

# 소프트웨어 기반 실시간 MPEG-4 비디오 부호화기

유 성민<sup>o</sup>, 채병조, 오승준, 정광수  
광운대학교 전자공학부

(smyou, bjchae, sjoh)@media.gwu.ac.kr and kchung@daisy.gwu.ac.kr

## A Software-Based Real-Time MPEG-4 Video Encoder

Sung-Min You<sup>o</sup>, Byung-Jo Chae, Seung-Jun Oh, and Kwangsue Chung  
School of Electronics Engineering, Kwangwoon University

### 요 약

본 논문에서는 소프트웨어 기반의 실시간 MPEG-4 비디오 부호화기를 제안하였다. MPEG-4에서 제공하는 MPEG-4 참조 소프트웨어 비디오 부호화기는 여러 가지 기능이 혼합되어 있는 복잡한 구조를 가지고 있기 때문에 실시간으로 부호화하는데 많은 어려움이 있다. 이를 PC에서 실시간으로 부호화할 수 있도록 목표를 심플 프로파일로 제한하여 이에 맞도록 구조를 수정하고, 고속 움직임 벡터 예측 알고리즘을 사용하여 움직임 예측시의 계산량을 줄였으며, 실시간으로 부호화할 시에 많은 부하를 주는 부분인 DCT, 움직임 예측 및 보상, 보간 부분을 MMX(MultiMedia Extention) 방법으로 구현하여 계산량과 부하를 줄임으로써 보다 향상된 성능을 제공한다. 기존 부호화기의 부호화 시의 복잡성을 감소시키고 윈도우 환경에서 간단한 조작만으로 부호화를 할 수 있도록 하였다.

### 1. 서론

광범위한 멀티미디어 응용의 기본이 되는 비디오 부호화는 그 표현을 위하여 대용량의 데이터가 필요하기 때문에 효율적인 비디오 압축 방법에 대한 연구가 활발하게 진행되어 왔다. 그리고 H.263, MPEG-1, MPEG-2 등과 같은 비디오 데이터 부호화 표준이 만들어졌다. 최근에는 컴퓨터와 TV, 전화 통신 사이의 경계가 무너지고, 콘텐츠 기반의 멀티미디어 통신에 대한 요구가 증가함에 따라 이러한 요구를 지원할 수 있는 새로운 표준인 MPEG-4가 만들어졌다.

MPEG-4는 콘텐츠 기반의 오디오/비주얼 객체의 통신과 접근, 처리를 목표로 한다. MPEG-4는 자연 영상뿐만 아니라 합성 객체의 부호화 방식, 초고능률 부호화 방식과 전송 오류에 강한 부호화 방식 등을 지향하기 때문에 실시간 통신, DVD, 콘텐츠 기반 저장과 검색 서비스, 디지털 TV 등의 광범위한 응용에서 사용될 수 있다 [1].

이러한 특징으로 인하여 MPEG-4는 이전의 표준들보다 더 복잡한 구조를 갖기 때문에 하드웨어로 구현하는 것은 특정 응용에만 국한될 수 있으므로 유연성과 이동성을 갖고, 새로운 도구를 첨가할 수 있도록 하는 소프트웨어로 구현하는 것이 바람직하다. 그러나 소프트웨어만으로 실시간 복부호화를 지원할 수 있으려면 대단히 높은 계산 능력이 요구된다.

본 논문에서는 이러한 단점을 보완하여 실시간으로

부호화를 할 수 있도록 구조를 단순화시키고, 고속 움직임 벡터 예측 알고리즘을 적용시키며, 인텔의 MMX 구조에 기반한 SIMD(Single Instruction Multiple Data) 함수를 사용하여 처리 속도를 향상시킨다.

본 논문의 2장에서는 MPEG-4 부호화기와 도입된 고속 움직임 벡터 알고리즘에 대해 간략히 설명하고, 3장에서는 MMX 기술에 대한 설명과 구현 과정을 보이고, 4장에서 구현된 부호화기의 실험 결과를 보이며, 5장에서 결론을 맺는다.

### 2. MPEG-4 비디오 부호화기와 고속 움직임 벡터 알고리즘

MPEG-4 비디오 부호화기의 구조가 MPEG-1과 MPEG-2에 비해 매우 복잡한 것은 [2] 여러 가지 다양한 기능들을 포함하기 때문인데, 사용되는 요소 기술을 정의한 프로파일과 그 프로파일에서 사용하는 영상의 크기 및 비트율로 정의되는 레벨에 따라 구조가 간단해질 수 있다. 본 논문에서는 이동통신 환경, 무선 환경에서 사용될 수 있는 저비트율의 데이터를 실시간으로 부호화하는데 초점을 맞추고 있으므로 SP, 즉 심플 프로파일에 해당하는 데이터만을 처리할 수 있는 부호화기를 만드는 데 목적을 둔다. 마이크로소프트(MS)사의 MPEG-4 버전 1 소프트웨어를 기반으로 하여 부호화기 구조를 단순화 시켰는데, MPEG-4가 가지고 있는 요소 기술 중에서 심플 프로파일에서는 필요하지 않은 영상

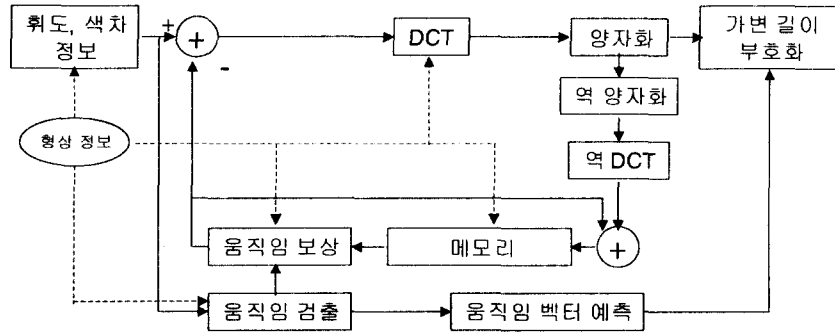


그림 1. MPEG-4 비디오 부호화기 구조

부호화 부분과 스프라이트 부호화 부분, 정지 텍스처 부호화 부분과 스케일러빌리티 부호화 부분 등을 제거하였다.

그림 1은 MPEG-4 비디오 부호화기 구조의 간단한 블록 다이어그램을 나타낸다. 그림에서 점선으로 표시된 형상 부호화 부분만 떼어내면 단순화된 부호화기의 구조가 된다.

또한 움직임 예측에서의 성능 향상을 위해 고속 움직임 벡터 예측 알고리즘을 도입하였다. 기존의 알고리즘은 FS(Full-Search) 방법을 사용하는데, 계산량이 너무 많으므로 이를 줄이기 위해 TSS(Three-Step-Search NMS(New fast Motion estimation search with Subsampling method) 방법을 적용시켰다[3]. 구현된 결과를 보면, 기존의 주관적 화질이나 PSNR을 어느 정도 유지하며 상당한 처리 속도 향상을 보이는 것을 알 수 있다.

3. MPEG-4 비디오 부호화기의 MMX 구현

인텔에서는 통신, 신호처리, 멀티미디어 응용 분야에 적합한 성능을 제공하기 위하여 기존의 인텔 구조를 확장시킨 MMX 구조를 개발하였다[4]. 본 논문에서는 부호화 속도 향상을 위해 움직임 예측에서의 SAD(Sum of Absolute Difference)값 계산, DCT 및 IDCT, 움직임 보상을 위한 보간(Interpolation) 모듈 등 세 부분을 MMX에서 제공하는 SIMD 명령어를 사용하여 부호화할 수 있도록 구조를 변경하였다.

MPEG-4에서의 SAD값 계산은 8-비트 데이터 성분을 갖는 16x16 또는 8x8 블록단위로 수행되는데, 각 데이터 성분의 크기가 8-비트이므로 8개의 데이터 성분이 MMX를 사용하여 동시에 수행될 수 있으므로, 처리 속도가 크게 향상된다. 16x16, 8x8 블록 단위로 SAD값을 구하는 두 가지 방법 모두를 MMX로 구현하였다.

DCT/IDCT 부분도 8x8 블록 단위로 수행되기 때문에 MMX로 구현하면 커다란 속도 향상을 얻을 수 있는 부

분이다. AAN DCT 알고리즘을 사용하고 인텔에서 제공하는 MMX DCT 방법을 참조하여 이 부분을 구현하였다.

보간은 반화소 정확도로 움직임 예측을 수행하기 위해 모든 복원된 I와 P 픽처에서 수행된다. 보간은 16-비트 데이터에 수행되기 때문에 4개의 데이터를 동시에 처리하는 것에 의해서 처리 속도의 향상을 얻을 수 있다. 이 부분 역시 MMX로 구현하였다.

4. 실험 결과

실험은 MPEG-4 버전 1 참조 소프트웨어 중에서 MS 사 비디오 부호화기의 구조를 수정하여 사용하였다. MMX를 지원하는 펜티엄 III CPU와 메모리 크기를 변경하며 실험하였다. 시험 시퀀스로는 MPEG-4 코덱 성능을 시험하기 위하여 가장 보편적으로 사용하는 Akiyo(qcif/cif, 250 프레임), Foreman(qcif/cif, 350 프레임), Suzie(qcif/cif, 프레임)를 사용하였다.

표 1. 각 영상의 방법별 프레임율 비교 (PIII 800MHz, 256MB RAM)

	Original	MMX_Only	TSS_Only	TSS_MMX	NMS_MMX
Akiyo (qcif)	21.4298	40.4596	36.9822	56.3444	60.5914
Suzie (qcif)	14.8588	32.5591	32.3555	52.0111	50.2681
Foreman (qcif)	12.9156	29.9222	30.9297	49.9286	49.1504
Akiyo (cif)	5.4176	10.1725	9.5354	14.289	15.6026
Suzie (cif)	3.4809	7.8918	8.0832	13.0708	12.3987
Foreman (cif)	2.816	6.9317	7.7016	12.5444	12.1439

표 1은 펜티엄 III 프로세서 800MHz CPU와 256M Ram을 갖는 PC에서 실험한 결과이다. 표에서 Original은 기존의 참조 소프트웨어를 나타내고, MMX\_Only는 MMX 방법만을 사용한 것, TSS\_Only는 TSS 방법을 사용한 것, TSS\_MMX는 위의 두 가지 방법을 다 사용한 것, NMS\_MMX는 NMS 방법과 MMX 방법(SAD 부분 제외)을 사용한 것이다. 여기서 비트율 제어는 사용하지

않았고, QP(Quantization Parameter) 값은 10으로 고정시켰다.

결과에서 알 수 있듯이 Original보다 MMX나 TSS 방법을 사용한 것이 fps(Frames/Sec)가 2배에서 3배까지 증가한 것을 볼 수 있고, 위의 두 가지 방법을 결합한 것과 NMS 방법을 사용한 것이 3배에서 4배 정도까지 증가함을 볼 수 있다.

표 2. 각 영상의 방법별 프레임율 비교 (PIII 500MHz, 128MB RAM)

	Original	MMX_Only	TSS_Only	TSS_MMX	NMS_MMX
Akiyo (qcif)	13.7468	25.6568	23.7756	36.0231	38.6458
Suzie (qcif)	9.4381	20.744	20.6897	33.2079	32.2789
Foreman (qcif)	8.181	18.9732	19.7897	32.1513	31.7432
Akiyo (cif)	3.441	6.4842	6.0723	9.1881	10.1071
Suzie (cif)	2.1988	4.9877	5.1279	8.3538	7.9504
Foreman (cif)	1.7796	4.3979	4.8744	7.9921	7.7875

표 2는 펜티엄 III 프로세서 500MHz CPU와 192MB 메모리를 갖는 PC에서 실험한 결과이다. 표 1과 비슷한 정도의 속도 향상이 있었다. 표 3은 MPEG-4 내의 TM 비트율 제어 방법을 사용하여 64 kbps로 부호화된 영상의 휘도 성분의 평균 PSNR을 비교한 것이다.

표 3. 각 영상의 평균 PSNR(Y) 비교

	Original	MMX_Only	TSS_Only	TSS_MMX	NMS_MMX
Akiyo (qcif)	39.0686	38.9416	39.1345	38.9199	39.055
Suzie (qcif)	34.4512	34.3047	33.9038	33.8811	34.1833
Foreman (qcif)	29.4159	29.4226	28.4745	28.4782	28.9084

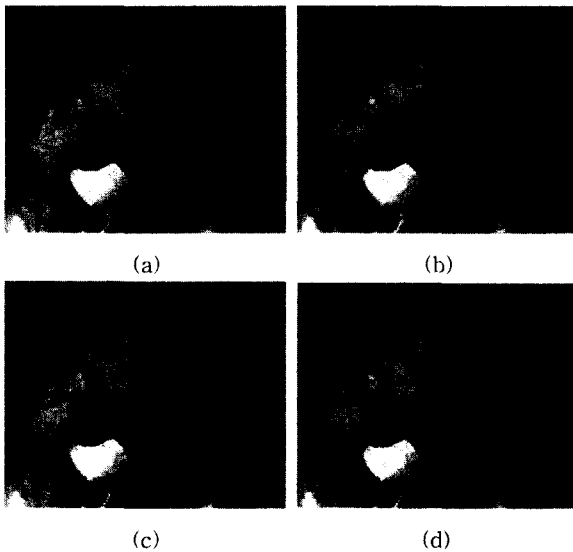


그림 2. 64 kbps Suzie(qcif) 영상. (a) 원 영상, (b) Original, (c) TSS\_MMX, (d) NMS\_MMX

움직임이 많은 Foreman 영상을 제외하고는 0.2~0.5 dB 차이로 PSNR의 차이가 그리 크지 않음을 알 수 있고, TSS\_MMX 방법보다 NMS\_MMX 방법이 PSNR이 좀 더 높음을 알 수 있다. 그림 2는 주관적 화질을 보여 준다. (b), (c), (d)의 화질의 차이가 거의 없음을 알 수 있다. 그림 3은 구현된 MPEG-4 비디오 부호화기이다.

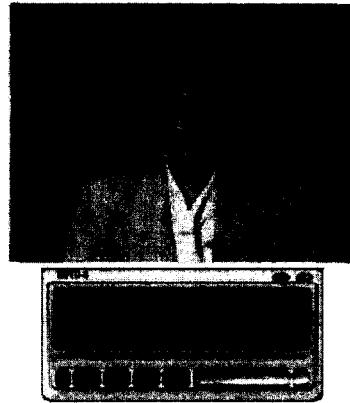


그림 3. MPEG-4 비디오 부호화기

### 5. 결론

본 논문에서는 소프트웨어 기반의 실시간 MPEG-4 비디오 부호화기를 제안하였다. 기존의 부호화기의 구조를 수정하여 특정 목표인 심플 프로파일로 구조를 단순화하고, 움직임 예측에서는 고속 움직임 벡터 예측 방법을 도입하였다. 그리고 MMX 방법을 사용하여 처리속도를 향상시켜서 큰 화질의 변화 없이 qcif 크기의 영상을 실시간으로 부호화하는 것이 가능함을 보였다. 향후 이동통신 서비스나 무선 환경에서의 응용에서 사용할 수 있도록 하기 위한 연구가 더 진행되어야 할 것이다.

### 6. 참고 문헌

- [1] ISO/IEC, FDIS 14496-2, "Generic Coding of Audio-Visual Objects - Part 2 : Visual", Oct, 1998
- [2] Thomas Sikora, "The MPEG-4 Video Standard Verification Model", *IEEE Trans. Circuits Syst. Vi Technol.*, vol. 7, no. 1, Feb. 1997
- [3] R.Li, B.Zeng, and M. L. Liou, "A New Three-step Search Algorithm for Block Motion Estimation" *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, 5, pp. 438-442, Oct. 1994.
- [4] Intel's MMX Technology Programmers Reference Manual, 1999