계층화 모션 추정법과 병렬처리를 이용한 차량 움직임 측정 알고리즘 개발 및 구현

강경훈[†] · 정성태[††] · 이상설[†††] · 남궁문[††††]

요 약

본 논문에서는 계층화 모션 추정법과 병렬 처리를 이용한 차량의 움직임 측정 알고리즘을 제안한다. 본 시스템에서는 CMOS 센서를 통하여 도로 영상을 캡쳐한다. 그 다음에 영상을 작은 블록들로 나누고 블록매칭을 이용하여 각 블록의 움직임을 계산한다. 그리고 움직임이 비슷한 블록들을 클러스터링하여 차량의 움직임을 측정한다. 본 논문에서는 실시간 동작을 위하여 계층화 모션 추정법과 병렬 처리에 의거한 블록매칭 알고리즘을 제안한다. 병렬처리를 위해서는 파이프라인과 데이터 플로우 기법을 도입하였다. 본 논문에서 제안된 시스템은 기존의 내장형 시스템을 이용하여 구현되었다. 제안된 블록매칭 알고리즘은 PLD(Programmable Logic Device)를 이용하여 구현하였고 클러스터링 알고리즘은 ARM 프로세서를 이용하여 구현하였다. 실험결과에 의하면 본 논문에서 구현된 시스템은 차량의 움직임을 실시간으로 추출할 수 있었다.

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

Kyung-Hoon Kang[†], Sung-Tae Jung[††], Sang-Seol Lee[†††] and Kung-Moon Nam[††††]

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 by using an embedded system. The proposed block matching algorithm has been implemented on PLDs(Programmable Logic Device) and clustering algorithm has been implemented by ARM processor. Experimental results show that the proposed system detects the motion of vehicles in real-time.

Key words: Block Matching, Parallel Processing, Image Based Vehicle Detection, Dataflow, Pipeline

---

본 연구는 한국과학재단 목적기초연구(R01-2000-000-00377-0)지원과 2001년도 원광대학교의 교비 지원에 의해서 수행되었음.
접수일 : 2003년 6월 21일, 완료일 : 2003년 7월 21일
[†] (주) 넷코덱 연구원
[††] 정회원, 원광대학교 전기전자 및 정보공학부 교수
[†††] 원광대학교 전기전자 및 정보공학부 교수
[††††] 원광대학교 토목환경 · 도시공학부 교수

1. 서 론

자동차의 급격한 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 도로정비 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 교통수요의 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, 설치시 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. 현재까지의 연구에서 고려된 검지기는 영상 검지기[1-7], 초음파 검지기[8,9], 초단파 검지기[10,11] 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 캡쳐 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 고가의 비용이 요구된다. 또한 PC라는 기본 플랫폼을 유지해야 하기 때문에 장치의 크기도 커지게 되어 실제 설치와 운용에 불편함을 겪게 된다. 그런데 영상 검지기의 영상 분석 부분을 하드웨어로 구현하면 기존의 PC기반의 영상검지기보다 간단하고 경제적으로 구성될 수 있다. 즉, VLSI 칩을 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 카메라[11]를 이용하여 획득한 영상을 작을 블록들로 분할한 다음에 블록매칭을 이용하여 각 블록의 움직임을 계산한다. 그리고 움직임이 비슷한 블록들을 클러스터링함으로써 차량의 움직임을 측정한다. 본 논문에서는 실시간 동작을 위하여 블록매칭 부분을 내장(embedded)형 시스템의 하드웨어상에서 병렬처리에 의한 계층화 모션 추정기법을 이용하여 구현하였다. 내장형 시스템의 PLD 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.

2. 계층화 모션 추정과 병렬처리 기반 영상 검지

2.1 블록매칭

본 논문에서 구현한 시스템은 CMOS 이미지센서를 사용하여 320×240 크기의 영상을 초당 60 프레임의 속도로 입력 받는다. 입력된 영상은 버퍼를 통해 메모리에 저장된 다음에 32비트 버스를 통해 블록매칭을 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 블록매칭이라 한다.

본 논문에서는 블록의 크기로 8×16을 사용하였고 각 블록과 가장 유사한 블록이 이전 영상에서 16×32 탐색 영역 안에 존재한다고 가정하였다. 현재 프레임의 블록과 이전 프레임에서의 탐색 영역의 관계는 그림 2와 같다. 이와 같은 가정에서 320×240

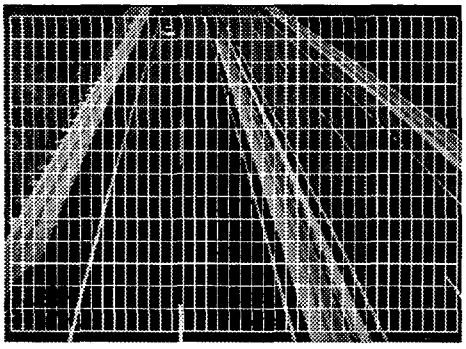


그림 1. 블록 분할

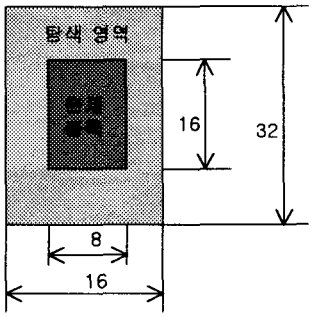


그림 2. 블록 설정

영상에 설정할 수 있는 블록의 개수는 그림 1에 나타나 있는 바와 같이 39×14개이다.

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, N은 블록의 가로방향과 세로방향의 픽셀 수를 나타낸다.

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

블록매칭 방법 중에서 가장 기본적인 방법은 전탐색 방법이다[13]. 모션 벡터의 최대 변위가 x축 방향으로 $\pm dx$이고 y축 방향으로 $\pm dy$라 할 때에 전탐색 방법에서는 모션벡터가 존재할 수 있는 모든 탐색공간 $(-dx\le u\le dx,\ -dy\le v\le dy)$에서 $SAD(u,v)$을 계산하여 그 중에서 $SAD(u,v)$값이 최소가 되는 위치의 블록을 선택한다. 이 방법은 탐색 영역의 모든 위치에서 두 블록 사이의 $SAD(u,v)$값을 계산하기 때문에 정확도는 높지만 $(2dx+1)\times(2dy+1)$번의 블록 비교를 하기 때문에 처리에 필요한 계산량이 많아서 실시간으로 동작할 수 있도록 구현하기에는 문제가 있다. 휘도차의 합 연산에서 한 픽셀의 연산에 한 클럭이 소요된다고 가정할 때에, d_x=4, d_y=8, 블록의 개수는 39×14, 현재 블록의 크기는 8×16, 이고 1초에 60프레임을 처리하므로 전체 블록에 대하여 전탐색을 수행하기 위해 필요한 주파수는 9×17×8×16×39×14×60=641,571,840MHz이다.

2.2 계층화 모션 추정

전탐색 방법이 많은 시간을 필요로 하므로 이에 비하여 계산량을 줄일 수 있는 여러 가지 방법들[14-18]이 제안되었다. 본 논문에서는 이들 방법 중에서 하드웨어에 의해 효율적으로 처리 가능한 방법으로 계층화 모션 추정 기법[18]을 채택하였다. 계층화 모션 추정 기법은 상위 계층에서 현재 블록과 탐색영역을 이완 샘플링하여 사용하는 방법으로 전탐색 기법에 비해 계산량을 현저히 줄일 수 있다.

본 논문에서는 두 단계 계층으로 나누어 상위 계층에서는 블록을 1/2 이완 샘플링을 하여 사용한다. 상위계층에서 발견된 매칭 블록의 위치가 (x, y)이면 하위 계층에서는 (x, y)를 중심으로 원래의 이미지를 사용하여 3×3 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 식 (2)와 같다.

$$H=(C_w\times C_h+C_w\times C_h\times(P_w-C_w+1)\times(P_h-C_h+1))\times B\times F \quad \text{식(2)}$$

식(2)에서 C_w , P_w는 각각 현재 블록과 이전 프레임 블록의 가로 픽셀 수이며, C_h, P_h는 각각 현재 블록과 이전 블록의 세로 픽셀 수이다. 또한 B는 처리해야 할 블록 수이며, F는 1초에 입력되는 프레임 수이다. 상위 계층에서는 1/2 이완 샘플링을 하기 때문에 C_w=4, P_w=8, C_h=8, P_h=16 그리고 처리해야 할 블록의 개수는 39 14개이고 F=60이므로 H=48,222,720Hz 이다. 식 (2)에서 괄호안의 덧셈의 첫 번째 항인 $C_w\times C_h$는 현재 블록을 PE에 미리 읽어 들이는 데 필요한

시간을 나타내고 두 번째 항은 SAD 연산에 필요한 주파수를 나타낸다.

하위 계층에 필요한 주파수도 식 (2)와 같이 구할 수 있는데, 하위 계층에서는 3 3 범위에서 전탐색을 수행하고 이완 샘플링을 사용하지 않고 원래의 영상을 사용하므로 C_w=8, C_h=16, P_w=10, P_h=18, B=39 14이고 F=60 이므로 H=41,932,800Hz이다.

그러므로 전체 블록매칭에 필요한 주파수는 90,155,520MHz이다. 이 계산은 다음 처리할 데이터를 미리 읽어 들이는 더블 버퍼링을 사용한다고 가정하여 영상 데이터를 저장하고 있는 외부 RAM에서 읽어 오는 시간을 고려하지 않은 것이다. 더블 버퍼를 사용하지 않는다면 각 블록의 SAD 연산시 데이터를 외부 메모리로부터 읽어 와야 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 블록매칭의 구현은 어렵다. 더블 버퍼를 만들 수 있는 PLD도 50MHz 이상에서 동작하기 힘들기 때문에 순차적으로 데이터를 처리하는 방법은 거의 구현이 불가능하다.

블록매칭을 실시간으로 처리하려면 좀더 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 블록매칭을 수행하는 방법을 제안한다.

2.3.1 계층화 모션 추적의 상위 단계

계층화 모션 추적의 상위 단계에서, 각 PE는 현재 블록의 한 줄에 대한 SAD 연산만을 수행하고 PE들의 계산 결과를 누적하여 전체 SAD 연산을 수행한다. 블록매칭을 위해서는 먼저, PE의 레지스터에 각 PE가 SAD 연산을 실행할 부분, 즉, 현재 블록 중에서 한 행이 저장된다. 그 다음에 각 PE는 이전 프레임의 영상 데이터가 입력되면 입력된 데이터와 저장하고 있는 현재 영상과 SAD를 구해 그 값을 누적하며 한 행에 대한 누적된 값을 출력한다. 이는 다음과 같은 식으로 나타낼 수 있다.

$$PL(m,n)\oplus CL(p)=\sum_{k=0}^{N-1}|P(m,n+k)-C(p,k)| \qquad \text{식 (3)}$$

$$SAD(i,j)=\sum_{l=0}^{M-1}PL(i+l,j)\oplus CL(l) \qquad \text{식 (4)}$$

식 (3)에서 ⊕ 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(4)의 M은 현재 블록의 세로 픽셀 수를 나타내며, $PL(i, j)$는 이전 프레임의 탐색 영역의 (i, j)에 위치하는 비교 블록의 한 행을 의미한다. 이와 유사하게 $CL(i)$는 현재 블록의 $(i, 0)$에 위치하는 한 행을 의미한다. 식 (4)는 탐색영역 중 하나의 비교 블록에 대한 SAD 연산을 보여주고 있다.

본 논문의 계층화 모션 추정법의 상위 계층에서는 1/2 크기로 이완 샘플링을 수행하므로 탐색 영역의 크기는 8×16이 되고 현재블록의 크기는 4×8이 된다. 따라서, 현재 블록이 8개의 PE에 한 줄씩 저장된다(한 줄이 4픽셀이고 픽셀 당 한 바이트이므로 4바이트가 저장된다). 그 다음에 탐색 영역이 PE에 순차적으로 입력된다. 각 PE는 탐색 영역의 픽셀 값이 입력되면 저장되어 있는 현재 블록의 픽셀 값과 SAD 연산을 하게 된다. SAD 연산을 병렬로 처리하는 과정이 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 신

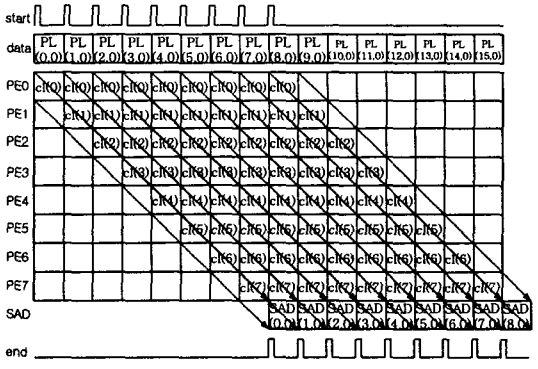


그림 3. 데이터 플로우 기법을 이용한 SAD 연산

호이다. 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 첫 번째 비교 블록의 첫 번째 줄의 SAD 연산, 즉 $PL(0,0)\oplus CL(0)$ 연산을 하며 결과는 PE1으로 전달된다. PL(1,0)이 입력되면, PE1은 첫 번째 비교 블록의 두 번째 줄의 SAD 연산, 즉 $PL(1,0)\oplus CL(1)$ 연산을 수행한 다음 그 결과 값을 PE0가 보내준 값에 누적하여 PE2에 전달한다. PE1이 연산을 하는 동안 PE0는 두 번째 비교 블록의 첫 줄을 위한 SAD 연산, 즉 $PL(1,0)\oplus CL(0)$ 연산을 수행하여 병렬 처리가 이루어진다. 마찬가지로 PL(2,0)이 입력되면 PE2는 첫 번째 비교 블록의 세 번째 줄에 대한 SAD 연산, 즉 $PL(2,0)\oplus CL(2)$ 연산을 수행하고, PE1은 두 번째 비교블록의 두 번째 줄의 SAD 연산, 즉 $PL(2,0)\oplus CL(1)$ 연산을 수행하며, PE0는 세 번째 비교 블록의 첫 번째 줄의 SAD 연산, 즉 $PL(2,0)\oplus CL(0)$ 연산을 수행한다.

연산이 계속 진행되어 PL(7,0)이 입력되면 8개의 PE가 병렬로 동작하고 PE7에서는 $PL(7,0)\oplus CL(7)$ 연산이 끝나면서 첫 번째 비교 블록의 SAD(0,0) 값을 출력한다. 그 다음에 PL(8,0)이 입력되고 계산이 완료되면 PE7에서는 두 번째 비교 블록의 SAD(1,0) 값이 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 되풀이된다. 결과적으로 탐색영역에 대한 최소 SAD값이 출력되고 출력된 최소 SAD값에 해당되는 좌표가 현재 블록이 이전 프레임의 탐색 영역 중 어느 곳에서 이동되었는가를 나타내는 좌표가 된다. 이와 같이 계층화 모션 추적 방법의 상위 단계에서, 한 블록의 최소 SAD를 결정하기 위한 주파수는 식 (5)와 같다.

$$H_1=(C_w\times C_h+P_h\times C_w\times(P_w-C_w+1))\times B\times F\text{식} \quad (5)$$

식(5)에서 C_w와 C_h는 각각 현재 블록의 가로 픽셀 수와 세로 픽셀 수이며, P_w와 P_h는 각각 탐색 영역의 가로 픽셀 수와 세로 픽셀 수이다. 또한 B는 처리해야 할 블록 수이며, F는 1초에 입력되는 프레임 수이다. 상위 계층에서는 1/2 이완 샘플링을 하기 때문에 C_w=4, C_h=8, P_w=8, P_h=16이다. 그리고 처리해야 할 블록의 개수는 39 14개이고 F=60이다. 그러므로 상위 계층을 처리하는데 필요한 주파수는 다음과 같다.

$$H_1=(4\times8+16\times4\times(8-4+1))\times39\times14\times60 = 11,531,520\text{Hz}$$

2.3.1 계층화 모션 추적의 하위 단계

상위 단계에서 8개의 4바이트를 저장할 수 있는 PE를 이용해서 현재 블록이 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 블록매칭을 수행한다. 하위 단계에서도 상위 단계와 동일한 파이프라인과 데이터 플로우 기법을 사용할 수 있는데, 이 때에는 현재 블록의 크기가 8×16이므로 8바이트를 저장할 수 있는 16개의 PE가 필요하게 된다. 하위계층에서는 현재 블록의 크기가 8×16이고 탐색 영역은 10×18이므로 블록매칭에 필요한 주파수는 다음과 같다.

$$H_2=(8\times16+18\times8\times(10-8+1))\times39\times14\times60= 18,345,600\text{Hz}$$

하위 단계의 주파수가 상위 단계의 주파수 보다 더 큰 이유는, 파이프라인의 한 단계에서 병렬로 동작하는 PE의 수가 상위 단계에서는 최대 8개인 반면에 하위 단계에서는 최대 3개이기 때문이다. 따라서, 하위 단계에서는 그림 4와 같이 데이터 흐름이 양방향으로 발생하도록 하여 보다 병렬성을 높이는 방법을 사용한다. 그림 4의 오른쪽에서 왼쪽으로 전달되는 데이터는 탐색영역의 영상이고 위에서 아래로 전달되는 데이터는 현재 블록의 영상이다. 하위 단계의 PE의 역할도 상위 단계와는 다르다. 상위 단계에서는 PE가 현재 블록의 한 줄의 픽셀 값들을 가지고 있으면서 그 줄에 대한 SAD 연산을 수행하여 누적된 값을 다음 PE에 전달했다. 그러나, 하위 단계에서는 각 PE는 SAD 값을 다른 PE에 전달하지 않고, 한 모션 벡터에 대한 전체 SAD 값을 누적하여 계산한다. 즉, PE0는 SAD(x-1, y-1) 값을, PE1은 SAD(x, y-1) 값을, 그리고 PE8은 SAD(x+1, y+1) 값을 계산한다.

계층화 모션 추적의 하위 단계에서, 현재 블록과

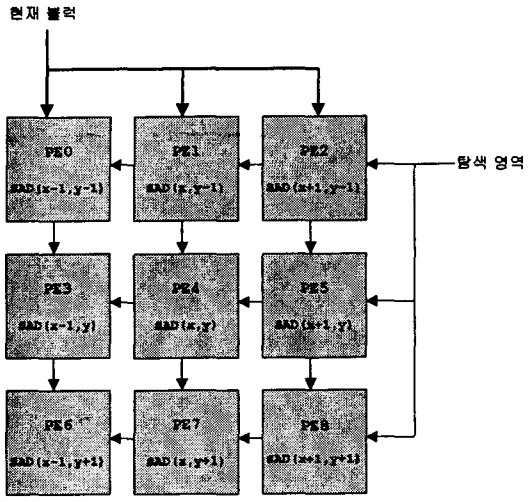


그림 4. 하위 단계 블록매칭을 위한 PE 구조

탐색 영역의 관계를 그림 5와 같다고 하자. 탐색 영역의 픽셀 주소가 좌측 상단 픽셀 P(0,0)으로부터 우측 하단의 P(17,9) 까지 그림 5와 같이 증가한다고 가정하자. 마찬가지로 현재 블록의 픽셀 주소도 C(0,0)으로부터 C(15,7) 까지 같은 방법으로 지정된다고 가정하자. 그림 5에서 점선으로 표시된 사각형이 현재 블록을 나타내는데, SAD(x-1,y-1), SAD(x,y), SAD (x+1,y+1) 연산을 위하여 탐색 영역에서 현재 블록과 비교되는 부분을 표시하고 있다.

그림 5와 같은 가정 하에 하위 단계에서 수행되는 SAD 연산을 식으로 표현하면 식(6)과 같다. 이 식에서 M은 현재 블록의 높이로써 16이고 N은 현재 블록의 넓이로써 8이다.

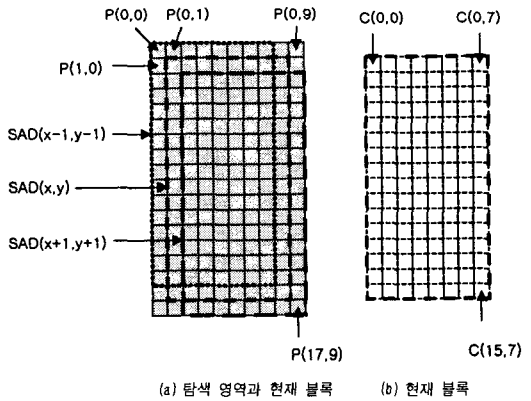


그림 5. 계층화 모션 추적의 하위 단계에서의 현재 블록과 탐색 영역의 관계

$$SAD(x+i,y+j)=\sum_{k=0}^{M-1}\sum_{l=0}^{N-1}|P(k+i+1,l+j+1)-C(k,l)| \quad \text{식 (6)}$$

하위 단계의 블록매칭은 먼저 현재 블록의 첫 번째 행을 읽어들이는 것으로 시작한다. 첫 번째 행의 8바이트의 픽셀 값이 순차적으로 읽어들여진 다음에 PE0, PE1, PE2에 동시에 전달된다. 각 PE는 픽셀 값이 전달될 때마다 8바이트 쉬프트 레지스터에 한 바이트씩 쉬프트 시켜가면서 저장한다. 그 다음에는 탐색 영역의 첫 번째 행의 픽셀 값들이 P(0,0)부터 순서대로 PE2, PE5, PE8에 전달된다. 이들 PE는 다음 픽셀값이 전달되면 이전 픽셀 값을 왼쪽 PE에 전달해 준다. 따라서, P(0,2)가 PE2에 입력될 때에 P(0,1)은 PE1에 입력되고, P(0,0)은 PE0에 입력되게 된다. 그리고 이때부터 SAD 연산이 시작된다. 즉, PE0는 P(0,0)와 C(0,0)과의 절대 차이값을, PE1은 P(0,1)과 C(0,0)과의 절대 차이값을, PE2는 P(0,2)와 C(0,0)과의 절대 차이값을 구한다.

C(0,0)과의 SAD 연산이 끝나면 C(0,0)의 값은 PE0, PE1, PE2에서는 더 이상 필요하지 않게 된다. 따라서 PE0, PE1, PE2는 C(0,0) 값을 PE3, PE4, PE5에 전달하고 동시에 다음 픽셀 값 C(1,0)을 읽어들인다. 그 다음에 P(0,3)이 PE2에 입력되면 PE0, PE1, PE2는 C(0,1)과의 SAD 연산을 수행한다. 그리고 C(0,1)을 PE3, PE4, PE5에 전달하고 C(1,1)을 읽어들인다. 이와 같이 계속해서 진행해나가다 P(0,9)가 PE2에 입력되면 PE0, PE1, PE2는 C(0,7)과의 SAD 연산을 수행하게 되고 PE0, PE1, PE2의 쉬프트 레지스터에는 현재 블록의 두 번째 행이 저장되고 PE3, PE4, PE5에는 현재 블록의 첫 번째 행이 저장된다. 그 다음에 탐색 영역의 두 번째 행이 입력되면 PE0, PE1, PE2는 현재 블록의 두 번째 행에 대한 SAD 연산을 PE3, PE4, PE5는 현재 블록의 첫 번째 행에 대한 SAD 연산을 수행한다. 같은 방법으로 탐색 영역의 세 번째 행이 입력되면 9개의 PE가 병렬로 동작하면서 SAD 연산을 수행하게 된다. 이와 같은 관계가 그림 6에 나타나 있다.

이와 같은 방식의 블록매칭에 필요한 주파수는 식(7)과 같다. 여기에서 C_w=8, C_h=16, P_w=10, P_h=18이므로 H_2 = 10,090,080Hz이다.

$$H_2=(C_w\times C_h+P_w\times P_h)\times B\times F \quad \text{식(7)}$$

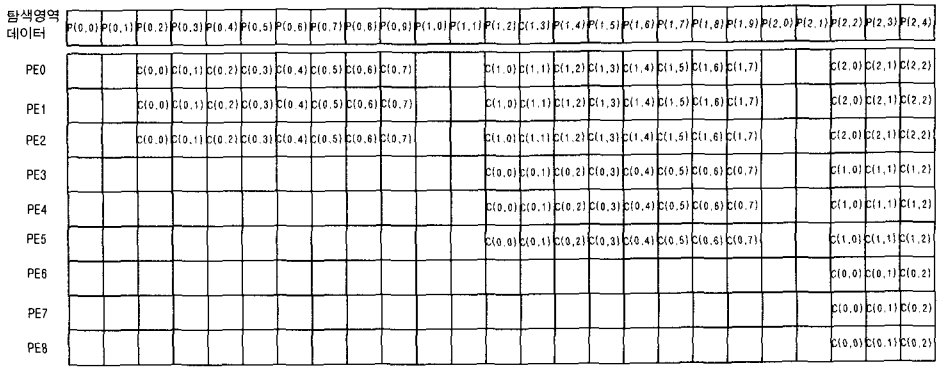


그림 6. 계층화 모션 추적의 하위 단계의 SAD 연산 흐름도

전체적으로 블록매칭에 필요한 주파수는 21,621,600Hz가 되어 하드웨어로 구현이 가능하다. 이 계산 결과는 순차적으로 계산했을 때 보다 4배 이상 빠르다. 그림 7에는 모션 벡터의 계산 결과가 나타나 있다. 그림 윗 부분의 차량은 멀리 있어서 두 영상 사이에 움직임이 감지되지 않았다.

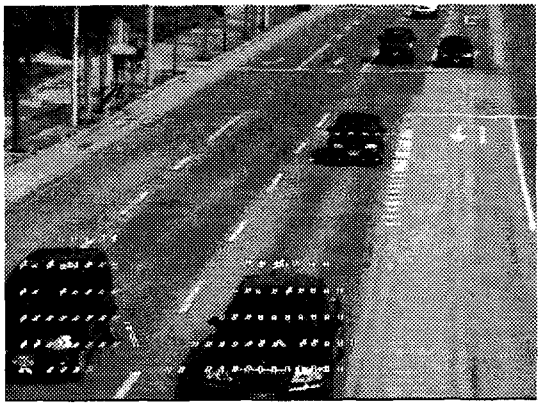


그림 7. 모션 벡터 계산 결과

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, K-평균 알고리즘, 이소데이터(isodata) 알고리즘 등 여러 가지 알고리즘[19]이 있는데, 본 논문에서는 이소데이터 알고리즘을 수정하여 다음과 같은 과정으로 차량을 추출하였다.

(1) 임의의 블록들을 초기 클러스터로 설정하고 각 클러스터의 중심을 계산한다. 클러스터의 중심은 군집화 기준으로 사용되는 값으로써 본 논문에서는 블록의 위치와 모션 벡터의 크기를 중심 값으로 사용하였다. 블록을 초기 클러스터로 설정하기 위해서는 블록을 x, y 방향으로 검색하여 주위 블록들의 움직임이 많은 블록을 선택하였다. 따라서, 초기 클러스터 수는 이동량이 많은 영역에 더 많이 배정된다.

(2) 블록의 모션 벡터가 0보다 큰 블록마다 가장 가까운 클러스터의 중심을 찾은 다음, 그 클러스터에 포함시킨다. 가장 가까운 클러스터의 계산에는 거리와 모션 벡터의 크기를 모두 고려한다. 블록을 추가한 다음에는 클러스터의 중심을 다시 계산한다.

(3) 포함된 샘플 수가 미리 정해진 최소 원소 수보다 적은 클러스터들을 제거한다.

(4) 각 블록에 대하여 주위의 블록과의 유사성을 검사한다. 즉, 블록 A의 주위 블록들의 대부분이 동일 클러스터에 속하면 블록 A도 같은 클러스터에 포함시킨다.

(5) 만약 두 개의 클러스터 사이의 거리가 클러스터 사이의 최소 거리보다 작다면 그들을 합병하고 다시 클러스터의 중심을 계산한다.

(6) 클러스터에 포함된 블록의 위치와 모션 벡터의 표준 편차가 큰 경우 비슷한 값을 가진 두 클러스터로 분할하고 각각의 중심을 계산한다.

(7) 정해진 반복 회수만큼 단계 (3)부터 (6)까지의 수행 중에 클러스터에 아무런 변화가 일어나지 않았거나 단계 (7)이 정해진 반복 회수만큼 수행되었으면 종료한다. 그렇지 않으면 단계 (3)으로 간다.

그림 8에는 군집화 예가 나타나 있다. 그림 8 (a)에는 단계 (1)의 초기 클러스터의 할당 예가 나타나 있

다. 여기에서는 6개의 초기 클러스터가 설정되었다. 그림 8 (b)에는 단계 (2)에서 블록들을 클러스터에 할당한 결과가 나타나 있다. 그림 8 (c)에는 단계 (4)에서 주위 블록과의 유사성 적용 결과가 나타나 있다. 여기에서는 차량의 한 부분이면서도 주위와의 색이 비슷한 이유로 모션 벡터가 0으로 계산된 블록이 차량이 속한 클러스터에 포함되는 것을 볼 수 있다. 그림 8 (d)에는 단계 (5)에서 블록을 합병한 결과가 나타나 있다. 단계가 진행되어도 더 이상의 변화가 없어 그림 8 (d)의 결과가 최종 군집화 결과이다.

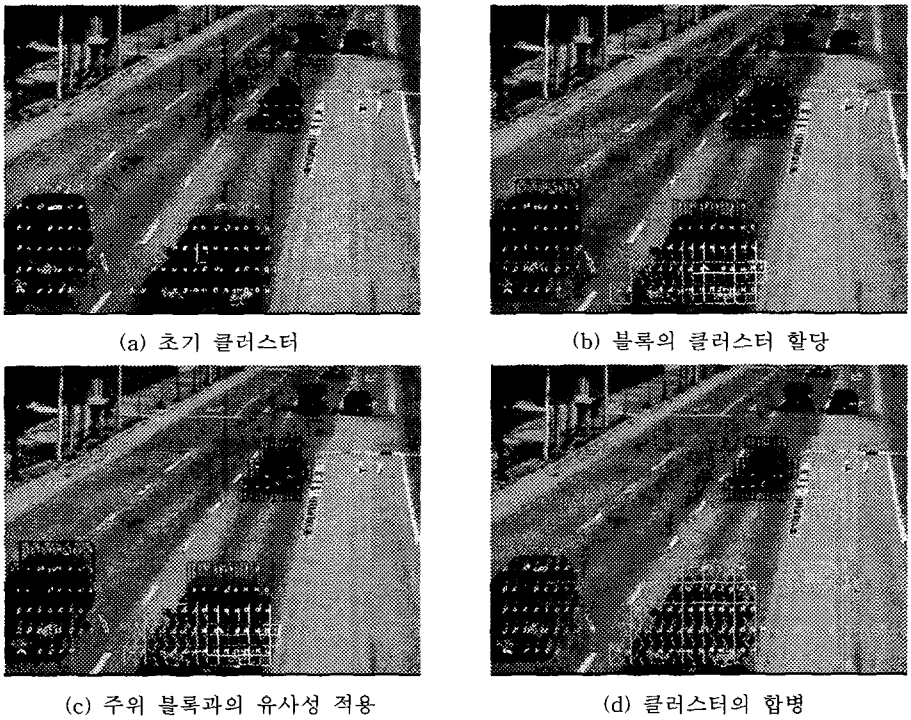
일반적인 이소데이터 알고리즘은 초기 씨드 포인트(seed point)를 임의의 지점에 일정한 수로 잡는다. 수정된 이소데이터 알고리즘에서는 씨드 포인트를 모션벡터가 밀집한 지역에, 자동차의 증가에 따라 씨드 포인트도 증가하도록 설정한다. 또한, 자동차 영상의 특성상 한대의 자동차는 같은 크기와 방향을 같는 모션벡터를 가지기 때문에 검지된 자동차 크기와 방향 벡터의 평균을 각각 구한 다음 임계치 이상의 차이를 보이면 이소데이터 알고리즘에 의해 검출된 자동차 영상이라도 검출 결과에서 제외했다. 이소데이터 알고리즘은 모든 영상데이터를 이용하는 것이 아니고 모션벡터를 이용하기 때문에 계산량이 적고, 알고리즘의 반복수행이 조건에 따라 달라지므로 병렬처리의 효율이 높지 않아 단일 프로세서에 의해 구현되었다.

3. 내장형 보드 상에서 검지기의 구현

영상입력을 위한 CMOS 이미지센서는 OmiVision사의 OV6120[12]를 사용하였고, 병렬처리에 의한 블록매칭을 실행하는 PLD는 10만 게이트의 소자를 집적할 수 있는 ALTERA FLEX 10K100ARC240을 사용하였으며, 이소데이터 알고리즘의 실행에는 50MIPS의 ARM프로세서[20]를 이용하였다. 그림 9는 영상검지기 구현에 사용된 내장형 시스템의 사진을 보여주고 있다.

그림 10은 전체 하드웨어의 블록도를 나타내고 있다. 시스템 버스는 32비트로 이루어졌으며, RAM에 접근하는 경우, 모두 burst 모드를 사용하여 버스의 효율성을 높였다.

영상 검지기의 중요 부분인 블록매칭부의 블록도는 그림 11과 같다. 시스템 내부에서 블록매칭을 위



(a) 초기 클러스터 (b) 블록의 클러스터 할당

(c) 주위 블록과의 유사성 적용 (d) 클러스터의 합병

그림 8. 군집화 예

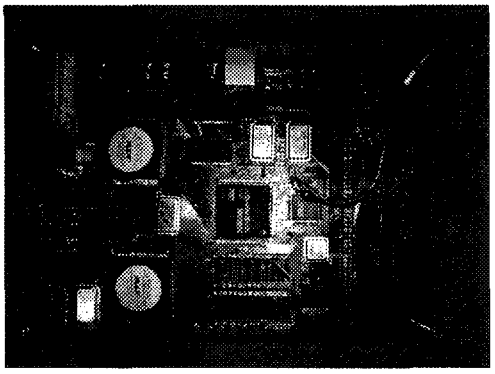


그림 9. 영상 검지기 구현에 이용된 내장형 시스템

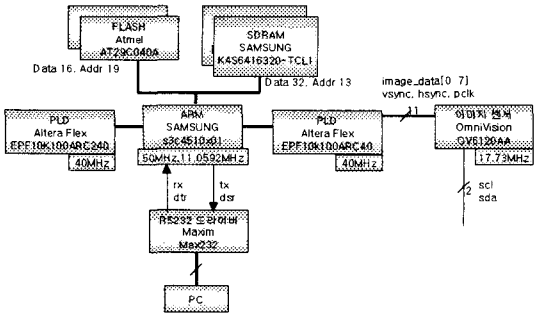


그림 10. 영상검지기의 하드웨어 블록도

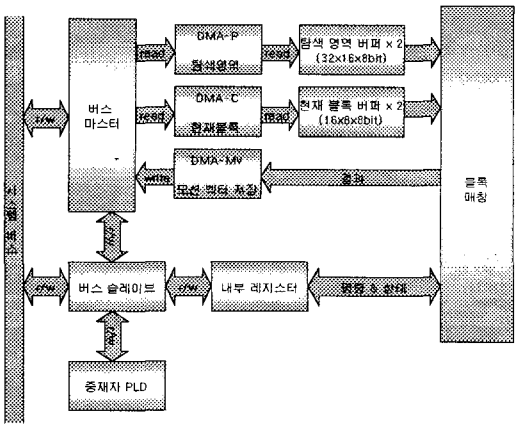


그림 11. 블록매칭부의 블록도

한 PLD, CPU, 메모리 중재자, 메모리는 버스를 공유한다. 버스 마스터는 버스 중재자에게 버스의 사용권을 얻어서 블록매칭을 수행할 데이터를 얻어오거나 계층화 모션 추정법을 수행한 결과인 최소 SAD값과 모션벡터를 시스템버스에 연결되어 있는 메인 메모리의 특정 위치에 저장하는 역할을 한다. 블록매칭을 수행하는데 필요한 데이터는 탐색의 기준이 되는 현재 프레임내의 임의의 위치에 존재하는 현재 블록(8×16 픽셀)과 이전 프레임에 존재하는 탐색영역(16×32 픽셀)이다. 그림11에서 DMA-C는 현재 프레임에서 하나의 비교 블록을 읽어 들여서 내부의 버퍼(8×16×8 비트)에 저장하는 역할을 한다. 그리고 DMA-P는 이전 프레임에서 탐색영역을 읽어 들여서 내부의 버퍼(16×32×8 비트)에 저장하는 역할을 한다. 그리고 DMA-MV은 블록매칭이 수행되고 난 후에 얻어지는 최소 SAD 값과 모션벡터를 메인 메모리에 저장하는 역할을 한다. 버스 슬레이브는 시스템버스 상에서 내부 레지스터에 대한 액세스가 발생할 때 이에 상응하는 응답이나 적절한 동작을 취하는 블록이다. CPU는 버스 슬레이브를 통해서 블록매칭부의 내부레지스터를 액세스 할 수 있고, 블록매칭 동작 파라미터를 전달할 수 있다.

그림 11의 각 블록들은 VHDL 언어를 이용하여 기술되었으며 시뮬레이션을 거친 다음 회로로 합성되었다. 그림 12는 VHDL 합성 도구인 MAX PLUS II를 이용해 ALTERA FLEX 10K100ARC240-2에 블록매칭부를 합성한 결과를 보여주고 있다. 합성 결과 로직셀은 4821개가 사용되었으며, EAB는 12개가 사용되었다. 그리고 최대 시간지연은 62.7ns로 15.94 MHz로 동작할 수 있었다.

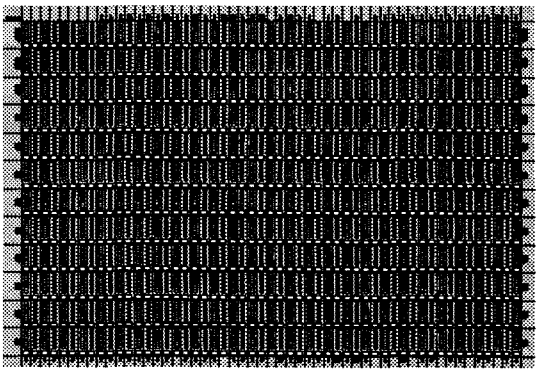


그림 12. ALTERA FLEX 10K100ARC240에 합성결과

4. 결 론

본 논문에서는 실시간 동작을 위하여 계층화 모션 추정법과 병렬 처리에 의거한 블록 매칭 알고리즘을 제안하였다. 데이터흐름과 파이프라인 기법을 적용

하여 여러 개의 PE가 PLD에서 활성화되어 동시에 수행되는 병렬처리에 의한 자동차의 움직임 분석을 실행하였다. 데이터흐름 방식과 파이프라인 구조를 사용함으로써 병렬 수행되는 각 PE에서 블록매칭에 요구되는 빈번한 메모리 접근을 줄일 수 있었다. 제안된 병렬처리 계층화 모션추정 기법을 내장형 시스템을 이용하여 구현한 결과 실시간으로 동작하였다. 본 논문에서 제안하는 병렬처리에 의한 계층화 모션 추정기법을 사용하면 PLD 상에서 여러 개의 PE가 병렬처리 방식에 의해 신속하게 연산이 수행되므로 저가로 구현이 용이하고 그 크기도 줄어들 수 있으므로 PC를 기반으로 하는 검지기에 비해 가격과 크기 면에서 장점을 갖는다.

참 고 문 헌

[1] Wu Yi-Ming; Ye Xiu-Qinc; Gu Wei-Kang, "A shadow handier in traffic monitoring system", Vehicular Technology Conference, Vol. 1, pp. 303-307, 2002.

[2] C. Setchell, E.L. Dagless, "Vision-based road-traffic monitoring sensor", IEE Proceedings on Vision, Image and Signal Processing, Vol. 148, No. 1, pp. 78-84, Feb. 2001.

[3] R. Cucchiara, M. Piccardi, P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system", IEEE Transactions on Intelligent Transportation Systems, Vol. 1, No. 2, pp. 119-130, June 2000.

[4] 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.

[5] M.Y. Siyal, M. Fathy, "A window-based image processing approach for real-time road traffic analysis", International Conference on Image Processing and Its Applications, Vol. 2, pp. 681-685, 1999.

[6] J. Badenas, F. Pla, "Segmentation based on region-tracking in image sequences for traffic monitoring", International Conference on Pattern Recognition, Vol. 2, pp. 999-1001, 1998.

[7] L. Wixson, K. Hanna, D. Mishra, "Improved illumination assessment for vision-based traffic monitoring", International Workshop on Visual Surveillance, pp. 34-41, 1997.

[8] H.-G. Kim, J.-H. Lee, S.-W. Kim, J.-I. Ko, D.-I. Cho, "Ultrasonic vehicle detector for side-fire implementation and extensive results including harsh conditions", IEEE Transactions on Intelligent Transportation Systems, Vol. 2 No. 3, pp. 127-134, Sep. 2001.

[9] H. Sumi, H. Takahashi, T. Izumi, N. Kiryu, W. Matsumoto, "Method of measuring travel time by using ultrasonic vehicle profile classifiers", Second International Conference on Road Traffic Monitoring, pp. 29-32, 1989.

[10] J. Wang, E.R. Case, D. Manor, "The Road Traffic Microwave Sensor (RTMS)", The 3rd International Conference on Vehicle Navigation and Information Systems, pp. 83-90, 1992.

[11] J.M. Milan, "Microwaves at Gilfillan division, ITT industries: past, present, and future", Microwave Symposium Digest, Vol. 3, pp. 1177-1180, 1999.

[12] Datasheet, "OV6120 single-chip CMOS CIF B&W digital camera", OmniVision Technologies, Inc., May 2000.

[13] L. De Vos, M. Stegherr, T.G. Noll, "VLSI architectures for the full-search blockmatching algorithm", International Conference on Acoustics, Speech, and Signal Processing, pp. 1687-1690, 1989.

[14] S. Lee, S.-I. Chae, "Two-step motion estimation algorithm for large search range using outlier pixel exclusion", Electronics Letters, Vol. 38, No. 2, pp. 68-69, Jan. 2002.

[15] H. Nisar, T.-S. Choi, "An advanced center biased three step search algorithm for motion estimation", IEEE International Conference on Multimedia and Expo, Vol. pp. 95-98, 2000.

[16] 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,

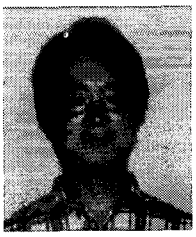
Jun. 1996.

[17] 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.

[18] C. W. Lin, Y.J. Chang, and Y. C. Chen, "Hierachical Motion Estimation Algorithm Based on Pyramidal Sucessive Elimination", 1998 International Computer Symposium, Proceedings of Workshop on Image Processing and Character Recognition, pp.41-44.

[19] E. Gose, R. Johnsonbaugh, S. Jost, Pattern Recognition and Image Analysis. Prentice Hall, 1996.

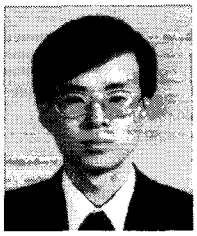
[20] R. Witek, J. Montanaro, "StrongARM: a high-performance ARM processor", Digest of Papers on Technologies for the Information Superhighway, pp. 188-191, 1996.



강 경 훈

2000년 2월 원광대학교 전기공학과 졸업
2002년 2월 원광대학교 전기공학과 석사학위 취득
2001년 1월~현재 (주)넷코덱 연구원

관심분야 : 컴퓨터구조, 영상처리, 통신시스템
E-mail : ghkang@netcodec.com



정 성 태

1987년 2월 서울대학교 컴퓨터공학과 졸업
1989년 2월 서울대학교 컴퓨터공학과 석사학위 취득
1994년 8월 서울대학교 컴퓨터공학과 박사학위 취득
1994년 9월~1995년 2월 한국전자통신연구소 박사후연수연구원
1999년 1월~1999년 12월 미국 Univ. of Utah 과학재단 지원 해외 Post-Doc.
1995년 3월~현재 원광대학교 전기전자 및 정보공학부 교수
관심분야 : 휴먼 컴퓨터 인터페이스, 영상 처리, VLSI/CAD
E-mail : stjung@wonkwang.ac.kr

이 상 설

1984년 2월 고려대학교 전자공학과 졸업
1989년 2월 한국과학기술원 전기 및 전자공학과 석사학위 취득
1994년 2월 한국과학기술원 전기 및 전자공학과 박사학위 취득
1994년~현재 원광대학교 전기전자 및 정보공학부 교수
관심분야 : 병렬컴퓨터구조, SoC, 영상 VLSI, 통신 VLSI, 인식 VLSI
E-mail : slee@wonkwang.ac.kr

남 궁 문

1984년 2월 원광대학교 토목공학과 졸업
1986년 2월 전북대학교 토목공학과 석사학위 취득
1992년 3월 히로시마대학교 토목공학과 박사학위 취득
1997년 1월~1998년 1월 미국 Univ. of Illinois 교환교수
1999년 12월~2000년 1월 Technische University Darmstadt 교환교수
2000년 7월~2000년 8월 Hiroshima University 교환교수
1992년 3월~현재 원광대학교 토목환경·도시공학부 교수
관심분야 : 교통행태, 인지공학, Simulation, CVM, GIS, Fuzzy, Neural남궁문
E-mail : ngmoon@wonkwang.ac.kr

교 신 저 자

이 상 설 570-749 전북 익산시 신용동 344-2 원광대학교 전기전자및정보공학부 교수