
Hue/Saturation 영상의 적응적 선택을 이용한 강인한 Mean-Shift Tracking

박한동, 오정수
부경대학교 융합디스플레이공학과

Robust Mean-Shift Tracking Using Adoptive Selection of Hue/Saturation

Han-dong Park, Jeong-su Oh
Department of Display Engineering, Pukyong National University
{esterlla, ojs}@pknu.ac.kr

요 약

Mean-Shift 알고리즘은 객체 모델과 객체 후보 영상에서 색상 히스토그램 분포의 유사도를 이용하여 객체를 추적하는 강인한 알고리즘이다. 그러나 색상정보를 이용한 Mean-Shift 알고리즘은 객체와 배경이 비슷한 색상 분포를 가질 경우에 추적에 실패할 수 있는 단점이 있다. 이러한 단점을 보완하기 위해 배경과 객체를 분리할 정보를 색상(hue)과 채도(saturation) 영상에서 각각 4비트의 bit-plane을 조합한 새로운 영상을 사용한 강인한 객체 추적 알고리즘을 구현한다.

ABSTRACT

The Mean-Shift is a robustness algorithm that can be used for tracking the object using the similarity of histogram distributions of target model and target candidate. However, Mean-shift using hue information has disadvantage of tracking a wrong target when the target and background has similar hue distributions. We then propose a robust Mean-Shift tracking algorithm using new image that combined upper 4bit-planes in hue and saturation, respectively.

키워드

Mean-Shift, Tracking, Hue, Saturation, HSV

I. 서 론

컴퓨터 비전 분야 중 객체를 추적하기 위하여 많은 알고리즘이 제안되고 있다. 그 중 Mean-Shift 추적 알고리즘은 가장 많이 사용되는 방법으로 객체 모델과 객체 후보 영역에서 히스토그램 분포의 유사도를 평가하여 그 히스토그램 분포가 가장 유사한 위치를 반복적으로 찾는 nonparametric 알고리즘이다[1]. 일반적인 Mean-Shift 알고리즘에서 사용하는 객체의 정보는 색상정보이다. 영상을 표현할 때 컬러 영상인 경우 보통 적(red), 녹(green), 청(blue) 각각 8비트로 구성되어 있다. 즉, 컬러정보를 모두 사용

하기 위해서는 24비트의 정보가 필요한데 이는 비효율적인 방법이다. 그러나 색상(hue), 채도(saturation), 명도(value)로 표현되는 HSV영상으로 변환하여 색상 영상만을 이용하여 8비트로 색상 정보를 얻을 수 있다. 색상 영상을 이용하면 영상은 채도, 명도 값을 배제하여 조명 등에 대한 영향을 줄일 수 있다.

하지만 색상 영상의 값으로 Mean-Shift 추적을 할 때, 배경의 색상과 객체의 색상이 유사한 경우에는 추적에 실패할 가능성이 있다. 본 논문에서는 색상의 유사성에 의한 실패 가능성을 줄이기 위해 색상 정보와 채도 정보를 조합하여 새로운 영상을 만들고 그 영상을 이용하여

객체 추적을 할 것이다. 그 조합 방법으로 8비트로 이루어진 색상, 채도 영상에서 각각 상위 4개의 비트를 조합하여 새로운 8비트 영상을 만드는 것이다. 8비트 영상의 bit-plane에서는 상위 4개의 비트가 영상과 가장 흡사한, 즉 중요도가 높은 비트라 할 수 있는데 색상, 채도의 조합 순서에 따라 색상 위주의 영상이 될 수도 있고, 채도 위주의 영상이 될 수도 있다. 이 조합 순서를 판명하기 위해서 영상 내에서 객체로 지정된 영역과 그 영역 외부에 위치한 배경 영역에서 색상, 채도 값을 구한 후 객체와 배경 영역의 차이 비율을 통해서 결정하게 된다.

제안된 알고리즘은 객체와 배경간의 색상 유사도가 큰 영상에 대해서도 추적이 가능하게 하고 기존에 추적이 가능했던 영상 역시 추적이 가능하다는 것을 보인다.

II. 본 론

2-1 Mean-Shift 추적 알고리즘

Mean-Shift 추적 알고리즘은 객체를 추적하는 알고리즘 중에서 가장 많이 사용되는 것으로써 관심영역 내에 데이터 분포가 가장 밀집된 최대 빈도점(peak point)을 찾아 관심영역 중심을 최대 빈도점으로 이동시키는 알고리즘으로 주로 히스토그램 역투영(histogram-backprojection)과 함께 사용된다. 객체(q)와 객체 후보 영역(p)의 히스토그램 빈 값(bin value)은 식 (1), 식 (2)를 이용해 계산한다[1-4].

$$q_u = C_q \sum_{i=1}^n k \left(\left\| \frac{x_i}{h_y} \right\|^2 \right) \delta(b(x_i) - u) \quad (1)$$

$$p_u = C_p \sum_{i=1}^{n_h} k \left(\left\| \frac{y - x_i}{h_p} \right\|^2 \right) \delta(b(x_i) - u) \quad (2)$$

여기서 히스토그램 q 와 p 는 각각 $\{q_u\}_{u=1 \sim m}$ 와 $\{p_u(y)\}_{u=1 \sim m}$ 이다. $b(x_i)$ 는 화소값이고 C 는 정규화를 위한 상수이다. K 는 관심영역의 중심에서 멀어질수록 가중치를 줄이는 커널 함수(kernel function)로 일반적으로 Mean-Shift에서 Epanechnikov kernel을 사용한다. δ 는 Kronecker delta function, $\delta(a) = \begin{cases} 1 & \text{if } a=0 \\ 0 & \text{otherwise} \end{cases}$ 이다. 위의 과정으로 입력된 영상은 다음과 같은 히스토그램의 비를 가진다[1-4].

$$w_i = \sum_{u=1}^m \left(\delta(b(y_i) - u) \sqrt{\frac{q_u}{p_u(y_0)}} \right) \quad (3)$$

여기서 y_0 은 객체 후보 영역에서 표적의 위치이고 이 식을 이용하여 Mean-Shift는 식 (4)와 같이 객체 후보 영역에서 객체 히스토그램 빈이 가장 많이 밀집된 위치(y_i)를 찾아간다[1-4].

$$y_i = \frac{\sum_{i=1}^n x_i w_i g \left(\left\| \frac{y_0 - x_i}{h_q} \right\|^2 \right)}{\sum_{i=1}^{n_i} w_i g \left(\left\| \frac{y_0 - x_i}{h_p} \right\|^2 \right)} \quad (4)$$

여기서 h 와 x_i 는 객체 후보 영역의 크기와 위치이고 g 는 앞서 언급한 k 의 shadow kernel 함수이다. 두 영역의 색상 분포의 유사도를 측정하기 위하여 Bhattacharyya 계수 ρ 를 사용하며 식 (5)와 같이 표현된다[1-4].

$$\rho(y) = \rho[q,p] = \cos(\theta) = \frac{\sum_{u=1}^m \sqrt{p_u(y)q_u}}{\sqrt{\sum_{u=1}^m p_u(y)q_u}} \quad (5)$$

색상을 통한 객체추적을 하게 되면 그림 1과 같이 배경과 구분되는 객체에 의해서 추적이 가능하게 된다. 하지만 그림 2의 영상처럼 객체와 배경의 색상 값이 유사할 경우에는 객체 추적에 실패할 가능성이 있다.

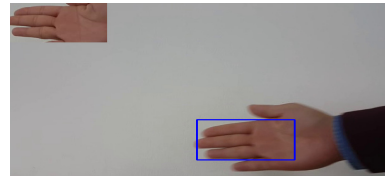


그림 1. 적절한 Mean-Shift Tracking

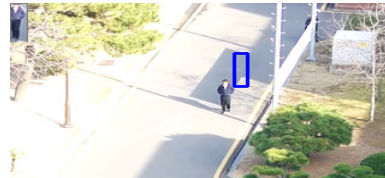


그림 2. 잘못된 Mean-Shift Tracking

그림 2에서 초기에 선택된 객체와 객체가 지나가는 배경영역에서 색상이 유사하다는 것을 색상 영상을 통해 알 수가 있는데 그림 3은 그림 2의 색상 값으로 구성된 8비트 영상으로 색상의 범위인 0~360°를 0~255로 정규화 시켜 나타내었다. 색상이 유사한 부분은 1채널 영상에서 밝기가 유사하게 나타나는 것을 볼 수 있다.



그림 3. 색상 값으로 구성된 영상

2-2 제안하는 알고리즘

그림 2, 그림 3의 경우 영상의 특성상 채도를 이용하면 더 효과적인 추적이 가능하다. 그림 4는 그림 3과 같은 프레임에서의 채도 값으로 구성된 영상으로 그림 3에 비해 객체, 배경간의 차이가 많이 나는 것을 볼 수 있다.



그림 4. 채도 값으로 구성된 영상

결과적으로 영상에 따라 색상 영상으로 추적하는 것이 효과적일 수도 있고, 채도 영상으로 추적하는 것이 더 효과적일 수도 있다. 또한 같은 영상 내에서도 프레임마다 그 환경이 달라 질 수도 있다. 따라서 환경에 따라 알맞은 영상에서 추적할 수 있는 기법이 필요하다.

그림 5는 추적영상 내에 객체 영역과 그 영역 주위로 크기가 2배가 되는 영역을 배경영역으로 설정하는 것을 나타내었다.

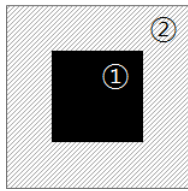


그림 5. 객체(①)와 배경(②)영역

위와 같이 분리된 객체와 배경영역에서 각각의 색상, 채도의 평균값을 구하여 배경에 대한 (배경-객체)의 비율을 식(6), 식(7)으로 계산한다.

$$Rate_H = \frac{|\overline{H_{Bg}} - \overline{H_{Ob}}|}{\overline{H_{Bg}}} \quad (6)$$

$$Rate_S = \frac{|\overline{S_{Bg}} - \overline{S_{Ob}}|}{\overline{S_{Bg}}} \quad (7)$$

여기서 식에서 \overline{H} 는 색상 값의 평균, \overline{S} 는 채도 값의 평균을 의미한다. 만약에 배경과 객체의 평균값이 차이가 많은 경우에는 Rate 값이 1 근처이거나 그 이상의 값을 가지게 되고 평균값의 차이가 적은 경우에 Rate값은 0 근처가 될 것이다. 이를 식(8)로 정리하였다.

$$\begin{cases} Rate \approx 0 & \text{if } bg \approx ob \\ Rate \approx 1 & \text{if } bg \gg ob \\ Rate \gg 1 & \text{if } bg \ll ob \end{cases} \quad (8)$$

식(8)을 근거로 Rate가 0.5 이상일 때, 배경의 평균값과 객체의 평균값의 차이가 많이 나는 것이

라 가정하고 $Rate_H$ 와 $Rate_S$ 의 값의 결과에 따라 표 1처럼 객체 추적을 할 영상을 선택하도록 하였다.

표 1. Rate 값에 따른 추적 영상 선택

	$Rate_S \geq 0.5$	$Rate_S < 0.5$
$Rate_H \geq 0.5$	색 상*	색 상
$Rate_H < 0.5$	채 도	색 상*

표 1에서 색상과 채도의 값이 비슷할 경우에는 색상을 선택(*)하도록 하였는데 각 영상의 특성상 색상영역에서 히스토그램이 채도 영역에서의 히스토그램보다 구분이 용이하기 때문이다.

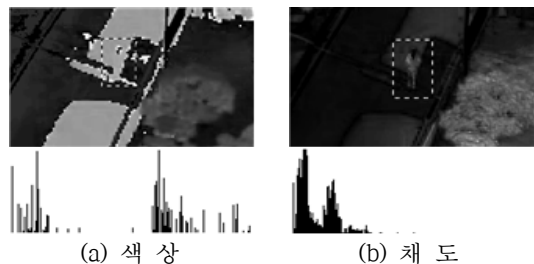


그림 6. 영상과 히스토그램

그림 (a), (b)는 Rate가 비슷한 프레임에서 각각 색상 영상과 채도 영상 및 각각의 히스토그램을 나타내었다. 색상에서 히스토그램의 표준편차 값은 약 72였고 채도에서의 표준편차는 약 18의 값을 나타내었다. 각 영상에서의 히스토그램을 비교하였을 때, 채도에 비해 색상 영역에서 히스토그램을 2개(객체와 배경)로 구분하기가 용이할 것이라고 예상할 수 있다.

이러한 기준을 통해 결정된 추적 영상은 주 추적 영상과 보조 추적 영상으로 나눌 수 있다. 예를 들어 영상 전체적으로 $Rate_H$ 가 강하다면 주 추적 영상은 색상 영상이 되고 채도 영상은 $Rate_H$ 가 일시적으로 감소되는 프레임에서 계속 추적이 가능하게 하는 보조 추적 영상이 된다. 대부분의 영상에서 보조 추적 영상은 객체 추적에 실패할 가능성이 있다. 그 이유는 보조 추적 영상에서의 객체와 배경의 차이가 주 추적 영상보다는 크지가 않기 때문이다. 이를 보완하기 위하여 색상, 채도 영상을 그대로 사용하지 않고 각각의 영상에서 상위 4개의 비트를 참고하여 새로운 8비트의 영상을 만들어서 사용하게 된다. 상위 4개의 비트를 사용하는 이유는 Bit-Plane에서 상위 4개의 비트($2^7 \sim 2^4$)가 원래의 8비트 영상과 흡사한 모습, 즉 중요도가 높은 비트이기 때문이다. 같은 이유로 새로운 영상으로 조합을 할 때, 비트 배치순서 또한 중요하게 되는데 색상이 위주로 되어야 하는 영상(표 1의 색상 선택 부분)은 그림 7처럼 색상 영상의 4개의 비트가

새로운 영상의 상위 4의 비트로 배치가 되어야 한다.

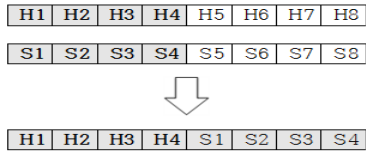


그림 7. 색상, 채도 조합 영상의 비트

이렇게 조합된 영상은 기존의 색상 영상이나 채도 영상에 비해 배경과 객체를 분리하기에 유리하다는 것을 그림 8을 통해서 알 수 있다.

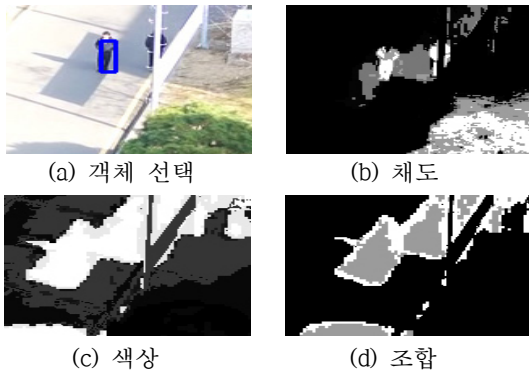


그림 8. 객체 역투영 영상

그림 8은 색상, 채도 및 두 조합 영상에서 배경에 선택된 객체를 투영한 결과 영상이다. 이 영상에서 주 추적 영상은 채도 영상이며 (b)처럼 배경에 비해 객체가 높은 값을 가지는 것을 볼 수 있다. 하지만 보조 추적 영상인 (c) 색상 영상에서는 배경을 객체에 역투영 했을 때 객체와 그 주변이 유사한 값을 보이는 것을 알 수 있다. 하지만 보조 추적 영상을 색상, 채도 조합 영상을 사용하였을 경우에는 (d)처럼 객체부분이 구분되어 높은 값을 가지게 되고 결과적으로 배경과 객체가 분리가 용이하여 추적알고리즘에 더 효율적이다.

2-3 실험결과

$Rate_H$ 와 $Rate_S$ 를 이용하여 적응적으로 추적영상을 선택하고 색상과 채도 값의 조합으로 새로운 영상을 만들어서 기존의 Mean-Shift 영상의 단점을 보완하였다. 그림 9는 기존의 Mean-Shift 알고리즘을 통한 방법과 제안된 기법을 이용한 Mean-Shift 알고리즘에서의 객체 추적 결과를 보여준다. 배경과 객체간의 색상 값 정보가 유사한 부분에서 기존의 Mean-Shift 추적 알고리즘은 그림 9의 진한 테두리처럼 추적에 실패하는 모습을 보이고 있지만 채도가 개입된 제안된 알고리즘에서는 그림 9의 얇은 테두리처럼 추적에 실패하지 않는 결과를 보여주고 있다.

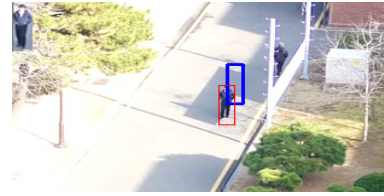


그림 9. 추적 결과의 비교

III. 결 론

본 논문에서는 Mean-Shift 추적 알고리즘에 사용되는 영상을 색상과 채도의 조합을 통해 새로운 영상을 만들어서 사용하였고 객체와 배경의 색상 및 채도의 차이의 비율을 통해서 색상과 채도의 비트 조합방법을 결정하여 주 추적 영상과 보조 추적 영상으로 나누어 프레임마다 알맞은 영상을 선택하여 추적이 이루어지도록 하였다. 기존의 Mean-Shift 추적 알고리즘에서 객체, 배경간의 컬러 정보가 유사하여 추적에 실패하였던 영상에 대한 추적을 제안된 알고리즘에서는 채도 정보의 개입으로 인해 추적에 성공하는 모습을 볼 수 있었고 색상 정보만으로 충분히 기존 알고리즘에서 추적이 가능했던 영상의 경우에도 개선된 알고리즘에서 여전히 추적에 성공하는 결과를 보였다.

참고문헌

[1] N. S. Peng and J. Yang “Mean-Shift Blob Tracking with Kernel-Color Distribution Estimate and Adaptive Model Update Criterion,” *Lecture Notes in Computer Science*, Vol. - No.3247, 2004

[2] D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, VOL.24, No.5, MAY 2002.

[3] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-Based Object Tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, May 2003.

[4] J. Ning, L. Zhang, D. Zhang, and C. Wu, “Robust Mean Shift Tracking with Corrected Background-Weighted Histogram,” *Published in IET Computer Vision*, Revised 27th June 2010.