

영역 궤적의 클러스터링을 이용한 비디오 영상에서의 움직이는 객체의 검출

권 영진^o, 이재호, 김회율

한양대학교 전자통신전파공학과

Moving Object Segmentation Using the Clustering of Region Trajectories

Young-Jin Kwon^o, Jae-Ho Lee, Whoi-Yul Kim

Division of Electrical and Computer Engineering, Hanyang University

yjkwon@vision.hanyang.ac.kr

요약

동영상에서 움직이는 객체 검출은 동영상의 내용을 표현하고 유사한 동영상을 검색하는 데 있어 중요한 특징 값을 추출하는 방법으로 사용된다. 그러나 복잡하게 카메라가 움직이는 동영상에서 움직이는 객체 검출은 아직까지 어려운 과제이다. 본 논문에서는 복잡한 카메라의 움직임이 있는 환경에서 움직이는 객체를 강인하게 검출하는 방법을 제안한다. 움직이는 객체 검출 방법은 입력 영상을 색상간의 클러스터링을 이용하여 각 영역으로 구분하는 Mean Shift 알고리즘과 인접한 프레임에서 구분된 영역을 대응시켜 영역의 모션 벡터를 구하는 영역 매칭, 유사한 궤적을 가지는 영역들의 클러스터링을 이용하여 객체를 검출하는 궤적 클러스터링 알고리즘을 사용한다. 제안한 영역 기반 알고리즘은 기존의 픽셀이나 블록 기반의 방법보다 움직이는 객체를 정확하게 검출하였다. 실험 결과 복잡하게 움직이는 카메라의 환경 속에서 움직이는 객체를 강인하게 검출하였다.

1. 서 론

멀티미디어와 네트워크의 발달로 텍스트와 영상 중심의 정보 교류에서 이제는 동영상이 중요한 정보 데이터로 사용되고 있다. 많은 동영상 데이터를 효율적으로 관리하기 위해서 움직이는 객체를 특징값으로 하는 동영상 인덱싱과 검색기능이 필요하다. 따라서 동영상에서 움직이는 객체를 검출하는 방법은 중요한 과제로 대두되고 있으며, 현재 다양한 연구가 진행중이다. Optical Flow를 이용한 움직이는 객체 검출 방법[1]은 인접하는 프레임을 픽셀 기반으로 매칭하여 Optical Flow를 계산하고, 계산된 flow의 히스토그램을 이용하여 global motion을 구하고 객체를 검출하는 기술로 잡음에 민감하고 연산량

이 많다는 단점이 있다. 움직이는 객체 검출을 위한 또 다른 방법으로 블록을 기반으로 매칭하여 모션벡터를 계산하는 블록 매칭 알고리즘이 제안되었다.[2] 이 방법은 블록 단위로 매칭하기 때문에 객체의 정확한 모양을 검출하기가 힘들고, 균일한 색상을 가진 영역에서는 부정확한 매칭 결과를 가진다. 이러한 단점을 개선하기 위해 본 논문에서는 칼라와 공간적 인접성을 고려하여 구분된 영역을 기반으로 객체를 검출하는 방법을 제안한다.

Mean Shift 알고리즘[3]을 사용하여 입력 영상을 균일한 색상을 가지는 영역으로 구분 짓고, 구분된 영역의 색상, 위치, 면적 등의 특징값을 이용하여 연속되는 프레임 사이에서 영역 매칭을 한다. 매칭된 영역들의 궤적을 계산하고, 유사한 궤적을 가진 영역들을 클러스터링하여 최종적으로 움직이는 객체를 검출한다.

2. 움직이는 객체 검출

본 논문에서 제안한 방법은 동영상에서 움직이는 객체를 검출하기 위해 Mean shift 알고리즘, 영역 매칭, 궤적 클러스터링의 절차를 가지며, 그림 1에서는 움직이는 객체 검출 방법의 대략적인 순서도를 보여준다.

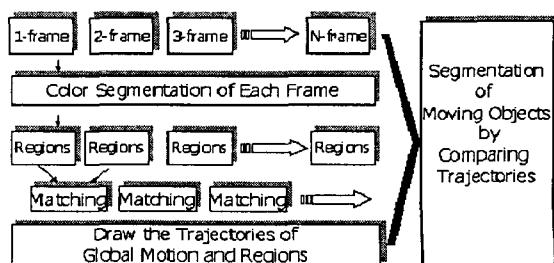


그림 1. 움직이는 객체 검출 과정

움직이는 객체 검출 과정은 다음과 같다.

1. 각 프레임을 대표 칼라들로 이루어진 영상으로 만든다.
2. 레이블링을 통해 각 영역을 구분짓고, 이 영역의 특징값(칼라, 무게중심, 면적 등)을 추출한다.
3. 인접하는 프레임에서 영역들의 특징값을 고려하여 합당한 영역끼리 매칭한다.
4. 매칭된 영역의 모션 벡터를 구하고, 모션 벡터들의 합을 이용하여 global motion 을 계산한다.
5. 1-4 과정을 동영상의 모든 프레임에 적용한다.
6. 매칭된 영역을 리스트로 만들어 그 영역의 궤적을 계산한다.
7. Global motion 과의 차이가 큰 영역 리스트를 추출하고, 이를 간의 클러스터링을 이용하여 움직이는 객체를 검출한다.

2.1 칼라 세그먼테이션 (Color Segmentation)

영역 기반 매칭을 통한 객체 검출을 하기 위해서 움직이는 객체 검출의 첫 단계로, 입력된 영상을 비슷한 색상으로 이루어진 영역들로 구분짓는 칼라 세그먼트 과정이 필요하다. 본 논문에서는 Mean Shift 알고리즘을 사용하여 다양한 색상으로 이루어진 영상을 공간과 색상의 특성을 고려하여 하나의 일관된 영역으로 구분짓는 칼라 세그먼테이션을 하였다. 인간의 시각적인 특성에 근접한 LUV 칼라 공간에서 입력 영상의 히스토그램을 계산하였다. 일정한 반경을 가진 구 모양의 원도우(S_h)안에 포함된 히스토그램들과 원도우 중심간의 차를 계산하고, 그 차가 가리키는 방향 벡터를 구하였다.

$$M_h(x) = \frac{1}{n_x} \sum_{x_i \in S_h(x)} [X_i - X] \quad (1)$$

n_x	S_h 안에 포함된 히스토그램의 합
S_h	일정한 반경을 가진 구 모양의 원도우
X_i	히스토그램
X	S_h 의 중심

식(1)의 $M_h(x)$ 를 Mean Shift Vector 라고 부르며, 항상 밀도가 최대로 증가하는 방향을 가리키고 있다. 이 벡터가 가리키는 방향으로 원도우를 이동하면 local density maximum 에 수렴해 간다.

Mean Shift 알고리즘에 절차는 다음과 같다.

1. Mean Shift Vector(M_h)를 계산한다.
2. M_h 방향으로 원도우(S_h)를 이동한다.
3. 1-2 과정을 반복하여, 수렴하는 local density maximum 을 찾는다.
4. 1-3 과정을 반복하여, 영상에서 존재하는 local density maximum 을 모두 계산한다.
5. 위에서 계산된 local density maximum 을 이용하여, 나머지 색상 영역은 제일 가까운 local density maximum 값으로 클러스터링 한다.

그림 2 는 1-2 과정을 통하여 local density maximum 에 이르는 경로를 보여준다. 최초 원의 중심(mean)은 밀도가 낮은 위치에 있지만, Mean Shift Vector 를 이용하여 local density maximum 으로 수렴하는 것을 보여준다.

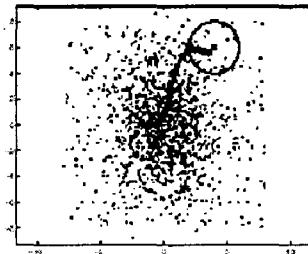
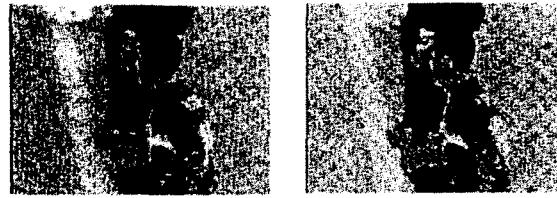


그림 2. Local density maximum에 이르는 경로

그림 3은 Mean Shift 알고리즘을 이용하여 비슷한 색상의 영역을 대표 칼라로 표현한 칼라 세그먼트된 영상을 보여준다. 그림 3 의 (a)는 다양한 색상으로 이루어진 원 영상이고, 그림 3 의 (b)는 Mean Shift 알고리즘을 사용하여 칼라 세그먼트된 영상이다.



(a) 원 영상

(b) 칼라 세그먼트된 영상

그림 3. Mean Shift 알고리즘을 사용한 결과

칼라 세그먼트된 영상에서 레이블링을 하여 하나의 색상으로 이루어진 영역을 구분하고, 각 영역의 색상, 면적, 무게중심, 외접 사각형 등의 특징값을 추출한다.

2.2 영역 매칭(Region Matching)

Mean Shift 알고리즘으로 구분된 영역들의 모션 벡터를 계산하고, 이를 이용하여 global motion 을 구하기 위해서 영역 매칭 과정이 필요하다. 연속되는 프레임 사이에 구분된 영역들의 레이블링 정보를 이용하여 영역 매칭을 한다.

$$Dist(i, j) = \sum_{n=0}^N w_n * (f_i - f_j)^2 \quad (2)$$

i	과거 프레임의 i 번째 영역
j	현재 프레임의 j 번째 영역
n	특징값의 번호
w	각 특징값에 대한 가중치
f	특징값

식(2)는 인접한 프레임사이에서 영역간의 차이를 정의하는 합수이다. 특징값으로 LUV 색상, 외접 사각형의 중심, 면적을 사용하여 영역간의 차이를 나타내는 합수를 정의한다. 연속되는 프레임 사이에서 식(2)에 의한 차이가 가장 작은 영역을 동일 영역으로 판단한다. 전체 프레임에 대하여 영역 매칭을 수행하고 이 매칭된 정보를 사용하여 특정 영역이 연결되는 리스트를 만들 수 있다. 이

것을 영역 리스트라고 부르며 이를 이용하여 영역의 움직임을 계산한다. 계산된 움직이는 객체를 검출하는데 중요한 특징값으로 사용된다. 그림 4는 인접한 프레임에서 영역 매칭의 결과를 보여준다.

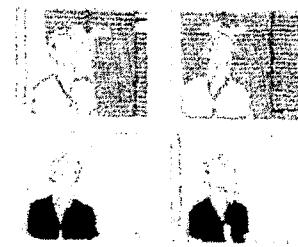


그림 4. 인접 프레임의 영역 매칭 결과

2.3 궤적을 이용한 움직이는 객체 검출

최종적으로 움직이는 객체 검출은 각 영역 리스트의 궤적을 비교하여 유사한 궤적들의 클러스터링을 통해 이루어진다. 영역 리스트의 궤적을 이용한 움직이는 객체 검출 과정은 다음과 같다.

1. 한 프레임에서 전체 영역들의 모션 벡터를 이용하여 global motion 계산
2. Global motion 과 궤적의 차이가 큰 영역 리스트 추출
3. 추출된 영역 리스트간의 클러스터링을 이용하여 움직이는 객체 검출

영역의 모션 벡터는 global motion 을 구하거나 객체 검출을 할 때 중요한 특징값이다. 영역의 모션 벡터를 정확히 계산하기 위해서 4 개의 점(영역의 시작점, 영역의 무게중심, 외접 사각형의 최소점, 외접 사각형의 최대점)을 모두 고려한다. 각 점들의 움직임을 구하고 이 중에서 차이가 많이 나는 한 점의 움직임만을 제외한 나머지의 평균을 구함으로써, 인접한 영역이 부분적으로 가리는 경우에도 움직임을 안정적으로 계산하도록 한다. 그림 5는 위에서 언급한 4 개의 점을 보여준다.

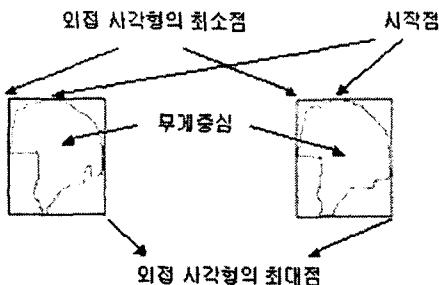


그림 5. 모션 벡터를 계산하기 위한 영역의 4 개의 점

Global motion 은 한 프레임에서 계산된 모든 모션 벡터를 합한 평균을 계산하고, 이 평균값과 차이가 많이 나

는 모션 벡터를 제외한 나머지의 평균을 계산함으로써 얻을 수 있다. 그림 6은 비디오 1 과 비디오 3 의 global motion 을 보여준다. 그림 6 의 (a)는 비디오 1 에서 오른쪽으로 움직이는 카메라를 잘 표현하고 있으며, 그림 6 의 (b)는 비디오 3 에서 카메라가 왼쪽으로 움직이며, 후반부에는 아래쪽으로 움직이는 것을 잘 표현하고 있다.

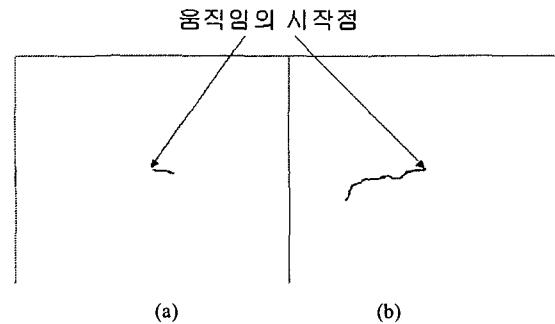


그림 6. 모션 벡터를 이용한 Global motion

움직이는 객체를 구성하는 영역 리스트를 추출하기 위해 global motion 과 영역 리스트의 궤적을 비교하여 차이가 큰 영역 리스트를 추출한다.

$$Dist(i) = \sum_{n=s}^f abs(\theta_i - \theta_G) * (m_{x_i} - m_{x_G})^2 * (m_{y_i} - m_{y_G})^2 \quad (3)$$

s	영역 리스트의 시작 프레임
f	영역 리스트의 마지막 프레임
i	영역 리스트의 번호
G	Global motion 을 나타내는 인덱스
x	x 방향을 표시하는 인덱스
y	y 방향을 표시하는 인덱스
θ	모션의 방향, 즉 $\tan^{-1} \frac{m_y}{m_x}$
m	모션의 정도

식(3)은 global motion 과 영역 리스트의 궤적 간의 차를 정의하는 함수이다. 객체의 궤적과 global motion 은 움직임의 방향에서 큰 차이를 보이므로, 움직임의 방향을 가중치로 사용하여 식을 정의하였다. Global motion 과 차이가 큰 궤적을 가지는 영역 리스트들간에 식(3)을 이용한 궤적의 차이가 작은 영역 리스트들을 클러스터링하여 움직이는 객체를 검출한다.

3. 실험 결과

3.1 실험 데이터

본 실험은 복잡하게 움직이는 카메라의 환경 속에서 객체의 전후좌우의 여러 가지 움직임을 가진 데이터에 대하여 실험하였고, 표 1 은 예시로 보이기 위해 실험 데이터 3 개의 움직이는 객체, 객체의 움직임, 카메라의 움직임을 설명한다.

표 1. 실험 데이터의 설명

비디오 클립		
비디오 1	움직이는 객체	Woman
	객체의 움직임	Move right
	카메라의 움직임	Pan right
비디오 2	움직이는 객체	Woman
	객체의 움직임	Move right Move forward
	카메라의 움직임	Pan left
비디오 3	움직이는 객체	Man
	객체의 움직임	Move left
	카메라의 움직임	Pan left Zoom in

3.2 실험 결과

실험은 비디오에서 발생하는 총 영역 리스트의 수를 구하고, 이 중에서 실제로 객체에 해당하는 영역 리스트의 수와 제안된 방법으로 검출된 객체의 영역 리스트의 수를 비교하였다. FRR(False Rejection Ratio)과 FAR(False Acceptance Ratio)을 정의하여 실험 결과를 쉽게 볼 수 있도록 하였다.

$$FRR = \frac{\text{실제 객체이면서 검출되지 않은 영역 리스트의 수}}{\text{실제 객체의 영역 리스트의 수}} \quad (4)$$

$$FAR = \frac{\text{실제 객체가 아니면서 검출된 영역 리스트의 수}}{\text{실제 객체가 아닌 영역 리스트의 수}} \quad (5)$$

표 2는 각 비디오 클립에서의 실험 결과를 보여준다.

표 2. 움직이는 객체의 추출 결과

동영상	총영역 리스트	객체영역 리스트	FRR	FAR
비디오 1	5	2	0%	33%
비디오 2	12	5	20%	14%
비디오 3	16	6	33%	10%



그림 7. 검출된 객체

그림 7은 위에서 실험 데이터로 사용한 비디오의 주요 두 장면과 검출된 움직이는 객체를 보여준다. 비디오 1의 움직이는 객체 검출 결과는 오른쪽으로 움직이는 여자를 추적하는 카메라의 환경에서 여자를 안정적으로 검출하는 것을 보여준다. 그러나 오른쪽 배경 영역은 동영상의 중간 부분에서 인접한 배경 영역과 합쳐지는데, 이로 인해 채적이 크게 바뀌어 객체로 구분되는 FRR이 발생했다. 비디오 2는 5개의 객체 영역 리스트 중 4개가 객체로 검출되며, 오른 팔 부분은 다른 영역 리스트와 채적의 차이가 커서 객체로 검출되지 않는 FAR이 발생했다. 비디오 3은 얼굴부분이 고개를 돌리는 과정에서 면적이 작아져 매칭이 끊기는 현상이 발생한다.

제안한 방법을 이용하여 동영상에서 움직이는 객체를 검출하는 데, 움직이는 객체를 카메라의 줌, 이동의 환경에서도 강인하게 찾아냈다.

4. 분석과 향후 과제

실험 결과 복잡하게 움직이는 카메라 환경에서 객체가 전후좌우의 다양한 움직임을 가진다 안정적으로 움직이는 객체를 검출하였다. 그러나 한 영역이 두 영역으로 나누어지거나 두 영역이 한 영역으로 합쳐지는 경우와 인접한 영역과 겹쳐서 완전히 가려지는 경우에는 영역 매칭이 제대로 되지 않아 움직이는 객체를 정확히 검출하는 데 어려움이 있었다. 이러한 문제점을 해결하기 위해 1 대 1 매칭이 아닌 새로운 매칭 알고리즘과, 중간에 끊기어 둘로 나누어진 영역 리스트를 하나의 영역 리스트로 합하는 알고리즘을 고려하여, 보다 안정적인 움직이는 객체 검출 기술을 개발할 계획이다.

5. 결 론

본 논문에서는 Mean shift 알고리즘, 영역 매칭, 영역 리스트 간의 채적을 이용한 클러스터링을 통해 움직이는 객체를 검출하는 방법을 제안하였다. 제안한 방법은 영역을 기반으로 하여 매칭의 정확도를 높였고, 복잡한 카메라의 움직임 환경에서도 움직이는 객체를 안정하게 검출하였다. 제안한 방법은 동영상의 내용을 표현하는데 중요한 특징값인 움직이는 객체를 검출할 수 있으며, 객체의 모양, 색상 등을 이용하여 유사한 객체를 가진 동영상을 검색하는 기능에 사용할 수 있을 것이다.

참고 문헌

- [1] R. Pless, T. Brodsky, Y. Aloimonos, "Detecting independent motion: the statistics of temporal continuity," Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 22 Issue: 8 , Aug. 2000
- [2] Pei-Yin Chen, Jer Min Jou, "An efficient blocking-matching algorithm based on fuzzy reasoning," Systems, Man and Cybernetics, Part B, IEEE Transactions on , Volume: 31 Issue: 2 , April 2001
- [3] D. Comaniciu, P. Meer, "Mean Shift Analysis and Applications," Computer Vision, 1999. IEEE International Conference on , Volume: 2 , 1999