

# 지하 주차장 차량 추적을 위한 객체의 이동 방향 추정

Nguyen Huu Thang<sup>†</sup>, 김재민<sup>†\*</sup>

## Estimation of Moving Direction of Objects for Vehicle Tracking in Underground Parking Lot

Nguyen Huu Thang<sup>†</sup>, Jaemin Kim<sup>†\*</sup>

### ABSTRACT

One of the highly reliable object tracking methods is to trace objects by associating objects detected by deep learning. The detected object is represented by a rectangular box. The box has information such as location and size. Since the tracker has motion information of the object in addition to the location and size, knowing additional information about the motion of the detected box can increase the reliability of object tracking. In this paper, we present a new method of reliably estimating the moving direction of the detected object in underground parking lot. First, the frame difference image is binarized for detecting motion energy, change due to the object motion. Then, a cumulative binary image is generated that shows how the motion energy changes over time. Next, the moving direction of the detected box is estimated from the accumulated image. We use a new cost function to accurately estimate the direction of movement of the detected box. The proposed method proves its performance through comparative experiments of the existing methods.

**Key words:** Moving direction estimation, Detected bounding box, Vehicle tracking, Accumulated binary frame difference image

### 1. 서 론

딥러닝 기반으로 검출된 객체를 연동하여 객체를 추적하는 방법은 고신뢰도 객체 추적 방법 중의 하나이다[1]. 검출된 객체는 Fig. 1에서와 같이 직사각형 상자로 표시된다. 검출된 상자는 객체의 위치, 넓이, 높이 정보를 가지게 되며, 이러한 정보는 객체 추적자(tracker)와 검출된 상자의 연동에 사용된다. 딥러닝 기반 객체 검출 방법의 성능 향상으로 이 방법의 신뢰도가 높아지고 있다[2,3].

하지만 다중 카메라 망(multi-camera network)에

서 다수의 객체를 추적하는 경우, 각 지역 카메라의 감시 영역(a local coordinate)에서 검출된 상자를 전역 감시 영역(the global coordinate)으로 매핑한 후 추적자와 검출 상자를 연동한다. 어안 렌즈 카메라를 사용하여 영역을 감시하는 경우, 영상의 외곽에서 발생하는 작은 매핑 에러는 전역 좌표에서 객체 위치에 큰 이동을 유발한다. Fig. 2(a)에서와 같이 두 카메라가 공동으로 감시하는 영역을 동일 형상을 가진 두 차량이 근접 교차 통과할 때, 매핑 에러로 인하여 Fig. 2(b)에서와 같이 추적자 T1(t-1)에 가장 근접한 검출 상자는 B2(t)가 되며, 추적자 T2(t-1)에 가장

\* Corresponding Author : Jaemin Kim, Address: (05649) 89, Bangi-dong, Songpa-gu, Seoul, Korea, TEL : +82-2-402-1634, +82-10-5683-1634, FAX : +82-2-320-1120, E-mail : jmkim@hongik.ac.kr  
Receipt date : Jan. 19, 2021, Revision date : Feb. 3, 2021  
Approval date : Feb. 16, 2021

<sup>†</sup> School of Electronic and Electrical Engineering, Hongik Univ.

(E-mail : huuthang96@gmail.com)

<sup>\*\*</sup> School of Electronic and Electrical Engineering, Hongik Univ.

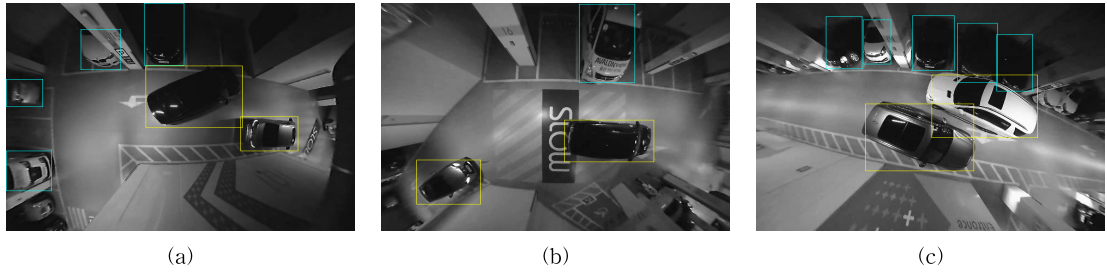


Fig. 1. Three test videos with boxes detected by YOLO v3[3]: (a) the 452th image of test video 1, (b) the 537th image of test video 2, and (c) the 541th image of test video 3. The blue and yellow boxes represent parked and running vehicles, respectively.

근접한 검출 상자는  $B1(t)$ 이 된다. 이는 객체의 아이디 스위치(ID switch)를 유발할 수 있다. 차량의 움직임은 지속성이 있다. 때문에 검출된 상자(차량)의 움직임이 Fig. 2(b)와 같다면, 추적자  $T1(t-1)$ 과 검출 상자  $B2(t)$ , 그리고 추적자  $T2(t-1)$ 과 검출 상자  $B2(t)$ 는 연동이 될 수 없다.

다수의 영상 프레임에 걸친 객체의 이동 궤적을 분석하면 좀 더 안정적인 객체 추적이 가능하지만 [4,5], 수백 대의 카메라를 연동하는 다중 카메라 망에서는 정보의 동기화 등의 문제가 있어 구현이 어렵다. 때문에 안정적인 객체 추적을 위해서는 검출 객체의 움직임 정보가 더욱 필요하다.

움직임 추정 방법은 객체 표면 화소값의 시연속성(temporal continuity)에 기반을 둔 움직임 추정 방법 [6,7]과, 객체의 특징점을 이용하는 움직임 추정 방법 [8,9] 등이 있다. 검출 상자로 나타내는 차량 영역의 모든 화소는 동일 방향으로 움직인다. 때문에 화소의 시연속성 기반 블록매칭 움직임 추정[7]과 특징점 기반 움직임 추정[8]은 검출 상자안의 객체의 움직임

추정에 매우 효과적이다.

하지만 시연속성에 기반을 둔 움직임 추정 방법은 차량의 헤드라이트, 조명의 변화 같은 외부 환경의 변화에 따라 성능이 저하될 수 있다. 특징점을 이용하는 움직임 추정 방법은 객체 표면에서 특징점을 안정적으로 찾기 어려운 경우 성능이 저하되는 문제점이 있다.

최근에는 딥러닝 기반으로 동영상에서 객체의 움직임을 찾는 방법이 있다[10]. 이 방법은 2차원 공간상의 특징을 추출하는 객체 검출에 비하여, 3차원 시공간상의 특징을 추출하기 때문에 많은 연산량을 필요로 한다.

객체의 움직임 검출에 있어 배경(background)과 전경(foreground)의 차분을 이용하는 방법이 널리 사용되고 있으나, 조명의 변화에 민감한 단점이 있다. 대안으로 조명 변화에 강인한 연속된 영상의 차분 영상을 누적한 후 누적 영상을 이진화하여 움직임을 검출하는 방법이 사용되고 있다[11,13,14].

객체의 움직임으로 인한 변화는 움직임 에너지를 나타낸다. 일정 크기 이상의 움직임 에너지는 연속된 영상의 차분 영상을 이진화하여 나타낼 수 있다. 이러한 이진 차분 영상을 누적하여 객체의 동작 분석에 사용하고 있다. 이는 시간의 흐름에 따라 차분의 크기로 표시되는 움직임 에너지가 변화하는 패턴을 분석하는 것이다[12,13].

본 논문에서는 안정적인 차량 추적을 위하여 딥러닝 기반으로 검출된 상자(차량)내의 차량의 움직임은 방향을 정확하게 찾고자 한다. 움직임 방향 추정은 동작 분석에 사용되는 누적 이진 차분 영상을 기반으로 한다. 검출 상자 영역 내의 누적 이진 차분 영상으로 차량의 움직임 방향을 정확하게 찾기 위한

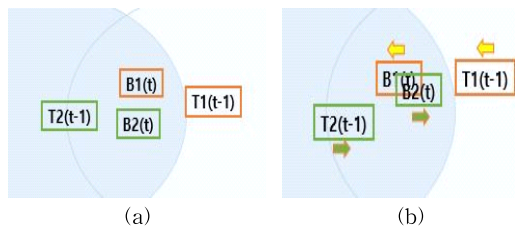


Fig. 2. Incorrect change of object location in the global coordinates due to mapping error: (a) trackers and boxes in the local coordinate and (b) incorrect location of trackers and boxes in the global coordinate. In (b) the arrows indicate the moving directions of the detected boxes and trackers.

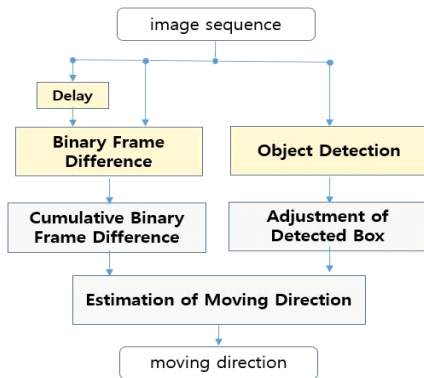


Fig. 3. The overall block diagram.

새로운 비용함수를 제안한다. 제안한 비용함수를 적용한 움직임 방향 추정은 객체 영역 내의 모든 화소들이 동일 방향으로 움직이는 차량의 움직임 방향 추정에 효과적이다. 검출 상자안의 차량 움직임 추정에 효과적인 블록 매칭 움직임 추정 방법과 특징점 기반 움직임 추정 방법으로 찾은 결과를 제안된 방법의 결과와 비교한다.

제안된 방법은 우선 객체의 이동 방향을 효과적으로 나타낼 수 있도록 차분 영상을 누적 저장한다. 이를 위하여 우선 차분 영상에서 변화가 있는 화소의 값은 255로, 변화가 없는 화소는 0으로 이진화한다. 다음으로 이진화된 영상을 다음과 같이 누적한다. 기존 누적된 영상의 모든 화소 값을 일정 값(D)만큼 감소시킨 후, 새로운 영상 프레임의 이진 차분 영상을 누적한다. 이렇게 누적된 이진 차분 영상은 Fig. 4(c),(d)에서와 같이 객체의 움직임 방향을 보여주고 있다. 누적 영상이 형성되면 움직임 방향을 추정한다. 제안된 방법은 전체 흐름도는 Fig. 1과 같다.

본론 2.1 절에서는 움직임이 있는 영역을 나타내는 이진 차분 영상의 생성에 관하여 기술한다. 2.2 절에서는 움직임 방향을 나타낼 수 있는 누적 이진 차분 영상의 생성에 관하여 기술한다. 2.3 절에서 누적 이진 차분 영상을 이용한 움직임 방향 추정에 관하여 기술한다.

## 2. 본 론

### 2.1 이진 차분 영상의 생성

연속된 영상에서 객체가 움직이면, 객체의 움직임이 있는 영역 내의 화소값에 변화가 발생한다. 화소

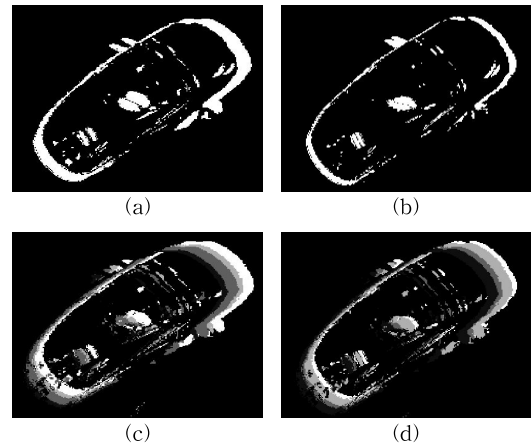


Fig. 4. Binary frame difference image and accumulated binary frame difference image: (a) the 432th binary frame difference, (b) the 433th binary frame difference, (c) accumulated frame difference including the 432th binary frame difference, (d) accumulated frame difference including the 433th binary frame difference. In (d), white, light gray, and dark gray pixels represent the areas of change according to the movement of an object in the 433th, the 432th, and the 431th frames, respectively.

값의 변화는 노이즈로 인해서도 발생한다. 움직임으로 인한 화소값의 변화를 분리 추출하는 임계값은 오츠(Otsu) 방법[15]으로 구한다. 차분영상의 화소값의 변화가 임계값보다 크면 해당 화소 값을 255로, 화소값의 변화가 임계값보다 작으면 0으로 이진화한다. 이진 차분 영상(binary frame difference, BFD)의 생성은 다음과 같다.

$$BFD(x, y, i) = \begin{cases} 255 & \text{if } |I(x, y, i) - I(x, y, i-1)| \geq Th, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

위 수식에서  $BFD(x, y, i)$ 는  $i$  번째 이진 차분 영상의 화소  $(x, y)$ 의 값을 나타낸다. 이 경우 그 값이 255인 화소  $(x, y)$ 는 이전 영상 프레임에서 움직이는 객체의 영역과 현 영상 프레임에서 움직이는 객체의 영역을 동시에 나타내기 때문에, 움직임이 있는 영역은 판단할 수 있지만, 움직임 방향과 크기는 알 수 없다. 따라서 객체의 움직임 방향을 나타낼 수 있는 방법이 필요하다.

### 2.2 누적 이진 차분 영상의 생성

$i-1$  번째 이진 차분 영상의 화소값이 255인 화소

$(x, y)$ 는  $i-2$  번째 영상과  $i-1$  번째 영상에서 움직이는 객체의 영역을 나타내고,  $i$  번째 이진 차분 영상의 화소값이 255인 화소  $(x, y)$ 는  $i-1$  번째 영상과  $i$  번째 영상에서 움직이는 객체의 영역을 나타낸다. 따라서  $i-1$  번째 이진 차분 영상의 화소  $(x, y)$ 의 값이 0 이고,  $i$  번째 이진 차분 영상의 화소  $(x, y)$ 의 값이 255인 화소  $(x, y)$ 는  $i$  번째 영상에서 움직이는 객체의 영역만을 나타낸다. 이와 같이  $i$  번째 이진 차분 영상에서 화소의 값이 255이고  $i-1$  번째 이진 차분 영상에서 화소의 값이 0인 화소들만 선택하여 한 영상에 누적하여 저장할 수 있다면, 시간이 지남에 따라 움직이는 객체의 영역이 어떻게 이동하는 지 알 수 있다. 이를 위하여 누적 이진 차분 영상(accumulated binary frame difference,  $ABFD$ )를 다음과 같이 생성한다.

$$ABFD(x, y, 1) = BFD(x, y, 1). \quad (2)$$

$$\begin{aligned} \text{for } i > 1 \\ ABFD(x, y, i-1) &= \max(ABFD(x, y, i-1) - D, 0). \\ ABFD(x, y, i) &= \begin{cases} 255 & \text{if } ABFD(x, y, i-1) = 0 \text{ and } BFD(x, y, i) = 255, \\ ABFD(x, y, i-1) & \text{otherwise.} \end{cases} \\ \text{end} \end{aligned} \quad (3)$$

위 수식에서  $ABFD(x, y, i)$ 는  $i$  번째 누적 이진 차분 영상의 화소  $(x, y)$ 의 값을 나타낸다. Fig. 4(a), (b)는 이진 차분 영상을, Fig. 4(c), (d)는 누적 이진 차분 영상을 보여주고 있다. Fig. 4(b)에서 화소값이 흰색(밝기=255)인 영역은 432번째 영상과 433번째 영상에서 움직이는 객체 영역을 함께 나타낸다. Fig. 4(d)에서 화소값이 흰색(밝기=255)인 영역은 433번째 영상에서 움직이는 객체 영역만을 나타내고 있다. Fig. 4(d)에서 화소값이 밝은 회색(밝기=185)인 영역은 432번째 영상에서 움직이는 객체 영역, 어두운 회색(밝기=105)인 영역은 431번째 영상에서 움직이는 객체 영역을 나타낸다. Fig. 4(d)에서 밝은 회색 띠의 두꺼운 폭은 432번째 영상에서 차량의 움직임이 클, 흰색 띠의 좁은 폭은 433번째 영상에서 차량의 움직임이 작음을 보여주고 있다.

### 2.3 누적 이진 차분 영상을 이용한 움직임 추정

누적 이진 차분 영상에서 차량의 움직임 추정은 다음과 같이 수행한다.

$$(v_{hor}, v_{ver}) = \underset{(x, y) \in \text{search range}}{\max} \sum_{(x, y) \in \text{Box}} f(x, y, v_{hor}, v_{ver}) \quad (4)$$

where

$$f(x, y, v_{hor}, v_{ver}) = \begin{cases} 2 & \text{if } ABFD(x, y, i) = 255 \text{ and } ABFD(x - v_{hor}, y - v_{ver}, i) = 255 - D \\ 2 - \alpha & \text{if } ABFD(x, y, i) = 255 \text{ and } ABFD(x - v_{hor}, y - v_{ver}, i) = 255 - 2D \\ 2 - \beta & \text{if } ABFD(x, y, i) = 255 - D \text{ and } ABFD(x - v_{hor}, y - v_{ver}, i) = 255 - 2D \\ 2 - \beta - \alpha & \text{if } ABFD(x, y, i) = 255 - D \text{ and } ABFD(x - v_{hor}, y - v_{ver}, i) = 255 - 2D \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

위 식 (4)에서 검출 상자 영역 내에서 모든 화소들의 매칭 값의 합을 최대화하는 움직임 벡터  $(v_{hor}, v_{ver})$ 를 최종 움직임 벡터로 선택한다. 최종적인 움직임 방향은 다음과 같다.

$$\angle = \text{atan}\left(\frac{v_{ver}}{v_{hor}}\right) \quad (6)$$

Fig. 4(c)에서 현 영상(432번째 영상)에서의 객체의 움직임만큼, 흰색 띠(432번째 영상에서 움직이는 객체의 영역( $ABFD(x, y, i) = 255, i = 432$ )을 나타냄)를 좌측 아래로 이동하면,  $ABFD(x - v_{hor}, y - v_{ver}, i-1)$ 는  $255 - D$  혹은  $255 - 2D$ 가 된다. Fig. 4(d)에서 현 영상(433번째 영상)에서 움직임만큼, 흰색 띠(433번째 영상에서 움직이는 객체의 영역( $ABFD(x, y, i) = 255, i = 433$ )을 나타냄)를 좌측 아래로 이동하면,

$ABFD(x - v_{hor}, y - v_{ver}, i-1)$ 는  $255 - D$ 가 된다. 하지만 2배의 움직임만큼 좌측 아래로 이동하면  $ABFD(x - 2v_{hor}, y - 2v_{ver}, i-1)$ 는  $255 - 2D$ 가 된다. 이러한 특성을 고려하여  $\alpha = 1$ 로 한 매칭 값을 식(5)와 같이 정의하였다.

객체 추적을 위한 추적자와 검출 상자를 연동함에 있어, 검출 상자의 움직임 크기보다 움직임 방향을 이용하는 것이 연동 신뢰도 향상에 낫다. 또한 차량의 진행 방향은 관성으로 인하여 유지되는 특성이 있다. 때문에 최근 프레임에서 움직임과 이전 프레임에서 움직임에 가중치를 곱하여 더한 결과를 움직임 방향 추정에 사용한다. 가중치를 내재적으로 포함하고 있는  $\beta > 0$ 을 사용하였다.

### 3. 실험 및 결과

실험에 사용한 테스트 비디오는 지하 주차장 천정에 설치된 어안렌즈 카메라로 주행하는 차량을 녹화한 영상이다. 영상은  $640 \times 480$  해상도를 가지는 그레이 영상이다. 카메라에서 한 프레임 단위로 압축하여 패킷으로 전송한 것을 서버에서 수신한 영상이 테스트 비디오다. 수백 대의 카메라가 연동된 관계로 수

신 과정에서 프레임이 누락되는 경우가 있어, 영상 프레임의 간격에 다소 변동이 있다. 때문에 연속된 두 영상 프레임 간의 차량 움직임 방향은 유지되지만, 움직임 크기에는 변동이 있다.

테스트 비디오 1은 어두운 주차장 주행로를 단독으로 주행하는 검정색 차량을 녹화한 76개 영상 프레임으로 구성되어 있다. Fig. 2(a)는 테스트 비디오의 452 번째 영상을 보여주고 있다. 차량 표면 화소들의 밝기 분포 변화가 작고 차량 화소와 배경 화소의 밝기 차이가 작아, 차량의 주요 특징 포인트를 찾기가 어렵다.

테스트 비디오 2는 주행 안내 사인이 표시된 노면에서 주행하는 검정색 차량을 녹화한 38개 프레임으로 구성되어 있다. Fig. 2(b)는 테스트 비디오 3에서 차량이 주행 안내 신호 위를 통과하는 순간을 보여주고 있다.

테스트 비디오 3은 상대적으로 밝은 주차장 주행로를 교차 주행하는 회색 차량과 흰색 차량을 녹화한 65개 영상 프레임으로 구성되어 있다. Fig. 2(c)는 테스트 비디오 2에서 두 차량이 교차 주행하는 순간을 보여주고 있다.

실측 차량의 움직임은 YOLO v3 방법으로 차량을 검출한 후, 검출 상자의 위치 및 크기를 포토샵에서 확인하면서 수정한 후, 수정된 검출 상자의 값을 이용하여 차량의 움직임을 실측하였다.

본 논문에서 제안한 방법과 절대차의 합(sum of absolute difference, SAD)을 비용 함수로 사용하는 블록매칭[7] 방법 그리고 주요 특징점 기반 움직임 추정 방법[8]으로 구한 움직임 방향 값을 비교하였다. Fig. 5는 3가지 방법으로 추정한 움직임 방향 오차 값을 보여주고 있다. 단위는 각도이다.

검출된 상자 영역안의 객체의 움직임을 추정함에 있어, 움직임 추정 방법에 따라 상자의 영역을 다소 조절하였다. 검출된 상자의 크기와 위치에 다소 오류가 있어, 상자의 크기를 확장하여 차량이 검출 상자 안에 포함하도록 하기 위한 것이다. 제안한 방법의 경우 검출 상자의 영역을 다소 증가(20 화소만큼 증가) 시킨 후 영역 내의 차량의 움직임을 추정할 때 안정적인 결과를 가져다주었다. 하지만 블록 매칭 움직임 추정 방법과 특징점 기반 움직임 추정 방법의 경우 검출 상자 영역을 그대로 적용하는 것이 좋은 결과를 가져다주었다. 제안한 방법은 배경에 영

향을 받지 않지만, 다른 두 방법은 배경에 영향을 받기 때문이다.

특징점 기반 움직임 추정 방법은 차량의 움직임이 작은 경우 상대적으로 오차가 크다. 그리고 차량의 특징점을 안정적으로 찾지 못하면 움직임 방향 오차가 매우 크게 증가하는 문제점이 있다. 직사각형으로 표현되는 검출 상자가 차량 주변 노면의 교통 안내 신호를 포함하는 경우, 노면의 신호가 특징점에 잘못 포함되면 추정한 움직임 방향 각도의 오차가 급격히 증가한다. Fig. 5(d)의 559 번째 붉은 그래프가 이를 보여주고 있다. 따라서 테스트 영상에서 특징점 기반 움직임 추정 방법은 검출 상자 내 객체의 움직임 방향 추정에 부적합하다. 때문에 실험에서는 특징점 기반 움직임 추정을 하는 경우 검출된 상자의 영역을 증가시키지 않았다.

블록 매칭 방법과 본 논문에서 제안한 방법은 성능이 비슷하나, 차량 표면 화소들의 밝기 분포의 변화가 작고 차량 화소와 배경 화소의 밝기 차이가 작은 경우 제안한 방법이 다소 우수한 성능을 보여주고 있다. 이는 Fig. 5(c)의 460번째 프레임 부근에서 발생한다.

Fig. 5의 그래프는 제안한 방법으로 구한 움직임 방향 오차가 최대 45도 이하이고, 대부분 프레임에서 20도 이내이다. 마주 보고 교차 주행하는 두 차량은 진행 방향의 각도 차이가 180도 내외이고, 교차로에서 교차 주행하는 차량의 경우도 그 각도의 차이가 90도 내외이다. 때문에 제안한 방법으로 추정한 검출 상자의 움직임 방향을 적용한다면, 추적자와 검출 상자의 잘못된 연동으로 인한 아이디 스위치를 줄일 수 있다.

Table 1은 움직임 방향 오차의 절대값의 평균과 표준편차를 보여주고 있다. 통계적으로 볼 때 모든 테스트 영상에서 95% 신뢰도로 제안한 방법으로 구한 움직임 방향의 오차는 20도 안으로 들어온다.

자동차의 경우에는 주행 중에도 그 형상의 변화가 적다. 때문에 누적 이진 차분영상으로 차량의 움직임은 방향과 크기를 매우 효과적으로 나타낼 수 있다. 반면 걷는 사람의 경우에는 그 형상의 변화가 상대적으로 크다. 이러한 경우에도 안정적으로 움직이는 방향과 크기를 나타내는 방법에 대한 추가 연구가 필요하다.

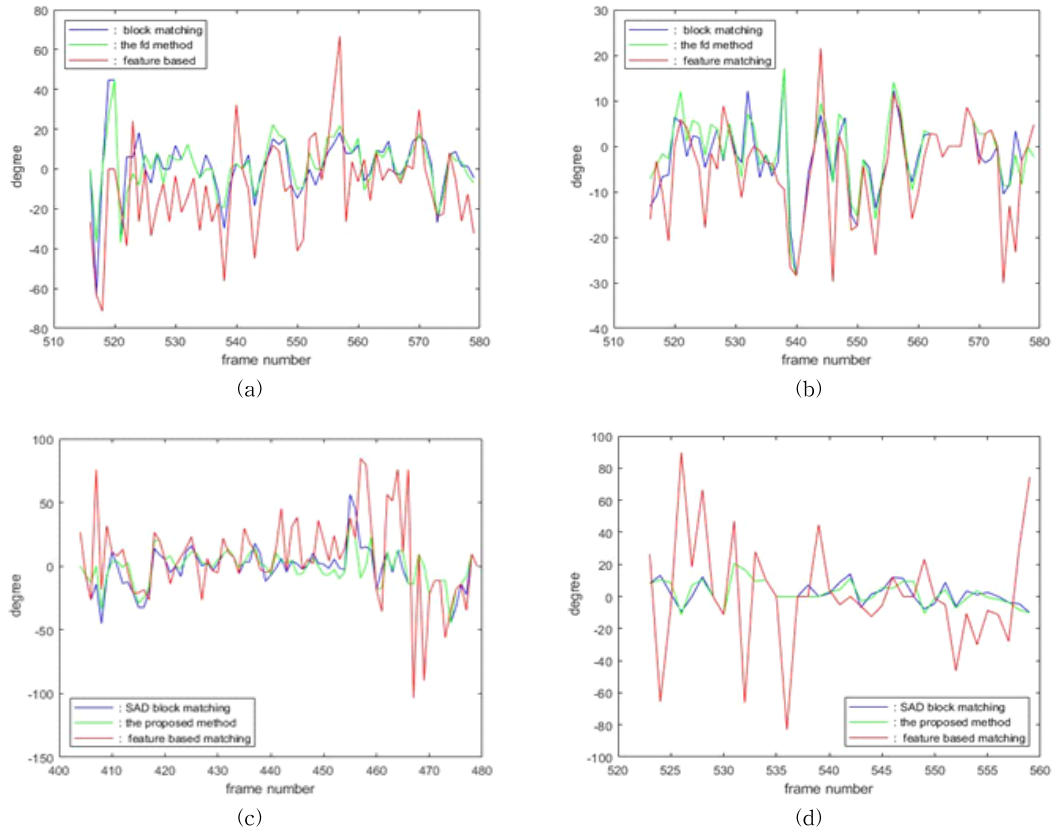


Fig. 5. The errors of the moving direction angle estimated by 3 difference methods: (a) the gray vehicle in the test video 3, (b) the white vehicle in the test video 3, (c) the black vehicle in test video 1, (d) the black vehicle in test video 2. The blue, green, and red graphs indicate the results of a SAD based block matching method[7], the proposed method, and a key point based motion estimation method[8], respectively.

Table 1. The means and standard deviations of the absolute values of the errors of the moving direction angle estimated by 3 difference methods: (a) the white vehicle in the test video 3, (b) the gray vehicle in the test video 3, (c) the black vehicle in the test video 2, (d) the black vehicle in the test video 3.

	(a)		(b)		(c)		(d)	
	mean	std	mean	std	mean	std	mean	std
the proposed method	6.1	5.6	9.7	9.4	9.7	9.0	6.2	5.1
block matching[7]	6.0	5.6	10.4	12	12	12	6.3	5.4
key point based method[8]	8.3	8.4	19	17	24	23	24	26

#### 4. 결 론

딥러닝 기반으로 검출된 객체를 연동하여 객체를 추적하는 방법은 고 신뢰도의 객체 추적 방법이다. 이러한 방법의 객체 추적에서 추적자와 검출된 객체의 연동 신뢰도를 향상시키기 위하여, 검출된 객체의 움직임 방향을 추정할 수 있는 방법을 제안하였다.

객체의 움직이는 방향을 효과적으로 보여주는 누적 이진 차분영상을 생성하고, 이를 기반으로 검출된 상자(객체)의 움직임 방향을 추정하였다. 동영상에서 기존 방법들과의 비교 실험을 통하여 제안한 방법으로 구한 움직임 방향은 그 오차가 작고 신뢰도가 높음을 입증하였다.



# REFERENCE

- [1] D. Held, S. Thrun, and S. Savarese, "Learning to Track at 100 FPS with Deep Regression Networks," *ECCV2016*, pp. 749-765, 2016.
- [2] J. Redmon and A. Farhadi, "Yolo9000: Better, Faster, Stronger," *CVPR2017*, pp. 6517-6525, 2017.
- [3] J. Redmon and A. Farhadi, *YOLOv3: An Incremental Improvement*, Technical Report, <https://arxiv.org/abs/1804.02767>, (accessed Dec., 30, 2020).
- [4] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple Hypothesis Tracking Revisited," *ICCV2015*, pp. 4696-4704, 2015.
- [5] J. BerclazFrancois, F. Fleuret, E. Turetken, and F. Fua, "Multiple Object Tracking Using K-Shortest Paths Optimization," *PAMI*, Vol. 33, No. 9, pp. 1806-1819, 2011.
- [6] S. Zhu and K. Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," *IEEE Trans. on Image Processing*, Vol. 9, No. 2, pp. 287-290, 2000.
- [7] J. Lu and M. Liou, "A Simple and Efficient Search Algorithm for Block-Matching Motion Estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol 7, No. 2, pp. 429-433, 1997.
- [8] J. Huang and W. Hsieh, "Automatic Feature-Based Global Motion Estimation in Video Sequences," *IEEE Trans. on Consumer Electronics*, Vol. 50, Issue 3, pp. 911-915, 2004.
- [9] Z. Meng, L. Meng, and H. Tomiyama, "Lucas-Kanade Optical Flow Based Camera Motion Estimation Approach," *2019 International Conference on Identification, Information and Knowledge in the Internet of Things*, pp. 20-26, 2020.
- [10] S. Zhang, M. Balamurali, and A. Kyme, "Deep Learning-Based Motion Estimation for Uninterrupted Tracking of Awake Rodents in PET," *2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings*, 2018.
- [11] M. Al-Smadia, K. Abdulrahimb, and R. A. Salama, "Cumulative frame differencing for urban vehicle detection," *Proc. of SPIE*, Vol. 10011, 100110G, pp. 1-7, 2016.
- [12] D. I. Ramddhan, I. P. Sari, and L. O. Sari, "Comparison of Background Subtraction, Sobel, Adaptive Motion Detection, Frame Differences, and Accumulative Difference Images on Motion Detection," *Singeri*, Vol. 22, No. 1, pp. 51-62, 2018.
- [13] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 3, pp. 257-267, 2001.
- [14] S. Yeon and J. Kim, "Realtime Vehicle Tracking and Region Detection in Indoor Parking Lot for Intelligent Parking Control," *Journal of Korea Multimedia Society*, Vol. 19, No. 2, pp. 418-427, 2016.
- [15] N. Otsu, "A Threshold Selection Method from Gray-level Histograms," *IEEE Trans. Systems, Man, Cybernet.* Vol. 9, Issue 1, pp. 62-66, 1979.



Nguyen Huu Thang

2014년~2019년 Danang

University of  
Technology Electrical  
and Communication

Engineering 학사

2020년~현재 홍익대학교 전자전

기공학부 석사

관심분야: 객체 검출 및 인식, 컴퓨터비전



김재민

1980년~1984년 서울대학교 전기  
공학과 학사

1984년~1986년 서울대학교 전기  
공학과 석사

1989년~1994년 Rensselaer  
Polytechnic Institute 전기  
공학과 박사

2001년~현재 홍익대학교 전자전기공학부 교수

관심분야: 영상처리, 패턴인식, 컴퓨터비전, 딥러닝