

H.264/AVC 무손실 화면내 부호화를 위한 향상된 CAVLC 부호화 방법

허진^o, 호요성
광주과학기술원 정보통신공학과
{jinheo, hoyo}@gist.ac.kr

요약

H.264/AVC 비디오 압축 표준에서 엔트로피 부호화 방법의 하나인 문맥 기반 적응적 가변길이 부호화(context based adaptive variable length coding, CAVLC)는 주로 손실(lossy) 압축 부호화 환경에 적합하게 설계되었다. 또한 H.264/AVC 고급 프로파일(Advanced 4:4:4)에서 지원하는 무손실(lossless) 부호화의 경우 손실 압축 기반으로 만들어진 CAVLC를 그대로 사용하므로 최적의 압축 성능을 제공하지 못한다. 본 논문에서는 H.264/AVC의 손실 압축과 무손실 압축 환경에서 잔여 신호의 통계적 특성에 분명한 차이가 있음을 확인하고, 무손실 화면내 압축 환경에서 잔여 신호의 통계적 특성을 고려하여 향상된 CAVLC 기반의 잔여 신호 부호화 방법을 제안한다. 실험 결과로부터 본 논문에서 제안하는 향상된 CAVLC 부호화 방법이 무손실 화면내 부호화 압축의 경우 기존의 CAVLC 부호화 방법에 비해 평균적으로 약 7% 정도의 비트 수를 감소시키는 것을 확인했다.

1. 서론

H.264/AVC 비디오 부호화 표준은 국제전기 통신 연합(International Telecommunication Union, ITU) 산하의 ITU-T와 국제 표준화 기구(International Standard Organization) 산하의 MPEG(Moving Picture Expert Group) 두 기관의 공동 작업으로 최근에 완성되었다. H.264/AVC는 이전의 비디오 부호화 표준인 MPEG-2/4, H.263 등에 비해 높은 부호화 효율을 제공하는데, 이는 가변 블록 기반의 움직임 보상, 복수 참조 영상, 1/4 화소단위 부화소 움직임 예측, 문맥 적응적 엔트로피 부호화와 같은 기술들이 사용되기 때문이다 [1] [2].

H.264/AVC 비디오 부호화 표준은 손실(lossy) 압축 기능과 무손실(lossless) 압축 기능을 모두 제공한다. 하지만, H.264/AVC는 주로 손실 압축 기반으로 연구되어 왔기 때문에 기존의 무손실 압축 기반으로 연구된 JPEG-LS [3], Motion JPEG2000 lossless [4]에 구현된 무손실 압축 알고리즘에 비해 비교적 효율이 낮다. 따라서 H.264/AVC 기반의 무손실 압축 성능 향상에 대한 많은 연구가 필요하다. 현재 H.264/AVC에 구현된 무손실 화면내 부호화의 경우 원 영상과 예측된 영상의 차분인 잔여 영상 신호에 대한 변환(transform)과 양자화(quantization) 과정을 수행하지 않고 엔트로피 부호화 과정만 수행한다.

따라서, 무손실 압축의 잔여 영상 신호와 손실 압축의 잔여 영상 신호 사이에는 명확한 통계적 분포의 차이가 있다. 본 논문에서는 무손실 압축 환

경에서 잔여 신호의 통계적 특성을 고려하여 이를 효율적으로 부호화하는 향상된 CAVLC (context based adaptive variable length coding)를 설계하는 방법을 제안한다. 표 1은 H.264/AVC 부호화 표준에서 매크로블록 부호화에 사용되는 구문을 나타낸다. 표 1에서 회색으로 처리된 구문들은 잔여 신호를 부호화하기 위해 사용되는 구문이다 [5]. 따라서, 본 논문에서는 회색으로 처리된 구문들에 대한 부호화 과정과 방법을 새롭게 제안했다.

표 1. 매크로블록 부호화에 사용되는 구문

<i>mb_type</i>
<i>prev_intra4x4_pred_mode_flag</i>
<i>rem_intra4x4_pred_mode_flag</i>
<i>intra_chrom_pred_mode</i>
<i>coded_block_pattern</i>
<i>mb_qp_delta</i>
<i>coeff_token</i>
<i>trailing_ones_sign_flag</i>
<i>level_prefix</i>
<i>level_suffix</i>
<i>total_zeros</i>
<i>run_before</i>

2. CAVLC 에서의 잔여 영상 부호화

CAVLC는 손실 부호화에서 다음과 같은 잔여 데이터의 특성을 고려하여 설계되었다:

- 1) 변환과 양자화 과정 후, 하위 블록 (4×4)은 일반적으로 많은 0을 포함한다.
- 2) 고주파 영역의 0이 아닌 계수들은 1과 같은 작은 값을 갖는다.
- 3) 0이 아닌 계수들은 저주파 영역으로 갈수록 값이 커지는 경향이 있다.

이와 같은 특성을 고려하여 CAVLC는 잔여 데이터를 효율적으로 부호화하기 위해 *coeff_token*, *trailing_ones_sign_flag*, *level_prefix*, *level_suffix*, *total_zeros*, *run_before*와 같은 구분들을 사용한다. 그림 1은 매크로블록 내 하위 블록 (4×4)에 대한 잔여 데이터 부호화 과정을 나타낸다 [6] [7] [8].



그림 1. CAVLC의 잔여 데이터 부호화 구조

처음 단계에서는 하위 블록 내에 존재하는 0이 아닌 계수의 수(*numcoeff*)와 가장 고주파 영역에 존재하는 절대값이 1인 계수의 수(*numtrailingones*)를 부호화한다. 여기서 *numtrailingones*는 3을 넘을 수 없다. *numcoeff*와 *numtrailingones*는 4개의 VLC 표 (3개의 VLC(variable length coding) 표와 1개의 FLC(fixed length coding) 표)를 사용하여 부호화한다. 표 2는 4개의 VLC 표를 나타낸다.

표 2. VLC 표 선택

예측된 0이 아닌 계수의 수	VLC 표
0, 1	<i>Num-VLC0</i>
2, 3	<i>Num-VLC1</i>
4, 5, 6, 7	<i>Num-VLC2</i>
8 이상	<i>FLC</i>

VLC 표는 이전에 부호화된 이웃하는 하위 블록

들; 왼쪽 블록과 위쪽 블록의 0이 아닌 계수를 수에 따라 선택된다. 표 2를 이용하여 이웃하는 블록들로부터 예측된 0이 아닌 계수의 수를 기반으로 현재 블록에 대한 적합한 VLC 표를 선택한다.

Trailing One Sign Flag는 처음 단계에서 발생한 trailing one에 대한 부호를 각각 한 비트를 사용하여 부호화한다. 양의 부호와 음의 부호는 각각 '0'과 '1'로 부호화된다.

레벨 부호화 과정에서는 trailing ones와 0 값들을 제외한 모든 계수들에 대한 절대값과 부호를 부호화한다. 이때, 계수들의 부호화는 지그재그 스캔된 순서의 역으로 진행된다. 각 레벨의 절대값(*abs_level*)은 7개의 VLC 표를 사용하여 부호화된다. VLC 표는 이전에 부호화된 레벨의 크기에 따라 선택된다. VLC 표 선택은 다음과 같다.

- 1) 만약 ($numcoeff > 10$ && $numtrailingones == 3$), *Lev-VLC1*로 초기화
이 조건을 만족하지 않으면,
*Lev-VLC0*로 초기화
- 2) 선택된 VLC표를 이용하여 레벨의 절대값 부호화
- 3) 레벨의 부호를 부호화
- 4) 만약 현재 부호화한 계수의 크기가 표 3에 정의된 문턱값(threshold)보다 크면, VLC표 증가

표 3. VLC표 증가를 위한 문턱값

<i>abs_level</i> 을 위한 <i>Lev-VLC</i> 표	VLC 표 증가를 위한 문턱값
<i>Lev-VLC0</i>	0
<i>Lev-VLC1</i>	3
<i>Lev-VLC2</i>	6
<i>Lev-VLC3</i>	12
<i>Lev-VLC4</i>	24
<i>Lev-VLC5</i>	48
<i>Lev-VLC6</i>	> 48

마지막으로 0의 계수 부호화 과정은 두 과정으로 구분된다. 우선 지그재그 스캔된 순서 상 가장 마지막에 위치한 0이 아닌 계수의 앞에 존재하는 모든 0의 수를 부호화한다. 다음으로 각 0의 위치를 부호화한다. 이때, 0의 위치 부호화 과정도 레벨 부호화 과정과 마찬가지로 지그재그 스캔된 순서의 역으로 진행된다.

3. 제안한 방법

3.1 무손실 환경에서 잔여 신호의 통계적 특성

무손실 부호화에서 잔여 데이터는 양자화 변환된 계수가 아니고 원 영상과 예측된 영상의 차분 값을 갖는 계수이다. 따라서 손실 부호화와 무손실

부호화의 잔여 데이터 사이에는 통계적 특성 차이가 있다. 무손실 부호화에서 잔여 데이터의 통계적 특성은 다음과 같다.

우선 0이 아닌 계수의 존재 확률 분포가 스캔 위치에 따라 독립적이고 (그림 2) 0이 아닌 계수의 수는 일반적으로 손실 부호화에서 0이 아닌 계수의 수보다 많다 (표 5). 또한, 0이 아닌 계수의 절대값은 스캔 위치가 커져도 (고주파 쪽으로 이동) 작아지지 않는다. 즉, 계수의 절대값은 스캔 위치에 독립적이다. 마지막으로 trailing one의 발생 확률이 손실 부호화에 비해 높지 않다 (표 4).

그림 2는 스캔 위치에 따른 0이 아닌 계수의 존재 확률 분포를 나타낸다. 예상했던 대로, 손실 부호화와 무손실 부호화의 잔여 데이터 사이에는 큰 통계적 특성 차이가 있음을 확인할 수 있다.

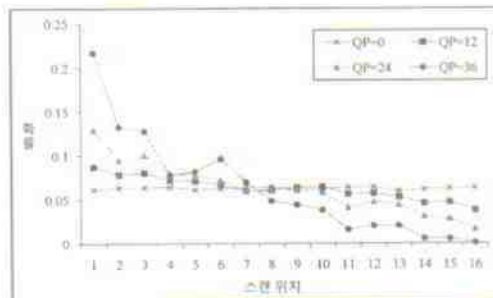


그림 2. 지그재그 스캔 위치와 계수 존재 확률 ('Paris', CIF)

표 4는 양자화 파라미터(QP)에 따른 trailing one의 발생 확률 분포를 나타낸다. 무손실 부호화의 경우 trailing one의 발생 확률이 손실 부호화에서 trailing one의 발생 확률에 비해 상대적으로 낮음을 확인할 수 있다.

표 4. trailing one의 발생 확률 분포

영상 \ QP	0	12	24	36
Foreman	0.258	0.760	0.903	0.958
Container	0.337	0.800	0.901	0.943
News	0.372	0.816	0.882	0.946
Silent	0.228	0.817	0.923	0.954
Mobile	0.222	0.636	0.837	0.929
Tempete	0.227	0.784	0.886	0.945
Paris	0.275	0.747	0.866	0.936
Football	0.142	0.787	0.904	0.952

위에서 언급한 무손실 부호화에서 잔여 데이터의 통계적 특성을 고려하여 CAVLC의 대응하는 부호화 부분을 변경함으로써, 무손실 화면내 부호화를 위한 향상된 CAVLC 부호화 방법을 제안한다.

그림 3은 이 논문에서 제안하는 잔여 데이터 부호화 구조를 나타낸다. 제안하는 방법의 부호화 과정은 다음과 같다. 처음 단계에서는 하위 블록 내에 존재하는 0이 아닌 계수의 수(numcoeff)를 부호화한다. 레벨 부호화 과정은 새로운 문턱값(threshold value)을 갖는 적응적인 VLC 표를 사용하여 각 계수들에 대한 절대값과 부호를 부호화한다. 마지막으로 0의 계수 부호화 과정은 CAVLC의 0의 계수 부호화 과정과 동일하게 가장 마지막에 위치한 0이 아닌 계수의 앞에 존재하는 모든 0의 수를 부호화한 후, 각 0의 위치를 부호화한다.

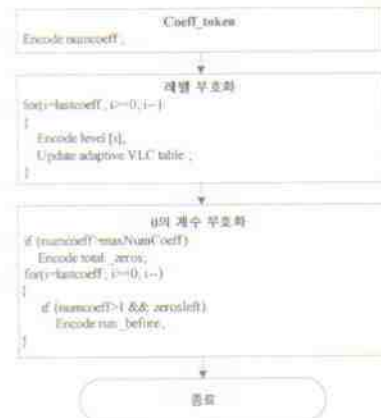


그림 3. 제안하는 잔여 데이터 부호화 구조

3.2 0이 아닌 계수의 수 부호화

무손실 부호화에서는 하위 블록 (4x4) 내에 0이 아닌 계수들의 수(numcoeff)가 많기 때문에 (표 5), 대부분의 경우 FLC 표가 선택된다. 무손실 화면내 부호화 환경에서 여러 실험 영상을 실험한 결과, FLC 표가 약 95% 정도 선택됨을 확인했다. 따라서 여기서 제안하는 방법은 3개의 VLC 표를 제거한 후, FLC 표만 사용하여 부호화를 수행한다. 또한 numtrailingones를 부호화하지 않고, numcoeff만 부호화한다.

표 5. 한 하위 블록의 0이 아닌 계수들의 평균 수

영상 \ QP	0	12	24	36
Foreman	13.92	8.10	3.49	1.09
Container	13.38	7.49	3.56	1.70
News	12.58	6.31	3.36	1.54
Silent	14.09	8.25	3.34	0.91
Mobile	14.19	10.28	6.31	2.53
Tempete	13.97	9.08	4.91	1.69
Paris	13.79	7.98	4.11	1.67
Football	15.06	9.38	4.63	1.21

기존의 FLC 표는 색차 DC를 제외하고 *numcoeff*에 4비트를 *numtrailingones*에 2비트를 할당하였다. 하지만 여기서 제안한 방법에서는 *numtrailingones*를 고려하지 않으므로 *numcoeff*를 단지 4비트만을 사용하여 부호화한다. 기존의 FLC 표의 경우에는 모든 *numcoeff*에 동일하게 4비트를 할당했지만, 여기서 제안하는 방법은 무손실 부호화의 통계적 특성을 고려하여 효율적으로 *numcoeff*를 부호화하기 위해 새로운 VLC 표를 설계한다.

우리가 제안한 VLC 표에서, *numcoeff*가 1-12번째 위치에 존재하면 4비트를, 13-16번째 위치에 존재하면 2비트를 할당한다. 복호기에서 모호함을 제거하기 위해 각 코드워드의 처음에 1비트의 체크 비트를 삽입한다. 최종적으로 *numcoeff*가 1-12번째 위치에 존재하면 5비트를, 13-16번째 위치에 존재하면 3비트를 할당한다. 표 6은 색차 DC를 제외한 *numcoeff*에 대한 코드워드를 나타낸다.

표 6. 색차 DC를 제외한 *numcoeff*의 코드워드

<i>numcoeff</i>	코드워드	
	체크 비트	비트
0	1	1111
1	1	0000
2	1	0001
3	1	0010
4	1	0011
5	1	0100
6	1	0101
7	1	0110
8	1	0111
9	1	1000
10	1	1001
11	1	1010
12	1	1011
13	0	00
14	0	01
15	0	10
16	0	11

색차 DC의 경우, CAVLC는 여러 칼라 포맷 (4:2:0, 4:2:2, 4:4:4)에 따라 적합한 VLC 표를 선택하여 색차 DC 잔여 데이터를 부호화한다. 제안한 방법은 무손실 부호화에서 색차 DC 데이터의 특성을 고려하여 새로운 VLC 표를 설계한다.

4:2:0 칼라 포맷에서 DC 데이터는 최대 4개까지 계수들을 가질 수 있다. 따라서 *numcoeff*에 각각 2비트를 할당하여 부호화를 수행한다. 또한 *numcoeff* > 0 과 *numcoeff* = 0을 구분하는 유일한 코드워드를 할당하기 위해 각 코드워드의 처음에 1비트의 체크 비트를 삽입한다. 표 7은 색차 DC의 *numcoeff*에 대한 코드워드를 나타낸다.

표 7. 색차 DC의 *numcoeff*에 대한 코드워드

<i>numcoeff</i>	코드워드	
	체크 비트	비트
0	1	x
1	0	00
2	0	01
3	0	10
4	0	11

3.3 레벨 부호화

레벨 부호화 과정에서, 각 0이 아닌 계수의 절대값 (*abs_level*)은 7개의 VLC 표 중 (*Lev-VLC0* ~ *Lev-VLC6*)에서 적합한 하나의 VLC 표를 선택하여 지그재그 스캔된 순서의 역으로 부호화를 수행한다. 각 VLC 표는 특정한 범위 내에서 효율적으로 *abs_level*을 부호화하도록 설계되었다. CAVLC에서 레벨 부호화를 위한 VLC 표는 *abs_level*이 고주파 영역에서 저주파 영역으로 갈수록 점점 증가한다는 사실을 기반으로 선택된다. 따라서 VLC 표 선택은 이전에 부호화된 *abs_level*에 따라 단조 증가 (*monotonically increasing*)한다.

그러나, 무손실 부호화에서 *abs_level*은 그림 4에서 보듯이 스캔 위치에 독립적이다. 따라서, 이전에 부호화된 *abs_level*에 따라 증가뿐 아니라 감소할 수 있는 적응적인 VLC 표 선택 방법을 설계한다.

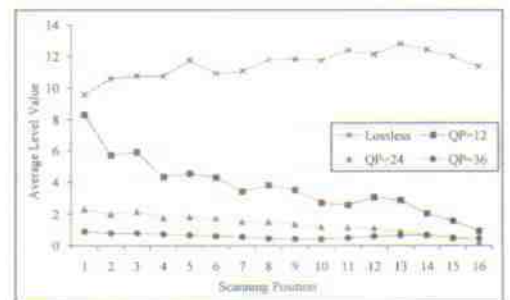


그림 4. 스캔 위치에 따른 계수의 평균 절대값 분포 ('Tempete', CIF)

손실 부호화의 레벨 부호화에서 CAVLC는 다음에 부호화할 *abs_level*이 일반적으로 현재 부호화한 *abs_level*보다 크다는 가정을 기반으로 선택 가능한 VLC 표 중에서 가장 작은 VLC 표를 선택하였다. 그러나, 무손실 부호화의 레벨 부호화에서 다음에 부호화할 *abs_level*은 일반적으로 현재 부호화한 *abs_level*보다 크다고 가정할 수 없다. 이론적으로, 무손실 부호화에서 각 *abs_level*에 대한 최적의 문턱값은 *abs_level*을 위한 부호화 범위 값의 평균값으로 계산된다. 왜냐하면, 다음에 부호화될 *abs_level*이 증가할지 혹은 감소할지 예측할 수 없

기 때문이다.

표 8은 각 Lev-VLC 표에서 abs_level을 효율적으로 부호화하기 위한 최적의 부호화 범위와 평균 문턱값을 나타낸다. 이를 기반으로 새로운 문턱값을 나타낸다. 제안한 방법에서 새로운 문턱값에 따라 VLC 표는 증가할 수도 감소할 수도 있다.

표 8. Lev-VLC 표에 따른 abs_level의 최적의 부호화 범위와 평균 문턱값

VLC 표	abs_level을 위한 최적의 부호화 범위	평균 문턱값	새로운 문턱값
Lev-VLC0	1	1	1
Lev-VLC1	1, 2	1.5	2
Lev-VLC2	1 - 5	3	3
Lev-VLC3	2 - 11	6.5	7
Lev-VLC4	4 - 23	13.5	14
Lev-VLC5	8 - 47	27.5	28
Lev-VLC6	> 15	-	-

그림 4에서 보듯이, abs_level은 손실 부호화와 무손실 부호화에서 큰 차이가 있다. 손실 부호화에서, 마지막 스캔 위치의 abs_level은 일반적으로 작은 값을 갖기 때문에, Lev-VLC0 혹은 Lev-VLC1의 VLC 표를 가지고 레벨 부호화를 시작한다. 그러나 무손실 부호화에서는 마지막 스캔 위치의 abs_level이 Lev-VLC0 혹은 Lev-VLC1의 VLC 표를 사용할 만큼 충분히 작지 않다. 무손실 화면내 부호화 환경에서 여러 실험 영상을 실험한 결과, 마지막 스캔 위치의 평균 abs_level은 Chroma DC 하위 블록과 다른 하위 블록들에서 각각 약 6.79와 10.70임을 확인했다. 따라서, 두 경우에 대해 각각 Lev-VLC3과 Lev-VLC4로 VLC 표를 초기화한다.

4. 실험 결과 및 분석

본 논문에서는 H.264/AVC 표준 소프트웨어 JM 13.2 [9]를 이용하여 실험했다. 제안한 알고리즘의 효율성을 보이고자, 4개의 QCIF(176×144) 해상도의 4:2:0 칼라 포맷 비디오 영상들(Foreman, Container, News, Silent)과 4개의 CIF(352×288) 해상도의 4:2:0 칼라 포맷 비디오 영상들(Mobile, Tempete, Paris, Football)을 이용하여 인트라 환경에서 부호화했다. 모든 비디오 영상들은 150 화면을 부호화했다. 자세한 실험 환경은 표 9에 나타났다.

표 9에서 'QPPrimeYZeroTransformBypassFlag'는 변환과 양자화 과정을 생략하고 곧바로 엔트로피 부호화 과정을 통해 잔여 영상 신호를 부호화하고자 하는 무손실 부호화를 위한 설정 변수이다. 단, 위 설정 변수는 양자화 파라미터 값을 0(무손실)으로 사용했을 경우에만 유효하다.

표 9. 실험 환경

실험 변수	값
ProfileIDC	244
IntraPeriod	1
QPISlice	0
SymbolMode	0
QPPrimeYZeroTransformBypassFlag	1

제안한 방법과 H.264/AVC 참조 소프트웨어 방법의 성능 비교를 위해 압축율과 비트수 이득을 각각 식 (1)과 식 (2)를 사용하여 비교했다.

$$\text{압축율} = \frac{\text{비트수}_{\text{원영상}}}{\text{비트수}_{\text{제안한방법}}} \quad (1)$$

$$\text{비트수이득} = \frac{\text{비트수}_{\text{H.264/AVC}} - \text{비트수}_{\text{제안한방법}}}{\text{비트수}_{\text{H.264/AVC}}} \times 100 \quad (2)$$

표 10은 성능 비교를 위해 실험한 결과를 나타낸다. 또한, 제안한 방법을 각각의 효율성을 보이고자 다음과 같이 각 방법에 대한 성능을 비교했다.

- 1) 새로운 VLC 표 설계 방법: 제안한 coeff_token 부호화 + trailing_one_sign_flag 제거
- 2) 제안한 최종 방법: 새로운 VLC 표 설계 방법 + 제안한 레벨 부호화 방법

표 10의 실험 결과로부터, 제안한 방법이 기존의 H.264 방법과 비교하여 QCIF 영상의 경우 약 5.9% 정도, CIF 영상의 경우 약 8.6% 정도의 비트 수가 감소함을 확인할 수 있다. 최종적으로 제안한 방법이 기존의 CAVLC 부호화 방법에 비해 평균적으로 약 7% 정도의 비트 수를 감소시키는 것을 알 수 있다.

5. 결론

본 논문에서는 H.264/AVC에서 손실 압축과 무손실 압축 환경에서 잔여 신호의 통계적 특성 차이를 분석하고, 이를 기반으로 무손실 압축 환경에 적합한 향상된 CAVLC (context based adaptive variable length coding) 기반의 잔여 신호 부호화 방법을 제안했다. 제안한 방법에서는 0이 아닌 계수들의 효율적인 부호화를 위해 새로운 VLC (variable length coding) 표를 설계했고, trailing ones의 부호를 부호화하는 부분을 제거했다. 마지막으로, 레벨 부호화를 위한 새로운 문턱값 (threshold value)을 갖는 적응적인 VLC 표를 사용했다. 실험 결과로부터 본 논문에서 제안하는 향상된 CAVLC 부호화 방법이 무손실 부호화 압축의 경우 현재 H.264/AVC에 사용되고 있는 CAVLC에 비해 평균적으로 약 7.26% 정도의 비트 수를 감소시키는 것을 확인했다.

표 10. 제안한 알고리즘의 성능 평가

영상	원 영상 크기 (bits)	방법	전체 비트수 (bits)	압축율	비트수 이득 (%)
Foreman	45619200	H.264 CAVLC	21119320	2.1601	0
		새로운 VLC 표 설계 방법	20502304	2.2251	2.922
		제안한 최종 방법	19897784	2.2927	5.784
Container	45619200	H.264 CAVLC	20201952	2.2582	0
		새로운 VLC 표 설계 방법	19659944	2.3204	2.683
		제안한 최종 방법	19224360	2.3730	4.839
News	45619200	H.264 CAVLC	19929400	2.2890	0
		새로운 VLC 표 설계 방법	19433624	2.3474	2.488
		제안한 최종 방법	18736216	2.4348	5.987
Silent	45619200	H.264 CAVLC	22109864	2.0633	0
		새로운 VLC 표 설계 방법	21452920	2.1265	2.971
		제안한 최종 방법	20561096	2.2187	7.005
Mobile	182476800	H.264 CAVLC	110168736	1.6563	0
		새로운 VLC 표 설계 방법	107456176	1.6982	2.462
		제안한 최종 방법	98218696	1.8579	10.847
Tempete	182476800	H.264 CAVLC	97856360	1.8647	0
		새로운 VLC 표 설계 방법	95257568	1.9156	2.656
		제안한 최종 방법	89482304	2.0393	8.557
Paris	182476800	H.264 CAVLC	89998496	2.0276	0
		새로운 VLC 표 설계 방법	87570432	2.0838	2.698
		제안한 최종 방법	83759728	2.1786	6.932
Football	182476800	H.264 CAVLC	101325088	1.8009	0
		새로운 VLC 표 설계 방법	98336912	1.8556	2.949
		제안한 최종 방법	93056088	1.9609	8.161
평균		H.264 CAVLC		2.0150	0
		새로운 VLC 표 설계 방법		2.0716	2.729
		제안한 최종 방법		2.1695	7.264

감사의 글

본 연구는 광주과학기술원 실감방송연구센터를 통한 대학IT연구센터(ITRC)의 지원에 의한 것입니다.

참고문헌

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. on Circuits and System for Video Technology, vol. 13, no. 7, pp. 560-576, July 2003.

[2] G. J. Sullivan and T. Wiegand, "Video compression-from concepts to the H.264/AVC standard," Proceedings of the IEEE, no. 1, pp. 18-31, Jan. 2005.

[3] M. J. Weinberger, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," IEEE Trans. on image processing, vol. 9, no. 8, pp. 1309-1324 Aug. 2000.

[4] Motion JPEG2000 Final Committee Draft, ISO/IEC JTC 1/SC29/WG 1, Document N2117, March 2001.

[5] Information Technology-Coding of audio-visual objects-part 10: Advanced Video Coding, International Standard, ISO/IEC 14496-10, Second edition, Dec. 2005.

[6] 호요성, 김승환, H.264/AVC 알고리즘 이해와 프로그램 분석, 두양사, 2008.

[7] L. E. G. Richardson, H.264 and MPEG-4 Video Compression, WILEY, 2003.

[8] G. Bjontegaard and K. Lillevold, Context-adaptive VLC (CVLC) coding of coefficients, ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16, Joint Video Team Document JVT-C028, May 2002.

[9] http://iphome/hhi.de/shehring/tml/download/old_jm/jm13.2.zip, Joint Video Team, Reference Software Version 13.2