

IJG JPEG 복호기의 구조

서 대환^o · 이 호석
호서대학교 컴퓨터공학과

The Architecture of the IJG JPEG Decompressor

Dae Hwan, Seo^o · Ho Suk, Lee
Dept. of Computer Engineering, Hoseo University

요 약

본 논문은 Independent JPEG Group의 JPEG 복호과정에 대해서 설명한다. IJG JPEG의 복호과정은 JPEG 표준 사양과 동일한 entropy decoding, inverse quantization, inverse DCT, MCU disassembly 과정을 거친다. IJG JPEG의 특징적인 개념으로는 iMCU, rowgroup, context, huffman optimization, 2-pass quantization, upsampling, downsampling등이 있다. 본 논문은 IJG JPEG의 복호과정에 대한 전반적인 설명과 구성 모듈에 대한 기능을 기술한다.

1. 서 론

IJG(Independent JPEG Group)는 ISO/IEC의 JPEG 표준사양에 부합되게 JPEG 부호기(compressor)와 복호기(decompressor)를 개발하였으나 ISO/IEC와는 별개의 독립된 기관이다. IJG JPEG은 사실상 산업계 표준 JPEG을 목표로 개발되었으며, 이식성(portability)를 높이기 위해서 C 프로그래밍 언어로 개발되었다. 이 JPEG 소프트웨어를 "Independent JPEG Group's JPEG software"라고한다. 이 소프트웨어는 Unix와 MS-DOS에서의 사용을 목적으로 개발되었지만 Apple 등 여러 플랫폼에서 사용될 수도 있다. IJG JPEG은 JPEG baseline, extended sequential, progressive DCT 를 지원하고 있지만 아직까지 hierarchical 그리고 lossless JPEG은 지원하지 않는다.[1,5]

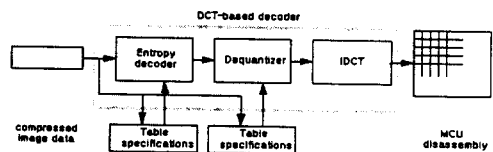
IJG JPEG 프로그램은 JPEG 표준사양에 일치하면서도 산업계 표준 JPEG을 목표로 개발되었기 때문에 높은 수준으로 모듈(module)화 되어있으며, 각 모듈의 작성방법도 비교적 일정한 규칙을 따르고 있다. 또한 프로그램의 가독성(readability)을 높이기 위해서 각 모듈에서 동일한 함수(function) 이름을 사용하고 있으며, 이를 function pointer를 사용하여 혼돈이 없도록 하고 있다.

IJG JPEG의 복호기는 JPEG proper와

postprocessing으로 크게 나뉘어진다. 본 논문에서는 복호 처리 과정에서 IJG JPEG의 특징으로 MCU(Minimum Coded Unit)의 확장된 개념인 iMCU, rowgroup, context에 대하여 기술하고, IJG JPEG의 복호과정에 대하여 기술한다.

2. IJG JPEG 복호기 전체 시스템 구조

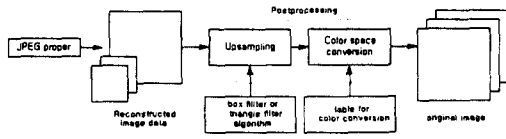
IJG JPEG 복호기는 크게 JPEG proper와 postprocessing으로 나뉘어진다. JPEG proper는 JPEG 표준에 따르는 복호 과정을 포함하고 있으며, [그림1]은 JPEG proper의 복호 과정을 나타내고 있다.



[그림 1] JPEG Proper

Postprocessing은 이미지를 구성하는 각 component의 sampling factor에 따라서 upsampling과 color space conversion을 수행하는 routine을 포함하고 있으며, 복호 속도를 향상시키기 위해 option으로 color quantization을 할 수가 있다. [그림2]는

postprocessing routine을 나타낸다.



[그림 2] Postprocessing

이 외에도 몇개의 중요한 controller, marker parser, data source module을 포함하고 있으며, memory management와 error handling module은 부호기와 공유한다. 다음은 복호 과정이다.[4,5]

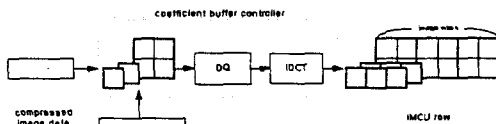
1. JPEG 복호 object의 할당과 초기화
2. JPEG data를 읽어들이기 위한 source의 지정
3. image 정보를 읽어들이기 위해 jpeg_read_header() 호출
4. master 복호 control의 초기화
5. active module의 선택
6. image height만큼 jpeg_read_scanlines() 호출
7. 복호 object 해제

3. Control module의 구조

JPG JPEG 복호기의 controller는 coefficient buffer, main buffer, postprocessing controller로 구성되어 있다.

3-1. Coefficient buffer controller

Coefficient buffer controller는 JPEG proper의 최상위 레벨에 위치하고 있으며, 기본적으로 main buffer controller에게 interleaved MCU row ("iMCU")를 제공한다. [그림3]에서는 iMCU생성에 대해 나타낸다.



[그림 3] Coefficient controller (한 블록은 8x8)

iMCU는 [그림3]에서와 같이 MCU를 이미지를 width 만큼 결합시킨 단위로서, 각 component에 대한 sample rows는 다음과 같이 정의 될 수 있다.

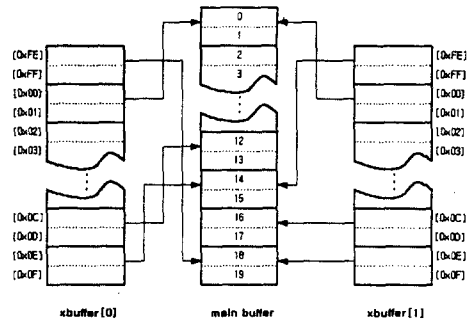
$$vertical_sampling_factor * DCT_scaled_size$$

iMCU는 Huffman, dequantization, IDCT까지 처리된 데이터를 포함한다.

3-2. Main buffer controller

Main buffer controller는 JPEG proper와 postprocessing사이에서 위치해 있으며, coefficient controller로부터 받아들인 iMCU 데이터를 rowgroup단위로 postprocessing에 제공한다. [그림4]는 main buffer

controller의 버퍼 구조를 나타내고 있다.



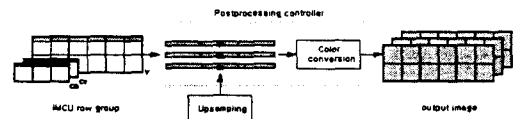
[그림 4] iMCU buffer의 구조

Main buffer는 실제 iMCU 데이터를 가지고 있으며, 두 개의 xbuffer는 main buffer를 포인트하고 있다. 각 xbuffer의 최상위의 포인터와 최하위의 포인터는 main buffer의 서로 다른 row group을 가리키고 있다. 이 row group을 context row라고 한다.[5]

Main controller는 iMCU의 각 row group을 3가지의 context_state에 따라 처리한다. Context_state는 prepare_for_iMCU, process_iMCU, postponded_row이다. prepare_for_iMCU는 새로운 iMCU를 처리하기 위한 초기화 작업을 한후 바로 process_iMCU_state로 바뀐다. Process_iMCU는 실제 iMCU의 각 row group 단위로 처리하고, 마지막 row group을 남겨두고 postponded_row_state로 바뀐다. Postponded_row_state에서는 마지막 row group을 처리하지 않은 상태로 xbuffer의 index가 switch되고, 다음 처리될 iMCU를 main buffer로 읽어들이기 후 이전 iMCU에서 처리되지 않은 row group을 처리한다.

3-3. Postprocessing controller

Postprocessing controller는 upsampling, color conversion을 다루는 것으로서, main controller로부터 받아들인 각 row group 단위로 처리를 한다. [그림5]는 postprocessing controller에 대해 나타낸다.



[그림 5] Postprocessing controller

4. JPEG 알고리즘

JPG JPEG의 복호 과정은 JPEG 표준을 따른다. 그 과정은 기본적으로 data source module로부터 압축 데이터를 읽어들이어 Huffman entropy encoding을 수행하고, 8x8 block에 대해 dequantization과 inverse DCT를 수행 후 마지막으로 upsampling과 color space

conversion을 수행한다. IJG JPEG 의 IDCT와 upsampling알고리즘에 대하여 살펴보겠다.

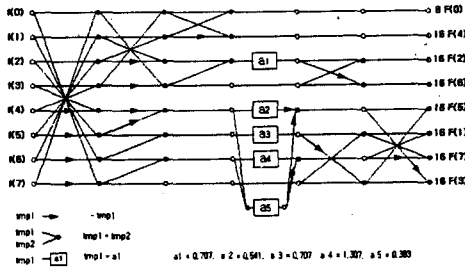
4-1. Arai, Agui, and Nakajima의 DCT algorithm

IJG에서는 IDCT과정에 AA&N 알고리즘을 활용하고 있다. 일반적인 2-Dimension IDCT 공식은 다음과 같다.[2,3]

$$S_{xy} = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C_u C_v S_{18} \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

where $C_u = C_v = \frac{1}{\sqrt{2}}$ for $u, v = 0$; otherwise $C_u = C_v = 1$

위의 IDCT식을 통해서 8*8 블록을 수행할 경우 곱셈이 1024, 덧셈이 896번 수행된다. 이러한 많은 연산을 피하기 위해서 IJG JPEG은 Fast DCT를 기반으로 한 AA&N 알고리즘을 사용한다. [그림6]은 8-point 1-D에 대한 AA&N 알고리즘에 대한 흐름을 나타낸다.



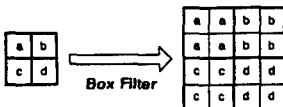
[그림 6] Flowgraph for 1-D DCT adapted from Arai, Agui, and Nakajima

그림에서 보듯이 8-point 1-D의 IDCT에 대해서 곱셈은 5번, 덧셈은 29번이 수행된다. 즉, 8*8 block에서 각 column과 row에 대해서 8-point 1-D IDCT를 수행하면 전체 곱셈은 80번, 덧셈은 464번이 수행된다.

IJG JPEG에서는 여러가지 IDCT 알고리즘을 사용하고 있으며, 그 목적은 연산의 횟수를 줄이는 것이다.[1.5]

4-2. Upsampling 알고리즘

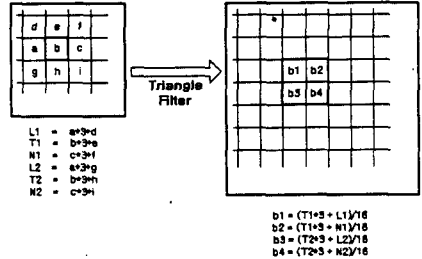
IJG JPEG에서는 downsample된 컴포넌트의 upsampling을 위한 두가지의 방식을 제공한다. 첫번째는 downsample된 입력 샘플 값을 이에 상응하는 출력 픽셀에 단순히 복사하는 방식으로, 이를 "Box filter"라고 한다. Box filter는 단순한 방식만큼 빠른 속도의 복호를 할 수 있다. [그림7]은 box filter를 통한 upsampling을 나타내고 있다.[3]



[그림 7] Box Filter (중복복사 pixel)

또 다른 upsampling으로 fancy processing 알고리즘이 있다. 이를 "triangle filter"라고 한다. 이는 속도와

이미지의 quality를 절충시킬 수 있는 알고리즘으로, 출력 픽셀은 본래의 입력 픽셀값과 인접 픽셀값 사이의 linear interpolation에 의해서 생성된다. [그림 8]은 triangle filter를 통한 upsampling을 나타낸다.



[그림 8] Triangle Filter (중복복사 pixel)

Upsampling과정에서 center sample에 3/4의 비중을 두고 인접한 픽셀에 1/4의 비중을 둠으로서, 출력 픽셀 값을 결정하게 된다. Triangle filter를 통한 upsampling은 Box filter 보다 이미지의 quality를 향상시킬 수가 있다.[3.5]

5. 결론

본 논문에서는 IJG(Independent JPEG Group)의 복호기에 대한 전반적인 내용과 중요한 특징 그리고 중요한 controller에 대하여 살펴보았다. IJG JPEG은 산업계 표준 JPEG을 목표로 개발된 만큼 프로그램이 매우 체계적으로 개발되어 있으며 높은 수준으로 module화 되어 있다.

IJG의 Huffman decoding, color space conversion등에 대한 알고리즘을 기술하지는 않았지만, 이러한 복호과정은 JPEG 표준을 따르고 있다

차후 IJG JPEG에서 제공되는 memory management routine과 color quantization에 대하여 연구할 예정이다.

6. 참고문헌

- [1] William B. Pennebaker, Joan L. Mitchell, "JPEG Still Image Data Compression Standard", Van Nostrand Reinhold, 1993.
- [2] Ralf Steinmetz, Klara Nahrstedt, "Multimedia: Computing, Communications & Applications", Prentice Hall Inc., 1995.
- [3] R. J. Clarke, "Digital Compression of Still Images and Video" Academic Press, 1995.
- [4] IJG, "IJG JPEG Source Code", 1996.
- [5] IJG, "IJG JPEG Document", 1996.