

고화질 영상에서 고속 안개 제거를 위한 SIMD 구조에 적합한 병렬메모리

이 형*

A Parallel Memory Suitable for SIMD Architecture Processing High-Definition Image Haze Removal in High-Speed

Hyung Lee*

요 약

Dark channel prior를 이용한 안개제거 알고리즘으로 만족할만한 연구결과가 발표된 이후로 이 알고리즘의 처리 속도를 높이기 위한 많은 연구들이 진행되었다. 이들 중에서 median dark channel prior를 이용한 알고리즘이 주목을 받고 있지만 여전히 낮은 처리속도의 한계를 갖고 있다. 그래서 본 논문에서는 고화질 영상에서 고속 안개 제거를 위한 SIMD 구조에 적합한 병렬메모리 모델을 제안한다. 제안하는 병렬메모리 모델은 n개의 화소들에 동시에 접근할 수 있으며, 3, 5, 7 또는 11의 크기를 갖는 4가지 종류의 median filter를 위한 간격들을 허용한다. 그래서 충분한 데이터 대역폭을 지원하기에 median dark channel prior를 이용한 알고리즘을 고속으로 처리할 수 있다.

▶ Keywords : 안개제거, Median Dark Channel Prior, 병렬메모리, SIMD 구조

Abstract

Since the haze removal algorithm using dark channel prior was introduced, many researches for improving processing speed have been addressed even if it presented impressive results. Remarkable one is using median dark channel prior. Although it has been considered as a very attractive method, processing speed is as low as ever. So, a parallel memory model which is suitable for SIMD architecture processing haze removal on high-definition images in high-speed is

•제1저자 : 이형 •교신저자 : 이형

•투고일 : 2014. 5. 14, 심사일 : 2014. 6. 14, 게재확정일 : 2014. 7. 4.

* 대전보건대학교 방송콘텐츠과(Dept. of Broadcasting Contents, Daejeon Health Sciences College)

※이 논문은 2012년도 대전보건대학교 교내연구비 지원에 의한 논문임.

introduced in this paper. The proposed parallel memory model allows to access n pixels simultaneously. It is also support stride 3, 5, 7, and 11 in order to execute convolution mask operations, e.g., median filter. The proposed parallel memory model can therefore support enough data bandwidth to process the algorithm using median dark channel prior in high-speed.

▶ Keywords : Haze Removal, Median Dark Channel Prior, Parallel Memory, SIMD Architecture

I. 서 론

최근 들어 실외영상을 처리하는 분야들이 늘어나고 있다. 대표적인 응용분야로 자동차 분야에서는 굵은 날씨에서 전방 확보 및 감지, 많이 보급되고 있는 전자광학 분야에서의 디지털카메라, 그리고 군사 분야에서의 해저탐사 등이 있겠다. 특히, 실외에서 촬영한 영상의 경우에는 연무 등의 다양한 외부 요인에 의해 영상 내의 객체가 가지고 있는 원래의 밝기나 색상 정보를 획득하기에 많은 어려움이 따른다. 그래서 좋은 결과를 얻기 위해 본처리 과정 전에 전처리 과정을 거치는데, 연무가 있는 환경에서 촬영한 영상을 청명한 기후 환경에서 획득한 영상과 유사하게 만드는 방법 등이 최근의 영상처리 분야에서 연구 주제들 중의 하나로 간주되고 있다.

한 장의 영상에서 안개를 제거하는 알고리즘[1-4]이 소개된 이후로 이들을 토대로 많은 후속 연구들이 진행되고 있는데[5-9] 이들 후속 연구들은 물리적 자원의 효율적인 활용 개선으로 처리 속도를 향상 시키거나, 특정한 환경에 종속적인 영상의 결과를 향상 시키려는 등의 두 가지로 분류될 수 있다. 전자에 해당하는 연구들 중에서 그나마 빠른 처리속도를 보이는 Gibson의 논문[7]에서는 400x600 영상을 처리하는데 소요되는 시간은 대략 1.2초 정도라고 기술하였다. 응용분야가 방송인 경우 1초에 29.97장(또는 23.976장)의 영상을 처리해야 하는데, SD급인 경우는 영상의 크기가 720x480이며, HD급인 경우에는 최고 1920x1080의 크기이다. 또한 고급디지털카메라의 경우는 4928x3264 크기의 영상을 초당 10장 이상 촬영할 수 있다. 앞선 연구들의 결과가 일반적인 상황에서 활용할 수 있는 깨끗한 영상을 제공하고 있지만 1초에 최소 15장에서 최대 30장을 처리해야 하는 분야에서는 처리시간의 단축이 필수적이다. 다양한 응용분야

에서 활용되려면 처리시간이 충분히 개선되어 이를 위한 실시간 처리가 가능한 단일부품으로의 개발이 필요하다.

영상/비디오처리 분야 외에도 방대한 양의 데이터를 실시간으로 처리하기 위한 고성능 컴퓨터에 대한 연구가 프로세서와 메모리 시스템, 그리고 프로그래밍 관점 등에서 활발히 진행되고 있다. 프로세서의 경우에서 볼 때, 중앙처리장치는 다중코어(multicore)로, 그래픽처리장치는 다수의 코어(many-core)로 집중되면서 특정 분야에 한정되어 연구되던 병렬처리구조, 특히 SIMD(Single Instruction Multiple Data) 구조에 대한 관심이 재차 대두되면서 활발히 연구되어 관련된 범용 제품들이 출시되고 있다. 또한 다수의 처리기들(processing elements)로 구성된 컴퓨터의 경우에는 처리기들이 처리할 방대한 양의 데이터를 지원하기 위해 메모리 대역폭을 확보하는 병렬메모리에 대한 연구들이 병행되고 있다. 또한 이러한 구조가 지원할 수 있는 최고의 성능을 확보하기 위해 다양한 연구들이 진행되고 있다[10].

고성능 컴퓨팅을 위한 데이터 대역폭을 확보하기 위한 메모리시스템들은 처리기들이 동시에 접근하도록 하는 것이다. 다수의 메모리 모듈들로 구성된 메모리시스템을 구축하는 방법 중의 하나로 메모리 모듈들에 대한 연속적인 메모리 접근 요청을 중첩시키는 것이다. 하지만 이러한 방법은 다수의 메모리 모듈들에 동시에 접근할 수 없기 때문에 SIMD 구조에 적합하지 않다. SIMD 구조를 위한 단순한 메모리 시스템은 접근하려는 데이터의 주소(a)를 메모리 모듈의 주소($a \% m$)에 매핑시키는 것이다. 여기서 m 은 메모리 모듈들의 개수로서 처리기들의 개수(n)와 일치할 때 최상의 성능을 발휘할 수 있으며, 구현상의 복잡도를 고려하여 2의 승수를 취하였다. 그러나 이러한 인터리브드 방식의 메모리시스템들은 일정한 간격과 다수의 접근형태들이 요구되는 상황에서는 다수의 메모리 모듈들에 동시에 접근할 경우에는 충돌이 발생하여 이를 해결하기 위한 추가된 기능 등으로 낮은 성능을 보였다.

이를 해결하기 위해 비선형 비틀림 (non-linear skewing) 메모리시스템에 대한 연구가 진행 되었지만, 이들 역시 한정된 접근형태들과 간격들, 그리고 접근 위치를 제한시켰다. 초기의 선형 비틀림 (linear skewing) 메모리 시스템[11]이 제안되면서 많은 연구자들이 이를 토대로 다양한 접근형태들과 임의의 간격을 지원하기 위한 연구들이 진행되었다 [12-14]. 최근에는 앞서 언급했던 메모리시스템들의 장단점들을 고려하여, 특정 응용에 적합한 컴퓨터 시스템들을 개발하면서 그 응용에 최적화된 메모리시스템에 대한 연구들도 진행되고 있다. 예를 들면, 특정 접근형태에 임의의 간격들만을 허용하고[15], 충돌 발생이 감지되면 직렬 접근을 수행하며 [16], 파이프라인 기법을 접목하여 계산된 주소를 큐로 관리하여 충돌을 회피하는[17] 방법들 등이 제시되었다.

이에 본 논문에서는 기존의 안개 제거를 위한 여러 연구들의 난제였던 고용량의 메모리 요구와 반복된 마스크 처리에 적합한 SIMD 구조를 위한 병렬메모리 모델을 제안한다. 이를 위해 제2장에서는 관련연구를, 제3장에서는 이를 위한 병렬메모리 모델을, 제4장에서는 병렬메모리를 포함한 SIMD 구조의 구현에 대한 고려사항들을 기술한다.

II. 관련 연구

다양한 날씨 조건에서도 깨끗한 영상을 얻기 위해 여러 장의 영상을 이용하거나 외부 정보를 활용하는 방법들이 제안되었으나, 추가적인 정보없이 하나의 영상만 존재할 경우에 대한 연구들이 최근에 제시되고 있다. 채도의 분포를 통해 화질을 개선하여 했던 R. Tan의 연구[1]은 과도한 채도 증가 현상 및 결과 영상에 후광이 생성되는 등의 단점이 존재했고, R. Fattal의 연구[2]는 안개가 짙은 영역에서는 깨끗한 영상을 얻을 수 없다는 단점을 갖고 있었다. 그러나 He 등이 제안한 dark channel prior를 이용한 안개제거 알고리즘[4]가 앞선 연구들에 비해 매우 깨끗한 결과 영상을 얻을 수 있으나, 낮은 처리속도와 고용량의 메모리 등이 단점으로 언급되었다.

일반적으로 안개가 있는 영상의 모델은 Narsimhan[18] 등이 처음 제안하여 여러 안개제거 알고리즘들에 사용되었는데, 이 모델 식은 다음과 같다.

$$I(x) = J(x)t(x) + A(1-t(x))$$

여기서 $I(x)$ 는 획득한 영상, A 는 안개의 밝기 정도, $t(x)$ 는 전달량으로 빛이 산란되지 않고 렌즈까지 도달된 정도를 나타낸다. $J(x)$ 는 원 영상의 밝기 값이기에 안개 제거는 입력

영상 $I(x)$ 로부터 A 및 $t(x)$ 와 $J(x)$ 를 구하는 것이다. $t(x)$ 는 1보다 작은 값을 가지므로 획득한 영상의 밝기는 안개가 없는 깨끗한 영상과 안개의 깊이의 조합으로 볼 수 있다

이러한 모델을 토대로 He 등이 제시한 방법[4]은 대략 4가지 단계를 거치는데, 1) 안개의 밀도를 예측 (dark channel prior), 2) 전달량 $t(x)$ 계산, 3) 후광 효과를 제거하기 위한 전경과 후경을 매팅(matting), 4) 전달량과 안개 값, 그리고 원 영상로부터 $J(x)$ 를 계산하여 안개가 제거된 결과영상을 얻게 된다. 이들 4가지 단계들 중 3단계에서 적용되는 매팅기법이 많은 메모리를 사용하며, 전반적으로 300x225 크기의 영상을 처리하는데 대략 10~20초 정도의 처리시간이 소요된다[4,5].

이러한 문제점들을 해결하기 위해 매팅기법을 계층적 cross bilateral filter 또는 가우시안 필터로 대체함으로써 어느 정도 개선을 보였으며[5,6], 기본적인 마스크 연산을 최소값 대신 중간값으로 대체함으로써 매팅 단계를 생략한 방법 [3,7] 등과 같이 메모리의 사용량을 낮추면서 처리 속도를 높이는 연구들이 진행되었다. 또한 이들의 방법론을 토대로 안개 강도맵을 구성하여 안개의 양을 예측하여 적용하거나[8], 영상의 에지 정보를 이용하여 전달량을 보전하는[9] 등 특정 상황에 따른 결과영상의 개선에 대한 연구들도 진행되었다. 하지만 Gibson[7] 방법조차도 400x600 크기의 이미지를 처리하는데 대략 1.2초 정도 소요된다고 언급하고 있다.

이렇듯 실시간 영상처리가 필요한 분야에서는 아직도 영상의 크기와 화소단위의 반복되는 연산들로 인하여 공간 복잡도와 시간 복잡도를 일정하게 유지하기 쉽지 않다. 하지만 화소단위로 반복되는 동일한 연산들의 규칙성으로 병렬도를 확보할 수 있기에 SIMD 구조를 갖는 하드웨어 구조 및 데이터의 대역폭을 지원하기 위한 병렬메모리에 대한 연구들이 지속적으로 연구되고 있다. 비록 He 등의 알고리즘을 병렬화시켜 발표된 논문[19]가 있지만, 이 역시 매팅 단계에서의 메모리 관리에 자유롭지 못했다.

이에 본 논문에서는 median dark channel prior를 토대로 어느 정도 결과 영상의 화질 유지 및 속도가 향상된 Gibson[7]의 방법을 토대로 고화질 영상에서 고속 안개 제거를 위한 SIMD 구조에 적합한 병렬메모리 모델을 제안하고자 한다. 이 방법은 화소 단위로 반복적인 median filter를 적용하는 것이기에 이 필터를 고속으로 처리하는 것이 전반적인 처리 속도 향상을 위한 주요한 요지라 할 수 있는데, 이를 위한 개별적인 연구들이 알고리즘 측면에서 지속적으로 진행되어 왔으며 이 중에서도 SIMD기반의 병렬처리에 대한 연구 [20]도 포함되어 있다.

III. 제안하는 병렬메모리 모델

본 논문에서 제안하는 병렬메모리 모델은 사용자 관점에서의 논리적인 접근모델과 실제 메모리 구성에 따른 물리적인 접근모델로 분리시켜 고려할 수 있다. 이를 위해 처리할 영상을 "가로의 길이 x 세로의 길이 y 채널수로 고려할 경우, 논리적인 접근모델은 $LM(x,y,ch)$ 로 나타낼 수 있으며, ch 는 RGB 중의 하나가 된다. 즉, 이는 특정 채널의 2차원 (x,y) -좌표계 내의 특정 화소를 가리키게 된다.

물리적인 접근모델은 접근하려는 특정 채널의 화소가 어느 메모리 모듈의 어떤 주소의 셀에 해당하는지를 결정해야 한다. 그렇기 때문에 물리적인 접근모델은 $PM(M,A)$ 로 나타낼 수 있으며, 이는 논리적인 접근모델에서 지정한 특정 채널(ch)의 특정 좌표(x,y)에 해당하는 화소 $LM(x,y,ch)$ 가 M 번째 메모리 모듈의 A 번째 주소에 해당하는 메모리 셀로 매핑된다는 것을 의미한다.

또한 병렬메모리 모델은 SIMD 구조를 고려하여 n 개의 처리기들이 동시에 n 개의 화소에 접근해야 한다. 만약 이들 n 개의 화소들이 동일한 메모리 모듈에 저장된다면 이들 화소들에 접근할 때 충돌을 일으켜 병렬성을 유지하지 못한다.

논리적인 접근모델을 물리적인 접근모델과 맵핑하기 위해서는 논리적인 접근모델의 인자들을 토대로 물리적인 접근모델의 인자들을 유추해 낼 수 있어야 하며, 또한 n 개의 화소들이 병렬성을 갖고 동시에 접근될 수 있어야 하기에 논리적인 접근모델에서 화소(P1)의 좌표와 화소(P2)의 좌표가 다르고, 이들이 물리적인 접근모델에서 같은 메모리 모듈에 저장되어 있다면 이들 화소들의 메모리 모듈 내의 주소는 달라야 한다. 또한 이들 화소들은 동시에 접근되어서도 안된다. 이러한 조건들을 토대로 $LM(x,y,ch,t) = PM(M,A)$ 를 만족시키는 M 과 A 를 결정하는 함수들이 제시되어야 한다.

이에 논리적인 접근모델에서 고려하는 메모리의 구성이 가로의 길이는 W 로, 세로의 길이를 H 로, RGB 각각으로 구성된 $W \times H \times 3$ 인 3차원 메모리로 간주하고, 동시에 접근할 화소들의 개수를 n 이라고 가정한다. 그리고 메모리 모듈들의 개수를 m 이라고 할 때, m 이 n 보다 큰 소수이면 소수의 모듈러 계산을 통해 $LM(x,y,ch) = PM(M,A)$ 를 만족시키는 메모리 모듈 M 은 수식 1에 의해서, M 번째 메모리 모듈 내의 메모리 셀의 주소 A 는 수식 2에 의해 각각 계산된다.

$$M(x,y,ch) = (q \times x + y + p \times q \times ch) \% m \quad (1)$$

$$A(x,y,ch) = x/(p \times r) + s1 \times (y/q) + s2 \times ch \quad (2)$$

여기서 p, q, r 은 디자인 인자들이며, $p \times q \times r = n$ 이라는 공식으로 정립시켜 n 이 동시에 접근할 수 있는 화소들의 개수이며 처리기들의 개수가 된다. 그리고 $s1 = \lceil W/pr \rceil$ 이고 $s2 = s1 \times \lceil H/q \rceil$ 의 특성을 갖게 되면 주소의 중복 없이 최대 n 개의 화소들이 동시에 접근될 수 있는 병렬성을 제공할 수 있다.

예를 들어, 논리적인 접근모델에서 고려하는 메모리의 크기가 $32 \times 32 \times 3$ 이고, $p = q = r = 2$ 이라고 가정한다면, n 은 8, m 은 11, 그리고 $s1$ 과 $s2$ 는 각각 8와 128이 된다. 즉, 물리적인 접근모델은 11개의 메모리 모듈들로 구성되고 동시에 접근되는 화소들의 개수는 8개가 된다. 즉, 11개의 메모리 모듈들 중에서 8개를 선택하여 8개의 화소들에 동시에 병렬로 접근한다는 것을 의미한다.

상기 수식 1과 수식 2를 확장시켜 동시에 접근할 수 있는 n 개의 화소들을 위한 메모리 모듈들과 그에 해당하는 주소들은 다음의 수식 3과 4에 의해 각각 계산된다.

$$(M(x,y,ch) + qk) \% m \quad (3)$$

$$A(x+k,y,ch), 0 \leq k < n \quad (4)$$

식 1을 토대로 $M(x+k,y,ch)$, $0 \leq k < n$ 을 고려할 때, 변수 k 가 k/q 로 대체되어 식 3이 도출될 수 있다.

앞선 예를 토대로, $LM(0,0,Red=0)$ 에서 8개 화소들의 좌표들 (0,0), (1,0), (2,0), (3,0), (4,0), (5,0), (6,0), (7,0)에 해당하는 모듈과 주소의 쌍들은 수식 3과 수식 4에 의해서 (0,0), (2,0), (4,0), (6,0), (8,1), (10,1), (1,1), (3,1)로 계산되며, 이들이 서로 중복되지 않기에 동시에 8개의 화소들에 접근할 수 있다. 하지만 물리적인 접근모델은 메모리 모듈들의 배열이 고정되어 있기 때문에 라우팅이 필요한데 이는 수식 5에 의해 결정된다.

$$Reg2((qk) \% m) \Leftarrow Reg1(k), 0 \leq k < n. \quad (5)$$

상기 모듈과 주소의 쌍들을 수식 5를 통해 라우팅을 하게 되면 (0,0), (1,1), (2,0), (3,1), (4,0), x, (6,0), x, (8,1), x, (10,1)로 배열이 바뀌게 된다. 이 때 5번과 7번 및 8번 메모리 모듈들('x'에 해당하며 이는 *don't care*를 의미함)는 설계 시 disable 상태가 되게 하여 동시에 접근할 경우에 충돌이 발생하지 않도록 한다. 이러한 라우팅은 메모리 쓰기 연산인 경우이며, 메모리 읽기 연산의 경우에는 라우팅

패턴이 반대가 된다.

연속적으로 반복 수행되는 median filter와 같은 마스크 연산의 경우, 앞서 언급된 알고리즘들에서 요구하는 마스크의 크기가 3x3, 5x5, 7x7, 11x11로 다양하기 때문에 간격이 3, 5, 7, 11이 허용되는 병렬메모리 모델이 제시되어야 한다. 이를 위해 논리적인 접근모델은 $LM(x, y, ch, t)$ 로 재정의 되어야 하며, t 는 1, 3, 5, 7, 11 중의 하나의 값을 갖는다. 예를 들어 $LM(0, 0, Red = 0, 3)$ 이고 동시에 접근되는 화소의 개수가 $n=4$ 인 경우에, Red 채널에서 영상 내에서 기준 좌표 (0,0)을 토대로 3의 간격을 갖는 (3,0), (6,0), (9,0)의 좌표들을 포함하여 모두 4개의 화소들이 동시에 접근되는 것을 의미한다. 이를 위해 $t > 0$ 이고 $t \% m \neq 0$ 라는 조건하에서 간격이 1, 3, 5, 7, 11을 만족하는 n 개의 메모리 모듈들과 주소들, 그리고 라우팅을 위한 함수들은 수식 3-5를 토대로 다음과 같이 수식 6-8로 각각 재정의 될 수 있다.

$$(M(x, y, ch) + qtk) \% m \tag{6}$$

$$A(x + kt, y, ch) \tag{7}$$

$$Reg2((qtk) \% m) \Leftarrow Reg1(k), 0 \leq k < n. \tag{8}$$

결과적으로 수식 3, 4, 5는 상기 수식 6, 7, 8에서 $t = 1$ 인 경우에 해당한다.

상기 예제들과 동일한 상황에서 Red 채널의 기준화소 좌표가 (1,1)이고 3의 간격으로 8개의 화소들에 동시에 접근하기 위한 논리적인 접근모델은 $LM(1, 1, 0, 3)$ 이며, 이에 해당하는 물리적인 메모리 모듈과 해당 주소의 쌍들은 수식 6과 7에 의해서 (3,0), (9,1), (4,1), (10,2), (5,3), (0,4), (6,4), (1,5)와 같이 계산되고, 라우팅은 수식 8에 의해서 (0,4), (1,5), x, (3,0), (4,1), (5,3), (6,4), x, x, (9,1), (10,2)와 같이 계산된다. 이렇게 함으로써 하나의 처리기가 3x3 마스크를 처리하는 동안 나머지 처리기들도 동시에 3의 간격을 갖는 서로 다른 화소들에 대해 동일한 연산을 수행할 수 있다. 즉, 처리기의 개수가 8개이면 8개의 마스크 연산을 동시에 수행하게 된다는 것을 의미하며 그렇기에 본 논문에서 제안하는 병렬메모리 모델에서 $n(=pqr)$ 을 어떻게 설정하느냐가 성능측면에서 최대 관건이라 할 수 있다.

IV. 설계 시 고려사항들

앞서 언급했듯이 전반적인 성능 측면을 고려한다면, 제안하는 병렬메모리 모델과 직접적으로 연결되는 처리기들의 개

수를 결정하는 것이 제일 중요한 관건이라 할 수 있다. 동시에 접근할 데이터들의 개수(m)를 결정하고 이에 따라 디자인 파라미터인 p, q, r 을 결정하는 것이다. 물론 논리적인 접근 모델에서 전체적인 3차원 메모리 구조의 크기($W \times H \times 3$)도 결정해야 한다. 여기에서 이론상으로는 n 의 개수를 최소 8개부터 고려할 수 있지만 n 에 의해서 메모리 모듈의 개수인 m 도 결정되기 때문에 $m-n$ 의 값이 작을수록 메모리 모듈의 활용도를 높일 수 있는데, 이는 $(A(W-1, H-1, 2) + 1) \times (m-n)$ 만큼의 메모리 셀들이 전혀 사용되지 않기 때문이다. 이는 비틀림 구조(skewing scheme)가 갖는 탄생적인 문제이지만, 이러한 구조이기에 간격을 갖고 충돌없이 n 개의 데이터에 동시에 접근할 수 있다. 2차원 병렬메모리 구조에서 이러한 사용되지 않는 메모리 셀들을 처리기들의 지역 버퍼로 활용하는 연구가 추가적으로 진행되고 있다.

SIMD 구조는 기본적으로 처리기들을 제어하는 하나의 제어기(control unit)와 여러 개의 처리기들, 그리고 메모리 시스템으로 구성될 수 있다. 본 논문에서 제안하는 병렬메모리 모델 자체만을 설계함에 있어서 앞서 언급했던 고려사항들로 설계하더라도 추가적으로 제어기와 처리기들을 설계함에 있어 이러한 병렬메모리의 특성을 고려하여 해야 한다. 즉, 설계한 병렬메모리의 읽기주기(read cycle)와 쓰기주기(write cycle)를 고려해야 하는데, 이는 병렬메모리를 어떻게 구성하여 설계하는냐에 달려 있다. 일반적으로 CPU가 직접적으로 주메모리에 접근하는 것과 달리 병렬메모리는 복잡한 연산이 필요한 메모리 모듈(M) 및 주소(A) 계산과 데이터 및 주소의 라우팅을 위한 네트워크가 필요하다. 아무리 여러 개의 데이터에 동시에 접근할 수 있다고 하더라도 설계가 미흡하면 충분한 대역폭을 확보할 수 없게 된다. 이를 위해서 처리기들은 메모리참조 명령들과 논리연산 등의 기타 명령들이 분리되어 동시에 수행할 수 있는 구조로 설계되어야 한다. 아울러 SIMD 구조 차원에서 제어기와 처리기들 및 병렬메모리가 파이프라인(pipe-lined) 방식으로 설계 되어야 하고 또한 병렬메모리가 모듈별로 구성되어 자체적으로 파이프라인 방식으로 동작하도록 설계하는 것이 본 논문에서 제안하는 병렬메모리 모델의 성능을 최대도로 높일 수 있을 것이다.

우선적으로 본 논문에서 제안하는 병렬메모리 모델을 어떻게 설계하느냐에 대한 기본적인 고려가 필요하다. 설계를 위해서 다양한 방법들이 거론될 수 있지만 기본적인 사항은 다음과 같다.

- $\lceil \log_2(W/p + 4WH/pqr) \rceil \times d$ -bit 크기를 갖는 m 개의 메모리 모듈들
- 간격 t 를 위한 3-bit 레지스터와 ch 를 위한 2-bit 레지

스터, 그리고 x 와 y 좌표를 지정하기 위한 레지스터들로 크기는 각각 $\lceil \log_2 W \rceil - bit$ 과 $\lceil \log_2 H \rceil - bit$

- 화소 데이터를 메모리 모듈들에 읽고 쓰기 위한 m 개의 t -bit 레지스터들
- 화소 데이터를 n 개의 처리기에 전송하기 위한 n 개의 t -bit 레지스터들
- m 개의 메모리 모듈들에서 n 개를 선택하는 모듈
- n 개의 선택된 메모리 모듈에 해당하는 n 개의 주소를 계산하는 모듈
- n 개의 데이터와 주소를 라우팅하는 모듈
- 전반적인 메모리 구조를 제어하고 처리기들과의 인터페이스 부분 등이 기본적으로 고려될 수 있다.

접근하려는 화소들에 관련된 메모리 모듈들과 주소를 계산하는 부분들이 곱셈과 나눗셈, 그리고 모듈러 연산들이 포함되어 있어 하드웨어 구현 시 시간복잡도와 공간복잡도가 높을 수 있다. 그러나 요즘 나오는 셀 라이브러리들이 최적화 되어 있어 그리 문제가 되지 않을 수 있지만 이들 함수들 역시 한정된 범위 내에서 반복적인 연산값을 갖기에 이들을 미리 계산하여 Latch나 SRAM 또는 ROM 등에 저장함으로써 해결할 수 있다.

예를 들어 n 개의 메모리 모듈들을 선택하는 수식 6과 수식 7의 경우는 특정 간격 t 에 대해 수식 9와 수식 10과 같이

수정함으로써 반복되는 상수값들을 미리 결정할 수 있다.

$$(M(0,0,0) + qk \times t) \% m, \tag{9}$$

$$A_t(a, 0, 0) + A(x, y, ch). \tag{10}$$

여기서 $a = x \% pr$ 이며 $0 \leq a < pr$, $0 \leq k < n$ 이다.

지금까지 기술한 고려 사항들을 토대로 본 논문에서 제안하는 병렬메모리 모델의 기능 블록도를 그림 1과 같이 구성할 수 있으며, 크게 메모리모듈 선택 모듈, 주소 계산 모듈, 데이터 및 주소 라우팅 모듈, 전체 수행 상태를 제어하기 위한 모듈, 그리고 m 개의 메모리모듈 등으로 구성할 수 있다.

모듈러 연산과 모듈러 연산의 결과로 m 개의 메모리 모듈들 중에서 n 개의 메모리 모듈들을 선택하는 부분은 연산 회로 구현상의 복잡도를 피하기 위해서 ROM1과 ROM2에 수식 9의 중간값들을 미리 계산하여 저장할 수 있다. 이는 미리 예측할 수 있는 계산값들이 반복되기 때문이며, ROM3과 그 아래의 모듈들은 주소 계산을 위한 수식 10을 위한 것이다. 또한 허용할 간격들이 결정 되었고, 라우팅을 제어할 데이터는 m 모듈러 연산의 결과(ROM1의 출력)이기 때문에 이들을 조합하여 데이터 및 계산된 주소들의 라우팅을 위해 Reordering 모듈과 Barrel Shifter를 구성하여 수식 8을 수행하도록 하였다.

SIMD 구조에서 제안하는 병렬메모리와 연계되는 n 개의 처리기들을 구성함에 있어 이들 처리기들이 메모리접근연산과 산술논리연산이 독립적임을 고려하여 파이프라인 방식으로 구현한다면 좀 더 성능을 높일 수 있을 것이다. 그래서 그림 1에서 좌측 상단의 State Machine을 추가하여 제안하는 병렬메모리 모델이 파이프라인 방식으로 동작할 수 있도록 제어함으로써 추후에 파이프라인 방식의 SIMD 구조에 적합한 메모리시스템으로써 활용될 수 있다.

이러한 고려사항들 중에서 파이프라인 방법을 적용하지 않고 본 논문에서 제안하는 병렬메모리 모델을 그림 1과 같은 구성으로 하드웨어 기술언어인 Verilog로 구현하고 기능 모의실험을 통해 읽기연산과 쓰기연산이 각각 최대 4개와 2개의 시스템 클럭 내에서 데이터의 안정성을 보장하는 것을 확인하였다.

추가적으로 마스크 연산의 특징이 이전에 읽었던 데이터를 재사용한다는 점을 고려하여 처리기들간의 데이터 전달을 허용하면 메모리 접근 회수를 크게 줄일 수 있게 된다. 예를 들어 $LM(0,0,0,3)$ 인 경우 PE0은 (0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2)를 읽게 되고 PE1은 (3,0), (3,1), (3,2), (4,0), (4,1), (4,2), (5,0),

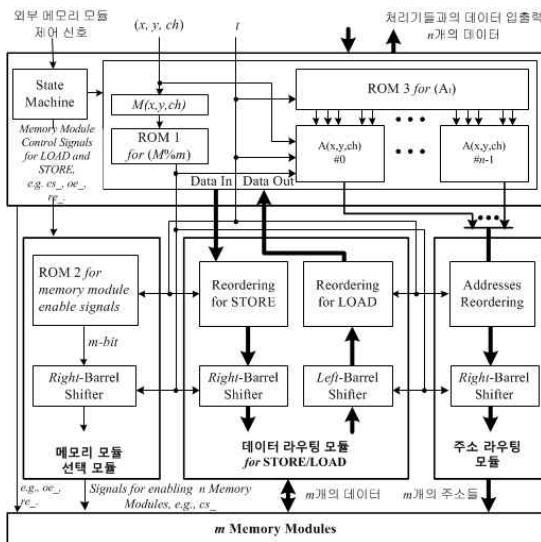


그림 1. 설계 시 고려사항들을 반영한 제안하는 병렬메모리 모델의 기능 블록도

Fig. 1. Functional block diagram of the proposed parallel memory model considering design issues

(5,1), (5,2)을 읽게 된다. 각각 읽은 9개의 화소들에 대한 데이터를 처리하고 난 후 일반적으로 인접 화소의 처리를 위해 $LM(1,0,0,3)$ 연산을 수행하게 되는데, 이 경우에는 PEO가 이전에 읽었던 데이터들 중에서 후반 6개는 동일하며 PE1이 이전에 읽었던 데이터들 중에서 전반 3개를 다시 읽어야 한다. 이러한 메모리접근 대신 PE들 간의 데이터 전송을 고려한다면 좀 더 빠른 처리속도를 갖도록 구현할 수 있지만 이는 알고리즘에 매우 종속적인 부분이라 할 수 있겠다.

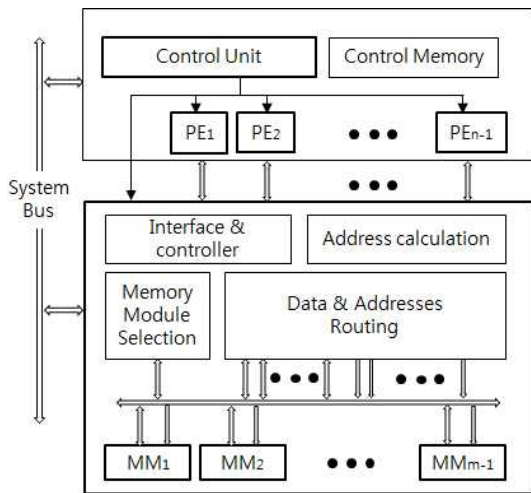


그림 2. 제안하는 병렬메모리를 활용한 시스템 구현의 예
 Fig. 2. An example of the implementation of a system using the proposed parallel memory

고려사항의 마지막으로 제어기(CU)와 n 개의 처리기들(PEs) 및 병렬메모리 시스템을 하나의 칩으로 구현할 것인지 아니면 기능별로 여러 개의 칩들로 나누어 구현할 것인지를 결정해야 한다. 참고로, 제안하는 병렬메모리를 포함한 전반적인 SIMD 구조를 갖는 시스템 구현을 위한 구성은 그림 2와 같이 제시될 수 있겠다. 이를 결정하기 위한 고려사항은 서로 물리적으로 분리되어 구성되는 m 개의 메모리 모듈들이다. 만약 이들이 하나의 칩에 집적될 수 없다면 m 개의 메모리 모듈들과 연결되는 데이터 버스와 주소 버스 및 메모리 제어 신호들을 위한 외부 핀들이 많다는 것이 단점이다. 만약 하나의 칩에 집적될 수 없다면 적용할 칩에 맞추도록 조정해야 하는데, 예를 들어 n 이 64라면 n 이 8인 병렬메모리를 8개로 구성하는 방법이 있다. 하지만 이런 방법은 앞서 설명한 바와 같이 64개의 단일 구성일 경우보다 더 많은 메모리 모듈들을 요구하게 되고 이로 인해 사용되지 않는 메모리 셀들을 더 많이 갖게 된다. 왜냐하면 $n=64$ 인 병렬메모리인 경우에는 메모리

모듈의 개수가 67이지만 $n=8$ 인 병렬메모리를 8개로 구성한다면 $88(=11 \times 8)$ 이 되기 때문이다. 구현에 있어서 개발할 환경에 따라 고려해야 할 많은 변수들이 있지만 본 논문에서 제안하는 병렬메모리모델은 그만큼 확장성을 지원할 수 있다.

본 논문에서 제안한 병렬메모리 모델은 SIMD 구조 내의 메모리시스템으로써 주변 구성 모듈들과 어떻게 상호연동 되도록 설계 되는냐에 따라 성능 측면에서 다양한 변화가 예측된다. 하지만 기본적으로 간격을 갖고 n 개의 화소들에 동시에 접근할 수 있다는 측면에서 처리할 데이터의 대역폭을 확보하기에 처리속도를 향상시킬 수 있다.

V. 결론

다양한 날씨 조건에서도 추가적인 정보없이 하나의 영상만으로 깨끗한 결과 영상을 만들어내는 일련의 연구들은 요즘처럼 대기 환경이 좋지 않은 상황에서 유용한 응용분야가 아닐 수 없다. 하지만 일련의 연구들을 살펴보면 처리 시간과 고용량의 메모리 요구 등으로 인해 실용화되기 위해서는 추가적인 연구들이 필요한 실정이다.

이에 본 논문에서는 화소별로 동일 연산들을 반복으로 수행한다는 점과 이를 위한 병렬구조로서 적합한 SIMD 구조를 위해 충분한 대역폭을 지원할 수 있는 병렬메모리 모델을 제안 하였으며 이를 설계할 경우에 고려해야 할 다양한 사항들에 대해서 제시하였다. 물론 구현 관점에서 볼 때 더 많은 연구들과 설계 기술들이 수반되어야 하겠지만 집적도가 높은 디바이스의 출현이 예측되는 상황에서 본다면 구현에 있어 크게 무리가 없으리라 예측된다.

또한 병렬 median filter에 대한 연구가 지속되고 있지만 이는 특정 하드웨어 구조에 종속적인 것이기에 본 논문에서 제안하는 일반적인 SIMD 구조를 위한 병렬메모리의 특성을 고려한 병렬 median filter에 대한 연구가 추가적으로 필요하며, 병렬 메모리의 접근 방식을 좀 더 다양화시켜 다양한 영상처리와 3차원 그래픽/이미지 응용분야에 접목시키는 연구가 향후 진행되어야 할 것이다.

참고문헌

[1] R. Tan, "Visibility in bad weather from a single image," Proc. CVPR, pp. 1-8, Alaska, USA, June 2008.
 [2] R. Fattal, "Single image dehazing," ACM Trans.

- Graphics, Vol. 27, No. 3, pp. 1-9, Aug. 2008.
- [3] J. P. Tarel and N. Hautière, "Fast Visibility Restoration from a Single Color or Gray Level Image," ICCV 2009, Kyoto, Japan
- [4] K. He, J. Sun, and X.Tang, "Single image haze removal using dark channel prior," in Proc. CVPR, pp. 1956-1963, Miami, USA, June 2009.
- [5] Jin-Hwan Kim, Chang-Su Kim, "Hierarchical Haze Removal Using Dark Channel Prior," Journal of KIEE, Vol. 5, No. 2, pp. 457-464, Feb. 2010.
- [6] Sangwook Kim, Dongwon Shin, "Improvement of dehazed image using optimal mask size of Gaussian Filter," ICROS 2012, Vol. 41, pp. 254~255, Apr. 2012.
- [7] K. Gibson, D. Vo, and T. Nguyen, "An Investigation of Dehazing Effects on Image and Video Coding," IEEE Transactions on Image Processing, Vol. 21, Issue. 2, pp. 662-673, Feb. 2012
- [8] Uhm, Tae-Ha, et. al, "Algorithm of haze removal using color analysis and compensation," Fall Conf. of KOSBE, pp. 19~22, Dec. 2012.
- [9] An, Jin-Woo, et. al, "Enhancement of haze removal using transmission compensation," Fall Conf. of KOSBE, pp. 19~22, Dec. 2012.
- [10] D. B. Kirk and Wen-mei W. Hwu, Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann Pub., 2010.
- [11] P. Budnik and D. J. Kuck, "The Organization and Use of Parallel Memories," IEEE Trans. Computers, Vol. 20, No. 12, pp. 1566-1569, Nov. 1971.
- [12] D. C. Van Voorhis and T. H. Morrin, "Memory System for Image Processing," IEEE Trans. Computers, Vol. 27, No. 2, pp. 113-125, Feb. 1978.
- [13] D. H. Lawrie and C. R. Vora, "The Prime Memory System for Array Access," IEEE Trans. Computers, Vol. 31, No. 5, pp. 435-442, May 1982.
- [14] J. W. Park, "An Efficient Buffer Memory System for Subarray Access," IEEE Trans. Parallel and Distributed Systems, Vol. 12, No. 3, pp. 316-335, Apr. 2001.
- [15] Eero Aho, Jarno Vanne, and T. D. Hamalainen, "Parallel Memory Architecture for Arbitrary Stride Access," In Proc. of the IEEE Workshop on Design and Diagnostics of Electronic Circuits and System, pp. 65-70, Apr. 2006.
- [16] J. K. Tanskanen and T. Pitkanen, "Parallel Memory Architecture for TTA Processor," Proc. of the 7th Int. Conf. on Embedded Computer Sys.: Architecture Modelling, and Simulation," pp. 273-283, 2007.
- [17] Dionysios Reisis and Nikolaos Vlassopoulos, "Conflict-free Parallel Memory Accessing Techniques for FFT Architecture," IEEE Trans. on Circuits and Systems, Vol. 55, No. 11, pp. 3438-3447, Nov. 2008.
- [18] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, "Instant dehazing of images using polarization," in Proc. CVPR, pp. 1984-1991, Hawaii, USA, Dec. 2001.
- [19] Hyung Lee, "Parallel System for Single Image Haze Removal," International Conference on Convergence Content 2012 (ICCC 2012), Vol. 10, No. 2, pp.207-208, Dec. 2012.
- [20] Kolte, P., Smith, R., and Su, W., "A Fast Median Filter Using Altivec," (ICCD '99) International Conference on Computer Design, pp. 384-391, Oct. 1999

저자 소개



이 형

1995: 충남대학교
컴퓨터과학과 이학사.

1997: 충남대학교
컴퓨터공학과 공학석사.

2005: 충남대학교
컴퓨터공학과 공학박사.

현 재: 대전보건의대학교
방송콘텐츠과 조교수

관심분야: 영상/비디오 처리

Email : hyung@hit.ac.kr