클러스터환경에서 MPI를 이용한 병렬 MPEG인코더의 설계 및 구현

이좌형* · 정인범**

Design and Implementation of Parallel MPEG Encoder with MPI on Cluster System

Joa-hyoung Lee* · In-bum Jung**

본 연구는 산업자원부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구결과임

요 약

최근 컴퓨터와 네트워크 기술이 빠르게 발전하고 널리 보급되면서 텍스트 위주로 작업을 하던 어플리케이션들의 비중이 낮아지고 멀티미디어 데이터를 처리하는 어플리케이션들의 비중이 점차 증가하고 있는 추세이다. 다양한 멀티미디어들 중에서 동영상 멀티미디어를 다루는 프로그램들은 멀티미디어 응용 어플리케이션들 중에서 큰 비중을 차지하고 있으며 실생활에서 널리 사용되고 있다. 대표적인 동영상 압축 표준인 MPEG의 경우 매우 높은 압축률을 제공하여 일반 사용자들도 손쉽게 동영상 데이터를 접하고 사용할 수 있는 기회를 제공한다. 하지만 MPEG 인코딩은 매우 많은 컴퓨팅 자원과 시간을 요하는 작업이다. 본 연구에서는 동영상 데이터를 인코딩 하는데 소요되는 시간을 줄이기 위해 클러스터환경에서 MPI를 이용하여 동영상 압축 표준인 MPEG 기반의 병렬 인코더를 설계 및 구현하였다.

ABSTRACT

As the computing and network technique move on and spread widly, the usage of multimedia application becomes in general while the usage of text based application becomes low. Especially the application which treats the streaming media such as video or movie, one of multimedia data, holds a majority in the usage of computing. MPEG, one of the typical compression standard of streaming media, provides very high compression ratio so that general users could be close to the streaming media with easy usage. However, the encoding of MPEG requires lots of computing power and time. In the paper, we design and implement a parallel MPEG encoder with MPI in cluster envrionment to reduce the encoding time of MPEG.

키워드

클러스터, MPEG, Encoder, 병렬처리, MPI

* 강원대학교 컴퓨터정보통신공학과 박사과정
** 강원대학교 컴퓨터정보통신공학과 교수 (교신저자)

접수일자 2008. 06. 04

I. 서 론

최근 컴퓨터와 네트워크 기술이 빠르게 발전하고 널리 보급되면서 텍스트 위주로 작업을 하던 어플리케이션들의 비중이 낮아지고 멀티미디어 데이터를 처리하는 어플리케이션들의 비중이 점차 증가하고 있는 추세이다. 다양한 멀티미디어들 중에서 영화같은 동영상 멀티미디어를 다루는 프로그램들은 멀티미디어 응용 어플리케이션들 중에서 큰 비중을 차지하고 있으며 실생활에서 널리 사용되고 있다. 컴퓨터 기술의 발전과 더불어 디지털 캠코더와 같이 동영상을 처리하는 가전기기들이 널리 보급되면서 일반 사용자들의 동영상에 대한 관심과 욕구가 날로 증가하는 상황이다. 하지만 영화와 같은 동영상 데이터의 경우 크기가 매우 방대하여 현재의 컴퓨터 사양에서 원 영상을 그대로 사용하기는 힘들다. 이러한 문제점을 해결하기 위하여 다양한 동영상 압축 알고리즘들이 개발되어져 왔다.

대표적인 동영상 압축 표준인 MPEG[1]의 경우 매우 높은 압축률을 제공하여 일반 사용자들도 손쉽게 동영상 데이터를 접하고 사용할 수 있는 기회를 제공한다. 그러나 MPEG의 경우 고사양화 되고 있는 일반 사용자들의 컴퓨팅 환경하에서 실시간으로 디코딩이 가능하지만 디코딩에 비해 보다 많은 자원과 시간을 필요로 하는 인코딩의 경우 아직 실시간으로 서비스를 제공하기 어려운 상황이다. 본 연구에서는 비디오 데이터를 인코딩하는데 소요되는 시간과 자원을 감소시키기 위해 클러스터환경에서 MPI를 이용하여 동영상 압축 표준인 MPEG-2 기반의 Parallel Encoder를 설계 및 구현하였다.

MPEG 인코딩의 경우 실시간으로 매우 많은 데이터가 입력되며 이를 압축하기 위해 많은 컴퓨팅 자원을 필요로 한다. 이러한 인코딩 작업을 순차적으로 수행할 경우 실시간으로 서비스가 어려우며 컴퓨팅 자원들을 효과적으로 사용하기가 어렵다. 하지만 MPEG의 경우 여러 단계의 계층을 기반으로 인코딩과 디코딩이 이루어지며 이를 기반으로 병렬화하여 실시간으로 서비스를 제공할 수 있다[2]. 본 연구에서는 MPEG 인코딩을 병렬로 처리하기 위하여 저가의 PC를 기반으로 한 클러스터 환경을 이용하였다. 인코딩 작업을 여러 노드에서 수행할 경우 노드간에 데이터 이동이 필요한데 이를 위하여 MPI표준[3]을 구현한 MPICH[4]를 이용하여 노드간 통신을 하도록 하였다.

본 논문에서 구현한 병렬 인코더를 이용하여 동영상 데이터를 압축해본 결과 8대 이상의 노드에서 작업을 병렬화 하여 수행할 경우 초당 25프레임 이상을 처리할 수 있는 것으로 측정되어 실시간으로 서비스를 제공할 수 있음을 확인하였다.

II. 관련연구

1990년대 초반에 국제 표준화기구(ISO)와 국제 전기기술위원회(IEC)에서 동영상 압축 표준인 MPEG표준을 제정한 이후로 인코딩을 실시간으로 처리하기 위하여 다양한 연구가 진행되어 왔다. 인코딩을 실시간으로 수행하기 위한 접근방식으로 크게 소프트웨어를 이용하는 방안과 하드웨어를 이용하는 방안이 제시되고 연구되어져 왔다. 소프트웨어를 기반으로 하는 인코더의 대표적인 예로 미국의 University of California at Berkeley에서는 Sun SPARCstation 10으로 구성된 클러스터환경을 기반으로 352x288크기의 영상을 초당 1.2 ~ 4.7 프레임 정도의 속도로 인코딩하는 병렬 인코더를 설계 및 구현하였다[5][6]. 이 연구에서는 MPEG의 특성을 분석하여 인코딩 성능을 향상시키기 위한 다양한 방안들을 모색하였으며 이후의 연구들에서 많은 참조가 이루어지고 있다. [7][8]의 연구에서는 위의 인코더를 개선하여 최대 초당 40프레임 이상 인코딩이 가능한 실시간 인코딩 시스템을 구현하였다. 하지만 이 연구들은 Touchstone Delta와 Paragon과 같은 슈퍼컴퓨터 환경에서 수백개의 프로세서를 이용하는 환경에서 이루어진 것들이기 때문에 널리 보급되기는 어려운 것들이라 할 수 있겠다. 이외에도 공유메모리를 기반으로 한 멀티프로세서 컴퓨터상에서 실시간 속도로 인코딩을 지원하는 연구들이 있었지만 대부분 수백개의 프로세서를 이용하여야 실시간의 속도로 인코딩이 가능하여 이들도 널리 보급에는 문제점이 있다[9][10][11][12][13]. 하드웨어를 이용한 인코더로는 MVP[14]나 VCP[15]와 같이 인코딩을 위한 전용의 프로세서를 이용하여 실시간으로 인코딩이 가능하도록 하였으나 확장성이 떨어진다고 볼 수 있다.

Ⅲ. 병렬인코더

MPEG-2 인코딩 작업을 병렬화 하기 위해서는 MPEG-2 의 다양한 계층들 중에서 어느 계층에서 병렬화를 할 것인지가 중요하다. 최상위 계층인 시퀀스층은 동영상 내에서 하나만 존재하므로 적당하지 않다. 픽처와 슬라이스 계층은 상호간의 의존성이 매우 커서 병렬수행시 많은 데이터를 주고받아야 하기 때문에 병렬화에 적합하지가 않다. GOP계층은 가장 독립적인 계층으로서 의존성이 적으며 매크로 블록은 의존성이 크지만 작업을 세분화하여 병렬화가 가능하다. 따라서 본 연구에서는 MPEG-2 video의 매크로블록과 GOP(Group of Picture)계층에서 병렬화를 시도하였으며 각 계층에서 병렬화를 수행한 후 인코딩 성능 측정을 통해 가장 적합한 병렬화 계층을 제안하고 최적의 parallel Encoder를 설계 및 구현하고자 하였다.

3.1 실험환경

실험을 위한 VODCA 서버는 HS 노드와 4개의 MMS 노드 그리고 복구 노드로 구성되며, 각 노드는 Linux 운영체제로 동작한다. MMS 노드, HS 노드, 클라이언트는 100Mbps 이더넷 스위치를 통해 연결되어 있다. 모든 MMS 노드와 복구 노드 또한 100Mbps 이더넷 스위치를 통한 내부 네트워크로 연결되어 있다. 표 1은 VODCA 시스템에서 각 MMS 노드의 하드웨어 구성을 나타낸다.

표 1. MMS 노드와 복구 노드 사양.
Table 1. Spec of MMS node and recovery node

CPU	Intel Pentium 4, 1.6 GHz
Memory	256 Mbyte DDR
Disk	Segate 40GB 7200RPM x 2
OS	RedHat 7.3 (Kernel 2.4.18)
Network	100 Mbps Fast Ethernet, 100Mbps Ethernet Switch

표 2. 실험에 사용된 영화 정보.
Table 2. Movie information

Movie name	John Q	Ice Age
Frame size	352×288	352×288
Frame rates	25	25
Bit rates	1,437.6	1,437.6
Running time	110	85
GOP size	124.1	120.8

표 2는 실험에 사용된 영화에 관한 세부 정보를 나타낸다. MPEG-2 영화를 사용하였으며 시스템의 성능을 측정하기에 충분한 상영시간을 가지고 있다. 또한, 표 2에서처럼 영화의 GOP들과 I 프레임의 크기를 측정하였다.

클러스터 내에서 노드간 통신을 위한 MPI라이브러리는 MPICH를 사용하였다.

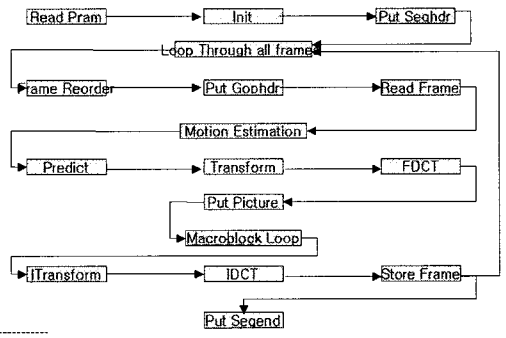

그림 1 순차적 인코더의 동작
Fig. 1. Procedure of sequential encoder

3.2 단일 인코더의 성능분석

그림1은 Sequential Encoder의 동작과정을 보여주며 그림 2는 Sequential Encoder 실행시 각 함수별 수행시간을 보여준다.

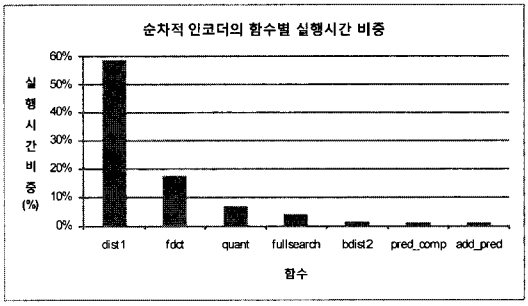

그림 2 순차적 인코더의 함수별 실행시간
Fig. 2. Execution time of function in sequential encoder

그림 2에서 보듯이 Sequential Encoding시 제일 많은 시간을 차지하는 것은 Dist1함수인데 이 함수는 Encoding 과정 중 Motion Estimation에 해당하는 함수이다. Motion Estimation은 각 매크로블록별로 이전 화면과

현재 화면 또는 현재 화면과 다음 화면간의 변화된 부분을 찾는 과정이다. 변화된 부분을 찾기 위해 주어진 범위 내에서 매크로블록과 일치하는 부분을 찾게 되며 일치하는 매크로블록과 현재 매크로블록 간의 거리를 계산하여 Motion Vector로 저장하게 된다. 이때 일치하는 매크로블록을 찾는 과정이 인코딩 과정 중 가장 많은 시간을 소비하는 것이며 이 시간을 줄이기 위해 이 부분을 병렬화 하였다.

3.3 매크로블록에서의 병렬화

매크로블록은 MPEG-2 video의 최하위 계층으로서 압축의 기본 단위이다. 매크로블록 계층에서의 병렬화는 매크로블록 간의 의존성이 다른 계층보다 크기 때문에 데이터를 주고받는 빈도가 많아지는 것이 문제이다. 매크로블록의 이러한 특성을 고려하여 본 연구에서는 노드간 통신량을 최소화하기 위해서 압축시 제일 많은 시간을 차지하는Motion Estimation 부분만의 병렬화를 시도하였다[6].

그림 3은 병렬인코더의 구조를 보여주며 그림 4는 병렬 인코더의 동작을 보여준다. 그림 3에서 보듯이 전체적인 인코딩작업을 수행하는 마스터노드와 Motion Estimation만을 수행하는 인코드 노드로 구성되며 MPI를 이용하여 데이터를 주고 받는다.

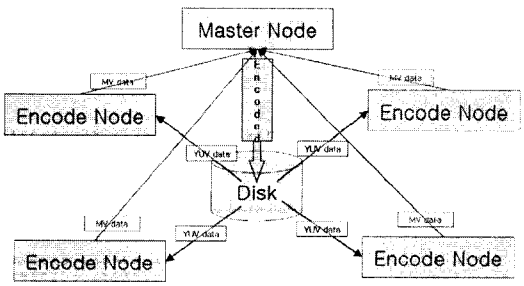

그림 3. 병렬 인코더의 구조(in Macroblock)
Fig. 3. Architecture of parallel encoder(in Macroblock)

전체적인 인코딩 작업은 마스터노드에서 수행되는데 인코딩 과정중 가장 많은 시간을 차지하는 부분인 Motion Estimation부분은 인코드 노드들이 병렬로 수행한 결과를 MPI를 통해 전달 받게 된다. 인코딩시의 입력인 YUV데이터는 NFS로 공유하여 모든 노드에서 접근이 가능하도록 하였으며 압축된 데이터는 마스터노드에서만 기록하도록 하였다.

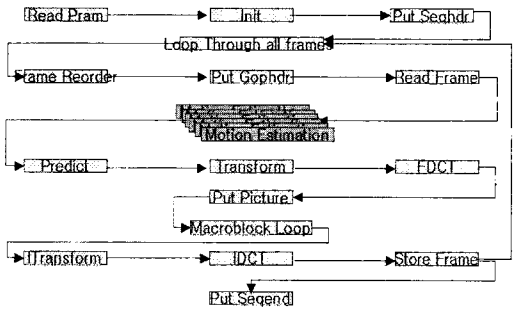

그림 4. 병렬 인코더의 동작과정(in Macroblock)
Fig. 4. Procedure of parallel encoder(in Macroblock)

그림 5는 매크로블록 계층에서의 병렬 인코더를 4개의 노드에서 실행하였을 때 함수별 수행시간을 보여준다. 그림 2와 비교해볼 때 전체적인 수행시간의 비중은 줄어들었지만 Motion Estimation에 사용된 시간이 여전히 많은 비중을 차지하는 것으로 나타났다. 이러한 현상의 원인으로는 비록 마스터 노드에서Motion Estimation을 수행하는 시간은 줄어들지만 다른 노드에서의 수행결과를 받는 작업이 많은 부분을 차지하기 때문인 것으로 판단되어진다.

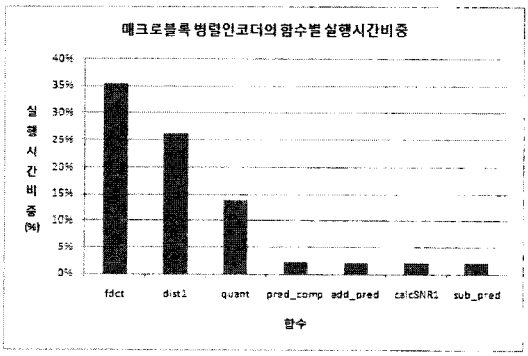

그림 5 매크로 블록에서의 병렬 인코더 수행시 함수별 실행시간(4 노드)
Fig. 5. Execution time of function in parallel encoder in macroblock (4 node)

이는 병렬인코더 수행시 MPI 함수를 통한 데이터 전송을 보여주는 그림6에서 확인할 수 있다. 인코더 노드에서 작업한 결과를 모으기 위해 마스터노드에서 MPI

함수인 broadcast를 사용하였다. 이 함수가 매 픽쳐마다 호출되면서 많은 시간을 차지하는 것을 그림 6에서 확인할 수 있으며 이를 통해 잦은 네트워크 사용이 문제인 것으로 판단되어졌다. (그림에서 흰색 부분이 broadcast함수가 호출된 곳임)

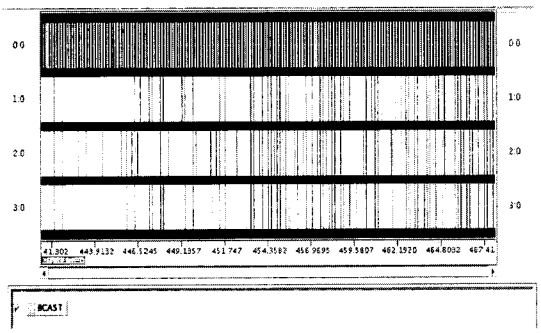

그림 6. MPI 함수의 사용량
Fig. 6. Usage of MPI function

그림 7은 Parallel Encoder 의 실행 성능을 나타내는데 잦은 네트워크 사용으로 인하여 성능향상이 미비하였다.

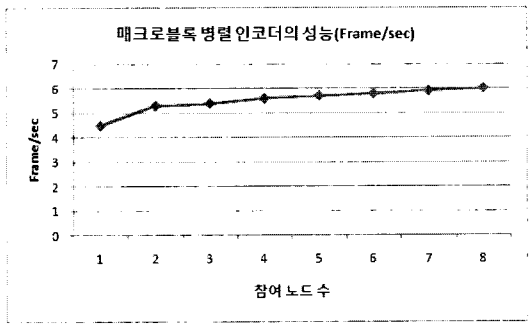

그림 7. 매크로블록에서의 병렬 인코더의 성능
Fig. 7. Performance of parallel encoder in Macroblock

3.4 GOP에서의 병렬화

GOP는 픽쳐들의 묶음이며 다른 GOP에 의존하지않고 독립적으로 인코딩이 가능한 최상위 계층이다. GOP 단위로 나누어서 인코딩을 할 경우 주고받는 빈도가 제일 적기 때문에 매크로블록에 비해 네트워크 부하는 적으나 각 노드에서 병렬 수행된 GOP를 하나의 파일로 묶어서 저장하는 데에 많은 네트워크가 사용된다.

그림 8은 GOP에서의 병렬 인코더의 구조를 보여준다. 그림 8에서 보듯이 병렬 인코더 시스템은 인코딩 작업을 수행하는 인코드 노드와 압축된 결과를 모아서 하나의 파일로 저장하는 컴바인 노드로 구성되며 각 노드들은 NFS를 통해 동영상 데이터를 공유하고 MPI를 통해 작업결과를 주고받는다. 인코드 노드들은 저장장치로부터 YUV데이터를 읽어 GOP단위로 압축을 수행하며 컴바인 노드는 순차적으로 인코드 노드들에서 GOP 데이터를 수신하여 저장장치에 저장한다.

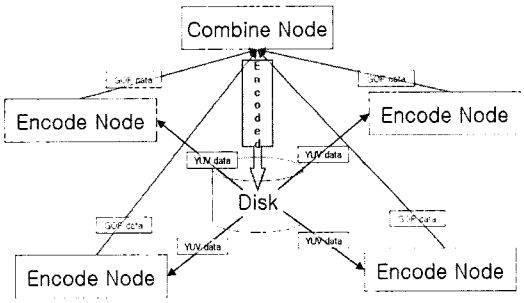

그림 8. GOP에서의 병렬 인코더의 구조
Fig. 8. Architecture of parallel encoder in GOP

그림 9는 GOP 계층에서의 병렬 인코더의 성능을 보여주는데 노드의 수가 7대가 넘으면서 실시간으로 인코딩을 수행함을 알 수 있다.

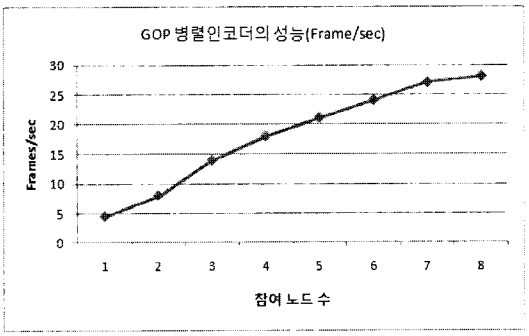

그림 9 GOP에서의 병렬 인코더의 성능
Fig. 9. Performance of parallel encoder in GOP

하나의 컴바인 노드에서 여러 인코드 노드로부터 데이터를 받아서 저장하기 때문에 컴바인 노드가 병목현

상의 원인이 될 수 있다. 모든 노드가 동시에 컴바인 노드로 데이터를 전송하려는 경우가 발생하면 Sequential Encoding이 될 가능성이 있는데 각 GOP마다 소요되는 시간이 다르므로 컴바인 노드가 각 인코드 노드와 통신하는 것이 한 시점에 집중되지는 않을 것으로 생각하였다. 이는 병렬 인코더에서 데이터를 주고받는 MPI함수의 동작을 보여주는 그림 10에서 각 노드가 무작위적으로 컴바인 노드로 데이터를 전송하는 것으로 나타나 문제가 되지 않음을 알 수 있다.

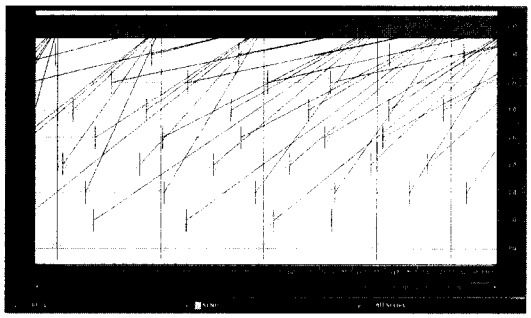

그림 10. GOP에서의 병렬인코더에서 MPI 함수의 동작과정
Fig. 10. Operation of MPI function in parallel encoder in GOP

Ⅳ. 결 론

본 연구에서는 동영상 데이터를 인코딩 하는데 소요되는 시간과 자원을 감소시키기 위해 클러스터환경에서 MPI를 이용하여 동영상 압축 표준인 MPEG-2 기반의 Parallel Encoder를 설계 및 구현하였다. 본 논문에서 구현한 병렬 인코더를 이용하여 동영상 데이터를 압축해 본 결과 8대 이상의 노드에서 작업을 병렬화 하여 수행할 경우 초당 25프레임 이상을 처리할 수 있는 것으로 측정되어 실시간으로 서비스를 제공할 수 있음을 확인하였다.

향후에는 최근에 모바일 환경이나 컴퓨터그래픽등에서 많이 사용되고 있는 MPEG-4 인코딩에 관한 연구를 진행할 계획이다.

참고문헌

[1] http://www.mpeg.org/MPEG/index.html

[2] 이호석, 김준기, "알기쉬운 MPEG-2소스코드 해설", 홍릉과학출판사

[3] http://www.mpi-forum.org/index.html

[4] http://www-unix.mcs.anl.gov/mpi/mpich/

[5] J. Maria Gonzalez, A. geweke, "The Design of a Parallel MPEG-2 Encoder", USC, http://www.cs.berkeley.edu/~chema/papers/NOW-mpeg/html/index.html

[6] K. L. Gong, "Parallel MPEG-1 Video Encoding", MS Thesis, Department of EECS, University of California at Berkeley, May 1994, Issued as Technical Report 811

[7] K. Shen, L. A. Rowe, E. J. Delp, "A Parallel Implementation of an MPEG1 Encoder: Faster Than Real-Time!", Proceedings of the SPIE - The International Society for Optical Engineering, vol.2419, pp. 407-418.

[8] K. Shen and E. J. Delp, "Parallel approaches to real-time mpeg video compression,",submitted to Journal of Parallel and Distributed Computing, 1997.

[9] K. Shen and E. J. Delp,"A spatial-temporal parallel approach for real-time mpeg video compression," Proceedings of the 25th International Conference on Parallel Processing, August 13{15, 1996, Bloomingdale, Illinois.

[10] Eiji Iwata and Kunle Olukotunm, "Exploiting Coarse-Grain Parallelism in the MPEG-2 Algorithm", In Proceedings of the International Conference on Supercomputing (ICS), 1999.

[11] S. M. Akramullah, I. Ahmad, and M. Liou. "A data-parallel approach for real-time MPEG-2 video encoding. Journal of Parallel and Distributed Computing, 30(2):129-146, November 1995.

[12] I. Agi and R. Jagannathan. "A portable fault-tolerant parallel software mpeg-1 encoder.", In Proceedings Second IASTED/ISMM International Conference on Distributed Multimedia Systems and Applications. IASTED/ISMM, August 1995.

[13] D.M. Barbosa et al., Parallelizing MPEG video encoding using multiprocessors, in: Proceedings of XII Brazilian Symposium on Computer Graphics and Image Processing, 1999, pp. 215-222.

[14] W. Lee, R. J. Gove, and Y. Kim, \Real-time MPEG video compression using the MVP," Proceedings of IEEE Data Compression Workshop, April 1994, Snowbird, Utah.

[15] P. Wayner, \Digital video goes real-time," Byte, vol. 19, no. 1, pp. 107{112, January 1994

저자소개

이좌형 (Joa-Hyoung Lee)

2003년 강원대학교 정보통신공학과 (공학사)
2005년 강원대학교 컴퓨터정보통신공학과(공학석사)
2005년 ~ 현재 강원대학교 컴퓨터정보통신공학과 (박사과정)
※관심분야: 멀티미디어 시스템, 센서 네트워크

정 인 범(In-Bum Jung)

1985년 고려대학교 전자공학과 학사.
1985년~1995년 ㈜ 삼성전자 컴퓨터 시스템사업부 선임 연구원.
1992년~1994년 한국과학기술원 정보통신공학과 석사
1995년~2000년 8월 한국과학기술원 전산학과 박사
2001년~현재 강원대학교 컴퓨터정보통신공학전공 교수
※관심분야 : 멀티미디어 시스템, 센서네트워크