第3章

数 据 存 储

本 章 导 读

多媒体数据在计算机中均以二进制形式存储，如图 3-1 所示。3.1 节将介绍数据存储的基本概念，3.2 节～3.5 节将详细介绍文本数据、音频数据、图像数据和视频数据是如何转换为二进制形式存储的。数字数据（整数和实数）的表示和组织方式已经在第 2 章讨论过，本章不再赘述。通过本章的学习，读者将理解多媒体数据是如何转换为二进制存储的。

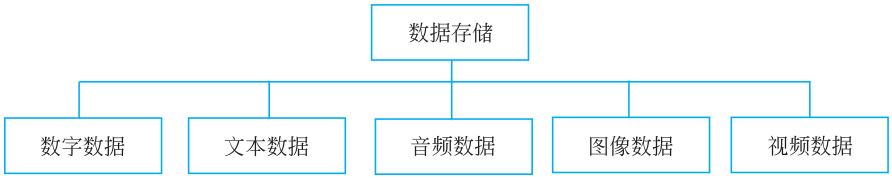


图 3-1　本章导读图

3.1 视频

3.1　数据存储概述

在计算机中，数据存储既涉及如何以有效和一致的方式组织数据，也包含如何在物理介质上保存数据。本章将重点介绍如何在计算机中表示和组织多媒体数据，具体包括数字、文本、音频、图像、视频等。

现实世界中数据是多种多样的，它们在计算机系统中扮演着关键角色。通常将数据划分为如下五类。

（1）**数字数据**：该类数据是最基本的数据类型，通常用于表示数值、统计信息和计算结果等。常见的数字数据包括整数和浮点数，它们的表示方法在第 2 章中已进行了详细介绍。

（2）**文本数据**：该类数据用于表示自然语言的字符、单词和句子。文本数据的应用非常广泛，从简单的文档编辑到复杂的自然语言处理系统，文本数据是现代计算机应用中的核心组成部分。

（3）**音频数据**：音频数据用于表示声音，包括音乐、语音和其他声音信号。通过采样、量化和编码的技术，计算机能够存储和还原高保真的声音信息。

（4）**图像数据**：图像数据是人类视觉的基础，也是自然景物的客观反映。照片、地图、卫星云图以及影视画面等都是图像的具体表示形式。

（5）**视频数据**：视频是由一系列图像组成的连续帧，通过适当的压缩和编码，计算机可以处理高分辨率视频内容。视频数据的存储和处理广泛应用于娱乐、教育等领域。


3.2 视频

3.2　存储文本

在计算机中文本数据存储是通过存储其字符编码的方式实现的。用户在键盘上输入的字母或汉字等文本，计算机都需要将其转换为二进制数据加以存储。这就引出了一个关键问题：如何在计算机中表示这些字符？其实思路很简单：对于每种字符（如英文字符和汉字）为其分配一个字符编码（整数），然后在计算机中存储该字符编码的二进制形式即可。3.2.1 节～3.2.3 节将分别介绍 ASCII 码、汉字编码以及 Unicode 编码等形式。

3.2.1　ASCII 码

在发明计算机的美国最早推出了 ASCII 码（American Standard Code for Information Interchange，美国信息交换标准代码）。ASCII 码是一种字符编码标准，用于在计算机和通信设备中表示文本数据。它通过为每个字符（包括字母、数字、标点符号和控制字符）分配一个唯一的数字代码来实现字符到计算机可处理数据之间的转换。ASCII 码使用 7 位二进制数来表示常用的英文字母、数字和符号，即分别用一个整数表示一种字符，ASCII 码值及其对应字符见表 3-1。

表 3-1　ASCII 码值及其对应字符

ASCII 码值	控制字符	ASCII 码值	控制字符	ASCII 码值	控制字符	ASCII 码值	控制字符
0	NUT（空字符）	14	SO（切换到后续字符）	28	FS（文件分隔符）	42	*
1	SOH（标题开始）	15	SI（切换到前置字符）	29	GS（组分隔符）	43	+
2	STX（文本开始）	16	DLE（数据链路转义）	30	RS（记录分隔符）	44	,
3	ETX（文本结束）	17	DC1（设备控制 1）	31	US（单位分隔符）	45	-
4	EOT（传输结束）	18	DC2（设备控制 2）	32	（space）	46	.
5	ENQ（询问）	19	DC3（设备控制 3）	33	!	47	/
6	ACK（确认）	20	DC4（设备控制 4）	34	"	48	0
7	BEL（响铃）	21	NAK（否定确认）	35	#	49	1
8	BS（退格）	22	SYN（同步）	36	$	50	2
9	HT（水平制表）	23	TB（终止块）	37	%	51	3
10	LF（换行）	24	CAN（取消）	38	&	52	4
11	VT（垂直制表）	25	EM（结束介质）	39	,	53	5
12	FF（换页）	26	SUB（替代）	40	(	54	6
13	CR（回车）	27	ESC（转义）	41	)	55	7

续表

ASCII码值	控制字符	ASCII码值	控制字符	ASCII码值	控制字符	ASCII码值	控制字符
56	8	74	J	92	/	110	n
57	9	75	K	93	]	111	o
58	:	76	L	94	^	112	p
59	;	77	M	95	_	113	q
60	<	78	N	96	、	114	r
61	=	79	O	97	a	115	s
62	>	80	P	98	b	116	t
63	?	81	Q	99	c	117	u
64	@	82	R	100	d	118	v
65	A	83	S	101	e	119	w
66	B	84	T	102	f	120	x
67	C	85	U	103	g	121	y
68	D	86	V	104	h	122	z
69	E	87	W	105	i	123	{
70	F	88	X	106	j	124	|
71	G	89	Y	107	k	125	}
72	H	90	Z	108	l	126	`
73	I	91	[	109	m	127	DEL(删除)

ASCII码表主要包含控制字符、可打印字符、数字、大小写字母和符号。控制字符用于设备控制,如换行LF和回车CR。可打印字符则用于显示文本,范围为32～126。这样的设计旨在确保字符的统一表示,方便计算机与人类之间的交流和信息处理,同时为早期计算机系统提供基本的字符编码支持。

3.2.2 汉字编码

相较于字母文字的语言系统,汉字因复杂性与数量多,需要一整套从输入、处理到显示的编码方案。

1. 汉字输入码

汉字输入码(或称为外码)是指用户通过键盘等设备输入汉字时所使用的编码方式。由于汉字不能像字母一样直接对应键盘上的按键,因此需要通过特定的输入编码将用户的输入映射为汉字。汉字输入码的类型多样,主要包括数字码、拼音码和字形码。

1) 数字码

数字码通过将汉字与数字进行关联,用户通过输入数字代码来选择特定的汉字。例如,区位码就是一种经典的数字码,它为每个汉字分配了唯一的数字组合。用户输入两个数字,分别表示汉字的“区”和“位”,从而确定汉字的位置。这种编码方式相对直接,但对于不熟悉

特定编码的用户可能不够直观。在现代多媒体输入环境中拼音码和字形码则更为常用。

2）拼音码

拼音码是最常见的汉字输入码，它基于汉字的发音（拼音）输入汉字，比如全拼输入法、双拼输入法等，如图 3-2 所示。拼音输入法的优点是学习曲线较低，缺点是存在重码和选择，但是随着智能拼音、自动联想以及预测输入等技术的发展，拼音已经成为现代中文输入法的主流。

图 3-2　双拼输入法

3）字形码

字形码是基于汉字的字形结构或笔画顺序来输入汉字的编码方式。这类输入法的核心思想是根据汉字的形状、部首、笔画等特征来输入。五笔输入法是较著名的字形码之一，如图 3-3 所示。它通过拆分汉字的结构，将汉字的不同部分对应到键盘的按键上，用户通过按键组合输入汉字。

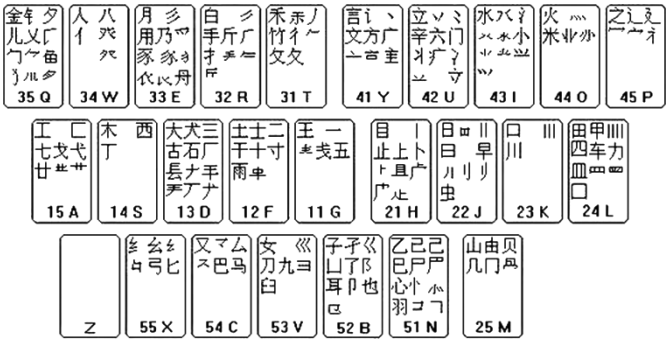


图 3-3　五笔输入法

汉字输入码的多样性反映了汉字输入的复杂性和灵活性。上述三种输入码基本解决了汉字与键盘的映射问题，各自适配了不同用户的输入习惯和需求，推动了中文信息处理的普及和发展。

2. 汉字国标码

1980 年我国发布了《信息技术汉字编码字符集基本集》，也称为简体中文的国家标准码（GB 2312—1980）。该标准定义了 6763 个常用汉字和 682 个图形符号的编码，采用双字节编码方式以便在计算机中表示。每个汉字的编码由两个字节组成，确保计算机能够有效处理和显示汉字，如图 3-4 所示。

2000 年我国又发布了新的扩展标准 GB 18030—2000。该标准收录了 27000 多个汉字，解决了 GB 2312 未能包含的少数民族文字和繁体字问题。而 GB 18030—2005 则进一步实现了对 GB 18030—2000 的扩展，它包含超过 70000 个汉字，并确保汉字与多语言字符

集的兼容性,为中文信息化的国际交流奠定了基础。

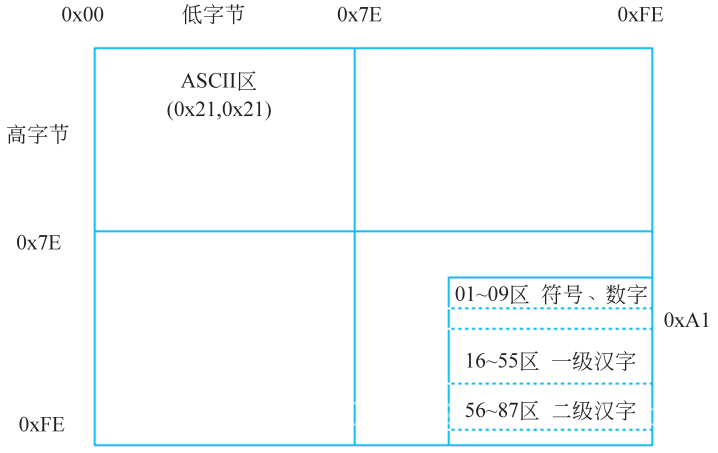


图 3-4 GB 2312

3. 汉字机内码

机内码指的是计算机系统内部存储和处理汉字使用的编码。输入码经过键盘被计算机接收后通过输入码转换模块转换为机内码。机内码和国标码之间有简单的转换规则,以便于不同编码系统之间的兼容性和互操作性。

4. 汉字字形码

汉字字形码用于汉字的显示或打印机输出,如图 3-5 所示。从实现角度看,字形码主要有点阵式和矢量式两种,如图 3-6 所示。点阵式编码是较早的汉字显示和打印方式之一。它通过将汉字的字形表示为一个点阵(像素网格)的形式来进行处理。汉字的每个字形都被分解成一个由点构成的矩阵。每个点的状态(黑或白)决定了汉字在显示或打印时的形状。点阵式编码优点在于直接、简单、易于实现。对于不同的显示设备(如光栅显示器、液晶屏等)和打印机,点阵式可以提供较为一致的字形显示。缺点在于分辨率依赖于点阵的大小。较小的点阵可能导致字形模糊。不易于放大或缩小,放大后容易出现锯齿状边缘。

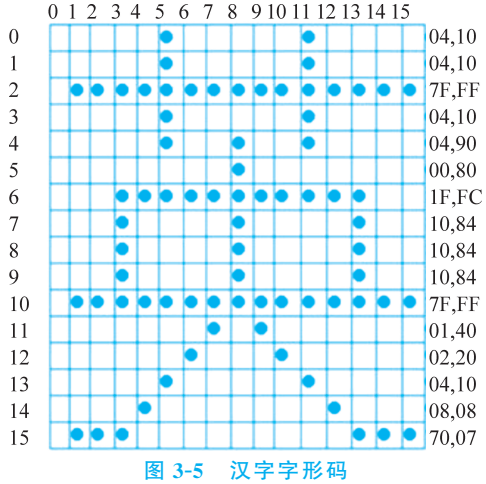


图 3-5 汉字字形码

图 3-6 汉字点阵式编码和矢量式编码

矢量式编码是一种更为现代的汉字显示和打印方式。它通过描述汉字的字形结构、笔

画和曲线来实现字形的显示。汉字的字形被表示为一组数学曲线和几何形状，这些曲线和形状可以根据需要进行缩放和变形。矢量式编码能够提供清晰的字形，无论放大或缩小都不会影响字形的质量。优点在于高质量显示和打印，字形清晰，无论缩放比例如何。缺点是矢量式编码可能需要更多的计算资源来渲染和处理，需要支持矢量图形的设备或软件。

3.2.3　Unicode 编码

虽然 ASCII 码为早期的计算机字符表示提供了有效的解决方案，但随着全球化和计算机应用的普及，ASCII 码的局限性逐渐显现出来。

1. 编码范围有限

ASCII 码使用 7 位二进制数，只能表示 128 个字符。尽管这足以涵盖英语字母、数字和常见符号，但对于其他语言，如中文、日文、韩文，以及很多特殊符号、数学符号和表情符号，ASCII 码无法表示。这使得它无法满足全球多语言环境的需求。

2. 不同地区的字符集冲突

为了弥补 ASCII 码的不足，各国和地区开发了自己的字符集标准，如欧洲的 ISO-8859 系列、中国的 GB 2312、日文的 Shift-JIS 等。这些字符集虽然能够支持各自的语言，但它们之间缺乏统一性，导致了字符编码冲突。例如，在不同字符集中，同一个二进制码点可能对应完全不同的字符，这增加了国际化软件开发的复杂性。

为了解决这些问题，Unicode 字符集应运而生，如图 3-7 所示。它的目标是为世界上所有的语言和符号提供唯一的、统一的编码方案。Unicode 编码是一种全球通用的字符编码标准，它使用更多位数的编码来表示几乎所有已知的书写系统符号和特殊字符。Unicode 字符集的设计初衷是克服 ASCII 码的局限性，并消除不同字符集之间的冲突。Unicode 字符集的编码范围远远超过 128 个字符，能够表示数十万个字符，如图 3-7 所示。

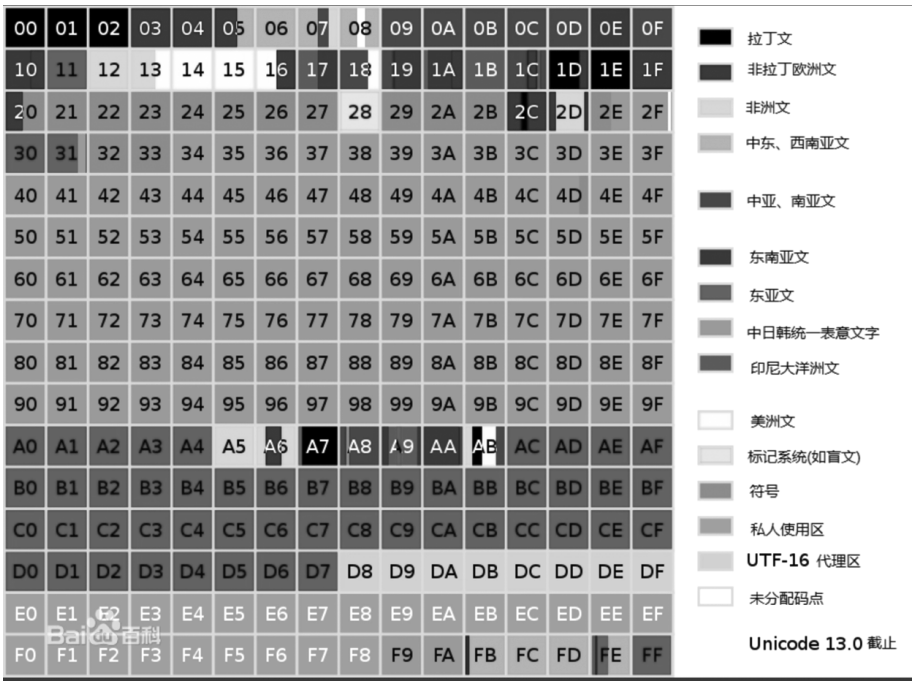


图 3-7　Unicode 字符集

假设读者要在计算机中存储一个简单的中文字符“你”。在 ASCII 码中这是不可能实现的。但在 Unicode 字符集中,“你”被分配了一个唯一的码点(U+4F60),可以通过 UTF-8 或其他编码方式进行存储。例如,“你”在 UTF-8 中表示为三个字节 11100100 10111100 10100000。

3.3 视频

3.3 存储音频

与文本和数字数据不同,声音是由物体振动产生的声波通过介质(如空气、固体或液体)传播,并能被人或动物的听觉器官所感知的波动现象。那么,对于音频数据该如何存储呢?众所周知,声波在时间和幅度上都是连续的,而二进制的数字信号是离散的。将连续的模拟信号(声波)转换为离散的数字信号是实现音频存储的前提,具体而言需要经过采样、量化和编码三个阶段。

3.3.1 采样

音频是典型的模拟数据,计算机不可能记录所有音频信号的幅值,但是可以记录其中的一部分幅值。采样是将连续的模拟信号转换为离散的数字信号的过程。它通过在特定的时间间隔内测量模拟信号的幅度来实现,如图 3-8 所示。根据奈奎斯特定理,为了准确重建原始信号,采样频率(每秒采样的次数)应至少为信号最高频率的 2 倍。常见的采样频率有 44.1kHz(CD 音质)和 48kHz(DVD 音质)。

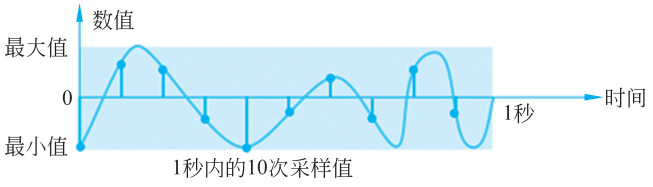


图 3-8 音频信号

例如,如果信号的最高频率是 10kHz,那么采样频率至少应为 20kHz。若采样频率低于这个标准,可能会导致混叠现象,即高频信号在采样过程中被错误地解释为低频信号,从而无法准确重建原始音频信号。

3.3.2 量化

量化是将采样得到的连续幅度值映射到有限的离散数值上的过程,即将模拟信号的幅度转换为数字信号的离散值。换言之,量化就是将样本的值截取为最接近的整数值的过程。例如,实际采样值 17.2 可截取为 17,实际采样值 17.7 则可截取为 18。

3.3.3 编码

编码是将量化后的数字信号转换为特定格式,以便存储或传输的过程。该过程包括数据压缩和格式转换,而数据压缩则进一步分为有损压缩和无损压缩。一般而言,量化后的数字信号会编码为二进制形式进行存储,从而实现模拟信号(音频)到数字信号的转换,进而存储到计算机中。而音频通过音响等外放设备播放时,需要从数字信号转换为模拟信号(即声

波),该转换之所以能顺利进行是由奈奎斯特定理所规定的采样频率所保证的。

3.4　存储图像


3.4 视频

图像是由像素组成的,适用于保存照片和复杂图像,通常用光栅图存储。而图形则由几何形状描述,通常用矢量图存储,适合存储图标和插图,因为它可以任意缩放而不会失真。

3.4.1　图像的基本概念

1. 像素(Pixel)

图像是由像素组成的网格,每个像素表示图像中的一个小点。图像的分辨率是由水平方向和垂直方向上的像素数量决定的,比如分辨率 1920×1080 表示图像有 1920 个水平像素和 1080 个垂直像素。每个像素的数据量取决于图像的颜色深度,图像可以是黑白的、灰度的或者彩色的。

2. 黑白图像

黑白图像只有两种颜色:黑色和白色。每个像素用 1 位(bit)来表示,0 代表黑色,1 代表白色。这样的图像文件体积较小,适合存储简单的图案。一个黑白图像可以用位图(bitmap)来存储,即每个像素用一位表示,按行存储所有像素。

3. 灰度图像

灰度图像包含从黑到白的多种灰色。例如,每个像素用 8 位(bit)表示的图形称为 8 位灰度图像,它表示 0～255 的灰度级别,0 是纯黑色而 255 是纯白色。灰度图像的存储依然是按行存储所有像素,每个像素需要 1 字节(byte)存储。

4. 彩色图像

彩色图像通常使用 RGB 模型,即每个像素由三个颜色通道(红、绿、蓝)的组合来表示。如果每个通道用 8 位表示,通道值范围为 0～255。因此,每个像素需要 3 字节(24 位),这就是 24 位彩色图像,每种颜色由 0～255 的 3 组数字表示。彩色图像的存储依然是按行存储每个像素的数据,每个像素包括三个通道的数据(R,G,B)。对于较复杂的图像,存储数据量会比较大,因此往往需要压缩。

3.4.2　点阵显示

光栅图(Raster Graphics)也称为位图(Bitmap),它将图像表示为一个由像素(点阵)组成的网格,每个像素包含颜色信息。光栅图的分辨率决定了网格的密度,分辨率越高,图像越清晰,但文件大小也会增大。它广泛应用于图像存储、显示以及字符的点阵编码等场景,如图 3-9 所示。光栅图采用逐位映射的方式得到字符的字模编码,其优点是易于定义显示,缺点是空间需求大,如图 3-10 所示为字符 A 的点阵位图及像素图案。

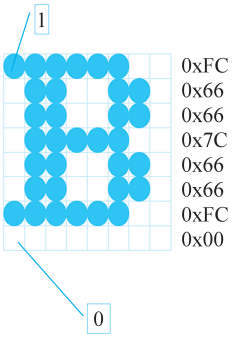


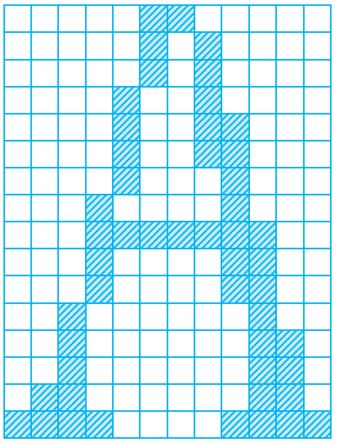
图 3-9　字符 B 的点阵显示

3.4.3　矢量显示

矢量图(Vector Graphics)是通过点、线、曲线和多边形的

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

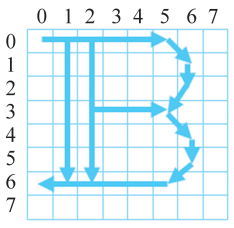
(a) 字符A的点阵位图



(b) 字符A的像素图案

图 3-10 字符 A 的点阵位图及像素图案

描述来表示图像。矢量图形由一系列几何图形的描述组成,通常用数学方程或命令来表示形状的大小、位置和其他特征。它与点阵图的区别在于,矢量图是基于描述图形的几何信息而不是每个像素的颜色。矢量字符显示将字符笔画分解为线段,以线段端点坐标为字符字模的编码,如图 3-11 所示,大写字母 B 可以拆分成不同的线段,存储时只需保存不同线段的起始位置即可。字符的大小和形状由线段的端点坐标决定,这些坐标被存储在矢量字符库中。右侧为字符每一行对应的矢量信息,包括线段的起点和终点坐标。矢量显示的优点在于空间需求小,由于矢量字符只存储线段的端点坐标,而不是每个像素的颜色信息,因此它们通常比点阵字符占用更少的存储空间。矢量显示的缺点在于时间开销大,它们需要经过光栅化过程才能在大多数显示设备上显示。光栅化是将矢量图形转换为点阵图形的过程,这可能会增加显示的计算负担,并在某些情况下影响显示性能。



00 50
50 61
61 62
62 53
53 64
64 65
65 56
56 06
10 16
20 26
23 53

图 3-11 字符 B 的矢量显示

点阵显示和矢量显示在图像处理和存储方面各有特点,它们在时间开销和空间开销上的表现形成了鲜明的对比。点阵显示通过存储每个像素的颜色信息来构建图像,这使得它在显示时无须复杂的转换,兼容性好,易于直接在各种设备上渲染。然而,这种存储方式空间开销较大,尤其是对于高分辨率的图像,因为每个像素都需要存储颜色信息,导致文件体积迅速增加。

与此相反,矢量显示通过存储几何信息和数学公式来定义图像,这大大减少了存储所需的空间,使得矢量图形在空间开销上更为经济。矢量图形在缩放和变换时不会损失清晰度,因为它们是基于数学公式的,这在处理复杂或需要高质量输出的图像时具有优势。但是,矢量图形在显示时需要经过光栅化过程,这可能会增加计算负担,导致在某些情况下显示速度较慢。如图 3-12 所示,它们都能显示相同的字符 B,它们都广泛地应用于图像存储与显示,适用于不同的应用场景和需求。

图 3-12　点阵字符、点阵字库中的位图表示、矢量轮廓字符

3.5　存储视频

视频实际上是一系列连续的静态图像，每秒钟展示的帧数被称为帧率（Frames Per Second，FPS）。例如，常见的视频帧率有 24 FPS、30 FPS 和 60 FPS，帧率越高，视频看起来越流畅。每一帧视频实际上就是一幅图像，因此视频的存储本质上是在存储声音的同时存储多幅图像。每帧的图像信息也会受到图像格式、分辨率等的影响。

原始视频数据量非常大，它通常需要压缩。压缩主要有无损压缩和有损压缩两种方式。无损压缩保留所有信息，不影响质量；有损压缩则丢弃部分信息来减小文件大小，通常不会明显影响视觉效果。视频编码通过对相邻帧之间的变化进行压缩，比如利用预测、运动补偿等技术。常见的视频编码标准有 H.264 和 H.265 等，与 H.264 相比（如图 3-13 所示），H.265 通过更高级的算法实现了更高的压缩率，如图 3-14 所示。H.265 的基本原理在于根据不同画面区域的复杂程度合理配置资源。具体而言，H.265 会将视频帧划分为更小的区域（称为编码单元），并根据每个区域的内容特征动态调整码率。例如，对于简单的、细节较少的画面区域，编码器会使用较少的数据进行存储；而对于细节丰富的区域，则分配更多的比特以确保画质不受影响。

图 3-13　H.264 帧间压缩

图 3-14　H.265 帧内压缩

总体而言，随着视频分辨率的提升和用户需求的变化，采用更高效的编码标准对于保障视频质量、降低带宽消耗具有重要意义。然而，实现这一目标仍需克服技术和设备兼容性方面的挑战。

3.6 小　　结

在计算机中,数据的存储不仅涉及如何在物理介质上保存数据,还包括如何在计算机内部以有效和一致的方式组织和表示这些数据。数据的种类繁多,不同类型的数据在计算机中以不同方式存储和处理。数字、文本、音频、图像和视频是主要的数据类型,每种类型都有其特定的存储要求和处理方法。文本数据在计算机中的表示依赖于字符编码标准。音频信号的存储过程从采样开始,通过测量信号的幅度,将连续的模拟信号转换为离散的数字信号。图像的存储包括点阵式和矢量式两种形式,它们适用于不同的应用场景和需求。视频数据由一系列连续的帧(图像)组成,帧率影响视频流畅度。另外,视频编码与压缩技术至关重要,它可以有效减少视频数据量而不显著影响视频质量。

习　　题

3-1　计算机中数据存储的主要数据类型有哪些?每种类型在存储和处理上有什么特点?

3-2　Unicode 编码与 ASCII 码的主要区别是什么?为什么 Unicode 编码会被广泛采用?

3-3　奈奎斯特定理在音频信号存储中有什么作用?

3-4　光栅图与矢量图的区别是什么?各自适合什么样的应用场景?

3-5　视频数据的存储过程中,压缩的重要性是什么?