{1}{3}} \quad (7)$$

$$(\sigma_i^j) = \frac{1}{N_j - 1} \sum_{x \in c^j} (x - m_i^j)^2 \quad (8)$$

그림 4 (c)와 (d)에서 원 영상의 클러스터링 결과와 단순화된 영상의 클러스터링 결과를 보여준다. 모션 마스크 영역에 단순화를 시킨 후 영역을 분할한 결과가 단순화하지 않은 영역 보다 분할된 영역의 갯수가 현저하게 줄어들음을 볼 수 있다.

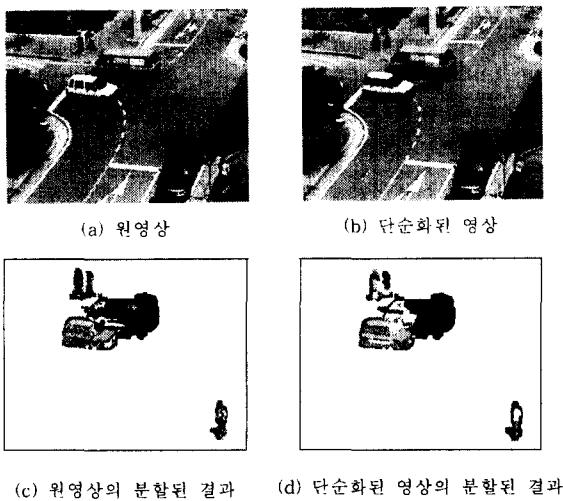


그림 4. 영역 분할 결과

IV. 움직임 추정 및 분할

본 단계에서는 움직이는 영역내 화소들의 모션 벡터를 구한다. 모션 추정은 컴퓨터 비전 영역에서 많이 연구되었다 [5]. 모션 벡터를 구하기 위해 연속된 두 프레임 I_t 와 I_{t+1} 의 두 영역의 유사성을 비교하는 matching window를 사용하였다 [6]. 즉, 영역 분할 단계에서 얻은 현재 프레임(I_k)내의 영역들이 다음 프레임(I_{k+1})의 어디에 대응되는지를 찾는다. 이때 영역의 밝기의 합이 최소가 되는 영역이 움직임에 의해 대응되는 영역이다. 아래에 주어진 오차 측정식 ME를 최소화하는 영역을 찾는다. 그림 5에서 모션 추정 결과의 결과를 보여준다.

$$ME(u_x, v_y) = \frac{1}{N_j} \sum_{(x,y) \in c^j} (I_k(x,y) - I_{k+1}(x+u_x, y+v_y)) \quad (9)$$

c^j : cluster j

$I_k(x,y)$: k번째 프레임의 (x,y)화소의 밝기

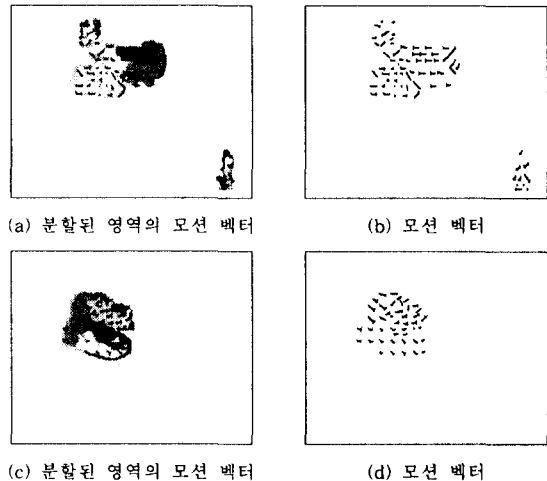


그림 5. 움직임 추정 결과

영역분할 단계에서 k-means 클러스터링 알고리즘을 사용하여 입력영상의 움직임이 있는 영역을 비슷한 밝기값을 가진 작은 영역들로 분할하였다. 그러나 K-means 클러스터링 알고리즘은 영상을 과분할하기 때문에 영역들을 다시 합쳐주는 단계가 필요하다. 본 단계에서는 비슷한 모션 벡터를 가진 영역들을 병합함으로써 과분할 문제와 더불어 겹침 문제도 해결한다. 그림 6에서 움직임 분할의 결과를 보여준다.

V. 실험결과

제안된 방법의 효율성을 검증하기 위해서 고정된 초점

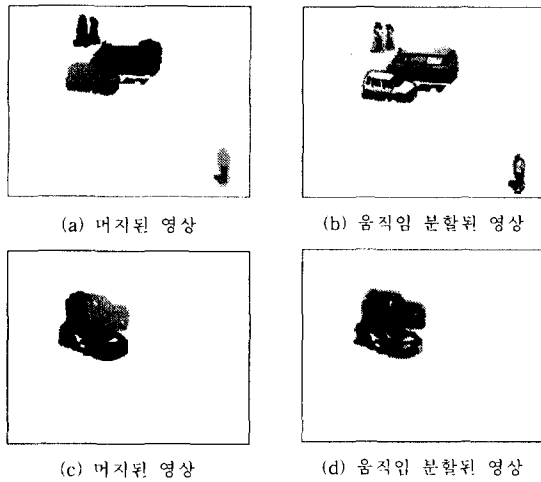


그림 6. 움직임 분할된 결과

의 복잡한 영상에 대해서 실험하였다. 초당 8 프레임으로 캡처된 320×240 영상을 사용하였다. 실험결과에서 움직이는 물체의 대략적인 위치를 알 수 있음을 보여주었다 또 다양한 조명에서 좋은 결과를 나타내었다. 움직이는 물체 분할의 정확도는 (제안한 방법으로 분할한 영역의 화소수 / 수동적으로 분할한 영역의 화소수) × 100 으로 나타내었다.

표 1. 영상 분할 결과

총 움직이는 물체 수	분할된 물체 수	정확도
131	124	93.6%

그림 7은 제안한 방법의 실험 결과를 보여준다. 그림 7(a)는 연속적으로 입력되는 영상, I_k, I_{k+1} , 과 배경 영상 (B) 이다. 그림 7(b)는 시간 차이 영상과 배경 차이 영상, 움직임을 나타내는 모션 마스크 바이너리 영상이다. 그림 7(c)는 k -means 클러스터링 방법을 이용한 영역 분할된 그림을 나타내며 각 분할된 영역의 유사한 모션 벡터에 의한 병합된 영역들을 보인다. 마지막으로 움직이는 물체가 추출된 그림을 나타낸다.

VI. 결론

본 논문에서는 연속영상에서 시공간 정보를 이용하여 움직이는 물체를 분할하는 방법을 제안하였다. 제안된 방법은 연속영상에서의 그레이 레벨과 모션정보를 기반으로 하였다. 그레이 레벨 정보는 변화하는 영역의 검출과 움직이는 영역의 분할에 사용되었고, 모션 정보는 유사한 모션벡터를 가진 영역을 병합하기위해 사용되었

다. 실험 결과 제안된 방법은 잡음과 물체의 겹침이 있는 영상에 대해서 좋은 분할 결과를 보였다.

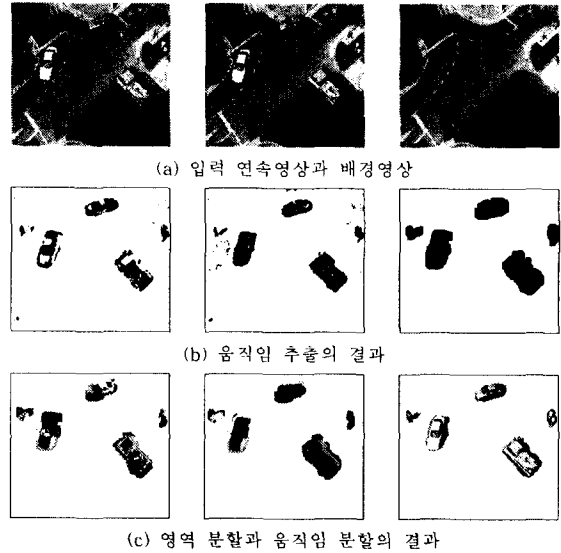


그림 7. 움직이는 물체의 분할 결과

참고 문헌

- [1] M. Betke, E. Haritaoglu and L. S. Davis, Multiple Vehicle Detection and Tracking in Hard Real-Time, *Proceedings of IEEE Intelligent Vehicles Symposium*, 1996, pp. 351-356.
- [2] H. Nariman, M. Alireza and B. Neil, Automatic Thresholding for Change Detection in Digital Video, *in Proc. SPIE*, 2000, vol. 4067, pp. 133-142.
- [3] S. Salaoda, Y. Shisikui, Y. Tanaka, and I. Yuyama, Image segmentation by integration approach using initial dependence of k -means algorithm, *in Proc. Picture Coding, Symp.97*, Berlin, Germany, Sept. 1997, pp. 265-269.
- [4] V. Markandey, A. Reid and S. Wang, Motion Estimation for Moving Target Detection, *IEEE Trans. Aerospace and Electronic System*, July, 1996, vol. 32, no. 3, pp. 866-874.
- [5] S. M. Smith, Asset-2: Real-Time Motion Segmentation and Object Tracking, *Real-Time Im.*, 1998, vol. 4, pp. 21-40.
- [6] S. M. Benoit, F. P. Ferrie, Monocular optical flow for real-time vision systems, *in Proc. ICPR*, 1996, vol. 1, pp. 864 -868.