

# 멀티코어를 이용한 안전하고 에너지 효율적인 MPEG 인코딩\*

이 성 주,<sup>†</sup> 이 은 지, 홍 승 우, 최 한 나, 정 용 화<sup>‡</sup>  
고려대학교

## Secure and Energy-Efficient MPEG Encoding using Multicore Platforms\*

Sungju Lee,<sup>†</sup> Eunji Lee, Seungwoo Hong, Hanna Choi, Yongwha Chung<sup>‡</sup>  
Korea University

### 요 약

컨텐츠 및 프라이버시 보호는 최근 보급되기 시작한 네트워크 기반 비디오 감시 시스템의 주요한 이슈가 되고 있다. 특히, 배터리로 동작하는 임베디드 시스템 기반의 비디오 센서가 압축 및 암호화 과정을 실시간으로 처리해야 하는 환경에서 실시간 요구사항과 에너지 효율성을 동시에 만족시키는 것은 쉽지 않은 문제이다. 본 논문에서는 비디오 감시 데이터를 압축 및 암호화하는 멀티코어 기반 솔루션을 제안하고, 제안 방법의 효율성을 실시간 처리와 에너지 소비 관점에서 평가한다. MPEG2/AES를 이용한 실험 결과, 실시간을 만족하는 범위 내에서 멀티코어 기반의 제안 방법이 통상적인 싱글코어 기반의 방법에 비하여 최대 30배까지 에너지 효율성을 개선할 수 있음을 확인하였다.

### ABSTRACT

Content security and privacy protection are important issues in emerging network-based video surveillance applications. Especially, satisfying both real-time constraint and energy efficiency with embedded system-based video sensors is challenging since the battery-operated sensors need to compress and protect video content in real-time. In this paper, we propose a multicore-based solution to compress and protect video surveillance data, and evaluate the effectiveness of the solution in terms of both real-time constraint and energy efficiency. Based on the experimental results with MPEG2/AES software, we confirm that the multicore-based solution can improve the energy efficiency of a singlecore-based solution by a factor of 30 under the real-time constraint.

**Keywords:** Protect Video Data, Multicore Platforms, Energy-Efficient

## 1. 서 론

비디오 감시 시스템은 아날로그 기반의 CCTV 시스템에서, Digital Video Recording (DVR), 그리고

최근의 유무선 네트워크 기반 디지털 시스템으로 진화하고 있다 [1]. 이렇게 비디오 감시 시스템이 네트워크로 연결됨에 따라, 네트워크를 통하여 전송되는 비디오 데이터에 대한 컨텐츠 및 프라이버시 보호에 대한 요구사항이 증가하고 있는 실정이다 [2]. 즉, 대부분의 비디오 감시 시스템에서의 전송 데이터가 방대하여 인코딩 단계에서의 압축 및 전송하는 비디오 데이터의 암호화에 대한 실시간 처리가 중요한 이슈가 되고 있다. 또한, 비디오 센서가 배터리로 동작하는 경우에는 에너지 효율성도 같이 고려해야 하는 주요 이슈가 된다.

접수일(2010년 4월 9일), 수정일(2010년 5월 10일),  
게재확정일(2010년 6월 10일)

\* 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국  
연구재단의 지원을 받아 수행된 연구임.

(No. 2009-0074784)

<sup>†</sup> 주저자, peacefeel@korea.ac.kr

<sup>‡</sup> 교신저자, ychungy@korea.re.kr

최근 멀티코어 프로세서가 출시되어 PC나 서버뿐만 아니라 임베디드 시스템용으로 활용되면서 [3], 멀티코어를 이용한 병렬처리 기법이 실시간성과 에너지 효율성을 동시에 만족할 수 있는 대안이 될 수 있다. 사실 멀티코어를 이용한 다중처리 (여러 개의 태스크를 동시에 수행하여 처리율을 개선) [4]와 비디오 압축에 대한 병렬처리 (하나의 태스크를 동시에 수행하여 응답시간을 개선) 서버 [5]에 대해서는 많은 연구가 수행되었다. 그러나, 멀티코어를 이용하여 안전한 비디오 감시 시스템을 구현하고, 병렬처리에 의하여 실시간성과 에너지 효율성을 동시에 만족하는 연구 결과는 아직 발표되고 있지 않은 실정이다.

본 논문에서는 비디오 감시 데이터를 압축 및 보호하기 위하여 멀티코어를 이용한 병렬처리 방법을 제안하고, 실시간성과 에너지 효율성 측면에서 그 타당성을 분석한다. 효과적인 병렬처리 솔루션을 구하기 위하여, 먼저 MPEG2와 AES [6]의 특성에 대하여 분석하였고, P/B-프레임의 압축 부하 (특히, "motion estimation" 함수)가 매우 불규칙적이라는 점을 확인하였다. 그리고 병렬 오버헤드를 최소화하기 위하여 작업부하가 상대적으로 예측 가능한 계산은 정적 (즉, 컴파일시 수행 코어를 결정)으로, 부하가 불규칙한 계산은 동적 (즉, 런타임시 수행 코어를 결정)으로 부하를 분산하였다. 제안된 방법은 이식성을 위하여 Pthread [7]로 병렬프로그램화 하였고, AMD Phenom II X4 955 멀티코어 프로세서에서 실험하였다. 마지막으로, 병렬처리 솔루션의 에너지 효율성을 평가하기 위해 디지털 멀티미터 WT210 [8]를 이용하여 전력 소모량을 측정하였다.

실험 결과, 멀티코어를 이용한 병렬처리 솔루션은 통상적인 싱글코어 솔루션에 비하여 에너지 효율성이 2배 이상 개선될 수 있음을 확인하였다. 또한, 본 논문에서 가정한 비디오 센서의 경우 연속적인 입력 데이터에 대하여 동일한 태스크 (압축 및 암호)만을 수행하는 임베디드 시스템이므로, Dynamic Voltage and Frequency Scaling (DVFS) [9] 기법을 적용하기 매우 용이하다. 따라서, 적절한 전압/주파수를 설정하면, 실시간성을 만족하면서 에너지 효율성을 30배까지 개선할 수 있음을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서는 MPEG2 특성에 대하여 설명하고, 비디오 압축 및 암호화를 멀티코어상에서 병렬로 처리하는 제안 방법을 3장에서 기술한다. 4장에서 제안 방법에 대한 실험 결과를 설명하고, 5장에서 결론을 맺는다.

## II. MPEG2 특성

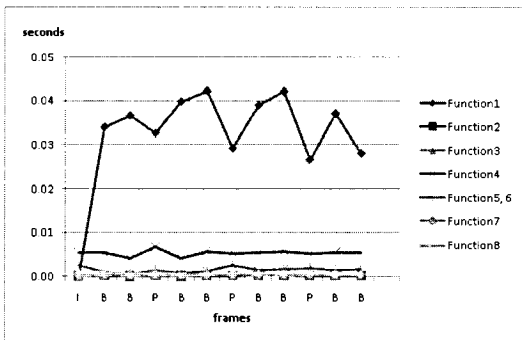
일반적인 MPEG 인코더의 구조는 I-, P-, B-프레임의 조합인 GOP (Group of Picture)로 구성된다 [10]. I-프레임은 정지영상의 표준인 JPEG와 같은 방법으로 인코딩되고, P- 혹은 B-프레임은 I-프레임의 데이터에 의존하여 움직임 보상을 수행하여 인코딩된다. 또한, 가까운 프레임에서 움직임 예측을 통한 유사한 블록을 찾지 못했을 경우, 해당 매크로 블록은 영상 내 부호화 모드로써 인코딩된다. 위 매크로 블록을 I-블록이라 한다. 즉, I-블록은 영상에서 많은 움직임이 있는 P- 또는 B-프레임의 매크로 블록이고, 표준 JPEG와 같은 방법으로 인코딩된다.

MPEG2를 멀티코어에서 효과적으로 병렬처리하기 위하여, 먼저 MPEG2 함수의 계산 특성을 분석한다 (표 1 참조). 첫번째 함수인 "motion estimation"은 움직임 예측에 따라 각 매크로블록을 어떻게 인코딩할지를 결정하며, 움직임 보상에 사용되는 움직임 벡터를 계산한다. 두번째 함수 "predict"는 움직임 보상을, 세번째 함수 "dct\_type\_estimation"은 DCT 변환 형태를, 네번째 함수 "transform"은 DCT 변환을 나타낸다. 다섯번째 함수 "quant"는 양자화를, 여섯번째 함수 "putpict"는 가변장 인코딩을, 일곱번째 함수인 "iquant"는 역양자화를, 마지막 "itransform"은 IDCT를 의미한다.

프레임별 각 함수의 수행시간을 분석하기 위해 비디오 감시 시스템용으로 많이 활용되는 Hallway 데이터 [11]를 AMD Phenom II X4 955 싱글코어에서 인코딩하였다. 그림 1에 IBBPBBPBBPBB 순서 (즉, 352×240 픽셀 크기를 갖는 12개 프레임)를 갖

(표 1) MPEG2 함수 (10)

Index	Function	Description
1	motion_estimation	Motion prediction, make MV, Set up block type
2	predict	Motion compensation using MV
3	dct_type_estimation	Calculate DCT type
4	transform	DCT
5	quant	Quantization
6	putpict	Variable length coding, Write the result on file
7	iquant	Dequantization
8	itransform	IDCT



(그림 1) MPEG2 각 함수의 I/P/B-프레임별 수행시간

는 하나의 GOP에 대한 수행시간을 나타내었다. 특히, P/B 프레임의 첫번째 함수에서 가장 많은 계산을 요구하며, 그 계산양도 매크로블록의 특성 (즉, intra, forward, backward 인코딩)에 따라 매우 상이하다는 것을 확인할 수 있다. 즉, 작업부하가 예측불가능하며, 이러한 부하 분산 문제를 효과적으로 해결하기 위해서는 오버헤드가 크더라도 작업부하를 동적으로 할당하여야 한다. 반면, P/B 프레임의 나머지 함수나 I 프레임은 계산양 자체가 크지 않을 뿐만 아니라 그 차이도 크지 않아, 오버헤드가 적은 정적인 부하 분산으로도 효과적인 병렬처리가 가능함을 예측할 수 있다.

### III. 제안방법

MPEG과 같은 영상 압축 기법은 대부분 대용량의 데이터를 처리해야하기 때문에 많은 계산량이 요구되며, 이러한 계산을 병렬처리하는 기존 방법은 태스크를 프로세서의 수만큼 나누어 정적으로 각 프로세서에 할당하는 방법이 이용되고 있다. 이러한 병렬처리 서버에서의 정적 할당 방식은 구현이 동적 할당 방식보다 쉬우며 분기 명령어가 많이 수행되지 않는 규격화된 계산특성을 갖는 응용에 적용할 경우 뛰어난 성능을 제공할 수 있다는 장점이 있다. 그러나, MPEG과 같은 영상 압축 기법들의 계산특성은 분기 명령어가 많이 수행되며 규격화된 계산특성을 갖지 않기 때문에 부하 불균형 문제가 발생 한다.

정적 할당 방식과 달리 동적 할당 방식을 규격화 되지 않은 계산특성을 갖는 응용에 적용할 경우 부하 불균형 문제를 해결할 수 있다. 그러나, 병렬처리 서버에서의 동적 할당 방식의 경우, 프로세서 상호간의 통신 오버헤드가 상대적으로 빈번히 발생하며, 오히려

성능을 저하시킬 수 있다. 이러한 통신 오버헤드 때문에 부하 불균형 문제를 해결하기 위해 동적 할당 방식을 병렬처리 서버에 적용하여 전체적인 성능을 향상 시키기란 어려운 일이지만, 최근 출시된 멀티코어 플랫폼에서는 코어간의 통신 오버헤드가 충분히 작기 때문에 동적 할당 방식을 적용하여 부하 불균형 문제를 해결하면서 동시에 전체적인 성능을 향상시킬 수 있다. 본 논문에서는, MPEG-2에서 많은 계산을 필요로 하는 "motion estimation"에 멀티코어 플랫폼을 이용하여 이러한 동적 할당 방식을 적용한다.

MPEG-2에서 "motion estimation"은 P-와 B-프레임을 생성하며, I-프레임의 경우에는 "motion estimation"이 수행되지 않는다. 따라서 MPEG-2의 프레임 생성 순서에 따라 I-프레임의 경우에는 정적 할당 방식을 적용하고, B-와 P-프레임의 경우에는 동적 할당 방식을 적용하여 "motion estimation"의 부하 불균형 문제를 해결한다. 그림 2 (a)는 I-프레임일 경우, 4-코어 플랫폼에서 수행되는 정적 할당 방식의 프로그램을 보여주고 있다. I-프레임의 경우 "motion estimation"이 수행되지 않기 때문에 B-와 P-프레임을 생성하기 위한 계산시간 보다 짧고, DCT와 IDCT 등의 규격화된 연산특성에 따라 정적 할당 방식만을 적용하여도 수행시간을 단축시킬 수 있다.

그림 2 (b)는 B-나 P-프레임의 경우에 동적 할당 방식을 적용하는 프로그램을 보여주고 있다. B-와 P-프레임일 경우, 작업부하 예측이 부정확한 "motion estimation"이 수행되어 부하 불균형 문제가 발생할 수 있기 때문에, 동적 할당 방식을 적용할 경우 효과적으로 수행될 수 있다.

마지막으로 콘텐츠의 보호를 위하여 본 논문에서는 AES-CTR를 적용하였다. AES는 대칭키 암호로서 블록암호의 표준이고 CTR (Counter)는 임의의 수 (Nonce)와 카운터 (Counter)를 이용해 만든 블록을 일정한 값으로 증가하여 암호화에 사용하기 때문에 블록암호를 스트림암호처럼 사용할 수 있게 해준다. 특히, 키와 Nonce를 미리 계산할 수 있기 때문에 데이터를 실시간으로 안전하게 암호화 할 수 있고, 병렬성이 뛰어나기 때문에 멀티미디어와 같은 콘텐츠 암호에 적합하다. 즉, AES-CTR의 경우 분기 명령어가 많이 수행되지 않으며 블록간 데이터 종속성이 없는 CTR의 경우 병렬처리가 용이 하기 때문에, 본 논문에서는 그림 2 (c)와 같이 AES-CTR의 병렬처리를 위하여 정적 할당 방식을 적용한다. 이 때, AES-CTR는 비디오 데이터가 I-, B-, P-프레임으

로 나뉘어 모두 압축이 된 후, 압축된 데이터를 블록 레벨에서 모두 암호화 한다. 사실, 데이터 종속성이 있는 AES-CCM의 경우에도 4-코어를 이용하여 병렬처리하는데 큰 어려움은 없다 [12]. 또한, 멀티미디어 데이터의 암호화 시간을 줄이기 위하여 멀티미디어 데이터의 일부를 암호화하는 방법 [13,14]들이 제안되고 있다. 본 연구에서는 MPEG2 (인코딩 시간이 상대적으로 많이 소요) 압축된 데이터 전체를 부분 암호화의 적용 없이 단순한 AES-CTR을 적용하여도 실시간성을 만족시킬 수 있어, 보안성 훼손을 감수하면서까지 부분 암호화를 적용하지 않는 것으로 가정한다.

```

Static task assignment I-Frame Encoding ( Frame_Data Frame ) {
    Number_of_Thread = 4;
    Thread_Data Thread[ Number_of_Thread ];
    For( 0 <= index < Number_of_Thread )
        Thread[ index ].Group_of_Block = Frame/Number_of_Thread;
    Create_Threads ( Number_of_Thread, Thread_working, Thread[] );
    Thread_working {
        Motion_Estimation( Thread_Group_of_Block );
        DCT( Thread_Group_of_Block );
        Gauntization( Thread_Group_of_Block );
        Putpicture( Thread_Group_of_Block );
        DeGauntization( Thread_Group_of_Block );
        IDCT( Thread_Group_of_Block );
    }
    Join_Threads ( Number_of_Thread );
    Synchronization( );
}
    
```

(a) I-프레임을 위한 정적 할당 방식

```

Dynamic task assignment B- or P-Frame( Frame_Data Frame ) {
    Number_of_Thread = 4;
    Thread_Data Thread;
    For( 0 <= index < Number_of_Thread )
        Thread[ index ].Group_of_Block = Frame/Number_of_Thread;
    Frame = Unlock;
    Create_Threads ( Number_of_Thread, Thread_working, Thread[] );
    Thread_working {
        For( 0 <= index < Number_of_Total_Block )
            For( Lock( Frame_Block[ index ] ) - Lock_Index+1 :
                Lock( Frame_Block[ index ] );
                if( Didn't_Execute_Motion_Estimation = Frame_Block[ index ] )
                    Motion_Estimation( Frame_Block[ index ] );
                Unlock( Frame_Block[ index ] );
            }
        Synchronization( );
        DCT( Thread_Group_of_Block );
        Gauntization( Thread_Group_of_Block );
        Putpicture( Thread_Group_of_Block );
        DeGauntization( Thread_Group_of_Block );
        IDCT( Thread_Group_of_Block );
    }
    Join_Threads ( Number_of_Thread );
    Synchronization( );
}
    
```

(b) B-, P-프레임을 위한 동적 할당 방식

```

Static task assignment AES-CTR ( Compressed Data )
{
    Number_of_Thread = 4;
    Thread_Data Thread[ Number_of_Thread ];

    For( 0 <= index < Number_of_Thread )
        Thread[ index ].Group_of_Block = Compressed_DATA/Number_of_Thread;
    Create_Threads ( Number_of_Thread, Thread_working, Thread[] );
    Thread_working
        AES-CTR ( Thread_Group_of_Block );
    Join_Threads ( Number_of_Thread );
    Synchronization( );
}
    
```

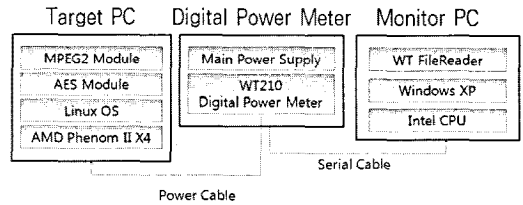
(c) AES-CTR을 위한 정적 할당 방식

(그림 2) 각 I/B/P-프레임과 AES-CTR을 위한 동적 할당 방식 및 정적 할당 방식

## IV. 실험결과

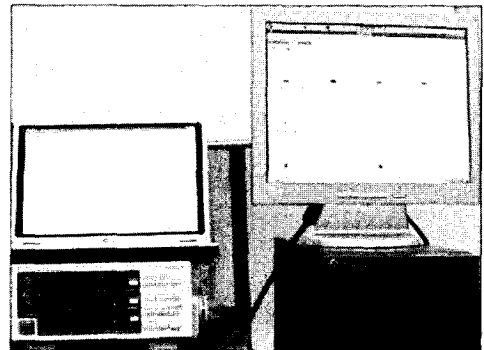
### 4.1 측정환경

본 논문에서는 에너지 효율성을 분석하기 위해 WT210 디지털 파워 미터기를 이용하여 전력을 측정하였다. 그림 3과 같이 WT210은 주전력을 공급해주고, 공급된 파워는 타겟 PC로 전달된다. 또한 동시에 WT210은 공급된 전력을 측정하여 시리얼 통신에 의해 모니터 PC에 데이터 값을 전달해 준다. 타겟 PC는 감시카메라 시스템에서 이용되는 플랫폼으로써 입력영상을 압축 및 암호화하는 MPEG2/AES 모듈이 수행되며, WT210을 이용하여 모듈이 수행될 때의 전력을 측정할 수 있다 [8].



(그림 3) 전력 측정 환경

감시카메라 시스템에서 소비되는 전력은 압축 코덱 방법에 따라 달라지며, 압축은 영상의 품질을 결정하는 중요한 요소이다. 또한 감시카메라 시스템의 단말은 자원 제약적이기 때문에 이러한 압축은 실시간성/영상품질을 고려하여 설계되어야 한다. 이러한 감시카메라 시스템을 고려하여 본 연구에서는 기초적인 연구 단계로 감시카메라 시스템을 구성하였다. 그러나 현재까지 멀티코어 임베디드 플랫폼이 국내에서 아직 출시되지 않았기 때문에 감시카메라 시스템의 카메라 단말



(그림 4) WT210를 이용한 소모전력 측정

을 AMD Phenom II X4 955 4-core Processor (3.2 GHz, RAM 3.0 GB)로 구성하였다. 압축 코덱은 MPEG2를 이용하였고, 비디오 포맷은 초당 25 YUV 프레임으로 구성하였다. 또한 프레임의 크기는 352×240이고 비디오 포맷에서 널리 쓰이는 4:2:0 포맷을 이용하였다 [11]. 그림 4는 실제 실험에 사용된 계측기 WT210과 모니터링 시스템, 그리고 타겟 PC의 모습을 보여주고 있다.

4.2 성능측정

비디오 데이터를 암호화하기 위해서는 일반적으로 MPEG2로 압축을 하여 데이터의 크기를 줄인 후, AES로 암호화한다. 이때, 디코더가 압축된 데이터를 디코딩 할 수 있도록 압축 데이터는 헤더 정보를 포함하고 있으며, 이러한 헤더가 암호화 되면 공격자는 MPEG2의 헤더를 모르기 때문에 암호화된 비디오 데이터를 재생할 수 없다. 그러나 MPEG2로 압축된 데이터에서 헤더의 정보는 표현 할 수 있는 정보가 매크로블록 종류, 프레임의 길이 등 잘 알려진 정보로 구성되기 때문에 공격자는 AES의 키를 알지 못해도 프레임을 재생하는 것이 어렵지 않다. 그림 5는 공격자가 헤더 정보를 알아내어도 I-, B-, P-프레임이 복원되지 않음을 보여주고 있다.

WT210으로부터 측정된 소비전력과 MPEG2/AES가 수행될 때의 수행시간을 측정하여 총 소모되는 에너지를 계산하였다. 표 2는 싱글코어와 멀티코어 플랫폼에서 MPEG2/AES 모듈이 1초 동안 수행되는 시간 및 전력을 나타내며, 소비하는 에너지는 시간과 전력을 곱하여 Joule 단위로 나타내고 있다. 싱글

(표 2) CIF 데이터 처리를 위한 수행시간과 에너지 효율성

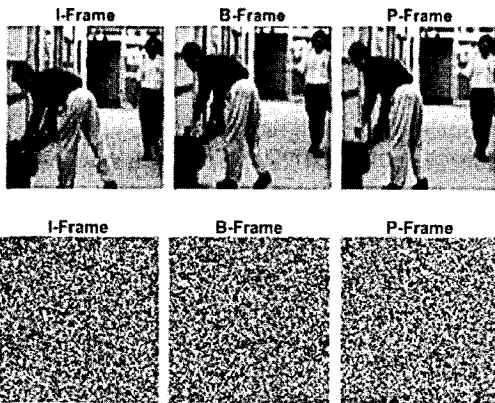
순차처리 소프트웨어				병렬처리 소프트웨어			
Power (W)	Delay (Sec)	Energy (J)	ED <sup>2</sup> (JS <sup>2</sup> )	Power (W)	Delay (Sec)	Energy (J)	ED <sup>2</sup> (JS <sup>2</sup> )
18	1.39	23.12	42.06	28	0.40	10.66	1.62

코어에서 MPEG2/AES의 경우, 1초 동안의 입력영상을 압축 및 암호화하는데 필요한 시간이 1.39초이기 때문에 실시간을 만족하지 않지만, 멀티코어 플랫폼에서는 0.4초로 실시간성을 만족하고 있다. 또한, 전력의 경우, 싱글코어에서는 평균 18W 정도의 전력을 요구하고 있지만, 멀티코어의 경우 28W로 오히려 싱글코어보다 높은 전력이 요구되었다. 그러나 에너지는 전력과 수행시간의 곱으로 계산되기 때문에 싱글코어의 에너지는 23J로 멀티코어의 10J 보다 많은 에너지를 소비함을 확인할 수 있다. 따라서 멀티코어 플랫폼을 이용한 감시카메라 시스템은 싱글코어 플랫폼을 이용할 때 보다 실시간을 만족할 수 있을 뿐만 아니라 에너지 효율성을 2배 이상 증가시켰다.

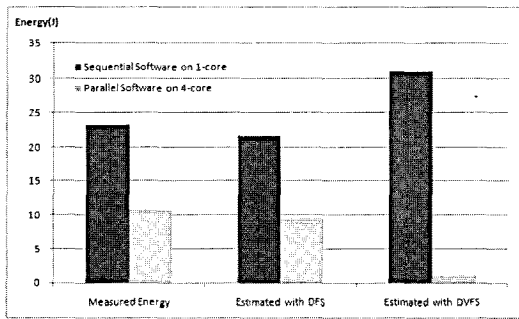
또한, 1초 동안 입력된 영상을 압축 및 암호화하는데 걸리는 시간이 0.4초이기 때문에, CPU 주파수를 수행시간이 1초가 넘지 않도록 낮추어 전력소비와 에너지의 효율성을 증가시킬 수 있다. 이렇게 CPU 주파수를 낮추어 전력소비와 에너지 효율성을 증가시키는 동적 주파수 전압 변환 기술 (Dynamic Voltage/Frequency Scaling)이 많이 연구되고 있으며, 멀티코어 플랫폼에서 이러한 DVFS 기술을 적용하면 에너지 효율성을 극대화시킬 수 있다.

4.3 에너지 효율성 분석

에너지 효율성을 향상시키기 위해 DVFS 기술을 감시카메라 시스템에 적용할 수 있다. 이때의 DVFS의 효과를 계산하기 위해 일반적으로 이용되는 에너지 소비모델은 수식 1 및 2와 같고, 이때 CI는 상수, F는 CPU의 주파수, V는 공급전압, T는 수행시간이다. 또한, 공급전압 V는 CPU의 주파수 F와 상수 C2의 곱으로 표현 될 수 있다. 즉, 멀티코어 시스템의 F는 3.2GHz, V는 1.425V, T는 1.39초일 때, WT210에 의해 E(=W·T)를 구하여 CI를 계산할 수 있다. 또한, V와 F의 관계는 수식 2에 따라 C2를 계산하여 구할 수 있다. 따라서 수식 1과 2를 이용하여, 주파수 F와 전압 V를 조절하여 전력의 효율성을 증가시키는 DVFS와 같은 기술을 적용할 때의 에너



(그림 5) AES로 암호화된 I/B/P-프레임과-프레임의 원영상과 복호화된 영상



(그림 6) DVFS 기술을 이용할 때의 에너지 효율성

지 효율성을 예측할 수 있다.

$$E = C1 \cdot F \cdot V^2 \cdot T \quad (1)$$

$$V = C2 \cdot F \quad (2)$$

싱글코어에서는 CPU 주파수를 높게 설정하여야 실시간으로 압축 및 암호화를 수행할 수 있다. 따라서 CPU의 동작 클럭을 3.9MHz로 증가시킨다면, 싱글코어 기반의 감시카메라 시스템도 실시간성을 만족할 수 있다. 또한, 에너지 효율성 측면에서도 그림 6과 같이, 수행시간이 감소하기 때문에 총 소비되는 에너지 역시 감소할 수 있다. 그러나 주파수가 높아지면 요구하는 전압 역시 높아지므로 결국 에너지 소비량은 증가하게 된다.

반면, 멀티코어 플랫폼에서는 0.4초로 실시간을 충분히 만족하고 있기 때문에 CPU 주파수를 1.1GHz로 낮추어 전력소비를 감소시킬 수 있다. 이 경우, CPU에서 요구하는 공급전압 역시 낮아지므로, 만약 1.1GHz와 0.475V로 주파수와 공급전압이 낮아지게 되면, DVFS 기술을 적용한 멀티코어 플랫폼의 에너지 효율성은 적용하지 않았을 때 보다 10배까지 증가하게 된다. 또한 싱글코어와 비교하였을 때는, 30배 정도로 효율성을 개선할 수 있다.

## V. 결 론

최근 임베디드 시스템을 기반으로 한 비디오 센서가 감시시스템 등의 응용에 이용되기 시작하면서, 컨테츠 및 프라이버시 보호가 주요 이슈가 되고 있다. 본 논문에서는 비디오 감시 데이터를 압축하고 보호하기 위하여 멀티코어 기반의 솔루션을 제안하였다. 먼저 MPEG2/AES의 계산 특성을 분석한 후, I-프레

임 압축은 정적, B/P-프레임의 "motion estimation"은 동적, 그리고 B/P-프레임의 나머지 함수와 암호화 부분은 정적으로 부하를 분산처리 하였다.

실험 결과에 의하면, 제안된 멀티코어 기반 솔루션은 통상적인 싱글코어 기반 솔루션에 비하여 2배 정도 개선된 에너지 효율성을 제공할 수 있음을 확인하였다. 특히, 저전력 처리를 위하여 최근 활발히 연구되고 있는 DVFS 기법을 적용한다면, 멀티코어 기반 솔루션은 실시간 조건을 만족하면서 에너지 효율성을 30배 이상 추가로 개선할 수 있을 것으로 기대된다. 따라서 PKI 등 계산요구량이 많은 정보보호 응용에 대하여 멀티코어를 이용한 병렬처리 기법은 처리시간 뿐만 아니라 에너지 효율성 측면에서도 좋은 솔루션이 될 수 있을 것으로 예상된다.

## 참고문헌

- [1] F. Nilsson and A. Communications, *Intelligent Network Video : Understanding Modern Video Surveillance Systems*, Taylor & Francis, Jan. 2008.
- [2] B. Furth and D. Kirovshi, *Multimedia Security Handbook*, CRC Press, Dec. 2004.
- [3] M. Levy and T. Conte, "Embedded Multicore Processors and Systems," *IEEE Micro*, vol. 29, no. 3, pp. 7-9, May 2009.
- [4] J. Hennessy and D. Patterson, *Computer Architecture : A Quantitative Approach-Fourth Edition*, Elsevier, Dec. 2007.
- [5] I. Ahmad, Y. He, and M. Liou, "Video Compression with Parallel Processing," *Parallel Computing*, vol. 28, no. 7, pp. 1039-1078, Aug. 2002.
- [6] M. Dworkin, "Recommendation for Block Cipher Modes of Operation - Methods and Techniques," *NIST Special Publication 800-38A*, Dec. 2001.
- [7] B. Barney, "POSIX Threads Programming," <http://www.llnl.gov/computing/tutorials/pthreads>, 2006.
- [8] N. Hirofumi, N. Naoya, and T. Katsuya, "WT210/WT230 Digital Power Meters,"

- Yokogawa Technical Report 35, Yokogawa Electric Corporation, Apr. 2003.
- [9] X. Huang, K. Li, and R. Li, "A Energy Efficient Scheduling Base on Dynamic Voltage and Frequency Scaling for Multi-core Embedded Real-Time System," Algorithms and Architectures for Parallel Processing, LNCS 5574, pp. 137-145, 2009.
- [10] J. Watkinson, MPEG Handbook, 2nd Ed., Focal Press, Nov. 2004.
- [11] <http://media.xiph.org/video/derf/>
- [12] 이은지, 이성주, 홍승우, 최한나, 최우용, 정용화, 민병기, "멀티코어를 이용한 AES-CCM 병렬처리," 한국정보보호학회 하계학술대회발표집, pp. 199-202, 2009년 6월.
- [13] T. Lookabaugh, D.C. Sicker, D.M. Keaton, W.Y. Guo, and I. Vedula, "Security Analysis of Selectively Encrypted MPEG2 Streams," Proc. of Multimedia Systems and Applications, pp. 10-21, Sep. 2003.
- [14] 정서현, 이은지, 이성주, 정용화, 민병기, "모바일 콘텐츠 보호를 위한 비트 레벨의 부분 암호화," 한국정보보호학회 영남지부학술대회발표집, pp. 68-72, 2010년 4월.

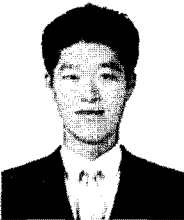
### 〈著者紹介〉



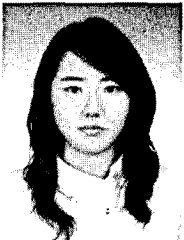
이 성 주 (Sungju Lee) 학생회원  
 2006년 2월: 고려대학교 전산학과 학사  
 2006년 3월~2008년 2월: 고려대학교 전산학과 석사  
 2008년 3월~현재: 고려대학교 전산학과 박사과정  
 <관심분야> 멀티코어, 에너지 효율성, 정보보호



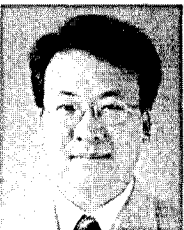
이 은 지 (Eunji Lee) 학생회원  
 2009년 2월: 고려대학교 전산학과 학사  
 2009년 3월~현재: 고려대학교 전산학과 석사과정  
 <관심분야> 멀티코어, 정보보호



홍 승 우 (Seungwoo Hong) 학생회원  
 2008년 2월: 고려대학교 전산학과 학사  
 2008년 3월~2010년 2월: 고려대학교 전산학과 석사  
 2010년 3월~현재: (주)슈프리마 연구원  
 <관심분야> 에너지 효율성, 생체인식



최 한 나 (Hanna Choi) 학생회원  
 2008년 2월: 고려대학교 전산학과 학사  
 2008년 3월~2010년 2월: 고려대학교 전산학과 석사  
 2010년 3월~현재: 한국인터넷진흥원  
 <관심분야> 생체인식, 정보보호



정 용 화 (Yongwha Chung) 종신회원  
 1984년: 한양대학교 전자통신공학과 학사  
 1986년: 한양대학교 전자통신공학과 석사  
 1997년: 미국 Univ. of Southern California 전기공학과(컴퓨터공학 전공) 박사  
 1986년~2003년: 한국전자통신연구원 생체인식기술연구팀장  
 2003년~현재: 고려대학교 컴퓨터정보학과 교수  
 <관심분야> 생체인식, 정보보호, 생체정보 보호