

가변길이 부호어 테이블의 재구성을 통한 효율적인 Coeff-Token 복호화 방식

정회원 문 용 호*

A New Coeff-Token Decoding Method based on the Reconstructed Variable Length Code Table

Yong-ho Moon* *Regular Member*

요 약

일반적으로, H.264/AVC 압축 표준에서 CAVLC 복원은 많은 양의 메모리 액세스를 필요로 한다. 이러한 메모리 액세스는 비디오폰 및 DMB와 같은 응용 서비스에 있어서 파워 소모 면에서 중요한 문제로 인식된다. 이를 해결하기 위하여 본 논문에서는 CAVLC의 구문요소중 하나인 coeff-token에 대한 효율적인 복호화 방식을 제안한다. 먼저 중복 액세스를 제거하기 위하여 바이트 단위로 구성되는 새로운 코드워드를 정의한다. 그리고 이를 기초로 가변길이 부호어 테이블을 재구성한 후 효과적인 메모리 액세스 기법을 제시한다. 모의 실험은 화질 저하 없이 제안 방식에서 약 85% 정도의 메모리 액세스가 절약됨을 보여준다.

Key Word : Coeff-Token, CAVLC, H.264/AVC, VLCT, VLSI

ABSTRACT

In general, a large amount of the memory accesses are required for the CAVLC decoding in H.264/AVC. It is a serious problem for the applications such as a DMB and videophone services because the considerable power is consumed for accessing the memory. In order to solve this problem, we propose an efficient decoding method for the coeff-token which is one of the syntax elements of CAVLC. In this paper, the variable length code table is re-designed with the new codewords which are defined by investigating the architecture of the conventional codeword for the coeff_token element. A new coeff_token decoding method is developed based on the suggested table. The simulation results show that the proposed algorithm achieves an approximately 85% memory access saving without video-quality degradation, compared to the conventional CAVLC decoding.

I. 서 론

다양한 멀티미디어 응용 서비스들의 출현과 발전은 높은 압축 효율을 전제로 하고 있다. ITU-T/ISO/IEC Joint Video Team에서 H.264/AVC라 명명된 새로운 동영상 표준안을 발표하였다.^[1] 이것은

기존 압축 표준 방식에 비하여 상당한 성능개선을 가져왔다. 그러나 H.264/AVC에서 채택된 새로운 특징과 기능들은 부호화뿐 아니라 복호화에서도 계산량의 증가를 야기하였다. 계산량의 증가는 H.264/AVC를 기반으로 하는 각종 서비스의 개발 비용 증가를 야기한다^[3]. 따라서 최근 H.264/AVC 복호

※ 이 논문은 2005학년도 부산외국어대학교 학술연구 조성비에 의하여 연구되었음.

* 부산외국어대학교 IT대학 디지털미디어학부 (yhmoon5@pufs.ac.kr)

논문번호 : KICS2006-08-355, 접수일자 : 2006년 8월 23일, 최종논문접수일자 : 2007년 3월 7일

화기의 효율적인 구현에 대한 관심이 크게 증가하고 있다³⁻⁵⁾.

그 중에서도 CAVLC(Context-based Adaptive Variable Length Coding) 복호화기에서 양자화 된 DCT 계수를 효과적으로 복원하기 위한 연구가 활발히 진행되고 있다. H.264/AVC의 baseline 프로파일에서 4x4 블록의 계수는 5개의 구문 요소(syntax element)들에 의해서 부호화, 복호화되어 진다. Di⁶⁾는 조합 논리로 구성되는 level 구문요소에 대한 복호화기를 제안하였다. Chang⁷⁾은 하드웨어 비용과 파워 소모를 감소시킬 수 있는 몇몇 기술들을 이용하여 CAVLC 복호화기 구조를 VLSI로 설계하였다. 이 구조에서는 H.264/AVC 표준에서 제시된 가변길이 부호어 테이블(VLCT)이 사용되기 때문에 많은 메모리 액세스가 수행된다. 한편 최근에 FSM(Finite State Machine) 기법을 이용하여 VLCT를 이용하지 않고 run-before 요소를 복원하는 새로운 복호화 방식이 제안되었다. 그러나 여전히 coeff-token, total zero의 복원을 위해서는 VLCT이용한 메모리 액세스가 요구된다.

일반적으로, VLCT를 이용한 복원 방식은 비트스 트림과 일치하는 코드워드를 발견할 때까지 재귀적으로 메모리를 액세스한다. 그러므로 VLCT 기반의 복원화기에서는 다수의 메모리 액세스가 이루어진다. 메모리 액세스는 시스템에서 파워 소모에 있어서 중요한 요소 중의 하나라는 것은 잘 알려진 사실이다. 특히 비디오폰이나 DMB와 같은 멀티미디어 서비스에서 파워 소모는 매우 중요한 고려 사항이다. 게다가 메모리 액세스는 메모리 주소를 생성하기 위한 부가 계산뿐만 아니라 시스템의 동작 속도에도 악영향을 끼친다. 따라서 메모리 액세스를 감소시키기 위한 새로운 복원 방식의 개발이 필요하다.

본 논문에서는 VLCT의 재구성을 통한 효율적인 coeff-token 복호화 방식을 제안한다. coeff-token에 사용되는 기존 코드워드의 구조 분석을 통하여 메모리의 중복 액세스를 방지할 수 있는 새로운 코드워드를 정의하고 이들로 구성되는 새로운 VLCT를 제시한다. 아울러 효율적인 메모리 구조와 복원 방식을 제시한다. 모의실험 결과는 화질 저하 없이 메모리 액세스 횟수가 크게 감소함을 보여준다.

II. H.264/AVC의 엔트로피 부호화

2.1 Exp-Golomb 부호화

Expo-Golomb(EG) codes⁹⁾는 움직임 벡터 차이,

매크로블록 타입과 같은 정보들의 부호화에 사용된다. 일반적으로 EG code는 [M_zeros][1][INFO]와 같이 구조를 지니고 있다. 여기서 M_zeros는 처음 [1]이 나오기 전까지의 0의 수를 가지는 사전코드(prefix code)이고, INFO는 처음 [1] 뒤의 정보를 가지는 사후코드(suffix code)이다. 예를 들어 코드 워드가 000010110인 경우, M_zeros는 0000 이고 INFO는 0110이다. 이 예에서도 볼 수 있듯이, M_zero와 INFO의 길이는 4로서 동일하다. 이 같은 구조적 규칙성은 EG code의 부호화 과정을 단순화시킨다.

그런데 EG code의 이 같은 규칙성은 M_zero 길이와 INFO의 길이간의 관계를 미리 정의함으로써 쉽게 일반화될 수 있다. 본 논문에서는 M_zeros와 INFO의 길이를 각각 PCL과 SCL로 정의한다. 위의 예제의 경우에서 PCL과 SCL은 모두 4로 동일하지만 SCL과 PCL이 항상 동일할 필요는 없다. 사전에 SCL과 PCL간의 관계가 약속되어 있기만 하면 복원에 문제가 야기되지 않는다.

2.2 CAVLC

H.264/AVC baseline 프로파일에서 4x4 블록에 대한 양자화된 계수들은 지그재그 순으로 재배열된 후 5개의 구문 요소들에 의해 부호화된다.

- coeff_token : '0'이 아닌 계수의 수(Tc)와 고주파 성분에서 크기가 1인 계수의 수(T1s)
- sign of T1s : 역 지그재그 순서에서의 T1s 부호
- level : T1s을 제외한 '0'이 아닌 계수의 크기
- total_zeros : 지그재그 순서에서 DC와 '0'이 아닌 마지막 계수간에 존재하는 크기가 '0'인 계수들의 수
- run-before : 역 지그재그 순으로 각각의 '0'이 아닌 계수 앞에 존재하는 '0'인 계수들의 수

coeff-token 구문요소를 부호화하기 위하여 압축할 때 1개의 고정길이 코드 테이블과 3개의 VLCT가 사용된다. 고정길이 코드 테이블에 존재하는 코드워드는 구조가 매우 규칙적이다. 그러나 VLCT의 경우 비교적 불규칙적이고 복잡한 구조의 코드워드로 구성되어 있다. 따라서 VLCT는 메모리를 사용하여 구현된다. 표 1은 VLCT들을 보여준다. 테이블의 선택은 이전에 복원된 이웃 블록의 Tc값으로부터 이루어진다. 표 1에서 각 코드워드는 특정 Tc와 T1s의 쌍으로 이루어져 있음을 볼 수 있다.

5개의 구문 요소들의 복원은 압축되어 전송되는

표 1. coeff-token에 대한 3개의 VLCT

Tc \ T1s	0	1	2	3
0	1	-	-	-
1	000101	01	-	-
2	00000111	000100	001	-
3	000000111	00000110	0000101	00011
4	0000000111	000000110	00000101	000011
5	00000000111	0000000110	000000101	0000100
⋮	⋮	⋮	⋮	⋮
15	0000000000000111	0000000000001010	00000000000001001	00000000000001100
16	0000000000000100	0000000000000110	0000000000000101	00000000000001000

(a) VLCT0 (0 ≤ nC < 2)

Tc \ T1s	0	1	2	3
0	11	-	-	-
1	001011	10	-	-
2	000111	00111	011	-
3	0000111	001010	001001	0101
4	00000111	000110	000101	0100
5	00000100	0000110	0000101	00110
⋮	⋮	⋮	⋮	⋮
15	000000000001001	000000000001000	000000000001010	00000000000001
16	000000000000111	000000000000110	000000000000101	000000000000100

(b) VLCT1 (2 ≤ nC < 4)

Tc \ T1s	0	1	2	3
0	11	-	-	-
1	001011	10	-	-
2	000111	00111	011	-
3	0000111	001010	001001	0101
4	00000111	000110	000101	0100
5	00000100	0000110	0000101	00110
⋮	⋮	⋮	⋮	⋮
15	000000000001001	000000000001000	000000000001010	000000000000001
16	000000000000111	000000000000110	000000000000101	000000000000100

(c) VLCT2 (4 ≤ nC < 8)

비트스트림으로부터 이루어진다. 이 중에서 sign of T1s, level, run-before 구문 요소는 산술 연산만으로 복원될 수 있다. 그러나 coeff-token 요소는 여전히 표 1과 같은 VLCT를 사용된다. 일반적으로 VLCT의 코드워드는 메모리에 저장되어 있다. 비트스트림이 주어지면, 스트림과 일치되는 코드워드를 찾기 위해 메모리를 반복적으로 액세스하여 코드워드와 비트스트림을 비교한다. 만약 일치된 코드워드가 찾아지면 메모리 주소로부터 Tc와 T1s를 결정한다. 그러나 일치된 코드워드가 없을 경우, 새로운 입력 비트스트림에 대하여 이러한 과정을 반복한다. 따라서 VLCT를 이용한 기존 복호화 방식은 많은

메모리 액세스를 필요로 한다. 그리고 메모리 주소 생성을 위한 추가적인 계산량과 메모리 액세스에 소요되는 지연시간은 시스템의 속도를 감소시킨다. 더구나 표 1의 코드워드 길이가 1~16비트인 점을 감안할 때 표 1은 구현에 있어서 메모리 낭비를 야기한다. 왜냐하면 기본적으로 메모리는 바이트 단위로 구성되고 액세스되기 때문이다. 따라서 메모리 낭비를 제거하고 과도한 메모리 액세스를 감소시킬 수 있는 효율적인 복호화 방식이 필요하다.

III. 제안하는 복호화 방식

3.1 기존 VLCT의 특성

효율적인 coeff-token 복호화 방식의 개발을 위해서 본 논문에서는 먼저 기존 VLCT의 특성을 조사하였다.

3.1.1 코드워드 정렬

표 1에 존재하는 coeff-token의 코드워드는 Tc와 T1s에 의하여 2차원 형식으로 정의되어 있다. 그러나 메모리는 물리적으로 1차원 구조를 지닌다. 그러므로 표1의 효율적인 구현을 위해서는 코드워드의 재배열이 필요하다. 표1의 VLCT를 보면 우하단보다 좌상단의 코드워드가 대체로 짧은 비트 길이를 지니고 있음을 알 수 있다. Kato^[10]는 이러한 현상에 기초하여 (Tc, T1s)쌍에 하나의 codenum을 대응시키는 테이블을 제시하였다. 그런데 이 방식을 이용하기 위해서는 여전히 메모리 액세스가 요구되는 제약이 존재한다.

본 논문에서는 메모리 액세스를 제거하기 위하여 Kato의 매핑 방식을 변경하였다. 표 2는 이를 보여 준다. 표 2에서 (Tc, T1s)쌍과 codenum을 살펴보면 매우 규칙적인 관계가 존재하며 이것은 간단한 수식으로 표현될 수 있음을 알 수 있다. 부호기에서 Tc와 T1s이 제시될 경우 대응하는 codenum은 식 (1)과 표현될 수 있다.

$$codenum = (Tc \ll 2) - 3 \cdot T1s \quad (1)$$

그리고 만약 codenum이 주어질 경우 Tc와 T1s는 식 (2)와 같이 쉽게 구해진다.

$$\begin{aligned} T1s &= codenum \% 4 \\ Tc &= (codenum + 3 \cdot T1s) \gg 2 \end{aligned} \quad (2)$$

식 (2)에서 %는 modular 연산을 의미한다. 위의 수

식들은 코드워드가 1차원으로 손쉽게 정렬될 수 있음을 보여준다. 또한 특정 코드워드의 메모리 주소를 codenum으로 설정할 경우 Tc와 TIs가 메모리 주소부터 직접 얻어진다는 것을 알 수 있다.

3.1.2 코드워드 분류

표 1의 코드워드는 기본적으로 EG code의 구조를 지니고 있음을 알 수 있다. 따라서 표 1의 코드워드를 EG code형식으로 분석해 보면, PCL이 0~14, SCL이 0~3의 값을 지니고 있음을 알 수 있다. 본 논문에서는 이러한 사실에 착안하여 표 1의 코드워드들을 PCL의 값에 따라 분류하였다. 표 3은 3개의 VLCT에 존재하는 코드워드들의 분류를 codenum으로 표현한 것이다. 표 3에서 밑줄 친 codenum은 각 그룹에서의 초기 codenum(iniNum)을 나타낸다.

표 2. VLCT에서 (Tc, TIs)쌍에 대한 codenum 매핑 규칙

Tc \ TIs	0	1	2	3
0	0	-	-	-
1	4	1	-	-
2	8	5	2	-
3	12	9	6	3
4	16	13	10	7
5	20	17	14	11
6	24	21	18	15
7	28	25	22	19
8	32	29	26	23
9	36	33	30	27
10	40	37	34	31
11	44	41	38	35
12	48	45	42	39
13	52	49	46	43
14	56	53	50	47
15	60	57	54	51
16	64	61	58	55

한편, 표 1과 표 3에서 다음과 같은 특징들을 발견할 수 있다.

- 1) 주어진 PCL이 주어질 경우, 기존 복호화 과정은 해당 그룹에 존재하는 코드워드들로 제한할 수 있다. 코드워드는 PCL로 분류되었기 때문에 그룹에서의 코드워드들을 비트 스트림과 비교하는 것으로 충분하다.
- 2) 동일 그룹내의 코드워드들은 INFO에 의해서 완벽히 구분된다. 그러므로 PCL이 구해진 경우, 코드워드의 INFO에 의해 일치되는 코드워드를 찾는 과정을 수행할 수 있다.
- 3) 일반적으로, 특정 그룹내의 코드워드들에서 SCL값들이 동일하지 않다.

표 1에서 SCL이 6 종류(0, 1, 2, 3, 1 or 2, 2 or 3)임을 알 수 있다. 그러므로 SCL은 INFO 비교를 위해서 반드시 제공되어야 한다. 4) 특정 그룹에서 codenum은 iniNum으로부터 증가하고, 그 증가량(INC)은 최대 6이다.

위의 특징들은 2가지 요점으로 정리할 수 있다. - iniNum과 INC을 이용한다면 메모리 액세스의 중복을 제거할 수 있다. - 기존 복호화 방식은 PCL을 구한 후 SCL에 따라 코드워드의 INFO를 비트스트림과 비교하는 것과 동일하다.

표 3. VLCT에서 PCL에 따른 코드워드 분류

Table PCL	VLCT0	VLCT1	VLCT2
0	0	0,1	0,1,2,3,7,11,15,19
1	1	2, 3, 7	5,6,9,10,13,14,17,23
2	2	4,5,6,9,11,15	4,8,12,18,21,22,25,27
3	3,4,5	8,10,13,19	16,20,24,26,28,29,30,31
4	6,7,11	12,14,17,23	32,33,34,35,36,37,38,39
5	8,9,10,15	16,18,20,21	40,41,42,43,44,45,46,48
6	12,13,14,19	22,24,25,27	47,49,50,52,53,56,57
7	16,17,18,23	26,28,29,30,31,32,33,35	51,54,60,61
8	20,21,22,27	34,36,37,38,39,40,41,44	55,58
9	24,25,26,28,29,30,31,32	42,43,45,46,47,48,49,52	64
10	33,34,35,36,37,38,39,40	50,53,54,56,57,60	-
11	41,42,43,44,45,46,47,48	55,58,61,64	-
12	50,51,52,53,54,55,56,57	51	-
13	58,,60,61,64	-	-
14	49	-	-

3.2 코드워드의 재정의 및 VLCT의 재구성

본 논문에서는 위에서 언급한 특징들을 기초로 하여 기존 VLCT를 MBASE와 MCODE라는 테이블로 재구성하고 효율적인 메모리 구조를 제안한다. MBASE는 앞에서 분류한 그룹의 iniNum과 SCL 유형 정보를 저장하고 있으며 MCODE는 특정 그룹에 속하는 코드워드의 INFO 비교를 위한 정보들을 저장하고 있다. 이들 테이블의 구성과 역할에 대한 상세한 내용은 다음과 같다.

비트스트림으로부터 PCL이 결정되면 이를 주소로 하여 MBASE가 역세스된다. MBASE는 6비트의 iniNum과 2비트의 TP1으로 구성된다. 앞에서 설명한 바와 같이 iniNum은 특정 그룹에 포함되는 코드워드들에 대한 초기 MCODE 주소이다. 그리고 TP1은 SCL의 기본 유형 정보를 제공하는 것으로 그 의미와 정의는 표 4와 같다. 표 4에서 TP1=11인 경우는 INFO가 존재하지 않은 경우를 의미하는 것으로 MCODE의 액세스없이 식(2)로부터 Tc와 T1s가 결정된다.

표 4. SCL에 대한 TP0와 TP1의 정의

특정 그룹에서의 SCL값의 종류	TP1	TP0	TP1의 의미
0	11	-	SCL=0
1	00	01	코드워드의 SCL이 동일
2		10	
3		11	
1 / 2	01	00 / 01	코드워드의 SCL이 1 또는 2
2 / 3	10	00 / 01	코드워드의 SCL이 2 또는 3

특정 그룹에 대한 INFO 비교는 MBASE에서 제공되는 iniNum을 초기 주소로 하여 시작된다. 앞에서 설명한 바와 같이 불필요한 중복 액세스없이 특정 그룹에서의 검색을 위해서는 INFO, SCL, INC가 제공되어야 한다. 이러한 이유로 MCODE는 [TP0][INFO][INC]의 형식으로 정의된 코드워드들로 구성된다. 표 5는 VLCT0에 있어서 MCODE 구조의 한 예를 보여준다.

표 5. VLCT0에 대하여 MCODE에 저장된 코드워드의 예

코드워드 codenum	TP0	INFO	INC
⋮	⋮	⋮	⋮
3	00	1xx	001
4	01	01x	001
5	01	00x	000
6	01	01x	001
7	00	1xx	100
8	10	11x	001
9	10	10x	001
⋮	⋮	⋮	⋮

표 5에서 INFO의 비트 길이, SCL은 TP0와 TP1의 합으로 쉽게 결정할 수 있다. 이것은 표 4로부터

쉽게 확인할 수 있다. 그리고 표 5에서 다음에 비교할 INFO의 메모리 주소는 현재 주소+INC에 의하여 결정된다. 따라서 INC가 0인 경우는 더 이상의 INFO비교가 필요하지 않다는 것을 의미한다.

3.3 제안하는 복호화 방식

제안하는 coeff-token의 복호화 방식은 다음과 같은 과정으로 구성된다.

- Step 1) 주위 블록으로부터 얻어진 nC로부터 VLCT를 결정한다.
- Step 2) 입력되는 비트 스트림으로부터 PCL을 결정한다.
- Step 3) PCL을 주소로 하여 MBASE를 액세스한다.
- Step 4) TP1이 11인 경우 PCL을 식(2)에 대입하여 T1s와 Tc를 구한다.
- Step 5) TP1이 11이 아닐 경우, iniNum을 주소로 하여 MCODE를 액세스한다.
- Step 6) MCODE의 INC, TP0을 이용하여 비트스트림과 동일한 INFO를 검색하고 일치하는 메모리 주소를 구한다.
- Step 7) 메모리 주소를 식(2)에 대입하여 T1s와 Tc를 구한다.

IV. 실험 결과 및 고찰

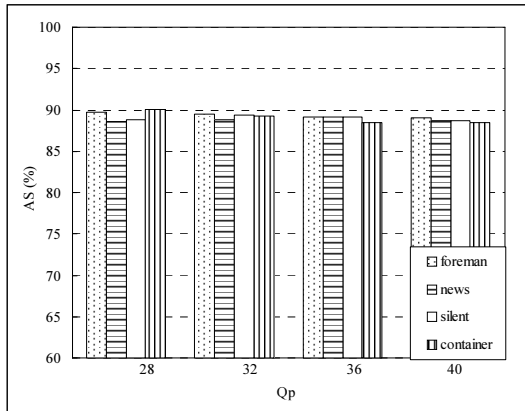
본 논문에서는 QCIF(176x144) 테스트 영상을 사용하여 제안 방식을 검증하였다. 탐색 범위는 16으로 설정하였고 5장의 예측 프레임을 적용하였다. R-D 최적화 기법은 압축된 테스트 영상에 대해 적용하였다. 그 외 다른 조건들은 H.264/AVC baseline 프로파일에 기반을 두고 있다.

제안 방식은 다양한 QP에 대해서 H.264/AVC 참조 software^[11] 복호화기와 비교하여 그 성능을 검증하였다. 그림 2는 이를 보여준다. 그림 2에서 AS는 제안된 방식에서의 메모리 액세스 이득을 의미하며 그 정의는 식 (4)와 같다.

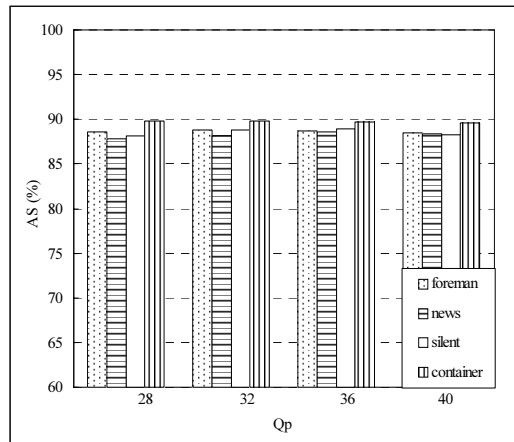
$$AS = (1 - \frac{\text{제안방식의 메모리 액세스수}}{M\text{에서의 메모리 액세스총수}}) \times 100 \quad (4)$$

그림 2는 제안 방식이 coeff-token의 복원에 있어서 메모리 액세스 수는 기존보다 약 85% 정도 절약했음을 보여준다.

그런데 기존 H.264/AVC 부호화기는 복호화 과정이 최적화되지 않은 것이다. 따라서 보다 더 명확한 성능 검증을 위해서 제안 방식에서의 평균 메모리 액세스를 조사하는 것이 필요하다. 표 6은 10Hz의 테스트 영상들에 있어서 4x4 블록당 평균 메모리 액세스를 보여준다. 표 6으로부터, 평균 메모리 액세스가 1정도임을 알 수 있다. 이것은 제안 방식에서 중복 메모리 액세스가 효율적으로 제거됨을 보여준다. 따라서 제안 방식은 복원 영상의 성능과 화질의 저하없이 파워 소모를 효율적으로 감소시킨다.



(a) 10 Hz



(b) 30 Hz

그림 2. 제안하는 복호화 방식에서의 메모리 액세스 이득

표 6. 제안 방식에서의 4x4 블록당 평균 메모리 액세스

Seq.	Foreman	News	Silent	Container
28	1.1	1.32	1.17	0.99
32	0.89	1.17	0.92	0.95
36	0.74	0.95	0.74	0.88
40	0.65	0.78	0.64	0.78

V. 결론

기존 CAVLC 복호화는 많은 메모리 액세스로 인하여 파워가 많이 소모된다는 문제를 지니고 있다. 특히, DMB나 비디오폰과 같은 서비스에 있어서 파워 소모는 매우 중요한 요소로 인식되고 있다. 따라서 파워 소모를 감소시킬 수 있는 효율적인 복호화 방식의 개발이 필요하다. 본 논문에서는 새로운 coeff-token 복원을 위한 새로운 VLCT와 효율적인 메모리 구조를 제안하였다. 기존 코드워드의 특징을 기반으로 새로운 코드워드 구조 및 VLCT 효율적인 메모리 구조를 설계하였다. 모의실험 결과는 제안 방식이 기존 방식보다 약 85% 정도 메모리 액세스 이득을 지니고 있음을 보여준다. 이것은 제안 방식이 H.264/AVC baseline 프로파일에서 메모리 액세스의 중복을 효율적으로 제거한다는 것을 의미한다. 또한 제안 방식에서 사용되는 코드워드는 1바이트로 구성되기 때문에 불필요한 메모리 낭비를 가져오지 않는 장점을 지닌다.

참고 문헌

- [1] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Joint Video Specification (ITU-T Rec. H.264|ISO/IEC 14496-10)-Joint Committee Draft," Doc. JVT-G050r1.doc, Dec. 2002.
- [2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 13, no.7 pp.560-576, July 2003.
- [3] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC baseline profile decoder complexity analysis," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 13, no.7 pp.704-716, July 2003.
- [4] V. Lappalainen, A. Hallapuro, and T. D. Hämäläinen, "Complexity of optimized H.26L video decoder," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 13, no.7 pp.717-725, July 2003.
- [5] Q. Peng and J. Jing, "H.264 Codec system-on-chip design and verification," *5th IEEE International Conference on ASIC*, vol. 2, pp.

- 922-925, Oct. 2003.
- [6] W. Di, G. Wen, H. Mingzeng, and J. Zhenzhou, "A VLSI architecture design of CAVLC decoder," *5th IEEE International Conference on ASIC*, vol. 2, pp. 962-965, Oct. 2003.
- [7] H. - C. Chang, C.-C. Lin, and J.-I. Guo, "A novel low-cost high-performance VLSI architecture for MPEG-4 AVC/H.264 CAVLC decoding," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 6110-6113, 2005.
- [8] Y. H. Moon, G. Y. Kim, and J.H. Kim, "An efficient decoding of CAVLC in H.264/AVC video coding standard," *IEEE Trans. on Consumer Electronics*, vol. 51, no. 3, pp. 933-938, Aug. 2005.
- [9] Iain E.G. Richardson, *H.264 and MPEG-4 Video Compression - video coding for next generation multimedia*, John Wiley & Sons, 2003, pp. 187-207.
- [10] S. Kato, K. Sugimoto, S. Adachi, and M. Etoh, "Structured truncated Golomb code for context-based adaptive VLC," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, pp. 405-408, Apr. 2003.
- [11] K. Sühring, Ed., (2006)JVT Reference Software JM 10.1, [Online]. Available: <http://bs.hhi.de/~suehring/tml>

문 용 호 (Yong-ho Moon)

정회원

한국통신학회 논문지 제31권 11C호 참조

<관심분야> 동영상 압축, 영상처리, 디지털시스템 설계 및 VLSI