

고해상도 정지영상 압축을 위한 효율적인 JPEG2000용 Context 추출부의 연산 방법 연구

이성목* · 송진근* · 해주영* · 이민우** · 강봉순*

*동아대학교 · **삼성전자

The Research of Efficient Context Coding Method for compression of
High-resolution image in JPEG 2000

Sung-mok Lee* · Jin-gun Song* · Joo-young Ha* · Min-woo Lee** · Bong-soon Kang*

*Dong-A University · **SAMSUNG Electronics Co. Ltd.

E-mail : jackal13@didec.donga.ac.kr

요 약

기존의 JPEG은 낮은 bit-rate에서의 화질열화현상과 고압축에서의 블록화 영상 등의 단점에 의해 새로운 정지영상 압축 방법이 요구되었다. 이에 차세대 정지영상 표준으로 등장한 것이 JPEG2000이다. JPEG2000 표준은 DWT(Discrete Wavelet Transform)과 EBCOT Entropy Encoding 기술을 기반으로 하고 있다. EBCOT(Embedded block coding with optimized truncation)은 JPEG2000 표준에서 실제 압축을 수행하는 가장 중요한 기술 중 하나이다. 하지만 EBCOT는 bit-level 처리를 하기 때문에 대부분의 연산 시간을 차지하고 있다. 이 때문에 EBCOT의 연산속도를 높이기 위한 연구가 많이 이뤄지고 있다. 이에 본 논문은 JPEG2000 표준의 특징을 이용하여 연산 구조를 개선시킨 Context 추출 방법을 제안한다. 제안한 알고리즘은 고해상도 Multi-Component 정지영상의 압축을 위한 JPEG 2000 Encoder Hardware에 적용될 것이다.

ABSTRACT

In order to overcome many defects in the current JPEG standard of still image compression, the new JPEG2000 standard has been development. The JPEG2000 standard is based on the principles of DWT and EBCOT Entropy Coding. EBCOT(Embedded block coding with optimized truncation) is the most important technology in the latest image-coding standard, JPEG2000. However, EBCOT occupies the highest computation time to operate bit-level processing. Therefore, many researches have achieved methods to minimize computation speed of EBCOT. Thus, this paper proposes the method of context-extraction that improves computational architecture. This paper proposes efficient context coding method. The proposed algorithm would apply to hard-wired JPEG2000 Encoder that is used for compression of high resolution image.

키워드

JPEG 2000, DWT(Discrete Wavelet Transform), EBCOT, Context,

1. 서 론

멀티미디어의 발달로 영상 압축은 저장매체와 전송선로 대역폭의 한계 등의 문제를 극복하기 위한 방법으로 필수적인 것이 되었다. 고화소의 디지털 카메라의 메모리에 많은 영상을 저장하고, 유무선 통신 분야에서 방대한 데이터를 효율적으

로 압축하여 빠르게 에러 없이 전송하는 것이 중요한 이슈로 부각되고 있는 것이다. 1992년에 JPEG(Joint Photographic Expert Group)이 정지 영상 압축 표준으로 채택된 이후 이 표준은 다양한 멀티미디어 응용분야에 사용되고 있다. 그러나 JPEG은 구현이 쉽다는 장점에도 불구하고 손실과 무손실 압축이 단일화되어 있지 않고 잡음이 많

은 저 비트율 환경에서 뚜렷한 성능 열화를 보이고 있으며 DCT(Discrete Cosine Transform)을 기반으로 압축과정을 수행하기 때문에 높은 비율로 압축을 할 경우 블록화 현상이 심하게 일어나는 단점을 가지고 있다. 이러한 문제점들을 해결하기 위해 JPEG의 표준화 작업을 담당하고 있는 ISO/IEC 산하의 JTC1/SC29/WG1 그룹에서 새롭게 JPEG2000을 개발 하였다. JPEG2000은 JPEG에서 단점으로 지적되었던 저비트에서의 화질 열화를 개선하고 다양한 부가기능이 추가된 차세대 정지영상 압축 표준이다. JPEG2000은 JPEG에서 보이던 단점을 극복하고 훨씬 높은 압축 성능을 보여주지만 JPEG에 비해 연산 과정이 복잡하고 연산량이 많은 단점을 가지고 있다. 또한 JPEG 2000에서 실제적으로 압축을 수행하고 있는 엔트로피 코딩이 다른 처리과정에 비해 높은 연산량과 연산시간을 차지하고 있다. 이에 본 논문은 엔트로피 코딩 중 가장 높은 연산량을 가지는 Context 추출방법의 연산구조를 개선시켜 고효소의 정지영상 압축용 JPEG2000 Encoder에 적용하기 위한 알고리즘을 제안한다.

II. JPEG 2000의 표준 구조

JPEG2000 인코딩과 디코딩 과정의 기본 블록 다이어그램은 아래 그림 1과 같다.

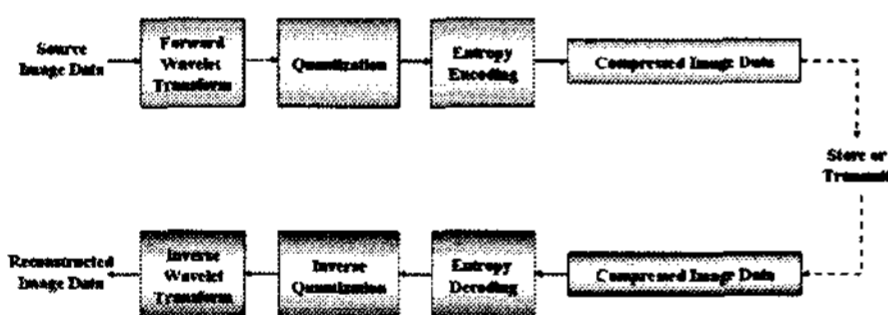


그림 5. JPEG2000의 기본 블록 다이어그램

JPEG2000의 인코딩 과정을 살펴보면 입력 영상을 이산 웨이블릿 변환을 수행하고 변환된 계수는 각 Sub-band 별로 양자화를 거친다. 이렇게 양자화된 웨이블릿 계수는 엔트로피 코딩 과정을 거친 후 최종적으로 JPEG2000 code-stream을 생성하게 된다. 디코딩은 앞에 언급한 과정의 정확한 역과정을 통해 복원된 영상을 사용자가 확인할 수 있게 된다.

JPEG2000의 연산의 복잡도는 JPEG에 비해 월등히 높은 편이다. JPEG2000 Encoding을 수행할 경우 각 처리 과정의 연산량을 표 1에 나타내었다[1]. 표 1에서 나타낸 바와 같이 JPEG2000 인코딩을 수행할 경우 가장 큰 연산 시간을 차지하고 있는 것은 EBCOT Tier-1 과정으로서 전체 연산의 절반 가량을 차지한다.

일반적인 3-component(RGB) 영상에 대한 JPEG 2000 압축 과정을 살펴보면 전체 연산량 중 웨이블릿 변환과 엔트로피 코딩부가 차지 하는 비율은 lossy 인코딩의 경우 웨이블릿 변환은 약 24%,

엔트로피 코딩부는 약 44%이다.

표 1. JPEG2000 인코딩의 주요 연산 백분율

Operation	Run Time Percentage			
	Gray Scale		Color Image	
	loseless	lossy	loseless	lossy
Color Transform	-	-	0.91	14.12
DWT	10.81	26.38	11.90	23.97
Quantization	-	6.42	-	5.04
EBCOT Tier-1	71.63	52.26	69.29	43.85
Pass 1	14.89	14.82	13.90	12.39
Pass 2	10.85	7.00	10.94	5.63
Pass 3	26.14	16.09	25.12	13.77
A.E	19.75	14.35	19.33	12.06
EBCOT Tier-2	17.56	14.95	17.90	13.01

아래와 그림 2는 JPEG2000 인코더 기본 블록 다이어그램을 나타내고 있다. 그림에서 보듯이 EBCOT는 Tier-1과 Tier-2로 구성되어 있다.

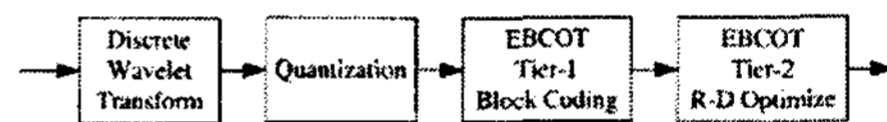


그림 6. JPEG 2000 인코더 기본 블록 다이어그램

Tier-1 블록은 웨이블릿 변환과 양자화를 거친 각 sub-band 별 양자화 계수를 EBCOT에서 처리하기 위한 최대 64X64 샘플의 code-block으로 나누며 이를 각각 sign bit-plane과 magnitude bit-plane으로 분리한다. 이것은 영상 압축을 수행 할 때 복원된 영상의 화질에 영향을 미치는 중요 인자는 MSB 쪽에 분포하고 있기 때문이다. 즉, 중요도가 높은 정보를 중심으로 압축하고 중요도가 낮은 LSB쪽은 필요에 따라 제거할 수 있는 특징을 가지고 있다. 이 분할된 bit-plane의 상태에 따라 상태값 Context와 샘플값 Decision을 추출한 후 덧셈연산과 쉬프트 연산을 사용하는 MQ-coder를 사용하여 실질적인 압축과정인 산술 부호화를 수행하게 된다. Tier-2 블록은 Tier-1에서 만들어진 압축된 bit-stream을 주어진 bit-rate에 맞게 데이터를 삭제하거나 점진적인 SNR Scalability와 Resolution Scalability가 가능하게 압축된 데이터를 재배열시킨다. 또한 Tier-2는 Tier-1을 통과한 bit-stream 내부에 File header와 같은 decoding 참조정보를 삽입하는 역할을 하고 있다.

그림 4는 JPEG 2000 EBCOT Tier-1의 스캔 패턴을 나타내고 있다. 그림에서 보는 바와 같이 분할된 코드블록의 첫 번째 열 세로 4샘플을 먼저 스캔하고 그 다음 열의 세로 4샘플을 스캔하는 형태로 코드블록의 가로크기만큼 반복 스캔 한 후 5행째로 넘어가서 동일한 패턴으로 반복 스캔 하게 된다.

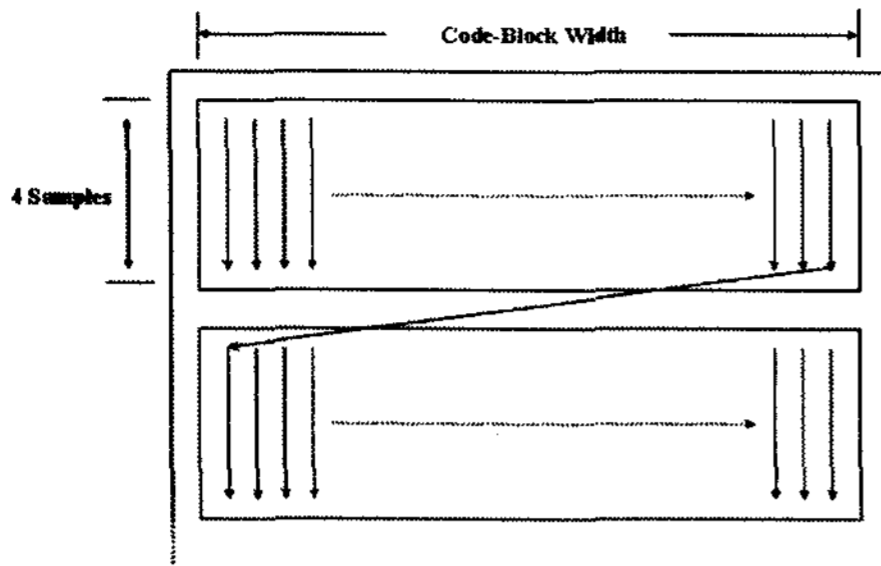


그림 7. context coder의 스캔 패턴

아래의 그림 4는 현재 샘플 X를 코딩하기 위한 주변 샘플을 나타낸 것이다. EBCOT에서는 현재 샘플과 부호비트, 주변의 상하 좌우 대각선 4개, 도합 8개의 샘플을 이용한다[2].

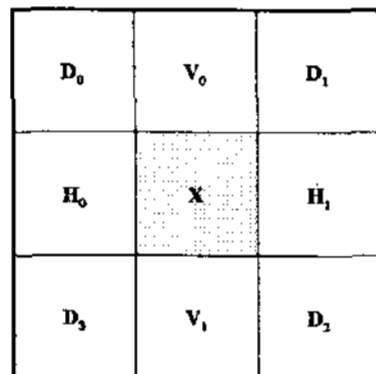


그림 8. 현재 샘플을 코딩하기 위한 주변 샘플

아래의 그림 5는 context 추출 과정의 전체적인 개요도를 나타낸 것이다. 그림 3과 그림 4에서 나타낸 스캔 패턴과 참조샘플을 이용하여 3가지의 코딩 패스(Significance Propagation Pass, Magnitude Refinement Pass, Clean-up Pass)와 4가지 코딩 오퍼레이션 ZC(Zero Coding), SC(Sign Coding), RLC(Run Length Coding), MR(Magnitude refinement)를 사용하여 총 19종류의 Context를 추출하고 그룹화하게 된다.

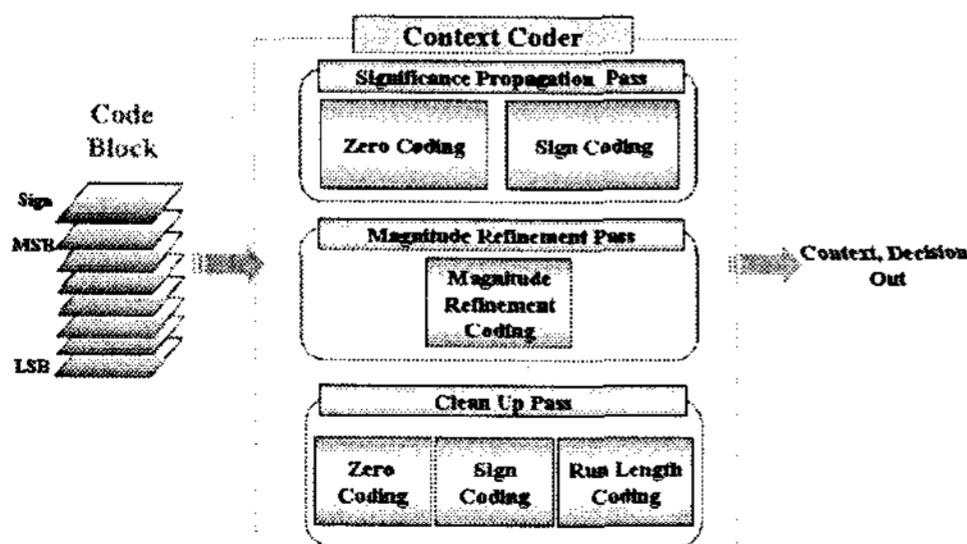


그림 9. EBCOT Tier-1 내부의 콘텍스트 추출 알고리즘의 개요도

III. 제안된 알고리즘

표준 알고리즘에서는 Sign bit-plane을 제외한 Magnitude bit-plane의 최상위 bit-plane에는 Clean-up pass만이 적용되고 코드블록 내의 모든 bit-plane은 일정한 순서로 3가지 코딩패스를 통과하며, bit-plane 내의 각각의 샘플들은 3가지 패스를 통과하는 동안 중복되지 않는 하나의 코딩 패스를 통해 코딩되게 된다. 이와 같이 JPEG2000 Tier-1은 bit-level 단위의 연산을 수행하고 하나의 샘플을 코딩 할 경우 3개의 코딩 패스를 모두 거쳐야 한다. 한 번의 코딩 과정 동안 실제로는 메모리 참조를 3번 수행해야 하므로 연산시간의 증가와 연산과정의 낭비가 많아지게 된다. 이는 하나의 샘플에 적용되는 코딩 패스는 하나의 샘플에 하나의 코딩패스만 적용되기 때문에 3번의 메모리 참조는 무의미하기 때문이다. 예를 들어 64X64 단위와 7비트의 Magnitude bit-plane을 가지는 코드블록을 연산 할 경우라면 64*64*7*3의 연산 시간이 필요하게 된다. 이와 같은 문제점은 고해상도 영상을 압축할 경우에 더욱 심각해지게 된다. 하지만 실제로 bit-plane의 하나의 샘플이 코딩되는 것은 앞에서 언급한 바와 같이 3가지 코딩 pass중 하나의 Pass만을 거쳐 코딩되게 된다. 즉, 절대 하나의 샘플에 코딩 패스가 다중으로 적용되는 일은 없다. 이에 착안한 것이 bit plane의 상태를 사전에 확인하고 코딩 패스를 결정하는 특성을 이용하여 메모리 참조 횟수를 줄이고자 하는 것이 본 논문에서 제안된 알고리즘이다. 아래의 그림 6에 제안된 알고리즘의 순서도를 나타내었다. 제안된 알고리즘은 코드블록 내의 샘플들은 한 개의 샘플이 하나의 코딩 패스만을 거치는 특징에 기반을 두었다. 또한 기존 패스 결정법이 bit-plane의 샘플값이 '1'이냐(이를 JPEG2000에서는 Significant한 상태라고 한다), '0'이냐에 따른 판단 기준을 가지고 있기 때문이다[3].

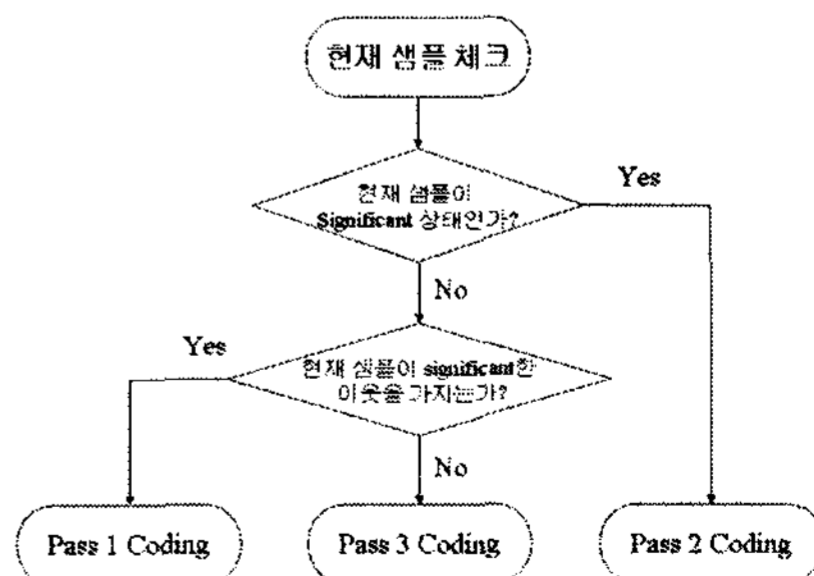


그림 10. 제안된 알고리즘의 순서도

최초 bit-plane의 현재 샘플의 상태를 체크하게 된다. 현재 샘플이 Significant한 상태 즉 '1'의 값을 가지면 2번 패스로 결정되고, 그렇지 않다면

현재 샘플 주변의 이웃 샘플들의 주변 값을 체크하여 Significant한 값이 하나라도 있다면 1번 패스, 그렇지 않고 주변 샘플값이 전부 0라면 3번 패스로 코딩되게 된다. 패스가 결정된 후 이 정보를 이용해 코드블록의 bit-plane을 한꺼번에 코딩한다. 이 같은 사전 Coding Pass 결정법은 JPEG 2000 표준알고리즘과는 달리 패스 결정 시 메모리 참조 1회와 context 추출 시 메모리 참조 1회 총 2회의 메모리참조로 Context를 추출할 수 있다.

IV. 알고리즘 비교 분석

제안된 알고리즘은 MATLAB을 통해 구현되었다. 그러므로 C와 JAVA로 구현된 표준 알고리즘과는 1:1 비교가 어려우므로 비교 분석으로 대체하였다.

본 논문에서 제안한 알고리즘을 표준의 샘플 기반의 CX 추출법을 사용한 것과 비교 분석하였다. 아래의 그림 7은 8X8의 코드 블록을 코딩하게 되면 아래와 같은 형태로 기본 3회의 스캔을 거쳐 Context와 Decision을 추출하게 된다.

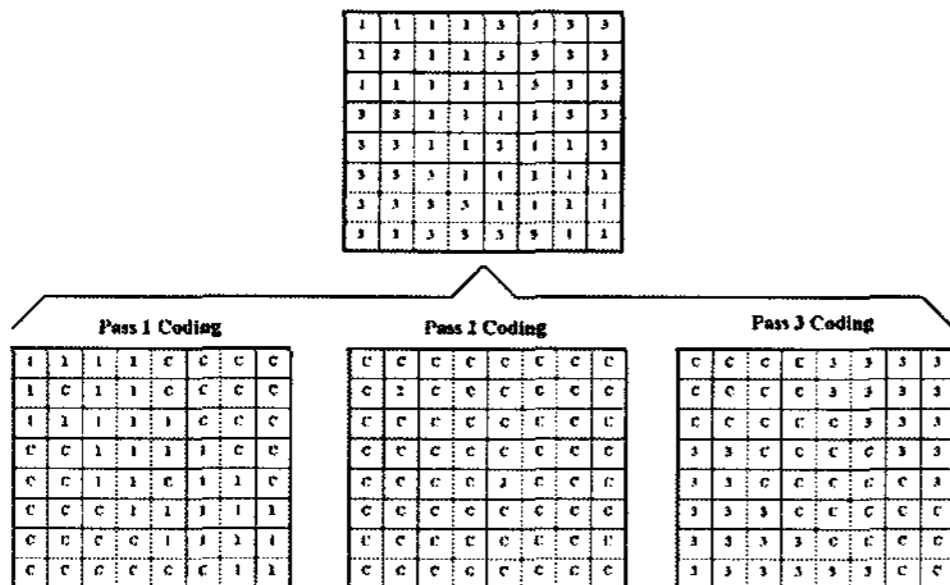


그림 11. 기존 알고리즘의 코딩 예시

이를 하드웨어로 구현한다면 메모리 참조회수가 늘어나고 클럭 소모가 늘어나 연산시간이 늘어날 수 밖에 없는 구조가 될 것이다.

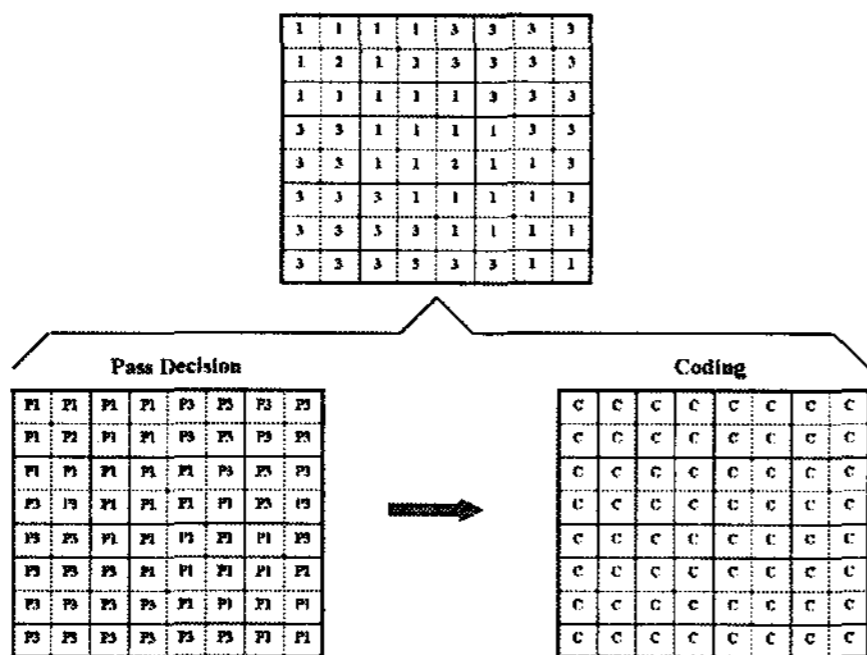


그림 12. 제안된 알고리즘의 코딩 예시

그림 8은 본 논문에서 제안된 방법의 코딩 예시이다. 제안된 알고리즘은 기존 알고리즘에 비해 약 33프로이상의 시간을 절약할 수 있게 된다. 그림에서 보듯이 사전에 코딩할 pass를 결정하고 이 정보를 이용하여 샘플을 한꺼번에 코딩한다. EBCOT Tier-1 블록의 기본처리 단위는 최대 64X64 크기로 규정되어 있다. 이는 하나의 코드블록이 손실되더라도 데이터에 미치는 영향을 줄이기 위한 것이다. 하지만 고해상도의 압축이 요구될 경우 해상도가 높아질수록 코드블록의 숫자는 늘어나게 된다. 기존 방식을 사용하게 되면 실제 연산에 쓰이지 않는 낭비 시간이 누적되게 되고 이는 전체 시스템의 처리속도에 영향을 미치게 된다. 이를 실제 하드웨어로 구현할 경우 메모리 참조 지연이 2~3 클럭이 소요되고 고해상도 영상을 압축할 경우 고용량의 외부 버퍼 메모리를 사용해야 하는데 이같은 경우는 더욱더 많은 메모리 참조 지연시간이 소모되게 된다. 즉 제안된 알고리즘은 이러한 낭비시간을 줄이므로 구현성 측면에서도 이득을 얻게 된다.

V. 결론

본 논문은 JPEG2000 엔트로피 코딩 알고리즘 중 가장 큰 연산량을 가지고 있는 context 추출부의 연산 방법의 개선에 관한 것이다. 기존의 방식은 하나의 샘플에 대해 모든 코딩패스를 적용하여 코딩을 수행하지만 본 논문에서는 코딩 패스의 사전판단을 통해 연산시간의 낭비를 줄였다. 제안된 context 추출 방법은 실제 하드웨어로 구현할 것이며 3메가 이상의 고해상도 영상의 압축에 사용될 JPEG 2000 Encoder에 적용하게 될 것이다. 이는 다양화되는 멀티미디어 기기에 실제 적용될 수 있을 것이다. 차후 다른 연산부와 연동하여 연산 속도를 더 고속화하고 하드웨어 최적화를 위한 알고리즘이 추가적으로 연구되어야 할 것이다.

참고문헌

- [1] C. Lian, "Analysis and Architecture Design of Block-Coding Engine for EBCOT in JPEG 2000", IEEE Transaction on Circuits and Systems for Video Technology, Vol. 13, No. 3, March 2003.
- [2] M. lee, J. Kim, B. Kang, "Hardware Implementation of EBCOT coder with ISO/IEC 15444-1 JPEG2000 standard", IT-SoC 2005 Conference, pp.675~ pp.678, March 2003.
- [3] T. Acharya, P. Tsai "JPEG2000 Standard for Image Compression Concepts, algorithms and VLSI architectures", John Wiley & Sons Inc, 2004.