

• 基本数据类型

- 1位图像
- 8位灰度图像
- 24位彩色图像
- 8位彩色图像
- 颜色查找表 (LUTs)

• 常用文件格式 -

JPEG、GIF、BMP等

图形和图像数据表示



讲师：肖俊

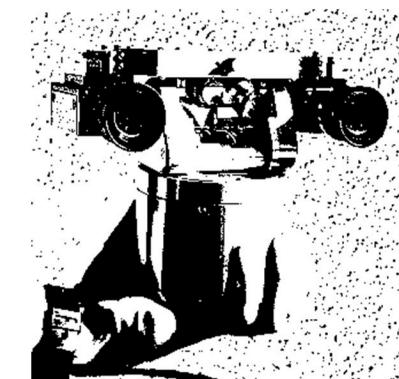
软件与技术学院

1. 基本图形/图像类型

- 1位图像
- 8位灰度图像
- 24位彩色图像
- 8位彩色图像
- 颜色查找表
- 如何设计颜色查找表

1.1 1位图像：案例

也称为二值图像或单色图像



1位图像示例

1.1 1位图像：特征

- 由开启和关闭的像素（数字图像中的像素——图像元素）组成
- 每个像素作为单个位（0或1）存储，0 - 黑色，1 - 白色

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

1.1 1位图像：大小与用途

- 存储
 - 分辨率为 640×480 的单色图像
 - $640 \times 480 / 8$ 字节
 - 所需存储空间：38.4KB

- 用途
 - 仅包含简单图形和文本的图片

1.2 8位灰度图像：案例



8位灰度图像示例

1.2 8位灰度图像：案例

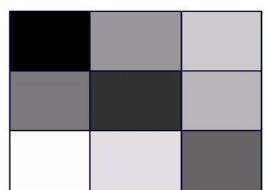


8位灰度图像与1位图像对比

1.2 8位灰度图像：特征

- 每个像素由一个字节表示——灰度值介于0到255之间

- 整个图像可视为一个二维像素值数组——称为位图



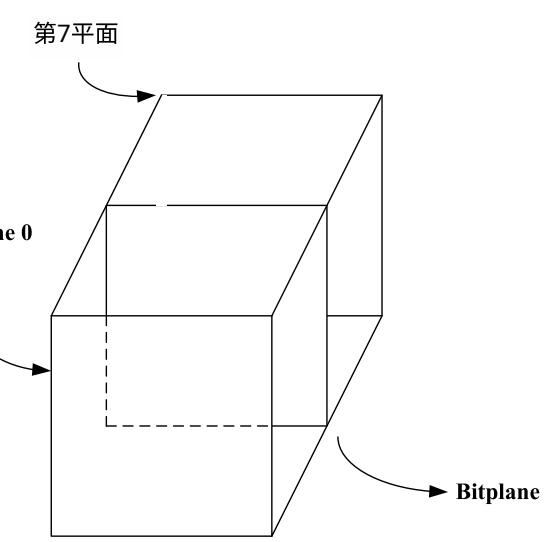
$$I = \begin{bmatrix} 0 & 150 & 200 \\ 120 & 50 & 180 \\ 250 & 220 & 100 \end{bmatrix}$$

1.2 8位灰度图像：特征

- 8位图像可看作一组1位位平面

- 每个平面由图像在某一级别的1位表示组成

- 所有位平面组成一个单字节，用于存储 $0 \sim 255$ 之间的值



1.2 8位灰度图像：尺寸

- 分辨率

- 高： 1600×1200
- 低： 640×480
- 宽高比：4:3

- 一幅 640×480 灰度图像 $-640 \times 480 = 307,200$ 所需的存储空间（字节）

- 存储图像数组的硬件
 - 帧缓冲器/“视频”卡

1.2 8位灰度图像：打印

- 如何在两级（1位）打印机上打印8位灰度图像？

- DPI

- 每英寸点数

- 打印此类图像很复杂

- 使用抖动处理

- 将强度分辨率转换为空间分辨率

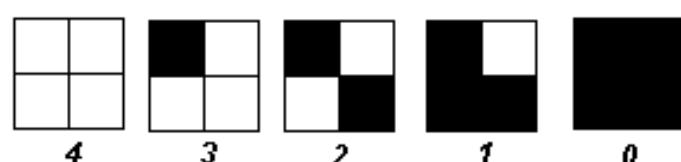
1.2 8位灰度图像：打印

- 抖动处理

- 主要策略是用更大的图案（例如 2×2 或 4×4 ）替换像素值，使得打印点的数量近似于模拟半色调印刷（例如报纸照片）中使用的不同大小的墨盘。

- 将颜色分辨率转换为空间分辨率。

- 一个 $N \times N$ 矩阵表示 $N^2 + 1$ 个强度级别
 - 2×2 图案可以表示五个级别：



1.2 8位灰度图像：打印

- 我们可以首先通过（整数）除以256/5，将0到255范围内的图像值重新映射到新的0到4范围。然后，例如，如果像素值为0，则在打印机输出的 2×2 区域不打印任何内容。但如果像素值为4，则打印全部四个点。

- 如果强度大于 $>$ 抖动矩阵条目，则在该条目位置打印一个黑点：将每个像素替换为一个 $n \times n$ 的点矩阵。

- 上述方法会增加输出图像的大小

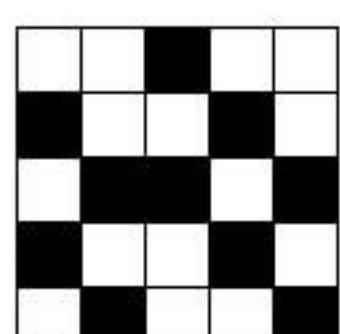
- 如果一个像素使用 4×4 模式，一个 $N \times N$ 的图像大小将变为 $4N \times 4N$ ，使图像增大16倍！

1.2 8位灰度图像：打印

- 一种更好的方法：避免放大输出图像——存储一个整数矩阵（标准图案），每个值范围从0到255

- 将灰度图像矩阵与图案进行比较，当值大于灰度值时打印点

0	14	22	5	8
18	9	1	19	13
6	24	16	7	23
21	2	12	20	3
10	15	4	11	17



一个25级灰度的例子：左边是标准图案，右边的灰度值为15

1.2 8位灰度图像：打印

- 一种使用 $n \times n$ 抖动矩阵的有序抖动算法如下：

```
BEGIN
    从  $x = 0$  到  $x_{\max}$                                 // columns
        从  $y = 0$  到  $y_{\max}$  // 行
             $i = x \bmod n$ 
             $j = y \bmod n$ 
            //  $I(x, y)$  是输入， $O(x, y)$  是输出,
            //  $D$  是抖动矩阵。
            如果  $I(x, y) > D(i, j)$ 
                 $O(x, y) = 1$ ;
            否则
                 $O(x, y) = 0$ 
END
```

1.2 8位灰度图像：打印

•示例

•使用 300*300DPI 的打印机在一张 (12.8*9.6 英寸) 的纸上打印一幅 (240*180*8 位) 的图像，每个像素（点）的尺寸是多少？

$$-(300*12.8)^*(300*9.6) = 3480 * 2880 \text{ 点}$$
$$-(3480/240)^*(2880/180) = 16 * 16 = 256$$

1.2 8位灰度图像：打印

•使用标准矩阵生成输出图像

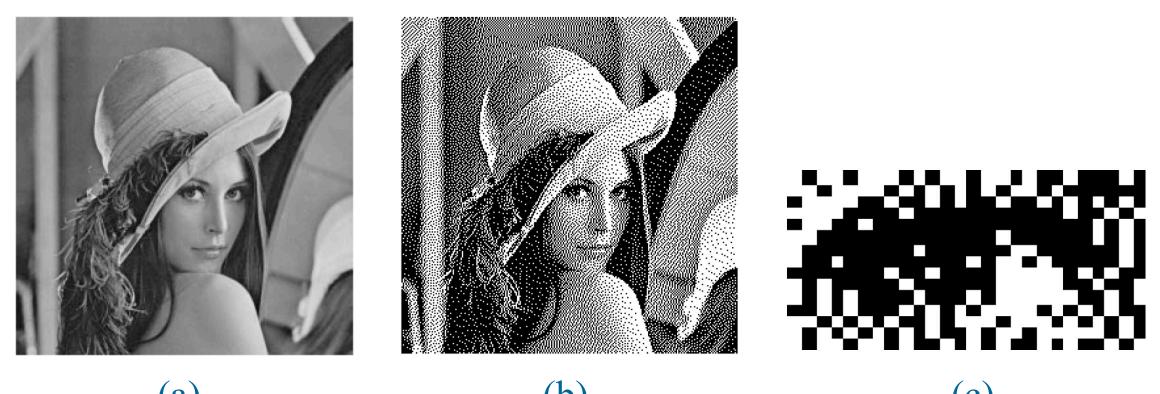


图3.4：灰度图像的抖动处理。

(a)：8位灰度图像“lenagray.bmp”。(b)：该图像的抖动处理版本。(c)：抖动处理版本的细节。

问题？

•用分辨率为 300*300DPI 的打印机在一张 8×6 英寸的纸上打印一幅 600×450×8 位的图像，每个像素（点）的尺寸是多少？

$$-(300*8)^*(300*6) = 2400*1800 \text{ 个点}$$
$$-(2400/600)^*(1800/450) = 4 * 4 \text{ 仅 } 17 \text{ 个灰度级}$$

- 将图像大小缩小至 150*113？
- 将灰度级从 256 降低到 16？

1.3 24位彩色图像：案例

编辑

牛肉味

云母屏风染上一层浓浓的烛影，
银河逐渐斜落启明星也已下沉。

嫦娥想必悔恨当初偷吃下灵药，

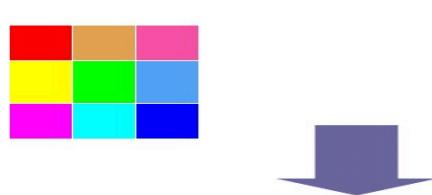
碧海青天夜夜心！

嫦娥奔月



1.3 24位彩色图像：特征

- 每个像素使用三个字节：代表 RGB
 - 取值范围从 0 到 255；
 - 支持 $256 \times 256 \times 256$ 种颜色，即 16,777,216 种
- 每个像素由 RGB 的不同灰度值描述



$$R = \begin{bmatrix} 255 & 240 & 240 \\ 255 & 0 & 80 \\ 255 & 0 & 0 \end{bmatrix} \quad G = \begin{bmatrix} 0 & 160 & 80 \\ 255 & 255 & 160 \\ 0 & 255 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 80 & 160 \\ 0 & 0 & 240 \\ 255 & 255 & 255 \end{bmatrix}$$

1.3 24 位彩色图像：大小

- 640×480 的 24 位彩色图像，921.6KB
 - $640 \times 480 \times 3$ 字节
- 许多 24 位彩色图像实际上以 32 位图像形式存储
 - 每个像素的额外数据用于存储一个值，指示特殊效果信息（例如，透明度标志）

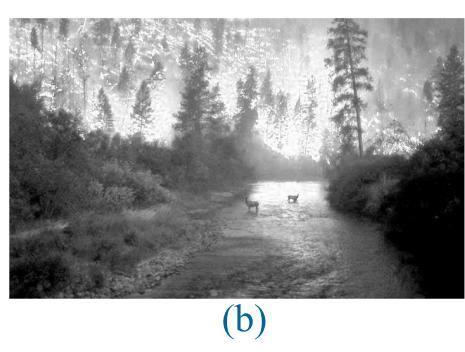


•半透明图像颜色 = 源图像颜色 $\times (100\% \times (1 - \text{透明度})) +$ 背景图像颜色 $\times \text{透明度}$

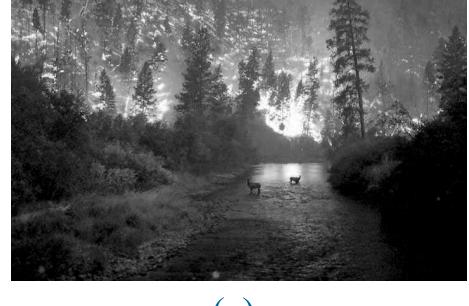
1.3 24位彩色图像



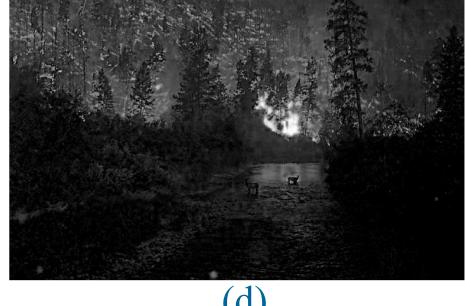
(a)



(b)



(c)

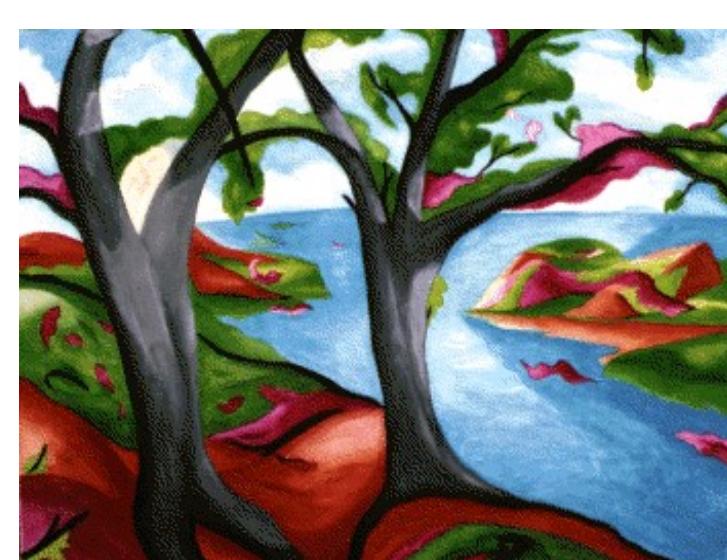


(d)

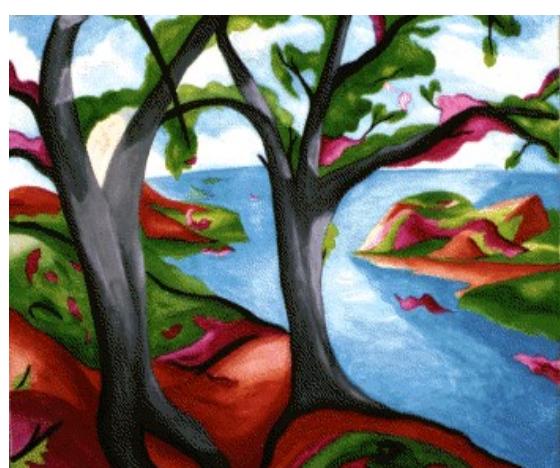
图3.5：高分辨率彩色图像及单独的 R、G、B 颜色通道图像。(a)：24位彩色图像“forestfire.bmp”示例。(b, c, d)：该图像的 R、G 和 B 颜色通道

1.4 8位彩色图像：案例

也称为 256 色图像



1.4 8位彩色图像：案例



8位彩色图像与24位彩色图像对比

1.4 8位彩色图像：特点

- 使用查找表（调色板）的思路

- 图像存储一组字节，而非实际颜色——字节值是一个3字节颜色表的索引——选择在表中放置哪些颜色很重要

- 选择最重要的256种颜色

- 通过对 $256 \times 256 \times 256$ 种颜色进行聚类生成——中值切割算法——中值切割算法的更精确版本

1.4 8位彩色图像



24位彩色图像



8位彩色图像

-注意8位图像相对于24位图像在空间上的巨大节省：一幅 640×480 的8位彩色图像仅需 300 kB 的存储空间，而一幅彩色图像则需要 921.6 kB（同样，未应用任何压缩）。

1.5 颜色查找表：案例

- 通过调整查找表改变颜色

- 查找表（LUT）小于图像，具有速度优势。示例：更改查找表

索引	R	G	B
1	255	0	0

转换为

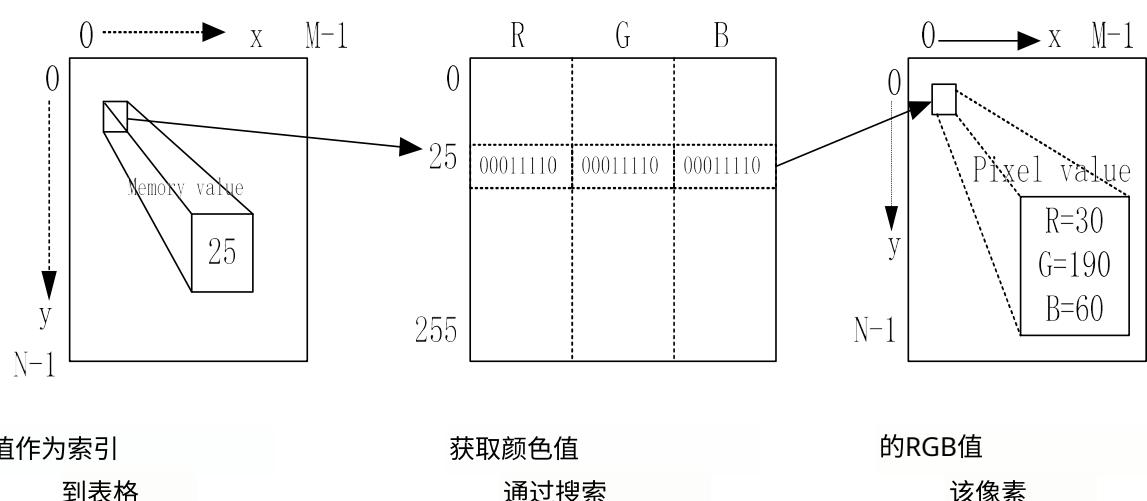
索引	R	G	B
1	0	255	0

将颜色索引加1，即将红色转换为绿色

- 一个重要应用：医学图像——将灰度图像转换为彩色图像

1.5 颜色查找表：案例

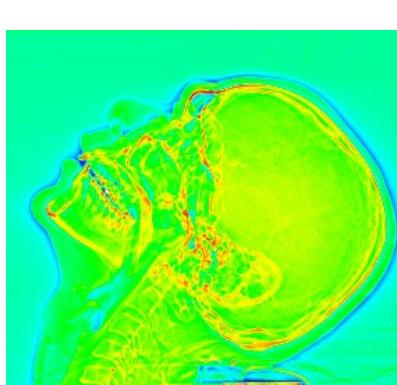
-8位彩色图像采用的思路是，仅为每个像素存储索引或代码值。例如，如果一个像素存储的值为25，这意味着要在颜色查找表（LUT）中查找第25行。



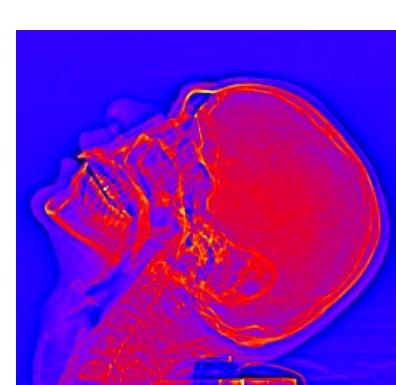
1.5 颜色查找表：医学图像



灰度图像



彩色图像 - 1



彩色图像 - 2

通过修改查找表将灰度图像转换为彩色图像

1.5 颜色查找表：医学图像

索引	R	G	B
0	0	0	0
1	1	1	1
2	2	2	2
3	3	3	3
...
64	64	64	64
65	65	65	65
66	66	66	66
67	67	67	67
68	68	68	68
69	69	69	69
...
254	254	254	254
255	255	255	255

正常灰度LUT

索引	R	G	B
0	0	0	0
1	0	0	7
2	0	0	15
3	0	0	23
...
64	0	255	255
65	0	255	247
66	0	255	239
67	0	255	231
68	0	255	223
69	0	255	215
...
254	255	248	248
255	255	252	252

彩虹编码LUT

索引	R	G	B
0	0	0	0
1	0	0	4
2	0	0	8
3	0	0	12
...
64	0	0	255
65	4	0	255
66	8	0	255
67	12	0	255
68	16	0	255
69	20	0	255
...
254	255	255	248
255	255	255	252

热金属编码LUT

1.6 如何设计颜色查找表

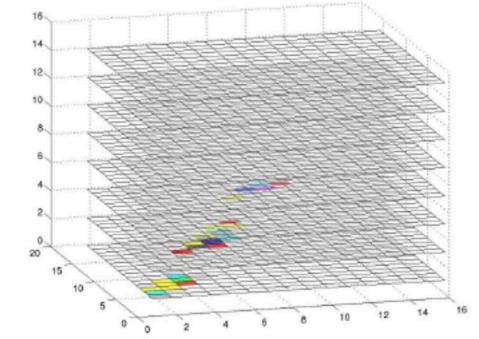
- 示例：8位彩色图像

- 人类对 R 和 G 比 B 更敏感
- 所以 $R = 3, G = 3$ 和 $B = 2$

RRRGGBB

- 基本思路——聚类

- 分析RGB颜色的三维直方图
- 聚类是一项代价高昂的且缓慢的过程



1.6 如何设计颜色查找表

- 将24位颜色转换为8位查找颜色的最直接方法是在每个维度上将RGB立方体划分为相等的切片。

- (a) 每个生成的立方体的中心将作为颜色查找表 (LUT) 中的条目，而只需将RGB范围0..255缩放到适当的范围即可生成8位代码。
- (b) 由于人类对 R 和 G 比 B 更敏感，我们可以将R范围和G范围0..255缩小到3位范围0..7，并将B范围缩小到2位范围0..3，这样总共构成8位。
- (c) 为了缩小 R 和 G ，我们可以简单地将 R 或 G 字节值除以 $(256/8) = 32$ ，然后截断。然后图像中的每个像素都被其8位索引替换，颜色查找表用于生成24位颜色。



1.6 如何设计颜色查找表

- 中值切割算法：一种简单的替代解决方案，能更好地解决这个颜色缩减问题。

- (a) 其思路是对 R 字节值进行排序并找出其中值；然后，小于中值的值标记为“0”位，大于中值的值标记为“1”位。
- (b) 这种方案确实会将比特集中在最需要区分大量相近颜色的地方。
- (c) 最容易可视化查找中值的方法是使用一个显示位置 0 到 255 计数的直方图。(d) 图 3.11 展示了 forestfire.bmp 图像的 R 字节值的直方图，以及这些值的中值（用垂直线表示）。

1.6 如何设计颜色查找表

- 中值切割算法

R	0	1						
G	00	01	10	11				
B	000	001	010	011	100	101	110	111
	...							
	...							
	...							

1.6 如何设计颜色查找表

- 然后图像中的每个像素都被其8位索引替换。

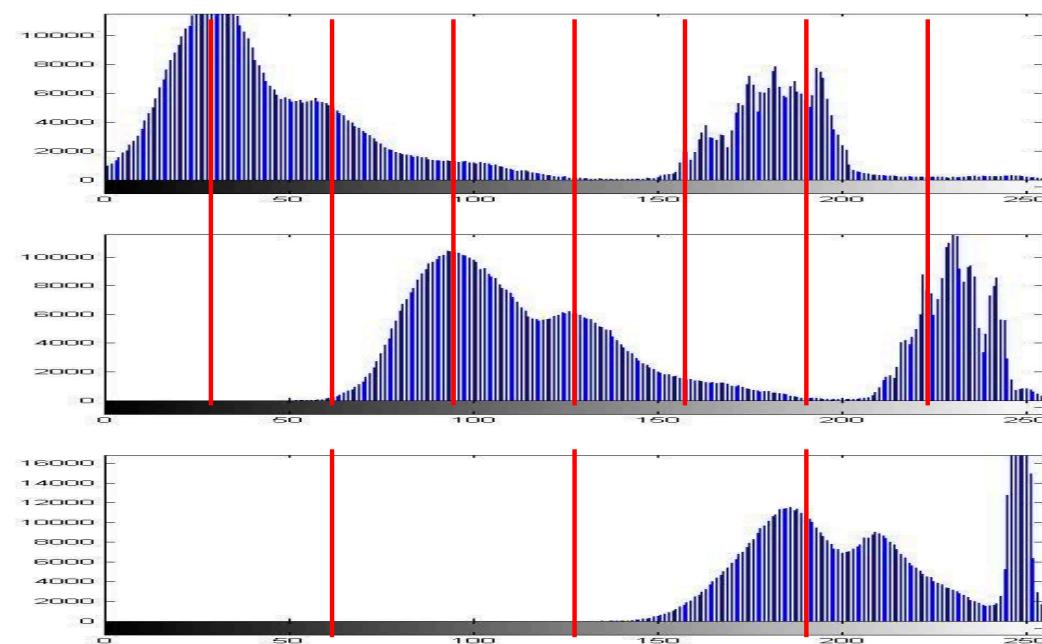
Eg.

- R: 16, 48, 80, 112, 144, 176, 208, 240
- G: 16, 48, 80, 112, 144, 176, 208, 240
- B: 32, 96, 160, 224

颜色为 [30, 129, 80] 的像素应转换为：
[16, 112, 96]

1.6 如何设计颜色查找表

- 我们能否取得更好的结果？



1.6 如何设计颜色查找表

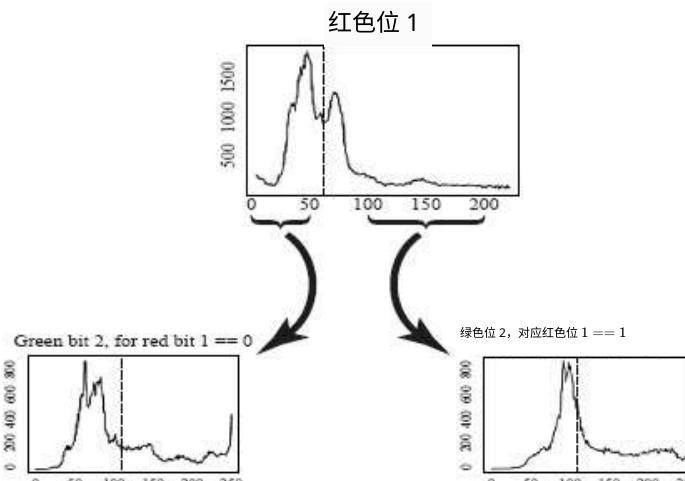


图 3.11 24 位彩色图像“forestfire.bmp”的 R 字节直方图为每个像素生成“0”位或“1”位标签。对于正在构建的颜色表索引的第二位，我们取小于 R 中位数的 R 值，并仅针对这些像素，根据其 G 值小于或大于 G 值的中位数，将这些像素标记为“0”或“1”，仅针对“0”红色位像素。对 R、G、B 继续处理 8 位，得到一个 8 位颜色查找表索引

2、常用图像文件格式

- GIF
- JPEG -
- BMP -
- PNG -
- TIFF -
- EXIF -
- 其他

2.1 GIF图像：案例



多媒体基础——图形与图像数据表示（2024年春季）

41

2.1 GIF图像：特点

- GIF（图形交换格式）
- 由优利系统公司（UNISYS Corporation）和Compuserve于1987年发明
- 最初用于通过电话线传输图形图像

- 不属于任何应用程序，目前几乎所有相关软件都支持

Fundamentals of Multimedia — Graphics and Image Data Representation (2024 Spring)

42

2.1 GIF图像：特点

- 使用LZW（莱姆佩尔 - 齐夫 - 韦尔奇）压缩算法
 - LZW算法是具有连续色彩的无损格式，压缩率约为50%
- 仅限于8位（256色）图像
 - GIF图像深度从1位到8位
 - GIF图像支持256种颜色

多媒体基础——图形与图像数据表示（2024年春季）

43

2.1 GIF图像：特性（续）

- 隔行扫描
 - 解码速度快
 - 采用隔行扫描方式存储
 - 可分四次逐步显示
- 支持GIF89a动画格式
 - 在一个图像文件中存储多个彩色图像

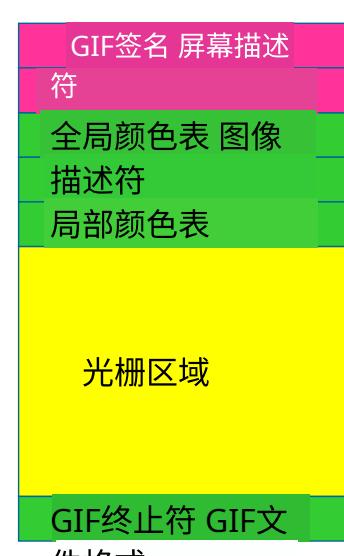


多媒体基础——图形与图像数据表示（2024年春季）44

2.1 GIF图像：案例分析

- 一张120*160的GIF图像

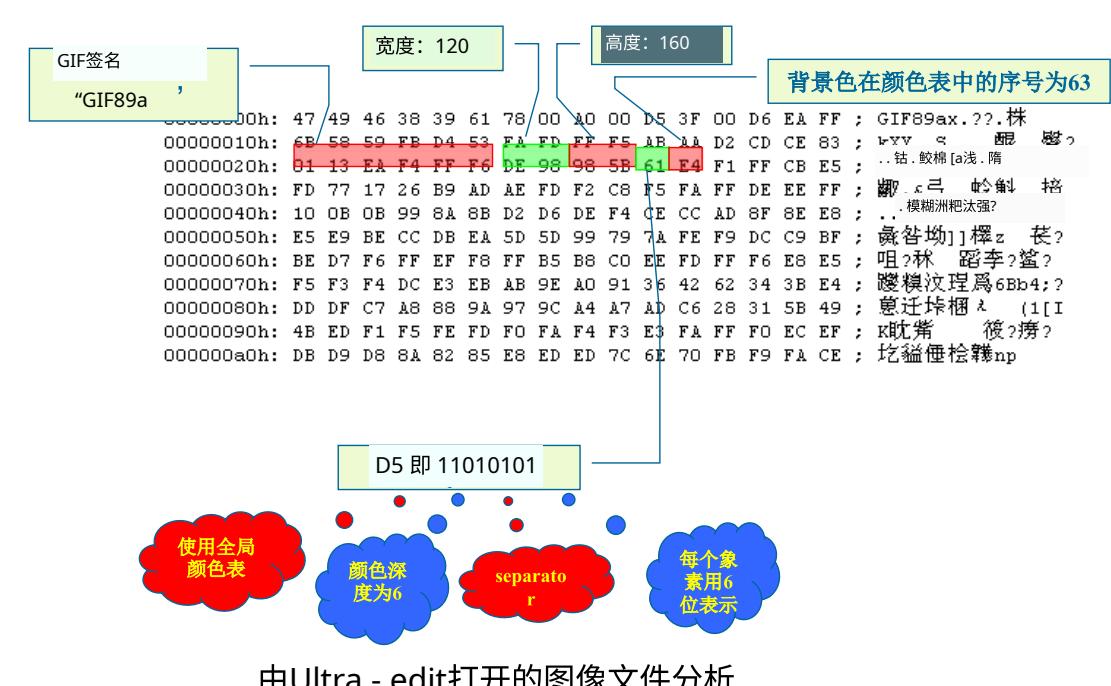
偏移量 长度 内容 0 3字节
节 "GIF"
3个3字节 "87a" 或 "89a"
6 2 bytes <Logical Screen Width>
8个2字节 <逻辑屏幕高度>
10个1字节 位0：全局颜色表标志 (GCTF) 位1..3：颜色分辨率 位4：全局颜色表排序标志 位5..7：全局颜色表大小： $2^{(1+n)}$ 1字节 <背景颜色索引
1f....GIF：文件头信息 偏移量、长度、内容



Fundamentals of Multimedia — Graphics and Image Data Representation (2024 Spring)

45

2.1 GIF图像：案例分析

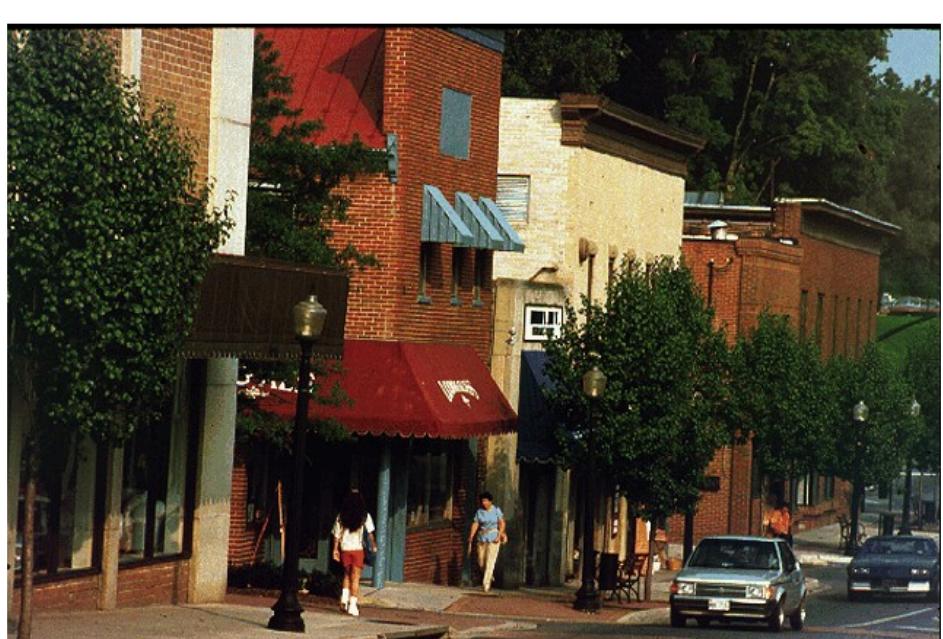


由Ultra-edit打开的图像文件分析

多媒体基础——图形与图像数据表示（2024年春季）

46

2.2 JPEG图像：案例



Fundamentals of Multimedia — Graphics and Image Data Representation (2024 Spring)

47

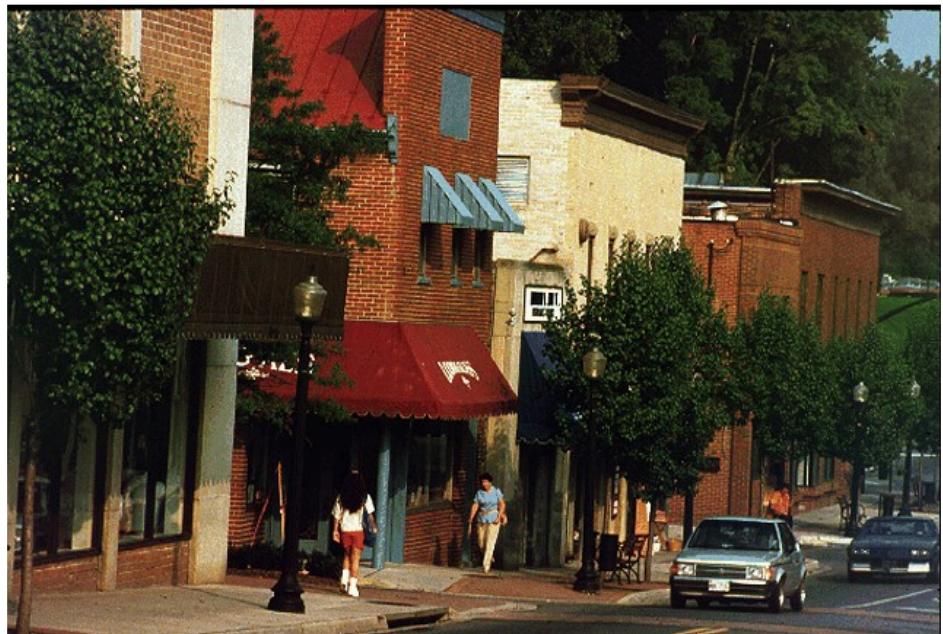
2.2 JPEG图像：特点

- JPEG（联合图像专家组）
- 由国际标准化组织（ISO）的任务组创建组织（ISO）
- 利用人类视觉的一些局限性系统
- JPEG实现高压缩率
- 一种有损压缩方法
 - 允许用户设置所需的质量级别或压缩比（输入除以输出）

多媒体基础——图形与图像数据表示（2024年春季）

48

2.2 JPEG图像：示例1



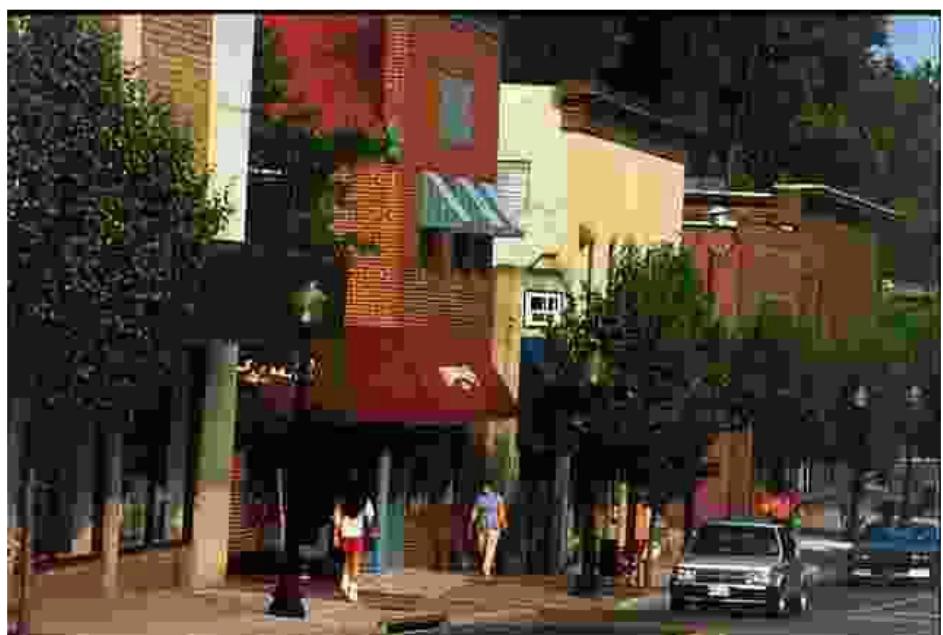
JPEG图像 (1): 252千字节

2.2 JPEG图像：示例2



JPEG图像 (2): 45.2千字节

2.2 JPEG图像：示例3



JPEG图像 (3): 9.21千字节

2.3 BMP图像

•由微软创建，作为Windows的主要图像格式，可存储1位、4位、8位数据以及真彩色数据

•BMP文件有三种存储形式：

- 无压缩的原始数据，最为常用
 - 行程长度编码：用于8位图像（256色）BI - RLE8
 - 行程长度编码：用于4位图像（16色）BI_RLE4

2.3 BMP图像

•BMP文件由四个部分组成：

- 文件头：类型及其他信息
- 位图信息头：长度、宽度、压缩算法等
- 调色板：颜色查找表，24位真彩色图像无调色板
- 图像数据：真彩色图像存储（R, G, B）三个分量，带调色板的图像存储调色板索引

2.3 BMP图像

•BMP文件：案例分析

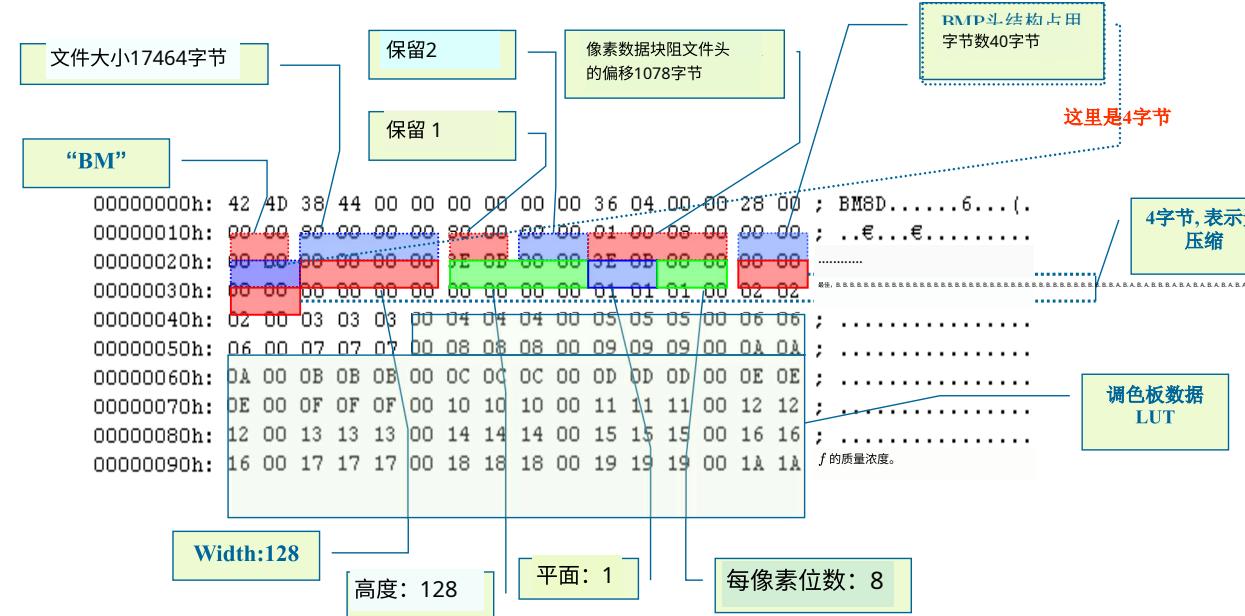
- 128×128 灰度 Lena 图像



偏移量 长度 内容 0 2 字
节 "BM"
2 4 字节 包含 "BM" 的总大小 6 2
字节 保留位 1
8 2 字节 保留位 2 10 4 字节
偏移字节数 14 4 字节 头部大小
(n) 18 n - 4 字节 头部(见右侧)
14 + n .. s - 1 图像数据 BMP 文
件格式：偏移量、长度
内容

偏移量 长度 内容 18 4 字节
宽度 22 4 字节 高度 282 2
每像素位数 30 4 字节
压缩方式 34 4 字节 图像大
小 38 4 字节 每米 X 像素数
42 4 字节 每米 Y 像素数
46 4 bytes Number of Colors
50 4 字节 重要颜色
54 (n - 40) 字节 OS/2 新扩
展字段

2.3 BMP 图像



通过 Ultra - edit 打开的图像文件分析

2.3 其他典型图像格式

- PNG（便携式网络图形）
- TIFF（标签图像文件格式）
- EXIF（可交换图像文件）
- 其他

结束

