

고속의 클러스터 추정을 위한 매니코어 프로세서의 디자인 공간 탐색

서준상*, 김철홍**, 김종면*

Design Space Exploration of Many-Core Processor for High-Speed Cluster Estimation

Jun-Sang Seo*, Cheol-Hong Kim**, Jong-Myon Kim*

요약

본 논문에서는 단일 명령어, 다중 데이터 처리 기반의 매니코어 프로세서를 이용하여 높은 계산량이 요구되는 차감 클러스터링 알고리즘을 병렬 구현하고 성능을 향상시킨다. 또한 차감 클러스터링 알고리즘을 위한 최적의 매니코어 프로세서 구조를 선택하기 위해 다섯 가지의 프로세싱 엘리먼트 (processing element, PE) 구조 (PEs=16, 64, 256, 1,024, 4,096)를 모델링하고, 각 PE구조에 대해 실행시간 및 에너지 효율을 측정한다. 두 가지 의료 영상 및 각 영상의 세 가지 해상도(128x128, 256x256, 512x512)를 이용하여 모의 실험한 결과, 모든 경우에 대해 PEs=4,096구조에서 최고의 성능 및 에너지 효율을 보였다.

▶ Keywords : 클러스터 추정, 차감 클러스터링 알고리즘, 병렬처리, 매니코어 아키텍처

Abstract

This paper implements and improves the performance of high computational subtractive clustering algorithm using a single instruction, multiple data (SIMD) based many-core processor. In addition, this paper implements five different processing element (PE) architectures (PEs=16, 64, 256, 1,024, 4,096) to select an optimal PE architecture for the subtractive clustering algorithm by estimating execution time and energy efficiency. Experimental results using two different medical images and three different resolutions (128x128, 256x256, 512x512) show that PEs=4,096 achieves the highest performance and energy efficiency for all the cases.

▶ Keywords : Cluster estimation, Subtractive clustering algorithm, Parallel processing, Many-core architecture

•제1저자 : 서준상 •교신저자 : 김종면

•투고일 : 2014. 6. 10, 심사일 : 2014. 7. 13, 게재확정일 : 2014. 9. 17

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

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

※ 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임
(No. NRF-2013R1A2A2A05004566)

I. 서론

최근 의료 영상 기기의 발달로 인해 고 해상도 의료 영상을 이용한 영상 처리 기술의 필요성이 증가하고 있다[1]. 이에 의료 영상의 정확한 분석 및 진단을 위해 고 해상도의 영상을 분할하는 의료 영상 분할 알고리즘의 중요성이 증가하고 있다[2].

대표적인 영상 분할 알고리즘인 퍼지 클러스터링 (fuzzy c-means clustering, FCM) 기법은 같은 클러스터에 속하는 픽셀 정보의 소속 정도를 이용하여 개체 또는 패턴을 분류하는 방법으로 패턴 인식, 이미지 분석, 유전자 분류 분야에서 널리 사용되어져 왔다[3-5]. 하지만 FCM 알고리즘은 초기에 클러스터 개수 및 클러스터 중심 값이 임의적으로 선택되기 때문에 정확한 클러스터링 결과를 보장하지 못하며, 또한 임계치를 만족하기 위해 반복 횟수가 증가하게 된다. 따라서 FCM 알고리즘의 적절한 초기 값을 선택하는 것은 시스템의 성능 향상에 매우 중요하다.

FCM 알고리즘의 초기 값을 계산하는 방법 중에 통계학적 연구에 의해 경험적 또는 선형적으로 얻어진 유효성 지표를 이용한 방법이 제안되어져 왔다[6-8]. 하지만 이와 같은 방법은 특정한 영상에 대한 지표 값을 기준으로 클러스터의 개수를 판별하기 때문에 여러 가지 영상 데이터의 클러스터링 환경에서 보편적으로 적용할 수 없는 문제점이 있다[5].

Chiu가 제안한 차감 클러스터링 알고리즘은 이러한 문제점을 해결하는 동시에 정확한 클러스터의 특징을 제공해 준다[9-10]. 차감 클러스터링 알고리즘은 하나의 픽셀 데이터를 기준으로 영상의 모든 픽셀 데이터와 비교하여 픽셀 데이터들의 밀집도가 가장 큰 클러스터의 중앙값을 구하는 알고리즘이다. 이와 같이 차감 클러스터링 알고리즘은 밀집도 비율과 데이터 군의 관련성을 비교하는 계산이므로 여러 가지 영상 데이터의 클러스터링 환경에 적용할 수 있다[11]. 하지만 하나의 픽셀 데이터를 기준으로 영상의 모든 픽셀 데이터와 비교하는 반복적인 연산을 수행하기 때문에 많은 실행 시간이 요구된다.

이러한 많은 실행 시간의 요구를 만족시키는 대안 중의 하나로 단일 명령어, 다중 데이터 (single instruction, multiple data, SIMD) 기반 매니코어 프로세서가 유망하다[12-17]. SIMD 기반 매니코어 프로세서는 지역성 (locality)이나 규칙성(regularity)이 있는 2차원 패턴의 이미지나 비디오 처리에 있어 최적의 프로세서 구조이다 [15]. 특히 차감 클러스터링 알고리즘은 지역성(locality)이나 규칙성(regularity)이 높으므로 SIMD 기반 매니코어 프로세서

에 적합하다.

본 논문에서는 매니코어 프로세서 상에서 차감 알고리즘을 병렬 구현하여 성능을 향상시킨다. 또한, 본 논문에서는 차감 클러스터링 알고리즘을 위한 최적의 매니코어 프로세서 구조를 탐색하기 위해 총 다섯 가지 PE 구조(PEs=16, 64, 256, 1,024, 4,096)에 대해 실행시간, 시스템 이용률 및 에너지 효율을 측정하였으며, 각 실험은 동일한 28nm 테크놀로지와 1 GHz의 동작 주파수를 사용하였다. 또한 두 개의 의료영상, 세 가지 해상도(128x128, 256x256, 512x512) 해상도 및 클러스터 개수를 달리하여 성능을 분석하였다.

본 논문의 구성은 다음과 같다. 2장에서는 차감 클러스터링 알고리즘 및 매니코어 프로세서 구조에 대한 관련 연구를, 3장에서는 차감 알고리즘의 병렬 구현에 대해 설명한다. 4장에서는 실험환경 및 PE 구조별 실행 시간, 시스템 이용률 및 에너지 효율을 분석하여 최적의 PE 구조를 선택한다. 마지막으로 5장에서는 본 논문의 결론을 맺는다.

II. 관련 연구

1. 차감 클러스터링 알고리즘

차감 클러스터링 알고리즘은 Mountain 클러스터링 알고리즘[18]의 확장된 형태로서 데이터의 밀집도에 의해 클러스터의 수를 구하는 알고리즘이다. 각 데이터는 다른 모든 데이터와의 비교를 통해 거리에 반비례하는 포텐셜 값을 계산하고 가장 큰 포텐셜 값을 갖는 데이터를 클러스터의 중심 값으로 결정한다. 식(1)과 같이 특정 데이터 포인트 x_i 에서 포텐셜 값을 정의한다[19].

$$p_i = \sum_{j=1}^n e^{-\alpha \|x_i - x_j\|^2} \quad (1)$$

위의 식(1)에서 $\alpha = \frac{4}{r_a^2}$ 이며 r_a 는 양의 정수 값이다. 식

(1)에서 정의된 포텐셜 값은 주위의 다른 데이터들의 거리에 따라 달라지며, 보다 가까운 데이터가 많을수록 포텐셜 값이 크다. 모든 데이터의 포텐셜 값을 구한 후 가장 큰 포텐셜을 가진 데이터 포인트를 첫 번째 클러스터 중심 x_1^* 이라 하고 그 포텐셜 값을 p_1^* 이라 한다. 주어진 클러스터 중심 x_1^* 과 포텐셜 p_1^* 값을 이용하여 데이터 포인트 x_1 에서의 포텐셜 값을 다음과 같이 변경한다.

$$p_i = p_i - p_1^* e^{-\beta \|x_i - x_1^*\|^2} \quad (2)$$

이때 $\beta = \frac{4}{r_a^2}$ 이며 양의 상수이다. 새로 갱신한 포텐셜 값은 첫 번째 클러스터 중심에 가까울수록 그 값이 작으며 갱신된 포텐셜 값이 가장 큰 데이터 포인트를 두 번째 클러스터의 중심으로 선택한다. 이와 같은 방법으로 식(3)과 같이 $k+1$ 번째 포텐셜 값은 k 번째 클러스터의 중심 x_k^* 를 이용하여 포텐셜 정도 및 클러스터 중심 값을 구한다[20].

$$p_i = p_i - p_k^* e^{-\beta \|x_i - x_k^*\|^2} \quad (3)$$

Chiu가 제안한 차감 클러스터링의 파라미터 값은 표 1과 같다.

표 1. 차감 클러스터링 알고리즘의 파라미터 값(20)
Table 1. Parameters of the subtractive clustering algorithm

파라미터	차감 클러스터링 알고리즘	
Clusterradius	r_a	(0.25, 0.50)
Sqash factor	η	1.25
Rejctratio	$\underline{\varepsilon}$	0.15
Acceptratio	$\bar{\varepsilon}$	0.5

여기서 r_a 는 데이터 공간 내의 클러스터 반경을 의미하며, $\eta = \frac{r_b}{r_a}$ 를 의미한다. 또한 r_b 는 포텐셜 내의 이웃 데이터 값의 측정된 감축 값을 의미한다. $\underline{\varepsilon}$ 는 클러스터 센터의 최대 허용 값이며 $\bar{\varepsilon}$ 는 최소 허용 값을 의미한다. 새로운 클러스터의 중심 값을 얻기 위한 반복 횟수는 다음과 같은 조건에 의해 정해진다[20].

```

if  $\frac{p_k}{p_1^*} > \bar{\varepsilon}$ 
    accept  $x_k^*$  as a cluster center and
    continue
else if  $\frac{p_k}{p_1^*} < \underline{\varepsilon}$ 
    reject  $x_k^*$  and end the clustering
process.
else
    let  $d_{\min}$  = shortest of the distances
    between  $x_k^*$  and all previously found
    cluster centers.
    
```

```

if  $\frac{d_{\min}}{r_a} + \frac{p_k}{p_1^*} \geq 1$ 
    accept  $x_k^*$  as a cluster
    center and continue.
else
    reject  $x_k^*$  and set the
    potential at  $x_k^*$  to 0.
    Select the data point with
    the next highest potential
    as the new  $x_k^*$  and reset.
end if
    
```

2. SIMD 기반 매니코어 아키텍처

그림 1은 SIMD 기반 매니코어 아키텍처의 블록 다이어그램을 보여준다[15]. 이중 격자구조의 PE 배열과 이를 제어하는 어레이 제어 유닛(array control unit, ACU), 외부 I/O와 PE 로컬 레지스터 간의 데이터 전송을 위한 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.

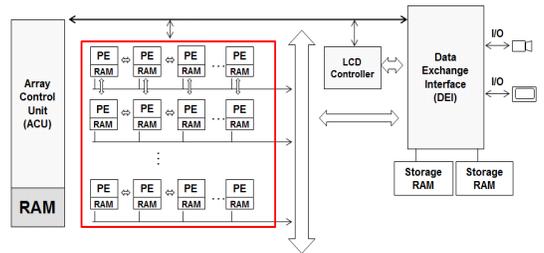


그림 1. SIMD 기반 매니코어 아키텍처
Fig. 1. SIMD-based many-core architecture

그림 2는 단일 PE 구조를 보여주며, 각각의 PE는 다음과 같은 특징을 가진다[15].

- 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 활성 및 비활성 시키는 Sleep 유닛

- 이웃하는 PE들과 데이터 통신을 위한 north-east-west-south(NEWS) 네트워크 및 serial I/O 유닛

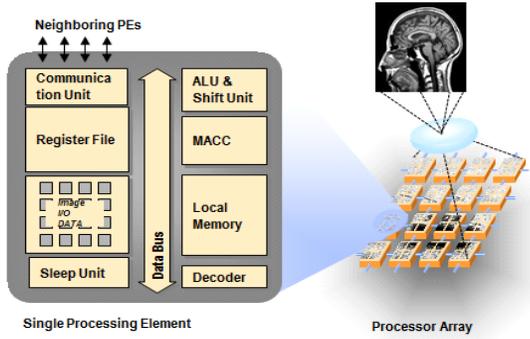


그림 2. 단일 PE 구조
Fig. 2. Single processing element architecture

3. SIMD 기반 영상처리 알고리즘의 병렬 구현

최근 고해상도를 이용한 영상처리 알고리즘은 많은 양의 연산량에 요구하여 실시간 애플리케이션에 적용하기에는 어려움이 있다. 이를 위해 SIMD 기반의 매니코어 프로세서 이용하여 이러한 영상처리에서 요구되는 많은 양의 연산을 효과적으로 수행하여 실시간을 만족시키는 연구가 활발히 진행 중이다[21-27]. 의료 영상을 포함한 특정 영상처리 알고리즘에 대한 최적의 매니코어 구조를 탐색하는 연구[21-24] 분야와 상용 GPU(graphics processing unit)를 이용하여 복잡한 영상처리 알고리즘의 성능을 향상시키는 연구[25-26] 분야

III. 차감 클러스터링 알고리즘의 병렬 구현

차감 클러스터링 알고리즘은 한 픽셀 데이터를 기준으로 전체 픽셀 데이터와의 비교를 통해 가장 큰 밀집도의 포텐셜 값을 구하는 연산을 반복 수행한다. 따라서 본 논문에서는 차감 클러스터링 알고리즘의 반복적인 연산을 매니코어를 이용하여 병렬 구현함으로써 성능 향상을 꾀한다. 본 논문에서 병렬 구현한 차감 클러스터링 알고리즘의 순서도는 그림 3과 같다.

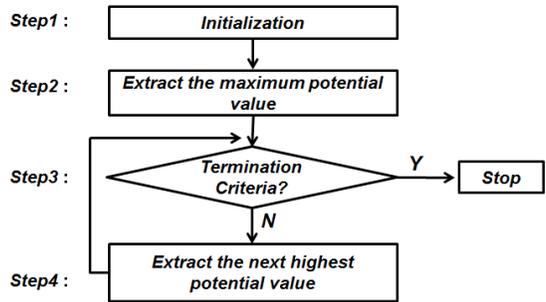


그림 3. 차감 클러스터링 알고리즘의 병렬 구현 순서도
Fig. 3. A flow diagram of a parallel implementation of the subtractive clustering algorithm

- 단계 1 : 차감 클러스터링 알고리즘의 초기화 단계로

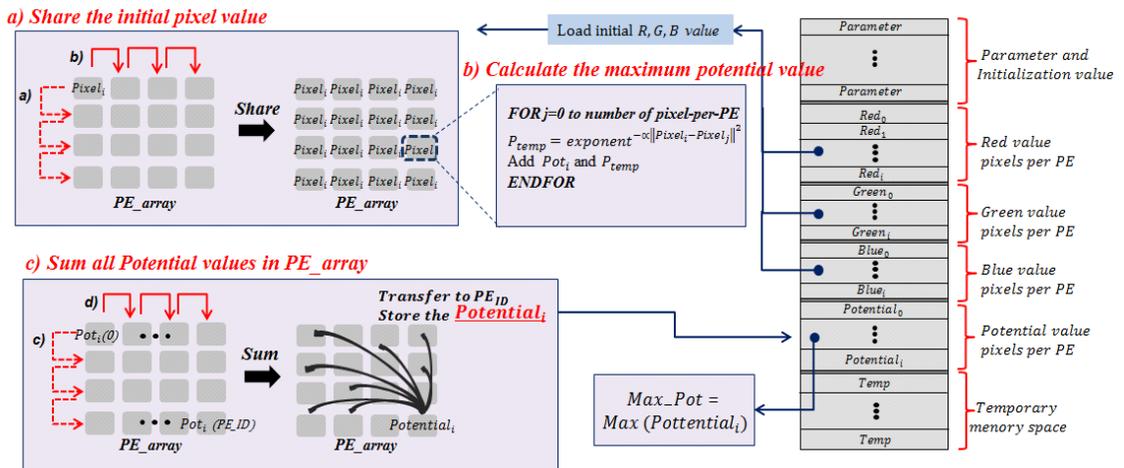


그림 4. 초기 최대 포텐셜 정도 추출단계의 블록도 및 수도코드
Fig. 4. A block diagram and pseudocode of an extraction step for initial maximum potential strength

Chiu가 제안한 파라미터 값 (표 1) 및 이미지 데이터는 각 PE의 메모리에 분배되어 저장되며, 각 PE의 픽셀 데이터는 식(4)와 같이 전체 픽셀의 최소값과 최대값을 통해 동시에 정규화 연산이 수행된다.

$$Pixel_i = \frac{Pixel_i - \min(Pixel_i)}{\max(Pixel_i) - \min(Pixel_i)} \quad (4)$$

- 단계 2 : 초기 포텐셜 최대값 추출 단계에서는 픽셀 데이터 값인 $Pixel_i$ 를 선택하는데 이는 영상에 포함된 모든 픽셀 데이터들을 참조한다. 매니코어 아키텍처의 경우 영상을 PE의 수만큼 분할하여 동시에 수행하기 때문에 PE 간 데이터 통신이 요구된다. 따라서 그림 4(a)와 같이 데이터 전송 명령어 (e.g., XFER NORTH, XFER EAST, XFER WEST, and XFER SOUTH)를 이용하여 픽셀 데이터를 공유한다. 그림 4(b)는 초기 최대 포텐셜 값을 추출하기 위한 연산 과정을 보여준다. 이 과정에서 사용되는 지수함수는 매니코어 프로세서에서 제공되지 않기 때문에 본 논문에서는 shift, branch, arithmetic 명령어 (e.g., LSH, BGE, MUL, ADD, and SUB)를 사용하여 식(5)와 같이 테일러급수를 이용하여 지수함수를 구현한다.

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad (5)$$

테일러급수를 이용한 지수함수 결과는 입력 x 와 차수 n

의 범위에 따라 큰 영향을 미친다. 따라서 본 논문에서는 식(6)과 같이 matlab에서 제공하는 exp함수의 결과와 테일러급수를 이용한 지수함수 결과와의 오차율을 구하여 적절한 입력 값과 차수 값을 구한다.

$$Error = \frac{\exp_{expfunc} - \exp_{taylor}}{\exp_{expfunc}} \quad (6)$$

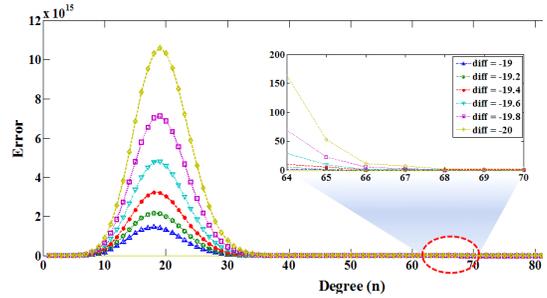


그림 5. 입력과 차수에 따른 exp함수와 테일러급수를 이용한 exp함수의 결과 오차율

Fig. 5. Resultant error rate between an exp. function according to input and degree and an exp. function using Taylor series

그림 5는 입력 값의 범위를 -20 ~ -19까지 0.1씩 증가시키며 비교한 결과이다. 이는 차감 클러스터링 알고리즘에서 테일러급수를 이용한 exp 함수의 입력 값이 -20까지 가장 많이 분포되어 있으며 -20 이하일 경우 exp 출력 값이 0에 가깝기 때문이다. 실험 결과, 가장 오차율이 큰 -20에서 차수 68부

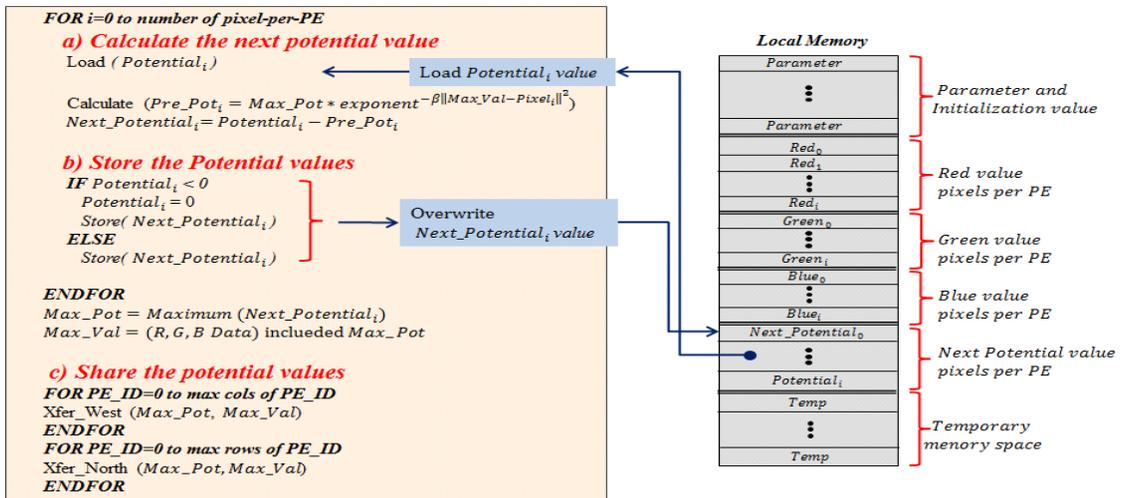
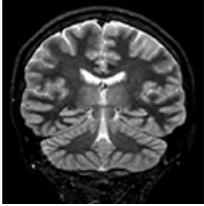


그림 6. 두 번째 최대 포텐셜 정도 추출단계 블록도 및 수도코드

Fig. 6. A block diagram and pseudocode of an extraction step for the second maximum potential strength

표 3. 시스템 파라미터
Table 3. Modeled system parameters

Parameters	Values					
Clock frequency	1GHz					
Interconnection network	Mesh					
Technology	28nm					
Image type and size						
	Image1		Image2			
	Size	Cluster	Size	Cluster		
	128x128	5	128x128	4		
	256x256	5	256x256	3		
512x512	4	512x512	3			
PEs	16	64	256	1024	4096	
Data/PE	128x128	1024	256	64	16	4
	256x256	4096	1024	256	64	16
	512x512	16384	4096	1024	256	64
Memory/PE (KB)	128x128	32.768	8.192	2.048	0.512	0.256
	256x256	131.072	32.768	8.192	2.048	0.512
	512x512	524.288	131.072	32.768	8.192	2.048

터 안정적인 오차율 값을 출력하므로 본 논문에서는 차수를 68로 정하며, 입력 값이 -20 이하일 경우 결과 값을 0 으로 출력한다. 이와 같은 연산은 각 PE에 할당된 영상 데이터 개수만큼 동시에 수행한다. 따라서 식(1)과 같이 모든 픽셀 데이터의 포텐셜 값을 합하기 위해 PE간 데이터 통신이 필요하다. 먼저 특정 픽셀 데이터의 포텐셜 값인 P_{temp} 의 누적 덧셈을 통해 각 PE_ID의 포텐셜 값 Pot_i 를 구한 후 그림 4(c)와 같이 데이터 전송 명령어를 이용하여 모든 PE_ID의 포텐셜 값의 덧셈을 수행한다. 수행 결과 마지막 PE_ID에 전체 픽셀 데이터의 포텐셜 값인 $Potential_i$ 가 저장되므로 모든 PE가 동일한 값을 갖도록 데이터 전송 명령어를 이용하여 전체 PE_ID에 공유한다. 이와 같은 연산은 영상의 크기인 $M \times N$ 만큼 수행하며 최종적으로 $Potential_i$ 값들의 비교 연산을 통해 최대값 Max_Pot 와 이에 해당하는 클러스터 중심값 Max_Val 을 선택한다.

■ 단계 3 : 초기 최대 포텐셜 값 Max_Pot 와 파라미터 값인 $Rejectratio$ 와 $Acceptratio$ 를 이용하여 차감 클러스터링 알고리즘의 반복 횟수를 결정한다. 하지만 초기 최대 포텐셜 값은 마지막 PE_ID에 위치하므로 이를 제외

한 나머지 PE들은 비활성화 시킨 후 연산을 진행한다. 2장 1절의 조건문과 같이 차감 클러스터링 알고리즘은 새로 갱신된 최대 포텐셜 값과 초기 포텐셜의 최대값의 나눗셈 연산을 한 결과가 $Rejectratio$ 이상이 될 때까지 반복적으로 수행 한다.

■ 단계 4 : 갱신할 최대 포텐셜 값은 이전에 계산한 Max_Pot 값과 Max_Val 값을 이용하여 계산한다. 그림 6과 같이 단계 2와 유사한 방법으로 연산을 수행하며, 먼저 로컬 메모리에 저장된 $Potential_i$ 을 불러와 두 번째 최대 포텐셜 값을 구하기 위한 연산을 동시에 수행한다. 그림 6(a)는 단계 2의 연산과는 다르게 이전에 계산된 포텐셜 값 $Potential_i$ 와 새로 계산된 포텐셜 값 Pre_Pot_i 와의 차를 통해 갱신된 포텐셜 값인 $Next_Potential_i$ 를 구한다. 만약 갱신된 포텐셜 값 $Next_Potential_i$ 이 0 이하의 값을 가질 경우 0 값으로 대체하고, 그렇지 않은 경우는 계산된 값을 유지한다. 한편 로컬 메모리의 저장 방식은 그림 6(b)와 같이 이전 포텐셜 값 $Potential_i$ 은 다시 사용하지 않기 때문에 재사용함으로써 메모리 저장 공간을 효율적으로 운영한다. 마

지막으로 갱신한 포텐셜 값의 최대값 Max_Pot 와 해당 하는 포텐셜의 클러스터 중심 값인 Max_Val 을 새로 추출한 후 그림 6(c)와 같이 공유한다.

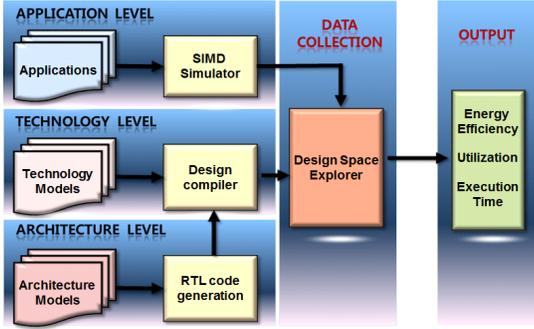


그림 7. 실험 방법론
Fig. 7. Experimental methodology

IV. 실험 결과 및 분석

1. 실험 방법론

그림 7은 매니코어 프로세서의 실행시간, 시스템 이용률 및 에너지 효율을 평가하기 위한 실험 방법론을 보여준다. 실험 방법론은 애플리케이션, 아키텍처 및 테크놀로지 레벨로 구성되어 있다. 애플리케이션 레벨에서는 매니코어 프로세서용 시뮬레이터를 이용하여 차감 클러스터링 알고리즘에 대한 실행 사이클 개수, 동적 명령어 빈도, 시스템 이용률 등의 데이터를 추출한다. 아키텍처 레벨에서는 register-transfer level (RTL) Verilog를 이용하여 모델링된 매니코어 아키텍처를 설계한다. 테크놀로지 레벨에서는 Synopsys Design Compiler 툴과 TSMC 28 nm 공정을 이용하여 아키텍처 레벨에서 설계되어진 RTL 코드를 합성한다. 마지막으로 디자인 공간 탐색기를 이용하여 PE 구조별 실행시간, 시스템 이용률 및 에너지 효율을 측정한다.

표 2. 매니코어 프로세서 성능평가 지표
Table 2. 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 이용률 (System utilization)	$U = \frac{N_{executedPE}}{N_{PE}} \times 100 [\%]$

2. 실험 평가지표

차감 클러스터링 알고리즘을 위한 매니코어 프로세서의 성능 평가를 위해 본 논문에서는 실행 시간(execution time), 에너지 효율(energy efficiency) 및 시스템 이용률(System utilization)을 측정하며, 이는 표 2와 같다. 실행 시간은 차감 클러스터링 알고리즘을 완료하는데 소요되는 시간을 뜻하며, 에너지 효율은 소비된 에너지 당 처리량, 시스템 이용률은 실행시간 동안 전체 PE 수에 대한 실제 동작 하는 PE 수의 비율로 정의된다. 여기서 C 는 사이클 수, f_{clk} 는 클럭 주파수, $Energy$ 는 28 nm 공정에서 소비된 에너지(Joule), $N_{executedPE}$ 는 실행시간 동안 실제 동작한 PE의 수, N_{PE} 는 실행시간 동안 전체 PE의 수를 의미한다.

표 3은 성능 분석을 위해 모델링된 매니코어 프로세서의 시스템 파라미터로써 각 실험은 다섯 가지의 PE 구조 ($PEs=16, 64, 256, 1024, 4096$)에 대해 28nm 테크놀로지와 1GHz 클럭 주파수를 사용하여 수행하였다. 또한 두 개의 의뢰영상, 각 영상마다 세 개의 해상도(128x128, 256x256, 512x512) 이용하여 성능을 분석하였다. 매니코어 프로세서 상에서 차감 클러스터링 알고리즘을 수행하기 위한 각 PE의 로컬 메모리 크기는 식 7과 같이 정의한다.

$$Mem_{PE} = 3 \times (Pixel_{PE}) + Pot_{num} + \alpha \quad (7)$$

여기서 $Pixel_{PE}$ 는 PE당 할당된 픽셀의 수를 의미하며, 3을 곱한 이유는 R, G, B 값들을 분리하여 저장하기 위함이다. 또한 Pot_{num} 는 한 픽셀에 해당하는 포텐셜 값을 의미한다.

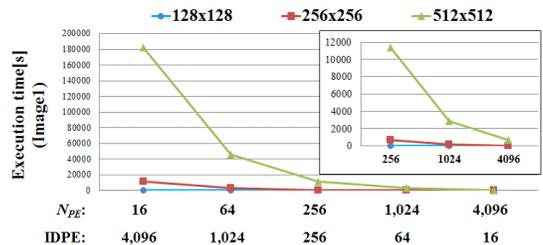


그림 8. 해상도에 따른 영상 1의 PE 구조별 실행시간
Fig. 8. Execution time of different PE architectures for different resolutions of image 1

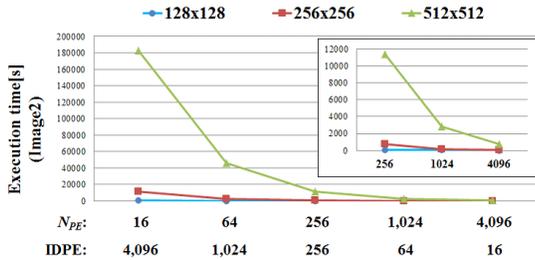


그림 9. 해상도에 따른 영상 의 PE 구조별 실행시간
Fig. 9. Execution time of different PE architectures for different resolutions of image 2

3. 실행시간

차감 클러스터링 알고리즘의 경우 전체 픽셀 값과 각 PE 에 할당된 픽셀 값의 비교 등을 통하여 포텐셜 값을 검출하기 때문에 많은 병렬성을 갖게 되어 그림 8, 9와 같이 사용되는 PE 개수의 증가와 거의 비례하여 실행시간이 감소되는 것을 볼 수 있다. 이는 3장 2단계에서 초기 포텐셜 최대값을 계산할 때 많은 실행시간을 소요하기 때문이다. 3장 2단계에서 설명하였듯이 테일러급수를 이용한 지수함수는 차수의 수만큼 나눗셈 연산을 수행한다. 더불어 초기 최대 포텐셜 값을 구하는 연산 과정은 영상의 모든 픽셀들과 한 PE당 할당된 픽셀들의 배수만큼 반복 연산을 수행하기 때문에 병렬성이 높으며, 다른 단계에 비해 약 90%이상의 실행시간이 소요된다. 따라서 이를 제외한 나머지 단계의 실행시간은 미비할 정도이므로 PE 구조별 실행시간 및 시스템 이용률 등에 큰 영향을 미치지 못한다. 위와 같은 이유로 클러스터 개수가 다른 영상의 경우라도 영상의 사이즈가 같은 경우 서로 비슷한 실행시간이 요구된다. 또한 해상도별 실행시간도 해상도 증가에 따라 실행시간이 약 16배 증가하는 것을 알 수 있다. 하지만 PE 수가 4배씩 증가하지만 실행시간은 4배씩 감소되지 않는 것을 확인할 수 있다. 이는 Amdahl 법칙(Amdahl's law)에 의해 각 PE들이 동시에 실행되는 연산 외에 사용자 정의 함수 호출 및 영상 픽셀을 PE로 분배하는 등 PE 수의 변화와 상관없이 일정한 사이클이 소요되기 때문이다. 또한 초기 픽셀 값의 공유, 각 PE의 초기 포텐셜 값의 덧셈연산을 위한 많은 내부 통신 및 PE 활성화/비활성화 명령어가 수행되어 PE 개수의 증가에 비례하여 실행시간이 감소되지 않는다.

4. 에너지 효율

에너지 효율은 소비된 단위 에너지 당 처리량을 의미하며, 그림 10, 11은 두 가지 의료 영상에 대한 PE 구조별 정규화된 에너지 효율을 보여준다. 세로축은 에너지 효율의 평균값

으로 정규화 되었기 때문에 세로축의 수치보다는 에너지 효율이 나타내는 모양이 중요하다. 에너지 효율은 매니코어 프로세서 상에서 차감 클러스터링 알고리즘을 수행할 때 소비된 에너지와 알고리즘의 실행 시간과 관계가 있다. 그림 10에서 보듯이 PE 수가 증가함에 따라 비례적으로 에너지효율이 증가하는 것을 볼 수 있다. 이는 그림 12(a)와 같이 PE 구조별 에너지의 변화폭은 일정하지만 실행시간은 PE의 수가 증가함에 따라 약 4배씩 감소하기 때문이다. 또한 해상도가 128x128에서 512x512로 증가할수록 더 높은 에너지 효율을 보이는데 이는 해상도가 증가할수록 더 많은 픽셀 데이터를 처리하고, 로컬 메모리의 사이즈 또한 증가하기 때문에 더 높은 효율성을 보인다. 그림 12(b)-(d)에서 보듯이 각 해상도마다 PE 구조별 에너지 변화가 다른 것을 확인할 수 있다. 이는 PE 수가 증가할수록 PE 활성화/비활성화 및 PE간 내부 통신 명령어를 더 많이 요구하여 에너지 증가 폭이 커지기 때문이다. 그림 12(b)에서 보듯이 128x128 해상도의 경우 PE당 할당된 픽셀 데이터의 양이 작기 때문에 PE 수가 256까지는 에너지가 감소하다 PE 수가 1024부터는 증가하는 것을 알 수 있다. 그림 12(c)와 같이 해상도가 256x256인 경우 PE당 할당된 픽셀 데이터의 양이 증가하여 에너지는 PE 수가 1024까지 감소하다 증가하는 형태를 보인다. 마지막으로 그림12(d)에서는 위와 같은 이유로 PE 구조별 에너지가 지속적으로 감소하는 것을 볼 수 있다.

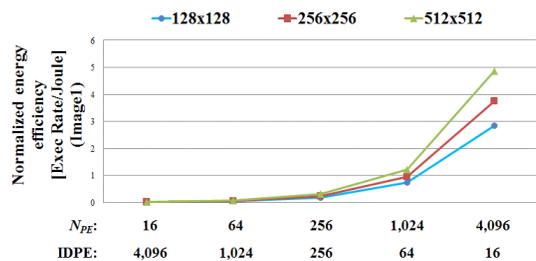


그림 10. 해상도에 따른 영상 1의 PE 구조별 정규화 된 에너지효율
Fig. 10. Normalized energy efficiency of different PE architectures for different resolutions of image 1

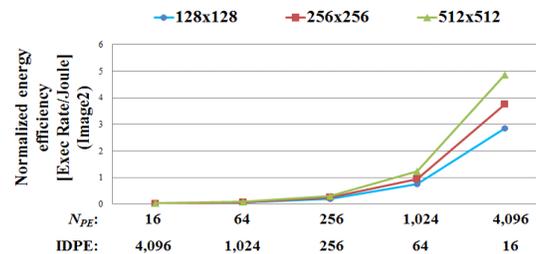


그림 11. 해상도에 따른 영상 2의 PE 구조별 정규화 된 에너지효율
Fig. 11. Normalized energy efficiency of different PE architectures for different resolutions of image 2

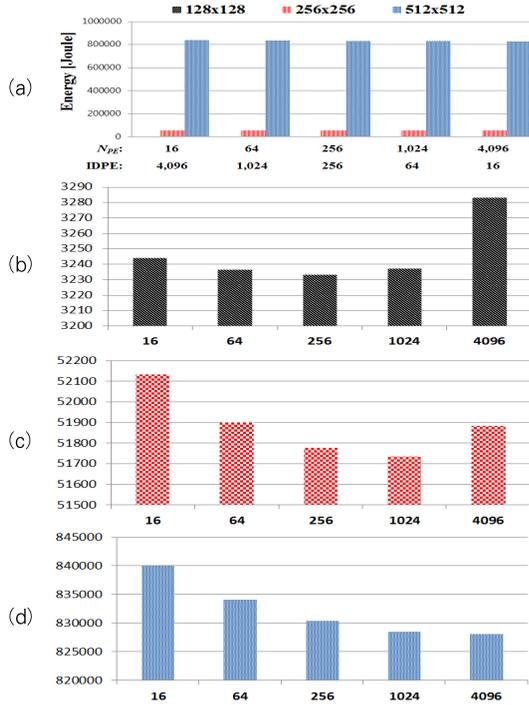


그림 12. 37지 해상도에 따른 영상 1의 PE 구조별 에너지 비교, (a) 전체 해상도, (b) 128x128, (c) 256x256, (d) 512x512
 Fig. 12. Energy comparison of different PE architectures for different resolutions of image 1. (a) all resolutions, (b) 128x128, (c) 256x256, (d) 512x512

5. 시스템 이용률

그림 13은 해상도에 따른 영상 1의 시스템 이용률을 보여 준다. 4.3절에서 설명하였듯이 PE 개수가 증가함에 따라 PE에 할당된 픽셀 연산에서 포텐셜 값 연산은 산술-논리 연산 명령어만으로 이루어져 PE 수가 256까지는 시스템 이용률이 크게 변하지 않는다. 하지만 PE 수가 256부터는 PE에 할당

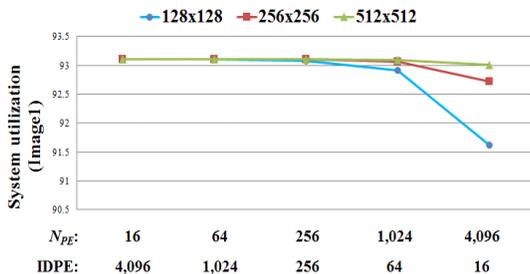


그림 13. 해상도에 따른 영상 1의 PE 구조별 시스템 이용률
 Fig. 13. System utilization of different PE architectures for different resolutions of image 1

된 데이터의 수가 점점 줄어들고 초기 픽셀 값 공유, 각 PE의 포텐셜 값의 덧셈 연산을 위한 내부 통신 및 PE 활성화/비활성화 명령어가 픽셀을 처리하는 명령어보다 더 큰 비중을 차지하기 때문에 시스템 이용률은 감소하게 된다. 위와 같이 해상도가 128x128에서 512x512로 증가할 경우 PE에 할당된 데이터의 수가 증가하기 때문에 시스템 이용률의 변화는 완만해 지게 된다.

그림 14는 해상도에 따른 영상 2의 시스템 이용률을 보여 준다. 그림 13과 비교해서 그래프의 곡선 및 수치가 거의 유사함을 알 수 있다. 이는 4.3절에 언급하였듯이 영상 1과 영상 2는 비슷한 실행시간을 보이며 PE 증가에 따른 내부 통신 및 PE 활성화/비활성화 명령어 수가 비슷하기 때문에 유사한 곡선을 보인다.

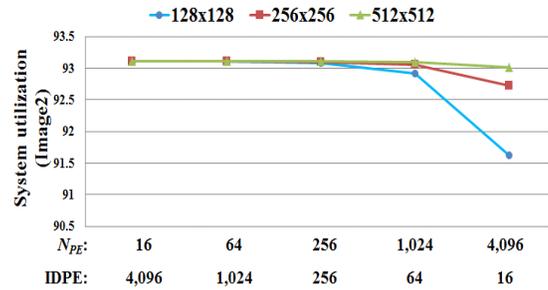


그림 14. 해상도에 따른 영상 2의 PE 구조별 시스템 이용률
 Fig. 14. System utilization of different PE architectures for different resolutions of image 2

6. 상용 GPU와의 성능 비교

본 절에서는 제안하는 매니코어 프로세서와 현재 널리 사용되고 있는 상용 고성능 GPU(NVIDIA Geforce GTX 560)와의 성능을 비교한다. 상용 GPU와의 정확한 비교는 적절하지 않지만 본 절의 목적은 상용 GPU와의 비교를 통해 선택한 최적의 매니코어 프로세서의 잠재가능성을 알아보기 위함이다. 성능 비교를 위해 256x256 크기의 의료영상을 사용하였으며 동일한 클러스터 추정 알고리즘 구현하여 실행시간을 측정하였다. 표 4는 상용 GPU와의 성능 비교 결과를 보여준다. PE 개수가 4,096개인 매니코어 프로세서는 상용 GPU보다 약 4.4배 더 높은 성능을 보였다.

표 4. 매니코어 프로세서와 상용 GPU와의 성능 비교
Table 4. Performance comparison between the many-core processor and a commercial GPU

	Units	GPU	Many-core processor
Number of core		336	4,096
Clock frequency	[GHz]	1.620	1
Execution time	[ms]	195.387	44.9

V. 결 론

본 논문에서는 매니코어 프로세서를 이용하여 높은 연산량을 요구하는 차감 클러스터링 알고리즘을 병렬 구현하고 성능을 향상시켰다. 또한 차감 클러스터링 알고리즘을 위한 최적의 매니코어 프로세서 구조를 탐색하기 위해 다섯 가지 PE 구조(PEs=16, 64, 256, 1,024, 4,096)를 모델링하고 이에 대한 실행시간, 시스템 이용률 및 에너지 효율을 측정하였다. 두 가지 의료영상 및 각 영상마다 3가지 해상도(128x128, 256x256, 512x512)를 사용하여 모의 실험한 결과, PEs=4,096구조에서 최고의 성능 및 에너지 효율을 보였다. 또한 선택된 매니코어 구조는 상용 GPU보다 약 4.4 배 더 높은 성능을 보였다.

향후 본 연구를 실제 FPGA 보드에 구현하여 본 논문의 결과를 검증할 계획이다.

참고문헌

- [1] M. Smelyanskiy, D. Holmes, J. Chhugani, A. Larson, D. M. Carmean, D. Hanson, P. Dubey, K. Augustine, D. Kim, A. Kyker, V. W. Lee, A. D. Nguyen, L. Seiler, R. Robb, "Mapping High-Fidelity Volume Rendering for Medical Imaging to CPU, GPU and Many-Core Architectures," *IEEE Trans. on Visualization and Computer Graphics*, Vol. 15, No. 6, pp. 1563-1579, 2009.
- [2] S. Krinidis, V. Chatzis, "A Robust Fuzzy Local Information C-Mans Clustering Algorithm," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1328-1337, 2010.
- [3] J. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithm*. New York: Plenum, 1981.
- [4] D. Pham, "An Adaptive Fuzzy C-Means Algorithm for Image Segmentation in the Presence of Intensity Inhomogeneities," *Pattern Recognition Letters*, vol. 20, pp. 57-68, 1999.
- [5] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone and J. C. Phillips, "An Initialization Method for Fuzzy C-means Algorithm using Subtractive Clustering," *Proceedings of IEEE*, vol.96, no.5, pp. 879-899, 2010.
- [6] R. J. Cho, M. Huang, M. J. Campbell, H. Dong, L. Steinmetz, L. Sapinoso, et al, "Transcriptional Regulation and Function during the Human Cell Cycle," *Nature Genetics* article, vol. 27, pp. 48-54, 2001.
- [7] Y. I. Kim, D. W. Kim, D. Lee, K.H. Lee, "A Cluster Validation Index for GK Cluster Analysis based on Relative Degree of Sharing," *Information Sciences*, vol. 168, pp. 225-242, 2004.
- [8] Y. Okada, T. Sahara, H. Mitsubayashi, S. Ohgiya, T. Nagashima, "Knowledge-assisted Recognition of Cluster Boundaries in Gene Expression Data," *Artif. Intell. Med.*, vol. 35, pp. 171-183, 2005.
- [9] S.L. Chiu, "Fuzzy Model Identification based on Cluster Estimation," *Journal of Intelligent and Fuzzy Systems*, vol. 2, pp. 267-278, 1994.
- [10] S.L. Chiu, "Extracting Fuzzy Rules from Data for Function Approximation and Pattern Classification," *Fuzzy Information Engineering: a Guide Tour of Applications*, pp. 149-162, 1997.
- [11] Z.h. Sun, "Study on Subtractive Clustering Video Moving Object Locating Method with Introduction of Eigengap," in the 9th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 609-612, 2012.
- [12] S.H. Lee, "The Design and Implementation of Parallel Processing System using the Nios^(R) II

- Embedded Processor," The Korea Society of Computer and Information, vol. 14, no. 11, pp. 97-103, Nov. 2009.
- [13] A. Gentile and D. S. Wills, "Portable Video Supercomputing," IEEE Trans. on Computers, vol. 53, no. 8, pp. 960-973, 2004.
- [14] Y.H. Kim and J.M. Kim, "Design Space Exploration of Optimal many-Core Processors for Discrete Wavelet Transform," Journal of Institute of Embedded Engineering of Korea, vol. 7, no. 5, pp. 277-284, 2012.
- [15] Y.M. Kim and J.M. Kim, "Design and Verification of High-Performance Parallel Processor Hardware for JPEG Encoder," Journal of Institute of Embedded Engineering of Korea, vol. 6, no. 2, pp. 100-107, 2011.
- [16] S. Sonntag, and F. Gilibert, "Design Space Exploration and Performance Evaluation at Electronic System Level for NoC-based MPSoC," IEEE/ACM International Conf. Computer-Aided Design, pp. 336-339, 2010.
- [17] 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.
- [18] R. Yager, D. Filev, "Generation of Fuzzy Rules by Mountain Clustering," Journal of Intelligent and Fuzzy Systems., vol. 2, no. 3, pp. 209-219, 1994.
- [19] S.G. Park, S.J. Choi, "Modeling of Left Ventricular Assist Device and Suction Detection Using Fuzzy Subtractive Clustering Method," Korea Intelligent Information System Society, vol. 22, no. 4, pp. 500-506, 2012.
- [20] R. Qun, L. Baron, and M. Balazinski, "Type-2 Takagi-Sugeno-Kang Fuzzy Logic Modeling using Subtractive Clustering," Fuzzy Information Processing Society., pp. 120-125, 2006.
- [21] B.-K. Choi, J.-M. Kim, "Implementation of Multi-Core Processor for Beamforming Algorithm of Mobile Ultrasound Image Signals," Journal of The Korea Society of Computer and Information, vol. 18, no. 2, pp. 1-8, 2011.
- [22] S.-M. Kang, J.-M. Kim, "Multimedia Extension Instructions and Optimal Many-core Processor Architecture Exploration for Portable Ultrasonic Image Processing," Journal of The Korea Society of Computer and Information, vol. 17, no. 8, pp. 1-10, 2012.
- [23] J.-Y. Kim, D.-G. Son, J.-M. Kim, H.-S. Jeon, "Parallel Implementation and Performance Evaluation of the SIFT Algorithm Using a Many-Core Processor," Journal of The Korea Society of Computer and Information, vol. 18, no. 8, pp. 1-10, 2013.
- [24] J.-S. Seo, M.-S. Kang, C.-H. Kim, J.-M. Kim, "Design Space Exploration of Embedded Many-Core Processors for Real-Time Fire Feature Extraction", Journal of The Korea Society of Computer and Information, vol. 18, no. 10, pp. 1-12, 2013.
- [25] I.-K. Jung, J.-S. Seo, M.-S. Kang, C.-H. Kim, J.-M. Kim, "Implementation and Performance Analysis of Fuzzy C-Means Algorithm Using GPGPU," Journal of Korean Institute of Next Generation Computing, vol. 9, no. 5, pp. 27-37, 2013.
- [26] S.-H. Yi, Y.-S. Woo, B.-N. Jang and Y.-M. Yi, "Efficient Local Binary Pattern Based Face Recognition Using OpenCL on the Embedded GPU," Journal of Korean Institute of Information Scientists and Engineers, vol. 40, no. 6, pp. 257-265, 2013.

저 자 소 개



서 준 상
2013: 울산대학교
컴퓨터정보통신공학부 공학사.
현 재: 울산대학교
전기전자컴퓨터공학과 석사과정.
관심분야: 병렬프로세서 구조,
임베디드시스템,
화재감지 알고리즘
Email : Siberiaj00@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
1995: 명지대학교 전기공학과 공학사.
2000: University of Florida
전기컴퓨터공학과 공학석사.
2005: Georgia Tech
전기컴퓨터공학과 공학박사
현 재: 울산대학교 전기공학부 교수
관심분야: 임베디드 SoC, 병렬처리.
Email : jmkim07@ulsan.ac.kr