

MPEG-4 Shape Decoder의 적합성 검사

정회원 황혜전*, 박인수**, 박수현*, 이병욱*

Conformance Test for MPEG-4 Shape Decoders

Hea-Juen Hwang*, In-Su Park**, Su-Hyun Park*, Byung-Uk Lee* *Regular Members*

요 약

MPEG-4 영상압축 방식은 객체지향적이다. 기존의 H.261, MPEG-1, MPEG-2에서 화면 전체를 부호화 해주는 것과 달리 MPEG-4에서는 한 화면을 물체, 배경 등 여러 객체들로 나누어 각각의 객체를 따로 부호화한다. 이렇게 전송된 객체들은 수신 단말기에서 다시 합성하여 완성된 화면으로 보여준다. MPEG-4의 형상 부호화는 화면에서 visual 객체에 해당하는 부분만을 부호화 해주는 부분이다. 본 논문에서는 MPEG-4 디코더가 형상 부호화의 표준 규격을 따르는지를 시험하는 방법을 제안한다. 본 논문은 BAB type을 결정하는 mode 부호화의 모든 경우와 Context-based Arithmetic 부호화의 Context computation에 대한 모든 경우에 대해 확인하는 방법을 제안하였다.

ABSTRACT

MPEG-4 visual coding is an object-based system. The current video coding standards, H.261, MPEG-1, and MPEG-2, encode frame by frame. On the other hand, MPEG-4 separately encodes several objects, such as video objects and audio objects, in the same frame. Each transmitted object is decoded and composed in one frame. Shape coding is a process of coding visual objects in a frame. In this paper we present conformance test method for MPEG-4 shape decoders. This paper reviews the basic shape decoding standard, and proposes conformance test methods for BAB type decoder, and CAE decoder for intra and inter VOPs. Our test generates all possible cases of shape motion vector difference and context.

I. 서 론

MPEG-4 부호화 방식은 기존의 MPEG-2까지의 개념과 달리 실시간 단말기에서 객체들을 받아서 합성할 수 있는 객체지향적인 시스템이다. 기존의 H.261, MPEG-1, MPEG-2의 영상 전송방식은 화면을 프레임별로 전송하기 때문에 보내야 할 데이터 양이 방대했다. 그러나 MPEG-4의 경우 각 화면을 비디오 객체, 오디오 객체 등 여러 가지 객체들로 나누어 각 객체별로 부호화하여 송신하게된다. 이렇게 송신한 각각의 데이터들은 수신기에서 다시 합성하여 화면을 구성한다. 그 결과 보내야 할 데이터

가 나누어짐으로써 한번 보낸 객체를 다시 전송하지 않아도 되므로 지연시간이 줄어들게 된다.

형상부호화(shape coding)는 전송할 객체의 형상을 부호화한다. 즉, 텍스처 영상에서 객체에 해당하는 부분을 alpha plane으로 masking해주어서 객체에 해당하는 부분만을 부호화한다. MPEG-4에서 형상정보는 alpha plane으로 표현한다. alpha plane은 이진(binary) 또는 gray scale 값으로 표현될 수 있다. 이진 alpha plane은 CAE (Context-based Arithmetic Encoding)에 의해 부호화되고 gray scale alpha plane은 텍스처 부호화와 유사한 움직임보상 DCT에 의해 부호화된다.

MPEG-4의 형상부호화에서는 Binary Alpha

* 이화여자대학교 정보통신학과 (bulee@mm.ewha.ac.kr), ** 서강대학교 전자공학과
논문번호 : 00070-0222, 접수일자 : 2000년 2월 22일

※ 본 연구는 현대전자산업주식회사의 연구지원에 의한 결과입니다.

Block(BAB)이 자료구조로 사용되며 주된 복호화 방법으로 block-based context-based binary arithmetic 복호화와 블록기반 움직임보상이 있다. 각 BAB은 16 화소 × 16 화소 크기의 사각형 블록으로 7가지 종류가 있다. BAB 종류에 따라 추가 변경되는 (update) 정보가 있는 경우 CAE (Context-based Arithmetic Encoding)가 사용된다. 본 논문에서는 7가지 BAB type을 결정하는 mode에 대한 시험방식과 각 CAE의 context 수에 대한 모든 경우의 수를 시험하는 방식을 제안하였다.

지금까지 MPEG-4 디코더 시험에 관한 연구는 거의 없으나, MPEG-2 표준에서 유사한 시험 과정을 살펴볼 수 있다^{[1][2]}. 적합성 검사에서는 시험할 부분에서 발생할 수 있는 모든 경우의 수를 포함하는 시험 비트열을 제작하고, 이 비트열을 시험 대상 디코더와 기준 디코더 (reference decoder)에 각각 입력시켜 그 결과를 비교하여, 그 결과가 동일한 경우 시험된 디코더가 올바르게 동작하는 것으로 판단한다^{[3][6]}. 본 논문은 형상 부호화의 각 과정 및 부호화의 시험 비트열을 제작하고 이에 대한 실험결과를 살펴보았다. 본 논문의 구성은 다음과 같다. 먼저 2장에서는 MPEG-4 형상 부호화 표준을 간단히 알아보고 3 장은 mode 복호화에서 I-VOP, P-VOP, B-VOP에 대한 BAB_type의 모든 경우의 수를 시험하는 방법을 설명한다. 4 장에서는 inter CAE와 intra CAE에 대한 모든 경우의 context 계산을 시험하는 방식을 제안한다.

II. 형상부호화 (Shape Coding) 방식

MPEG-4의 경우 객체를 기반으로 형상 부호화한다. 비주얼 부호화기에서는 텍스처 화면의 객체에 해당하는 곳만을 통과시키는 alpha plane을 사용하여 비디오 객체에 해당하는 부분만의 정보를 송신한다. 수신기에서는 비주얼 객체 부분의 정보를 받아 복호화한다. 복호화된 여러 객체들, 즉 배경, 오디오 객체, 자연영상 객체, 컴퓨터 그래픽과 같은 인위적 영상 객체들을 합성하여 완전한 화면으로 구성한다. 그림 1은 Akiyo 영상의 형상 부호화의 예이다. 그림 1 (a)는 형상부호화물 하기 전의 텍스처 영상이고 그림 1 (b)는 객체의 위치를 나타내는 이진 alpha plane 영상이다. 그림 1 (c)는 형상부호화 후의 결과로 객체부분이 이진 alpha plane으로 인해 배경으로부터 분리되었음을 볼 수 있다. 객체의 크기와 위치는 alpha plane에 의해 결정되는데

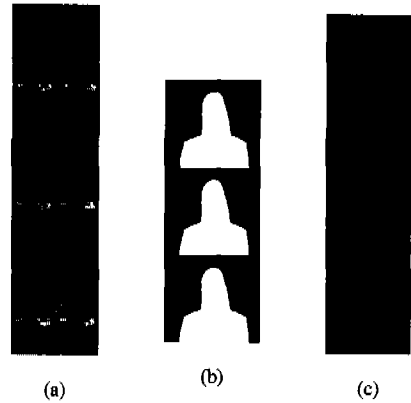


그림 1. shape coding의 예 (a) texture영상 (b) alpha plane (c) shape coding 결과

alpha plane의 흰색부분이 객체뿐 의미하며 그곳의 텍스처 정보만을 부호화한다.

영상 부호화는 기본적으로 각 매크로블록별로 부호화가 되고 이진 alpha plane은 I-VOP (Intra-coded video object plane)를 위한 intra mode와 P-VOP (Predictive-coded video object plane)와 B-VOP (Bidirectionally predictive-coded video object plane)를 위한 inter mode로 부호화된다. 기본적으로 사용되는 부호화로 block-based Context-based Arithmetic Encoding (CAE)과 블록기반 움직임보상이 함께 쓰인다. 형상 부호화의 구체적인 예를 살펴보자. 176 화소 × 144 화소 화면에서 임의로 48 화소 × 80 화소의 사각형이 객체라 가정하여 여기에 해당하는 alpha plane을 만들어준 경우 그 곳의 위치에 속한 텍스처 정보만이 부호화되어 수신기상으로 전달되게 된다. 그 결과 수신기 상에 재현된 VOP 크기는 실제 화면의 크기인 176 화소 × 144 화소가 아닌 객체의 크기인 48 화소 × 80 화소로 나타난다.

본 논문의 모든 실험에서 사용된 영상의 크기는 176 화소 × 144 화소이다. 각 BAB의 크기는 16 화소 × 16 화소로 정의되었으므로 한 영상은 99 (11 블록 × 9 블록) 개의 블록을 포함한다.

III. Mode 복호화 시험

객체의 위치와 형상정보를 가진 alpha plane의 정보는 16 화소 × 16 화소 BAB의 형태로 부호화된다. BAB type은 표 1과 같이 각 VOP에 따라 나타나는 BAB_type의 종류가 다르다. I-VOP는 2 (무명), 3 (블부명), 4 (intra CAE)로 이루어지며

P-VOP와 B-VOP의 경우 0 ~ 6까지 7 가지 모든 BAB type을 가진다⁵⁾.

MPEG-4 표준안의 규격의 디코더 적합성 검사에서도 mode 복호화 시험을 규정하고 있다. MPEG-4 표준 규격안의 시험 비트열 #SH-2에서 주어진 profile@ level 구조의 영상 움직임 벡터와 BAB type 부호화, MB bandwidth, 참조 memory bandwidth에 대한 시험이 요구된다⁷⁾. 본 장에서는 정적 시험에 해당되는 BAB_type의 부호화와 영상 움직임 벡터에 관한 시험 방법을 제안한다.

1. I-VOP 의 mode 복호화

I-VOP은 앞에서 언급한 바와 같이 3가지 경우의 BAB_type을 갖는다. VOP상의 블록 (i, j)에 해당하는 BAB의 BAB_type을 f(i, j)라 한다면 f(i, j)의 BAB type의 codeword는

$$C = 27 \times (f(i-1, j-1) - 2) + 9 \times (f(i, j-1) - 2) + 3 \times (f(i+1, j-1) - 2) + (f(i-1, j) - 2)$$

으로 계산된다. 이때 만약 f(i, j)가 VOP 외부의 BAB을 가리키는 경우 f(i, j)는 BAB_type 2를 갖는 투명으로 가정하며, 현재 프레임 외부의 BAB들 또한 BAB_type을 투명으로 취급한다⁴⁾.

표 1. BAB_type 의 종류

BAB type	Semantic	Used in
0	MVDs=0 and No update	P-, B-VOPs
1	MVDs≠0 and No update	P-, B-VOPs
2	Transparent	All VOP types
3	Opaque	All VOP types
4	Intra CAE	All VOP types
5	MVDs=0 and Inter CAE	P-, B-VOPs
6	MVDs≠0 and Inter CAE	P-, B-VOPs

본 장에서는 표 1에서 I-VOP인 경우 나올 수 있는 모든 codeword C를 수신기에서 올바르게 계산할 수 있는지를 시험하는 mode 복호화의 시험방법을 제안한다. codeword C는 alpha plane의 영상 정보로부터 계산하므로 그림 2에서 현재 BAB type (f(i, j))가 2 (투명), 3 (불투명), 그리고 4 (intra) 인

경우 각각에 대해서 codeword C의 값이 나올 수 있는 모든 경우의 수의 alpha plane을 제작하였다. 위의 식에 따라 codeword C 값은 주위의 매크로블록에 따라 0~80 사이의 값을 가질 수 있다. 시험 비트열을 발생시키기 위한 alpha plane은 3 블록 × 2 블록을 하나의 단위로 하여 각각의 codeword C 값을 발생시키는 BAB_type을 부여하는 형태로 제작하였다. 그림 2는 각각 현재 BAB type이 2, 3, 4의 경우에 대하여 제작된 alpha plane의 일부이며 흰색 부분은 객체를 나타낸다.

앞에서 제안한 방식에 따라서 제작된 그림 2와 같은 alpha plane과 Akiyo 영상을 입력하여 영상을 부호화하여 시험 비트열을 제작하였다. 생성된 비트열은 다시 복호화하면서 디코더에서 codeword C 값을 출력하여 확인하였다. 그 결과 모든 경우에 따라 제작된 비트열이므로 정상적인 디코더에서는 각각 현재 BAB type이 2 (투명), 3 (불투명), 그리고 4 (intra) 경우에 codeword C 값이 0~80 사이의 모든 수가 나오는 것을 확인할 수 있었다. 만약 정상적인 디코더가 아니거나 codeword C 값을 계산하는 부분이 잘못된 디코더라면 모든 숫자가 나오지 않거나 0 미만 또는 80을 초과하는 숫자가 나올 수 있다.

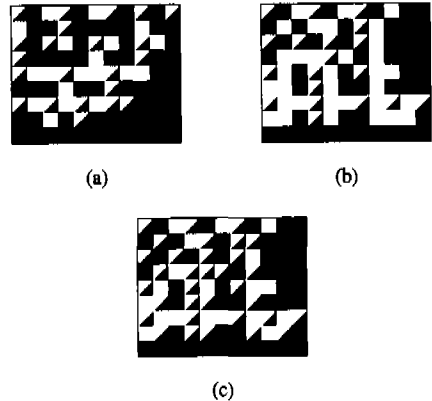


그림 2. I-VOP mode decoding test에 사용된 alpha plane (a) BAB_type=2인 경우 (b) BAB_type=3인 경우 (c) BAB_type=4인 경우 2. P-VOP & B-VOP에서의 mode 복호화

P-VOP와 B-VOP에서 현재 BAB type을 복호화할 때 참조할 VOP에 대해 동일한 위치의 BAB을 참조한다. 이때 참조하는 VOP는 전참조 (forward reference) VOP와 후참조 (backward reference) VOP가 있다. 전참조 VOP는 가장 최근 과거의 비어있지 않은 I-VOP나 P-VOP으로 정의되고 후참조

VOP는 가장 가까운 미래의 비어있지 않은 I-VOP나 P-VOP으로 정의된다. P-VOP의 경우 전참조 VOP가 사용되고 B-VOP의 경우 전참조와 후참조 모두 사용된다¹⁾.

P-VOP나 B-VOP의 BAB type은 0 ~ 6까지 7개 type 값을 모두 가질 수 있다. 이때 2, 3, 4의 경우 I-VOP일 때와 같으므로 P-VOP와 B-VOP에서 중복하여 하지 않았다. BAB_type 0, 1, 5, 6의 경우 표 1과 같이 BAB의 MVDs (Motion Vector Difference for shape) 값과 inter CAE의 유무에 따라 결정이 된다. 이때 MVDs값은 MVs (Motion Vector for shape)와 MVPs (Motion Vector Prediction for shape)의 차이값에 의해 결정된다. 즉, $MVDs = MVs - MVPs$ 이다. 현재 블록의 형상 움직임 벡터가 MVs이고 MVPs는 주변의 MVs와 MV에 의해 결정된다. 현재 형상 매크로블록의 MVPs는 후보 값에 의해 결정된다. 이때 후보 값은 그림 3과 같이 MVs1, MVs2, MVs3, MV1, MV2, MV3의 순서로 우선 순위를 갖고 우선 순위가 큰 곳에 값이 있을 경우 그곳의 값이 MVPs가 된다.

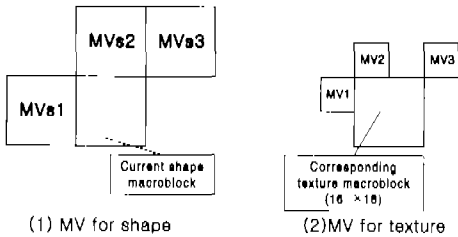


그림 3. MVPs로 취하는 후보값

1) P-VOP의 복호화

시험을 위해 제작된 alpha plane은 176 화소 × 144 화소 크기의 임의의 이진 영상이다. 이러한 영상은 각각의 매크로블록이 분극적인 값을 취하기 때문에, 형상 움직임 벡터 값과 새로 변경되는 값에 따라서 BAB_type이 민감하게 변한다. 그 결과 비트열 제작시 정확한 실험결과를 얻을 수 있다. 실험에서 입력되는 영상의 VOP형태는

$$I_1, P_2, P_3, P_4, P_5$$

로 설계하였다. I-VOP의 BAB_type 결과는 임의의 값을 가진 매크로블록이므로 각각의 매크로블록의 모든 화소 값을 취하기 위해 intra CAE를 하며 나타나는 BAB_type은 4가 된다. 각 P-VOP에 해당하

는 alpha plane은 각각의 실험의 조건에 따라 이전 VOP의 값을 취하여 제작하였다.

BAB_type이 0이 될 조건은 MVDs = 0이며 각각 BAB에 추가 변경 (update) 되는 값이 없고 참조 BAB과 동일한 경우이다. MVDs가 0인 조건을 만족시키기 위해서 그림 4와 같이 블록간의 움직임을 똑같이 주어 MVDs 즉 MVs와 MVPs의 차이가 0이 되도록 해주었다.

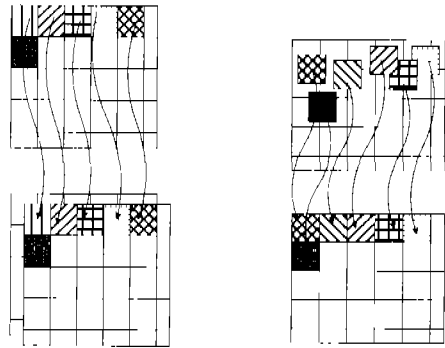


그림 4. BAB_type = 0 시험용 alpha plane
그림 5. BAB_type = 1 시험용 alpha plane

BAB_type = 1의 조건은 MVDs ≠ 0이고 변경되는 값이 없는 경우이다. MVDs가 0이 아니기 위해서는 MVs와 MVPs의 값이 차이가 0이 아닌 값을 가져야한다. 또한 추가 변경되는 값이 없으므로 BAB_type = 0일 때와 마찬가지로 현재 BAB의 픽셀 값을 취할시 움직임 벡터만을 주고 참조하는 영상의 값은 그대로 취한다. BAB_type = 1을 위한 시험영상은 그림 5와 같다. MVDs가 0이 아니기 위하여 현재의 BAB이 가지는 형상 움직임 벡터와 예측값 MVP가 되는 후보 값들의 차이가 감지 않은 조건을 만족시키는 alpha plane을 제작하였다. 각각의 블록의 움직임 벡터를 줄 때 현재 블록의 움직임 벡터가 예측된 움직임 벡터와 다른 값이 주어져야 한다. 예측을 할 때 가져오는 블록은 그림 3에서 살펴본 바 있다. 그 결과 각각의 블록의 MVDs는 0이 아닌 값을 갖고 추가 변경되는 값은 없다. 본 논문의 실험에서는 형상 움직임 벡터의 모든 경우에 대한 복호화를 확인하기 위해 MPEG-4 표준안의 모든 가능한 경우의 형상 움직임 벡터 (-16 ~ 16) 값을 모두 포함한 비트열을 만들었다. BAB_type = 5는 MVDs = 0 이면서 이전 프레임에 비해 추가 변경되는 값이 있어서 inter CAE를 하는 경우이다. MVDs = 0이 되도록 하는 조건은

앞의 BAB_type = 0과 같다. 각각의 블록의 화소 값이 참조 영상의 블록 값에 비해 추가 변경되는 값을 가질 때 inter CAE를 하게 된다. 이러한 조건에 만족시키기 위해 현재 프레임의 MVD = 0인 각각의 모든 매크로블록의 화소 값에서 임의로 7 개 정도의 화소 값을 반전시켜주었다. 그 결과 전 화면에 비해 각각의 매크로블록의 화소 값이 추가 변경된다.

BAB_type = 6 인 경우 MVDs ≠ 0 이면서 추가 변경되는 값이 있으므로 inter CAE를 하게 된다. MVDs ≠ 0인 블록을 생성하는 것은 BAB_type = 1의 블록을 생성시킨 방법과 같다. inter CAE를 위해 BAB_type = 5에서 사용했던 방식으로 각각의 16 × 16의 블록에서 7 개 정도의 화소값을 반전시켜 시험 alpha plane을 만들어 주었다.

앞에서 제작된 alpha plane과 텍스처 영상에 해당하는 akiyo영상을 부호화기에 넣어주어 각각의 BAB_type에 따른 시험비트열을 생성하였다. 제작된 모든 시험 비트열을 본 연구진이 제작한 비트열 분석기를 사용하여 분석한 결과 모든 매크로블록의 BAB type이 각각의 조건에 만족하는 결과를 갖는 것을 확인할 수 있었다. 그 결과 정상적인 디코더의 경우 각각의 BAB type 시험비트열의 조건에 맞는 결과가 나온다. 즉 그림 6 (a) 과 같이 BAB type 0인 비트열의 P-VOP에 해당하는 프레임의 모든 매크로블록의 BAB type이 0을 나타내야 한다. 그림 6 (b) 는 BAB type 6인 비트열의 정상적인 부호화 결과물비트열 분석기에서 본 결과이다. 만약 디코더가 P-VOP의 mode decoding에서 잘못된 연산을 수행하게되면 각각의 BAB_type 시험비트열의 결과가 잘못된 BAB_type을 산출해내게 되거나 P-VOP 부호화 하는 중간에 오류가 발생하게된다.

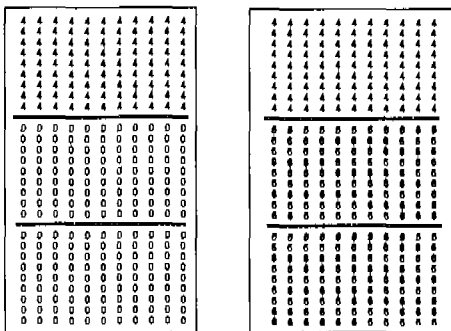


그림 6. mode decoding test bitstream의 정상적인 디코딩 결과 (a) Bab_type = 0인 결과 (b) Bab_type = 6인 결과

2) P-VOP의 복호화

P-VOP의 경우 전참조하므로 과거의 영상을 참조하게 된다. B-VOP의 경우 참조영상이 전참조와 후참조 모두 사용되어, 조건에 따라 과거 혹은 미래의 영상이 참조된다. 참조하는 영상을 결정할 때는 움직임 보상이 쓰이는데 항상 부호화의 경우 텍스처 부호화에 비해 비교적 간단하다. 현재 B-VOP에 대하여 전참조 영상이 후참조 영상에 비하여 시간적 차이가 같거나 작을 경우 전참조를 하고 그 이외의 경우 후참조를 하게 된다. 그 예로 만약 영상을

$$\dots P_1, B_2 B_3 B_4 P_5 \dots$$

형태로 부호화하는 경우 B₂와 B₃는 전참조를 하여 앞의 P₁를 참조하나, B₄는 후참조를 하여 뒤의 P₅를 참조하게된다. 본 시험에서는 이와 같은 조건을 고려하여 입력영상을

$$I_1 P_2 B_3 B_4 B_5 P_6 B_7 B_8 B_9 P_{10}$$

형태로 주었다. 시험 비트열을 제작시 앞의 P-VOP 모드 부호화 한 경우와 비슷한 방법으로 BAB type 0, 1, 5, 6의 시험 영상을 만들었다. 만들어진 시험 영상을 부호화하여 비트열로 만든 후 그 결과를 software로 제작된 기준 디코더에서 확인하여 시험의 적합성 여부를 살펴보았다. 그 결과 제작된 비트열은 전참조의 후참조 두가지 경우의 모든 BAB type을 시험할 수 있었다. 본 시험비트열을 정상적인 디코더에 넣어준 경우 각각의 BAB type에 대해 정확한 BAB type의 값을 가지나 비정상적인 디코더의 경우 P-VOP의 시험에서 언급한바와 같이 잘못된 값을 산출하게된다.

IV. Context based arithmetic decoding 시험

영상의 이진 행상정보를 부호화하는 방법으로 context based arithmetic coding 방법이 사용된다. Context based arithmetic encoding의 알고리즘은 다음과 같다.

- ① Context 수를 조사한다.
- ② Context 수를 사용하여 확률 테이블을 참조한다.
- ③ 인덱스 확률을 사용하여 산술 인코더를 동작시킨다.

MPEG-4 디코더 적합성 검사 표준 규격안에서는 시험 비트열의 #SH-1에서 intra CAE, inter CAE 등의 검사를 요구한다⁷⁾. 이번 장에서는 intra CAE 와 inter CAE의 시험을 위한 시험 비트열의 제작방법을 살펴보겠다.

1. intra CAE의 context computation

그림 7은 BAB type이 intra인 경우 현재 화소의 context 수를 계산하기 위한 주위의 화소들을 보여 주고 있다. 그림 7에서 주위의 화소에 따른 현재 context 수는

$$C = \sum_k c_k 2^k$$

와 같이 표현할 수 있다. 여기서 c_k 는 화소가 투명일 경우에는 0, 불투명일 경우에는 1을 갖는다. 만약 이 과정에서 주위의 화소들 중에서 현재 VOP 외부에 있거나 현재 프레임 외부에 있는 것은 그 화소값을 0으로 가정하고 context 수를 계산한다.

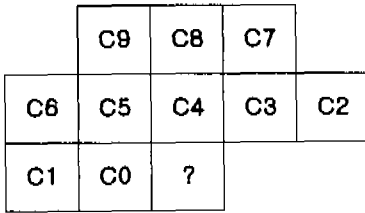


그림 7. INTRA template와 context 구조

본 논문에서는 context 가 가질 수 있는 모든 경우의 수 (0 ~ 1023)가 나올 수 있도록 alpha plane를 제작하여 입력 영상으로 하였을 때 올바른 context 수가 나오는 지를 확인하는 방식을 택하였다. 모든 가능한 context 들을 만들기 위해서 5 화소 x 3 화소 크기의 블록단위로 context 수를 만들었다. 그림 8의 (a)는 현재 화소값이 투명인 경우 모든 경우의 수가 나올 수 있도록 제작한 alpha

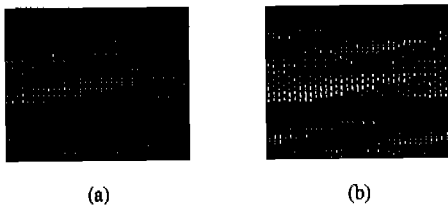


그림 8. Intra context number test를 위한 alpha plane.
(a) current pixel이 transparent인 경우,
(b) current pixel이 opaque인 경우.

plane 이며 그림 8의 (b)는 현재 화소값이 불투명인 경우 모든 경우의 수가 나올 수 있도록 제작한 alpha plane이다.

디코더의 context 수를 계산하는 과정을 실험하기 위해서 앞에서 제안한 영상을 alpha plane으로 하고 이 두 영상을 정상적인 인코더로 부호화 한 후 여기에서 제작된 비트열을 디코더에 넣어 복호화하면서 디코더의 context 수를 계산하는 부분에서 context 수를 출력해 보았다. 그 결과 각각의 영상에 대해서 context 수 C가 0 ~ 1023 사이의 모든 경우의 수가 바르게 계산되어 나오는 것을 확인할 수 있었다.

2. P-VOP & B-VOP inter CAE에서의 context computation

그림 9는 BAB type이 inter mode인 경우 현재 화소의 context 수를 계산하기 위한 주위의 pixel들을 보여 주고 있다. inter CAE에서 P-VOP의 경우 앞 프레임의 값을 참조하며, B-VOP의 경우 3장의 B-VOP mode decoding test에서 언급한 바와 같이 앞과 뒤 프레임 모두를 참조한다. 그림 9에서 주위의 화소에 따른 현재 context 수는 intra CAE에서 context 수를 계산하던 방식과 동일하게 표현될 수 있다. inter CAE의 경우 그림 9와 같이 현재의 BAB에서 4 개, 참조하는 BAB에서 5 개의 화소를 참조한다. inter의 경우 BAB_type이 5인 경우와 6인 경우가 있으므로 시험방식을 두 가지로 나누었다.

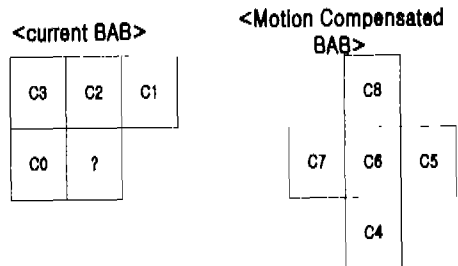


그림 9. INTER template 와 context 구조

먼저 BAB type = 5의 inter CAE 하는 경우의 시험방식을 살펴보겠다. 본 논문에서는 intra CAE의 시험과 마찬가지로 context 수가 가질 수 있는 모든 경우의 수 (0 ~ 511)가 나올 수 있는 alpha plane를 제작하여 이를 입력 영상으로 부호화를 해주어 올바른 context 수가 나오는 지를 확인하는 방식을 택하였다. 먼저 주변 화소가 가질 수 있는 정

우의 수는 2^9 으로 512가지이고 현재의 화소 값이 불투명인 경우와 투명인 경우가 있으므로 결과적으로 1024가지의 경우가 나오게된다

P-VOP의 경우 그림 9에서 움직임 보상할 BAB의 위치는 앞 프레임이 된다. 그림 9의 왼쪽의 그림이 현재 프레임의 BAB이고 물음표 부분이 현재의 화소이다. 오른쪽 그림은 참조하는 프레임의 BAB으로 P-VOP의 경우 이전 프레임의 값을 참조하며 c6이 위치하는 곳이 현 화소의 위치와 같게된다. context 계산의 모든 경우를 발생시키기 위해 참조할 프레임과 참조될 프레임에 위치할 두 개의 3×3 화소 블록을 한 단위로 각 c0, c1, c2, c3, c4, c5, c6, c7, c8에 해당하는 값을 주어서 $2^9 \times 2 = 1024$ 가지 모든 경우의 수를 발생시키는 블록을 만들었다. 이러한 블록을 BAB_type = 5인 이미지에 일정한 단위의 간격으로 움직임 벡터가 주어지는 것을 고려하여 정확한 위치에 각각의 단위 블록 값을 주었다. 그림 10 (a)는 일정한 간격으로 현재 화소를 지정한 후 MVDs가 0이면서 inter CAE인 모든 경우의 context 값을 발생시킨 영상이다. 실제 alpha plane으로 적용할 때 그림 10 (b)에서 보는 것과 같이 앞의 모드 복호화에서와 같은 방법으로 제작된 BAB_type = 5인 alpha plane에 그림 10 (a)에서 현재 화소와 주변의 참조하는 화소에 위치한 값을 그림 10 (a)와 같이 바꾸어 주었다. BAB_type = 6인 경우 inter CAE context 수를 발생하는 것은 BAB_type = 5와 비슷하지만 프레임의 블록들의 형상 움직임 벡터가 각각 다르기 때문에 참조 영상의 값을 가져올 때 그 블록이 움직인 형상 움직임 벡터를 추적하여야 한다. 시험방식은 앞

의 BAB_type = 5인 경우와 같고 BAB_type = 6인 프레임을 움직임을 고려한 모든 경우의 수를 고려하여 만든 영상의 화소값에 해당하는 위치 값을 바꾸어서 1024가지 경우가 발생함을 확인하였다. 그림 11 (a)은 context 수를 발생시킨 경우이고 실제 입력된 alpha plane은 그림 11 (b)와 같다. 그 결과 이러한 방법으로 제작된 시험비트열이 현재 화소로 가장된 각각의 화소에서 context 수의 모든 경우를 발생시키는 것을 확인할 수 있다.

위에서 설명한 제작방법에 따라 만들어진 alpha plane들을 부호화하여 모든 경우의 context 수를 계산하는 것을 시험하는 시험비트열을 제작하였다. 또한 시험비트열이 복호화시 정상적인 디코더에서 모든 경우를 발생하는 것을 확인하였다. 본 논문에서 제작된 intra CAE와 BAB_type 5, BAB_type 6에 대한 inter CAE를 시험하는 비트열을 정상적인 디코더에 넣을 경우 모든 경우의 context 수 C가 빠짐없이 산출되게된다. 만약 정상적인 디코더가 아닌 경우 시험비트열을 부호화한 결과에서 모든 경우의 context 수 C가 산출하지 않거나 context 수 C가 MPEG-4 표준안에서 지정된 범위를 벗어난 값이 나오게 된다.

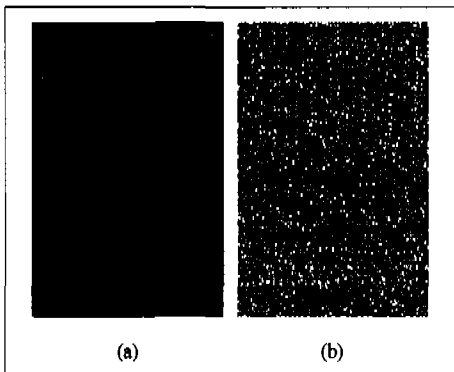


그림 10. BAB_type = 5인 inter CAE context computation 시험에 사용된 alpha plane.
(a) 모든 경우의 context number만을 나타내는 영상
(b) 실제 사용한 alpha plane.

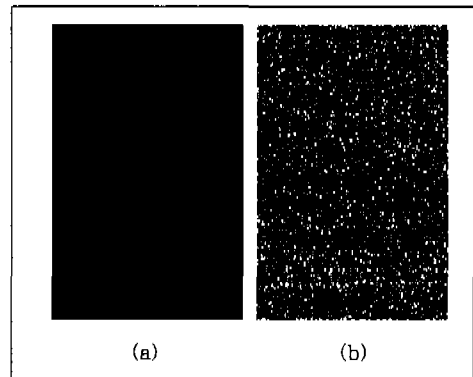


그림 11. BAB_type = 6 일 때 Inter CAE context computation number 시험에 사용된 alpha plane
(a) 모든 경우의 context number만을 나타내는 영상
(b) 실제 사용한 alpha plane.

V. 결론

본 논문은 MPEG-4 visual 정합 실험을 위해 사용될 수 있는 형상부호화 시험 비트열을 제작한 후 그 결과를 확인하였다. 제작된 비트열을 통해 MPEG-4 디코더가 형상부호화의 표준안의 내용을

따르고 있는지 여부를 확인할 수 있었다. 본 논문에서 제안된 시험 방식을 통하여 영상 복호화에서 모든 가능한 BAB_type을 올바르게 복호화함을 확인하고 intra mode가 가질 수 있는 모든 context 값을 확인하였다. 또한 CAE에 있어서도 intra CAE, inter CAE에 대한 모든 경우의 수를 확인하는 시험 비트열 제작 및 실험을 완료하였다. 본 논문에서 제안한 비트열을 시험 대상인 MPEG-4 hardware 또는 software 디코더에서 복호화하여, 그 결과가 기준 복호화기의 출력과 동일한지의 여부를 확인하여 영상 부호화에 관한 적합성시험을 할 수 있다. 본 논문에서는 정적 시험 (static test)에 국한하였으나, 충분히 빠른 속도로 복호화하는지를 시험하는 동적 시험 (dynamic test)에 대한 연구가 진행되어야 한다.

감사의 글

본 연구에 많은 도움을 주신 현대전자산업(주)의 오대일 수석 연구원, 정재원 책임 연구원, 강현수 선임연구원께 감사말 드립니다.

참고 문헌

[1] 임동근, 김대희, 호요성, 양수경, 고종석, "MPEG-2 적합성 검사의 구현", *신호처리합동학술대회*, vol. 10, no. 1, pp. 79-82, 부산대학교, 1997년 9월.

[2] C. M. Kim, B. U. Lee, and R.-H. Park, "Design of MPEG-2 video test bitstreams," *IEEE Trans. Consumer Electronics*, vol. CE-45, no. 4, pp. 1213-1220, Nov. 1999.

[3] ISO/IEC JTC1/SC29/WG11 N2550 Information technology- Very Low Bitrate Audio-Visual Coding: Conformance ISO/IEC 14496-4 Committee Draft.

[4] ISO/IEC 14496-2 Information Technology- Generic Coding of Audio-visual Objects (part 2: visual).

[5] ISO/IEC JTC1/SC29/WG11 MPEG99/N2687 Coding of moving pictures and associated audio Information.

[6] ISO/IEC JTC1/SC29/WG11 MPEG99/M4973 MPEG-4 Visual Core Profile Verification Bitstream Specifications Draft.

[7] ISO/IEC JTC1/SC29/WG11 N2883 Information Technology- Generic coding of Moving pictures and Audio: Conformance ISO/IEC 14496-4/Amd 1.

황혜진(Hea-Juen Hwang)

1999년 8월 : 이화여자대학교 전자공학과졸업

1999년 9월~현재 : 이화여자대학교 정보통신학과

석사과정

<주관심 분야> 영상압축, 컴퓨터 비전

박인수(In-Su Park)

1999년 2월 : 조선대학교 제어계측공학과졸업

1999년 3월~현재 : 서강대학교 전자공학과 석사과정

<주관심 분야> 멀티미디어 통신 시스템

박수현(Su-Hyun Park)

1999년 2월 : 이화여자대학교 전자공학과졸업

1999년 3월~현재 : 이화여자대학교 정보통신학과

석사과정

<주관심 분야> 영상압축, 컴퓨터 비전

이병욱(Byung-Uk Lee)

정회원

이화여자대학교 정보통신학과 부교수, 정회원,

한국통신학회논문지 제22권6호 참조