

일반논문-10-15-4-02

# 고정 카메라에서의 시공간적 경계 정보를 이용한 이동 객체 윤곽선 검출 방법

곽재호<sup>a)</sup>, 김희율<sup>a)†</sup>

## Moving Object Contour Detection Using Spatio-Temporal Edge with a Fixed Camera

Jae-Ho Kwak<sup>a)</sup> and Whoi-Yul Kim<sup>a)†</sup>

### 요 약

본 논문에서는 시간적, 공간적 경계 정보를 이용한 이동 객체의 윤곽선 검출 방법을 제안한다. 일반적으로 이동 객체의 경우 객체의 윤곽선(contour)을 구성하는 경계(edge) 픽셀에서 시간 축과 공간 축을 중심으로 큰 변화량(gradient)을 가진다. 따라서 시간 축과 공간 축을 중심으로 변화량이 큰 경계 픽셀을 구하면 이동 객체의 윤곽선을 검출할 수 있다. 본 논문에서는 임의의 픽셀에 대하여 시간 축을 중심으로 한 경계 정보를 구하기 위해 Temporal Edge라는 새로운 형태의 변화량 계산 방법을 제안한다. Temporal Edge는 시간  $t$  와,  $t-2$ 에서 입력된 두 그레이 스케일 영상의 차를 시간 축을 기준으로  $x$ 방향,  $y$ 방향으로 Sobel Mask를 적용하여 구한다. 검출된 Temporal Edge를 이용하여 이동 객체의 윤곽선이 존재하는 후보 영역을 검출하고, 검출된 후보 영역을 중심으로 공간적 경계 정보를 구하여 이동 객체의 대강의 윤곽선을 검출한다. 후처리 과정에서 검출된 대강의 윤곽선으로부터 배경 경계와 노이즈 픽셀을 제거한 후 최종적으로 이동 객체의 윤곽선을 검출한다. 제안한 방법은 기존의 배경 차 방법과는 다르게 별도의 배경 영상을 만들지 않기 때문에 배경 차 방법이 가지는 문제점을 극복하였으며, 빠른 연산 속도로 실시간 적용이 가능하다. 실험을 통하여 야간에도 강인한 윤곽선을 검출할 수 있음을 확인하였고, 엔트로피 방법과의 비교를 통해 제안하는 방법의 우수성을 보였다.

### Abstract

In this paper, we propose a new method for detection moving object contour using spatial and temporal edge. In general, contour pixels of the moving object are likely present around pixels with high gradient value along the time axis and the spatial axis. Therefore, we can detect the contour of the moving objects by finding pixels which have high gradient value in the time axis and spatial axis. In this paper, we introduce a new computation method, termed as temporal edge, to compute an gradient value along the time axis for any pixel on an image. The temporal edge can be computed using two input gray images at time  $t$  and  $t-2$  using the Sobel operator. Temporal edge is utilized to detect a candidate region of the moving object contour and then the detected candidate region is used to extract spatial edge information. The final contour of the moving object is detected using the combination of these two edge information, which are temporal edge and spatial edge, and then the post processing such as a morphological operation and a background edge removing procedure are applied to remove noise regions. The complexity of the proposed method is very low because it dose not use any background scene and high complex operation, therefore it can be applied to real-time applications. Experimental results show that the proposed method outperforms the conventional contour extraction methods in term of processing effort and a ghost effect which is occurred in the case of entropy method.

Keyword : Spatio-temporal edge, Contour detection, Moving object detection

## I. 서론

최근 늘어나는 강력 범죄로 인하여 보안에 대한 요구가 증가함에 따라 CCTV나 카메라와 같은 영상 장비를 활용한 지능형 영상 보안 시스템의 개발이 중요한 연구 분야로 대두되고 있다<sup>[1]</sup>. 지능형 영상 보안 시스템이란 영상 장비로부터 입력되는 영상으로부터 이동 객체의 정보를 자동으로 추출하고, 이 정보를 바탕으로 이동 객체의 움직임 정보나 이동 경로를 자동으로 검출하여 비정상적인 행위에 대해서 정보를 알려 줌으로써 인간을 대신하여 감시 역할을 수행할 수 있는 자동화 된 시스템을 말한다. 지능형 영상 보안 시스템의 가장 기본이 되는 기능은 영상 장비로부터 입력되는 영상에서 자동으로 이동 객체를 검출하는 것이다. 이동 객체의 검출이 선행된 후에 움직임 정보나 이동 경로에 대한 분석이 가능하다. 따라서 이동 객체의 검출은 지능형 영상 보안 시스템에 가장 핵심이 되는 기술 요소이며, 이동 객체의 검출 성능에 따라 시스템의 성능이 좌우된다.

일반적으로 카메라에서 연속적으로 입력되는 이미지 시퀀스에서 이동 객체를 검출하는 방법에는 배경 영상을 이용한 방법<sup>[2-4]</sup>이 주로 이용된다. 배경 영상을 이용한 방법이란 카메라로부터 연속적으로 입력되는 영상에서 배경에 해당하는 영상을 유추하고, 이를 활용하여 현재 시간  $t$ 에서 입력되는 영상과의 차를 구하여 이동 객체를 검출하는 방법이다. Hariataoglu<sup>[2]</sup>는 매 시간 입력되는 영상에서의 각 픽셀 값에 대한 평균과 분산 값을 이용하여 배경 영상을 생성 하였으며, Sauffer<sup>[3]</sup>는 픽셀 값을 가우시안 모델을 이용하여 모델링 하여 배경 영상을 생성하였다. Elgammal<sup>[4]</sup>은 커널 밀도 추정(Kernel Density Estimation) 기법을 이용하여 매 시간 입력되는 영상에서의 픽셀 값들에 대한 확률적 분포를 구하여 배경을 생성 하였다. 배경 영상 차 방법은 카메라로부터 연속적으로 입력되는 영상에서 유추한 배경을 기반으로 움직이는 객체를 검출하기 때문에 신뢰성 있는 배경 영상을 생성하는 것이 가장 큰 관건이라고 할 수

있다. 외부 조명의 변화가 심하지 않은 상황에서 배경 영상이 잘 유추 되었을 경우 이 방법은 뛰어난 검출 결과를 보인다. 그러나 자연광이나 인공조명등의 외부 요인에 의해 갑작스러운 조명 변화가 있을 경우 신뢰성 있는 배경 영상이 생성되기 어렵기 때문에 검출 성능이 저하된다는 단점이 있다. 신뢰성 있는 배경을 유추하기 위해서는 다양한 형태의 부가적인 연산이 포함되게 되고 결국 많은 연산 량을 필요로 하게 되므로 속도가 느려진다는 단점도 있다. 또한 배경 영상 유추 단계에서, 장시간 정지 상태에 있는 객체의 경우 배경 영상으로 간주되어 배경 영상이 만들어 지며, 이 객체가 다시 움직일 경우 객체가 배경으로 모델링 된 부분이 잘못 검출되는 잔상효과가 나타난다. 잔상효과는 배경 차이 기법이 가지고 있는 가장 큰 문제 중에 하나이다.

배경 영상을 사용하지 않고 이동 객체를 검출하는 방법으로는 시공간적 경계정보<sup>[5-10]</sup>를 활용하는 방법이 있다. 이 방법은 카메라로부터 입력되는 이미지 시퀀스에서 시공간적 경계정보를 추출하고 이를 중심으로 이동 물체를 검출하는 방법이다. 배경 영상을 활용하는 방법과는 달리 시공간적 경계 정보만을 추출하기 때문에 별도의 배경 이미지를 만들지 않아 배경 영상법이 가지고 있는 단점을 효율적으로 극복할 수 있다. 본 논문에서는 이러한 장점을 활용하여 시공간적 경계정보를 이용한 이동물체 검출 방법을 제안한다. 본 논문에서는 임의의 픽셀에 대하여 시간 축을 중심으로 한 시간적 경계 정보를 구하기 위해 Temporal Edge라는 새로운 형태의 변화량 계산 방법을 제안한다. Temporal Edge는 시간  $t$ 와,  $t-2$ 에서 입력된 두 그레이 스케일 영상의 차를 시간 축을 기준으로 x방향, y방향으로 Sobel Mask를 적용하여 구한다. 제안된 Temporal Edge는 알고리즘의 초기 단계에서 이동 객체의 윤곽선이 존재하는 후보 영역을 검출하는데 활용된다. 검출된 후보 영역에 속하는 픽셀들을 중심으로 공간적 경계 정보를 추출하고, 이로부터 이동 객체의 대강의 윤곽선을 검출한다. 그러나 이렇게 검출된 대강의 윤곽선에는 우리가 원하지 않는 배경 경계와 노이즈 성분이 포함되어 있다. 따라서 모폴로지(morphology) 연산과 배경 에지 제거법과 같은 후처리 과정을 통해 노이즈를 제거한 후 최종적인 이동 객체의 윤곽선을 검출한다.

본 논문의 구성은 다음과 같다. 제 II장에서는 같은 관점

a) 한양대학교 전자컴퓨터통신공학과  
 Hanyang University, Department of Electronics Computer Engineering  
 † 교신저자 : 김희율(wykim@hanyang.ac.kr)  
 \*접수일(2009년12월28일), 수정일(2010년6월30일), 게재확정일(2010년7월12일)

에서 문제에 접근한, 시공간적 경계정보 검출에 대한 기존 논문들을 고찰하고, 제III장에서는 Temporal Edge를 포함한 제안한 알고리즘을 설명한다. 제IV장에서는 기존 방법과의 비교 실험을 통해 제안한 방법의 성능을 분석하고, 제V장에서는 결론을 맺는다.

## II. 관련 연구

시공간적 경계 정보 추출은 1989년대부터 많이 연구된 분야이다. 여기에서는 시공간적 경계 정보를 추출하는 다양한 방법들에 대해 고찰하고자 한다.

Stein<sup>[5]</sup>은 이미지의 시퀀스를  $x, y, t$  축을 중심으로 3차원 구형 패치로 모델링 하고, 이 구형 패치를 일정한 각도로 나눈 후, 각 구간내의 픽셀들에 대한 픽셀 값(intensity)에 대한 각도의 변화량을 히스토그램으로 구하여 시공간 경계정보를 추출하였다. 이미지 상에서 물체가 움직일 경우  $x, y$  축을 중심으로 공간상의 변화가 발생하고 이 변화는  $x$  축에 대한  $y$  축의 각도의 변화  $\theta_s$ 로 나타내었으며, 이를 공간상의 변화정보로 활용하였다. 동일한 개념을 시간 축  $t$ 에 적용하여 시간의 흐름에 따라 물체가 이동하면 이미지 시퀀스 상에서 물체의 이동 정보는  $x$  축을 중심으로 시간 축  $t$ 의 각도 변화  $\theta_t$ 로 나타내었다. 이 각도의 변화량은 물체의 이동 속도와도 연관성이 높다. 즉 물체의 이동 속도가 빠를 경우 각도의 변화가 크게 나타나고, 이동 속도가 작을 경우 각도의 변화가 작게 나타난다. Stein 기준 픽셀  $(x, y, t)$ 를 중심으로 기준 픽셀이 포함된 구형의 패치 분절(구형 패치가 일정 간격으로 나뉜 부분)내의 픽셀에 대한  $(\theta_s, \theta_t)$ 쌍으로 시공간적 경계정보를 표현하였다.

Kim<sup>[6]</sup>은 엔트로피(entropy)와 크로스엔트로피(cross-entropy) 개념을 이용하여 시공간적 경계정보를 추출하였다. 임의의 픽셀을 중심으로 주변 픽셀과의 밝기 변화량을 엔트로피의 개념을 이용하여 구하고 이를 공간적 경계 정보로 활용하였다. 또한 시간  $t$ 에서의 엔트로피와 시간  $t-1$ 에서의 엔트로피간의 상호 관계를 크로스엔트로피 개념을 이용하여 정의한 후 이를 공간적 변화량으로 활용하였다. 이 두 변화량의 조합으로 시공간적 경계 정보를 추출하였다.

Chen<sup>[7]</sup>은 piecewise planar (linear) function을 이용하여 이미지를 모델링 하고 이를 기반으로 이미지를 픽셀의 값에 대해 변화량이 큰 부분과 변화량이 작은 부분으로 구분하고 변화량이 큰 부분을 중심으로 시간적 경계정보를 추출하였다. 이 방법은 시간 축을 이용하여 분석할 때 흔히 이용되는 옵티컬플로워(optical flow)를 구하지 않고, 단지 시간의 순서대로 입력되는 이미지 시퀀스의 2차원 미분을 통하여 시간 축  $t$ 를 기준으로 이차원 미분한 값이 제로(zero)가 아닌 위치를 중심으로 이동 물체의 경계를 추출하였다.

Suzuki<sup>[8]</sup>는 가우시안 함수의 이차 도함수를 이용하여 이동 물체의 시간적 경계 정보를 추출하였다. 가우시안 함수의 이차 도함수를 시간  $t$ 와  $t-1$ 인 이미지에 적용하여 두 값의 곱이 음수인 부분을 픽셀 변화량이 발생한 부분이라고 간주하고 이를 중심으로 시간적 경계정보를 검출하였다. 이 시간적 경계정보는 시간 축을 중심으로 한 이동 객체의 윤곽선 정보가 된다.

Duncan<sup>[9]</sup>은 Suzuki와 마찬가지로 가우시안 함수의 이차 도함수를 이용하여 시간적 경계 정보를 추출하였다. 그러나 Suzuki와는 달리 Duncan은 가우시안 함수의 2차 도함수를 시간  $t$ 에 입력되는 이미지  $f_t$ 와 시간적, 공간적 축을 기준으로 컨볼루션(convolution)시켜 이들의 상관관계를 통해 시공간적 경계정보를 추출하였다.

Heyden<sup>[10]</sup>은 시공간적 경계 정보를 활용한 이미지 분할 방법을 제안하였다. Heyden은 Prewitt Gradient를 활용하여 공간적 경계 정보를 추출하고, 이 때 발생하는 공간적 틈(gaps)을 시간적 경계 정보를 활용하여 보완하였다. 연속적으로 입력되는 이미지의 상호 관계를 통하여 시간적 경계정보를 추출하고, 경계 개선(edge enhancement), 경계 연결(edge linking)등의 기법을 통해 이를 개선하였다. 개선된 시간적 경계정보는 공간적 경계정보의 틈을 보완하는데 활용되었고 이러한 시공간적 경계정보의 상호 조합을 통해 이미지 분할을 효과적으로 구현하였다.

## III. 이동 객체 윤곽선 검출 방법

본 논문에서 제안하는 이동 객체 윤곽선 검출 방법은 현

제 시간  $t$ 에서 입력되는 영상  $f_t$ 만을 사용한다는데 그 특징이 있다. 배경 영상을 생성하는 기존 방법의 문제점을 극복하기 위해 배경 영상을 생성하지 않고,  $f_t$  영상만으로 이동 객체의 윤곽선을 검출한다. 본 논문에서 제안하는 이동 객체의 윤곽선 검출 방법은 크게 세 가지 부분으로 나누어진다. 첫 번째 부분은 현재 시간  $t$ 에서 입력되는 영상  $f_t$ 를 이용하여 이동 객체의 윤곽선이 존재하는 후보 영역을 검출하기 위해 Temporal Edge를 구하는 단계이다. 두 번째 부분은 구해진 Temporal Edge를 기반으로 공간적 경계 정보를 추출하여 이동 객체에 대한 대강의 윤곽선을 검출하는 단계이다. 마지막으로 대강의 윤곽선으로부터 배경 경계와 노이즈를 제거하기 위한 후처리 과정이다. 그림 1은 본 논문에서 제안하는 알고리즘의 전체 흐름도를 나타낸 것이다. 그림 1에서 원안에 있는 - 기호는 두 이미지 사이의 차(subtraction)를, ^는 두 영상의 공통부분, 즉 AND 연산을 의미한다.

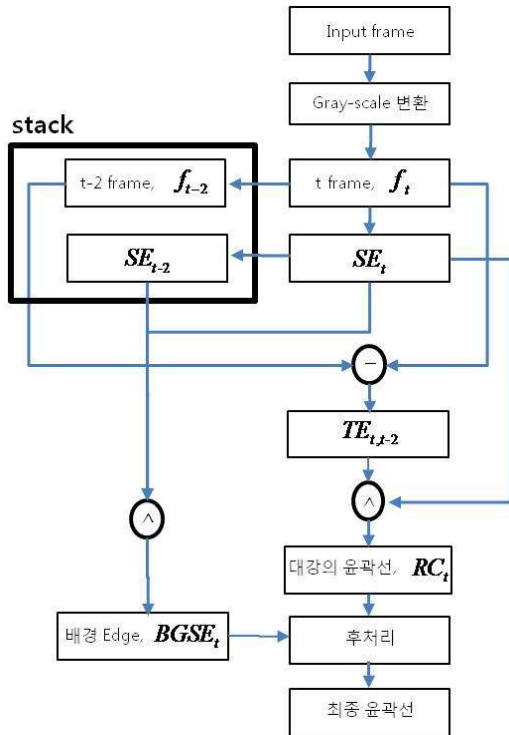


그림 1. 전체 알고리즘 흐름도  
Fig. 1. Flowchart for the proposed algorithm

### 1. Temporal Edge 검출

이동 객체의 윤곽선을 검출하기 위해 우선 윤곽선이 존재하는 후보 영역을 먼저 검출한다. 후보 영역을 먼저 검출하는 이유는 윤곽선을 검출하기 위해 영상 전체를 검색하기 보다는 윤곽선이 존재하는 후보 영역을 먼저 검출하고 그 후보 영역에서 윤곽선을 검출함으로써 연산량을 줄여 속도를 빠르게 할 수 있기 때문이다. 이동 객체의 윤곽선이 존재하는 후보 영역을 검출하기 위해 시간 축을 중심으로 한 새로운 형태의 변화량 계산법인 Temporal Edge를 제안한다. Temporal Edge는 Spatial Edge와 대응되는 용어로서 기존에 공간상에서 에지 연산자를 이용하여 추출한 경계 정보를 Spatial Edge라고 하면, 이와 유사한 개념으로 시간 상에서 에지 연산자를 이용하여 추출한 경계 정보를 Temporal Edge라 칭하기로 한다. 공간상에서 경계를 구할 경우 영상 상에서 각 픽셀을 중심으로 그림 2와 같은 Sobel Mask를 이용하여 구한다.

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}, \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

그림 2.  $x$  축,  $y$  축 방향에 대한 Sobel Mask  
Fig. 2. Sobel Mask for the  $x$  and  $y$  directions

Temporal Edge  $TE_{f_t, f_{t-2}}$ 는 Sobel Mask를 그림 3과 같이 시간 축을 중심으로  $x$  방향,  $y$  방향으로 적용하여 구한다. 이를 식으로 표현하면 수식 (1)과 같다.

$x > 1, y > 1$ 에 대해서

$$TE_x(x, y) = (f_t(x-1, y-1) - f_{t-2}(x-1, y-1)) + 2(f_t(x-1, y) - f_{t-2}(x-1, y)) + (f_t(x-1, y+1) - f_{t-2}(x-1, y+1))$$

$$TE_y(x, y) = (f_t(x-1, y-1) - f_{t-2}(x-1, y-1)) + 2(f_t(x, y-1) - f_{t-2}(x, y-1)) + (f_t(x+1, y-1) - f_{t-2}(x+1, y-1)) \quad (1)$$

$$TE_{f_t, f_{t-2}} = \sqrt{TE_x^2 + TE_y^2}$$

$$\theta = \arctan\left(\frac{TE_x}{TE_y}\right)$$

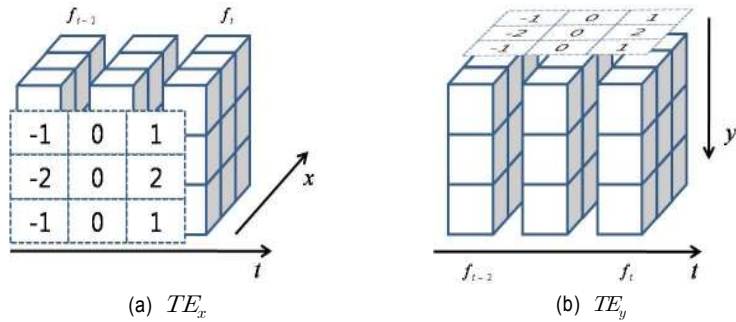


그림 3. Sobel Mask를 이용한 Temporal Edge 검출  
Fig. 3. Temporal Edge extraction using Sobel Mask

그림 4는 수식 (1)을 이용하여 구한 Temporal Edge의 예이다. 그림 4에서 보듯이 Temporal Edge는 환형(ring shape)을 나타낸다. 그 이유는 Temporal Edge가 시간 축을 중심으로 픽셀의 변화량을 계산한 값이기 때문이다. 즉 이동 객체의 경우 객체의 경계에 해당하는 부분의 경우 시간 축을 중심으로 큰 변화량을 보이는 반면 몸통에 해당하는 부분은 작은 변화량을 보인다. 따라서 이동 객체의 윤곽선이 존

재 하는 주변으로는 강한 Temporal Edge가 나타나지만, 그렇지 않는 부분에서는 Temporal Edge가 약하거나 나타나지 않게 된다.

## 2. Spatial Edge 검출

카메라로부터 연속적으로 입력되는 매 영상마다 Canny

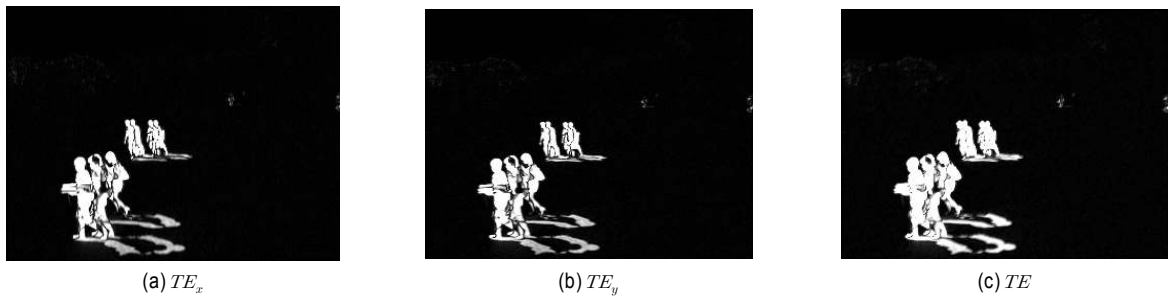


그림 4. Temporal Edge의 예  
Fig. 4. An example of Temporal Edge

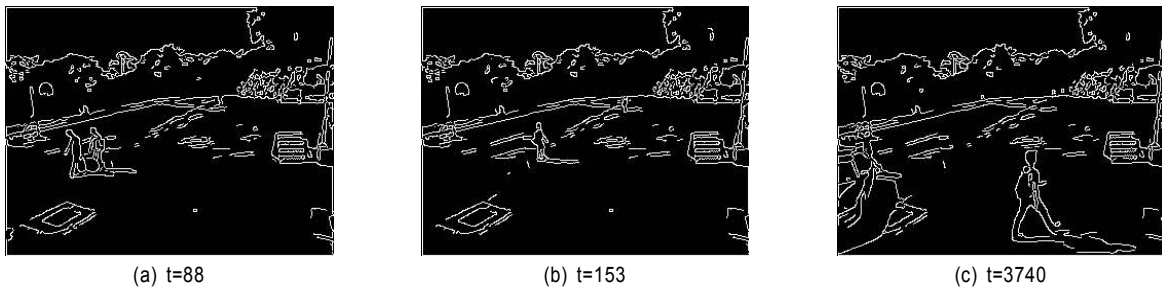


그림 5. Canny edge detector를 이용한 Spatial Edge의 예  
Fig. 5. Examples of Spatial Edge using Canny edge detector

Edge<sup>[11]</sup>를 검출하여 이를 Spatial Edge 정보로 활용한다. 그림 5는 검출된 Spatial edge 정보  $SE_t$ 의 예를 보인 것이다. 이렇게 추출된 Spatial Edge 정보는 Temporal Edge 정보와 함께 조합되어 이동 객체의 윤곽선을 검출 하는데 활용된다. 뿐만 아니라 후처리 과정을 위해 배경 경계 정보를 만드는 데도 활용된다.

### 3. 대강의 윤곽선 검출

III.2에서 검출한 Spatial Edge 정보  $SE_t$ 로부터 이동 객체의 윤곽선을 검출하기 위해 III.1에서 추출한 Temporal Edge  $TE_{f_t, f_{t-2}}$ 를 활용한다. 서론에서 우리는 이동 객체의 윤곽선을 구성하는 픽셀의 경우 시간적, 공간적 축을 중심으로 높은 변화량(gradient)을 가진다고 가정했었다. III.1에서 구한 Temporal Edge는 시간 축을 중심으로 변화량을 구한 값이고 III.2에서 구한 Spatial Edge는 공간 축을 중심으로 변화량을 구한 값이다. 따라서 이 두 값이 동시에 높은 픽셀들을 추출하면 이동 객체의 윤곽선을 검출 할 수 있다. 본 논문에서는 Temporal Edge를 이동 객체의 윤곽선이 존재하는 후보 영역으로 삼아, Temporal Edge에 속하는 임의의 픽셀을 중심으로 그 주변(본 논문에서는 5x5 window 영역 내)에 존재하는 Spatial Edge 픽셀을 검출하여 이동 객체에 대한 대강의 윤곽선(rough contour)  $RC_t$  검출한다. 여기서 “대강의”라고 명시한 이유는 이 단계에서 검출된 윤곽선을 후처리 과정을 거친 최종 윤곽선과 구분하기 위해서이다. 이 단계에서 검출된 윤곽선은 최종 윤곽선과는 달리

우리가 원하지 않은 노이즈와 배경 경계 픽셀을 포함한다. 따라서 이러한 노이즈와 배경 경계 픽셀을 제거하기 위해 후처리 과정이 필요하다. 표 1은 대강의 윤곽선  $RC_t$ 를 구하기 위한 Pseudo-code 이다.

표 1. 대강의 윤곽선  $RC_t$ 를 구하기 위한 Pseudo-code  
Table 1. Pseudo-code for extraction rough contour  $RC_t$

```

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 ) {  $RC_t(x+c, y+r) = se$ ; }
    }
}
    
```

그림 6은 이동 객체에 대한 대강의 윤곽선 검출 결과이다. 서론에서 우리는, 이동 객체의 윤곽선은 공간적으로나 시간적으로 변화량이 큰 부분에 존재 한다고 가정했었다. 그림 6에서 보듯이 이동 객체에 대한 대강의 윤곽선이 Temporal Edge와 Spatial Edge가 동시에 큰 부분에서 검출됨에 따라 이 가정이 정당함을 확인 할 수 있다. 그러나

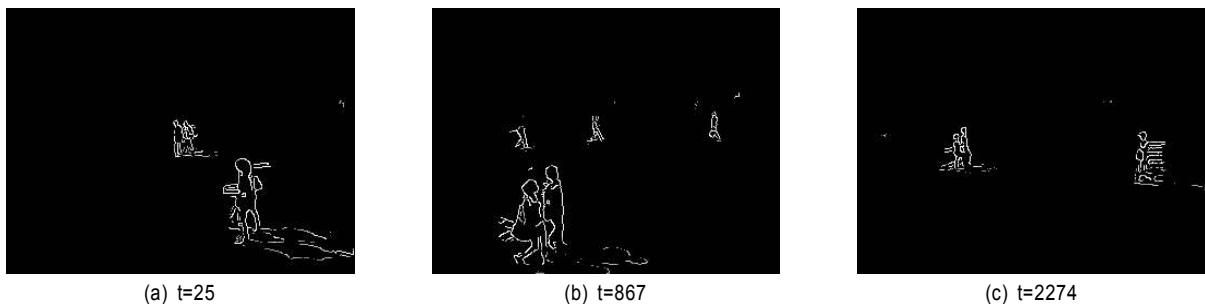


그림 6. 이동 객체에 대한 대강의 윤곽선 검출  
Fig. 6. Rough contour extraction results of moving object

그림 6의 결과에는 우리가 원하지 않는 노이즈 픽셀이 많이 포함되어 있다. 이는 후보 영역  $TE_{f_t, f_{t-2}}$  픽셀을 중심으로 주변 영역(5x5 window)에 있는 Spatial Edge 픽셀들을 고려하다보니 배경 경계가 포함되어 검출되었기 때문이다. 따라서 배경 경계를 제거하기 위한 후처리 과정이 필요하다.

#### 4. 배경 경계 정보 생성

III.3에서 구한 대강의 윤곽선  $RC_t$ 에는 이동 객체의 실제 윤곽선뿐만 아니라 우리가 검출하기를 원치 않는 배경경계 픽셀과 노이즈 픽셀이 포함되어 있다. 따라서 이를 제거하기 위한 후처리 과정이 필요하다. 배경 경계를 제거하기 위해서는 배경 경계 정보를 알아야 하며, 이를 위해 배경 경계 정보를 생성할 필요가 있다. 우리는 매 시간  $t$ 에서 입력되는 영상  $f_t$ 로부터 공간적 경계 정보  $SE_t$ 를 검출하고 있다. 배경 경계라 함은 매 시간 입력되는 영상에서 이동 객체와는 별개로 항상 고정되어 나타나는 공간적 경계를 말한다. 따라서 매 시간  $t$ 에서 검출한 공간적 경계 정보  $SE_t$ 상에서 동일한 위치에서 나타날 확률이 높다. 따라서 공간적 경계 정보  $SE_t$ 에서 반복적으로 나타나는 경계 픽셀에 대해 연속적으로 검출되는 횟수를 누적하여 배경 경계 정보  $BGSE_t$ 를 생성한다. 시간  $t-1$ 에서  $SE_{t-1}$ 상에서의 임의의 픽셀이 공간적 경계 픽셀인 경우 시간  $t$ 에서 다시 이 픽셀이 공간적 경계로 나타나게 되면 이는 배경 경계 픽셀일 확률이 높으므로 누적 횟수를 1 증가 시키고 반대로 공간적 경계로 나타나지 않는 경우 누적 횟수를 1 감소시킨다. 결국  $SE_t$ 상에서 연속적으로 공간적 경계로 나타나는 픽셀의 경우 배경 경계 일 확률이 높아져 높은 누적 횟수를 나타내게 되고 그렇지 않은 경우 낮은 누적 횟수를 나타내게 된다. 배경 경계 정보  $BGSE_t$ 는 수식 (2)와 같이 구한다.

$$\begin{cases} count_t(x,y) = count_{t-1}(x,y) + 1 & \text{if } SE_t(x,y) > 0 \wedge SE_{t-1}(x,y) > 0 \\ count_t(x,y) = count_{t-1}(x,y) - 1 & \text{Otherwise} \end{cases} \quad (2)$$

$$\begin{cases} BGSE_t(x,y) = 255 & \text{if } count_t(x,y) > th_1 \\ BGSE_t(x,y) = 0 & \text{Otherwise} \end{cases}$$

여기서  $count_t(x,y)$ 는 시간  $t$ 에서  $x,y$  위치의 임의의 픽셀에 대해 공간적 경계로 나타나는 횟수를 누적한 것이며,  $th_1(=10)$ 는 임의의 경계(threshold) 값이다. 그림 7은 생성된 배경 경계 정보를 보인 것이다.



그림 7. 생성된 배경 경계 정보  
Fig. 7. Estimated background edge information

#### 5. 후처리

III.3 절에서 구한 대강의 윤곽선  $RC_t$ 에는 우리가 원하지 않는 배경 경계 픽셀과 노이즈 픽셀이 포함되어 있다. 따라서 배경 경계 픽셀 제거와 노이즈 픽셀 제거를 위한 후처리 과정이 필요하다. III.4절에서 생성한 배경 경계 정보를 이용하여 대강의 윤곽선으로부터 배경 경계 픽셀에 해당하는 픽셀을 제거하여 최종적인 이동 객체의 윤곽선을 검출한다. 또한 주변에 이동 객체의 윤곽선이 존재하지 않는 점 픽셀을 노이즈 픽셀로 간주하고 이를 제거한다. II.3절에서 구한 대강의 윤곽선  $RC_t$ 내에 존재하는 픽셀 중 배경 경계 정보  $BGSE_t$ 와 동일한 위치에 존재하는 픽셀을 배경 경계 픽셀로 간주하고 이를 제거한다. 이를 식으로 표현하면 수식 (3)과 같다.

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

여기서  $RC_t$ 는 III.3절에서 구한 대강의 윤곽선이며,

$BGSE_t$ 는 시간  $t$ 에서의 배경 경계 정보이고,  $th_2 (= 128)$ 는 임의의 경계(threshold) 값 이다. 그림 8은 III.3 절에서 구한 대강의 윤곽선에 대한 배경 경계 제거 결과 이다.

#### IV. 실험 결과

제안한 방법은 Intel Core2 2.4 GHz 환경에서 Visual C++ 2008을 이용하여 구현하였으며, 실험 영상은 320x240 크기의 다양한 실외 환경에서 촬영된 영상을 사용 하였다. 그림 9는 실험에 사용된 영상들로 다양한 환경에서 촬영된 실외 영상들로 구성되어 있다.

제안한 방법의 검출 성능을 평가하기 위해 Precision 과 Recall을 이용하였다. Precision과 Recall은 정확도를 평가 하는 척도로서 널리 알려진 방법이다. Precision은 검출 결과 가 얼마나 정확한가를 평가하는 척도이며, Recall은 검출 해야 하는 것 중에서 실제로 얼마나 검출 했느냐를 평가하는 척도이다. Precision과 Recall을 구하는 식은 식 (4), (5) 와 같다.

$$Precision = \frac{t_p}{t_p + f_p} \quad (4)$$

$$Recall = \frac{t_p}{t_p + f_n} \quad (5)$$

$t_p$ : 제안된 방법으로 검출된 윤곽선 픽셀 중 GT와 일치(존재)하는 픽셀 수  
 $f_p$ : 제안된 방법으로 검출된 윤곽선 픽셀 중 GT와 일치(존재)하지 않는 픽셀 수  
 $f_n$ : 제안된 방법으로 검출되지 않은 픽셀 중 GT상에 실제 이동 객체의 윤곽선 픽셀인 픽셀 수

여기에서,  $t_p$ 는 true positive,  $f_p$ 는 false positive를 그리고  $f_n$ 는 false negative를 나타낸다. 즉  $t_p$ 는 제안된 방법을 통해 검출된 이동 객체의 윤곽선 픽셀들 중, GT(ground truth) 이미지 상의 실제 이동 객체의 윤곽선 픽셀과 위치가 일치 하는 픽셀의 개수를 말한다. 이때 GT는 그림 9의 각 실험 영상에서 물체의 이동이 있는 하나의 구간에 대해 100장의 연속된 이미지를 추출하고 이를 기반으로 수작업으로 이동 객체의 윤곽선에 해당하는 픽셀을 표시하여 이진(binary) 영상을 만들었다.  $f_p$ 는 제안된 방법을 통해 이동 객체의 윤곽선 픽셀로 검출되었으나 GT 이미지 상에는 실제 이동 객

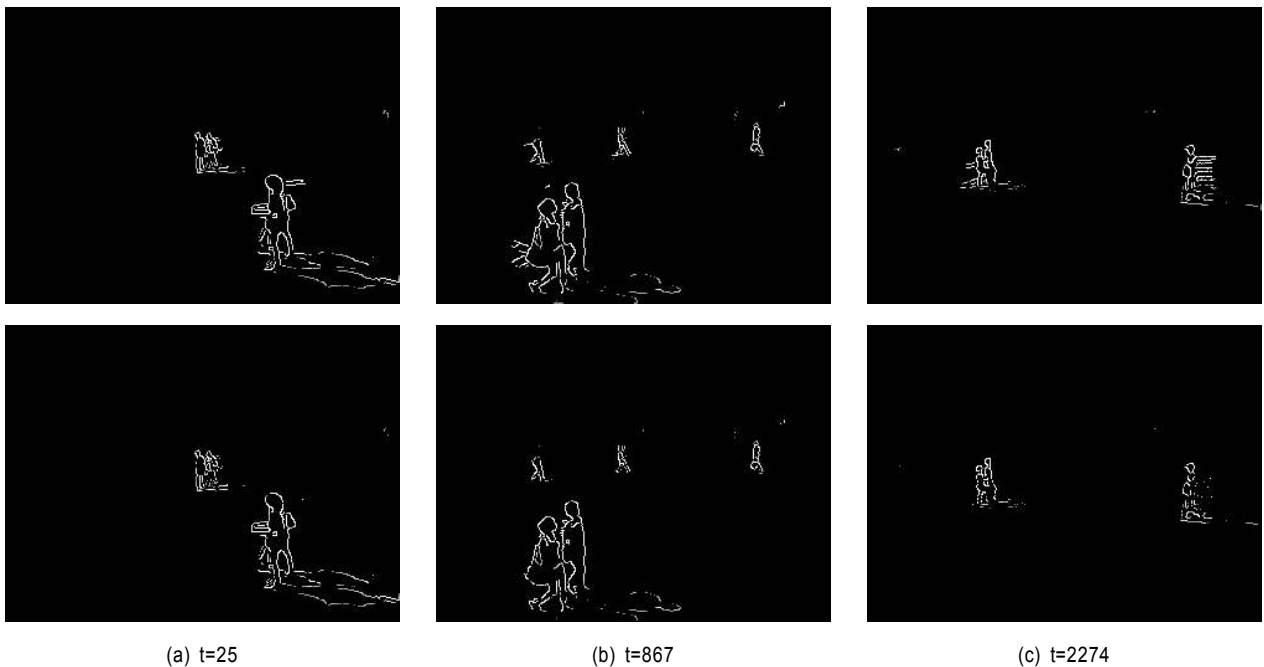


그림 8. 이동 객체의 윤곽선에 대한 배경 경계 제거

Fig. 8. Results of background edge removing on the final contour of the moving object.



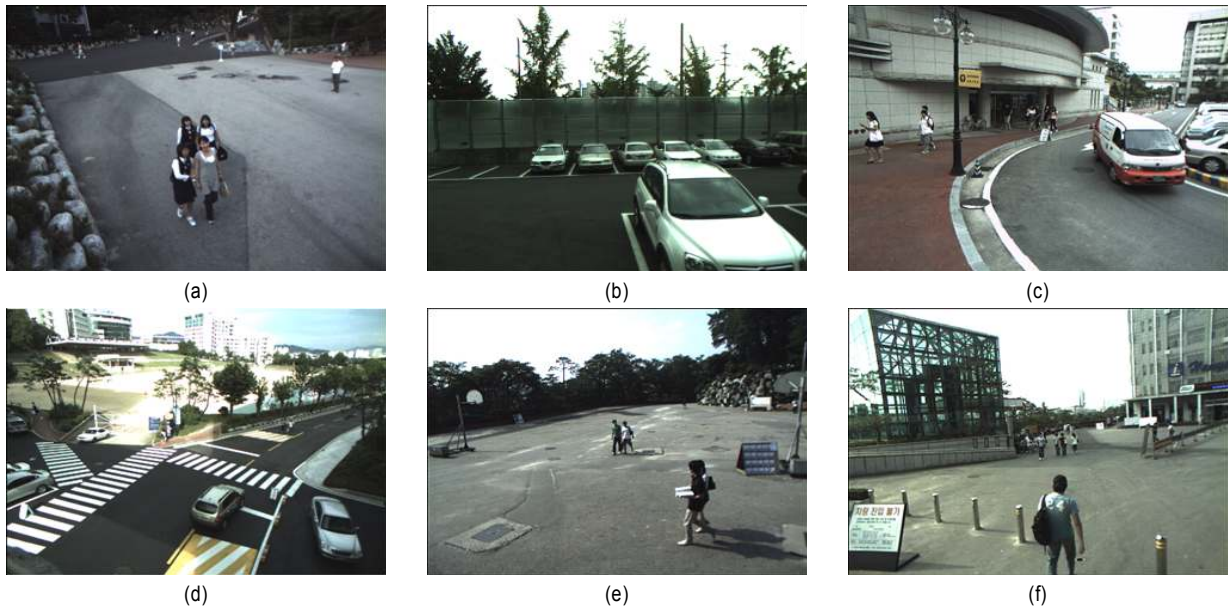


그림 9. 실험에 사용된 실외 영상들  
 Fig. 9. Outdoor scenes used in experiment

체의 윤곽선 픽셀이 아닌 픽셀의 수를 말한다. 마지막으로  $f_n$ 는 제안된 방법으로는 검출되지 않은 픽셀들 중 GT 이미지 상에는 실제 이동 객체의 윤곽선 픽셀인 픽셀의 개수를 말한다. 표 2는 그림 9의 실험 영상에 대한 이동 객체 윤곽선 검출의 정확도에 대한 평균값을 나타낸 것이다. 표 2에서 보듯이 제안한 방법은 다양한 실외 환경에 대해 이동 객체의 윤곽선 검출 정확도가 높음을 볼 수 있다. (c) 비디오의 경우 이동 객체의 뭉침 정도가 많고 이동 객체의 양이 많아 제안한 방법으로 검출한 윤곽선의 검출 정확도가 다소 떨어짐을 볼 수 있다.

표 2. 그림 9의 실험 영상에 대한 윤곽선 검출 정확도에 대한 평균값  
 Table 2. Moving object contour detection average accuracy for Fig. 9

	(a)	(b)	(c)	(d)	(e)	(f)
Precision	0.978	0.944	0.851	0.909	0.911	0.871
Recall	0.901	0.965	0.811	0.923	0.976	0.910

제안한 방법은 밝기의 변화에도 강인하게 이동 객체의 윤곽선을 검출 할 수 있음을 실험을 통해 확인할 수 있었

다. 그림 9의 (e) 비디오 클립에 대해 동일한 장소에서 서로 다른 시간대를 이용하여 밝기의 변화가 다른 실험 영상을 취득하였다. 그림 10은 동일한 장소에서의 다른 시간대에 촬영된 영상에 대한 이동 객체 윤곽선 검출 결과를 보인 것이다.

서론에서 우리는 배경 영상을 사용하는 기존 방법에 대해 잔상효과의 문제점을 지적 했었다. 즉, 배경 영상을 사용하는 기존 방법의 경우 아주 천천히 움직이는 객체나 오랜 시간 정지해 있는 객체는 배경으로 포함되어 모델링 되고 이후 이 객체가 다시 움직일 경우 배경 영상에 잔상으로 남아 있게 되어 이동 객체로 잘못 검출되는 문제가 있다. 그러나 제안한 방법은 별도의 배경 영상을 만들지 않고 현재 시간  $t$ 에서 입력되는 영상  $f_t$ 만을 사용하기 때문에 이러한 잔상효과를 원천적으로 제거할 수 있다. 그림 11은 잔상 효과에 따른 검출 결과를 비교한 것이다. 본 논문에서 제안한 이동 객체 윤곽선 검출 방법의 결과를 검증하기 위해 엔트로피를 이용한 방법<sup>[6]</sup>을 이용하여 성능을 비교한 결과가 그림 12에 나와 있다. 이 결과들을 보면 다른 방법들에 비해 제안한 방법이 이동 객체 윤곽선을 강인하게 검출함

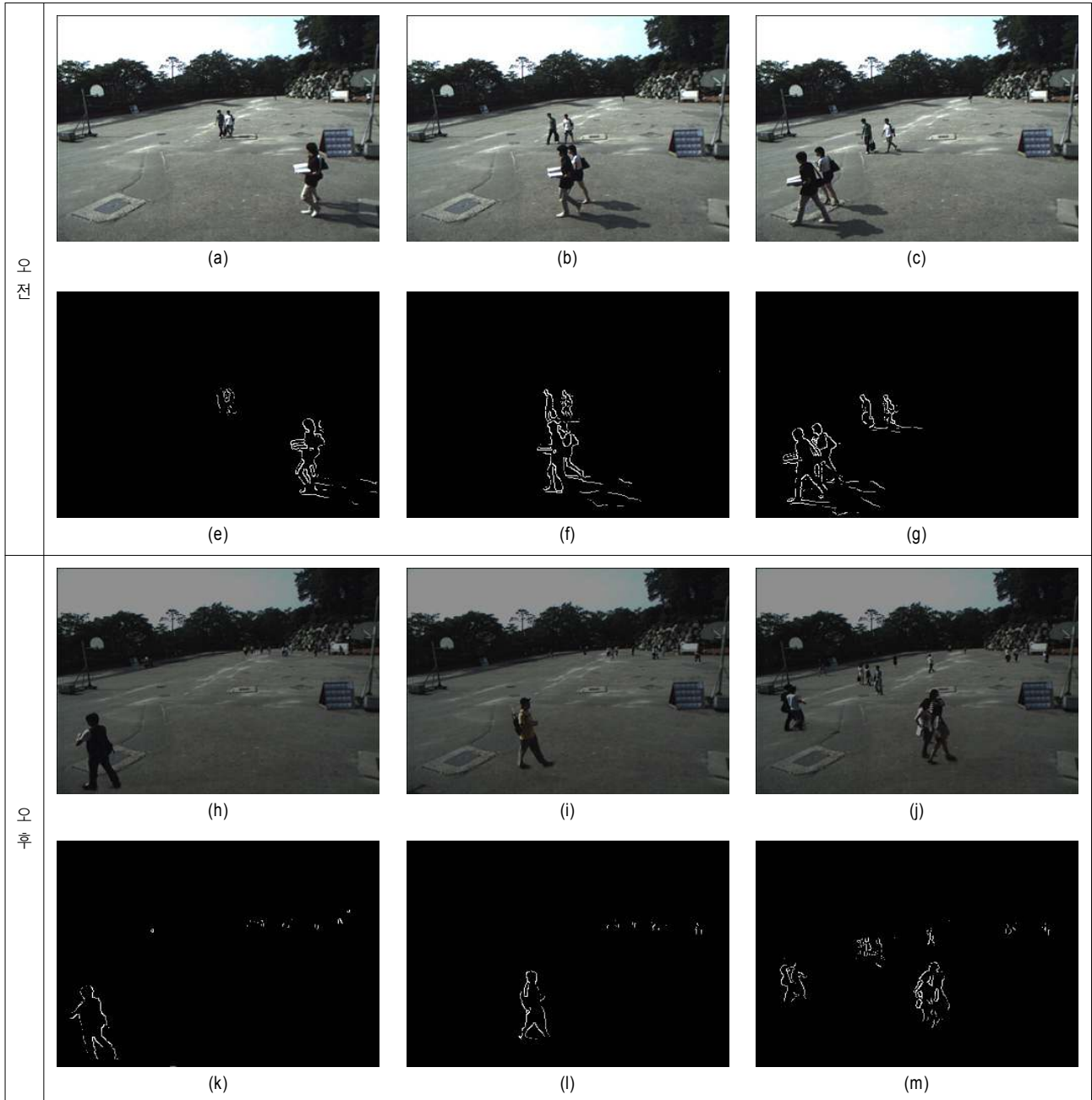


그림 10. 밝기 변화에 따른 이동 객체 검출 결과 비교

Fig. 10. Comparison of contour extraction results with respect to brightness change

을 볼 수 있다. 엔트로피를 이용한 이동 객체 윤곽선 검출 방법은 중심 픽셀 주변의 밝기 변화를 확률적으로 모델링하기 때문에 주변 픽셀의 밝기 변화가 크기 않은 경우에는

윤곽선 검출이 잘 안됨을 볼 수 있다. 그러나 제안하는 방법은 중심 픽셀과 주변 픽셀과의 밝기 차를 3차원으로 고려하기 때문에 이동 객체의 윤곽, 즉 픽셀의 밝기의 변화량이


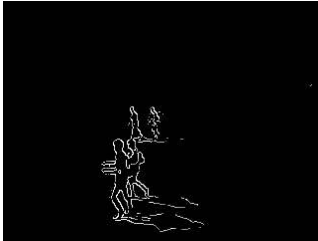
	입력 영상	배경 영상	윤곽선 검출 결과
배경 영상차 방법			
제안한 방법	(a)		

그림 11. 잔상효과에 따른 검출 결과 비교  
 Fig. 11. Comparison of contour extraction results with respect to the ghost effect

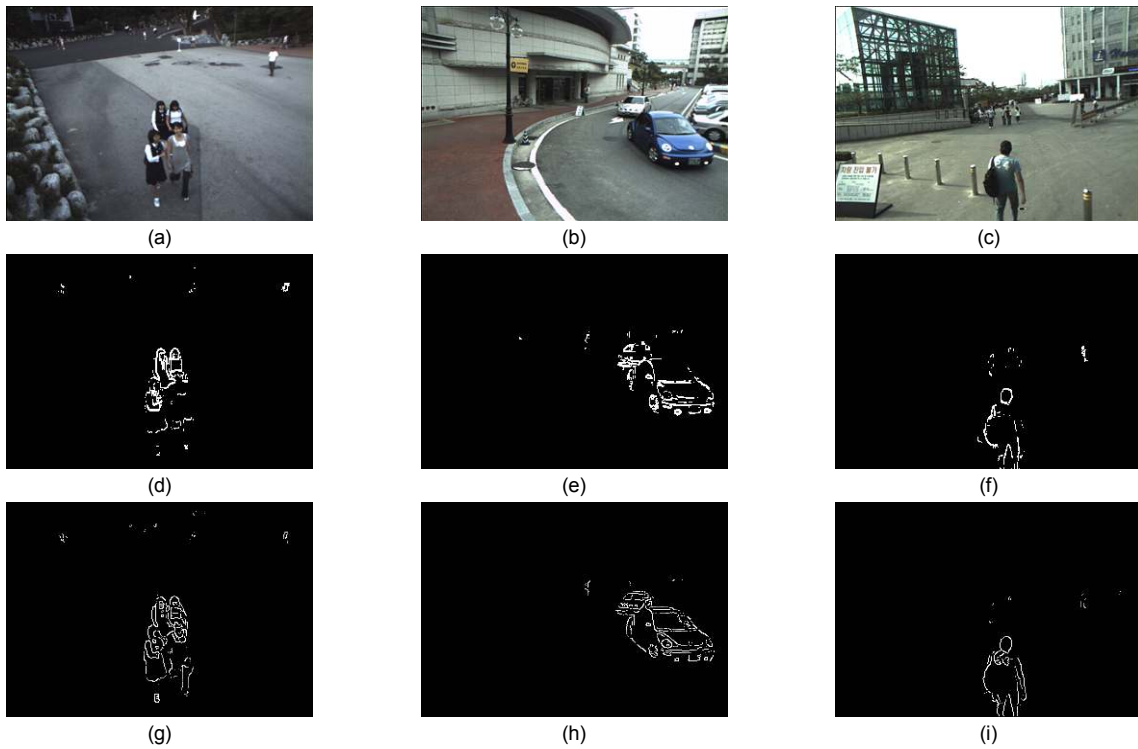


그림 12. 윤곽선 검출 결과 비교: (a)-(c) 원본 영상, (d)-(f) 엔트로피 방법, (g)-(h) 제안한 방법  
 Fig. 12. Comparison of moving objects contour extraction results: (a)-(c) Original images, (d)-(f) Entropy method, (g)-(i) Proposed method

큰 부분에서 강인한 윤곽선이 검출됨을 볼 수 있다.

## V. 결 론

본 논문에서는 시간적, 공간적 경계 정보를 이용한 이동 객체 윤곽선 검출 방법을 제안 하였다. 제안한 방법은 시간  $t$ 와  $t-2$ 에서 입력되는 영상을 이용하여 시간적 경계 정보를 추출하였으며, 시간  $t$ 에서 입력되는 영상에서 Canny Edge를 검출하여 공간적 경계 정보로 활용하였다. 시간적 경계 정보를 추출하기 위해 Sobel Mask를 이용한 새로운 형태의 시간적 경계 정보 검출 방법인 Temporal Edge를 제안하였다. Temporal Edge로부터 검출된 시간적 경계 정보를 이동 객체의 윤곽선이 존재하는 후보 영역으로 사용하였으며, 공간적 경계 정보를 이용하여 최종적인 이동 객체의 윤곽선을 검출 하였다. 시간  $t$ 에서 검출한 시간적 경계 정보를 확률적으로 누적하여 배경 경계 정보를 생성하고 이를 이용하여 최종 윤곽선에 포함되어 있는 배경 경계를 제거 하였다. 다양한 실험 영상을 통해 제안한 방법의 우수성을 검증 하였다. 특히 동일한 장소에서 밝기가 서로 다른 영상을 취득하여 밝기 변화에도 강인하게 이동 객체의 윤곽선을 검출할 수 있었다. 배경 영상을 생성하지 않아 잔상 효과를 제거할 수 있었으며, 엔트로피 방법과의 비교를 통해 제안한 방법이 우수한 것을 확인할 수 있었다.

본 논문은 시간적 경계 정보를 검출하기 위해 새로운 변화량 계산 방법인 Temporal Edge를 제안했다는 점에서 그 의의가 있다고 하겠다. 시간  $t$ 를 중심으로 한 3개의 이미지와 차 영상만으로 구성되어 연산속도가 매우 빨라 실시간 응용이 가능하다. 또한 Temporal Edge를 통해 추출되는 시간 축에 따른 이미지 픽셀의 변화량 정보는 연속적인 이미지에서의 이동 물체를 검출하는 다른 어플리케이션에서 다

양하게 활용이 가능하며, 그 활용도가 높다고 할 수 있다. 그러나 제안한 방법은 고정 카메라만을 고려하여, 카메라가 움직이는 상황에 대해 취약함을 드러낸다. 따라서 카메라가 움직이는 상황에도 이동 물체의 윤곽선을 검출 할 수 있는 연구가 필요하다.

## 참 고 문 헌

- [1] 유장희, 문기영, 조현숙, "지능형 영상보안 기술현황 및 동향", 전자통신동향분석, 제23권, 제4호, pp.80-88, 2008년, 8월
- [2] I. Hariataoglu, D. Harwood, and L. Davis, "W4: A real time surveillance of people and their activities," Proc. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 809-830, 2000
- [3] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," Proc. Computer Vision and Pattern Recognition, pp. 246-252, 1999
- [4] A. Elgammal, D. Harwood, and L.S. Davis, "Non-parametric model for background subtraction," Proc. European Conference Computer Vision, vol. 2, pp. 751-767, 2000
- [5] A. N. Stein and M. Herbert, "Local Detection of Occlusion Boundaries in Video," IVC(27), No. 5, 2 pp. 514-522, 2009
- [6] S. H. Kim, "A Novel Approach to Moving Edge Detection Using Cross Entropy," GVIP05, pp. 21-24, 2005
- [7] W. Chen and N. Nandhakumar, "A Simple scheme for motion boundary detection," Proc. IEEE Inter Conf. on Systems, Man and Cybernetics. 1994.
- [8] S. Suzuki and J. Sklansky, "Extracting nonrigid moving objects by temporal edges," Proc. IEEE Inter Conf. on Pattern Recognition, pp. 69-73, 1992
- [9] J. H. Duncan and T. C. Chou, "Temporal edges: the detection of motion and the computation of optical flow," University of Maryland, Center for Automation Research, CAR-Tech. Rep.-362, 1988
- [10] C. H. Hayden, R. C. Gonzalez, and A. Ploysongsang, "A temporal edge-based image segmentor," PR(20), No. 3, pp. 281-290, 1987
- [11] Canny, J., "A Computational Approach To Edge Detection," Proc. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8, pp. 679-714, 1986

---

저 자 소 개

---



곽재호

- 1999년 2월 : 한양대학교 전자전자통신전파공학군 공학사
- 2001년 2월 : 한양대학교 전자공학과 공학석사
- 2007년 9월 ~ 현재 : 한양대학교 전자컴퓨터통신공학과 박사과정
- 주관심분야 : 영상처리, 패턴인식, 내용기반 멀티미디어 분석



김희을

- 1980년 : 한양대학교 전자공학과 공학사
- 1983년 : Pennsylvania State University 전자공학과 공학석사
- 1989년 : Purdue University 전자공학과 공학박사
- 1989년 ~ 1994년 : University of Texas 조교수
- 1994년 ~ 현재 : 한양대학교 전자컴퓨터통신공학과 정교수
- 주관심분야 : 영상처리, 컴퓨터비전, 패턴인식, 머신비전, MPEG-7