

객체 검출 결과를 반영하는 영상 내 목표물 추적 기법

홍은태, 임종우

한양대학교 컴퓨터소프트웨어공학

honggeuntae@hanyang.ac.kr, jlim@hanyang.ac.kr

요 약

본 논문에서는 객체 검출 결과를 반영하여 객체를 효과적이고 안정적으로 추적하는 방법을 제안한다. 먼저 객체 검출 결과를 통해 추적하고자 하는 객체의 초기 위치와 크기를 초기화한다. 그리고 인접한 프레임에서의 객체의 위치를 Lucas-Kanade 기법을 사용해 점진적으로 추적한다. 이와 같은 방법으로 추적을 하는 경우 조도변화, 가려짐과 같은 환경변화에 취약하기 때문에 보다 효과적으로 주성분 기저벡터를 갱신하여 객체를 추적하는 방법은 제안하였다. 또한, 추적이 실패하는 지점을 찾고 추적과 객체 검출을 병행하여 일반적으로 추적이 어려운 상황에서도 추적하고자 하는 객체를 강건하게 추적하는 방법을 제안한다.

1. 서론

영상에서의 객체 추적은 무인자동차, 보행자 검출과 같은 자동차 관련 산업에서 이슈가 되고 있다. 이처럼 영상 내에서 원하는 객체를 찾기 위한 방법으로는 추적기(Tracker)를 사용하는 방법과 검출기(Detection)를 사용하는 방법으로 크게 나눌 수 있다. 추적기의 경우 일반적으로 인접한 프레임에서 템플릿(Template)과 가장 유사한 윈도우를 가까운 지점에서 찾는 방법이며 대표적으로 Lucas-Kanade 와 Particle Filter 를 사용하는 방법이 있다 [1][2]. 추적기(Tracker)는 템플릿(Template) 근처의 영상정보를 사용하기 때문에 보다 안정적인 결과를 보여주지만, 초기 템플릿(Template)을 사용자가 직접 지정을 해주어야 하며, 다양한 환경변화에서 추적이 실패하는 경우 다시 객체를 찾지 못하는 단점이 있다. 반면 특징 기반의 검출기(Detection)는 모든 프레임에서 찾고자 하는 객체를 찾을 수 있지만, 영상 전체를 검출하여 결과를 도출하기 때문에 잘못된 결과를 포함하는 경우가 많고 올바른 객체를 검출하더라도 결과가 안정적이지 못하다. 이에 본 논문에서는 두 가지 알고리즘을 복합적으로 사용하여 안정적인 모션 모델을 선택하고, 가려짐과 조도변화 등 환경적인 변화에도 안정된 결과를 보이는 추적기술을 제안한다.

본 논문의 구성은 2 장에서 제안하는 올바른 추적 결과로 구성된 Eigen-space 와 이를 사용하는 추적기(Tracker)를 소개하고 추적이 실패하는 구간을 검출하고 검출결과를 추적기(Tracker)에 반영하는 방법이 3 장에 소개되며 4 장과 5 장에서는 실험 결과와 논문에 대한 결론을 맺는다.

2. 추적기(Tracker)

장시간 추적에서 추적하고자 하는 객체는 항상 조도변화, 가려짐, 타겟 윈도우 배경변화 등 다양한 변화가능성을 내포한다. 이에 대하여 장시간 안정적인 추적을 하기 위해서는 단순히 타겟 윈도우에 대하여 픽셀단위의 추적 알고리즘을 적용하는 것은 불가능하다. 본 논문에서는 IVT(Incremental Visual Tracking) 에서의 추적결과로 만들어진 Subspace 를 끊임 없이 갱신하고 이를 추적결과에 반영해 다양한 환경에서 강건하게 객체를 추적하는 방법을 사용한다 [3]. 이때, 모든 추적결과가 아닌 올바른 추적결과를 기반으로 기저벡터 공간을 구축하여 검출(Detection)과 함께 하나의 타겟을 장시간 추적하는데 유리한 방법을 제안한다.

2.1 Lucas-Kanade Algorithm

기존의 IVT 논문에서는 파티클필터를 사용하여 모션 모델을 추정한다. 파티클필터는 현재 가장 많이 사용되는 확률적 기반의 모션모델 추정 기법 중 한가지 이지만 가우스분포를 따르는 랜덤의 샘플 모델을 생성하기 결과가 안정적이지 않다. 그러므로 본 논문에서는 보다 안정적인 모션 모델을 추정하기 위한 방법으로 Lucas-Kanade 알고리즘을 사용한다. 하지만 optical flow 기반의 Lucas-Kanade 알고리즘은 조도변화에 민감하다는 단점을 가지고 있고, 이를 해결하기 위하여 취득된 영상의 모서리(Edge) 특징을 사용하였다.

Lucas-Kanade 알고리즘은 모션모델을 추정하

기 위한 방법으로 Gradient descent 를 사용한다. $W(x; p)$ 를 warping parameter set 으로 보면 p 는 parameter vector 이다. $T(x)$ 는 추적하고자 하는 target object 의 템플릿, $I(x)$ 는 취득된 이미지로 그리고 $x = (x, y)^T$ 는 pixel coordinates 으로 나타낼 수 있다. Lucas-Kanade 알고리즘의 목표는 템플릿 $T(x)$ 와 이미지에서 모션모델 parameter 를 통해 warping 된 이미지 $I(W(x; p))$ 의 차이를 최소화하는 모션모델을 추정하는 것이므로 아래와 같이 나타낼 수 있다.

$$e = I(W(x; p)) - T(x) \quad (1)$$

여기서 템플릿 이미지 $T(x)$ 는 이미지 평균값 μ 을 중심으로 하는 기저벡터로부터 확장된 subspace 라고 가정하고 이를 Eigen space 한다. 이에 우리는 새로운 열 벡터 e^* 를 다음과 같이 정의할 수 있다.

$$e^* = \sum_{i=1}^t C_i U_i \quad (2)$$

위 식에서의 는 Eigen-space 방향의 주성분 벡터 U 를 템플릿과 warping 이미지의 차이인 e 의 subspace 로 프로젝션하는 변수로, U 와 e 를 내적하여 구한다. 결국 기저벡터를 e 에 반영하는 식은 아래와 같다.

$$\sum_x [I(W(x; p)) - T(x) - e^*]^2 \quad (3)$$

위 식(4)는 최소자승법 문제이며 이는 Gradient descent 를 통해 아래와 같이 추정된다.

$$\sum_x [I(W(x; p + \Delta p)) - T(x) - e^*]^2 \quad (4)$$

$$p \leftarrow p + \Delta p \quad (5)$$

위 식(5)를 반복적으로 수행해 최적화된 모델 파라미터 p 를 추정할 수 있다. 이때, Δp 를 구하기 위하여 식(3)을 테일러 급수로 나타내면 다음과 같고,

$$\sum_x [I(W(x; p)) + \nabla I \frac{\delta W}{\delta p} \Delta p - T(x) - e^*]^2 \quad (6)$$

이를 Δp 에 대하여 정리하면

$$\Delta p = H^{-1} \sum_x [\nabla I \frac{\delta W}{\delta p} \Delta p]^T [T(x) - I(W(x; p))]^2 \quad (7)$$

이고, 이때 H 는 Gauss-newton method 의 Hessian matrix 로 $J * J^T$ 이다. 결국 식(9)의 Δp 를 반복적으로 식(5)에 대입하여 적절한 모션모델을 추정할 수 있다.

2.2 Incremental Visual Tracking

주어진 하나의 영상 데이터를 I_i , 이 데이터들을

행렬로 나타낸 것을 $A = \{I_1, \dots, I_m\}$ 라고 한다면 A 에 대하여 SVD 를 함으로서 쉽게 각 영상의 기저 벡터를 구할 수 있다. 하지만 지속적인 영상에 대해 A 는 지속적으로 증가하게 된다. 이때 추가된 영상 데이터 배열을 $E = \{I_{n+1}, \dots, I_m\}$ 라고 하면 우리가 구하고자 하는 기저벡터는 $A' = (A | E)$ 에 대한 SVD 결과가 된다. 이처럼 구하고자 하는 배열의 크기가 지속적으로 변하는 경우 IVT 논문에서 제안하는 R-SVD 를 통하여 효율적으로 기저벡터를 구할 수 있다 [3].

IVT 의 방식을 사용하면 추적하고자 하는 객체의 기저벡터로 이루어진 Eigen-space 를 지속적으로 갱신하고 이를 반영하여 다양한 환경에서 안정적인 모션모델을 추정할 수 있다. 하지만 모든 추적 결과를 사용해 기저벡터를 갱신한다는 것은 항상 정확한 결과를 보장할 수는 없다. 또한 잘못된 결과로 기저벡터가 갱신되는 경우 추적기(Tracker)는 잘못된 객체를 추적하게 될 가능성이 커진다. 이에 대하여 보다 안정적으로 객체를 추적하기 위하여 본 논문에서는 객체

검출 결과와 전후추적결과(Forward-Backward Tracking) 사용하여 보다 강건하게 Eigen-Space 를 갱신하고 추적을 오래 지속할 수 있는 방법을 제안한다.

3. 검출결과를 반영하는 추적 기법

일반적인 실외 환경에서의 추적은 추적하고자 하는 객체와 환경의 변화로 인하여 지속적으로 온전한 모션모델을 구하는 것은 불가능하다. 또한, 추적하고자 하는 객체를 사용자가 직접 지정해야 하는 것 역시 어플리케이션으로서는 치명적인 문제점이다. 이를 보완하기 위해서 본 구현에서는 Latent-SVM 알고리즘의 검출(Detection)결과를 반영하고 초기화 및 추적이 실패하는 구간을 찾아 잘못된 Eigen-space 가 업데이트 되는 문제점을 해결하는 더욱 강건한 추적기법을 제안한다.

3.1 Forward-Backward Error

추적이 실패하는 지점을 찾기 위하여 본 구현에서 사용한 방법은 Forward-Backward Error 과 오버랩이다 [4]. 원래 Forward-Backward Error 는 region tracking 에서 특징 혹은 픽셀에 대한 신뢰도를 구하는데 사용된 기법이나, 본 논문에서는 추적이 실패하는 지점을 검출하는데 사용되었다.

아래 그림 1. 처럼 전-후로의 추적을 수행하여 추정된 모션모델을 비교해 두 개의 인접한 프레임에서 현재 타겟의 상태가 추적하기에 적합한 환경 인지를 예측할 수 있다. Forward-Backward error 가 크다면 조도변화, 가려짐, 타겟 윈도우 내 변하는 배경으로부터 추적이 영향을 받았다고 판단할 수 있으며, 이 경우에는 추적 결과 이미지를 Eigen-Space 를 위한 데이터로 사용하지 않는다.

이를 통해 더욱 강건한 Subspace 를 만들 수 있다.

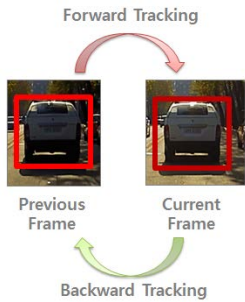


그림 1. Forward-Backward Tracking

2.2 Data Association

Latent-SVM 검출(Detection)은 아래 그림 2. 처럼 한 개 이상의 결과를 보인다 [5]. 이에 추적하고자 하는 객체와 검출결과와의 관계를 정의해야 한다. 본 논문에서는 추적기(Tracker)와 검출(Detection) 결과의 오버랩(Overlap)을 통해 관계를 정의했으며, 추적결과와의 오버랩(Overlap)이 일정 수준보다 낮으면 추적이 완전히 실패했다고 가정하고, 새로운 검출결과로 추적알고리즘의 정보를 초기화 하였다.

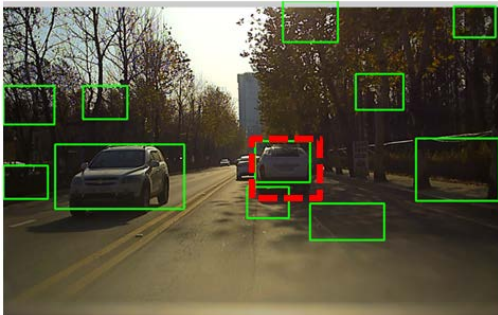


그림 2. Latent-SVM 검출과 추적 결과. 위 그림에서 초록색 윈도우는 Latent-SVM 검출 결과를, 그리고 빨강색 점선은 추적 결과를 나타냄.

2.2 Data Selection

Data association 으로 타겟에 대한 검출 결과를 찾더라도 모든 결과를 추적의 초기화에 사용할 수는 없다. 아래 그림 3. 은 검출(Detection) 결과 중 오버랩(Overlap)을 통해 정의된 현재 추적하는 차량에 대한 검출 결과를 보여준다. 그림 상단의 초록색 윈도우는 검출(Detection)결과 중 잘못된 결과를 보여주는데, 이러한 잘못된 검출 결과를 사용해 추적기(Tracker)를 초기화한다면 오히려 지속적으로 잘못된 결과를 추정할 가능성이 커진다. 이에 본 논문에서는 현재 프레임에서 관계가 정의된 검출결과와 이전 프레임에서 추적기(Tracker)의 추정결과 윈도우의 차를 최소화 하는 모션모델을 추정 후 이를 비교하여 검출결과 중 초기화에 사용하기에 적합한 결과를 선택한다.

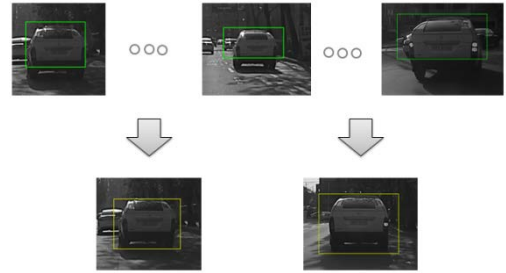


그림 3. Data Association 으로 정의된 추적하는 추적하는 타겟에 대한 검출(Detection)결과. 상단의 초록색 윈도우는 잘못된 결과를 포함한 모든 결과를, 그리고 하단의 노란색 윈도우는 제안하는 알고리즘으로 선택된 검출 결과를 나타냄.

2.2 Integrator

IVT 는 지속적으로 Eigen-Space 와 이미지 평균을 업데이트 함으로서 다양한 환경변화에도 안정적으로 추적이 가능하다. 하지만 잘못된 Eigen-Space 의 구성으로 인한 드리프트(Drift)의 가능성을 가지고 있다. 이러한 단점을 보완하기 위하여 최대한 성공적인 추적결과를 Eigen-Space 갱신에 반영해야 한다. 또한 차량의 좌우회전은 optical flow 관점에서 추적하는 객체가 완전히 다른 객체로 인식되는 위험성을 내포하고 있다. 이렇게 객체가 변하는 경우는 현재까지 구성된 Eigen-Space 를 초기화하고 새로운 객체로서 추적하는 객체를 인식하는 것이 바람직하다. 이에 본 구현에서는 Forward-Backward Error 를 통해 1 차적으로 추적이 실패하는 지점을 검출한다. 하지만 Forward-Backward Error 는 모든 추적실패 구간을 검출하지는 못한다. 실험결과 빠른 객체의 이동 혹은 심각한 가려짐의 경우 확실하게 실패지점을 찾아낼 수 없었다. 이에 대해 본 구현에서는 Data Association 이 확실하다면 안정적 모션모델은 아닐지라도 Latent-SVM 결과는 타겟을 탐지했다고 가정한다. 그리고 Forward-Backward Error 가 발견하지 못한 추적 실패 지점은 심각한 가려짐 혹은 빠른 객체 이동 같은 이후의 추적 역시 확실하게 실패하게 되는, 추적하는 윈도우에 대한 큰 변화이기 때문에 Eigen-space 와 이미지 평균을 초기화한다. 이로 인해 긴 시퀀스에 대한 추적으로 Eigen-space 가 잘못 구성되어 추적이 완전히 실패하는 경우에도 새로운 객체에 대한 추적이 가능하다.

4. 실험 결과

IVT 는 대부분의 영상 데이터에 대하여 2D 기반 추적 알고리즘 중 좋은 성능을 보여준다. 하지만 장시간에 추적에서 다양한 환경변화의 에러의 누적은 복원될 수 없는 실패를 초래한다. 본 논문과 기존의 IVT 와의 차이점은 첫째, 모션 모델 추정을 위한

방법으로 Lucas-Kanade 를 사용하여 여러 프레임에 대하여 안정적인 모션모델 추정을 가능하게 하였다. 둘째, 환경적 요인으로 인해 객체 추적이 실패하는 지점을 검출하여 올바른 모션 모델을 통해 Eigen-Space 를 구성하도록 하였으며, 마지막으로 추적하고자 하는 객체가 심각하게 변하는 경우를 탐지하여 Eigen-Space 및 영상 벡터 평균을 다시 초기화하여 긴 시퀀스에 대하여 사용자의 도움 없이 안정적인 추적을 가능하게 한다.



그림 4. 차량 블랙박스 영상에 대해 본 논문 결과와 IVT 그리고 ground truth 를 윈도우로 나타낸 그림. 빨간색 윈도우는 ground truth 를 나타내고, 파란색은 IVT 결과를 그리고 초록색 윈도우는 본 논문에서 제안하는 방법의 결과를 나타냄.

위 그림 4. 의 하단의 두 개 영상은 추적기만 사용할 때와 본 논문에서 제안하는 방법의 차이점을 보여준다. 추적기만 사용하는 경우는 아래 그림처럼 타겟의 다양한 변화 때문에 추적이 실패하는 경우 지속적으로 추적을 수행할 수 없지만, 제안하는 방법의 경우 다양한 환경에서도 강건히 추적이 가능하다.

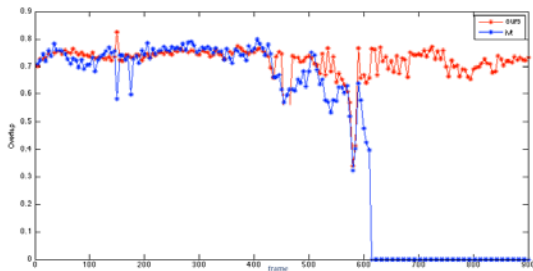


그림 5. 자동차 블랙박스 영상 내 차량 추적 결과. x 축은 프레임, y 축은 Ground Truth 와의 오버랩을 나타내며 파란색은 IVT 결과를, 빨간색은 본 논문의 결과를 나타냄

위 그림 5. 는 IVT 와 본 구현의 결과와 Ground truth 의 오버랩 그래프이다. 약 600 프레임부터 IVT 만으로는 추적이 불가능 했으며 추적이 되는 지점까지의 표준편차는 본 구현은 0.0182 그리고 IVT 는 0.0385 로서 본 구현이 좀더 안정적인 결과를 나타내었다. 또한 아래 그림 4. 와 그림 5. 는 Latent-SVM 과 본 논문의 스케일 파라미터의 변화량을 통해 본 구현이 훨씬 안정적인 모션모델을 추정 가능함을 보여준다.

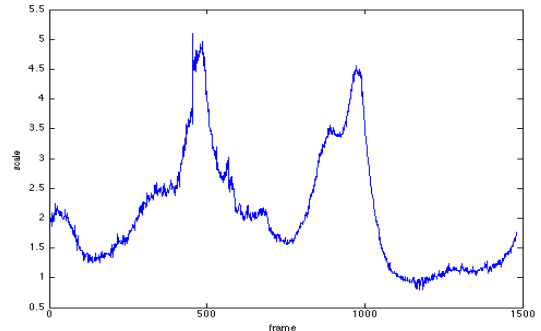


그림 6. Latent-SVM 검출의 스케일 값

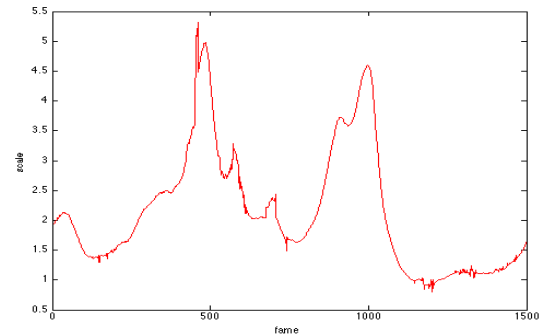


그림 7. 본 논문 결과의 스케일 변화 값

5. 결론

본 논문에서 제안하는 방법은 전후추적결과를 통해 보다 강건하게 하나의 타겟을 추적하고, 검출 결과를 사용해 실외의 다양한 환경에서 장시간 추적을 가능하게 한다. 위 실험 결과처럼 본 논문은 효과적으로 Eigen-Space 를 갱신하여 다양한 환경에서 강건히 추적이 가능함을 보였다. 제안하는 방법은 검출결과를 사용하여 추적기(Tracker)의 성능을 향상시키지만, 지속적으로 잘못된 검출결과를 사용하게 될 경우 전체적인 성능에 부정적인 영향을 초래하는 한계점이 있다. 이러한 문제점은 올바른 추적결과를 통해 검출(Detection)기법에서 사용하는 분류기를 지속적으로 학습하고, 추적하는 타겟에 대한 검출성능을 향상시켜 해결할 수 있다. 또한 구성된 Eigen-Space 를 사용해 검출기법의 성능을 향상시킬 수 있을 것으로 예측된다.

참고문헌

- [1] Baker, Simon, and Iain Matthews. "Lucas-kanade 20 years on: A unifying framework." *International journal of computer vision* 56.3 (2004): 221-255.
- [2] Isard, Michael, and Andrew Blake. "Condensation—conditional density propagation for visual tracking." *International journal of computer vision* 29.1 (1998): 5-28.
- [3] Ross, David A., et al. "Incremental Visual Tracking."
- [4] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. "Forward-backward error: Automatic detection of tracking failures." *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010.
- [5] Felzenszwalb, Pedro, David McAllester, and Deva Ramanan. "A discriminatively trained, multiscale, deformable part model." *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008.