

# BMP 파일 이해

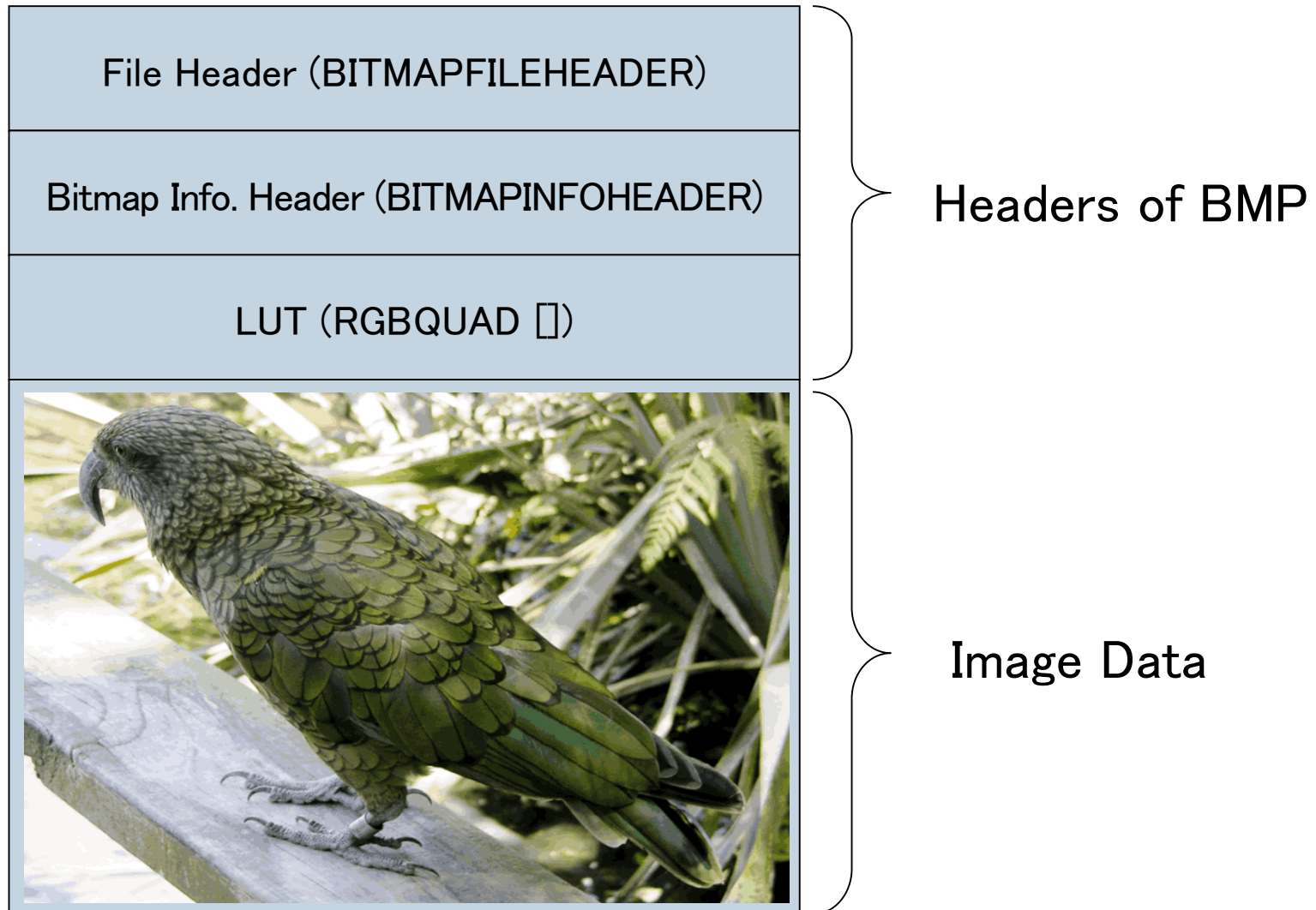
김성영교수  
금오공과대학교  
컴퓨터공학과

# 학습 내용

---

- BMP 파일의 구조
- BMP 파일의 특징

# BMP 파일의 구조



```
typedef struct tagBITMAPFILEHEADER
{
    WORD          bfType;          // Specifies the file type
                                   // It must be "BM" (4D42)

    DWORD         bfSize;          // 파일의 크기 (byte)

    WORD          bfReserved1;     // reserved (항상 0)

    WORD          bfReserved2;     // reserved (항상 0)

    DWORD         bfOffBits;       // 픽셀 데이터의 시작 오프셋
} BITMAPFILEHEADER;
```

```
typedef struct tagBITMAPINFOHEADER {  
    DWORD biSize;                // 구조체의 크기 (bytes)  
    LONG   biWidth;               // 비트맵의 가로 길이 (pixels)  
    LONG   biHeight;             // 비트맵의 세로 길이 (pixels)  
    WORD   biPlanes;             // 비트 플레인 수 (항상 1)  
    WORD   biBitCount;           // 픽셀당 비트수 (1,4,8,16,24,32)  
    DWORD  biCompression;        // 압축 유형 (BI_RGB)  
    DWORD  biSizeImage;          // 비트맵 데이터의 크기 (bytes)  
    LONG   biXPelsPerMeter;       // 수평 해상도 (pixels/meter)  
    LONG   biYPelsPerMeter;       // 수직 해상도 (pixels/meter)  
    DWORD  biClrUsed;            // LUT에 포함된 칼라 인덱스의 개수  
    DWORD  biClrImportant;       // 비트맵을 화면에 출력하기 위해  
                                   // 사용된 칼라 인덱스의 개수  
} BITMAPINFOHEADER;
```

```
typedef struct tagRGBQUAD
{
    BYTE        rgbBlue;           // B component
    BYTE        rgbGreen;          // G component
    BYTE        rgbRed;            // R component
    BYTE        rgbReserved1;      // reserved
} RGBQUAD;
```

bfOffBits: 0x36

biWith: 0x96(150)

biHeight: 0x64(100)

biBitCount: 0x18(24)

biClrUsed: 0x00(0)

true color image

0	42 4D C8 B0 00 00 00 00 00 00 00 36 00 00 00 28 00	B M È °				6	(
10	00 00 96 00 00 00 64 00 00 00 01 00 18 00 00 00	-	d		↑		
20	00 00 92 B0 00 00 12 0B 00 00 12 0B 00 00 00 00	,	°	↑	♂	↑	♂
30	00 00 00 00 00 00 36 4A 59 40 53 60 4B 5D 6B 68				6 J Y @ S ` K ] k h		
40	79 88 7F 91 9B 8C 9D A6 97 A8 B0 90 A2 A9 8A 9C	y ^ ▯ ‘ > ⌘   - ° ¢ ⊙ Š œ					
50	A4 94 A5 AD 9B AE B7 8A 9A A6 80 91 9F A8 B7 BE	¤ ” ¥ - > ⊙ · Š š ! € ‘ Ÿ · ¾					
60	C5 CE CE CF D6 D6 C9 D0 D2 C6 CD CF C3 CB CE C6	À Î Ï Ï Ö Ö É Ð Æ Í Î Ä Ê Æ					
70	CD D1 C8 CF D2 C9 CF D2 CB D1 D2 CC D1 D3 CE D3	Í Ñ È Ï Ò É Ï Ò Ë Ñ Ò Ì Ñ Ó Í Ó					
80	D4 CE D3 D5 CE D3 D5 D0 D4 D5 CF D4 D5 CE D2 D4	Ô Î Ó Õ Î Ó Ö Ð Ô Õ Î Ô Õ Î Ô					
90	D0 D4 D5 D1 D4 D6 D4 D7 D7 D3 D6 D7 D3 D6 D6 D2	Ð Ô Õ Ñ Ô Õ Ô × × Ó × Ó Ô Ö Ò					
A0	D5 D6 D2 D6 D7 D5 D8 D9 D4 D7 D8 D2 D6 D8 D3 D7	Õ Ö Ò Ô × Õ Ø Õ × Ø Ò Ô Ø ×					
B0	D9 D3 D6 D8 D4 D8 DA D5 D9 DB D5 D8 DA D5 D9 DB	Ù Ó Ø Ø Ø Ú Õ Ù Ô Ø Ø Ú Õ Ù Ô					
C0	D3 D7 D9 D4 D8 DB D6 DA DC D3 D8 DB D1 D6 D9 D4	Ó × Ù Ô Ø Ú Ó Ú Û Ó Ø Ú Ñ Ö Ù Ô					
D0	D9 DC D3 D9 DB CC D0 D3 AB C1 C2 63 A1 95 54 8E	Û Ü Ó Ù Ô Û Û Ð Ó « Á Â ¸ ¡ • T Ž					
E0	81 6F A5 98 6B 9F 91 60 90 85 5C 95 88 50 85 7B	ø ¥ ~ k Ÿ ‘ ˘ ... \ • ^ P ... {					



bfOffBits: 0x436

biWith: 0x96(150)

biHeight: 0x64(100)

biBitCount: 0x08(8)

biClrUsed: 0x00(0)

gray-scale image

0	42	4D	98	3F	00	00	00	00	00	00	36	04	00	00	28	00	B	M	~	?				6	↓	(					
10	00	00	96	00	00	00	64	00	00	00	01	00	08	00	00	00		-		d		↑	▣								
20	00	00	62	3B	00	00	12	0B	00	00	12	0B	00	00	00	00		b	;	↑	σ	↑	σ								
30	00	00	00	00	00	00	00	00	00	01	01	01	00	02	02									↑	↑	↑	↑	↑	↑	↑	
40	02	00	03	03	03	00	04	04	04	00	05	05	05	00	06	06	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
50	06	00	07	07	07	00	08	08	08	00	09	09	09	00	0A	0A	-	●	●	●	▣	▣	▣								
60	0A	00	0B	0B	0B	00	0C	0C	0C	00	0D	0D	0D	00	0E	0E		σ	σ	σ	♀	♀	♀								
70	0E	00	0F	0F	0F	00	10	10	10	00	11	11	11	00	12	12	♫	♫	♫	♫	♫	♫	♫	♫	♫	♫	♫	♫	♫	♫	♫
80	12	00	13	13	13	00	14	14	14	00	15	15	15	00	16	16	↑	!!	!!	!!	♀	♀	♀	↑	↑	↑	↑	↑	↑	↑	↑
90	16	00	17	17	17	00	18	18	18	00	19	19	19	00	1A	1A	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
A0	1A	00	1B	1B	1B	00	1C	1C	1C	00	1D	1D	1D	00	1E	1E	→	←	←	←											
B0	1E	00	1F	1F	1F	00	20	20	20	00	21	21	21	00	22	22								!	!	!	"	"	"	"	"
C0	22	00	23	23	23	00	24	24	24	00	25	25	25	00	26	26	"	#	#	#	\$	\$	\$	%	%	%	&	&	&	&	&
D0	26	00	27	27	27	00	28	28	28	00	29	29	29	00	2A	2A	&	'	'	'	(	(	(	)	)	)	*	*	*	*	*
E0	2A	00	2B	2B	2B	00	2C	2C	2C	00	2D	2D	2D	00	2E	2E	*	+	+	+	,	,	,	-	-	-	.	.	.	.	.



0	42	4D	C8	B0	00	00	00	00	00	00	36	00	00	00	28	00
10	00	00	96	00	00	00	64	00	00	00	01	00	18	00	00	00
20	00	00	92	B0	00	00	12	0B	00	00	12	0B	00	00	00	00
30	00	00	00	00	00	00	36	4A	59	40	53	60	4B	5D	6B	68
40	79	88	7F	91	9B	8C	9D	A6	97	A8	B0	90	A2	A9	8A	9C
50	A4	94	A5	AD	9B	AE	B7	8A	9A	A6	80	91	9F	A8	B7	BE
60	C5	CE	CE	CF	D6	D6	C9	D0	D2	C6	CD	CF	C3	CB	CE	C6



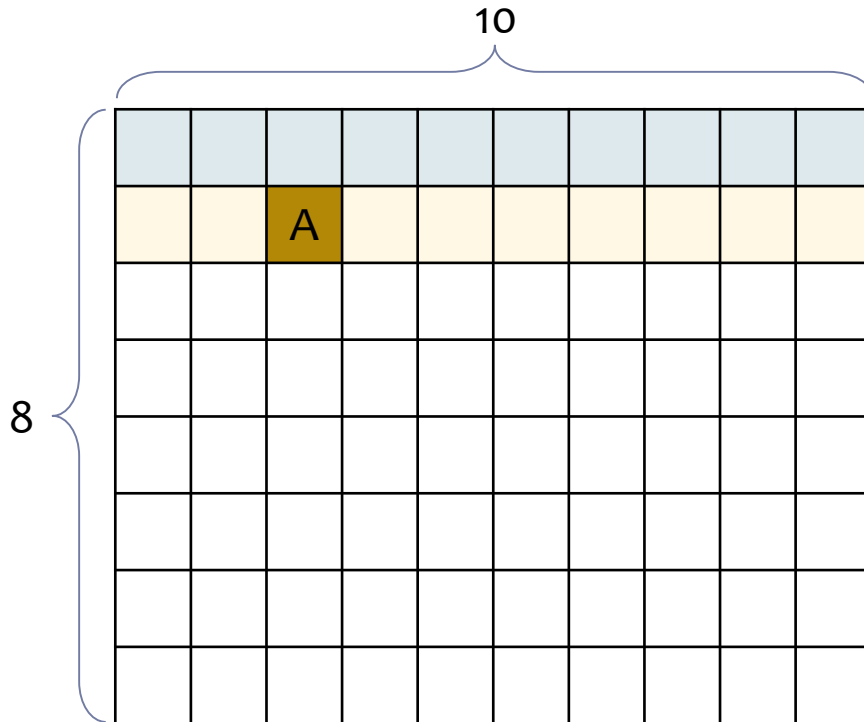
0	42	4D	C8	B0	00	00	00	00	00	00	36	00	00	00	28	00
10	00	00	96	00	00	00	64	00	00	00	01	00	18	00	00	00
20	00	00	92	B0	00	00	12	0B	00	00	12	0B	00	00	00	00
30	00	00	00	00	00	00	00	00	FF	40	53	60	4B	5D	6B	68
40	79	88	7F	91	9B	8C	9D	A6	97	A8	B0	90	A2	A9	8A	9C
50	A4	94	A5	AD	9B	AE	B7	8A	9A	A6	80	91	9F	A8	B7	BE
60	C5	CE	CE	CF	D6	D6	C9	D0	D2	C6	CD	CF	C3	CB	CE	C6





# BMP 파일의 특징

$$(x, y) \Rightarrow [width * y + x]$$

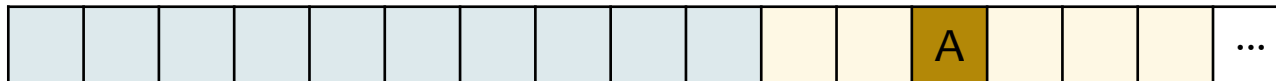


2차원 배열: data

$$A \Rightarrow \text{data}[1][2]$$

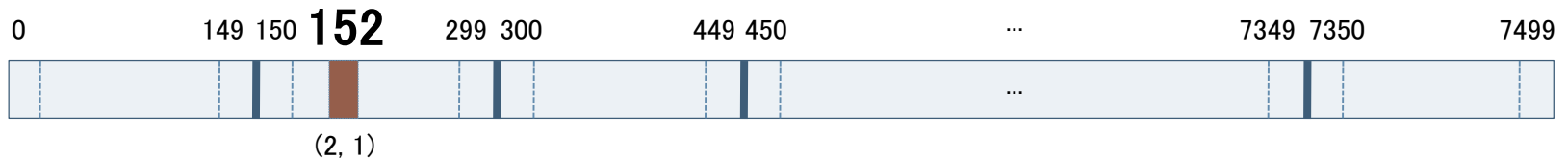
1차원 배열: data

$$A \Rightarrow \text{data}[12] \Rightarrow [1 * 10 + 2]$$

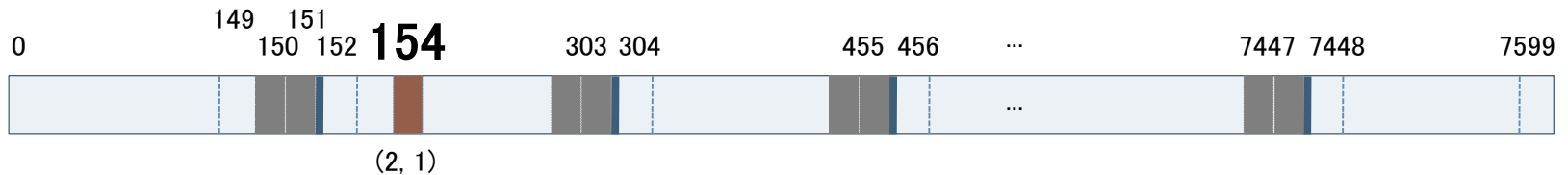


$$(x, y) \Rightarrow [widthStep * y + x]$$

150 x 50 8bit gray-scale image       $(x, y) = (2, 1) ? \Rightarrow [152] = [150 * 1 + 2]$



각 행의 데이터는 4byte의 배수로 저장



$$(x, y) = (2, 1) \Rightarrow [154] = [152 * 1 + 2]$$

# widthStep

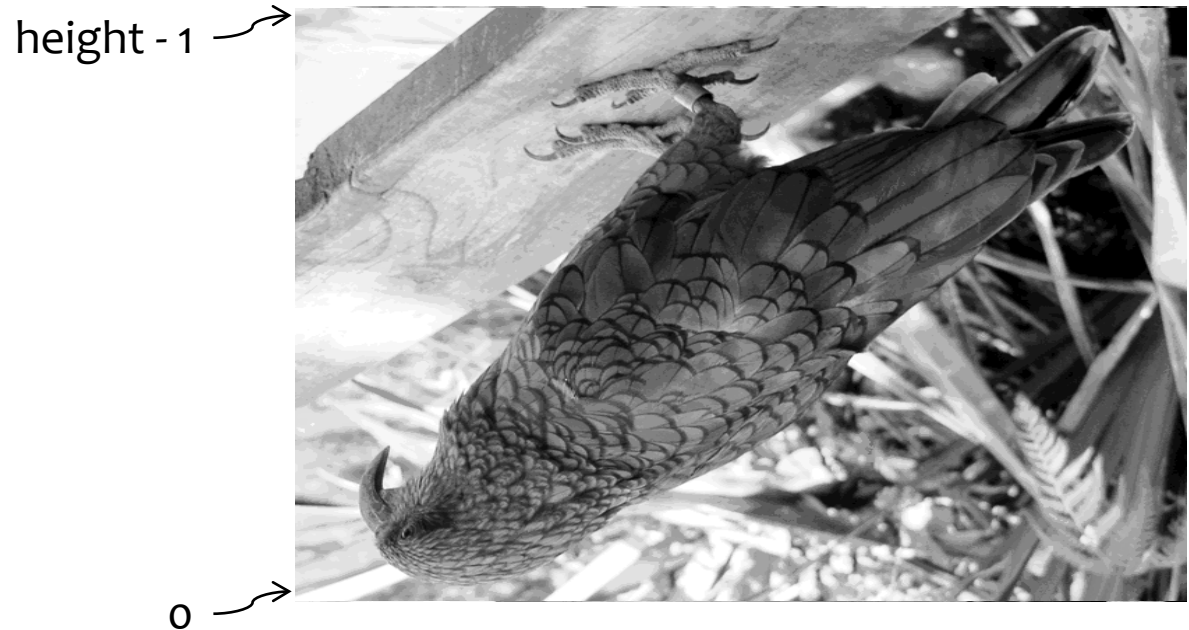
$$\mathit{width} \Rightarrow \mathit{widthStep} = ((\mathit{width} + 3) / 4) * 4$$

width가 150인 8bit gray-scale image의 경우

$$\mathit{widthStep} = ((150+3) / 4) * 4 = 38 * 4 = 152$$

$$(x, y) \Rightarrow [\mathit{widthStep} * y + x]$$

# Vertical Flip



$$(x, y) \Rightarrow [widthStep * (\textit{height} - y - 1) + x]$$

- BMP 파일의 구조

- Headers of BMP → File Header, Bitmap Info. Header, LUT
- Image Data

- BMP 파일의 특징

- 한 행의 데이터는 4바이트의 배수
- 상하반전된 데이터

- $(x, y) \Rightarrow [\textit{widthStep} * (\textit{height} - y - 1) + x]$