

# 공간 영역 예측에 의한 정지 영상 부호화

조상규<sup>\*</sup> · 문준 · 황재정

<sup>\*</sup>군산대학교

Advanced Image Coding based on spacial domain prediction

Sang-Gyu Cho<sup>\*</sup> · Joon-Moon · Jae-Jeong Hwang

<sup>\*</sup>Kunsan National University

E-mail : sgcho@kunsan.ac.kr

## 요 약

본 논문은 차세대 영상 부호화 표준, H.264/MPEG-4 Part 10 AVC(Advanced Video Coding)와 기존의 정지 영상 부호화 표준, JPEG(Joint Photographic Experts Group)이 결합된 진보적인 정지 영상 부호화(Advanced Image Coding, AIC) 알고리즘에 대하여 살펴본다. AIC 알고리즘은 H.264의 인트라 프레임 블록 예측 방법과 컨텍스트 기반 적응형 이진 산술 부호화(Context-based Adaptive Binary Arithmetic Coding, CABAC), 그리고 JPEG 표준의 DCT 변환이 결합된 것이다. 본 논문에서는 AIC와 JPEG, JPEG-2000 등의 정지 영상 표준들의 성능을 비교 분석하고 그 결과를 제시한다.

## ABSTRACT

This paper is made up Advanced Image Coding(AIC) that combines algorithms from next generation image coding standard, H.264/MPEG-4 Part 10 advanced video coding(AVC) and still image compression standard, JPEG(Joint Photographic Experts Group). AIC combines intra frame block prediction from H.264 with a JPEG style discrete cosine transform and quantization, followed by Context-based Adaptive Binary Arithmetic Coding(CABAC) as used in H.264. In this paper, we analyzes the efficiency of the AIC algorithm and JPEG and JPEG-2000, and it presents of result.

## 키워드

AIC, Intra frame block prediction, CABAC, DCT

## 1. 서 론

압축(compression)이란 데이터를 보다 적은 수의 비트(bits)로 만드는 것이다. 비디오나 이미지 압축(video/image coding)은 디지털 영상을 적은 수의 비트로 만들거나 압축하는 과정을 의미한다.

대부분의 비디오 코딩 방법은 시간적(temporal) 그리고 공간적(spacial) 중복성(redundancy)을 이용하여 그 과정을 수행한다. 이에 반해서, 이미지 코딩은 공간적으로 인접에 있는 각 화소사이의 높은 상관관계(correlation)를 이용한다. 이러한 내용을 기반으로 이미지와 비디오 압축은 20년이 넘는 기간 동안 매우 활발한 연구와 개발이 진행된 분야이며, 압축과 복원을 위한 많은 종류의 시스템과 알고리즘이 제안되었고 개발 되었다. 그러한 비디오나 이미지 압축을 위한 국제 표준안들

중에서 대표적으로 JPEG[1, 2], MPEG[3] 그리고 H.26x[4, 5] 시리즈(series)가 이에 속한다.

최근에 등장한 H.264/MPEG-4 Part 10 AVC(Advanced Video Coding) 표준안[4, 6, 7, 8]은 시각 정보의 부호화된 표현을 위한 차세대 영상 압축 표준이다. 압축의 효율을 증대시키기 위해서 새로운 기술들을 많이 채택하였다. 그 중에서도 인트라(intra) 부호화를 위한 방향성 공간 예측(directional spatial prediction)과 컨텍스트 기반 적응형 산술 부호화(Context-based Adaptive Binary Arithmetic Coding, CABAC)[5]가 대표적이다.

본 논문에서는 새로운 알고리즘은 아니지만, H.264 표준안의 인트라 부호화 방법과 CABAC 부호화 방법을 JPEG의 DCT 변환과 결합한 진보적 이미지 코딩(Advanced Image Coding, AIC)

알고리즘에 대해서 살펴본다. 그리고 AIC의 복원 영상을 JPEG[1]과 JPEG-2000[2]의 비교 분석을 통해서 그 성능을 고찰해 본다.

## II. 본 론

일반적으로 JPEG 압축의 기본 원리는 시각적으로 중요한 부분과 덜 중요한 부분으로 분리하여 시각적으로 중요한 부분은 살리고 덜 중요한 부분은 손실시킴으로써 데이터를 줄이는 것이다. AIC 알고리즘은 새로운 알고리즘이 아니며 기존 표준안에서 특정 부분만을 추출하여 조합한 것이다. 입력 영상의 컬러 정보를 줄이기 위하여 RGB 데이터를 YCbCr로 변환하고, H.264 표준안의 인트라 프레임 블록 예측(intra frame block prediction)과 JPEG의 DCT, 양자화 그리고 H.264 표준안의 CABAC 부호화를 한다.

그림 1은 AIC의 인코더와 디코더의 블록다이어그램이다.

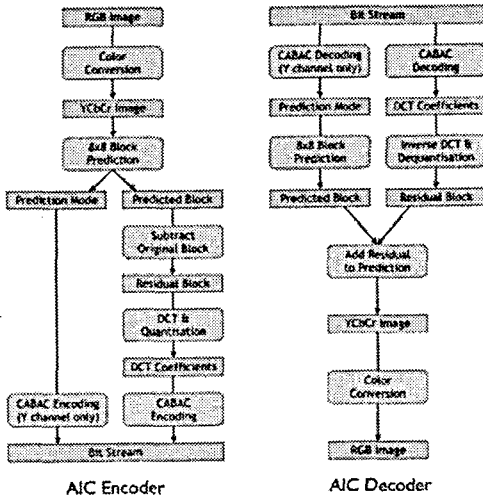


그림 1. AIC 알고리즘  
Fig. 1. AIC algorithm

### 1. 컬러 변환(color conversion)

입력 영상이 컬러 영상인 경우에 시각 정보를 줄이기 위해서 RGB 컬러 모델을 YCbCr 컬러 모델로 변환한다. 단, 그레이스케일(gray-scale) 영상은 휘도 성분(Y)만을 가지고 있기 때문에 제외한다. 컬러 모델 변환식은 다음과 같다[1].

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ Cb &= -0.169R - 0.331G + 0.5B \\ Cr &= 0.5R - 0.419G - 0.081B \end{aligned}$$

그림 2는 RGB Lena 영상의 YCbCr 컬러 변환 결과이다.



그림 2. RGB, YCbCr 컬러 변환  
Fig. 2. RGB to YCbCr color conversion

### 2. 8x8 블록 예측(8x8 block prediction)

컬러 변환된 Y, Cb, Cr 각 성분들은 8x8 블록으로 나누어지고 그림 3(a)처럼 왼쪽에서 오른쪽 그리고 위쪽에서 아래쪽으로 각 라인별로 부호화된다. AIC의 블록 예측은 이전에 인코딩되고 복원된 블록에 의해서 현재 블록을 예측하는 H.264 표준의 인트라 코드 방법을 이용한다. H.264는 4x4 또는 16x16 휘도(Y) 블록들을 예측하기 위한 알고리즘들을 가지고 있다.

그림 3(b)처럼 8x8 블록으로 확장된 H.264의 4x4 블록 알고리즘을 이용하며, 예측을 위해서 이전에 인코딩되고 복원된 블록들로부터 위쪽과 왼쪽의 25개 화소(blue pixels)를 이용한다. 그림 4는 AIC 알고리즘과 H.264 표준의 9가지 예측 모드와 Lena 영상의 예측 결과 예를 나타낸다. 각 예측 모드들은 예측을 위해서 필요로 하는 모든 화소들(위쪽과 왼쪽의 25개 화소)이 이용 가능한 경우 해당되는 예측 모드를 사용된다. 예를 들어 Mode 0은 위쪽의 16개 화소가 모두 이용 가능할 때 이용된다. 단, 처음 인코딩된 블록의 경우는 예측에 이용 가능한 화소들이 없기 때문에 화소 값 128이나 50% 그레이가 사용된다[4, 6, 7].

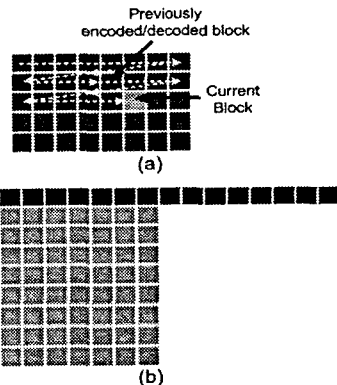


그림 3. 8x8 블록 예측  
(a) 블록 스캔, (b) 확장된 8x8 블록

Fig. 3. 8x8 Block prediction, (a) line order scanning, (b) extended 8x8 block

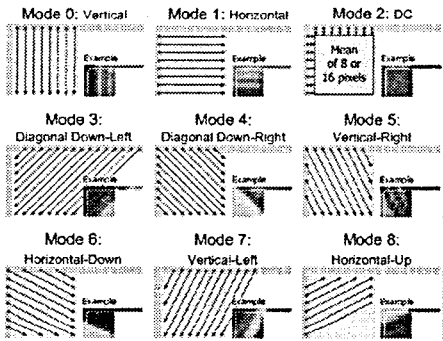


그림 4. 예측 모드  
Fig. 4. Prediction modes

AIC는 모든 휘도 블록을 위해서 9개의 예측 모드를 사용하며, 이들 모드들은 예측보다는 예측된 블록과 원래 블록의 차이를 최소화 한다. 선택된 예측 모드는 스트림(stream)으로 인코딩되며, 색차성분(Cb, Cr)은 대응하는 휘도(Y) 블록의 예측모드와 동일한 예측모드를 사용한다[4, 6, 7].

그림 5는 Lena 영상의 원본과 예측된 영상의 휘도 성분의 오차 영상이다.

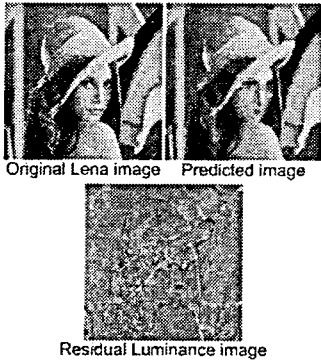


그림 5. 휘도 성분의 예측 오차  
Fig. 5. Residual luminance image

### 3. DCT 변환과 양자화(DCT and quantization)

그림 5의 오차 블록의 엔트로피(entropy)를 보다 더 줄이기 위해서 DCT 변환하고 선택된 양자화 레벨(quality level)로 DCT 계수(coefficients)를 양자화 한다.

AIC에 적용된 DCT는 고화질의 영상을 생성하기 위해서 부동 소수점 DCT(floating point DCT)을 수행한다. 그리고 DCT 변환의 고속 연산을 위해서 AAN(Arai, Agui and Nakajima) DCT 알고리즘을 이용한다[9].

### 4. CABAC

(context-based adaptive binary arithmetic coding) 양자화된 DCT 계수와 예측 모드는 모두 스트

림으로 저장된다. 이때 필요한 비트를 최소화하기 위해서 H.264의 CABAC에 의해서 인코딩 된다.

JPEG에서는 양자화된 DCT 계수가 지그재그 스캔(zigzag scan)과, 가변 길이 부호화(Variable Length Coding, VLC)를 통하여 인코딩되고 전송된다[1]. CABAC는 RLC(Run Length Coding)을 사용하지 않기 때문에 DCT 계수의 재정렬(reorder)이 필요하지 않다. AIC는 양자화된 DCT 계수 값을 라인별 스캔 순서로 전송하게 된다.

CABAC는 요소(element's)의 컨텍스트에 따라 각 선택스(syntax) 요소에 대한 확률 모델 선택한다. 그리고 내부적인 통계를 기반으로 확률을 변경하고 가변 길이 코딩이 아닌 산술 코딩을 사용하여 좋은 압축 성능을 나타낸다.

CABAC의 부호화 과정은 (a) 이진화(binization: 0, 1)를 하고 (b) 컨텍스트 모델을 선택한 다음 (c) 산술 인코딩(arithmetic encoding)을 한다. 마지막으로 실제 코딩된 값을 기반으로 (d) 선택된 컨텍스트 모델의 확률(context model probability update)을 갱신한다[4,6,7].

디코더(decoder) 과정은 부호화된 스트림 데이터(stream data)를 받아 그림 1의 AIC 디코더 과정 즉, 인코더의 역 과정을 수행하면 된다.

## III. 시뮬레이션 (Simulation)

시뮬레이션 환경은 다음의 표 1과 같다.

표 1. 시뮬레이션 환경

OS	· Windows XP Professional
Tool	· Visual C++ 6.0 · Jasper JPEG-2000 library · JPEG library · H.264 ref. software JM80.zip
Test image	· RGB 24bits color image - Lena 512x512 size - Baboon 256x256 size

시뮬레이션은 2개의 RGB 테스트 영상을 visual C++ 6.0에서 컴파일된 JPEG과 Jasper JPEG-2000 그리고 AIC(수정된 H.264 ref. JM8.0)[7]의 입력영상으로 하였다. 각 코드의 양자화 파라미터 값과 비트 Rate를 변경시키면서 PSNR과 인코딩 속도를 측정해 본다.

그림 6은 512x512 Lena RGB 영상의 JPEG과 JPEG-2000 그리고 AIC의 bpp(bit pel pixel)에 따른 PSNR을 나타낸 것이다.

그림 7은 JPEG과 JPEG-2000 그리고 AIC를 이용하여 약 0.48 bpp(bit per pixel)로 부호화된 256x256 Baboon RGB 영상을 나타낸 것이다.

JPEG-2000의 영상이 smooth와 blur 처리되어 bit 손실이 있지만 시각적인 면에서는 가장 원본

영상과 유사해 보였고, JPEG과 AIC의 경우와 비교할 때 PSNR이 가장 높았다. 하지만, 인코딩 속도 면에서는 JPEG과 AIC보다 느렸다.

AIC의 경우는 JPEG 보다도 PSNR이 높았으나 인코딩 속도 면에서는 다소 느렸다.

적화의 필요성을 과제로 제시하였다.

참고문헌

[1] <http://www.jpeg.org>, the ISO JPEG standard committee  
 [2] <http://www.ece.uvic.ca/~mdadams/jasper>, the JasPer Project Home Page  
 [3] ISO/IEC 14496-2, Amend. 1, Info. tech. coding of audio visual object -Part 2: Visual, 2001  
 [4] ITU-T Rec. H.264/ISO/IEC 11496-10, "Advanced Video Coding", Final Committee Draft, Doc. JVT-E022, Sep. 2002  
 [5] D. Marpe, G Blättermann and T Wiegand, "Adaptive Codes for H.26L", ITU-T SG16/6 Doc. VCEG-L13, Eibsee, Germany, Jan. 2001  
 [6] Iain E. G. Richardson, "H.264 and MPEG-4 video compression: Video coding for Next-generation MultiMedia", John Wiley&Sons Ltd, 2003  
 [7] <http://www.vcodex.fsnet.co.uk/h.264.html>, "H.264/MPEG-4 Part 10 Tutorials"  
 [8] <http://bs.hhi.de/~suehring/tml>, H.264 ref. software JM80.zip  
 [9] Y. Arai, T. Agui, and M. Nakajima. "A fast DCT-SQ scheme for images", Trans. IEICE, vol. E-71, no. 11, pp. 1095-1097, Nov. 1988

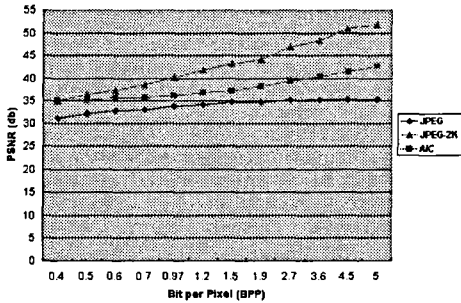


그림 6. PSNR(JPEG, JPEG-2000, AIC)  
 Fig. 6. PSNR(JPEG, JPEG-2000, AIC)



Original: Baboon 256x256, 24bpp      AIC: 0.48bpp, PSNR: 22.85db



JPEG: 0.47bpp, PSNR: 22.51db      JPEG-2000: 0.47bpp, PSNR: 23.17db

그림 7. 256x256 baboon 이미지의 PSNR  
 Fig. 7. PSNR of 256x256 baboon images

V. 결 론

본 논문에서는 정지 영상을 부호화하기 위해서 공간 영역에 예측을 통한 진보적 정지 영상 부호화(AIC) 알고리즘에 대해서 살펴보았다. 시뮬레이션에서 AIC 알고리즘은 JPEG 보다는 PSNR이 높았고, JPEG-2000보다는 낮을 것을 확인하였다. 또한, 인코딩 속도 면에서는 JPEG-2000보다는 다소 빨랐으나 JPEG 보다는 느렸다. 차후에 화질 개선(PSNR)과 고속 연산을 수행하기 위한 코드의 최