

EBCOT 알고리즘의 bit-plane 부호화 개선에 대한 연구

호서대학교 컴퓨터공학부
이 호 석
hslee@office.hoseo.ac.kr

A study on the bit-plane coding improvement of EBCOT algorithm

Department of Computer Engineering
Hoseo University
Ho Suk Lee

요 약

본 논문은 EBCOT 알고리즘의 소개와 개선 방법을 제안한다. EBCOT 알고리즘은 웨이블릿 변환과 블록기반 bit-plane 부호화 방법을 활용한 알고리즘이다. EBCOT 에서 사용하는 bit-plane 부호화 방법을 블록기반 fractional bit-plane 방법이라고 한다. 이 방법은 bit-plane 전체를 한번에 부호화하는 것이 아니라 블록으로 나누어 부호화를 수행하고 또한 하나의 bit-plane 에 대하여서도 4 번의 pass 를 거치면서 bit 의 context 에 따라서 부호화를 수행한다. EBCOT 는 웨이블릿 변환에 의하여 resolution 스케일러빌리티를 지원하고 fractional bit-plane 부호화에 의하여 SNR 스케일러빌리티를 지원하며 블록기반 부호화에 의하여 ROI 에 대한 random 접근 기능을 지원한다. 그리고 EBCOT 는 부호화가 완료된 다음에 bit reduction 과정을 수행한다. 이러한 특징들은 이전의 EZW 나 SPIHT 방법에 비하여 장점들이라고 할 수 있다. 그러나 bit-plane 부호화를 수행하는 과정에서 효율을 개선할 수 있으며 본 논문은 이에 대한 방법을 제안한다.

1. EBCOT 알고리즘

EBCOT(Embedded Block Coding with Optimized Truncation) 알고리즘[1]은 JPEG2000[2]에서 채택한 정지영상 부호화 방법이다. EBCOT 알고리즘은 기본적으로 이산 웨이블릿 변환과 블록기반 bit-plane 부호화 방법을 통합한 방법이다. 블록기반 bit-plane 부호화라는 것은 bit-plane 자체를 32x32 혹은 64x64 크기의 블록(EBCOT에서는 코드블록이라고 함)으로 나누고 각 코드블록에 대하여 독립적으로 부호화를 수행한다는 것을 의미한다. EBCOT에서는 이 코드블록을 다시 16x16 크기의 서브블록(sub-block)으로 나누어서 부호화 과정을 수행한다. EBCOT에서 정의하여 사용하는 bit-

plane 을 “fractional bit-plane”이라고 한다. “fractional”이라는 단어가 의미하듯이 bit-plane 으로 부터 부호화를 할 때 bit 의 context 를 조사하여 4 번의 pass 를 거쳐서 context 조건을 만족하는 bit 양 만큼씩 bit-plane 을 부호화한다. 그리고 그 결과를 출력 layer 로 출력한다. 출력 layer 도 최소 1 개에서 최대 50 개까지 구성할 수가 있다. 이러한 방법은 EBCOT 가 최대한의 스케일러빌리티를 갖도록 하기 위하여 설계되었다. EBCOT 는 다음의 장점을 가진다. (1) SNR 스케일러빌리티 (2) resolution 스케일러빌리티 (3) 코드블록 기반 부호화에 의한 ROI 에 대한 random 접근 (4) PCDR(Post Compression Rate Distortion). PCDR 알고리즘을 사

용함으로써 부호화를 모두 수행하고 난 다음에 bit reduction 과정을 수행한다. EBCOT 알고리즘은 각 서브밴드를 다시 작은 코드블록으로 분할하여 수행한다. 이 코드블록의 크기는 보통 32×32 혹은 64×64 이다. 그리고 서브밴드에 속한 각 코드블록에 대하여 독립적으로 부호화를 수행하여 bitstream 을 생성한다. 보통 $2,560 \times 2,048$ 크기의 영상을 생각할 경우 가로에 40 개 세로에 32 개의 코드블록이 있으므로 전체적으로 1,280 개의 코드블록들이 있다. 이와 같이 많은 수의 독립적인 코드블록들로 인하여 EBCOT 알고리즘은 각 코드블록에 의하여 생성되는 bitstream 과 영상 전체에 대한 bitstream 을 별개로 생각하여, 영상에 대한 전체 부호화 과정을 두 단계의 구조(2-tier structure)로 나누어 생각한다. 좀더 구체적으로 설명하면, 전체 bitstream 은 다수개의 quality layer 로 구성되며 각 코드블록들은 생성된 bit 들을 각 quality layer 에 제공하여 전체 bitstream 을 구성하는 것이다. 여기서 두 단계의 구조라는 개념은 quality layer 와 다음 단계의 엔트로피 부호화 과정하고는 독립적으로 생각할 수 있다는 것이다. 즉, quality layer 를 구성하는 여러 개의 layer 의 bitstream 과 엔트로피 부호기에 입력되는 bitstream 을 독립적으로 생각하여 처리할 수 있다. 이러한 특징 때문에 전체 시스템 부호화는 각 코드블록들이 각 quality layer 에 생성하는 상대적인 bit 양을 결정하는 정책을 비교적 자유롭게 선택하여 영상 전체의 quality layer 를 구성할 수가 있다. 예를 들면, 오직 하나의 layer 만이 생성될 수 있다. 이러한 경우에는 이 bitstream 에는 스케일러빌리티 기능이 없으며 PCRD 알고리즘을 사용하여 정확하게 bit rate 를 조절할 수가 있다. 또 다른 경우에는 비교적 많은 수의 layer (최대 50 개)가 생성될 수 있다. 이런 경우에는 각 layer 가 전체 bitstream 에 대하여 부분적인 정보를 담고 있어 스케일러빌리티를 적용할 수가 있다. 그리고 EBCOT 는 많은 수의 layer 을 처리하는데 소요되는 비용을 줄이기 위하여 각 코드블록에 대한 요약(summary) 정보만을 가지고 부호화 과정을 수행하는 별개의 부호화

과정을 가지고 있다.

2. EBCOT 의 Block 부호화

Block 부호화는 기본적으로 bit-plane 부호기이다. 여기에 다음과 같은 특징들이 있다. (1) fractional bit-plane (2) 효율적인 산술 부호화를 위한 context 생성 (3) 32×32 혹은 64×64 코드블록을 16×16 서브블록으로 다시 구분. Fractional bit-plane 의 사용은 더욱 세밀한 bitstream embedding 을 위한 것이다. 서브블록의 크기는 16×16 이다. 서브블록의 사용은 서브블록 자체의 significance 를 먼저 부호화하고 다음에 서브블록이 포함하고 있는 bit 값들의 significance 를 고려하여 부호화하기 위함이다. 서브블록 significance 부호화에 대한 기본 가정은 significant bit 값들은 일반적으로 그룹을 형성하기 때문에 그룹을 대표하는 하나의 bit 값을 부호화함으로써 다른 많은 bit 값들을 부호화할 필요가 없다는 것이다. Bit-plane 부호화 방법을 사용한 이유는 가장 중요한(most significant) bit 를 먼저 부호화하고, 다음에 bit 의 중요도 순서에 따라 순차적으로 부호화하기 위함이다. 만약 bitstream 이 나중에 truncate 된다면 코드블록은 덜 중요한 bit 부터 잃게 될 것이다. Bit-plane 부호기는 ZC(Zero Coding), RLC(Run-Length Coding), SC(Sign Coding), MR(Magnitude Refinement) 모듈들을 가지고 있다. ZC 는 양자화된 bit 값을 부호화한다. ZC 는 bit 에 대하여 근접 이웃(immediate neighbor) bit 들과의 관계로부터 9 개의 context 를 구성하여 부호화한다. RLC 는 산술 부호기에 입력되는 bitstream 의 길이를 줄이기 위함이다. SC 는 bit 값들의 방향성을 부호화하기 위한 것이다. 예를 들어, LH 서브밴드에서 수평으로 이웃하고 있는 bit 값들은 동일한 sign 을 가진 것으로 간주하고 반면에 수직으로 이웃하고 있는 bit 값들은 반대의 sign 을 가진 것으로 간주한다. SC 는 이와 같은 sign 정보의 방향성을 부호화하기 위함이다. MR 은 significant bit 값들을 부호화한다. Bit 값들은 이전 bit-plane 의 bit 값이나 이웃하고 있는 bit 값들과 상관성이 약하다. MR context 는 state variable 과 바로 이전의 수평, 수직 bit 들에 관계가 있다.

3. Fractional bit-plane 과 scan order

각 bit-plane 에 대하여 4 pass 의 부호화 과정을 거친다. Bit-plane 에 대하여 4 pass 를 하는 이유는 각 코드블록들로 하여금 세밀한 bitstream 을 출력하도록 하기 위해서 이다. 각 pass 는 forward significance propagation pass, reverse significance propagation pass, magnitude refinement pass, 그리고 normalization pass 이다. Forward significance propagation pass 는 각 서브블록의 bit 값을 scan-line 순서로 검사하면서 insignificant 하거나 혹은 “선호하는 이웃(preferred neighborhood)” bit 를 가지지 못한 bit 들은 처리하지 않는다. 여기서 “선호하는 이웃” 이란 LH 와 HL 서브밴드에서는, 만약 bit 가 적어도 하나의 수평 이웃(horizontal neighbor)에 1 값을 가진다면 “선호하는 이웃”을 가진다고 한다. LL 서브밴드도 동일한 방법으로 처리한다. HH 서브밴드에서는 하나 혹은 그 이상의 대각선 bit 가 1 이라면 “선호하는 이웃”을 가진다고 한다. 이와 같은 bit 에 대하여서는 이 bit 가 현재의 bit-plane 에서 처음으로 significant 하게 된 것인지를 판단하기 위하여 ZC, RLC 를 적용해 본다. 만약 그렇다면, SC 를 적용하여 sign 을 부호화한다. 이 pass 를 부르는 이유는 significant 한 것으로 판명된 bit 는 다음의 scan 방향으로 significance 를 전파하는 역할을 하기 때문이다. Reverse significance propagation pass 는 개념적으로 전 단계의 pass 와 동일하다. 차이점은 bit 값들이 반대의 순서로 검사되고 “선호하는 이웃”이 8 개의 이웃 bit 로 확대 적용 되어 처리된다. Magnitude refinement pass 에서는 이미 이전의 두 단계에서 검사된 bit 들에 대하여 MR 을 호출하여 처리한다. Normalization pass 에서는 이전의 단계에서 처리되지 않은 least significant bit 들을 처리한다.

4. EBCOT 알고리즘 개선

이상으로 EBCOT 알고리즘에 대하여 자세하게 살펴해보았다. EBCOT 알고리즘은 이전의 알고리즘과 같이 웨이블릿 변환을 사용하고 있으며 코드블록을 도입한 것이 큰 특징이다. 코드블록은 서로 독립적으로 처리되고 있으나 서로 완전히 독립적인

것은 아니다. 코드블록들은 웨이블릿 변환의 결과로서 서브밴드를 구성하고 있기 때문에 서로 연관되어 있다. 그리고 EBCOT 알고리즘도 양자화를 수행하기 때문에 웨이블릿 계수값에 있어서 약간의 손실이 있을 수 있다. 이로 인하여 역 웨이블릿 변환을 수행하면 원래의 값과는 다른 값이 생성될 수 있다. 현재는 이러한 점들이 허용 범위에 있으며 같은 기준으로 실험한 경우 SPIHT[3] 알고리즘의 경우보다 우수한 결과를 나타내고 있다[1]. EBCOT 알고리즘의 bit-plane 부호화에 대하여서는 다음의 개선 방법을 생각해 볼 수 있다. [Least significant bit-plane 부호화] : Least significant bit-plane 에는 대부분 0 이 존재한다. 이러한 bit-plane 이 포함하고 있는 0 bit 을 코드블록 부호화 과정에서 RLC 를 사용하여 부호화할 필요는 없다고 생각된다. Least significant bit-plane 에 대하여 bitwise inclusive OR 연산을 사용하면 bit-plane 이 포함하는 1 의 개수를 알 수 있으며 bitwise AND 연산을 사용하면 1 의 위치를 알 수 있다. 이러한 방법을 사용하여 1 bit 값을 가지지 않는 bit-plane 에 대하여서는 코드블록 부호화 이전에 미리 부호화를 수행하고, 1 bit 값을 약간만 가지고 있는 bit-plane 은 1 bit 값의 위치만을 따로 부호화하고, 1 bit 값을 상당히 많이 가지고 있는 bit-plane 에 대하여서는 일반적인 코드블록 부호화 방법을 사용할 수 있다.

5. 참고문헌

- [1] D. Taubman, "High Performance Scalable Image Compression with EBCOT," IEEE Trans. on Image Processing, Vol., 9, No., 7, pp. 1158-1171, July 2000.
- [2] C. Christopoulos and A. Skodras, "JPEG2000 The next generation still image compression standard," ICIP'99, Kobe, Japan, 1999.
- [3] A. Said and W. Pearlman, "A New, Fast and Efficient Image Codec based on Set Partitioning in Hierarchical Trees," IEEE Trans. on Circuits and Systems for Video Technology, Vol., 6, No., 3, pp. 243-250, June 1996.