論文96-33B-1-5

컴퓨터 비젼 응용을 위한 태스크 레벨 파이프라인 멀티컴퓨터 "RV860-PIPE"의 구현

(Implementation of a Task Level Pipelined Multicomputer "RV860-PIPE" for Computer Vision Applications)

李 忠 煥 *, 金 俊 聲 *, 朴 圭 皓 *

(Choong Hwan Lee, Jun Sung Kim, and Kyu Ho Park)

요 약

본 논문에서는 컴퓨터 비젼응용을 위한 태스크레벨 파이프라인 멀티컴퓨터인 ''RV860-PIPE(Realtime Vision i860 system using PIPEline)''를 구현하고 성능평가 하였다. RV860-PIPE는 MIMD형태의 아키텍처를 가지며, 시각처리에 적합한 ring 형태의 상호연결망(interconnetion network)을 갖는 매시지 패싱 타입의 컴퓨터로 구현되었다. RV860-PIPE의 단위 노드 컴퓨터는 일반적인 시각알고리즘의 수행을 위하여 범용성과 강력한 처리능력을 갖도록 64-bit 프로세서를 사용하여 설계하였다. 또한 단위 노드간 양방향 통신및, 영상 입출력기와의 통신 부담을 덜기위한 전용화된 고속 통신 채널을 설계하였다. 구현된 시스템은 edge 검출, 실시간 이동물체 추적, 실시간 얼굴인식과 같은 시각 처리 응용프로그램들을 성능평가함으로써 컴퓨터 비젼응용에 실용적으로 적용가능함을 보였다.

Abstract

We implemented and evaluated the performance of a task level pipelined multicomputer ''RV860-PIPE(Realtime Vision i860 system using PIPEline)'' for computer vision applications. RV860-PIPE is a message-passing MIMD computer having ring interconnection network which is appropriate for vision processing. We designed the node computer of RV860-PIPE using a 64-bit microprocessor to have generality and high processing power for various vision algorithms. Furthermore, to reduce the communication overhead between node computers and between node computer and a frame grabber, we designed dedicated high speed communication channels between them. We showed the practical applicability of the implemented system by evaluting performances of various computer vision applications like edge-detection, real-time moving object tracking, and real-time face recognition.

I. 서 론

2차원 혹은 3차원 영상을 분석하기 위한 컴퓨터 시각 처리(computer vision processing)는 처리해야 할 데이타 양은 물론 처리 알고리즘의 복잡도로 인한 많은 계산 요구에 의해 실제 응용및 구현에 어려움을 겪고 있다. 따라서 실시간 컴퓨터 시각 처리를 위해서는 시각 처리 알고리즘에 내재되 있는 병렬성을 최대한 활용하고 이를 극대화 할 수 있는 적합한 아키텍처를 찾는 것이 필수적이다. 이를 위해서는 시각 처리 알고리즘의 분석및 병렬성(parallelism)의 광범위한 고

* 正會員, 韓國科學技術院 電氣 및 電子工學科 컴퓨터 工學 硏究室

(Computer Engineering Research Laboratory Department of Electrical Engineering)

接受日字:1995年2月8日, 수정완료일:1995年12月18日

찰이 필요하다[1][2].

대부분의 시각 처리 알고리즘은 3단계의 추상화된 처리 레벨(abstracted processing level)로 구성되어 있다. 이러한 추상화의 첫번째 단계로는 입력 영상의 전처리에 해당하는 저수준 처리(low-level processing)가 있다. 이 단계에서는 응용에 따라 영상(image), 라벨 영상(labeled image), 다해상도 영상(multiresolution image)등 수치적 화소 데이타(numerical pixel data)에 대해 필터링(filtering), 콘볼루션(convolution), 특징 추출(feature extraction), 변환(transformation)과 같은 처리가 적용 된다. 이러한 처리의 대부분은 각 화소에 동시에 동일한 처리를 할 수 있는 데이타 레벨의 공간 병렬성(spatial parallelism)을 가지고 있으므로 이를 이용한 대규모 병렬(massively parallel) 시스템을 구현할 수 있다. 이는 주로 SIMD(Single Instruction-stream Multiple Data-stream)구조의 영상처리 전용의 VLSI 형태로 구현된다. 이러한 SIMD 아키텍처로는 그 연결 형태에따라 메쉬(mesh), 어레이 프로세서(array processor), 하이퍼큐브(hypercube), 피라미드 프로세서(pyramid processor)등이 있다. 대표적으로 메쉬형태의 아키텍처로는 CLIP7A[3], MPP[4]등을 들 수 있으며, CM(Connection Machine)[5]의 경우는 국소(local) 통신을 위해서는 NEWS network을 이용하며, 먼거리의 프로세서사이의 통신을 위해서는 하이퍼 큐브 연결을 이용하는 형태를 취한다. 피라미드 아키텍처로는 [6], SPHINX[7] 등을 들 수 있다.

두번째 처리단계인 중간단계처리(intermdiate-level processing)에서는 부호화된(symbolic) 데이타뿐 아니라 수치적 데이타를 동시에 처리 하게 된다. 대표적인 처리로는 Grouping, 모델 변환(회전, 축적, 이동, 투영(projection)) 등이 있다. 이러한 계산은 대부분 데이타에 의존성을 갖게 되고, 불규칙한 통신 패턴을 요구하게 된다. 따라서 이 단계에는 medium-grain 혹은 coarse grain형태의 병렬성이 내재하게 된다. 이러한 병렬성을 이용하기 위해서, 중간단계처리용 아키텍처는 국소, 전역(global) 데이타의 통신을 지원해야 하며, 분산 독립적인 제어가 가능 해야 한다. SIMD와 MIMD(Multiple Instruction-stream Multiple Data-stream)형태의 두 아키텍처가 공존하는 형태의 구조가 위의 요구를 만족 시킬 수 있다. 이러한 형태의 시스템은 PASM[8], IUA[9], VisTA[10]와 같은 전체 시각 시스템(integrated vision system)의 중간단계처리기에서 찾아 볼 수 있다.

세번째 처리로서 고수준처리(high-level processing)는 시각처리의 마지막 단계에 해당하며 앞의 처리 결과를 이용하여 최종적인 결정(decision), 탐색(search), 매칭(matching), 추론(inference), 검증(verification)등을 행하게 된다. 이러한 처리 역시 중간단계와 마찬가지로 부호화된 데이타와 수치적 데이타의 처리가 공존하게 되지만, 중간단계보다 보다 복잡하고 불규칙적인 통신 패턴과 계산이 요구 된다. 따라서 MIMD형태의 coarse-grain 병렬성을 이용하는것이 적합하며 이에따라 PASM,IUA,VisTA와 같은 시각 시스템에서도 고수준처리를 위해서 MIMD 형태의 아키텍처를 지원하고 있다.

한편 위에서 설명한 3단계의 시각처리 단계는 전체 시각 태스크(vision task)를 수행하는데 있어서는 시간병렬성(temporal parallelism)을 이용할 수 있다. 일반적인 시각처리가 전 단계의 출력을 다음 단계의 입력으로 사용하여야 하는 의존성을 갖기때문에 이부분은 어쩔수 없이 순차적인 수행을 해야 하는 부분으로 남는다. 이러한 상황에서 이용할 수 있는 병렬화 방법으로는 파이프라인(pipeline) 형태의 처리 방법을 들 수 있으며 이때 파이프라인상의 각 단계들은 전체 시각 태스크를 분할한 부태스크(subtask)들로 구성된다. 이러한 형태의 유사병렬(pseudoparallel) 파이프라인형 아키텍처에 대해서는 [11]에서 찾아 볼 수 있으며, 앞에 언급된 전체 시각 시스템[9][10]에서도 각 추상화된 단계간에 파이프라인 형태의 처리를 지원하고 있다.

본 논문에서 설명하고자 하는 파이프라인형 멀티컴퓨터인 ``RV860-PIPE''는 위에 설명된 3단계의 추상화된 처리로 이루어진 전체 시각 태스크의 수행을 목표로 한 시스템이다. 앞에서 살펴본 CLIP, MPP, SPHINX와 같은 시스템은 저수준 시각처리를 위한 전용 시스템으로 중간단계, 고수준과 같은 시각 처리를 위해서는 적합하지 않다. 반면에 PASM, IUA, VisTA, 유사병렬 아키텍처와 같은 전체 시각 태스크의 수행을 위한 시스템의 경우는 각 단계별로 적합한 아키텍처를 전용화(dedication)하고 이들간에 연결망(interconnection)을 구축하는 형태를 취하고 있다. 그러나 이같은 시스템을 실제 구현하기 위해서는 각

단계를 전용화 해야 하는 부담과 함께, 전용화된 단계 사이에 이에 적합한 연결망을 구축해야하는 부담 또한 안게 된다. 이는 구축 비용의 증가는 물론 구축 기간 또한 오래 걸리게 된다. 본 논문에서 설명하는 RV860-PIPE는 위와 같은 문제점의 실용적인 타협점으로 전체 비젼 태스크를 부태스크단위로 단위 프로세서에 매핑하고, 프로세서간에는 전용 통신망을 통해 계산및 데이타의 파이프 라인을 유지하는 방법을 사용한다. 따라서 RV860-PIPE는 MIMD형태의 아키텍쳐를 갖게 되며, 단위 프로세서는 각기 다른 부태스크가 독립적으로 수행될 수 있다. 한편 테크놀로지의 발전으로 단위 프로세서의 성능이 강력해 짐에 따라, 단위 프로세서를 강력하게 함으로써 각 레벨의 전용화에 근사한 처리속도를 가질 수 있으며, 전체 시스템의 범용성 유지및 구현에도 유리하다.

본 논문에서는 컴퓨터 시각처리를 목적으로 개발된 파이프라인형 멀티컴퓨터인 RV860-PIPE의 하드웨어 및 소프트웨어 구조에 대해서 설명하고, 이의 성능평가로서 몇가지 시각처리 응용(application)에 대한 적용과 이의 성능에 대해서 언급한다. II장에서는 RV860-PIPE의 단위 프로세서및 연결망의 구성, 호스트 컴퓨터와의 상호연결(interface)등에 대한 전체 하드웨어 구조와, 단위 프로세서의 구동을 위한 node-kernel의 구조및 전체 시스템의 소프트웨어 구조에 대해서 설명한다. III장에서는 구현된 RV860-PIPE상에서 시각처리 프로그램을 병렬화하기 위한 기법과 그에 따른 성능분석에 대해서 언급한다. 그리고 IV장에서는 구현된 시스템에 성능평가를 위한 시각처리 응용에 대한 실험결과를 언급하고, 마지막으로 IV장에서 결론을 맺는다.

II. RV860-PIPE의 구현

1. 개 요

하드웨어로 구현된 파이프라인형 멀티컴퓨터는 RV860-PIPE로 명칭하며, 멀티컴퓨터를 구성하기 위한 단위 프로세싱 노드는 RV860이라 명칭한다.

RV860-PIPE시스템의 전체 구성은 그림 1과 같다. 시스템의 호스트는 PC(Personal Computer)이며 영상의 입출력을 담당하는 영상 취득기(frame grabber)와 RV860으로 구성된 멀티컴퓨터(4노드 실제구현)로 구성되있다.

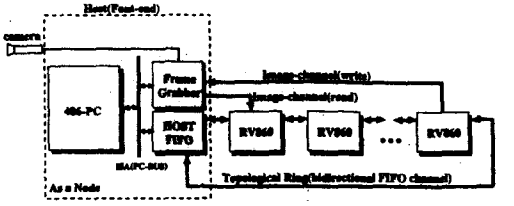


그림 1. 파이프라인형 멀티컴퓨터 RV860-PIPE의 전체 구성도

Fig. 1. System block diagram of pipelined multicomputer RV860-PIPE.

RV860끼리의 상호 연결은 양방향 통신 채널에 의해 이루어지며, 호스트를 포함해서 이중연결된 링(Doubly Linked Topological Ring)을 형성한다. 호스트와의 interface는 HOST FIFO 모듈을 통함으로써 호스트 자체를 링 상의 하나의 프로세싱 노드로 보이도록 하였다. 프레임 그레버는 입출력되는 영상 데이타를 컴퓨팅 모듈인 RV860과 주고 받을 수 있도록 2채널을 가지고 있다. 한편 RV860은 프로세서로 Intel의 i860-XR(80 MFLOPS)를 사용하였으며, 8 MByte의 메모리와 다른 노드와의 상호연결을 위한 2개의 FIFO통신 channel을 가지고 있다. 따라서 태스크 레벨의 파이프라인을 형성하기 위한 데이타의 경로는 프레임 그레버로부터 링 상의 첫번째 노드와의 Image-channel(read)과 RV860사이의 양방향 FIFO채널, 마지막으로 링 상의 마지막 노드와 프레임 그레버사이의 Image-channel(write)을 통해 이루어 지며, 모든 데이타 채널의 대역폭은 10 MByte/sec를 유지하고 있다.

한편 각 단위 노드에는 시스템의 관리및 통신을 위한 kernel이 구현되어 있으며, 이를 이용하여 각 태스크간에 동기적/비동기적 메시지패싱(synchronous/asynchronous message passing)이 가능하다. 또한 kernel에서는 STORE-AND-FORWARD 방식에 의해 노드간의 메시지 라우팅(routing)을 담당하고 있다. 전체 알고리즘은 통신과 계산에 해당하는 부하(load)를 균등(balancing)하게 하는 방향으로 각 노드에 매핑되어 수행된다.

2. 하드웨어의 구조

RV860-PIPE를 구성하는 하드웨어는 크게 실제 태스크를 수행하는 프로세서인 단위 노드 컴퓨터들과 호스트를 ring 연결망에서 하나의 노드처럼 연결하기 위

한 Host-FIFO Module, 영상의 입출력및 단위 노드와의 영상의 전송을 담당하는 영상취득기, 그리고 호스트 컴퓨터로 나누어 볼 수 있다. 단위 노드 컴퓨터들은 ring형태의 연결망을 지원하는 RV860상의 양방향 FIFO channel에 의해 연결되어 host 컴퓨터의 후위 컴퓨터로(back-end computer)로 존재하게 된다. 다음 부절들에서는 RV860-PIPE를 구성하기 위한 하드웨어에 대해서 설명한다.

RV860-PIPE 시스템이 일반적인 비젼시스템을 목표로 하고 있으므로 단위 노드 컴퓨터에서 수행되는 시각 태스크는 저수준의 많은 부동 소수점 계산은 물론, 고수준의 복잡한 알고리즘을 포함한다. 이는 단위 노드 컴퓨터가 여러 알고리즘을 무리없이 수행시킬 수 있는 일반성을 가져야 함을 의미한다. 또한 RV860에서의 부태스크의 수행시간은 전체 태스크 파이프라인의 사이클 타임을 결정하므로서 전체 수행시간에 큰 영향을 미치게 된다. 따라서 단위 노드 컴퓨터는 위의 일반성과 함께 고성능의 처리능력을 요구 하게 된다.

본 연구에서는 위의 조건을 만족시키기 위해서 64-bit의 고성능 마이크로 프로세서인 INTEL의 I860-XR을 사용하여 단위 노드 컴퓨터를 구성하였다. 다음은 i860-XR 프로세서의 제원[12] 이다.

i) Instruction throughput : 80 MFLOP(Million Floating-Point Operation per Second) in single -precision at 40 Mhz
ii) Cache : 8 Kbyte data cache(640Mbyte/sec bandwidth at 40 Mhz) 4Kbyte instruction cache(320 Mbyte/sec bandwidth at 40 Mhz)
iii) 160 Mbyte/sec external bus bandwidth at 40 Mhz
iv) 32-bit(4 Gbyte) address space with on-chip paging unit
v) Parallel integer and floating-point operation(dual-instruction) support
vi) 3D graphic instruction support.

그림 2와 그림 3은 각각 단위 노드 컴퓨터를 구성하고 있는 DRAM부와 I/O부의 구성도이다.

우선 단위 노드 컴퓨터의 DRAM부는 프로그램과 데이타를 위한 8 MByte의 DRAM과 이를 구동하기 위한 DRAM controller로 구성되어 있다.

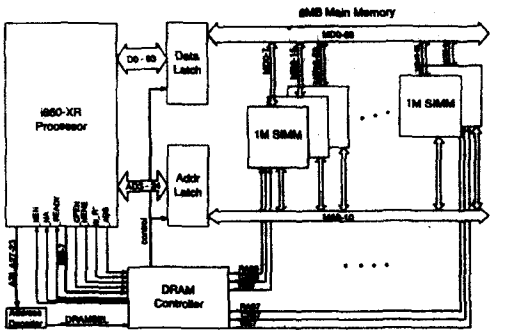


그림 2. Processor-DRAM부의 구성도
Fig. 2. Block diagram of the Processor-DRAM part.

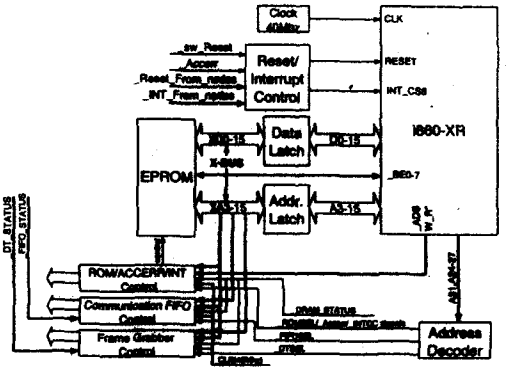


그림 3. Processor-I/O부의 구성도
Fig. 3. Block diagram of the Processor-I/O part.

DRAM controller는 i860 프로세서의 external bus에 대한 pipelined cyle을 지원 할 수 있도록 EPLD(Erasable Programmable Logic Device)로 설계되어 DRAM에 대한 pipelined Load/Store의 instruction의 경우 40 Mhz에서 50ns에 수행시간을 갖는다. 한편 단위 노드 컴퓨터의 I/O부는 시스템의 부팅(booting)및 모니터링을 위한 EPROM(Erasable Programmable Read-Only Memory), interrupt controller, 다른 노드와의 통신을 위한 FIFO controller, 영상 취득기와의 interface를 위한 controller로 구성되어 있다. 이들 I/O를 위한 address/data bus로는 16 bit폭의 X-BUS를 별도로 두고 있다. 모든 I/O control은 i860 프로세서의 메모리 영역을 이용한 Memory-mapped I/O형식을 취하며, 각 controller를 위한 state machine들은 EPLD를 이용하여 구현되었다.

2) Ring Interconnection

RV860-PIPE는 전체 시각 태스크의 태스크 수준의 파이프라인 처리를 목표로 한 시스템이므로, 파이프라인에 요구되는 태스크간의 데이타 통신을 원활히 할 수 있는 적합한 연결망이 요구된다. 이를 위해서 RV860-PIPE에서는 대부분의 시각 처리 태스크가 전단의 데이타를 입력으로 받아 자신에 해당하는 프로세싱을 거친후 다음단으로 출력하는 형태의 직렬적인 연결 구조를 요구 한다는 가정하에 ring형태의 연결망을 채택하고 있다. 따라서 단위 노드 컴퓨터인 RV860는 이웃하는 노드끼리의 연결망을 구성하기 위한 2개의 FIFO 통신 채널을 제공하므로써 전체노드사이에 ring 연결망을 형성할 수 있다. 그림 4에는 이와같은 연결망을 형성하기 위한 FIFO 채널부의 구성도이다.

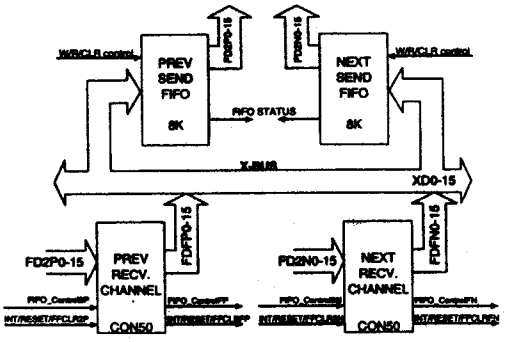


그림 4. FIFO channel부의 구성도
Fig. 4. Block diagram of FIFO channel part.

단위 노드 컴퓨터상에는 그림 4에서 보는 것과 같이 양방향의 노드로 송신을 하기위한 각각 8Kbyte의 FIFO와 ring상의 이전 노드와 다음 노드로 부터의 수신을 위한 2개의 수신채널이 존재한다. FIFO는 동시에 read/write가 가능하므로 채널이 연결된 두 노드 사이에 asynchronous 통신이 가능하다. 송신과 수신을 위한 데이타 버스는 그림 3에서 설명된 I/O를 위한 X-BUS에 연결되며, 통신을 위한 콘트롤은 그림 3의 FIFO controller에서 담당하고 있다. 한편 수신을 위한 2개의 채널은 노드간의 interrupt, reset, FIFO Clear를 위한 신호들의 경로로서도 사용된다.

3) 영상 취득기 Interface

영상의 입출력을 위한 영상 취득기로는 호스트 컴퓨터의 ISA-BUS상에 존재하는 Data Translation사의 DT2871[13]을 사용하였다. DT2871의 제원은 다음과 같다.

i) Four 256 Kbyte(512 × 512 × 8-bits) frame buffer for RGB/HSI color image
ii) 30 frames/sec rate for capturing and displaying
iii) RGB to HSI, HSI to RGB conversion at 10 Mhz support
iv) External I/O data ports at 10 Mhz support

RV860-PIPE에서는 영상 입출력기가 호스트에 존재함으로써 오는 영상 입출력의 병목현상을 해결하기 위해 DT2871에서 제공하는 영상 입출력을 위한 I/O data port와 노드 컴퓨터인 RV860과의 interface를 마련하였다. 이를 위한 구성은 그림 5와 같으며 DT 2871과 RV860사이의 간단한 handshake protocol에 의해 영상 데이타의 송수신이 이루어 진다. 이러한 영상 채널의 데이타 버스는 FIFO 통신을 위한 I/O와 마찬가지로 RV860의 X-BUS에 연결되며, 10 Mbyte /sec의 전송률을 가진다.

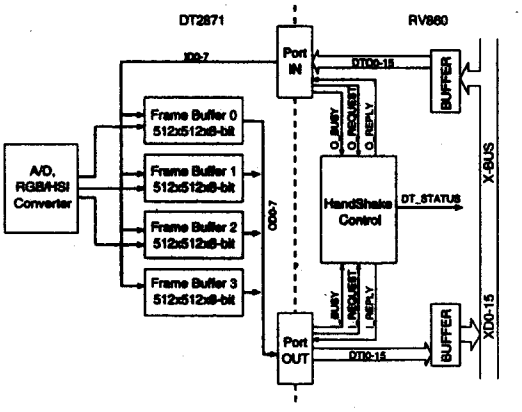


그림 5. Image channel부의 구성도
Fig. 5. Block diagram of Image channel part.

4) Host Interface

RV860-PIPE는 host와의 interface를 위한 별도의 Host FIFO 모듈을 가지고 있다. 이의 구성은 그림 4의 노드 컴퓨터사이의 통신을 위한 FIFO 채널과 동일한 설계를 유지하고, 호스트 버스와의 interface logic을 추가 함으로써 구현되었다. 따라서 호스트 버스상에 존재하며 FIFO 채널을 통해 자신의 이전과 다음의 노드와 연결됨으로써 호스트를 ring상의 하나의 노드처럼 보이게 만들수 있는 장점을 지니게 된다. 이로서

hardware및 software구조의 간편성과 함께 호스트가 바뀜으로써 생기는 노드컴퓨터와의 interface 구조의 변화가 필요 없으므로 확장성을 지니게 된다.

한편 RV860-PIPE의 호스트 컴퓨터는 RV860보드들과 함께 passive backplane에 존재할 수 있는 486DX50프로세서 CPU 보드와 영상의 저장및 프로그램 개발을 위한 하드디스크, 모니터로 구성되고 호스트 컴퓨터의 운영체제로는 DOS가 이용되었다.

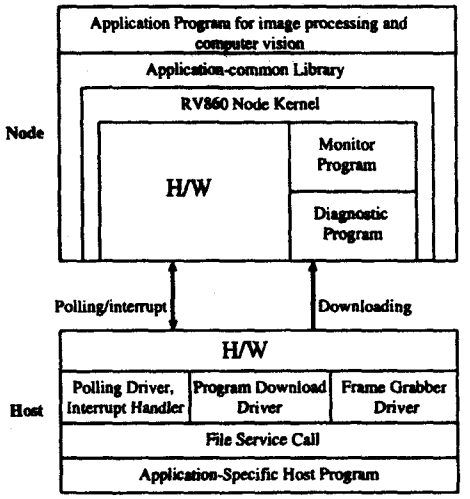


그림 6. RV860-PIPE 시스템의 소프트웨어 계층 구조
Fig. 6. Software hierarchy of RV860-PIPE system.

3. 소프트웨어의 구조

RV860-PIPE의 운용을 위한 소프트웨어의 구성은 크게 RV860을 위한 것과 호스트 컴퓨터를 위한 것으로 나누어 볼 수 있다. RV860을 위해서는 먼저 hardware 진단을 위한 program과 EPROM상에 존재하면서 시스템의 부팅과 kernel의 download및 debugging을 담당하고 있는 monitor program이 있다. 또한 노드컴퓨터의 자원관리, 운용및 통신을 지원하는 노드 kernel과 호스트에서 수행될 시각처리를 위한 공통의 library가 존재한다. 이들을 이용한 실제 시각검사 프로그램들은 호스트에서 개발되 download를 통해 수행되게 된다. 호스트를 위한 소프트 웨어로는 RV860과의 interface를 위한 polling, interrupt driver와 개발된 시각검사 프로그램을 monitor와의 통신을 통해 각 노드에 broadcast하기 위한 download driver, 또한 DT2871을 구동하기 위한 device driver로 구성되어 있고 실제 응용에 따라 특정한 프로그램이 수행되게 되어있다. 이와같은 소프트웨어의 계층 구조는 그림 6에 나타나 있다.

한편 RV860을 위해 구현된 노드 kernel은 다음과 같은 기능들을 제공한다.

i) System initialization and reset
ii) Trap/interrupt handler
iii) Memory management(4 Gbyte virtual addressing support)
iv) Synchronous, asynchronous communication primitive
v) I/O(host, DT2871) driver and system call
vi) File service system call

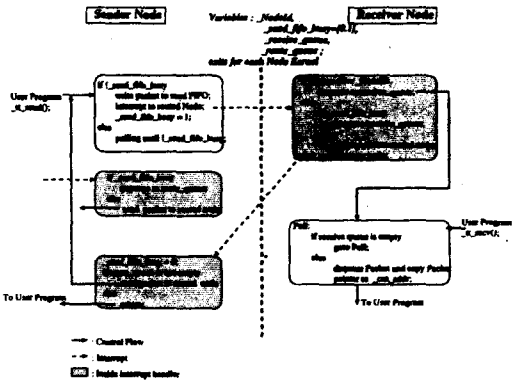


그림 7. RV860 node kernel에서 STORE-AND-FORWARD routing 방법
Fig. 7. STORE-AND-FORWARD routing scheme in RV860 node kernel.

특히 노드간의 통신은 kernel에 의한 STORE-AND-FORWRD 형태의 routing을 통해 이루어 질 수 있으며, 또한 user program에서 직접 routing을 고려한 program을 함으로써 이루어 질수도 있다. 기본적으로 이웃한 두 노드사이의 통신은 polling과 interrupt 두 방식을 모두 사용할 수 있으며, polling의 경우는 원하는 데이타 크기만큼을 EPLD에서 hardware적으로 상태를 polling함으로써 이루어 진다. 한편 kernel에 의한 routing을 할 경우에는 패캣 단위의 통신을 사용하고 interrupt handler에서 수신 interrupt가 발생 했을 경우 수신과 routing을 담당하고 있다. 그림 7은 이같은 routing방법을 설명하고 있다. Routing 알고리즘은 간단하게 ring상에서 통신

을 위해 거쳐야 하는 node의 hop수가 작은 방향으로 라우팅함으로써 구현된다.

위와같은 소프트웨어 구조하에서 프로그램의 병렬화는 전체 시각 태스크를 각 노드에 해당하는 부태스크로 나누어 구성하고 노드간의 데이타 전송및 동기를 위해서는 kernel에서 제공되는 통신 primitive를 사용하여 정적으로 이루어 진다. 따라서 RV860-PIPE 시스템이 전체 시각 태스크의 태스크수준의 파이프라인을 목표로 개발된 만큼 전체 태스크를 어떻게 각 노드에 매핑하느냐에 따라 전체 태스크의 수행에 영향을 주게 된다. 이에 따른 병렬화 모델및 성능에 대해서는 다음장에서 설명된다.

Ⅲ. RV860-PIPE상에서 시각응용을 위한 병렬처리 기법

본장에서는 RV860-PIPE상에서 시각응용을 위한 프로그램을 병렬화 하기위한 가능한 2가지 모델인 파이프라인 병렬화 기법과 데이타 분할 기법[14]에 대해서 설명하고 성능분석을 한다.

1. 데이타 분할 병렬화 기법

데이타 분할기법은 처리하고자 하는 전체 데이타를 각 프로세서에 균등히 분배한후에 각각의 프로세서가 동일한 처리를 한 후 부분적 결과 데이타를 다시 조합(merge)하는 형태의 SPMD(Single Program over Multiple Data steam) 형태의 병렬처리 방법이다. 따라서 동일한 알고리즘에 많은 데이타 처리를 요구하는 저수준의 시각처리에 적합하고, 알고리즘의 수정및 확장이 용이하다는 장점을 갖는다. 또한 프로세서의 증가에 따른 비례적(linear) 성능향상을 기대 할 수 있지만, 이에 따른 통신에 의한 부담(Overhead)을 고려한 알고리즘의 병렬화가 수반되어야 한다.

RV860-PIPE에서 데이타 분할 기법을 사용하여 시각처리 태스크를 수행할때의 프로세서간의 태스크 파이프라인 활동도는 그림 8과 같다. 각 프로세서에 할당된 태스크는 처음에 데이타의 분배를 위한 통신과 분배된 데이타의 처리 그리고 처리된 결과에 대한 통신의 순서로 파이프라인을 형성하고 있다. 이때 통신데이타양과 처리 데이타양은 각 프로세서가 동시에 처리를 끝내고 통신을 할 수 있도록 적절한 분배가 필요하다. 그렇지 않으면 프로세서가 처리를 끝내고 통신을 기다리는 시간의 낭비에 의해 프로세서의 효율이 떨어지게 된다.

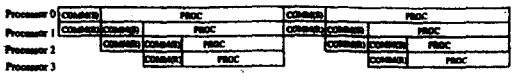


그림 8. 데이타 분할 병렬화 기법에 의한 전형적인 파이프라인 활동도(PROC은 processing, COMM은 communication을,(S)는 send, (R) receive를 의미함

Fig. 8. Typical pipeline activity diagram for data partitioning parallelization model.(PROC means processing, COMM means communication, (S) means send, (R) means receive each other)

한편 D를 전체 시각 태스크가 처리 해야할 데이타라고 할때 데이타 처리 기법을 이용한 병렬화를 위해 $D=d_1,d_2,\ldots,d_n$과 같이 n개의 분할을 할 경우 성능은 다음과 같다. p개의 프로세서를 사용하고, $\alpha=\frac{n}{p}$라고 정의하고, T_{p_i}는 프로세서 i 의 수행시간, T_{total}은 알고리즘의 전체 수행시간이라 하면,

$$T_{p_i}=T_{proc\_i}+T_{comm\_i} \tag{1}$$

$$T_{p\_\max}=MAX(T_{p_1},T_{p_2},\ldots,T_{p_p}) \tag{2}$$

$$T_{total}=\begin{cases}\alpha T_{p\_\max} & \text{if } \alpha>1,\\ T_{p\_\max} & \text{if } \alpha\le 1\end{cases} \tag{3}$$

와 같이 구해진다. 여기서 T_{proc_i}는 분할된 하나의 데이타 블럭을 i 프로세서에서 처리하는 시간이고, T_{comm_i}은 i프로세서에서 하나의 데이타 블럭을 송수신하는데 걸리는 총 통신시간에 해당한다.

2. 파이프라인 병렬화 기법

파이프라인 기법은 각 프로세서에 각기 다른 프로그램을 할당하여 이들의 연동을 통해 전체 알고리즘을 수행시키는 기법이다. 그러므로 복잡한 알고리즘으로 구성되어 있는 시각 알고리즘이나, 알고리즘의 데이타 의존도가 높은 중간단계와 고수준 시각처리 알고리즘의 경우는 이 기법을 사용하여 성능을 향상 시킬 수 있다. 그러나 프로세서간에 서로 다른 태스크와 데이타가 매핑 되어, 프로세서간 불균등한 부하에 의한 효율저하를 초래 할 수 있으므로 태스크의 부하를 균등하게 분할 하기위한 기법[15]을 필요로 한다.

그림 9은 RV860-PIPE에서 파이프라인 병렬화 기

법을 통해 태스크간에 파이프라인을 형성하여 전체 시각 태스크가 수행될때의 활동도이다. 프로세서에 매핑되는 태스크는 데이타의 처리(processing)와 통신을 위한 부분으로 구성되며, 프로세서간의 통신은 FIFO를 통한 비동기 통신으로 중첩(overlap)되어 이루어질 수 있다. 한편 p는 프로세서의 수, m은 한 태스크내의

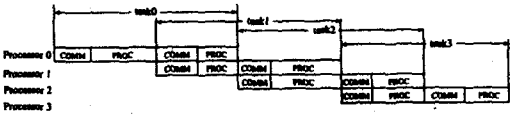


그림 9. 파이프라인 병렬화 기법에 의한 전형적인 파이프라인 활동도(PROC은 processing, COMM은 communication을 의미함)

Fig. 9. Typical pipeline activity diagram for pipeline parallelization model.(PROC means processing, COMM means communication)

통신 패킷의 수, T_{p_i}는 프로세서 i 의 수행시간, T_{total}은 알고리즘의 전체 수행시간 이라할때 파이프라인 병렬화에 의한 성능분석은 다음과 같다. T_{proc_i}, T_{comm_i}를 각각 i 프로세서에서 하나의 데이타 패캣에 대한 프로세싱 시간, 통신 시간 이라 할때 T_{p_i}, T_{total}은

$$T_{p_i} = T_{proc_i} + T_{comm_i} \tag{4}$$

$$T_{p_{max}} = MAX(T_{p_1}, T_{p_2}, \cdots, T_{p_p}) \tag{5}$$

$$T_{total} = [\ m + (p-1)\ ] \times T_{p_{max}} \tag{6}$$

로 주어진다. 만약 $m \gg p$ 라면, 전체수행시간은 $T_{total} \approx mT_{p_{max}}$로 주어진다. 따라서 전체 수행시간은 통신량과 한 프로세서에서 걸리는 최대 지연시간에 좌우된다.

Ⅳ. 컴퓨터 시각처리에의 응용 및 성능 평가(Application to Computer Vision and Performance Evaluation)

본장에서는 구현된 RV860-PIPE의 성능평가를 위해 몇가지 시각처리 응용 프로그램의 RV860-PIPE상의 구현과 결과에 대해서 다룬다. 시각 응용 프로그램으로는 전체 시각알고리즘에 대해 일반적인 성능을 측정하기 위하여 비교적 저수준의 응용에 해당하는 edge 검출과 저수준, 중간단계 처리를 포함하고 있는 실시간 이동물체 추적 알고리즘, 그리고 3단계의 처리를 모두 포함하고 있는 실시간 얼굴인식 알고리즘을 구현하였다.

한편 구현된 응용프로그램의 RV860-PIPE에서의 수행시간 평가를 위해서는 호스트 컴퓨터상의 타이머를 이용하였다. 따라서 측정된 시간에는 RV860에서 시간을 기록하기 위한 interrupt를 호스트에 걸고 호스트의 interrupt handler에서 이에 반응하기 위한 시간이 포함 되어있다. 다른 시스템과의 비교평가를 위해 측정된 SUN-SPARC STATION II에서의 수행시간은 두 시스템간의 자원(resource)및 I/O 성능의 차이로 RV860-PIPE시스템에서의 수행시간과 근사적인 비교자료에 해당한다. 또한 기술된 수행시간은 실험에 사용된 영상에 따라 가변되므로 일반적인 영상에 대한 평균값을 기술한다.

1. 병렬 Canny 에지 검출(Parallel Canny Edge Detection)

Edge detector중 비교적 복잡한 Canny edge detector[16][18]를 구현하였다. Canny의 알고리즘은 노이즈를 제거하기위한 gaussian mask operation과 미분 성분을 구하기 위한 gradient mask operation과 같은 저수준의 처리를 포함하고 있으므로 병렬화를 위해서는 III장에서 설명된 데이타 분할기법을 사용하였다. 표 1은 $256 \times 256 \times 8$-bit 입력영상에 대한 수행시간으로 프로세서의 수에 비례하는 정도의 속도향상(speedup)을 보였다. 그림 10의 (a)는 실제 RV860-PIPE에서의 수행결과 이다.

표 1. Canny edge detection의 수행시간 비교

Table 1. The comparison of Run times for the Canny edge detection.

	SUN	1 노드	4 노드
수행시간 [sec]	9	3.5	1

2. 실시간 이동 물체 추적(Real Time Moving Object Tracking)

움직이는 물체를 추적하기 위한 알고리즘으로 연속영상의 시간에 따른 변화량을 구하고 그것을 가지고 임계값에 의한 세그멘테이션(segmentation)을 통해

가장 움직임이 큰 물체를 추적하는 알고리즘을 구현하였다[17][18][19].

전체 알고리즘이 여러단계의 처리를 필요로 하므로 III장의 파이프라인 병렬화 방법에 따라 알고리즘을 분할하여 프로세서에 매핑하였다. 각 프로세서에 부하를 고려한 태스크의 매핑은 다음과 같다. 프로세서 0에는 시간에 따른 미분처리, 1에는 subsampling, 2에는 노이즈제거와 세그멘 테이션, 3에는 움직이는 물체의 중심을 구하고 표시하는 태스크가 각각 할당 되었다. 512×512×8-bit 해상도의 영상에서 수행결과, 표 2에서와 같은 성능을 얻었다.

표 2. 이동물체 추적 알고리즘의 성능 비교
Table 2. The comparison of performance for the moving object tracking algorithm.

	SUN	4 노드
throughput [frame/sec]	0.5	4

3. 실시간 얼굴 인식(Real Time Face Recognition)

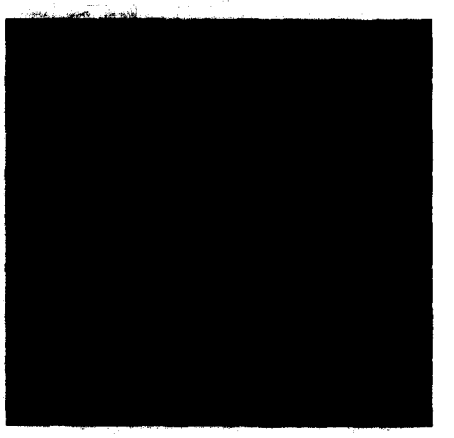
얼굴인식을 위하여 HSI(hue-Saturation-Intensity)공간에서의 color정보를 이용하여 얼굴의 특징영역을 분할하고, 분할된 특징영역의 이동, 축적, 회전에 대한 정규화를 거쳐 모델 템플리트(template)와 상호상관(cross-correlation)을 통해 얼굴을 인식해 내는 알고리즘을 구현하였다[18][20][21]

구현된 알고리즘이 3단계의 시각처리를 모두 요구할 뿐만아니라, 호스트와의 I/O 또한 다른 응용에 비해서 많이 필요하므로 파이프라인 병렬화 방법에 따라 각 프로세서에 적당한 부하를 가지도록 전체 알고리즘을 분할하여 할당하였다[18]

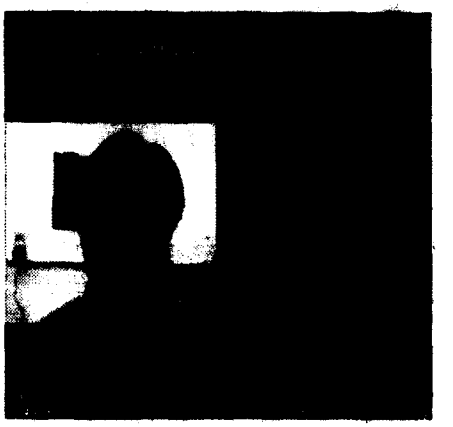
표 3은 구현된 얼굴인식 알고리즘의 수행시간 비교로, SUN에 비해서 월등한 성능향상을 볼 수 있다. 여기서 normal-manner의 수행시간은 512×512×24-bit 입력 정지영상에 대해서 인식결과를 모니터에 출력하기 까지의 총시간이며, pipelne-manner의 경우는 5장의 연속영상에 대해서 연속적인 인식을 수행할때의 throughput에 해당한다. 한편 인식률의 경우는 10사람의 템플리트 모델에 대해 평균 95%의 인식률을 나타냈으며, pipeline-manner로 인식시스템을 구동할 경우에 인식률에 대해 높은 안전성을 보여 주었다. 그림 10 (b)는 실제 RV860-PIPE상에서 수행된 얼굴인식 결과 영상이다.

표 3. 얼굴인식 알고리즘의 수행시간 비교
Table 3. The comparison of run time for face recognition.

	SUN	4 노드 normal-manner	4 노드 pipeline-manner
결과	23.25 sec	3.5 sec	1.5 frame/sec



(a)



(b)

그림 10. (a) RV860-PIPE상에서 parallel canny edge detection 수행 결과 영상, (b) RV 860-PIPE상에서 Real-Time Face Recognition 수행결과 영상
Fig. 10. (a) Result image of parallel canny edge detection on RV860-PIPE, (b) Result image of Real-Time Face Recognition on RV860-PIPE.

V. 결 론

본 연구에서는 컴퓨터 비젼응용을 위한 태스크레벨 파이프라인 멀티 컴퓨터인 RV860-PIPE를 구현하고, 성능평가 하였다.

RV860-PIPE는 MIMD형태의 아키텍처로서 프로세서간의 상호연결로는 시각처리에 적합한 ring구조를 가지고 있다. RV860-PIPE의 단위 노드 컴퓨터는 일반적인 시각처리 알고리즘을 무리없이 수행할 수 있으며, 전체 태스크 파이프라인의 수행시간에 큰 영향을 주지 않토록 고성능 처리능력을 가진 64-bit의 범용 프로세서를 사용하여 구현하였다. 또한 단위 노드컴퓨터간의 매시지 패싱을 위해서는 양방향의 FIFO 채널을 제공하므로써 고속의 비동기 통신이 가능하도록 하였고, 영상의 입출력을 위해서는 노드 컴퓨터와의 전용 채널을 사용하여 입출력에 대한 부담도 최소화 하였다.

한편 성능평가를 위해 edge detection, 실시간 이동물체 추적, 실시간 얼굴인식과 같은 일반적인 컴퓨터 비젼응용에 적용하였고, 그 결과 구현된 시스템은 성능면에서 실시간에 근접한 속도를 얻었으며, 응용면에서는 전체 비젼태스크를 구현하는데 있어 적합함을 보였다. 따라서 RV860-PIPE 시스템이 컴퓨터 비젼응용에 실용적으로 적용가능함을 보였다.

추후과제로는 전체 시각처리 알고리즘의 각 프로세서로의 매핑시 최적의 성능을 얻을 수 있는 매핑기법의 연구및 컴퓨터 비젼을 위한 많은 다른 응용 프로그램의 이식을 통해, 구현된 시스템의 문제점및 개선점 도출등이 필요하다.

참 고 문 헌

[1] Charles C. Weems, "Architectural Requirements of Image Understanding with Respect to Parallel Processing," *Proceedings of the IEEE*, vol. 79, No. 4, April 1991.

[2] Charles C. Weems et al., "The DARPA Image Understanding Benchmark for Parallel Computers," *J. Parallel and Distributed Computing*, pp. 1-24, Jan. 1991.

[3] Terry J. Fountain, K. N. Matthews, and Michael J. B. Duff, "The CLIP7A Image Processor," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 10, No. 3, pp.310-319, MAY 1988.

[4] K. Batcher, "Design of a massively parallel processor," *IEEE Transaction on Computer*, vol. c-29, pp. 836-840, 1980.

[5] L. W. Tucker, "Architecture and application of the Connection Machine," *IEEE Computer*, pp. 26-38, Aug 1980.

[6] C. R. Dyer, "A VLSI Pyramid Machine for Hierarchical Parallel Image Processing," *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing*, pp. 381-386, 1981.

[7] A. Merigor, B. Zavidovique, and F. Devos, "SPHINX, A pyramidal approach to parallel image processing," in *Proc. IEEE Workshop Computer Architecture for Pattern Analysis and Image Database Management*, pp. 107-111, Nov 1985.

[8] H. J. Siegel, L. J. Siegel, F. C. Kemmerer, P. T. Mueller, JR., J. E. Smalley, JR.,and S. D. Smith, "PASM:A Partitionable SIMD/MIMD System for Image Processing and Pattern Recognition," *IEEE Transactions on Computers*, vol. c-30, No. 12, pp. 934-947, Dec 1981.

[9] C. Weems, A. S. P. Levitan, A. R. Hanson, E. M. Riseman, J. G. Nash, and D. B. Shu, "The image understanding architecture," *Int. J. Computer Vision*, vol. 2, pp. 251-282, 1989.

[10] Myung Hoon Sunwoo, "VisTA: An Integrated Vision Tri-Architecture System," *Ph. D Dissertation*, Univ. of Texas at Austin, Aug. 1990.

[11] Dharma P. Agrawal, and Ramesh Jain, "A Pipelined Pseudoparallel System Architecture for Real-Time Dynamic Scene Analysis," *IEEE Transaction on Computers*, Vol. c-31, No. 10, pp. 952-962, October 1982.

[12] Neal Margulis, *i860 Microprocessor Architecture* , McGraw-Hill, 1990.

[13] *.DT2871 User Manual*, DATA TRANSLATION, 1991.

[14] S. Yalamanchili and J. K. Aggarwal, "Analysis of a Model for Parallel Image Processing," *Pattern Recognition*, vol. 18, no. 1, pp. 1-16, 1985.

[15] Alok N. Choudhary, Mun K. Leung, Thomas S. Huang and Janak H.Patel, "Parallel Implementation and Evaluation of Motion Estimation System Algorithms on a Distributed Memory Multiprocessor using Knowledge Based Mappings," *Int Conf. on Pattern Recognition*, vol. 2. pp.337-342, 1990.

[16] John Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.

[17] Claude L. Fennema and William B. Thompson, "Velocity Determination in Scenes Containing Several Moving Objects," *Computer Graphics and Image Processing*, vol. 9, pp. 301-315, 1979.

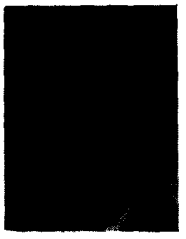
[18] 이 충환, 박 규호, "Performance Evaluation of RV860-PIPE for Computer Vision Application," *KAIST Internal Report*, 1994.

[19] 김 준성, "A Real-Time Face Recognition System," *MS. Thesis*. 전기및전자공학과. 한국과학기술원. Dec, 1994

[20] 정 영기, "A Study on Face Recognition Algorithm under Varying Pose." *MS. Thesis*. 전기및전자공학과. 한국과학기술원. June, 1994

[21] C. H. Lee, J.S. Kim, Y.K. Jung, K.H. Park, "A Real-Time, Parallel Face Recognition Task-Level Pipelined Multicomputer," *Proceedings of IEEE Int. Conf. on SMC*, vancouver, cadada, Vol. 3, pp. 2289-2294, Oct, 1995.

저 자 소 개



李 忠 煥(正會員)

1991년 한양대학교 전자공학과 졸업(학사). 1993년 한국 과학기술원 전기 및 전자공학과 졸업(석사). 1995년 ~ 한국 과학기술원 전기 및 전자공학과 박사과정. 주관심분야는 컴퓨터 그래픽스, 비쥬얼라이제이션, 컴퓨터 비젼, 병렬처리

金 俊 聲(正會員)

1993년 부산대학교 전자공학과 졸업(학사). 1995년 한국 과학기술원 전기 및 전자공학과 졸업(석사). 1995년 ~ 현재 한국 과학기술원 전기 및 전자공학과 박사과정. 주관심분야는 컴퓨터 그래픽스, 비쥬얼라이제이션, 컴퓨터 비젼, 병렬처리

朴 圭 晧(正會員) 第 22卷 第 9號 參照