

특집논문-05-10-4-03

Scalable Video Codec을 위한 효율적인 비트율 제어기법

박 내 리^{a)}, 전 동 산^{b)}, 김 재 곤^{b)}, 한 종 기^{a)†}

Efficient Bitrate Control Scheme for Scalable Video Codec

Nae-Ri Park^{a)}, Dong-San Jeon^{b)}, Jae-Gon Kim^{b)} and Jong-Ki Han^{a)†}

요 약

본 논문에서는 현재 JSVM2.0에 적용된 비트율 조절방법과 비교하여 효율적인 비트율 조절방법을 제안한다. 우선, 기존 방법의 문제점을 서술하고 이 문제를 해결하기 위한 효율적 비트할당 방법을 제시한다. JSVM2.0에서 비트율 조절은 초기 양자화 파라미터(QP)와 스케일링 요소를 적용하여 재조정된 QP값에 의해 이루어진다. 즉, 초기 QP 값과 스케일링 요소에 의해 영상의 화질과 발생 비트량이 결정된다. 그런데 기존 방법은 모든 영상에 대해 적절한 초기 QP값을 알아야 한다. 또한, 인코더 파라미터에 의해 선택적으로 매크로블록에 대한 QP변화량을 조절할 수 있지만 이 변화량의 조절은 매우 비효율적이기 때문에 임의의 비트율에서 최적의 화질을 얻기 어렵다. 본 논문에서는 각 프레임의 복잡도에 따라 발생 비트량을 조절함으로써 초기 QP의 영향력을 줄이고 더 나아가 매크로블록의 특성에 따라 매크로블록 단위마다 적절한 QP를 적용하여 프레임 내의 비트율 조절에 대한 효율성을 높인다. 본 논문에서 제안하는 방법은 목표 비트율에 근사한 비트량을 발생 시키도록 부호화하는 것을 가능하게 하며, 기존 방법들과 비교했을 때 같은 비트율에서 개선된 화질의 영상을 얻을 수 있다.

Abstract

In this paper, we propose a new bitrate control scheme to improve the quality of image encoded by SVC and to resolve the problems of conventional scheme. In JSVM2.0, bitrate of a frame is controlled by an initial quantization parameter and scaling factor that it has different value according to frame. It is difficult to get the best of video quality at arbitrary bitrate because the conventional scheme has two defects. One is that we have to know proper initial QP's for all sequences. Another is that QP's control skill for macroblocks is very inefficient. In this paper, we propose an efficient bit allocation algorithm to reduce the effect of the initial QP and to increase the efficiency of bit allocation by using proper QP's for macroblocks. In simulation results, it can be seen that using the proposed scheme enables the SVC encoder to control the bitrate by the macroblock unit and outperforms the conventional schemes in the respect of rate-distortion.

Keyword: scalable video codec(SVC), rate control

I. 서 론

최근 인터넷 스트리밍 비디오 서비스와 다양한 휴대 이

동 단말기의 등장은 새로운 코덱기술의 개발을 이끌었다. 인터넷 스트리밍 비디오등과 같이 유무선 네트워크를 이용한 서비스를 제공하기 위해 가변 채널 환경을 고려한 비디오 압축 기술의 필요성이 증가하였다. 또한, 다양한 휴대 이동 단말기 사용의 증가는 다중 해상도와 다중 프레임율을 지원하는 비트스트림을 필요로 하기 시작하였다.

a) 세종대학교 정보통신공학과
Dept. of Information & Communications Engineering, Sejong University.
b) 한국전자통신연구원 디지털방송연구단
Broadcasting Media Research Group, ETRI

이러한 필요성에 의해 등장한 스케일러블 비디오 코덱은 이전의 코덱(MPEG-1, MPEG-2, MPEG-4)들과 비교하여 여러 가지 다른 특징을 갖는다. 이전의 코덱은 정해진 채널 용량에서 비트스트림의 전송 및 복호가 가능한 반면, 스케일러블 비디오 코덱은 채널 용량의 변화에 따라 적응적으로 비트스트림의 복호 과정을 조절한다. 이를 가능하도록 하기 위해 SVC는 크게 세 가지의 스케일러빌리티 기능을 제공한다^{[2][4]}. 첫째, 해상도 스케일러빌리티(Spatial scalability) 기능은 채널용량의 변화에 따라 수신된 데이터의 일부분을 이용하여 다른 해상도의 비디오 신호로 복원하도록 한다. 둘째, 프레임율 스케일러빌리티(Temporal scalability) 기능은 수신 데이터를 여러가지 프레임율을 갖는 비디오 신호로 복호될 수 있도록 한다. 마지막으로 화질 스케일러빌리티(SNR scalability) 기능이다. 이 기능은 채널 용량의 제약으로 정보의 일부분만 수신하여도 그 일부분의 신호만 복호함으로써 다른 화질의 신호로 재생할 수 있도록 한다.

위와 같은 특징들을 갖는 SVC코덱의 성능을 향상시키기 위해 여러 가지 효율적인 방법들이 제안되어 왔다. 즉, 가변 GOP기술, 효율적 움직임 추정기술, 최적화된 양자화기 등이 그 예이다. 이러한 기술들 가운데 각 부호화 요소별 비트할당 기술은 SVC 코덱의 성능을 향상시키기 위한 가장 중요한 기술들 중 하나이다. 이 비트할당 기술은 코덱 내 다른 모듈(예를 들면, 움직임 추정, 양자화)들에 영향을 끼치면서 영상의 화질을 결정하기 때문이다. 따라서 SVC를 위한 효율적인 비트할당 기술은 SVC 코덱의 성능을 향상시키기 위해서 필수적으로 개발되어야 할 기술이다. 그러나, 기존의 SVC 코덱에서 비트율 조절 방법은 모든 영상에 대해 적절한 초기 QP값을 알아야 한다. 또한, 인코더 파라미터에 의해 선택적으로 매크로블록에 대한 QP변화량을 조절할 수 있지만, 이 변화량의 조절은 매우 비효율적이기 때문에 임의의 비트율에서 최적의 화질을 얻기 어렵다. 따라서, 각 프레임의 복잡도에 따라 발생 비트량을 조절하고, 각 매크로블록 단위마다 적응적으로 QP를 적용하여 프레임 내의 비트율 조절에 대한 효율성을 높일 수 있는 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 현재 SVC에

서 사용하고 있는 비트율 제어 방법을 간단히 설명한다. 3장에서는 기존 방법의 문제점을 해결하고 코덱의 성능을 개선하기 위한 알고리즘을 제안한다. 제안하는 알고리즘과 기존 알고리즘을 비교한 실험 결과를 4장에서 설명한다. 5장에서 본 논문의 결론을 맺는다.

II. SVC의 비트율 제어

현재SVC(JSVM2.0)^[3]의 비트율 조절과정은 초기 QP 설정과 스케일링 요소에 의해 QP를 재조정 하는 과정으로 이루어져있다. 초기 QP는 임의의 값을 가질 수 있으며 QP가 6씩 증가함에 따라 양자화 스텝 사이즈는 두 배가 된다. 따라서, QP가 커짐에 따라 발생 비트율은 감소한다. 또한, 스케일링 요소는 MCTF^[5-8]의 특성을 고려하여 결정된다. MCTF에서 시간적 분해를 반복할수록 스케일링 요소는 큰 값을 가지며, 이 값이 커짐에 따라 QP는 감소한다. 이러한 특징을 갖는 두 요소의 결합으로 QP는 재조정되고, 재조정된 QP에 의해 영상의 발생 비트율과 화질이 결정된다. 다음의 1절에서 QP의 의미와 결정 방법을 기술하고, 2절에서 스케일링 요소의 결정 과정을 살펴본다. 마지막으로 3절에서는 각 프레임마다 결정된 QP를 각 매크로블록을 고려하여 재조정하는 방법을 기술한다.

1. 양자화 파라미터(QP)의 의미와 결정

JSVM2.0은H.264/AVC와 호환 가능하도록 하기 위해 이를 기반으로 모델링 되었다. 따라서 H.264/AVC에서의 양자화 방법과 동일하며 기본적인 양자화 연산 방법은 다음과 같다.

$$Z = \left(\frac{|\omega| + f \cdot \Delta}{\Delta} \right) \tag{1}$$

식(1)에서 Z는 양자화 후의 샘플 값을 나타낸다. ω 는

표 1. H.264에서의 양자화 스텝사이즈와 양자화 파라미터의 관계
Table 1. Relations of quantization step size and quantization parameter in H.264

QP	0	1	2	3	4	5	6	7	8	9	10
Qstep	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2
QP	18	24	30	36	42	51
Qstep	5	10	20	40	80	224

주파수 변환한 결과 샘플 값이고 f_n 는 오프셋을 나타낸다. Δ (Qstep)는 양자화 스텝 사이즈를 나타내는데 이 값에 의해 양자화의 정도가 결정된다. 이 양자화 스텝 사이즈는 총 52가지 경우를 갖는다. QP는 이 양자화 스텝 사이즈를 나타내기 위한 인덱스를 의미한다. 표 1에서 양자화 스텝 사이즈와 QP의 관계를 나타내었다. SVC에서 양자화 과정은 H.264/AVC와 마찬가지로 표. 1의 양자화 스텝 사이즈와 QP와의 관계를 이용한다. 이러한 QP의 적용은 영상의 발생 비트율과 화질을 결정하므로 영상의 특성을 이용하여 적절한 QP를 결정하는 것이 중요하다. SVC에서 QP결정 방법은 다음의 식(2)와 같다. 식(2)에서 iQP 는 임의의 초기 QP값을 나타내며, $dScalingFactor$ 는 SVC에서 MCTF의 특성을 고려하여 결정되는 스케일링 요소를 나타낸다. 이 $dScalingFactor$ 를 결정하는 방법은 다음 2절에서 자세히 설명한다^[1-4].

$$dQP = iQP - (6 \times \log_2(dScalingFactor)) \quad (2)$$

$$QP = \max(0, \min(51, \lfloor dQP + 0.5 \rfloor)) \quad (3)$$

식(2)에서 초기QP(iQP)와 스케일링 요소($dScalingFactor$)로 결정된 dQP 는 식(3)과 같이 QP를 나타낼 수 있는 0부터 51까지의 범위 내에서 정수형의 값으로 조절된다.

2. 스케일링 요소(scaling factor)의 결정 과정

스케일링 요소, 즉 식(2)의 $dScalingFactor$ 값은 SVC의 프레임울 스케일러빌리티를 지원하는 MCTF의 특성을 고려하여 결정된다^[5-8]. 그림 1은 한 GOP에 대한 MCTF 분해

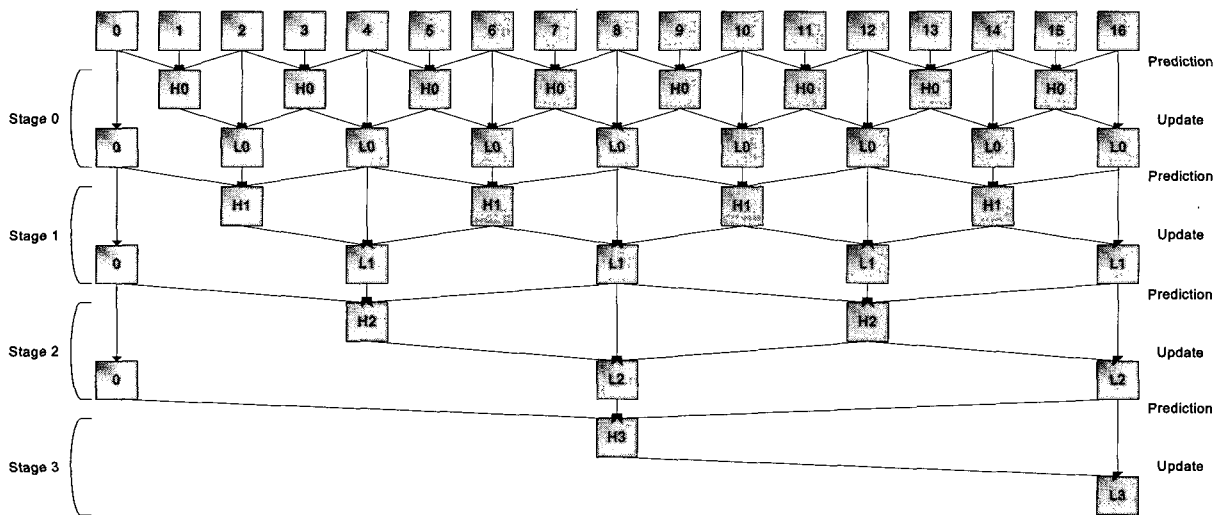


그림 1. MCTF 분해 과정
Fig1. MCTF Decomposition process

과정을 보여준다. 그림 1과 같이 MCTF 분해 과정은 GOP 단위로 수행되며, 각 예측과정과 업데이트 과정을 거쳐 High-Pass 프레임(H)과 Low-Pass 프레임(L)을 생성한다^{[2][4]}. 이 때 H 프레임은 대상 프레임과 예측 프레임의 차이 신호이며, L 프레임은 다음 분해 단계의 H 프레임을 생성하기 위해 원 신호의 고주파 성분을 제거하여 재구성한 신호이다. 그림 1에서 stage는 MCTF 분해 단계를 나타낸다. 분해단계가 증가함에 따라 즉, MCTF 분해과정이 반복됨에 따라 한 분해단계내의 프레임 수는 이전 분해 과정의 반으로 줄어든다. 이러한 과정은 분해 단계 내에 H프레임과 L프레임이 각각 하나가 남을 때까지 반복된다.

예측과 업데이트 과정은 단방향 예측과 양방향 예측 방법을 사용하여 H 프레임과 L 프레임을 생성하는데, 각 예측 방법과 프레임의 종류에 따라 표 2와 같이 다른 정규화 변수를 갖는다. 이 정규화 변수는 스케일링 요소를 결정하는 역할을 한다.

표 2. 예측 방법과 프레임의 종류에 따른 정규화 변수
Table 2. Normal variables according to prediction type and frame type

	단방향 예측 (Haar filter)	양방향 예측 (5/3 filter)
H 프레임	$\sqrt{1/2}$	$\sqrt{23/32}$
L 프레임	$\sqrt{2/1}$	$\sqrt{3/2}$

스케일링 요소를 결정하는 또 다른 요소는 한 프레임내의 connected pixels의 비율^[7]로, 식 (4)-(6)과 같이 계산되어진다. 예측 과정과 업데이트 과정의 불일치로 인해 프레임 내에 connected pixels과 unconnected pixels이 발생되는데, 이때 connected pixels은 각 블록이 얼마나 신뢰할 수 있는 블록 인지를 나타내는 잣대가 된다. 따라서 MCTF 분해 단계에 따라 참조 프레임의 신뢰도를 connected pixels의 비율로 나타낸다. 한 프레임에서 connected pixels의 비율은 각 예측타입에 따라 다음의 식으로 나타낸다.

$$RateL0 = N_{L0} / N_{Total} \tag{4}$$

$$RateL1 = N_{L1} / N_{Total} \tag{5}$$

$$RateBi = N_{Bi} / N_{Total} \tag{6}$$

식(4)-(6)에서 N_{Total} 은 한 프레임내의 총 픽셀의 수이다. 또한 N_{L0} , N_{L1} , N_{Bi} 는 각각 순방향 예측, 역방향 예측, 양방향 예측된 픽셀의 수를 나타낸다. 식(4)-(6)에 의해 결정된 connected pixels의 비율과 정규화 변수를 적용하는 과정을 그림 2에서 나타내었다. 그림 2의 각 변수들은

< CASE 1 >

$$\begin{aligned} & (RateBi[iFrame-1] + RateBi[iFrame+1]) \times (Factor_53_LP-1) / 2 \\ & + (RateL1[iFrame-1] + RateL0[iFrame+1]) \times (Factor_22_LP-1) / 2 + 1 \end{aligned} \tag{7}$$

< CASE 2 >

$$\begin{aligned} & RateBi[iFrame-1] / 2 \times (Factor_53_LP-1) \\ & + RateL1[iFrame-1] \times (Factor_22_LP-1) + 1 \end{aligned} \tag{8}$$

< CASE 3 >

$$\begin{aligned} & RateBi[iFrame+1] / 2 \times (Factor_53_LP-1) \\ & + RateL0[iFrame+1] \times (Factor_22_LP-1) + 1 \end{aligned} \tag{9}$$

< CASE 4 >

$$\begin{aligned} & RateBi[iFrame] \times (Factor_53_HP-1) \\ & + (RateL0[iFrame] + RateL1[iFrame]) \times (Factor_22_HP-1) + 1 \end{aligned} \tag{10}$$

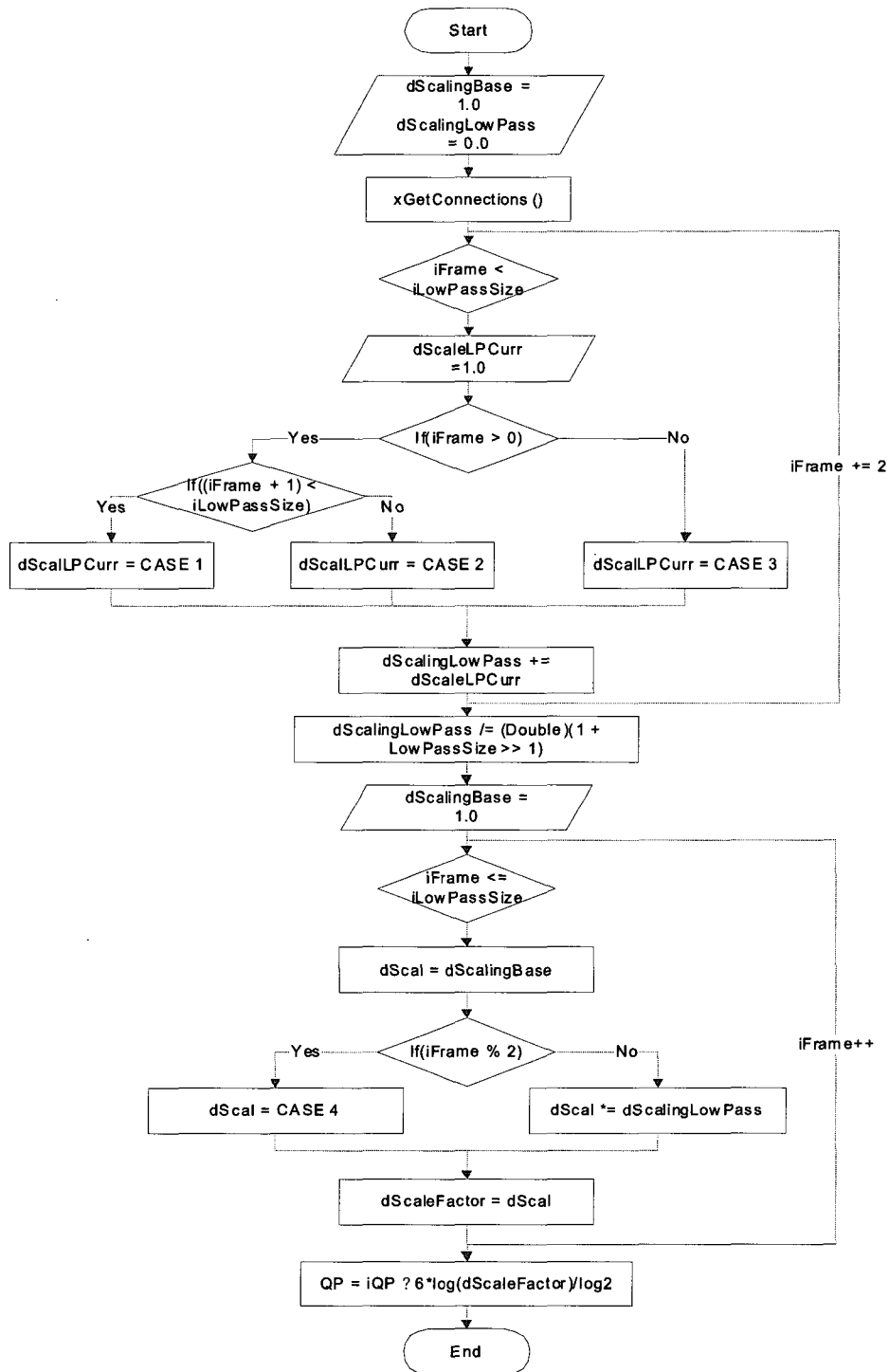


그림 2. 스케일링 요소의 계산과정
 Fig 2. Computational process of scaling factor

JSVM2.0 소프트웨어 내의 변수와 동일하게 표기 하였다.

또한 그림 2의 $xGetConections()$ 는 connection 비율을 계산하는 함수이다.

그림 2에 의해 그림 1과 같은 GOP수의 경우 프레임 넘버가 0이나 16이 아닐 경우에 L 프레임의 스케일링 요소는 CASE 1에 의해 계산되고, 16일 경우 CASE 2, 0일 경우 CASE 3에 의해 계산 된다. 각 CASE 1, 2, 3은 식(7)-(9)에서 나타내었다.

위의 표 2에서 정규화 변수는 다음과 같은 변수로써 재정의할 수 있다. 단방향 예측을 사용한 H 프레임은 $Factor_22_HP$ 로 표기하며 단방향 예측을 사용한 L 프레임은 $Factor_22_LP$ 로 나타낸다. 또한, 양방향 예측을 사용한 H 프레임은 $Factor_53_HP$ 로 표기하며, L 프레임은 $Factor_53_LP$ 로 나타낸다.

식(7)-(9)는 각 프레임의 connected pixels의 비율과 정규화 변수를 이용하여 스케일링 요소를 계산하는 방법을 보여준다. 식(7)-(9)에 의해서 결정된 L 프레임의 스케일링 요소는 각 분해 단계 내에서 평균을 취하여 해당 분해단계 내에 H 프레임의 스케일링 요소를 결정하기 위한 초기값이 된다. H 프레임의 스케일링 요소는 식(10)에 의해 최종적으로 결정된다.

위의 과정에 의해 스케일링 요소는 분해 단계가 높아질수록, 즉 MCTF 분해과정을 반복할수록 큰 값을 갖는다. 또한, 프레임의 종류에 대해 L 프레임의 스케일링 요소가 H 프레임의 스케일링 요소보다 큰 값을 갖는다. 스케일링 요소 값이 클수록 위의 식(2)에 의해 작은 양자화 파라미터 값으로 결정되므로 스케일링 요소값이 큰 프레임에 더 많은 비트가 할당된다. 이는 양자화 파라미터가 프레임의 에너지에 반비례적으로 결정되어야 하는 특성과 일치한다.

3. 매크로블록을 위한 QP의 적용

앞서 설명한 1, 2절의 방법에 의해 결정된 각 프레임의 QP는 인코더 파라미터 파일에서 선택된 QP의 변화량만큼 매크로블록 단위로 조절되며, 이 변화량은 H프레임의 차이 신호 블록에 대해서만 적용된다. QP의 변화량을 ΔQP

라고 하자. 이 때 각 매크로블록이 가질 수 있는 QP의 최소값과 최대값은 각각 식 (11), (12)와 같다.

$$MinQP = QP - \Delta QP \tag{11}$$

$$MaxQP = QP + \Delta QP \tag{12}$$

예를들어, 식 (11), (12)의 ΔQP 가 1이고, QP가 30이라고 할 때 한 프레임 내 매크로블록이 가질 수 있는 QP는 29, 30, 31이 된다. 위와 같이 결정된 QP는 프레임 내 매크로블록에 순차적으로 적용된다. 즉 0번째 매크로블록을 위한 QP는 29이고, 1번째 매크로블록을 위한 QP는 30이 된다. 또한, 2번째 매크로블록을 위한 QP는 31의 값을 갖는다. 이와 같은 순으로 마지막 매크로블록까지 3개의 QP값이 반복적으로 할당된다. 이렇게 할당된 QP는 각 매크로블록이 가지고 있는 성질을 고려하고 있지 않기 때문에 비트율 대비 PSNR면에서 효율적인 QP를 할당한다고 판단하기 힘들다. 즉, 한 프레임 내 모든 매크로블록에 대해 같은 QP를 할당했을 때와 거의 동일한 결과를 보인다. 따라서 현재 JSVM에 적용되어 있는 매크로블록당 QP의 조절 방법은 매우 비효율적이기 때문에 이를 개선하기 위한 새로운 QP 조절방법이 필요함을 알 수 있다.

III. 제안하는 알고리즘

기존의 알고리즘은 모든 시퀀스에 대해 적절한 초기 QP 값을 알고 있을 때 고화질의 영상을 얻을 수 있다. 그러나 영상에 대한 초기 QP값을 알 수 없을 때 여러 가지 문제점이 생긴다. 예를 들어, 움직임이 많은 영상을 큰 QP로 부호화 할 경우 저화질의 영상을 얻게 될 것이다. 또한, 움직임이 적은 영상을 작은 QP로 부호화 할 경우 발생 비트량이 커질 것이다. 본 논문에서는 이런 기존 알고리즘의 문제점을 해결하고 목표 비트량에 맞춰 부호화 할 수 있는 비트량당 알고리즘을 제안한다.

제안하는 알고리즘에서는 양자화 파라미터를 프레임의 에너지에 반비례적으로 결정한다. 프레임의 에너지를 화소의 제곱의 평균으로 나타냈을 때, 이 값이 큰 프레임은 에너지를 많이 포함한 프레임이 될 것이다. 따라서 작은 QP로 부호화하여 많은 비트를 할당함으로써 화질을 유지하도록 한다. 반대로 제곱의 평균값이 작은 프레임은 에너지를 적게 포함하고 있으므로 큰 QP로 부호화할 때 적은 비트량으로도 일정 화질을 유지할 수 있다. 즉, 프레임의 에너지에 따라 다른 QP를 할당할 때에 효율적인 부호화가 가능해진다. 기존의 알고리즘에서는 connected pixels의 비율과 정규화 변수에 의해 결정된 스케일링 요소를 이용하여 프레임의 에너지를 결정하는데, 이 방법은 프레임의 복잡도를 이용하여 프레임의 에너지를 결정하는 방법에 비해 비효율적이다. 기존 알고리즘의 connected pixels의 비율은 예측 과정에서 참조 프레임의 신뢰도를 나타내기 위한 값이며 현재 프레임에 대한 정보를 포함하지 않기 때문이다. 따라서 본 논문에서는 프레임의 복잡도를 고려하여 이를 근거로 프레임당 목표 비트를 할당하는 방법을 제안한다. 또한, 매크로블록의 특징을 고려하여 각 매크로블록마다 적합한 양자화 파라미터를 할당함으로써 더욱 효율적으로 비트할당이 이루어지도록 한다. 다음의 1절과 2절에서는 제안하는 알고리즘의 각 단계를 서술한다.

1. 비트할당(bit allocation)

SVC에서는 부호화 전에 GOP단위로 준비과정을 수행한다. 즉, 각 다른 해상도의 레이어에 대해 MCTF 과정이 수

행되어 GOP내에 하나의 L프레임과 (N-1)개의 H 프레임을 생성한다. 이 때, N은 GOP크기를 의미한다. 따라서 부호화 전에 한 GOP내의 프레임 종류와 MCTF분해 단계를 알 수 있을 뿐만 아니라, 부호화하고자 하는 프레임의 복잡도 또한 알 수 있다. 부호화 전에 프레임의 모든 정보를 알고 있다는 특징을 이용하여 각 프레임을 위한 비트할당을 GOP단위로 수행하고, 프레임의 종류와 분해 단계에 따라 다른 목표 비트를 할당하도록 한다.

그림 3은 MCTF 분해 과정에 의해 생성된 프레임은 나타낸다. GOP크기 N에 대해 $2^K = N$ 을 만족하는 K번의 분해과정을 거치는데, 이 분해과정이 반복됨에 따라 프레임의 수는 반으로 줄어든다. 각 분해 단계에 따라 H프레임을 각각 H0, H1, H2로 표기하였다. 이 때 분해과정을 반복하여 높은 단계에 속한 프레임일수록 현재 프레임과 멀리 떨어진 프레임에서 움직임이 예측되므로 큰 잔여신호를 포함하고 있다. 또한 이러한 프레임은 복호할 때 낮은 단계에 있는 프레임으로부터 참조됨으로 중요도가 높다. 즉, 그림 3의 H프레임은

$$H2 > H1 > H0$$

순으로 차이잔여신호를 많이 포함하며 중요도가 높다.

프레임의 종류와 분해 단계에 따라 프레임의 복잡도를 고려하여 목표 비트를 할당하기 위한 방법으로 아래의 식을 제안한다. L 프레임에 대한 목표 비트(T_L)는 식(13)을 이용하여 계산한다. 식(13)에서 L, HS2, HS1, HS0는 각각 L 프레임, stage 2에서의 H 프레임, stage 1에서의 H

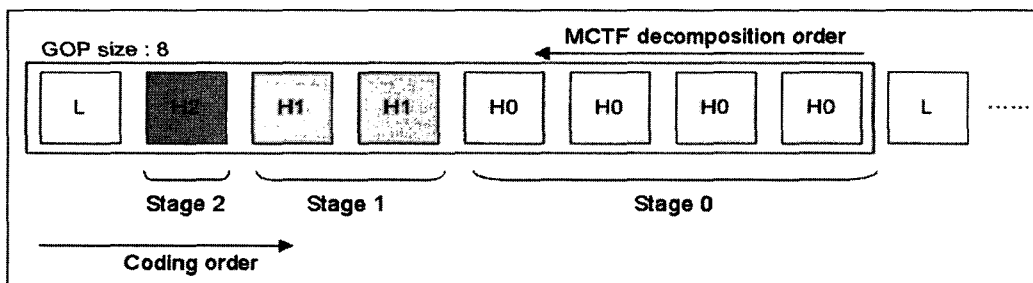


그림 3. MCTF 분해 과정으로 생성된 프레임
Fig 3. Generated frames from MCTF decomposition process

$$T_L = \left(\frac{R_G}{N_L + N_{HS2} \frac{Avg(X_{HS2})}{Avg(X_L)} + N_{HS1} \frac{Avg(X_{HS1})}{Avg(X_L)} + N_{HS0} \frac{Avg(X_{HS0})}{Avg(X_L)}} \right) \times \frac{X_L}{Avg(X_L)} \quad (13)$$

$$T_{HS2} = \left(\frac{R'_G}{N_{HS2} + N_{HS1} \frac{Avg(X_{HS1})}{Avg(X_{HS2})} + N_{HS0} \frac{Avg(X_{HS0})}{Avg(X_{HS2})}} \right) \times \frac{X_{HS2}}{Avg(X_{HS2})} \quad (14)$$

$$T_{HS1} = \left(\frac{R'_G}{N_{HS1} + N_{HS0} \frac{Avg(X_{HS0})}{Avg(X_{HS1})}} \right) \times \frac{X_{HS1}}{Avg(X_{HS1})} \quad (15)$$

$$T_{HS0} = \left(\frac{R'_G}{N_{HS0}} \right) \times \frac{X_{HS0}}{Avg(X_{HS0})} \quad (16)$$

프레임, stage 0에서의 H 프레임을 의미한다. R_G 는 한 GOP내에 할당된 목표 비트량이며 목표 비트 T 에 대해 $R_G = (T / \text{frame rate}) \times N$ 으로 계산된다.

$N_i, i = L, HS2, HS1, HS0$ 는 GOP내에 존재하는 각 프레임 종류별 수를 나타낸다. X_i 는 각 프레임의 복잡도를 나타내며, $Avg(X_i)$ 는 프레임의 복잡도를 분해단계 별로 평균한 값이다. 식(13)과 같은 개념으로 stage2의 H 프레임을 위한 목표 비트 계산식은 식(14)로 서술될 수 있다. 식(14)의 R'_G 는 GOP에 할당된 비트량에서 이전 프레임 즉, L프레임 부호화 과정에서 발생된 비트량을 제외한 비트량이다. 식(15), (16)은 각각 stage1과 stage0 내의 프레임에 대한 목표 비트량 계산 식을 나타낸다.

위의 식(13)-(16)에 의해 프레임의 종류와 분해단계에 따라 다른 비트량을 할당하고 한 단계내에서도 프레임의 복잡도를 고려하여 비트할당을 다르게 하였다. 식(13)-(16)에서 이용된 복잡도는 프레임내 화소의 제곱의 합으로써 계산될 수 있다. 따라서 MCTF분해 후 생성된 H 프레임과

L 프레임 내의 화소값을 이용하여 프레임의 에너지를 계산할 수 있다. 각 프레임의 복잡도는 $X_L, X_{HS2}, X_{HS1}, X_{HS0}$ 로 표기하며 이 값은

$$X_L > X_{HS2} > X_{HS1} > X_{HS0}$$

의 경향을 갖는다. 각 프레임의 복잡도는 식(17)를 이용하여 계산된다. 식(17)에서 P_i 는 각 프레임의 화소값을 나타낸다.

$$X_i = \sqrt{\sum (P_i)^2}, i = L, HS2, HS1, HS0 \quad (17)$$

식(13)-(16)에서 계산된 프레임에 대한 목표 비트량은 식(18)의 범위 내에서 할당되도록 한다. GOP 단위 내의 부호화 과정에서 나중 순서에 있는 프레임을 위한 비트할당을 적절히 하기 위해 목표 비트량의 최소값과 최대값을 제한한다.

$$\frac{TargetBit}{FrameRate} \times \frac{1}{2} \leq T_L, T_{HS2}, T_{HS1}, T_{HS0} \leq \frac{TargetBit}{FrameRate} \times 2 \quad (18)$$

2. 비트율 조절(Rate Control)

이 절에서는 MPEG-2 TM5^[10]의 비트율 조절 방법을 응용하여 가상버퍼의 충만도에 따라 각 매크로블록에서 사용될 양자화 파라미터의 참조값을 정하는 과정을 설명한다. 이 때 참조값은 현재 부호화하려는 매크로블록의 양자화 파라미터를 결정하기 위한 기준값이 된다. 식(13)-(16)에 의해 구해진 각 프레임의 목표 비트량과 실제로 부호화 하여 발생한 비트량의 차를 이용하여 현재 매크로 블록에서 사용될 양자화 파라미터를 조절한다. 식(19)-(22)은 j 번째 매크로블록에서 사용될 QP를 결정하는 과정을 설명한다.

$$D_j^i = D_0^i + \left(B_{j-1}^i - \frac{T_i \times (j-1)}{MB_total} \right), \quad i = L, HS2, HS1, HS0 \quad (19)$$

$$D_0^i = (iQP - AvgQP^i) \times \frac{\gamma}{52}, \quad \gamma = 2 \times \frac{bit\ rate}{frame\ rate} \quad (20)$$

위 식(19)에서 B_{j-1}^i 은 초기 양자화 파라미터를 이용하여 부호화했을 때 $j-1$ 번째 매크로블록을 부호화할 때까지 누적 발생된 비트량을 의미한다. 식(20)에서 D_0^i 는 초기 가상버퍼의 충만도이며, γ 는 reaction 파라미터를 나타낸다. 또한, $AvgQP^i$ 는 같은 분해 단계 내에 존재하는 이전 프레임의 평균 양자화 파라미터이다. 식(19), (20)에 의해 현재 매크로블록을 부호화하기 직전의 가상버퍼충만도를 계산할 수 있다. 가상버퍼의 충만도를 이용하여 식 (21)와 같이 j 번째 매크로블록의 양자화 파라미터를 계산한다. 식 (21)에 의해 매크로블록 단위의 양자화 파라미터를 결정함으로써 프레임 내에서 효율적인 비트할당이 가능해졌다. 이 때, 초기양자화 파라미터 iQP 는 식(22)과 같은 방법으로 결정한다.

$$QP_j = \frac{D_j \times 52}{\gamma} \quad (21)$$

$$\begin{aligned} & \text{if } (bpp \leq 0.2) \quad iQP = 38; \\ & \text{else if } (bpp \leq 0.5) \quad iQP = 33; \\ & \text{else if } (bpp \leq 0.8) \quad iQP = 28; \\ & \text{else} \quad iQP = 23; \end{aligned} \quad (22)$$

식(22)에서 bpp 는 $bitrate / (frame\ rate \times width \times height)$ 와 같이 한 프레임 내의 화소 당 비트수(bits/pixel)로 나타낸다.

IV. 실험 결과

본 장에서는 제안하는 알고리즘의 성능을 평가하기 위한 실험 결과를 나타낸다. 제안하는 알고리즘의 성능을 확인하기 위해 JSVM 2.0에 적용되어 있는 MCTF 구조에서 실험할 뿐만 아니라, Poznan 회의를 통해 non-normative 부분에서 채택된 closed-loop hierarchical B picture 구조에서의 실험을 병행한다^{[11][12]}. SVC에서 예측구조는 spatial base layer에서 open-loop hierarchical B picture 구조를 가지며, spatial enhancement layer에서 MCTF 구조를 갖는다. MCTF 또한 open-loop 예측 구조를 지원한다. 이는, 예측 시 양자화 에러를 포함한 복원된 영상으로부터 예측하는 방법이 아닌, 원 영상에서 예측을 수행한다. 반면, closed-loop 예측 구조는 양자화 에러를 포함한 복원된 영상으로부터 예측하는 방법을 의미하며 상당한 화질 향상을 가져왔다^[11]. 따라서 두 가지 다른 예측 구조에서 제안하는 방법이 가져오는 화질 향상 정도를 비교해 본다. 또한, 제안하는 알고리즘의 비교 실험을 위해 제2장에서 설명한 양자화 파라미터 결정방법뿐만 아니라 추가의 두 가지 다른 모드들과도 비교한다. 제 2장에서 설명한 방식과 같이 스케일링 요소를 계산하여 프레임에 따라 다른 양자화 파라미터를 적용한 방법을 'Adaptive QP'로 표기한다. 또 다른 비교 방법으로써 'Standard QP'는 스케일링 요소계산 시 connected pixels의 비율을 일정한 값으로 고정 한 후 양자화 파라미터에 적용한 방법이다^[1-4]. 이때 connected pixels의 비율은 다음의 값으로 고정된다.

$$\begin{aligned} RateL0 &= 0.2 \\ RateL1 &= 0.2 \\ RateBi &= 0.6 \end{aligned}$$

마지막으로 다른 비교 대상은 'Fixed QP' 모드이다. 실험을 위해 모든 프레임에 대해 동일하게 양자화 파라미터를 적용하였다. 이때 양자화 파라미터는 초기 양자화 파라미터와 동일하다. 위에서 설명한 'Adaptive QP'와 'Standard QP' 모드는 JSVM2.0의 인코더 파라미터 파일에서 결정 가능하다^[3].

SVC는 spatial scalability를 지원하는데, 이 때 spatial base layer는 H.264/AVC와 호환 가능하므로 SVC에 도입된 MCTF 방식을 수행하지 않고 'Hierarchical B picture' 방법으로 프레임 스케일러빌리티를 가능하도록 한다. 따라서 MCTF와 달리 업데이트 과정이 없기 때문에 connected pixels의 개념이 존재하지 않는다. 이와 같은 이유로 spatial base layer에서 양자화 파라미터를 결정짓는 요소인 connected pixels의 비율은 'Standard QP'와 같이 고정된 비율만 지원된다.

또한, Spatial enhancement layer는 MCTF의 예측과정과 업데이트 과정에서 다른 예측 모드로 인해 다른 정규화 변수를 가지는 반면 'Hierarchical B picture'를 사용하는 spatial base layer는 'update' 과정이 존재하지 않으므로 base layer를 위한 새로운 정규화 변수 값을 가지며^{[3][9]}, 식(23), (24)와 같다.

$$Factor_53_HP_BL = 1.0 \quad (23)$$

$$Factor_22_HP_BL = 1.0 \quad (24)$$

제안하는 알고리즘의 두 단계를 적용한 실험 결과를 나타내기 위한 테스트 영상은 'Football', 'Crew', 'Foreman', 'Mobile' 을 사용하였다. Spatial base layer는 QCIF@15Hz의 해상도와 프레임율을 갖고, spatial enhancement layer는 CIF@30Hz의 해상도와 프레임율을 갖는다. 기존 알고리즘과 제안하는 알고리즘의 성능을 RD curve로 비교하였으며, 프레임별 발생 비트율을 그래프로 나타내었다.

그림 4-7은 각각 spatial base layer만을 부호화 하고 복호화 한 'Football', 'Crew', 'Foreman', 'Mobile' 영상의 실험 결과이다. 제안하는 알고리즘은 open-loop 예측 구조에서 'Fixed QP'와 'Standard QP' 두 가지 모드와 비교하였으며, 기존의 비트율 제어 기법인 TM5와 비교하였다. 실험을 위한 'Football' 영상의 목표 비트율은 각 192 / 224 / 256 / 320 / 384 kbit/sec로 하였다. 이 때, 각 비교모드의 비트율을 맞추기 위한 초기 양자화 파라미터는 각 31 / 30 / 29 / 27 / 25의 값으로 설정하였다. 'Crew' 영상의 타겟 비트율은 96 / 112 / 128 / 160 / 192 kbit/sec로 하였다. 각 비교방법의 비트율을 맞추기 위한 초기 양자화 파라미터는 각 34

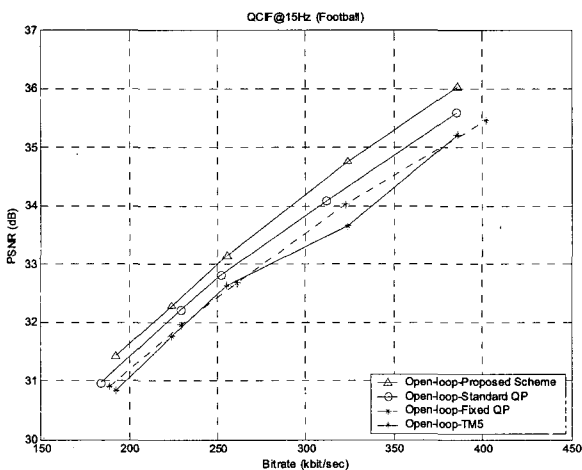


그림 4. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Football', QCIF, Open-loop)
Fig 4. RD curve of conventional algorithm and proposed algorithm ('Football', QCIF, Open-loop)

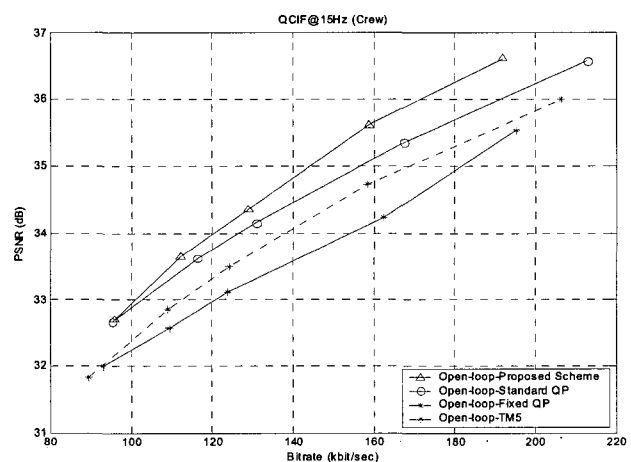


그림 5. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Crew', QCIF, Open-loop)
Fig 5. RD curve of conventional algorithm and proposed algorithm ('Crew', QCIF, Open-loop)

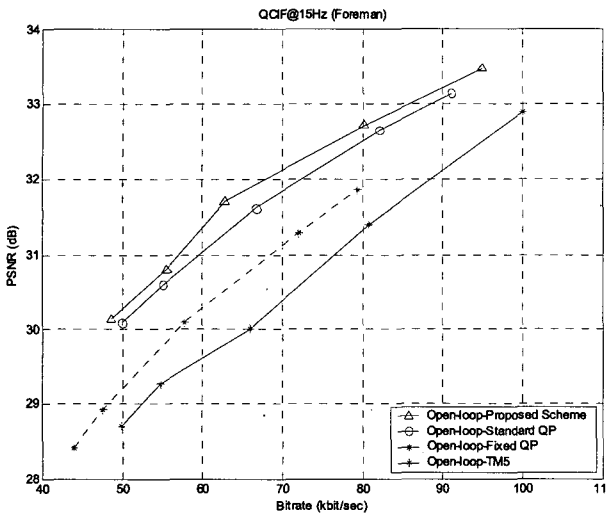


그림 6. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Foreman', QCIF, Open-loop)
 Fig 6. RD curve of conventional algorithm and proposed algorithm ('Foreman', QCIF, Open-loop)

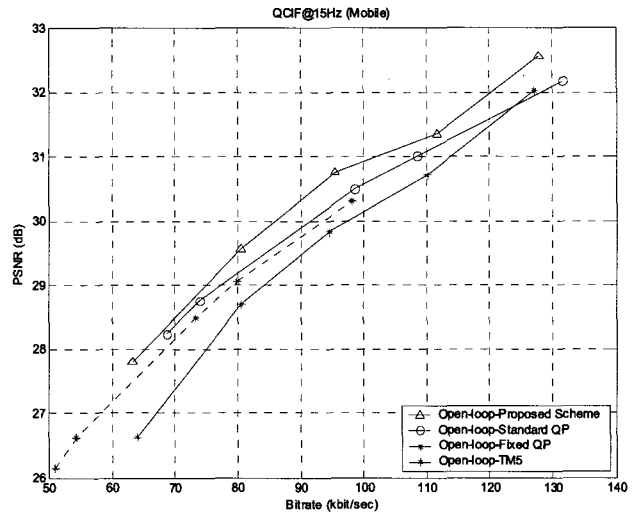


그림 7. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Mobile', QCIF, Open-loop)
 Fig 7. RD curve of conventional algorithm and proposed algorithm ('Mobile', QCIF, Open-loop)

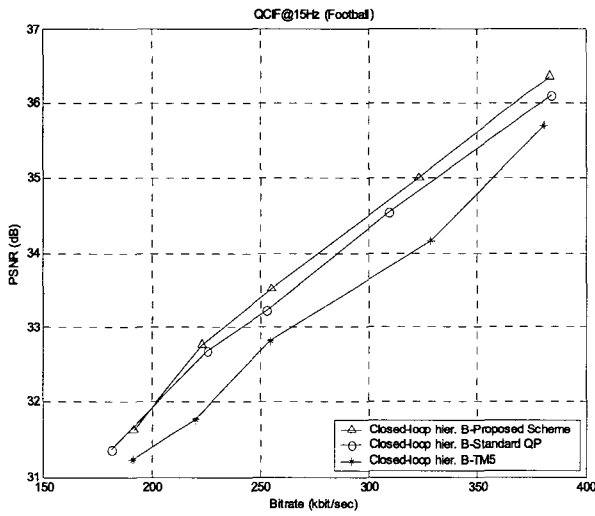


그림 8. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Football', QCIF, Closed-loop)
 Fig 8. RD curve of conventional algorithm and proposed algorithm ('Football', QCIF, Closed-loop)

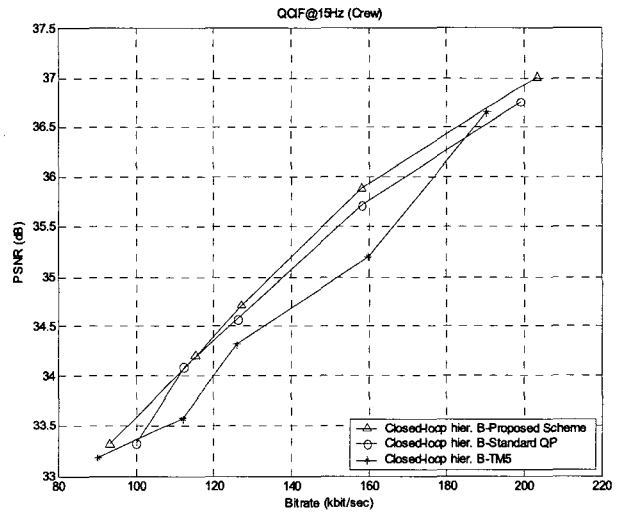


그림 9. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Crew', QCIF, Closed-loop)
 Fig 9. RD curve of conventional algorithm and proposed algorithm ('Crew', QCIF, Closed-loop)

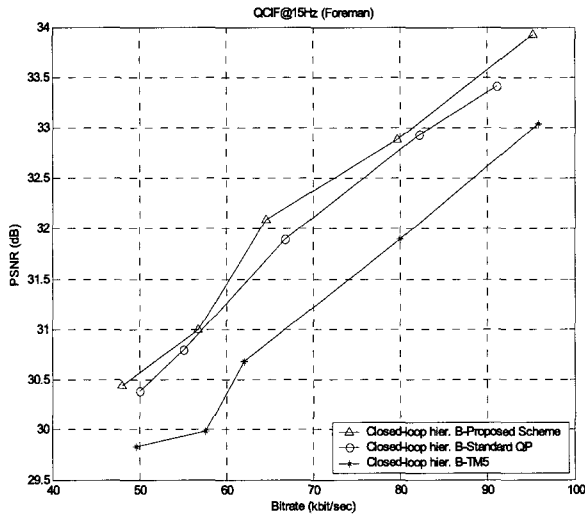


그림 10. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Foreman', QCIF, Closed-loop)

Fig 10. RD curve of conventional algorithm and proposed algorithm ('Foreman', QCIF, Closed-loop)

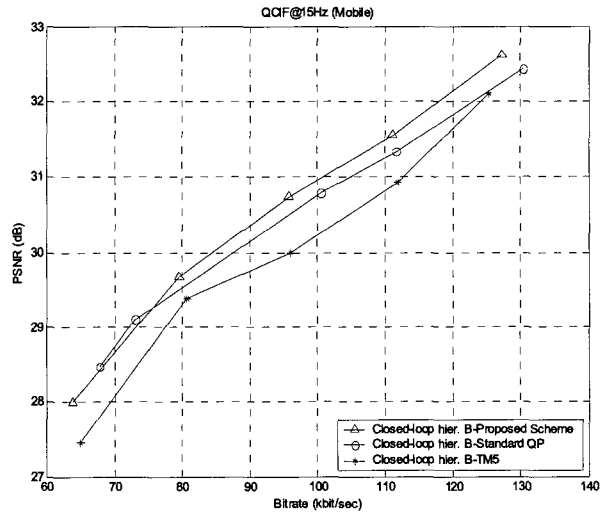


그림 11. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Mobile', QCIF, Closed-loop)

Fig 11. RD curve of conventional algorithm and proposed algorithm ('Mobile', QCIF, Closed-loop)

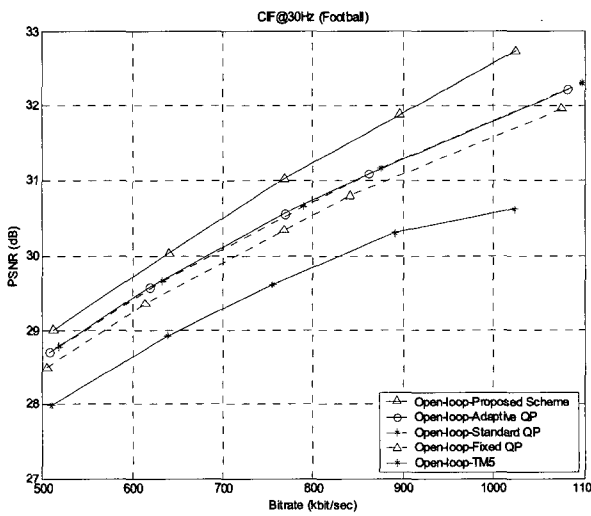


그림 12. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Football', CIF, Open-loop)

Fig 12. RD curve of conventional algorithm and proposed algorithm ('Football', CIF, Open-loop)

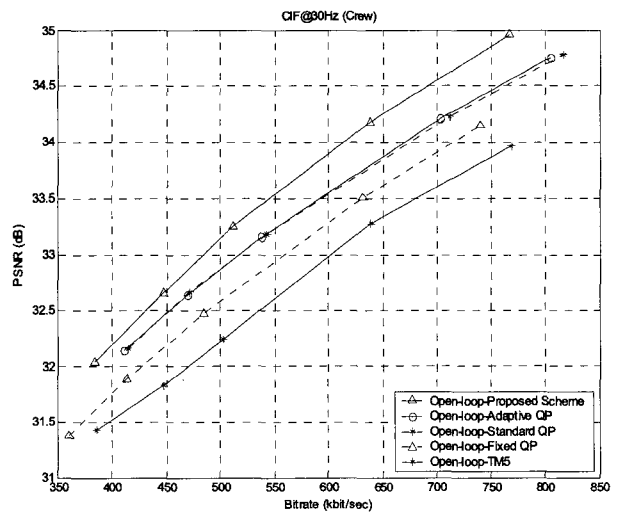


그림 13. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Crew', CIF, Open-loop)

Fig 13. RD curve of conventional algorithm and proposed algorithm ('Crew', CIF, Open-loop)

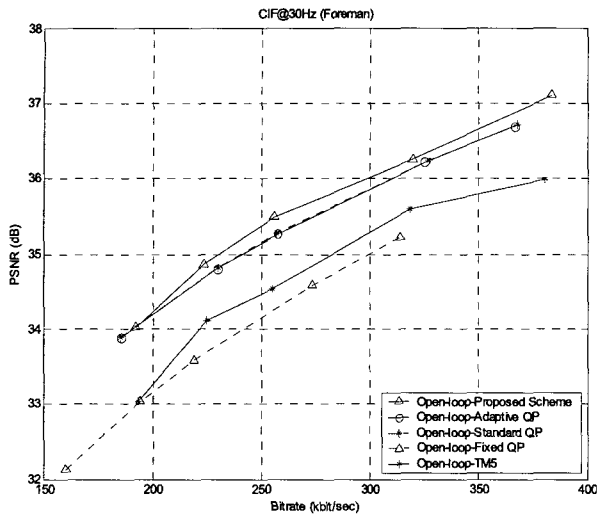


그림 14. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Foreman', CIF, Open-loop)
 Fig 14. RD curve of conventional algorithm and proposed algorithm ('Foreman', CIF, Open-loop)

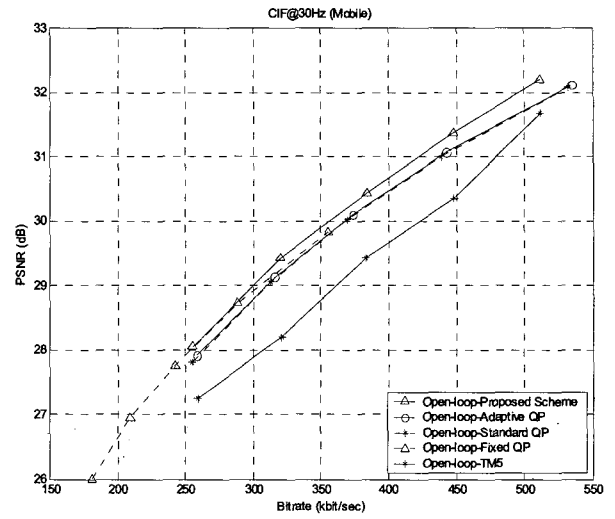


그림 15. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Mobile', CIF, Open-loop)
 Fig 15. RD curve of conventional algorithm and proposed algorithm ('Mobile', CIF, Open-loop)

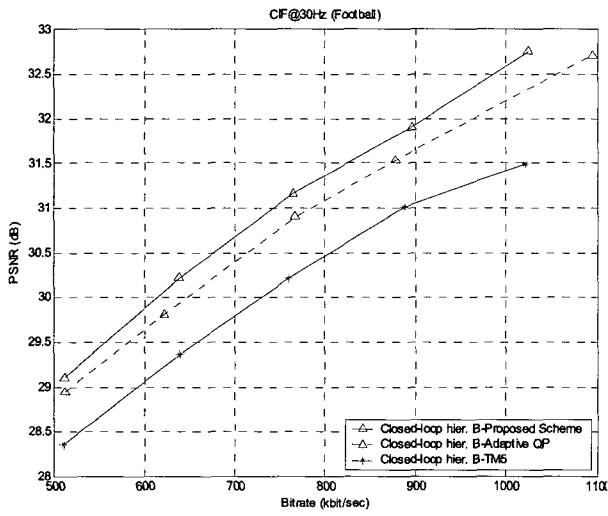


그림 16. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Football', CIF, Closed-loop)
 Fig 16. RD curve of conventional algorithm and proposed algorithm ('Football', CIF, Closed-loop)

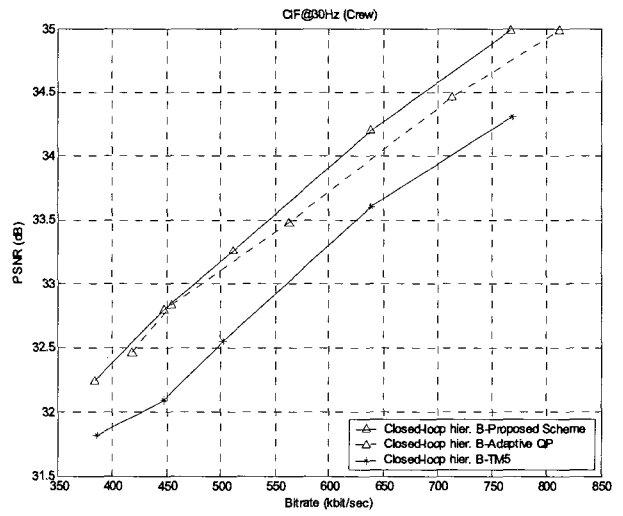


그림 17. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Crew', CIF, Closed-loop)
 Fig 17. RD curve of conventional algorithm and proposed algorithm ('Crew', CIF, Closed-loop)

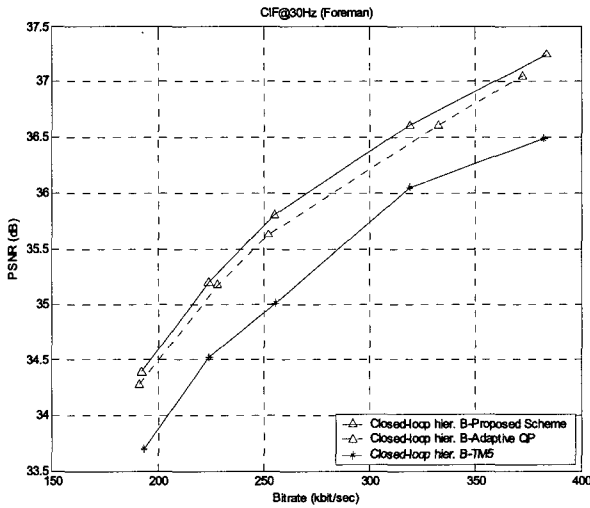


그림 18. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Foreman', CIF, Closed-loop)

Fig 18. RD curve of conventional algorithm and proposed algorithm ('Foreman', CIF, Closed-loop)

/ 32 / 31 / 29 / 27의 값으로 설정하였다. 마찬가지로 'Foreman' 영상의 타겟 비트율은 48 / 56 / 64 / 80 / 96 kbit/sec , 초기 양자화 파라미터는 39 / 38 / 36 / 34 / 33 이며, 'Mobile' 영상의 타겟비트율은 64 / 80 / 96 / 112 /

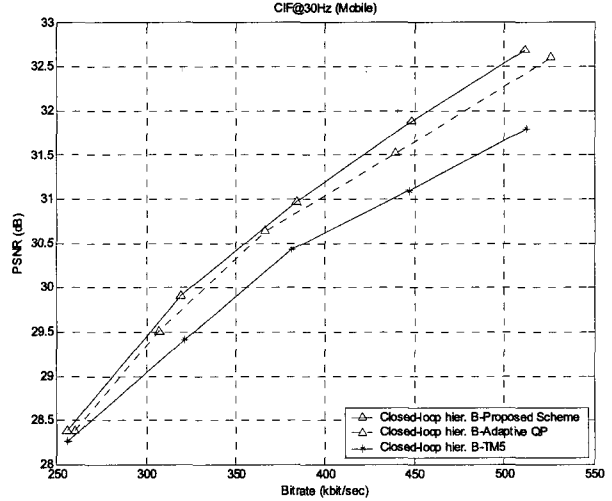


그림 19. 기존 알고리즘과 제안하는 알고리즘의 RD curve 비교 ('Mobile', CIF, Closed-loop)

Fig 19. RD curve of conventional algorithm and proposed algorithm ('Mobile', CIF, Closed-loop)

128 kbit/sec, 초기 양자화 파라미터는 43 / 42 / 39 / 38 / 36의 값으로 설정하였다. 그림 8-11은 동일한 영상에 대해 closed-loop 예측 구조에서의 RD curve를 나타낸다. 제안하는 알고리즘과의 성능 분석을 위해 'Standard QP' 모드,

표 3. 'Football', 'Crew' 영상에 대한 각 레이어의 QP

Table 3. QP of each layer for 'Football' and 'Crew' sequence

'Football'			'Crew'		
Layer0(QCIF@15Hz)	Layer1(QCIF@30Hz)	Bitrates(kbits/s)	Layer0(QCIF@15Hz)	Layer1(QCIF@30Hz)	Bitrates(kbits/s)
35	41	512	35	36	384
	39	640		35	448
	37	768		34	512
	36	896		32	640
	34	1024		31	768

표 4. 'Foreman', 'Mobile' 영상에 대한 각 레이어의 QP

Table 4. QP of each layer for 'Foreman' and 'Mobile' sequence

'Foreman'			'Mobile'		
Layer0(QCIF@15Hz)	Layer1(QCIF@30Hz)	Bitrates(kbits/s)	Layer0(QCIF@15Hz)	Layer1(QCIF@30Hz)	Bitrates(kbits/s)
35	37	192	35	49	256
	35	224		46	320
	34	256		44	384
	32	320		42	448
	31	384		40	512

TM5와 비교하였다. 그림 12-15는 동일한 영상에 대해 2 레이어로 부호화 한 후 복호화 한 결과이다. 제안하는 알고리즘은 'Fixed QP'와 'Standard QP', 'Adaptive QP' 세 가지 방법뿐만 아니라, TM5와 비교 실험하였다. 서로 다른 spatial layers에 대해 실험을 하기 위해서는 spatial base

layer의 QP를 고정시킨 후, spatial enhancement layer의 QP를 변화시켜 그 때의 발생 비트율과 PSNR을 구한다. 각 영상의 목표 비트율과 그에 따른 각 layer의 QP값들은 표 3, 4에서 나타내었다. 위의 실험과 마찬가지로 open-loop 예측 구조 에서뿐만 아니라 closed-loop 구조에서 성

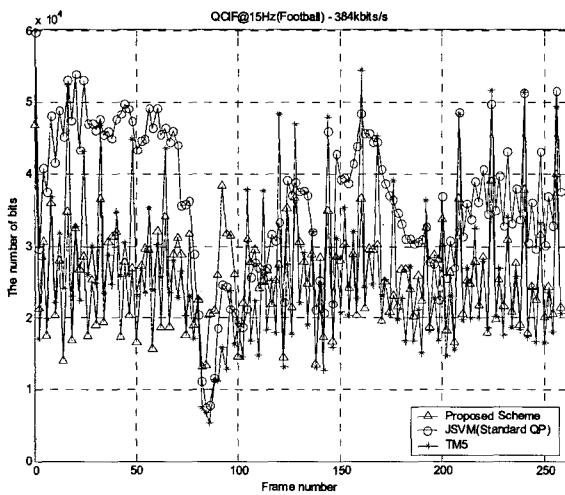


그림 20. 프레임 별 발생 비트량 비교 ('Football', 384kbts/s)
Fig 20. The number of generated bits according to the frame ('Football', 384kbts/s)

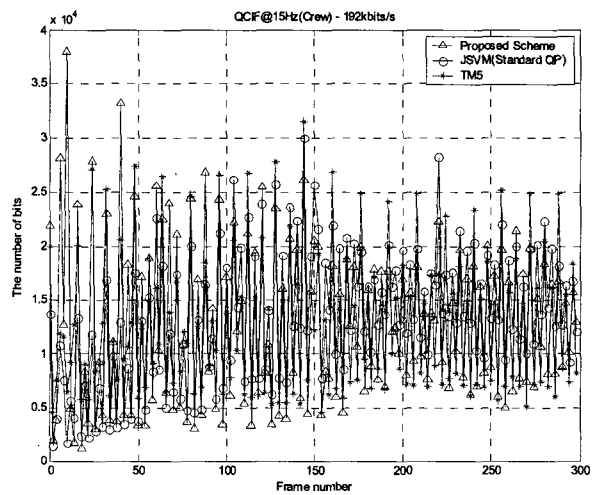


그림 21. 프레임 별 발생 비트량 비교 ('Crew', 192kbts/s)
Fig 21. The number of generated bits according to the frame ('Crew', 192kbts/s)

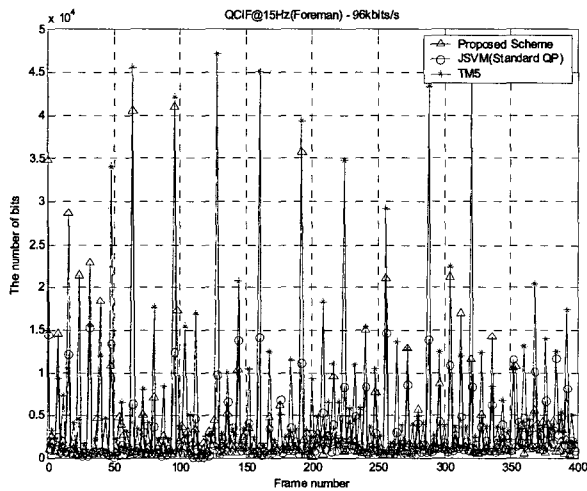


그림 22. 프레임 별 발생 비트량 비교 ('Foreman', 96kbts/s)
Fig 22. The number of generated bits according to the frame ('Foreman', 96kbts/s)

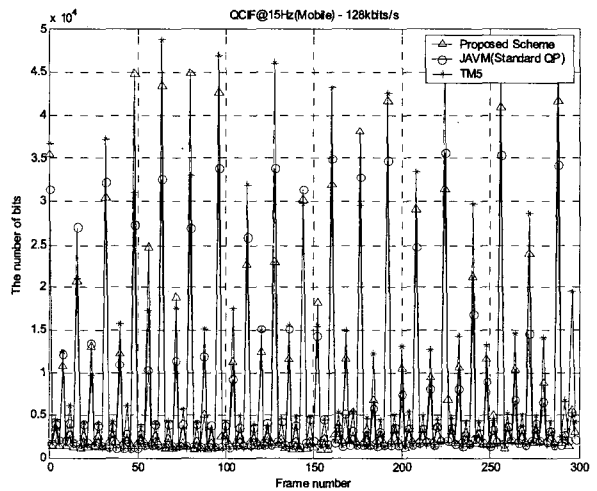


그림 23. 프레임 별 발생 비트량 비교 ('Mobile', 128kbts/s)
Fig 23. The number of generated bits according to the frame ('Mobile', 128kbts/s)

능 측정을 위한 실험을 수행하였하였으며, 실험 결과는 그림 16-19에서 나타내었다. 마지막으로, 각 영상의 프레임별 발생비트량은 그림 20-23과 같다. 결과에서 알 수 있듯이, 복잡도가 큰 프레임에 많은 비트가 할당되며, 복잡도가 작은 프레임에 적은 비트가 할당된다. 따라서, 제안하는 방법은 효율적인 비트할당 기법이 된다.

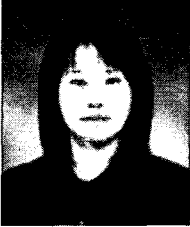
V. 결 론

제안하는 비트할당 알고리즘은 기존의 알고리즘과 비교하여 크게 두 가지의 특성을 갖는다. 첫째, 초기 QP에 의존하여 영상의 화질과 비트율이 결정되며, 주어진 영상들에 적절한 QP를 알아야 하는 기존 방법과 달리, 초기 QP 값에 대한 의존도를 낮추고 프레임의 복잡도에 따라 QP값 조절이 가능하도록 한다. 따라서 같은 비트율에서 더 나은 화질의 영상을 얻을 수 있다. 둘째, 기존의 비트율 조절 방법은 매크로블록에 대해 일정범위의 QP를 반복적으로 적용하는 반면, 제안하는 알고리즘은 매크로블록의 특성을 고려하여 각 매크로블록에 따라 적절한 QP를 적용하도록 한다. 제안하는 두 가지 방법을 사용함으로써, 주어진 영상 정보에 대해 효율적으로 비트할당이 이루어질 수 있다. 다양한 영상에 대한 실험을 통해 비트율 대비 PSNR 측면에서 제안하는 알고리즘이 기존 알고리즘보다 SVC 코덱의 성능을 더욱 향상시키는 것을 확인할 수 있다.

참 고 문 헌

- [1] J. Reichel, H. Schwarz and M. Wien, "JSVM 1.0 Software", Joint Video Team of ISO/IEC MPEG and ITU-T VCEG N6900, January 2005, Hongkong, China
- [2] J. Reichel, H. Schwarz and M. Wien, "JSVM 1.0 Reference Encoding Algorithm Description", Joint Video Team of ISO/IEC MPEG and ITU-T VCEG N6899, January 2005, Hongkong, China.
- [3] J. Reichel, H. Schwarz and M. Wien, "JSVM 2.0 Software", Joint Video Team of ISO/IEC MPEG and ITU-T VCEG N7085, April 2005, Busan, Korea.
- [4] J. Reichel, H. Schwarz and M. Wien, "JSVM 2.0 Reference Encoding Algorithm Description", Joint Video Team of ISO/IEC MPEG and ITU-T VCEG N7084, April 2005, Busan, Korea.
- [5] P. Chen and John W. Woods, "Bidirectional MC-EZBC With Lifting Implementation", IEEE Trans. Circuits and Systems. Vol.14, No.10, Oct.2004.
- [6] John W. Woods and P.Chen, "Improved MC_EZBC with Quarter-pixel Motion Vector", MPEG contribution, M8366, Fairfax.
- [7] S.J Choi and John W. Woods, "Motion Compensation 3D sub-band coding of video", IEEE Transaction on Image Processing, vol.8, no.2, pp153-167, Feb. 1999.
- [8] J. R. Ohm, "Temporal domain subband video coding with motion compensation", IEEE Transaction on Image Processing, vol.3, no.5, pp559-571, Sept. 1994.
- [9] I. Amonou, N. Cammas, S. Kervadec and S. Pateux, "Modification of the calculation of Scaling Factors in the JSVM", Joint Video Team of ISO/IEC MPEG and ITU-T VCEG JVT-O045, April 2005, Busan, Korea.
- [10] Test Model 5, ISO/IEC JTC1/SC29/WG11, N0400, Apr. 1993.
- [11] H. Schwarz, D. Marpe and T. Wiegand, "Comparison of MCTF and closed-loop hierarchical B picture", Joint Video Team of ISO/IEC MPEG and ITU-T VCEG JVT-P059, July 2005, Poznan, Poland.
- [12] J. Reichel, H. Schwarz and M. Wien, "JSVM 3.0 Reference Encoding Algorithm Description", Joint Video Team of ISO/IEC MPEG and ITU-T VCEG N7311, July 2005, Poznan, Poland.

 저 자 소 개

**박 내 리**

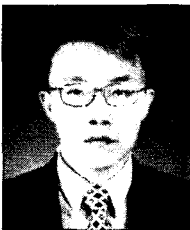
- 2005년 : 세종대학교, 정보통신공학과, 공학사
- 2005년~현재 : 세종대학교, 정보통신공학과, 석사과정
- 주관심분야 : 비디오 신호처리 및 압축, Scalable Video Coding

**전 동 산**

- 2002년 2월 : 부산대학교 전자공학과(공학사)
- 2004년 2월 : KAIST 전기 및 전자공학과(공학석사)
- 2004년 5월~현재 : ETRI 디지털방송연구단 방송미디어연구그룹 연구원
- 주관심분야 : 영상통신, 이미지 프로세싱, 패턴인식, Scalable Video Coding, TV-Anytime/MPEG-7/MPEG-21

**김 재 곤**

- 1990년 2월 : 경북대학교 전자공학과(공학사)
- 2005년 2월 : KAIST 전기 및 전자공학과(공학박사)
- 2001년 9월~2002년 11월 : 뉴욕 콜롬비아대학교 연구원
- 1992년 9월~현재 : ETRI 디지털방송연구단 방송미디어연구그룹 선임연구원/ 방통융합미디어연구팀장
- 주관심분야 : 영상통신, 비디오신호처리, 디지털방송, 멀티미디어 프레임워크, TV-Anytime/ MPEG-7/MPEG-21

**한 종 기**

- 1992년 : 한국과학기술원(KAIST), 전기 및 전자공학과, 공학사
- 1994년 : 한국과학기술원(KAIST), 전기 및 전자공학과, 공학 석사
- 1999년 : 한국과학기술원(KAIST), 전기 및 전자공학과, 공학 박사
- 1999년~2001년 삼성전자 디지털 미디어 연구소, 책임연구원
- 2001년~현재 : 세종대학교 정보통신공학과 조교수
- 주관심분야 : 비디오 신호처리 및 압축, Transcoding, Scalable Video Coding