

# Mean Shift를 이용한 교차로 교통량 측정 시스템 개발\*

## The Development of Vehicle Counting System at Intersection Using Mean Shift

천인국\*  
(In-Gook Chun)

### 요 약

비디오 카메라에서 입력된 동영상을 분석하여 교차로에서 교통량을 자동적으로 측정하는 시스템을 설계, 구현하였다. 입력 영상에서 배경과 전경을 분리하기 위하여 3가지 방법을 비교 분석하였으며 이 중에서 좋은 결과를 보여주는 Li의 방법을 선택하였다. 전경 영상에서 연결 성분 분석을 이용하여 각각의 블로브들을 분리하였으며 분리된 블로브들은 블로브 추적기를 이용하여 프레임 별로 추적된다. 가장 기본적인 추적기는 블로브의 크기와 위치 정보들을 이용한다. 블로브들 간에 충돌이 있는 경우에는 블로브 안의 컬러 분포를 이용하는 mean shift 알고리즘이 사용되었다. 제안된 시스템은 실제 교차로 동영상을 이용하여 테스트되었으며 휴리스틱을 추가할 경우, 좋은 감지율과 오차율을 보였다.

### Abstract

A vehicle counting system at intersection is designed and implemented using analyzing a video stream from a camera. To separate foreground image from background, we compare three different methods, among which Li's method is chosen. Blobs are extracted from the foreground image using connected component analysis and the blobs are tracked by a blob tracker, frame by frame. The primary tracker use only the size and location of blob in foreground image. If there is a collision between blobs, the mean-shift tracking algorithm based on color distribution of blob is used. The proposed system is tested using real video data at intersection. If some huristics is applied, the system shows a good detection rate and a low error rate.

**Key words** : Vehicle counting, traffic counting, background estimation, traffic measurement

## I. 서 론

도로의 교통량을 측정하는 것은 지능형 교통 시스템(ITS : Intelligent Transport Systems)에서 중요한 요소이다. 대표적인 교통 정보 수집 장치에는 루프

검지기(ILD, Inductive Loop Detector)를 이용하는 방법과 비디오 카메라를 이용하는 방법이 있다 [1]. 유도식 측정기는 많이 사용되는 신뢰성 있는 방법이지만 도로에 매설되어야 하므로 설치와 이동이 어렵다. 또한 자기장의 간섭의 변화량을 측정하는

본 연구는 2005학년도 순천향대학교 교수연구년제로 지원을 받아 수행하였음.

\* 주저자 : 순천향대학교 공과대학 컴퓨터학부 교수

† 논문접수일 : 2008년 6월 4일

† 논문심사일 : 2008년 6월 10일

† 게재확정일 : 2008년 6월 23일

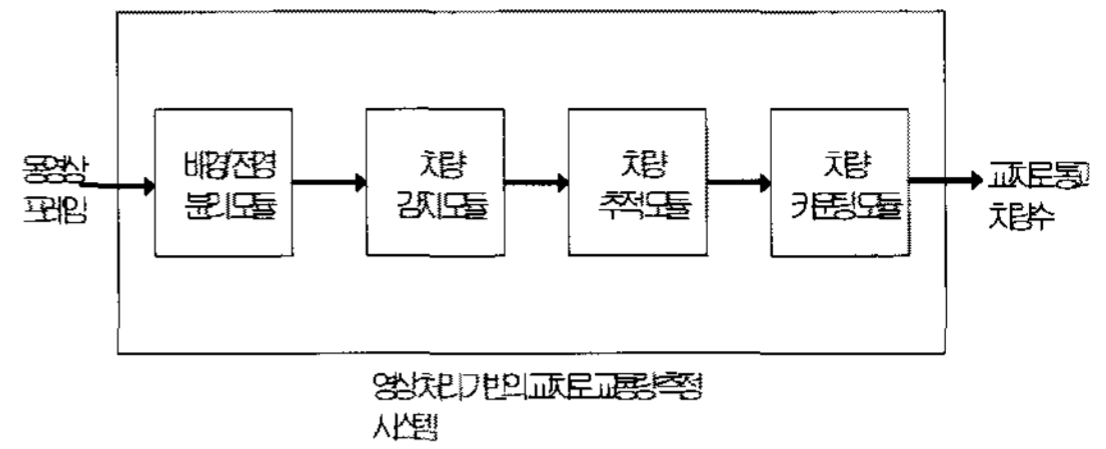
방법이므로 차량이 천천히 움직이거나 정지하는 경우에는 검출이 힘들다. 반면에 비디오 카메라를 이용하는 방법은 동영상을 입력받아 분석하는 방법으로서 설치와 이동이 간단하지만 아직까지도 신뢰성은 낮은 편이다. 그러나 비디오 카메라를 이용하게 되면 영상 처리를 통해 얻어진 도로 정보와 영상을 중앙 통제소에 전달하여 도로의 상황에 맞는 적절한 통제를 할 수 있는 추가적인 장점도 있다 [2].

영상 처리를 이용하여 교통량을 측정하는 방법으로는, 크게 나누어 루프 검지기처럼 영상 내에 검지 영역을 설정하여 차량을 인식하는 방법과 전체 영상에서 차량을 분할하고 추적하는 기법이 있다. 영상 내에 검지 영역을 설정하는 방법에는 동일한 차선에 연속적인 두 개의 검지 영역을 설정하여 차량의 속도를 판단하는 기법 [3], 차선과 나란한 지역적인 영역에서 차량 영역을 분할하여 추적하는 기법 등이 있다 [4]. 전체 영상에서 차량을 분할하여 추적하는 기법에는 차량과 배경 영상과의 차이를 이용하여 차량을 분리하는 방법 [5]과 차량의 움직임을 인식하여 차량을 분리하는 방법 [6] 등이 있다. 지금까지의 대부분의 교통량 측정 시스템은 혼잡한 교차로가 아닌 일반 도로를 대상으로 한 것으로 이들 방법들은 모두 차량의 흐름이 복잡하지 않고 가려짐이 없는 환경에서는 잘 동작하지만 혼잡한 교차로에서는 인식율이 현저하게 떨어진다.

본 논문에서는 교차로 동영상에서 개개의 차량을 인식하고 추적하여서 좌회전, 직진, 우회전 등의 교통량 정보를 집계하는 시스템을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 전체 시스템의 구성을, 3장에서 배경과 전경의 분리 알고리즘에 대하여 기술하였다. 4장에서는 차량의 감지와 추적 알고리즘을, 5장에서는 차량을 카운팅하는 알고리즘에 대하여 기술하였다. 6장에서는 실제 입력 동영상을 이용한 실험 결과와 분석을 하였고 7장에서는 결론과 문제점에 대하여 기술하였다.

## II. 전체 시스템의 구성

교통량 측정 시스템의 입력은 비디오 카메라에



<그림 1> 교차로 교통량 측정 시스템의 구성  
<Fig. 1> Modules of vehicle counting system for intersection

서 전달되는 동영상 스트림이다. 비디오 카메라에서 입력되는 영상을 분석하여 배경에서 전경을 분리하고 각각의 자동차를 탐지하고 추적하고, 필요하다면 교차원적인 분석을 수행한다. 교차원적인 분석이란 예를 들어 차량의 크기, 차량의 종류 등을 분석하는 것을 의미한다. 교차로 교통량 시스템의 전체 구성은 <그림 1>과 같다.

교차로 교통량 측정 시스템은 다음과 같은 모듈들을 가진다. 각각의 모듈에 대하여 순차적으로 기술한다.

- ① 배경/전경 분리 모듈: 입력 영상의 각각의 픽셀이 배경인지 전경인지를 결정한다.
- ② 차량 감지 모듈: 전경 영상을 받아서 새로운 차량이 영상 안에 나타났는지를 감지한다.
- ③ 차량 추적 모듈: 차량 리스트를 이용해서 각각의 차량을 추적한다.
- ④ 차량 카운팅 모듈: 진출 감지선과 진입 감지선을 지나는 차량들을 분류하여 카운팅한다.

## III. 배경과 전경의 분리

교통량을 측정하기 위해서는 먼저 입력 영상에서 배경이 분리되어야 한다. 배경과 전경을 분리하는 것은 시스템의 첫 번째 단계이자 매우 중요한 작업이다. 이 단계의 정확도는 전체 시스템의 정확도에 많은 영향을 끼친다.

일반적으로 입력 영상이 실내에서 촬영되었다면 배경과 전경의 분리 작업은 비교적 쉽다. 그러나 교차로 교통량 측정 시스템처럼 실외에서 촬영되는

경우, 카메라의 흔들림, 흔들리는 나무 가지, 밝기의 변화, 그림자 등으로 인하여 배경이 끊임없이 변화한다. 따라서 고정된 배경 영상으로는 좋은 분리를 할 수 없고, 배경 영상은 반드시 시간에 따라서 변화되어야 한다.

현재까지 배경과 전경을 분리하기 위하여 많은 방법들이 고안되었다 [7, 8]. 본 논문에서는 일반적으로 많이 사용되는 3가지 방법을 구현하여 테스트 하였다.

첫 번째 방법은 각 픽셀 위치에서 시간적인 메디안 필터링을 사용하는 방법이다. 이 방법에서 배경 영상의 각 픽셀은 최근 N개 영상의 픽셀값들의 중간값(median)으로 설정된다. 이 방법은 하나의 위치에서 시간적으로 가장 많이 등장하는 픽셀값이 배경일 거라는 논리에 근거한다. 이 방법은 비교적 구현하기가 쉽고, 성능도 우수한 편이지만 중간값을 구하려면 정렬을 해야 하기 때문에 계산량이 많은 편이다.

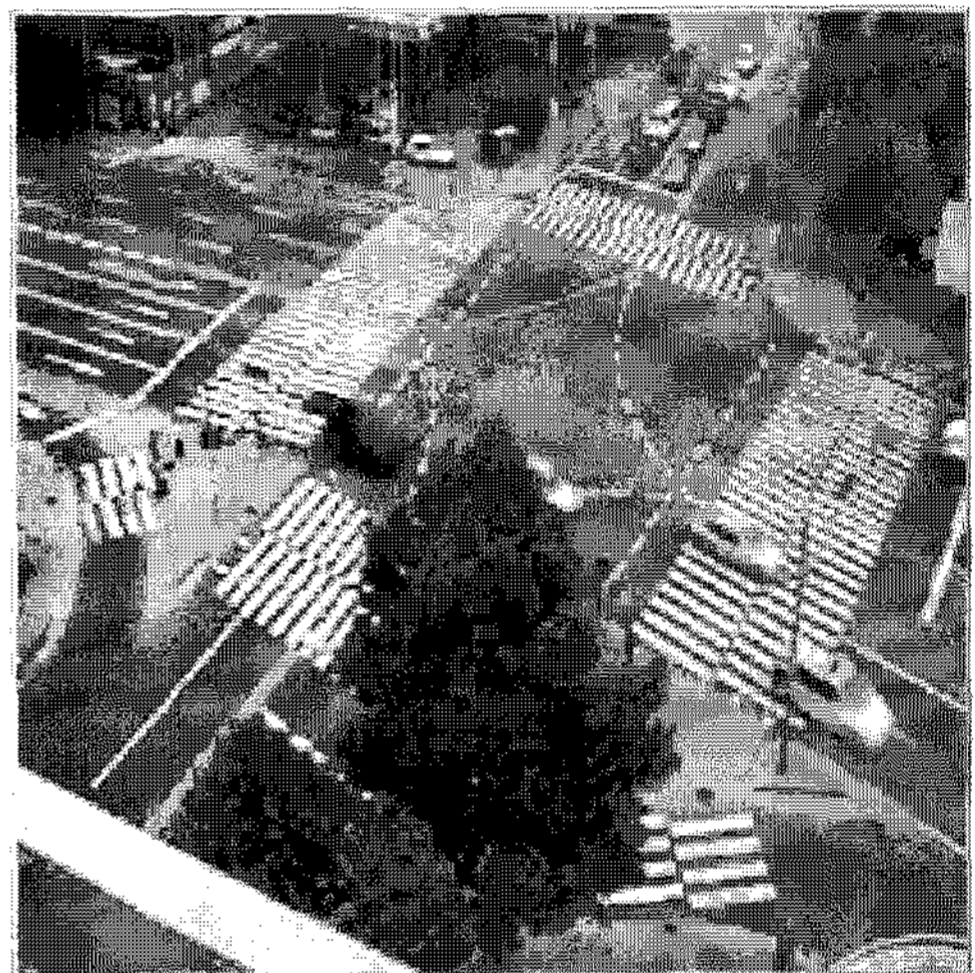
두 번째 방법은 Li에 의하여 제안된 방법이다 [9]. Li의 방법은 정지 물체와 움직이는 물체가 동시에 존재하는 복잡한 배경도 처리할 수 있다고 주장되었다. 이 방법은 선택된 특징 벡터로부터 배경과 전경을 분리하기 위하여 Bayes 법칙을 사용한다. 정지 물체 배경은 색상 특징으로 기술되고, 움직이는 배경 물체는 색상 동시 발생(co-occurrence) 특징으로 표현된다. 픽셀의 색상과 색상 동시 발생 분포는 히스토그램에 의하여 표현된다. Bayes의 법칙이 하나의 픽셀을 배경, 또는 전경으로 분류하는데 사용된다. 픽셀의 분류가 끝나면 배경이 업데이트된다. 이 알고리즘은 배경이 점차적으로 변하는 경우나 배경의 갑작스러운 변화, 정지 물체 들을 성공적으로 처리한다.

세 번째 방법은 각각의 픽셀을 독립적으로 K개의 Gaussian 확률 분포의 혼합체(Adaptive Mixture of Gaussian)로 나타내는 방법이다[10]. 이 방법에서 하나의 픽셀은 다음과 같은 식으로 표현될 수 있다.

$$P(I_t) = \sum_{i=1}^K \omega_{i,t} \eta(I_t; \mu_{i,t}, \sigma_{i,t}^2)$$

여기서 K=4이고  $\omega$ 는 가중치,  $\eta$ 는 Gaussian 분포를 의미한다. 이 방법에서는 전경이 검출되기 전에 배경이 업데이트된다.

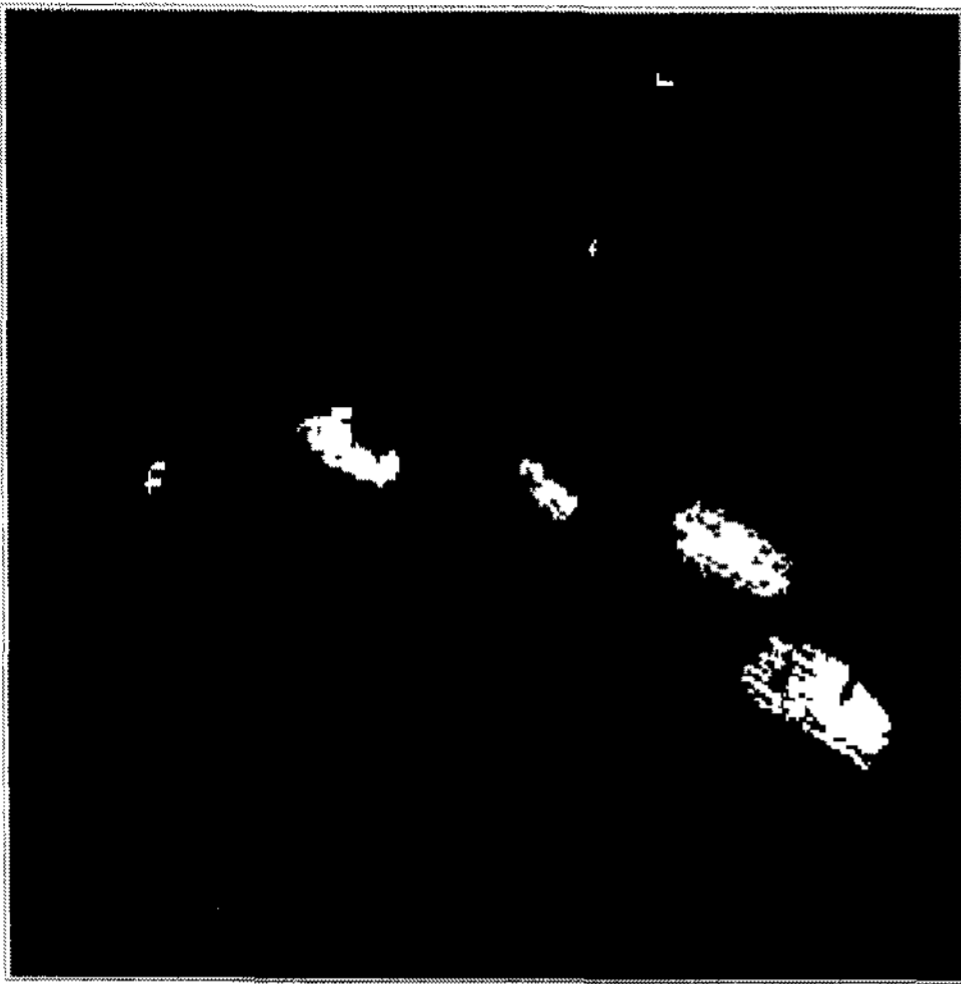
이들 세 가지 방법 중에서 Li의 방법이, 배경이 갑자기 변환하였을 경우, 복원력이 우수하고, 실행 시간 면에서도 양호한 것으로 판단되어서 본 논문에서 채택되었다. <그림 2>는 동일한 실험 영상에 대하여 위의 세 가지 방법을 적용하여 전경 영상을 구해본 결과이다.



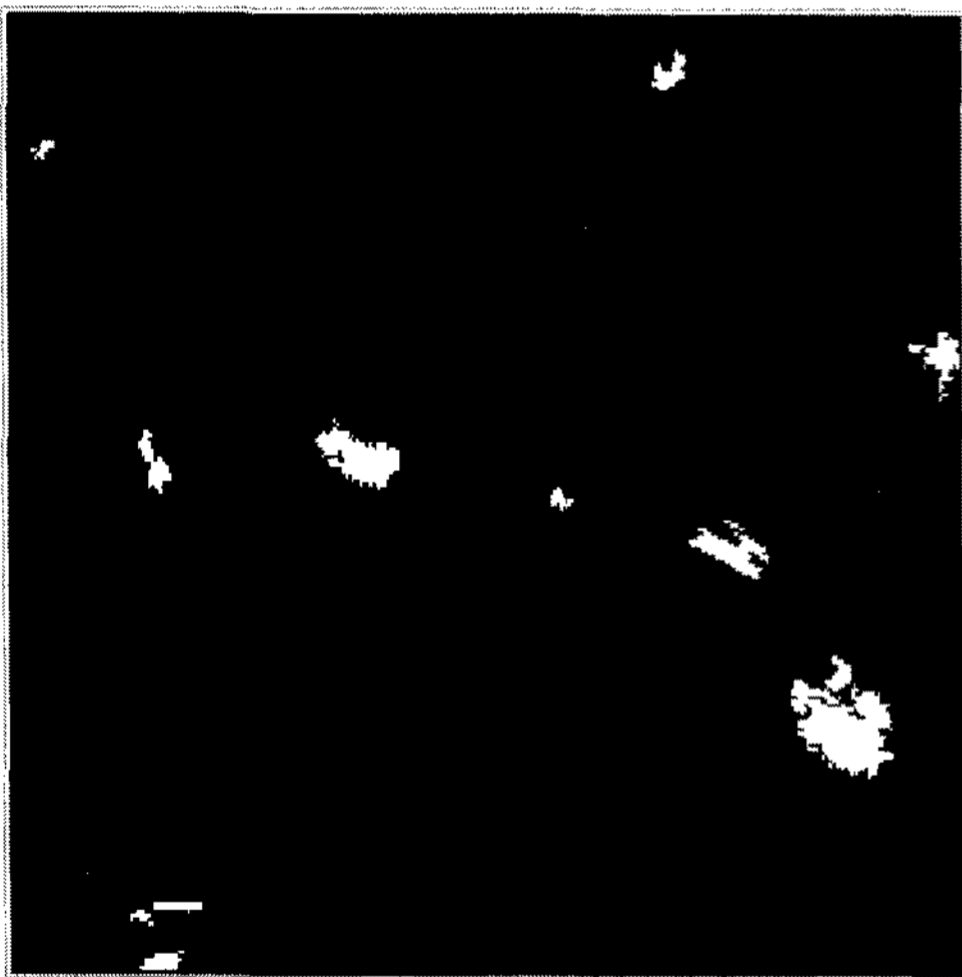
(a) 입력 영상  
(a) Input image



(b) 메디안 필터링 방법  
(b) Median filtering



(c) Li의 방법  
(c) Li's method



(d) K개의 Gaussian 확률 분포의 혼합체  
(d) K Gaussian mixture model

<그림 2> 전경과 배경의 분리 방법 비교  
<Fig. 2> Comparison of foreground and background separation method

#### IV. 차량 추적

실시간으로 물체를 추적하는 것은 많은 컴퓨터 비전 응용에서 중요한 작업이다. 교통량 측정에서도 추적 기능은 필요하다. 왜냐하면 현재 감지된 차량이 앞의 프레임에 있었던 차량인지, 새롭게 등장한 차량인지를 구별하지 못하면 교통량 측정의 결

과가 달라지기 때문이다.

차량 추적 모듈은 차량을 식별하여서 각 프레임에서의 차량의 위치와 크기 정보를 제공한다. 여기서는 2가지의 방법을 혼용하여 사용하였다. 첫 번째 방법은 단순 블로브 추적기이다. 단순 블로브 추적기는 블로브 사이에 겹침이 없는 경우에 신뢰성 있고 빠른 추적 성능을 보여준다. 만약 블로브 사이에 겹침이 있다면 mean shift 알고리즘을 사용한다.

본 논문에서의 차량 추적 알고리즘은 기본적으로 연결 성분 분석(connected component analysis)에 기반을 두고 있다. 먼저 배경에서 분리된 전경 영상에 대하여 연결 성분 분석을 수행한다. 이 결과로 생성되는 각각의 연결 성분, 즉 블로브(blob)는 차량일 가능성이 있는 물체 후보로 간주된다. 블로브들을 추적하기 전에 블로브들 중에서 크기가 아주 작은 블로브는 크기 필터(size filter)를 이용하여 제거한다.

또한 두 개의 블로브 리스트가 사용된다. 하나는 추적 중인 블로브들을 관리하는 blob\_list이고 또 하나는 현재 프레임에서 새롭게 발견된 블로브들을 저장한 new\_blob\_list이다. new\_blob\_list는 매 프레임마다 초기화된다. 전경 영상에 대하여 연결 성분들을 추출하고 이것을 new\_blob\_list에 등록한다. 이어서 blob\_list에 있었던 기존의 블로브에 대해서 칼만 필터(Kalman filter)를 이용하여 블로브의 위치를 예측하여 기록한다. 다음 절차는 blob\_list에 있는 블로브와 new\_blob\_list에 있는 블로브들을 매칭하는 단계이다. blob\_list에 있는 각각의 블로브에 대하여 new\_blob\_list에 있는 블로브가 대응되는지를 점검한다. 만약 블로브들의 바운딩 박스가 겹쳐지면 일단 동일한 블로브라고 판단하고 이것을 블로브의 hypothesis\_list에 기록한다. 만약 대응되는 블로브의 개수가 하나이면 단순 블로브 추적기를 사용하고, 대응되는 블로브의 개수가 두 개 이상이면, 충돌이라고 가정하고 mean shift 알고리즘을 이용하여 추적한다.

##### 1. 단순 블로브 추적기

블로브의 hypothesis\_list에 기록된 블로브의 개수가 하나인지 두 개인지를 검사한다. 만약 하나이면

대응되는 블로브는 이미 찾은 것이다. 만약 대응되는 블로브가 두 개 이상이면 추적중인 블로브와 대응되는 후보 블로브들 사이의 거리를 계산한다. 가장 짧은 거리를 가지는 블로브가 매칭 블로브가 된다.

## 2. mean-shift 블로브 추적기

차량들이 근접하여 지나가는 경우, 전경 영상에서 존재하는 연결 성분 정보만 가지고는 정확히 추적할 수 없다. 따라서 입력 영상의 추가적인 정보를 이용하여야 한다. 하지만 차량의 형태를 가지고 차량을 추적하는 방법들의 경우, 추적 알고리즘이 간단해지는 장점이 있으나 차량이 기울어지거나 차량의 일부가 가로수에 의하여 가려지는 경우에 형태의 변화가 일어나 정확히 추적하기 힘들다. 이러한 단점을 해결하는 방법으로 컬러 분포를 이용하는 방법이 각광을 받고 있다 [11]. 차량의 컬러 분포는 차량의 기울어짐이나 일부가 가려지더라도 크게 영향을 받지 않는다. 컬러 분포는 흔히 컬러 히스토그램으로 나타내며 여기서는 RGB 컬러 공간을 사용하고 총  $m$ 개의 빈(bin)이 사용된다.

먼저 과거 프레임으로부터 추적할 차량의 컬러 분포를 추출하여 차량 모델로 가정한다. 현재 프레임에서 추적 차량의 위치를 예측하고 이 위치를 중심으로 주변을 탐색하여 컬러 분포가 가장 유사한 후보 영역을 찾아낸다. 단순한 탐색 알고리즘을 이용하면 계산량이 너무 많아지므로 최근 많이 사용되는 mean shift 알고리즘을 사용하였다 [11].

먼저 컬러 분포를 이용하여 차량 모델과 차량 후보를 확률 밀도 함수(pdf) 형태로 나타내면 다음과 같다.

- ① 차량 모델:  $q = \{q_u\}_{u=1\dots m}$        $\sum_{u=1}^m q_u = 1$
- ② 차량 후보:  $p = \{p_u\}_{u=1\dots m}$        $\sum_{u=1}^m p_u = 1$

여기서 특정한 위치의 픽셀을 컬러 빈에 대응시키는 함수  $b: R^2 \rightarrow \{1\dots m\}$ 을 가정하면 차량 모델과 차량 후보의 확률을 다음과 같이 나타낼 수 있다.

$$q_u = C \sum_{i=1}^n k(\|x_i\|^2) \delta[b(x_i) - u]$$

$$p_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y-x_i}{h}\right\|\right) \delta[b(x_i) - u]$$

여기서  $n$ 와  $n_h$ 는 차량을 나타내는 블로브 안의 픽셀의 개수이고  $\delta$ 는 Kronecker delta 함수이다.  $C$ 와  $C_h$ 는 전체 확률을 1로 만들기 위한 정규화 상수이다.  $k(x)$ 는 단순히 중심점에서 멀리 떨어진 픽셀에 더 적은 가중치를 주기 위한 커널 함수이다.  $h$ 는 커널의 대역폭(bandwidth)을 의미한다 [11].

차량 모델의 확률 밀도 함수  $q_u$ 와 영상 내의 중심 좌표가  $y$ 인 차량 후보의 확률 밀도 함수  $p_u(y)$  사이의 유사성 척도로는 다음과 같이 정의되는 Bhattacharyya 계수가 많이 사용된다 [11].

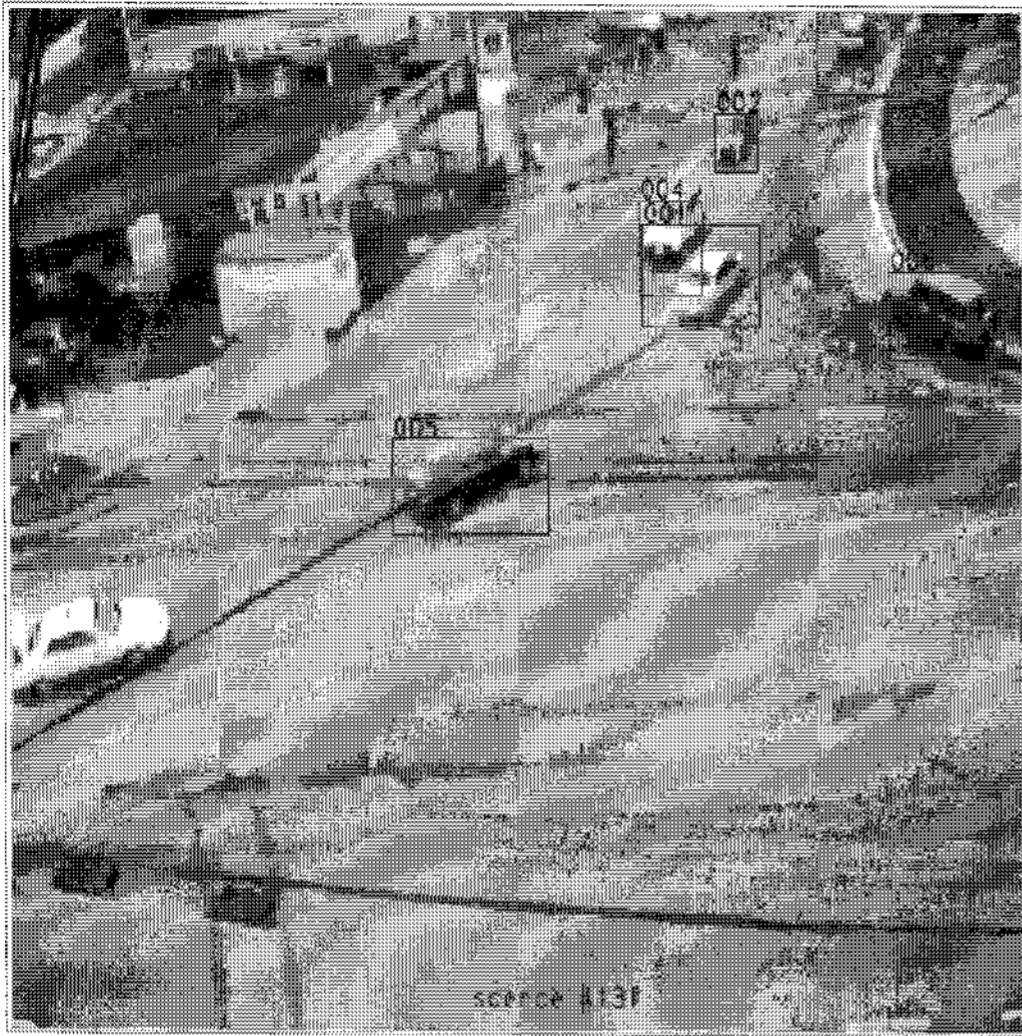
$$\rho(y) = \rho[p(y), q] = \sum_{u=1}^m \sqrt{p_u(y)q_u}$$

영상에서의 어떤 위치에서 계산된 Bhattacharyya 계수의 크기는 그 위치에서 추적하려고 하는 물체가 존재할 수 있는 확률을 나타내게 되며, 따라서 물체를 추적하는 과정은 Bhattacharyya 계수가 최대가 되는 위치를 추적하는 과정과 동일하다. 차량 모델이 중심 좌표  $y_0$ 인 위치에 있을 때 Bhattacharyya 계수가 최대로 증가되는 새로운 위치의 중심 좌표  $y_1$ 은 mean shift 벡터를 이용하여 다음 식으로 표현할 수 있으며, 이 식을 좌표값의 변화가 일정값 이하로 작아질 때까지 몇 번 반복 적용하여 Bhattacharyya 계수가 최대인 위치를 구할 수 있다.

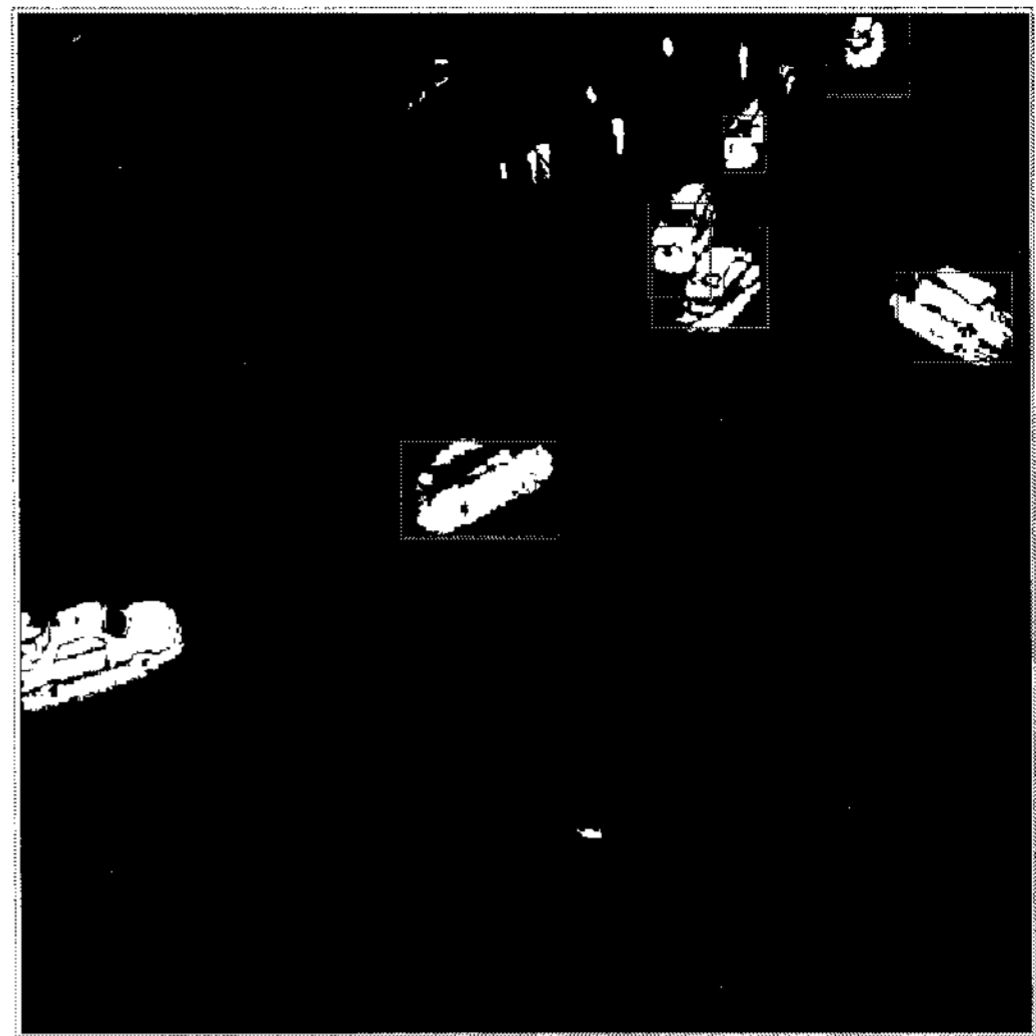
$$y_1 = \frac{\sum_{i=1}^{n_h} x_i \omega_i g\left(\left\|\frac{y_0-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} \omega_i g\left(\left\|\frac{y_0-x_i}{h}\right\|^2\right)}$$

여기서  $g(x) = -k'(x)$ 이고,  $\omega_i$ 는 가중치로서 다음과 같이 계산된다.

$$\omega_i = \sum_{u=1}^m \sqrt{\frac{q_u}{p_u(y_0)}} \delta[b(x_i) - u]$$



(a) 131번째 프레임  
(a) 131th frame



(b) 131번째 프레임의 블로브 영상  
(b) 131th frame blob mage



(c) 150번째 프레임  
(c) 150th frame



(d) 150번째 프레임의 블로브 영상  
(b) 150th frame blob image

<그림 3> Mean-shift 알고리즘을 이용한 차량 추적의 예  
<Fig. 3> Example of vehicle tracking using mean shift algorithm

Mean shift 기법은 전체 탐색을 사용하지 않으므로 계산량이 적고 실시간 구현에 적합하다. Mean shift을 이용한 차량 추적의 예를 <그림 3>에서 보였다. <그림 3>에서 추적 번호 001인 차량과 004인 차량이 근접하여 주행하는 경우에, 만약 연결 성분

만을 이용하여 처리하였다면 두 대의 차량은 서로 연결된 것으로 인식되고 따라서 하나의 차량으로 간주된다. 본 논문에서는 차량이 일정 거리 안으로 접근하게 되면 mean shift 기법을 이용하여 차량을 추적하기 때문에 두 대의 차량을 분리하여 추적할

수 있다.

### 3. 사후 처리

블로브 중에서 블로브의 상태가 정상이 아니면 블로브는 에러 카운터가 증가된다. 블로브의 위치, 크기, 움직임, 평균 밝기 등을 종합적으로 판단하여서 이상 여부를 판단한다. 만약 블로브의 에러 카운터가 일정 횟수 이상이 되면 블로브는 블로브 리스트에서 삭제된다.

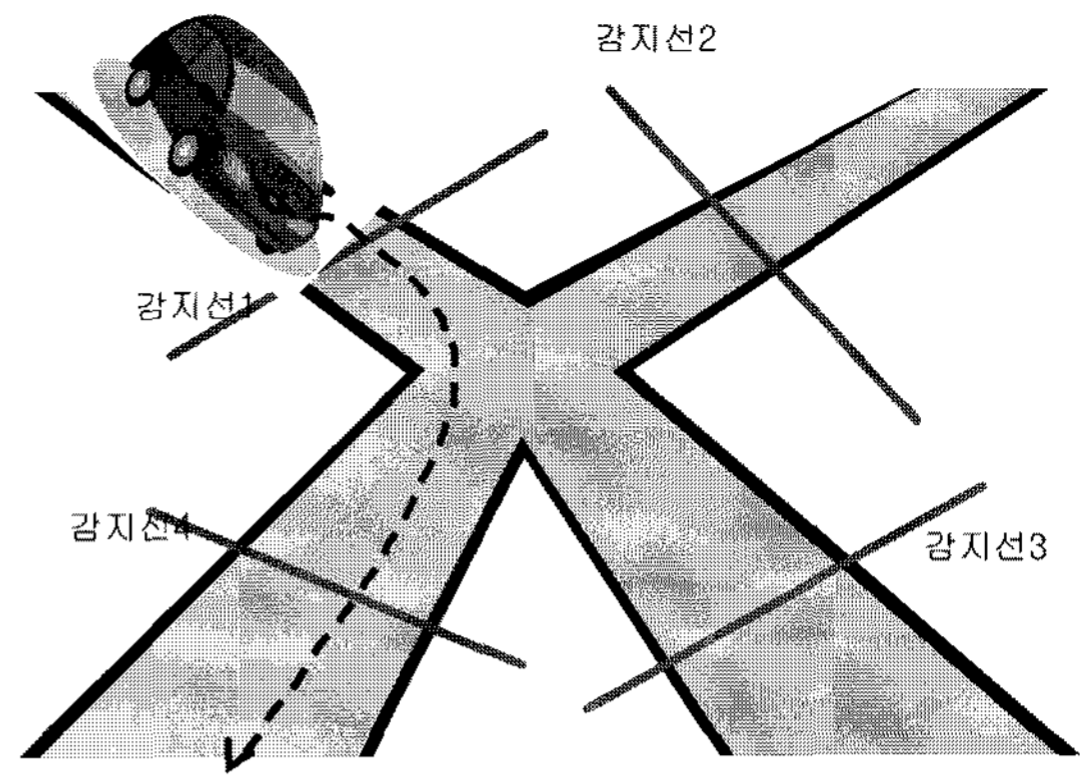
블로브가 완전히 삭제되기 전까지는 이전에 계산되었던 이동 속도로 계속 움직이는 것으로 간주한다. 이것은 차량이 일시적으로 나무 등에서 의하여 가려졌다가 다시 등장하는 경우에 새로운 차량으로 인식하지 않고 이전의 차량과 매치시키기 위한 것이다. 이 기능은 또한 정지하고 있는 차량의 경우, 일정한 횟수의 프레임이 경과되면 자동적으로 배경으로 인식하게 한다.

## V. 차량 카운팅 모듈

교차로에서 교통량 측정을 위하여 사용자는 화면에 측정을 원하는 위치에 감지선을 표시한다. 하나의 감지선을 통과한 자동차가 다른 감지선을 통과하면 해당되는 카운팅 배열의 요소가 증가되게 된다.

예를 들어서 <그림 4>에서 차량은 감지선 1을 통과하여 교차로에 진입한 후에 감지선 4를 거쳐서 교차로에서 진출하였다. 따라서 진입 감지선 1과 진출 감지선 4에 해당되는 차량 카운팅 배열의 요소가 1만큼 증가하게 된다. 이런 방법으로 교차로에서 직진이나 좌회전, 우회전, 유턴 등의 차량을 모두 카운팅할 수 있다.

카운팅을 보다 신뢰성있게 하기 위하여 차량의 범위 안의 픽셀을 모두 조사하여 각각의 감지선에 해당하는 픽셀들의 숫자를 계산하여서 가장 많은 픽셀을 가지는 감지선을 선택한다. 또한 감지선 픽셀의 개수가 일정한 임계값보다 커야만 감지선을



차량 카운팅 배열

	진출 감지선1	진출 감지선2	진출 감지선3	진출 감지선4
진입감지선1	0	0	0	1
진입감지선2	0	0	0	0
진입감지선3	0	0	0	0
진입감지선4	0	0	0	0

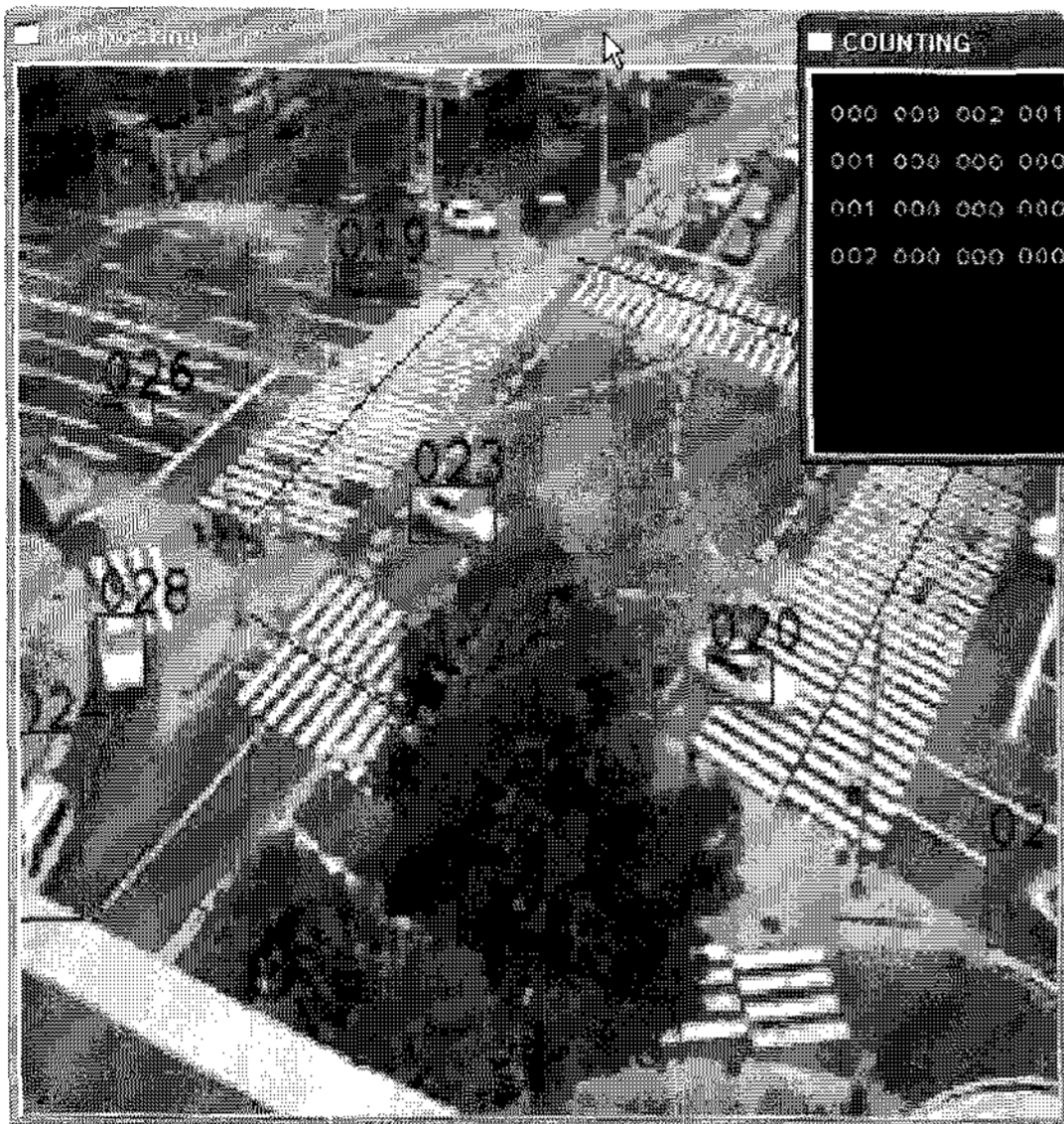
<그림 4> 차량 카운팅 알고리즘  
<Fig. 4> Vehicle counting algorithm

통과한 것으로 인식하였다.

차량의 상태는 “교차로 외부”, “감지선 통과중”, “교차로 내부” 중의 하나가 된다. 차량의 상태가 “교차로 외부”로 되어 있는 차량이 진입 감지선을 통과하는 중이면 차량의 상태가 “감지선 통과중”로 변경된다. 차량이 계속하여 진입 감지선을 벗어나면, 차량의 상태가 “교차로 내부”로 기록된다. 현재 “교차로 내부”에 있는 차량이 다시 진출 감지선을 통과하면 차량의 상태가 “교차로 외부”가 되고 이때 차량 카운팅 배열의 해당되는 요소가 1만큼 증가하게 된다.

## VI. 실험 결과

실험 영상은 실제 교차로를 촬영한 약 40초간의 320×240 크기의 24비트/픽셀의 컬러 영상이다. 실험에 사용된 PC는 2.66GHz의 Intel Core2 Duo CPU를 가지고 있고, 운영 체제로 Windows XP 환경을 사용하고 있다. 알고리즘의 성능을 비교하는 기준으로는 인간이 판단한 이상적인 결과에 대하여 감



<그림 5> 교통량 측정 시스템의 전체 모습  
<Fig. 5> Screenshot of vehicle counting system

<표 1> 세 가지 방법의 실험 결과  
<Table 1> Experimental results of three methods

사용된 방법	총 차량 수	감지된 차량	감지율 (%)	잘못 측정된 차량 수	오차율 (%)	측정 시간 (초)
①연결 성분	31	23	74	8	48	142
②연결 성분 + mean shift	31	20	64	0	0	290
③연결 성분 + mean shift + 휴리스틱	31	24	77	0	0	314

지율과 오차율을 계산하였다.

<표 1>에서 방법 ①은 연결 성분만을 가지고 차량을 추적하는 것이고 방법 ②는 연결 성분을 사용하되 차량끼리 충돌이 발생했을 경우에는 mean-shift를 사용한다. 방법 ③은 연결 성분과 mean-shift를 사용하면서 추가로 휴리스틱을 사용하는 것이다. 적용한 휴리스틱의 예는 다음과 같다. 만약 교차로 안에서 진입 정보가 없는 차량이 갑자기 감지

되었을 경우, 현재 속도를 기반으로 진입 감지선을 추측한다. 이는 차량 두 대가 붙어서 진행하는 경우, 진입시에는 하나의 차량으로 인식되다가, 교차로 내부에서 분리되면 차량이 갑자기 하나 더 나타난 것처럼 인식되기 때문이다.

<표 1>에서 감지율은 교차로를 지나가는 전체 차량 대수 중에서 알고리즘이 감지한 차량 대수의 비율로 정의하였다. 또 오차율은 사람이 추적하는 경우에 비교하여서 감지된 차량 대수 중에서 잘못 감지된 차량의 대수로 정의하였다.

<표 1>의 결과에서 보면 연결 성분 정보만을 사용하는 것은 감지율은 70%정도지만 48% 정도의 상당히 높은 오차율을 보인다. 오차율의 원인을 분석해본 결과, 연결 성분만 가지고 추적을 하다 보니, 교차로에서 빠져나가는 차량이 교차로에 진입하는 차량과 근접하여서 지나치는 경우, 진입하는 차량의 영향으로 추적이 실패하게 되고, 기존의 추적 박스가 반대 방향으로 이동하면서 차량이 다시 교차로에 진입한 것으로 잘못 카운팅된다. 이 경우에는 감지하는 차량의 대수는 증가하지만 동시에 오차율도 높아지게 된다. 결국은 차량들이 근접하여서 지나갈 때 연결 성분끼리 잘못된 매칭을 하는 경우가 많음을 알 수 있다.

Mean shift을 사용하는 경우에는 충돌을 mean shift로 해결하기 때문에 이런 현상이 없어지게 된다. 연결 성분에 mean shift, 휴리스틱을 사용하는 방법은 약 77%의 정확한 감지율을 보여준다. 사용한 휴리스틱으로는 앞에서 설명한 바와 같이 감지선 안에서 차량이 갑자기 나타났을 경우, 현재 속도를 기반으로 진입 감지선을 추측하는 방식이다. 휴리스틱을 사용하지 않으면 이러한 차량을 전혀 카운팅되지 않는다.

일반적으로 진행 방향이 한 방향인 연속류 도로에서의 감지율인 90%대보다는 못치는 감지율로서 주된 원인은 시작 단계부터 근접하여서 한덩어리로 움직이는 차량을 제대로 분리하지 못하기 때문이다. 이에 대한 해결책으로는 동영상을 역방향으로 재생하면서 차량과 차량을 분리하는 방법을 생각할 수 있다. 또 다른 방법으로는 교차로의 각 차선마다



가상적인 감지선을 별도로 설치하여서 차선별로 별도로 차량 대수를 카운팅하는 방법도 생각해 볼 수 있을 것이다.

## VII. 결 론

본 논문에서는 교차로 통과하는 차량의 대수를 비디오 카메라를 이용하여 자동적으로 측정하는 시스템에 대하여 연구하였다. 동영상에서 배경과 전경을 분리하기 위하여 Li의 방법을 적용하였으며 전경 영상에서 연결 성분을 추출하여 차량을 인식하였다. 인식된 차량은 연결 성분의 크기, 위치, 평균 밝기 등의 속성을 이용하여 추적되었으며 만약 두 개의 연결 성분이 충돌을 일으켜서 분리가 어려운 경우에는 mean shift 기법을 이용하여 추적하였다.

직진, 좌회전, 우회전을 분리하기 위하여 동영상 위에 사용자가 가상의 진입 감지선과 진출 감지선을 그려서 이 가상의 선을 통과하는 차량의 수를 카운팅하였다.

감지율을 더욱 향상하기 위해서는 서로 근접하여 진행하는 차량을 분리하여 카운팅할 수 있어야 할 것이며 카메라의 흔들림이나 잡음에 더욱 강인하고 실행 속도 면에서 더욱 효율적인 알고리즘을 필요로 한다. 또한 차량의 그림자나 건물의 그림자에 의한 오차를 줄이는 방안도 고려하여야 한다.

## 참고문헌

[1] 이대호, 박영태, "시공간 영상 분석에 의한 강건한 교통 모니터링 시스템," *정보과학회 논문지: 소프트웨어 및 응용*, 제31권, 제11호, pp. 1534-1542, 2004. 11.

[2] 하동문, 이종민, 김용득, "비전 기반 차량 검출 및 교통 파라미터 추출," *정보과학회 논문지, 시스템 및 이론*, 제30권, 제11호, pp. 610-620, 2003. 12.

[3] O. Masoud, N. P. Papanikolopoulos, and E.

Kwon, "The use of computer vision in monitoring weaving sections," *IEEE Trans. Intelligent Transportation Systems*, vol. 2, no. 1, pp. 18-25, Jan. 2001.

[4] S. Gupte, O. Masoud, R. F. K. Martin, N. P. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Trans. Intelligent Transportation Systems*, vol. 3, no. 1, pp. 37-47, Jan. 2002.

[5] P. G. Michalopoulos, "Vehicle detection video through image processing: The autoscope system," *IEEE Trans. Vehicular Technology*, vol. 40, no.1, pp. 21-29, Jan. 1991.

[6] L. Wixson, K. Hanna, and D. Mishra, "Illumination assessment for vision-based traffic monitoring," *Proc. IEEE Workshop on Visual Surveillance*, pp. 34-41, 1998.

[7] A. Yilmaz, O. Javed, and M. Shar, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 1-45, April 2006.

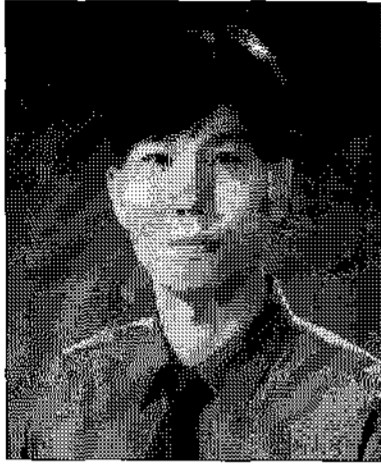
[8] T. Chen, H. Haussecker, A. Bovyrin, R. Belenov, K. Rodyushkin, A. Kuranov, and V. Eruhimov, "Computer vision workload analysis: Case study of video surveillance systems," *Intel Technol. J.*, vol. 9, pp. 109-118, Sept. 2005.

[9] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, "Foreground object detection from video containing complex background," *ACM Multimedia*, 2003.

[10] R. Hammond and R. Mohr, "Mixture densities for video objects recognition," *Proc. IEEE Int. Conf. Pattern Recognition*, pp. 71-75, 2000.

[11] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *Proc. IEEE Int. Conf. Pattern Recognition*, vol. 2, pp. 142-149, Feb. 2000.

저자소개



천 인 국 (Chun, In-Gook)

1983년 2월 : 서울대학교 공학학사 (전자전공)

1985년 2월 : KAIST 공학석사(전자전공)

1985년 3월 ~ 1988년 2월 : 삼성전자 종합연구소 주임연구원

1993년 8월 : KAIST 공학박사(전자전공)

1993년 9월 ~ 현재 : 순천향대학교 컴퓨터학부 교수

2005년 3월 ~ 2006년 2월 : 캐나다 UBC 대학교 방문교수