

# 칼만 필터와 차량 특징 정보를 이용한 중첩된 다중 차량 추적에 관한 연구

준회원 강은구\*, 정회원 김성동\*\*, 최기호\*

## A study on tracking of multiple vehicle occlusions in road images using Kalman filter and vehicle feature information

Eun-ku Kang\* Associate Member Seong-Dong Kim\*\*, Ki-ho Choi\* Regular Members

### 요 약

본 논문은 고정된 카메라를 통해 들어오는 도로영상에서 추적되는 다중 차량들의 겹침(occlusion) 발생시 칼만 필터와 차량의 특징정보를 이용하여 개별 차량을 분할하고 추적 가능한 시스템을 제안하고 구현하였다.

다중 차량을 추적할 시 가장 큰 문제점이 되고 있는 차량 겹침을 해결하기 위해 카메라와의 거리를 이용하여 해결하는 방법과 3D 모델을 이용하여 해결하는 방법, 겹침 추론 등 차량 겹침을 해결하기 위한 여러 가지 방법들이 제시되고 있다. 그러나 영상에 연속적으로 나타나는 다중 차량의 겹침을 단일 차량으로 인식할 수 있는 단점이 있다. 따라서 칼만 필터와 차량의 특징 정보로서 차량의 높이와 넓이의 비, 추적에 사용되는 박스에서 차량과 여백의 비를 이용함으로써 연속적으로 나타날 수 있는 차량 겹침을 분할하고 추적 가능하게 하는 시스템을 구현하고 실험하였다.

본 시스템에서는 256 X 256 의 크기로 15 frames/sec로 저장된 AVI 파일 형식의 동영상을 사용하여 실험에 이용하였으며, 시내 도로에서의 차량들의 실험 결과 기존의 방법 보다 차량 특징 정보를 이용한 방법이 연속적 겹침에 대한 처리에 우수함을 보였다.

### ABSTRACT

This paper is about a proposal and implementation of a system that can segment and track vehicles using the Kalman filter and the information of vehicle features when there is occlusion of vehicles in road image captured by a fixed camera. When tracking vehicles, the most prevalent problem is vehicle occlusion. While many methods suggested for vehicle occlusion have used the distance between the camera and vehicle along with 3D models, none have completely eliminated occlusion as the first image appearance is processed to the second image. The resulting disappearance can not solved.

In this paper, a solution is shown for continuous vehicle occlusion in a fixed camera schema using the information of vehicle features using the ratio of width and height of tracking box and the distribution of a vehicle in the tracking box. In this system, AVI files, which are sized by 256 x 256 and captured 15 frames/sec, were used for the experiment. The experimental result showed the method of the information of vehicle features is better than any other method.

\* 광운대학교 컴퓨터공학과

\*\* 계원조형예술대학 정보통신과 교수(sdskim@ns.kaywon.ac.kr)

논문번호: K01064-0210, 접수일자: 2001년 2월 10일

## I. 서론

차량 증가에 따른 도로 확충의 부족으로 교통 문제가 증가함에 따라 교통 감시는 현 사회에 중요한 주제가 되고 있다. 따라서 건설교통부에서는 이에 대한 방안으로 지능형 교통 시스템(ITS: Intelligent Transport System)을 구축 중에 있다. 지능형 교통 시스템이란 첨단 정보 통신 기술을 활용한 교통 관리 시스템으로서 우리나라에는 고속도로와 시내도로에 일부 구축되어 있고 현재 일반 국도에 구축되고 있다. ITS에 쓰이는 기술에는 LVDS (Loop Vehicle Detection System), IVDS(Image Vehicle Detection System)이 이용되고 있다. LVDS는 도로에 원형이나 사각형의 루프 코일을 설치하여 이 루프 코일 위를 차량이 통과할 경우 코일에서 발생하는 자장의 세기를 감지하여 교통상황의 정보를 제공하는 시스템이다. 그러나 이 방식은 루프를 도로 표면에 설치하게 되므로, 도로의 노화나 다른 도로상의 공사로 인해 노면에 파손이 발생한 경우에 작동이 되지 않는 경우가 발생하는데, 이를 수리하기 위해서는 교통의 흐름을 통제하여야 하므로 교통의 혼잡이 발생하며, 이를 막기 위해서는 야간에 작업을 진행하게 되면 비용상의 경제적인 문제도 발생한다. 따라서 유지, 보수가 상당히 어렵다<sup>[1],[2]</sup>. 따라서 설치 유지 보수가 용이한 IVDS에 관심이 높아지고 있다. IVDS는 영상 화면의 변화를 감지하여 필요한 교통 정보를 얻는 시스템으로써 픽셀들의 변화량과 변화정도를 분석하여 속도나 차종 등 필요한 정보를 추출할 수 있다. 이에 따라 카메라에서 얻어진 동영상을 통한 교통 감시 시스템에 대한 관심이 높아지면서 컴퓨터 비전 연구가 활발히 이루어지고 있는 추세이다. 비전 시스템을 통해서 얻어진 도로 상황의 혼잡 정도, 사고의 발생, 차선의 위반 등의 정보를 중앙의 통제 시스템에 보내어서 적절한 대응을 할 수 있다면 효율적인 차량 통제가 이루어질 수 있을 것이다.

그러나 카메라의 영상을 이용하여 교통 정보를 얻는데 있어서 차량들의 겹침(occlusion) 현상이 영상 화면의 픽셀들을 분석하는데 문제점이 되고 있다. 이에 대한 연구로 California Berkeley대학에서는 영상 화면에서 차량과 카메라의 거리를 이용하여 해결 하고 있으며<sup>[3]</sup>, 홍콩 대학에서는 3D 모델과 지역 비(area ratio)을 사용하여 차량의 겹침 현상 해결에 관하여 연구 되고 있다<sup>[4]</sup>. 그리고 광주

과학 기술원에서는 Explicit/Implicit occlusion reasoning 을 제시하여 차량 겹침 현상을 해결하고 있다<sup>[5]</sup>. 그러나 홍콩 대학 연구는 차량의 3D 모델을 미리 알아야 하는 문제점이 있고, 특히 위 세 연구의 공통된 문제점으로는 두 차량이 처음 영상에 나타날 때부터 사라질 때까지 계속 겹침 현상을 보였을 경우에는 이에 대한 해결이 어렵다는 문제점이 생기가 된다.

따라서 본 논문에서는 물체 화소의 원 분포와 윤곽선까지 거리의 최대 최소 비를 차량의 특징 정보로 사용하여 occlusion reasoning의 단점을 보완하고 다중 차량 추적에서 나타나는 차량 겹침 현상을 해결하기로 한다. 특히 화면 처음부터 마지막까지 계속해서 나타나는 겹침 현상을 해결하기 위해 추출된 차량의 특징 정보를 사용하기로 한다.

## II. 다중 차량 추적 시스템

본 논문의 다중 차량 추적 시스템의 구성은 그림 2-1과 같다. 먼저, 배경 영상과 연속되어 들어오는 영상으로부터 차량을 추출하여 차량의 위치정보를 얻는 차량 분할(Vehicle Segmentation), 차량 분할에서 얻어진 정보를 가지고 차량을 추적하는 차량 추적(Vehicle Tracking), 마지막으로 앞에서 얻어진 정보를 분석하여 교통 파라미터들을 구하는 교통 파라미터 추출(traffic parameter extraction)부분으로 나눌 수 있다.

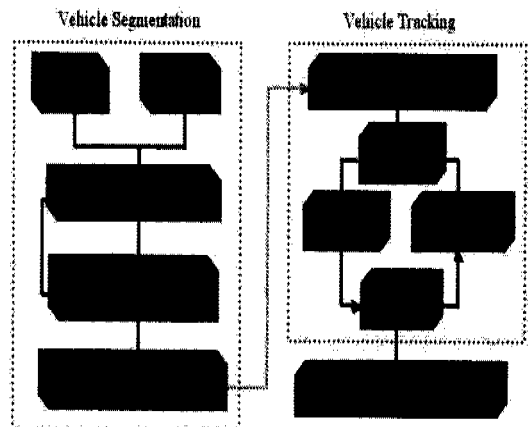


그림 2-1. 다중 차량 추적 시스템 전체 구성도

### 2.1 동작 분할

분할(segmentation)은 영상분석에 있어서 가장 기본이 되는 기법으로 영상을 겹쳐지지 않는 동질 영

역들의 집합으로 나누는 기법이다. 컴퓨터 비전에 있어서 다양한 분야에 적용되는 영상 분석의 기본 기법으로 영상처리에 있어서 중요한 위치를 차지하고 있다. 최근 영상 부호화에도 응용범위가 확대됨에 따라 그 중요성은 더욱 증대되고 있다. 영상분할(video sequence image segmentation)에서는 원하는 객체를 추출하기 위해서 공간영역 정보를 이용하여 균질 영역(homogenous region)을 추출하게 된다. 영상분할에서의 목적은 여러 가지가 있을 수 있지만 주로 객체와 배경을 구분하는 것에 의미를 부여하고 있다. 따라서 배경과 객체간의 영역을 정확하게 구분하는 것이 필요하게 된다.

연속 영상에서 고정된 배경으로부터 움직이는 물체만을 뽑아내는 가장 손쉬운 방법은 현재의 입력 영상과 고정된 배경사이의 차 영상을 이용하는 방법이다. 이 방법을 통하여 차량의 위치정보를 얻을 수 있다. 배경모델은 시스템 초기화 단계에서 수행되는 배경 영상 획득과정과 추적단계에서 수행되는 적응 배경 모델로 이루어진다. 배경 영상 획득 과정은 시스템이 가동되는 초기에만 수행된다. 배경 영상은 날씨 등과 같은 주위 환경 변화에 따라 배경의 새로운 반영이 필요하다. 날씨, 밤낮 등의 주위 환경 변화가 존재하는 연속 영상으로부터 움직이는 물체만을 찾도록 하기 위해서는 주위 환경 변화에 따라 적응 가능한 배경 모델을 구현하여야 한다.

임  $f_i(x,y)$ 사이의 변화를 검출하기 위한 가장 쉬운 방법 중 하나는, 두 영상의 모든 화소와 화소를 비교하는 것이다. 이를 수행하기 위한 한 과정은 차이 영상을 만드는 것이다. 움직이지 않는 정적인 성분만을 가지고 있는 배경 영상을 같은 환경에서 움직이는 물체를 포함한 다음 영상과 비교하게 되면, 두 영상의 차이에서는 정적 성분은 제거되고 비정적 영상 성분에 해당하는 영이 아닌 요소만이 남아 있는 결과를 갖게 된다.

한 두 영상간의 차이 영상은 다음 식 (2.1)과 같이 정의될 수 있다.

$$M_{ij}(x,y) = \begin{cases} \text{graylevel}_{ij} & \text{if } |f_b(x,y_i) - f_i(x,y)| > \theta \\ 0 & \text{그 이외의 경우} \end{cases} \quad (2.1)$$

여기서  $\theta$  는 문턱치(threshold)이다.  $d_{ij}(x,y)$ 는 공간좌표  $(x,y)$ 에서 두 영상간의 명암도 차이가 그 좌표에서 꽤 다를 경우에만 문턱치  $\theta$  에 의해 결정되어져 각 픽셀의 gray level값을 유지하게 된다.

동영상 해석에서는  $M_{ij}(x,y)$ 에서 gray level값을 가진 모든 화소들은 물체의 움직임 결과로 고려된다. 이 방법은, 두 영상이 같이 조절되어 있고 조명은  $\theta$  에 의해 설정된 범위 안에서 비교적 일정한 경우에만 적용 가능하다. 실제로는,  $M_{ij}(x,y)$ 에서 gray level값을 가진 요소들은 종종 잡음의 결과로 발생한다. 일반적으로 이러한 요소들은 차이 영상에서 분리 점들로 되고, 이들의 제거를 위한 간단한 방법은  $M_{ij}(x,y)$ 안에서 gray level로 된 4-연결 또는 8-연결 영역을 형성하여 미리 정해진 요소의 수 보다 작은 수를 가진 영역들은 모두 무시하는 것이다. 비록 이 방법은 작거나 천천히 움직이는 물체를 무시하는 결과를 가져 올 수도 있으나, 이 방법은 차이 영상에서 남아 있는 요소들이 실제로 움직임을 결과로 될 가능성을 높인다.

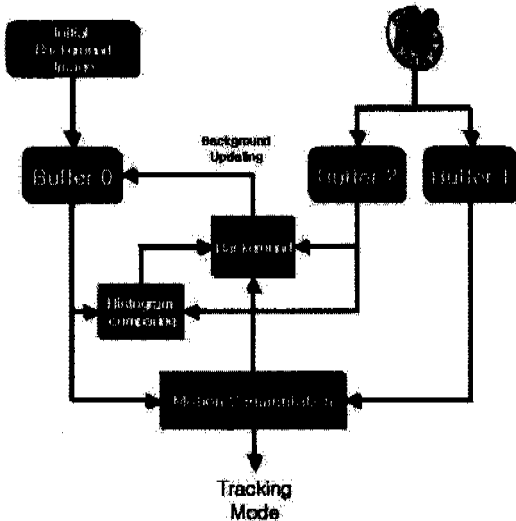


그림 2-2. 다중 차량 분할 시스템 구조

2.2 차량 분할

배경 프레임  $f_b(x,y)$ 와 연속 되어 들어오는 프레

2.3 배경 이미지 처리

2.3.1 히스토그램 개요

히스토그램은 화소 밝기(pixel intensity)의 막대 그래프(bar graph)이다. x축은 화소의 밝기를 나타내고, y축은 각 화소 밝기가 영상 내에서 나타나는 빈도수를 나타낸다. 영상 히스토그램은 영상의 밝기 분포를 나타내는 유용한 정보이다. 또한 영상의 명암대비(contrast) 정보도 내포하고 있다

이미지의 밝기 특성은 밝기 히스토그램으로 알려

진 도구를 가지고 나타낼 수 있다. 일반적으로 히스토그램은 수 집합의 분포 그래프이다. 밝기 히스토그램은 디지털 이미지 내 화소의 gray level 의 분포 그래프이다. 히스토그램은 0에서 255(8bit gray scale)의 수평축은 밝기를, 수직 축은 화소 수를 가지는 그래프로 나타낸다. 이 그래프를 사용하여 우리는 즉시 이미지가 기본적으로 어두운지 밝은지와 높거나 낮은 화질(contrast)을 볼 수 있다.

화질은 이미지 밝기 속성을 묘사하는데 종종 사용되는 말이다. 이미지의 화질은 밝기 히스토그램에 분명히 나타내어진다. 낮은 화질은 gray scale의 화소 밝기의 그 성분이 밀집되게 나타나고 다른 gray level은 최소화로 남겨나 완전히 채워지지 않게 된다. 높은 화질은 2개의 peak가 바깥쪽에 존재하는 bi-modal 히스토그램으로서 나타낸다. 히스토그램은 또 한 우리에게 얼마나 많은 유용한 활동 범위가 이미지에 의해서 사용되어지는가를 알려준다. 이미지의 actual dynamic range는 gray scale안에 얼마나 많은 gray level이 채워져 있는지에 의해 나타내어진다



그림 2-4. 히스토그램 구조

2.4 히스토그램 적용

모든 프레임이 동일한 수의 픽셀을 가지고 있다고 했을 때, 하나의 프레임은 n개의 구별되는 명암 값들 즉 gray level 값들의 집합으로 특징지어질 수 있으며, 명암 히스토그램  $H_i$ 는 명암 버킷(bucket)  $h_i$ 가 화상 내 gray level  $i$ 의 픽셀 수를 포함하는 벡터  $(h_1, h_2, \dots, h_n)$ 들의 합이 된다. 따라서, 명암 히스토그램은 다음 식 (2.2)과 같이 나타낼 수 있다.

$$H_i = \sum_{i=0}^{255} h_i \tag{2.2}$$

여기서  $H_i$ 는 화상 안에서 명암  $i$ 의 전체 명암 히스토그램을 나타낸다,  $i$ 의 범위는 8bit gray level RAW image에서 0~255까지 나타낸다.

명암 히스토그램의 차이를 구하는 식은 일반적으로 다음 식 (2.3)와 같이 많이 사용된다.

$$d(I_i, I_{i+1}) = \sum_{j=0}^{255} |H_i(j) - H_{i+1}(j)| \tag{2.3}$$

여기서  $I$ 는 동영상의 프레임을 나타내며,  $d(I_i, I_{i+1})$ 는 연속된 프레임간의 히스토그램의 차이를 나타낸다. 히스토그램 차이 값에서 차량을 추출 할 수 있는 경우와 없는 경우의 경계선을 임계값으로 정한다. 이 임계값을 넘었을 경우 적응 배경 모델에서 버퍼 2에 있는 이미지를 버퍼 0으로 업그레이드 한다.

2.4.1 차량 추적

움직이는 물체를 추적하기 위하여서는 효과적인 칼만 필터를 사용하게 된다. 본 연구에서도 측정된 화상 정보로부터 표적물을 추적하기 위하여 칼만 필터를 사용하였다. 칼만 필터는 필터의 구조가 선형으로 구성되어 있으며, 선형시스템에서 모든 변수들의 분포가 가우시안이고 부가되는 잡음들이 백색 가우시안의 특성을 지니고 있을 경우에 그 특성이 최적으로 나타남이 잘 알려져 있다. 그리고 차량의 겹침 (occlusion)을 해결하기 위하여 차량의 특징들을 이용하여 차량을 추적하였다.

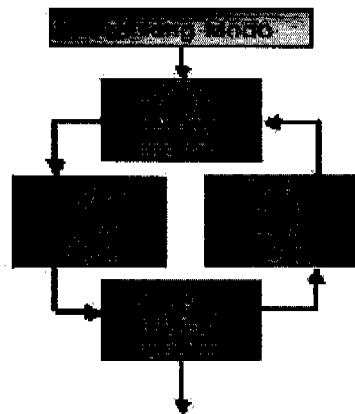


그림 2-5. 다중 차량 추적 시스템 구조

2.5 추적 필터

2.5.1 칼만 필터 적용

칼만 필터란 잡음에 의해 간섭 받는 선형 동적 시스템(linear dynamic system)에서 상태 벡터  $x_k$ 의 최적의 추정치  $\hat{x}_k$ 를 구하기 위한 순환적인

(recursive) 알고리즘이다. 여기서  $k$ 는 시간을 나타낸다. 칼만 필터링은 크게 예측(prediction), 측정(measurement), 수정(update)의 세 가지 단계로 나누어진다.

예측 단계에서는 다음 상태의 상태 벡터  $\hat{x}_{k+1(-)}$  과 상태 추정 오차의 공분산 행렬  $P_{k+1(-)}$  을 예측하는 단계이다. (-)은 아직 측정을 통한 수정 단계가 이루어지지 않았음을 나타낸다. 상태 추정 오차의 공분산 행렬을 수식으로 표현하면 다음 식 (2.4)과 같다.

$$P_{k+1(-)} = E \left[ \begin{pmatrix} x_k - \hat{x}_k \\ x_k - \hat{x}_k \end{pmatrix} \begin{pmatrix} x_k - \hat{x}_k \\ x_k - \hat{x}_k \end{pmatrix}^T \right] \quad (2.4)$$

예측 단계에서 이루어지는 상태 벡터와 상태 추정오차의 공분산 행렬의 예측을 나타내는 식은 각각 식 (2.5)와 식 (2.6)으로 주어진다.

$$\hat{x}_{k+1(-)} = f(\hat{x}_{k(+)} + w_k \quad (2.5)$$

$$P_{k+1(-)} = P_{k(+)} + Q_k \quad (2.6)$$

여기서  $f$ 는 주어진  $\hat{x}_{k(+)}$ 의 예측 함수이고,  $w_k$ 는 시스템의 모델 잡음(model noise),  $Q_k$ 는  $w_k$ 의 공분산 행렬이다. 측정 단계는 다음 식 (2.7)으로 표현된다.

$$z_k = H \hat{x}_k + v_k \quad (2.7)$$

$z_k$ 는 측정 벡터,  $H$ 는 상태 벡터와 측정 벡터사이의 관계를 나타내는 관측 행렬(observation matrix),  $v_k$ 는 측정 오차를 나타내며, 모델 잡음  $w_k$ 와 측정 오차  $v_k$ 는 비상관(uncorrelated) 되었다고 가정한다.

마지막으로 수정 단계에서는 측정 단계에서 구해진 측정값을 바탕으로 상태 벡터와 상태 추정 오차를 새로운 값으로 수정한다. 수식으로 표현하면 다음 식 (2.8), 식 (2.9), 식 (2.10)과 같다.

$$K_k = P_{k(-)} H_k^T [H_k P_{k(-)} H_k^T + R_k]^{-1} \quad (2.8)$$

$$P_{k(+)} = [I - K_k H_k] P_{k(-)} \quad (2.9)$$

$$\hat{x}_{k(+)} = \hat{x}_{k(-)} + K_k [z_k - (H_k \hat{x}_{k(-)})] \quad (2.10)$$

### 2.5.2 칼만추적 필터의 적용

이 장의 목적은 차량에 맞는 위치의 동작 파라미터들을 정하는 것이다. 측정된 것들은 동작 분할 모듈(motion segmentation module)에서 각 이미지마다 더욱 정확히 계산된 motion estimate들이다. 추적되는 상자의 좌상과 우하의 좌표, 순환적 측정 기인 칼만 필터를 사용할 것이다.

먼저 본 논문의 운동 모델에서 사용된 가정은 다음과 같다.

1. 관성 운동을 하는 물체는 일반적으로 완만하게 움직이므로 운동 파라미터의 급격한 변화가 생기지 않는다.
2. 각각의 추적 목표의 속도는 인접한 시간 축상에서는 상수 값을 가지는 등속 운동을 한다. 차량의 운동학 모델은 다음 식 (2.11), 식 (2.12), 식 (2.13)으로 표현된다.

$$X_{k+1} = 1/2 a_k \Delta t^2 + v_k \Delta t + X_k \quad (2.11)$$

$$v_{k+1} = a_k \Delta t + v_k \quad (2.12)$$

$$a_{k+1} = a_k = a \quad (2.13)$$

그리고 차량의 위치를 나타내는 좌표(X,Y)를 이용하는 데 있어서 다음 식과 같이 두 가지 식 (2.14)과 식 (2.15)로 나눈다.

$$X_k = \begin{bmatrix} x_k \\ v_{xk} \\ a_{xk} \end{bmatrix} \quad (2.14) \quad Y_k = \begin{bmatrix} y_k \\ v_{yk} \\ a_{yk} \end{bmatrix} \quad (2.15)$$

본 논문의 대상이 되고 있는 지상의 차량들에 관한 시스템에서도 그 위치와, 속도, 그리고 가속도를 상태변수로 선택하여 이산식으로 정리될 수 있다.

$$\begin{bmatrix} p_{k+1} \\ v_{k+1} \\ a_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \\ a_k \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w_k \quad (2.16)$$

$$z_k = [1 \quad 0 \quad 0] \begin{bmatrix} p_k \\ v_k \\ a_k \end{bmatrix} + v_k \quad (2.17)$$

여기에서  $p_k$ 는 표적물의 위치를,  $v_k$ 는 표적물의 속도를,  $a_k$ 는 표적물의 가속도를, 그리고  $\Delta t$ 는 표적물의 위치에 관한 측정이 진행되는 시간을 나타내고 있다. 또한 측정방정식에 주어지는  $v_k$ 는 측정 잡음으로 영평균 백색잡음을 나타내고, 가속도를 나타내는 식에서의  $w_k$ 는 백색 가우시안 프로세스로서 차량에 가해지는 여러 가지 힘의 결과를 얻어지는 차량의 가속도를 의미하며,  $v_k$ 와 마찬가지로 영평균 백색잡음으로 고려되었다. 추적시스템에서는 차량의 가속도를 알 수 없으므로 이를 시스템잡음으로 처리하게 되며, 시스템의 특성에 따라 적절히 모델링할 수 있다. 칼만 필터에 적용되는 값은 차량을 추적하는데 사용되는 박스의 좌상의 좌표와 우하의 좌표를 이용한다. 칼만 필터에 적용하는 식들은 같기 때문에 여기서 우리는 X좌표에 관해서만 기술하기로 한다. 계산 과정은 상태방정식과 측정 방정식에 의해 제어된다.

다음 식은 상태 방정식에 관한 식이다.

$$X_{k+1} = A_k X_k + w_k \quad (2.18)$$

측정 방정식은 다음과 같다.

$$z_k = H_k X_k + v_k \quad (2.19)$$

$w_k$ 는 시스템의 모델 잡음이며,  $v_k$ 는 측정 잡음을 나타내며 서로 비상관이며 가우시안 분포를 가졌다 고 가정한다.

$$P(w) \rightarrow N(0, Q) \quad (2.20)$$

$$P(v) \rightarrow N(0, R) \quad (2.21)$$

그리고 칼만 필터는 다음과 같은 서로 다른 측정 벡터를 사용한다.

$$z_{xk} = \begin{bmatrix} x_k \\ v_{xk} \end{bmatrix} \quad (2.22) \quad z_{yk} = \begin{bmatrix} y_k \\ v_{yk} \end{bmatrix} \quad (2.23)$$

A행렬은 현 상태와 다음 상태와의 관계를 나타내고 H행렬은 현 상태와 측정간의 관계를 나타내는 주어진 위의 식에 따라서 다음과 같이 정의 할 수 있다.

$$A_k = \begin{bmatrix} 1 & 1 & 0.5 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.24)$$

$$H_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (2.25)$$

### III. 차량 특징 정보를 이용한 방법

#### 3.1 차량 특징 정보의 개요

다른 시스템의 경우 화면에서 차량들의 겹침이 생기거나 풀릴 경우에만 적용되었을 경우의 처리를 보여주고 있다. 그러나 화면 내에서 계속 차량 겹침 현상이 일어날 경우에는 이를 인지하지 못하는 경우가 있다. 따라서 이에 대한 방안으로 차량을 추적하는데 사용되는 박스를 이용하여 이를 해결하고자 한다.

차량 특징 정보는 카메라에서 들어오는 영상을 이용하여 얻는 정보로써 화면에서 차량이 차지하는 부분의 픽셀 값을 이용하는 것이다. 차량 분할에서 얻어진 차량의 정보를 가지고 각각의 특징들을 살펴봄으로써 얻을 수 있다.

사람이 아닌 컴퓨터 등 기계들을 통해 차량을 인식하고 추적하기 위해서는 화면 즉 프레임마다 그 내용들을 분석하고 분석을 통해 차량을 인지하고 각 차량별로 구분을 하여야 한다. 그러므로 각 차량마다 가지고 있는 특징들을 이용하여 차량들을 구분하고 한다. 이러한 특징들은 차량을 구분하여 인지하는 것 외에도 차량 겹침이 일어났을 때에도 사용될 수 있다. 차량의 겹침이라는 것은 차량 두 대가 겹쳐 보임으로써 기계가 인식할 때 두 대가 아닌 한 대로 인식할 수도 있다. 이런 오류를 해결하기 위해서 또한 차량의 특징들을 이용한다.

먼저 물체 화소의 원 분포를 이용하는 방법이다. 차의 윤곽선 중심과 최소거리에 있는 윤곽선 한 점의 거리를 반지름 r로 하는 원 내부의 화소 분포를 특징으로 택했기 때문에 물체의 회전, 크기 변화에 영향을 받지 않는다. 물체 화소의 원 분포는 식3-1과 같다

$$F^k = \frac{\sum_{(x,y) \in A} I(x,y)}{A} \quad (3.1)$$

A는 윤곽선 중심과 윤곽선 점들과의 최소 거리이다.

그리고 각 차량의 넓이와 높이를 이용한다. 각 차량의 대한 넓이를  $r_{wide}$ 라 하고,  $r_{height}$  높이를 라고 하면 다음과 같이 나타낼 수 있다.

$$R^k = \frac{r_{width}}{r_{height}} \quad (3.2)$$

차량을 추적할 때 각 차량마다 배경 이미지와 다른 물체 즉 차량들이 추출 될 경우 추출된 차량과 박스 픽셀 수의 비율을 비교한다.

추출된 차량의 픽셀의 수를  $p_{car}$  라 하고, 박스의 배경 픽셀 수를  $p_{box}$  이라 정의하면 다음과 같이 식을 나타낼 수 있다.

$$P^k = \frac{p_{car}}{p_{box}} \quad (3.3)$$

위의 특징들은 다음 그림 3-1에서 보듯이 추출된 차량에서 추적에 이용되는 추적 Box를 이용하여 각각의 특징들을 얻을 수 있다.

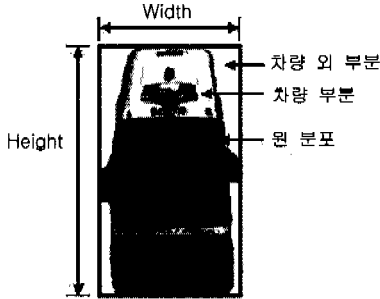


그림 3-1. 차량의 특징

먼저  $F^k$ 는 원안의 있는 픽셀들의 gray level값들을 더한 후 원의 지름을 나눈 값을 이용한다. 이 특징은 차량이 카메라에 잡히는 동안 점침 크기가 커지는데 이 특징은 차량의 크기에 상관없이 일정한 비율을 가지고 있어서 이를 특징으로 사용할 수 있다.  $R^k$ 은 추적에 사용되는 박스의 넓이와 높이의 비율로써 단일 차량의 경우와 겹침이 일어난 경우의 비율이 다르다는 이론 하에 사용되는 특징이다. 이 특징만으로는 단일 차량과 겹침이 일어난 차량들을 구분하기에 부족하기 때문에  $P^k$  값 즉, 추적에 사용되는 박스에서 차량부분과 차량 외 부분의 비율도 같이 사용하게 된다.

### 3.2 겹침 상태에서 차량 특징 적용

도로에서 차들이 주행을 하면서 카메라에 잡히는 경우는 세 가지의 경우가 있다. 첫째는 explicit occlusion으로써 처음에는 구분되어 나타난 후에 점차 겹침 상태로 측정되는 것이고 두 번째는 implicit occlusion으로 초기에 차량들이 단일 차량으로 나타난 후 분리되는 경우이다. 마지막으로 초기부터 마지막까지 계속 겹침 현상이 일어나는 경우이다. 다른 시스템의 경우 첫번째와 두 번째의 경우에 적용되나 마지막의 차량 겹침에 관하여서는 효율적이지 못했다. 따라서 위의 특징들을 적용하여 이를 해결 할 수 있다. 계속되어 나타나는 겹침의 경우

폭 : 84 높이 : 147 $R^k : P^k : 0.920.57$	폭 : 57 높이 : 105 $R^k : P^k : 0.980.54$	폭 : 63 높이 : 83 $R^k : P^k : 0.950.76$
폭 : 84 높이 : 169 $R^k : P^k : 0.890.50$	폭 : 58 높이 : 129 $R^k : P^k : 0.830.45$	폭 : 80 높이 : 130 $R^k : P^k : 0.630.62$

그림 3-2. 각 차량의 특징 정보인  $R^k$ 와  $P^k$ 의 예

다른 시스템에서는 단일 차량으로 인식하게 된다. 만일 다중 차량이 겹침 현상이 일어난 후 단일 차량으로 분리되면 분리되기 전의 상태가 겹침이라는 것을 인식 하게 되지만 카메라의 시야에 벗어 날 때 까지 계속 겹침 현상이 일어나게 된다면 단일 차량으로 인식하게 되는 오류가 생긴다. 본 논문에서 제시하는 방법은 단일 차량의 경우와 겹침이 일어난 경우를 차량의 특징을 이용하게 된다. 밑의 그림의 예를 보듯이 단일 차량의 경우에는 추적 박스에서 거의 대부분을 차량이 차지 하게 된다. 그러나 겹침의 경우에는 추적되는 박스에 빈 공간이 많이 나타나는 것을 볼 수 있고 높이와 넓이의 비율이 보통 단일 차량의 것과 다르게 나타난다. 이를 이용하면 처음부터 겹침이 일어나더라도 조기에 인지 할 수 있게 된다.

위의 그림은 여러 가지 경우의 차량들을 보여주고 있다. 상단의 그림은 겹침 현상이 없을 경우 추적되었을 때의 상태를 나타내며 하단의 그림은 차량들의 겹침 현상이 나타났을 때의 경우를 보여주고 있다.

위와 같이 여러 상황을 테스트 하여  $R^k$  와  $P^k$  를 이용하여 차량이 추출 되었을 때 차량 겹침인지의 판단을 할 수 있다.

- ◆ 단일 차량의 경우 :  $R^k > 0.51$   $P^k > 0.92$ 를 모두 만족할 경우
- ◆ 차량 겹침이 일어난 경우 :  $R^k < 0.51$  또는  $P^k < 0.92$ 일 경우

위에서 보듯이 임계값을 정하여 위에 만족하는 부분에 맞게 단일 차량과 차량 겹침이 일어난 경우를 판단 할 수 있다.

#### IV. 시스템 구현 및 실험

##### 4.1 실험 환경

Computer : Pentium II 333MHz PC  
 Video Camera : Digital 8mm camera  
 영상 보드 : Ziemax사의 Megapeg board, 15 frame/sec  
 실험 영상 : 시내 일반 도로 (창동역 근처, 안양 계원 예술대 앞)  
 256\*256 8bit gray level AVI 영상

실험 영상은 차량들의 속도가 평균 50 km/h가 되는 도로에서 촬영하였다. 카메라의 높이는 5M정도에서 촬영하였다. 구현된 멀티 차량 추적 프로그램은

다음 그림과 같이 구현 되었으며 각 부분에 대한 설명은 다음과 같다.

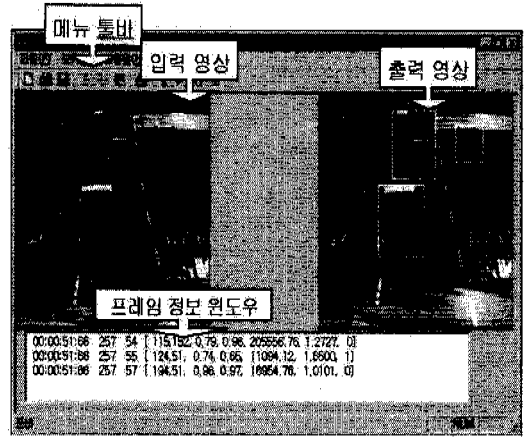


그림 4-1. 다중 차량 추적 시스템의 인터페이스

- 볼바  
 동영상을 불러오기, 이미지 파일 저장하기, 동영상 파일 실행하기, 실행 중 잠깐 멈춤, 실행 중단이 있으며 동영상을 실행하게 되면 차량 추적이 시작된다. 차량이 추적되면서 나오는 출력영상을 BMP 파일로 저장할 수 있도록 하였다.
- 입력 영상과 출력 영상  
 입력 영상은 현재 들어오는 차량의 영상들로서 이 영상을 처리 하여 차량의 정보를 추출하고 그 정보에 의해 차량을 추적하게 되면 차량 겹침 현상도 판단하게 된다. 출력 영상은 입력 영상에 의해 처리된 차량의 정보를 표시하는 부분으로 차량 추적 박스를 표시하여 추적이 되고 있음을 보여주었다.

##### ● 프레임 정보 윈도우

차량 추적을 하기 위한 프레임마다 처리에서 얻어지는 차량의 정보를 보여 주고 있다. 프레임별 시간, 프레임 번호, 차량의 순번, 그리고 차량의 정보들이 보여지고 차량의 정보는 다음 식 5.1과 같이 차량의 각 정보들을 나열하고 있다.

$$O_n^p (p=1, \dots, K) = f(C, F, R, P) \quad (4.1)$$

여기서 n은 시간을 나타내고, K는 움직이는 blob 수, f()는 특성들의 집합 즉, C는 중심 좌표, F는 차량 확소의 원 분포 그리고 R은 넓이와 높이의 비율, P는 상자 안의 차량에 대한 픽셀의 분포를 나



```

Multi_Vehicle_Track()
{
    Back_Inital();           // 초기 배경 화면 처리
    Buffer0 = Back_Inital(); // buffer0에 배경화면 입력
    Buffer1 = GetFrame();    // 현재 frame을 Buffer1에 입력
    Buffer2 = GetFrame();    // 배경처리를 위한 현재 frame 입력
    Vehicle_segmentation() // 차량 추출
    {
        Vehicle_seg = Buffer1 - Buffer0; // 차(difference) 이미지 얻기
        get_vehicleFeatur(Vehicle_seg) // 차량 정보 얻기
        {
            int Vehicle_center[2]; // 차량 중심
            float PixelCircle;     // 원 분포
            float WidthHeight;    // 넓이와 높이 비
            float RatioVehicleBox; // 차량의 분포
        }
        Adaptive_background() // 배경 이미지 처리
        {
            HistogramInital = Histogram(Buffer0); // 배경이미지 히스토그램
            HistogramCurrent = Histogram(Buffer2); // 현재이미지 히스토그램
            Diff = |HistogramInital - HistogramCurrent|;
            if ( diff > threshold )
                Buffer0 = Buffer2; // 새 배경이미지 입력
        }
    }
    Vehicle_Tracking(VehicleFeatur)
    {
        kalmanFilter(); // 차량 위치 측정, 예측, 수정
        Occlusion()
        {
            if ( WidthHeight > 0.51 )
            {
                if (RatioVehicleBox > 0.92)
                    occlusion = 0; // 위 조건을 만족하면 겹침 현상 없음
            }
            else
                occlusion = 1; // 겹침 현상 있음
        }
    }
}

```

그림 4-2. 멀티 차량 추적 알고리즘

타낸다

끝으로 차량 겹침이 있을 경우에는 1을 없을 경우에는 0으로 표시하여 차량 겹침의 유무를 표시하였다.

4.2 차량 추적 알고리즘

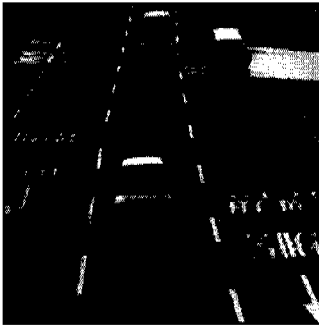
다음 그림4-2은 차량 추적 시스템의 알고리즘을 정리한 것이다.

4.3 실험 결과

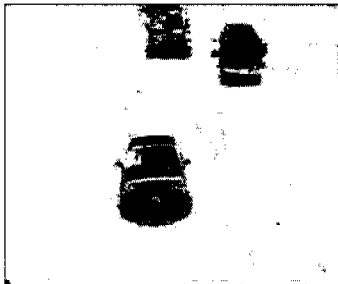
실험은 Offline에서 되었으며, 먼저 적응적 배경 모델을 이용하여 차량을 추출하는 방법과 연속적 배경 모델을 이용한 방법을 비교 하였다. 그 후 각 차량의 정보와 칼만필터를 이용하여 차량을 추적하

였다. 추적한 결과를 시각적으로 보이기 위하여 도료를 통과한 차량의 수를 측정하여 실제 수와 비교하여 추적이 효과적으로 이루어 졌는지에 대하여 실험하였다. 추적하는 과정에서 나타나는 차량 겹침을 처리하고 위하여 차량 특징 정보들을 이용하였다.

연속되어 들어오는 이미지와 배경 이미지와의 차(difference)를 이용하여 차량을 추출하게 된다. 여기서 얻어진 차량들은 추적되는 박스에 의해 각 차량마다 특징 정보들을 얻게 된다. 그리고 배경 이미지는 명암 히스토그램을 이용하여 연속되어 들어오는 이미지와 비교, 배경이미지의 Update 판단을 하게 된다.



(a) 입력 영상



(b) 차량 분할 영상

그림 4-3. 차량 추출

4.3.2 차량 추적

차량의 추적은 다음 그림4-4와 같이 차량을 추적하는 상자를 사용한다. 칼만 필터를 통하여 예측된 좌표의 상자 안의 차량을 추적하고 각 차량의 특징을 이용하여 겹침 현상을 해결한다.

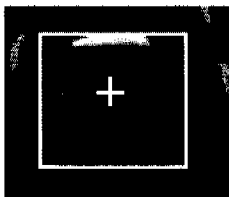


그림 4-4. 추적을 위한 박스

차량의 추적의 결과를 보이기 위하여 도로를 지나가는 차량의 수를 통해 나타내기로 하였다. 차량의 대수는 결과 분석의 기준 값을 마련할 수 있으며, 추적기의 기본적인 기능을 대표하는 기능으로 성능을 평가하는 주 항목으로 사용된다.

실험은 일반 3차선 도로에서 이루어 졌으며 오후 2시부터 6시까지 시간당 10분씩 측정하였다. 날씨는 태양과 구름이 있었고 주위 건물(아파트)의 그림자가 도로에 영향을 주었으나 전체적으로 드리워짐으로 인해 차량을 추출하는 데는 크게 문제가 되지 않았다.

겹침 현상은 차량의 특징들을 이용하여 각 차량들을 구분하여 주었으나 겹침이 심한 경우에는 차량 인식이 어려웠다. 이 부분에 대해서는 추적기를 통한 예측된 위치에 크게 의존하게 하여 추적하였으며 차량 구분이 되었을 때 그 값을 비교해 주었다.

표 1. 추적을 통한 차량 대수의 실험 결과

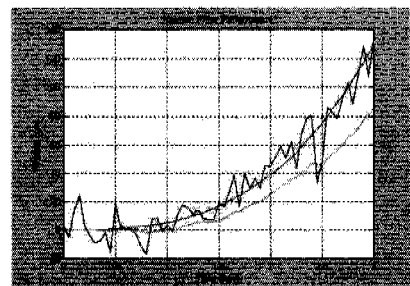
	날 씨	시 간	실제 통과대수	Missing	Error
1	태양 있고 맑음 (선명한 그림자)	14:00~ 14:10	78	4	5.1%
2	태양 있고 맑음 (선명한 그림자)	15:00~ 15:10	83	6	7.2%
3	태양 없고 흐림 (넓은 그림자)	16:00~ 16:10	98	10	10.2%
4	태양 없고 흐림 (그림자 적음)	17:00~ 17:10	106	11	10.4%
5	저녁 무렵	18:00~ 18:10	177	23	13%

표1에서 보는 바와 같이 에러율은 평균 9.18%를 보여주고 있다. 에러의 85%를 차지하는 가장 큰 이유 중에 하나는 큰 차 뒤에 작은 차가 따라오면서 화면에 가려져 추적이 어려운 상태이다.

다음은 칼만 필터를 이용하여 각 움직이는 차량의 궤도에 관한 예측과 실제로 측정된 궤도와의 오차를 Matlab을 이용하여 보여주고 있다.



(a)



(b)

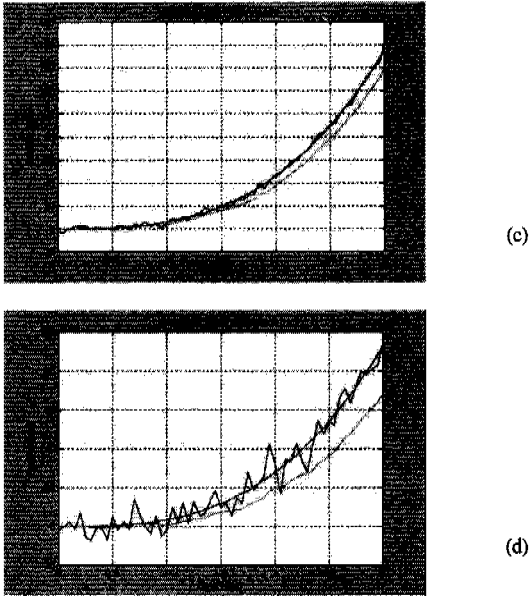


그림 4-5. 칼만 필터를 이용한 다중 차량 추적  
 (a) 입력된 이미지 (b) a의 차량  
 (c) b의 차량 (d) c의 차량

특징 정보를 이용하여 실험한 결과는 다음 그림 4-6과 표2와 같다.

표 2. 차량 특징 정보

Image	Car	C (중심좌표)	$R^k$	$P^k$	Occlusion 여부
1	1	x:208 y:91	0.87	0.95	X
	2	x:131 y:65	0.5	0.97	X
	3	x:94 y:43	0.81	0.97	X
2	1	x:222 y:125	0.82	0.94	X
	2	x:131 y:74	0.57	0.97	X
	3	x:94 y:60	0.77	0.96	X
3	1	x:236 y:164	0.70	0.95	X
	2와 3	x:114 y:100	0.82	0.65	0
4	2와 3	x:117 y:128	0.93	0.65	0
5	2와 3	x:117 y:171	0.56	0.62	0
6	2	x:134 y:230	1.37	0.92	X
	3	x:108 y:159	0.84	0.93	X
7	3	x:108 y:199	0.75	0.95	X
	4	x:94 y:17	0.99	0.97	X
8	4	x:85 y:40	1.00	0.96	X
	5	x:179 y:23	0.58	0.95	X

### 4.3.3 차량 겹침

차량 겹침 현상은 차량의 정보를 가지고 추적하는 가운데서 일어난다. 따라서 화면에 보이는 차량들을 추적하면서 차량 겹침의 유무 판단을 하여야 한다. 차량 겹침 유무의 판단은 차량 추적에 이용되는 상자의 넓이와 높이의 비율  $R^k$ 와 상자 안의 차량이 화소 비율  $P^k$ 을 이용한다.

차량의 번호는 위의 실험에 보이는 차량의 앞부분부터 정하여 놓았다. 위의 표에서 보는 것과 같이 각각의 단일 차량을 추적하는 것과 각 단일 차량 두 대가 겹침 현상을 보인 후, 해제되는 결과를 보이고 있다.

다음은 연속되어 나타나는 겹침에 관한 실험 결과이다.

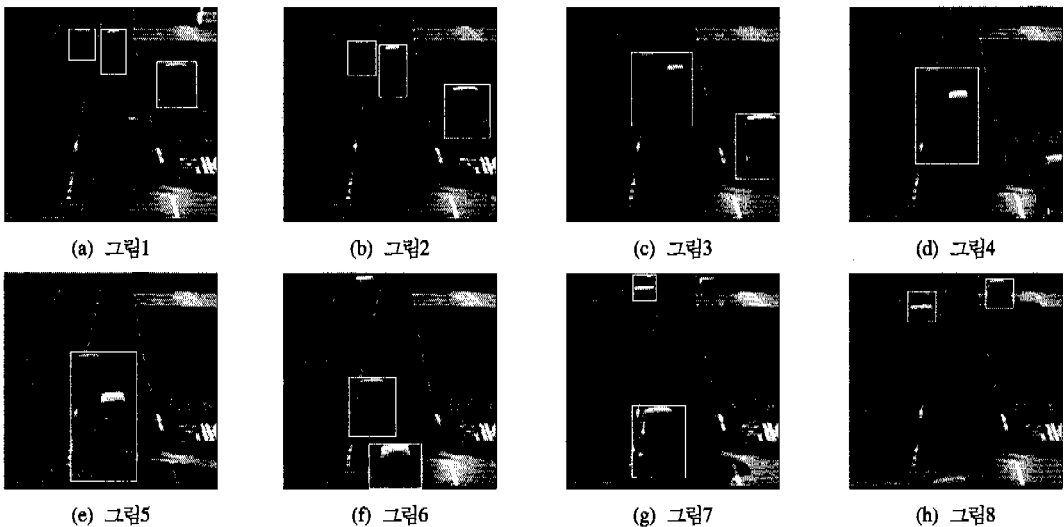


그림 4-6. 차량 추적과 겹침

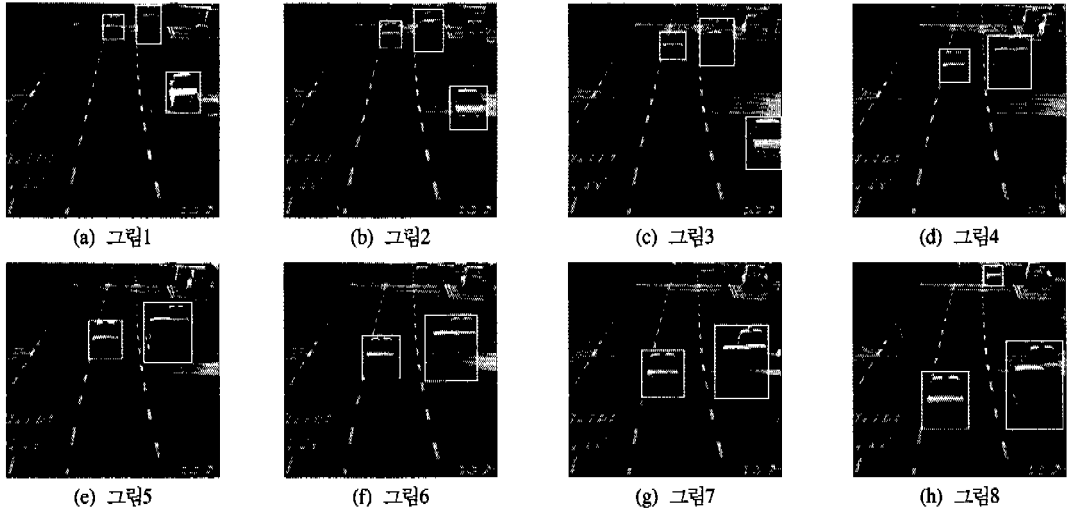


그림 4-7. 차량 추적과 겹침 #2

표 3. 차량 특징 정보 #2

Image	Car	C (중심좌표)	$R^k$	$P^k$	Occlusion 여부
1	1	x:213 y:105	0.98	0.94	X
	2와3	x:171 y:23	0.63	0.89	0
	4	x:128 y:28	0.97	0.97	X
2	1	x:222 y:125	0.98	0.94	X
	2와3	x:174 y:31	0.75	0.85	0
	4	x:128 y:37	0.98	0.97	X
3	1	x:236 y:165	0.68	0.95	X
	2와 3	x:178 y:46	0.75	0.83	0
	4	x:125 y:48	0.98	0.65	-
4	2와 3	x:188 y:68	0.81	0.78	0
	4	x:122 y:75	0.93	0.95	X
5	2와 3	x:193 y:80	0.82	0.78	0
	4	x:117 y:91	0.88	0.95	X
6	2와 3	x:199 y:97	0.8	0.78	0
	4	x:117 y:111	0.89	0.94	X
7	2와3	x:208 y:117	0.68	0.79	0
	4	x:114 y:131	0.95	0.93	X
8	2와 3	x:216 y:142	0.66	0.80	0
	4	x:111 y:162	0.83	0.93	X
	5	x:169 y:14	1.06	0.94	X

표 4. 차량 겹침 처리 방법 비교

비교 \ 방법	카메라와의 거리를 이용하는 방법	3D 모델을 이용한 방법	Occlusion Reasoning을 이용한 방법	본 시스템
각 차량 구별	가능	가능	불가	가능
차량에 대한 사전 지식	불필요	필요	불필요	불필요
차량 구별	어려움	가능	어려움	가능
연속적인 겹침에 대한 처리	불가	불가	불가	가능

#### 4.4 고찰

본 논문에서 제안한 다중차량 시스템에 적용될 적응적 배경모델을 이용한 배경이미지 개선에 관한 실험과 칼만 필터와 차량의 특징들을 이용하여 차량들을 추적하고 겹침 현상이 일어나도 다중 차량들을 추적하는 것을 보였다.

다중 차량을 추적하면서 각 차량의 정보를 연계 시키기 위하여 각 차량의 특징 즉, 픽셀의 원 분포도를 이용하고, 특히, 차량 겹침 현상이 일어났을 경우 차량 추적에 이용되는 차량의 넓이와 높이의 비율, 추적 상자 안의 차량이 차지하는 화소의 비율을 이용하여 겹침의 유무를 판단할 수 있었고 차량의 겹침의 경우에는 단일 차량을 추적하는 방법과 같이 하나 후에 겹침이 해제 되었을 경우 각각의 차량은 그 전의 정보를 공유할 수 있었다.

다음은 다른 시스템에서 제안된 방법과 본 시스템에 사용된 방법을 비교하였다.

하지만 사람의 눈으로도 인지할 수 없는 경우 즉 큰 차 뒤에 작은 차가 계속 따라오면서 보이지 않은 경우에는 추적 시스템에서도 인지가 되지 않았

다. 이것은 카메라의 위치가 차량의 정면에 높이 4.5M의 위치해 있음으로써 나타날 수 있는 현상이었다. 따라서 카메라의 위치를 변경하여 이를 해결할 수 있을 것이다.

## V. 결론

현재 차량의 수가 급증을 하면서 교통 관제에 대한 관심이 높아지고 있으며 이를 컴퓨터를 이용하여 자동적으로 해결하기 위해 많은 연구들이 진행되고 있다. 이 중에 다중 차량들을 추출하고 추적하는 것이 가장 핵심적인 연구이며 특히 다중 차량을 추적하는데 있어서 가장 문제가 되고 있는 차량의 겹침을 해결하기 위한 여러 가지 방법들이 제시되고 있다.

본 논문은 도로 영상에서 차량들을 추출하고 추적하는데 있어서 아직까지도 해결되지 못하고 있는 차량의 겹침 해결에 관하여 논하였다. 현재 교통 관제 시스템을 연구하는 곳에서는 여러 가지 방법들을 제시하고 있으나 대부분 제한된 조건에서 그 문제들을 해결하고 있다. 본 논문은 카메라 영상에 처음 나타날 때부터 카메라의 시야에 벗어날 때 까지 계속 차량 겹침이 일어나는 경우에도 이를 인지하고 추적할 수 있는 방법에 관하여 실험하였다. 차량의 특징을 이용하여 다른 시스템에서 다루지 않는 영상 이미지에서 계속 되어지는 차량 겹침 현상을 다루었고 효과적으로 결과를 얻을 수 있었다.

다중 차량 추적 시스템은 앞으로 고속도로, 국도 및 시내 도로에서 교통 흐름 파악, 제어에 활용될 수 있을 것이며, 특히 차량의 겹침을 처리함으로써 정확한 차량의 흐름을 파악하고 각 도로의 활용방안에 대한 정보를 제공하여 교통 제어에 원활한 대처를 가능케 할 것이다.

그러나 한 차량이 다른 차량에 의해 영상에 나타나는 모습이 거의 적을 경우는 차량 추적하는 시스템에서 공통적으로 갖는 제약이다. 이러한 제약을 해결하기 위한 지속적인 연구가 계속되어야 할 것이다

## 참고 문헌

[1] 도로교통 안전협회, "영상처리 차량검지기 개발 및 효율성평가에 관한 연구", 1994  
 [2] 도로교통 안전협회, 교통개발연구원, "영상검지 체계 실용화 방안 및 확대 방안에 관한 연구",

1996.12  
 [3] Dieter Koller, Joseph Weber, and Jitendra Malik, "Robust Multiple Car Tracking with Occlusion Reasoning", UCB Technical Report UBC/CSD-93-780, University of California, October, 1993.  
 [4] Nelson H.C. Yung and Adrew H.S. Lai "Detection of Vehicle Occlusion using a Generalized Deformable Model" Circuits and Systems, IEEE International Symposium on Volume: 4, 1998, Page(s): 154 -157 vol.4  
 [5] Young-Kee Jung and Yo-Sung Ho "Traffic Parameter Extraction using Video-based Vehicle Tracking" Intelligent Transportation Systems, 1999 IEEE/IEEE/ISAI International Conference, Page(s): 764 -769  
 [6] J.Badenas, J.M.Sanchiz and F.Pla, "Using Temporal Integration for Tracking Regions in Traffic Monitoring Sequences" IEEE Multimedia, Oct.-Dec. 2000, pp.1137-1140.  
 [7] R.G. Brown and P.Y.C.Hwang, Introduction to Random Signals and Applied Kalman Filtering, 2<sup>nd</sup> ed., John Wiley & Sons, Inc. 1992.  
 [8] D. Beymer, P. McLauchlan, and B.coifman, and J. Malik, "A Real-time Computer Vision System for Measuring Traffic Parameter," Proc. of CVPR 97, pp. 495-501, July 1997.  
 [9] 이철현, 김형태, 설성욱, 남기곤, 이장명, "칼만 필터를 이용한 다중 차량 추적 알고리즘," 대한전자공학회 논문지, 제36권, 5편, 제3호 pp.89-96, 1999.  
 [10] N. McFalane and C. Scholfield, "Segmentation and Tracking of Piglets in Images", Machine Vision and Application, Vol. 8, pp. 187-193, Oct. 1995  
 [11] Y.K.Jung and Y.S.Ho, "Robust Vehicle Detection and Tracking for Traffic Surveillance", Picture Coding Symposium '99, pp.227-230, April 1999  
 [12] A.M. tekalp, Digital Video Processing, Prentice Hall, 1993  
 [13] Karen Gettman, A Simplified Approach To Image Processing, Prentice Hall, 1997

강 은 구(Eun-ku Kang)                          준회원

2001년 2월: 광주대학교 대학원 컴퓨터 공학과  
석사과정

<주관심 분야>: 멀티미디어 영상처리, 영상통신

김 성 동(Seong-Dong Kim)                      정회원

한국통신학회 논문지 제24제26권 제1B호 참조

최 기 호(Ki-Ho Choi)                            정회원

한국통신학회 논문지 제24제26권 제1B호 참조