

## 图形和图像的数据表现

- ▶ 1 图像的基本属性
- ▶ 2 图形与图像的类型
- ▶ 3 常见的图形与图像文件格式

林福宗： c5.4、c5.5：彩色数字图像基础  
Ze-Nian Li： c3：图形和图像的数据表现



## 1 图像的基本属性

### ▶ 1.1 分辨率-Resolution

- ▶ **显示分辨率**：指显示屏上能够显示出的像素数目。同样大小显示屏能够显示的像素越多，说明显示设备的分辨率越高，显示的图像质量也就越高。（ $640 \times 480$ ， $1024 \times 768$ ）
- ▶ **图像分辨率**：指组成一副图像的像素的密度，一般用单位长度上包含像素的个数来衡量。常用单位为DPI（dots per inch），即每英寸多少点。

### ▶ 1.2 像素深度

- ▶ 像素深度是指存储每个像素所用的位数。像素深度决定彩色图像每个像素可能有的颜色数，或者确定灰度图像每个像素可能有的灰度级数。
- ▶ **RGB16位，24位，32位，8位（256色）**

例 图像像素深度为3，则可显示8色

R	G	B	颜色
0	0	0	黑色
0	0	1	蓝色
0	1	0	绿色
0	1	1	青色
1	0	0	红色
1	0	1	品红色
1	1	0	黄色
1	1	1	白色

3

## 2 图形与图像的类型

▶ 2.1 按照有无颜色信息来分\_灰度图像、彩色图像

▶ ① 1位图像

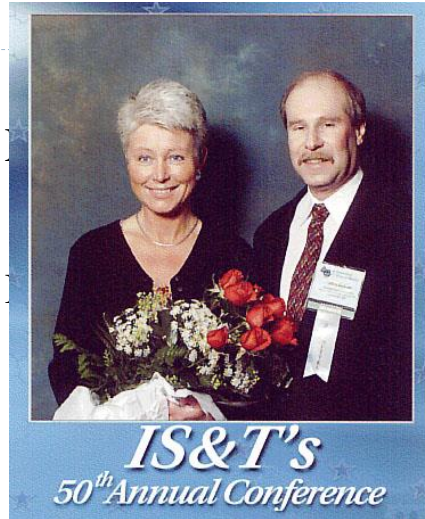
▶ 二值图像：每个像素取值0或者1

▶ 也称单色图像

▶ 存储空间要求小



## Standard test image

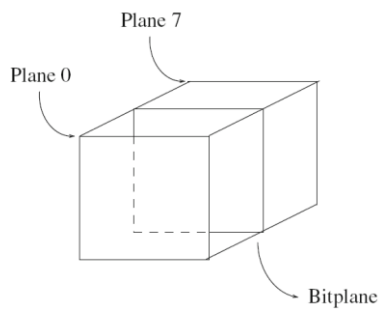


**Imaging Experts Meet Lenna in Person in Boston on May 1997**

▶ 5

## ②8位灰度图像

- ▶ 每个像素取值0~255（1个字节）
- ▶ 整幅图像是由像素值组成的二维数组，bitmap
- ▶ 存储空间要求比二值图像大
- ▶ 位平面 Bitplane



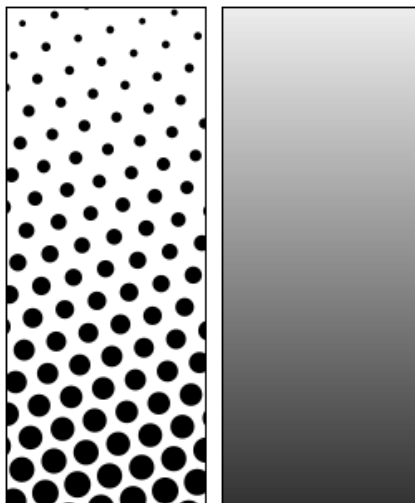
▶

## 思考：激光打印机如何打印图像

- ▶ 假设激光打印机设备分辨率**600 DPI** (dot-per-inch)
  - ▶ 每个点只能是 打印/不打印 两种状态
  - ▶ 那么600x600图像，只能打印在1英寸见方的区域内
  - ▶ 很明显，打印出来的图像太小了
- ▶ 如何打印出更大的图像呢？
  - ▶ 抖动-Dithering，一种用空间分辨率代替亮度分辨率的策略。
  - ▶ 类似在模拟处理中的半调（halftone）打印



## 半调（halftone）打印



## 抖动-Dithering

- ▶ 用 $N \times N$ 点的矩阵来呈现 $N^2+1$ 级的灰度级效果
- ▶ 例如 $2 \times 2$ 抖动模版，如右图所示，可以呈现5级灰度级
- ▶ 算法规则：
  - ▶ 先将图像转换成5级灰度级
  - ▶ 如果灰度级大于抖动矩阵元素值则打印
- ▶ 抖动模版越大，能呈现的灰度级越大
- ▶  $4 \times 4$ 模板？  $3 \times 3$ 模板？

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

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

5	3	7
1	0	2
8	4	6



## 抖动-Dithering

抖动会使图像的空间分辨率增大 $n \times n$ 倍

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

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

0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
1	0	1	0	1	0	1	0	1	1	1	0	1	1	1	0
0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1



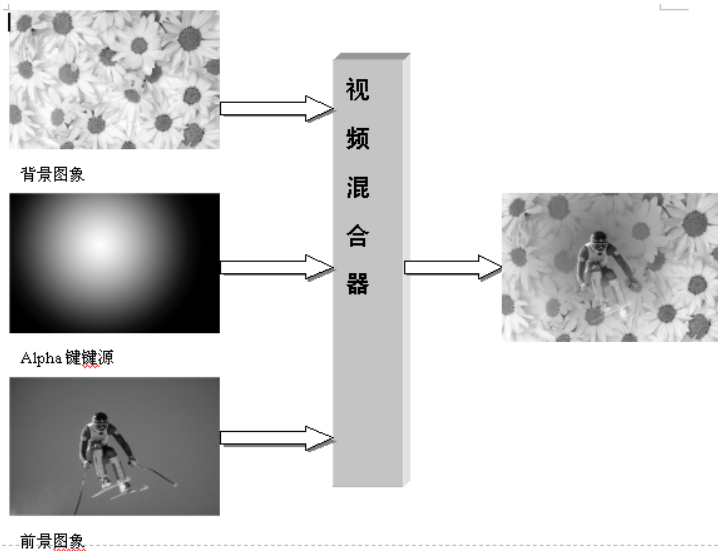


## 阿尔法(α)通道

- ▶ **α**用于表示像素在对象中的透明度
  - ▶ 例如，用两幅图像A和B混合成一幅新图像，新图像C的像素为：
 
$$ColorC_i = \alpha \times ColorA_i + (1 - \alpha)ColorB_i$$
  - ▶ 又如，一个像素(A,R,G,B)的四个分量都用规一化的数值表示时，
    - ▶ 像素值为(1,1,0,0)时显示红色，表示红色强度为1。
    - ▶ 像素值为(0.5,1,0,0)时，使用α通道中的预乘数0.5与R，G，B相乘，其结果为(0.5,0.5,0,0)，表示红色强度为0.5。

▶ 13

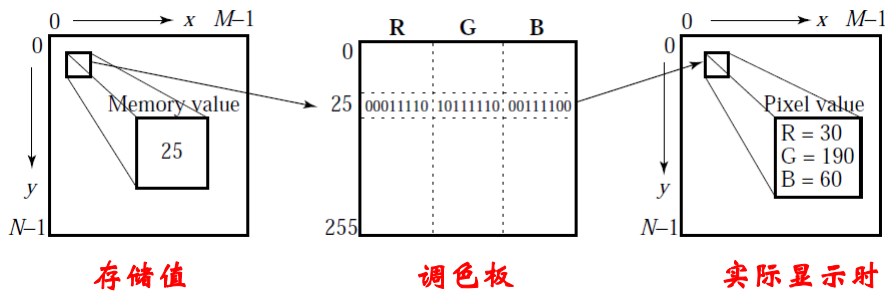
## α键控混合技术



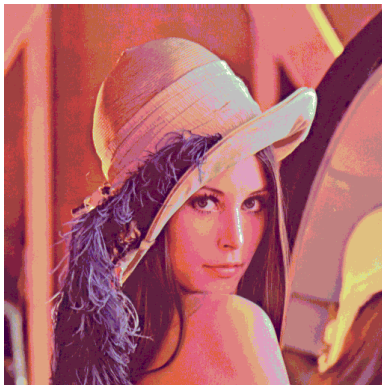
▶ 14

## ④8位彩色图像

- ▶ 利用8位颜色信息（也称为256色）
- ▶ 图像存储的不是颜色，是字节的集一个查找表（也被称为调色板）的
- ▶ 存储空间较小



## 24位和8位图像对比



16 256色标准图像



24位标准图像



## 24位和8位图像对比



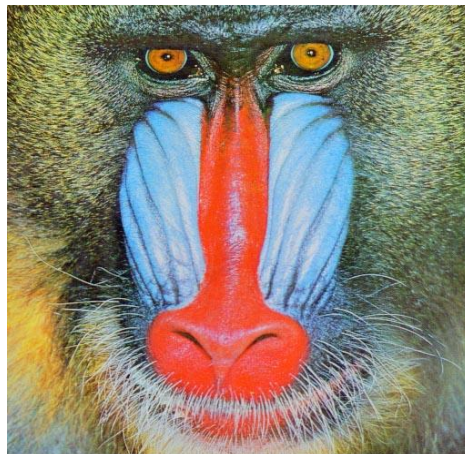
17 256色标准图像



24位标准图像



256色标准图像



24位标准图像

## 真彩色、伪彩色与直接色

- ▶ **真彩色 (true color)** : 在组成一幅彩色图像的每个像素值中, 有R, G, B三个基色分量, 每个基色分量直接决定显示设备的基色强度
- ▶ **伪彩色 (pseudo color)** : 每个像素的颜色不是由每个基色分量的数值直接决定, 而是把像素值当作彩色查找表(调色板)的表项入口地址, 去查找一个显示图像时使用的R, G, B强度值, 用查找出的R, G, B强度值产生彩色
- ▶ **直接色 (direct color)** : 每个像素值分成R, G, B分量, 每个分量作为单独的索引值对它做变换。也就是通过相应的彩色变换表找出基色强度, 用变换后得到的R, G, B强度值产生的彩色称为直接色。

▶ 19

## 复习回顾

### 2.1 按照有无颜色信

- ▶ 一个640\*480分辨率

某图片分辨率为48775×5441, 颜色深度32位,  
 $48775 \times 5441 \times 32 / 8 = 1061539100$  字节  
 $\approx 0.99\text{GB}$

- ▶ 1位图像

$$640 \times 480 / 8 = 37.5\text{KB}$$

- ▶ 8位灰度图像

$$640 \times 480 \times 8 / 8 = 307200\text{B} = 300\text{KB}$$

- ▶ 24位彩色图像

$$640 \times 480 \times 24 / 8 = 307200 \times 3\text{B} = 900\text{KB}$$

- ▶ 8位彩色图像

某专业单反相机拥有3630万像素, 其最大分辨率  
 $7360 \times 4912$ ,  $3630\text{万} \times 24 / 8 \approx 103.8\text{MB}$

▶

## 2.2 从生成、显示、处理和存储的角度划分 ——矢量图和点位图

### 矢量图

- ▶ 矢量图是用一系列计算机指令来表示一幅图
- ▶ 存放矢量图的存储格式称为**矢量图格式**；
- ▶ 存储的数据主要是绘制图形的数学描述和指令。
- ▶ 将每个图元对象作为一个独立体处理，缩放、旋转、移动时不会影响其它对象。



▶ 21

### 矢量图

- ▶ **优点：**计算生成，显示精度高、操作灵活性大。
- ▶ **缺点：**图像显示时花费时间比较长；真实世界的彩色图像难以转化为矢量图。

#### 常用文件格式：

- ▶ 如Adobe Illustrator的\*.AI、\*.EPS和\*.SVG、AutoCAD的\*.dwg和\*.dxf、Corel DRAW的\*.cdr、windows标准图元文件\*.wmf和增强型图元文件\*.emf 等
- ▶ Adobe Illustrator、AutoCAD、Corel Draw 等绘图软件都是基于矢量的系统。

矢量图



放大后的矢量图



▶ 22

## 点位图 (bit map)

- ▶ 用像素值阵列表示的图，是图像在空间上的离散化。
- ▶ 画位图或编辑位图的软件称为**画图程序**(**paint programs**)；
- ▶ 存放位图的格式称为位图格式；存储的内容是描述像素的数值。
- ▶ 位图的获取通常用扫描仪、数码相机、摄像机、录像机、视像光盘和相关的数字化设备

点阵图



放大后的点阵图



▶ 23

## 点位图 (bit map)

### 优点：

- ▶ 适合表现细致、色彩和层次比较丰富的图像，是所有视觉图像表示方法的基础。
- ▶ 显示速度快。真实世界的图像可以通过扫描仪、数码相机、摄像机等设备方便的转化为位图。

### 缺点：

- ▶ 存储和传输时数据量比较大。
- ▶ 缩放、旋转时算法复杂且容易失真。

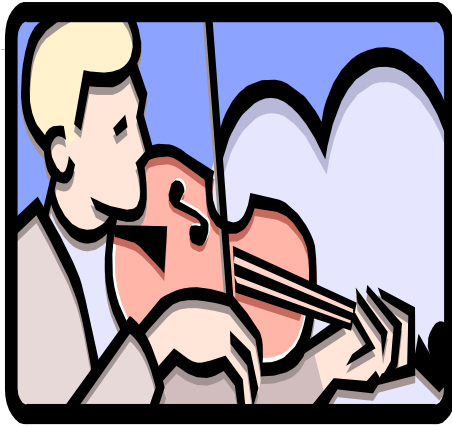
点阵图



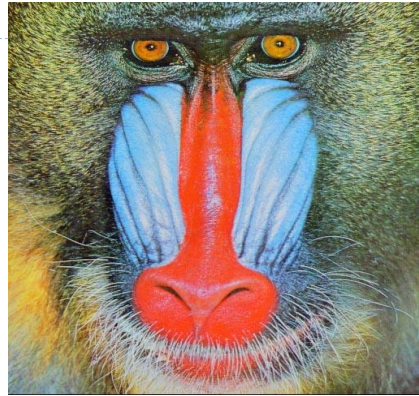
放大后的点阵图



▶ 24



矢量图



点位图

**Q: 矢量图和点位图可以互换吗?**

photoshop里的栅格化(raster)

▶ 25

## 2.3 图形与图像的区别与联系

### 从显示内容的角度

- ▶ 一般来说，图像所表现的显示内容是自然界的真实景物，或利用计算机技术逼真地绘制出的带有光照、阴影等特性的自然界景物
- ▶ 图形实际上是对图像的抽象，组成图形的画面元素主要是点、线、面或简单立体图形等，与自然界景物的真实感相差很大
- ▶ **图形更多的用矢量图表示，图像更多的用位图表示**

一旦在屏幕上显示，图形与图像无异

- ▶ 都是以一定的分辨率和颜色深度在屏幕上以点阵的形式显示出来。

▶ 26

### 3 常见的图形与图像文件格式

**Table 3.1** Adobe Premiere  
file formats

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

### 3 常见的图形与图像文件格式

- ▶ BMP文件格式
- ▶ GIF文件格式
- ▶ JPG文件格式（以后的章节中）
- ▶ 需要说明的：
  - ▶ 图像存储时一般由两部分组成：**图像说明**部分和**图像数据**部分
    - ▶ 图像说明部分：图像的**格式、深度、高度、宽度、调色板、压缩方法**等
    - ▶ 图像数据部分：描述图像每个像素的数据

有关各种图形图像文件的文档参考网站：  
<http://www.martinreddy.net/gfx>



### 3.1 BMP图像文件格式

位图文件(Bitmap-File , BMP)格式是Windows采用的图像文件存储格式，在Windows环境下运行的所有图像处理软件都支持这种格式。BMP位图文件默认的文件扩展名是bmp或者dib。

BMP文件大体上分为四个部分：

位图文件头BITMAPFILEHEADER
位图信息头BITMAPINFOHEADER
调色板Palette(可选)
实际的位图数据ImageData

#### ①位图文件头

00 0C 00 36

0000 1100 0000 0000 0011 0110

	偏移量	域的名称	大小	内容
图像文件头	0000h	标识符 (Identifier)	2 bytes	两字节的内容用来识别位图的类型： ‘BM’ : Windows 3.1x, 95, NT, ... ‘BA’ : OS/2 Bitmap Array ‘CI’ : OS/2 Color Icon ‘CP’ : OS/2 Color Pointer ‘IC’ : OS/2 Icon ‘PT’ : OS/2 Pointer
	0002h	File Size	1 dword	用字节表示的整个文件的大小
	0006h	Reserved	1 dword	保留，设置为0
	<div>00000000h: 42 4D 36 00 0C 00 00 00 00 00 36 00 00 00 28 00 ; BM.....6... (. 00000010h: 00 00 00 02 00 00 00 02 00 00 01 00 18 00 00 00 ; ..... 00000020h: 00 00 00 00 0C 00 00 00 00 00 00 00 00 00 00 00 ; ..... 00000030h: 00 00 00 00 00 00 39 16 53 39 16 52 3E 20 60 BE ; .....9.S9.R&gt; ? 00000040h: 1C 5D 41 1E 61 39 15 5E 3E 1A 5C 3C 1C 5D 44 26 ; .]A.a9.^&gt;.\&lt;.]D&amp; 00000050h: 62 42 1B 5E 3A 17 59 42 1B 5D 4B 1B 5F 45 1C 60 ; bB.^:.YB.]K._E. \</div>			

### ①位图文件头

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	
00000000h:	42	4D	36	00	0C	00	00	00	00	00	36	00	00	00	28	00	; BM6.....6... (
00000010h:	00	00	00	02	00	00	00	02	00	00	01	00	18	00	00	00	; .....
00000020h:	00	00	00	00	0C	00	00	00	00	00	00	00	00	00	00	00	; .....
00000030h:	00	00	00	00	00	00	39	16	53	39	16	52	3E	20	60	BE	; .....9.S9.R> `?
00000040h:	1C	5D	41	1E	61	39	15	5E	3E	1A	5C	3C	1C	5D	44	26	; .]A.a9.^>.\<.]D&
00000050h:	62	42	1B	5E	3A	17	59	42	1B	5D	4B	1B	5F	45	1C	60	; bB.^:.YB.]K._E.`

图像文件头	00000h	(Identifier)	2 bytes	“CP” : OS/2 Color Pointer “IC” : OS/2 Icon “PT” : OS/2 Pointer
	0002h	File Size	1 dword	用字节表示的整个文件的大小
	0006h	Reserved	1 dword	保留, 设置为 0
	000Ah	Bitmap Data Offset	1 dword	从文件开始到位图数据开始之间的数据 (bitmap data) 之间的偏移量

### ①位图文件头结构体

```
typedef struct
tagBITMAPFILEHEADER {
    WORD    bfType;
    DWORD   bfSize;
    WORD    bfReserved1;
    WORD    bfReserved2;
    DWORD   bfOffBits;
} BITMAPFILEHEADER;
```

#### **bfType**

指定文件类型, 必须是 0x424D, 即字符串"BM", 也就是说所有.bmp文件的头两个字节都是"BM"

#### **bfSize**

指定文件大小, 包括这14个字节

#### **bfReserved1, bfReserved2**

为保留字, 不用考虑

#### **bfOffBits**

为从文件头到实际的位图数据的偏移字节数, 即 前三个部分的长度之和。



②位图信息头

000Eh	Bitmap Header Size	1 dword	位图信息头(Bitmap Info Header)的长度,用来描述位图的颜色、压缩方法等。下面的长度表示: 28h - Windows 3.1x, 95, NT, ... 0Ch - OS/2 1.x F0h - OS/2 2.x
0012h	Width	1 dword	位图的宽度,以像素为单位
0016h	Height	1 dword	位图的高度,以像素为单位
001Ah	Planes	1 word	位图的位面数
			每个像素的位数
<div>00000000h: 42 4D 36 00 0C 00 00 00 00 00 36 00 00 00 28 00 ; BM6.....6... (. 00000010h: 00 00 00 02 00 00 00 02 00 00 01 00 18 00 00 00 ; ..... 00000020h: 00 00 00 00 0C 00 00 00 00 00 00 00 00 00 00 00 ; ..... 00000030h: 00 00 00 00 00 00 39 16 53 39 16 52 3E 20 60 BE ; .....9.S9.R&gt; \? 00000040h: 1C 5D 41 1E 61 39 15 5E 3E 1A 5C 3C 1C 5D 44 26 ; .]A.a9.^&gt;.\&lt;.]D&amp; 00000050h: 62 42 1B 5E 3A 17 59 42 1B 5D 4B 1B 5F 45 1C 60 ; bB.^: .YB.]K. _E. ^</div>			
			21 - 21bit (true color) bitmap 32 - 32bit (true color) bitmap

②位图信息头

<div>00000000h: 42 4D 36 00 0C 00 00 00 00 00 36 00 00 00 28 00 ; BM6.....6... (. 00000010h: 00 00 00 02 00 00 00 02 00 00 01 00 18 00 00 00 ; ..... 00000020h: 00 00 00 00 0C 00 00 00 00 00 00 00 00 00 00 00 ; ..... 00000030h: 00 00 00 00 00 00 39 16 53 39 16 52 3E 20 60 BE ; .....9.S9.R&gt; \? 00000040h: 1C 5D 41 1E 61 39 15 5E 3E 1A 5C 3C 1C 5D 44 26 ; .]A.a9.^&gt;.\&lt;.]D&amp; 00000050h: 62 42 1B 5E 3A 17 59 42 1B 5D 4B 1B 5F 45 1C 60 ; bB.^: .YB.]K. _E. ^</div>			
图像信息头			0Ch - OS/2 1.x F0h - OS/2 2.x
	0012h	Width	1 dword 位图的宽度,以像素为单位
	0016h	Height	1 dword 位图的高度,以像素为单位
	001Ah	Planes	1 word 位图的位面数
001Ch	Bits Per Pixel	1 word	每个像素的位数 1 - Monochrome bitmap 4 - 16 color bitmap 8 - 256 color bitmap 16 - 16bit (high color) bitmap 24 - 24bit (true color) bitmap 32 - 32bit (true color) bitmap

图像信息	001Eh	Compression	1 dword	压缩说明 0 - 无压缩 1 - RLE8 (每行 4 字节表示) 2 - RLE4 (每行 8 字节表示) 3 - Bitfields (也使用 BI_BITFIELDS 表示)
	0022h	Bitmap Data Size	1 dword	用字节数表示的位图数据的大小。该数必须是 4 的倍数
	0026h	HResolution	1 dword	用像素/米表示的水平分辨率
	002Ah	VResolution	1 dword	用像素/米表示的垂直分辨率

$biWidth * biHeight$   
其中  $biWidth$  是 4 的倍数，  
水平分辨率 \* 像素深度之后  
取最近的整倍数。

```

00000000h: 42 4D 36 00 0C 00 00 00 00 00 36 00 00 00 28 00 ; BM6.....6... (.
00000010h: 00 00 00 02 00 00 00 02 00 00 01 00 18 00 00 00 ; .....
00000020h: 00 00 00 00 0C 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000030h: 00 00 00 00 00 00 39 16 53 39 16 52 3E 20 60 BE ; .....9.S9.R> \?
00000040h: 1C 5D 41 1E 61 39 15 5E 3E 1A 5C 3C 1C 5D 44 26 ; .]A.a9.^>.\<.]D&
00000050h: 62 42 1B 5E 3A 17 59 42 1B 5D 4B 1B 5F 45 1C 60 ; bB.^.:YB.]K._E.\

```

```

00000000h: 42 4D 36 00 0C 00 00 00 00 00 36 00 00 00 28 00 ; BM6.....6... (.
00000010h: 00 00 00 02 00 00 00 02 00 00 01 00 18 00 00 00 ; .....
00000020h: 00 00 00 00 0C 00 00 00 00 00 00 00 00 00 00 00 ; .....
00000030h: 00 00 00 00 00 00 39 16 53 39 16 52 3E 20 60 BE ; .....9.S9.R> \?
00000040h: 1C 5D 41 1E 61 39 15 5E 3E 1A 5C 3C 1C 5D 44 26 ; .]A.a9.^>.\<.]D&
00000050h: 62 42 1B 5E 3A 17 59 42 1B 5D 4B 1B 5F 45 1C 60 ; bB.^.:YB.]K._E.\

```

图像信息	0022h	Bitmap Data Size	1 dword	用字节数表示的位图数据的大小。该数必须是 4 的倍数
	0026h	HResolution	1 dword	用像素/米表示的水平分辨率
	002Ah	VResolution	1 dword	用像素/米表示的垂直分辨率
	002Eh	Colors	1 dword	位图使用的颜色数。如 8-位/像素表示为 100h 或者 256。
	0032h	Important Colors	1 dword	指定重要的颜色数。当该域的值等于颜色数时，表示所有颜色都一样重要

### ③调色板

调色板数据	0036h	Palette	N * 4 byte	调色板规范。对于调色板中的每个表项，这4个字节用下述方法来描述 RGB 的值： 1 字节用于蓝色分量 1 字节用于绿色分量 1 字节用于红色分量 1 字节用于填充符(设置为 0)
-------	-------	---------	---------------	---



### ④ 实际的图像

这地址偏移量对应的是256色彩色图

图像数据	0436h	Bitmap Data	x bytes	该域的大小取决于压缩方法，它包含所有的位图数据字节，这些数据实际就是彩色调色板的索引号
------	-------	-------------	---------	---

- ▶ 紧跟在调色板之后的是图像数据字节阵列。对于用到调色板的位图，图像数据就是该像素颜色在调色板中的索引值（逻辑色）。对于真彩色图，图像数据就是实际的R、G、B值。
- ▶ 对于2位，4位，8位彩色图而言，一个字节各表示多少个像素？



## 位图数据

- ▶ 对于**2色**位图，用1位就可以表示该像素的颜色（一般0表示黑，1表示白），所以一个字节可以表示8个像素。
- ▶ 对于**16色**位图，用4位可以表示一个像素的颜色，所以一个字节可以表示2个像素。
- ▶ 对于**256色**位图，一个字节刚好可以表示1个像素。
- ▶ 对于**真彩色**图，三个字节才能表示1个像素。耗费空间，但图的颜色显得更亮丽，有得必有失。

▶ 39

## 特别需要注意两点

- ▶ 每一行的字节数必须是4的整倍数，如果不是，则需要补齐。这在前面介绍**biSizeImage**时已经提到了。（注意文件中0出现的位置）
- ▶ 一般来说，BMP文件的数据是**从下到上，从左到右的像素顺序**。也就是说，从文件中最先读到的是图像最下面一行的左边第一个像素，然后是左边第二个像素…接下来是倒数第二行左边第一个像素，左边第二个像素…依次类推，最后得到的是最上面一行的最右一个像素。

▶ 40

## 关于BMP文件可进一步扩展的问题

- ▶ 24位转2位，4位，8位，16位 (r5g5b5, r5g6b5) 位图如何操作？
- ▶ 8位图像转24位图像如何操作？
- ▶ 更改调色板？
- ▶ 假设取得某526色BMP位图数据和调色板，如何显示？
- ▶ ...

▶ 41

## 3.2 GIF图像文件格式

GIF ( Graphics Interchange Format ) 格式由 CompuServe公司于87年开发，版本号**GIF87a**，89年扩充后版本号为**GIF89a**。

GIF图像文件**以块(block)为单位存储信息**。一个GIF文件由表示图形/图像的数据块、数据子块以及显示图形/图像的控制信息块组成，称为**GIF数据流(Data Stream)**。数据流中的所有控制信息块和数据块都必须在文件头(Header)和文件结束块(Trailer)之间。

▶ 42

GIF图像文件格式

1	Header	GIF文件头	
2	Logical Screen Descriptor	逻辑屏幕描述块	
3	Gloabal Color Table	全局颜色表	
	。 。 。 。 。	扩展模块（任选）	
4	Image Descriptor	图像描述块	
5	Local Color Table	局部颜色表	可
6	Table Based Image Data	基于表的图像数据	重
7	Graphic Control Extension	图像控制扩展块	复
8	Plain Text Extension	无格式文本扩展块	n
9	Comment Extension	注释扩展块	次
10	Application Extension	应用程序扩展块	
	。 。 。 。 。	扩展模块（任选）	
11	GIF Trailer	GIF文件结束块	

▶ 43

1	Header	GIF文件头	
---	--------	--------	--

GIF文件头：6个固定字节组成的数据块，它们用来说明使用的文件格式是GIF格式及当前所用的版本号： GIF87a , GIF89a.

▶

2

Logical Screen Descriptor

逻辑屏幕描述块

- ▶ 逻辑屏幕描述块：包含定义[每个]图像显示区域的参数，包括背景颜色信息。

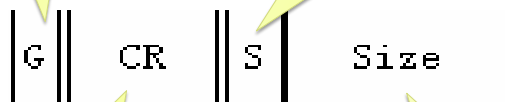
7	6	5	4	3	2	1	0	字节号	域的名称	类型
Logical Screen Width								0	逻辑屏幕宽度	Unsigned
								1	(以像素为定单位)	
Logical Screen Height								2	逻辑屏幕高度	Unsigned
								3	(以像素为定单位)	
G	CR			S	Size			4	包装域	见图6-04
Background Color Index								5	背景颜色索引	Byte
Pixel Aspect Ratio								6	像素宽高比	Byte

45

- ▶ 第5个字节（包装域）的分析：

b7: 有无全局彩色表

b3: 全局彩色表中的颜色是否按重要性排序



b6b5b4: 每种基色的位数

b2b1b0: 每个像素的位数

46

全局彩色表: Global Color Table Flag

- ▶ 彩色表的表项数目等于 $2^{(n+1)}$ ，其中 $n = \text{b2b1b0}$
- ▶ 每个表项由3个字节组成，分别代表R、G、B的相对强度，因此彩色表的字节数就等于 $3 \times 2^{(n+1)}$ 。

7 6 5 4 3 2 1 0	字节号	域的名称	数据类型
red intensity	0	红色索引 000	Byte
green intensity	1	绿色索引 000	Byte
blue intensity	2	蓝色索引 000	Byte
red intensity	3	红色索引 001	Byte
green intensity	4	绿色索引 001	Byte
blue intensity	5	蓝色索引 001	Byte
...	...	...	
...	...	...	
red intensity	745	红色索引 255	Byte
green intensity	746	绿色索引 255	Byte
blue intensity	767	蓝色索引 255	Byte

▶ 47

## GIF图像文件格式

- ▶ GIF图像文件格式可包含数量无限的图像，而且也没有一个固定的存放顺序，仅用一个**字节的图像分隔符**(Image Separator)来判断是不是图像描述块。
- ▶ 每一幅图像都由一个**图像描述块**(Image Descriptor)、**可有可无的局部彩色表**(Local Color Table)和**图像数据**组成。
- ▶ 每幅图像必须要落在逻辑屏幕描述块(Logical Screen Descriptor)中定义的逻辑屏(Logical Screen)尺寸范围里。

▶ 48



## GIF图像文件格式的特点

- ▶ GIF文件采用了LZW无损压缩算法来存储图像数据。
- ▶ GIF文件允许设置背景的透明属性。
- ▶ GIF文件格式可在一个文件中存放多幅彩色图像并且制作出幻灯片或者动画效果。
- ☞ GIF文件支持图像数据的交叉存储方式，这样一个大的图像可以逐步显示，让用户首先看到图像概貌，然后逐步清晰。
- ☞ GIF文件定义的形象中可以加入文本。

## 3.3 TIFF-图像文件格式

- ▶ 标签图像文件格式（Tagged Image File Format）
- ▶ 一种主要用来存储包括照片和艺术图在内的图像的文件格式
- ▶ 它最初由Aldus公司与微软公司一起为PostScript打印开发
- ▶ 是一种无损的图片格式

### 3.3 PNG-图像文件格式

- ▶ 便携网络图形格式Portable Network Graphic Format的简称
- ▶ 20世纪90年代中期开始开发的图像文件存储格式，其目的是企图替代GIF和TIFF文件格式，同时增加一些GIF文件格式所不具备的特性
- ▶ 名称来源于非官方的“PNG's Not GIF”，是一种位图文件存储格式
- ▶ 用来存储灰度图像时，灰度图像的深度可多达16位，存储彩色图像时，彩色图像的深度可多达48位，并且还可存储多达16位的 $\alpha$ 通道数据
- ▶ PNG使用从LZ77派生的数据无损压缩算法

▶ 51

### 本章小结

- ▶ 1 图像的基本属性
  - ▶ 分辨率
  - ▶ 像素深度
- ▶ 2 图形与图像的类型
  - ▶ 灰度图、彩色图
  - ▶ 矢量图、位图
- ▶ 3 常见的图形与图像文件格式
  - ▶ BMP（重点）
  - ▶ GIF和其他（了解特点）

▶

## 作业

- ▶ 分别建立矢量图和点位图, 体会并简述二者的不同。
- ▶ 教材P54: 回答练习3, 并设计一个3\*3图像, 画出其抖动后的效果图。
- ▶
- ▶ 教材中练习8(c), 写出用于计算图像RGB数据颜色直方图的伪代码。
- ▶ \* 选做练习: 仔细分析一个BMP文件的结构。 (使用工具UltraEdit)