

비디오 응용에서 화질 제약을 고려한 응용 수준의 에너지 최적화 기법

임채석^o 하순회
서울대학교 전기·컴퓨터공학부
{csim^o, sha}@iris.snu.ac.kr

Application-Level Energy Optimization Technique for Video Applications with Video Quality Constraint

Chaeseok Im^o Soonhoi Ha
School of Electrical Engineering and Computer Science, Seoul National University

요 약

이 논문은 프레임 기반의 비디오 응용에 대해서 화질 (video quality) 제약을 고려한 응용 수준 (application-level)에서의 에너지 최적화 기법을 제안한다. 화질과 에너지 소모 사이에는 상관관계 (trade-off)가 있음을 이용하여, 본 논문은 H.263 인코더의 화질을 실시간으로 모니터링해서 프레임 속도 (frame rate)를 자동으로 조절하는 알고리즘을 제안한다. 기존 동적 전압 스케줄링 (DVS: dynamic voltage scheduling) 기법은 유휴 시간 (slack time)을 주어진 것이라고 가정하는 반면, 제안하는 기법은 유휴 시간 자체를 증가시켜서 DVS 기법의 효과를 향상시킨다. 제안하는 기법이 주어진 화질 제약을 만족하며 상당한 에너지 소모를 감소함을 실험을 통해서 알 수 있다.

1. 서 론

최근 화상 전화라든지 비디오카메라 등의 비디오 응용이 이동 환경 (mobile environment)에서 중요하게 부각되고 있다. 이런 응용은 배터리를 기반으로 동작하기 때문에 에너지 소모를 줄이는 것이 중요한 설계 목표가 된다. 이에 많은 저전력 설계 기법이 회로 수준에서 OS 수준까지 다양한 설계 수준에서 제안되었다 [1].

동적 전압 스케줄링 (DVS: dynamic voltage scheduling)은 마이크로프로세서 기반의 시스템에서 가장 주목받는 OS 수준의 저전력 기법이다 [2]. 이는 태스크 수행 후 남은 유휴 시간 (idle time)을 활용하여 프로세서의 공급 전압 및 동작 주파수를 낮춤으로써 전체 에너지 소모를 감소시키는 기법이다. 따라서 이 기법의 핵심은 유휴 시간의 이용률을 최대화해서 주어진 성능 제약 내에서 최대한로 공급 전압을 낮추는 것이다.

지금까지 많은 정적 혹은 동적 DVS 기법들이 제안되었는데 [4, 5, 6, 7, 8, 9, 10], 이들의 한계는 유휴 시간을 100% 활용하지 못한다는 점이다. 이에 우리는 DVS에 버퍼를 활용할 것을 제안하여 이 문제를 해결했다 [11]. 그럼에도 기존 DVS 기법들은 여전히 한계를 가지고 있는데, 주어진 유휴 시간 내에서만 이용률 극대화를 꾀한다는 점이 그것이다. 이에 본 논문은 응용의 알고리즘을 수정하여 유휴 시간 자체를 증가시킴으로써 DVS의 효과를 보다 향상시킬 것을 제안한다. 즉 우리는 응용 수준 (application-level)의 에너지 최적화 기법을 제안하는 것이다.

우리는 비디오 응용에서 화질은 프레임 속도 (frame

rate)와 화면의 움직임 정도 (degree of movement)에 영향을 받는다는 점에 주목해서, 프레임 속도와 화질의 상관관계 (trade-off)를 이용한 에너지 최적화 문제를 정의한다. 즉 주어진 화질 제약 하에서 에너지 소모를 최소화하는 것이다. 본 논문은 H.263 인코더의 화질을 실시간으로 모니터링해서 프레임 속도를 자동으로 조절하는 알고리즘을 제안하여 주어진 화질 제약 하에서 에너지 소모를 최소화한다. 제안하는 기법은 기존 인코더 알고리즘을 조금만 수정하여 적용할 수 있다.

2장에서는 본 논문의 동적인 화질 제약 하에서 에너지 최적화 문제의 기본 아이디어에 대해서 설명하고, 3장에서는 제안하는 기법의 알고리즘을 제시한다. 4장에서는 실험을 통해서 제안하는 기법이 효과적으로 DVS 기법의 효율을 향상시킴을 보이고, 마지막으로 5장에서 결론을 맺는다.

2. 동 기

양적인 분석을 위해서 아래 수식의 PSNR (peak signal noise rate)을 객관적인 화질의 척도로 사용한다.

$$PSNR = 10 \times \log_{10} \left[\frac{1}{E} \times 255^2 \right] [dB]$$

E 는 원본 (original) 프레임과 재생 (reconstructed) 프레임 간의 밝기 요소 (luminance component)의 제곱근 평균 제곱 오차 (root mean square error)이다.

제안하는 기법의 기본 아이디어는 인코딩 시 원본 프레임과 이전 재생 프레임 사이의 PSNR이 미리 정해진 화질 제약보다 크면 프레임을 생략하는 것이다. 이는 실질적인 프레임 속도 (effective frame rate)를 감소시키는

데 따라서 유휴 시간은 증가하고 이를 DVS에 적용하여 에너지 감소를 이룰 수 있다.

그런데 화면의 움직임 정도는 화면의 종류에 따라 달라서, 프레임 속도 감소에 따른 화질의 저하도 그에 따라 달라진다. 그러므로 다양한 종류의 화면 - Coastguard, Hall monitor, Foreman, Container - 으로 구성된 비디오 입력을 고정 프레임 속도 (FFR: fixed frame rate)로 인코딩하면, PSNR 패턴은 그림 1처럼 변화가 크게 된다. 즉, 화면의 움직임 정도가 큰 'Coastguard'와 'Foreman'은 화면의 움직임 정도가 작은 'Hall monitor'와 'Container'보다 상대적으로 낮은 PSNR을 보여준다. 하지만 가변 프레임 속도 (VFR: variable frame rate)로 인코딩하면, 화면의 움직임이 적은 'Hall monitor'와 'Container'에서 프레임 속도를 줄임으로써, 화질 제약은 만족하면서 평균 프레임 속도를 더욱 떨어뜨릴 수 있고 따라서 보다 많은 에너지 감소를 이룰 수 있다.

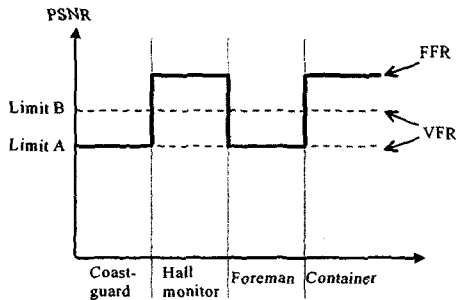


그림 1. 예제 비디오 입력에 대한 PSNR.

3. 제안하는 기법

```

1:  frameInterval = fixedFrameInterval;
2:  for (;;) {
3:    readFrame(currentFrame);
4:    PSNR = estimatePSNR(currentFrame,
        previousReconstructedFrame);
5:    if (PSNR > LB + α)
6:      frameInterval = frameInterval + 1;
7:    else if (LB - α ≤ PSNR < LB)
8:      frameInterval = frameInterval / 2;
9:    else if (PSNR < LB - α)
10:     frameInterval = fixedFrameInterval;
11:    if (frameInterval < fixedFrameInterval)
12:      frameInterval = fixedFrameInterval;
13:    else if (frameInterval > maxFrameInterval)
14:      frameInterval = maxFrameInterval;
15:    DVS(frameInterval);
16:    doEncoding(currentFrame);
17:    skipFrames(frameInterval - 1);
    }
    
```

그림 2 제안하는 알고리즘

제안하는 알고리즘을 그림 2에 제시하였다. 여기서는 프레임 속도 대신에 프레임 간격 (frame interval)을 조절하는 방식을 취했다. 매 프레임 입력을 읽은 후 (3줄), 프레임 간격을 조절할지 여부를 결정하기 위해서 현재 입력 프레임과 이전 재생 프레임의 PSNR을 구한다 (4 줄). 주어진 화질 제약은 만족하기 위해서 두 개의 인자 - 하한 (LB: lower bound)과 여지 (α: margin) - 를 사용해서 PSNR이 (LB, LB + α) 범위 안에 들어가도록 프레임 간격을 조절하는데, PSNR의 범위에 따라서 각각 아래와 같이 수행한다.

i) $PSNR > LB + \alpha$: 이전 프레임과의 화면 차이가 미미함을 의미한다. 그러므로 프레임 간격을 1 증가한다 (5-6줄). 프레임 간격을 천천히 증가한 것은 화질 저하가 눈에 띄지 않게 하기 위해서이다.

ii) $LB \leq PSNR \leq LB + \alpha$: PSNR이 (LB, LB + α) 범위 안에 들어갔음을 의미한다. 이는 우리가 원하는 것이므로 프레임 간격을 변화시키지 않고 현재 프레임 속도를 유지한다.

iii) $LB - \alpha \leq PSNR < LB$: 이전 프레임과의 화면 차이가 큼을 의미한다. 화질을 빨리 회복시키기 위해서 프레임 간격을 반으로 감소한다 (7-8줄).

iv) $PSNR < LB - \alpha$: 화질이 심각하게 떨어졌음을 의미한다. 이는 종종 화면 전환의 경우에 발생한다. 이 경우에는 처음 FFR 방식의 화질을 복구하기 위해서 프레임 간격을 초기값 (fixed frame interval)으로 설정한다 (9-10줄).

마지막 두 경우에 대해서는 몇몇 PSNR 값이 LB보다 아래로 떨어질 수 있다. 우리는 PSNR 값의 1% 미만이 LB 아래로 떨어졌을 때 해당 인코딩 기법이 화질을 만족했다고 정의한다. 최적의 (LB, α) 인자는 충분한 실험을 통해서 결정되는데 자세한 내용은 생략하겠다.

이후, 조정된 프레임 속도를 반영하여 DVS를 수행하고 (15줄) 해당 프레임을 인코딩한 후 (16줄), (프레임 간격-1) 개의 프레임을 건너뛰어 다음 프레임을 인코딩한다 (17 줄).

4. 실험

H.263 인코더로 TMN 테스트 코드 3.0 버전 [12]를 사용하였다. 5MHz에서 1.2V/2.4mW를 소모하고 80MHz에서 3.4V/381mW를 소모하는 ARM8 프로세서를 대상으로 컴파일하고 시뮬레이션 했다. 실제 수행 사이클은 시뮬레이터에서 얻었다. 동작 주파수는 최고 주파수를 프레임 간격으로 나누어서 결정했다. ARM8 프로세서의 전력-주파수 표 [13]로부터 해당 전력 소모를 얻고, 프레임별 에너지 소모 및 전체 에너지 소모를 계산하였다.

화면 변화의 실제적인 환경을 조성하기 위해서, 300 프레임 CIF (common intermediate format) 비디오 클립 - 'Coastguard', 'Hall monitor', 'Foreman', 'Container' - 을 4개 이어서 1200 프레임 비디오 입력을 만들었다. 화면의 움직임 정도의 차이 때문에 'Coastguard'와 'Foreman'은 낮은 PSNR 값을 가지고 'Hall monitor'와 'Container'는 높은 PSNR 값을 가진다. 최고 프레임 속도는 30 fps로, 최저 프레임 속도는 30/16 fps로, 그리

고 최대 프레임 간격은 16으로 설정하였다.

앞서 밝힌 바와 같이 입력 프레임의 1% 미만이 화질 제약보다 낮은 PSNR 값을 가졌을 때 해당 인코딩 결과는 화질 제약을 만족했다고 정의한다. 즉 이 실험에서는 1200 프레임 중 12 프레임이 화질 제약보다 낮은 PSNR 값을 가진 경우이다.

표 1. FFR과 VFR 방식의 성능 비교

인코딩 방식	프레임 속도 (fps)	화질 제약 (PSNR)	에너지 (J)	
			버퍼 없음	버퍼 사용
FFR (30 fps)	30	29.37	138.1	59
VFR (29, 2)	23.6	29.33	103.6	11.9
FFR (15 fps)	15	19.32	25.3	16.1
VFR (27, 2)	11.7	18.57	19.6	6.4
FFR (7.5 fps)	7.5	16.45	7.2	5.4
VFR (26, 2)	6.9	16.45	6.9	3.9
FFR (3.7 fps)	3.7	13.56	2.6	2.4
VFR (23, 2)	3.4	13.16	2.6	2.1

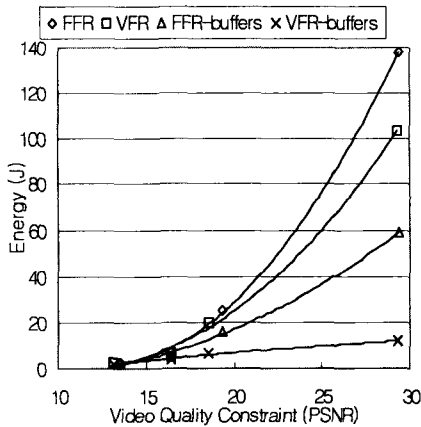


그림 3. 화질 제약에 따른 에너지

표 1은 여러 가지 프레임 속도에 대해서 FFR과 VFR 방식의 성능을 비교한 것이다. 우선 FFR 방식의 결과로부터 화질 제약을 얻어내고, 이와 비슷한 화질 제약을 보이는 VFR 방식을 찾아내었다. 최적의 (LB, a)는 각각 (29, 2), (27, 2), (26, 2), (23, 2)로 결정되었다. VFR 방식의 평균 프레임 속도가 FFR 방식에 비해서 20% 정도 감소되었다.

표 1은 또한 두 가지 DVS 기법 - 버퍼를 사용한 것과 그렇지 않은 것 - 에 대해서 제안하는 방법이 보다 적은 에너지를 소모함을 보여준다. 30 fps의 경우에 FFR에 비해 버퍼를 사용하지 않을 때는 34.5J (25%)을, 버퍼를 사용할 때는 47.1J (80%)을 절감한다. 다른 프레임 속도에서의 비교에서도 마찬가지로의 결과를 보여준다.

그림 3은 표 1의 결과를 이용하여 화질 제약에 따른 에너지 소모를 추세선으로 나타내었다. 결국, 제안하는 VFR 방식은 FFR 방식보다 같은 화질 제약에서 보다 에너지 효율적임을 보여준다.

5. 결 론

본 논문은 화질과 에너지의 상관관계로부터 에너지 최적화 문제를 정의하고, H.263 인코더의 화질을 실시간으로 모니터링해서 프레임 속도를 자동으로 조절하게 알고리즘을 고찰으로써 이 문제를 해결할 것을 제안한다.

참 고 문 헌

- [1] M. Pedram and J. M. Rabaey, *Power Aware Design Methodologies*, Kluwer Academic Publishers, June 2002.
- [2] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473-484, April 1992.
- [3] <http://www.itu.int/ITU-T/>
- [4] Y. Shin, K. Choi, and T. Sakurai, "Power Optimization of Real-Time Embedded Systems on Variable Speed Processors," *Proc. of Int'l Conference on Computer-Aided Design*, pp. 365-368, 2000.
- [5] S. Lee and T. Sakurai, "Run-Time Voltage Hopping for Low-Power Real-Time Systems," *Proc. of Design Automation Conference*, pp. 806-809, 2000.
- [6] I. Hong, D. Kirovski, G. Qu, M. Potkonjak and M. B. Srivastava, "Power Optimization of Variable Voltage Core-Based Systems," *Proc. Design Automation Conf.*, pp. 176-181, 1998.
- [7] I. Hong, M. Potkonjak, and M. B. Srivastava, "On-line Scheduling of Hard Real-Time Tasks on Variable Voltage Processor," *Proc. of Int'l Conference on Computer-Aided Design*, pp. 653-656, 1998.
- [8] T. Okuma, T. Ishihara and H. Yasuura, "Real-Time Task Scheduling for a Variable Voltage Processor," *Proc. Int'l Symp. on Systems Synthesis*, pp. 24-29, 1999.
- [9] Y. Shin and K. Choi, "Power Conscious Fixed Priority Scheduling for Hard Real-Time Systems," *Proc. Design Automation Conf.*, pp. 134-139, 1999.
- [10] D. Shin, J. Kim, and S. Lee, "Intra-Task Voltage Scheduling for Low-Energy Hard Real-Time Applications," *IEEE Design & Test of Computers*, vol. 18, no. 2, March-April 2001.
- [11] C. Im, H. Kim, and S. Ha, "Dynamic Voltage Scheduling Technique for Low-Power Multimedia Applications using Buffers," *Proc. of Int'l Symposium on Low Power Electronics and Design*, 2001.
- [12] <http://www.telenor.com/rd/>
- [13] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, "A Dynamic Voltage Scaled Microprocessor System", *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571-1580, November 2000.