레이어 단위 멀티 프로세싱을 이용한 스케일러블 비디오 인코더 S/W 개발

*이주영, 김제우, 김용환, 최병호
*전자부품연구원
*jyyi@keti.re.kr

Development of Scalable Video Encoder S/W using layer based multi-processing

*Yi, Joo-Young, Kim Jewoo, Kim Yong-Hwan, Choi Byeong Ho
*Korea Electronics Technology Institute

요약

스케일러블 비디오 코딩은 레이어간 예측 기능을 이용하여 시뮬케스트 코딩에 비해 코딩 효율을 높인다. 하지만 스케일러블 비디오 코딩의 레이어간 예측으로는, 인트라 픽셀(inta pixel). 모션 정보(motion vector information), 레지듀얼(residual)등의 예측이 수행되는데, 이는 많은 계산 시간을 소요하게 되며, 시뮬케스트에 비해 코딩 시간이 증가하게 된다. 특히 인코더의 경우, 가장 최적의 모드를 선택하기 위하여, 기존 H.264 AVC에서 사용하는 예측을 수행한 뒤, 부가적으로 스케일러블 비디오 코딩의 레이어간 예측을 수행하기 때문에, 하나의 영상이 많은 레이어를 포함할수록 인코딩에 의한 계산 부하가 매우 증가하게 된다. 본 논문에서는 이를 해결하기 위해, 멀티 코어 이용하여 레이어별로 병렬처리가 가능하도록 하는 스케일러블 비디오 인코더의 구조를 제안한다. 이로써 하나의 영상이 포함하는 레이어의 수가 증가함에 따라 발생하는 인코딩 계산 부하를 줄이도록 하였다. 그리하여 본 논문에서 제안하는 구조를 적용하였을 때, 2개의 공간영역으로의 스케일러빌리티를 가지는 영상들에 대해서는 평균 24.8%의 속도가 향상되었고, 1개의 공간영역과 1개의 화질 영역으로의 스케일러빌리티를 가지는 영상들에 대해서는 평균 82%의 속도 향상을 보였다.

1. 서론

최근 다양한 네트워크를 통하여, 다양한 단말기를 대상으로 하는 비디오 커뮤니케이션이 점점 더 관심을 받고 있다 [1]-[3]. 이러한 산업계의 요구를 받아 들여, 하나의 비디오 스트림에 spatial, temporal, quality영역에서 여러 resolution을 지원하는 Scalable Video Coding (SVC)이 H.264/AVC의 amendment 3으로써 2007년 말에 표준화 되었다. SVC에서는 기존 simulcast 코딩에 비해 코딩 효율을 높이기 위해 레이어간 예측 기능을 포함시켰다. 하지만 SVC의 레이어간 예측을 수행하기 위해서는 많은 계산 시간을 소요하게 된다. 특히 SVC 인코더의 경우, 매크로 블록을 코딩 할 때, 제일 적합한 코딩 모드를 찾아내기 위해 기존의 H.264 AVC에서 사용하는 예측 뿐만 아니라, SVC에서 사용하는 레이어간 예측도 수행하여야 하기 때문에, 계산 시간이 매우 증가하게 된다. 또한 하나의 영상이 포함하는 레이어의 개수가 증가함에 따라, 인코더의 부하가 증가하게 되며, 이는 실시간 인코딩을 요하는 시스템에서 방해의 요인이 된다.

본 논문에서는 SVC 인코더의 속도를 개선하는 병렬처리 구조를 제안한다. 이는 레이어 별로 각각 다른 코어에서 인코딩을 수행하도록 하여, 인코더의 속도를 증가시켰으며, 레이어가 증가함에 따른 인코더 연산부하를 줄이고자 하였다.

본 논문에서 제안하는 멀티 프로세싱 구조를 적용하기 위해서는 먼저, 레이어 인코딩에 필요한 기능들이 모두 매크로블록 단위로 수행되어야 하는데 이를 2장에서 설명하고, 3장에서는 본 논문이 제안하는 레이어 단위의 멀티 프로세싱의 구조를 설명한다. 그리고 4장에서는 시뮬레이션 결과를, 5장에서는 결론을 기술한다.

2. 매크로블록 단위 인코딩

본 논문에서 제안하는 멀티 프로세싱 구조를 적용하기 위해서는 먼저 인코딩을 수행에 필요한 기능들이 모두 매크로블록 단위로 처리 되어야 한다. 그림 1은 하나의 매크로블록에 대한 인코딩순서를 나타낸다.

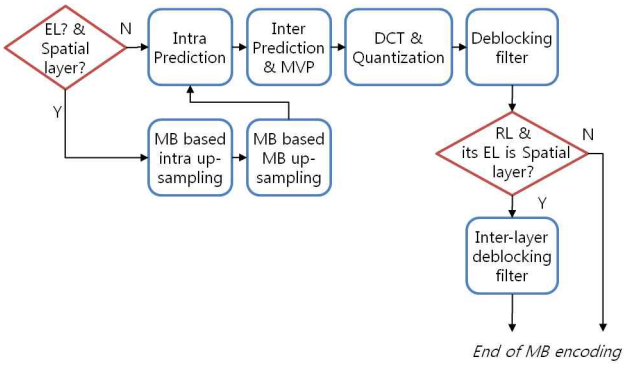

그림 1. 하나의 매크로블록에 대한 인코딩 순서

그림 1을 보면, 상위 레이어(EL)의 경우, 레이어 종류가 spatial 레이어 일 때, 레이어간 예측을 위한 업샘플링을 매크로블록 단위로 수행함을 알 수 있다. 먼저 매크로블록 단위 인트라 업샘플링의 경우, 그림 2와 같이 참조 레이어(RL)로부터 업샘플링에 사용될 픽셀의 위치

를 계산 한 뒤, 그 것을 바탕으로 업샘플링을 수행하도록 한다[6]. 만약 EL이 인트라 업샘플링에 참조하고자 하는 RL의 블록이 인터 픽셀을 포함할 경우, 해당 인터 픽셀을 주변 인트라 픽셀을 이용하여 패딩을 수행한다[7].

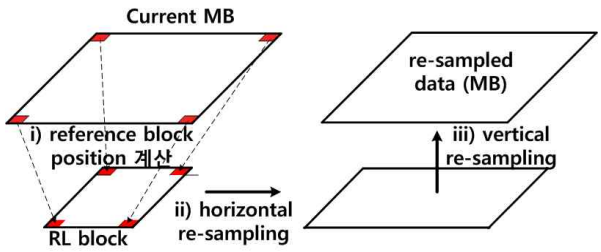

그림 2. 매크로블록 단위 인트라 업샘플링

그림 1에서 보이는 것과 같이, 매크로블록 단위로 레이어간 예측에 사용하는 매크로블록의 타입을 RL로부터 계산한다. 이때 매크로블록의 타입이 인터인 경우, 모션 정보도 함께 업샘플링을 수행한다.

인트라 예측을 수행할 때에는 기존 H.264 AVC에서의 인트라예측뿐만 아니라, RL로부터 업샘플링하여 구한 인트라샘플을 이용한 예측을 함께 수행하여 최적의 모드를 계산한다. 인터 예측에서는 모션 정보 예측할 때에 RL로부터 업샘플링하여 계산 된 모션 정보를 이용한 예측을 함께 수행한다.

현재 인코딩 되는 레이어가 RL로 사용이 되고 그 레이어의 EL이 공간영역으로 스케일러빌리티를 가질 경우에는 그림 1에 보이는 것과 같이 매크로블록 단위 레이어간 디블록킹 필터를 수행하도록 한다. 하나의 엑세스 유닛(access unit (AU))을 인코딩 하기 전에 모든 레이어의 특성을 분석하여 해당 RL이 사용되는 EL의 레이어 정보를 저장하여, 해당 RL 인코딩시 레이어간 디블록킹 필터 수행 여부를 알게 함으로써, 레이어간 디블록킹 필터를 RL 인코딩시 매크로블록 단위로 처리하게 하였다.

3. SVC 인코더의 멀티 프로세싱 구조

본 논문에서는 스케일러블 비디오 인코더의 속도를 높이기 위한 레이어 단위 멀티프로세싱 구조를 제안한다. 먼저 본 논문에서 제안하는 멀티 프로세싱 구조를 적용하기 위해서는 논문의 2장에 기술된 것과 같이 인코딩에 필요한 기능들이 매크로블록단위로 처리되어야 한다.

그림 3은 제안하는 레이어 단위 멀티 프로세싱 구조를 나타낸다. 그림 3에서 보이듯이, 레이어 인코딩은 매크로블록 단위로 수행되며, 하나의 레이어 인코딩이 하나의 스레드에서 수행되도록 하였다. 레이어간 예측을 위해서는 첫 번째 레이어(layer 0)의 인트라 픽셀 및 매크로블록 모드 정보가 두 번째 레이어(layer 1)에서 사용 될 수 있기 때문에, 스레드간 동기화가 필요하다. 즉 그림3의 thread0에서 layer0의 인코딩이 수행 된 후의 layer0 정보를 layer1에서 레이어간 예측에 이용할 수 있기 때문이다. 본 논문에서 제안하는 구조에서는 멀티프로세싱의 동기화의 부하를 줄이기 위하여 매크로 블록단위 레이어 인코딩이 영상 사이즈의 한 열이 끝나는 시점에만 하도록 한다. 그림 3에 보이는 것과 같이, thread0에서 layer0의 매크로블록 인코딩을 수행하면서 영상 한 열이 끝나는 때에 thread1에 이벤트 신호를 주어 layer0의 한 열의 인코딩이 끝났음을 알린다. 그 후 대기 하고 있던 thread1은 thread0으로부터 이벤트 신호를 받아 매크로블록 인코딩을 시작하게 된다. layer0의 영상의 크기가 layer1보다 작을 경우에는 thread1의 대기 시간이 짧아 질것이며, 또한 thread0의 인코딩이 모두 수행되고 난 후부터는 thread1은 대기 없이 매크로블록 인코딩을 수행하도록 한다.

본 논문에서 제안하는 멀티 프로세싱 구조를 이용하면, 레이어 별로 별도의 스레드를 이용하여 처리하기 때문에, 레이어 증가로 인한 연산 부하를 줄일 수 있으며, 이로써 전체 인코더의 속도를 향상시킬 수 있다.

4. 시뮬레이션 결과

본 장에서는 스케일러블 비디오 인코더에 제안하는 멀티 프로세싱 구조를 적용하기 전과 후의 속도를 비교한다. 인코딩의 속도를 측정하기 위해 사용된 시스템은 Intel(R) Core(TM) i7 CPU 860@2.8GHz, 6GB RAM을 이용하였고, 운영체제는 Windows 7이다. 또한 정확한 측정을 위해 한 개의 영상에 대해 인코딩을 네 번 연속 수행한 뒤, 첫 번째 수행에 의해 프로그램 코드와 비트스트림이 캐시 메모리에 로드 될 것이므로[8], 첫 번째 수행에 의한 결과는 제외하고 나머지 세 결과에 대해 중간값을 구하여 속도를 측정하였다.

그림 3. 레이어 단위 멀티 프로세싱 구조

표 1은 시뮬레이션을 위한 인코딩 조건을 나타낸다. configuration a에 영상 4가지를 configuration b에 영상 4가지를 사용하였다.

표 1 시뮬레이션에서 사용한 인코딩 조건

Parameter	Value
Profile	Scalable Baseline
Frame rate	30 Hz
# frames	150
GOP size	4 (IPPP)
Intra period	16
# reference frame	1
ME mode	Fast search
Search range	64
Entropy coding	CAVLC
QPs	26,31,36
layer configuration	configuration (a)
	1 base layer + 2 spatial layers resolution 176x144p/352x288p/704x576p
	configuration (b)
	1 base layer + 1 spatal layer + 1 CGS layer resolution : 352x240p/704x480p/704x480p

표 2에 인코딩 시간을 비교한 것을 보인다. 표 2에서 보이는 것과 같이, 제안한 알고리즘을 적용하였을 경우, 2개의 spatial layer를 가지는 영상에 대해서는 평균 24.8%를 1개의 spatial layer와 1개의 CGS layer를 가지는 영상에 대해서는 평균 82%의 속도 향상이 됨을 알 수 있다.

표 2 인코딩 시간 비교

	stream	Qp	Original (FPS)	Proposed (FPS)	Improved (%)
configuation (a)	city	26	7.79	9.74	25.03
		31	8.19	10.27	25.40
		36	8.68	10.88	25.35
	harbour	26	7.73	9.65	24.84
		31	8.09	10.07	24.47
		36	8.59	10.74	25.03
	ice	26	9.44	11.62	23.09
		31	9.69	12.1	24.87
		36	9.89	12.34	24.77
	soccer	26	8.32	10.39	24.88
		31	8.76	10.94	24.89
		36	9.28	11.59	24.89
	average	31	8.70	10.86	24.79
configuation (b)	bluesky	26	6.1	11.09	81.80
		31	6.27	11.33	80.70
		36	6.44	11.75	82.45
	pedest	26	6.46	11.87	83.75
		31	6.68	12.25	83.38
		36	6.88	12.58	82.85
	rushhour	26	6.32	11.45	81.17
		31	6.64	12.12	82.53
		36	6.96	12.6	81.03
	station	26	6.05	11.03	82.31
		31	6.31	11.48	81.93
		36	6.57	11.95	81.89
	average	31	6.47	11.79	82.15

또한 표 2에서 보이듯이 RL의 영상의 크기와 EL의 영상의 크기가 비슷할수록 큰 속도 향상을 보인다. 이는 기존 방식은 각 레이어의 크기가 클수록 인코딩 속도가 오래 걸리는 반면, 제안한 멀티 프로세싱 구조를 적용하였을 때에는 인코딩의 속도가 최상위 레이어의 크기에 비례하고 최 상위 레이어가 모두 인코딩 되는 시간에 하위 레이어들의 인코딩이 모두 마치기 때문이다.

5. 결론

본 논문에서는 스케일러블 비디오 코딩이 가지는 연산 부하 문제를 해결하기 위한 멀티 프로세싱 구조를 제안하였다. 스케일러블 비디오 코딩을 레이어가 증가함에 따라서 인코딩의 연산시간이 매우 증가 하게 되는데, 레이어 단위 멀티 프로세싱을 적용하여 레이어의 증가에 따른 인코딩의 연산 부하를 줄이도록 하였다. 본 논문에서 제안하는 방법을 적용하였을 경우, 2개의 spatial layer를 가지는 영상에 대해서는 평균 24.8%를 1개의 spatial layer와 1개의 CGS layer를 가지는 영상에 대해서는 평균 82%의 속도를 향상 시킬 수 있다.

[참고문헌]

[1] T. Wiegand, L. Noblet, and F. Rovati, "Scalable Video Coding for IPTV services," *IEEE Trans. Broadcasting*, vol.55, no.2, pp.527-538, June2009.

[2] T. Schierl, T. Stockhammer, and T. Wiegand, "Mobile video transmission using Scalable Video Coding," *IEEE Trans. Circuits Syst. Video Technol*, vol. 17, no. 9, pp. 1204-1217, Sep. 2007.

[3] *Candidate standard: ATSC Mobile DTV Standard, Part 7 - AVC and SVC Video System Characteristics*, A/153 Part7:2009, ATSC, May 2009.

[4] *Advanced video coding for generic audiovisual services,* ITU-T Rec. H.264 and ISO/IEC 14496-10, ITU-T and ISO/IEC JTC 1, Jan. 2009.

[5] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol*, vol. 17, no. 9, pp. 1103-1120, Sep. 2007.

[6] Y.-H. Kim, J.-Y. Yi, and B.-H. Choi, "Fast and memory-efficient up-sampling methods for H.264/AVC SVC with extended spatial scalability," *IEEE Trans. on Consumer Electronics*, vol56,no2,pp. 695-703, May 2010

[7] 이주영, 김용환 "스케일러블 비디오 코딩을 위한 MB 단위 고속 constrained intra re-sampling 방법" *대한전자공학회 하계학술대회 논문집 2010*

[8] V. Lappalainen, A. Hallapuro, and T. D. Hamalainen, "Complexity of optimized H.26L video decoder implementation," *IEEE Trans. Circuits Syst. Video Technol*, vol.13, no.7, pp.717-725, July 2003.