

물체 추적 성능 향상을 위한 추적 실패의 검출 방법

이광무, 최진영

서울대학교 전기컴퓨터 공학부, 서울대학교 자동화시스템공동연구소

Detection of Tracking Failures for Improving Object Tracking Performance

Kwang Moo Yi, Jin Young Choi

Seoul National University Electrical Engineering and Computer Science Department, ASRI

Abstract - Mean Shift 알고리즘은 매우 빠르며 실시간 추적이 가능하다는 장점을 지니지만 추적의 정확도와 관련하여 scale의 변화와 관련된 적응 문제, model의 변화에 대한 적응 문제 등 여러 문제점을 지닌다. 따라서 시스템의 안전성을 보장하기 위해서는 추적 실패를 검출할 수 있는 별도의 검출 방법이 필요하다. 본 연구는 별도의 추가적인 연산 없이 Bhattacharyya coefficient의 변화를 추정하여 물체 추적 실패를 검출하는 방법을 제안한다. 또한 이를 실제 추적 시스템에 구현하여 실험하여 그 성능을 확인하였다.¹⁾

1. 서 론

영상 처리에 있어서 물체 추적은 주요 분야 중 하나이다. 물체 추적은 감시 시스템, 로봇, 등 여러 영상 처리를 요하는 시스템에 사용되며 그 목적 또한 행동 인식, 대상 감시, 등으로 다양하다. 이러한 다양한 목적에 사용되는 물체 추적은 기본적으로 여러 다양한 어플리케이션들이 돌아가는 상황에서 수행되기에 보다 적은 연산 량으로 정확하게 쫓아가는 것이 중요하다.

Mean Shift 알고리즘은 적은 연산 량으로 비교적 정확히 물체를 추적하는 알고리즘으로 이러한 실시간 응용 분야에 적합하다[1]. 하지만 Mean Shift 알고리즘은 추적의 정확도와 관련하여 scale의 변화, model의 변화에 관한 적응에 취약하며 이로 인해 추적 실패가 발생하기도 한다. 특히, global maxima 또는 minima가 아닌 local maxima 또는 minima를 찾아가 엉뚱한 물체를 추적 대상으로 인식하는 문제가 발생하기도 한다. 이 경우 물체 추적 오류는 전체 시스템의 안정성을 훼손시킬 수도 있다. 예를 들어 감시 시스템이 추적 도중에 엉뚱한 물체를 추적 대상으로 인식할 경우 제자리에 멈춰있는 것으로 인식하여 이로 인해 침입자의 추적 실패 또는 엉뚱한 사람을 침입자로 판단하는 등의 문제가 발생할 수 있다. 이러한 문제를 해결하기 위해서는 추적 시스템이 자체적으로 추적의 성공/실패 여부를 판단할 수 있어야 한다.

Bhattacharyya coefficient의 경우 Bayes error와 밀접하게 연관되어 있으며, 따라서 유사도를 나타내는 factor로 보아도 무방하다. 따라서 우리는 본 논문에서 별도의 추가적인 연산 없이 추적 연산 내에서 발생하는 결과들을 이용하여 Bhattacharyya coefficient를 추정하고, 이를 이용해서 물체 추적의 성공/실패 여부를 판단하는 방법을 제안한다. 또한, 이를 실제 Bhattacharyya coefficient를 구한 결과와 비교, 분석하여 제안된 방법의 효용성을 알아보고, 이를 실제 추적 시스템에 구현하여 그 성능을 확인한다.

2. 본 론

2.1 Mean Shift Tracking

Mean Shift Tracking은 Mean Shift 알고리즘을 사용하여 물체를 추적하는 방법이다[1]. 이 방법은 가해진 어떤 한 점에서 출발하여 주어진 샘플 분포의 최대 점을 반복적인 방법을 사용하여 찾아가는 방법으로 매우 빠르고 효율적으로 분포의 최대 점 즉, 추적 대상의 위치를 찾는 방법이다. Mean Shift Tracking은 물체의 전체적인 color histogram을 사용하므로 partial occlusion, motion blur등에 강인한 특징을 지닌다.

일반적인 Mean Shift Tracking은 다음과 같이 수행된다.²⁾ 각각 target model과 target candidate의 m-bin histogram을 \hat{q} 와 $\hat{p}(y)$ 라 하고, target model의 분포 $\{\hat{q}_u\}_{u=1\dots m}$ 과 이전 프레임으로부터의 target model의 추정 위치 \hat{y}_0 가 주어졌을 경우:

1. Target candidate의 위치를 \hat{y}_0 로 초기화, 분포 $\{\hat{p}_u(\hat{y}_0)\}_{u=1\dots m}$ 계산 및 $\rho[\hat{p}(\hat{y}_0), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0)\hat{q}_u}$ 의 계산
2. weight $\{w_i\}_{i=1\dots n_x}$ 의 계산
3. Mean shift vector를 기반으로 target의 새로운 위치 산출,

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_x} x_i w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|\right)}{\sum_{i=1}^{n_x} w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|\right)} \quad (1)$$

1) 이 연구는 산업자원부와 삼성 테크윈의 지원에 의해 연구되었음.
2) [1]의 4.2의 Bhattacharyya Coefficient $\rho[\hat{p}(y), \hat{q}]$ Maximization 알고리즘

4. 만약 $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$ 일 경우 이를 만족하지 않을 때까지 $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_0 + \hat{y}_1)$ 반복.
5. 만약 $\|\hat{y}_1 - \hat{y}_0\| < \epsilon$ 일 경우 멈춤. 아닐 경우 $\hat{y}_0 \leftarrow \hat{y}_1$ 로 설정하고 1 단계로.

여기서 $w_i = \sum_{u=1}^m \delta[b(x_i) - u] \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}}$ 3), $g(\cdot)$ 는 사용된 커널 $K(\cdot)$ 의 프로필을 $k(\cdot)$ 라고 할 경우 $g(\cdot) = -k'(\cdot)$.

위의 알고리즘에서 4의 단계는 거의 일어나지 않으며⁴⁾ 물체의 추적의 성공/실패 여부를 검사할 경우 사실상 불필요한 단계이다. 또한, Bhattacharyya coefficient의 계산은 histogram의 bin의 개수에 따라 상당한 연산을 요구하며 4단계 외의 경우 불필요하다. 따라서 위의 알고리즘은 물체 추적 성공/실패를 검출하는 시스템의 경우 다음과 같이 줄여 연산 량을 감소시킬 수 있다.

1. Target candidate의 위치를 \hat{y}_0 로 초기화 및 weight $\{w_i\}_{i=1\dots n_x}$ 계산.
2. Mean shift vector를 기반으로 target의 새로운 위치 산출
$$\hat{y}_1 = \frac{\sum_{i=1}^{n_x} x_i w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|\right)}{\sum_{i=1}^{n_x} w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|\right)}$$
3. 만약 $\|\hat{y}_1 - \hat{y}_0\| < \epsilon$ 일 경우 멈춤. 아닐 경우 $\hat{y}_0 \leftarrow \hat{y}_1$ 로 설정하고 1 단계로.

2.2 Bhattacharyya Coefficient

추적해야 하는 target model의 색 그리고/또는 texture를 나타내는 특성 z가 density function q_z 를 갖으며, y에 위치한 target candidate가 $p_z(y)$ 에 따라 특성이 분포되어 있을 때, 결국 물체 추적은 $p_z(y)$ 와 q_z 가 가장 유사한 y를 찾는 것과 동일하다. Bhattacharyya coefficient는 Bayes error, 즉 분류 시 잘못 분류할 확률과 밀접하게 연관되어 있으며 따라서 target의 위치를 찾는 문제는 target model과 target candidate의 Bhattacharyya coefficient를 최대화 하는 지점을 찾는 문제와 동일하다. Bhattacharyya coefficient는 일반적으로 다음과 같이 정의 된다.[2]

$$\rho(y) \equiv \rho[p(y), q] = \int \sqrt{p_z(y)q_z} dz \quad (2)$$

Bhattacharyya coefficient의 특성들에 관해서는 [2, 3]에서 다루고 있다. 분포 p와 q를 히스토그램을 사용하여 생성할 경우,

$$\hat{q} = \{\hat{q}_u\}_{u=1\dots m} \quad \left(\sum_{u=1}^m \hat{q}_u = 1\right) \quad (3)$$

$$\hat{p}(y) = \{\hat{p}_u(y)\}_{u=1\dots m} \quad \left(\sum_{u=1}^m \hat{p}_u(y) = 1\right) \quad (4)$$

이며 따라서 Bhattacharyya coefficient의 estimate는 다음과 같이 정의된다.

$$\hat{\rho}(y) \equiv \rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(y)\hat{q}_u} \quad (5)$$

3) 함수 $b: R^2 \rightarrow \{1\dots m\}$ 는 픽셀 위치 x_i 와 픽셀의 색이 히스토그램의 해당하는 위치 $b(x_i)$ 를 연관시켜주는 함수
4) [1]에 의하면 0.1% 미만

2.3 Failure Detection

물체 추적의 성공/실패 검출은 현재 갖고 있는 target model과 target candidate 간의 유사도를 비교하여 수행할 수 있다. 이들의 유사도를 검출하는 방법은 여러 가지가 있지만 Bhattacharyya coefficient의 값을 봄으로써 간단히 비교할 수 있다. Bhattacharyya coefficient는 Bayes error와 밀접한 관련이 있으며 따라서 물체의 유사도를 나타내는 척도로 사용할 수 있다.

현재 프레임에서의 target의 추정 위치를 \hat{y}_0 라 할 때, Bhattacharyya coefficient는 $\hat{p}_u(\hat{y}_0)$ 근처에서 Taylor expansion을 통해 근사화하면 다음과 같이 주어진다.[1]

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} + \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k \left(\left\| \frac{y - x_i}{h} \right\|^2 \right) \quad (6)$$

(6)에서 우변의 좌항의 경우 y에 관하여 변하지 않는 상수이므로 다음의 우항만을 고려해도 $\rho[\hat{p}(y), \hat{q}]$ 의 변화를 관찰할 수 있다.

$$\frac{C_h}{2} \sum_{i=1}^{n_h} w_i k \left(\left\| \frac{y - x_i}{h} \right\|^2 \right) \quad (7)$$

최종적으로 target을 잡았을 경우 위의 (7)에서 각 i에 대하여 $k \left(\left\| \frac{y - x_i}{h} \right\|^2 \right)$ 의 값은 각 x_i 의 y에 상대적인 위치에 따라 결정된다. 즉, $k \left(\left\| \frac{y - x_i}{h} \right\|^2 \right)$ 의 값은 윈도우 내에서 각 x_i 의 w_i 가 끼치는 영향을 나타낼 뿐이다. 따라서 $k \left(\left\| \frac{y - x_i}{h} \right\|^2 \right)$ 는 y의 변화와는 상관이 없는 값이다. 즉, (7)의 값은 w_i 와의 연관성이 강하며, w_i 의 변화를 관찰하는 것으로 전체 Bhattacharyya coefficient의 값의 변화를 추정하는 것이 가능하다.

전체적인 w_i 의 변화를 관찰하기 위하여 w_i 의 평균값 사용하였다. 평균값을 사용함으로써 다음의 장점들을 갖는다.

1. 영상 전체의 정보가 고르게 반영된다.
2. Epanechnikov kernel[4, p. 139]을 사용할 경우 (1)에서 g가 상수 값을 가지므로 $\sum_{i=1}^{n_h} w_i$ 를 추적 알고리즘을 수행하는 동안 필연적으로 계산하므로 별도의 연산 량 증가가 거의 없다.

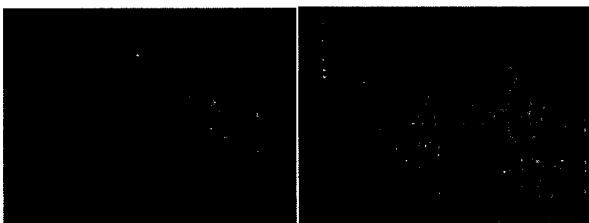
이 weight는 Bhattacharyya coefficient의 크기를 결정하므로 Bayes error와 관계된 factor로서 실질적으로 물체 간의 유사도의 변화를 나타내는 인자로서 활용할 수 있다. 또한, 히스토그램을 이용하여 얻어지므로 물체의 유형에 관계없이 사용이 가능하며 이미지의 축척에 영향을 받지 않는다는 장점을 지닌다. 이 weight를 이용하여 추적 성공/실패를 검출하는 방법은 다음의 순서로 수행된다.

처음 시작할 때의 target model의 w_i 의 평균값 $w_{avg,0} = 1$ 이므로:

1. 최종적으로 target으로 잡은 부분의 w_i 의 평균 값 $w_{avg,1}$ 계산.
2. $likeliness \leftarrow |1 - w_{avg,1}|$
3. $likeliness < threshold$ 이면 추적 실패
 $likeliness \geq threshold$ 이면 추적 성공.

3. 결 과

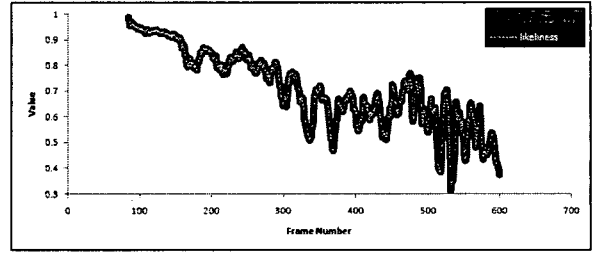
본 논문에서는 실제로 Bhattacharyya coefficient를 계산하여 물체 추적을 수행하였을 때와 이를 계산하지 않고 제안된 weight들의 sum을 이용하여 Bhattacharyya coefficient의 변화를 추정하였을 때의 연산시간의 차이, 추정의 정확도를 과격한 랜덤 움직임을 보이는 사람의 얼굴을 대상으로 하여 실험하였다. <그림 1>은 실제 추적 결과로 추적 대상(사람)이 추적 화면을 벗어나 추적 대상을 놓친 경우이다.



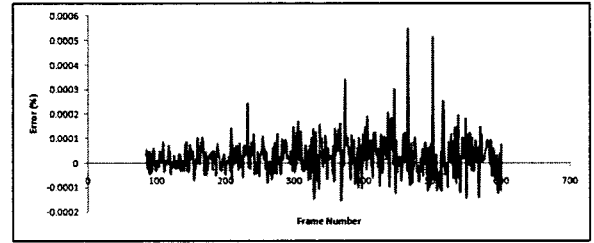
<그림 1> 실제 Tracking 구현 예
좌측 - 물체 추적 성공 예, 우측 - 물체 추적 실패 예

3.1 추정의 정확도

<그림 2>과 <그림 3>은 각각 실제 Bhattacharyya coefficient와 제안된 weight들의 sum을 사용하여 추정된 값과 그 둘 사이의 오차이다.



<그림 2> Bhattacharyya coefficient와 weight를 이용한 추정 값(likeliness)의 비교



<그림 3> Bhattacharyya coefficient와 likelihood 사이의 오차(%)

위의 <그림 2>와 <그림 3>으로부터 매우 적은 오차를 갖는 것을 볼 수 있다. 이로부터 본 제안된 방법이 효율적으로 Bhattacharyya coefficient를 추정하고 있음을 확인할 수 있다.

3.2 추정으로 인한 속도 향상

다음의 <표 1>은 Bhattacharyya coefficient를 추정하였을 때와 추정하지 않았을 경우의 연산시간의 차이이다.

<표 1> Histogram Dimension에 따른 평균 연산 시간

histogram dimension	Bhattacharyya coefficient를 계산한 경우 (ms)	Bhattacharyya coefficient를 계산하지 않은 경우 (ms)	연산 시간 차이 (%)
10 × 10 × 10	7.598456	7.029014	7.494182
16 × 16 × 16	6.804642	5.901354	13.27459
24 × 24 × 24	8.675048	7.170213	17.34671

OS: Windows Vista(TM) Business K, CPU: Intel® Core(TM) 2 T7200 2.0Ghz, RAM: 2.0GB

평균 연산 시간을 보면 histogram dimension이 클수록 Bhattacharyya coefficient를 구하는데 필요한 연산 시간이 크고 따라서 dimension이 클수록 보다 많은 연산 시간이 단축되는 것을 볼 수 있다. 추적 시스템은 적은 부하를 갖고 수행될수록 좋다는 점을 감안하면 이는 시스템을 보다 효율적으로 개선하였다 할 수 있다.

4. 결 론

본 논문에서는 Mean Shift Tracking시의 연산 과정에서 사용되는 weight 값들의 변화를 사용하여 Bhattacharyya coefficient를 추정하고 이를 이용하여 물체 추적의 성공/실패 여부를 검사하는 방법을 제안하였다. 연산 과정에서 계산하는 값을 이용함으로써 별도의 추가적인 연산 없이 추정의 안정성을 높일 수 있었으며, 그 효율성을 실험적으로 확인하였다. 별도의 연산 량 증가 없이 매우 정확한 추정이 가능하였으며 이를 이용하여 실제 시스템에 구현하여 활용이 가능함을 확인할 수 있었다. 추후의 연구 과제로는 이를 보다 범용적으로 개선하는 것과 histogram외의 방법을 사용하는 물체 추적으로 이를 확장하는 것이 있다.

[참 고 문 헌]

- [1] D. Comaniciu, V. Ramesh, P. Meer "Real-Time Tracking of Non-Rigid Objects using Mean Shift", BEST PAPER AWARD, IEEE Conf. Computer Vision and Pattern Recognition (CVPR'00), Hilton Head Island, South Carolina, Vol. 2, 142-149, 2000
- [2] A. Djouadi, O. Snorrason, F.D. Garber, "The Quality of Training-Sample Estimates of the Bhattacharyya Coefficient", IEEE Trans. Pattern Analysis Machine Intell., 12:92-97, 1990.
- [3] T. Kailath, "The Divergence and Bhattacharyya Distance Measures in Signal Selection", IEEE Trans. Commun. Tech., COM-15:52-60, 1967.
- [4] D.W. Scott, "Multivariate Density Estimation", New York: Wiley, 1992.