JPEG (HW3)

MediaLab.
Dowan Kwon

Contents of Table

❖ JPEG Algorithm

bitset library 활용 가능

① Color Space Conversion - Lossless
② Chroma subsampling - Lossy
③ DCT (Discrete Cosine Transform) - Lossless
④ Quantization (Quantization Table) - Lossy
⑤ Zigzag Scan
⑥ DC : DPCM (Differential Pulse Code Modulation)
AC : RLC (Run Length Coding)
⑤ Entropy Coding (Huffman Coding) - Lossless
⑥ File 생성

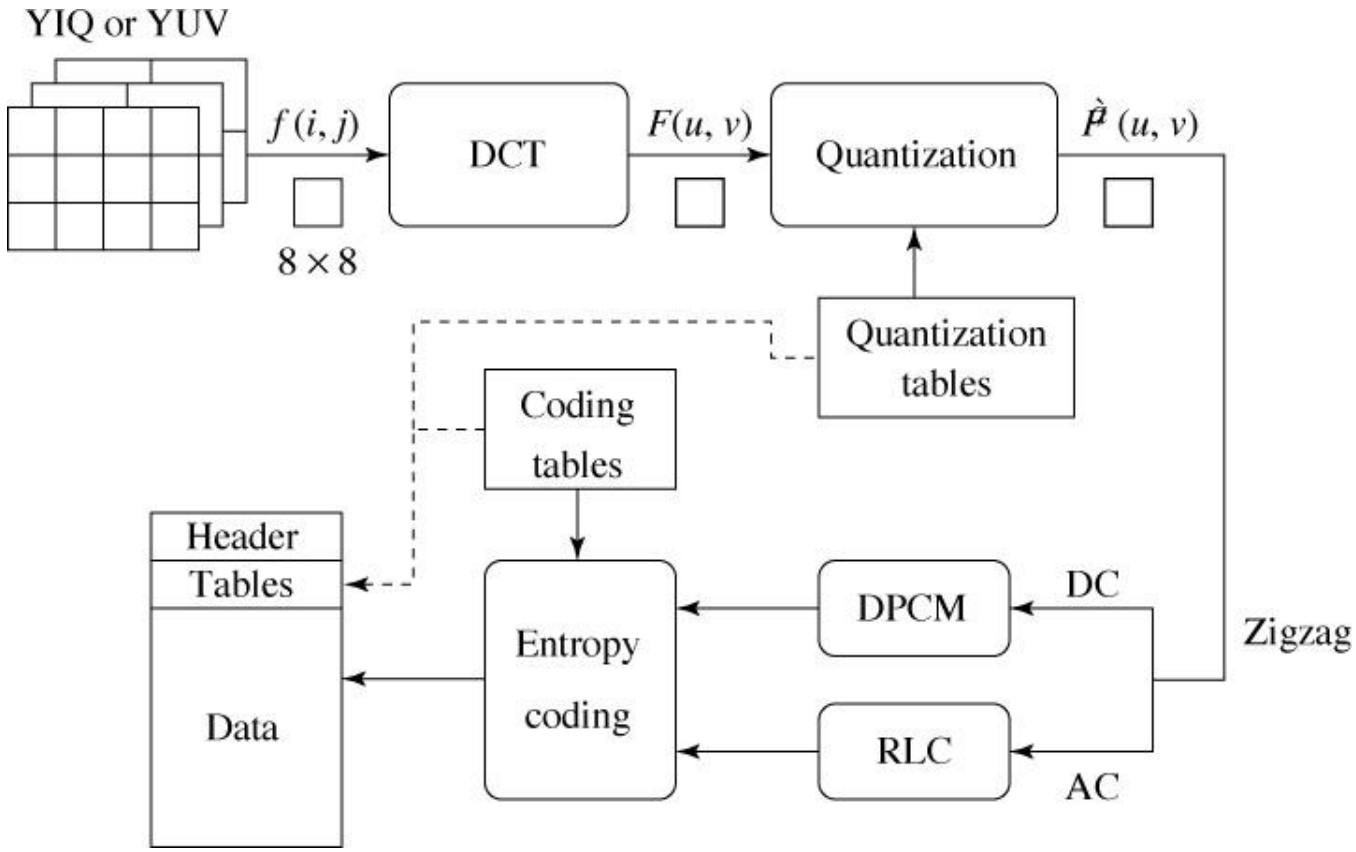


Fig 1. JPEG 알고리즘

❖ JPEG Algorithm

③ DCT (Discrete Cosine Transform)

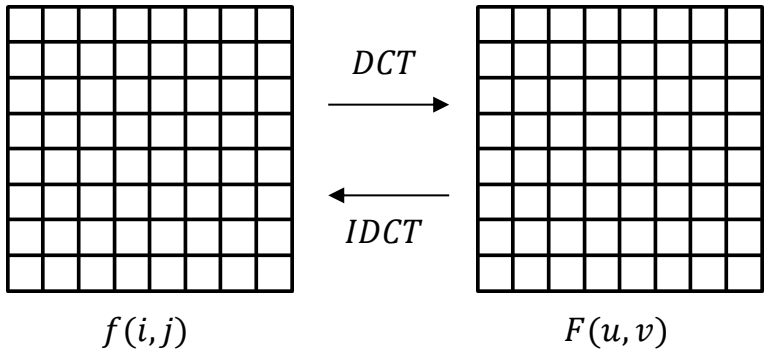
- 입력된 Bmp 파일을 8×8 MacroBlock으로 나눠서 DCT 한다. $f(i,j) \rightarrow F(u,v)$

$$F(u,v) = \frac{C(u)C(v)}{4}\sum_{i=0}^{7}\sum_{j=0}^{7}\cos\left(\frac{(2i+1)u\pi}{16}\right)\cos\left(\frac{(2j+1)v\pi}{16}\right)f(i,j), \qquad C(0) = \frac{1}{\sqrt{2}}, C(p) = 1\ (p \neq 0)$$

Fig 2. DCT

$$f(i,j) = \sum_{u=0}^{7}\sum_{v=0}^{7}\frac{C(u)C(v)}{4}\cos\left(\frac{(2i+1)u\pi}{16}\right)\cos\left(\frac{(2j+1)v\pi}{16}\right)F(u,v), \qquad C(0) = \frac{1}{\sqrt{2}}, C(p) = 1\ (p \neq 0)$$

Fig 3. IDCT



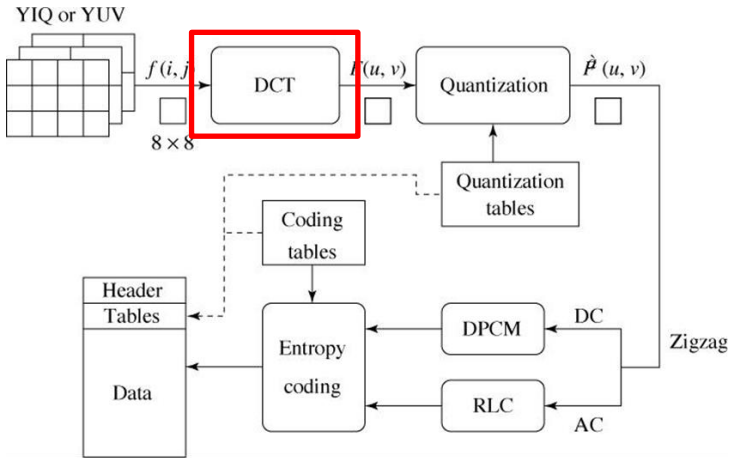


Fig 1. JPEG 알고리즘

❖ JPEG Algorithm

④ Quantization (Quantization Table)

- 미리 정한 Quantization Table를 이용해 Quantization 을 한다. $F(u,v) \rightarrow \hat{F}(u,v)$

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Fig 4. Quantization Table

$$\hat{F}(u,v) = Round\left(\frac{F(u,v)}{Q(u,v)}\right)$$

Fig 5. Formula of Quantization

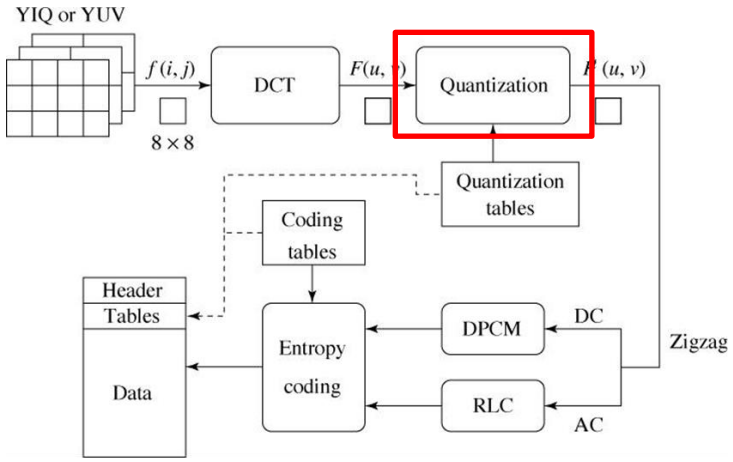
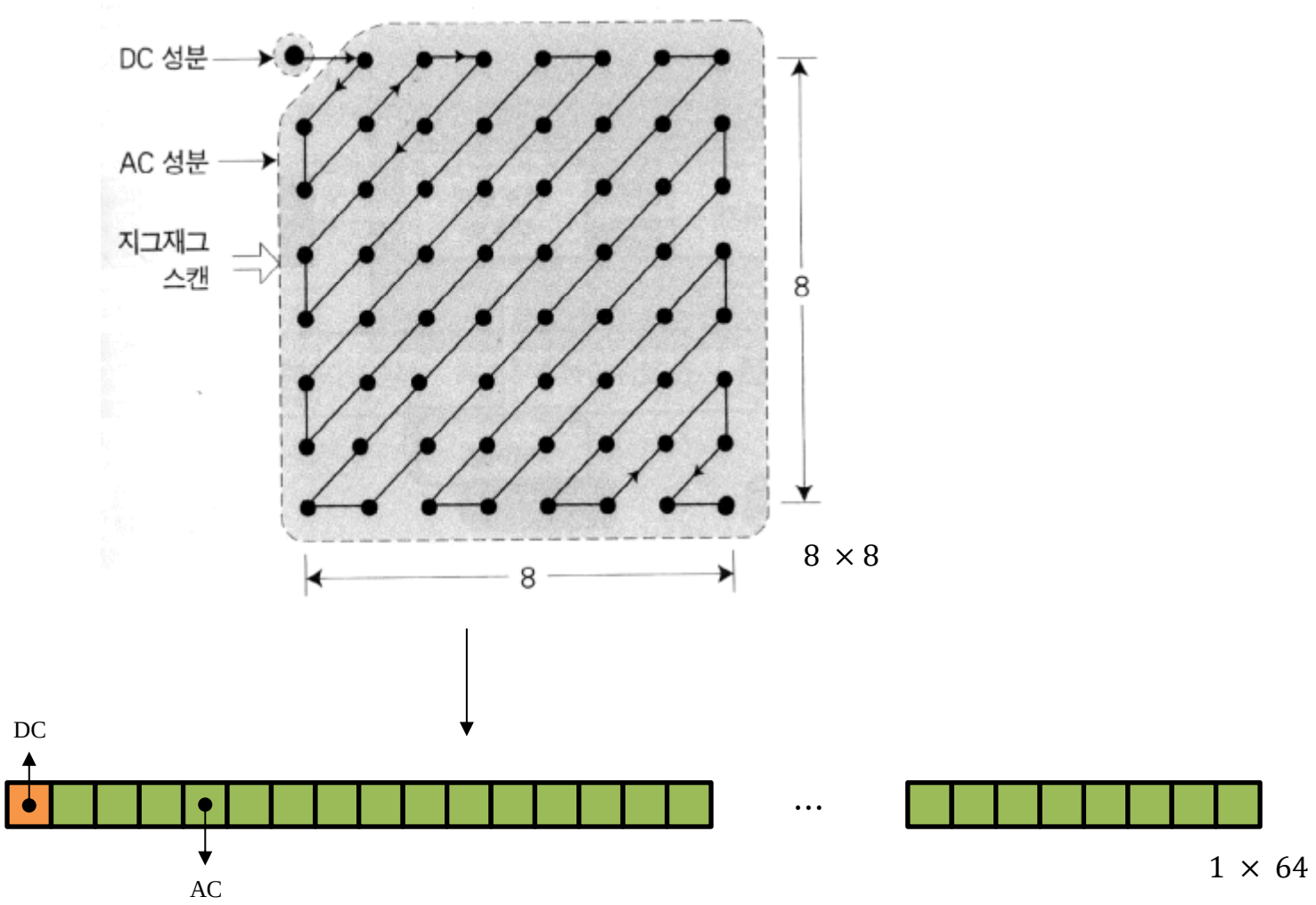


Fig 1. JPEG 알고리즘

❖ JPEG Algorithm

⑤ Zigzag Scan

- $\hat{F}(u,v)$를 Zigzag Scan 한다.



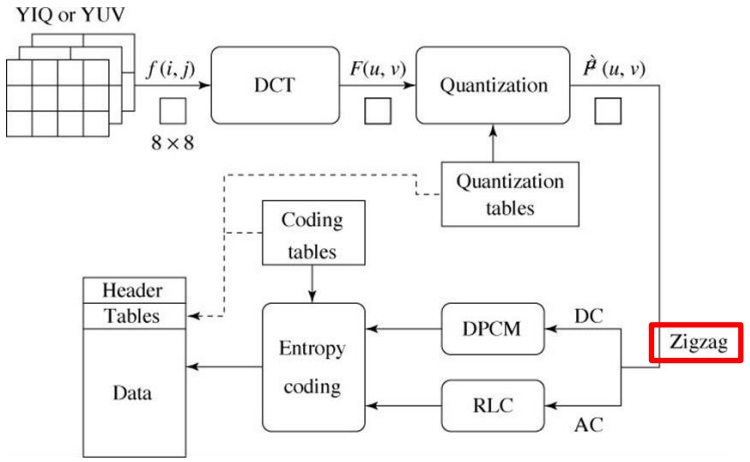
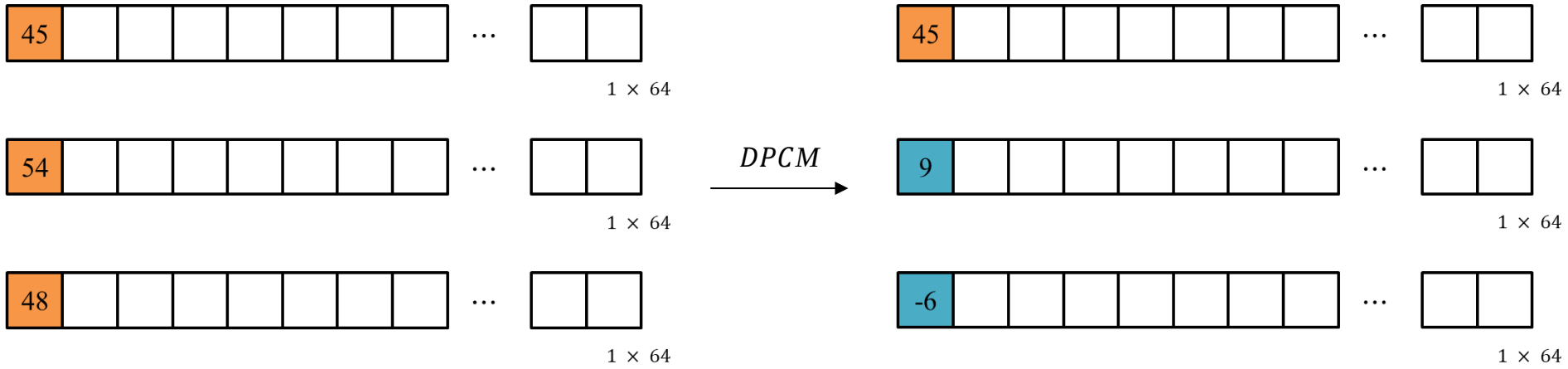


Fig 1. JPEG 알고리즘

❖ JPEG Algorithm

⑥ DC : DPCM (Differential Pulse Code Modulation)

- DC값들을 따로 모아서 DPCM을 수행



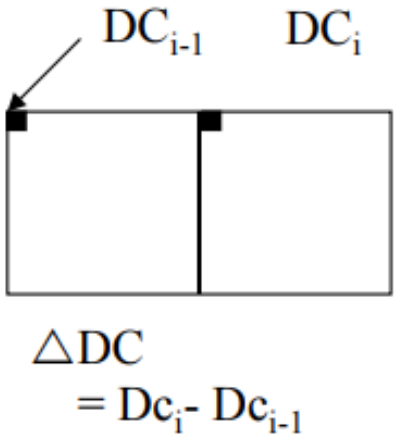


Fig 6. Formula of DPCM

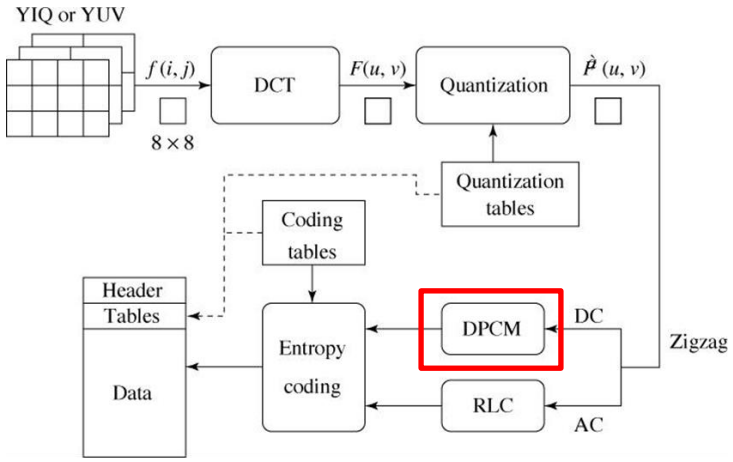
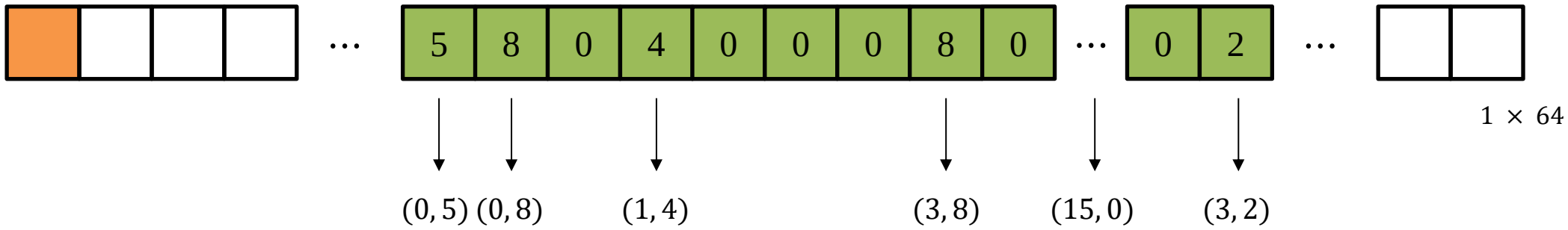


Fig 1. JPEG 알고리즘

❖ JPEG Algorithm

⑥ AC : RLC (Run Length Coding)

- AC 값들을 RLC를 수행
- (Skip, Value) : Skip의 최대 길이는 15
- (0, 0) end of block



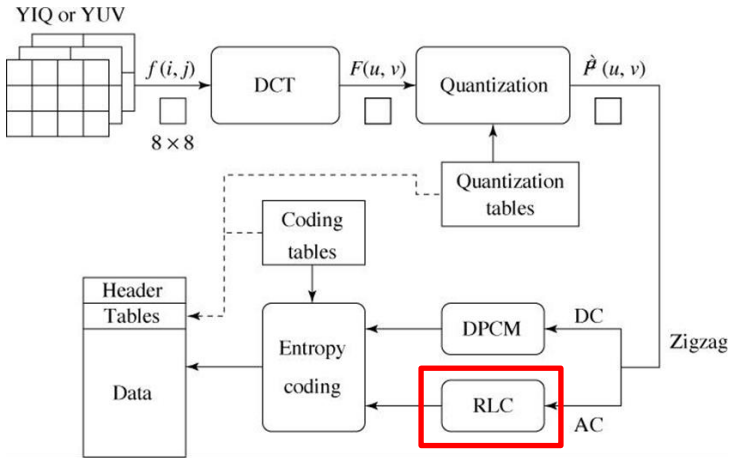


Fig 1. JPEG 알고리즘

❖ JPEG Algorithm

⑦ 모은 값들을 Huffman encoding 한다.

- DPCM 완료한 DC 값의 길이로 Huffman encoding → DC Length Huffman Table
- RLC 완료한 AC값들로 Huffman encoding → AC Huffman Table

SIZE	Code Length	Code
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Fig 7. Example of DC Length Huffman Table

※ DC Example

DC sequence : 48 **40** …

DPCM sequence : 48 **-8** …

-8 : 1010111

0111 : -8 (1의 보수를 취한 2진수의 표현)

101 : DC Length Huffman Table의 Size 4 일 때

SIZE	Value	Code
0	0	---
1	-1,1	0,1
2	-3, -2, 2,3	00,01,10,11
3	-7,..., -4, 4,..., 7	000,..., 011, 100,...111
4	-15,..., -8, 8,..., 15	0000,..., 0111, 1000,..., 1111
.		.
.		.
11	-2047,..., -1024, 1024,... 2047	...

Fig 6. DC Component Size and Value Table

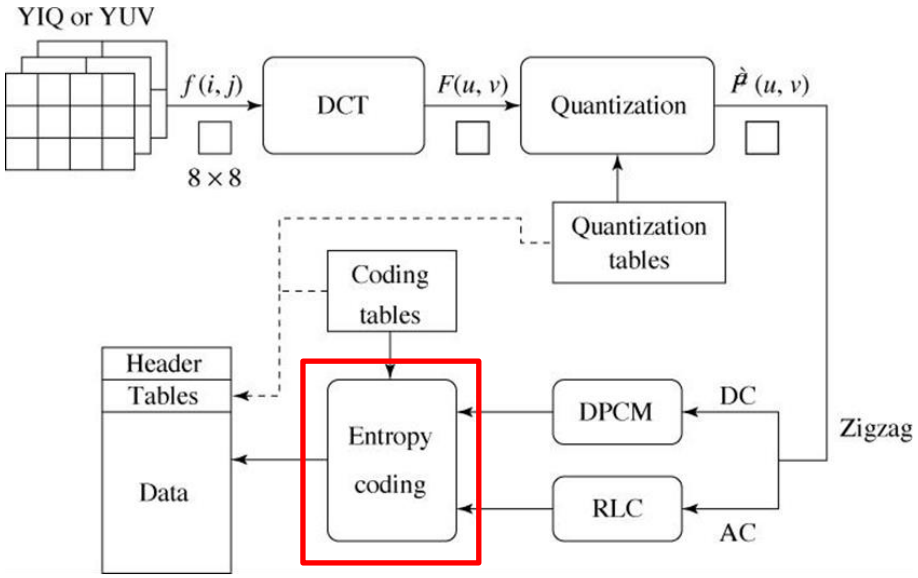


Fig 1. JPEG 알고리즘

❖ JPEG Algorithm

⑦ 모은 값들을 Huffman encoding 한다.

- DPCM 완료한 DC 값의 길이로 Huffman encoding → DC Length Huffman Table
- RLC 완료한 AC값들로 Huffman encoding → AC Huffman Table

SIZE	Value	Code
0	0	---
1	-1,1	0,1
2	-3, -2, 2,3	00,01,10,11
3	-7,..., -4, 4,..., 7	000,..., 011, 100,...111
4	-15,..., -8, 8,..., 15	0000,..., 0111, 1000,..., 1111
.		.
.		.
11	-2047,..., -1024, 1024,... 2047	...

Run/ SIZE	Code Length	Code
0/0	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	1111000
0/7	8	11111000
0/8	10	1111110110
0/9	16	1111111110000010
0/A	16	1111111110000011

Run/ SIZE	Code Length	Code
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	1111111110000100
1/7	16	1111111110000101
1/8	16	1111111110000110
1/9	16	1111111110000111
1/A	16	1111111110001000
... 15/A	More	Such rows

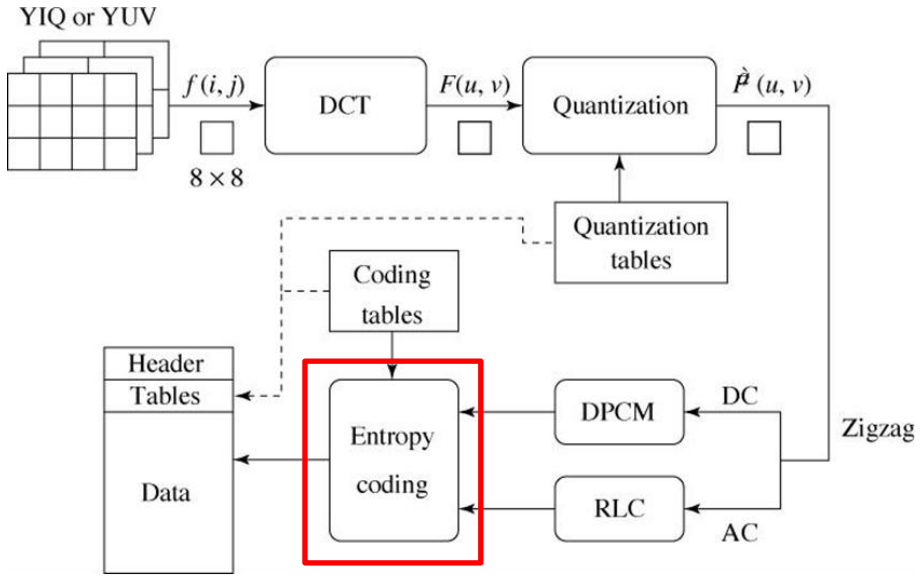
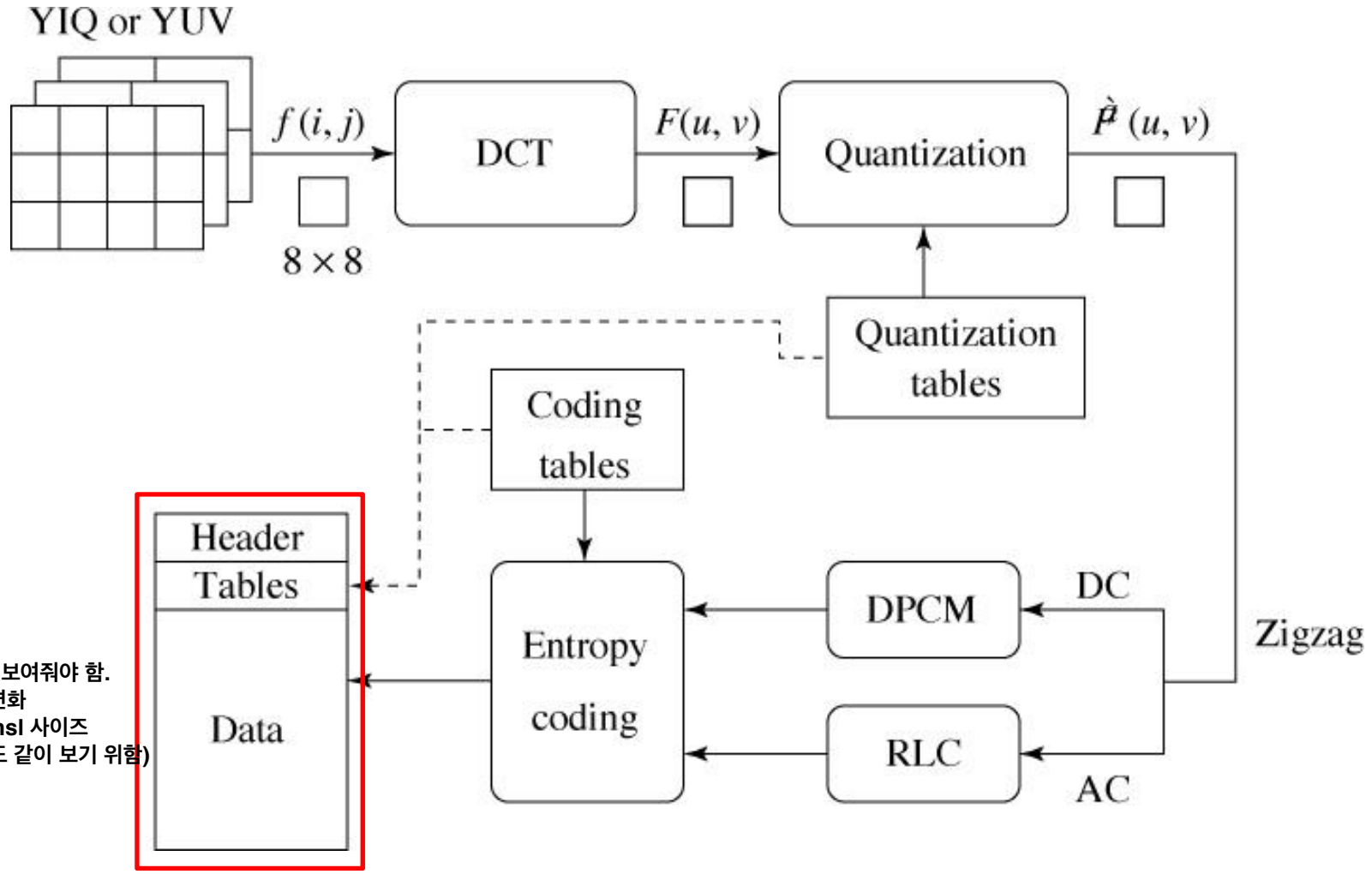


Fig 1. JPEG 알고리즘

❖ JPEG Algorithm

⑧ 압축 파일에 Quantization Table, DC Length Huffman Table, AC Huffman Table 등 Decoding에 필요한 정보들을 담고 데이터들을 저장한다.



얼마나 압축됐는지 비교한거 보여줘야 함.
1. 데이터 사이즈 변화
2. 입력 bmp 사이즈 와 msl 사이즈
(압축으로 인한 헤더파일의 크기도 같이 보기 위함)

Fig 1. JPEG 알고리즘

❖ Assignment

- JPEG Encoder / Decoder
- 입력 영상 Lena.bmp

Input: Lena.bmp

JPEG Encoder → Bitstream (Lena.msl) → JPEG Decoder

Output: RecLena.bmp

헤더도 전달해서 bmp로 디스플레이 하는게 어려우면
어떻게든 디스플레이 하면 됨.

- 소스코드 (.c, .cpp, .py, .ipynb), 실행파일 (c/c++의 경우)
- 보고서 (.pdf) 파일 (bmp original input data VS jpeg compressed output data size)
- 이름_학번_HW3.zip 형식으로 제출해주세요


Thank
you