INFO 430 MULTIMEDIA

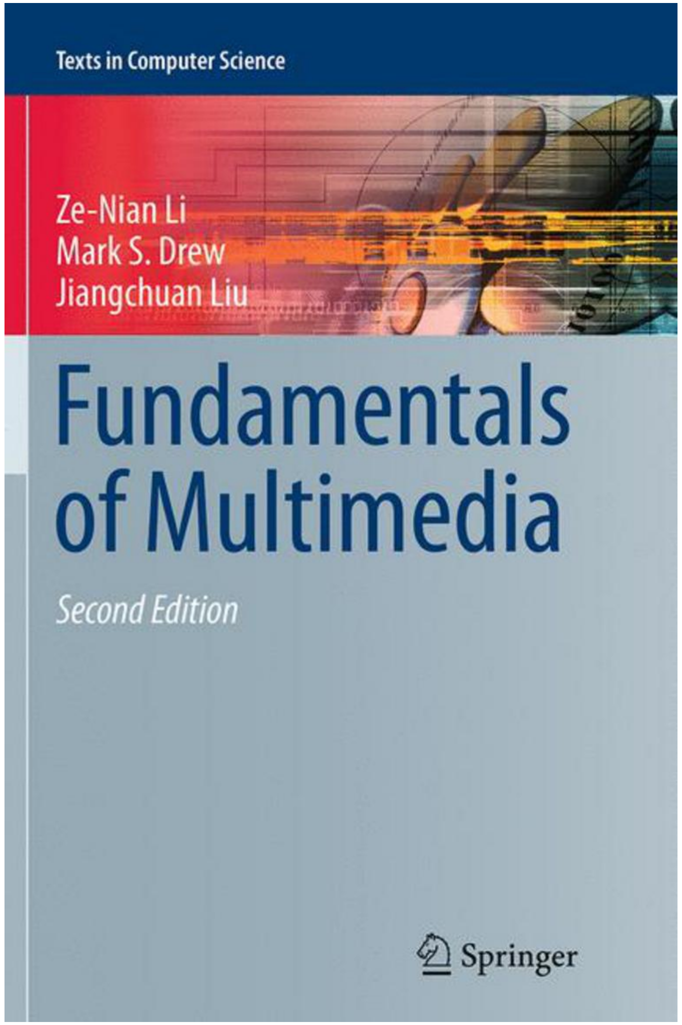
Dr Siba Haidar

Course Outline

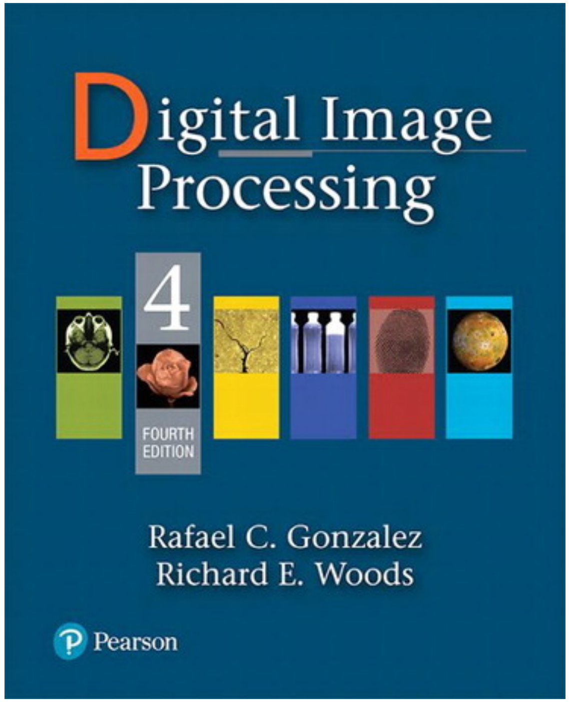
- Chapter 1 - Introduction to multimedia
- Chapter 2 - Digital representation of graphics and images
- Chapter 3 - Colors in images and video
- Chapter 4 - Fundamentals video
- ~~Chapter 5 - Digital Audio~~
- Chapter 6 - Lossless compression algorithms
- Chapter 7 - Lossy compression algorithms (including JPEG standard)
- Chapter 8 - Video Coding (MPEG)
- Chapter 9 - Image Processing

- Tutorials: Mastery of techniques for handling and processing of media streams with MATLAB.

Textbooks



- then for the last lectures :



GRAPHICS AND IMAGE DATA REPRESENTATIONS

Chapter 2

Chapter Outline

- Graphics/Image Data Types
  - 1-bit images
  - 8-bit gray-level images
    - Ordered Dithering
  - 24-bit color images
  - 8-bit color images
    - Look-Up Table
    - Median Cut
- Popular File Formats
  - GIF
  - others

Chapter Objectives

- the number of file formats used in multimedia continues to proliferate
- study few to develop a sense of how they operate
  - concentrate on GIF and JPG image file formats
    - GIF file format is one of the simplest and contains several fundamental features
    - JPG file format is arguably the most important overall

Image	Sound	Video
BMP,	AIFF,	AVI, MOV,
GIF, JPG,	AAC, AC3,	DV, FLV,
EPS, PNG,	MP3, MPG,	MPG,
PICT, PSD,	M4A, MOV,	WMA, WMV,
TIF, TGA	WMA	SWF, M4V, MP4, MXF

  - Table 3.1: Adobe Premiere file formats
- first we discuss the features of file formats in general

10

Image Data Types

1-bit Images

- each pixel stored as a single bit 0 |1
  - binary image
  - 1-bit monochrome image since it contains no color

- Fig. 3.1 shows a 1-bit monochrome image
  - Lena
  - multimedia scientists
  - standard image used to illustrate many algorithms

Fig. 3.1: Monochrome 1-bit Lena image.

8-bit Gray-level Images

- Each pixel has a gray-value between 0 and 255.
  - → a single byte;
- e.g.
  - dark pixel → 10
  - bright pixel → 230
- Bitmap:
  - 2D array of pixel values that represents the graphics/image data
- Fig. 3.3 shows the Lena image again, but this time in grayscale.

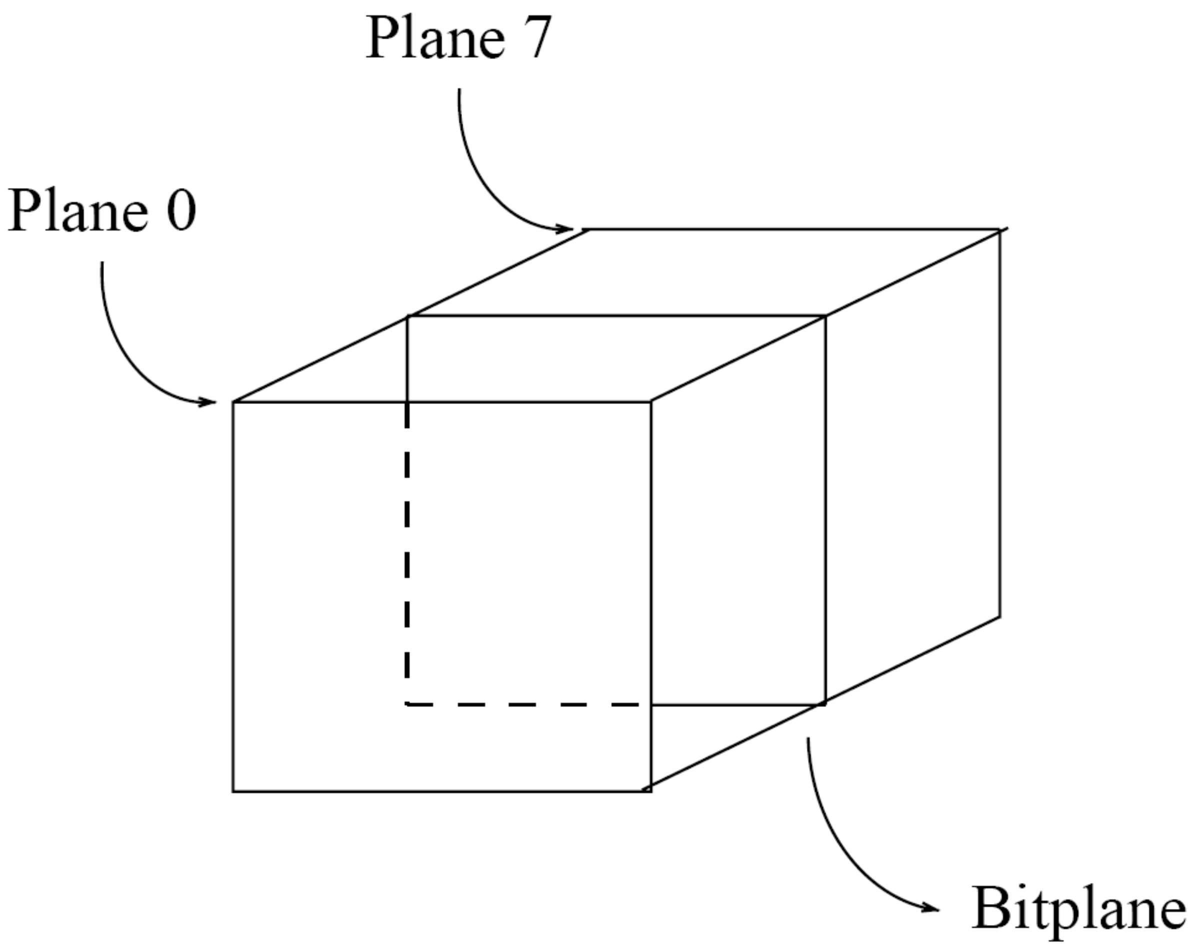


Fig. 3.2: Bit-planes for 8-bit grayscale image.

Fig. 3.3: Grayscale image of Lena.

Image resolution

- refers to the number of pixels in a digital image
- higher resolution → better quality

- fairly high resolution for such an image might be
  - 1,600 x1,200,
- lower resolution might be
  - 640 x 480

Example

Lena 128x128

Lena 1024x1024

Example

Lena 128x128 → ZOOM IN

Lena 1024x1024 → ZOOM IN

Li & Drew

Image Size

- Each pixel is usually stored as a byte
  - a value between 0 to 255
  - 640 x 480 grayscale image requires 300 kB
    - 640 x 480 = 307, 200 bytes

Dithering

When an image is printed, the basic strategy of dithering is used, which trades intensity resolution for spatial resolution to provide ability to print multi-level images on 2-level (1-bit) printers.

(a)

(b)

(a) detail

(c)

Fig. 3.4: Dithering of grayscale images. (a): 8-bit grey image "lenagray.bmp". (b): Dithered version of the image. (c): Detail of dithered version.

Dithering

- Dithering is used to calculate patterns of dots such that values from 0 to 255 correspond to patterns that are more and more filled at darker pixel values, for printing on a 1-bit printer.

- The main strategy
  - replace a pixel value by a larger pattern, say 2 x 2 or 4 x 4
  - such that the number of printed dots approximates the varying-sized disks of ink used in analog in halftone printing

- Half-tone printing
  - analog process that uses smaller or larger filled circles of black ink to represent shading, for newspaper printing.
  - For example, if we use a 2×2 dither matrix

.

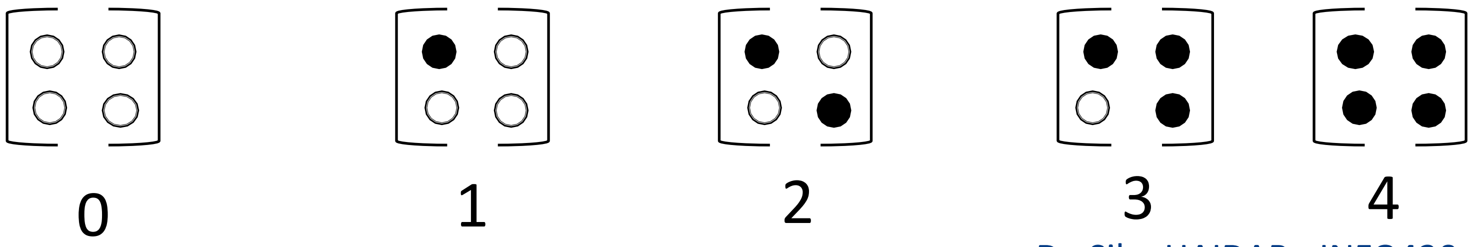
$$\begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}$$

Dithering

- for example, if we use 2x2 dithering matrix

$$\begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}$$

- we can first re-map image values in
  - 0..255 →0..4
  - by (integer) dividing by 255/4
    - if pixel value is 0 we print nothing
    - if pixel value is 4 we print all four dots



Dithering

- The rule is:
  - replace each pixel by an n x n matrix of dots
  - if the intensity > the dither matrix entry
    - then print an ON dot at that entry location
- Note that the image size may be much larger
  - for a dithered image
  - replacing each pixel by a 4 x 4 array of dots
  - makes an image 16 times as large

Ordered Dithering

- A clever trick can get around this problem.
- Suppose we wish to use a larger, 4 x 4 dither matrix, such as

$$\begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}$$

- An ordered dither consists of turning ON the printer out-put bit for **a pixel**
  - if the intensity level > the particular matrix element **just at that pixel position**.

Dithering

- An algorithm for ordered dither, with n x n dither matrix, is as follows:

```
BEGIN
     for x = 0 to xmax                      // columns
             for y = 0 to ymax              // rows
                     i = x mod n
                     j = y mod n
                     // I(x, y) is the input, O(x, y) is the output,
                     //D is the dither matrix.
                     if I(x, y) > D(i, j)
                             O(x, y) = 1;
                     else
                             O(x, y) = 0;
END
```

Small Example

- `>>lena=imread('lena512.bmp');`
- `>> lenaeye=lena(250:281,250:281);`
- `>> lenaeye17=uint8(double(lenaeye)*16/255);`

172	188	189	186	198	195	195	192	185	146	123	125	132	137	143	150	156	150	150	151	151	148	145	140	132	125	116	104	92	75	67	59
178	187	189	192	197	195	189	177	136	118	121	131	139	145	145	152	153	153	152	146	157	162	155	150	149	143	129	122	109	96	93	82
188	190	196	197	199	193	171	124	116	120	122	134	137	142	151	153	154	149	142	123	150	157	159	163	160	154	139	133	126	126	115	97
191	193	197	199	192	158	111	110	123	130	130	131	133	137	146	147	142	146	138	117	122	144	165	163	160	155	143	127	129	138	126	91
196	199	199	189	149	108	111	113	120	126	125	123	124	138	135	134	132	138	131	115	79	123	155	157	155	157	137	113	132	137	118	91
200	200	182	130	100	98	108	113	113	114	120	117	118	125	120	117	124	129	110	88	51	90	129	149	146	139	116	96	119	125	91	89
204	178	117	85	100	96	104	108	107	101	94	95	98	102	102	105	109	107	87	62	47	69	111	145	132	124	83	68	101	112	73	103
173	100	84	85	95	98	96	100	103	100	96	74	75	73	67	82	73	81	59	54	47	56	81	105	94	75	48	46	67	69	64	84
94	79	81	79	94	87	77	69	70	87	84	64	64	67	62	57	59	57	54	50	44	50	53	60	55	56	47	43	47	48	48	56
72	78	86	77	80	72	71	60	52	59	64	56	54	57	53	50	56	49	50	53	51	53	53	50	49	45	43	45	43	41	40	44
71	70	71	68	67	63	58	53	51	54	52	51	52	52	49	50	47	49	48	46	50	59	48	56	57	54	65	44	46	40	38	44
63	75	59	54	55	57	53	50	53	52	52	58	51	47	50	52	46	48	46	47	44	51	55	63	84	71	55	51	46	40	46	39
67	67	57	53	50	47	48	53	53	51	53	55	51	53	45	44	42	47	46	47	48	79	66	98	119	111	92	70	57	46	42	39
59	66	51	48	48	52	47	48	48	47	55	47	51	48	46	44	46	47	44	50	56	70	90	138	162	150	142	104	78	56	42	41
70	58	41	47	49	48	49	48	53	53	47	43	54	49	50	40	46	48	76	62	69	94	128	162	193	192	175	149	119	98	59	42
60	50	42	41	42	51	53	61	77	67	43	47	62	60	45	39	36	40	88	87	65	86	121	174	202	204	191	176	163	154	97	69
49	53	44	42	44	51	65	82	90	81	50	60	80	79	48	38	36	39	56	90	65	50	72	163	204	204	199	192	192	185	139	107
53	51	42	46	52	56	70	92	103	90	50	57	94	93	76	48	41	43	69	112	77	56	66	157	204	204	198	198	198	193	172	144
51	48	49	45	47	66	71	95	115	107	71	52	92	98	90	72	66	66	90	108	74	53	87	177	207	204	204	206	203	200	191	171
66	59	46	43	45	69	74	93	117	121	99	67	58	84	98	86	86	91	83	72	57	66	126	197	209	206	208	210	205	206	200	182
86	78	55	47	51	69	83	90	108	127	115	88	59	60	79	87	90	80	77	55	65	113	173	207	211	210	211	211	205	206	202	182
100	96	74	50	43	51	68	92	105	114	125	113	89	62	57	54	61	58	65	77	107	160	198	208	206	208	207	209	202	200	192	176
117	108	86	65	47	47	73	99	113	112	117	120	113	95	82	63	66	75	88	127	158	188	202	203	202	202	206	204	198	195	188	163
124	118	99	83	67	67	82	99	115	119	119	124	132	120	108	101	102	116	137	163	186	197	198	202	205	207	209	207	202	198	176	151
134	133	114	106	95	91	93	95	94	110	119	130	139	141	131	133	138	147	158	182	187	193	197	199	203	206	209	206	199	182	158	144
141	140	130	124	116	107	99	105	83	75	113	125	135	138	146	149	157	157	159	168	173	176	187	187	193	196	193	185	175	167	170	155
143	141	135	136	129	112	109	114	105	108	120	114	111	116	133	133	143	151	155	151	158	169	167	172	178	178	170	177	180	187	189	177
141	143	143	141	134	120	118	125	126	127	126	108	109	107	124	111	96	122	133	136	146	150	145	147	167	160	141	159	162	164	175	172
144	143	146	146	137	134	132	136	138	131	122	109	104	119	125	100	89	118	129	130	132	127	105	106	128	115	101	127	159	163	168	167
142	140	145	149	143	140	144	141	142	133	129	109	119	128	122	110	92	128	128	137	134	139	113	110	128	123	102	125	166	152	147	156
142	146	141	145	144	144	150	149	148	142	131	119	129	129	123	113	110	134	127	136	138	143	123	108	134	138	129	134	166	146	146	151
146	148	139	143	140	143	147	150	151	144	132	131	140	137	130	124	126	132	127	136	136	133	133	119	140	149	142	144	160	153	153	150

11	12	12	12	12	12	12	12	12	9	8	8	8	9	9	9	10	9	9	9	9	9	9	9	8	8	7	7	6	5	4	4
11	12	12	12	12	12	12	11	9	7	8	8	9	9	9	10	10	10	10	9	10	10	10	9	9	9	8	8	7	6	6	5
12	12	12	12	12	12	11	8	7	8	8	8	9	9	9	10	10	9	9	8	9	10	10	10	10	10	9	8	8	8	7	6
12	12	12	12	12	10	7	7	8	8	8	8	8	9	9	9	9	9	9	7	8	9	10	10	10	10	9	8	8	9	8	6
12	12	12	12	9	7	7	7	8	8	8	8	8	9	8	8	8	9	8	7	5	8	10	10	10	10	9	7	8	9	7	6
13	13	11	8	6	6	7	7	7	7	8	7	7	8	8	7	8	8	7	6	3	6	8	9	9	9	7	6	7	8	6	6
13	11	7	5	6	6	7	7	7	6	6	6	6	6	6	7	7	7	5	4	3	4	7	9	8	8	5	4	6	7	5	6
11	6	5	5	6	6	6	6	6	6	6	5	5	5	4	5	5	5	4	3	3	4	5	7	6	5	3	3	4	4	4	5
6	5	5	5	6	5	5	4	4	5	5	4	4	4	4	4	4	4	3	3	3	3	3	4	3	4	3	3	3	3	3	4
5	5	5	5	5	5	4	4	3	4	4	4	3	4	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	3	4	4	3	4	3	3	3	2	3
4	5	4	3	3	4	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	4	5	4	3	3	3	3	3	2
4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	4	6	7	7	6	4	4	3	3	2
4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	4	6	9	10	9	9	7	5	4	3	3
4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	4	4	6	8	10	12	12	11	9	7	6	4	3
4	3	3	3	3	3	3	4	5	4	3	3	4	4	3	2	2	3	6	5	4	5	8	11	13	13	12	11	10	10	6	4
3	3	3	3	3	3	4	5	6	5	3	4	5	5	3	2	2	2	4	6	4	3	5	10	13	13	12	12	12	12	9	7
3	3	3	3	3	4	4	6	6	6	3	4	6	6	5	3	3	3	4	7	5	4	4	10	13	13	12	12	12	12	11	9
3	3	3	3	3	4	4	6	7	7	4	3	6	6	6	5	4	4	6	7	5	3	5	11	13	13	13	13	13	13	12	11
4	4	3	3	3	4	5	6	7	8	6	4	4	5	6	5	5	6	5	5	4	4	8	12	13	13	13	13	13	13	13	11
5	5	3	3	3	4	5	6	7	8	7	6	4	4	5	5	6	5	5	3	4	7	11	13	13	13	13	13	13	13	13	11
6	6	5	3	3	3	4	6	7	7	8	7	6	4	4	3	4	4	4	5	7	10	12	13	13	13	13	13	13	13	12	11
7	7	5	4	3	3	5	6	7	7	7	8	7	6	5	4	4	5	6	8	10	12	13	13	13	13	13	13	12	12	12	10
8	7	6	5	4	4	5	6	7	7	7	8	8	8	7	6	6	7	9	10	12	12	12	13	13	13	13	13	13	12	11	9
8	8	7	7	6	6	6	6	6	7	7	8	9	9	8	8	9	9	10	11	12	12	12	12	13	13	13	13	12	11	10	9
9	9	8	8	7	7	6	7	5	5	7	8	8	9	9	9	10	10	10	11	11	11	12	12	12	12	12	12	11	10	11	10
9	9	8	9	8	7	7	7	7	7	8	7	7	7	8	8	9	9	10	9	10	11	10	11	11	11	11	11	11	12	12	11
9	9	9	9	8	8	7	8	8	8	8	7	7	7	8	7	6	8	8	9	9	9	9	9	10	10	9	10	10	10	11	11
9	9	9	9	9	8	8	9	9	8	8	7	7	7	8	6	6	7	8	8	8	8	7	7	8	7	6	8	10	10	11	10
9	9	9	9	9	9	9	9	9	8	8	7	7	8	8	7	6	8	8	9	8	9	7	7	8	8	6	8	10	10	9	10
9	9	9	9	9	9	9	9	9	9	8	7	8	8	8	7	7	8	8	9	9	9	8	7	8	9	8	8	10	9	9	9
9	9	9	9	9	9	9	9	9	9	8	8	9	9	8	8	8	8	8	9	9	8	8	7	9	9	9	9	10	10	10	9

0 11	**8** 12	**2** 12	**10** 12	12	12	12	12	12	9	8	8	8	9	9	9	10	9	9	9	9	9	9	9	8	8	7	7	6	5	4	4
12 11	**4** 12	**14** 12	**6** 12	12	12	12	11	9	7	8	8	9	9	9	10	10	10	10	9	10	10	10	9	9	9	8	8	7	6	6	5
3 12	**11** 12	**1** 12	**9** 12	12	12	11	8	7	8	8	8	9	9	9	10	10	9	9	8	9	10	10	10	10	10	9	8	8	8	7	6
15 12	**7** 12	**13** 12	**5** 12	12	10	7	7	8	8	8	8	8	9	9	9	9	9	9	7	8	9	10	10	10	10	9	8	8	9	8	6
12	12	12	12	9	7	7	7	8	8	8	8	8	9	8	8	8	9	8	7	5	8	10	10	10	10	9	7	8	9	7	6
13	13	11	8	6	6	7	7	7	7	8	7	7	8	8	7	8	8	7	6	3	6	8	9	9	9	7	6	7	8	6	6
13	11	7	5	6	6	7	7	7	6	6	6	6	6	6	7	7	7	5	4	3	4	7	9	8	8	5	4	6	7	5	6
11	6	5	5	6	6	6	6	6	6	6	5	5	5	4	5	5	5	4	3	3	4	5	7	6	5	3	3	4	4	4	5
6	5	5	5	6	5	5	4	4	5	5	4	4	4	4	4	4	4	3	3	3	3	3	4	3	4	3	3	3	3	3	4
5	5	5	5	5	5	4	4	3	4	4	4	3	4	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	3	4	4	3	4	3	3	3	2	3
4	5	4	3	3	4	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	4	5	4	3	3	3	3	3	2
4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	4	6	7	7	6	4	4	3	3	2
4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	4	6	9	10	9	9	7	5	4	3	3
4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	4	4	6	8	10	12	12	11	9	7	6	4	3
4	3	3	3	3	3	3	4	5	4	3	3	4	4	3	2	2	3	6	5	4	5	8	11	13	13	12	11	10	10	6	4
3	3	3	3	3	3	4	5	6	5	3	4	5	5	3	2	2	2	4	6	4	3	5	10	13	13	12	12	12	12	9	7
3	3	3	3	3	4	4	6	6	6	3	4	6	6	5	3	3	3	4	7	5	4	4	10	13	13	12	12	12	12	11	9
3	3	3	3	3	4	4	6	7	7	4	3	6	6	6	5	4	4	6	7	5	3	5	11	13	13	13	13	13	13	12	11
4	4	3	3	3	4	5	6	7	8	6	4	4	5	6	5	5	6	5	5	4	4	8	12	13	13	13	13	13	13	13	11
5	5	3	3	3	4	5	6	7	8	7	6	4	4	5	5	6	5	5	3	4	7	11	13	13	13	13	13	13	13	13	11
6	6	5	3	3	3	4	6	7	7	8	7	6	4	4	3	4	4	4	5	7	10	12	13	13	13	13	13	13	13	12	11
7	7	5	4	3	3	5	6	7	7	7	8	7	6	5	4	4	5	6	8	10	12	13	13	13	13	13	13	12	12	12	10
8	7	6	5	4	4	5	6	7	7	7	8	8	8	7	6	6	7	9	10	12	12	12	13	13	13	13	13	13	12	11	9
8	8	7	7	6	6	6	6	6	7	7	8	9	9	8	8	9	9	10	11	12	12	12	12	13	13	13	13	12	11	10	9
9	9	8	8	7	7	6	7	5	5	7	8	8	9	9	9	10	10	10	11	11	11	12	12	12	12	12	12	11	10	11	10
9	9	8	9	8	7	7	7	7	7	8	7	7	7	8	8	9	9	10	9	10	11	10	11	11	11	11	11	11	12	12	11
9	9	9	9	8	8	7	8	8	8	8	7	7	7	8	7	6	8	8	9	9	9	9	9	10	10	9	10	10	10	11	11
9	9	9	9	9	8	8	9	9	8	8	7	7	7	8	6	6	7	8	8	8	8	7	7	8	7	6	8	10	10	11	10
9	9	9	9	9	9	9	9	9	8	8	7	7	8	8	7	6	8	8	9	8	9	7	7	8	8	6	8	10	10	9	10
9	9	9	9	9	9	9	9	9	9	8	7	8	8	8	7	7	8	8	9	9	9	8	7	8	9	8	8	10	9	9	9
9	9	9	9	9	9	9	9	9	9	8	8	9	9	8	8	8	8	8	9	9	8	8	7	9	9	9	9	10	10	10	9

11	12	12	12	12 (0)	12 (8)	12 (2)	12 (10)	12	9	8	8	8	9	9	9	10	9	9	9	9	9	9	9	8	8	7	7	6	5	4	4
11	12	12	12	12 (12)	12 (4)	12 (14)	11 (6)	9	7	8	8	9	9	9	10	10	10	10	9	10	10	10	9	9	9	8	8	7	6	6	5
12	12	12	12	12 (3)	12 (11)	11 (1)	8 (9)	7	8	8	8	9	9	9	10	10	9	9	8	9	10	10	10	10	10	9	8	8	8	7	6
12	12	12	12	12 (15)	10 (7)	7 (13)	7 (5)	8	8	8	8	8	9	9	9	9	9	9	7	8	9	10	10	10	10	9	8	8	9	8	6
12	12	12	12	9	7	7	7	8	8	8	8	8	9	8	8	8	9	8	7	5	8	10	10	10	10	9	7	8	9	7	6
13	13	11	8	6	6	7	7	7	7	8	7	7	8	8	7	8	8	7	6	3	6	8	9	9	9	7	6	7	8	6	6
13	11	7	5	6	6	7	7	7	6	6	6	6	6	6	7	7	7	5	4	3	4	7	9	8	8	5	4	6	7	5	6
11	6	5	5	6	6	6	6	6	6	6	5	5	5	4	5	5	5	4	3	3	4	5	7	6	5	3	3	4	4	4	5
6	5	5	5	6	5	5	4	4	5	5	4	4	4	4	4	4	4	3	3	3	3	3	4	3	4	3	3	3	3	3	4
5	5	5	5	5	5	4	4	3	4	4	4	3	4	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	3	4	4	3	4	3	3	3	2	3
4	5	4	3	3	4	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	4	5	4	3	3	3	3	3	2
4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	4	6	7	7	6	4	4	3	3	2
4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	4	6	9	10	9	9	7	5	4	3	3
4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5	4	4	6	8	10	12	12	11	9	7	6	4	3
4	3	3	3	3	3	3	4	5	4	3	3	4	4	3	2	2	3	6	5	4	5	8	11	13	13	12	11	10	10	6	4
3	3	3	3	3	3	4	5	6	5	3	4	5	5	3	2	2	2	4	6	4	3	5	10	13	13	12	12	12	12	9	7
3	3	3	3	3	4	4	6	6	6	3	4	6	6	5	3	3	3	4	7	5	4	4	10	13	13	12	12	12	12	11	9
3	3	3	3	3	4	4	6	7	7	4	3	6	6	6	5	4	4	6	7	5	3	5	11	13	13	13	13	13	13	12	11
4	4	3	3	3	4	5	6	7	8	6	4	4	5	6	5	5	6	5	5	4	4	8	12	13	13	13	13	13	13	13	11
5	5	3	3	3	4	5	6	7	8	7	6	4	4	5	5	6	5	5	3	4	7	11	13	13	13	13	13	13	13	13	11
6	6	5	3	3	3	4	6	7	7	8	7	6	4	4	3	4	4	4	5	7	10	12	13	13	13	13	13	13	13	12	11
7	7	5	4	3	3	5	6	7	7	7	8	7	6	5	4	4	5	6	8	10	12	13	13	13	13	13	13	12	12	12	10
8	7	6	5	4	4	5	6	7	7	7	8	8	8	7	6	6	7	9	10	12	12	12	13	13	13	13	13	13	12	11	9
8	8	7	7	6	6	6	6	6	7	7	8	9	9	8	8	9	9	10	11	12	12	12	12	13	13	13	13	12	11	10	9
9	9	8	8	7	7	6	7	5	5	7	8	8	9	9	9	10	10	10	11	11	11	12	12	12	12	12	12	11	10	11	10
9	9	8	9	8	7	7	7	7	7	8	7	7	7	8	8	9	9	10	9	10	11	10	11	11	11	11	11	11	12	12	11
9	9	9	9	8	8	7	8	8	8	8	7	7	7	8	7	6	8	8	9	9	9	9	9	10	10	9	10	10	10	11	11
9	9	9	9	9	8	8	9	9	8	8	7	7	7	8	6	6	7	8	8	8	8	7	7	8	7	6	8	10	10	11	10
9	9	9	9	9	9	9	9	9	8	8	7	7	8	8	7	6	8	8	9	8	9	7	7	8	8	6	8	10	10	9	10
9	9	9	9	9	9	9	9	9	9	8	7	8	8	8	7	7	8	8	9	9	9	8	7	8	9	8	8	10	9	9	9
9	9	9	9	9	9	9	9	9	9	8	8	9	9	8	8	8	8	8	9	9	8	8	7	9	9	9	9	10	10	10	9

Color Image Data Types

Color Image Data Types

- The most common data types for graphics and image file formats — 24-bit color and 8-bit color.
- Some formats are restricted to particular hardware / operating system platforms, while others are "cross-platform" formats.
  - Even if some formats are not cross-platform, there are conversion applications that will recognize and translate formats from one system to another.
- Most image formats incorporate some variation of a compression technique due to the large storage size of image files. Compression techniques can be classified into either lossless or lossy.

24-bit Color Images

- each pixel is represented by 3 bytes, usually RGB
  - 256 x 256 x 256 possible combined colors
  - = 16,777,216 possible colors
  - → storage penalty
    - 640 x 480 24-bit color image → 921.6 kB without compression

- An important point:
  - many 24-bit color images are actually stored as 32-bit images
  - extra byte to store an alpha value
    - special effect information (e.g., transparency)


(a)


(b)


(c)


(d)

Fig. 3.5: (a): 24-bit high-resolution color image "forestfire.bmp".
(b, c, d): separate R, G, and B color channels for this image.

8-bit Color Images

- Many systems can make use of 8 bits of color information (the so-called “256 colors”) in producing a screen image.

- Such image files use the concept of a lookup table to store color information.
  - image stores **NOT** color
  - **BUT** instead just an index
  - into a table with 3-byte values that specify the color

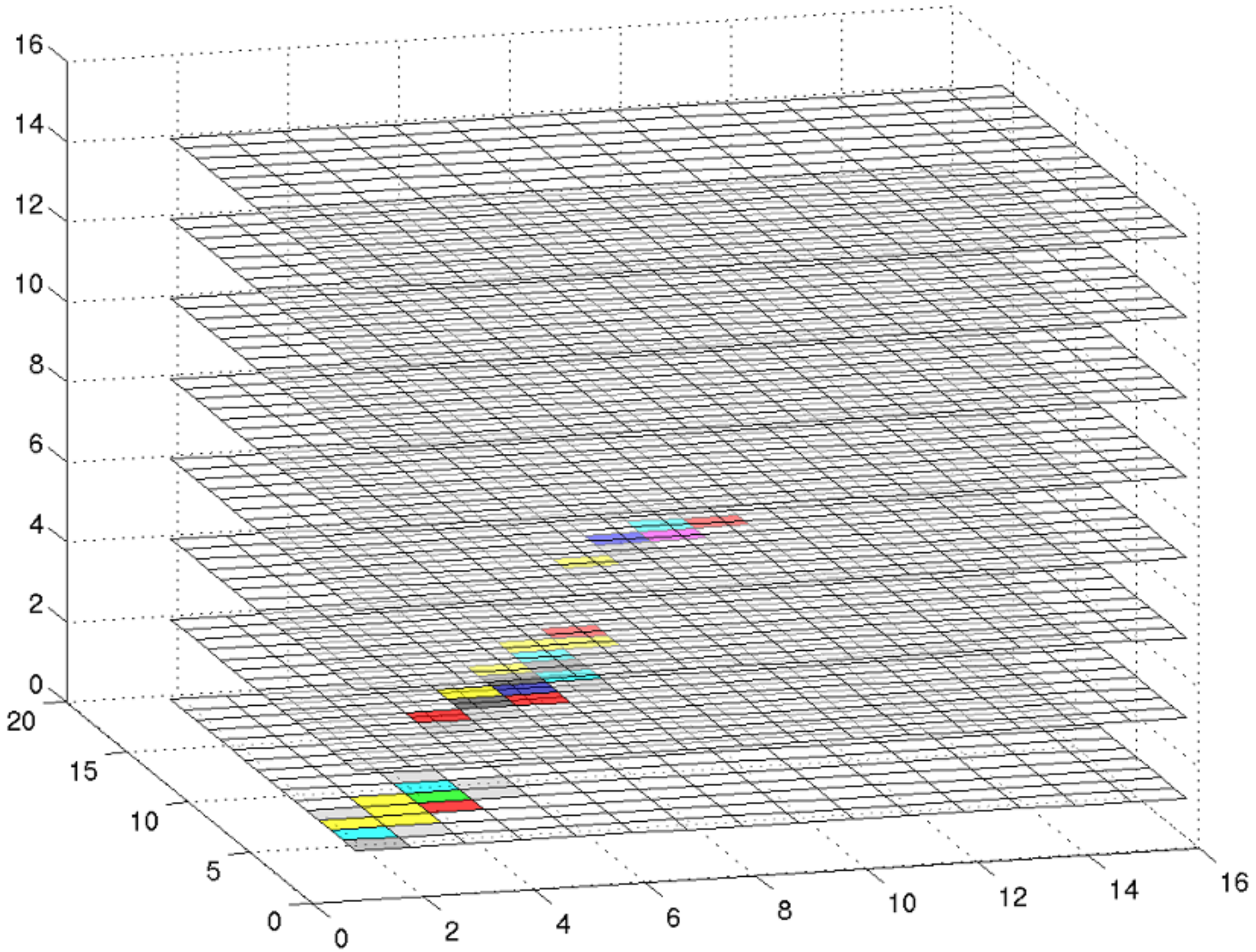


Fig. 3.6: 3D histogram of the RGB colors in forestfire.bmp.

Fig. 3.7 Example of 8-bit color image. (GIF)

Note the great savings in space for 8-bit images, over 24-bit ones: a 640 x 480 8-bit color image only requires 300 kB of storage, compared to 921.6 kB for a color image (again, without any compression applied).

Color Look-up Tables (LUTs)

- idea in 8-bit color images
  - store only the index for each pixel
  - if a pixel stores 25→ go to row 25 in LUT to find color

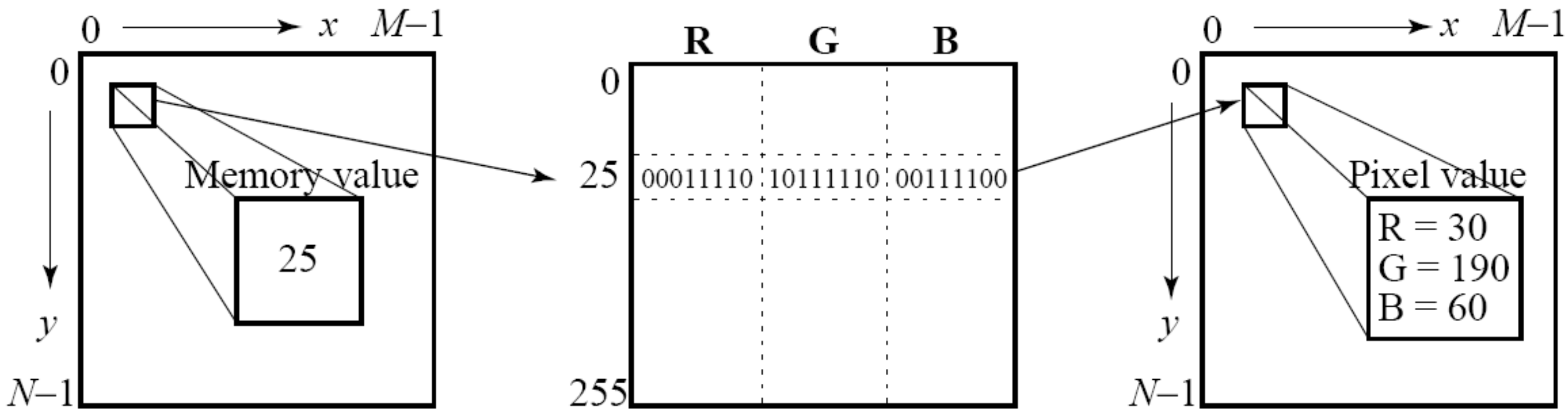


Fig. 3.8: Color LUT for 8-bit color images.

Color-picker

- array of fairly large blocks of color
  - mouse-click selects color
  - https://www.google.com/search?q=color+picker
- color-picker displays palette colors associated with index values from 0 to 255
  - Fig. 3.9 displays the concept of a color-picker
    - user selects color block with index value 2
    - color = cyan : RGB values (0, 255, 255)
- simple animation process
  - keep the indexes
  - but change the LUT
  - →color cycling | palette animation

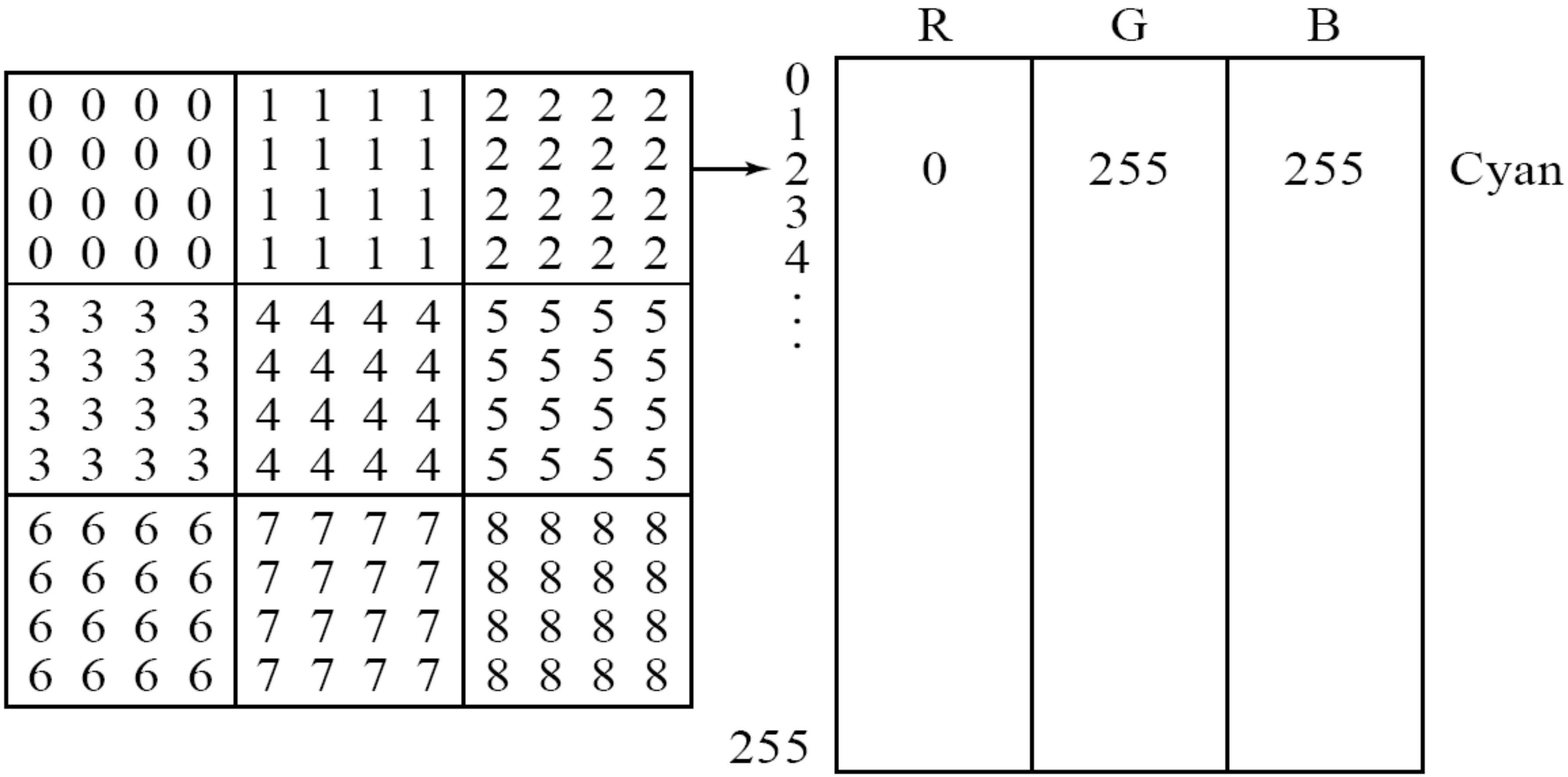
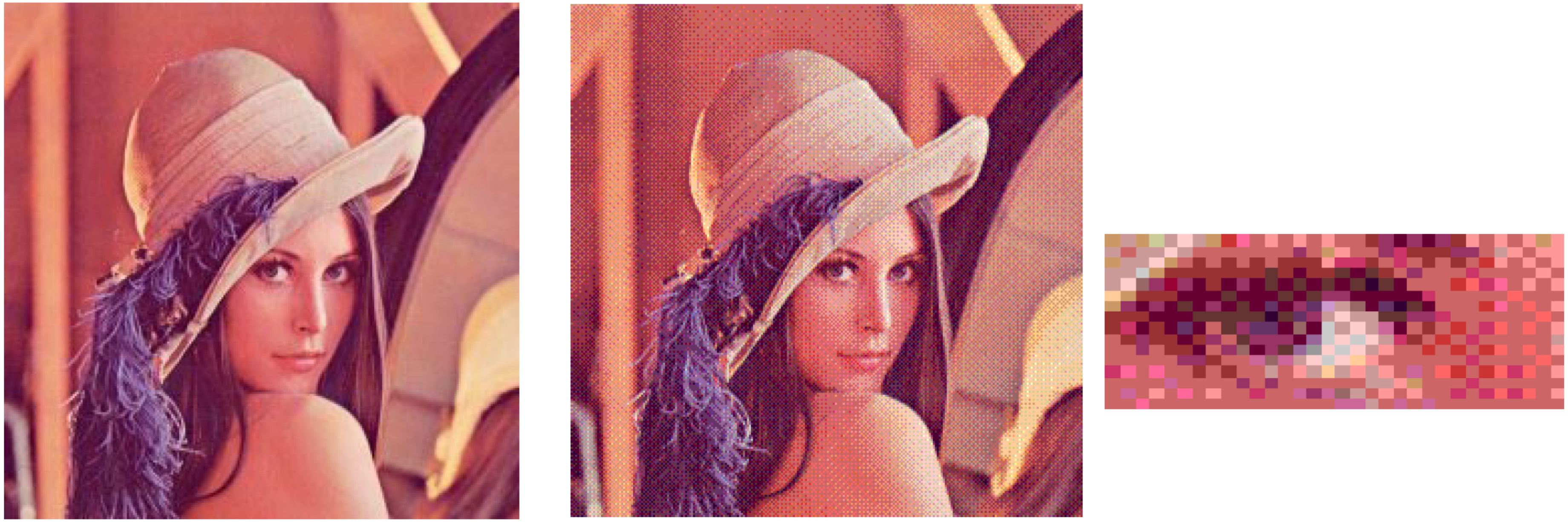


Fig. 3.9: Color-picker for 8-bit color: each block of the color-picker corresponds to one row of the color LUT

Dithering for Color Image

- Dithering can also be carried out for color printers
  - 1 bit per color channel
  - space out color with R, G, and B dots
- if printer|screen can print only a limited number of colors
  - using 8 bits instead of 24
  - color → printable, even if not in LUT
  - apparent color resolution increased by averaging intensities of neighboring pixels
- trick the eye into perceiving colors that are not available
  - because it carries out a spatial blending that can be put to good use

- Example



(a) (b) (c)

Fig. 3.10 (a) shows a 24-bit color image of “Lena”, and Fig. 3.10 (b) shows the same image reduced to only 5 bits via dithering. A detail of the left eye is shown in Fig. 3.10 (c).

How to devise a color LUT

Color Reduction Problem

How to make 8-bit LUT for a 24-bit color Image?

1) Straightforward LUT creation

- divide RGB cube into equal slices in each dimension

- (a) centers of each cube → entries in LUT
  - scale each range R, G, B ranges 0..255 → appropriate ranges
  - generate total 8-bit code

- (b) Since humans are more sensitive to R and G than to B
  - shrink R & G range 8 into 3-bit range 0..7 → keep most significant 3 bits!
  - shrink B range down to 2-bit range 0..3 → keep MS 2 bits!

- (c) Finally
  - each pixel gets replaced by its 8-bit index
  - color LUT serves to generate 24-bit color

2) Median-cut algorithm

- better yet simple solution
  - → concentrate bits where they most need to differentiate between high populations of close colors
- sort the R byte values and find their median
  - values <= median → labelled "0"
  - values > median → labelled "1"
- visualize find median using a histogram
  - Fig. 3.11

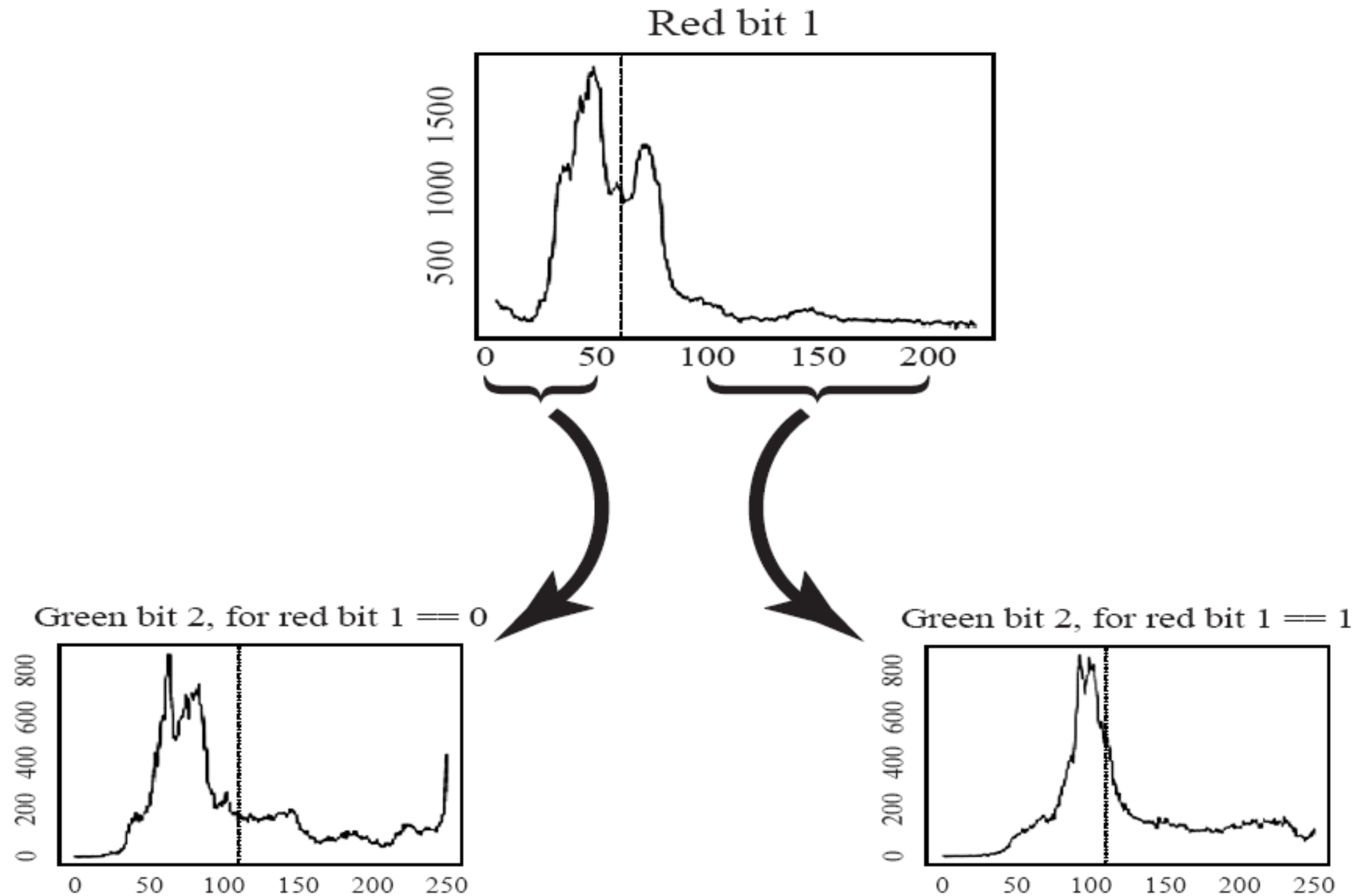


Fig. 3.11 Histogram of R bytes for the 24-bit color image "forestfire.bmp" results in a "0" bit or "1" bit label for every pixel. For the second bit of the color table index being built, we take R values less than the R median and label just those pixels as "0" or "1" according as their G value is less than or greater than the median of the G value, just for the "0" Red bit pixels. Continuing over R, G, B for 8 bits gives a color LUT 8-bit index.

Median Cut Exercise

- check the median cut exercise in a separate file
  - partiel.09.10

Popular File Formats

8-bit GIF : one of the most important formats because of its historical connection to the WWW and HTML markup language as the first image type recognized by net browsers

JPEG: currently the most important common file format

GIF

- standard simple yet contains many common elements
- 8-bit (256) color images only
  - acceptable color images
  - best suited for images with few distinctive colors (graphics |drawing)
- GIF standard supports interlacing
  - successive display of pixels in widely-spaced rows
  - by a 4-pass display process
- GIF actually comes in two flavors:
  - 1. GIF87a: The original specification
  - 2. GIF89a: The later version
    - supports simple animation via a Graphics Control Extension block in the data
    - provides simple control over delay time, a transparency index, etc.

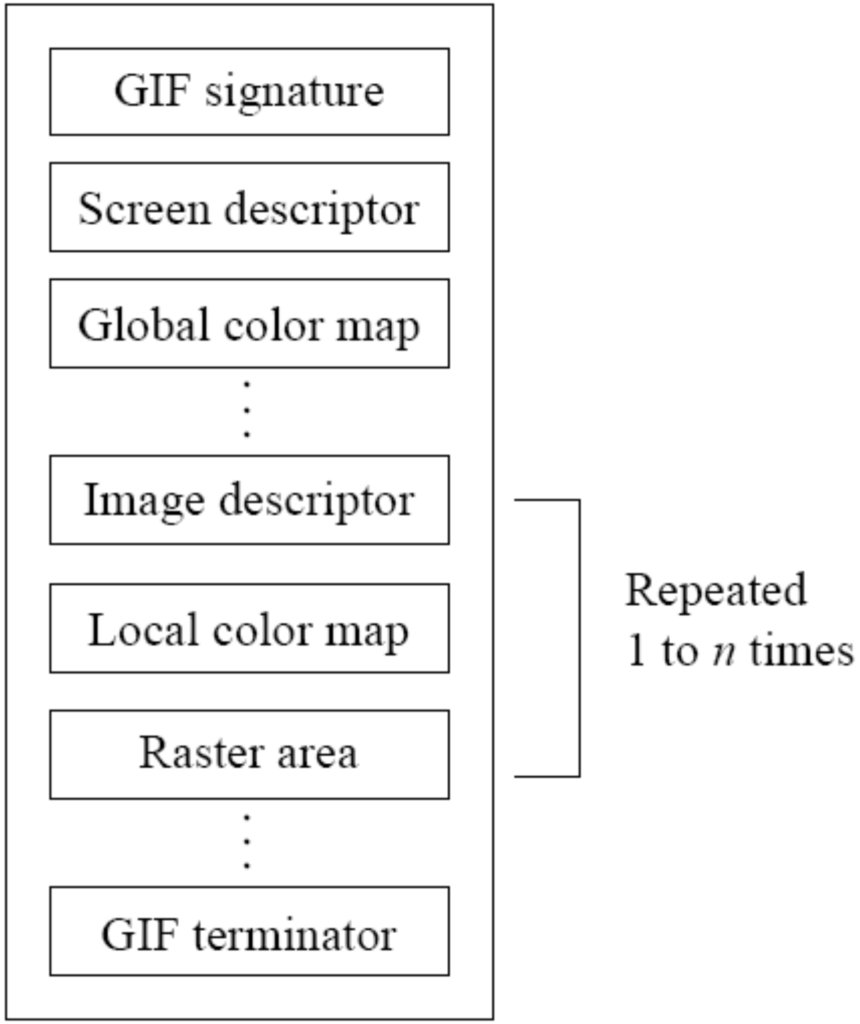


Fig. 3.12: GIF87 file format. Standard Specification

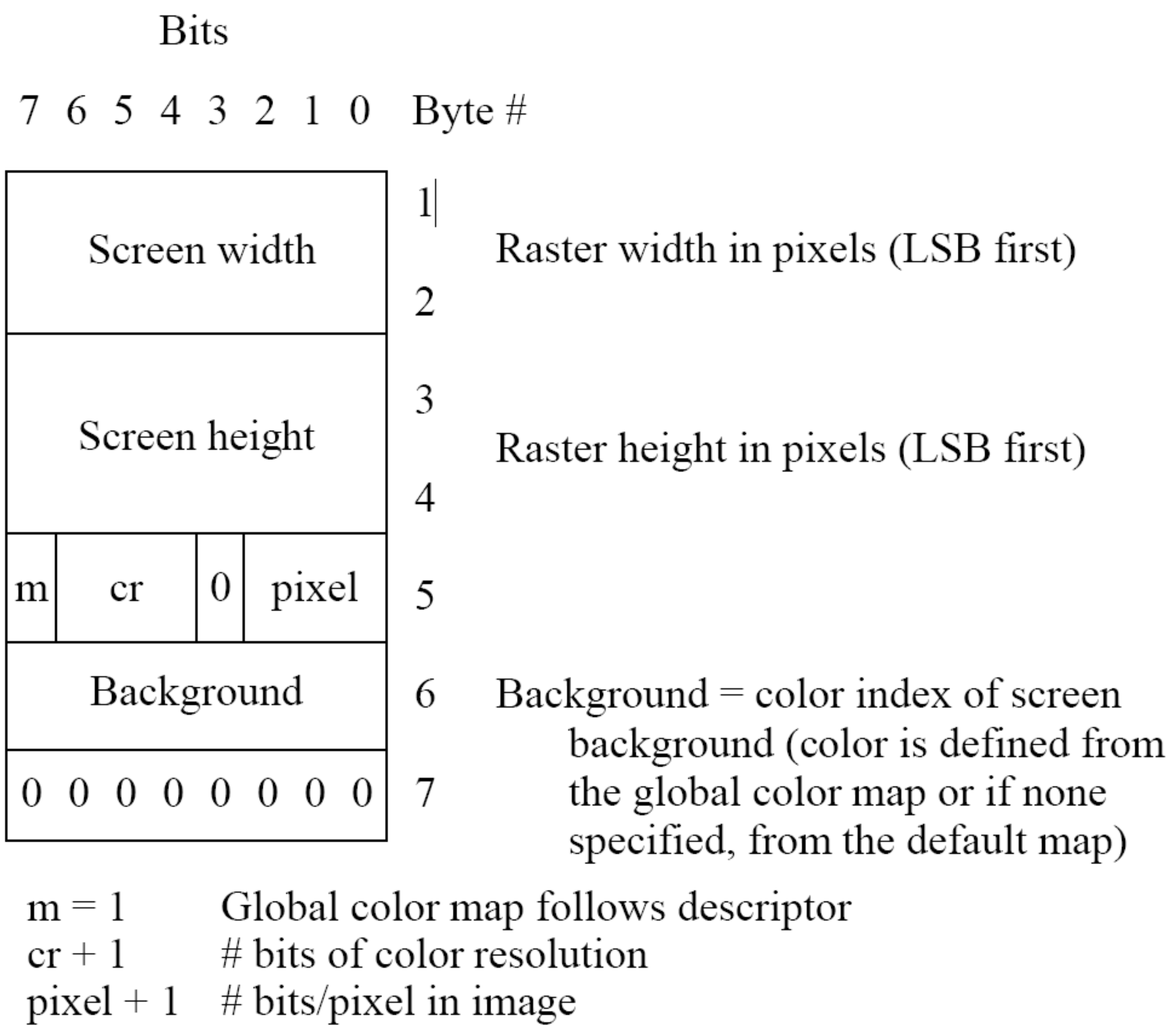


Fig. 3.13: GIF screen descriptor.

Screen Descriptor comprises a set of attributes that belong to every image in the file. According to the GIF87 standard, it is defined as in Fig. 3.13.

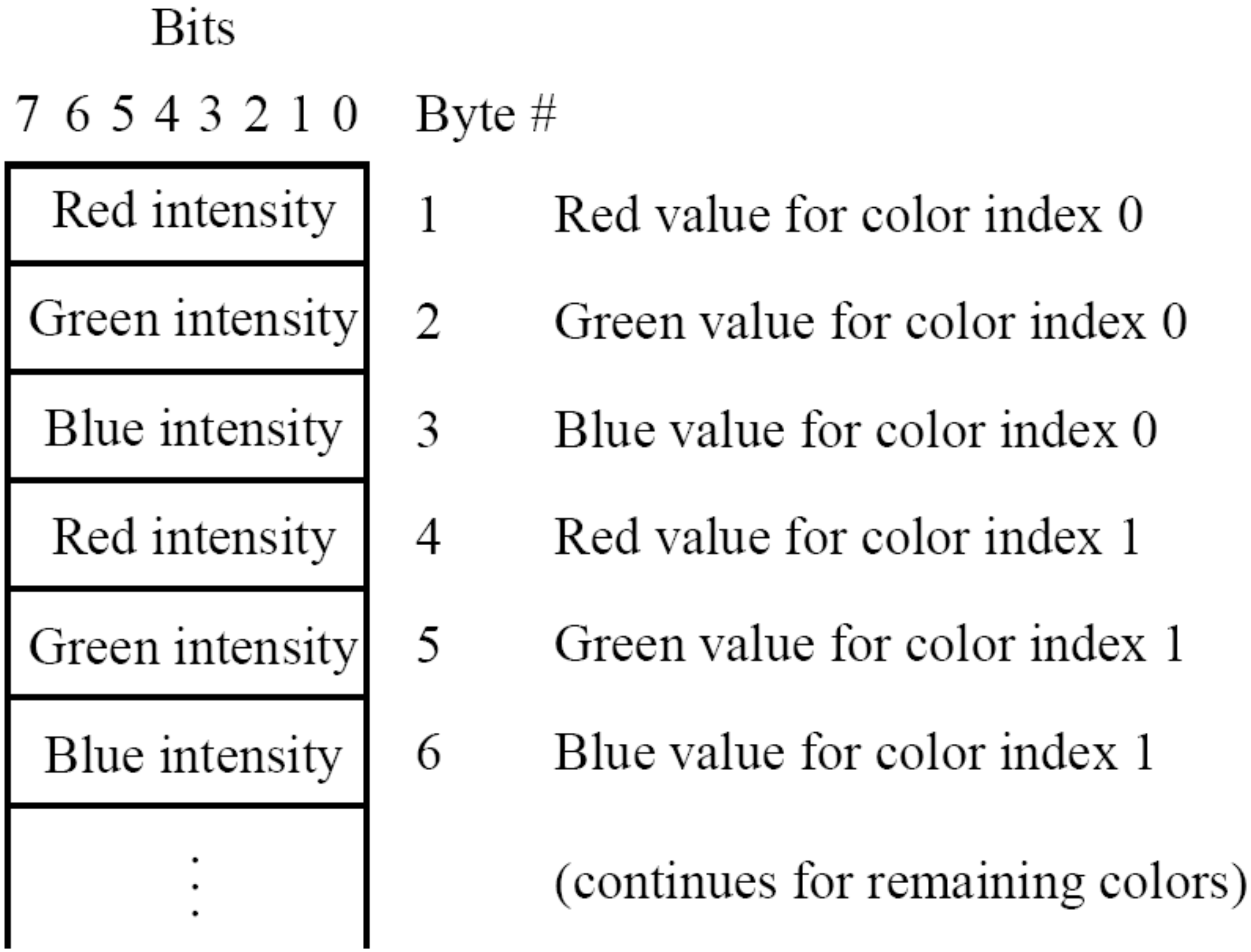


Fig. 3.14: GIF color map.

Color Map is set up in a very simple fashion as in Fig. 3.14. However, the actual length of the table equals $2^{(pixel+1)}$ as given in the Screen Descriptor.

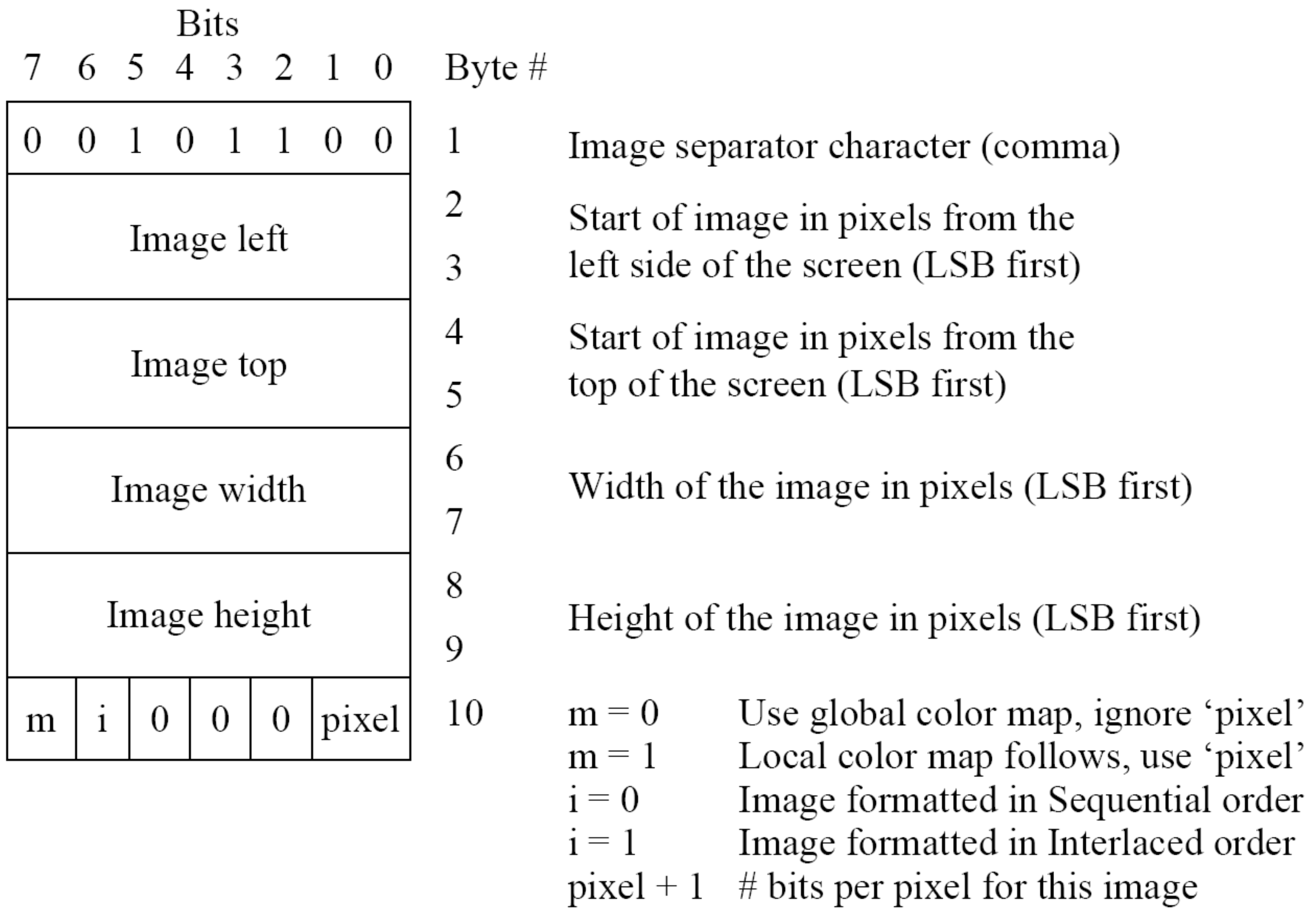


Fig. 3.15: GIF image descriptor.

Each image in the file has its own **Image Descriptor**, defined as in Fig. 3.15.

Image row	Pass 1	Pass 2	Pass 3	Pass 4	Result
0	*1a*				*1a*
1				*4a*	*4a*
2			*3a*		*3a*
3				*4b*	*4b*
4		*2a*			*2a*
5				*4c*	*4c*
6			*3b*		*3b*
7				*4d*	*4d*
8	*1b*				*1b*
9				*4e*	*4e*
10			*3c*		*3c*
11				*4f*	*4f*
12		*2b*			*2b*
⋮					

Fig. 3.16: GIF 4-pass interlace display row order.

If the “interlace” bit is set in the local Image Descriptor, then the rows of the image are displayed in a four-pass sequence (Fig.3.16).

GIF

- We can investigate how the file header works in practice by having a look at a particular GIF image.
- Fig. 3.7 is an 8-bit color GIF image
  - in UNIX, issue the command: `od -c forestfire.gif | head -2`
  - and we see the first 32 bytes interpreted as characters:

```
G   I   F   8   7   a   \208  \2  \188  \1  \247  \0  \0  \6  \3  \5
J   \132  \24  |  )  \7  \198  \195  \  \128  U  \27  \196  \166
&   T
```

  - To decipher the remainder of the file header (after "GIF87a"), we use hexadecimal: `od -x forestfire.gif | head -2`
  - with the result

```
4749 4638 3761 d002 bc01 f700 0006 0305 ae84 187c
2907 c6c3 5c80 551b c4a6 2654
```

GIF

- file header
  - G I F 8 7 a \208 \2 \188 \1 \247 \0 \0 \6 \3 \5 J \132 \24 |) \7 \198 \195 \ \128 U \27 \196 \166 & T
  - GIF87a
  - 4749 4638 3761 d002 bc01 f700 0006 0305 ae84 187c 2907 c6c3 5c80 551b c4a6 2654

JPEG

- most important current standard for image compression
- human vision system
  - has some specific limitations
  - JPEG takes advantage to achieve high rates of compression
- user set a desired level of quality | compression ratio
  - input divided / output
- Fig. 3.17 shows forestfire image with quality factor Q=10%
  - size ~ 1.5% of original size
  - JPEG Q=75% → size ~ 5.6% of original
  - GIF → compresses ~ 23.0% of uncompressed image size

Fig. 3.17: JPEG image with low quality specified by user.

PNG

- Portable Network Graphics
  - to supersede GIF standard and extend it in important ways
- special features of PNG files include:
  - 1. Support for up to 48 bits of color information — a large increase.
  - 2. Files may contain
    - gamma-correction information for correct display
    - alpha-channel information for uses: as control of transparency
  - 3. Progressive display of pixels in a 2-dimensional fashion
    - show few pixels at a time over seven passes through each 8 × 8 block of an image

TIFF

- Tagged Image File Format
- support for attachment of additional information (tags) provides a great deal of flexibility
  - 1. The most important tag is a format signifier:
    - what type of compression etc. is in use in the stored image
  - 2. TIFF can store many different types of image:
    - 1-bit, grayscale, 8-bit color, 24-bit RGB, etc.
  - 3. TIFF was originally a lossless format
    - but now a new JPEG tag allows one to opt for JPEG compression
  - 4. The TIFF format
    - developed by the Aldus Corporation in the 1980's, and
    - later supported by Microsoft

Graphics Animation Files

- few dominant formats aimed at storing graphics animations
  - series of drawings or graphic illustrations
  - as opposed to video (series of images)
  - difference:
    - animations less demanding of resources than videos
- 1. FLC is an animation or moving picture file format
  - created by Animation Pro
  - FLI, is similar to FLC
- 2. GL produces somewhat better quality moving pictures
  - GL animations can also usually handle larger file sizes
- 3. Many older formats:
  - DL or Amiga IFF files
  - Apple Quicktime files
  - animated GIF89 files

PS and PDF

- Postscript
  - important language for typesetting
  - many high-end printers have a Postscript interpreter built
  - vector-based picture language
    - rather than pixel-based
    - page element definitions are essentially in terms of vectors
- 1. Postscript includes text as well as vector/structured graphics.
- 2. GL bit-mapped images can be included in output files.
- 3. Encapsulated Postscript files add some additional information for inclusion of Postscript files in another document.
- 4. Postscript page description language itself does not provide compression; in fact, Postscript files are just stored as ASCII.

PS and PDF

- parallel Postscript
- Adobe Systems Inc. includes LZW compression in its Portable Document Format (PDF) file format
- PDF files
  - that do not include images
  - have same compression ratio, 2:1 or 3:1
  - as files compressed with other LZW-based compression tools

Some Other JPEG Formats

- Microsoft Windows: WMF: the native vector file format for the Microsoft Windows operating environment:
  - 1. Consist of a collection of GDI (Graphics Device Interface) function calls, also native to the Windows environment.
  - 2. When a WMF file is "played" (typically using the Windows PlayMetaFile() function) the described graphics is rendered.
  - 3. WMF files are ostensibly device-independent and are unlimited in size.

Some Other JPEG Formats

- BMP
  - major system standard graphics file format for Microsoft Windows
  - used in Microsoft Paint and other programs
  - many sub-variants within the BMP standard
- Macintosh: PAINT and PICT:
  - 1. PAINT was originally used in the MacPaint program, initially only for 1-bit monochrome images.
  - 2. PICT format is used in MacDraw (a vector-based drawing program) for storing structured graphics.
- X-windows: PPM: the graphics format for the X Window system. PPM supports 24-bit color bitmaps, and can be manipulated using many public domain graphic editors, e.g., xv.