

# PVM을 이용한 컴퓨터비전 플랫폼

김경남, 고종국, 김상호, 서영진, R.S.Ramakrishna

광주과학기술원 정보통신공학과

## Computer Vision Platform using PVM

Kyung-Nam Kim, Jong-Guk Ko, Sang-Ho Kim, Young-Jin Suh and R S Ramakrishna  
Department of Information and Communications  
Kwang-Ju Institute of Science and Technology(K-JIST)

### 요 약

컴퓨터 비전은 많은 계산을 요구하는 작업으로 구조적인 계산작업(low-level vision)과 미구조적 계산작업(high-level vision)을 가지고 실시간 처리를 요구한다. 이러한 점에서 비전 작업의 병렬처리와 그것들의 구현에 대한 스케줄링 schemes이 본 논문에서 중요시 된다. 그리고 PVM이 동작하는 저가의 네트워크로 연결된 워크스테이션 클러스터상에서 구현될 알고리즘을 구현하고 제안된 아이디어는 실용적인 예 (eye location from image sequence)를 들어서 보였다. 저대대의 멀티미디어 환경은 이러한 고성능의 컴퓨팅 플랫폼을 사용하리라 기대된다.

### 1. 서 론

컴퓨터 비전 응용은 공장에서의 물품검사, 지동 시물 추적(automatic object tracking), 문자 인식 그리고 의학분야에서의 이미지 처리 등 여러 분야에서 이용되고 있다. 이러한 응용분야는 일반적으로 실시간 처리를 필요로 하고, 게다가 비전 작업은 매우 복잡하고 많은 계산을 요구한다. 이러한 점에서 병렬처리는 중요한 위치를 차지한다. 지금까지 병렬 비전 환경은 몇몇 학교 및 연구소에 의해 연구되어 왔다 [1,2,3]. 또한 비전 작업의 특성상 low, intermediate, high 레벨로 나뉘어져 각기 다른 계산 구조상의 특성들을 가진다.

본 논문의 구성으로서, 2장은 병렬비전환경과 heterogeneous networked computing 에 대해 설명한다. 3장에서는 병렬 알고리즘과 스케줄링 schemes에 대해 설명되고, 4장에서는 실험 결과를 보여준다. 그리고, 마지막에 결론과 앞으로의 과제에 대해 설명한다.

### 2. 병렬 비전 환경

#### 2.1 병렬비전환경 및 요구사항

병렬 구조를 갖는 몇몇 시스템이 비전 알고리즘 개발의 속도 향상의 관점에서 제안되어 왔다 [1,2,3]. 다음은 그들 중 몇몇 예이다.

- Paragon (Cornell Univ), Warp (Carnegie Mellon Univ), DISC (Purdue Univ), IUA (Univ of Massachusetts), etc

- 이들 시스템들은 data parallel 문제들의 효과적인 알고리즘을 개발하는데 도움을 준다.
- 단점이라면 매우 고가이며, 특정 아키텍처에 의존한다는 것이다.

병렬 비전 환경에서의 요구사항들은 다음과 같다.

- 이상적으로 병렬 비전 환경은 여러 시스템 아키텍처를 지원해야 한다. 경제적 측면에서 현재 사용하는 하드웨어를 그대로 사용하는 것이 바람직하며 그들은 transparent한 방법으로 다양한 상황에서 기능을 제공해야 한다.
- Task management, parallel time optimization, and load balancing 통해 이들은 유용성을 높인다 [5].

#### 2.2 Heterogeneous Networked Computing PVM

Parallel Virtual Machine (PVM)은 연구과제의 부실률로서 고성능의 과학 계산 소프트웨어 개발자들에게는 매우 잘 알려져 있다. PVM은 서로 다른 기종의 컴퓨터들을 하나의 병렬 가상머신으로 사용할 수 있게 해준다 [6].

그리고 PVM으로 이기백척화 데이터포맷은 transparent한 방법으로 다루어 질 수 있다. 그러나 예상할 수 없는 computational speed, machine load, 그리고 network load는 확실히 고려되어야 해서 사용자의 소프트웨어 개발자는 이러한 문제를 내치하는 어떤 전략을 구상해야 한다. 그리고 PVM은 완전한 분산 시스템을 지원하지 않는다. 원래로 Single system image의 개념으로 location transparency가 보장 되지 못한다. 이러한 연구는 분산시스템 분야에서 매우 활발히 진행되고 있다. 대략적인 제안된 병렬 비전 환경의 각 layer들을 그림 1에서 보여 준다.

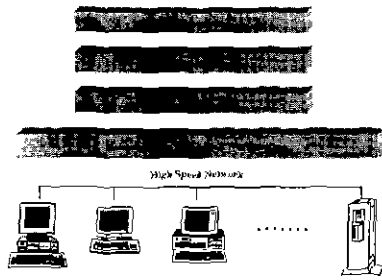


그림 1 Heterogeneous Networked Computing Environment

### 3. 병렬 알고리즘과 작업 스케줄링

#### 3.1 비진 오퍼레이션의 분류

이 레이어들은 [5]에서 볼 수 있고 전반적인 작업의 분류를 설명하는데 충분하리라 생각된다

	Local operators		Global operators
	pixel	mask	
Data-independent	Arithmetic Multiple operand Single operand Complex, arithmetic Nonlinear functions Trigonometry Bitwise operators Comparison logic Data conversion Format conversion Classification	2D convolution	FFT, DCT Hadamard Statistics Size operators Histogram Shape analysis Fractal analysis
Data-dependent	Feature analysis Edge-linking Eigen values / Eigen vectors		

#### 3.2 스케줄링 schemes

이 schemes들은 그림 2,3에서 보일 것이다 두 가지 schemes인 local scheme과 global scheme은 잘 알려져 있다 local 스케줄링을 이용해 각각의 오퍼레이션들이 최고의 성능을 얻기 위해 최적화 될 수 있다. 또한, 몇몇 작업들로 이루어진 global job은 task graph와 Gantt chart로 스케줄링될 수 있다 그리고, 쉽게 열려서, 작업들은 프로세서들에게 더디나리하게 분산된다

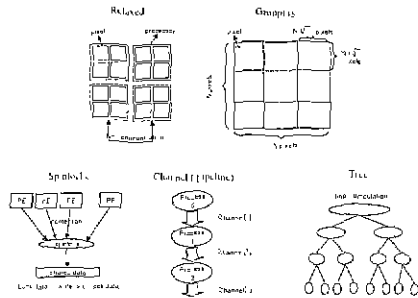


그림 2 스케줄링 Scheme 1

Dynamic load balancing은 중앙집중식(centralized) 완전분산식(fully distributed), 반 분산식(semi-distributed)으로 분류될 수 있다 중앙 집중식 load balancing에서 global

state 정보는 알고리즘이 프로세서들에게 작업을 나누는 작업을 잘 할 수 있게 한다 하지만 이 방법은 확장성에 문제가 된다 완전 분산식 load balancing 은 프로세서들이 주변 프로세서들과 정보를 교환 할 수 있게 해준다 이것의 잇점은 스케줄링 오버헤드가 적지만 중앙 집중식만큼 워크로드가 잘 분산되지 않는다. 그리고 반 분산식은 위의 두 schemes의 결함이나 여기에 더하여 유용한 load balancing과 합리적인 task런리 가 더 좋은 효율성을 위해 포함되어야 한다 [5]

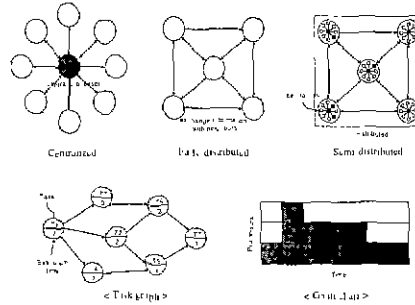


그림 3 스케줄링 Scheme 2

각각의 오퍼레이션은 자신의 계산 특성에 일치하는 특정 schemes을 요구한다 3.1에 비진 오퍼레이션의 분류에 따라서 각 레벨을 위한 적당한 local 스케줄링 schemes이 아래에 있다 data-dependant 한 오퍼레이션을 위한 병렬 알고리즘을 만들기는 무척 힘들다 [4]

#### ▶ Data-independent

- Local operators (Pixel) Relaxed, Grouping
- Local operators (Mask) Partially relaxed, Grouping
- Global operators Spinlock, Channel, Grouping, Tree

#### ▶ Data-dependent

- Partitioning, Grouping, Tree
- AI-approach ( quite difficult )

프로세서간의 통신, data sharing, synchronization, data partitioning, 그리고 message passing은 각각의 오퍼레이션을 수행하기 위해, 그리고 사용자의 복잡한 응용프로그램을 속내 유기적으로 결합되어야 한다

### 4. 실험결과

#### 4.1 예제 Vision Task

얼굴 이미지 sequence로부터 눈의 위치를 찾아내는 작업이 실험에 이용되었다 이 task는 매우 계산 집약적이며, 실시간 처리가 요구되기 때문에 실험에게로서 적당하다고 할 것이다 알고리즘의 세세한 부분은 여기서 중요하기 않으므로 생략하고 단지 이해를 돕기 위해 계산에 관련된 부분만 실험해 것이다

눈의 위치를 찾는 응용기술은 histogram template matching이다 [8] Histogram을 계산하는 부분이 신개요

퍼레이션을 지연시키는 bottle-neck이므로 병렬처리를 이용해서 가속화 시켜야 할 것이다 계산과징의 중요한 세 부분 task를 언급하면 (1) RGB값을 변환, (2) search range내의 영역의 histogram들을 계산, (3) template histogram과 계산되어진 histogram들과의 비교를 들 수 있다 여기서 task(2)가 전체 코드의 70~80%의 차지해서 가장 계산의 비중이 높다고 할 것이다 Task(1)은 프로세서간의 communication이 필요없는 완전한 data parallelization이 가능하고, task(2)에는 synchronization 및 그룹화가 적용되어 있고, task(3)에는 tree scheme과 그룹화가 적용되어 있다

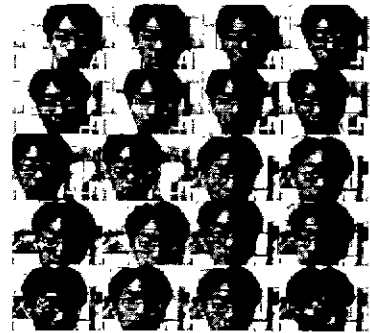


그림 7. 눈위치 검출후의 이미지

4.2 결과 및 이미지

그림 4,5,6은 execution time, speedup, efficiency 및 evaluation factor에 대한 실험결과 그래프를 보여준다

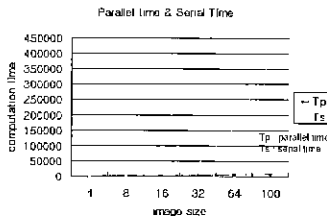


그림 4 Parallel Time & Serial Time

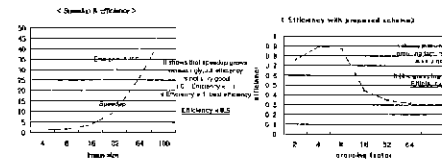


Figure 6 Speedup & Efficiency

그림 6 제한된 scheme 사용시의 efficiency

이 그래프들은 특검 실험 ( 눈위치검출 )에 대한 만족할 만한 결과를 보여준다. 그림 7은 눈 검출 알고리즘을 적용한 후의 결과이미지이다

5. 요약 및 향후과제

본 논문에서는 네트워크로 연결된 상이한 종류의 워크스테이션들을 이용한 실용적인 예제의 vision task들을 병렬 처리하는 방법을 보여주었다 이들 범용의 워크스테이션을 이용하는 것이 경제적이고 실용적인 측면에서 보다 선호할 만한 것이라고 말할 수 있다

확장성의 측면에서 계산 노드들 더하고 빼는 것이 쉽게 이루어지지만 작업의 균등한 분배의 문제는 아직 해결되어야 할 과제이다 또한 사용자에게 보다 쉽게 이용할수 있는 transparent한 분산 미진 시스템을 개발하는 것도 중요한 과제라 할 수 있다

참고 문헌

- [1] C-L Wang, P B Bhat, and V K Prasanna, "High-performance computing for vision," Proc of The IEEE 84, pp 931--946, Jul 1996
- [2] L H Jameson, E J Delp, and C.-C Wang, "A software environment for parallel computer vision," Computer 25(2), pp 29 -45, 1997
- [3] V K Prasanna, "Parallel architectures and algorithms for image understanding ch Software Environments", pp 453- 562, Academic Press INC, 1991
- [4] B P. Lester, "The Art of Parallel Programming", Prentice HALL, 1993
- [5] C. Lee, Y-F Wang and T Yang, "Global optimization for mapping parallel image processing tasks on distributed memory machines," J Parallel and Distrib. Computing 45, pp 29--45, 1997
- [6] A Geist, A Beguelin, J Dongarra, W Jiang, R Manchek, and V Sundream, "PVM Parallel Virtual Machine, A Users' Guide and Tutorial for Networked Parallel Computing", MIT Press, 1994
- [7] N Gul and E L Zapata, "Fast hough transform on multiprocessor A branch and bound approach," J Parallel and Distrib Computing 45, pp. 82--89, 1997
- [8] K-I Kang, J-H Ra, and M-H Kim, "Tracking the eye trajectories in dynamic images using adaptive template matching," in Proc of the 9th Spring Conf of Korea Informanion Processing Society, 1998