

## 고속 영상 검지기 시스템 개발에 관한 연구

### Study On Development of Fast Image Detector System

임 태 현, 이 종 민, 김 용 득  
아주대학교 컴퓨터 네트워크 연구실  
전화 : (031) 219-2372 / 팩스 : (031) 212-9531  
H.P 번호 : 011-9875-6825

Tae Hyun Lim, Jong Min Lee, Yong Deak Kim  
School of Electronics Engineering Ajou University  
E-mail : earthmountain@hanmail.net

#### Abstract

Nowadays image processing is very useful for some field of traffic applications. The one reason is we can construct the system in a low price, the other is the improvement of hardware processing power, it can be more fast to processing the data.

In this study, I propose the traffic monitoring system that implement on the embedded system environment. The whole system consists of two main part, one is host controller board, the other is image processing board. The part of host controller board take charge of control the total system, interface of external environment, and OSD(On screen display). The part of image processing board takes charge of image input and output using video encoder and decoder, image classification and memory control of using FPGA, control of mouse signal. And finally, for stable operation of host controller board, uC/OS-II operating system is ported on the board.

#### I. 서론

한정된 도로 및 시설자원에서 가장 극대화한 효율을 내기 위해서는 교통의 편중을 균등하게 하는 전문적이고 세밀한 교통 정보 전달 시스템의 구축과 함께 도로 시설의 파손 및 교통의 흐름을 방해하는 요소를 제거하는 정책이 필요하다. 이 때 도로의 상황을 파악하여 정보를 생성하는 것이 중요하며 영상을 이용한 차량 인식 기법이 필수적으로 요구된다.

현재 국내외에서 개발되어 실용화된 대부분의 차량 인식 시스템은 검지센서 방식으로 각종 검지센서들을 도로에 매설하거나 도로변에 설치하여 데이터를 수집한 후 이를 기반으로 차량을 검지하는 방법을 사용하고 있다. 이러한, 검지센서 방식인 경우에는 유도식 루프 검지센서, 초음파, 초단파, 광 검지센서 등을 사용하여 차량을 인식하는 방식으로서 보편적으로 많이 사용되고 있다. 하지만 이러한 방식은 검지센서의 유지 보

수에 필요한 비용과 인적 자원이 과다하고 여러 개의 검지센서를 포함하는 복잡한 현장 설비를 필요로 하는 단점을 가지고 있다.

하지만 영상을 이용한 차량 인식 시스템은 도로에 손상을 주지 않고서도 설치가 가능하다. 따라서 설치 시에는 교통 통제 등 불필요한 교통량 부하를 주지 않을 수 있고, 설치가 검지센서 방식에 비해 매우 간단하므로 설치비 또한 절감할 수 있으며, 유지 보수 또한 간편하며, 자체 설비의 가격 또한 저렴하다는 장점을 가지고 있다.

반면에 영상을 기반으로 한 차량 인식 시스템은 날씨, 그림자, 차량 배경 등의 환경적인 요인에 민감하게 반응하여 인식률이 크게 변화하므로 이를 보완해 주는 알고리즘의 개발이 필수적이다.

본 논문에서는 ARM7 마이크로컨트롤러와 FPGA를 이용하여 저가의 고성능 차량 인식 시스템의 구현을 제안하고자 한다.

호스트 컨트롤러 보드의 안정성을 높이고 어플리케이션의 개발을 편리하게 하기 위해서 uC/OS-II를 포팅 하였다.

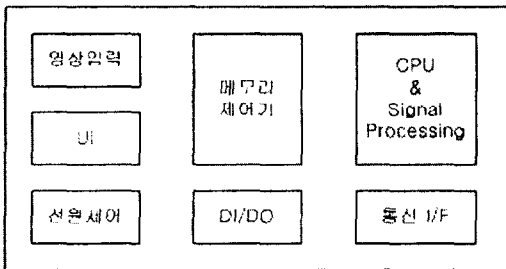
본 논문의 2장에서는 시스템의 하드웨어 구성을 살펴볼 것이며, 3장에서는 시스템의 Firmware 구성, 4장에서는 영상처리 알고리즘의 적용과정을 살펴 볼 것이다. 5장 결론에서는 개발된 시스템의 테스트 결과를 살펴 보도록 할 것이다.

## II. 시스템 하드웨어 구성

영상 검지기는 실시간 화면으로 검지할 수 있고, 도로와의 비접촉식으로 설치가 용이할 뿐 아니라 다양한 교통변수 계측이 가능하므로 적은 비용으로 많은 기능을 수행할 수 있는 시스템이다.

영상검지기는 교통 데이터 수집을 통한 정확한 교통상황의 파악과 이에 대한 대처능력이 필요하다. 그리고 본 영상검지기는 실제 환경에서 운영되어야 하므로 성능뿐만 아니라 안정성에도 중점을 두어 설계하였다.

본 논문에서 제작하려고 하는 하드웨어의 기본 블록도는 그림 1 과 같다.



<그림 1> 전체 하드웨어 블록도

하드웨어는 두장의 PCB로 구성되어 있으며 전체 시스템을 제어하는 호스트 컨트롤 보드(이하 HST 보드)와 영상처리 알고리즘이 포팅 되어 입력된 영상을 처리하는 영상처리 보드(이하 IMD 보드)로 나뉜다. 이는 영상처리 알고리즘의 처리와 전체 시스템 제어를 분리시킴으로서 CPU의 과부하로 인한 오동작을 막기 위함이며, 전체적인 시스템의 제어를 하나의 CPU에 맡김으로서 독립적으로 시스템의 제어를 수행하여 안정적인 시스템 구축에 목적이 있다.

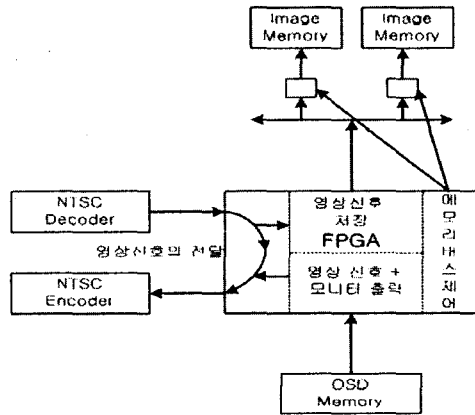
두 보드간의 통신은 각 CPU 간의 RS-232 통신을 이용하는 경로와 HST 보드의 CPU와 IMD 보드의 FPGA의 Data Bus와 Control Bus를 통한 경로를 이용하여 통신이 가능하다.

외부와의 인터페이스는 HST 보드의 RS-232를 이용한 PC와의 통신 경로와 IMD 보드의 마우스를 이용한 인터페이스가 있다.

본 시스템에서 사용되는 FPGA는 Spartan-II를 사용하였으며 역할은 크게 네 가지로 나뉜다.

- 영상 신호의 입력 및 출력
- 영상 신호의 메모리 저장
- 메모리 버스 제어
- On Screen Display

영상신호의 입력과 출력은 그림 2에서와 같이 영상신호의 입력은 NTSC Decoder에 의해서 이루어진다. 디코더를 거친 영상신호는 디코더의 클럭에 맞춰 입력이 FPGA에 들어가게 된다. FPGA는 항상 모니터의 입력을 유지해 주어야 하므로 입력된 신호를 곧바로 NTSC Encoder로 보내주게 되고 이는 모니터를 통해 출력이 나가게 된다.



<그림 2> FPGA 기능도

그리고 영상 신호의 메모리 저장은 입력된 영상 신호는 NTSC Decoder를 통해 바로 나가는 동시에 CPU에 의해 영상처리 과정을 거치기 위해서 Image Memory에 저장이 되어야 한다. 하나의 영상입력의 시작은 NTSC Decoder의 VRESET 신호부터이며, 입력되는 영상신호는 매 클럭마다 1 byte의 데이터보내기 때문에 이를 메모리에 저장하기 위해서는 각 데이터마다 어드레스를 만들어 주어야 한다. 따라서 FPGA는 VRESET 신호가 입력되는 순간에 메모리 주소를 초기화 하며, 입력되는 메모리 버스를 전환하여 전과 다른 영상 메모리에 저장을 시작한다. 입력이 시작되는 순간부터 매 클럭마다 어드레스를 생성하며 입력된 데이터를 메모리에 저장한다.

FPGA의 메모리 버스 제어부분은 하나의 영상을 처리하기 위해서는 320\*240 byte의 정보를 처리해야만 한다. 이러한 데이터가 초당 60 프레임이 입력되기 때문에 1초당 처리해야 되는 데이터의 양은 4500Kbyte (=320\*240\*60)이다. 이런 데이터를 모두 실시간에 처리한다는 것은 위험부담이 있으므로 CPU의 처리 시간을 좀더 안정적으로 가져가기 위해서 이미지 저장 메모리를 2개 설치하게 되었다. 따라서 1번 메모리에 FPGA

에서 영상 신호를 저장하고 있는 동안 CPU는 2번 메모리에 저장되어 있는 영상 신호를 처리하고, 반대로 FPGA가 2번 메모리에 영상 신호를 저장하고 있는 동안 CPU는 1번 메모리에 저장되어 있는 데이터를 처리하게 되면 보다 안정적인 시스템을 구축할 수가 있게 된다.

또한 FPGA는 On Screen Display 부분을 담당한다. 본 시스템의 특성상 외부에 대한 인터페이스는 RS-232를 통해 PC와 통신을 하거나 PS2 마우스를 통해 직접 시스템과 통신을 하는 방법이다. 전자의 경우는 실험실 내에서 개발 당시 사용하게 될 통신 방법이며, 후자의 경우 사용자는 시스템에 연결되어 있는 모니터를 통해서 시스템과 통신을 하는 방법이며, 이는 실제 필드에서 사용될 방법이다. 따라서 모니터를 통해서 시스템이 사용자에게 알려줄 사항을 표시하여야만 한다. 이를 위해서 본 시스템에서는 FPGA를 이용하여 출력 영상 신호를 적절히 가공하여 모니터를 통해 사용자와 통신할 수 있는 수단을 제공한다. 모니터에 글이나 그림을 인위적으로 표시하기 위해서는 각 글이나 그림의 좌표를 미리 파악하여 그 위치에 점을 찍음으로서 표시를 하게 된다. 본 시스템에서는 OSD 메모리를 따로 가지고 있다. OSD 메모리는 FPGA 영상 출력 신호에 OSD 메모리의 내용을 OR 연산을 통해서 중첩 시키는 역할을 함으로서 모니터에 각종 정보를 표시하게 된다.

### III. 시스템의 Firmware 구성

본 시스템에서는 컨트롤러 보드의 안정성을 높이고 어플리케이션의 개발을 편리하게 하기 위해서 uC/OS-II를 포팅하였다. uC/OS-II의 포팅은 HST 보드에 하게 된다. HST 보드는 전체 시스템을 제어하는 역할을 하므로 자체의 안정성이 매우 중요하다. 또한 시스템의 업그레이드 시에 편리한 인터페이스를 제공하기 위해서도 RTOS가 포팅 될 필요가 있다.

uC/OS-II는 매우 뛰어난 성능을 제공하는 RTOS는 아니지만 소스가 공개되어 있으며 구조가 간단하여 짧은 시간에 기능을 파악하기가 용이하며, 현재 사용되는 RTOS들이 제공하는 대부분의 기능(선점형 커널, 세마포 메시지 큐등의 다양한 프로세스간의 통신과 멀티태스킹 가능)들을 모두 가지고 있어 본 시스템에 적용하는데 무리가 없다.

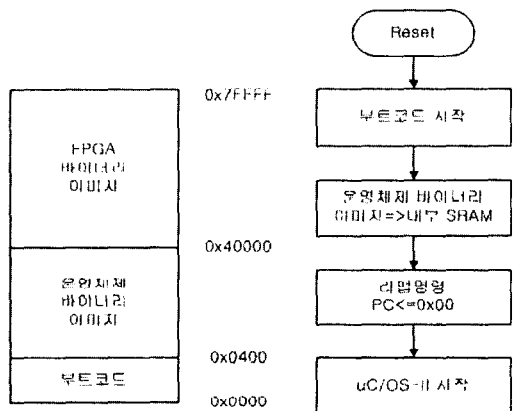
uC/OS-II는 실제로 운영체제를 포팅하기 위해서 CPU의 독립적인 부분을 제외한 나머지 부분들, 즉 하드웨어에 비독립적인 스택 생성 함수, 컨택스트 스위칭을 위한 함수, 그리고 CPU의 10ms 타이머가 필요하다.

외부 플래시 메모리에는 매우 중요한 세 가지 정보가 담겨져 있다. 처음으로 CPU가 동작을 시작할 때 필요한 부트 코드와 근본적으로 동작을 시켜야할 운영

체제의 바이너리 이미지, 그리고 IMD 보드에 다운로드할 FPGA의 바이너리 이미지에 대한 데이터이다. 이러한 데이터들은 그림 15에서와 같이 저장되어 있는데 부트 코드를 제외하고는 모두 갱신이 가능하다.

부트 코드의 역할은 CPU의 부팅 이후에 내부 SRAM으로 플래시에 있는 운영체제의 바이너리 이미지, 또는 PC로부터 RS-232를 이용하여 다운로드 되는 이미지를 복사해 넣고, 리셋 명령을 수행하여 이들 다운로드한 이미지로 제어를 넘기게 된다.

### IV. 영상처리 알고리즘 적용과정



외부 플래시 메모리의 구조 부트코드의 순서도

그림 3 플래시 메모리의 구조와 부팅과정

영상 내 특정 위치에서의 검지를 위해 영상내에 가상의 검지선을 정하게 되는 데 이렇게 설정된 가상의 검지선내에 차량이 들어왔을 경우에 차량의 검지선 인식하게 된다. 즉 매 프레임마다 검지선의 화소값 분포를 감시하여 차량의 검지선에 진입 유무에 대해 분석하므로써 검지하는 것이다. 차량이 검지선을 통과하지 않았을 경우에는 노면에 대한 즉 배경의 화소값이 나타나게 되고, 차량이 검지선에 진입했을 경우에는 화소값이 변하게 된다. 이 때, 검지선의 화소값들은 통과 차량의 색상에 따라 노면 화소값들보다 높아질 수도 있고 낮아질 수도 있다. 이 두가지 경우에 모두 대비하기 위하여 차량 진입시 화소값들의 높아짐에 대한 최대 임계값과 낮아짐에 대한 최소 임계값을 정한다. 이 임계값의 범위는 차량이 통과하지 않았을 때 검지선의 화소값들의 분포 범위에 의하여 얻을 수 있고, 검지선에 추출된 화소값들이 이 임계값 범위밖에 있을 때 차량의 진입을 인식할 수 있다.

각 검지선에서 추출된 화소값들이 임계값 범위 안에 있는 지 판단하는 방법으로는 검지선의 화소값들을 임계값과 차례대로 비교하는 방법과 그 평균값을 임계값

과 비교하는 방법이 있다.

환경에 따른 배경영상 즉, 도로의 화소값의 변화에 대처하기 위하여 신경망 이론을 이용한 학습방법을 이용한다. 즉 검지의 기준이 될 화소값과 임계값을 입력되고 있는 곳의 환경변화에 맞게 학습하여 이용함으로써 실시간으로 언제, 어디서든지 기능을 발휘할 수 있도록 하는 것이다.

영상 검지기 알고리즘은 입력영상을 32화소의 일차원 배열로 정규화한다. 정규화된 32화소중 임의로 설정된 임계값을 넘는 화소수를 계수하여 이 값이 25%를 초과하면 라인이 점유되었다고 판단한다.(차량 흐름의 검지로 판단한다.) 임계값 및 점유율은 고정된 값이 아니며 수시로 값을 변경하여 지정할 수 있게 하여 배경 영상의 변화에 대한 적응력을 높인다. 영상이 입력되면 8장의 영상에 대해서 정규화가 이루어지고 매 20분마다 기준 영상을 갱신한다

### V. 결론 및 추후 연구

본 논문에서 제안한 시스템을 검증하기 위하여 실제로 도로 영상을 녹화한 비디오를 영상입력으로 하여 테스트를 하였다. 테스트의 조건은 320\*240의 영상 입력을 초당 60프레임을 받아 들였고, 한 프레임의 영상에서 8군데의 검지영역을 설정하였다. 그 결과 차량 검지에 대해서는 다음과 같은 검지 성능을 보여주었다. (테스트는 10일 동안 주간 정오 와 야간 자정에 특정 도로를 통과하는 차량 100대를 대상으로 실험하였다)

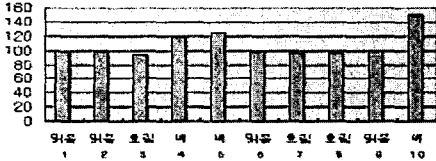


그림 4 날짜별 주간 검지된 차량수

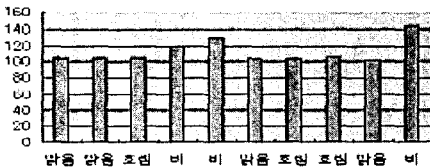


그림 5 날짜별 야간에 검지된 차량수

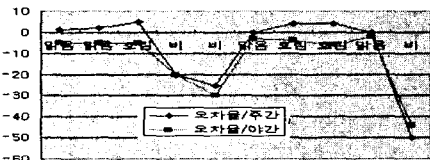


그림 6 주야간 기후상태에 따른 오차율

본 논문에서는 고성능의 안정적인 영상 처리 시스템을 구현하기 위한 설계를 제안하였다. 높은 성능을 얻기 위하여 하나의 CPU에 영상 처리 알고리즘 연산을 전담시켰으며 이를 통해서 320\*240 픽셀의 영상 데이터를 1초에 30프레임 이상을 처리할 수 있는 처리 속도를 가지게 되었다. 하지만 이러한 데이터를 영상 디코더로부터 직접 받을 때는 한 픽셀의 데이터가 0.035us마다 입력이 되기 때문에 이러한 신호를 영상 디코더로부터 직접 받아 실시간으로 처리한다는 것은 데이터를 놓칠 수 있는 가능성을 내포하게 된다. 따라서 본 시스템은 영상 디코더로부터의 신호를 FPGA를 통해서 따로 확보해 둔 영상 데이터 메모리에 저장해 두고 CPU는 영상 디코더로부터의 신호가 아닌 메모리의 데이터를 처리하게 된다. 또한 영상 데이터 메모리를 하나 더 이용하여 FPGA는 이 메모리들을 번갈아 접근하여 하나의 메모리에는 Even 필드의 데이터, 다른 하나의 메모리에는 Odd 필드의 데이터만을 저장하게 되고, 영상 처리 CPU는 FPGA가 접근하지 않는 영상 데이터 메모리에 접근하여 처리를 하게 됨으로써, 영상 데이터 처리에 있어서 보다 안정적인 동작을 보장 받을 수 있게 된다. 또한 더욱 복잡한 영상처리 알고리즘을 필요로 할 경우에는 FPGA를 통하여 그 영상 데이터의 전처리 알고리즘을 수행할 수 있으므로 시스템의 성능을 또 다시 비약적으로 발전시킬 수 있는 여지를 가질 수 있다.

### 참고문헌(또는 Reference)

- [1] ARM, "ARM Architecture Reference Manual", February 2000
- [2] ARM, "ARM7TDMI Data Sheet", Aug 1995
- [3] Jean J. Labrosse, "MicroC/OS-II The Real-Time Kernel", R&D Books, 1999
- [4] Ed Sutter, "Embedded Systems Firmware Demystified", CMP Book, 2002
- [5] ATMEL, "AT91 ARM Thumb Microcontrollers Data Sheet", 2002
- [6] 김효준, "uC/OS-II EP7209(ARM7) 포팅 " 2001
- [7] Webber H C(ed), "Image processing and transputers", IOS Press, 1992
- [8] M.Schlett, "Trends in Embedded-Microprocessor Design", Computer, vol. 31, no. 8,pp.44-49, Aug,1998
- [9] S. C. Chan, H. O. Ngai, K. L. Ho "A programmable image processing system using FPGA"
- [10] ARM, "ARM Software Development Toolkit - Users Guide", February 2000
- [11] ARM, "ARM Software Development Toolkit - Reference Guide", February 2000