

슈퍼픽셀 기반의 그래프 컷을 이용한 객체 추적

이대연, 김창수
고려대학교

daeyounlee@mcl.korea.ac.kr, cskim@mcl.korea.ac.kr

Visual Object Tracking Using Superpixel-Based Graph Cuts

Dae-Youn Lee, Chang-Su Kim
Korea University

요 약

본 논문에서는 슈퍼픽셀(superpixel) 단위의 그래프 컷 알고리즘을 적용하여 객체 추적의 정확도를 향상시키기 위한 방법을 제안한다. 먼저 영상 분할 기법을 사용하여 입력 영상을 슈퍼픽셀로 분할하고 각 슈퍼픽셀에서 색상 히스토그램을 이용한 특성 벡터를 생성한다. 그리고 특성 벡터에 지지벡터기계(support vector machines)를 사용하여 각 슈퍼픽셀의 객체 확률 값을 추정한다. 객체 확률 값을 데이터 항(data term)으로, 이웃한 슈퍼픽셀 간의 특성 벡터 차 값을 스무드 항(smooth term)으로 하여, 그래프 컷(graph cuts) 알고리즘으로 슈퍼픽셀들을 객체와 배경으로 분류하고 객체 슈퍼픽셀을 최대한으로 포함하는 객체 윈도우를 찾는다. 실험 결과는 제안하는 기법이 기존 기법들보다 객체 추적 성능이 우수함을 보여준다.

1. 서론

테러와 같은 반사회적 범죄의 발생 빈도가 증가함에 따라 영상 보안 기술에 대한 관심이 지속적으로 증가하고 있다. 영상 보안 기술의 하나로써 객체 추적 기법에 관한 연구가 많이 진행되고 있다[1].

Comaniciu 는 객체 윈도우 단위의 색상 히스토그램을 이용한 객체 추적 기법을 제안하였다[2]. 중심점 이동을 통하여 객체 위치를 빠른 속도로 찾지만 객체 윈도우 단위의 단일 객체 모델을 사용함으로써 객체의 외형을 표현하는 데 한계를 갖는다. Wang 은 슈퍼픽셀 단위의 색상 히스토그램으로 객체와 배경 모델을 생성하여 객체 추적의 정확도를 높였다[3]. 하지만 배경 슈퍼픽셀이 객체 슈퍼픽셀과 유사한 경우 객체 추적의 정확도가 하락하는 단점을 갖는다.

본 논문에서는 슈퍼픽셀 단위의 그래프 컷 알고리즘을 사용하는 효과적인 객체 추적 기법을 제안한다. 슈퍼픽셀 단위의 색상 히스토그램으로 특성 벡터를 생성하고 특성 벡터에 지지벡터기계를 적용하여 객체 확률 값을 추정하고 데이터 항에 사용한다. 또한 인접한 슈퍼픽셀간의 색상 히스토그램 차이를 스무드 항에 사용한다. 그래프 컷을 이용하여 각 슈퍼픽셀을 객체와 배경으로 분류하고 객체 슈퍼픽셀을 최대한으로 포함하는 객체 윈도우 위치를 계산한다. 실험 결과는 제안하는 기법이 기존 기법들보다 정확한 객체 추적 성능을 제공함을 보여준다.

* 이 논문은 2013 년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구(No. 2012-011031) 및 2012 년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2012-0000916).

본 논문의 구성은 다음과 같다. 2 절에서는 제안하는 기법에 대해서 설명하고 3 절에서는 제안한 기법의 성능을 실험을 통해서 확인한다. 마지막으로 4 절에서는 본 논문에 대한 결론을 맺는다.

2. 제안하는 기법

2.1 외형 모델

본 논문에서는 SLIC 기법[4]을 적용하여 입력 영상을 슈퍼픽셀로 분할한다. 그리고 각 슈퍼픽셀에서 HSV 색상 히스토그램을 생성한다. HSV 색상 히스토그램은 각 색상 채널마다 16 개의 빈(bin), 총 48 개의 빈을 사용하였다. i 번째 슈퍼픽셀의 HSV 색상 히스토그램에서 생성한 특성 벡터를 \mathbf{f}_i 이라고 정의한다.

첫 프레임에서는 객체 특성 벡터와 배경 특성 벡터를 생성하고 지지벡터기계[5]를 학습한다. 객체 특성 벡터는 사용자에 의해 정의된 객체 영역에 포함된 슈퍼픽셀에서 추출하고 배경 특성 벡터는 객체 영역 밖 불규칙한 위치의 슈퍼픽셀에서 300 개를 추출한다. 객체 특성 벡터들의 클래스를 1 로 배경 특성 벡터들의 클래스를 -1 로 정의하고 다음 학습데이터집합 L 에 대하여 지지벡터기계를 학습한다.

$$L = \{(\mathbf{f}_i, c_i) | \mathbf{f}_i \in \mathbb{R}^p, c_i \in \{-1, 1\}\}_{i=1}^n. \quad (1)$$

위 식에서 $p=48$, c_i 는 i 번째 특성 벡터의 클래스를 나타낸다.

2.2 객체 위치 추정

본 논문에서 제안하는 객체 위치 추정은 크게 3 단계로 구성된다. 먼저 각 슈퍼픽셀에서 특성 벡터에 지지벡터기계를 적용하여 객체 확률 값을 계산한다. 그리고 그래프 컷을 사용하여 슈퍼 픽셀을 객체와 배경으로 분류한다. 마지막으로

객체 슈퍼픽셀을 최대도 포함하는 객체 윈도우 위치를 찾는다.

지지특성벡터의 다차원 평면과 입력 특성 벡터 \mathbf{f}_i 사이의 거리를 d_i 로 정의할 때, 슈퍼픽셀 i 의 객체 확률 값은 다음과 같다.

$$p(d_i) = \begin{cases} 0.5 + \frac{d_i}{d_{\max}} & \text{if } d_i \geq 0, \\ 0.5 + \frac{d_i}{d_{\min}} & \text{if } d_i < 0. \end{cases} \quad (2)$$

d_{\max} 는 $D = \{d_i | i = 1, \dots, n\}$ 에서 최대값, d_{\min} 는 D 에서 최소값을 나타낸다.

그래프 컷을 이용하여 n 개의 슈퍼픽셀 레이블을 결정한다. ℓ_i 를 i 번째 슈퍼픽셀의 레이블이라고 할 때, 객체 추적에서는 객체 ($\ell_i = 1$) 및 배경 ($\ell_i = 0$)의 두 개 레이블 값을 갖는다. 레이블링 문제는 $L = \{\ell_1, \ell_2, \dots, \ell_n\}$ 으로 표현되며 에너지 함수는 다음과 같다.

$$E(L) = \sum_{i \in S} D_i(\ell_i) + \sum_{(p,q) \in N} V_{p,q}(\ell_p, \ell_q). \quad (3)$$

$D_i(\ell_i)$ 는 데이터 항으로써 객체 확률 값을 사용하여 정의한다.

$$D_i(\ell_i) = \begin{cases} p(d_i) & \text{if } \ell_i = 0, \\ 1 - p(d_i) & \text{otherwise.} \end{cases} \quad (4)$$

또한 $V_{p,q}(\ell_p, \ell_q)$ 는 스무드 항으로써 인접한 슈퍼픽셀 p 와 q 사이의 특성 벡터 \mathbf{f}_p 와 \mathbf{f}_q 를 사용하여 다음 식과 같이 주어진다.

$$V_{p,q}(\ell_p, \ell_q) = \gamma \exp(-\chi(\mathbf{f}_p, \mathbf{f}_q)) [\ell_p \neq \ell_q]. \quad (5)$$

$$\chi(\mathbf{f}_p, \mathbf{f}_q) = \sqrt{\frac{\sum_{j=1}^{48} (f_{p,j} - f_{q,j})^2}{(f_{p,j} + f_{q,j})}} \text{이고 } \gamma = 0.004 \text{이다.}$$

슈퍼픽셀의 레이블이 정해지고 난 후, 객체인 슈퍼픽셀을 최대도 포함하는 객체 윈도우 위치를 결정한다.

2.3 외형 모델 갱신

본 논문에서는 시간에 따른 객체의 변화를 반영하여 지지벡터기계를 갱신한다. 지지벡터기계를 갱신에는 첫 프레임 및 4개의 갱신 프레임을 사용한다. 갱신 프레임이란 객체 추적에 끝난 프레임들 중에서 객체 윈도우 내에 객체 레이블을 갖는 슈퍼픽셀이 10개 이상 있는 경우를 의미한다. 4개의 새로운 갱신 프레임이 정해지면 지지벡터기계를 갱신하게 된다. 각 갱신 프레임에서 객체 특성 벡터는 $d_i > 1$ 이면서 객체 윈도우 내에서 객체 레이블을 갖는 슈퍼픽셀에서 생성하고 배경 특성 벡터는 객체 윈도우 밖의 불규칙한 위치에서 300개의 슈퍼픽셀에서 생성한다.

3. 실험 결과

본 논문에서는 제안하는 기법의 성능을 평가하기 위하여 그림 1의 “Board,” “Lemming,” “Liquor” 실험 영상을 사용한다.

제안하는 기법의 성능 분석을 위해 STRUCK[6] 기법과 SPT[3] 기법과 비교하여 평균 오차 거리를 계산한다. 여기서 평균 오차 거리는 객체 추적 결과와 실측 정보의 거리의 차이 값을 나타낸다. 표 1은 각 실험 영상에서 제안하는 기법과 비교 기법들의 평균 오차 거리를 나타낸다. 본 논문에서 제안된 기법과 SPT 기법을 비교하였을 때 오차를 측면에서 평균적으로 약 55%의 성능향상이 있다.



Board Lemming Liquor
그림 1 실험 영상. 노란 윈도우 내부는 추적해야 하는 객체.

표 1 제안하는 기법과 STRUCK 기법 [6], SPT 기법 [3]의 객체 추적 성능 비교.

	STRUCK	SPT	Proposed
Board	38.8	167.6	21.3
Lemming	23.5	35.1	17.3
Liquor	128.0	63.4	62.1
Average	63.4	74.6	33.6

4. 결론

본 논문에서는 슈퍼픽셀 단위의 색상 그래프 컷 알고리즘을 이용한 객체 추적 기법을 제안하였다. 제안하는 기법에서는 각 슈퍼 픽셀에서 색상 히스토그램을 추출하고 지지벡터기계 분류기를 사용하여 객체 확률을 구한다. 그리고 객체 확률을 데이터 항으로 슈퍼 픽셀간의 색상 히스토그램 거리를 스무드 항으로 하여 그래프 컷 기법으로 객체인 슈퍼픽셀을 결정하고 최종적으로 객체 위치를 결정한다. 실험을 통하여 제안하는 기법이 기존 방식에 비하여 향상된 객체 추적 성능을 보임을 확인하였다.

5. 참고문헌

- [1] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Comput. Surv.*, vol. 38, no. 4, Dec. 2006.
- [2] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–575, May 2003.
- [3] S. Wang, H. Lu, F. Yang, and M.-H. Yang, “Superpixel tracking,” in *Proc. ICCV*, 2011, pp. 1323–1330.
- [4] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk “SLIC superpixels”, *EPFL Technical Report*, no. 149300, June 2010.
- [5] G. Cauwenberghs and T. Poggio, “Incremental and decremental support vector machine learning,” in *Advances in Neural Information Processing Systems*, volume 13, 2001.
- [6] S. Hare, A. Saffari, and P.H.S. Torr, “STRUCK: Structured output tracking with kernels,” in *Proc. ICCV*, Nov. 2011, pp. 263–270.