

CCSDS 122.0-B-1 : 고해상도위성의 영상압축 기술

서석배*, 구인회**

CCSDS 122.0-B-1 : An Image Compression Technology for High Resolution Satellites

Seok-Bae Seo*, In-Hoi Koo**

ABSTRACT

In recent, image data compression method of high resolution satellite image is the important issue for its application and development. The CCSDS (Consultative Committee for Space Data Systems) published a standard for the high resolution image data compression, CCSDS 122.0-B-1, in the end of 2005, which is expected to be widely applied in process of compression for the high resolution satellite images. In this paper, it is explained that the current trends of image compression methods for high resolution satellites, and then the comparison results between CCSDS 122.0-B-1 standard and JPEG are described.

초 록

최근 고해상도위성의 개발 및 활용이 활발해지면서, 데이터 전송 및 처리를 위한 영상압축의 중요성이 대두되고 있다. 이에 발맞추어 CCSDS (Consultative Committee for Space Data Systems)에서는 고해상도위성의 영상압축 권고안인 CCSDS 122.0-B-1을 2005년 말에 발표했으며, 이는 향후 고해상도위성의 영상압축에 널리 적용될 것으로 기대된다. 본 논문에서는 고해상도위성의 최근 동향을 알아보고, CCSDS 122.0-B-1 기반의 고해상도 위성 영상압축 기술에 대해 설명하며, JPEG과 성능을 비교한 결과를 기술한다.

Key Words : Image Compression, High Resolution Satellite, CCSDS 122.0, JPEG, and Wavelet Transform.

* 서석배, 한국항공우주연구원 우주응용센터 지상체계개발팀
sbseo@kari.re.kr

** 구인회, 한국항공우주연구원 우주응용센터 지상체계개발팀
freewill@kari.re.kr

1. 서론

고해상도위성영상의 정의는 그 활용분야에 따라 다르다. 최근 고해상도위성영상의 주 활용분야가 군용에서 상용으로 옮겨가고 있으며, 1m 이하의 해상도를 고해상도로 간주하고 있다. 현재 활용되거나 개발될 1m 이하 해상도의 위성은 다음 표 1과 같다.[1]

표 1. 1m 이하 고해상도 위성 현황

| 위성 | 국가 | 발사일 년-월-일 | 해상도 PAN/MS | 관측폭 Km |
|----------------|------|--------------|---------------|-----------|
| IKONOS-2 | 미국 | 99-09-24 | 1.0/4.0 | 11 |
| QuickBird-2 | 미국 | 01-10-18 | 0.6/2.4 | 16 |
| EROS B1 | 이스라엘 | 06-04-25 | 0.7/- | 7 |
| Resurs DK-1 | 러시아 | 06-06-15 | 1.0/3.0 | 28 |
| KOMPSAT-2 | 한국 | 06-07-28 | 1.0/4.0 | 15 |
| IRS Cartosat-2 | 인도 | 07-01-10 | 0.8/- | 10 |
| WorldView-1 | 미국 | 07-09-18 | 0.5/- | 16 |
| WorldView-2 | 미국 | Mid-2009 | 0.5/1.8 | 16 |
| GeoEye-1 | 미국 | 08-09-06 | 0.4/1.64 | 15 |
| EROS C | 이스라엘 | 09-04-01 | 0.7/2.8 | 11 |
| KOMPSAT-3 | 한국 | 2011 | 0.7/2.8 | 15 |
| Pleiades-1 | 프랑스 | 10-03-01 | 0.7/2.8 | 20 |
| Pleiades-2 | 프랑스 | 11-03-01 | 0.7/2.8 | 20 |

고해상도위성의 영상은 용량이 크며 높은 공간해상도를 요구하므로 특성상 대부분 저궤도에서 촬영된다. 저궤도 위성의 경우 지상과 접촉할 수 있는 시간이 한정적이므로, 제한된 시간에 많은 양의 데이터를 전송하기 위해서는 영상압축 기술이 반드시 필요하다. 화소당 해상도(BPP, Bits Per Pixel)가 8이고, PAN은 하나의 채널, MS는 네 개의 채널이며 PAN 해상도의 1/4일 때 데이터크기 변화를 그림 1에 나타내었다.

그림 1에서, 관측폭이 10Km 일 때(수신 영상의 세로의 길이는 관측폭과 같다고 가정함, 10Km), 해상도가 1m에서 0.7m로 증가하면 용량이 2배, 0.5m로 증가하면 4배로 증가하며, 관측폭이 넓어지면 그 증가세도 큰 폭으로 증가됨을 볼 수 있다. 따라서 관측폭이 커지고 해상도가 높아질수록 용량이 커지는 데이터를 위성에서 지상으로 전송하기 위해서는 영상압축이 반드시 필요함을 알 수 있다.

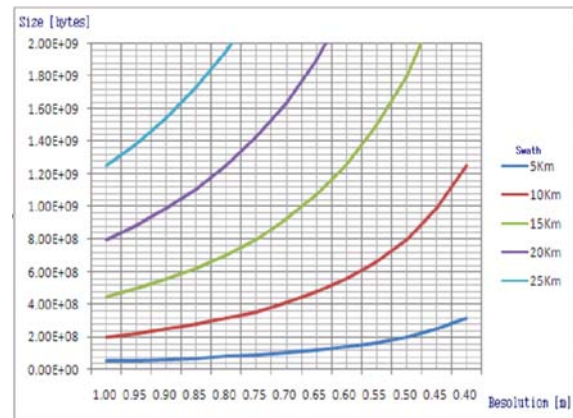


그림 1. PAN/MS 해상도 1m/4m 일 때 영상의 크기변화

본 논문에서는 2005년에 CCSDS (Consultative Committee for Space Data Systems)에서 고해상도 위성영상의 압축의 권고로 발표한 122.0-B-1을 설명하며, 이를 위해서 근간이 되는 JPEG, JPEG 2000의 개략적인 내용을 정리하였다.

본 논문의 2장에서는 현재 널리 사용되는 압축 알고리즘인 JPEG에 관해서 간략히 살펴보고, 3장에서는 본 논문의 핵심인 CCSDS 122.0-B-1 압축 알고리즘(이하 122.0)을 설명한다. 다음으로 4장에서는 JPEG 알고리즘과 122.0과의 성능비교결과를 살펴본다.

2. JPEG 압축

영상압축은 손실압축(lossy compression)과 무손실압축(lossless compression)으로 구분된다. 용어에 함축된 의미처럼 손실압축은 압축 전 영상과 압축을 복원한 영상에 차이가 있는 것으로 데이터 손실이라는 단점을 가지는 대신 높은 압축 성능 때문에 인터넷, 디지털카메라, 디지털방송 등 상용분야에 널리 사용되고 있다. 무손실압축의 경우 압축 전 영상과 압축이 복원된 영상이 동일한 것으로 영상의 손실 없이 압축을 하고자 할 때 사용되며, 손실압축에 비해서 낮은 압축률을 가진다.

본 장에서는 현재 널리 사용되는 JPEG의 손실압축(2.1)과 무손실압축 알고리즘(2.2)에 대해서 간략히 알아본다. 이어서 122.0의 근간이 되는 JPEG 2000(2.3)에 대해서 설명하였다.

2.1 JPEG : 손실 JPEG

JPEG은 정지영상을 위해 만들어진 손실 압축 방법으로, ISO와 ITU-T에서 1994년에 표준으로 제정하였다(ISO 10918-1). 영상의 품질을 감소시키는 단점이 있으나, 10분의 1 정도의 압축률을 달성할 수 있어 현재 영상을 저장하는 포맷으로 널리 사용되고 있다. JPEG 압축의 과정은 다음과 같이 요약된다.[2]

- 색공간변환 : 칼라영상의 경우 RGB로 구분된 데이터를 YCbCr이라는 공간으로 변환한다.
- 크로마 서브샘플링 : Cb와 Cr 성분 수를 줄인다.
- DCT : 전체 영상은 8×8 화소의 크기를 갖는 섹션으로 나뉘지고 각각에 대해서 DCT(이산 코사인 변환)을 수행한다. DCT의 장점은 영상을 주파수영역으로 변환했을 때 높은 밀집도를 가지는 특성이 있어 압축의 성능을 높일 수 있다는 것이다.
- 엔트로피 부호화 : DCT의 결과를 저주파에서 고주파 순으로 정렬한 다음(지그재그 스캔), 고주파에 반복되는 0에 대해서 RLE (Run-Length Encoding)을 수행한다. 다음으로 데이터의 빈도수에 따라서 가변적인 비트를 할당하는 허프만 부호화를 수행한다.

그림 2는 이상 설명한 손실 JPEG을 설명한 것이다.

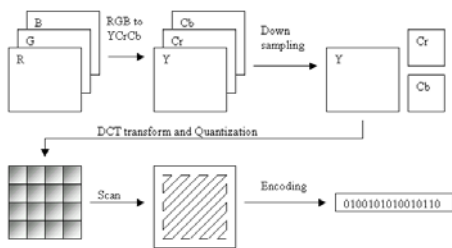


그림 2. 손실 JPEG 알고리즘

2.2 Lossless JPEG : 무손실 JPEG

1993년에 표준으로 정립된 lossless JPEG은 2분의 1 정도로 영상을 압축할 수 있다.[2] Lossless JPEG의 알고리즘은 크게 예측과 엔트로피부호화로 나눌 수 있으며, 예측에서는 영상에서 인접 화소들은 유사한 값을 가진다는 특성을 이용하여 DPCM (Differential Pulse Code Modulation)으로 변환한 다음, 그 결과를 손실 JPEG과 유사하게 엔트로피 부호화를 수행한다. 그림 3은 무손실 JPEG의 알고리즘을 설명한 것이다.

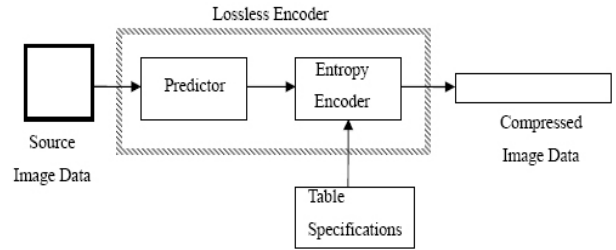


그림 3. 무손실 JPEG 알고리즘

2.3 JPEG 2000

JPEG 2000은 웨이브렛 변환을 기본으로 하여 정립된 표준으로 (ISO/IEC 15444-2:2000), 기존의 압축 알고리즘에 비해서 다음의 특징을 갖는다.[3]

- 최신의 저 bit-rate 압축 성능
- 영상 품질, 해상도, 컴포넌트 또는 공간 지역성에 의한 점진적 전달의 가능
- 손실 또는 무손실 압축 모두 지원.
- Bitstream에 대한 Random 또는 공간적 접근의 가능
- Rotation, Cropping 등의 압축된 영역 처리
- 관심 영역(ROI, Region of Interest)의 coding 가능
- 제한된 메모리에서의 수행 가능

그림 4는 JPEG 2000의 압축 알고리즘을 설명한 것으로 크게 웨이브릿 변환과, 변환된 결과에 대해서 코딩을 수행하는 부분으로 나눌 수 있다.

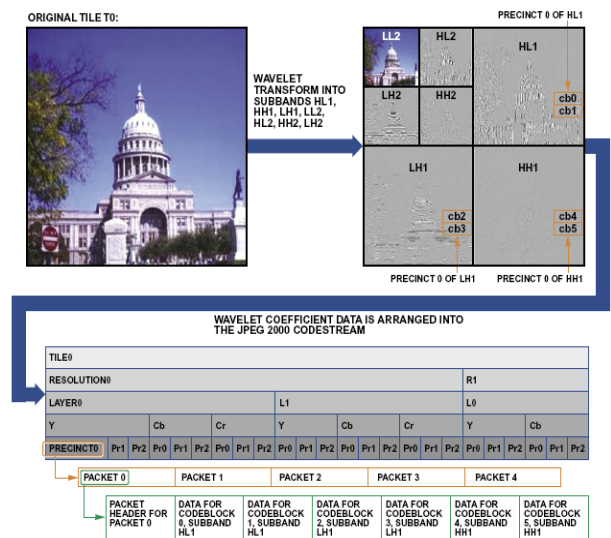


그림 4. JPEG 2000 알고리즘

3. CCSDS 122.0-B-1

CCSDS는 다음의 요구사항에 기반하여, 2005년도에 122.0을 고해상도 위성영상의 압축의 표준으로 결정하였다.[5]

1. Frame 기반의 영상 및 Push-broom 기반 영상 모두 적용 가능할 것
2. 압축률과 품질의 조정이 가능할 것. (품질의 측면에서 lossless 압축 가능할 것)
3. 최소 4-bit 영상에서 최대 16-bit 영상에 대해 적용 가능할 것.
4. 위성체내 압축 관련 전자 장치에서의 실시간 처리가 가능할 것.
5. 영상의 미소 영역에서의 packet 손실 효과를 제한할 것.

이렇게 표준으로 제정된 122.0은 JPEG 2000과 다음과 같은 차이점을 갖는다. [6]

1. 전송 channel의 bandwidth 감소
2. Buffering 및 Data 저장 공간 감소
3. 주어진 전송률에서의 Data 전송 시간 감소

위에서 언급한 3가지 특징 이외에는 JPEG 2000과 많은 부분이 유사하다.

JPEG 2000에서 설정할 수 있는 많은 옵션 부분이 있는데, 122.0에서는 이를 고해상도 위성영상에 초점을 맞추어 고정하여 알고리즘을 간소화시킴으로써 고해상도 위성영상 압축분야에서 표준으로 활용할 수 있는 장점이 있다.

122.0의 압축은 DWT (Discrete Wavelet Transform) 와 BPE (Bit Plane Encoder)로 크게 나눌 수 있으며 다음 절부터 그 알고리즘에 대해서 설명한다. 그림 5에 이상 설명한 122.0 알고리즘을 간략하게 나타내었다.

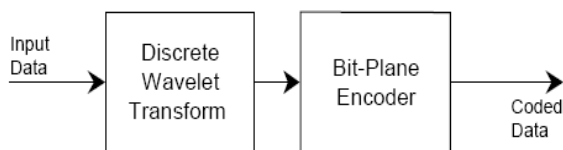


그림 5. 122.0-B-1 알고리즘

3.1 DWT

웨이블릿변환(Wavelet transforms)은 JPEG에 사용되는 이산코사인변환(DCT, Describe Cosine Transform)이나 푸리에변환(FT, Fourier Transform)에 비하여 불연속성, 급격한 변화, 비주기적 특성을 가지는 신호를 잘 표현할 수 있는 장점이 있다. DWT는 디지털신호나 영상과 같이 이산적인 형태에 적용되는 웨이블릿변환이다.[2]

영상을 DWT하면 네 개의 주파수밴드(LL, LH, HL, HH)로 나뉘지고, 가장 저주파대(LL)의 영역에 DWT를 반복해서 수행하면 서브밴드 코딩된 영상을 얻을 수 있다. DWT는 JPEG처럼 영상을 작은 크기(8×8)로 나뉘서 처리하지 않기 때문에, DWT를 이용한 영상 압축에서는 복원영상에 블록화 현상이 발생하지 않는 장점이 있다. 그림 6은 영상에 대해서 3번의 DWT를 수행한 예를 보여준다.

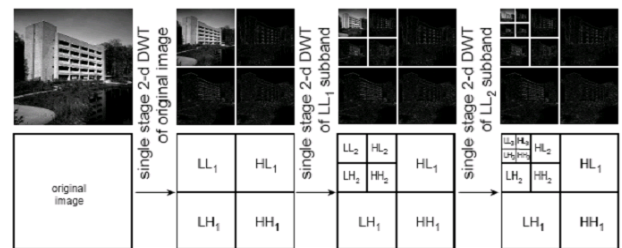


그림 6. 3번의 DWT를 수행한 결과

DWT는 크게 Float Transform과 Integer Transform으로 나뉜다.

122.0에서는 Float Transform을 위해서 9/7 필터를 표 2와 같이 정의하고 있으며, 역변환(Inverse Float Transform)을 위해서는 별도의 9/7필터를 정의하고 있다.

표 2. 9/7 Float Transform을 위한 계수

| i | Low Pass Filter (h _i) | High Pass Filter (g _i) |
|----|-----------------------------------|------------------------------------|
| 0 | 0.852698679009 | -0.788485616406 |
| ±1 | 0.377402855613 | 0.418092273222 |
| ±2 | -0.110624404418 | 0.040689417609 |
| ±3 | -0.023849465020 | -0.064538882629 |
| ±4 | 0.037828455507 | · |

Float Transform 역변환을 위한 9/7 필터의 계수값은 저주파필터와 고주파 필터의 값이 바뀌고, 부호가

반대이다. 필터를 이용한 (웨이브렛) Float Transform의 핵심은 필터 설계 즉, 필터의 크기와 그 계수를 결정하는 것인데, 122.0에 크기와 그 계수가 모두 정의되었기 때문에 일반적인 필터 बैं크 (filter bank)의 처리하는 방법에 따라 쉽게 구현할 수 있다.[2] Float Transform의 경우 웨이브렛 변환한 결과는, 이후 처리과정인 BPE에서 가장 가까운 정수로 근사화되므로 역변환시 완전복원(lossless compression)이 되지 않는 특징이 있다.

Integer Transform은 다음의 (수식 1)과 (수식 2)를 이용해서 처리할 수 있으며, 여기서 N은 처리할 영상의 (가로 또는 세로) 길이, C_j는 Low Pass Filer의 계수, D_j는 High Pass Filter의 계수이다.

$$x_0 = C_0 + \left[-\frac{D_0}{2} + \frac{1}{2} \right]$$

$$x_{2j} = C_j + \left[-\frac{D_{j-1} + D_j}{4} + \frac{1}{2} \right] \quad \text{for } j = 1, \dots, N-1 \quad (\text{수식 1})$$

$$x_1 = D_0 + \left[\frac{9}{16}(x_0 + x_2) - \frac{1}{16}(x_2 + x_4) + \frac{1}{2} \right]$$

$$x_{2j-1} = D_j + \left[\frac{9}{16}(x_{2j} + x_{2j+2}) - \frac{1}{16}(x_{2j-2} + x_{2j+4}) + \frac{1}{2} \right] \quad \text{for } j = 1, \dots, N-3$$

$$x_{2N-3} = D_{N-2} + \left[\frac{9}{16}(x_{2N-4} + x_{2N-2}) - \frac{1}{16}(x_{2N-6} + x_{2N-2}) + \frac{1}{2} \right]$$

$$x_{2N-1} = D_{N-1} + \left[\frac{9}{8}x_{2N-2} - \frac{1}{8}x_{2N-4} + \frac{1}{2} \right]$$

$$C_0 = x_0 - \left[-\frac{D_0}{2} + \frac{1}{2} \right]$$

$$C_j = x_{2j} - \left[-\frac{D_{j-1} + D_j}{4} + \frac{1}{2} \right] \quad \text{for } j = 1, \dots, N-1 \quad (\text{수식 2})$$

해당 데이터를 Integer Transform하면 그 개수의 2배의 계수가 생성되는데, 서브밴드 코딩을 위해서 1/2로 서브샘플링(sub-sampling) 해야 하며, 이 과정에서 완전복원(lossless compression)을 위하여 DC 부분을 제외한 각각의 서브밴드에 대한 weight를 고려해야 한다.

122.0의 첫 번째 단계인 Float Transform 또는 Integer Transform을 선택해서 DWT를 마치면, 실제적인 압축이 이루어지는 BPE를 수행한다.

3.2 BPE (Bit Plane Encoder)

영상에 대해서 DWT를 수행하면 영상과 같은 크기로 DWT 변환결과가 생성된다. 그림 7은 3번의 DWT를 수행한 결과이다.

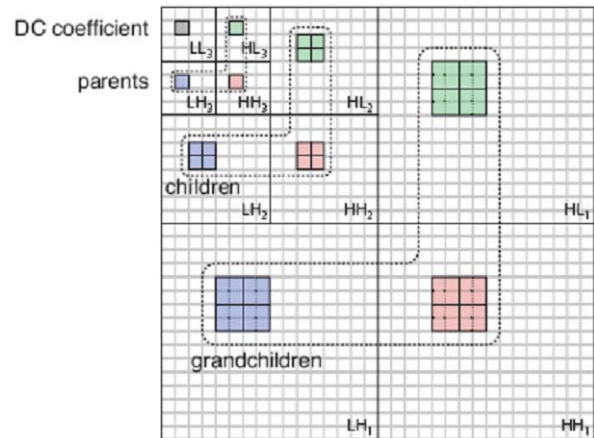


그림 7. DWT 수행 결과

BPE 구체적인 알고리즘을 설명하기 전에, 122.0에서 정의한 용어(definition)에 대해서 알아본다.

- 블록(block) : DWT 결과에서 하나의 DC 값과 이와 관련된 모든 AC 값들의 집합. 그림 8에 블록의 구성을 나타내었다.
- 세그먼트(segment) : 블록을 일정한 개수로 묶은 것으로, 16과 2²⁰ 사이로 설정한다.
- gaggle : 세그먼트 내의 데이터 묶음 단위로 16개의 블록으로 구성된다. 단 하나의 세그먼트에서 마지막 gaggle은 16이하의 블록으로 구성된다.
- BitDepthDC : 한 세그먼트 내에서 DWT 결과의 최대 DC값에 할당된 비트 수
- BitDepthAC : 한 세그먼트 내에서 DWT 결과의 최대 AC값에 할당된 비트 수

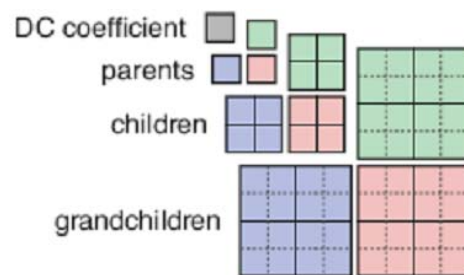


그림 8. 정의된 블록(block)의 구성

BPE는 각 세그먼트단위로 수행되며, 최종결과는 표 3과 같은 포맷으로 저장된다. 즉 각 세그먼트에 대해서 하나의 세그먼트 헤더가 있으며, 해당 세그먼트에 대해서 DC 및 AC 계수를 계산한 다음, AC 계수의 각 유효 비트에 대해서 코딩이 수행된다.

표 3. BPE 포맷

| |
|---|
| 3.2 (1) : Segment Header |
| 3.2 (2) : Initial coding of DC coefficients |
| 3.2 (3) : Coded AC coefficient bit depths |
| 3.2 (4) : Coded bit plane b = BitDepthAC-1 |
| 3.2 (4) : Coded bit plane b = BitDepthAC-2 |
| ⋮ |
| 3.2 (4) : Coded bit plane b = 0 |

(1) Segment Header

세그먼트 헤더는 BPE를 수행하며 생성되는 여러 가지 설정 또는 계산 값을 포함하고 있으며, 표 4와 같이 크게 네 가지(Part1~ Part4)로 구성된다.

표 4. 세그먼트 헤더의 구성

| Part | 크기 (bytes) | 포함된 정보 |
|------|------------|---|
| 1A | 3 | 영상에서 처음과 마지막 세그먼트 표시, 세그먼트 카운터, BitDepthDC, BitDepthAC, Part2 ~ Part4에 대한 플래그 |
| 1B | 1 | DWT를 위해 추가된 열(row)의 수 |
| 2 | 5 | 세그먼트의 제한 길이, 압축에 대한 정보, 압축을 위한 bit fill 정보 |
| 3 | 3 | 블록 내의 세그먼트 크기, code option k 정보 (표 5) |
| 4 | 8 | DWT 종류, 입력 영상의 부호 유무, 입력 영상의 BPP, 영상의 transpose 유무, codeword 길이, 서브밴드의 가중치 |

Part1은 A와 B로 나뉘어져 있으며 자세한 각각의 항목을 그림 9에 나타내었다. Part1에는 세그먼트마다 항상 포함되는 파트로, 영상의 첫 번째 세그먼트와 마지막 세그먼트, 옵션 파트(Part2~Part4)에 대한 플래그, 세그먼트에 대한 카운터 등을 포함한다. 설명에서 항목의 괄호 내의 숫자는 할당된 bit 수이다.

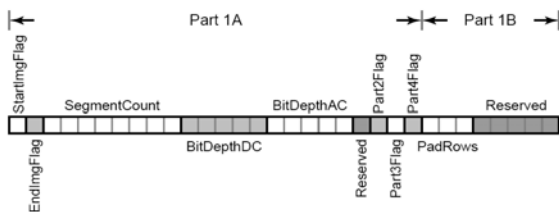


그림 9. Segment Header : Part 1A & 1B

- StartImgFlag(1) : 영상은 세그먼트 단위로 나뉘어 처리되는데, 영상에서 첫 번째 세그먼트 일 때 1이 된다.
- EndImgFlag(1) : 영상에서 마지막 세그먼트 일 때 1이 된다.

- SegmentCount(8) : 세그먼트의 연속성을 체크하기 위한 링 카운터이다.
- BitDepthDC(5) : 한 세그먼트 내에서 DWT 결과의 최대 DC값에 할당된 비트 수로 2의 보수 형태로 저장된다.
- BitDepthAC(5) : 한 세그먼트 내에서 DWT 결과의 최대 AC값에 할당된 비트 수이다.
- Reserved(1) : 0으로 채워짐
- Part2Flag(1) : Part2 헤더의 유무를 저장함. 1일 때 Part2 헤더가 있음.
- Part3Flag(1) : Part3 헤더의 유무를 저장함.
- Part4Flag(1) : Part4 헤더의 유무를 저장함.
- PadRows(3) : DWT에서 채워진 0의 수로 역DWT를 수행한 후에는 제거 됨.
- Reserved(5) : 모두 0으로 채움

그림 10에는 Part2의 세부 항목을 나타내었다. Part2에는 압축 시 한 세그먼트 내의 바이트 수와 압축에 대한 여러 가지 정보를 포함하고 있다.



그림 10. Segment Header : Part 2

- SegByteLimit(27) : 압축 시 한 세그먼트 내의 바이트 수 이다. (mod 2^{27} 로 저장됨)
- DCStop(1) : DC 계수만을 압축 할 때 1 셋팅됨.
- BitPlaneStop(5) : 비트 레벨로 압축할 때 Stop Bit Plane을 나타냄. 이 때 DCStop = 0 임.
- StageStop(2) : 비트 레벨로 압축할 때 Stop stage를 나타냄. 00=stage1, 01=stage2, 10=stage3, 11=stage4.
- UseFill(1) : 각 세그먼트에서 SegByteLimit를 위해서 bit fill이 사용될 때 1로 셋팅됨.
- Reserved(4) : 0으로 채움

그림 11에는 Part3의 세부 항목을 나타내었다. Part3에는 블록 내의 세그먼트의 크기, 양자화에 대한 정보 등을 포함하고 있다.

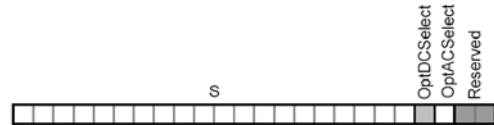


그림 11. Segment Header : Part 3

- S(20) : 블록 내의 세그먼트의 크기를 나타내며, $S \bmod 2^{20}$ 으로 저장됨.
- OptDCSelect(1) : DC 계수를 양자화 할 때 알고리즘에 의한 (optimum) 값을 사용했을 때는 1로, 사용자가 임의로 설정한(heuristic) 값이면 0으로 셋팅함.
- OptACSelect(1) : AC 계수를 양자화 할 때 알고리즘에 의한 값을

사용했을 때는 1로, 사용자가 임의로 설정한 값이면 0으로 셋팅함.

- Reserved(2) : 0으로 채움

그림 12에는 Part4의 세부 항목을 나타내었다. Part4에는 DWT에 대한 정보, 영상에 대한 정보, 서브밴드에 대한 가중치 등의 정보를 포함하고 있다.

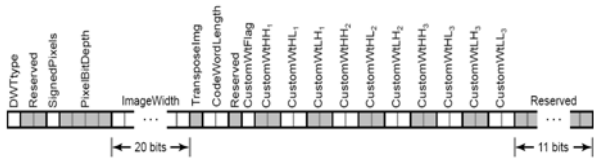


그림 12. Segment Header : Part 4

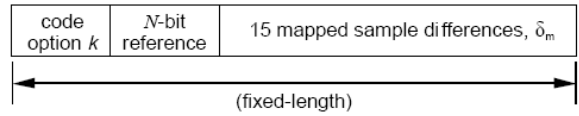
- DWType(1) : Float DWT일 때 0, integer DWT일 때 1로 셋팅.
- Reserved(2) : 0으로 채움.
- SignedPixels(1) : 영상의 각 화소가 부호가 없을 때 1, 있을 때 0으로 셋팅
- PixelBitDepth(4) : 영상 각 화소의 BPP(Bit Per Pixel).
- ImageWidth(20) : 영상의 가로 길이
- TransposeImg(1) : 복원 후 영상을 transepose하면 1, 아닐 경우 0으로 셋팅
- CodeWordLength(2) : codeword의 길이가 8bit일 때 00, 16bit일 때 01, 24bit일 때 10, 32bit일 때 11로 셋팅.
- Reserved(1) : 0으로 채움.
- CustomWtFlag(1) : 서브밴드의 가중치를 미리 정의된 적용할 때 0, 사용자가 정의한 것을 사용하면 1로 셋팅하고 다음의 10개 항목을 설정함.
- CustomWtHH₁(2) : 가중치가 2⁰ 일 때 00, 2¹일 때 01, 2²일 때 10, 2³일 때 11로 채움. 단 CustomWtFlag가 0일 때는 00으로 채워짐.
- CustomWtHL₁(2), CustomWtLH₁(2), CustomWtHH₂(2), CustomWtHL₂(2), CustomWtLH₂(2), CustomWtHH₃(2), CustomWtHL₃(2), CustomWtLH₃(2), CustomWtLL₃(2) : 상동
- Reserved(11) : 모두 0으로 채움.

(2) Initial coding of DC coefficients

BitDepthDC에 따라서 차별적으로 양자화 한 다음, DC 계수 간의 차이를 이용하여 계산한다. BitDepthDC가 3 이하일 때는 양자화를 수행하지 않으며, 다른 경우는 BitDepthAC를 함께 고려하여 양자화의 정도를 결정한다.

DC 계수의 초기화 결과는 다음의 그림 13 또는 그림 14의 포맷으로 저장된다. (gaggle 단위로 처리되는 것에 주의)

a) first gaggle in a segment:



b) subsequent gaggles:

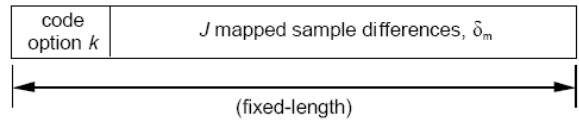
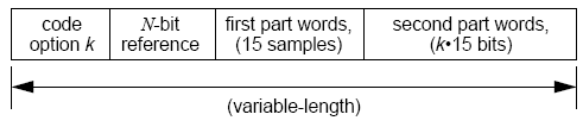


그림 13. DC 계수 초기화 : coding option 미선택

a) first gaggle in a segment:



b) subsequent gaggles:

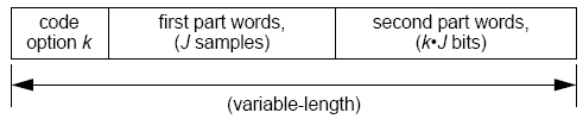


그림 14. DC 계수 초기화 : coding option 선택

그림 10 및 그림 11의 차이는 결과 값의 코딩 유무이고, 이는 code option k 부분을 보면 알 수 있다. 그리고 N은 BitDepthDC-q(양자화 정도)로 계산되며 10 이하의 값이다.

그림15및 그림16에서 code option k 부분에는 기준이 되는 N bit의 값(N-bit reference)이 위치하고, 다음에는 기준 N bit를 기준으로 그 차이를 계산하여 δ_m 에 저장한다. δ_m 은 다음 (수식 3) 및 (수식 4)에 의해서 계산된다. 여기서 c_m 은 기준이 되는 N bit의 값의 계수이고, c_{m-1} 은 현재 계산하는 비트에 대한 계수 값이다.

$$\delta'_m = c'_m - c'_{m-1} \tag{수식 3}$$

$$\delta_m = \begin{cases} 2(\delta'_m), & \text{if } 0 \leq \delta'_m \leq \theta_m \\ 2|\delta'_m|-1, & \text{if } -\theta_m \leq \delta'_m < 0 \\ \theta_m + |\delta'_m|, & \text{otherwise} \end{cases} \tag{수식 4}$$

여기서, $\theta_m = \min(c'_{m-1} - x_{\min}, x_{\max} - c'_{m-1})$,
 $x_{\min} = -2^{N-1}, x_{\max} = 2^{N-1} - 1$

Code option k의 내용은 다음 표 5와 같다.

표 5. Code option k

| k | N=2 | 2<N≤4 | 4<N≤8 | 8<N≤10 |
|----------|-----|-------|-------|--------|
| 0 | 0 | 00 | 000 | 0000 |
| 1 | - | 01 | 001 | 0001 |
| 2 | - | 10 | 010 | 0010 |
| 3 | - | - | 011 | 0011 |
| 4 | - | - | 100 | 0100 |
| 5 | - | - | 101 | 0101 |
| 6 | - | - | 110 | 0110 |
| 7 | - | - | - | 0111 |
| 8 | - | - | - | 1000 |
| un-coded | 1 | 11 | 111 | 1111 |

(3) Specifying the AC bit depth in each block

AC 계수의 비트를 결정하는 방법은 BitDepthAC에 따라서 차별적으로 양자화 한 다음, AC 계수 간의 차이를 이용하여 계산한다. BitDepthAC가 0일 때는 양자화를 수행하지 않으며, BitDepthAC가 1일 때는 0 또는 1의 값으로 계산된다. 나머지는 계산방법과 결과 포맷은 DC계수와 유사하다.

(4) Bit Plane Coding

이 부분은 BitDepthAC에 따라서 계산되며, 5개의 stage로 나뉘어, 한 세그먼트 내에서 각 블록 별로 반복되며 계산된다.

Stage 0 : 하나의 블록에 대해서, 'BitDepthAC-1' 번째 비트에 대한 DC 계수의 유무를 코딩한다. (블록 당 한 번만 수행)

Stage 1-3 : 'BitDepthAC-1' 번째 비트부터 하나씩 감소하며 0이 될 때까지 반복하며 계산한다. 각각의 반복에서 계산되는 결과는 표 6과 같다.

표 6. Bit Plane Coding 결과 : stage1~ stage3

| 구분 | 최대 길이 | 불가능한 값 |
|---------------------------------------|-------|--------|
| types _b [P] | 3 bit | - |
| types _b [C _i] | 4 bit | - |
| types _b [H _{ij}] | 4 bit | 0000 |
| tran _D | 3 bit | 000 |
| tran _G | 3 bit | - |
| tran _H | 4 bit | 0000 |

'types'는 비트별로 처리할 때 현재 처리하는 비트와 미래에 처리하는 비트에 1의 개수를 나타내는 것이다. 예를 들어 BitDepthAC가 5라고 하면, Bit Plane Coding은 4→3→2→1→0 번째 비트 순으로 계산된다. 현재 2 번째 비트를 처리하고 있다고 가정하면 types₂[P]는 P 서브밴드에서 2이하의 비트(2,1,0)에

포함된 1의 개수이다. 'tran'은 transition word의 의미로, 값이 변하는 지점을 표시하기 위해서 사용된다. stage1~ stage3에서 계산된 결과는 엔트로피 부호화되어서 저장된다.

Stage 4 : 'BitDepthAC-1' 번째 비트부터 하나씩 감소하며 0이 될 때까지 반복하며 계산한다. 각각의 반복에서 처리하는 비트 보다 큰 값을 가지는 AC 계수에 대한 정보를 저장한다.

그림 15에 이상 설명한 122.0을 요약하여 설명하였다. 그림에서 'BPE for Segment'는 BitDepthAC 개만큼 반복된다.

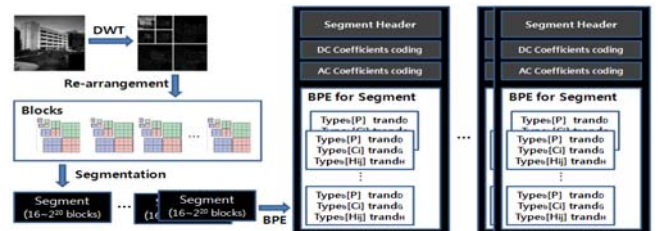


그림 15. 122.0-B-1 알고리즘 요약

4. 성능비교

마지막으로 이상 설명한 122.0과 JPEG 2000의 압축 성능비교에 대한 결과는 표 7과 같다.[5]

표 7. 압축 성능비교 : 122.0-B-1 vs JPEG 2000

| Rate (bits /pixel) | PSNR(dB) | | | | Maximum Absolute Error | |
|--------------------------------|--------------------|-------------|--------------------|--------------------|------------------------|--------------------|
| | CCSDS | | JPEG2000 | CCSDS | | JPEG2000 |
| | Floating point DWT | Integer DWT | Floating point DWT | Floating point DWT | Integer DWT | Floating point DWT |
| Average for 8-bit test images | | | | | | |
| 2.00 | 41.37 | 40.22 | 41.47 | 18.00 | 15.20 | 13.20 |
| 1.00 | 35.76 | 35.18 | 35.52 | 34.40 | 29.00 | 32.20 |
| 0.50 | 32.37 | 31.95 | 32.06 | 60.60 | 52.20 | 52.80 |
| 0.25 | 29.89 | 29.50 | 29.48 | 93.40 | 81.80 | 89.20 |
| Average for 10-bit test images | | | | | | |
| 2.00 | 54.70 | 53.26 | 54.92 | 26.00 | 23.50 | 18.30 |
| 1.00 | 47.76 | 47.10 | 47.80 | 63.20 | 53.40 | 53.90 |
| 0.50 | 42.97 | 42.60 | 42.90 | 115.50 | 95.10 | 113.20 |
| 0.25 | 39.36 | 39.12 | 39.32 | 204.80 | 188.30 | 195.30 |
| Average for 12-bit test images | | | | | | |
| 2.00 | 65.93 | 64.30 | 66.49 | 33.33 | 28.00 | 22.70 |
| 1.00 | 61.18 | 60.17 | 61.20 | 59.00 | 50.67 | 46.30 |
| 0.50 | 58.57 | 57.87 | 58.48 | 88.67 | 78.67 | 83.70 |
| 0.25 | 56.62 | 56.12 | 56.29 | 142.00 | 141.33 | 139.30 |

표 7의 결과는 모두 손실압축의 결과로, 동일한 압축률에서 floating point DWT와 JPEG2000은

유사한 결과를 나타내었고, integer DWT는 성능이 다소 떨어짐을 볼 수 있었다.

Integer DWT는 주로 무손실압축에 사용된다는 점을 고려하면, 위성영상압축에서 손실압축에는 floating point DWT를 이용하고, 무손실압축에서는 Integer DWT는 사용하면 될 것으로 판단된다.

5. 결 론

본 논문에서는 영상압축의 기술인 JPEG에 대해서 간략히 살펴보고, 2005년에 CCSDS에서 표준으로 제정한 영상압축인 CCSDS 122.0-B-1에 관하여 설명하였다. CCSDS 122.0-B-1 압축 표준은 현재 개발 중인 아리랑위성 3호 등, 향후 개발 및 활용될 1m 이하의 고해상도영상압축에 널리 적용될 것으로 기대된다.

참고문헌

1. ASPRS 홈페이지, "Publication & Information → Satellite Information", 2008, www.asprs.org.
2. Wikipedia 홈페이지, "키워드 검색 : JPEG, lossless JPEG, JPEG 2000, Wavelet, filter bank", 2008, <http://en.wikipedia.org/>
3. Michel W. Mareclin, "An overview of JPEG 2000", IEEE Data Compression Conference, pp. 523-541, 2000.
4. Christine Bako, "JPEG 2000 Image Compression", Analog사 홈페이지, www.analog.com
5. Pen-Shu Yeh, "The New CCSDS Image Compression Recommendation", IEEE Aerospace Conference, pp.4138-4145, 2005
6. CCSDS, "Image Data Compression, Recommended Standard CCSDS 122.0-B-1", CCSDS 홈페이지, 2008. <http://public.ccsds.org/>