

H.263 인코딩 속도향상연구

유환중, 강의선, 강석찬, 김영환, 김진구, 임영환
 송실대학교 컴퓨터학과

H.263 Encoding Speed up Research

Hwan-Jong Yu, Eui-Sun Kang, Seock-Chan Kang, Young-Hwan Kim,
 Jin-Goo Kim, Young-Hwan Lim
 Dept. of Computer Science Soongsil University

요약

PSTN(Public Switch Telephone Network)에서 동영상을 전송하기 위해 H.263 이라는 표준이 발표되었다. 저속의 전송률을 가지는 PSTN 을 이용해서 영상회의나 영상전화 등을 구현하기 위해서는 기존의 코딩방식으로는 데이터를 전송하는데 문제점이 많았다. 이를 위해서 개발된 것이 H.263 이다. H.263 은 H.261 에 기반을 두고 있으며 H.261 에 비해서 동일화질을 제공하는데 반정도의 데이터 양으로도 가능하게 해준다.

영상 압축 Encoder 는 일반적으로 Decoder 에 비하여 영상을 처리하는데 많은 시간이 소요된다. 그러나 VOD 등과 같이 실시간으로 압축할 필요가 없는 경우에 대해서는 인코더가 많은 시간을 소비하더라도 큰 문제가 없는 반면에, 영상 회의나 영상 전화 등은 실시간 영상 Encoding, Decoding 을 수행해야 한다. 그러기 위해서 고가의 하드웨어를 사용하게 된다. 이와 같은 이유에서 본 연구에서는 H.263 을 소프트웨어만으로 Encoding 속도향상을 꾀하고자 하는 것이 이 논문의 목표이다.

1. INTRODUCTION

H.261 이나 H.263 을 Encoding 하기 위해서는 Hardware 에 의존해왔고, 현재는 Software 만으로 Encoding 을 구현하는데 속도저하가 심각한 원인이 되고 있다.

MOPS 라는 단위로 H.261 을 분석한 Compression 에 대한 비교는 다음과 같다.(6)

TABLE I Compression MOPS

COMPRESSION	MOPS
RGB to YCbCr conversion	27
Motion estimation (exhaustive search for p=8)	608
Inter-/Intraframe coding	40
Loop filtering	55
Pixel prediction	18
2-D DCT	60
Quantization and zig-zag scanning	44
Entropy coding	17
Frame reconstruction	99
TOTAL	968

우선 우리는 각 Module 별로 시간을 분석하였다.

아래의 Table 은 TMN H.263 Encoding Source 를 분석한 표이다.

TABLE II TMN Profile

Count
207734.3 72.6 207734.3 72.6 25974000 _SAD_Macroblock (block_functions)

16344.39	5.7	16344.39	5.7	32959 _FindHalfPel (mot_est)
10580.99	3.7	10580.99	3.7	334 _ReadImage (picture_functions)
7940.584	2.8	7940.584	2.8	198396 _Dct (dct)
6450.121	2.3	6450.121	2.3	198396 _Quant_blk (quant)
6309.244	2.2	269132.2	94.1	333 _CodeOneOrTwo (p_picture)
4489.742	1.6	4489.742	1.6	65934 _LoadArea (block_functions)
2885.834	1	2885.834	1	666 _InitializeMV (mot_est)
2761.585	1	2761.585	1	333 _InterpolateImage (picture_functions)
2048.022	0.7	2048.022	0.7	12600 _idctref (dct)
1959.471	0.7	1959.471	0.7	33066 _Clip (block_functions)
1871.956	0.7	1871.956	0.7	334 _WriteImage (picture_functions)
1807.861	0.6	1807.861	0.6	334 _ComputeSNR (snr)

여기서 대부분을 시간을 Motion Estimation 하는데 소비하였다.

Motion Estimation technique 의 Complexity 는 세가지로 정의될 수 있다.(6)

- 1 Search Algorithm
- 2 Cost Function
- 3 Search range

Search 는 움직임 측정의 가장 기본이 되는 기술이다. 현재 매크로 블록이 이전의 어느 위치에 있는지를 찾기 위해서는 이전 프레임의 Search Area 에서 적당한 algorithm 를 수행시켜야 한다. 보통 search algorithm 으로 blocking matching algorithm 를 사용한다.

또한 Search Area 를 결정해 주는 것이 search range parameter 이다. Search Range 는 매크로 블록 가장 자리 사

이의 거리를 나타낸다. 일반적으로 H.263 에 Search Range 의 값은 6 이다.

이전 프레임의 search area 에서 매크로 블록 만큼의 데이터를 찾아서 매크로 블록을 비교하여서 유사성이 얼마나 있는지를 측정해야 하는데 이 경우 Cost function 을 사용하여 알아볼 수 있다.

본 논문에서 접근하고자 하는 방식은 두 가지 방법으로 할 것이다.

- 효율적 Search Algorithm 의 적용
- Cost Function 의 최적화 (using MMX™ Instructions)

2. H.263

이제까지 화상정보는 디지털로 저장을 하여도 그 정보량이 매우 비대하여서 일반통신선로나 시스템에서 사용하는 데는 애로사항이 매우 많았다.

그러나 화상정보의 압축 기술이 급속하게 발전하면서 화상정보도 일반통신선로나 시스템에서 사용할 수 있게 되었다.

여러 종류의 압축 기술들에 대한 표준으로는 정지화상용인 JPEG, 음성용으로 MPEG 오디오, 동화상용으로 H.261, MPEG(1/2/4), MPEG 시스템 기술, 여러 가지의 미디어를 동시에 취급할 때 여러 개의 작은 멀티미디어로 표현된 정보 객체(Multimedia Hypermedia Object)로 나누어 기술하는 MHEG, 그리고 end-to-end 까지의 멀티미디어 서비스를 위한 시스템 전체의 표준을 제정하기 위하여 결성된 DAVIC(Digital Audio Visual Council) 등이 있다.

ITU-T 는 PSTN 을 이용한 오디오/비디오 서비스에서 동영상정보를 압축하기 위한 표준인 H.263 을 규정하였다. ITU-T 가 규정하는 표준은 DCT intra frame, motion estimation 그리고 DCT transformed inter frames 을 포함하는 hybrid coding 이다. ITU-T 가 규정하고 있는 표준은 현재의 PSTN 상의 video phone 을 포함한 application 과 PSTN 의 전송속도와 유사한 미래의 이동 네트워크에 응용하기에 적합하다. (1)

3. Mmx

인텔의 MMX(Multi Media eXtension) 테크놀로지는 이름처럼 멀티미디어 어플리케이션 특히 그래픽, 음성 등 아날로그 데이터를 빠르게 처리하기 위한 목적으로 만들어진 명령어이다.

MMX 아키텍처는 강력한 57 개의 명령어가 추가되었고, 하나의 명령어로 다중 데이터를 같은 함수 내에서 처리할 수 있는 SIMD(Single Instruction Multiple Data) 그리고, 32KB 로 늘어난 캐시 크기 등이 주요 특징이다.

MMX 테크놀로지에 추가된 명령어를 MMX 명령어라고 부르며, MMX 명령의 특징은 곱셈(multiply) 명령어로 덧셈과 곱셈을 한꺼번에 처리한다. MMX 의 명령어 이용하는 레지스터는 8 개가 있지만 레지스터가 추가된 것이

아니고 부동소수점용(FPU) 레지스터를 조작하는 명령이 추가된 것이다.

모든 응용 프로그램이 MMX 명령을 이용하는 것은 아니다.

MMX 명령은 전혀 범용성이 없다. 디지털 처리용 명령인 것이다. 따라서 MMX 의 이용이 기대되는 것은 디지털 처리를 하는 응용 프로그램에 한해서다.

멀티미디어 응용 프로그램중 주로 화상이나 음성의 리얼타임 처리가 필요한 것이 MMX 지원 효과도 크다.

또 이미지 프로세싱 처리를 하는 화상처리 응용 프로그램도 MMX 효과가 기대된다. 이들 프로그램에서는 처리하는데 기다리는 시간이 줄어들 지도 모른다. 액션 게임에서는 3D 처리에 이용할 수 있다.

가장 큰 효과를 볼 수 있는 것이 이미지 분야이다.

이는 컴퓨터 그래픽이 아니라 포토샵에서 이미지를 필터링시키는 것과 같은 효과를 말한다.

사운드 분야도 마찬가지다. 이는 인텔 자체의 벤치마크 테스트 결과에서도 나타나는 사항이다. 아마 응용 프로그램이 독단적으로 MMX 기능을 이용한다고 볼 때 눈에 띄게 빨라지는 프로그램은 포토샵과 같은 이미지 처리 프로그램과 버추얼 사운드 캔버스와 같은 소프트웨어 미디 프로그램 정도일 것이다.

4. RELATED WORK

H.263 인코더 실행 시 가장 문제점이 되는 부분은 Search algorithm 에 있다. 움직임 벡터를 찾기 위한 Algorithm 에는 Exhaustive search algorithm 처럼 Search area 의 모든 부분을 Search 하는 Full search algorithm, Fast search algorithm 이 존재한다.

위의 문제점에서 본 것처럼 기존의 H.263 인코더 실행 시 가장 문제점이 되는 부분은 Search algorithm 에 있다. 움직임 벡터를 찾기 위한 Algorithm 에는 Exhaustive search algorithm 처럼 Search area 의 모든 부분을 Search 하는 Full search algorithm 뿐만 아니라 Fast search algorithm 이 존재한다. 그러므로 인코더의 수행 속도를 저하시키는 Full search algorithm 을 대체하여 Fast search algorithm 를 사용하여 인코더의 수행 속도를 향상 시킬 수가 있을 것이다.

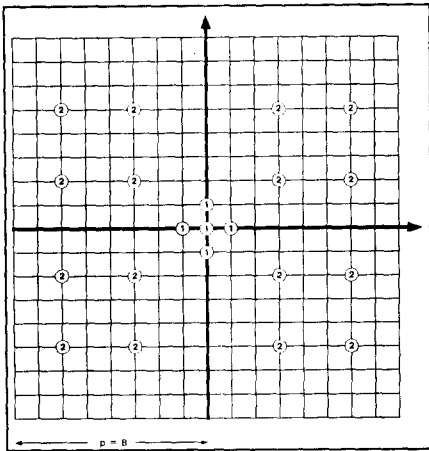
본 연구에서는 인코더의 수행 속도를 저하시키는 Full search algorithm 과 Fast search algorithm 을 대체하여 DCUPS(Densely-Centered Uniform-P Search) algorithm 를 사용하여 인코더의 수행 속도를 향상 시킬 수가 있을 것이다.

DCUPS(Densely-Centered Uniform-P Search) algorithm 의 구현은 다음과 같다. (6)

DCUPS algorithm 의 균등한 지역은 조사 지역에 샘플링 점들의 숫자를 단순히 균등하게 분배한다. 이 샘플링은 같은 공간이고 사실상 어떤 배치에 대한 계산이 아닌 것

을 요구한다. 그러므로 알고리즘의 오버헤드는 작다. DCUPS algorithm 는 Fig. 1.에 묘사되어 있다.

조사 알고리즘은 모든 단계 조건하에서 잘 실행이 예상된다. 그 빠른 조사들은 그림의 모시리나 모퉁이에 조사 지역을 맞게 조사 벡터를 클립 하여야 한다. DCUPS 의 하나의 바람직한 특징은 클립핑 상태에 순서적으로 분해를 하는 것이다. 4 개의 알고리즘은 DCUPS 처리를 경과 하는 것을 예상된다.



DCUPS(Densely-Centered Uniform-P Search) algorithm

- 조사지역의 왼쪽 픽셀 숫자
- 조사지역의 오른쪽 픽셀 숫자
- pup 는 조사지역의 위쪽 스캔라인의 숫자
- pdown 는 조사지역의 아래쪽 스캔라인의 숫자

5. SUMMARY

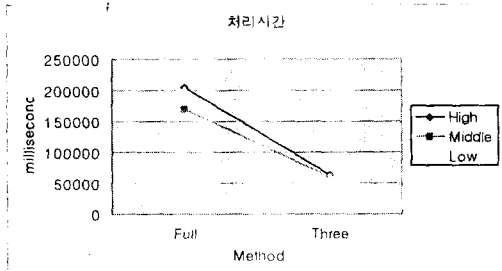


Fig. 1. Three Step Algorithm Processing Time Analysis

Three Step Algorithm 실행 결과를 분석한결과 약 2.5 배의 향상을 가져왔다. 따라서 DCUPS Algorithm 을 적용시 더 많은 향상이 이루어 질것으로 기대된다.

아울러 MMX 로 작성된 인코딩 속도를 바탕으로 한 전체적인 속도향상은 62.43%이다. 이것은 전체 루틴의 평균 속도향상(60.68%)에 아주 가깝다. 그러나, 이 루틴들

의 전체 실행 시간의 공유는 스칼라 버전에서는 63.21% 이고 MMX 버전에서는 40.37% 이다. (5)

TABLE III Min. execution time, average execution time, execution frequency and average speedup for each routine

Routine	Version	Min.			
Time	Average				
Time	Times				
Executed	Average				
Speedup					
DCT+Q	MMX	1035	1226	63954	58.6
DCT+Q	Scalar	2498	2961	64062	
IQ+IDCT	MMX	1071	1198	63954	57.83
IQ+IDCT	Scalar	2661	2841	64062	
CalcSADFF16	MMX	275	396	152655	65.71
CalcSADFF16	Scalar	1075	1155	149759	
CalcSADHF16	MMX	663	770	21306	66.9
CalcSADHF16	Scalar	2067	2326	21335	
CalcSADFH16	MMX	566	757	21296	70.1
CalcSADFH16	Scalar	2076	2532	21330	
CalcSADHH16	MMX	1078	1239	42568	63.97
CalcSADHH16	Scalar	3148	3439	42616	
CalcSADFF8	MMX	107	131	92076	62.78
CalcSADFF8	Scalar	331	352	92652	
CalcSADHF8	MMX	215	239	90184	61.26
CalcSADHF8	Scalar	571	617	90288	
CalcSADFH8	MMX	189	231	93492	66.13
CalcSADFH8	Scalar	580	682	93064	
CalcSADHH8	MMX	317	347	106740	61.4
CalcSADHH8	Scalar	835	899	106800	
CalcIntraDecision	MMX	318	515	10791	63.73
CalcIntraDecision	Scalar	1130	1420	10791	
PreCalcDiffMB	MMX	1478	2808	10659	5.39
PreCalcDiffMB	Scalar	1816	2968	10677	

6. REFERENCE

- [1] ITU-T DRAFT H.263, Video Coding for Low Bitrate Communication, 2 May, 1996.
- [2] ITU-T Study Group 15, "Video Codec Test Model Near-Term, Version 7 (TMN7)," 25 February 1997.
- [3] Cote, G., Gallant, M. and Kossentini, F. "Efficient Motion Vector Estimation and Coding for H.263-Based Very Low Bit Rate Video Compression." Online document available at URL <http://www.cce.ubc.ca/spmg/>, 1997.
- [4] CCITT H.263-Image Compression - Online document available at URL <http://www.stud.ec.eitb.ch/~rmp/rm/h263.html>
- [5] Performance Analysis of Intel MMX Technology for an H.263 Video Encoder - Ville Lappalainen
- [6] Borko Furht, Joshua Greenberg and Raymond Westwater, Kluwer Academic Publishers, "Motion Estimation Algorithms for Video Compression", 1997