

# 라인 매칭 기법을 이용한 실시간 움직임 검출과 추적기법

이 재 호, 장 석 환, 김 회 율  
한양대학교 전자전기공학부 영상공학 연구실

## Real-time Motion Detection and Tracking using Line-matching Algorithm

Jaeho Lee, Seokhwan Jang and Whoiyul Kim  
IELab, Division of Electrical and Computer Engineering, Hanyang Univ.  
E-mail : jhlee@vision.hanyang.ac.kr

### 요 약 문

본 논문에서는 Pan/Tilt 움직임이 있는 카메라 영상에서 실시간으로 이동하는 물체를 검출하고 추적하기 위한 라인매칭(Line-matching)알고리즘을 제안한다. 또한 물체를 추적하기 위해 색상 성분의 분포와 물체의 움직임을 동시에 이용하여 특징 값을 매칭 하는 모션-칼라 매칭(Motion-Color matching)방법을 제안한다. 본 논문에서 제시한 라인매칭 알고리즘은 움직이는 카메라 영상 안에서 움직이는 물체를 추적하는데 있어 효율적으로 카메라의 움직임을 보정하며, 그에 따른 연산 시간도 현저히 줄일 수 있는 방법이다. 실험에 의하면 카메라로부터 입력되는 영상에서 움직임을 검출 추적하는데 있어 초당 10~12 frame의 연산 속도를 보였으며, 추적하는 대상에 대하여 배경의 움직임이나 주위의 환경에 영향을 받지 않는 강인한 추적 결과를 보였다.

### 1. 서론

최근 들어 비디오 영상으로부터 움직임 검출과 추적에 대한 중요성이 인식됨에 따라 이에 대한 많은 연구가 진행되어지고 있다[1]. 현재까지 움직임 검출에 대한 대부분의 연구가 고정된 카메라를 사용한 환경에 대하여 진행되어 왔다[2]. 고정된 카메라에서 움직이는 물체를 검출해 내고 추적하는 알고리즘들은 카메라의 움직임을 고려한 시스템보다 비교적 쉽게 구현이 된다. 이러한 기법의 대부분은 카메라로부터 입력되는 영상을 이용하여 배경 이미지를 제작하고, 이 배경 이미지와 현재 입력 영상과의 차를 이용하여 현재 영상에서의 움직이는 물체를 검출하는 방법을 취하고 있다[3]. 움직이는 물체

에 대한 추적방법도 위와 같은 방법으로 배경 이미지와 현재 입력영상과 차이를 이용하여 수행된다[2][4][5]. 그러나 이러한 기법들은 카메라가 고정된 환경에 대해서만 적용이 가능하며. 카메라가 움직이는 경우에는 영상 전체에 대한 변화를 계산하여 전체의 움직임을 보정하고 이를 보정된 영상을 이용하여 움직임을 검출하는 방법이 필요하다. 기존의 카메라 움직임에 대한 보정 방법들로는 Optical Flow를 이용한 방식, Block Matching 방식, affine motion model방식 등의 방법을 사용하였는데, 이러한 방식은 그 연산의 양이 많아 이를 실시간으로 처리하는 시스템에 적용하기에는 다소 문제가 있다 [6][7]. 이에 본 논문에서는 시스템의 특성에 맞는 실시간 움직임 보정 알고리즘으로 라인매칭(Line-matching) 기법을 개발하였다.

물체의 추적을 위해서는 물체로부터의 특징 값을 추출하는 과정이 필요하다. 물체의 특징 값으로는 모양, 색상, 크기, 속도 등 여러 가지의 요소를 사용할 수가 있는데, 어떠한 요소를 선택할지 여부는 그 시스템의 적용 및 응용 환경에 따라 많이 좌우된다. 예를 들어 추적 물체의 모양이 변화지 않는 물체에 대해서는 그 물체의 모양 특징 값이 중요한 위치를 차지하게 된다[8]. 본 논문에서는 추적 대상이 그 위치와 방향에 따라 모양의 변화가 현저하게 나타나는 사람이나 자동차등임을 감안하여, 특징 값으로 크기의 변화와 모양의 변화에 강인한 물체의 색상성분 분포를 이용하고, 신뢰도의 향상을 위하여 움직임의 정보를 추가한 모션-칼라 매칭(Motion-Color matching) 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 움직이는

카메라에서 영상간의 보정을 위한 라인 매칭 기법과 검출된 물체에 대하여 분리하는 segmentation 방법, 그리고 분리된 물체에 대하여 추적하는 모션-칼라 매칭 방법에 대하여 기술하였다. 3장에서는 본 연구를 바탕으로 제작한 실시간 움직임 검출 및 추적 시스템을 이용한 실제 실험 결과를 도시하였으며 마지막으로 4장은 본 연구의 결론으로 구성되어 있다.

## 2. 움직임 검출과 추적 기법

### 2.1 카메라 움직임 보정 알고리즘(라인 매칭)

카메라의 움직임은 Boom up, down(카메라 중심축의 상/하 움직임), Track left, right(카메라 중심축의 좌/우 움직임), Dolly backward, forward(카메라 중심축의 전/후 움직임), Pan left, right(카메라 중심축은 고정하고 좌/우로 움직임), Tilt up, down(카메라 중심축은 고정하고 상/하로 움직임), 그리고 Roll right, left(카메라를 시계/반시계 방향으로 회전)로 나누어진다. 현재까지 많은 연구들이 이러한 카메라의 움직임을 입력되는 영상으로부터 얻어내기 위하여 8개의 파라미터를 계산하는 perspective motion model을 이용하는 알고리즘과 6개의 파라미터를 계산하는 affine motion model을 이용하는 알고리즘들을 사용하였다[9]. 그러나 본 논문에서는 구현하는 시스템은 카메라의 중심축은 고정되어있고 단지 Pan과 Tilt 모션만이 카메라의 움직임으로 간주되므로 이러한 시스템의 특성에 맞게 실시간으로 카메라 움직임을 보정해주는 알고리즘이 필요하였다

라인 매칭(Line-matching)알고리즘에서는 현재와 과거의 프레임들 비교할 때 각각의 이미지 라인(low, column)별로 위치를 변경하면서 유사도를 비교하며, 전체의 이미지에서 가장 빈도가 많은 움직임을 찾아서 이를 카메라의 전체 움직임으로 간주하는 것이다. 아래는 이미지에서 카메라의 Pan 움직임 값을 구하는 라인 매칭 알고리즘의 요약이다.

1) 현재 이미지의 각 라인에 대하여 이전 이미지에서의 라인을 좌우로 움직이면서 그 차의 합을 구한다.

$$D_j = \sum_{x=0}^n abs[I_1(x) - I_2(x-j)]$$

where  $j$  : amount of shift,  
 $n$  : length of low

2) 가장 값이 작게 나타나는 움직임을 구하여 이를 그 라인의 움직임으로 결정한다.

$$Y_{shift\ low} = arg\ MIN(D_j)$$

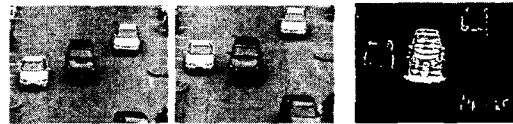
where  $Y$  : column number

3) 각 라인의 움직임 값들을 projection하여 그 중에서 가장 크게 나타나는 값을 그 이미지의 전체 움직임으로 결정한다.

$$G_{shift\ low} = arg\ MAX\{P(Y_{shift\ low})\}$$

where  $P(\ )$  : projected number,  
 $G$  : Global Image motion

1)번 연산에서 비교할 영역은 한 프레임 사이에 최대 로 움직일 수 있는 범위로 한정한다. 2)번 연산은 1)번 연산이 한 라인에 대해 수행된 후, 그 값들 중에서 가장 적은 값을 연산한 라인의 움직임으로 결정하는 것이다. 1)번과 2)번 연산으로 모든 라인에 대해 움직임 값들이 구해지면, 전체 이미지의 움직임을 구하기 위해 각각의 라인에서 구한 움직임 값들 중 가장 많은 빈도를 가진 값을 선택하는 연산이 3)번이다. 위와 같은 방법으로 카메라의 Tilt 움직임 값도 구한다. 이렇게 구하여진 Pan과 Tilt의 움직임 값을 이용하여 연속되는 두 이미지의 차를 구하면 카메라의 움직임과 독립적으로 움직이는 물체를 구별해 낼 수 있다. 이 라인매칭 기법은 카메라 움직임을 구할 시에 각 라인 움직임 값의 빈도 중 가장 많은 값을 찾으므로, 이미지 안의 독립적으로 움직이는 물체로 인하여 발생하는 카메라 움직임 값의 계산 오류를 배제할 수 있다. 또한 pan/tilt 움직임이 동시에 일어났을 경우에 대하여서도 이어지는 두 프레임간에는 움직임이 현저하게 나타나지 않으므로 계산 값에 오차가 발생하지 않는다. Pan/Tilt의 움직임 값을 구하기 전에 전처리 과정으로 입력되는 영상에 대하여 이미지의 윤곽선(edge)을 추출하고, 이렇게 추출된 윤곽선 이미지를 이용하여 라인매칭 값을 구하였다. 이는 영상 자체에 대하여 매칭 값을 계산하는 것보다 윤곽선 영상을 이용하면 그 값의 차이가 크므로 안정적으로 Pan/Tilt 움직임 값을 구할 수 있기 때문이다. 그림 1은 위에서 설명한 라인매칭기법으로 카메라의 움직임을 검출하고 이를 복원하여 두 영상간의 차를 구한 결과이다. (a)의 입력 영상에서 중앙의 자동차만 움직이고 있다.



(a) 입력 영상 (b)보정후의 영상

그림 1. 카메라의 움직임에서 움직이는 물체의 검출, tilt motion = -9. pan motion = 3

위의 그림에서 보면 카메라의 움직임을 복원하여도 움직임이 일어나지 않은 물체의 경계선 부분이 표시가 되는 것을 볼 수 있다. 이는 이 논문에서 제시한 알고리즘이 카메라의 Pan/Tilt 모션의 영향으로 발생하는 이미지에 투영되는 물체의 크기 변화에 대한 보정 연산이 없기 때문이다. 그러므로 움직이지 않는 물체의 경계선 부분에 이로 인해 생기는 노이즈가 발생하게 된다. 이렇게 생기는 노이즈들은 Morphology 기법을 이용해 실시간으로 간단하게 제거 될 수 있다..

## 2.2. 움직이는 물체의 분리(Segmentation)

물체의 움직임이 검출되었으면 이 물체에 대한 특징 값을 추출하기 위하여 물체 자체만을 분리하는 과정이 필요하다. 이 과정을 segmentation이라고 하며, 본 연구에서는 연속되는 세 프레임에서의 이미지간의 차를 이용하여 물체를 분리하는 방법을 제안한다.

카메라가 Pan/Tilt로 움직이면서 라인매칭을 이용하여 물체의 움직임이 검출되면 검출된 위치에서 카메라를 고정시키고 연속되는 프레임에서 그 전 프레임 간의 차를 구한다. 이 연산은 연속되는 세 프레임 간에서 수행하며, 이는 움직이는 물체가 이미지의 영역 안에서 벗어나기 전에, 배경 영역이 최대한 들어가지 않도록 물체만을 분리해 내기 위함이다.

연속되는 세 프레임에서 두개의 차(difference) 영상이 얻어지며, 이를 이용하여 물체를 분리하는 과정은 다음과 같다.

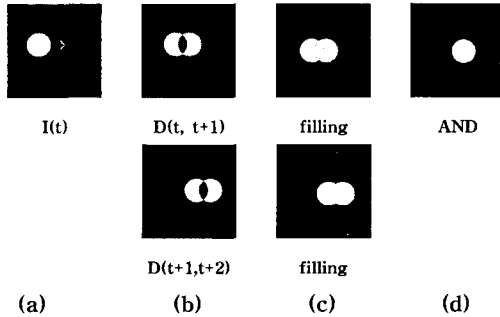


그림 2. 움직이는 물체의 분리

위의 그림에서 (a)는 오른쪽으로 움직이는 물체를 나타내고 있으며 (b)의 상단은 t+1 프레임과의 이미지 차를 나타내고 하단은 t+1 프레임과 t+2 프레임과의 이미지 차를 나타내고 있다. (c)는 이 두 차 영상에 대하여 filling으로 만들어진 영상이고 (d)는 (c)의 상단과 하단 이미지에 대하여 AND 연산을 취한 것이다.

위의 그림에서 채워지는 영역 크기의 결정은 추적할 대상의 이미지에 나타나는 크기를 미리 검토하고 그 크기를 이용한다. 채워진 두 차 영상의 공통 부분만을 추출하면 배경영역이 가장 적게 들어가는 움직이는 물체에 대한 이미지가 분리되어진다.

## 2.3. 색상성분을 이용한 매칭 알고리즘

본 연구에서는 추적하는 대상이 사람이나 자동차등 거리와 방향에 따라 그 모양과 크기가 변하는(Non-rigid body)모델이므로 이러한 변화에도 강인하게 적용하기 위하여 색상성분을 특징 값으로 적용하였다.

이미지의 칼라모델로는 RGB, YUV, HSI 등 많은 모델들이 있으나 본 연구에서는 하나의 값으로 색상을 표시할 수 있는 hue값을 특징 값으로 채택하였다. 입력되

는 이미지는 RGB형식으로 되어있기 때문에 hue로 변환하는 과정이 필요하다. hue 변환 식은 다음과 같다.

$$H = \arccos\left(\frac{0.5 * (RG + RB)}{\sqrt{(RG * RG + RB * RB)}}\right)$$

where  $RG=R-G$ ,  $RB=R-B$ ,  $GB=G-B$

이러한 hue 값은 칼라 성분의 값을 하나의 원에 있는 각각의 각도 값으로 대응시켜 놓은 것으로, 색상 값이 하나의 특징 값으로 표시되는 장점이 있으나, 이미지의 값이 흰색과 검은색 계통의 그레이에 가까운 색일 경우에는 그 표현방법이 없는 단점이 있다. 그러므로 움직이는 물체가 흰색이나 검은색의 성분일 경우에 대하여는 정확한 값을 계산하는데 문제가 발생하는 것을 알 수 있었다. 이를 보완하기 위하여 특징 값의 범위를 픽셀의 값이 칼라의 성분으로 표현 가능한 경우인 361개와 그렇지 않을 경우의 그레이로 변환한 256개의 값을 더한 617개로 정하였다.

분리된 영상에 대하여 각각의 픽셀 값을 이러한 특징 값으로 변환한 후, 그 값을 히스토그램(histogram)으로 나타낸 것이 그림 3이다. 그 중 빈도수가 많이 나타나는 값들을 선택한 후 이 값들을 이 물체가 나타내는 특징 색상으로 정의한다.

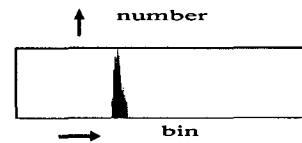


그림 3. 분리된 물체의 특징 값 분포도.

물체의 특징 색상 값이 정해지면 이어지는 영상에 대하여 앞에서 정의된 특징 값으로 이미지를 변환하고 이를 이용하여 움직이는 물체를 추적해 나간다. 이때 전체 영역에 대하여 이를 표시하면 비슷한 특징 값을 갖고 있는 물체에 영향을 받을 수가 있으므로 앞에서 제안한 라인매칭기법을 이용하여 움직임이 발생한 영역에 대하여만 그 특징 값을 산출해 내고 그렇지 않은 영역에 대해서는 연산을 수행하지 않는다. 또한 신뢰도 향상을 위하여, 움직임이 일어난 영역 중 이전 프레임에서 추적하고 있는 물체의 영역에 대하여 일정구간 안에 존재하는 움직임들만을 의미 있는 움직임으로 간주한다. 위와 같은 방법으로 현재 이미지에서 특징 값을 산출할 후보 영역을 정의한다.

후보 영역이 정의되면, 영역 안의 이미지에 대하여 특징 값을 계산한다. 이렇게 표시된 특징 값들에 대하여, 오류 영역을 배제하기 위하여, 특징 값들의 위치에 관한 평균과 분산을 구한 후 분산의 2배 안에 존재하는 픽셀들만 선택한다. 추적 대상이 있는 영역이 성공적으로 찾지 않으면 그 영역이 입력영상의 중앙에 위치하도록 카메라의 모션을 제어하면서 추적해 나간다. 그림으로 도시하면 다음과 같다.



(a) (b) (c)  
 그림 4. 움직이는 물체에 대한 특징 값 표시, (a) 입력 영상 (b) 라인 매칭으로 움직임을 검출한 영상 (c) 검출된 영역 안에서의 칼라 특징 값 표시 영상

### 3. 실험 결과

본 논문에서 제안한 방법으로 카메라로부터 입력되는 영상들에 대하여 실시간(10~12 frame/sec)으로 움직임을 검출하고 추적할 수 있는 시스템을 제작하였다.

아래의 그림들은 앞에서 설명한 움직임 검출과 추적기법을 이용하여 제작한 실제 시스템을 사용하여 움직이는 물체에 대한 검출과 추적 결과이다.

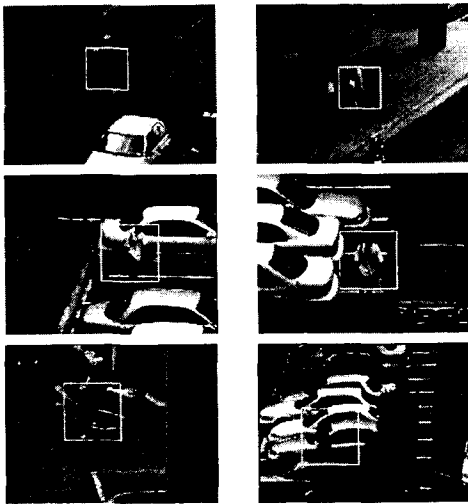


그림 5. 움직이는 물체의 검출과 추적결과

detection rate	man	98%	
	car	100%	
tracking rate	man	single	96%
		multi	84%
	car	single	87%
		multi	76%

표 1. 검출 및 추적 실험 결과

움직이는 물체의 검출 과정에서 사람의 크기가 자동차의 크기보다 작아 검출 시 효율이 다소 떨어지기는 하였으나 검출 성능은 매우 높게 나타남을 알 수 있다. 그러나 추적하는 과정에서 물체의 속도가 너무 빠르거나 비슷한 특성의 이동하는 물체와 근접하는 경우에는

추적 효율이 현저히 줄어드는 것을 알 수 있었다. 이에 대한 보완으로 추적 과정에서 이동 물체에 대한 정확한 예측과정 도입과 효율적인 특징 값의 연구가 필요하다.

### 4. 결론

본 연구는 Pan/Tilt 드라이버를 이용하여 카메라를 움직이면서 실시간으로 움직이는 물체를 검출하고 추적하는 지역 감시 시스템의 제작을 목적으로 하였다. 이러한 지역 감시 시스템을 구현하기 위하여 실시간으로 카메라의 움직임을 보정하여 카메라의 움직임과 움직이는 물체의 움직임을 구별하는 라인 매칭(Line-matching) 알고리즘을 제시하였다. 또한 움직임이 검출된 후 검출된 물체를 효율적으로 추적하기 위한 알고리즘으로 물체의 색상성분 분포와 움직임의 정보를 이용한 모션-칼라 매칭(Motion-Color matching)기법을 구현하였다.

본 연구의 실용성 향상을 위하여 더욱 빠르고 정교한 카메라 모션의 복원 알고리즘의 개발과 신뢰도의 향상을 위하여 추적하는 물체에 대한 예측 알고리즘 등의 효율적인 적용이 필요하다

### 참고 문헌

- [1] M. shah and R. Jain. Motion-based Recognition. Kluwer Academic, 1997.
- [2] R. Rosales and S. Sclaroff, Improved Tracking of Multiple Humans with Trajectory Prediction and Occlusion Modeling, IEEE Conf, on CYPR, 1998
- [3] P. M. Ngan, Motion detection in temporal clutter, Tech, Rep, 693, Industrial Research Limited, Auckland, New Zealand, March 1997,
- [4] McFarlane N. J .B., Schofield, C.P. Segmentation and tracking of piglets in images, MVA, Vol. 8, pp. 187-193, 1995.
- [5] K Rohr, Incremental recognition of pedestrians from image sequences, Computer Vision and Pattern Recognition, pages 8-13, 1993.
- [6] Pentland A. and Horowitz B. Recovery of nonrigid motion and structure, IEEE Trans, on Pattern Analysis and Machine Intelligence, 13(7): 730: 730-742, July 1991.
- [7] M. Isard, A. Blake, CONDENSATION-conditional density propagation for visual tracking, Int. J. Computer Vision, 1998,
- [8] Cootes T. J., Taylor C .J., Cooper D. H., and Graham J.. Training models of shape from sets of examples, In British Machine Vision conference, pages 9-18, September 1992,
- [9] Ali Tabatabai, A Method for Indexing and Searching Video Databases based on 3-D Camera Velocity, MPEG7 proposal, 1999