

# H.264/AVC의 향상된 CAVLC 부호화 방법을 이용한 무손실 위성 영상 압축

허진, 호요성

광주과학기술원 정보기전공학부

{jinheo, hoyo}@gist.ac.kr

## Lossless Satellite Image Compression using Improved CAVLC of H.264/AVC

Jin Heo and Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)

### 요약

본 논문에서는 무손실 위성 영상 압축의 효율을 높이기 위해 H.264/AVC의 무손실 화면내 부호화에 기반한 새로운 문맥 기반 적응적 가변 길이 부호화 (CAVLC) 방법을 제안한다. 기존의 H.264/AVC의 CAVLC는 손실 압축에 적합하게 설계되었기 때문에, 변환과 양자화 과정을 수행하지 않는 무손실 압축에서 최적의 압축 성능을 제공하지 못한다. 본 논문에서는 손실 압축과 무손실 압축 환경에서 잔여 영상 신호의 통계적 차이가 있음을 확인하고, 무손실 위성 영상 압축 환경에서 잔여 신호의 통계적 특성을 고려하여 향상된 CAVLC 기반의 무손실 위성 영상 압축 방법을 제안한다. 제안한 방법을 사용하여 위성 영상을 압축한 결과 기존의 대표적인 무손실 영상 압축 방법인 JPEG-LS와 CALIC 보다 약 12% 정도 비트 수를 감소시켰다.

### 1. 서론

최근 센서 및 전자·통신 기술의 발달로 위성 영상의 해상도가 높아짐에 따라, 다양한 응용 분야에서 위성 영상이 사용되고 있다. 일반적으로 위성 영상은 정확성과 정밀도가 보장되어야 하므로, 손실 압축을 적용할 수 없다. 따라서, 위성 영상은 부호화/복호화 과정을 거치더라도 원 영상과 복원 영상이 완전히 일치하여 품질 및 신뢰도를 보장할 수 있는 무손실 압축 방법을 사용한다.

무손실 정지 영상 압축에 사용되는 JPEG-LS(lossless joint photographic experts group)와 CALIC(context-based adaptive lossless/nearly lossless image coding)은 손실 압축의 높은 압축률에 비해, 그 압축 효율이 약 1.5:1 정도에 불과하다. 따라서 고해상도, 고화질의 위성 영상의 경우 데이터의 양이 크기 때문에 효율적인 무손실 압축이 필수적이다.

현재 H.264/AVC [1]에 구현된 무손실 화면내 부호화는 원 영상과 예측된 영상의 차분인 잔여 영상 신호에 변환과 양자화 과정을 수행하지 않고, 엔트로피 부호화만 수행한다. 따라서, 무손실 압축의 잔여 영상 신호와 손실 압축의 잔여 영상 신호 사이에는 명확한 통계적 분포 차이가 있다. 본 논문에서는 무손실 압축 환경에서 잔여 신호의 통계적 특성을 고려하여 향상된 CAVLC(context-based adaptive variable length coding) [2] 기반의 효율적인 무손실 위성 압축 방법을 제안한다.

### 2. H.264/AVC 정지 영상 압축 방법

그림 1은 무손실 위성 영상을 압축하기 위해 H.264/AVC를 사용한 무손실 정지 영상 압축의 구조를 나타낸다. 그림 1에서 보듯이, 먼저 화면내 (인트라) 예측을 사용하여 예측 영상을 만든 후, 원 영상과 예측 영상의 차분인 잔여 영상 신호를 엔트로피 부호화를 사용하여 압축한다.

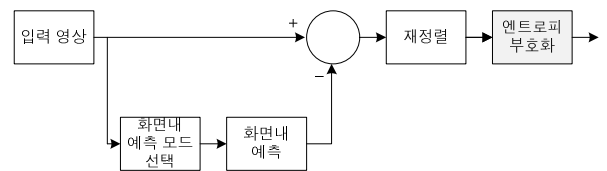


그림 1. H.264/AVC를 사용한 무손실 정지 영상 압축 구조

H.264/AVC는 블록 간의 화소 상관도를 고려하여 화면내 예측 부호화를 수행한다. 이미 부호화된 화소의 정보를 이용하여 블록마다 적절한 예측 방향을 선택해서 부호화하기 때문에 부호화 효율을 향상시킬 수 있다. 그림 2와 그림 3은 각각 4×4 화면내 예측에서 사용하는 9가지 예측모드와 16×16 화면내 예측에서 사용하는 4가지 예측모드를 나타낸다.

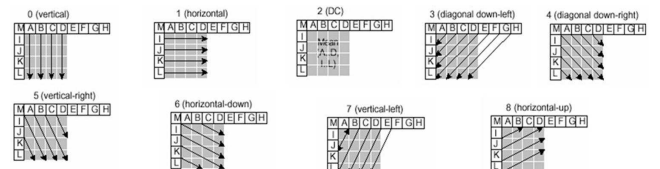


그림 2. 4×4 화면내 예측의 9가지 예측모드

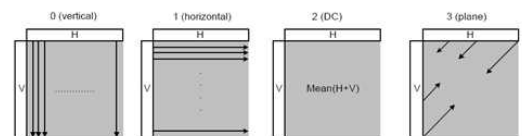


그림 3. 16×16 화면내 예측의 4가지 예측모드

화면내 예측을 수행한 후, 생성된 잔여 데이터는 마지막으로 엔트로피 부호화(CAVLC)를 수행하여 부호화한다. H.264/AVC의 CAVLC

는 잔여 데이터를 효율적으로 부호화하기 위해 그림 4와 같은 잔여 데이터 부호화 과정을 수행한다.

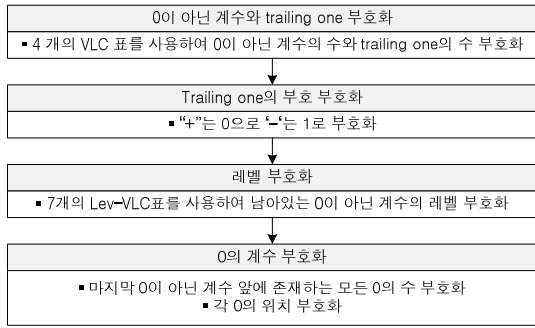


그림 3. 16×16 화면내 예측의 4가지 예측모드

처음 단계는 하위 블록 내에 존재하는 0이 아닌 계수의 수와 가장 고주파 영역에 존재하는 절대값이 1인 계수의 수를 부호화한다. 여기서 가장 고주파 영역에 존재하는 절대값이 1인 계수의 수는 3을 넘을 수 없다. 여기서는 4개의 VLC 표 (3개의 VLC(variable length coding) 표와 1개의 FLC(fixed length coding) 표)를 사용하여 부호화한다.

두 번째 단계는 처음 단계에서 발생한 가장 고주파 영역에 존재하는 절대값이 1인 계수에 대한 부호를 각각 한 비트를 사용하여 부호화한다. 양의 부호와 음의 부호는 각각 '0'과 '1'로 부호화된다.

레벨 부호화 과정에서는 가장 고주파 영역에 존재하는 절대값이 1인 계수와 0 값을 제외한 모든 계수에 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.

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. 향상된 CAVLC 부호화 방법

#### 가. 무손실 환경에서 잔여 신호의 통계적 특성

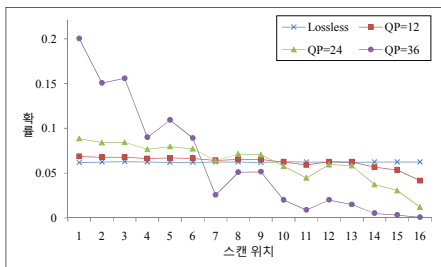


그림 5. 스캔 위치에 따른 0이 아닌 계수의 존재 확률

손실 압축과 무손실 압축의 잔여 데이터의 통계적 분포 차이는 변환과 양자화 유·무 때문이다. 변환과 양자화 과정을 수행하지 않는 무손실 압축의 경우, 잔여 데이터(0이 아닌 계수와 0 계수)의 분포가 스캔 위치에 따라 독립적이고, 잔여 데이터의 절대값이 고주파 쪽으로 이동해도 작아지지 않는다. 즉, 잔여 데이터의 절대값도 역시 스캔 위치

에 독립적이다. 그림 5는 스캔 위치에 따른 0이 아닌 계수의 존재 확률을 나타낸다. 예상대로, 손실 압축과 무손실 압축의 잔여 데이터 사이에는 통계적 특성 차이가 있음을 확인할 수 있다.

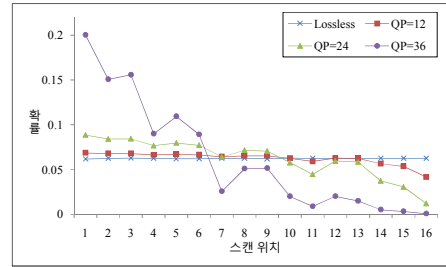


그림 5. 스캔 위치에 따른 0이 아닌 계수의 존재 확률

그림 6은 무손실 부호화에서 잔여 데이터의 통계적 특성을 고려한 새로운 CAVLC 방법을 나타낸다. 처음 단계에서는 새로운 VLC 표를 사용하여 하위 블록 (4×4) 내에 존재하는 0이 아닌 계수의 수를 부호화한다. 레벨 부호화는 새로운 문턱값을 갖는 적응적인 Lev-VLC 표를 선택하여 부호화를 수행한다. 마지막으로 0의 계수 부호화는 부호화 방향을 변경한 후, 새로운 방법을 사용하여 각 0의 위치를 부호화한다.



그림 6. 제안한 CAVLC의 잔여 데이터 부호화 구조

#### 나. 제안하는 CAVLC 부호화 방법

무손실 화면내 부호화 환경에서 위성 영상을 실험한 결과, 0이 아닌 계수의 수 부호화에서 FLC(fixed length code) 표가 약 95% 정도 선택됨을 확인했다. 따라서 제안하는 방법은 기존의 3개의 VLC(variable length code) 표를 제거하고, 하나의 FLC 표만 사용하여 부호화를 수행한다. 모든 0이 아닌 계수의 수에 동일하게 4 비트를 할당하는 FLC 표 대신, 무손실 압축의 통계적 특성을 고려하여 새로운 VLC 표를 설계한다.

```
//write VLC for numcoeff with modified FLC
if (numcoeff > 12) // (13 ≤ numcoeff ≤ 16)
    length = 3; // 1(check bit) + 2 (numcoeff bit)
else // (0 ≤ numcoeff ≤ 12)
    length = 5; // 1(check bit) + 4 (numcoeff bit)
if (numcoeff > 0)
{
    if (numcoeff > 12)
        info = numcoeff - 13;
    else
        info = numcoeff - 1;
}
else // numcoeff is equal to 0
    info = 15;
//makes codeword
if (length == 3)
    bitpattern = 0x00; //check bit
else
    bitpattern = 0x01; //check bit
```

제한한 VLC 표에서, 0이 아닌 계수의 수가 1~12이면 4 비트를, 13~16이면 2 비트를 할당한다. 복호기에서 모호함을 제거하기 위해 각 코드워드의 접두사에 1 비트의 체크 비트를 삽입한다. 앞에서는 제한한 VLC 표의 코드워드에 대한 의사 코드(pseudo-code)를 보인다.

레벨 부호화에서 각 0이 아닌 계수의 절대값은 7개의 Lev-VLC 표 중 적합한 하나의 Lev-VLC 표를 선택하여 부호화를 수행한다. 기존의 레벨 부호화에서 Lev-VLC 표는 계수의 절대값이 고주파 영역에서 저주파 영역으로 갈수록 증가한다는 사실을 기반으로 선택된다. 그러나 무손실 부호화에서 계수의 절대값은 그림 7에서 보듯이 스캔 위치에 독립적이다. 따라서, 이전에 부호화된 계수의 절대값에 따라 증가뿐만 아니라, 감소할 수 있는 적응적인 Lev-VLC 표 선택 방법을 제안한다.

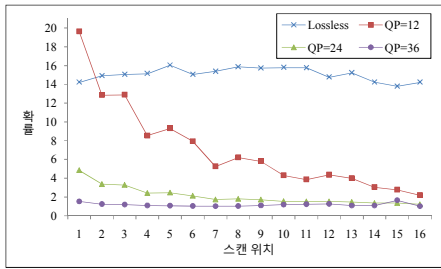


그림 7. 스캔 위치에 따른 계수의 평균 절대값 분포

무손실 부호화의 레벨 부호화에서 다음 부호화할 계수의 절대값은 일반적으로 현재 부호화한 계수의 절대값보다 크다고 가정할 수 없다. 다음에 부호화할 계수의 절대값이 증가할지 혹은 감소할지 예측할 수 없으므로, Lev-VLC 표는 이전에 부호화된 계수들의 절대값의 가중합을 사용하여 결정한다.

$$T(abs\_level_i) = \frac{1}{a_i + 1} a_i \cdot avg_i + abs\_level_i, \quad (1)$$

$$a_i = \begin{cases} 0, & i = lastcoeff \\ 1, & i = lastcoeff - 1, lastcoeff - 2, \\ 2, & otherwise \end{cases}$$

$$avg_i = \frac{1}{(lastcoeff - i + 1)} \left\{ \sum_{k=lastcoeff}^i abs\_level_k \right\} \quad (2)$$

여기서  $a_i$ 와  $abs\_level_i$ 는  $i$ 번째 가중 인자와 계수의  $i$ 번째 절대값을 나타낸다. 그리고  $T(abs\_level_i)$ 와  $lastcoeff$ 는 Lev-VLC 표를 선택하기 위한 새로운 문턱값과 마지막 0이 아닌 계수의 위치를 나타낸다. 표 1은 새로운 문턱값에 대응하는 Lev-VLC 표를 나타낸다.

표 1. 새로운 문턱값에 대응하는 Lev-VLC 표

Lev-VLC 표	$T(abs\_level_i)$
Lev-VLC0	0
Lev-VLC1	2
Lev-VLC2	4
Lev-VLC3	9
Lev-VLC4	19
Lev-VLC5	39
Lev-VLC6	>39

그림 7에서 보듯이, 마지막 계수의 절대값은 손실 부호화와 무손실 부호화에서 큰 차이가 있다. 손실 부호화에서 마지막 계

수의 절대값은 일반적으로 작은 값을 갖기 때문에, 작은 값의 부호화에 효율적인 Lev-VLC0 혹은 Lev-VLC1의 Lev-VLC 표를 선택하여 레벨 부호화를 시작한다. 그러나 무손실 부호화에서 마지막 계수의 절대값은 Lev-VLC0 혹은 Lev-VLC1의 Lev-VLC 표를 사용하여 부호화를 시작할 만큼 충분히 작지 않다. 여러 위성 영상을 사용하여 실험한 결과, 마지막 계수의 평균 절대값은 약 13임을 확인했다. 따라서, 무손실 환경에서 레벨 부호화는 Lev-VLC4를 사용하여 부호화를 시작한다.

0 부호화는 두 과정으로 이루어진다. 우선 0이 아닌 가장 고주파 계수 앞에 존재하는 모든 0의 수를 부호화한 후, 각 0이 아닌 계수에서 0의 위치를 부호화한다. 이때, 0의 위치는 각 0이 아닌 계수 앞에 존재하는 연속하는 0의 수(run\_before)와 0이 아닌 가장 고주파 계수부터 시작하여 부호화되지 않은 0의 수(zerosleft)를 고려하여 선택된 VLC 표를 사용하여 부호화한다. 각 0의 위치 부호화는 더 이상 부호화할 0이 없거나 혹은 마지막 0이 아닌 계수에서 종료한다.

그림 8에서 보듯이, 손실 부호화에서 0은 고주파 영역으로 갈수록 증가한다. 그러나 무손실 부호화에서 0의 분포는 스캔 위치에 독립적이다. 따라서, 0의 위치 부호화 방향을 바꾸고 0이 아닌 가장 고주파 계수 앞에 존재하는 모든 0의 수 부호화를 생략한다. 0이 아닌 가장 고주파 계수 앞에 존재하는 모든 0의 수를 부호화하는 이유는 역방향으로 각 0의 위치를 부호화할 때, 0이 아닌 가장 고주파 계수의 위치를 모르기 때문이다. 하지만 제안하는 방법은 각 0의 위치 부호화 방향을 순방향으로 변경하였으므로, 0이 아닌 가장 고주파 계수 앞에 존재하는 모든 0의 수를 부호화할 필요가 없다.

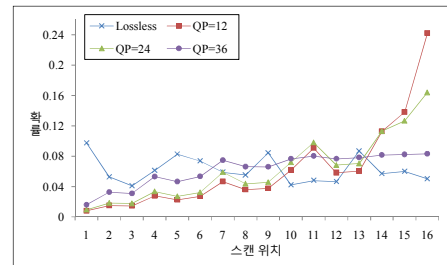


그림 8. 스캔 위치에 따른 0이 아닌 계수의 존재 확률

제한한 0의 위치 부호화 순서가 역방향에서 순방향으로 변경되었으므로, 새로운 0의 위치 부호화 방법을 제안한다. 각 0의 위치를 부호화하기 위해 사용하는 VLC 표는 각 0이 아닌 계수 앞에 존재하는 연속하는 0의 수와 DC 계수부터 시작하여 아직 부호화되지 않은 0의 수(all\_zeros)를 고려하여 선택한다. 또한 0의 위치 부호화의 방향이 변경되었기 때문에, 마지막 0이 아닌 계수에서도 0의 위치 부호화를 수행해야 한다. 그림 9는 제안한 0 부호화 과정을 나타낸다.

스캔 위치	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
계수 레벨 값	0	3	-2	0	1	0	0	0	0	0	0	0	0	0	0	0
부호화 순서	<i>abs_level</i>		<i>all_zeros</i>									<i>run_before</i>				
1	3		13									1				
2	-2		12									0				
3	1		12									1				

그림 8. 스캔 위치에 따른 0이 아닌 계수의 존재 확률

표 3. 제안한 알고리즘의 성능

영상	원 영상 크기 (bytes)	방법	전체 비트수 (bytes)	압축율	비트수 이득 (%)
서울 도심 영상	806439	JPEG-LS	633276	1.2734	-13.73
		CALIC	619990	1.3007	-11.35
		제안한 방법	556818	1.4483	0
부산 도심 영상	806439	JPEG-LS	627850	1.2844	-13.09
		CALIC	614484	1.3124	-10.68
		제안한 방법	555170	1.4526	0
평균		JPEG-LS		1.2789	-13.41
		CALIC		1.3066	-11.02
		제안한 방법		1.4505	0

손실 부호화에서 각 0의 위치를 부호화하기 위해 사용한 VLC 표의 run\_before 범위는 0~14이다. 이때, 코드워드의 길이는 1~11이다. 왜냐하면, 기존의 0의 위치 부호화는 마지막 0이 아닌 계수에서는 0의 위치를 부호화하지 않았기 때문이다. 그러나 무손실 부호화에서 제안한 방법은 마지막 0이 아닌 계수에서도 0의 위치를 부호화하므로, 새로운 코드워드를 설계해야 한다. 따라서 run\_before가 15인 경우, 코드워드 길이가 12이고 코드워드 값이 1인 새로운 코드워드를 설계한다.

#### 4. 실험 결과 및 분석

본 논문에서는 H.264/AVC 표준 소프트웨어 JM 13.2 [3]를 이용하여 실험했다. 그림 10은 1120×720 해상도의 위성 영상으로 Google earth [4]에서 제공하는 Google earth 5 프로그램을 사용하여 획득했다.



(a) 서울 도심 영상 (b) 부산 도심 영상

그림 10. 위성 영상

표 2는 H.264/AVC를 사용한 무손실 위성 영상 부호화의 실험 환경을 나타낸다.

표 2. 실험 환경

실험 변수	값
ProfileIDC	244 (High 4:4:4)
QPISlice	0
SymboleMode	0 (CAVLC)
QPPrimeYZeroTransformBypassFlag	1 (Lossless)

제안한 방법의 성능 평가를 위해 무손실 정지 영상 압축에 널리 쓰이는 JPEG-LS [5]와 CALIC [6]을 사용하여 비교했다. 성능 평가는 압축율과 비트수 이득(%)을 사용했다.

$$\text{압축율} = \frac{\text{비트수 원영상}}{\text{비트수 제안한 방법}} \quad (3)$$

$$\text{비트수이득} = \frac{\text{비트수 제안한 방법} - \text{비트수 무손실영상압축}}{\text{비트수 제안한 방법}} \times 100 \quad (4)$$

표 3의 실험 결과로부터, 제안한 방법이 기존의 JPEG-LS와 CALIC 보다 효율적임을 확인할 수 있다.

#### 5. 결론

본 논문에서는 H.264/AVC의 무손실 화면내 부호화에서 손실 압축과 무손실 압축의 잔여 신호 통계적 특성 차이를 분석했다. 이를 기반으로 무손실 위성 영상 압축 환경에 적합한 향상된 문맥 기반 적응적 가변 길이 부호화 (CAVLC) 방법을 제안했다. 제안한 방법에서는 0이 아닌 계수들의 효율적인 부호화를 위해 새로운 VLC 표를 설계했다. 그리고 레벨 부호화를 위한 새로운 문턱값을 갖는 적응적인 Lev-VLC 표를 선택 방법을 제안했다. 마지막으로, 효율적인 0 부호화를 위한 새로운 연속하는 0 계수의 부호화 방법을 제안했다. 실험 결과로부터 본 논문에서 제안하는 향상된 CAVLC 부호화 방법이 기존의 무손실 압축을 위해 사용되는 JPEG-LS와 CALIC에 비해 약 12% 정도의 비트 수를 감소시키는 것을 확인했다.

#### 감사의 글

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음. (NIPA-2009-(C1090-0902-0017))

#### 참고 문헌

- [1] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560-576, July 2003.
- [2] I.E.G Richardson, H.264 and MPEG-4 Video Compression. Wiley, 2003.
- [3] [http://iphome/hhi.de/shehring/tml/download/old\\_jm/jm13.2.zip](http://iphome/hhi.de/shehring/tml/download/old_jm/jm13.2.zip), Joint Video Team, Reference Software Version 13.2.
- [4] <http://earth.google.com>
- [5] M.J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," IEEE Trans. Image Process., vol. 9, no. 8, pp. 1309-1324, Aug. 2000.
- [6] X. Wu and N. Memon, "Context-Based, Adaptive, Lossless Image Coding," IEEE Trans. Comm., vol. 45, no. 4, pp. 437-444, April 1997.