

논문 2004-41SP-1-2

실시간 객체기반 비디오 서비스를 위한 MPEG-4 Encoder 분석

(Analysis of MPEG-4 Encoder for Object-based Video)

김민훈*, 이선영*, 문석주*, 장의선**

(Min Hoon Kim, Euee Seon Jang, Sun young Lee, and Seok ju Moon)

요약

본 논문에서는 현재 MPEG-4 비디오의 인코더를 분석하고 인코더의 복잡함을 줄일 수 있는 효율적인 압축기술을 제안하였다. 지금까지의 객체기반 비디오에서는 유/무선 저 전송률 부호화환경에 맞추기 위하여 형상부호화(Shape coding)를 제외한 인코더 최적화가 주를 이루었다. 최근에 본 연구팀이 수행한 형상부호화의 복잡도를 줄이기 위한 실험을 통하여 객체기반 부호화에서 형상부호화가 차지하는 계산비중이 상대적으로 높다는 것을 밝혀냈다. 본 논문에서는 최근까지의 영상부호화 분야와 형상부호화 분야의 최적화 기술이 성공적으로 결합되어 최적의 객체기반 부호화 기술을 유도할 수 있는지 실험하였다. 먼저 영상부호화 부분에서는 기존의 MEMC(Motion Estimation Motion Compensation)에서 사용된 검색 방법인 나선형 검색 대신에 MVFAST(Motion Vector Field Adaptive Search Technique) 기술을 채택하였고, 형상부호화 부분에서는 복잡함을 줄이기 위해 기존의 MEMC에서 사용된 나선형 검색을 생략하고 IVOPF(Intelligent VOP Formation) 대신 TRB(Tightest Rectangular Boundary)을 채택하여 최적화에 적용해 보았다. 실험결과, 객체기반 부호화를 위한 제안된 최적화 방식이 기존의 reference software 보다 57.3% 향상되었음을 알 수 있었다. 아울러, 본 논문에서 제안된 기술은 형상부호화 부분에만 적용된 최적화 기술보다 50.8% 향상된 실험결과를 보여주었다.

Abstract

In this paper, we have analyzed the current MPEG-4 video encoding tools and proposed efficient coding techniques that reduce the complexity of the encoder. Until recently, encoder optimization without shape coding has been a major concern in video for wire/wireless low bit rate coding services. Recently, we found out that the computational complexity of MPEG-4 shape coding plays a very important role in the object-based coding through experiments. We have made an experiment whether we could get optimized object-based coding method through successfully combining latest optimized texture coding techniques with our proposed optimized shape coding techniques. In texture coding, we applied the MVFAST method for motion estimation. We chose not to use IVOPF(Intelligent VOP Formation) but to use TRB(Tightest Rectangular Boundary) for positioning VOP and, finally, to eliminate the spiral search of shape motion estimation to reduce the complexity in shape coding. As a result of experiment, our proposed scheme achieved improved time complexity over the existing reference software by 57.3% and over the optimized method on which only shape coding was applied by 48.7%, respectively.

Keywords: MPEG-4, OVC, MEMC, 형상부호화, MVFAST

I. 서론

* 학생회원, 한양대학교 정보통신공학과
(Department of Information and Communication Engineering,
Hanyang University)

** 정회원, 한양대학교 정보통신학부
(The College of Information & Communications, Hanyang
University)

※ 이 논문은 2003년 한양대학교 교내연구비 지원으로 연구되었음
접수일자 : 2003년9월5일, 수정완료일 : 2003년12월26일

1993년에 시작된 MPEG-4의 첫 번째 표준이 2000년에 국제 표준으로 완성되었다. 사용자는 MPEG-4의 중요한 특징인 객체기반 부호화를 사용하여 MPEG-4 비트스트림 안에서 오디오 비주얼한 객체를 편집하고 재사용 할 수 있다. MPEG-4의 비주얼 파트는 전통적

인 비디오 부호화뿐만 아니라 얼굴이나 몸 애니메이션과 2D/3D mesh coding같은 그래픽 객체의 부호화를 포함한다.

초기 상업용 MPEG-4 서비스에서는 객체기반 비디오 부호화(Object-based video coding)가 아닌 프레임기반 비디오 부호화(Frame-based video coding)를 사용하였다. 현재 MPEG-4 비디오 부호화는 Microsoft Windows Media Technologies와 Realnetwork Real같은 경쟁자들과 MPEG-4 비디오 부호화 보다 두 배 좋은 성능을 갖는 MPEG-4(part 10) Advanced Video coding(AVC)에 의해서 위협을 받고 있다^{[1][2][3][4]}.

하지만 MPEG-4 비디오 부호화기술의 경쟁력은 아직 남아 있다. 바로 객체기반 부호화 이다. 객체를 가지고 있는 하나의 화면 안에서 사용자들은 화면 안의 내용을 편집하거나 재사용 할 수 있다. 비록 내용기반의 편집이 전혀 새로운 것은 아니지만 비디오 객체의 재사용이 가까운 미래에 양방향 멀티미디어 서비스에 사용될 수 있음을 시사해 주고 있다.

실시간 객체기반 서비스(OVC)의 성공을 위해서는 복잡도의 감소가 무엇보다 큰 문제이다. 지금까지 MPEG-4 객체기반 비디오 부호화의 복잡함은 소프트웨어와 하드웨어 분야에서 연구 되어져 왔다^{[5][6]}. 대부분의 이전 연구들은 인코더와 디코더의 기본구조를 유지하면서 복잡함을 줄이는데 있었다. 실시간 객체기반 비디오 부호화 서비스의 성공을 이루기 위해서는 부호화와 복호화 모두의 최적화가 필요하다. 하지만 복호화 과정이 부호화 과정보다 상당히 빠르기 때문에 부호화 과정에서 더 많은 최적화 작업이 이루어져야 한다. 예를 들어, 디지털 방송, 모바일용 hand-set과 인터넷 PC등에서는 속도가 빠른 디코더뿐만 아니라 인코더가 필요하다.

인코더의 부호화 과정 중 가장 큰 부분을 차지하는 것은 MEMC (motion estimation motion compensation) 과정 이다. 이것은 객체기반 비디오 부호화의 복잡도 감소를 위해서는 MEMC 과정의 최적화가 필요함을 의미한다. 현재 reference software의 MEMC 과정에서는 움직임을 찾는 방법으로 형상과 영상부분 모두 나선형 검색을 사용한다. 그러나, 나선형 검색은 화면 전체를 검색하여 움직임 벡터를 찾는 방법이기 때문에 복잡도가 증가한다.

본 논문에서는 객체기반 부호화의 복잡도를 줄이기 위한 몇 가지 방법을 제안한다. 형상부분에서 나선형

검색을 제거하고 영상부분에서는 화면 전체를 검색하는 나선형 검색대신 영역별 검색방법인 MVFAST를 적용하여 화질의 변화 없이 복잡도 줄일 수 있었다.

본 논문은 다음과 같이 구성된다. IV장에서는 제안된 효율적인 방법들과 III장과 IV장에서는 실험 결과 및 결론에 대해 기술한다.

II. Proposed efficient encoding conditions

그림 1은 MPEG-4 OVC 부호화 과정의 구조를 나타낸다. 인코더는 형상부호화(shape coding), 움직임추정(motion estimation),움직임보상(motion compensation)과 영상부호화(texture coding)로 구성된다. 표 1은 자세한 부호화 과정의 여러 가지 도구이다. 도구 중에서 계산의 복잡도가 큰 것들은 'X'로 표시 하였다. MEMC는 형상(Shape)과 영상(texture)부분에서 한번씩 처리되고, 부호화 과정에서 많은 시간을 차지한다. 본 논문에서는 MEMC 과정의 최적화를 위한 방법과 부호화 과정의 최적화를 위해 IVOP와 이진 형상부호화를 분석하여 최적화를 위한 효율적인 방법을 제안 하였다.

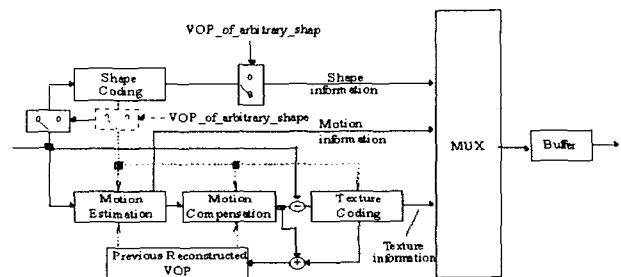


그림 1. MPEG-4 OVC 인코더 구조

Fig. 1. MPEG-4 OVC encoder structure

2.1 Tightest rectangle 모드

객체 기반의 형상부호화에서 각 객체는 IVOPF에 의해서 VOP(video object plane) formation을 갖게 된다. 만약 비디오 객체가 사각형이 아닌 다양한 형상이면 MB는 전체 픽셀 값이 255인 all opaque와 일부 픽셀 값이 255인 partially opaque, 전체 픽셀 값이 0인 all transparent로 나누어진다.

그림 2는 VOP을 찾는 방법을 보여준다. VOP 위치를 정하는 방법은 TR(Tightest rectangle)의 좌측상단을 첫 번째 MB로 선택하는 것이다. IVOPF는 TR의

좌측상단을 control MB의 우측하단으로 control MB를 정한다. Control block에서 짝수 열의 픽셀들이 VOPF와 MB의 수를 분석하기 위해 사용된다. 사용되는 픽셀의 수는 64(8*8)이다.

짝수열의 픽셀을 사용하는 이유는 비디오 포맷이 4:2:0 포맷으로 색채(U or V)값이 4개의 휘도(Y) 값에 포함되기 때문이다. Control block내 64개의 픽셀의 첫 번째 픽셀은 첫 번째 MB의 위치 후보 픽셀이 된다. 그 후에 전체 MB개수, all transparent MB개수, non-transparent(all opaque와 partially transparent) MB개수를 계산할 수 있다. 모든 경우(64가지)를 계산한 후, non-transparent MB개수가 가장 작은 픽셀을 선택한다. IVOPF의 단점은 반복되는 검색이고 최악의 경우에 64가지를 계산하여 non-transparent MB수를 비교한다.

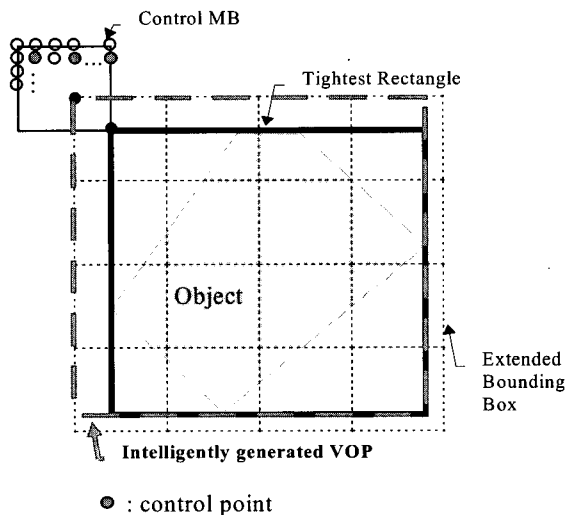


그림 2. Intelligent VOP 구성
Fig. 2. Intelligent VOP formation

본 논문에서는 IVOPF의 압축효율을 가지면서 복잡도를 줄이기 위한 5개의 다른 vop formation에 관한 실험을 하였다. 표 2는 5가지 모드의 실행시간에 대한 시뮬레이션 결과이다. '8*8 Bnd(8*8)'모드는 DCT coding이 8*8 block인 점을 고려하여 블록을 16*16에서 8*8로 줄였고 boundary texture coding을 피하기 위해서 8*8 boundary 블록이 가장 작은 것을 선택하게 하였다. 'Worst'모드는 non-transparent MB가 가장 많은 경우(IVOPF의 반대)를 선택하게 하였다. 'TMB'모드는 MEMC와 형상부호화의 복잡도를 최소화하기 위해서 nonopaque 블록을 최소화 하였다. 'AM'모드는 모든 블록의 숫자가 최소가 되게 하였고 'TR'모드는

IVOPF를 사용하지 않고 가장 작은 bounding box를 선택하게 하였다.

2.2 형상 부호화의 MEMC

움직임 추정과 움직임 보상(MEMC)은 부호화 과정에서 가장 많은 시간을 차지하고 있는 부분이다. [9][10][11][12]

표 1. MPEG-4 OVC 부호화 도구
Table. 1. MPEG-4 OVC encoding tools

Category	Tool	Functionality	Cost
General	VOP formation	Compression	X
Shape	Mode decision	Compression	
	MEMC	Compression	X
	subsampling	Rate control	X
	Binary shape coding	Compression	X
	Grayscale shape coding	Compression	
Motion	padding	Compression	X
	MEMC	Compression	X
Texture	Padding	Compression	X
	DCT	Compression	X
	Quantization	Compression	
	AC/DC prediction	Compression	
	VLC encoding	Compression	
Other	Error Resilience	Error Resilience	
	Scalable coding	Scalability	
	Sprite coding	Compression	

실험결과 IVOPF를 사용하지 않은 TR모드의 복잡도가 가장 작음을 알 수 있다.

표 2. 새로운 VOP formation 모드
Table. 2. New VOP formation modes

Selection MODE	Semantics
IVOPF	Find the minimum number of nontransparent MBs
8*8 Bnd(8*8)	Find the minimum number of 8*8 boundary blocks
Worst	Find the maximum number of nontransparent MBs
Trans/Bnd Min.(TBM)	Find the minimum number of transparent and boundary MBs
All Min(AM)	Find the minimum number of MBs(or try to find the minimal size of VOP)
Tightest Rectangle(TR)	Do not use IVOPF at all.

MEMC는 motion vector predictor(MVP)가 결정된 후에 움직임 추정과 움직임 보상의 과정을 따른다. MVP를 결정하기 위해서 형상과 영상 MV(motion vector)를 사용한다.

그림 3은 현재 BAB의 MV 계산을 위한 전 MV를 나타낸다. MVP는 첫 번째 MV1이 결정되고 MV2가 유효하지 않다면 0으로 세팅된다. MVP가 결정 되면 MVP의 16*16 픽셀의 움직임 보상 에러가 +/- 16 픽셀 범위 안에서 수평 수직방향으로 계산된다. 현재 reference software는 영상의 움직임 추정에서 사용한 나선형 검색을 형상에서도 사용한다.

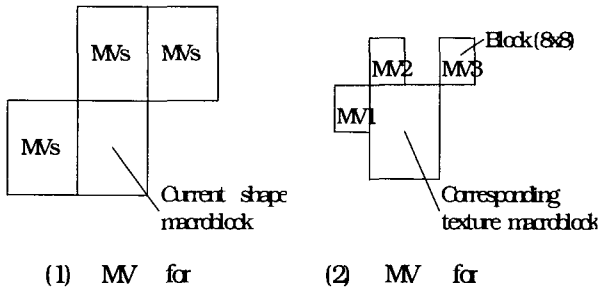


그림 3. MVPs를 위한 후보 MVs
Fig. 3. Candidature MVs for MVPs

형상부호화를 위한 움직임 추정은 영상부호화와 유사한 복잡도를 가진다. 예상된 MV의 중앙의 화소로부터 다음에 오는 MB후보는 나선형 검색을 통해 얻어진다. 이 방법은 중앙 점으로부터 가장 가까운 픽셀을 쉽게 정할 수 있다. 그러나 +/- 16픽셀을 거쳐 반복처리를 하는 나선형 검색은 인코더를 복잡하게 만든다. 형상을 위한 MV가 결정되면, 움직임 보상을 위해서 inter CAE coding을 위한 참조 BAB를 준비한다.

MPEG-4 OVC에서 MEMC는 형상과 영상에서 사용된다. 형상에서의 블록은 이진 값을 비교하기 때문에 형상부호화를 위한 MEMC는 영상부호화를 위한 MEMC보다 복잡도가 감소한다. 형상을 위한 MEMC의 최적화를 위해서 본 논문에서는 어떠한 블록비교도 하지 않도록 하였다. 나선형 검색을 생략함으로써 MVP가 현재 MV가 되도록 하였고 형상부호화의 복잡도를 감소시켰다.

2.3 영상 부호화의 MEMC

현재 사용되고 있는 reference software에서는 영상부호화의 MEMC를 위해서 나선형 검색이 사용되고 있다. 나선형 검색 방법은 전체화면을 검색하여 움직임 벡터를

찾는 방법으로, 복잡도가 증가하는 단점이 있다. 그래서 우리는 나선형 검색 방법대신 지금까지 제안된 빠른 MEMC방법인 MVFAST를 적용하였다. 현재 제안된 빠른 MEMC방법에는 MVFAST, PMVFAST(Predictive Motion Vector Field Adaptive Search Technology)등이 존재한다^{[7][8]}. MVFAST와 PMVFAST는 나선형검색 방법과 달리 전체화면 대신 영역별로 검색하여 움직임 벡터를 찾는 방법이다. PMVFAST는 MVFAST에 비해 빠른 검색방법이지만 메모리의 낭비와 알고리즘이 복잡한 단점이 있고 MPEG-4에서 사용된 비디오 영상 이외에는 검증되지 않았다. 표 3은 MVFAST와 PMVFAST의 비교이다.

표 3 MVFAST와 PMVFAST의 비교
Table. 3 Comparison of MVFAST and PMVFAST

Category	MVFAST	PMVFAST
임계값	임계값이 없거나 하나의 임계값, 모든 환경에서 검증	반드시 여러개의 임계값을 사용, MPEG-4 이외의 환경에서 검증되지 않음
알고리즘의 복잡도	작은 복잡도	높은 복잡도
메모리의 크기	필요하지 않음	메모리의 사용이 필요

MVFAST는 블록 정합 시 유사성을 판단할 때 SAD(sum of absolute difference)를 사용한다. 이때 현재 블록과 후보 블록간의 최소 SAD를 갖는 후보 블록이 움직임 벡터로 선택된다. 만약 SAD가 Threshold T보다 작으면 그 블록의 움직임 벡터는 (0,0)이 되고 검색 대상에서 제외된다. 그림 4는 MB에서 Local motion vector field가 n=3인 ROS(region of support)를 갖는 것을 나타낸다.

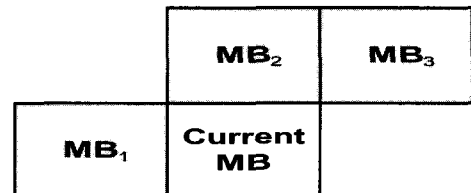


그림 4. 참조 지역
Fig. 4. Region of support(ROS)

주변블록의 길이는 $V_i=(x_i,y_i)$ 는 $|v_i|=|x_i|+|y_i|$ 로 정의되고 L은 모든 V_i 에 대한 최대 값 $MAX\{|v_i|\}$ 이다. Motion

Activity를 설정하는데 조건은 식 (1)로 표현되어진다.

$$\begin{aligned} \text{Motion Activity} &= \text{Low, if } L \leq L1; \\ &= \text{Medium, if } L1 < L \leq L2; \\ &= \text{High, if } L > L2; \end{aligned} \quad (1)$$

그림 5는 주변 블록의 Li를 나타낸다. (a)는 L1=2이고 (b)는 L2=1이다.

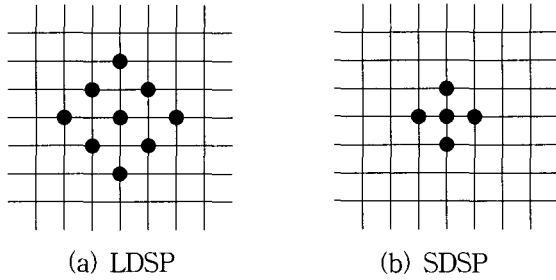


그림 5. 검색 패턴
Fig. 5. Search pattern

만약 Motion Activity가 Low 또는 Medium일때 검색 시작점은 origin이 되고 High 일때는 최소 SAD를 갖는 곳이 시작점이 된다. 또 Motion Activity가 Low 또는 High이면 그림 4의 (b)처럼 Small Diamond Search Pattern(SDSP)를 사용하고 Medium이면 (a)처럼 Large Diamond Search Pattern(LDSP)를 사용한다. SDSP와 LDSP의 검색은 다음과 같은 과정을 따른다.

1. Small Diamond Search (SDS)

- 1) 1 단계 : Small diamond search pattern의 검색 시작점은 중심에 있다. 주위의 모든 checking point의 SAD를 검사한 후에 시작점이 가장 작은 SAD를 가지면 중심이 움직임벡터를 나타낸다. 아니면 2단계로 진행한다.
- 2) 2 단계 : 중심의 SDSP의 시작점이 최소 SAD지점으로 움직인다. 주위의 모든 checking point의 SAD를 검사한 후에 시작점이 가장 작은 SAD를 가지면 중심이 움직임벡터를 나타낸다. 아니면 이 과정을 반복한다.

2. Large Diamond Search(LDS)

- 1) Large diamond search pattern의 검색 시작점은 중심에 있다. 주위의 모든 checking point의 SAD를 검사한 후에 시작점이 가장 작은 SAD를 가지면 중심이 움직임벡터를 나타낸다. 아니면 2단계로 진행한다.

- 2) 2 단계 : 중심의 LDSP의 시작점이 최소 SAD지점으로 움직인다. 주위의 모든 checking point의 SAD를 검사한 후에 시작점이 가장 작은 SAD를 가지면 3단계로 진행한다. 아니면 이 과정을 반복한다.

- 3) 3 단계 : LDSP를 SDSP로 바꾸고 시작점이 가장 작은 SAD를 가지면 움직임벡터를 나타낸다.

표 4는MVFAST의 검색 모드를 나타낸다.

표 4. MVFAST의 검색 모드
Table. 4. The search modes for MVFAST

Motion Activity	Search Center	Search Strategy
Low	Origin	SDS
Medium	Origin	LDS
High	The position of the vector in set V that yields minimum SAD	SDS

III. Simulation results

3.1 Test Conditions

본 논문에서 사용된 인코더와 디코더 소프트웨어는 MPEG-4 reference software 2001년 버전(a Microsoft version)을 사용하였다^[9]. 실험을 위해 512 MB의 Pentium 4 1.8GHz PC를 사용하였다. 실행시간 분석은 소프트웨어의 시간 모듈을 이용하였고 3번 실험의 평균 값을 사용하였다. 표 4는 사용된 비디오 영상이다. 부호화 시의 설정은 IPPP, 10Hz, no rate control 과 I-VOP와 P-VOP를 위한 QP값을 10으로 하였다.

3.2 Experiment results

본 논문에서는 객체기반 비디오 부호화의 복잡도를 줄이는 방법을 제안 하였다. 제안한 부호화 방법 PEM(Proposed Encoding Method)은 형상 부호화 부분에서 MEMC 과정중 하나인 나선형 검색의 제거하고 IVOPF 대신 TR(Tightest Rectangle)을 사용하였고 영상 부호화 부분에서는 나선형검색 대신 MVFAST 검색 방법을 적용하였다. 그림 6과 그림 7은 PEM과 reference software의 화질 변화를 비교한 것이다. 이 결과는 본 논문에서 제안한 방법이 화질에 영향을 주지 않는다는 것을 나타낸다.

표 5는 PEM의 실행시간을 나타낸다. 본 논문에서 제안한 PEM의 경우 나선형 검색을 하지 않고 TR를 사용

한 영상부호화 최적화 기술인 SEM(Shape Encoding Method)에 비하여 50.8%의 복잡도 감소를 가져왔고, refrence software의 속도에 비해 57.3% 향상 되었다. 그리고 가장 두드러진 효과를 나타낸 saxophone 영상은 67%의 복잡도가 감소 되었다.

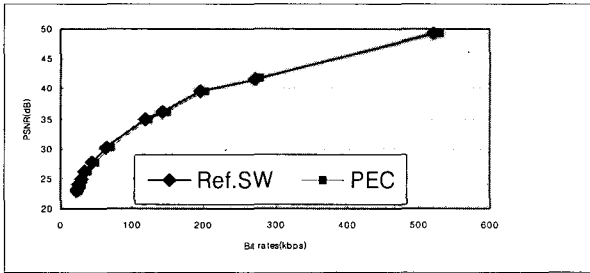


그림 6. Stefan 영상의 PSNR 비교
Fig. 6. PSNR comparison of stefan sequence

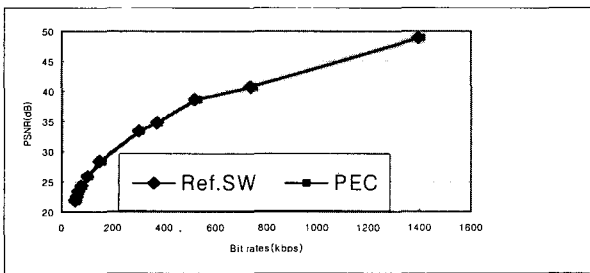


그림 7. Fishandlogo 영상의 PSNR 비교
Fig. 7. PSNR comparison of fishandlogo sequence

표 5. PEC의 실행시간 결과
Table. 5. Run time results of PEC

Test sequence	Ref.SW	SEC	PEC
akiyo	105.3	88.7	53.65
coastguard	169.3	161.6	70.18
container	122.5	119	59.25
dancer1	73.4	62.1	34.04
foreman	118.4	105.9	48.07
news	141.2	134.6	69.37
saxophone	200.6	167.4	66.12
singer	62.0	53.7	34.79
stefan	65.5	55.9	32.05
bream	120.8	101.2	45.23
child	103.6	81	41.42
cyclamen	201.8	179.6	82.05
fishandlogo	127.2	94.4	47.33
weather	76.4	64.5	37.97
Average	120.6	104.6	51.5

IV. Conclusion

객체기반 비디오 부호화 과정에 있어서 IVOPF와 형상부분의 MEMC 과정인 나선형검색의 사용은 화질의 개선과 압축량의 감소 성능에 비해 매우 높은 복잡도를 초래하였다. 본 논문에서는 영상부호화 부분의 IVOPF와 나선형 검색을 생략하고 영상부호화부분의 MEMC과정인 나선형 검색대신 MVFAST를 적용함으로써 객체기반 비디오 부호화 과정의 성능을 개선하였다.

향후 DMB 등 지능형 방송이 등장할 가까운 장래에 등장할 것으로 예상되고, 개인 휴대통신 등에서 복원 뿐만 아닌 실시간 압축이 예상되는 바, 객체기반 비디오 부호화의 성능개선은 계속적으로 이루어져야 할 분야로 판단된다. 특히, 모바일 응용을 위한 실시간 압축 기술의 필요성이 매우 높고 따라서 향후에 주로 연구되어야 할 분야 중 하나로 생각된다.

참고 문헌

- [1] Y.Ozer, "Will MPEG-4 Fly?", *PCMgazine* April 3, 2001.
- [2] Microsoft Windows Media Technologies home page, <http://www.microsoft.com/korea/windows/windowsmedia/software/player/v7.asp>.
- [3] Realnetworks homepage, <http://www.realnetworks.com/>.
- [4] MPEG Homepage, <http://mpeg.telecomitalia.com/>.
- [5] W.Zheng, I.Ahmad, and M.L.Liou, "Benchmark the software based MPEG-4 Video codec", in the *8th International IEEE Conference on Electronics, Circuits and Systems*, V.1, pp.289-292, 2001.
- [6] H.-C.Chang, Y.-C.Wang, M.-Y.Hsu, and L.-G. Chen, "Efficient algorithms and architectures for MPEG-4 object-based video coding," in *IEEE workshop on Signal Processing Systems*, pp.13-22, 2000.
- [7] Optimised Reference Software for Coding of Audio-Visual object version 4.0, ISO/IEC/JTC1/SC29/WG11, N4554, Pattaya, December. 2001.
- [8] Yu-Wen. Huang, Shyh-Yih. Ma, Chun-Fu. Shen, and Liang-Gee. Chen "Predictive Line Search: An

- Efficient Motion Estimation Algorithm for MPEG-4 Encoding Systems on Multimedia Processors," *IEEE Trans. on Circuits and Systems for Video Technology*, V.13, pp.111-117, January. 2003.
- [9] MPEG-4 Video Verification Model 15.0, ISO / IEC / JTC1 / SC29 / WG11, N3093, Maui, DEC. 1999.
- [10] J. Ostermann, E.S.Jang, J.-S. Shin, and T. Chen, "Coding of arbitrarily shaped video objects in MPEG-4," in *IEEE International Conference on Image Processing 1997*, Santa Barbara, CA, 1997.
- [11] N.Brady, "MPEG-4 standardized methods for the compression of arbitrarily shaped video objects," *IEEE Trans. on Circuits and Systems for Video Technology*, V.9, No. 8, pp. 1170-1189, Dec.1999.
- [12] S.-H. Son, E.S.Jang, S-H.Lee, D.-S.Cho, J.-s.Shin, and Y.-S.Seo, "Scan interleaving based scalable binary shape coding," *Signal Processing: Image Communication* 15(2000) 619-629.

저 자 소 개



김민훈(학생회원)

2002년 2월 금오공과대학교 산업공학과 졸업,

2002년 3월 ~ 현재 : 한양대학교 정보통신공학과 석사과정 <주관심 분야 : MPEG-4, Image processing>



이선영(학생회원)

1996년 2월 숭실대학교 인공지능학과 졸업, 2003년 3월 ~ 현재 : 한양대학교 정보통신공학과 석사과정 <주관심 분야 : MPEG-4>



문석주(학생회원)

2002년 2월 전남대학교 통계학과 졸업, 2002년 3월 ~ 현재 : 한양대학교 정보통신공학과 석사과정 <주관심 분야 : MPEG-4, Image processing>



장의선(정회원)

1991년 2월 전북대학교 전자계산기공학과(공학사), 1994년 6월 State University of New York at Buffalo 전기 및 컴퓨터공학과(공학석사), 1996년 2월 State University of New York at Buffalo 전기 및 컴퓨터 공학과(공학박사), 1996년 10월 ~ 2000년 2월 : MPEG-4 SNHC Group 의장, 1995년 U.S Army Research Lab, Adelphi, MD, USA Research Associate, 1996년 1월 ~ 2002년 2월 삼성종합기술원 전문연구원(책임연구원) 멀티미디어랩, 2002년 3월 ~ 현재 : 한양대학교 정보통신학부 조교수 <주관심 분야 : 멀티미디어 압축, 컴퓨터 그래픽스>