

# 워터마킹과 MPEG4를 적용한 DVR 시스템과 실시간 처리 속도 향상에 관한 연구

김자환\* · 허창우\* · 류광렬\*  
\*목원대학교, 대학원 IT공학과

## A Study on the DVR System Realization with Watermarking and MPEG-4 for Realtime Processing Speed Improvement

Ja-Hwan Kim\* · Chang-wu Hur · Kwang-Ryol Ryu\*  
\*Mokwon University, Dept. of IT Engineering  
E-mail : ryol@mokwon.ac.kr

### 요 약

본 논문은 비디오 영상의 워터마킹과 MPEG4를 적용한 DVR 시스템 구현 및 실시간 처리 속도 향상에 관한 연구이다. DVR 실시간 처리를 위해 DSP 프로세서와 데이터 전송은 DSP의 QDMA를 사용하였고, 보안을 위해 워터마킹과 상용성을 위해 MPEG4 압축기법을 적용한다. 알고리즘은 연산 시 프로세서의 내부 메모리에서 처리하는 구조와 반복 연산을 하는 부분을 DSP 프로세서 구조에 적합한 형태로 최적화 하여 구현 한다. 실험 결과, 구현된 DVR 시스템에서 D1 크기의 동영상에 워터마킹과 MPEG4 알고리즘 연산 결과 프레임 당 연산에 소요되는 처리 시간이 12%이상 개선이 되었다.

### ABSTRACT

The DVR system realization with watermarking and MPEG-4 for real time processing speed improvement is presented in this paper. For the real time processing the system is used the DSP processor, Quick DMA for data transmission, watermarking for security and MPEG-4 compression for facility. The algorithms are that the operational structure has the internal memory of processor, and the optimal realization is suitable to form the DSP processor structure r processed for the iterative operations. The experimental result shows the real time processing is improved 12% over for the D1 image in comparison with the other system.

### 키워드

DVR, Watermarking, QDMA, MPEG-4, Real-time, DSP Processor

### 1. 서 론

DVR 시스템은 1997년 PC 기반으로 상업화에 성공하여 현재 임베디드 시스템으로까지 발달이 되었다. 또한, 단순한 녹화기능에서 보안 시스템 영역까지 적용이 되고 있는 실정이다. 따라서 DVR 시스템을 구현하기 위해 영상 압축 기술 위주로 해서 시스템이 개발이 되어 왔으나 보안 및

영상 데이터에 대한 보호를 위해 보안 기술을 추가하여 시스템을 구성을 하게 되었다. 본 논문에서는 임베디드 DVR 시스템을 구현하기 위해 압축 기술로는 MPEG4 알고리즘과 보안 기술로는 비디오 워터마킹을 적용하였다.[1-7] DVR 시스템에서 압축 기술과 보안 기술을 연산하기 위해 DSP 프로세서를 사용하였고, 상위 호스트와의 통신과 저장 매체를 제어하기 위해 Host 프로세

서를 사용하여 시스템을 구성하였다. 또한 MPEG 4 압축 알고리즘과 워터마킹 알고리즘을 D1 영상 크기의 동영상을 1초에 30 프레임 실시간 처리를 하기 위해 DSP 프로세서의 구조에 적합한 형태로 데이터 처리 방법과 알고리즘에서 반복 루프 및 판단 루프 등의 코드를 최적의 형태로 수정하였다. 이러한 여러 가지의 기법들을 적용하여 실시간 임베디드 DVR 시스템의 처리 속도를 향상시키고자 한다.

## II. 시스템 구현

### II-1 시스템 구성

시스템 구성은 그림 1과 같이 비디오 및 OSD(On-Screen Display)를 처리하는 비디오 모듈과 알고리즘들을 연산하는 DSP 모듈, 그리고 상위 통신 및 저장 매체를 제어하는 호스트 모듈로 구성되는 임베디드 DVR 시스템이다.

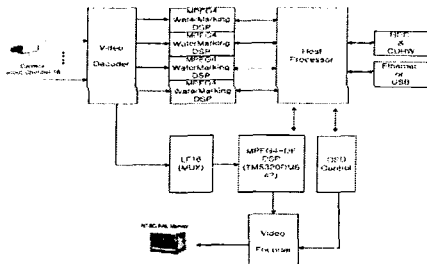


그림 1. 임베디드 DVR 시스템

#### 1) 비디오 모듈

비디오 모듈은 그림2와 같이 디코더부분과 인코더부분으로 구성이 된다. 디코더부분은 입력 신호로 16개 CCD 카메라로부터 영상을 력하기 위한 비디오 입력 모듈로 구성이 된다. 입력되는 비디오 신호들의 동기화 및 제어할 수 있도록 FPGA를 이용한 부분과 입력된 신호들을 DSP 프로세서에 정합되는 모듈이다. 인코더부분은 카메라로부터 입력된 영상을 직접 모니터에 출력하는 라이브 영상과 호스트 프로세서로부터 영상 압축 데이터를 DSP에서 복원한 비디오 데이터를 아날로그 모니터에 출력하기 위한 비디오 출력 모듈로서 구성이 된다.

#### 2) DSP 모듈

DSP 모듈은 그림 1에서와 같이 입력된 영상 데이터를 압축 알고리즘과 워터마킹 알고리즘을

수행한 후, 생성된 영상 압축 데이터를 호스트 프로세서에 데이터 전송을 하는 인코더 블록과 호스트로부터 영상 압축된 데이터를 받아서 복원하여 비디오 출력 단에 데이터를 내보내는 디코더 블록으로 구분이 된다.

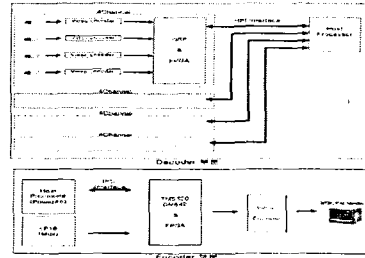


그림 2. 비디오 모듈

인코더 블록에서 하나의 DSP가 각각 4개의 비디오 신호 받아서 처리할 수 있는 구조로서 총 4개의 DSP를 이용하여 16채널 비디오 입력 신호를 받아서 처리하는 구조로 되었다. 디코더 블록은 하나의 DSP로 16개의 영상 압축 데이터를 복원할 수 있는 구조로 설계하였다. 사용된 DSP 프로세서는 64비트 연산을 하는 프로세서이며 영상 처리에 적합하도록 디자인된 전용 프로세서를 사용하였다. 여기서 사용된 DSP 프로세서의 구조에서 영상 입력 신호를 받아들이는 부분을 프로세서 내부에 구성을 하므로 외부 필드메모리에 저장하고 저장된 데이터를 읽어서 들이는 시간에 대한 입출력 액세스 시간을 줄일 수 있는 구조로 되어 있다.

#### 3) 호스트 모듈

그림 3는 호스트 모듈로 DSP 인코더 모듈들로부터 전송되는 영상 압축 데이터들을 저장 매체인 하드디스크나 CDROM등에 저장한다. 그리고 상위 네트워크로 데이터를 전송하는 기능을 수행하고 저장된 영상 압축 데이터를 복원하기 위해 DSP의 디코더 모듈로 데이터를 전송하는 기능을 수행한다.

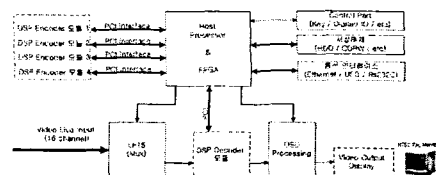


그림 3. 호스트 모듈

여기서 DSP 모듈간의 인터페이스와 저장 매체 간의 인터페이스는 실시간 처리를 위해 PCI 버스를 사용한다.

II-2 알고리즘 구현 및 처리 속도 개선

D1 크기의 동영상은 1초당 30 프레임 처리를 하기 위해 알고리즘을 DSP 프로세서의 구조에 적합한 형태로 변경하여 처리 속도를 개선시킨다.

- 1) 데이터 최적화 : 사용된 DSP프로세서는 정수형으로 연산을 수행하므로 알고리즘 구현을 위해 연산에 사용되는 모든 데이터 형태를 정수형으로 변환하였다. 변환하는 방법은 일반적으로 큐포맷이라는 방법을 가장 많이 사용한다. 즉, 실수형의 데이터를 소수점 몇째 자리까지 연산할 것인지 정해서 정수형으로 변환 시키는 방법이다.
- 2) 데이터 전송 최적화 : CPU가 데이터를 연산하기 위해 데이터를 억세스하는데 걸리는 사이클이 최적의 상태가 14사이클을 필요하게 된다. 따라서 외부 메모리의 억세스를 줄여 억세스 시간을 최소화하기 위해 캐쉬나 QDMA 기능을 이용하였다. 먼저 캐쉬의 경우 캐쉬 제어 로직에 의해 하드웨어적으로 데이터들이 억세스가 되고, QDMA의 경우 데이터를 전송할 때 데이터들이 버스트 형태인 매 사이클 마다 데이터를 전송하기 때문에 CPU 데이터들을 제어하는 것보다 빠르게 데이터를 억세스 해서 연산을 수행할 수 있다. 연산은 주로 프로세서 내부 메모리에서 연산을 할 수 있는 구조로 변경하였으면 이때 QDMA와 캐쉬를 이용 한다.
- 3) 반복 루프 최소화 : C 코드에서 for나 while등의 반복 횟수를 최소한으로 소스 코드를 재구성을 하였다. for나 while등의 반복문은 기본적으로 반복하기 위한 횟수를 판단하기 위해 판단문인 if 등의 조건 비교 루틴을 수행하는데 DSP 프로세서의 구성상 분기시키는 명령이 6사이클이 소유가 된다. 따라서 반복 횟수를 줄이면 그만큼 분기 명령 실행 횟수가 줄어들기 때문에 그만큼 처리 시간을 줄일수있다.
- 4) DSP의 코드 최적화 : DSP 프로세서는 내부에서 연산하는 방법이 파이브라인 개념을 도입하여 하드웨어 파이브 구조와 소프트웨어 파이브 구조로 연산을 수행한다. 하드웨어 파이브는 일반 프로세서에가 가지는 ALU의 기능을 4개의 형태로 분리하여 수행할 수 있도록 하였는데 이를 유닛이라고 부르면 총 8개의 유닛으로 구성이 되어 있

다. 그러므로 1사이클에 8개의 명령을 동시에 수행할 수 있는 구조로 하드웨어 파이브가 구성이 되어 있다. 따라서 실행 코드가 1 사이클에 8개의 명령을 수행할 수 있는 구조로 실행 코드를 생성을 시켰다. 예로,  $c += a[i] * b[i]$  연산하기 위해 실행코드를 만들었을 경우 1회 수행하는데 16사이클이 필요하여 40회 반복을 할 경우 640사이클이 수행이 된다. 이를 소프트웨어 파이브의 형태로 실행코드를 변경을 하면 28 사이클 만에 결과를 얻을 수 있다. 따라서 알고리즘의 실행 코드를 최대한 소프트웨어 파이브 구조로 변경한다.

5) 비디오 데이터 처리 최적화 : 카메라로부터 들어오는 영상은 동영상을 구현할 때 비월주사 즉, interlace방식을 사용하고 디지털 영상은 순차 주사 즉, Non-interlace방식을 채택하는데, 아날로그 영상을 디지털 영상으로 변환 때는 A/D 변환 이외에 De-interlace과정이 필요하게 된다. 아날로그 영상은 1 프레임을 홀수 필드와 짝수 필드로 각각 표현을 하는데, 아날로그 영상의 경우 홀수 필드 204개, 짝수 필드 240개로 1초당 60 필드로 표현을 한다. 디지털 영상의 경우, 홀수 필드 240개, 짝수 필드 240개로 두 개를 합하여 480 라인을 1초에 30프레임을 표현한다. 움직이는 동영상의 경우 홀수 필드와 짝수 필드 간에 1/60초의 시간 겹의 문제점이 발생이 된다. 이를 해결하기 위한 알고리즘과 그 외에 어두운 환경에 DVR 녹화 시 카메라로부터 들어오는 불규칙한 노이즈, 움직임이 많은 D1 사이즈 영상의 경우 Odd 필드와 Even 필드의 불일치 등으로 추가되는 알고리즘과 영상의 화질을 개선하기 위해 추가되는 알고리즘, 입력 영상이 불완전한 경우 자동으로 녹화를 중지하는 기능 등을 프로세서에 맞게 구현 한다.

6) MPEG4상에서 움직임 벡터 연산 최적화 : 움직임 벡터 탐색의 경우 MPEG4 알고리즘에서 전체 계산량의 30-50 %까지 소요 되므로 정확한 움직임 벡터 탐색시 비트율 감소에 결정적인 역할을 한다. 예로, 16x16 매크로 블록으로 Motion Vector를 위한 계산량은 하나의 연산을 하는데  $31 \times 31 \times (16 \times 16) = 246K$  클럭이 필요하게 되므로 뺄셈, 절대 값, 덧셈을 수행하게 되면 총 738K 사이클이 필요하며, 영상 크기가 CIF(320x240)이면 약 221M 사이클이 필요로 하게 된다. 따라서 이 부분을 프로세서의 내부에서 연산을 수행할 수 있는 구조로 변경을 하였고 전체 처리 속도가 개

선이 되었다.

7) 워터마킹 구현 최적화 : 워터 마킹 구현 시 일반적으로 난수를 발생하여 처리하나 실시간 처리를 위해서 난수를 발생하지 않았다. 영상에 존재하는 픽셀들의 형태에 따라 특정 표시를 만들었다. 픽셀들이 모인 평면 공간으로 만들어져야만 표현이 가능하며, 인접한 픽셀들은 거의 비슷한 값을 가지게 된다. 이러한 특성은 비슷한 값을 가지는 픽셀들 간의 미세한 값의 변화는 사람의 눈으로 확인하기 어렵다는 특성을 이용한 특정 표시를 이용하여 워터마크를 삽입하여 처리 속도를 개선한다.

### III. 실험 및 고찰

그림 4는 16 채널 임베이드드 DVR 시스템을 구현한 보드이다.

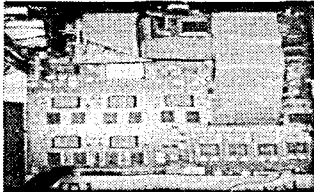


그림 4. 16 채널 임베이드드 DVR 시스템

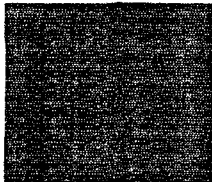


그림 5. 60% 이상의 테스트 동영상

표 1. 실험 결과

실험조건	기존 시스템	구현 시스템
60% 움직임이 있는 영상	29.5 ms	26.2 ms
데이터 최적화	29.5 ms	0.1ms
데이터 전송 최적화	29.5 ms	0.6ms
반복 루프 최소화	29.5 ms	0.3ms
DSP의 코드 최적화	29.5 ms	0.7ms
비디오 데이터 처리 최적화	29.5 ms	0.1ms
움직임 벡터 연산 최적화	29.5 ms	0.5ms
워터마킹 구현 최적화	29.5 ms	1.0ms

구현된 임베이드드 DVR시스템으로 MPEG4 압

축 알고리즘과 워터마킹 알고리즘의 수행 처리 속도를 개선하기 위해 여러 형태의 방법으로 DSP 프로세서에 적합한 형태로 코드를 재구성하였다. 재구성한 코드를 수행한 결과 D1 크기의 움직임이 60%이상인 동영상에서도 1초당 30 프레임 실시간 처리를 구현하였다. 1 프레임 연산에 소요된 시간은 26.2ms가 소요되었다. 실험에 사용된 60% 이상의 움직임을 갖는 동영상은 그림 5와 같은 테스트 영상을 사용하였다. 표1 실험 결과에서 실시간 시스템 구현시 사용되는 프로세서에 적합한 형태로 최소한의 연산 과정을 수행하기 위한 구현 방법에 따라 처리속도에 영향을 주는 것을 알 수 있다. 워터 마킹 알고리즘을 좀 더 최적화하여 기능 추가시 실시간으로 처리가 가능하도록 추가적인 개선 작업이 필요하다.

### IV. 결론

본 논문에서는 구현된 16채널 임베이드드 DVR 시스템에서 MPEG4 알고리즘과 워터 마킹 알고리즘을 적용하여 실시간 시스템을 구현하였다. 실험결과 DSP 프로세서에 적합한 코드를 재구성하여 구현하여 전체적이 연산 시간이 약 3.3ms정도 개선되어 1초당 30프레임 실시간 처리가 가능하도록 개선이 되었다. 향후, 임베이드드 DVR시스템이 보안 시스템에 적용이 되므로 기존에 사용되었던 워터마킹보다 강인한 워터마킹 연구와 다양한 동영상의 실험을 수행 할 필요가 있다.

### 참고문헌

- [1] Rao K.R. and Hwang J. J., "Techniques and Standards for Image, Video, and Audio Coding", Prentice-Hall, USA, 1996
- [2] Iain E.G. Richardson, "H.264and MPEG4", WILEY, 2003
- [3] Hartung F. and B. Girod, "Watermarking of Uncompressed and Compressed Video", Signal Processing, vol. 66, no. 3 (Special issue on Watermarking), pp. 283-301, May, 1998.
- [4] Texas Instrumnets, "TMS320C6000 CPU and

Instruction Set Reference Guide", 2003

- [5] 김자환, 류광렬, "DSP 프로세서를 이용한 Digital Video Recorder System 구현에 관한 연구", 한국해양정보통신학회. 2004
- [6] 김자환, 류광렬, "DSP기반 실시간 다채널 VA DVR 감시시스템 설계 구현", 한국정보기술학회. 2005
- [7] 김자환, 류광렬, "비디오 위터마킹을 이용한 실시간 DVR 시스템 구현", 한국정보기술학회. 2005