

실시간 화재 특징 추출을 위한 임베디드 매니코어 프로세서의 디자인 공간 탐색

서준상*, 강명수*, 김철홍**, 김종면*

Design Space Exploration of Embedded Many-Core Processors for Real-Time Fire Feature Extraction

Jun-Sang Suh*, Myeongsu Kang*, Cheol-Hong Kim**, Jong-Myon Kim*

요 약

본 논문에서는 많은 연산량이 요구되는 화재 특징 추출 알고리즘을 위한 최적의 매니코어 프로세서에 대한 디자인 공간을 탐색한다. 최적의 매니코어 디자인 공간을 선택하기 위해 매니코어를 구성하는 프로세서 엘리먼트 (PE)의 개수와 로컬 메모리 사이즈를 변화시키면서 시뮬레이션을 수행하여 성능, 에너지 효율 및 시스템 면적 효율에서 최적인 매니코어 구조를 결정한다. 본 논문에서는 256×256 해상도의 30 프레임으로 구성된 화재/비화재 비디오 영상을 대상으로하여 움직임 검출, 색상 분할 및 이산 웨이블릿 변환으로 구성된 화재 특징 추출 알고리즘을 여섯 가지 매니코어 구조(PEs=16, 64, 256, 1,024, 4,096, 16,384)를 사용하여 모의 실험한 결과, 모든 화재/비화재 비디오 영상에 대해 1,024개와 4,096개의 PE를 갖는 매니코어 구조가 각각 최적의 시스템 면적 효율과 에너지 효율을 보였다. 또한, 실험에서 사용한 여섯 가지 매니코어 구조 모두가 실시간 비디오 처리에서 요구되는 초당 30 프레임 처리 기준을 만족하였다.

▶ Keywords : 화재 특징 추출, 매니코어 프로세서, 시스템 면적 효율, 에너지 효율

Abstract

This paper explores design space of many-core processors for a fire feature extraction algorithm. This paper evaluates the impact of varying the number of cores and memory sizes for the

•제1저자 : 서준상 •교신저자 : 김종면

•투고일 : 2013. 2. 8, 심사일 : 2013. 4. 2, 게재확정일 : 2013. 5. 30

* 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan)

** 전남대학교 전자컴퓨터공학부(School of Electronic and Computer Engineering, Chonnam National University)

※ 이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구이며(No. NRF-2013R1A2A2A05004566), 또한 중소기업청에서 지원하는 2012년도 산학연공동기술개발사업(No. C0031322)의 연구수행으로 인한 결과물이며, 산업통상자원부의 광역경제권 선도산업 육성사업의 일환인 "동남광역경제권 선도산업 지원단"의 2013년 연구비 지원으로 수행되었음(R0001220).

many-core processor and identifies an optimal many-core processor in terms of performance, energy efficiency, and area efficiency. In this study, we utilized 90 samples with dimensions of 256×256 (60 samples containing fire and 30 samples containing non-fire) for experiments. Experimental results using six different many-core architectures (PEs=16, 64, 256, 1,024, 4,096, and 16,384) and the feature extraction algorithm of fire indicate that the highest area efficiency and energy efficiency are achieved at PEs=1,024 and 4,096, respectively, for all fire/non-fire containing movies. In addition, all the six many-core processors satisfy the real-time requirement of 30 frames-per-second (30 fps) for the algorithm.

▶ Keywords : Fire feature extraction, many-core processors, system area efficiency, energy efficiency

I. 서 론

최근 공장, 선박, 도시기반 시설 등의 내부에 전기, 난방 및 가스 등의 기간시설이 포함되어 있어 화재의 발생 가능성과 위험성이 크게 증가되고 있다. 이러한 시설의 화재안전 확보를 위해서는 화재를 조기에 신속하게 감지하고 진압하여 피해를 최소화 하여야 한다. 기존의 화재 감지는 열과 연기 센서의 신호를 받아들여 화재 여부를 판단하였으나[1], 센서를 이용한 화재감지 시스템은 탐지범위가 제한되어 있고, 화재가 발생한 후 일정시간이 지난 후에 열이나 연기가 확산되어야 감지가 가능하므로 조기 화재 감지가 어렵다[2]. 이러한 단점들을 해결하기 위해 영상 처리를 통한 화재 감지 알고리즘 연구가 활발히 진행되고 있다[3-5]. 특히 화재의 색상 및 형태를 이용하여 화재 영역을 결정하는 방법들 가운데 이산 웨이블릿 변환(discrete wavelet transform, DWT)을 이용하는 경우가 많은데 이는 영상 전체를 대상으로 주파수 변환을 수행함으로써 배경에 비해 높은 고주파 성분을 갖고 상승방향성을 가진 화재의 특징 추출에 적합하기 때문이다[6]. 하지만 이산 웨이블릿 변환은 전체 영상에 대해 저대역/고대역 통과 필터를 반복적으로 수행해야하므로 많은 연산량이 요구되어 실시간 화재 감지에 제약이 된다.

이러한 실시간 처리 문제의 단점을 해결할 수 있는 대안으로 칩 내부에 다수의 코어를 탑재하여 적은 전력소모로 높은 성능을 기대할 수 있는 매니코어 프로세서가 유망하다[7-10]. 매니코어 프로세서 구조로는 다음과 같이 크게 세 가지 유형으로 나뉘는데, 동일한 명령어로 여러 데이터를 동시에 처리하는 SIMD(single-instruction, multiple-data)

방식, 여러 개의 프로세서를 사용하여 서로 다른 명령어를 수행하지만 하나의 데이터 스트림을 처리하는 MISD(multiple-instruction, single-data)방식, 그리고 모든 프로세서들이 서로 다른 명령어를 실행하면서 여러 개의 데이터를 처리하는 MIMD(multiple-instruction, multiple-data)방식이 이에 해당한다. GPGPU(general purpose graphics processing unit)의 경우 기본적으로 SIMD방식에 해당되지만 더 발달된 형태로써 하나의 명령어로 많은 스레드를 제어하는 SIMT(single-instruction, multiple-thread)방식이라고 할 수 있다.

GPU 기술은 CPU에서 실시간으로 처리하기 힘든 3차원 그래픽스와 같은 연산을 효율적으로 처리함으로써 응용프로세서의 부담을 줄여줌과 동시에 병렬처리를 통해 연산속도를 가속화하는 효과를 얻을 수 있다[8],[9]. GPU와 같이 명령어 레벨(instruction-level)이나 스레드 레벨(thread-level) 프로세서들은 실리콘 면적을 멀티포트 레지스터 파일(multiported register file), 캐쉬(cache), 파이프라인(deep pipelined) 기능 유닛 등으로 사용하는 반면, SIMD 기반 매니코어 프로세서는 수십, 수백 개의 저비용 프로세싱 엘리먼트(processing element, PE)들을 이용하여 고성능을 추구하고 동시에 저장장소와 데이터 통신 요구를 최소화하기 위해 프로세싱 엘리먼트와 데이터 입-출력을 동일위치에 배치함으로써 저전력을 만족시킨다[10]. 특히 이산 웨이블릿 변환을 이용한 화재의 특징 추출 방법은 지역성(locality)이나 규칙성(regularity)을 가지고 있으므로 SIMD 기반 매니코어 프로세서에 적합하다.

본 논문에서 선택한 화재 특징 추출 알고리즘은 기존의 화재 및 연기 감지에 많이 사용되는 배경 차분(background subtraction) 기반 움직임 영역 검출과 화재의

RGB(red-green-blue) 색상 분포 정보를 이용한 색상 분할 알고리즘을 통해 화재 후보 영역을 검출한다[11]. 다음으로 화재 후보 영역에 대해 이산 웨이블릿 변환을 수행함으로써 화재의 특징을 추출한다[12]. 또한 기존 고성능 프로세서의 디자인 공간 탐색 기법[13],[14]을 응용하여 본 논문에서는 화재 특징 추출 알고리즘을 위한 최적의 매니코어 구조를 탐색하기 위해 총 여섯 가지 PE 구조(PEs=16, 64, 256, 1,024, 4,096, 16,384)에 대해 각각 시스템 면적 효율 및 에너지 효율을 측정하였으며, 각 실험은 동일한 130 nm 테크놀로지와 100 MHz의 클럭 주파수를 사용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 화재감지 알고리즘을 병렬 구현한 관련 연구를, 3장에서는 매니코어 프로세서의 구조 및 실험 방법론을, 4장에서는 화재의 특징 추출을 위한 알고리즘의 병렬 구현 방법에 대해 각각 설명한다. 5장에서는 PE 구조 별 실험 시간, 시스템 면적 효율 및 에너지 효율 결과를 바탕으로 화재 특징 추출 알고리즘을 위한 최적의 매니코어 프로세서를 결정한다. 마지막으로 6장에서는 본 논문의 결론을 맺는다.

II. 관련 연구

1993년 Healey가 제안한 화재 색상 정보를 이용한 실시간 화재감지 시스템을 시작으로 영상처리 기반 화재감지 알고리즘에 대한 연구가 활발히 진행되고 있다[15]. 이후 Phillips 등은 화재의 색상 정보뿐만 아니라 화재의 동적인 특성을 고려하여 연속된 프레임 간 변화 정보를 화재감지에 사용하기도 하였다[16]. 하지만 붉은색 옷을 입고 움직이는 사람, 자동차 헤드라이트 등 화재와 유사한 특징을 포함하고 있는 객체들이 영상에 포함될 수 있으므로 화재감지의 성능 향상을 위해서는 보다 뚜렷한 화재의 특징 추출에 대한 연구가 요구되어 왔다. 따라서 Liu 등은 화재 영역에 대한 푸리에 계수의 변화율과 화재 모양에 대한 푸리에 계수를 화재의 시간-공간적 특징으로 사용함으로써 화재감지에 대해 좋은 성능을 보였다[18],[19]. 하지만 이와 같은 특징 추출 알고리즘은 많은 연산량을 요구하므로 실시간 사용에 제약이 된다. 따라서 최근 화재감지 알고리즘에 대한 실시간 처리를 위해 먼저 영상을 분할하고 분할된 영역을 다수의 코어에서 동시에 실행시켜 화재의 특징을 추출하는 등 실시간 영상처리를 통한 조기 화재감지를 위한 연구가 활발히 진행 중에 있다 [20],[21].

III. 시뮬레이션 환경

1. SIMD 기반 매니코어 프로세서

그림 1은 SIMD 기반 매니코어 프로세서의 블록 다이어그램을 보여준다. 2중 격자구조의 PE와 이를 제어하는 어레이 컨트롤 유닛(array control unit, ACU), 외부 I/O와 PE 로컬 레지스터 간의 데이터를 전송하는 데이터 교환 인터페이스(data exchange interface, DEI)로 구성되어 있다. ACU는 명령어를 인출(fetch)하여 전체 PE에 동시에 전송(broadcasting)하며, 또한 특정 PE를 활성화(active) 또는 비활성화(deactivate) 시키는 역할을 한다. ACU에 의해 데이터가 각 PE에 분배되면 각 PE는 명령어를 동시에 수행한다.

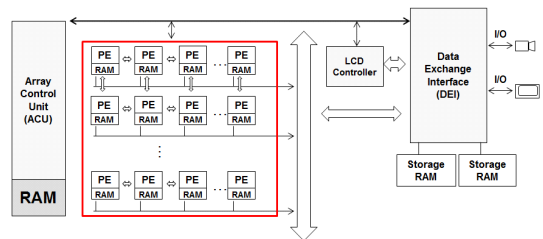


그림 1. SIMD 기반 매니코어 프로세서 아키텍처
Fig. 1. SIMD based many-core processor architecture

그림 2는 단일 PE의 구조를 보여주며, 각각의 PE는 다음과 같은 특징을 가진다.

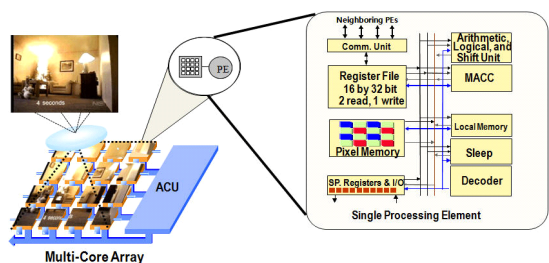


그림 2. 단일 PE 구조
Fig. 2. Single processing element

- 32비트 폭의 로컬 메모리
- 32비트 폭의 16개 3포트 범용 레지스터
- 기본적인 산술/논리 연산을 수행하는 ALU
- 64비트 곱셈 및 누산기 (multiply accumulator)
- 산술/논리 시프트 연산을 수행하는 배럴 시프트(barrel shift)

- 지역 정보를 이용해 각 PE들을 활성화 및 비활성 시키는 Sleep 유닛
- 이웃하는 PE들과 데이터 통신을 위한 north-east-west-south(NEWS) 네트워크 및 serial I/O유닛

2. 실험 방법론

그림 3은 SIMD 기반 매니코어 프로세서의 성능, 시스템 면적 효율 및 에너지 효율을 평가하기 위한 실험 방법론을 보여준다. 실험 방법론은 애플리케이션, 아키텍처 및 테크놀로지 레벨로 구성되어 있다. 애플리케이션 레벨에서는 매니코어 프로세서용 시뮬레이터를 이용하여 화재 특징 추출 알고리즘을 병렬 구현하여 실행 사이클 개수, 동적 명령어 빈도, 시스템 이용률 등의 실험 결과를 얻는다. 아키텍처 레벨에서는 아키텍처 모델링 툴을 사용하여 모델링된 아키텍처의 디자인 변수들을 계산한다[22]. 테크놀로지 레벨에서는 아키텍처 레벨에서 구해진 디자인 변수들을 Generic System Simulator(GENESYS)의 입력으로 사용하여 아키텍처 모델들의 사이클 시간(cycle time), 와이어 지연(wire latency), 전력(power), 클록 주파수(clock frequency)를 계산한다[23]. GENESYS는 매니코어 배열 및 멀티 클러스터 프로세서 등의 다양한 시스템 구조를 모델링하기 위한 테크놀로지 모델링 툴로써 각 코어들은 동기 ASIC 칩으로 표현되는 계층적 모델로 구성된다. 이러한 계층적 모델은 기본요소, 재료, 디바이스, 회로, 시스템 등 5-레벨로 이루어져 있다. 처음 세 가지 레벨은 재료특성과 스위칭 디바이스 특성의 물리적 효과를 모델링하며, 회로 레벨은 신호 지연, 동적 및 정적 에너지와 같은 게이트의 특징을 모델링한다. 마지막으로 시스템 레벨은 싱글 ASIC 칩을 묘사하기 위한 아키텍처, 연결구조, 패키징 상세 정보를 포함한다. 마지막으로 디자인 공간 탐색기(design space explorer)를 이용하여 위의 세 가지 레벨에서 구해진 파라미터들을 조합하고, PE 구조 별 실행시간, 시스템 면적 효율 및 에너지 효율을 측정한다.

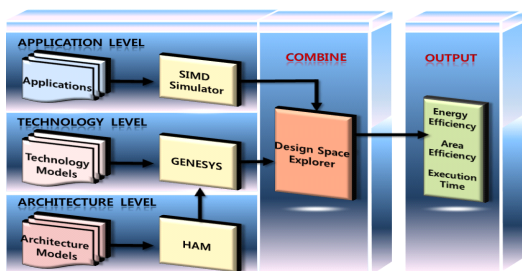


그림 3. 매니코어 프로세서를 위한 실험 방법론
Fig. 3. Simulation methodology for many-core processors

IV. 화재 특징 추출을 위한 알고리즘의 병렬 구현

I장에서 언급한 바와 같이 본 논문에서는 화재 특징 추출을 위해 먼저 움직임 영역 검출 및 색상 분할 알고리즘을 매니코어 상에 병렬 구현하여 화재 후보 영역을 검출한다. 이후 1단계 이상 웨이블릿 변환을 병렬 구현한 후 4가지 서브밴드(LL, LH, HL, HH)중 LH와 HL밴드에서 웨이블릿 계수를 이용하여 구한 에너지와 분산 값을 화재의 특징으로 하며, 자세한 내용은 아래와 같다.

1) 움직임 영역 검출

일반적으로 화재가 포함된 영상의 경우라도 다양한 사물 및 복잡한 배경 등이 포함되어 있으며, 이는 화재의 특징 추출을 어렵게 한다. 따라서 본 논문에서는 먼저 화재가 갖는 동적인 특성을 이용하여 움직임 영역 검출을 통해 화재 후보 영역을 검출한다. 그림 4는 매니코어 프로세서 상에서 움직임 영역 검출 과정을 보인다. 현재 프레임과 이전 프레임의 픽셀 간 감산 결과가 실험적으로 정의된 임계값보다 클 경우 해당 픽셀을 움직임 영역으로 판단한다. 이 때 본 논문에서는 움직임 영역과 배경을 구분하기 위해 움직임 영역의 경우는 해당 픽셀 위치에 '1'을 표기하여 움직임 영역임을 표기하였고, 배경의 경우는 움직임 영역의 픽셀 정보를 저장하고 있는 PE를 비활성화 한 후 '0'을 표기함으로써 배경임을 표기하였다. 이와 더불어 PE가 저장하고 있는 픽셀정보가 모두 배경일 경우 해당 PE를 비활성화 한 후 화재 특징 추출을 위한 다음 단계를 진행하도록 하였다.

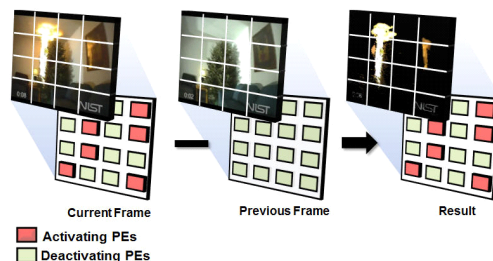


그림 4. 매니코어 프로세서 상에서 움직임 영역 검출
Fig. 4. Movement-containing region detection on the reference many-core processor

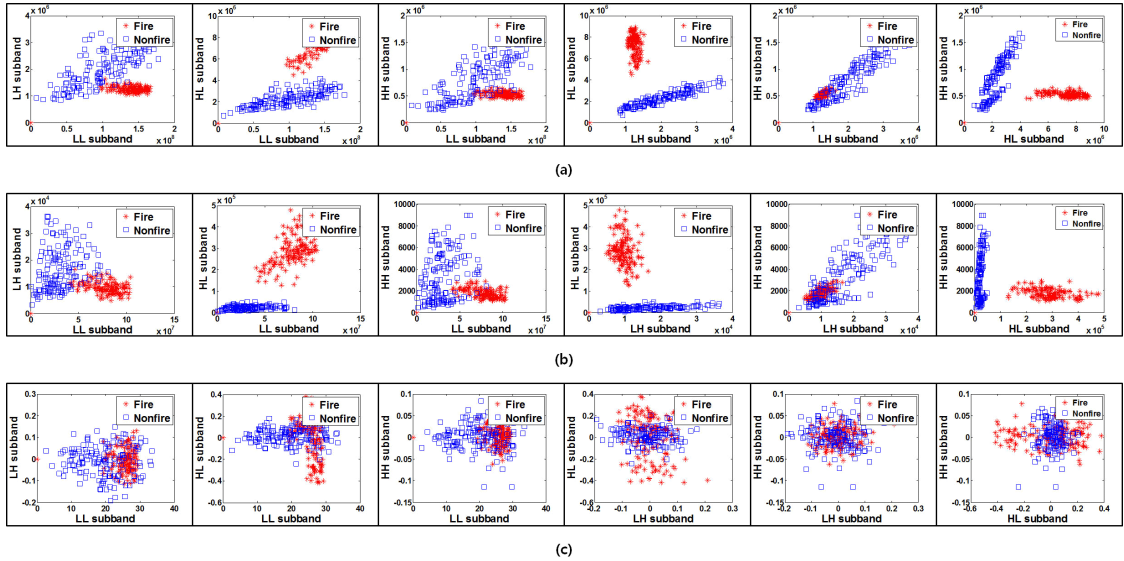


그림 6. 웨이블릿 서브밴드별 특징 분포. (a) 에너지, (b) 분산, (c) 평균
 Fig. 6. Feature distribution for different wavelet subband. (a) Energy, (b) variance, and (c) mean

2) 색상 분할

움직임 영역 검출 이 후 붉은색 옷을 입고 움직이는 사람 혹은 움직이고 있는 붉은색 승용차 등은 화재 후보 영역에 포함될 수 있으므로 본 논문에서는 화재가 갖는 색상 정보를 이용하여 색상 분할을 수행함으로써 보다 정확한 화재 후보 영역을 검출한다.

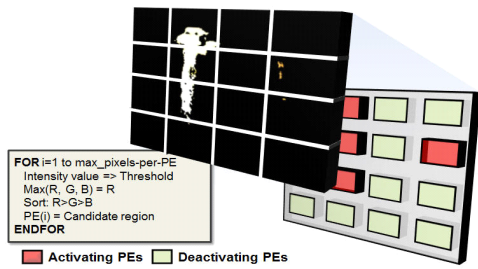


그림 5. 매니코어 프로세서 상에서 색상 분할
 Fig. 5. Color segmentation on the reference many-core processor

그림 5는 움직임 영역 검출 후 활성화된 PE에서 색상 분할 수행 과정을 보여주며, 이 후 움직임 영역 검출과 유사하게 PE 활성화/비활성화를 통해 화재 색상과 유사한 픽셀 위치에는 '1'을, 그렇지 않은 픽셀 위치에는 '0'을 표기하고 움직임 영역 검출 결과와의 AND 연산을 통해 최종 화재 후보 영역을 검출한다.

3) 화재 특징 추출

움직임 영역 검출과 색상 분할 이후에도 여전히 화재와 유사한 객체가 영상에 포함 될 수 있으며 이를 위해 본 논문에서는 1단계 이산 웨이블릿 변환을 통해 화재 감지를 위한 화재의 특징을 추출한다. 또한 이산 웨이블릿 변환 시 계산 복잡도가 크지 않으면서도 데이터 손실이 적은 4탭 Daubechies 필터를 이용한다[24].

본 논문에서는 50개의 화재 영상과 50개의 비 화재 영상에 대해 움직임 영역 검출 및 색상 분할 후 화재 후보 영상에 대해 1단계 이산 웨이블릿 변환을 수행하고 서브밴드별 평균, 분산, 에너지 값의 분포를 살펴보았으며, 그림 6에서와 같이 LH와 HL 서브밴드에서 추출한 에너지와 분산은 화재의 특징으로 적합함을 확인할 수 있었다. 또한 그림 6(a)와 (b)에서 볼 수 있듯이 LL 서브밴드에 대한 에너지와 분산 역시 LH와 HL 서브밴드와 더불어 화재의 특징으로 사용이 가능하나 LL 서브밴드는 화재의 특징 중 하나인 고주파 성분을 충분히 반영하지 못하는 경우가 종종 발생하여 화재와 비 화재를 구분하는 특징 추출을 위한 밴드로 사용하기에는 다소 어려움이 있었다. 또한 HHH 서브밴드의 경우는 화재의 특징인 고주파 성분은 잘 반영하나 저주파 성분인 화재의 외곽선이나 방향성 등의 특징은 잘 반영하지 못하였다. 평균값의 경우에는 모든 서브밴드에 대해서 화재의 특징으로는 적합하지 않았는데 이는 화재의 동적인 특성으로 인해 그 값의 분포가 고르지 못하기 때문이다. 앞서 언급하였듯이, 화재의 특징 추

출을 위해서는 색상 분할 후 발생한 화재 후보 영상에 대해 1 단계 이산 웨이블릿 변환을 매니코어 프로세서를 이용하여 구현하였으며, 그림 7은 고대역/저대역 통과 필터링을 통한 1 단계 이산 웨이블릿 변환 결과이다. 이산 웨이블릿 변환 결과 각 PE는 4개의 서브밴드(LL, LH, HL, HH)에 대한 웨이블릿 계수를 자신의 로컬 메모리에 저장한다.

다음으로 화재의 특징으로 사용할 LH와 HL 서브밴드에서 웨이블릿 계수에 대한 에너지와 분산을 계산하기 위해서는 PE간 데이터 이동이 불가피 하다. 먼저 그림 8(a)와 (b)에서 볼 수 있듯이 LL과 LH 서브밴드에 해당하는 웨이블릿 계수는 각 PE의 좌측(혹은 서쪽)으로, HL과 HH 서브밴드에 해당하는 웨이블릿 계수는 각 PE의 우측(혹은 동쪽)으로 이동시킴으로써 세로축 기준 저대역/고대역 웨이블릿 계수의 좌/우 정렬을 가능하게 한다. 이와 유사한 방법으로 그림 8(c)와 (d)에서와 같이 PE간 데이터 이동을 할 경우 가로축 기준 저대역/고대역 웨이블릿 계수의 상/하 정렬 또한 가능하다.

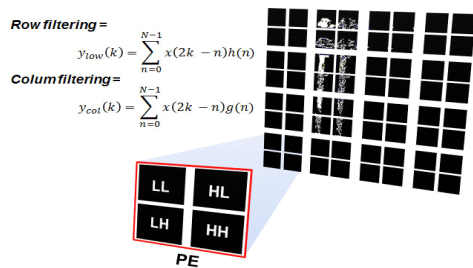


그림 7. 매니코어 프로세서 상에서 1단계 이산 웨이블릿 변환 Fig. 7. One-level discrete wavelet transform on the reference many-core processor

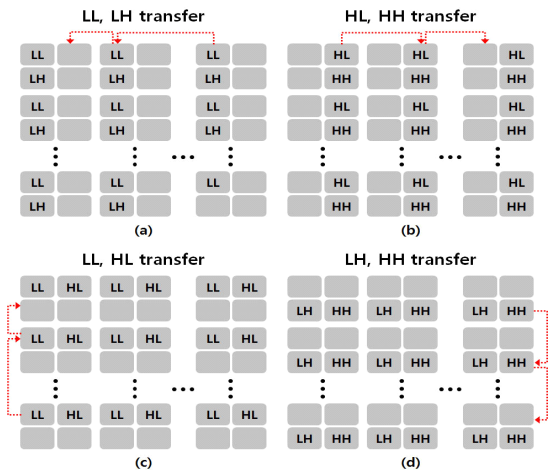


그림 8. 웨이블릿 계수 재 정렬을 위한 PE간 내부 통신 Fig. 8. PE inter-communications for wavelet coefficients reordering

이는 결과적으로 왼쪽 윗부분, 왼쪽 아랫부분, 오른쪽 윗부분, 오른쪽 아랫부분에 위치한 PE로 LL, LH, HL, HH 서브밴드에 대한 웨이블릿 계수를 재 정렬하는 결과를 받게 되며 그림 9와 같다. 마지막으로 LH와 HL 서브밴드에 대한 웨이블릿 계수를 이용하여 에너지와 분산을 구하기 위해서는 해당 정보 이외에 나머지 정보를 저장하고 있는 PE들을 비활성화 시킴으로써 가능하다. 위와 같은 과정은 시스템 이용률에 많은 영향을 미치게 되며, 화재의 특징 추출에 사용되는 PE의 수가 늘어날수록 많은 PE를 비활성화 시켜야하는 결과를 초래하게 되어 시스템이용률을 저하시킨다.

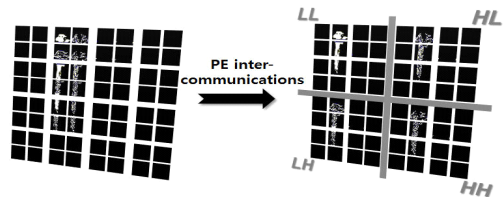


그림 9. 웨이블릿 계수 재 정렬 결과 Fig. 9. Result of wavelet coefficients reordering

V. 실험 결과 및 분석

1. 성능 평가 지표 및 실험 환경

화재의 특징 추출을 위한 매니코어 프로세서의 성능 평가를 위해 본 논문에서는 실행 시간(execution time), 시스템 이용률(system utilization), 시스템 면적 효율(area efficiency), 에너지 효율(energy efficiency)을 측정하며, 표 1과 같다[25]. 실행 시간은 화재 특징 추출을 완료하는데 소요되는 시간을 뜻하며, 시스템 면적 효율은 단위 면적당 처리량, 에너지 효율은 소비된 에너지 당 처리량으로 각각 정의된다.

표 1. 매니코어 프로세서 성능평가 지표 Table 1. Evaluation metrics for the many-core processor

실행 시간 (execution time)	$t_{exec} = \frac{C}{f_{clk}}$
에너지 효율 (energy efficiency)	$\eta_E = \frac{1}{t_{exec} \cdot Energy} \left[\frac{1}{s \cdot Joule} \right]$
시스템 면적 효율 (area efficiency)	$\eta_A = \frac{1}{t_{exec} \cdot Area} \left[\frac{1}{s \cdot mm^2} \right]$

표 3. 화재/비화재 비디오 영상에 대한 PE 구조 별 프레임 당 평균 실행시간
 Table 3. Average execution times per frame with varying PE configurations for fire or non-fire-containing movies

실행시간 (ms)	Movie 1				Movie 2				Movie 3			
	Overall	MRD	CS	FE	Overall	MRD	CS	FE	Overall	MRD	CS	FE
PEs=16	17.869	12.71	1.621	3.534	18.116	12.88	1.643	3.583	18.142	12.90	1.646	3.588
PEs=64	4.723	3.125	0.398	1.199	4.786	3.167	0.403	1.215	4.795	3.173	0.404	1.217
PEs=256	1.326	0.763	0.097	0.465	1.342	0.772	0.098	0.470	1.344	0.774	0.098	0.471
PEs=1,024	0.423	0.186	0.023	0.213	0.427	0.188	0.024	0.215	0.428	0.188	0.024	0.215
PEs=4,096	0.179	0.043	0.005	0.130	0.180	0.043	0.005	0.131	0.1809	0.043	0.005	0.131
PEs=16,384	0.132	0.009	0.001	0.121	0.132	0.009	0.001	0.121	0.132	0.009	0.001	0.121

여기서 C 는 사이클 수, f_{clk} 는 클럭 주파수, $Energy$ 는 130 nm 공정에서 소비된 에너지(Joule), $Area$ 는 시스템의 단위 면적(mm^2)을 나타낸다. 또한 실험에 사용된 PC는 Intel Core™ i5-2500 Quad-Core 3.3 GHz이며, 성능 평가에 사용된 비디오 영상은 그림 10과 같이 256×256 해상도의 30 프레임으로 구성된 화재/비화재 영상을 사용하였다.



그림 10. PE 구조 별 성능 평가에 사용된 비디오 예
 Fig. 10. Video examples for performance evaluation with varying PE configurations

표 2는 성능분석을 위해 모델링된 프로세서의 시스템 파라미터로써 각 실험은 여섯 가지의 PE 구조(PEs=16, 64, 256, 1024, 4096, 16,384)에 대해 130 nm 테크놀로지와 100 MHz 클럭 주파수를 사용하였으며, PE간 내부 통신은 메쉬(mesh) 구조를 사용하였다. 본 논문에서 선택한 6개의 PE 구조는 256×256 해상도 비디오 영상의 각 프레임을 처리하는데 있어 PE 당 처리할 이미지 픽셀 개수와 관련 있다.

표 2. 시스템 파라미터
 Table 2. Modeled system parameters

Parameters	Values					
PEs	16	64	256	1,024	4,096	16,384
Memory/PE (KB)	64	16	4	1	0.25	0.125

2. 실행 시간

표 3은 화재/비화재 비디오 영상에 대해 움직임 검출

(movement-containing region detection, MRD), 색상 분할(color segmentation, CS), 화재 특징 추출(feature extraction, FE), 전체 알고리즘(Overall)에 대해 여섯 가지 PE 구조 별 프레임 당 평균 실행 시간을 나타낸다. 움직임 검출 및 색상 분할의 경우는 각 PE에 할당된 픽셀 값과 임계 값 사이의 비교를 통하여 화재 후보 영역을 검출하기 때문에 많은 병렬성을 갖게 되어 표 3에서와 같이 사용되는 PE 개수의 증가와 거의 비례하여 실행 시간이 감소되는 것을 볼 수 있었다. 하지만 PE의 수가 4배씩 증가하였음에도 불구하고 4배씩의 실행 시간이 감소되지 않는 이유는 Amdahl의 법칙(Amdahl's law)을 따르는 것으로[12], 예를 들어 영상 픽셀을 PE로 분배, 알고리즘 내부에서 사용자 정의 함수(user defined function)를 호출하는 경우에는 PE 수의 변화와는 상관없이 일정한 사이클이 소요되므로 실행 시간의 감소가 PE 수의 증가와 비례하지 않게 되는 것이다. 특징 추출 부분의 경우는 III장에서 설명한 바와 같이 적절한 위치로 웨이블릿 계수의 재 정렬, LH와 HL 서브밴드에서 에너지와 분산 계산을 위해 PE간 많은 내부 통신 및 PE 활성화/비활성화가 요구되어, 결과적으로 PE의 수가 증가하더라도 실행 시간 측면에서는 큰 성능 향상을 보이지 않았다.

화재 특징 추출 시 특히 색상 분할 단계에서는 화재와 유사한 색상을 갖는 영역을 추출하기 위해 프레임에 속한 R채널의 픽셀 값이 설정한 임계값보다 클 경우 해당 위치(혹은 좌표)에서 RGB 최댓값을 임계값보다 큰 R채널의 픽셀 값으로 치환하거나 RGB 픽셀 값을 재 정렬하는 등의 추가적인 처리 작업이 요구되므로 영상정보에 따라 프레임별 실행 시간이 달라질 수 있다. 30 프레임으로 구성된 화재/비화재 비디오 영상에 대해 실행 시간을 측정할 결과 16개의 PE를 사용하였을 때 Movie 1, 2, 3에 대해 각 17.87 ms, 18.12 ms, 18.14 ms의 프레임 당 평균 실행 시간이 소요되어 모든 PE 구조에서 실시간 영상처리에서 요구하는 초당 30 프레임의 처리(33 ms)를 만족하였다.

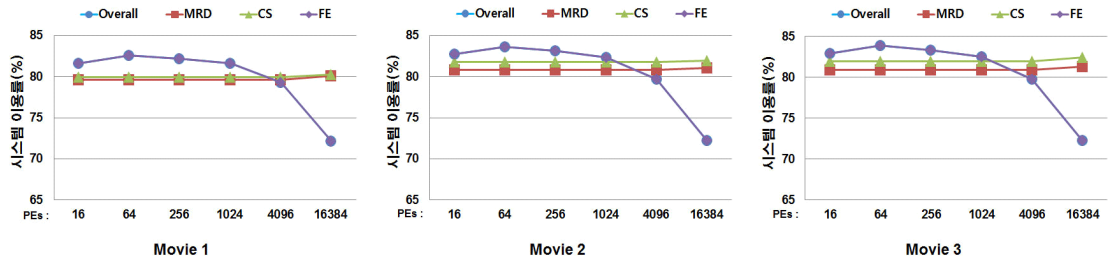


그림 11. 다양한 PE 구조에 따른 화재/비화재 비디오 영상에 대한 시스템 이용률
 Fig. 11. System utilization with different PE configurations for fire/non-fire videos

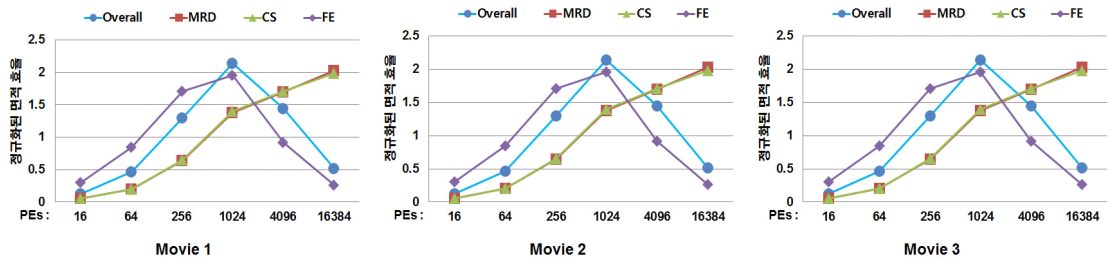


그림 12. 화재/비화재 비디오 영상에 대한 PE 구조 별 정규화된 시스템 면적 효율
 Fig. 12. Normalized system area efficiencies with varying PE configurations for fire and non-fire containing videos

3. 시스템 이용률

그림 11은 PE 구조에 따른 시스템 이용률로, III.1절과 III.2절에서 설명한 것과 같이 움직임 영역 검출(MRD)과 색상 분할(CS)의 경우는 화재 후보 영역을 검출함에 있어 배경과 움직임 영역을 표기하고, 화재와 유사한 색상 정보를 포함하고 있는 영역을 추출하기 위해 PE 활성화/비활성화가 요구된다. 이때 요구되는 PE 활성화/비활성화 횟수는 영상에 정보에 의존적인 것으로 사용되는 PE 수와 비례하여 증가하거나 감소하지 않는다. 따라서 그림 11에서와 같이 PE가 증가하더라도 시스템 이용률은 거의 일정하게 된다. 그림 11에서 Movie 2와 3의 경우 움직임 영역 검출 및 색상 분할 단계에서 시스템 이용률이 Movie 1의 것보다 다소 높은 것을 확인할 수 있는데 이는 Movie 2와 3의 경우 화재 후보 영역에 해당하는 부분이 Movie 1보다 더 넓게 퍼져있어 움직임 영역 검출 및 색상 분할을 위해 더 많은 PE가 활성화되었기 때문이다.

특징 추출(FE)의 경우는 배경 부분의 픽셀 값을 '0'으로, 화재 후보 영역의 픽셀 값은 처리되는 현재 프레임의 픽셀 값으로하여 256×256 크기의 영상을 대상으로 이산 웨이블릿을 수행한다. 따라서 특징 추출 부분은 영상 정보와 상관없이 PE 수가 증가할수록 웨이블릿 계수의 재 정렬을 위해 상·하·좌·우로 더 많은 PE가 요구되는 내부 통신으로 인해 PE 비

활성화 횟수 또한 증가하게 되므로 시스템 이용률이 감소하게 된다. 전체 알고리즘(Overall)에 대한 시스템 이용률의 경우는 모든 화재/비화재 비디오 영상에 대해 PE 구조 별 움직임 영역 검출, 색상 분할 알고리즘에 대한 시스템 이용률이 일정한 이유로 화재 특징 추출에 대한 시스템 이용률에 의존하여 그 모양이 나타난다.

4. 시스템 면적 효율

시스템 면적 효율은 단위 면적당 시스템의 처리량을 뜻하며, 그림 12는 화재/비화재 비디오 영상에 대한 PE 구조 별 정규화된 시스템 면적 효율을 보여준다. 그림 12에서 움직임 영역 검출(MRD) 및 색상 분할(CS)의 경우는 PE의 수가 증가할수록 시스템 면적 효율 또한 지속적으로 증가하는 것을 볼 수 있는데 이는 실행 시간과 시스템 면적과 관계가 있다. 즉 실행 시간의 경우 표 3에 나타나듯이 PE 수가 4배씩 증가함에 따라 Movie 1, 2, 3의 경우 실행 시간은 최소 1.2배에서 최대 3.77배로 감소하는 반면 시스템 면적은 최소 1.04배에서 최대 3.35배로 선형적으로 증가하여 시스템 면적 증가 대비 실행시간의 감소폭이 더 크므로 화재/비화재 비디오 영상에 대해 PE 구조 별 시스템 면적 효율이 계속적으로 증가한다(표3과 표4 참조). 반면 화재 특징 추출(FE)과 전체 알고리즘(Overall)의 경우는 시스템 면적 증가대비 실행 시간

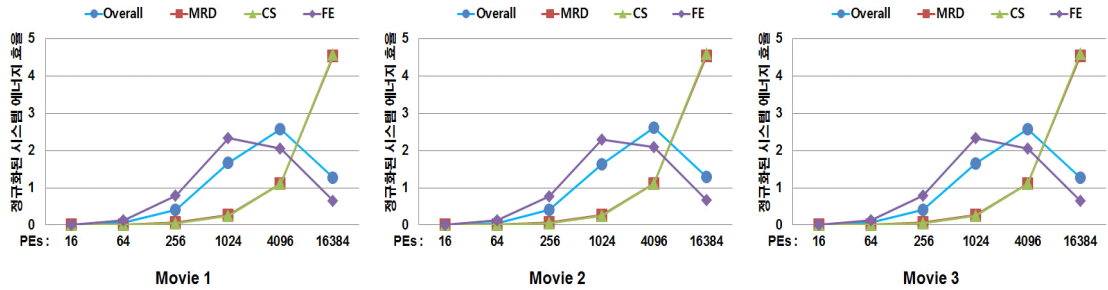


그림 13. 화재/비화재 비디오 영상에 대한 PE 구조 별 정규화된 시스템 에너지 효율

Fig. 13. Normalized energy efficiencies with different numbers of processing elements for fire/non-fire containing videos

표 5. 최적의 매니코어 프로세서와 상용 프로세서와의 성능 비교

Table 5. Performance comparison between the optimal many-core processors and a commercial processor

Parameter	Units	TI TMS320C6416	Many-core processors	
Number of core		1	1,024	4,096
Technology	(nm)	130	130	130
Clock frequency	(MHz)	720	100	100
Area	(mm ²)	529	198.7	645.9
Average power	(mW)	950	616	1930
Execution time	(ms)	142.83	0.428	0.18
Energy	(mJoule)	135.69	0.264	0.35
Area efficiency	(1/s·mm ²)	0.013	11.7586	8.6012
Energy efficiency	(1/s·Joule)	51.597	8850184	15873015

의 감소가 가장 두드러진 1,024개의 PE를 사용하였을 때 모든 비디오 영상에 대해 가장 높은 시스템 면적 효율을 보였다. 또한 그림 12에서 화재/비화재 비디오 영상에 대해 그래프의 모양이 유사하게 나타나는 이유는 표 3에서 볼 수 있듯이 움직임 영역 검출 및 색상 분할 단계에서 프레임들을 구성하고 있는 영상 정보에 따라 실행 시간의 차이가 발생하나 그 차이가 그래프의 모양이 달라질 정도로 실행 시간의 차이가 두드러지지 않기 때문이다.

표 4. PE 구조 별 시스템 면적

Table 4. System area for each PE configuration

PEs	시스템 면적 (mm ²)	PEs	시스템 면적 (mm ²)
16	82.06	1,024	198.73
64	85.82	4,096	645.93
256	107.12	16,384	2164.01

5. 에너지 효율

에너지 효율은 소비된 단위 에너지 당 처리량을 의미하며, 그림 13은 화재/비화재 비디오 영상에 대한 PE 구조 별 정규화된 에너지 효율을 나타낸다. 시스템 면적 효율과 유사하게 에너지 효율 또한 알고리즘 수행 시 소비된 에너지와 알고리즘 실행 시간과 관계가 있으며, 움직임 영역 검출(MRD) 및 색상 분할

(CS)의 경우 Movie 1,2,3에 대해 PE 구조 별 소비된 에너지 증가에 비해 실행 시간의 감소가 더 크기 때문에 에너지 효율은 PE 수가 증가함에 따라 계속적으로 증가하는 것을 확인할 수 있었다(표 3 참조). 화재의 특징인 웨이블릿 계수를 이용한 에너지 및 분산을 계산하기 위해서는 이산 웨이블릿 변환 후 웨이블릿 계수의 재 정렬이 요구되는데, 이는 IV.1절에서 언급한 바와 같이 PE 수가 증가할수록 잦은 PE 활성화/비활성화 및 많은 PE간 내부 통신을 야기한다. 즉 PE 수가 증가함에 따라 Sleep 유닛과 Communication(XFER 명령어) 유닛에 의해 소비되는 에너지의 증가폭이 커져 1,024개 이상의 PE를 사용할 경우 모든 화재/비화재 비디오 영상에 대해 에너지 효율이 감소하는 결과를 보였다. 이는 결과적으로 전체 알고리즘(Overall)에 영향을 미치게 되어 4,096개의 PE를 사용하였을 때 모든 화재/비화재 비디오 영상에 대해 가장 높은 에너지 효율을 보였다.

6. 상용 프로세서와의 성능 비교

본 절에서는 최적의 매니코어 프로세서와 상용 프로세서(TI TMS320C6416)와의 효율을 비교한다. 상용 프로세서와의 정확한 비교는 적정하지 않지만 본 절의 목적은 상용 프로세서와의 비교를 통해 선택한 최적의 매니코어 프로세서의

잠재가능성을 알아보기 위함이다. 실험에 사용된 비디오 가운데 Movie 2의 실행 시간이 가장 많이 소모되어 본 논문에서는 Movie 2를 대상으로 화재 특징 추출 알고리즘 수행을 위한 실행 시간, 시스템 면적 효율 및 에너지 효율을 측정하였고, 표 5는 TI TMS320C6416과의 비교 결과를 나타낸다. 비교 결과, TI TMS320C6416 DSP보다 1,024개와 4,096개의 PE를 각각 사용하였을 경우 실행 시간에서는 약 333.7배와 약 793.5배, 시스템 면적 효율에서는 약 904.5배와 약 661.6배, 에너지 효율에서는 약 171,525배와 307,635배의 향상을 보였다.

VI. 결 론

본 논문에서는 화재 감지를 위한 효과적인 특징 추출을 위해 먼저 영상의 움직임 영역 검출 및 색상 분할을 통해 화재 후보 영역을 검출하였으며, 검출된 화재 후보 영상을 대상으로 1단계 이상 웨이블릿 변환을 수행 한 후 LH와 HL 서브밴드의 웨이블릿 계수를 이용해 화재의 특징을 추출하였다. 위의 모든 알고리즘은 SIMD 기반 매니코어 프로세서 상에서 병렬 구현하였으며, 256×256 해상도의 30 프레임으로 구성된 화재/비화재 비디오 영상을 대상으로 가장 효율적인 에너지 소비와 시스템 면적을 사용하는 최적의 매니코어 프로세서 구조를 탐색하기 위해 총 여섯 가지 PE 구조(PEs=16, 64, 256, 1,024, 4,096, 16,384)에 대해 각각 면적 효율 및 에너지 효율을 측정하였다. 실험 결과 모든 화재/비화재 비디오 영상에 대해 1,024개와 4,096개의 PE를 각각 사용하였을 때 각각 최적의 시스템 면적 효율과 에너지 효율을 보였다. 또한, 실행 시간을 측정한 결과, 한 프레임에서 화재의 특징을 추출하는데 최대 약 18 ms의 시간이 소요되어 모든 PE 구조에서 일반적인 실시간 비디오 처리에서 요구하는 초당 30프레임 처리 기준을 만족하였다.

향후 추출된 특징과 신경망 회로 및 서포트 벡터 머신 등의 분류기와 결합하여 하나의 통합 화재 감지 시스템을 위한 최적의 매니코어 구조 탐색과 관련한 연구를 계속 진행할 계획이다.

참고문헌

- [1] B.Y. Lee, S.T. Park, S.H. Hong, and D.H. Baek, "A Study on the Fire Detection Algorithm for Early Fire Detection of Electrical Fire," 2009 The Korean Institute of Electrical Engineers Summer Conf., pp. 2162-2163, Muju Resort, Jeonbuk, 2009.
- [2] K.H. Cheong, B.C. Ko, and J.Y. Nam, "Vision-Based Early Fire Detection System," Korean Society for Imaging Science & Technology, Vol. 13, No. 1, pp. 62-71, 2007.
- [3] T.X. Truong, and J.-M. Kim, "An Effective Four-Stage Smoke-Detection Algorithm using Video Images for Early Fire-Alarm Systems," Fire Safety Journal, Vol. 46, No. 5, pp. 276-282, 2011.
- [4] S. Noda, and K. Ueda "Fire Detection in Tunnels Using an Image Processing Method," Vehicle Navigation and Information Systems Conf., pp. 57-62, Japan Highway Public Corp., Osaka, 1994.
- [5] T. Nguyen, M. Kang, Y.-K. Kwon, and J.-M. Kim, "A Study on Effective Fire Detection Algorithm Combining Multiple Heterogeneous Methods", The 2012 International Conf. on Advanced Information Technology and Sensor Application, Vol. 1, No. 1, pp. 19, Sunshine Hotel, Daejeon, 2012.
- [6] S.J. Ham, and B.C. Ko, "Fire-Flame Detection Using Fuzzy Finite Automata," Journal of The Korean Institute of Information Scientists and Engineers, Vol. 37, No. 9, pp. 712-721, Sept. 2010.
- [7] S.H. Lee, "The Design and Implementation of Parallel Processing System using the Nios^(R) II Embedded Processor," Journal of The Korea Society of Computer Information, Vol. 14, No. 11, pp. 97-103, Nov. 2009.
- [8] N. Singhal, J.W. Yoo, H.Y. Choi, and I.K. Park, "Implementation and Optimization of Image Processing Algorithms on Embedded GPU," IEICE Trans. Inf. & Syst., Vol. E95-D, No. 5, pp. 1475-1484, May 2012.
- [9] I.K. Park, N. Singhal, M.H. Lee, S. Cho, and C.W. Kim, "Design and Performance Evaluation of Image Processing Algorithms on GPUs," IEEE

[1] B.Y. Lee, S.T. Park, S.H. Hong, and D.H. Baek, "A Study on the Fire Detection Algorithm for Early Fire Detection of Electrical Fire," 2009 The

- Trans. on Parallel and Distributed Systems, Vol. 22, No. 1, pp. 91-104, Jan. 2011.
- [10] A. Gentile, and D.S. Wills, "Portable Video Supercomputing," IEEE Trans. on Computers, Vol. 53, No. 8, pp. 960-973, 2004.
- [11] T.X. Troung, and J.-M. Kim, "An Early Smoke Detection System based on Motion Estimation", Proceedings of the 5th International Forum on Strategic Technology, pp. 455-458, Ulsan, 2010.
- [12] Y. Dedeoglu, B.U. Toreyin, U. Gudukbay, and A.E. Cetin, "Real-Time Fire and Flame Detection in Video," IEEE International Conf. Acoustics, Speech, and Signal Processing, Vol. 2, pp. 669-672, Ankara, Turkey, 2005.
- [13] J.-W. Choi, M. Kang, and J.-M. Kim, "Implementation of an Optimal SIMD-based Many-core Processor for Sound Synthesis of Guitar," Journal of The Korea Society of Computer Information, Vol. 17, No. 1, pp. 1-10, Jan. 2012.
- [14] H.G. Lee, U.Y. Ogras, R. Marculescu, and N. Chang, "Design Space Exploration and Prototyping for On-chip Multimedia Applications," Proceedings of the 43rd Annual Design Automation Conf., pp. 137-142, 2006.
- [15] G. Healey, D. Slater, T. Lin, B. Drda, A.D. Goedeke, "A system for real-time fire detection", IEEE Computer Vision and Pattern Recognition Conference (CVPR'93), 1993, pp.605-606.
- [16] W. Phillips III, M. Shah, N. da Vitoria Lobo. "Flame Recognition in Video", in Proceedings of the Fifth IEEE Workshop on Applications of Computer Vision, 2000, pp. 224-229.
- [17] C.B. Liu, N. Ahuja. "Vision based fire detection", in IEEE International Conference on Pattern Recognition, vol. 4, 2004, pp. 134-137.
- [18] B.U. Toreyin, Y. Dedeoglu, U. Gudukbay, A.E. Cetin. "Computer vision based method for real-time fire and flame detection", Pattern Recognition Letters, 2005.
- [19] B.U. Toreyin, Y. Dedeoglu, U. Gudukbay, A.E. Cetin. "Realtime Fire and Flame Detection in Video", International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05), 2005.
- [20] J. Zhu, Y. Liu, K. Bao, Y. Chang. "Realtime Simulation of Burning Solids on GPU with CUDA", Proceedings of the 2nd International Conference on Interaction Sciences, 2009, pp. 1335-1340.
- [21] D. Xie, R. Tong, H. Wu. "Multi-channel video-based parallel fire detection acceleration method using multi-cores", in 2010 IEEE 10th International Conference on Computer and Information Technology (CIT), 2010, pp. 1219-1224.
- [22] S.M. Chai, T. Taha, D.S. Wills, and J.D. Meindl, "Heterogeneous Architecture Models for Interconnect-Motivated System Design", IEEE Trans. on VLSI Systems, Vol. 8, No. 6, pp. 660-670, 2000.
- [23] J.C. Eble, V.K. De, D.S. Wills, and J.D. Meindl, "A Generic System Simulator (GENESYS) for ASIC Technology and Architecture beyond 2001," The 9th Annual IEEE International ASIC Conf., pp. 193-196, 1996.
- [24] Y.J. Hur, and S.H. Park, "A Unit-Based Volume Data Compression Scheme Using Daubechies D4 Wavelet Filter," HCI2006 Conference., pp. 1201-1206, Pheonixpark, Gangwon, 2006.
- [25] D.H. Woo, and H.S. Lee, "Extending Amdahl's Law for Energy-Efficient Computing in the Many-Core Era," IEEE Computers, Vol. 41, No. 12, pp. 24-31, 2008.

저 자 소 개



서 준 상
2012: 울산대학교
컴퓨터정보통신공학부 공학사.
현 재: 울산대학교
전기전자컴퓨터공학과 석사과정.
관심분야: SoC설계,
매니코어 프로세서 이키텍처,
임베디드시스템
Email : stoking111@nate.com



강 명 수
2008 : 울산대학교 컴퓨터공학과 학사
2010 : 울산대학교
전기전자컴퓨터공학과 공학석사
2010 - 현재 : 울산대학교
전기전자컴퓨터공학과 박사과정
관심분야 : 임베디드 시스템,
음향신호처리, 고장진단
Email: ilmareboy@gmail.com



김 철 홍
1998 : 서울대학교 컴퓨터공학사.
2000 : 서울대학교 컴퓨터공학부 석사.
2006 : 서울대학교
전기컴퓨터공학부 박사
2005-2007년 : 삼성전자
반도체총괄 책임연구원
2007-현재 : 전남대학교
전자컴퓨터공학부 교수
관심분야 : 임베디드시스템, 컴퓨터구조,
SoC설계, 저전력 설계
Email: cheolhong@gmail.com



김 종 면
1989: 명지대학교 전기공학과 공학사.
2000: University of Florida
전기컴퓨터공학과 공학석사.
2004: Georgia Tech.
전기컴퓨터공학과 공학박사.
현 재: 울산대학교 전기공학부 교수
관심분야: 컴퓨터구조, 병렬프로세서,
멀티미디어 신호처리,
SoC설계
Email : jongmyon.kim@gmail.com