

# 교통 상황 감시 시스템을 위한 미들웨어 인터페이스의 설계 및 구현

강영모<sup>0</sup>, 김재영, 남종호  
서강대학교 컴퓨터학과

{jec1979<sup>0</sup>, young0kr}@mlneptune.sogang.ac.kr, jhnang@ccs.sogang.ac.kr

## Design and Implementation of Middleware Interface for Traffic Surveillance System

Youngmo Kang<sup>0</sup>, Jaeyoung Kim, Jongho Nang  
Department of Computer Science, Sogang University

### 요 약

교통 상황 감시 시스템은 감시 카메라로 촬영된 영상을 입력으로 받아 도로 상의 자동차를 탐지, 추출, 추적 및 인식하고 그것으로부터 고급 수준의 정보 즉, 특정한 이벤트를 탐지하기 위한 목적으로 널리 사용되고 있다. 하지만 기존의 감시 시스템들은 불법주차나 교통 상황 분석과 같은 특정 목적만을 위해 설계되었기 때문에, 확장성이나 호환성이 고려되지 않고 설계 및 구현되었다. 본 논문에서는 이러한 기존 시스템의 문제를 해결하고 보다 효율적인 시스템의 설계를 위하여, 교통 상황 감시 시스템에 대한 총 5 계층의 미들웨어를 제안한다. 그리고 제안하는 미들웨어를 구성하기 위해서 총 45개의 인터페이스를 직접 구현하였고, 이를 통해 기존 시스템의 문제점인 확장성과 호환성을 해결하였다.

### 1. 서 론

교통은 현대의 도시 생활에서 의식주와 비견될 만큼, 시민들의 생활에 있어서 필수적인 요소로 자리 잡았다. 하지만 교통의 대중화에 따른 교통량의 증가로 인해, 도로 상의 교통 상황을 어떻게 효율적으로 관리할 것인가에 대한 논의가 활발히 이루어지고 있다. 최근에는 감시 카메라를 활용하여 교통 상황을 수동으로 직접 관리하고 있지만, 이러한 방법은 사람이 촬영된 영상을 직접 관찰하고 실시간으로 처리하는 데에 있어서 번거로운 작업이 많다는 문제점이 있다. 따라서 교통 상황을 자동으로 관리할 수 있는 시스템이 필요하다.

이러한 자동화된 감시 시스템은 카메라로 촬영된 영상을 컴퓨터 비전과 이미지 프로세싱 기법을 통해, 도로 상의 자동차를 탐지, 추출, 추적 및 인식한다. 그리고 나아가 자동차의 움직임을 기록 및 분석한 후, 사람이 이해할 수 있는 고급 수준의 정보를 기술한다. 즉, 이것은 사람의 개입이 없이도 도로 상에서 발생하는 비정상적인 사건 및 사고를 탐지할 수 있게 한다.

기존의 교통 상황 감시 시스템들을 살펴보면, 다양한 고급 정보를 추출하기 위해서 Traffic Flow Monitoring System[1], Abnormal Incident Detection System[2], Intelligent Transportation System[3] 등과 같은 여러 가지 시스템들이 있다. 하지만 이들은 불법 주차, 교통 상황 분석 등과 같은 특정한 목적을 위해 설계되었기 때문에, 다른 목적으로 사용되기 위한 확장성이 고려되지 않았고, 다른 시스템과의 호환성 측면에서 문제가 있다. 따라서 본 논문에서는 이러한 기존 시스템의 문제점인 호

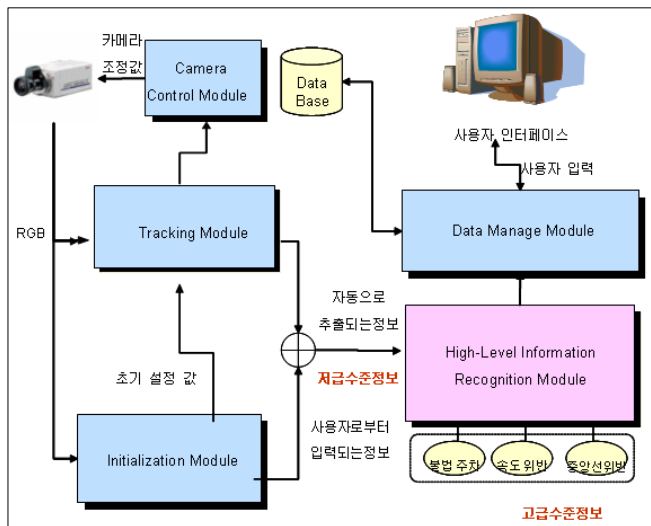
환성과 확장성을 해결하고 보다 효율적인 시스템의 설계를 위해서, 교통 상황 감시 시스템에 대한 미들웨어 인터페이스를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 교통 상황 감시 시스템의 전체적인 구조를 살펴보고, 3장에서는 교통 상황 감시 시스템을 구성하기 위해 본 논문에서 제안하는 미들웨어 인터페이스의 설계 및 구현에 대해서 다루며, 마지막 4장에서는 본 연구에 대한 결과에 대해 서술하고 마지막으로 결론을 맺도록 한다.

### 2. 교통 상황 감시 시스템의 전체적인 구조

본 논문에서 제안하는 교통 상황 감시 시스템은 기본적으로 카메라 조절 모듈, 시스템 초기화 모듈, 물체를 추적하는 모듈, 고급 수준 정보를 추출하는 모듈, 데이터를 관리하는 모듈로 구성<그림 2-1>된다. 카메라 조절 모듈은 원하는 도로의 영상을 촬영하기 위하여 pan, tilt, zoom 동작을 할 수 있도록 카메라의 파라미터 값을 조정하는 모듈이다. 그리고 시스템 초기화 모듈은 감시 카메라의 설치를 위한 초기화 값을 조절하고 도로의 구조 정보를 구성하는데 사용된다. 물체를 추적하는 모듈은 카메라로부터 입력된 영상에서 배경이미지를 생성하여 오브젝트를 추출하고 추적하는 모듈이다. 고급 수준 정보 모듈은 추적 모듈에 의해 알게 된 저급 수준 정보를 이용하여 자동차가 어떤 행동을 하는지 판단하는 모듈이다. 마지막으로 데이터 관리 모듈은 고급 수준 정보를 데이터베이스에 기록하는 역할을 담당한다.

이와 같은 전체 시스템 구성에서의 입력 정보는 감시 카메라로 촬영되는 영상이고, 출력 정보는 탐지되는



<그림 2-1> 교통 감시 시스템의 전체적인 구조

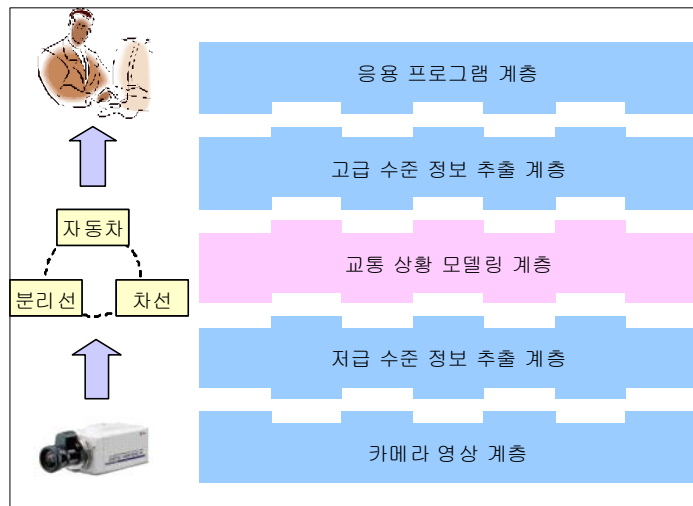
이벤트가 된다. 실제 출력 정보인 이벤트 즉, 고급 정보를 추출하기 위해서는 먼저 영상 이미지에서의 저급 수준 정보 다시 말해, 특징(Feature)값들을 추출해야만 하고 이렇게 얻은 저급 수준의 정보를 사용하여 실제 교통 상황 감시 시스템에서 필요한 고급 정보를 구성하게 된다. 고급 수준의 정보는 크게 세 가지로 분류할 수 있는데, 하나는 자동차의 상태에 대하여 기술(Description)할 수 있는 고급 수준 정보이고, 다른 하나는 도로의 상태에 대하여 기술할 수 있는 고급 수준 정보이다. 그리고 마지막 하나는 자동차의 행동에 의해 발생하는 이벤트를 기술할 수 있는 고급 수준 정보이다.

자동차의 상태를 기술할 때 사용되는 정보로는 자동차의 색상, 종류, 자동차 번호 등이 있고, 도로 상의 교통 상황을 기술하는 정보로는 현재 도로가 교통 체증으로 인해 막히는 구간인지 아닌지에 대한 기술이 포함된다. 그리고 도로의 상태를 기술하는 정보에는 도로 상에서 일어나는 비정상적인 사건들 예를 들면, 중앙선 침범, 속도위반, 앞지르기 금지위반과 같은 여러 가지 위반 행위들이 포함된다.

기존의 시스템들은 저급 수준의 정보로부터 고급 수준의 정보를 추출하기까지의 과정이 하나의 프로세스로 동작하는 구조를 갖고 있다. 따라서 새로운 알고리즘을 적용하거나 기존의 것을 수정할 때, 전체 프로세스를 고쳐야 하는 문제점을 지닌다. 또한 서로 다른 목적을 위해 설계되었기 때문에, 새로운 시스템과의 호환성 측면에서도 문제점을 갖고 있다. 따라서 이러한 확장성과 호환성을 높이고, 다양한 시스템을 설계할 때 개발비용을 절감하고 효과적인 시스템 설계를 위해서, 표준화된 인터페이스가 필요하다.

### 3. 교통 상황 감시 시스템을 위한 미들웨어 인터페이스

본 논문에서 제안하는 교통 감시 시스템의 미들웨어는 <그림 3-1>와 같이 총 5개의 계층으로 이루어져 있다. 이 중 가장 핵심적인 계층은 중앙에 위치한 모델링 계층



<그림 3-1> 제안된 교통 상황 감시 시스템의 구조

으로서, 세 가지 객체인 자동차, 차선, 차선 분리선으로 구성되고, 각 객체간의 관계를 통해 이벤트를 정의한다.

각각의 객체는 하위 계층의 저급 수준 정보를 이용하여 구성된다. 저급 수준의 정보는 두 종류로 나눌 수 있는데, 하나는 카메라를 이용해 촬영된 영상으로부터 자동으로 추출할 수 있는 정보이고, 다른 하나는 사용자의 입력에 의해 구성되는 정보이다. 자동으로 추출되는 정보에는 배경(Background)과 전경(Foreground) 사이의 픽셀 값 차이를 이용하여 얻을 수 있는 오브젝트 정보 [4]와, 그 오브젝트가 움직이는 궤적(Trajectory) 정보가 포함된다. 이러한 저급 수준의 정보를 통해서 자동차 객체에 대한 속성값을 얻을 수 있다. 예를 들어, 자동차 객체는 차종, 색상, 위치의 속성을 가질 수 있고, 추가적으로 자동차의 번호 또한 인식이 가능한데, 영상 내에서 번호판의 위치를 추적하여 번호판의 이미지를 구한 다음, 학습을 통해 자동차의 번호를 인식[5]하는 방법을 사용한다. 그리고 사용자의 입력으로부터 얻을 수 있는 저급 수준 정보에는 도로에 대한 구조 정보 즉, 도로의 영역이나 종류, 차선의 곡선과 종류, 주행 방향의 정보 등이 포함된다. 마찬가지로 차선 객체와 차선 분리선 객체의 속성값도 사용자가 입력한 저급 수준 정보를 사용해 구성할 수 있다.

교통 상황 모델링 계층에서는 각 객체들의 관계를 이벤트로 정의할 수 있는데, 교통 상황의 감시가 그 목적이므로, 우리의 시스템에서는 자동차 객체를 중심으로 하는 상관관계만 필요하다. 예를 들어, '자동차 객체와 차선 객체의 관계'로부터 주행 방향이나 주행 속도 또는 차종에 관련된 교통 범칙 이벤트를 추출해낼 수 있다. 마찬가지로 '자동차와 차선 분리선의 관계'로부터는 중앙선 침범, 유턴 금지, 진로 변경 제한과 같은 이벤트들이 추출 가능하며, '자동차와 자동차의 관계'를 이용하여 안전거리 미확보, 끼어들기 등의 이벤트를 추출할 수 있다.

카메라 영상 계층에서는 입력으로 사용될 영상 이미지를 제공하는 인터페이스들을 제공한다. 그리고 저급 수준 정보 추출 계층에서는 영상 이미지를 제공받아 원하는 오브젝트의 정보와 도로 구조의 정보를 생성하는 인

모듈	함수	기능 설명
자동차와 차선 사이의 이벤트 모듈	void DetectEventByDirection(void)	주행 방향에 의한 이벤트를 탐지
	void DetectEventBySpeed(void)	주행 속도에 의한 이벤트를 탐지
	void DetectEventByCarModel(void)	차종에 의한 이벤트를 탐지
자동차와 차선 분리선 사이의 이벤트 모듈	void DetectEventByLaneChange(void)	차선 변경에 의한 이벤트를 탐지하는 함수
	boolean IsOverlapping(CarEntity *, DividingLineEntity *)	자동차와 차선 분리선 위에 있는지 판단
자동차 사이의 이벤트 모듈	void DetectEventByCarDistance(void)	자동차들의 거리에 의한 이벤트를 탐지
	CarEntity * FindRearCar(CarEntity *, LaneEntity *)	뒤에 위치한 자동차를 찾음
	CarEntity * FindFrontCar(CarEntity *, LaneEntity *)	앞에 위치한 자동차를 찾음
	CarEntity * FindSideCar(CarEntity *, LaneEntity *)	옆 차선에 위치한 자동차를 찾음
	int GetDistance(CarEntity *, CarEntity *)	두 자동차 사이의 거리를 계산

<표 3-1> 고급 수준 이벤트 탐지 계층의 인터페이스

터페이스들을 제공한다. 오브젝트의 정보의 경우는 자동적으로 추출이 가능하지만, 도로 구조의 정보의 경우는 사용자의 입력을 받을 수 있는 GUI가 사용되도록 설계하였다. 그리고 교통 상황 모델링 계층의 경우는 앞서 추출된 저급 수준 정보들을 이용하여 자동차, 차선, 차선 분리선의 세 가지 객체들을 생성하는 인터페이스들을 갖고 있다. 그리고 가장 핵심적인 고급 수준 이벤트 탐지 계층에서는 각 객체간의 관계 정보를 얻고, 그 정보를 활용하여 여러 가지 이벤트를 탐지해내는 인터페이스들을 제공한다.

전체 시스템의 실제 구현은 MFC를 사용하여 제작되었으며, 오브젝트 추출 및 추적을 위해 인텔사에서 개발한 OpenCV라는 비전 프로세스 라이브러리를 사용하였다. 구현된 인터페이스들은 5개의 계층을 통틀어, 총 45개로 이루어져 있으며, 각각의 인터페이스는 자동차와 차선, 자동차와 차선 분리선, 자동차와 자동차의 관계를 이용하여 고급 수준의 이벤트들을 탐지하는 역할을 한다. 그림 <표 3-1>는 4번째 계층인 고급 수준 이벤트 탐지 계층의 실제 설계 및 구현한 인터페이스의 예를 보여주고 있다.

#### 4. 연구 결과와 결론

본 논문에서 설계한 미들웨어 인터페이스들은 기존의 시스템들이 가졌던 문제점인 확장성과 호환성뿐 만 아니라, 개방성, 직교성, 일관성을 포함한 총 5가지 측면에서의 요구사항을 만족시킨다. 이를 증명하기 위해, 확장성은 제안한 인터페이스들을 통해 실제 도로 상의 다양한 이벤트들을 탐지해낼 수 있음을 보임으로써 확인하였고, 호환성은 시스템의 입력을 카메라 영상뿐 만 아니라, 일반적인 동영상 파일로 교체할 수 있음을 통해 보일 수 있었다. 그리고 개방성은 모델을 변형함으로써 새로운 이벤트 역시 탐지할 수 있음을 통해 보였다. 또한 직교성을 보이기 위해서 인터페이스가 제공하는 기능을 비교

한 결과, 서로 간에 중복되는 기능이 없음을 확인하였다. 마지막으로 일관성을 보이기 위해서는 각 계층 간에 사용하는 입출력 정보를 비교하였고 그 결과, 각 계층이 사용하는 정보는 그 계층에서만 사용 및 관리됨을 확인하였다.

이와 같이 본 논문에서 제안한 미들웨어 인터페이스를 사용하여 교통 상황 감시 시스템을 개발한다면, 객체 간의 다양한 조합을 통해 새로운 이벤트의 정의 및 추출할 수 있고, 기존 시스템들이 가졌던 확장성 및 개방성 문제점들을 추가적으로 해결할 수 있다.

#### 5. 참고 문헌

[1] K. Kanayama, Y. Katoh, and K. Tsuji, "A Car Detection System Using Neural Network for Image Processing," in Proc. of 25th Int. Symp. Automotive Technology and Automation, Road Transport Informatics and Intelligent Vehicle System, 1992.

[2] H. Ikeda, T. Matsuo, K. Tsuji, and Y. Kaneko, "Abnormal Incident Detection System Employing Image Processing Technology," in Proc. of IEEE Int. Conf. on Intelligent Transportation Systems(ITS'99), pp. 748-752, October 1999.

[3] M. Aoki, "Imaging and Analysis of Traffic Scene," in Proc. of IEEE Int. Conf. on Image Processing(ICIP'99), Vol.4, pp. 1-5, 1999.

[4] S. Chang, W. Chen, and H. Meng "A Fully Automated Content-Based Video Search Engine Supporting Spatiotemporal Queries," in IEEE Trans. on Circuit and System for Video Tech., vol.8, no.5, September 1998.

[5] Y.Huang, S. Lai, and W. Chuang, "A Template-Based Model for License Plate Recognition," in Proc. of Int. Conf. on Networking, Sensing and Cotrol, March 2004.