

다중 카메라 환경에서의 실시간 객체 추적

조상현⁰, 강행봉
가톨릭 대학교 컴퓨터 정보 공학부
{cshgreat⁰, hbkgang}@catholic.ac.kr

Real-time object tracking in Multi-Camera environments

Sang-Hyun Cho⁰ and Hang-Bong Kang
Dept. of Computer Eng. The Catholic University of Korea

요 약

비디오 시퀀스에서의 객체 추적은 보안 및 감시 시스템(Security and surveillance system), 비디오 원격 회의(Video teleconferencing)등과 같이 컴퓨터 비전 응용 분야에 널리 이용되어, 점점 그 중요성이 증가하고 있다. 여러가지 이유로 인해 카메라 뷰(View)로부터 객체의 가시 상태가 변하는 경우, 하나의 뷰만을 이용해서는 좋은 결과를 가지기 어렵기 때문에 본 논문에서는 객체가 가장 잘 나타나는 뷰를 선택해서 객체를 추적하는 방법을 제안한다. 각각의 카메라 뷰에서 객체를 추적하기 위해 본 논문에서는 다중 후보가 결합된 Mean-shift 알고리즘을 이용한다. 제안된 시스템의 경우, 복잡한 환경으로 인해 객체의 가시 상태가 변하는 환경에서 단일 뷰를 이용하는 경우와 비교해 더 나은 성능을 가질 수 있었다.

1. 서 론

실시간 객체 추적(Real-time Object Tracking)은 비디오 감시 시스템, 비전 기반 네비게이터와 같은 비전 응용 산업이 발달하면서 그 중요성이 더해지고 있는 분야이다.

다중 카메라를 이용한 추적 방법은 Doubek 등[1]에 의해 디지털 카메라와 컴퓨터로 구성된 비주얼 룸 시스템(Visual Room System)에 사용되었다. Comaniciu 등[2]은 Mean-shift 알고리즘을 이용해서 적은 계산량을 필요로 하는 시스템을 제안했고, Khan 등[3]은 캘리브레이션 하지 않은 다중 카메라를 이용해 사람을 추적하는 시스템을 제안했다. 또한 Nummiaro 등[4]은 Particle filter를 이용해서 다중 카메라 환경에서의 칼라를 기반으로 한 객체 추적 방법을 제안했다.

본 논문에서는 다중 후보를 결합한 Mean-shift 알고리즘을 이용해 다중 카메라 환경에서 객체를 추적하는 방법을 제안한다. 제안된 시스템의 경우, 다중 카메라 환경에서 객체의 가시 상태가 가장 좋은 뷰를 선택해 추적을 함으로써 객체의 가시상태가 변하는 환경에서 비교적 좋은 성능을 가질 수 있었다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 본 논문에서 제안하는 시스템의 각각의 구성에 대해 설명하고

3장에서는 실험결과를 기술한다.

2. 다중 후보가 결합된 Mean-shift를 이용한 다중 카메라 환경에서의 실시간 객체 추적

2.1. 시스템의 구성

본 논문에서 제안하는 시스템의 구성은 그림 1과 같다.

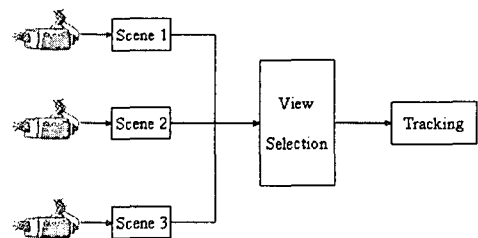


그림 1. 시스템 구성도

먼저, 카메라들로부터 얻은 이미지를 획득한 후, 각 이미지에서의 객체 가시 상태를 판단한 후, 객체의 가시상태가 가장 좋은 이미지를 선택하고, 그 이미지를 이용해 객체를 추적한다.

2.2. 다중 후보가 결합된 Mean-shift

Fukunaga와 Hostetler[5]에 의해 제안된 Mean Shift 알고리즘은 Point Sample Distribution의 가장 가까운 모드(Mode)를 찾는 비매개변수적(Non-Parameteric) 방법으로 객체의 칼라정보와 Mean shift알고리즘을 이용한 객체 추적 알고리즘은 구현이 간단하고, 적은 계산 복잡도를 갖는 장점이 있어서 실시간 객체 추적 시스템에 적합하다고 할 수 있다. 하지만, 지역 모드(Local Mode)로 수렴하는 특성으로 인해 객체의 주변(neighborhood)에 지역 모드가 존재하는 복잡한 환경에서는 좋은 성능을 나타내지 못하는 단점을 가지고 있다. 또한, 단일 후보를 가지기 때문에 복잡한 환경으로 인해 잘못된 위치로 수렴했을 경우, 추적에 실패하게 된다.

이러한 문제점을 극복하기 위해 본 논문에서는 복수개의 후보를 사용하는 방법을 이용한다. 먼저 후보들을 생성하기 위해 후보 생성 범위를 결정해야 하는데. 이를 위해 객체의 모션(motion) 정보가 이용된다.

$I(x,y,t)$ 와 $I(x,y,t+1)$ 을 각각 $t, t+1$ 프레임의 이미지라고 할 때, 다음을 만족한다고 가정한다.

$$I(x + p(x,y,t), y + q(x,y,t), t + 1) = I(x, y, t) \quad (1)$$

여기서, $(p(x,y,t), q(x,y,t))$ 는 $t, t+1$ 프레임 사이의 객체의 모션에 의한 이동(displacement)이다. 테일러 전개를 이용해 (1)식을 근사시키면 다음과 같다.

$$I(x + p, y + q, t + 1) = I(x, y, t) + pI_x + qI_y + I_t \quad (2)$$

여기서, $I_x = \frac{\partial I(x, y, t)}{\partial x}, I_y = \frac{\partial I(x, y, t)}{\partial y}$,

$I_t = \frac{\partial I(x, y, t)}{\partial t}, p = p(x, y, t), q = q(x, y, t)$ 이다.

(1)식과 (2)식에 의해, 다음 조건을 만족한다[6].

$$pI_x + qI_y + I_t = 0 \quad (3)$$

따라서, 다음과 같은 에러함수를 최소화 하는 (p, q) 가 객체의 모션 정보이다.

$$Err^{(t)}(p, q) = \sum_{(x,y) \in R} (pI_x + qI_y + I_t)^2 \quad (4)$$

객체의 위치를 추정하기 위해 객체의 상태벡터 \mathbf{s} 를 다음과 같이 정의한다.

$$\mathbf{s} = (x, y) \quad (5)$$

여기서, x, y ,는 객체의 중심좌표이다.

모션 정보를 이용해 후보를 다음과 같이 생성한다.

$$\mathbf{s}_t = A(\mathbf{s}_{t-1} + (p, q)) + \mathbf{w}_{t-1} \quad (6)$$

여기서, A 는 model의 deterministic component, (p, q) 는 모션, \mathbf{w}_{t-1} 은 multivariate random variable이다
다음으로, 생성된 샘플들을 이용해서 각 샘플에 대응되는 칼라 분포는 다음과 같이 표현된다.

$$p_y^{(n)} = f \sum_{i=1}^l k \left(\frac{\|\mathbf{y}^n - \mathbf{x}_i^n\|}{a} \right) \delta[h(\mathbf{x}_i^n) - u] \quad (7)$$

여기서, n 은 샘플 인덱스(Sample Index), l 는 타원 안의 pixel의 수, δ 는 Kronecker delta 함수, a 는 $\sqrt{H_x^2 + H_y^2}$, f 는 normalization factor이다.

생성된 샘플들의 칼라 분포를 이용해서, 다음의 weight를 다음과 같이 계산한다.

$$w_i^n = \sum_{u=1}^m \sqrt{\frac{\hat{q}}{\hat{p}_n(\mathbf{y}_0)}} \delta[b(\mathbf{x}_i^n) - u] \quad (8)$$

Mean-shift 알고리즘과 계산된 weight들을 이용해서, Mean-shift 벡터를 다음과 같이 계산한 후, 후보들의 위치를 Mean-shift 벡터 방향으로 옮겨준다.

$$\mathbf{y}^n = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i^n w_i^n g \left(\left\| \frac{\mathbf{y}_0^n - \mathbf{x}_i^n}{h} \right\|^2 \right)}{\sum_{i=1}^{n_h} w_i^n g \left(\left\| \frac{\mathbf{y}_0^n - \mathbf{x}_i^n}{h} \right\|^2 \right)} - \mathbf{y}_0^n \quad (9)$$

타겟과 칼라 분포가 비슷한 후보에게 높은 가중치를 주기 위해, 다음과 같은 weight function을 정의한다.

$$\pi_i = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1 - (\sum_{u=1}^m \sqrt{\hat{p}^{(u)} q^{(u)}})}{2\sigma^2}} \quad (10)$$

각 단위(시간) t 에서의 객체의 상태는 다음과 같이 추정된다.

$$E[S] = \sum_{n=1}^N \pi^{(n)} S^{(n)} \quad (11)$$

2.3. 뷰의 선택

객체의 가시 상태가 가장 좋은 뷰를 선택하기 위해 본 논문에서는 초기 타겟이 가지는 칼라 분포와 각 뷰에서 계산되는 위치에서의 칼라 분포의 유사도를 다음과 같이 측정한다.

$$\rho[p, q] = \sum_{u=1}^m \sqrt{p^{(u)} q^{(u)}} \quad (12)$$

여기서, q 는 초기 타겟이 가지는 칼라 분포이고, p 는 뷰에서 계산된 타겟위치에서 가지는 칼라 분포이다. 따라서, 객체의 가시 상태가 가장 좋은 뷰는 (12)식의 값이 최대가 되는 뷰이다.

3. 실험 및 결과

본 실험은 P4-1.5Ghz의 성능을 가지는 PC에서 수행했으며, 모든 Sequence는 320x240 해상도를 가지는 Bitmap sequence를 사용했고, 후보개수(N)은 10을 사용했다.

그림 2는 다중 카메라 환경에서 얼굴 추적을 한 결과이다. (a)는 제안된 방법을 이용해 객체의 가시 상태가 가장 좋은 뷰를 선택해 추적한 결과이고, (b)는 동일 프레임에서의 각 카메라(A, B, C)별 추적 결과이다. 32 프레임을 보면 객체의 가시상태가 A 카메라 뷰에서 가장 좋은 것을 볼 수 있다. 따라서, 제안된 시스템은 A 카메라의 뷰(1-A)를 선택해 좋은 추적 결과를 나타낸다. 하지만, 143 프레임에서는 A 카메라에서의 객체 가시 상태보다 B 카메라에서의 가시 상태가 더 좋기 때문에 제안된 시스템은 B 카메라의 뷰(2-B)를 선택해 좋은 추적 결과를 나타낸다.

하지만, A 카메라의 경우 32 프레임에서는 좋은 결과(1-A)를 나타내지만, 143 프레임에서는 그렇지 못하다(2-B). B, C 카메라의 경우도 비슷한 결과를 볼 수 있다.

4. 결론

기존의 단일 뷰에서의 객체추적의 경우, 뷰 안에서의 객체의 가시 상태가 좋지 못할 경우, 좋은 결과를 얻기가 힘들었다. 하지만, 다중 뷰를 이용해, 하나의 카메라 뷰에서 객체의 가시 상태가 좋지 못하더라도, 객체의 가시 상태가 좋은 다른 카메라 뷰를 이용해서 추적을 실행함으로써 단일 뷰를 이용할 때보다 좋은 결과를 얻을 수 있었다.

하지만 시스템의 성능을 결정하는 파라미터들을 적절하게 결정해야 하는 단점을 가지고 있다. 이러한 시스템 파라미터들을 적응적으로 결정 할 수 있는 연구가 추가로 수행되는 것이 필요하다.

참고문헌

[1] P. Douthek, T. Svoboda and L. Van Gool, Monkeys - a Software Architecture for ViRoom - Lpw-Cost Multicamera System, ICVS, pp. 386-395, 2003.
 [2] D. Coumaniciu, V. Ramesh and P. Meer, Real-Time Tracking of Non-Rigid Objects using Mean-shift, CVPR, pp. 142-149, Vol. 2, 2000.
 [3] S. Khan, O. Javed and M. Shah, Tracking in Uncalibrated Cameras with Overlapping Field of View, PETS, 2001.
 [4] K. Nummiaro, E. K. Meier, T. Svoboda, D. Roth and L.V.Gool, Color-Based Object Tracking in Multi-Camera Environments, DAGM03, pp.591-599, 2003.
 [5] K. Fukunaga and D. Hostetler " The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition" IEEE trans. Information Theory vol. IT-21, 32-40, 1975.
 [6] B.K.P. Horn and B.G. Schunck. Determining optical flow. Artificial Intelligence, 185-203, 1981.

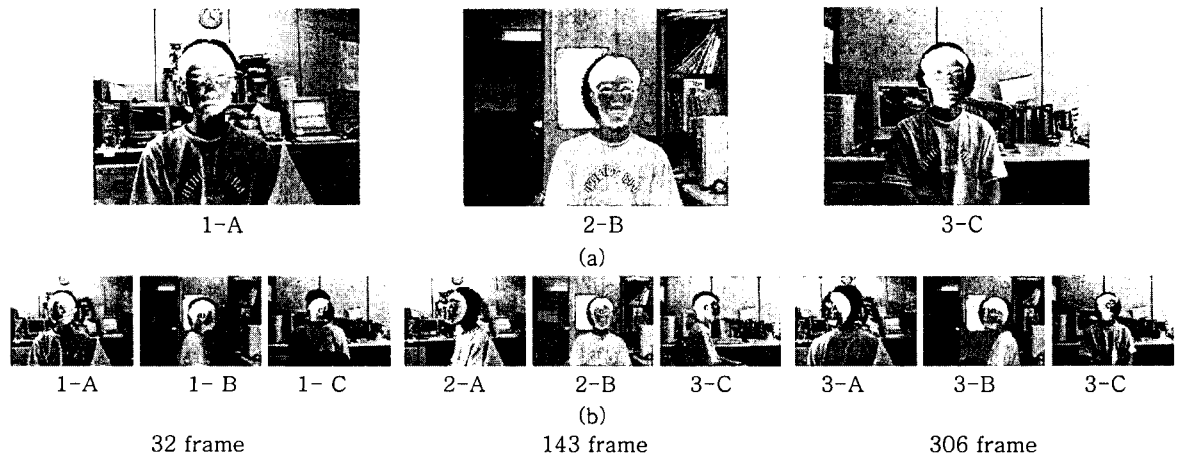


그림 2. 다중 카메라 환경에서의 얼굴 추적 결과 : (a) 제안된 시스템에서의 추적결과 (b) 각 카메라 별 추적 결과