

# 계층화 모션 추정법과 병렬처리 기반의 차량 움직임 측정 알고리즘 개발 및 검증<sup>1)</sup>

강경훈, 심현진, 이은숙, 정성태, \*남궁문, \*\*김기정, 이상설  
원광대학교 전기전자 및 정보공학부, \*원광대학교 도시·토목환경공학부, \*\*명지대학교 SOC  
공학부

전화 : 063-850-6889 / 핸드폰 : 018-626-0125

## Design and Verification of Algorithms for the Motion Detection of Vehicles using Hierarchical Motion Estimation and Parallel Processing

Kyung-Hoon Kang, Hyun-Jin Shim, Eun-Sook Lee, Sung-Tae Jung, \*Kung-Moon Nam,  
\*\*Ki-Jung Kum, Sang-Seol Lee

Division of Electric Electronic and Information, Wonkwang University

\*Division of Architecture, Urban, Civil & Environmental Engineering Wonkwang University

\*\*Division of Civil and Transportation Engineering, MyongJi University

E-mail : pandol@wonkwang.ac.kr

### Abstract

This paper presents a new method for the motion detection of vehicles using hierarchical motion estimation and parallel processing. It captures the road image by using a CMOS sensor. It divides the captured image into small blocks and detects the motion of each block by using a block-matching method which is based on a hierarchical motion estimation and parallel processing for the real-time processing. The parallelism is achieved by using the pipeline and the data flow technique. The proposed method has been implemented with an embedded system. Experimental results show that the proposed method detects the motion of vehicles in real-time.

### I. 서론

자동차화의 급격한 진전과 이용률의 증가에 따라 자동차 이용의 효율성에 관한 사회적 관심은 매우 급격히 증가하고 있다. 특히, 도시 내 도로의 용량 부족이나 자동차 교통량 증가에 적절히 대응할 수 있는 도로 정비 등이 원활히 추진되지 못하고 있는 상황을 고려할 때 도로교통 문제를 개선할 수 있는 방안의 하나로 교통정보를 활용함으로써 도로교통 기능의 효율성을

높이는 방법에 대한 관심이 증가되고 있다.

이러한 배경에서 교통 관리 및 통제에 관한 관심이 증대되고 있다. 교통 관리 및 통제 기법은 현재의 교통 상태를 실시간으로 파악하여 이를 기초로 교통 수요의 조절과 시설 운영 능력의 향상을 통해 기존 시설의 효율성을 증대시키는 기법이다. 따라서 현재의 교통 상태를 나타내는 자료들을 실시간으로 획득하는 방법의 개발이 요구되고 있다.

교통 상태 정보의 추출기로는 루프검지기가 많이 사용되고 있는데, 도로 공사 및 파손에 의한 단선, 설치 시 도로 차단에 의한 교통 혼잡, 유지 보수의 어려움 등의 단점이 지적되고 있다. 이러한 문제를 극복하고 수집할 수 있는 교통 상태 정보의 질적 및 양적 다양화를 도모하기 위하여 루프 검지기를 대체할 수 있는 다양한 연구가 진행되고 있다[1,2,3]. 현재까지의 연구에서 고려된 검지기는 영상 검지기, 초음파 검지기, 초단파 검지기 등이 있으며 이중 영상 검지기가 가장 유력한 검지기로서 인식되고 있다.

영상을 이용한 교통 상태 정보 검지에 대한 다양한 기법들이 제안되었는데, 많은 시스템들은 PC를 기반으로 하고 값비싼 영상 캡처 보드를 이용하여 영상을 획득하며 일반적으로 소프트웨어적으로 영상 분석을 수행하고 있다. PC는 범용을 목적으로 하는 운영체제를 채택하고 있으므로 특수목적의 처리를 위해서는 고성능의 값비싼 영상처리 장치를 장착해야 하기 때문에 가격이 고가가 된다. 또한 규모도 PC라는 기본 플랫폼을 유지해야 하기 때문에 커지게 되어 실제 설치와 운용에 불편함을 겪게 된다. 그런데 영상 검지기의 영상 분석 부분을 하드웨어로 구현하면 기존의 PC기반의 영상검지기보다 간단하고 경제적으로 구성될 수 있다.

1) 본 연구는 한국과학재단 목적기초연구 (과제번호: 2000-2-31300-001-3) 지원으로 수행되었음

즉, VLSI칩을 이용하여 주행하는 자동차의 움직임에 대한 영상 처리를 수행할 수 있는 알고리즘을 구현함으로써 영상처리 보드의 기능을 대신할 수 있기 때문에 크기도 작아지게 되고, 가격도 저렴하게 된다.

따라서, 본 논문에서는 저가의 하드웨어로 구현하기에 적합하고 실시간으로 차량의 움직임을 측정할 수 있는 차량 움직임 측정 방법을 제안하고 이를 소형 내장형 시스템을 이용하여 구현하였다. 제안된 방법은 CMOS 카메라를 이용하여 획득한 영상을 작은 블록들로 분할한 다음에 블록 매칭을 이용하여 각 블록의 움직임을 계산한다. 그리고 움직임이 비슷한 블록들을 클러스터링함으로써 차량의 움직임을 측정한다. 본 논문에서는 실시간 동작을 위하여 블록 매칭 부분을 내장(embedded)형 시스템의 하드웨어상에서 병렬처리에 의한 계층화 모션 추정기법을 이용하여 구현하였다. 내장형 시스템의 FPGA(Field Programmable Gate Array) 상에서 동일한 처리능력을 갖고 여러 개가 병렬 동작되는 처리단위(Processing Element : PE)를 도입하였으며 각 PE는 병렬로 계층화 모션 추정을 실행하게 된다. 파이프라인 구조로 구성된 PE에 영상데이터가 연속적인 데이터흐름(data flow) 형태로 입력되고 이 때 해당 PE가 활성화 되어 효율적인 병렬 연산이 실행된다. 데이터흐름 형태로 주입되는 영상자료는 병렬연산을 위한 빈번한 메모리 접근을 줄이는 효과를 갖게 된다. 실험결과에 의하면 본 논문에서 제안한 시스템은 차량의 움직임을 실시간으로 측정할 수 있었다.

## II. 계층화 모션 추정과 병렬처리 기반 영상 검지

### 2.1 블록 매칭

본 논문에서 구현한 시스템은 CMOS이미지센서를 사용하여 320x240 크기의 영상을 초당 60 프레임의 속도로 입력 받는다. 입력된 영상은 버퍼를 통해 메모리에 저장된 다음에 32비트 버스를 통해 블록 매칭을 위한 PLD에 입력된다. 영상은 그림 1과 같이 여러 블록들로 분할된 다음 각 블록에 대하여 그 블록과 가장 유사한 블록이 이전 프레임 영상에서 어디에 위치하는가를 탐색하는데, 이를 블록 매칭이라 한다.

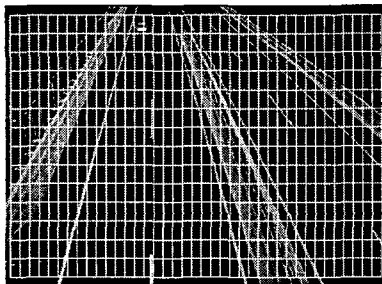


그림 3. 블록 분할

본 논문에서는 블록의 크기로 8x16을 사용하였고 각

블록과 가장 유사한 블록이 이전 영상에서 16x32 탐색 영역 안에 존재 한다고 가정하였다. 현재 프레임의 블록과 이전 프레임에서의 탐색 영역의 관계는 그림 2와 같다. 이와 같은 가정에서 320x240 영상에 설정할 수 있는 블록의 최대 크기는 그림 1에 나타나 있는 바와 같이 39x14개이다.

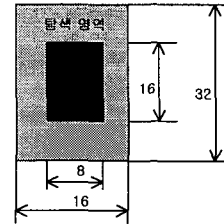


그림 4. 블록 설정

일치하는 블록을 찾기 위해서는 식 (1)과 같이 정의되는 휘도차의 합(SAD: Sum of Absolute Difference) 값을 이용한다. 여기에서  $L(i,j)$ 는 현재 프레임 블록 안의 픽셀들에 대한 휘도값을 나타내고,  $L''(i,j)$ 는 이전 프레임 블록 안의 픽셀들에 대한 휘도값을 나타낸다.  $M, N$ 은 블록의 가로방향과 세로방향의 픽셀 수를 나타낸다.

$$D(u, v) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |L''(i+u, j+v) - L(i, j)| \quad \text{식(1)}$$

블록 매칭 방법 중에서 가장 기본적인 방법은 전탐색 방법이다. 전탐색 방법은 정확도는 높지만 처리에 필요한 계산량이 많아서 실시간으로 동작할 수 있도록 구현하기에는 문제가 있다.

### 2.2 계층화 모션 추정

전탐색 기법은 계산량이 많기 때문에 탐색 공간을 줄여서 탐색하면서도 탐색 결과의 정확도는 많이 떨어지지 않는 여러 가지 방법들[4,5,6]이 제안되었다. 본 논문에서는 이들 방법 중에서 하드웨어에 의해 효율적으로 처리 가능한 방법으로 계층화 모션 추정 기법[6]을 채택하였다. 계층화 모션 추정 기법은 상위 계층에서 현재 블록과 탐색영역을 이완 샘플링하여 사용하는 방법으로 전탐색 기법에 비해 계산량이 현저히 줄일 수 있다.

본 논문에서는 두 단계 계층으로 나누어 상위 계층에서는 블록을 1/2 이완 샘플링을 하여 사용한다. 상위 계층에서 발견된 매칭 블록의 위치가  $(x, y)$ 이면 하위 계층에서는  $(x, y)$ 를 중심으로 3x3 영역을 탐색하여 매칭 블록을 찾는다. 그러나 계층화 모션 추정법을 사용한다 하더라도 순차적으로 데이터를 처리한다면 상당히 많은 계산이 필요하다. 순차적으로 처리할 경우에 본 논문에서 적용한 계층화 모션 추정법의 상위 계층을 실시간으로 처리하는데 필요한 주파수는 SAD연산을 한 클럭에 한다고 가정할 경우 100MHz가 넘게 된다. 이 계산은 다음 처리할 데이터를 미리 읽어 들이는 더블 버퍼링을 사용한다고 가정하여 메모리로부터

영상 데이터를 저장하고 있는 외부 RAM에서 읽어 오는 시간을 고려하지 않은 것이다. 더블 버퍼를 사용하지 않는다면 각 블록의 SAD 연산시 데이터를 외부 메모리로부터 읽어와야 하기 때문에 몇 배의 주파수가 필요하다. 만약 일반적인 프로세서를 이용하여 구현한다면 절대 값 연산과 차분 연산을 한 클럭에 끝내지 못할 뿐 아니라 더블버퍼도 사용하기 힘들기 때문에 고성능 프로세서를 사용한다 하더라도 순차적인 방법을 이용한 실시간 블록 매칭의 구현은 어렵다. 더블 버퍼를 만들 수 있는 PLD나 FPGA등도 50MHz이상에서 동작하기 힘들기 때문에 순차적으로 데이터를 처리하는 방법은 거의 구현이 불가능하다.

블록 매칭을 실시간으로 처리하려면 더 낮은 주파수로 동작이 되어야 하는데 이를 위해서는 병렬처리가 필수적이다. 또한 메모리의 충돌을 방지하기 위해 메모리로부터 읽어오는 데이터를 최소화하고, 메모리를 접근하는 통로를 단일화해야 하는데 이것에 적합한 방법은 순차적인 데이터 접근이다.

### 2.3 병렬처리 기반 탐색방법

본 논문에서는 여러 개의 처리기(PE : Processing Element)가 데이터 플로우 기법을 사용하여 병렬로 블록 매칭을 수행하는 방법을 제안한다. 계층화 모션 추적의 상위 단계에서는, 각 PE가 현재 블록의 한 줄에 대한 SAD 연산만을 수행하고 전체 PE의 계산 결과를 누적하여 전체 SAD 연산을 수행한다. 블록 매칭이 시작되면, PE의 레지스터에 각 PE가 SAD 연산을 실행할 부분, 즉, 현재 블록 중에서 한 가로 줄이 저장된다. 그 다음에 각 PE는 이전 프레임의 영상 데이터가 입력되면 입력된 데이터와 저장하고 있는 현재 영상과 SAD를 구해 그 값을 누적하며 한 가로 줄에 대한 누적된 값을 출력한다. 이는 다음과 같은 식으로 나타낼 수 있다.

$$PL(m,n) \oplus CL(p) = \sum_{k=0}^{N-1} |P(m,n+k) - C(p,k)| \quad \text{식 (2)}$$

$$SAD(i,j) = \sum_{l=0}^{M-1} PL(i+l,j) \oplus CL(l) \quad \text{식 (3)}$$

식 (2)에서  $\oplus$  연산을 정의하였다. N은 현재 블록의 가로 픽셀 수이며, P(i, j)와 C(i, j)는 각각 비교블록과 현재 블록에서 (i, j)에 위치하는 픽셀 값을 의미한다. 그리고 식(3)의 M은 현재 블록의 세로 픽셀 수를 나타내며, PL(i, j)는 이전 프레임의 탐색 영역의 (i, j)에 위치하는 비교 블록의 한 가로 줄을 의미한다. 이와 유사하게 CL(i)는 현재 블록의 (i, 0)에 위치하는 한 가로 줄을 의미한다. 식 (3)은 탐색영역 중 하나의 비교 블록에 대한 SAD 연산을 보여주고 있다.

앞에서 설명한 바와 같이 탐색영역의 크기는 16x32이고 현재 블록의 8x16이다. 본 논문의 계층화 모션 추정법의 상위 계층에서는 1/2 크기로 down 샘플링을 수행하므로 탐색 영역의 크기는 8x16이 되고 현재블록의 크기는 4x8이 된다. 현재 블록이 8개의 PE에 한 가로 줄씩 저장된다(한 줄이 4픽셀이고 픽셀 당 한 바이트이므로 4바이트가 저장된다). 그 다음에 탐색 영역이 PE에 순차적으로 입력된다. 각 PE는 탐색 영역의 픽

셀 값이 입력되면 저장되어 있는 현재 블록의 픽셀 값과 SAD 연산을 하게 된다. SAD 연산을 병렬로 처리하는 방법이 그림 3에 나타나 있다.

그림 3에서 PE0 - PE7은 각 PE를 나타내고 data 신호는 PE에 입력되는 탐색 영역 데이터를 나타낸다. End 신호는 PE에서 결과가 출력됨을 알리는 신호이다. Start 신호는 각 PE의 시작을 나타내는 신호이다. 앞에서 설명한 바와 같이 가로 4개의 픽셀을 나타내는 것이 PL(i, j)이다. PL(0,0)이 입력되면 PE0가 SAD 연산을 하며 결과는 PE1으로 전달된다. PL(1,0)이 입력되면 PE0는 두 번째 비교 블록의 첫 줄을 위한 SAD 연산을 하며, PE1은 첫 번째 비교 블록의 두 번째 줄의 SAD 연산을 수행하게 된다. 마찬가지로 PL(2,0)이 입력되면 PE0는 세 번째 비교 블록의 첫 번째 줄의 SAD 연산을 수행하게 되며, PE1은 두 번째 비교블록의 두 번째 줄의 SAD연산을 수행하며, PE2는 첫 번째 비교 블록의 세 번째 줄에 대한 SAD연산을 수행하게 된다.

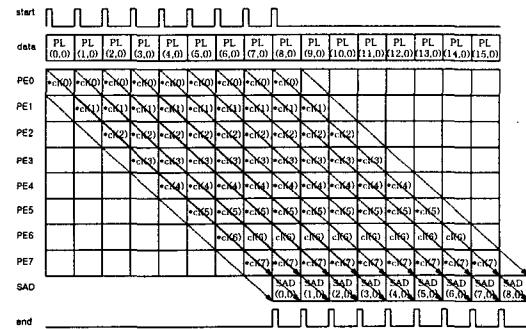


그림 3. 데이터 플로우 기법을 이용한 SAD 연산

연산이 계속 진행되어 PL(7,0)이 입력되면 8개의 PE가 동작하고 PE7에서는 PL(7,0)의 계산이 끝나면서 첫 번째 비교 블록의 SAD 결과를 출력한다. 그 다음에 PL(8,0)이 입력되고 계산이 완료되면 PE7에서는 두 번째 비교 블록의 SAD 결과가 출력된다.

출력된 SAD결과가 첫 번째 비교 블록의 결과보다 작으면 최소 SAD값으로 두 번째 비교 블록이 저장되고 그렇지 않으면 첫 번째 비교 블록의 결과가 최소 SAD로 저장된다. 이 과정이 마지막 비교 블록의 계산이 끝날 때까지 되풀이 된다. 결과로 하나의 탐색 영역에 대한 최소 SAD값이 출력되고 출력된 최소 SAD값에 해당되는 좌표가 현재 블록이 이전 프레임의 탐색 영역 중 어느 곳에서 이동 되었는가를 나타내는 좌표가 된다. 이와 같이 8개의 4바이트를 저장할 수 있는 PE를 이용해서 상위 계층에서 이동되었을 것으로 추정되는 위치 (x,y)를 추적한 다음에는, (x,y)와 그 주변 8개 위치 즉, (x-1, y-1)부터 (x+1, y+1) 사이의 영역에 있는 위치에 대하여 다운 샘플링되지 않은 원래의 영상을 사용하여 블록 매칭을 수행한다. 이때에는 현재 블록의 크기가 8x16이므로 8바이트를 저장할 수 있는 16개의 PE가 필요하게 된다.

계층화 모션 추적 방법의 상위 단계에서, 한 블록의

최소 SAD를 결정하는데 걸리는 시간은 탐색영역의 넓이(탐색영역의 세로크기 현재 영상의 세로크기)에 비례한다는 것을 알 수 있다. 이것으로 전체 블록을 실시간으로 처리하는데 대한 주파수는 식 (4)와 같다.

$$H = (P_h \times C_w) \times (P_w - C_w + 1) \times B \times \quad \text{식 (4)}$$

식(4)에서  $C_w$ 와  $C_h$ 는 각각 현재 블록의 가로 픽셀 수와 세로 픽셀 수이며,  $P_w$ 와  $P_h$ 는 각각 탐색 영역의 가로 픽셀 수와 세로 픽셀 수이다. 또한  $B$ 는 처리해야 할 블록 수이며,  $F$ 는 1초에 입력되는 프레임 수이다. 상위 계층에서는 1/2 이완 샘플링을 하기 때문에  $C_w=4$ ,  $C_h=8$ ,  $P_w=8$ ,  $P_h=16$ 이다. 그리고 처리해야 할 블록의 개수는  $39 \times 14$ 개이고  $F=60$ 이다. 그러므로 상위 계층을 처리하는데 필요한 주파수는 다음과 같다.

$$H_1 = (16 \times 4) \times (8 - 4 + 1) \times (39 \times 14) \times 60 = 10,483,200 \text{Hz}$$

하위 계층에서는 현재 블록의 크기가  $8 \times 16$ 이고 탐색 영역은  $10 \times 18$ 이므로 블록 매칭에 필요한 주파수는 다음과 같다.

$$H_2 = (18 \times 8) \times (10 - 8 + 1) \times 39 \times 14 \times 60 = 14,152,320 \text{Hz}$$

전체적으로 블록 매칭에 필요한 주파수는  $24,635,520 \text{Hz}$ 가 되어 하드웨어로 구현이 가능하다. 이 계산 결과는 영상처리에 최적화된 하드웨어를 순차적으로 계산했을 때 보다 4배 이상 빠르다. 현재 팬티엄 4 1.5G에서 MMX 명령어를 사용해 수행한 결과가  $320 \times 240$ 의 이미지를 초당 20프레임 정도 처리할 수 있는 결과에 비해서는 단순한 클럭 비교만 했을 때 250 정도 빠르다.

#### 2.4 군집화

블록들의 모션 벡터를 구한 다음에, 복수의 차량을 서로 다른 물체로 인식하고 이동하는 차의 크기를 인식하기 위해서 군집화 알고리즘을 사용하였다. 군집화 알고리즘으로는 ForgY 알고리즘[7], K-means 알고리즘[7] 등 여러 알고리즘이 있는데, 본 논문에서는 이소데이터 알고리즘(isodata algorithm)[7]을 수정하여 사용하였다. 군집화는 계산량이 적고, 병렬처리의 효율이 높지 않아 내장형 시스템의 단일 프로세서에 의해 구현되었다.

### III. 내장형 시스템 상에서 검지기 구현

영상입력을 위한 CMOS 이미지센서는 OmiVision사의 OV6620이 이용되었고, 병렬처리에 의한 블록 매칭을 실행하는 PLD는 10만 게이트의 소자를 직접 할 수 있는 ALTRA FLEX 10K100ARC240이 이용되었으며, 이소데이터 알고리즘의 실행에는 50MIPS의 ARM프로세서가 이용되었다. 그림 4는 구현된 영상처리기의 사진을 보여주고 있다.

### IV. 결 론

본 논문에서는 실시간 동작을 위하여 계층화 모션 추정법과 병렬 처리에 의거한 블록 매칭 알고리즘을

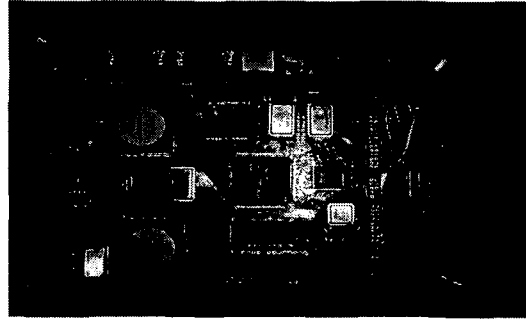


그림 4. 내장형 시스템을 이용한 영상 검지기 구현

제안하였다. 데이터흐름과 파이프라인 기법을 적용하여 여러 개의 PE가 FPGA에서 활성화 되어 동시에 수행되는 병렬처리에 의한 자동차의 움직임 분석을 실행하였다. 데이터흐름 방식과 파이프라인 구조를 사용함으로써 병렬 수행되는 각 PE에서 블록매칭에 요구되는 빈번한 메모리 접근을 최소화 하였다.

본 논문에서 제안하는 병렬처리에 의한 계층화 모션 추정기법을 사용하면 FPGA 상에서 여러 개의 PE가 병렬처리 방식에 의해 신속하게 연산이 수행되므로 저가로 구현이 용이하고 그 크기도 줄어 들 수 있으므로 PC를 기반으로 하는 검지기에 비해 가격과 크기 면에서 장점을 갖는다.

#### 참고문헌

- [1] M. Hasan, D. Cunneo and A. Chachich, "Analysis of Traffic Video to Develop Driver Behavior Models for Microscopic Traffic Simulation," MIT Center for Transportation Studies, Technical Report, 1999.
- [2] 박세현, 정기철, 허준구, 김항준, "영 기반의 실시간 교통 감시 시스템, 전자공학회논문지, 제 36 권, c편, 제 8 호, pp.582-589, 1999.
- [3] 정영기, 호요성, "차량의 영상추적을 통한 지능적인 교통정보 추출 알고리즘," Telecommunications Review, 제 9 권, 4호, pp.582-596, 1999.
- [4] R. Li, B. Zeng and M.L. Liou, "A New Three-step Search Algorithm for Block Motion Estimation," IEEE Trans. on Circuits and Systems for Video Technology, vol. 4, no. 4, pp. 438-442, Aug. 1994.
- [5] L. Po and W. Ma, "A novel four-step search algorithm for fast block motion estimation", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 6, No. 3, pp. 313-317, Jun. 1996.
- [6] C. W. Lin, Y.J. Chang, and Y. C. Chen, "Hierarchical Motion Estimation Algorithm Based on Pyramidal Successive Elimination", 1998 International Computer Symposium, Proceedings of Workshop on Image Processing and Character Recognition, pp.41-44.
- [7] E. Gose, R. Johnsonbaugh, S. Jost, Pattern Recognition and Image Analysis. Prentice Hall, 1996.