

차량 그림자 누적을 통한 검지 영역 자동 설정 및 교통량 측정 방법

김지완 · 이재성*

Automatic Detection of Vehicle Area Rectangle and Traffic Volume Measurement through Vehicle Sub-Shadow Accumulation

Jee-Wan Kim · Jaesung Lee*

Department of Electronic Engineering, Korea National University of Transportation, Chungbuk 380-702, Korea

요 약

기존 영상 검지기 분야에는 다양한 고성능 알고리즘들이 존재하지만 실시간 연산 요구량이 너무 많아 시스템 장비가 고가, 고전력을 소모하는 단점이 있었다. 이에 본 논문에서는 저가, 저전력 영상 검지 시스템 구현을 위해 안드로이드 플랫폼의 성능 사양에 적합한 저연산량의 영상 검지 알고리즘을 제안한다. 본 방법은 차량 하부에만 생성되는 sub-shadow 를 분리하여 이를 누적함으로써 차선 및 검지 영역을 정밀하게 설정하고 이 검지 영역을 통과하는 차량 자체와 차량 sub-shadow 의 통과패턴을 판단하여 차선별 교통량 뿐만 아니라 상행 및 하행 교통량까지 자동으로 분류할 수 있다. 실험 결과 제안하는 알고리즘은 하행 차량의 경우 평균 97.1%, 상행 차량의 경우 평균 94.1%의 검지율을 보였다. 이 결과는 상용 루프검지기의 성능 95% 에 버금가는 수준으로 만족스러운 성능을 보였다.

ABSTRACT

There are various high-performance algorithms in the area of the existing VDSs (vehicle detection systems). However, they require a large amount of computational time-complexity and their systems generally are very expensive and consume high-power. This paper proposes a real-time traffic information detection algorithm that can be applied to a low-cost, low-power, and open development platform such as Android. This algorithm uses a vehicle's sub-shadow to set ROI(region of interest) and to count vehicles using a location of the sub-shadow and the vehicle. The proposed algorithm is able to count the vehicles per each road and each direction separately. The experiment results show that the detection rate for going-up vehicles is 94.1% and that for going-down vehicles is 97.1%. These results are close to or surpasses 95%, the detection rate of commercial loop detectors.

키워드 : 영상 검지기, sub-shadow, 연산복잡도, 안드로이드, 교통량

Key word : VDS, sub-shadow, time complexity, Android, traffic volume

접수일자 : 2014. 04. 22 심사완료일자 : 2014. 05. 13 게재확정일자 : 2014. 05. 26

* **Corresponding Author** Jaesung Lee(E-mail:jaesung.lee@ut.ac.kr, Tel:+82-43-841-5163)

Department of Electronic Engineering, Korea National University of Transportation, Chungbuk 380-702, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2014.18.8.1885>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

국내 교통 정보 검지기 설치 현황은 루프검지기가 70%, 영상검지기가 20%, 기타 검지기가 10%를 차지하고 있다. 그러나 루프검지기는 처음 도로를 타설할 때 설치하거나 기존 도로를 컷팅하여 설치해야 한다. 길게는 몇 백 미터의 전선을 끌어와 연결해야 하기 때문에 시공이 어렵고 전류로 차량 흐름을 체크하기 때문에 이 긴 전선에 항상 전류가 흘러야 해서 전력소모가 크다. 또한 과적 차량에 의해서 또는 도로가 여름철에 뜨겁게 달궈졌다가 겨울에 얼어붙는 온도의 변화에 의해서 루프선이 뒤틀리곤 하는데, 이 과정에서 끊어지거나 오작동이 발생할 가능성이 매우 높다. 이렇게 오작동하는 루프는 재시공해야하기 때문에 멀쩡한 도로를 파손하게 되어 도로의 수명을 줄이게 되고 공사 기간 중 도로 혼잡비용을 높이는 요인이 된다. 이렇듯 설치 및 유지보수의 어려움, 도로 미관 저해 등 기존 루프 검지기가 갖는 많은 단점들을 극복하기 위해 최근에는 영상 검지기가 많이 설치되고 있다.

적외선, 레이저, 레이더, 지자기 센서 등 다른 대안들도 있지만 영상 검지 시스템은 기본 교통 정보 이외에도 기후 측정, 유고 검지, 도로 상황 등 다양한 추가 정보의 수집이 가능하고 보안 목적을 위한 영상 기록 및 모니터링이 가능하다는 장점 때문에 선호되고 있는 실정이다.

하지만 국내에 설치되고 있는 영상 검지기들을 면밀히 살펴보면 많은 문제점을 내재하고 있다. 관할 구역 및 관할 기관이 다름에 따라 지역마다 또는 도로 구간마다 천차만별의 다양한 영상검지기가 설치되어 있다. 게다가 다양한 제조업체들 간 기술적 표준이 없어 펌웨어 등 시스템 유지보수 관리가 폐쇄적(in-house platform)이다.

이는 새로운 또는 개정된 도로교통법 시행령이 발효될 때마다 새로운 장치를 개발해 교체 설치하는 형식으로 이어져 하드웨어 자원의 낭비를 유발하게 된다. 고가의 납품 단가(대당 1~3천만원) 및 유지보수비(대당 연간 5~7백만원)는 어찌보면 당연한 결과이다. 뿐만 아니라 최적화 설계에 대한 노력이 부진함에 따라 전력소모가 크다는 문제도 많이 제기되고 있으며 이 문제 때문에 신규 설치지역에 추가 전기 공사를 수반하기도 한다.

영상검지기 관련 국내 연구개발 현황을 살펴보면 SMART-I [1] 와 같이 영상검지기에서 교통 정보의 검지 효율을 높이기 위한 연구는 몇몇 수행되고 있으나 본 논문에서 언급한 문제점들을 해결하기 위한 연구는 요원한 실정이다.

따라서 본 저자는 스마트폰과 같은 안드로이드 기반 저전력 플랫폼에서 앱 형식으로 영상 검지기를 구현하는 개발 방식을 제안하였다[2]. 그렇게 함으로써 small feature size 에 따른 저전력, 저비용(설치 공사비) 실현이 가능하게 되고 온라인 업데이트를 통해 유지보수 비용도 대폭 절감할 수 있게 된다. 무엇보다도 오픈된 개발 플랫폼을 제공함으로써 여러 능력 있는 개발자들이 선의의 경쟁을 통해 만든 우수한 앱들만 설치되기 때문에 가장 좋은 기술들만 현장에 적용될 수 있게 되고 그에 따라 일관성 있는 검지율을 확보할 수 있을 것으로 판단한다.

최근 컴퓨터 시스템의 고성능화에 따라 Optical Flow Vector [3] 를 이용하거나 SIFT (Scale Invariant Feature Transform)[4]와 같이 차량의 특징점 추출 알고리즘으로부터 차량 특징 포인트의 descriptor를 이용하여 가상의 입체 터널을 통과하는 차량을 실시간으로 인지해 그로부터 교통 정보를 추출하는 등 좀 더 정교하고 고급스러운 방법이 선보이고 있다. 하지만 그러한 알고리즘들을 실시간으로 구현하는 데에는 단위 시간당 요구 연산량이 매우 방대하여 안드로이드 시스템과 같은 소규모 시스템이 실시간으로 감당하기에는 역부족이다.

일반적으로 고성능 데스크톱 PC 대비 안드로이드 플랫폼(갤럭시S4기준)의 연산 성능(computing power)의 차이는 약 4~5배이다[5]. 따라서 최근 고가의 시스템에 적용되고 있는 optical flow 나 SIFT 와 같이 연산량이 매우 많은 알고리즘을 안드로이드 플랫폼에 적용하여 실시간으로 동작하도록 하는 것은 현실적으로 불가능하다.

이러한 문제를 해결하기 위해 본 논문에서는 안드로이드 기반 실시간 영상검지 시스템에 적용이 가능한 저연산량의 검지 알고리즘을 제안한다. 연산량이 줄면 검지 성능이 떨어지기 마련이나 본 논문에서 제안하는 알고리즘은 저연산량에도 불구하고 검지율은 기존 최고급 알고리즘들을 이용한 검지율 수준을 유지한다.

II. 관련 연구

기존의 저연산량 영상검지 알고리즘은 평균배경영상과 입력영상 간 차영상으로부터 thresholding 과정을 통하여 이진화(binazation)한 후 수동으로 설정한 고정된 ROI(region of interest) 내의 VAR(Vehicle Area Rectangle) 을 결정하고 이를 시간대 별로 카운트하여 기본 교통 정보(교통량, 속도, 점유율 등)을 측정하였다 [6, 7]. 하지만 이 방법은 차량 간 간격이 좁은 경우 또는 카메라의 흔들림 등에 의해 2대 이상의 차량이 큰 차한 대로 인식되는 문제가 자주 발생하였다.

본 논문은 그러한 기존의 영상검지기 알고리즘의 단점을 보완하고 검지율을 높이기 위해 그림자 누적 모델을 이용한 자동 ROI설정 및 차량 카운트 방법을 제안한다. 차량 하부 그림자(이하 sub-shadow)는 같은 시간대의 모든 차량에 대해 거의 유사한 픽셀값을 가지며 차량이 만들어내는 그림자 중 가장 어두운 특징이 있다. 차량의 전면 또는 후면 바로 아래에는 항상 sub-shadow가 관찰되므로 차량 카운트 및 분리의 기준으로 활용하기에 매우 적합하다. 기존에도 sub-shadow를 이용한 차량검지의 시도가 있었다[8, 9]. 하지만 기존의 연구들은 주로 주행 중 전방에 차량이 있음을 감지하기 위한 용도, 주차장의 차량 대수 카운트하기 위한 용도, VAR에 상관없이 단방향(하행)의 차량 수를 카운트하는 용도로만 sub-shadow를 이용하였다. 본 논문이 제안하는 방법은 경제성을 고려하여 한 대의 카메라로 설치 방향에 상관없이 상행, 하행 이동 방향을 인지하여 각 차선별 그리고 방향별 동시 카운트가 가능한 방법이다.

Sub-shadow는 도로 지면에 가장 가까이 생성되기 때문에 도로 위 차량위치를 비교적 정확하게 유추할 수 있으며, 도로구간을 알아낼 수 있다. 이를 착안하여 능동적인 ROI 설정을 적용하였고, 기존의 차영상을 이용한 VAR 결정 방법에 sub-shadow를 이용한 판단 과정을 추가하여 검출률을 높인다.

III. 그림자 누적 모델을 이용한 영상 검지 알고리즘

제 3절에서는 제안하는 차량검출 시스템의 구현을 위해 필요한 영상 처리 알고리즘 파이프라인을 구성하

고 각 블록에 대해 자세히 설명한다. 다음 그림 1은 제안하는 차량 카운트 방법의 파이프라인 블록도이다.

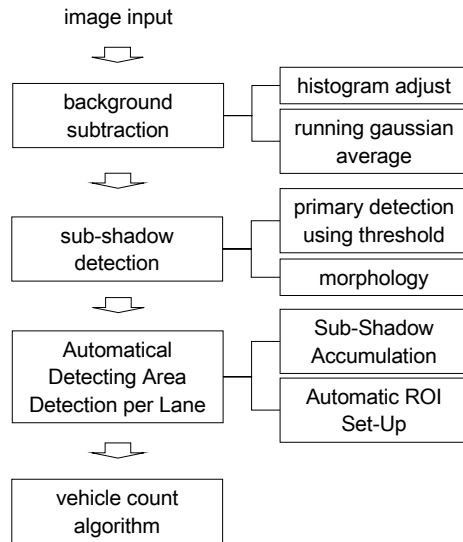


그림 1. 제안하는 차량 카운트 방법에 대한 블록도
Fig. 1 Block diagram of the proposed vehicle counting method

3.1. 도로 배경 제거 (Road Background Subtraction)

본 논문이 제안하는 차량검출 알고리즘의 첫 단계는 카메라의 입력 프레임으로부터 차량 이외의 배경을 제거하는 것이다. 검지용 카메라는 항상 같은 도로 배경을 촬영하고 있기 때문에 연속한 두 프레임에서 변화가 있는 픽셀구간(움직이는 사물 구간) 외에는 비슷한 값을 가지게 된다. 이점을 이용하여 지속적으로 모니터링을 하면 움직이는 사물을 배경으로부터 분리할 수 있다. 우선 sub-shadow 를 더욱 잘 검출하고, 오검출을 줄이기 위해 입력영상 중 어두운 영역의 스펙트럼(히스토그램)을 늘려준다. 그런 다음 연산속도가 가장 빠르고 시스템 메모리 요구사항이 가장 적은 것으로 알려진 running gaussian average [10] 알고리즘을 적용해 이동 평균배경을 만들어 현재의 입력영상과 이동평균배경 간 차영상을 구함으로써 배경을 제거하게 된다.

3.1.1. 히스토그램 조정(Histogram Adjustment)

카메라로 촬영되어 시스템으로 들어오는 입력 영상은 시간과 날씨에 따라 다양한 밝기를 가지게 된

다. 상황에 따라 sub-shadow 픽셀과 sub-shadow가 아닌 픽셀간의 값 차이가 크지 않을 수 있다. 그렇게 되면 sub-shadow 영역 검출을 위해 설정하는 임계값으로부터 픽셀값 사이의 격차가 작아 검출 정확도가 떨어질 수 있기 때문에 전체적인 히스토그램을 조정 (adjustment)할 필요가 있다. sub-shadow 검출은 가장 어두운 값에 관심을 가지기 때문에 영역의 스펙트럼을 늘려주었다. 다음 식(1)은 히스토그램 조정을 수식으로 나타낸 것이다.

$$y = \left(\frac{x-a}{b-a}\right)^\gamma \times (d-c) + c \quad (1)$$

x 는 입력 영상의 픽셀값이고 y 는 출력 영상의 픽셀값이다. 본 식은 x 가 a 이상 b 이하의 값들에 대해서만 적용하는 수식이며 a 이하의 입력 픽셀값들은 모두 $y = c$ 로 단일화되고, b 이상의 값은 모두 $y = d$ 로 단일화된다. a 이상 b 이하의 입력 픽셀값들은 γ 값에 따라 비선형으로 변환할 수 있다. 본 논문에서는 어두운 픽셀 부분의 스펙트럼을 넓히기 위해 γ 값을 0.5로 지정하였다. 상수 a, b, c, d 는 시간대별, 날씨별로 설정되는 값이며 일상적인 낮 시간대에는 a, c 는 0, b, d 는 1 (즉, max. 픽셀값인 255)로 하여도 전체 시스템 성능에 문제가 없음을 확인하였다. 그림 2는 그렇게 히스토그램을 조정한 결과이다.



그림 2. 원본영상과 히스토그램 조정 결과 예제
Fig. 2 Original image and histogram adjustment

그림 2에서 알 수 있듯이 조정된 영상에서는 어두운 픽셀 영역의 스펙트럼이 늘어나 일반 그림자 영역과 sub-shadow 영역이 구별이 가능해지며 이를 통해 적절한 임계값을 설정할 수 있고, sub-shadow 검출에 있어 잡음을 최소화 할 수 있게 된다.

3.1.2. 가우시이동평균(Running Gaussian Average)

배경을 제거를 위해서는 먼저 배경영상을 확보할 필요가 있는데, 연속적으로 입력되는 이미지 간 각 픽셀에 대해 running average 를 계산하면 배경을 얻을 수 있다. 아래의 식 (2)는 running average 계산 수식을 나타내고 있다.

$$\mu_t = \alpha I_t + (1 - \alpha)\mu_{t-1} \quad (2)$$

위 식에서 I_t 는 현재 프레임 t 에서 입력된 영상의 픽셀 값, μ_t 는 프레임 t 까지의 평균, 그리고 α 는 가중치를 나타낸다. 여기서 가중치 α 는 새로 입력된 자료 I_t 가 이전까지의 평균값 μ_{t-1} 에 직접적으로 얼마만큼의 영향을 주는지에 대한 값으로, 평균을 위한 자료량의 역수가 된다. 본 연구에서는 240 x 320, 30fps 영상을 이용하였으며, 적절한 이동평균배경을 얻기 위해 자료량을 200프레임으로 설정하였다. 이를 위 식에 적용하면 α 는 1/200이 된다. 본 가중치 설정은 카메라의 성능과 설치환경에 따라 달라져야 한다.



그림 3. 평균배경 예제
Fig. 3 Example of average background

배경영상은 정규분포를 따른다고 가정하고, 배경영상과 현재의 입력 영상의 픽셀 값을 비교하여 표준편차 σ_t 를 벗어나는 픽셀은 차량과 그 그림자 영역으로 판단하고 식 (3)을 이용하여 그 영역을 구별한다.

$$|I_t - \mu_t| > k\sigma_t \quad (3)$$

아래 그림 4는 식 3을 판단 기준으로 하여 이진화 (threshold)한 영상이다.

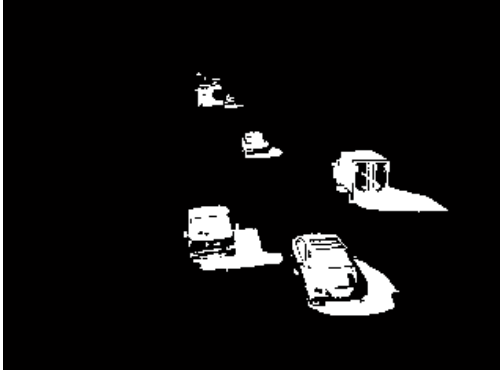


그림 4. Thresholding 으로 배경을 제거한 결과
Fig. 4 Removing the background by thresholding

이제 sub-shadow 검출을 위해서 그림 2의 히스토그램이 조정된 영상을 그림 4의 이진화 이미지(즉, 픽셀값이 1 또는 0)와 픽셀간 상호 곱하여 투영하고 결과가 0인 배경 픽셀들을 max 픽셀값으로 변환하면 다음 그림 5를 구할 수 있다.



그림 5. 차량영상 투영 및 배경 변환
Fig. 5 Reflection of adjusted vehicle area and its shadow area and background transformation

3.2. 차량 밑 그림자 검출(Sub-Shadow Detection)

Sub-shadow는 도로영상에서 가장 낮은 픽셀값을 가지기 때문에 적절한 임계값으로 어두운 영역을 검출해야 한다. 그런 다음 검출된 영역의 잡음을 제거하는 과정이 필요하다.

3.2.1. 임계값을 통한 1차 검출(Primary Detection using Threshold)

우선 어두운 영역 검출을 위해 그림 5의 결과에서 가장 낮은 픽셀값을 얻는다. 이 값은 시간과 날씨에 따라 항상 다른 값을 가지게 되므로, 유동적으로 적용하기 위해 가장 낮은 값에 대한 running average를 적용한다. 이는 다음 식 (4)와 같이 표현할 수 있다.

$$D_t = D_{t-1} + \frac{(\min(I_t) + \beta) - D_{t-1}}{\alpha} \quad (4)$$

D_t 는 가장 낮은 픽셀값이고, I_t 는 입력 이미지의 픽셀값, β 는 I_t 의 최솟값에 대한 표준편차이며 α 는 가중치이다. 그림 5의 결과에서 D_t 를 임계값으로 설정하여 식 (4)에서 D_t 값 이하의 픽셀을 추출하면 그림 6과 같이 sub-shadow 영역을 1차적으로 추출할 수 있다.

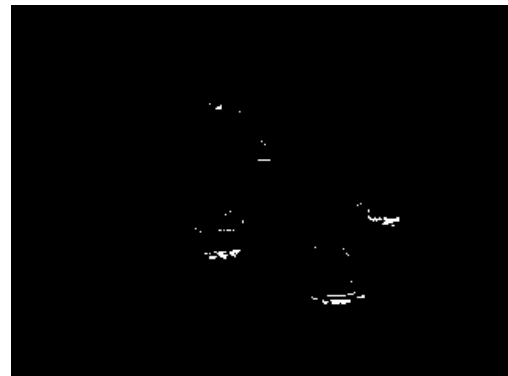


그림 6. Sub-shadow 1차 추출결과
Fig. 6 The first result of sub-shadow extraction

3.2.2. 모폴로지(morphology)

sub-shadow 분리는 밝기 값을 기준으로 구분하기 때문에 잡음이 발생할 수 있다. 차량 검지를 위해 필요한 sub-shadow 영역은 차량의 바로 밑 수평 방향의 그림자인데 차량 측면의 하단 등에 생기는 어두운 그림자는 잡음으로 작용하여 차량 카운트 시 오검출을 야기할 수도 있다. 한편 차량의 상태에 따라 sub-shadow가 얇게 생성되거나 끊어진 형태를 가지기도 한다. 이러한 점들을 보완하기 위해 두 번의 morphology 연산을 실시하였다. 우선 얇거나 끊어진 sub-shadow 픽셀을 보충하기 위해 이진영상에서 픽셀을 팽창하는 dilation 연산을 적용한다. 다음 식 (5)는 b연산의 정의이다.

$$A \oplus B = \bigcup_{(x,y) \in B} A_{(x,y)} \quad (5)$$

여기서 A 는 이미지, B 는 0으로 이루어진 3x3 정방형마스크를 나타낸다. 위 연산에서 마스크 B 는 이진 이미지 A 의 값과 일치할 경우 0을 반환하고, 하나라도 일치하지 않는 경우는 1을 반환한다. 이를 통해 sub-shadow는 상, 하, 좌, 우 방향으로 팽창된 결과를 얻을 수 있다. 이후 차량 측면에 세로로 발생하는 잡음을 제거하기 위해 모든 픽셀의 수평길이를 수축하는 erosion 연산을 수행한다. 다음 식 (6)은 erosion 연산의 정의이다.

$$A \ominus B = (x,y) : B_{(x,y)} \subseteq A \quad (6)$$

위 연산에서 B 는 1로 이루어진 1x5 수평마스크를 나타낸다. 위 연산에서 마스크 B 는 이진 이미지 A 의 값과 마스크의 값이 일치할 경우 1을 반환하고, 일치하지 않을 경우는 0을 반환한다. 이를 통해 sub-shadow는 좌, 우 방향으로 침식된 결과를 얻을 수 있다.

아래 그림 7은 그림 6의 결과로부터 모폴러지를 적용한 결과이다. 그림에서 알 수 있듯이 차량 및 sub-shadow 영역만 뚜렷하게 남게 된다.



그림 7. 모폴러지가 적용된 sub-shadow 이미지
Fig. 7 Sub-shadow image after morphological operations

3.3. 차선별 자동 검지 영역 검출(Automatical Detecting Area Detection per Lane)

Sub-shadow는 차량의 전면부 바로 아래 생성되며 그 폭은 차량의 폭과 비슷하기 때문에, sub-shadow를 통해

차량의 위치를 비교적 정확하게 알아낼 수 있다. 이를 응용하여 sub-shadow를 누적하면 쉽게 자동으로 차선별 검지 영역을 검출할 수 있다.

3.3.1. 차량 밑 그림자 누적(Sub-Shadow Accumulation)

모폴러지 연산이 적용된 sub-shadow는 다른 차선의 sub-shadow와 확연히 분리된다. 각 프레임의 sub-shadow 영역의 픽셀값('1')들을 계속 누적해 가면 차량이 지나간 길을 기록하게 되고, 이로부터 각 차선 영역을 파악할 수 있게 된다. 기존에도 이와 비슷한 방식으로 중앙선을 검출을 응용한 연구[11]가 있었으나 개별 차선이 아닌 도로 전체영역을 검출하여 ROI(Region of Interest)를 설정하는데 이용하였다. 본 논문의 방법은 도로 전체영역이 아니라 개별 차선을 검출할 수 있는 진보된 방법이다. 중간에 차선을 바꾸는 차량이 있지만 누적량이 1 이상인 픽셀들에 대해 평균 값을 구하고 그 값을 임계값으로 하여 thresholding 처리하기 때문에 간헐적으로 값이 누적된 픽셀들은 임계값에 미달하여 결국 도로 검출 결과에는 영향을 주지 않는다. 결과적으로 얻은 이미지는 다음의 그림 8과 같다.

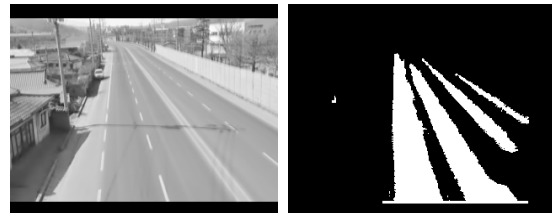


그림 8. 평균배경과 sub-shadow 누적을 이용해 얻은 도로영역
Fig. 8 Running average background and accumulated sub-shadow model

3.3.2. 자동 ROI 설정(Automatic ROI Set-Up)

기존 검지기들은 ROI를 수동으로 설정하기 때문에 오랜 시간동안 카메라가 흔들림에 따라 발생하는 카메라 앵글 변화에 따라 주기적으로 ROI를 수정해줘야 하는 불편함이 있었다. 물론 카메라 앵글 변화에 대한 자동 보상을 위한 image calibration 알고리즘도 연구된 바 있다. 하지만 기본적으로 카메라 앵글이 어떻게 변했는지 에 대한 정보를 알아내기 힘들다는데 그 한계가 있다. 본 논문에서는 카메라 앵글 변화 정보 없이도 자동으로 ROI를 설정할 수 있는 방법을 고안하였다.

원활한 차량 카운트를 위해서 평균 크기의 차량이 도로를 차지하는 면적을 검지 영역(ROI)으로 설정한다. 우선 평균 차량의 길이에 해당되는 간격으로 도로를 두 개의 수평선을 설정한 후 그림 9의 (a)와 같이 그 바깥 영역의 픽셀 값들은 모두 0으로 처리한다. 그런 다음 그림 9의 (b)와 같이 rectangle blob 처리를 수행하여 각 차선별 ROI를 설정한다. 이러한 방법은 차선의 수와 방향에 관계없이 적용할 수 있다.

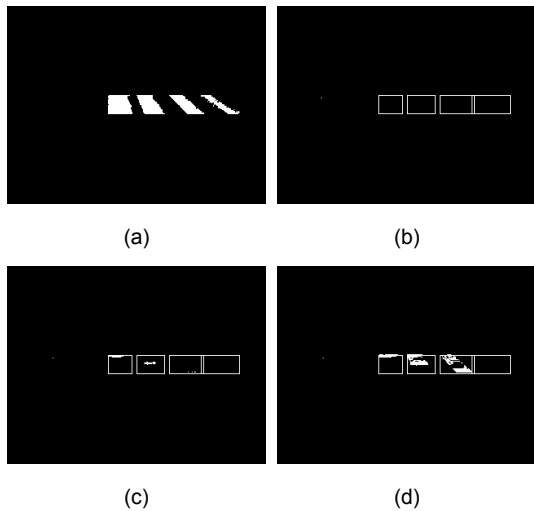


그림 9. (a) 누적도로 ROI 구간분할, (b) ROI 영역 Blob처리 영상, (c) ROI 영역에서 관찰된 도로영역내의 모폴로지 처리된Sub-shadow 영상(즉, 그림 7과 그림 8 간 AND 영상), (d) ROI 영역에서 관찰된 도로영역내의 배경제거 영상(즉, 그림 4와 그림 8 간 AND 영상)

Fig. 9 (a) ROI division of accrued road, (b) ROI blobbing, (c) Morphological sub-shadow image projected into ROI and the accrued road, (d) Vehicle image projected into ROI and the accrued road

3.4. 차량 카운트 알고리즘(Vehicle Count Algorithm)

차량을 카운트하는 방법 중 연산량이 가장 적은 방법은 ROI 영역 내의 픽셀 히스토그램(이진 이미지에서는 객체의 면적을 의미)을 이용하는 것이다. 본 논문에는 그림 9의 (c), (d) 각 이미지에서 픽셀값이 '1'인 영역의 면적을 ROI 내의 차선 면적으로 나눈 비율을 체크하여 단계적으로 모니터링을 수행한다.

또한 본 논문에서는 차량 진행 방향 인지를 위해 앞서 지정한 ROI 영역을 위, 아래로 반씩 나누어 위 영역을 R1, 아래 영역을 R2로 정의하여 각각 차량의 주행방

향에 따라 다른 역할을 하도록 한다.

하행 차량의 카운트 알고리즘은 다음과 같다. 하행 차량의 sub-shadow는 차량의 전면부 아래에 생성되므로, 차량이 ROI에 접근할 때 해당 차량의 Sub-shadow가 차량 본체 영역보다 먼저 들어오는 특징이 있다. 이를 착안하여, 그림 9 (c)의 sub-shadow 위치와 그림 9 (d)의 히스토그램 상태에 따라 세 개의 단계(Level)를 설정하고, 각 단계를 순차적으로 통과하여야만 카운트를 수행하도록 한다. 본 과정은 각 차선별로 수행된다. 초기 0 단계에서 R1 영역에 그림 9 (c)와 같이 sub-shadow의 픽셀이 포함되면 변수 $Level_{down}$ 을 1단계로 설정한다. 여기서 sub-shadow의 포함은 픽셀의 수로 판단하는데, 본 연구에서는 sub-shadow가 ROI지역을 지날 때 그 픽셀의 수가 최소 20 개 이상으로 관찰되어 임계값을 20으로 설정하였다. 이 상태에서 차량이 각 ROI 구간(R1+R2)으로 진입하여 그림 9 (a) 대비 그림 9 (d) 간 면적 비율이 50% 이상이면 하행 차량 카운트(변수 Down)를 증가시키고 $Level_{down}$ 를 2단계로 설정한다. 모든 단계에 대해 R1영역에서의 면적 비율(즉, 그림 9 (a) 대비 그림 9 (d)의 면적 비율)이 30% 미만일 경우는 $Level_{down}$ 을 초기상태인 0단계로 되돌린다. 아래 그림 10은 하행차량 카운트알고리즘을 그림으로 나타낸 것이다.

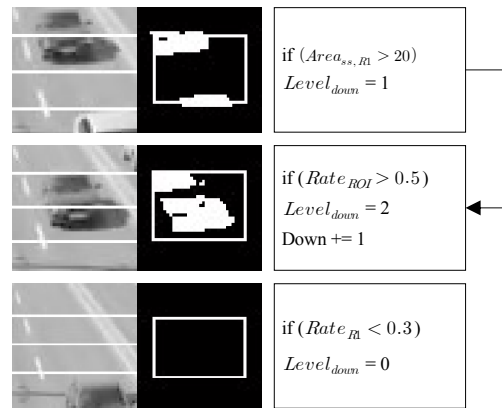


그림 10. 하행 차량 카운트 알고리즘

Fig. 10 The Count Algorithm for vehicles which are going down

상행 차량의 카운트알고리즘은 다음과 같다. 상행 차량의 sub-shadow는 차량의 후미 아래에 생성되므로,

차량이 ROI에 접근할 때 차량의 sub-shadow가 가장 나중에 들어오는 특징이 있다. 이를 착안하여, 하행의 경우와 마찬가지로 세 개의 단계를 설정하고, 각 단계를 순차적으로 통과하는 경우에만 카운트를 수행한다. 초기 0 단계에서 그림 9 (a) 와 그림 9 (d) 간 면적 비율이 50% 이상이고, R2 영역에서의 sub-shadow 픽셀수가 임계값 20 미만이면 $Level_{up}$ 을 1단계로 설정한다. 이 상태에서 차량이 전진하여 sub-shadow의 픽셀수가 임계값 20을 넘으면 상향카운트를 증가시키고 $Level_{up}$ 을 2 단계로 설정한다. 모든 단계에 대해 R1에서의 면적 비율이 30% 미만일 경우는 $Level_{up}$ 을 초기 0 단계로 되돌린다. 아래 그림 11은 하행차량 카운트알고리즘을 설명하고 있다.

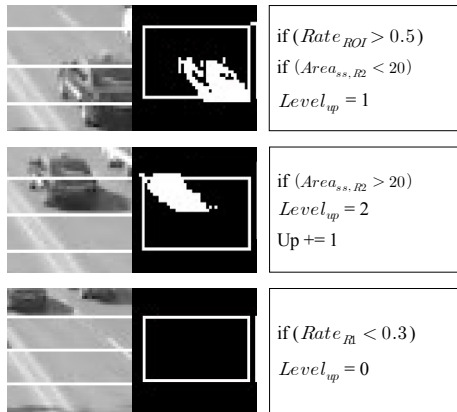


그림 11. 상행 차량 카운트 알고리즘
Fig. 11 The Count Algorithm for vehicles which are going up

참고로 상행 차량은 하행의 경우와 달리 sub-shadow의 누적과정을 통해 차선과 ROI 영역이 설정된 이후에 그 다음 차량부터 카운트가 발생하기 때문에 위 알고리즘을 최초로 적용할 때에는 상행 차량 카운트 결과에 한 대를 가산해줘야 한다.

본 알고리즘은 상행, 하행 방향 및 차선 수를 자동으로 인식하여 카운트하기 때문에 상행, 하행 별 그리고 각 차선별 교통량 측정이 가능하다.

3.5. 성능 측정(Performance Test)

지금까지 소개된 알고리즘을 프로그램 하여 다양한 실제 도로 영상에 적용하여 그 실험 결과를 표 1에 정리

하였다. 적용된 시나리오는 7가지로서 3~4월 일몰 후인 09시부터 일몰 전인 18시까지의 낮 상황에 대한 영상 샘플들이다. 영상간의 시간간격은 최대 2시간 30분정도이다.

표 1. 제안하는 차량카운트방법 실험 및 비교데이터
Table. 1 Experiment result of the proposed vehicle count method and comparison with the existing method

No	(A)	(B)	(C)	(D)
1	66, 81	110(74.8%)	68, 79	97.0%, 97.5%
2	33, 28	60(98.4%)	34, 24	97.0%, 85.7%
3	42, 35	74(96.1%)	41, 34	97.6%, 97.1%
4	55, 61	101(94.8%)	54, 58	98.2%, 95.1%
5	38, 50	99(88.9%)	38, 51	100%, 98.0%
6	22, 15	43(86.0%)	24, 14	91.7%, 93.3%
7	67, 71	151(91.4%)	68, 77	98.5%, 92.2%
		90.1%		97.1%, 94.1%

* (A)실제 하행차량수와 상행 차량수, (B)기존 카운트 방식(ICA) 결과, (C)제안하는 알고리즘으로 카운트한 결과, (D)제안하는 알고리즘의 검지율
* (A) The number of vehicles per up/down lane, (B) Result of ICA method, (C) Count result using the proposed method, (D) Detection rate of the proposed method

실험 결과 제안하는 알고리즘은 하행 차량의 경우 평균 97.1%, 상행 차량의 경우 평균 94.1%의 검지율을 보였다(표 1의 (D) 열). 알려진 바에 따르면 기존 VAR 방식[7]이 89.8%, 상행 및 하행 구분을 못하고 차량을 카운트 하는 기존 ICA 방법[6]이 90.1% ((B) 열) 인 것으로 측정된 것을 감안하면 상당히 개선된 수준이다. 또한 각 차선별, 상, 하행별 자동 카운트가 가능하고 특히, 가장 많이 설치 운영되고 있는 루프검지기의 성능 95%를 능가하기 때문에 상용 가치에 근접한 것으로 보인다.

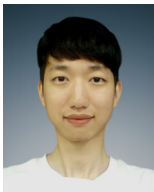
본 실험의 시나리오 중에는 바람이나 기타 환경변화로 인한 카메라의 흔들림이 심한 경우도 있었으나 실험 결과 다른 일반 시나리오의 결과와 차이가 거의 없어 촬영 환경 변화에 강한 것을 확인할 수 있었다. 또한, 시간대별 태양 위치에 따른 그림자 길이에 관계없이 안정적인 검출이 가능함도 확인할 수 있었다.

IV. 결 론

영상 검지기 분야에는 기존에 많은 성능 좋은 알고리즘이 있지만 실시간 연산 요구량이 너무 많아 시스템 장비가 고가, 고전력을 소모하였다. 이에 본 논문에서는 저가, 저전력 영상검지시스템 구현을 위해 안드로이드 플랫폼의 성능 사양에 적합한 저연산량의 알고리즘을 고안했으며 그 결과 검지율이 상용 루프검지기에 버금가는 95% 이상인 것으로 파악되어 만족스러운 연구 결과를 얻었다. Sub-shadow을 이용하여 차선간격을 자동으로 설정하고 실시간으로 그 간격을 조정하고 sub-shadow 와 차량 객체 영역을 상호 보완적으로 이용하여 차량 카운트를 수행하기 때문에 측정 결과가 매우 정확한 편이다. 또한, 주행방향 및 차선에 따라 독립적으로 계수가 가능하였다. 또한, 본 알고리즘은 차량 바로 아래 생기는 sub-shadow 를 이용하기 때문에 밤 시간이나 태양의 고도가 매우 낮아 차량 아래 그림자가 생기지 않는 경우는 검지율이 낮아진다. 따라서 향후 후속 연구에서는 sub-shadow가 생기지 않는 야간 또는 새벽시간대에서 sub-shadow 대신 차량라이트를 이용하여 교통량을 측정할 수 있도록 알고리즘을 보완하는 한편, 하행차량 보다 낮은 상행차량의 검지성능을 높이는 데 집중할 예정이다.

감사의 글

본 연구의 일부는 미래창조과학부 및 정보통신산업진흥원의 IT융합고급인력과정지원사업의 연구 결과로 수행되었으며(NIPA-2014-H0401-14-1007) 나머지 일부는 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2012R1A1A1038515)



김지완(Jaesung Lee)

2010년 3월 ~ 현재 한국교통대학교 학부생
 ※관심분야: 영상 처리, 멀티미디어 SoC 설계

REFERENCES

- [1] Suk-Tae Kim et. al., "The Next Generation Smart Vehicle Detection Technology, SMART-I", *The Road Traffic*, pp 26-33, no. 129, 2012.
- [2] Suk-il Song, Jaesung Lee, Gyoony-Byung Ko, and Chul Moon, "Research Trend on Elementary Technologies of the Next Generation Smart Transportation System", *Information and Communications Magazine*, pp. 18-pp. 24, vol. 30, no. 10, October 2013.
- [3] Barron, J.L., D.J. Fleet, S.S. Beauchemin, and T.A. Burkitt. Performance of optical flow techniques. *CVPR*, 1992.
- [4] Lowe, David G., "Object recognition from local scale-invariant features", *Proceedings of the International Conference on Computer Vision*, pp. 1150-1157. 1999.
- [5] http://www.inpai.com.cn/doc/hard/198143_-3.htm
- [6] Tekalp, A. M., Smolic, A., Vetro, A., and Onural, L., Eds. "Tracking and counting vehicles in traffic video sequences using particle filtering ," vol. 99, 4. *Proceedings of the IEEE*, 2011.
- [7] In Jung Lee, Joon Young Min, Hyoung Lee, "An Algorithm for Analysing Occulsion using ICA Method", *Proceedings of KITA conference* pp.457-466, 2004.
- [8] Kim, Giseok, and Jae-Soo Cho. "Vision-based vehicle detection and inter-vehicle distance estimation." *Control, Automation and Systems (ICCAS), 2012 12th International Conference on. IEEE*, 2012.
- [9] Jun-Hee Cho, Hee-Sung Kim, "Counting the number of cars waiting at the traffic light by computer vision process" *Proc. of the 30th KIISE Fall Conference*, Vol. no.2, pp.562-564, Oct. 2003.
- [10] Piccardi, Massimo. "Background subtraction techniques: a review." *Systems, man and cybernetics, 2004 IEEE international conference on*. Vol. 4. 2004.
- [11] Bong-Keun Kim, "A Study of Non-ROI Real-time CCTV Visibility Measurements for Highway Fog Warning System." *Journal of the Korea Academia-Industrial cooperation Society conference*, 709-712 , 2009 .



이재성(Jaesung Lee)

2008년 8월 서울대학교 전기컴퓨터공학부 박사
2001년 2월 ~ 2011년 8월 한국전자통신연구원(ETRI) 선임연구원
2011년 9월 ~ 2013년 9월 한국교통대학교 조교수
2013년 10월 ~ 현재 한국교통대학교 부교수
※관심분야 : 영상 처리, 멀티미디어 SoC 설계