

영상과 음향 기반의 교차로내 교통사고 검지시스템의 구현

An Implementation of Traffic Accident Detection System at Intersection based on Image and Sound

이형석*, 김영욱, 권대길, 박기현, 이경복, 한민홍

(Hyung-Suk Lee, Young-Wook Kim, Tai-Gil Kwon, Kee-Hyun Park, Kyong-Bok Lee, and Min-Hong Han)

Abstract : The frequency of car accidents is very high at the intersection. Because of the state of a traffic signal, quarrels happen after accidents. At night many cars run away after causing an accident. In this case, accident analyses have been conducted by investigating evidences such as eyewitness accounts, tire tracks, fragments of the car or collision traces of the car. But these evidences that don't have enough objectivity cause an error in judgment.

In the paper, when traffic accidents happen, the traffic accident detection system that stands on the basis of images and sounds detects traffic accidents to acquire abundant evidences. And, this system transmits 10 seconds images to the traffic center through the wired net and stores images to the Smart Media Card. This can be applied to various ways such as accident management, accident DB construction, urgent rescue after awaring the accident, accident detection in tunnel and in inclement weather.

Keywords : intersection, image processing, sound processing, accident detection, neural network

I. 서론

날로 심각해져 가는 교통문제를 효과적으로 해결하기 위한 도로교통시스템의 지능화 작업이 '90년대 초반부터 시작되어 현재 기획단계를 지나 현장시험, 시범운영, 서비스 제공 단계에 이르렀다. 도심에서는 교차로내 사고 발생빈도가 전체 사고의 50% 이상을 차지하고 있다. 특히 교차로 사고는 신호등의 상태에 따라 빈번하게 발생하므로 이로 인한 사고 후 분쟁 발생이 많다. 또한 야간 교차로 사고시에는 사고 후 도주하는 차량이 많이 발생하는데, 이 경우 목격자의 증언이 절대적으로 필요하게 된다. 그러나 사고발생시 목격자 증언의 오보나 혹은 사고 사실 조작으로 인해 사고 처리시 억울한 피해자가 지속적으로 발생하고 있으며 이는 사회적인 문제로서 자주 대두되곤 한다.

현재까지의 사고분석은 타이어 스키드 자국이나 파편 및 차체의 충돌 혼적 등의 사후 증거만을 활용하여 진행되어 왔다. 이러한 사후 증거는 그 증거자료가 불충분할 뿐 아니라 잘못된 결과를 일으킬 가능성이 많다. 그러나 만일 교차 소지를 없앨 수 있고, 사고처리 비용 및 시간을 줄일 수 있어 도심 교통 체증 완화에도 기여할 수 있을 것이다.

따라서 본 논문은 이러한 문제점에 착안하여 교차로내에서 차량 사고시 사고발생시점의 영상을 유선망을 통해 교통

센터와 Smart Media Card로 전송하게 하였다. 사고발생시점을 검지하는 부분은 음향과 영상 2가지를 적용한다. 먼저 음향에 의한 사고검지 방법은 음향처리보드에서 FFT 변환 과정과 고역 통과 필터(High Pass Filter)를 통과 시켜서 만든 음향의 강도(Intensity)와 신경망(Frequency)을 획득하여 역전파(Neural Network)이론 중 Back Propagation 알고리즘을 적용하여 사고음을 학습시키고 이를 인식하도록 하였다. 또한 영상에 의한 사고검지방법은 초당 일정frame을 처리 할 수 있는 영상처리보드에 입력되는 RGB 칼라영상 중 특정값을 추출해서 영상처리를 통하여 사고를 인식하도록 하였다.

II. 시스템 구성

교차로내 교통사고 검지 시스템은 크게 영상처리부와 음향처리부로 나눌 수 있다. 그림 1에서 영상처리부는 CCD 카메라, 신호등 정보, 영상처리 DSP보드로 구성되며 음향처리부는 마이크, 음향처리DSP보드로 구성된다.

음향처리부는 교차로에 설치된 마이크로부터 들어오는 아날로그 음향을 음향처리보드에서 FFT 형태의 디지털데이터로 변환 후 음향분석을 통해 사고시점을 판단한다. 만약 사고라고 판단되면 영상처리보드에서 사고전후 영상을 획득 할 수 있도록 사고발생 신호를 보내준다.

영상처리부는 교차로 주변 폴대에 설치된 카메라와 신호등에서 교차로 주변영상과 신호등 상태에 대한 정보가 영상처리보드에 실시간으로 입력이 된다. 이렇게 입력된 영상들은 사고검지를 위해 영상분석을 실시한 후 영상처리보드내의 SDRAM에 저장된다.

SDRAM내부에는 현재시점을 기준으로 항상 최근 100frame 영상이 저장되고 갱신되어진다. 만약 영상과 음향분석에 의해 사고라고 판단되어지면 SDRAM에 저장된 사

* 책임저자(Corresponding Author)

논문접수 : 2003. 9. 15., 채택확정 : 2004. 3. 9.

이형석, 김영욱 : 고려대학교 산업시스템정보공학과
(trflight@korea.ac.kr/ksksyw@korea.com)

권대길 : 한국전자부품연구원(onekid@korea.ac.kr)

박기현 : TI KOREA(keehyun@hotmail.com)

이경복, 한민홍 : 고려대학교 산업시스템정보공학과
(masket@hotmail.com/mhhan@korea.ac.kr)

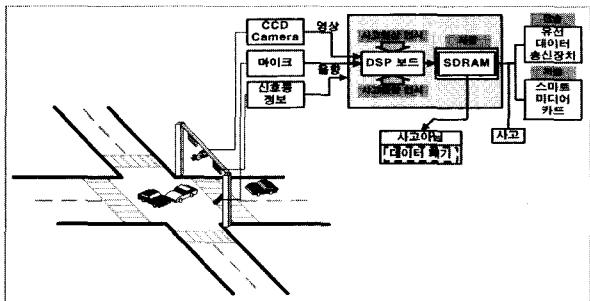


그림 1. 전체 시스템 구성도.

Fig. 1. A block diagram of full system.

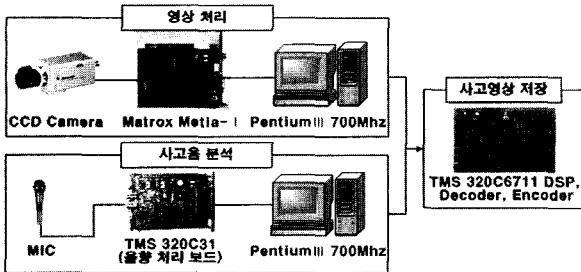


그림 2. 사고 영상 분석 및 저장을 위한 실험 환경.

Fig. 2. Experimental environment for analyzing an storing accident image.

고 전 영상 50프레임과, 사고 후 영상 50프레임을 유선 송신 장치를 통해 교통센터로 전송한 후 다시 Smart Media Card에 저장을 하게 된다. 이때 사고영상의 데이터량을 줄이고 전송속도를 빨리하기 위해 JPEG압축알고리즘을 이용하여 압축을 실시하게 된다[4,5].

III. 주행선 인식 알고리즘

본 연구에 사용된 시스템은 음향처리를 위한 TI사의 TMS320C31 개발 보드와 영상처리 및 영상 저장을 위해 직접 제작한 TMS320C6711 보드와 음향 분석을 위한 컴퓨터 프로그램으로 구성된다[15]. 그림 2는 사고 영상 분석 및 저장을 위한 실험 환경을 나타내고 있다.

사고 음향 분석을 위해 마이크로 사고음을 입력받아 오디오 코덱을 이용해 디지털 변환을 수행하고, 사고음을 분석할 수 있는 사고음 분석부와 사고영상 처리/저장을 위해 디코더와 인코더, FIFO RAM, SDRAM, SBSRAM이 장착된 TMS320C6711 보드부 그리고 영상획득을 위한 카메라로 구성된다. 또한 DSP 내에서 처리한 영상 처리 결과를 검증하기 위해 컴퓨터 상에 Matrox 영상 처리 카드를 장착하여 동일한 영상처리 알고리즘을 구현해 보았다[14].

영상처리를 위해서는 언급한바와 같이 카메라에서 출력되는 NTSC(National Television System Committee)로 구성된 영상신호를 디지털로 변환해주는 디코더부와 영상처리가 끝난 디지털 영상 데이터를 출력하기 위해 NTSC 신호로 변환해주기 위한 인코더로 구성되는데, NTSC 신호는 명암을 표현하는 휘도신호와 색을 표현하는 색신호 그리고, 화면의

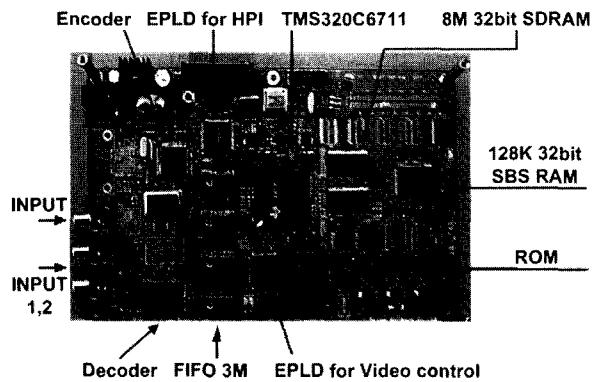


그림 3. 영상 처리 보드.

Fig. 3. Image processing board.

수직, 수평 동기를 맞추는 동기신호로 나누어진다[13].

디코더는 KS0127을 사용하였고 인코더는 BT868을 사용하였다. KS0127을 이용하여 NTSC 신호에서 Y8bit, U4bit, V4bit로 디지털화하고 이를 FIFO 메모리에 임시저장한다. 저장된 데이터를 DSP에서 Y신호만을 추출하여 영상 검지를 수행하고, 다시 RGB565포맷으로 변환한 후 SDRAM에 저장한다. 저장된 영상 데이터는 다시 FIFO 메모리에 임시저장되었다가 BT868에 의해 NTSC 신호로 변환되는 것이 본 영상처리저장보드의 중요 기능이다.

그림 3은 영상 처리 및 저장을 위해 직접 제작한 TMS320C6711 영상처리 보드이다[1]-[3]. 영상 입력을 위해 2개의 입력단이 구성되어 있고, 이를 디지털 신호로 변환할 수 있는 비디오 디코더(Video Decoder)가 연결되어있다. 비디오 디코더에서는 200*200 크기의 16bit RGB 컬라 영상으로 변환하여 4M의 FIFO RAM에 획득(Capture)된 순서대로 저장한다.

저장된 Image는 Decoder의 Vsync 신호에 의해 DSP에 인터럽트를 발생시키고, External Interrupt가 발생하면 16bit 단위로 DMA 7번을 이용하여 전송하고, 이미지 데이터는 SDRAM에 저장된다. SDRAM은 소프트웨어적으로 Ring 메모리로 구성하여 100개의 이미지를 순환하면서 종단의 이미지는 폐기시키는 방법으로 메모리를 관리한다. SDRAM은 총 64Mbyte의 용량을 가지는데 전반부의 50Mbyte는 100 프레임의 Raw 데이터 저장을 위해 사용되고, 후반부의 14Mbyte는 100 프레임 JPEG압축후의 데이터를 저장하는데에 사용된다. 알고리즘은 Flash ROM 또는 HPI를 통해 컴퓨터에서 다운로드 되도록 하여 SBSRAM으로 전송되도록 한다. 알고리즘은 SBSRAM에서 처리하고, 처리된 영상은 SDRAM에 저장되도록 하였다[11,12].

영상처리시 처리 후 결과 영상을 얻어내는 것이 매우 필요한데 이 보드에서는 기본적으로 엔코더(Encoder)인 BT868을 사용하여 결과 영상을 출력할 수 있는데, DSP에서 FIFO에 처리 후 결과 화면을 전송하면 EPLD에 엔코더의 타이밍을 마스터 모드로서 동기화할 수 있도록 로직을 작성하였다. 엔코더로 데이터 전송시에는 16비트로 설정된 DMA6번을 이용하여 데이터를 전송하며 Encoder에서는 640*480의

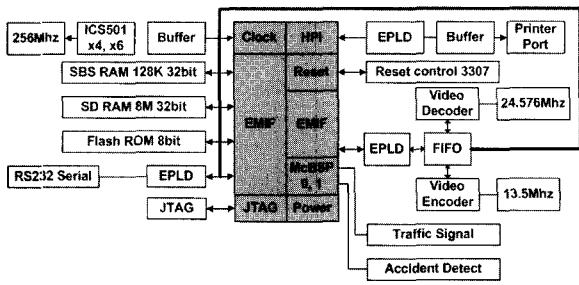


그림 4. 영상 처리 보드의 흐름도.

Fig. 4. Flowchart of image processing board.

아날로그 신호를 최종적으로 만들어 낸다. EDMA 카운터는 2차원으로 설정하여 0x4F000으로 설정하였다.

또한 Encoder외에 RS232 통신을 57600bps이상의 고속 설정을 통해 시리얼로 컴퓨터에 전송하도록 하였는데, 이를 위해 Serial 전용 칩이 사용되었다.

영상처리시 입력은 16비트 RGB 포맷으로 읽어들이고, 저장역시 RGB 16비트 포맷으로 수행한다. 그러나 영상처리시에는 알고리즘의 적절한 적용을 위해 RGB의 G 6비트를 상위비트로 사용하여 Gray 레벨로 알고리즘을 구현하였다.

그림 4는 영상처리 및 저장보드의 개략도를 나타낸다.

본 연구를 위해 제작된 보드는 각기능에 따라 다른 기능을 가진 3개의 Lattice사의 EPLD를 사용하였다. 첫 번째 EPLD는 디코더, 인코더, FIFO등의 영상 데이터 처리관련 칩을 제어하기 위한 것으로 DSP의 EMIF와 연결되어 있다. 두 번째 EPLD는 컴퓨터에서 프로그램을 SBS RAM에 전송할 수 있도록 HPI 제어를 위해 장착되었고, 버퍼를 통해 컴퓨터 프린터 포트와 연결된다. 세 번째 EPLD는 시리얼 칩과 각종 스위치 등의 입출력 신호를 지원하기 위한 I/O port를 위해 장착되었으며 DSP의 EMIF부와 연결되어 있다.

신호등 신호 입력은 적색, 녹색, 좌회전, 노랑색의 4가지가 입력되고, 사고 발생 측정시에 데이터를 출력하는데 McBSP 0번의 CLKS, CLKR, CLKX, FSR0, FSX0을 입출력 포트로 사용하였고, DX0, DR0을 이용하여 디코더, 앤코더 칩을 제어하기 위해 I2C 통신을 수행하였다. Clock은 칩의 구입상의 문제와 가격적인 면을 고려하여 클락 분주기인 ICS501 칩을 사용하여 쉽게 구할 수 있는 25MHz를 기준 Clock으로 하여 x4, x6으로 분주하였다. 이때 DSP에는 150Mhz의 Clock이 입력되고, EMIF부인 SDRAM, SBSRAM은 100Mhz가 입력된다. 이 보드에서는 전압이 불규칙해도 언제나 일정한 작동을 수행해야 하므로 전압변화가 0.5V 이상 차이가 나면 Reset을 다시 발생할 수 있도록 3307이라는 칩을 사용하여 Reset을 관리하였다.

IV. 사고시점 검지

사고시점 검지는 본 논문에서 핵심이 되는 사항으로 음향분석과 영상분석을 동시에 수행하여 측정한다. 음향으로 인한 사고시점 검지는 사고발생시 측정되는 음향을 분석하므로 높은 음향을 내는 사고음 측정시 신뢰도가 매우 높게 나타난다. 그러나 교차로 주변의 잡음이 발생할 경우나 낮은

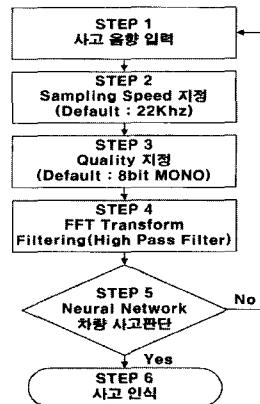


그림 5. 사고검지 음향처리 알고리즘 순서도.

Fig. 5. Flowchart of sound processing algorithm for accidental detection.

음향의 사고일 경우는 사고판단 인식율이 좋지 않았다.

그래서 이를 보완하기 위해 영상처리에 의한 사고시점 판단 방법을 재안하게 되었다. 그러나 영상처리에 의한 사고시점 판단방법도 차량이 많은 혼잡한 도로의 경우나 카메라로 들어오는 영상의 명도가 어두울 경우 오판을 가져올 가능성이 존재했다. 그 결과 음향과 영상에 의한 사고 시점 판단은 어느 한쪽만이 모든 환경 하에서 정확하다고 판단 할 수 없었다. 그래서 서로 배타적 관계가 아닌 상호보완의 관계로 적용 되고 있다.

1. 음향분석 처리

음향 분석을 위해 음향처리 보드에 입력되는 음향은 CODEC에 의해 아날로그 신호가 디지털 신호로 변환된다. 변환된 신호는 필요로하는 고주파 신호만을 남기고 저주파 신호를 제거하는 고역여파기 알고리즘을 사용하고 이를 FFT변환하여 진폭을 구한 후 샘플 음향의 주파수와 연관된 데이터를 생성해 낸다.

음향인식을 통한 사고발생 시점 판단을 위해서는 사고음의 주파수 패턴을 판별해야 한다. 일반적인 교통상황과 사고 발생에 따른 음향이 어떠한 영역의 주파수대에 있는지를 판단하여 사고 여부를 측정하게 되며 그림 5와 같은 절차를 따라 처리된다.

STEP 1

음향 처리의 순서를 살펴보면 먼저 마이크에 의해 교차로 주변의 잡음과 함께 사고음이 입력되게 된다.

STEP 2, 3

입력된 아날로그 음향을 CODEC에 의해 A/D 변환을 하는데, 이때 음질(8bit mono)과 샘플링 스피드(22Khz)를 정한 후 변환을 수행한다.

STEP 4

이렇게 변환된 디지털신호는 DSP에 의해 FFT(Fast Fourier Transform)을 수행하게 된다. 음향이 FFT변환이 되면 시간 영역에서 주파수 영역으로 변환되므로 신호 분석이 쉽게된다. FFT 변환된 주파수는 음, 양의 신호인 주기적인 파형으로 나타나는데 이 신호를 이용하여 주파수 스펙트럼으로 구하기 위해 절대값을 취해 준다. 주파수 스펙트럼으로

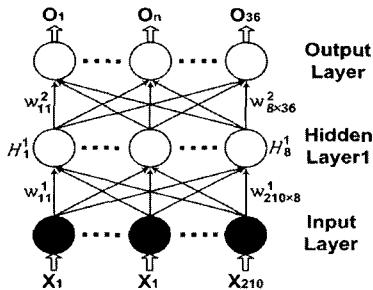


그림 6. 역전파 신경망의 구조.

Fig. 6. Structure of BPN.

부터 나온 신호를 고역 통과 필터를 통해 필터링 해주면 대부분의 저주파수 대역의 잡음이 제거된다.

STEP 5

사전에 사고음향이라고 판단된 데이터를 BPN을 이용해 학습시킨 후 이를 이용해 일정시간동안 모여진 음향 데이터가 사고음향인지를 판단한다.

STEP 6

사고가 인식되면 영상 저장을 위한 DSP보드에 신호를 보내게 되어 5초 후 영상입력이 종료되고, 영상 데이터가 저장된다.

2. 음향분석을 위한 Neural Network

그림 6은 음향 분석을 통하여 교통사고를 검지하는데 사용된 역전파신경망(BPN) 알고리즘의 신경망 구조를 나타내고 있다. 이 신경망은 원하는 고정된 출력패턴을 목표로 하는 Supervised 학습방법을 택하고 있다. 그래서 먼저 사고음향이라고 판단되는 음향데이터를 이용하여 사고사점 판단에 대한 선형 학습이 필요하다.

그림 6의 신경망 구조를 보면 입력층(input layer)에는 입력에 대응하는 210개의 뉴런(neuron)이 있으며, 은닉층은 8개, 출력층(output layer)에는 인식 음성에 대응하는 36개의 뉴런을 가진다. 은닉층을 구성하는 뉴런의 개수 및 alpha, beta 값들과 각 층 사이의 무게는 우수한 출력을 갖도록 시행착오적으로 설정하였다. 명령교시를 위해 사용된 사고음향은 36가지로 분류하였다. 각 출력 뉴런은 대응하는 인식 음향의 입력에 대하여 1이 출력되고 그 외의 경우에는 0이 출력되도록 학습하였다.

<입력층>

음향처리보드를 통해서 FFT변환을 거쳐 디지털 데이터로 변환된 음량 데이터를 일정한 시간 간격별로 모은다. 이렇게 모여진 시간대별로 모여진 데이터를 그래프로 나타내면 그림 7처럼 일정한 패턴의 계곡모양을 하고 있다.

여기서 시간대별로 그려진 하나의 계곡모양의 패턴들이 각각 하나의 입력층으로 사용되어진다. 이러한 하나의 입력 패턴은 210개의 노드수를 가지면 0에서 1사이의 값들로 일정하게 나타나게 된다. 즉 가장 큰 값은 '1'로 결정하고 가장 작은 값은 '0'으로 하여 그 사이에 나머지 값을 실제 음량의 비율에 맞게 집어넣는 것이다.

<출력층>

출력층은 총 36개의 노드로 구성 되어있는데 각 노드를

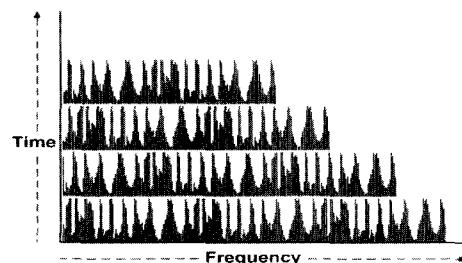


그림 7. 시간대별 음량 그래프.

Fig. 7. Volume graph each time zone.

표 1. 음향에 따른 출력 노드.

Table 1. Output node by sound.

음향	출력 노드
스키드음, 급브레이크음	1~10
충돌음	11~16
경적음	
차량이 고속으로 지나가는 소리	17~32
싸이렌음	
사고가 없는 교차로 잡음	33~34
아무런 소리의 입력이 없을 때	35~36

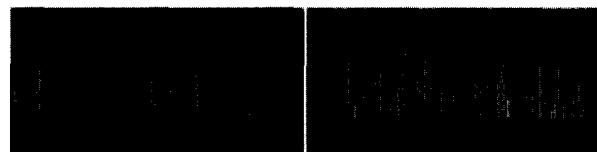


그림 8. 아무소리 없을 때.

Fig. 8. Noiseless.



그림 9. 충돌 사고음.

Fig. 9. Collision accident sound.

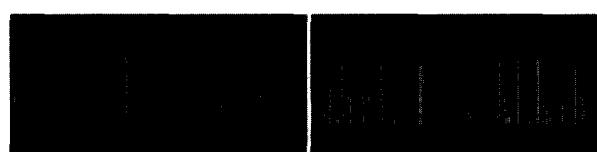


그림 10. 교차로 잡음.

Fig. 10. Intersection noise.



그림 11. 스키드 사고음.

Fig. 11. Skid accident sound.

은 각각 사고음의 한 종류나 잡음의 한 종류를 대변한다.

즉 수집된 샘플들을 선별하여 품질이 좋은 음향들을 각각의 노드에 대응시킨 것이다. 이것을 자세하게 살펴보면 표 1과 같다.

<학습>

학습에 필요한 실제 교차로 사고음향은 현실상 녹취하기 힘들어 사고음향을 많이 보유하고 있는 기관 및 업체를 통해 입수한 음향을 가지고 학습을 실시하였다.

학습은 그림 8~그림 11과 같은 입력패턴들을 가진 36개의 샘플에서 입력 자료를 추출하였다. 소리는 아날로그로 전송되기 때문에 같은 소리를 들더라도 매번 조금씩 다른 FFT 과형을 보여준다. 그러므로 학습 후 나타나는 예러를



그림 12. 사고검지 영상처리 알고리즘 순서도.

Fig. 12. Flowchart of image processing algorithm.

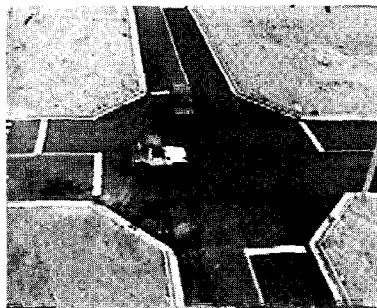


그림 13. 획득한 영상의 예.

Fig. 13. An example of acquired image.

줄이기 위해 하나의 샘플에 대해 3번의 입력 자료를 추출하였다. 그러므로 한번에 학습시킨 자료수의 개수는 108개 (36×3)이다. 이 자료수를 30,000번 학습시켰으며 에러가 잘 줄지 않는 경우 alpha와 beta 값을 변형시키고 각 무게들의 랜덤 시드를 변형시켜 학습시켰다. 에러는 각 출력 노드에 신경망으로 계산되어져 나온 값과 실제 목표값(입력한 샘플을 대표하는 노드만 '1' 나머지는 '0')의 제곱 합으로 계산하였다.

3. 영상분석 처리

카메라에서 출력되는 NTSC(National Television System Committee)로 구성된 아날로그 영상신호는 KS0127 디코더를 거치면서 디지털 신호로 변환하여 RGB 565포맷을 가진 칼라 영상신호를 얻는다. 이렇게 얻어진 RGB신호중 Green 값만을 추출하여 160×120 크기의 Gray 영상으로 변환 후 SDRAM에 저장된다[7-9].

이렇게 저장된 영상은 그림 12와 같은 사고검지 영상처리 과정을 거치게 된다.

STEP 1

사고 영상분석을 위한 영상을 입력 받는다. 사고 영상은 실제로 취득하기 어려우므로 그림 13과 같이 가상적인 사고 발생 장면을 만들고 이를 기반으로 사고 시점 분석을 수행하였다.

STEP 2

전체 화면을 영상처리하기에는 하드웨어 성능에 지장을

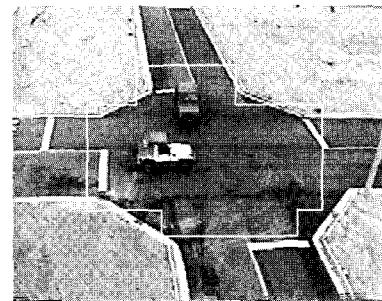


그림 14. 처리 영역의 지정.

Fig. 14. Appointment of processing region.

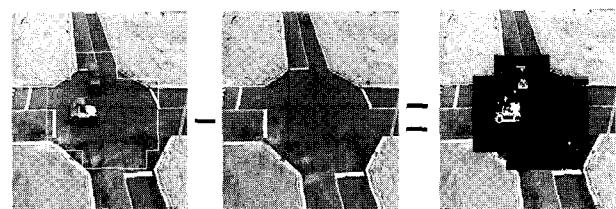


그림 15. 차 영상 생성결과.

Fig. 15. Result of subtracted image.

줄 수 있으므로 그림 14와 같이 관심 있는 부분만 선택하여 영상처리 영역을 지정한다.

STEP 3

교차로에 차량이 없는 영상을 참조영상으로 지정한 후 이를 주기적으로 갱신한다. 실제적으로 신호가 바뀌는 순간에 차량이 교차로 내에 없는 순간이 발생하므로 실질적으로 타당성이 있다고 고려된다. 또한 갱신은 3시간에 1번만 이루어져도 영상 처리에 큰 문제는 없는 것으로 고려된다. 구해진 참조영상과 입력된 영상과의 차영상을 그림 15와 같이 만들어 영상처리의 기본 전처리를 수행한다.

차영상 생성은 다음과 같다. (B : Background, S_2 : Next Image)

$$S_2(x, y) = S_1(x, y) - B(x, y)$$

$$S_3(x, y) = 1 \ll S_2(x, y) \gg T_1$$

$0 \ll \text{Otherwise}$

STEP 4

입력 영상과 참조영상과의 차영상을 통해 차량만을 추출해낸다. 만일 참조영상이 기상의 상태에 따라 완벽하게 입력영상과 배경이 일치할 수 없는 경우가 발생한다. 이러한 경우 산발적인 잡음이 생기게 되는데 이를 효과적으로 제거하기 위해 Morphology 기법인 침식연산(Erosion)과 팽창연산(Dilation)을 연속적으로 수행하여 그림 16과 같이 잡음이 없는 깨끗한 영상을 구한다.

STEP 5

차량의 특징점을 파악하여 차량의 외곽과 중심점을 구해낸다. 차량의 색이 도로면과 화면히 구별되면 문제가 없으나, 도로면과 유사한 색의 차량 진입시 차량의 앞부분과 뒷부분만이 측정된다. 따라서 텁색영역내의 차량에 대해 Edge

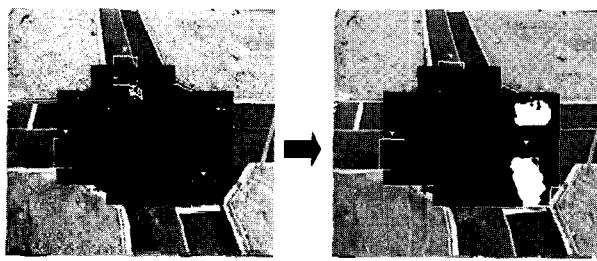


그림 16. Morphology 기법 적용 후 결과 영상.

Fig. 16. A result image implemented on the morphological method.

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

그림 17. X축, Y축 경계를 추출하기 위한 소벨 연산자.

Fig. 17. Sobel operator for extracting edges of X, Y.



그림 18. 차량종류에 따른 차량외곽 사각형과 차량중심.

Fig. 18. Contour rectangle and center of vehicle by vehicle types.

를 추출한다. Edge추출에는 기본적인 Mask의 하나인 Sobel Mask를 X축과 Y축으로 2회 적용한다. Sobel Mask는 (1)과 같이 표현할 수 있으며 X축과 Y축의 Mask는 그림 17과 같이 표현된다. 연산된 결과값이 +로 나오면 상승 에지(=상승부)로 규정하고, -로 나오면 하강 에지로 규정하며, x축과 y축을 기준으로 하여 좌표값이 가장 큰 상승 에지점과 가장 작은 하강 에지점과 연결하여 그림 18과 같이 차량의 외곽을 나타내는 사각형을 만들고, 차량의 중심점을 파악한다.

$$\begin{aligned} \Delta_x f(i, j) &= [f(i+1, j-1) + 2f(i+1, j) + f(i+1, j+1)] \\ &\quad - [f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1)] \quad (1) \end{aligned}$$

STEP 6

교차로에서 차량의 진입 시작구간에 차량 추적을 위한 검지 트랙을 그림 19와 같이 설치한다. 차량이 최초 검지트랙에 들어서면 차량의 이동 벡터에 따라 차량 검지 트랙을 이동하여 차량을 추적해간다.

차량 검지 트랙은 다음과 같은 제한 조건을 가진다.

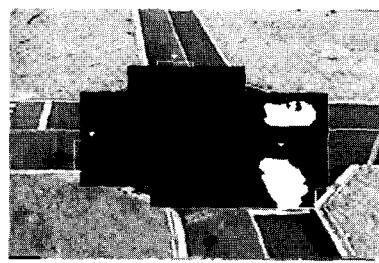


그림 19. 차량 진입구간에 설치된 4개의 차량 검지 영역.

Fig. 19. 4 regions of perceived vehicle installed in entry section.

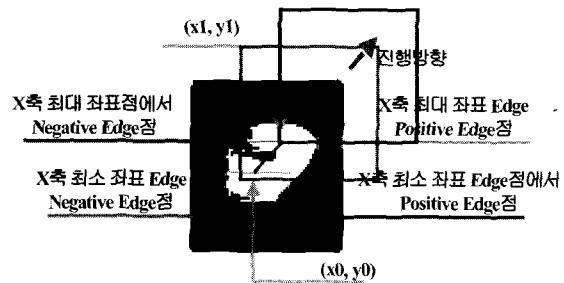


그림 20. 영상처리 속도에 따른 차량 추적 영역 변경.

Fig. 20. Zone change for vehicle detection by image processing speed.

- 차량 검지 영역의 크기가 도로 폭을 넘지 않아야 한다. (30pixel*30pixel)
- (반대편 차선의 차량과 겹쳐 오류가 발생할 요인이 크다.)
- 차량 검지 영역이 차량보다 작을 경우 진행하는 차량의 앞부분을 추적한다.
- 연산 속도가 느려 초당 처리하는 프레임수가 낮아지면 차량 추적시 잊어버릴 확률이 높으므로 연산속도에 따라 다른 임계치를 설정한다.

- 차량추적에 실패하면 검지 영역은 초기화 된다.

STEP 7

최초 검지 구역에 차량이 추적되면 차량의 이동 방향 벡터를 분석하여 차량 예상 이동 지점에 새로운 검지 영역을 설정한다.

검지영역은 영상 처리 속도에 따라 예상 검지 영역의 위치를 조절한다. 만일 새로운 검지 영역이 이전 프레임보다 5픽셀이 이동한 지역에 설치된다면 20프레임의 연산 속에서 는 추적에 문제가 없으나 만일 연산량의 증가로 인해 10프레임으로 연산속도가 떨어진다면 고속으로 진행하는 차량의 추적에 실패 할 수 있다. 따라서 (2)를 이용하여 다음 프레임의 새 검지 구역의 중심점을 계산한다.

$$x(\text{next frame}) = (x_1 - x_0)/k + x_1, \quad y(\text{next frame}) = (y_1 - y_0)/k + y_1 \quad (2)$$

k:연산처리 속도에 따른 임계치(25frame 연산시 0.5가 적합)

그림 20과 같이 차량이동 벡터는 x축 최대 상승 에지점에서의 하강 에지를 구하고, 최소 하강 에지에서 상승 에지를 구해 선분을 생성하고, 생성된 선분의 중심점을 연결하여 그 중심점을 연결하면 이동 방향 벡터를

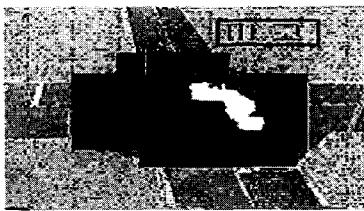


그림 21. 최종 사고영상 판단 1.

Fig. 21. Final decision of accident image 1.

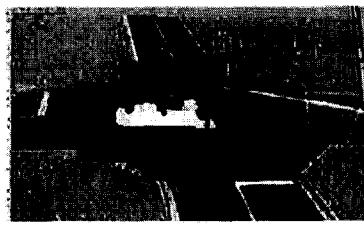


그림 22. 최종 사고영상 판단 2.

Fig. 22. Final decision of accident image 2.

구할 수 있다. 이동방향 벡터를 통해 생성된 이동 방향은 10도 단위로 36방향으로 구성되고, 1 프레임에 5도 이상 차이가 날 경우는 벡터 계산의 오류로 판단하여 이전 프레임에서 계산된 벡터를 사용하여 새로운 검지 영역을 설정한다 [6,10].

STEP 8

차량 충돌 검사는 다음과 같은 2단계의 절차를 거쳐 수행한다.

① 1차 차량 사고 판단 조건

- 두대의 차량이 접촉하면 차량 추적영역이 근접한다.
- Dilation처리에 의해 차량의 연결된 이미지가 얻어지면 Edge를 이용한 차량의 외곽선이 겹친다.

② 2차 차량 사고 판단 조건

- 1차 충돌 조건을 100프레임(약 5초)간 유지하면 사고로 판단 (1차 사고 판단 조건만 적용할 경우 차량이 많이 존재할 때에는 근접 주행을 사고로 오인할 수 있다.)

차량 추적이 시작되면 차량 경계를 나타내는 사각형과 화살표가 이동하게 된다. 차량이 검지 영역보다 큰 경우에는 차량의 전방을 추적할 수 있도록 하였다. 차량의 외곽선 영역이 겹치게 되어 차량의 외곽선이 차량의 전방으로 가려는 성질 때문에 2개의 차량 검지 사각형의 중첩 현상이 발생한다.

탐색 영역 사각형의 좌측 상단을 $xb1, yb1$ 우측 하단을 $xb2, yb2$ 라 하면 서로 다른 차량의 외곽선간의 충돌을 (3)과 같이 정의할 수 있다.

$$(xa1 < xb2) \& (xb1 < xa2) \& (ya1 < yb2) \& (yb1 < ya2) == 1 \quad (3)$$

교차로내 검출된 차량이 N개라면 충돌 검사해야 할 횟수는 (4)와 같이 정의 할 수 있다.

$$\text{충돌 검사 횟수} = 1 / 2 \times N \times (N - 1) \quad (4)$$

(N: 트래킹된 차량 수)

차량의 충돌 여부는 각각의 트래킹된 차량이 알고리즘 상에서 구별되어 차량 외곽선을 생성할 수 있다. 사각형이 최초 Interlace 될 때를 1차 사고 시점으로 인식하고, 지속적으로 5초간 유지될 경우 최종 충돌로 인식한다. 이는 차량의 근접 주행으로 인한 오류를 방지하기 위한 수단으로 적용하였다. 그림 21은 오른쪽에서 진입한 차량과 아래쪽에서 진입한 차량의 서로 다른 외곽선이 충돌하는 사고 발생 영상을 나타낸 것이고, 그림 22는 좌측에서 진입한 차량과 아래쪽에서 진입한 차량의 충돌을 나타낸다. 최종 충돌로 인식되면 화면 상단에 사고라는 메시지를 표시한다.

V. 사고영상 획득

CCD카메라로부터 영상처리보드로 들어오는 RGB RAW 데이터영상은 160 * 120 사이즈 RGB Color 565포맷 영상이다. 이러한 Raw영상은 그림 23처럼 SDRAM의 시작번지부터 차례로 최대 100프레임까지 순차적으로 저장이 된다.

그리고 만약 101개째 프레임을 저장하려고 한다면 다시 SDRAM의 시작번지로 돌아가 기존에 저장되었던 영상위에 덮어서 저장하게 된다. 마치 이것은 자료구조 알고리즘에 나오는 순환큐(Circular Que)의 구조와 거의 흡사하다. 이런 방식으로 항상 최근 영상데이터 100프레임을 SDRAM에 저장하고 있다가 사고신호를 받으면 즉시 SDRAM 저장되었던 사고 영상을 교통센터로 전송하고 Smart Media Card에 저장하게 된다. 이때 데이터량을 줄이고 전송속도를 빨리하기 위해 JPEG알고리즘을 이용하여 압축을 실시하게 된다.

그런데 여기서 그냥 사고영상을 처음부터 전송하게 되면 교통센터에서는 시간적으로 순차적인 사고영상 데이터를 받을 수 없다. 그 이유는 SDRAM에 데이터를 저장할 때 오버플로 발생을 막기 위해서 처음번지부터 순회하여 다시 저장하기 때문에 순차성(Sequence)을 갖는 시간파는 순서가 어긋나기 때문이다. 그래서 영상을 전송하기 전에 먼저 사고영상 데이터를 시간상의 오름차순으로 정렬을 해야 한다. 사고영상을 시간순으로 정렬하려면 영상 그 자체를 정렬하게 되면 데이터가 크기 때문에 오버헤드가 발생하여 시간이 많이 소요된다. 그래서 사고발생 신호가 들어오면 SDRAM상의 주소 인덱스를 기록해두었다가 영상을 SDRAM에서 다시 읽을 때 이전에 기록해 두었던 인덱스를 참조하는 방법을 사용하였다. 이 방법은 평상시의 영상이 항상 SDRAM에 저장되다가 사고신호가 발생하면 가장 최근 영상 50프레임이 사고영상이 되고 그 이후에 들어오는 50프레임의 영상은 사고 후영상이 된다. 그래서 다시 읽어올 때 이전에 기록해 두었던 SDRAM의 인덱스보다 작은 번지의 영상에서 50프레임의 첫 번째 번지를 계산하고 순차적으로 읽으면서 사고 전 영상 50프레임을 그림 24처럼 JPEG으로 하나씩 압축하여 SDRAM의 시작번지부터 하나씩 저장해 둔다. 그 이후 사고 전 영상 50프레임이 압축 및 저장이 끝나면 다시 인덱스 주소부터 시작하여 사고 후 영상 50프레임을 앞의 과정과 똑같이 반복하여 이전에 기록된 사고 전 압축영상 이후부터 순서대로 저장된다.

이런 식으로 사고영상이 정렬되면 영상을 전송한다. 그러나 사고영상만 전송하게 된다면 교통사고의 발생시점을 알

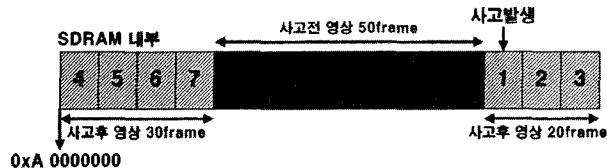


그림 23. 사고영상이 정렬이 안된 SDRAM내부.

Fig. 23. Inside SDRAM of nonarranged accident image.

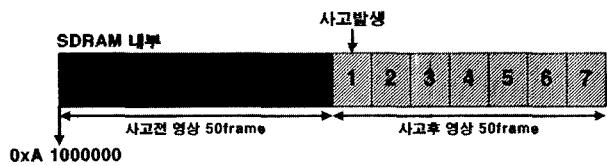


그림 24. 사고영상이 압축 및 정렬된 SDRAM내부.

Fig. 24. Inside SDRAM of compressed and arranged accident image.

수는 있으나 사고 후에 교통사고의 잘못을 가릴 수 있는 증거 자료로는 불충분하다. 만약 교차로 주변의 신호등 신호의 상태를 보낼 수만 있다면 완벽하지는 않지만 객관적인 자료로 충분히 활용 될 수 있다. 그래서 실제 교차로에 설치된 신호등과 연결될 수 있는 장치는 아니지만 영상처리 보드내의 포트에서 신호상태를 받을 수 있도록 가상의 신호등을 제작하게 되었다. 가상신호등은 실제 교차로처럼 빨강, 페랑, 좌회전, 노랑색의 LED신호를 주기적으로 발생하여 신호상태를 영상처리보드의 입력포트로 보낸다. 이렇게 입력된 신호등 신호는 사고영상을 보내기 전에 영상처리보드에서 신호등 상태를 저장해 두었다가 각각의 사고영상 프레임 앞에 신호상태에 대한 헤더정보를 먼저 보내고 그 다음에 사고영상을 전송한다. 이러한 전송방식은 Smart Media Card에도 똑같은 형태로 적용된다.

VI. 실험 및 결과

음향에 의한 사고검지 판별 실험은 5가지 종류의 기본 사고음을 변환하여 20종류의 사고음에 대해서 표 2와 같은 사고 음향 인식 결과를 얻었다.

비교적 음량이 높은 스키드음이나 차량 충돌음일 경우 더욱 높은 사고 인식률을 나타내며 전체적으로 양호한 결과를 얻을 수 있었다. 그러나 실험한 사고 음향의 경우 음향의 종류도 다양하지 않고 실제 교차로에서 실험을 한 것이 아니다. 그래서 더욱 다양한 패턴의 사고음향과 교차로 주변 잡음이 섞인 사고음향에 대한 더 많은 실험이 필요하다고 판단된다. 만약 실제 교차로에서 실험을 할 경우 주변 잡음이 사고음향 인식에 영향을 미쳐 표 2와 같이 좋은 결과를 얻을 수 없을 것으로 생각되며 향후 주변잡음제거에 대한 알고리즘 개선 방향에 대한 연구가 필요할 것으로 생각된다.

영상에 의한 사고 인식률 실험을 위해 가상으로 사고 영상을 취득하여 실험을 하였다. 실험 결과는 가상 실험 영상이 비교적 단순하고 차량속도가 낮아 표 3과 같이 우수한

표 2. 사고 음향 인식 결과.

Table 2. A result of accident sound awareness.

음향종류	사고 인식률	비고
교차로 잡음	82%	사고음향은 50가지 종류의 변형된 음원으로 구성
경적소리	85%	
스키드음	97%	
충돌음(금속)	99%	
충돌음(유리)	98%	

표 3. 사고 영상 인식 결과.

Table 3. A result of accident image awareness.

오류종류	사고 인식률	비고
1종 오류 (사고 발생시 사고로 인식 못함)	99%	오후 5~8시까지 촬영한 도로 영상. 2종 오류보다 1종 오류의 문제가 더 심각한 오류임
2종 오류 (사고 발생시 않을경우 사고로 인식)	95%	
전체	97%	

결론을 얻을 수 있었다. 그러나 본 논문에서는 한정된 데이터를 기준으로 하였기 때문에 높은 신뢰도를 보였으나 실제로 사고 발생시 다양한 형태의 사고영상이 발생하므로 지속적인 보완작업이 필요 할 것으로 생각된다.

VII. 결론

교차로내에서 교통사고 발생시 사고음향과 영상을 통하여 사고를 검지하고 사고영상을 교통센터로 전송하고 Smart Media Card에 저장하는 교통사고 검지 시스템을 개발 하였다. 이 시스템의 교차로내 교통사고 검지 판별 기능은 아직 완벽하지는 않지만 앞으로 무인 자동화 교통사고검지 시스템에 대한 가능성을 보여주었다고 생각 되어진다.

향후 이러한 교통사고 검지시스템은 유고관리 및 사고 DB구축, 사고 인지시 자동으로 긴급구난, 터널 및 악천 후시 사고검지 등 여러 방면에서 활용이 가능하다. 그 결과 대형사고로 인한 사망이나, 야간 추돌 사고와 같은 증거가 불충분한 교통사고의 중요한 단서가 되어 경찰 및 보험사의 사고처리에 있어 시간 및 비용단축에 크게 기여할 것이며 신속한 사고처리로 교통흐름을 원활히 하는 기대효과를 예상할 수 있다.

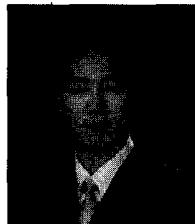
참고문헌

- [1] TMS320C6211/6711 PERIPHERALS REFERENCE GUIDE.
- [2] TMS320C6211/6711 PROGRAMMER'S GUIDE.
- [3] TMS320C6211/6711 OPTIMIZING C COMPILER USER'S GUIDE.
- [4] 홍상기, “TMS320C6201를 이용한 고속 웨이브릿 영상 압축 시스템 개발”, 부산대학교 석사학위 논문, 1999. 2.
- [5] 조경석, “TMS320C6201을 이용한 H.263 부호화기의 구현 및 동영상 처리를 위한 개선 방안”, 한국과학기술원 석사학위 논문, 2000. 2.
- [6] 이상길 번역, “C언어에 의한 디지털 영상처리”, Ohm

- 사], 1999.
- [7] Socott E Umbaugh, "Computer vision and image processing", *Prentice-hall International*, 1998.
- [8] Milan Sonka Vaclav Hlavac Roger, "Image processing analysis, and machine vision", *PWS publishing*, 1999.
- [9] Randy Crane, "Simplified approach to image processing", *Prentice-hall International*, 1996.
- [10] 장동혁, "Visual C++을 이용한 digital image processing", PC어드밴스, 2000.

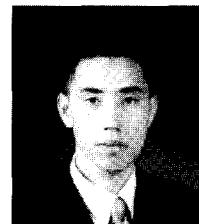
- [11] 한국과학 기술원, 아주대학교, 중앙대학교, "고성능 DSP Core 개발에 관한 연구(2차년도 연차보고서)", 1997. 6.
- [12] 오현호, 김성윤, 윤대희, "범용 DSP칩에서의 구현을 위한 고속 DCT 알고리듬의 성능 비교분석", 제 10회 신호처리학회 논문집, 1997.
- [13] "CCTV SYSTEM GUIDE", 삼성항공산업주식회사, 1997.
- [14] "In-system programmable solution", Lattice, 2000.
- [15] "Technology innovations", Texas Instruments, Nov., 2000.

이 형 석



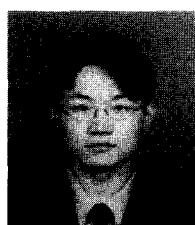
1979년 12월 1일생. 2003년 고려대 산업시스템정보공학과 졸업(공학사). 2003년~현재 동대학원 석사과정. 관심분야는 Computer Vision, Image Processing, ITS.

김 영 육



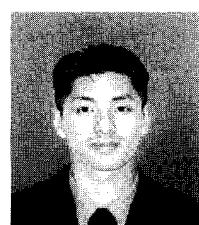
1975년 10월 10일생. 2002년 한국외국어대 산업공학과 졸업(공학사). 2002년~현재 고려대 산업시스템정보공학과 석사과정. 관심분야는 ITS, Embedded System, Robotics.

권 대 길



1975년 11월 26일생. 2001년 동의대 산업공학과 졸업(공학사). 2003년 고려대 산업시스템정보공학과 졸업(공학석사). 2003년~현재 한국전자부품 연구원 연구원. 관심분야는 Embedded System, Image Processing, Neural Network.

박 기 현



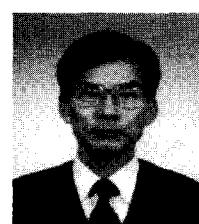
1973년 5월 3일생. 1999년 경희대 기계공학과 졸업(공학사). 2001년 고려대 산업시스템정보공학과 졸업(공학 석사). 2001년~현재 TI KOREA 연구원. 관심분야는 Image Processing, DSP, Embedded System.

이 경 복



1978년 10월 6일생. 2001년 남서울대 산업공학과 졸업(공학사). 2003년 고려대 산업시스템정보공학과 졸업(공학석사). 2003년~현재 고려대 산업시스템정보공학과 박사과정. 관심분야는 Robotics, Embedded System, Neural Network, DSP, 무인자동차.

한 민 흥



1942년 3월 2일생. 1964년 서울대 기계공학과 졸업(공학사). 1973년 University of Minnesota 산업공학과 졸업(공학석사). 1985년 Georgia Institute of Technology 산업공학과 졸업(공학박사). 1981년~1982년 숭실대학교 산업공학과 조교수. 1985년~1988년 Texas A&M University 산업공학과 조교수. 1988년~1990년 포항공과대학 산업공학과 부교수. 1991년~현재 고려대학교 산업시스템정보공학과 정교수. 관심분야는 지능형 자동차, 지능 로봇, ITS, 컴퓨터비전.