

# 임베디드 환경에 적합한 H.264|AVC 비디오 코덱에 관한 연구

\*유영일, \*김용태, \*\*이승준, \*강동욱, \*김기두

\*국민대학교 대학원 전자공학과

\*\*엠큐브웍스(주)

[young@kookmin.ac.kr](mailto:young@kookmin.ac.kr)

## H.264|AVC video codec for embedded environments

\*Youngil Yoo, \*Yongtae Kim, \*\*Seungjun Lee, \*Dongwook Kang, \*Kidoo Kim

\*Electrical engineering, graduated school, kookmin university

\*\*McubeWorks corp.

### 요약

H.264|AVC (baseline profile)은 MPEG-4 (simple profile)에 비하여 약 2배 정도 향상된 압축 성능을 나타내는 것으로 알려져, DMB 방송, DVD 등에 사용될 예정이며, 다양한 멀티미디어 서비스에 사용될 차세대 비디오 압축 표준으로 인정 받고 있다. 하지만, 부호기의 복잡도는 MPEG-4 simple profile에 비하여 16배 정도 높아져서, 실시간을 요구하는 서비스, 또는 임베디드 환경을 갖는 서비스에 사용하는데 심각한 제약을 갖는다. 따라서, 기존의 표준 코덱들을 대체하여 다양한 통신 환경에서 향상된 화질의 비디오 서비스를 제공하기 위해서는 코덱의 복잡도 개선이 매우 중요하다. 본 논문은 H.264|AVC 비디오 코덱을 임베디드 환경을 갖는 통신 서비스에 적용하기 위하여 코덱의 복잡도를 개선하기 위한 방법을 제시한다.

### I. 서론

본 논문은 H.264|AVC 비디오 코덱을 임베디드 환경을 갖는 통신 서비스에 적용하기 위하여 코덱의 복잡도를 개선하기 위한 방법을 제시한다. H.264|AVC 비디오 코덱은 위성 DMB, 지상파 DMB, 그리고 차세대 DVD 표준 코덱으로 사용 되는 것이 확정 되었다. 하지만, H.264|AVC의 특성상 임베디드 환경과 같이 메모리 및 계산 용량 등의 자원이 부족한 시스템에서의 사용은 매우 어렵다. 예를 들면, MMS 와 같은 서비스를 제공하기 위해서는 PDA나 혹은 카메라가 내장된 핸드폰에 H.264|AVC 비디오 코덱이 내장 되어야 한다. 본 논문에서는 표 1과 같은 서비스 품질 및 환경에서 H.264|AVC baseline profile 부호기의 사용을 목표로 하고 있다.

표 1. 서비스 품질 및 부호화 환경

Frame format	QCIF
Frame rate	7.5 fps
Data rate	32~64 kbps
Processor	ARM920T 255MHz(60 MIPS)
Memory	32MB

논문의 구성은 다음과 같다. 2장에서는 H.264|AVC baseline profile의 부호화 요소들을 임베디드 환경에 맞도록 설정하고, 그러한 설정 하에서 표준 고속 움직임 추정 [4] [5] [6] [7] 및 모드 결정의 복잡도를 알아본다. 3장에서는 비용 효율적인 고속 움직임 추정 및 모드 결정 방법을 제시한다. 4장에서는 모의 실험 결과를 보이고, 5장에서 결론을 맺는다.

### II. 표준 고속 움직임 추정 알고리즘의 복잡도

#### 1. H.264|AVC baseline profile 부호기 설정

고려하는 H.264|AVC는 baseline profile을 따르며, 움직임 추정 및 모드 결정 알고리즘은 JM8.2[8]의 표준 고속 알고리즘의 절차를 따른다. 하지만, baseline profile의 몇몇 기법들은 부호화 효율의 증가에 비해 복잡도가 높다. 따라서 비용 효율이 낮은 기법들을 제거하고 부호기를 단순화하는 것이 필요할 것이다. 다중 참조 프레임 을 사용할 경우(5개를 사용할 경우), 얻을 수 있는 압축률은 3% 미만인데 비하여, 복잡도는 단일 참조 프레임 을 사용할 때의 약 5배의 메모리와 계산량이 필요하다. 따라서, 다중 참조 프레임의 사용은 비용 효율적이 비디오 코덱을 개발하는데 있어서는 고려 대상이 되지 못 한다. 또

한 하다마드 변환의 경우 압축률에는 영향을 주지 않으며, 단지 PSNR의 향상에 조금 기여할 뿐이다. 따라서, 하다마드 변환도 부호화 설정에서 제외한다. 그리고 RD 최적화 기법을 사용할 경우, 5%의 성능 향상이 있으나, 모드 결정 시, 각 모드에 대하여 DCT 변환/역변환과 엔트로피 부호화 과정의 수행이 필요하므로 복잡도가 매우 높다. 따라서, 복잡도를 고려하여, RD최적화 기법 또한 사용하지 않았다. 표 2에 고려하는 H.264|AVC의 부호기 설정을 나타내었다.

표 2. 부호기 설정

Profile	Baseline profile	
Possible coding mode	Inter	All (7 block shapes)
	Intra	All(I4MB:9directions, I16MB 4directions)
RD-optimization	Off	
Hadamard transform	Off	
# of ref.	1	
Search range	8	
MV resolution	1/4	

## 2. 표준 고속 알고리즘의 계산적 복잡도

움직임 추정 알고리즘의 계산량은 SAD연산의 횟수로 간단히 측정할 수 있다. 공정한 비교를 위해, 4x4 블록 크기의 SAD 연산의 횟수는 16x16 블록 크기의 연산 횟수의 1/16로 개수된다. 사용한 타겟 보드는 ARM920T 266MHz 머신으로, 움직임 추정 알고리즘의 요구 계산량은 정수 화소 움직임 추정에는 매크로 블록당 30 SAD, 부화소 움직임 추정에는 15 SAD 이하의 연산 횟수를 가져야 한다. 몇 개의 실험을 통해 표준 고속 움직임 추정 알고리즘의 SAD 연산 횟수를 조사해 본 결과, 정수 화소 움직임 벡터를 찾는데 매크로 블록 당 평균 143번의 SAD 연산, 부화소 움직임 벡터를 찾는데 45번의 SAD 연산이 수행되는 것을 확인 하였다. 영상 시퀀스 별 SAD 연산 횟수는 표 4에 나타내었다. 결국, 타겟 보드의 계산적 제한 조건은 복잡한 H.264|AVC의 표준 움직임 추정 및 모드 결정 기법을 그대로 수용할 수 없게 만든다. 따라서, 표준 고속 알고리즘을 단순화 시키는 것이 필요하다.

## III. 비용 효율적인 고속 움직임 추정 알고리즘

H.264|AVC baseline profile 부호기의 부호화 이득을 가져오는 부호화 기법 중 가변 블록 크기 움직임 추정 기법이 가장 많은 부호화 이득을 가져온다. H.264|AVC baseline profile 부호기의 인터 모드는 16x16(mode 1), 16x8(mode 2), 8x16(mode 3), 8x8(mode 4), 8x4(mode 5), 4x8(mode 6), 4x4(mode 7) 블록 크기로 총 7가지 이다. 복잡도는 단순 계산으로 7배 증가하여야 하지만, 각 모드간 움직임 벡터는 높은 상관성을 가지고 있으므로, 이것을 이용할 경우, 실제 움직임 추정 복잡도는 다른 부호화 도구에 비하여 크게 증가하지 않는다. 따라서, 이러한 모드간 움직임 벡터의 상관성을 최대한 이용하기 위한 탐색 패턴 및 모드 결정 초기 종료 조건 등을 정의 하는 것이 필요하다. 제한하는 움직임 추정 및 모드 결정 알고리즘의 모드 별 탐색 패턴을 그림 1과 그림 2에 나타내었다. 부화소 단위 움직임 추정의 탐색 패턴을 그림 3에 나타내었다. 그림 4는 하나의 매크로 블록의 모드를 결정하기 위한 전체 움직임 추정 및 모드 결정 알고리즘의 흐름도 이다.

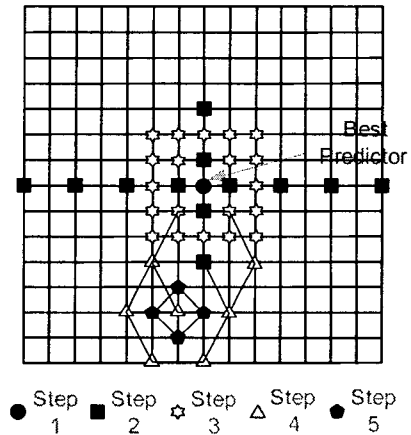


그림 1. 16x16, 8x8 블록에 대한 움직임 추정 탐색 패턴

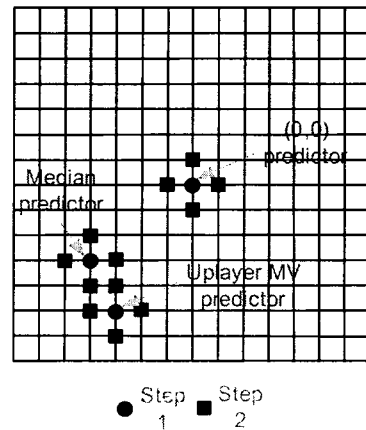


그림 2. 16x8, 8x16, 8x4, 4x8, 4x4 블록에 대한 움직임 추정 탐색 패턴

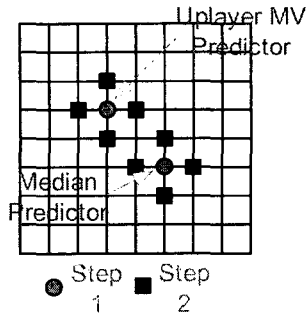


그림 3. 부화소 움직임 추정에 대한 탐색 패턴

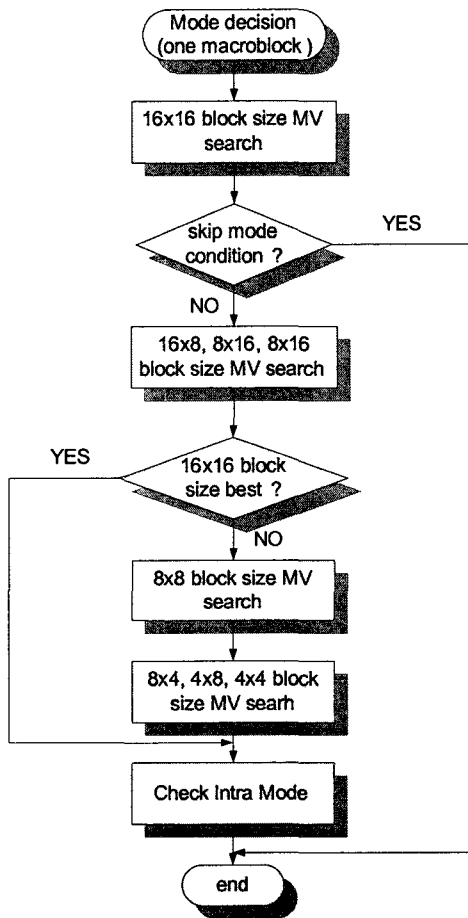


그림 4. 모드 결정 알고리즘 흐름도

16x16 및 8x8 블록 크기에 대한 움직임 추정의 경우, 표준 고속 움직임 추정 알고리즘의 탐색 패턴과 Big-Hexagon 탐색 패턴을 생략하는 것을 제외하고는 모두 동일하다. 반면, 16x8, 8x16, 8x4, 4x8, 4x4 블록 크기에 대해서는 중간값 예측기, 상위 모드의 움직임 벡터, (0,0) 벡터와 각 예측 벡터의 상, 하, 좌, 우 화소만을 탐색하여 모드의 움직임 벡터를 정한다. 이러한 전략은 부화소 단위 움직임 추정 전략에도 그대로 사용된다. 16x16과 8x8 블록에 대한 탐색과정 중 조기 종료 조건은 표준 고속 움직임 추정에서 사용하는 조기 종료 조건을 그대로 사용하

였다. 모드 결정 알고리즘은 16x16 블록 크기(mode 1)에 대한 움직임 추정 후, skip-mode 조건을 확인하여 skip-mode 조건이 만족될 경우, 모든 모드 결정 과정을 건너뛰는 방법을 사용하였고, 16x16, 16x8, 8x16 블록 크기를 평가해 본 후, 16x16 블록 크기가 최소의 cost 값을 가질 경우에 8x8 이하 블록 크기에 대한 평가를 생략하는 내용을 포함하였다. 이는 위의 조건이 만족될 경우, 하위 모드로의 진행은 단지 움직임 벡터의 표현에 필요한 비트량의 증가만을 가져오고 SAD의 감소는 미미할 것을 예상할 수 있기 때문이다. 즉, 하위 모드로의 진행은 비용 함수 값의 단조 증가 형태로 모델링 된다는 가정을 바탕으로 하고 있다.

#### IV. 모의 실험 결과

모의 실험에 사용된 비디오 시퀀스는 표 3과 같다. 부화소의 부화화 설정은 표 2와 같다. 표준 고속 알고리즘의 RD 성능 및 SAD 연산량을 표 4에 정리하였다. 제안하는 고속 알고리즘의 성능을 표 5에 정리하였다.

표 3. 실험에 사용된 비디오 시퀀스 및 비디오 포맷

Sequence name	Frame format	Frame rate	QP
Forman	QCIF	7.5 fps	30
Irene			28
Students			25
Mother & daughter			26

표 4. 표준 고속 알고리즘의 RD 성능 및 SAD 연산 횟수

Seq.	PSNR (dB)	Bit Rate (kbps)	# of SAD op.	
			Int.	Frac.
Foreman	33.89	53.12	195.81	49.38
Irene	37.13	57.29	110.75	45.06
St.	37.79	55.49	126.56	42.56
M & D	37.32	52.45	139.94	43.68
Average	36.53	54.59	143.27	45.17

표 5. 제안하는 고속 알고리즘의 RD 성능 및 SAD 연산 횟수

Seq.	PSNR (dB)	Bit Rate (kbps)	# of SAD op.	
			Int.	Frac.

Foreman	33.64	56.6	29.63	14.88
Irene	36.95	58.62	17.19	11.68
Students	37.6	59.45	18.5	11.31
M & D	37.13	55.26	19.69	12.25
Average	36.33	57.48	21.25	12.53

표 6. 표준 고속 알고리즘과 제안하는 알고리즘의 성능 비교

Sequence	PSNR Dec. (dB)	Bit Rate Inc. (%)	# of SAD op. Saving (%)	
			Int.	Frac.
foreman	0.25	6.55	84.9	69.9
Irene	0.18	2.33	84.5	74.1
students	0.19	7.14	85.4	73.4
M & D	0.19	5.36	85.9	72.0
Average	0.2	5.34	85.2	72.3

표 4에서 볼 수 있듯이 RD 성능은 표 1에 제시한 서비스 화질 요구 사항을 만족하지만, SAD 연산량은 요구 계산량을 4배 이상 넘어선다. 반면, 표 5, 6에서 볼 수 있듯이 제안하는 알고리즘은 평균적으로 5% 정도의 비트를 증가를 보이지만, RD 성능은 서비스 요구 조건을 만족한다. 또한 알고리즘의 요구 복잡도를 만족하는 SAD 연산 횟수를 보인다.

## V. 결론

모의 실험 결과 하위 모드에 대한 움직임 벡터가 중간 값 예측기, 상위 모드 움직임 벡터, (0,0) 벡터로 예측이 가능함을 보였다. 이러한 예측기와 약간의 보정 만을 통하여 부호기의 RD 성능에 평균적으로 5% 정도의 열화를 가져 오지만, 약 80% 정도의 SAD 연산 횟수를 줄일 수 있어 본 논문에서 고려하고 있는 임베디드 환경에서 H.264|AVC baseline profile 부호기를 사용할 수 있음을 보였다.

## 참조논문

[1] Minhua Zhou, "Evaluation and Simplification of H.26L Baseline Coding Tools", JVT-B030.doc, Joint Video Team(JVT) of ISO/IEC MPEG &

ITU-T VCEG, 2<sup>th</sup> meeting, Geneva, CH, Jan. 29  
 □ Feb. 1, 2002

[2] Iain E.G. Richardson, "H.264 and MPEG-4 VIDEO COMPRESSION", John Wiley & Sons Ltd, 2003

[3] Peter Kuhn, "Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation", Kluwer Academic Publishers, 1999

[4] Zhibo Chen, Peng Zhou, Yun He, "Fast Integer Pel and Fractional Pel Motion estimation for JVT", JVT-F017r1.doc, Joint Video Team(JVT) of ISO/IEC MPEG & ITU-T VCEG, 6<sup>th</sup> meeting, Awaji, JP, 5-13 December. 2002

[5] Zhibo Chen, Peng Zhou, Yun He, "Fast Motion estimation for JVT", JVT-G016.doc, Joint Video Team(JVT) of ISO/IEC MPEG & ITU-T VCEG, 7<sup>th</sup> meeting, Pattaya II, Thailand, 7-14 March, 2003

[6] Zhibo Chen, JianFeng Xu, Yun He, "Simplifications on fast motion estimation", JVT-H026.doc, Joint Video Team(JVT) of ISO/IEC MPEG & ITU-T VCEG, 8<sup>th</sup> meeting, Geneva, Switzerland, 23-27 May, 2003

[7] Byeungwoo Jeon, Jeyun Lee, "Fast mode decision for H.264", JVT-J033.doc, Joint Video Team(JVT) of ISO/IEC MPEG & ITU-T VCEG, 10<sup>th</sup> meeting, Waikoloa, USA, 8-12 Decemver, 2003

[8] JVT Reference Software version 8.2, <http://bs.hhi.de/~suehring/tml/download>