

# 공간적, 시간적 경계 정보를 이용한 이동 객체 윤곽선 검출 방법

\*곽재호 \*김희율

한양대학교 전자통신컴퓨터공학과

\*jhkwak@vision.hanyang.ac.kr

## Moving Object Contour Detection using Spatial and Temporal Edge

\*Kwak, Jae-Ho \*Kim, Whoi-Yul

Dept. of Electrical and Computer Engineering, Hanyang University

### 요약

본 논문에서는 Spatial Edge와 Temporal Edge를 이용한 이동 객체의 윤곽선 검출 방법을 제안한다. 카메라로부터 연속적으로 입력되는 영상에서 이동 객체의 윤곽선이 존재하는 후보 영역을 검출하기 위해, 새로운 방법의 Temporal Edge를 제안한다. Temporal Edge를 통해 검출된 후보 영역을 중심으로 Spatial Edge를 구하고, 후처리 과정을 통해 노이즈를 제거한 후 최종적으로 이동 객체의 윤곽선을 검출한다. 제안한 방법은 실험을 통해 그 성능을 확인하였고, 배경 차 방법과 비교 하였다.

### 1. 서론

최근 늘어나는 강력 범죄로 인해 보안에 대한 요구가 증가함에 따라 CCTV나 카메라와 같은 영상 장비를 활용한 지능형 감시 시스템 개발이 중요한 연구 분야로 대두되고 있다 [1]. 지능형 감시 시스템은 카메라로부터 입력되는 영상에서 이동 객체를 자동으로 감지하고, 이를 추적함으로써 인간을 대신하여 감시 역할을 수행할 수 있는 자동화된 감시 시스템을 말한다. 지능형 감시 시스템의 가장 기본이 되는 기술은 카메라로부터 입력되는 영상에서 이동 객체를 검출하는 것이다. 이동 객체의 검출 성능에 따라 시스템 성능이 좌우된다.

이동 물체를 검출하는 대표적인 방법은 배경 차이 기법[2-5]과 윤곽선 추출 기법[6-7]이 있다. 배경 차이 기법은 이동 객체가 없는 배경 영상과 움직임이 존재하는 현재 입력 영상의 차이를 구하여 배경 부분을 제외한 이동 객체를 분할하는 방법으로써, 이동 객체 검출에 가장 많이 쓰이는 방법이다. 이전 영상들로부터 배경이 잘 유지되었을 경우, 그리고 조명의 변화가 심하지 않을 경우 뛰어난 검출 결과를 보인다. 하지만 배경 영상의 유추가 잘못 되었거나 자연광이나 인공조명등의 갑작스러운 조명 변화가 있는 경우 검출 성능이 저하된다는 단점이 있다. 뿐만 아니라 장시간 정지 상태에 있는 객체나 천천히 이동하는 객체가 있는 상황에서는 배경 검출이 어려운 단점이 있다.

윤곽선 추출 기법은 입력된 영상에서 Edge추출하고, 추출된 Edge에서 배경에 해당하는 픽셀을 제거한 후 물체를 검출하는 방법이다. 이 방법은 Edge 정보를 활용한다는 점에서 배경 차이 기법보다 조명 강인하다는 장점이 있다. 그러나 입력된 영상으로부터 Edge 정보를 추출하기 위한 연산 량이 많다는 단점이 있다. 뿐만 아니라 Edge 정보를 추출하는 방법에 따라 윤곽선의 추출 성능이 좌우 된다는 단점이 있다.

본 논문에서는 Spatial Edge 정보와 Temporal Edge 정보를 활용하여 이동 객체의 윤곽선(contour) 검출 알고리즘을 제안한다. 알고리즘의 초기 단계에서 이동 객체의 윤곽선이 존재하는 후보 영역을 검출하기 위해 새로운 방법의 Temporal Edge를 제안한다. 현재 시간  $t$ 에 입력된 영상을 중심으로  $t-2$ 시간에 입력된 영상과의 차를 6개의

요소 위치를 기준으로 구한 후, 이 값들을 산술적 누적하여 Temporal Edge를 구한다. Temporal Edge를 통해 검출된 후보 영역을 중심으로 Spatial Edge를 구하고, 후처리 과정을 통해 노이즈를 제거한 후 최종적으로 이동 객체에 대한 윤곽선을 검출한다.

제안한 방법의 배경 영상을 생성하지 않기 때문에 배경 영상을 사용하는 방법에 비해 연산 속도가 빠르며, 배경 영상을 이용함으로써 발생하는 여러 가지 문제들을 극복할 수 있다. 배경 차이 기법의 경우 조명의 변화가 심한 경우 신뢰성 있는 배경 유추가 이루어지지 않아 높은 검출 오류를 보이는 반면, 제안하는 방법은 영상의 Edge정보를 활용하기 때문에 조명 변화에 강인하여 높은 검출 성능을 보인다. 뿐만 아니라 카메라로부터 오랜 시간 입력되는 영상을 이용하여 배경을 유추하는 배경 차이 기법의 경우 오랜 시간 정지해 있던 물체가 갑자기 움직이는 경우 물체의 잔상 현상(ghost effect)으로 잘못된 검출 결과를 보이지만, 제안하는 방법은 시간  $t, t-1, t-2$ 에서 입력되는 세 개의 영상만을 활용하기 때문에 Ghost 현상이 발생하지 않아 움직이는 물체의 윤곽선을 정확하게 검출할 수 있는 장점이 있다.

본 논문의 구성은 다음과 같다. 2장에서 Temporal Edge를 포함한 제안한 알고리즘 방법을 설명한다. 3장에서는 실험을 통해 제안한 방법의 성능을 분석하고, 4장은 결론을 기술한다.

### 2. 제안한 방법

제안한 방법의 전체 흐름도는 그림 1과 같다.

#### 가. Temporal Edge 검출

이동 객체의 윤곽선이 존재하는 후보 영역을 검출하기 위해 새로운 방법의 Temporal Edge를 제안한다. Temporal Edge는 임의의 두 장의 Gray 영상을 이용하여 다음의 방법으로 구한다.

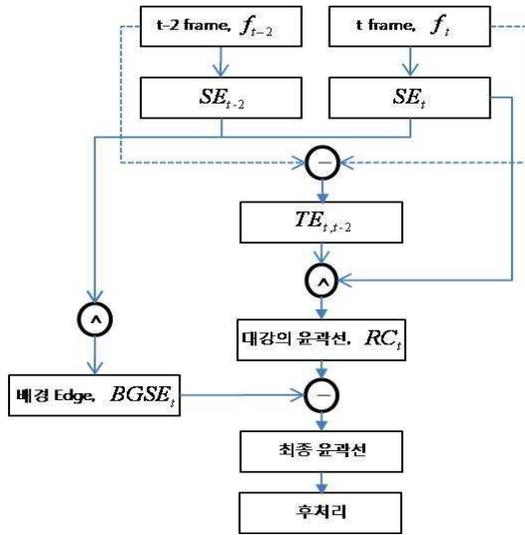


그림 1 전체 흐름도

시간  $t$ 에서의 영상을  $f_t$ 이라 하고, 시간  $t-2$ 에서의 영상을  $f_{t-2}$ 라고 하면 두 개의 영상으로부터 구해지는 Temporal Edge  $TE_{t,t-2}$ 는 식(1) 과 같이 구해진다.

$$TE_{t,t-2}(x,y) = \sum_{j=0}^5 sum[j]$$

$$sum[j] = \sum_{i=-1}^1 abs(\alpha - \beta), j = 0,1,2$$

$$sum[j] = \sum_{i=-1}^1 abs(\gamma - \delta), j = 3,4,5 \quad (1)$$

단,  
 $\alpha = f_t(x+(j-1), y+i)$   
 $\beta = f_{t-2}(x+(j-1), y+i)$   
 $\gamma = f_t(x+i, y+(j-4))$   
 $\delta = f_{t-2}(x+i, y+(j-4))$

식(1)에서  $TE_{t,t-2}$ 를 구하기 위해 6개의 요소 위치를 중심으로 두 영상의 차를 계산 하였다. 그림 2는  $TE_{t,t-2}$ 를 구하기 위해 사용되는 요소 위치  $sum[j]$ 를 보인 것이다.

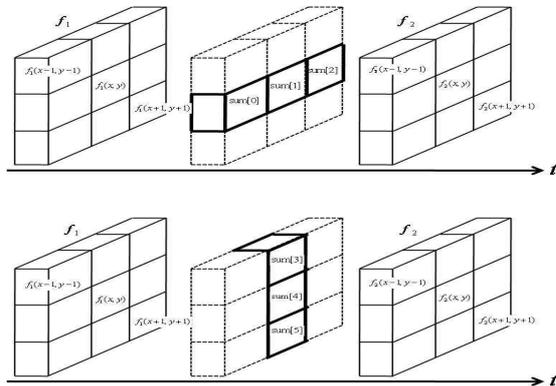


그림 2 Temporal Edge 계산을 위한 요소 위치

그림 3은 식 (1)을 이용하여 검출한 Temporal Edge 이다 .

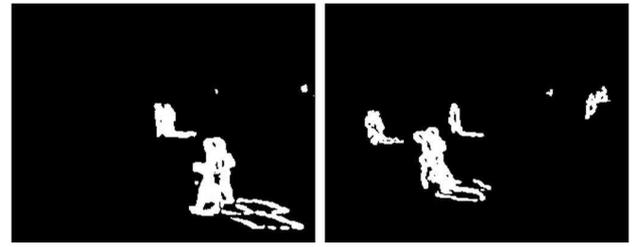


그림 3 Temporal Edge

#### 나. Spatial Edge 검출

카메라로부터 연속적으로 입력되는 매 영상마다 Canny Edge [8]를 검출하여 이를 Spatial Edge 정보  $SE_t$ 로 활용한다. 그림 4는 검출된 Canny Edge를 보인 것이다.

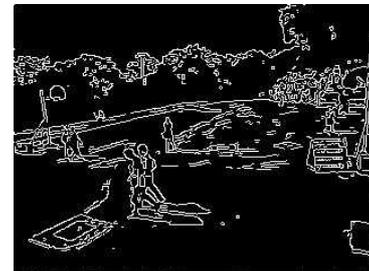


그림 4 Spatial Edge(Canny Edge)

#### 다. 대강의 윤곽선 추출

연속적으로 입력되는 매 영상으로부터 검출한  $SE_t$ 로부터 실제 이동 객체의 윤곽선을 검출하기 위해 Temporal Edge 정보  $TE_{t,t-2}$ 을 활용한다. 일반적으로 객체의 윤곽선이 존재하는 픽셀의 경우 높은 값의 기울기(gradient)를 보인다. 이동 객체의 경우 윤곽선에 해당하는 픽셀과 그 주변(일반적으로 3x3 window 영역 내)을 중심으로 큰 픽셀 값 차이와 픽셀의 기울기(gradient) 차이를 보인다. 이러한 속성은 Temporal Edge와 Spatial Edge 모두에 반영되어 나타난다. 따라서 Temporal Edge 픽셀을 중심으로 그 주변에 Spatial Edge가 나타날 확률이 높다. Temporal Edge를 기준으로 검출한 후보영역  $TE_{t,t-2}$  내에 존재하는 각 픽셀들을 중심으로 일정 크기(본 논문에서는 5x5 window) 영역 내에 존재하는 Spatial Edge를 검출하여 이동 객체에 대한 대강의 윤곽선(Rough Contour)  $RC_t$ 를 검출한다. 표 1은  $RC_t$ 를 검출하는 Pseudo-code 이다.

그림 5는 대강의 객체 윤곽선 검출 결과이다. 그림 5에서 보듯이 이동 객체에 대한 대강의 윤곽선이 검출됨에 따라 앞서 언급한 확률적 가정이 정당함을 볼 수 있다. 그러나 그림 5의 결과에는 우리가 원하지 않는 윤곽선 정보가 많이 포함되어 있다. 이는  $TE_{t,t-2}$  픽셀을 중심으로 주변 영역(5x5 window)에 있는 픽셀들을 고려하다보니 배경 Edge가 포함되어 검출되었기 때문이다. 따라서 배경 Edge를 생성하여 배경 Edge를 제거하는 후처리 과정이 필요하다

```

영상 상에서 기준 좌표 x, y에 대해
MxN 크기의 window를 고려하면

int m = (int)(M/2);
int n = (int)(N/2);

for( int r=-n; r<=n; r++ ) {
  for( int c=-m; c<=m; c++ ) {
    int se =  $SE_t(x+c, y+r)$ ;
    if( se>0 ) {  $RC_t(x+c, y+r) = se$ ; }
  }
}

```

표 1 대강의 윤곽선  $RC_t$ 를 구하기 위한 Pseudo-code

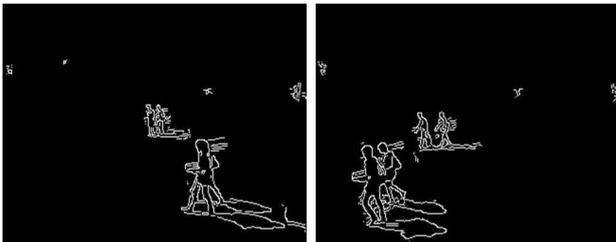


그림 5 검출된 대강의 윤곽선

### 라. 배경 Edge 생성

앞에서 구한 대강의 객체 윤곽선  $RC_t$ 에는 객체의 실제 윤곽선에 해당하는 픽셀뿐만 아니라 배경 Edge에 해당하는 픽셀이 포함되어 있다. 따라서 이를 제거하기 위해서 배경의 Edge 정보가 필요하다. 우리는 매 시간 입력되는 영상으로부터 Spatial Edge정보  $SE_t$ 를 추출하고 있다. 배경 Edge의 경우 매 영상 추출되는 Spatial Edge 상에 동일한 위치에 나타날 확률이 높다. 따라서 배경 Edge로 나타나는 횟수를 누적하여 배경 Edge에 대한 Likelihood를 구하고 이를 배경 Edge 정보로 활용한다.  $SE_t$ 상에서 임의의 위치의 픽셀이 Edge 인 경우  $BGSE_t$ 의 동일 위치상의 픽셀 값을 1 증가 시키고, 반대로 Edge가 아닌 경우에는  $BGSE_t$ 의 동일 위치상의 픽셀 값을 1 감소 시킨다.  $SE_t$  상에서 연속적으로 윤곽선으로 나타나는 픽셀은 배경 Edge일 확률이 높으며 결국  $BGSE_t$  상에서 높은 픽셀 값을 타나내게 된다. 배경 Edge 정보  $BGSE_t$ 는 식 (2)과 같이 구한다.

$$\begin{cases}
 BGSE_t(x,y) = \min(a, 255) \text{ if } SE_t(x,y) > th_1 \\
 BGSE_t(x,y) = \max(b, 0) \text{ Otherwise}
 \end{cases}
 \quad (2)$$

단,  
 $a = BGSE_{t-1}(x,y) + 1$   
 $b = BGSE_{t-1}(x,y) - 1$

여기에서  $BGSE_t(x,y)$ 는 Gray 영상이며, 시간  $t$ 에서 좌표  $x, y$  위치에서의 픽셀 값을 말한다.  $SE_t$ 는 2나 절에서 구한 값으로

시간  $t$ 에 입력된 영상에 대한 Spatial Edge 정보를 말한다.  $th_1 (= 0)$ 는 임의의 경계값(threshold)이다. 그림 6은 구해진 배경 Edge 영상을 보인 것이다.



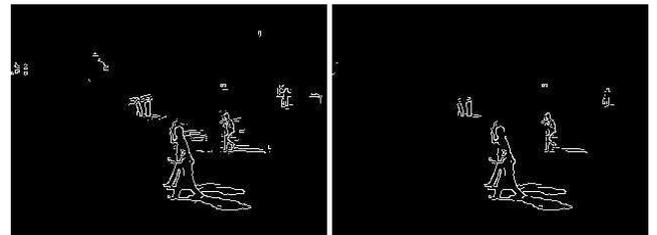
그림 6 배경 Edge

### 마. 배경 Edge 제거

생성된 배경 Edge 영상을 이용하여 노이즈를 제거하는 방법은 식 (3)과 같다. 앞 절에서 구한 대강의 윤곽선 영상  $RC_t$  내에 존재하는 픽셀 중 배경 Edge 영상 내에 존재하는 픽셀과 동일한 위치에 존재하는 경우, 이를 배경 Edge로 간주하고 제거한다.

$$RC_t(x,y) = 0, \text{ if } BGSE_t(x,y) > th_2 \quad (3)$$

여기서  $RC_t(x,y)$ 는 시간  $t$ 에서의 대강의 윤곽선 영상(Rough Contour)이며,  $th_2 (= 128)$ 는 임의의 경계값(threshold)이다. 그림 7은 임의의 이동 객체에 대한 배경 Edge를 제거하기 전의 윤곽선과 제거 후 최종적으로 검출 윤곽선을 보인 것이다.



(a) 배경 Edge 제거 전 (b) 배경 Edge 제거 후

그림 7 최종 윤곽선 검출

### 바. 후처리

최종적으로 검출된 윤곽선에는 주변에 자신을 제외한 윤곽선 픽셀이 존재하지 않은 노이즈 픽셀이 존재하며 후처리 과정에서 이러한 노이즈 픽셀을 제거한다.

## 3. 실험결과

제안된 방법은 배경 차 방법 [2]과 성능을 비교하여, 배경 영상을 생성하지 않고도 움직이는 물체의 윤곽선을 효율적으로 검출할 수 있음을 보였다. Intel Core2 2.4 GHz 환경에서 Visual C++ 2008 을 이용하여 구현하였으며, 실험 영상은 320x240 크기의 영상에 대해 다양한 실외에서 환경에서 촬영된 영상을 사용하였다.

실험을 통하여 배경 영상을 이용한 방법의 경우 물체가 오랜 시간 정지해 있다가 갑자기 움직이는 경우 잔상(ghost)이 남는 반면, 제안한 방법은 잔상이 남지 않음을 볼 수 있었다.

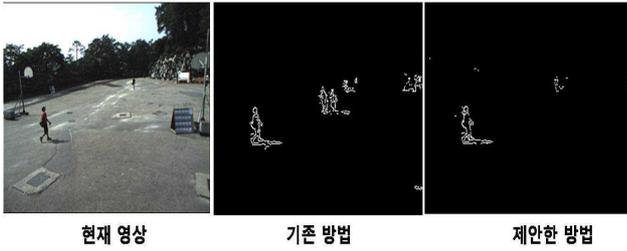


그림 8 잔상 효과에 따른 검출 결과 비교

그림 8은 잔상 효과에 배경 영상을 이용한 방법과 제안한 방법과의 비교를 보여준다. 그림 8에서 현재 영상에 움직이는 사람이 한명 있다. 그러나 배경 영상을 활용한 경우 오랜 시간동안 중심에 머물러 있던 사람이 갑자기 사라져서 중심에 잔상이 남아 있고 이 때문에 마치 가운데 사람이 있는 것처럼 윤곽선이 검출된다. 그러나 제안한 방법은 정확하게 윤곽선을 추출함을 볼 수 있다.

배경 영상을 이용하는 방법은 신뢰성 있는 배경을 추출하는 것이 알고리즘 성능을 좌우하며, 이를 위해 많은 양의 연산을 필요로 한다. 즉 급격한 조명의 변화나 여타의 다른 외부 환경의 변화에 강인한 배경을 유추하기 위해 많은 부분들을 고려하기 때문에 부가적인 연산들이 추가되어 상당히 많은 연산 량을 요구한다. 그러나 제안한 방법은 별도의 배경 영상을 생성하지 않고 카메라로부터 0, t-2, t 시간에 입력되는 세장의 영상만을 사용 사용하기 때문에 연산 시간이 매우 빠르다. 따라서 실시간으로 응용이 가능하다. 표 2는 각 방법에 대한 연산 시간을 비교한 것으로 1000개의 영상에 대한 평균 연산 시간을 비교한 것이다. 표 2에서 보듯이 제안한 방법이 배경 영상을 사용하는 방법에 비해 2배 빠른 연산 시간을 보였다.

표 1 연산 시간 비교

	영상 당 연산 시간 (ms)
기존 방법	13.91
제안한 방법	7.68

#### 4. 결론

본 논문에서는 Spatial Edge와 Temporal Edge를 이용한 이동 객체 윤곽 검출 방법을 제안 하였다. 새로운 방법의 Temporal Edge를 제안하여 이동 객체의 윤곽선이 존재하는 후보 영역을 검출 하였으며, Canny Edge를 이용하여 Spatial Edge를 검출하였다. 6개의 요소 점을 기준으로 한 새로운 형태의 Temporal Edge는 연속적으로 입력되는 영상에서 시간에 따른 픽셀 경계 변화 정보를 효과적으로 검출할 수 있었다. 제안한 방법은 기존에 배경 영상을 활용하는 기법이 가진 잔상 현상의 문제점을 해결하였다. 뿐만 아니라 배경 영상을 생성하지 않기 때문에 많은 배경 영상의 신뢰성에서 생길 수 있는 문제점들을 원천적으로 해결 하였으며, 수행속도가 빨라 실시간 응용 프로그램에 적용이 가능하다.

[1] 문기영, 조현숙, “지능형 영상보완 기술 현황 및 동향”, 전자통신동향분석, 제23권, 제4호, 2008

[2] I Haritaoglu, D. Harwood, and L Davis, “W4: A real time surveillance of people and their activities,” Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, pp. 809-830, 2000.

[3] A Elgammal, D Harwood, and L. Davis, “Non-parametric model for background subtraction,” Proc. 6th European Conference on Computer Vision, Dublin, Ireland, 2000

[4] 이진형, 김용철, “배경 차이법과 텍스처 분할을 이용한 차량 검출,” 제어·자동화·시스템공학회지, 제6권, 제4호, pp. 20-27, 2000

[5] 남석우, 오해석, “동영상에서 배경 영상을 이용한 차량 영상 검출,” 정보처리학회논문지B, 제10-B권, 제6호, pp. 705-710, 2003

[6] Yankun Wei and Wael Badawy, “A new moving object contour detection approach,” Proc. IEEE International Workshop on Computer Architectures for Machine Perception (CAMP), pp. 231-236, 2003

[7] 김보경, 전병민, “GVF 스테이크를 이용한 윤곽선 추출,” 한국콘텐츠학회/한국통신학회 추계 종합학술대회, 제1권, 제2호, pp313-317, 2003

[8] Canny, J., “A Computational Approach To Edge Detection,” Proc. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 8, pp. 679-714, 1986.