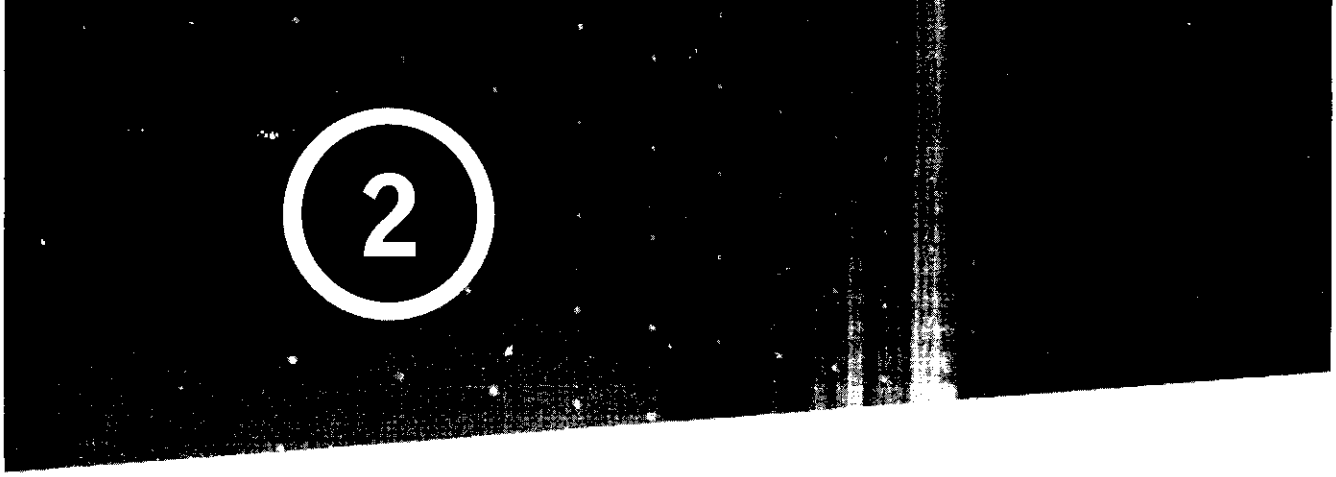


2 Multimedia information representation

2.1 Introduction

All types of multimedia information are stored and processed within a computer in a digital form. In the case of textual information consisting of strings of characters entered at a keyboard, for example, each character is represented by a unique combination of a fixed number of bits – known as a **codeword** – and hence the complete text by a string of such codewords. Similarly, computer-generated graphical images are made up of a mix of lines, circles, squares, and so on, each represented in a digital form. A line, for example, is represented by means of the start and end coordinates of the line relative to the complete image, each coordinate being defined in the form of a pair of digital values.

In contrast, devices such as microphones and many video cameras produce electrical signals whose amplitude varies continuously with time, the amplitude of the signal at any point in time indicating the magnitude of the sound-wave/image-intensity at that instant. As we indicated in Section 1.2, a

signal whose amplitude varies continuously with time is known as an **analog signal**. In order to store and process such signals – and hence types of media – within a computer, it is necessary first to convert any time-varying analog signals into a digital form. In addition, the signals used to operate devices such as loudspeakers – for speech and audio – and computer monitors – for the display of digitized images for example – are also analog signals. Thus digital values representing such media types must be converted back again into a corresponding time-varying analog form on output from the computer.

The conversion of an analog signal into a digital form is carried out using an electrical circuit known as a **signal encoder**. This, as we will expand upon in the next section, operates by first **sampling** the amplitude of the analog signal at repetitive time intervals and then converting the amplitude of each sample into a corresponding digital value. Similarly, the conversion of the stored digitized samples relating to a particular media type into their corresponding time-varying analog form is performed by an electrical circuit known as a **signal decoder**.

In this way, all the media types associated with the various multimedia applications discussed in the last chapter are stored and processed within a computer in an all-digital form. This means that the different media types can be readily integrated together and, equally important, the resulting integrated bitstream can be transmitted over a single all-digital communications network. This chapter is concerned with the way the different media types are represented in their digital form and, where appropriate, the conversion operations that are used. Before discussing this, however, it will be helpful if we first gain an understanding of the general principles relating to how analog signals are converted into their digital form and vice versa.

2.2 Digitization principles

2.2.1 Analog signals

The general properties relating to any time-varying analog signal are shown in Figure 2.1. As we can see in part (a) of the figure, the amplitude of such signals varies continuously with time. In addition, a mathematical technique known as **Fourier analysis** can be used to show that any time-varying analog signal is made up of a possibly infinite number of single-frequency sinusoidal signals whose amplitude and phase vary continuously with time relative to each other. As an example, the highest and lowest frequency components of the signal shown in Figure 2.1(a) may be those shown in Figure 2.1(b).

The range of frequencies of the sinusoidal components that make up a signal is called the **signal bandwidth** and two examples are shown in part (c) of the figure. These relate to an audio signal, the first a speech signal and the second a music signal produced by, say, an orchestra. In terms of speech, humans produce sounds – which are converted into electrical signals by a

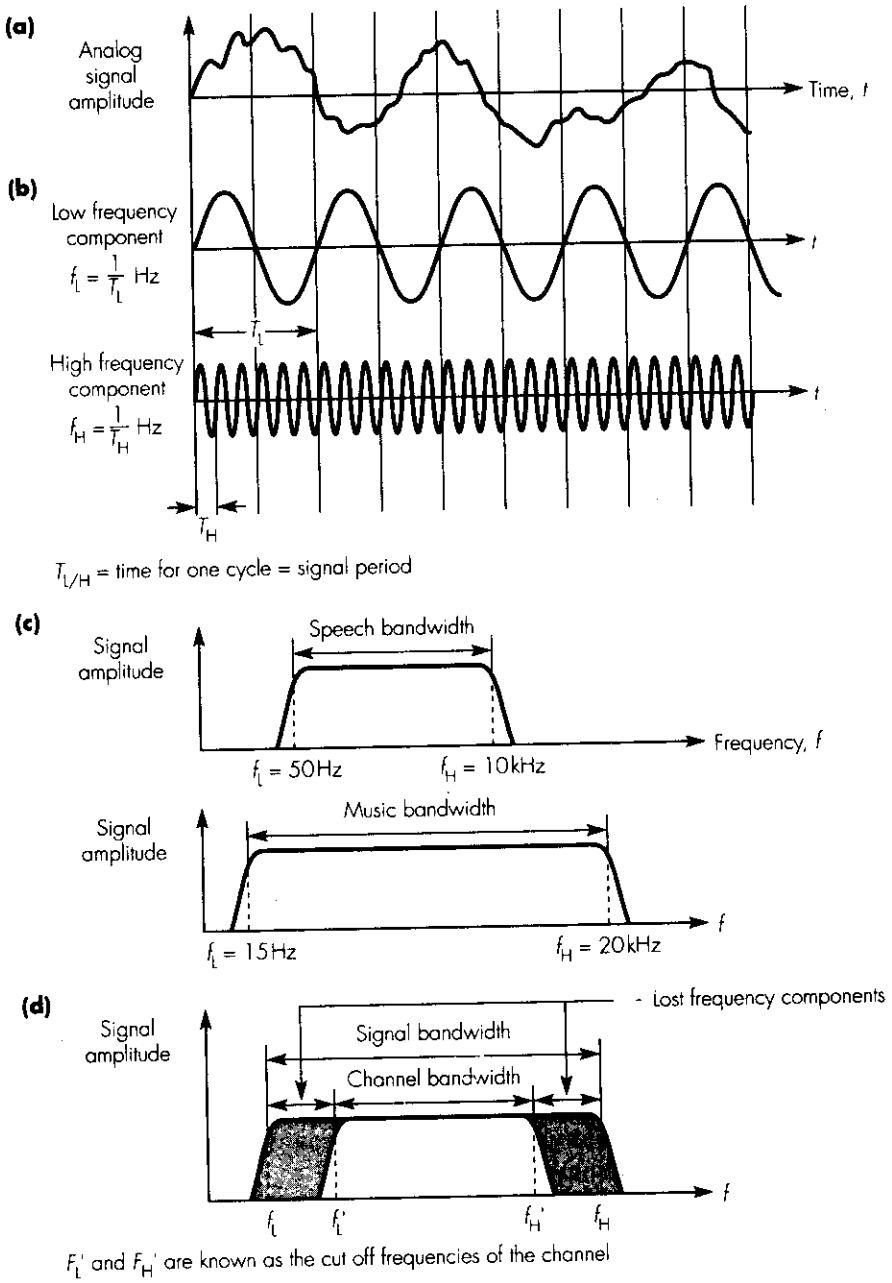


Figure 2.1 Signal properties: (a) time-varying analog signal; (b) sinusoidial frequency components; (c) signal bandwidth examples; (d) effect of a limited bandwidth transmission channel.

microphone – that are made up of a range of sinusoidal signals varying in frequency between 50 Hz and 10 kHz. In the case of a music signal, however, the range of signals is wider and varies between 15 Hz and 20 kHz, this being comparable with the limits of the sensitivity of the ear.

Ideally, when an analog signal is being transmitted through a network, the bandwidth of the transmission channel – that is, the range of frequencies the channel will pass – should be equal to or greater than the bandwidth of the signal. If the bandwidth of the channel is less than this, then some of the low and/or high frequency components will be lost thereby degrading the quality of the received signal. This type of transmission channel is called a **bandlimiting channel** and its effect is shown in Figure 2.1 (d).

2.2.2 Encoder design

As indicated in Section 2.1, the conversion of a time-varying analog signal – of which an audio signal is an example – into a digital form is carried out using an electronic circuit known as a (signal) encoder. The principles of an encoder are shown in Figure 2.2 and, as we can see in part (a), it consists of two main circuits: a **bandlimiting filter** and an **analog-to-digital converter** (**ADC**), the latter comprising a sample-and-hold and a quantizer. A typical **waveform set** for a signal encoder is shown in part (b) of the figure. The role of the bandlimiting filter, as we shall see below, is to remove selected higher-frequency components from the source signal (A). The output of the filter (B) is then fed to the **sample-and-hold** circuit which, as its name implies, is used to sample the amplitude of the filtered signal at regular time intervals (C) and to hold the sample amplitude constant between samples (D). This, in turn, is fed to the **quantizer** circuit which converts each sample amplitude into a binary value known as a codeword (E).

The most significant bit of each codeword indicates the polarity (sign) of the sample, positive or negative relative to the zero level. Normally, a binary 0 indicates a positive value and a binary 1 a negative value. Also, as we can deduce from the time-related set of waveforms shown in Figure 2.2(b), to represent the amplitude of a time-varying analog signal precisely, requires firstly, the signal to be sampled at a rate which is higher than the maximum rate of change of the signal amplitude and secondly, the number of different quantization levels used to be as large as possible. We shall consider each of these requirements separately.

Sampling rate

In relation to the sampling rate, the **Nyquist sampling theorem** states that: in order to obtain an accurate representation of a time-varying analog signal, its amplitude must be sampled at a minimum rate that is equal to or greater than twice the highest sinusoidal frequency component that is present in the signal. This is known as the **Nyquist rate** and is normally represented as either Hz or, more correctly, **samples per second** (**sps**). Sampling a signal at a rate

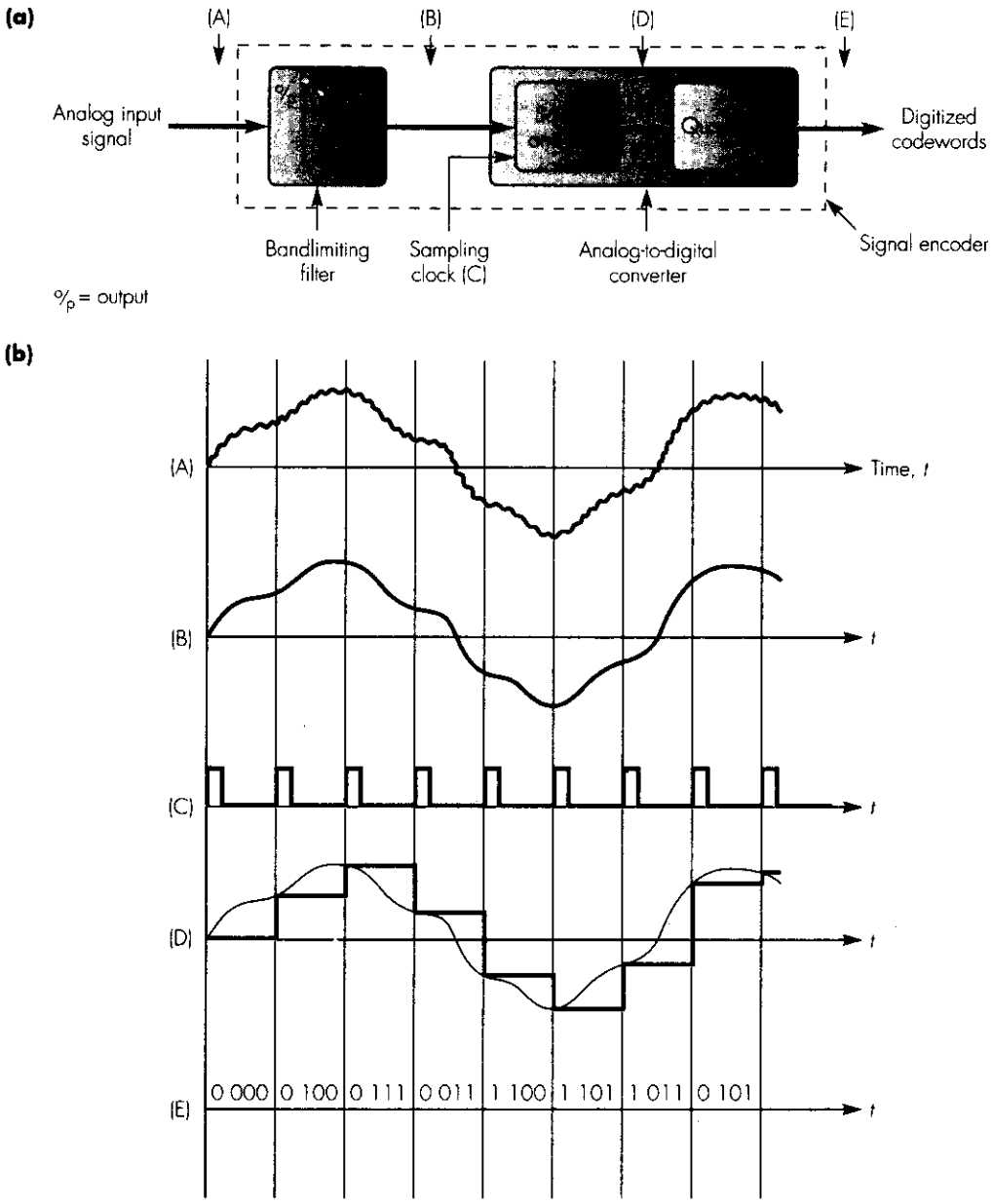


Figure 2.2 Signal encoder design: (a) circuit components; (b) associated waveform set.

which is lower than the Nyquist rate results in additional frequency components being generated that are not present in the original signal which, in turn, cause the original signal to become **distorted**.

The distortion caused by sampling a signal at a rate lower than the Nyquist rate is best illustrated by considering the effect of undersampling a single-frequency sinusoidal signal as shown in Figure 2.3.

In this example, the original signal is assumed to be a 6 kHz sinewave which is sampled at a rate of 8 ksps. Clearly this is lower than the Nyquist rate of 12 ksps (2 × 6 kHz) and, as we can see, results in a lower-frequency 2 kHz signal being created in place of the original 6 kHz signal. Because of this, such signals are called **alias signals** since they replace the corresponding original signals.

In general, this means that all frequency components present in the original signal that are higher in frequency than half the sampling frequency being used (in Hz), will generate related lower-frequency alias signals which will simply add to those making up the original source signal thereby causing it to become distorted. However, by first passing the source signal through a bandlimiting filter which is designed to pass only those frequency components up to that determined by the Nyquist rate, any higher-frequency components in the signal which are higher than this are removed before the signal is sampled. Because of this function the bandlimiting filter is also known as an **antialiasing filter**.

In practice, as we indicated earlier, the transmission channel used/available often has a lower bandwidth than that of the source signal. In such cases, in order to avoid distortion, it is the bandwidth – and hence frequency range – of the transmission channel that determines the sampling rate used rather than the bandwidth of the source signal. Since in such cases the source signal may have higher frequency components than those dictated by the Nyquist rate of the transmission channel, it is necessary first to pass the source signal through a bandlimiting filter which is designed to pass only those sinusoidal frequency components which are within the bandwidth of the transmission channel. In this way, the generation of any alias signals caused by **undersampling** the source signal is avoided.

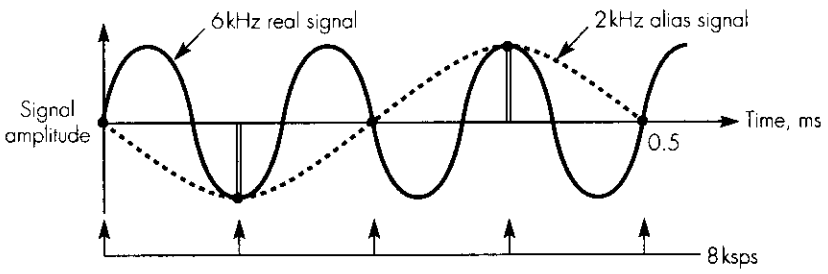
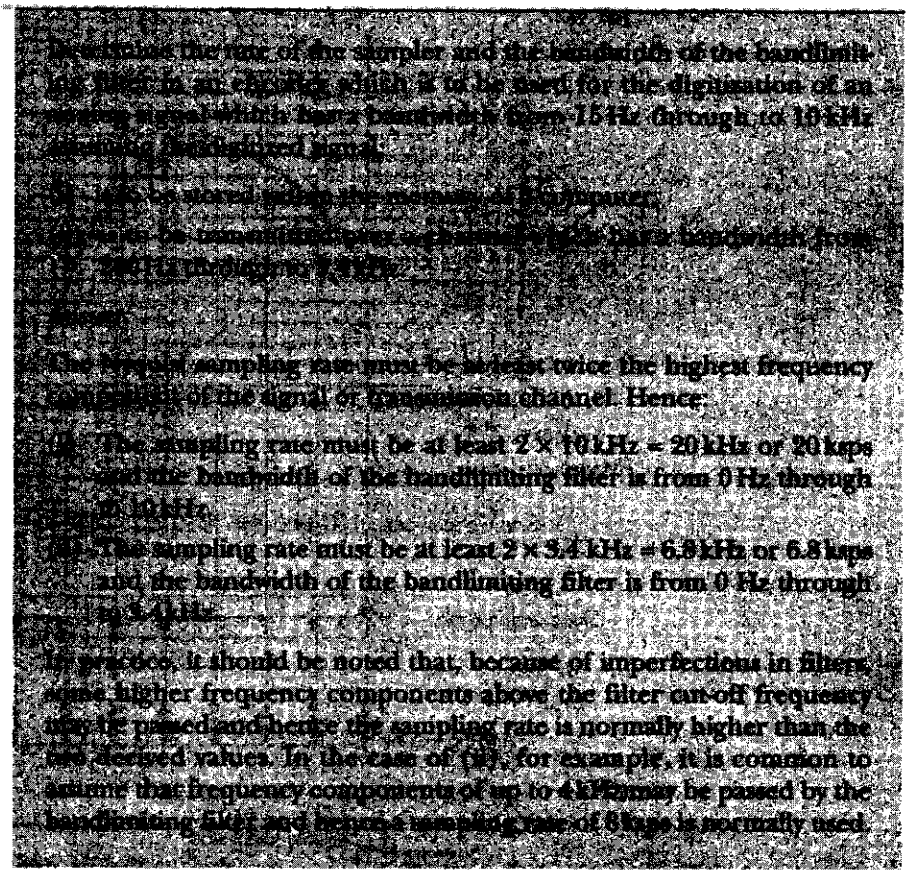


Figure 2.3 Alias signal generation due to undersampling.

Example 2.1



Quantization intervals

To represent in a digital form the amplitudes of the set of analog samples shown earlier in Figure 2.2 precisely, would require an infinite number of binary digits since, when a finite number of digits is used, each sample can only be represented by a corresponding number of discrete levels. The effect of using a finite number of bits can be seen by considering the example shown in Figure 2.4. In this example we use just three bits to represent each sample including a sign bit. This results in four positive and four negative quantization intervals, the two magnitude bits being determined by the particular quantization interval the analog input signal is in at the time of each sample.

As we can deduce from part (a) of the figure, if V_{max} is the maximum positive and negative signal amplitude and n is the number of binary bits used, then the magnitude of each **quantization interval**, q, is given by

$$q = \frac{2V_{max}}{2^n}$$

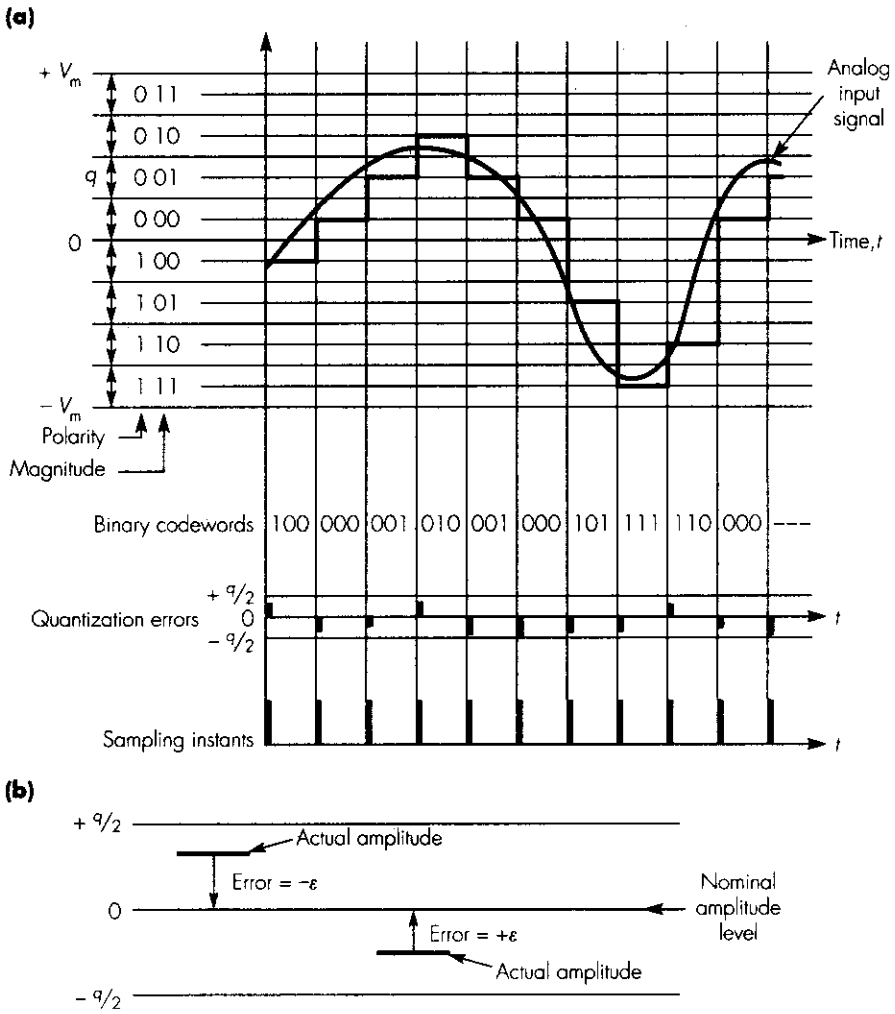


Figure 2.4 Quantization procedure: (a) source of errors; (b) noise polarity.

Also, as we can see, a signal anywhere within a quantization interval will be represented by the same binary codeword. This means that each codeword corresponds to a nominal amplitude level which is at the center of the corresponding quantization interval. Hence the actual signal level may differ from this by up to plus or minus $q/2$.

The difference between the actual signal amplitude and the corresponding nominal amplitude is called the **quantization error** and, for the example shown in Figure 2.4, the quantization error values are shown expanded in part (b) of the figure. Normally, the error values will vary randomly from sample to sample and hence the quantization error is also known as **quantization noise**,

the term "noise" being used in electrical circuits to refer to a signal whose amplitude varies randomly with time.

Another related factor which influences the choice of the number of quantization intervals used for a particular signal is its smallest amplitude relative to its peak amplitude. With high-fidelity music, for example, it is important to be able to hear very quiet passages without any distortion created by quantization noise. The ratio of the peak amplitude of a signal to its minimum amplitude is known as the **dynamic range** of the signal, D. Normally it is quantified using a logarithmic scale known as **decibels** or **dB**:

$$D = 20 \log_{10} (V_{max}/V_{max}) \text{dB}$$

Hence when determining the quantization interval – and thus number of bits to be used – it is necessary to ensure that the level of quantization noise relative to the smallest signal amplitude is acceptable.

Example 2.2

An analog signal has a dynamic range of 40 dB. Determine the magnitude of the quantization noise relative to the minimum signal amplitude if the quantizer uses (i) 6 bits and (ii) 10 bits.

Answer:

$$D = 20 \log_{10} \frac{V_{max}}{V_{min}} \text{ dB} \quad \text{Quantization noise} = \pm \frac{q}{2} = \pm \frac{V_{max}}{2^n}$$

Hence $40 = 20 \log_{10} \dfrac{V_{max}}{V_{min}}$

and $V_{min} = \dfrac{V_{max}}{100}$

(i) $n = 6$ bits

Hence $\dfrac{q}{2} = \pm \dfrac{V_{max}}{2^6} = \pm \dfrac{V_{max}}{64}$

(ii) $n = 10$ bits

Hence $\dfrac{q}{2} = \pm \dfrac{V_{max}}{2^{10}} = \pm \dfrac{V_{max}}{1024}$

As we can see from these values, with 6 bits the quantization noise is greater than V_{min} and hence is unacceptable. With 10 bits, however, the quantization noise is an order of magnitude less than V_{min} and hence will have a much reduced effect.

2.2.3 Decoder design

As indicated in Section 2.1, although analog signals are stored, processed and transmitted in a digital form, normally, prior to their output, they must be converted back again into their analog form; loudspeakers, for example, are driven by an analog current signal. The electronic circuit that performs this conversion operation is known as a (signal) decoder, the principles of which are shown in Figure 2.5.

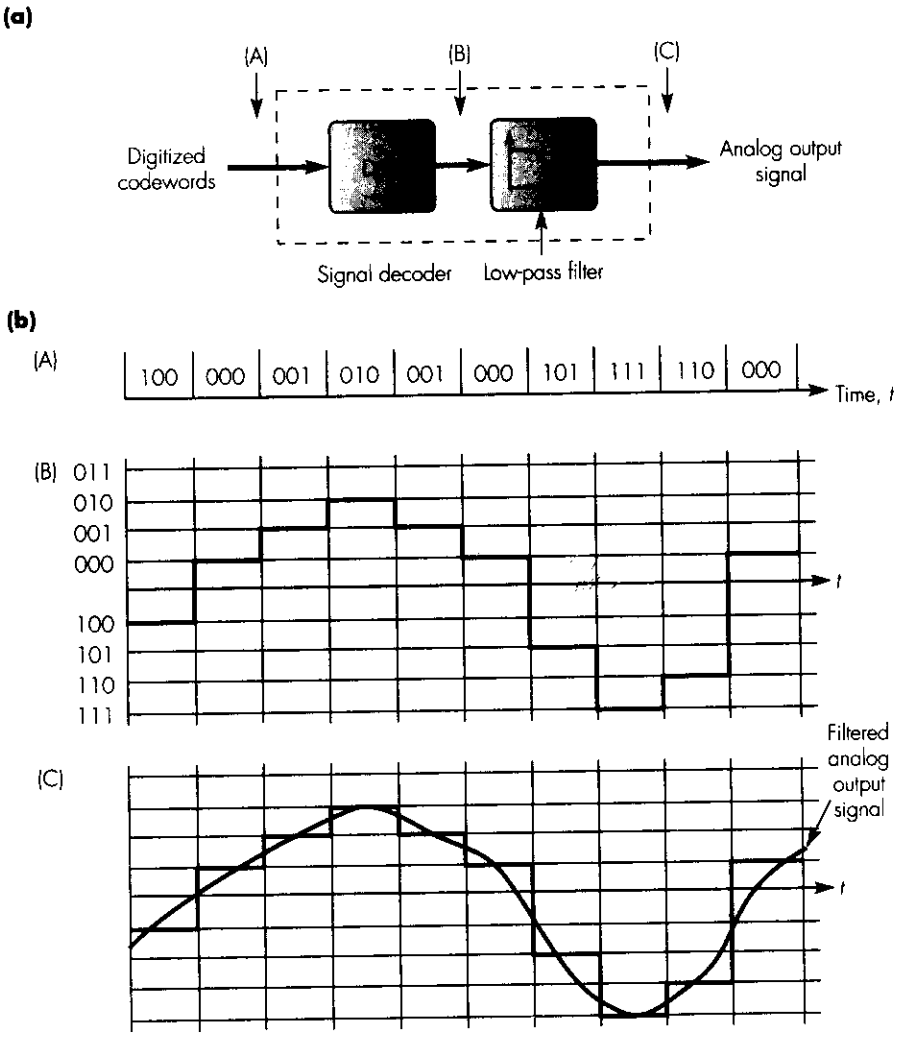


Figure 2.5 Signal decoder design: (a) circuit components; (b) associated waveform set.

First, each digital codeword (A) is converted into an equivalent analog sample using a circuit called a **digital-to-analog converter** or **DAC**. This produces the signal shown in (B), the amplitude of each level being determined by the corresponding codeword. Since this is a time-varying signal, as indicated earlier, Fourier analysis can be used to show that this type of signal comprises not just the sinusoidal frequency components that make up the original (filtered) analog signal, but also an infinite number of additional higher-frequency components. In order to reproduce the original signal, the output of the DAC is passed through a **low-pass filter** which, as its name implies, only passes those frequency components that made up the original filtered signal (C). Normally, the high-frequency cut-off of the low-pass filter is made the same as that used in the bandlimiting filter of the encoder. Because of its function, the low-pass filter is also known as a **recovery** or **reconstruction filter**.

Finally, since in most multimedia applications involving audio and video the communications channel is two-way simultaneous, the terminal equipment must support both input and output simultaneously. Hence the audio/video signal encoders and decoders in each terminal equipment are often combined into a single unit called an **audio/video encoder-decoder** or simply an **audio/video codec**.

2.3 Text

Essentially, there are three types of text that are used to produce pages of documents:

- **unformatted text:** this is also known as **plaintext** and enables pages to be created which comprise strings of fixed-sized characters from a limited character set;
- **formatted text:** this is also known as **richtext** and enables pages and complete documents to be created which comprise of strings of characters of different styles, size, and shape with tables, graphics, and images inserted at appropriate points;
- **hypertext:** this enables an integrated set of documents (each comprising formatted text) to be created which have defined linkages between them.

We shall discuss the three types separately.

2.3.1 Unformatted text

Figure 2.6 illustrates two examples of character sets that are widely used to create pages consisting of unformatted text strings.

The table shown in part (a) tabulates the set of characters that are available in the **ASCII character set**, the term "ASCII" being an abbreviation for the **American Standard Code for Information Interchange**. This is one of the

(a)

Bit positions			7	0	0	0	0	1	1	1	1
			6	0	0	1	1	0	0	1	1
			5	0	1	0	1	0	1	0	1
4	3	2	1								
0	0	0	0	NUL	DLE	SP	0	@	P	\	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	"	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	'	7	G	W	g	w
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	|
1	1	0	1	CR	GS	-	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	~
1	1	1	1	SI	US	/	?	O	—	o	DEL

(b)

Bit positions			7	0	0	0	0	1	1	1	1
			6	0	0	1	1	0	0	1	1
			5	0	1	0	1	0	1	0	1
4	3	2	1								
0	0	0	0					@	P		
0	0	0	1					A	Q		
0	0	1	0					B	R		
0	0	1	1					C	S		
0	1	0	0					D	T		
0	1	0	1					E	U		
0	1	1	0					F	V		
0	1	1	1					G	W		
1	0	0	0					H	X		
1	0	0	1					I	Y		
1	0	1	0					J	Z		
1	0	1	1					K	[		
1	1	0	0					L	\		
1	1	0	1					M	]		
1	1	1	0					N	^		
1	1	1	1					O	—		

Figure 2.6 Two example character sets to produce unformatted text: (a) the basic ASCII character set; (b) supplementary set of mosaic characters.

most widely used character sets and the table includes the binary codewords used to represent each character. As we can see, each character is represented by a unique 7-bit binary codeword. The use of 7 bits means that there are 128 (2^7) alternative characters and the codeword used to identify each character is obtained by combining the corresponding column (bits 7–5) and row (bits 4–1) bits together. Bit 7 is the most significant bit and hence the codeword for uppercase M, for example, is 1001101.

In addition to all the normal alphabetic, numeric and punctuation characters – collectively referred to as **printable characters** – the total ASCII character set also includes a number of **control characters**. These include:

- format control characters: BS (backspace), LF linefeed), CR (carriage return), SP (space), DEL (delete), ESC (escape), and FF (form feed);
- information separators: FS (file separator) and RS (record separator);
- transmission control characters: SOH (start-of-heading), STX (start-of-text), ETX (end-of-text), ACK (acknowledge), NAK (negative acknowledge), SYN (synchronous idle), and DLE (data link escape).

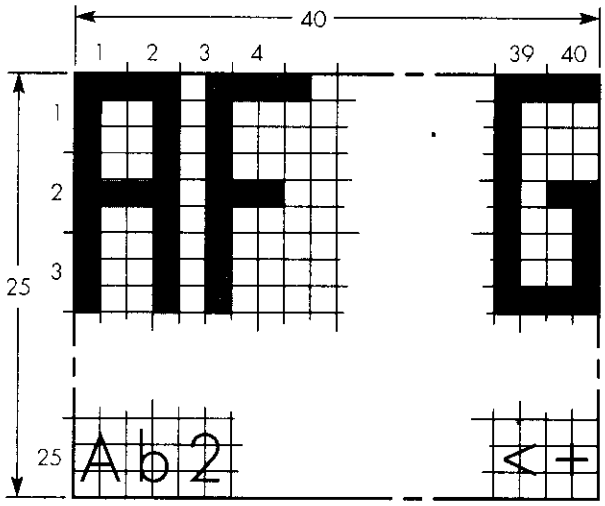
The latter, as we shall describe later in Section 6.4.3, are sometimes used to control the transmission of blocks of characters which are made up of the other characters in the set.

The character set tabulated in Figure 2.6(b) is a supplementary version of that shown in part (a). Here the characters in columns 010/011 and 110/111 are replaced with the set of **mosaic characters** shown. These are then used, together with the various uppercase characters illustrated, to create relatively simple graphical images.

An example application of this particular character set is in **Videotex** and **Teletext** which are general broadcast information services available through a standard television set and used in a number of countries. Some examples of typical Teletext/Videotex symbols are shown in Figure 2.7. As we can see, although in practice the total page is made up of a matrix of symbols and characters which all have the same size (in terms of their height and width), some simple graphical symbols and text of larger sizes can be constructed by the use of groups of the basic symbols.

2.3.2 Formatted text

An example of formatted text is that produced by most word processing packages. It is also used extensively in the publishing sector for the preparation of papers, books, magazines, journals, and so on. It enables documents to be created that consist of characters of different styles and of variable size and shape, each of which can be plain, bold, or italicized. In addition, a variety of document formatting options are supported to enable an author to structure a document into chapters, sections and paragraphs, each with different headings and with tables, graphics, and pictures inserted at appropriate points.



Note: Grid only included as a template.

Figure 2.7 Example Videotex/ Teletext characters.

To achieve each of these features, the author of the document enters a specific command which, typically, results in a defined format-control character sequence – normally a reserved format-control character followed by a pair of other alphabetic or numeric characters – being inserted at the beginning of the particular character string, table, graphic or picture. In this way, each page of the document comprises the string of characters that make up the textual part of the page – plus, where appropriate, the associated tables, graphics, or pictures – with the corresponding format-control character sequences interspersed at appropriate points. An example formatted text string is as shown in part (a) of Figure 2.8 and the printed version of this string is shown in part(b).

As we can deduce from the above, in order to print a document consisting of formatted text, the printer must first be set up – that is, the microprocessor within the printer must be programmed – to detect and interpret the format-control character sequences in the defined way and to convert the following text, table, graphic, or picture into a line-by-line form ready for printing. In addition, to help the author visualize the layout of each page prior to printing, commands such as *print preview* are often provided which cause the page to be displayed on the computer screen in a similar way to how it will appear when it is printed. This is the origin of the term **WYSIWYG**, an acronym for what-you-see-is-what-you-get.

(a)

```
<B><FONT SIZE=4><P>Formatted Text</P>
</B></FONT>
<P>This is an example of formatted text, it includes:</P>
<FONT SIZE=2>
</FONT><I><P>Italics,</I> <B>Bold</B> and <U>Underlining</P>
</U>
<FONT FACE="French Script MT"><P>Different Fonts</FONT> and <FONT
SIZE=4>Font Sizes</P>
```

(b)

Formatted text

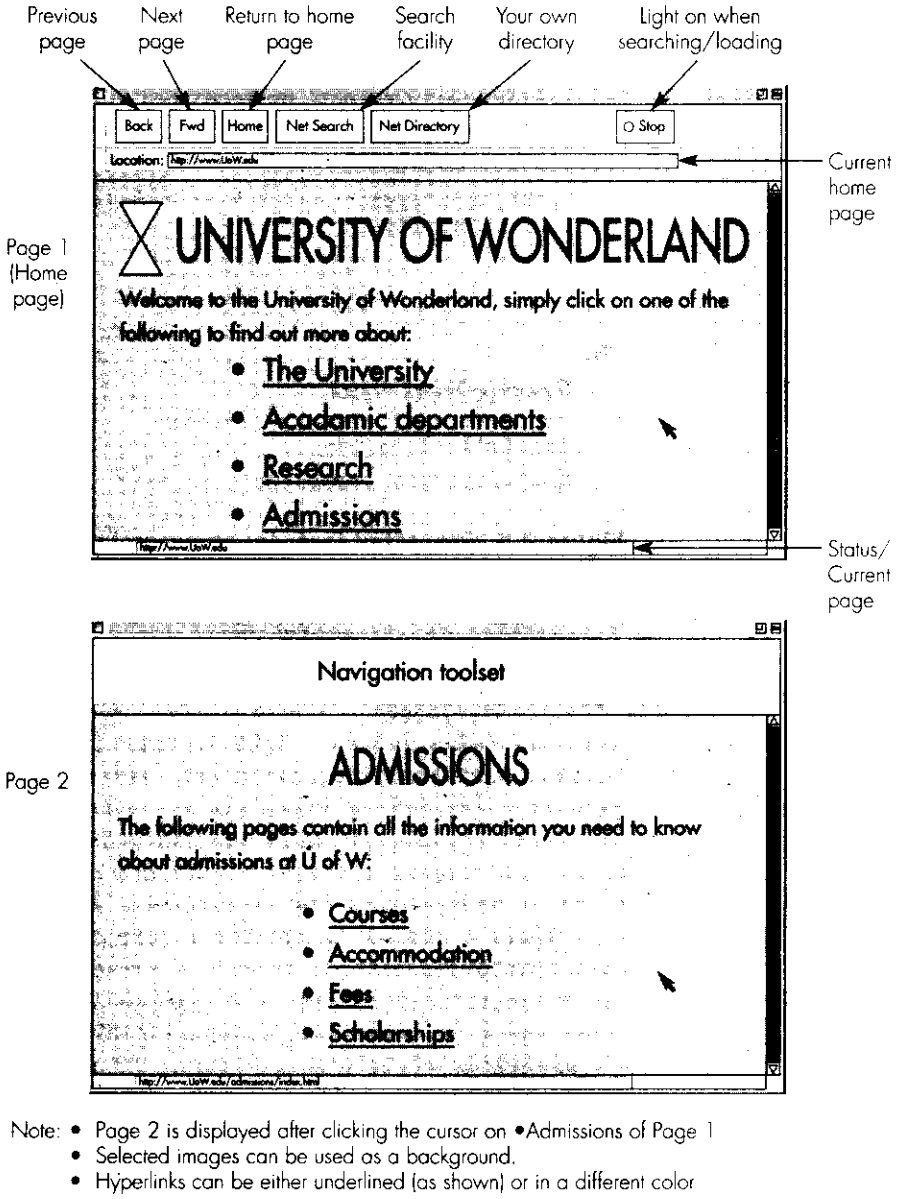
This is an example of formatted text, it includes:
Italics, **Bold** and Underlining
Different fonts and Font Sizes

Figure 2.8 Formatted text: (a) an example formatted text string; (b) printed version of the string.

2.3.3 Hypertext

As we saw earlier in Section 1.4.2 hypertext is a type of formatted text that enables a related set of documents – normally referred to as **pages** – to be created which have defined linkage points – referred to as **hyperlinks** – between each other. For example, most universities describe their structure and the courses and support services they offer, in a booklet known as a prospectus. Like most such booklets, this is organized in a hierarchical way and, in order for a reader to find out information about a particular course, and the facilities offered by the university, typically, the reader would start at the index and use this to access details about the various departments, the courses each offers, and so on, by switching between the different sections of the booklet.

In a similar way, as we show in Figure 2.9, hypertext can be used to create an electronic version of such documents with the index, descriptions of departments, courses on offer, library, and other facilities all written in hypertext as pages with various defined hyperlinks between them to enable a person to browse through its contents in a user-friendly way. Typically, the linked set of pages that make up the prospectus would all be stored in a single server computer. However, should a particular department choose to provide a more in-depth description of the courses and facilities it offers – for example, the contents of courses, current research projects, staff profiles, or publications – these can also be implemented as a linked set of pages on a different computer and, providing all the computers at the site are connected to the same network (and use the same set of communication protocols), additional hyperlinks between the two sets of pages can be introduced.



Note: • Page 2 is displayed after clicking the cursor on •Admissions of Page 1
• Selected images can be used as a background.
• Hyperlinks can be either underlined (as shown) or in a different color

Figure 2.9 Example of an electronic document written in hypertext.

The linked set of pages that are stored in a particular server are accessed and viewed using a client program known as a **browser**. This can be running in either the same computer on which the server software is running or, more usually, in a separate remote computer.

Associated with each set of linked pages is a page known as the **home page**. Normally, this comprises a form of index to the set of pages linked to it, each of which has a hyperlink entry-point associated with it. Typically, hyperlinks take the form of an underlined text string and the user initiates the access and display of a particular page by pointing and clicking the mouse on the appropriate string/link.

Associated with each link, in addition to the textual name of the link and the related format-control information for its display, is a unique network-wide name known as a **uniform resource locator** or **URL**. This comprises a number of logical parts – including the unique name of the host computer where the page is stored and also the name of the file containing the page – which collectively enables the browser program to locate and read each requested page. Hence to access the home page of a particular server, the user first enters its URL – in response to a prompt by the browser program – and, in turn, the browser uses this, first to locate the server computer on which the particular page is stored and then to request the page contents from the server. The page contents are stored in a specific formatted text and, on receipt of the contents, the browser displays these on the client computer screen using the included format control commands. In this way, after accessing the home page associated with a site, a user is able to access and browse through the contents of the linked set of pages in the order he or she chooses.

An example of a hypertext language is **HTML** which stands for **HyperText Markup Language**. We shall discuss HTML in detail in section 15.3 and restrict oursleves here to its essential features in order to gain an understanding of the form of representation of a typical hypertext page. HTML is an example of a more general set of what are known as **mark-up languages**. These are used to describe how the contents of a document are to be presented on a printer or a display, the term "mark-up" being that used by a copy editor when the printing of documents was carried out manually. Other mark-up languages include **Postscript** (known as a (printed) page description language), **SGML** (the acronym for **Standard Generalized Mark-up Language** on which HTML is based), **Tex**, and **Latex**. In general, the output of these languages is similar to that produced by many word-processing systems but, unlike word processors, they are concerned only with the formatting of a document in preparation for its printing or display.

As the name implies, HTML is concerned solely with hypertext and has been designed specifically for use with the World Wide Web and, in particular, for the creation of Web pages. It is concerned primarily with the formatting of pages – to enable a browser program running on a remote computer to display a retrieved page on its local screen – and for the specification of hyperlinks – to enable a user to browse interactively through the contents of a set of pages linked together by means of hyperlinks.

The page formatting commands – known as **directives** in HTML and each sandwiched between a pair of **tags** (<>) – include commands to start a new paragraph (<P>), start and end boldface (<B>text</B>), present in the

form of a bulleted list (<HL>list</HL>), include an image (<IMG SRC=" an image">), and so on. Other media types such as sound and video clips can also be included, giving rise to the term **hypermedia**. Indeed, the terms "hypermedia" and "hypertext" are often used interchangeably when referring to pages created in HTML.

The specification of a hyperlink is made by specifying both the URL of where the required page is located, together with the textual name of the link. For example, the specification of a hyperlink to a page containing 'Further details' would have the form <A HREF="URL">Further details </A>. Hence, as we can see, a hypertext string is similar to a formatted text string, except that at the linkage points within a page additional text strings are found that define the URL of the linked page.

2.4 Images

Within the context of this book, images include computer-generated images – more generally referred to as **computer graphics** or simply **graphics** – and digitized images of both documents and pictures. Although ultimately all three types of image are displayed (and printed) in the form of a two-dimensional matrix of individual picture elements – known as **pixels** or sometimes **pels** – each type is represented differently within the computer memory or, more generally, in a computer file. Also, each type of image is created differently and hence it is helpful for us to consider each separately.

2.4.1 Graphics

There is a range of software packages and programs available for the creation of computer graphics. These provide easy-to-use tools to create graphics that are composed of all kinds of visual objects including lines, arcs, squares, rectangles, circles, ovals, diamonds, stars, and so on, as well as any form of hand-drawn (normally referred to as **freeform**) objects. Typically, these are produced by drawing the desired shape on the screen by means of a combination of a cursor symbol on the screen – a pencil or paint brush for example – and the mouse. Facilities are also provided to edit these objects – for example to change their shape, size, or color – and to introduce complete predrawn images, either previously created by the author of the graphic or selected from a gallery of images that come with the package. The latter is often referred to as **clip-art** and the better packages provide many hundreds of such images. Textual information can also be included in a graphic, together with precreated tables and graphs and digitized pictures and photographs which have been previously obtained. Objects can overlap each other with a selected object nearer to the front than another. In addition, you can add fill or add shadows to objects in order to give the complete image a three-dimensional (3-D) effect.

A computer's display screen can be considered as being made up of a two-dimensional matrix of individual picture elements – pixels – each of which can have a range of colors associated with it. For example, **VGA (video graphics array)** is a common type of display and, as we show in Figure 2.10(a), consists of a matrix of 640 horizontal pixels by 480 vertical pixels with, for example, 8 bits per pixel which allows each pixel to have one of 256 different colors.

All objects – including freeform objects – are made up of a series of lines that are connected to each other and, what may appear as a curved line, in practice is a series of very short lines each made up of a string of pixels which, in the limit, have the resolution of a pair of adjacent pixels on the screen. Some examples are shown in Figure 2.10(b)

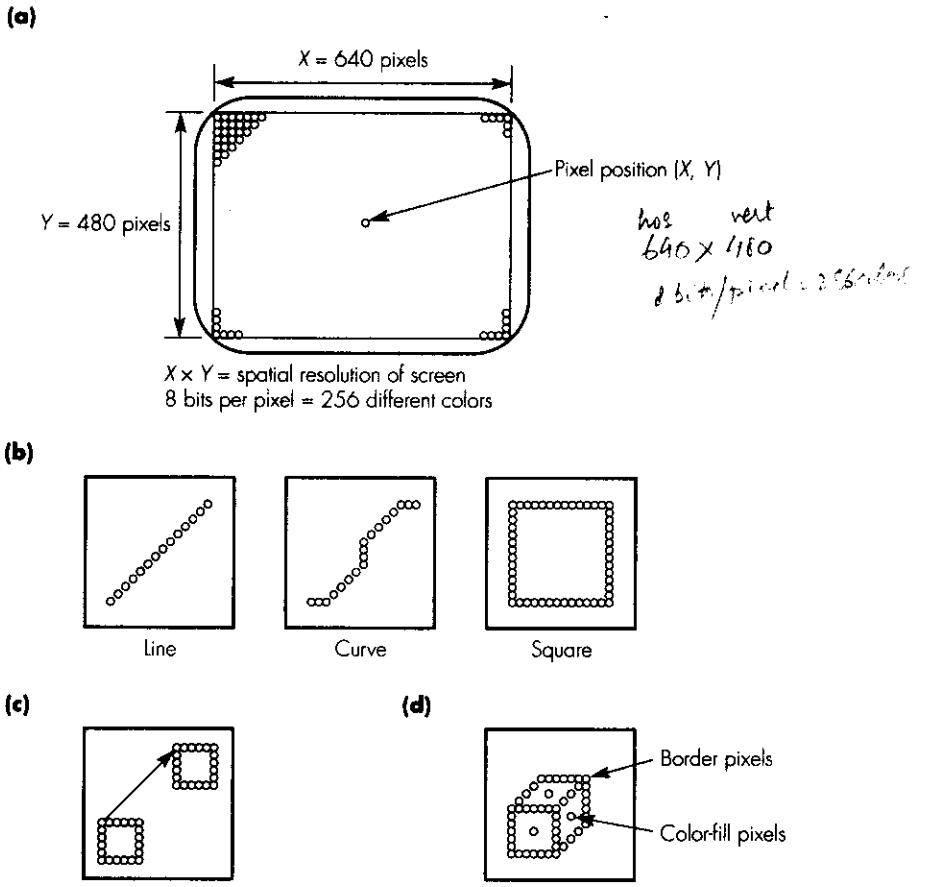


Figure 2.10 Graphics principles: (a) example screen format; (b) some simple object examples; (c) effect of changing position attribute; (d) solid objects.

Each object has a number of **attributes** associated with it. These include its shape – a line, a circle, a square, and so on – its size in terms of the pixel positions of its border coordinates, the color of the border, its shadow, and so on. In this way, editing an object involves simply changing selected attributes associated with the object. For example, as we show in Figure 2.10(c), we can move a square to a different location on the screen by simply changing its border coordinates and leaving the remaining attributes unchanged.

An object shape is said to be either open or closed. In the case of an open object, the start of the first line and the end of the last line that make up the object's border are not connected – that is, they do not start and end on the same pixel – whilst with a closed object they are connected. In the case of a closed object, the pixels enclosed by its border can all be assigned the same color – known as the **color-fill** – to create solid objects as shown in Figure 2.10(d). This operation is also known as **rendering**. In this way, all objects are drawn on the screen by the user simply specifying the name of the object and its attributes – including its color-fill and shadow effect if required – and a set of more basic lower-level commands are then used to determine both the pixel locations that are involved and the color that should be assigned to each pixel.

As we can deduce from the above, the representation of a complete graphic is analogous to the structure of a program written in a high-level programming language. For instance, a program consists of a main body together with a number of procedures/functions, each of which has a set of parameters associated with it and performs a specific function. In the same way, a graphic consists of the set of commands (each with attributes) that are necessary to draw the different objects that make up the graphic. Also, in the same way that the procedures/functions in a program can be a mix of those created by the writer of the program and those available as library procedures/functions, so the objects associated with a graphic can be either those created by the author or those selected from the set of standard objects or the clip-art gallery. And in the same way that a procedure/function in a program may, in turn, call a number of lower-level functions, so the commands associated with objects use the lower-level commands to display the objects on the screen. Finally, in the same way that the main body of a program is concerned with invoking the various procedures/functions in the order necessary to implement a particular computational task, so the main body of a graphic representation is concerned with invoking the different object commands in the correct sequence to create the desired graphic taking into account any overlapping objects.

We can conclude that there are two forms of representation of a computer graphic: a high-level version (similar to the source code of a high-level program) and the actual pixel-image of the graphic (similar to the byte-string corresponding to the low-level machine code of the program and known more generally as the **bit-map format**). A graphic can be transferred over a network in either form. In general, however, the high-level language form is

much more compact and requires less memory to store the image and less bandwidth for its transmission. In order to use the high-level language form, however, the destination must of course be able to interpret the various high-level commands. So instead the bit-map form is often used and, to help with this, there are a number of standardized forms of representation such as **GIF** (**graphical interchange format**) and **TIFF** (**tagged image file format**). There are also software packages such as **SRGP** (**simple raster graphics package**) which convert the high-level language form into a pixel-image form. We shall discuss a selection of these in the next chapter.

2.4.2 Digitized documents

An example of a digitized document is that produced by the scanner associated with a **facsimile** (**fax**) **machine**, the principles of which are shown in Figure 2.11.

The scanner associated with a fax machine operates by scanning each complete page from left to right to produce a sequence of scan lines that start

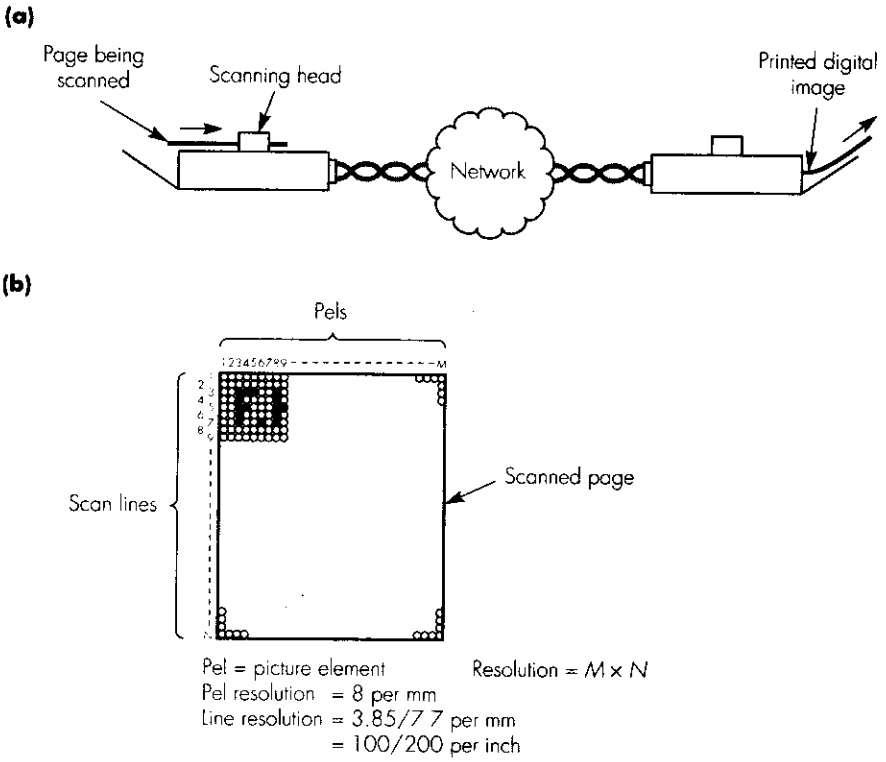


Figure 2.11 Facsimile machine principles: (a) schematic; (b) digitization format.

at the top of the page and end at the bottom. The vertical resolution of the scanning procedure is either 3.85 or 7.7 lines per millimeter which is equivalent to approximately 100 or 200 lines per inch. As each line is scanned, the output of the scanner is digitized to a resolution of approximately 8 picture elements – known as **pels** with fax machines – per millimeter.

Fax machines use just a single binary digit to represent each pel, a 0 for a white pel and a 1 for a black pel. Hence the digital representation of a scanned page is as shown in Figure 2.11(b) which, for a typical page, produces a stream of about two million bits. The printer part of a fax machine then reproduces the original image by printing out the received stream of bits to a similar resolution. In general, the use of a single binary digit per pel means that fax machines are best suited to scanning bitonal (black-and-white) images such as printed documents comprising mainly textual information.

2.4.3 Digitized pictures

In the case of scanners which are used for digitizing continuous-tone monochromatic images – such as a printed picture or scene – normally, more than a single bit is used to digitize each picture element. For example, good quality black-and-white pictures can be obtained by using 8 bits per picture element. This yields 256 different levels of gray per element – varying between white and black – which gives a substantially improved picture quality over a facsimile image when reproduced. In the case of color images, in order to understand the digitization format used, it is necessary first to obtain an understanding of the principles of how color is produced and how the picture tubes used in computer monitors (on which the images are eventually displayed) operate.

Color principles

It has been known for many years that the human eye sees just a single color when a particular set of three primary colors are mixed and displayed simultaneously. In fact, a whole spectrum of colors – known as a **color gamut** – can be produced by using different proportions of the three primary colors red (R), green (G), and blue (B). This principle is shown in Figure 2.12 together with some examples of colors that can be produced.

The mixing technique used in part (a) is known as **additive color mixing** which, since black is produced when all three primary colors are zero, is particularly useful for producing a color image on a black surface as is the case in display applications. It is also possible to perform the complementary **subtractive color mixing** operation to produce a similar range of colors. This is shown in part(b) of the figure and, as we can see, with subtractive mixing white is produced when the three chosen primary colors cyan (C), magenta (M), and yellow (Y) are all zero. Hence this choice of colors is particularly useful for producing a color image on a white surface as is the case in printing applications.

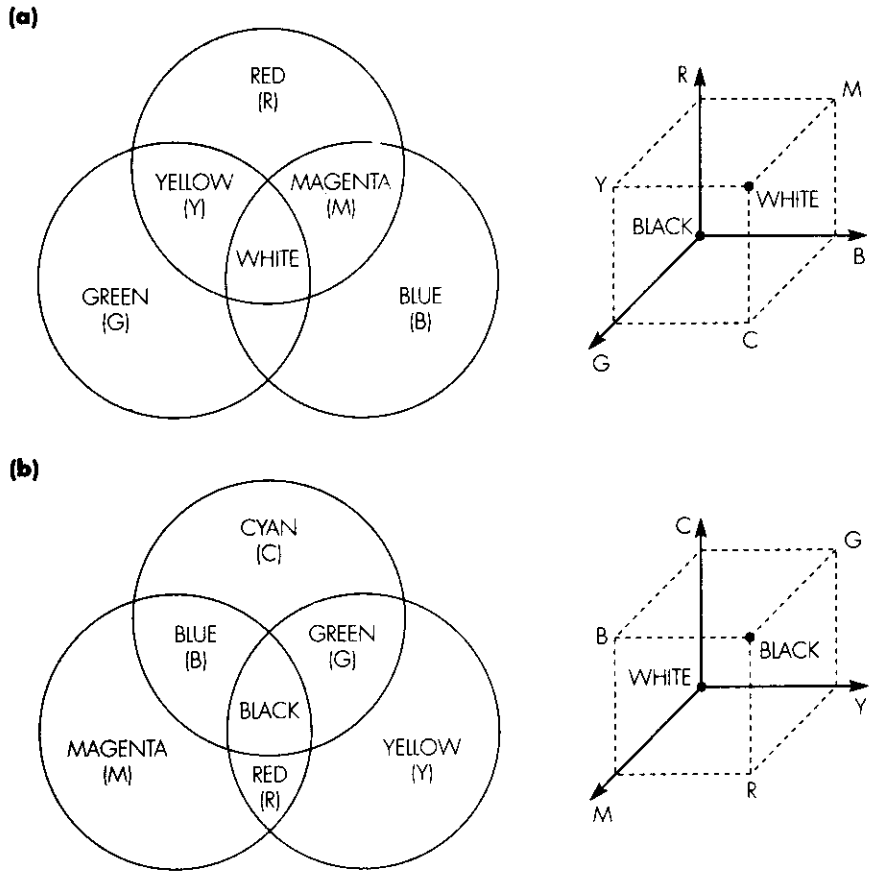


Figure 2.12 Color derivation principles: (a) additive color mixing; (b) subtractive color mixing.

The same principle is used in the picture tubes associated with color television sets with the three primary colors R, G, and B. Also, in most computer monitors since, in general, those used with personal computers use the same picture tubes as are used in television sets. Hence, in order to be compatible with the computer monitors on which digital pictures are normally viewed, the digitization process used yields a color image that can be directly displayed on the screen of either a television set or a computer monitor. The general principles associated with the process are shown in Figure 2.13.

Raster-scan principles

The picture tubes used in most television sets operate using what is known as a **raster-scan**. This involves a finely-focussed electron beam – the raster – being scanned over the complete screen. Each complete scan comprises a number of discrete horizontal lines the first of which starts at the top left

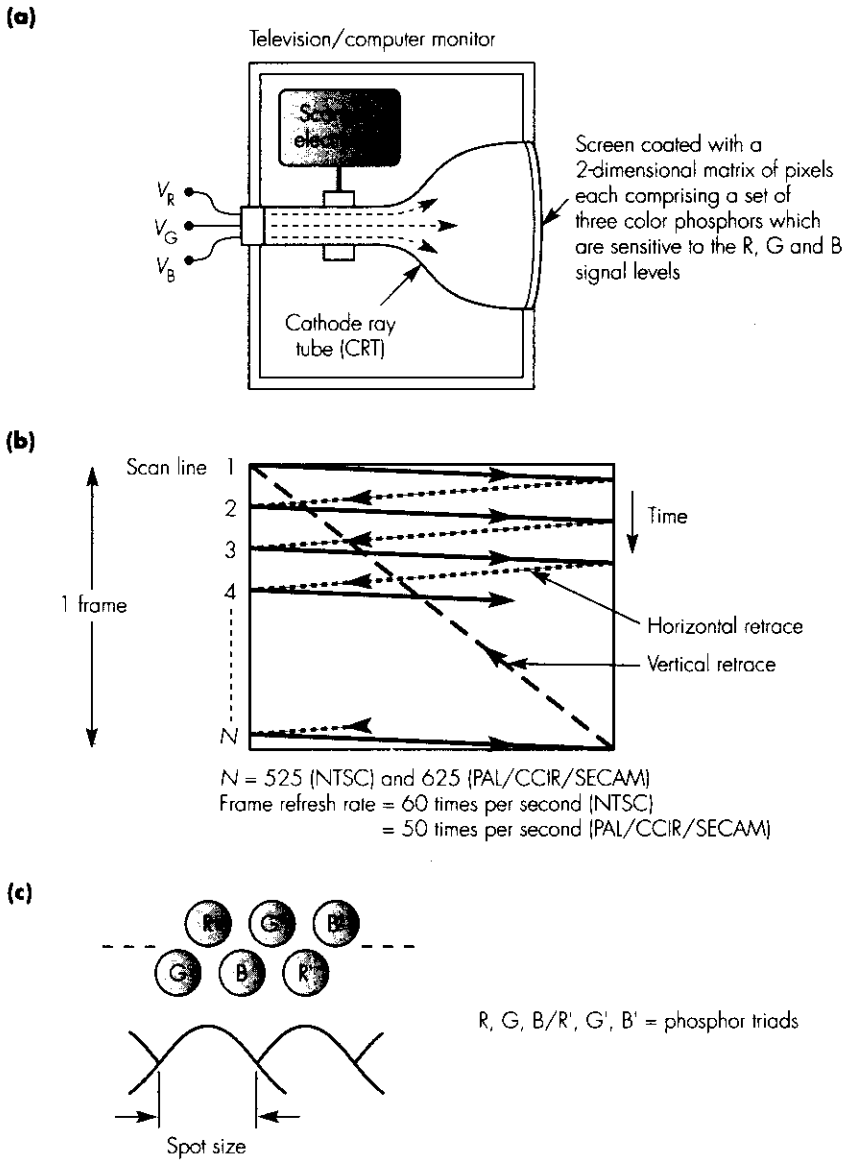


Figure 2.13 Television/computer monitor principles: (a) schematic; (b) raster-scan principles; (c) pixel format on each scan line.

corner of the screen and the last of which ends at the bottom right corner. At this point the beam is deflected back again to the top left corner and the scanning operation repeats in the same way. This type of scanning is called **progressive scanning** and is shown in diagrammatic form in Figure 2.13(b).

Each complete set of horizontal scan lines is called a **frame** and, as we can see, each frame is made up of N individual scan lines where N is either 525 (North and South America and most of Asia) or 625 (Europe and a number of other countries). The inside of the display screen of the picture tube is coated with a light-sensitive phosphor that emits light when energized by the electron beam. The amount of light emitted – its brightness – is determined by the power in the electron beam at that instant. During each horizontal (line) and vertical (frame) retrace period the electron beam is turned off and, to create an image on the screen, the level of power in the beam is changed as each line is scanned.

In the case of black-and-white picture tubes just a single electron beam is used with a white-sensitive phosphor. Color tubes use three separate, closely-located beams and a two-dimensional matrix of pixels. Each pixel comprises a set of three related color-sensitive phosphors, one each for the red, green, and blue signals. The set of three phosphors associated with each pixel is called a **phosphor triad** and a typical arrangement of the triads on each scan line is as shown in Figure 2.13(c). As we can deduce from this, although in theory each pixel represents an idealized rectangular area which is independent of its neighboring pixels, in practice each pixel has the shape of a **spot** which merges with its neighbors. A typical spot size is 0.025 inches (0.635 mm) and, when viewed from a sufficient distance, a continuous color image is seen.

Television picture tubes were designed, of course, to display moving images. The persistence of the light/color produced by the phosphor, therefore, is designed to decay very quickly and hence it is necessary to continuously **refresh** the screen. In the case of a moving image, the light signals associated with each frame change to reflect the motion that has taken place during the time required to scan the preceding frame, while for a static/still image, the same set of light signals are used for each frame. The **frame refresh rate** must be high enough to ensure the eye is not aware the display is continuously being refreshed. A low refresh rate leads to what is called **flicker** which is caused by the previous image fading from the eye retina before the following image is displayed. To avoid this, a refresh rate of at least 50 times per second is required. In practice, the frame refresh rate used is determined by the frequency of the mains electricity supply which is either 60 Hz in North and South America and most of Asia and 50 Hz in Europe and a number of other countries.

Most current picture tubes operate in an analog mode, that is, the amplitude of each of the three color signals is continuously varying as each line is scanned. In the case of digital television – and digitized pictures stored within the memory of a computer – the color signals are in a digital form and comprise a string of pixels with a fixed number of pixels per scan line. Hence in order to display a stored image, the pixels that make up each line are read from memory in time-synchronism with the scanning process and converted into a continuously varying analog form by means of a digital-to-analog converter.

Since in practice the area of the computer memory that holds the string of pixels that make up the image – the pixel image – must be accessed continuously as each line is scanned, normally a separate block of memory known as the **video RAM** – RAM being the acronym for **random access memory** – is used to store the pixel image. In this way, the graphics program needs only to write into the video RAM whenever either selected pixels or the total image changes. An example architecture showing the various steps involved is given in Figure 2.14

Typically, the graphics program is used to create the high-level version of the image interactively (using either the keyboard or a mouse) and the **display controller** part of the program interprets sequences of display commands and converts them into displayed objects by writing the appropriate pixel values into the video RAM. The latter is also known, therefore, as the **frame/display/refresh buffer**. Normally the **video controller** is a hardware subsystem that reads the pixel values stored in the video RAM in time-synchronism with the scanning process and, for each set of pixel values, converts these into the equivalent set of red, green, and blue analog signals for output to the display.

Pixel depth

The number of bits per pixel is known as the **pixel depth** and determines the range of different colors that can be produced. Examples are 12 bits – 4 bits per primary color yielding 4096 different colors – and 24 bits – 8 bits per

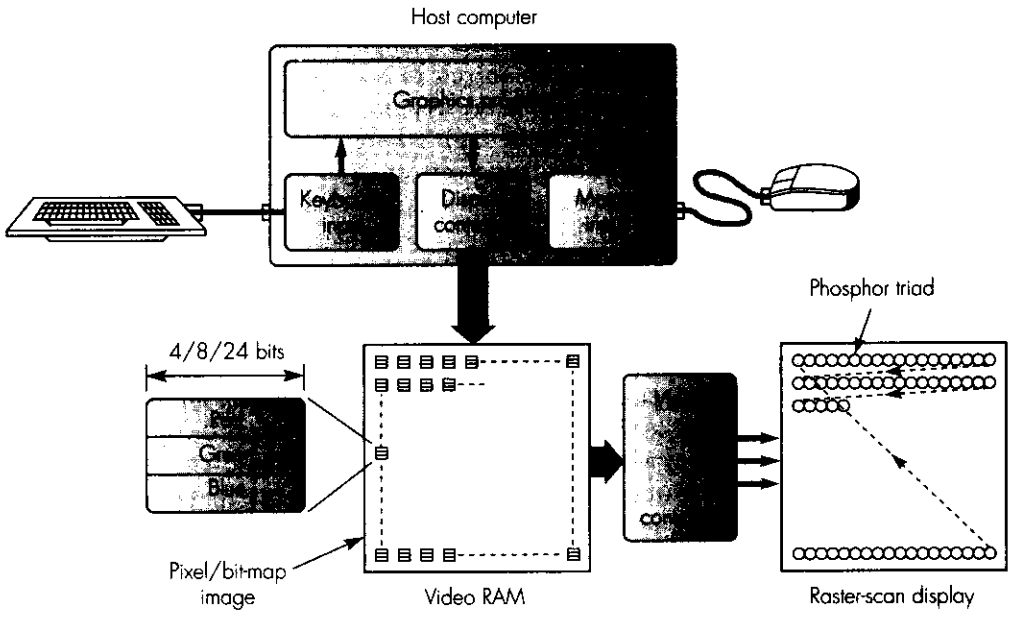


Figure 2.14 Raster-scan display architecture.

primary color yielding in excess of 16 million (2^{24}) colors. In practice, however, the eye cannot discriminate between such a range of colors and so in some instances a selected subset of this range is used. The selected colors in the subset are then stored in a table and each pixel value is used as an address to a location within the table which contains the corresponding three color values. The table is known as the **color look-up table** or **CLUT**. For example, if each pixel is 8 bits and the CLUT contains 24 bit entries, this will provide a subset of 256 (2^8) different colors selected from a palette of 16 million (2^{24}) colors. In this way, the amount of memory required to store an image can be reduced significantly.

Aspect ratio

Both the number of pixels per scanned line and the number of lines per frame vary, the actual numbers used being determined by what is known as the **aspect ratio** of the display screen. This is the ratio of the screen width to the screen height. The aspect ratio of current television tubes is 4/3 with older tubes – on which PC monitors are based – and 16/9 with the wide-screen television tubes.

In the United States, the standard for color television has been defined by the **National Television Standards Committee** (**NTSC**) while in Europe three color standards exist **PAL** (UK), **CCIR** (Germany), and **SECAM** (France). As we indicated earlier, the NTSC standard uses 525 scan lines per frame while the three European standards all use 625 scan lines. In neither case, however, are all lines displayed on the screen since some are used to carry control and other information. In practice, therefore, the number of visible lines per frame – which is equal to the vertical resolution in terms of pixels – is 480 with an NTSC monitor and 576 with the other three standards. Thus in order to avoid distortion on a screen which has a 4/3 aspect ratio – for example when displaying a square of, say, ($N \times N$) pixels – it is necessary to have 640 pixels (480 × 4/3) per line with an NTSC monitor and 768 (576 × 4/3) pixels per line with a European monitor. This produces a lattice structure that is said to produce **square pixels** and is shown in diagrammatic form in Figure 2.15. Some example screen resolutions associated with the more common computer monitors based on television picture tubes – together with the amount of memory required to store the corresponding image – are shown in Table 2.1

As we can deduce from the table, the memory requirements to store a single digital image can be high and vary between 307.2 kbytes for an image displayed on a **VGA** (**video graphics array**) screen with 8 bits per pixel through to approximately 2.36 Mbytes for a **SVGA** (**Super VGA**) screen with 24 bits per pixel. It should be noted that the more expensive computer monitors are not based on television picture tubes and hence are not constrained by the 4/3 aspect ratio. An example is 1280 × 1024 × 24 which may have a refresh rate as high as 75 frames per second to produce a sharper image.

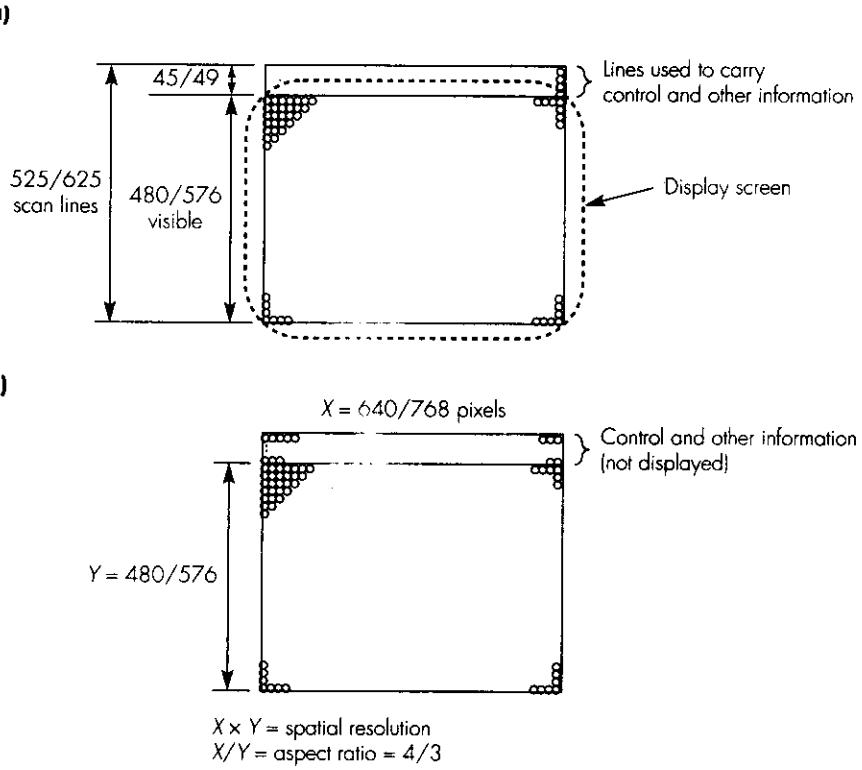


Figure 2.15 Screen resolutions: (a) visible lines per frame; (b) digitization spatial resolution.

Table 2.1 Example display resolutions and memory requirements.

Standard	Resolution	Number of colors	Memory required per frame (bytes)
VGA	640 × 480 × 8	256	307.2 kB
XGA	640 × 480 × 16	64K	614.4 kB
	1024 × 768 × 8	256	786.432 kB
SVGA	800 × 600 × 16	64K	960 kB
	1024 × 768 × 8	256	786.432 kB
	1024 × 768 × 24	16M	2359.296 kB

Example 2.3

Derive the time to transmit the following digitized images at both 64 kbps and 1.5 Mbps:

a 640 × 480 × 8 VGA-compatible image,

a 1024 × 768 × 24 SVGA-compatible image.

Solution:

The number of bits in the VGA image is:

$$640 \times 480 \times 8 = 2.4576 \text{ Mbits}$$

The number of bits in the SVGA image is:

$$1024 \times 768 \times 24 = 18.874368 \text{ Mbits}$$

The time to transmit each image is:

$$\text{At 64 kbps: VGA} = \frac{2.4576 \times 10^6}{64 \times 10^3} = 38.4 \text{ s}$$

$$\text{SVGA} = \frac{18.874368 \times 10^6}{64 \times 10^3} = 294.912 \text{ s}$$

$$\text{At 1.5 Mbps: VGA} = \frac{2.4576 \times 10^6}{1.5 \times 10^6} = 1.6384 \text{ s}$$

$$\text{SVGA} = \frac{18.874368 \times 10^6}{1.5 \times 10^6} = 12.5829 \text{ s}$$

As we can see, the times to transmit a signal image at 64 kbps are such that interactive access would not be feasible, nor at 1.5 Mbps with the higher-resolution SVGA image.

Digital cameras and scanners

A typical arrangement that is used to capture and store a digital image produced by a scanner or a digital camera – either a still-image camera or a video camera – is shown in Figure 2.16(a). In the figure it is assumed that the captured image is transferred to the computer directly as it is produced. Alternatively, in the case of digital cameras, a set of digitized images can be stored within the camera itself and then downloaded into the computer at a later time.

An image is captured within the camera/scanner using a solid-state device called an **image sensor**. This is a silicon chip which, in digital cameras, consists of a two-dimensional grid of light-sensitive cells called **photosites**. When the camera shutter is activated, each photosite stores the level of intensity of the light that falls on it. A widely-used image sensor is a **charge-coupled**

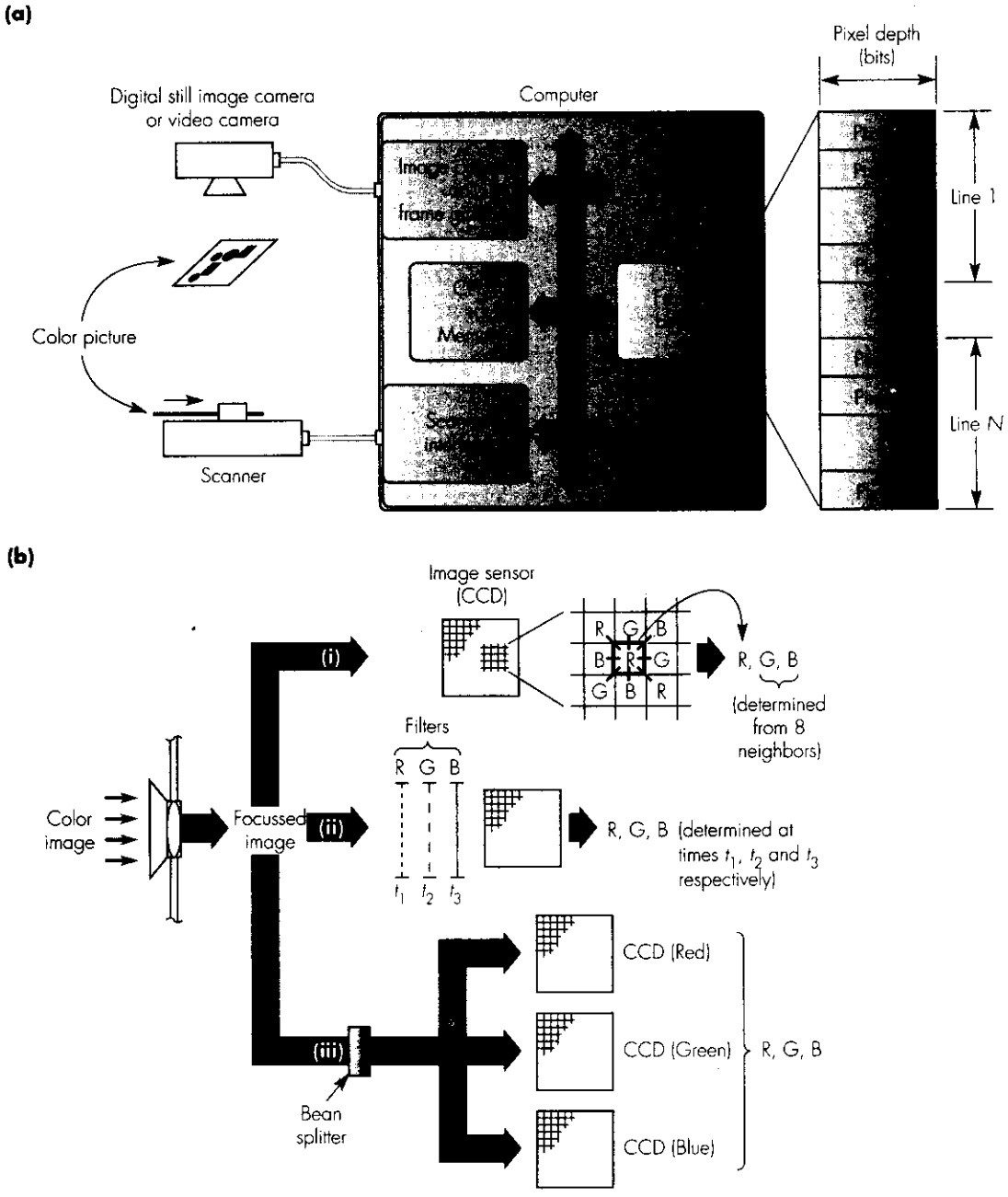


Figure 2.16 Color image capture: (a) schematic; (b) RGB signal generation alternatives.

device (CCD). This comprises an array of photosites on its surface and operates by converting the level of light intensity that falls on each photosite into an equivalent electrical charge. The level of charge – and hence light intensity – stored at each photosite position is then read out and converted into a digital value using an ADC. A similar technique is used in scanners except the image sensor comprises just a single row of photosites. These are exposed in time-sequence with the scanning operation and each row of stored charges are read out and digitized before the next scan occurs.

For color images, the color associated with each photosite – and hence pixel position – is obtained in a number of ways. These include the three methods shown in Figure 2.16(b).

(i) In this method, the surface of each photosite is coated with either a red, green, or blue filter so that its charge is determined only by the level of red, green, or blue light that falls on it. The coatings are arranged in a 3×3 grid structure as shown in the figure. The color associated with each photosite/pixel is then determined by the output of the photosite – R, G, or B – together with each of its 8 immediate neighbors. The levels of the two other colors in each pixel are then estimated by an interpolation procedure involving all nine values. This method is used with most consumer-quality cameras.

(ii) This method involves the use of three separate exposures of a single image sensor, the first through a red filter, the second a green filter, and the third a blue filter. The color associated with each pixel position is then determined by the charge obtained with each of the three filters – R, G, and B. Since three separate exposures are required for each image, this approach cannot be used with video cameras. It is used primarily with high-resolution still-image cameras in locations such as photographic studios where the camera can be attached to a tripod.

(iii) This method uses three separate image sensors, one with all the photosites coated with a red filter, the second coated with a green filter, and the third coated with a blue filter. A single exposure is used with the incoming light split into three beams each of which exposes a separate image sensor. This method is used in professional-quality high-resolution still and moving image cameras since, in general, they are more costly owing to the use of three separate sensors and associated signal processing circuits.

Once each image/frame has been captured and stored on the image sensor(s), the charge stored at each photosite location is read and digitized. Using a CCD, the set of charges on the matrix of photosites are read a single row at a time. First the set of charges on the first row of photosites are transferred to what is called the **readout register**. Each of the photosites in a row is coupled to the corresponding photosites in the two adjoining rows and, as each row is transferred to the readout register, the set of charges on each of

the other rows in the matrix move down to the next row of photosite positions. Once in the readout register, the charge on each photosite position is shifted out, amplified and digitized using an ADC. This procedure then repeats until the set of charges on all rows have been read out and digitized.

For a low-resolution image of 640 × 480 pixels and a pixel depth of 24 bits – 8 bits each for the R, G, and B signals – the amount of memory required to store each image is 921 600 bytes. If this is output directly to the computer, then the bit-map can be loaded straight into the frame buffer ready to be displayed. If it is to be stored within the camera, however, then multiple images of this size need to be stored prior to them being output to a computer. The set of images are stored in an integrated circuit memory that is either on a removable card or fixed within the camera. In the first case, the card is simply removed and inserted into the PCMCIA slot of a computer and in the second case the contents of the memory are downloaded to the computer by means of a cable link. Once within the computer, software can be used to insert the digital image(s) into a document, send it by email, and so on. Alternatively, photo-editing software can be used to manipulate a stored image; for example, to change its size or color

There are a number of file formats used to store a set of images. One of the most popular is a version of the tagged image file format (TIFF) called **TIFF for electronic photography (TIFF/EP)**. This allows many different types of image data to be stored in the image file including data (such as the date and time and various camera settings) associated with each image.

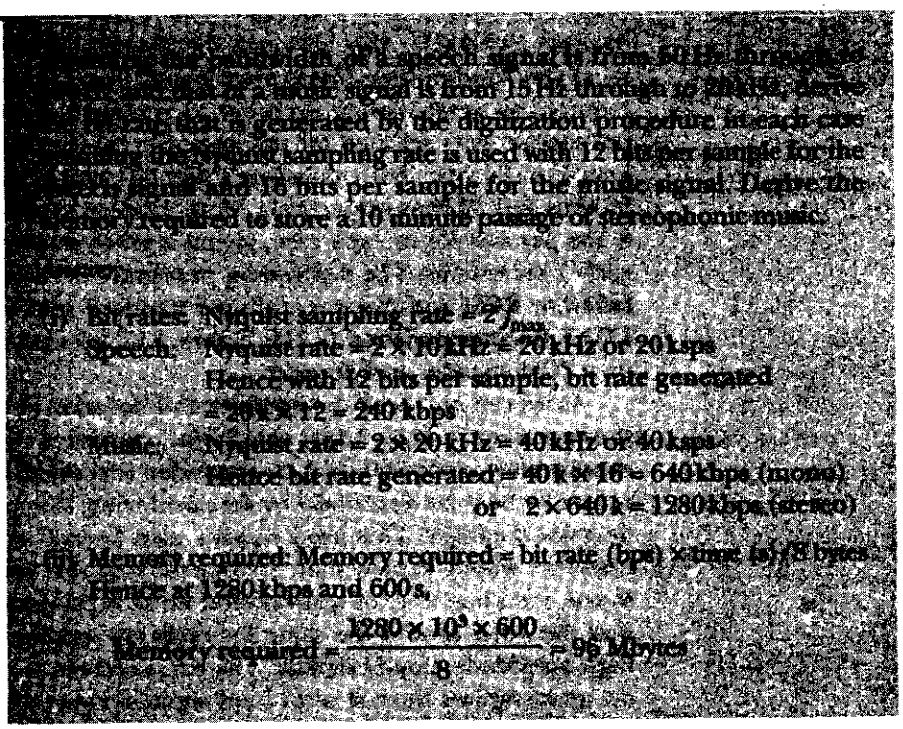
2.5 Audio

Essentially, we are concerned with two types of audio signal: speech signals as used in a variety of interpersonal applications including telephony and video telephony, and music-quality audio as used in applications such as CD-on-demand and broadcast television. In general, audio can be produced either naturally by means of a microphone or electronically using some form of synthesizer. In the case of a synthesizer, the audio is created in a digital form and hence can be readily stored within the computer memory. A microphone, however, generates a time-varying analog signal and in order to store such signals in the memory of a computer, and to transmit them over a digital network, they must first be converted into a digital form using an audio signal encoder. Also, since loudspeakers operate using an analog signal, on output of all digitized audio signals the stream of digitized values must be converted back again into its analog form using an audio signal decoder.

We discussed the general principles behind the design of a signal encoder and decoder earlier in the chapter in Section 2.2 and here we will simply apply these principles to explain the digitization of both speech and music produced by a microphone. We shall then discuss the format of synthesized audio in a separate section.

The bandwidth of a typical speech signal is from 50 Hz through to 10 kHz and that of a music signal from 15 Hz through to 20 kHz. Hence the sampling rate used for the two signals must be in excess of their Nyquist rate which is 20 ksps (2 × 10 kHz) for speech and 40 ksps (2 × 20 kHz) for music. The number of bits per sample must be chosen so that the quantization noise generated by the sampling process is at an acceptable level relative to the minimum signal level. In the case of speech, assuming linear (equal) quantization intervals, tests have shown that this dictates the use of a minimum of 12 bits per sample and for music 16 bits. In addition, since in most applications involving music stereophonic (stereo) sound is utilized (and hence two such signals must be digitized) this results in a bit rate double that of a monaural (mono) signal.

Example 2.4



In practice, both the sampling rate used and the number of bits per sample are often less than these values. In the case of speech, for example, the bandwidth of the network used in many interpersonal applications is often much less than the bandwidth of the source signal thus dictating a lower sampling rate with fewer bits per sample. Similarly, with music, the sampling rate is often lowered in order to reduce the amount of memory that is required to store a particular passage of music. A practical example of the digitization parameters used in each case is now presented.

2.5.1 PCM speech

As we described in Chapter 1, most interpersonal applications involving speech use for communication purposes a public switched telephone network (PSTN). Because this has been in existence for many years the operating parameters associated with it were defined some time ago. Initially, a PSTN operated with analog signals throughout, the source speech signal being transmitted and switched (routed) unchanged in its original analog form. Progressively, however, the older analog transmission circuits were replaced by digital circuits. This was carried out over a number of years and, because of the need to interwork between the earlier analog and newer digital equipments during the transition period, the design of the digital equipment was based on the operating parameters of the earlier analog network. The bandwidth of a speech circuit in this network was limited to 200 Hz through to 3.4 kHz. Also, although the Nyquist rate is 6.8 kHz, the poor quality of the bandlimiting filters used meant that a sampling rate of 8 kHz was required to avoid aliasing. In order to minimize the resulting bit rate, 7 bits per sample were selected for use in North America and Japan and 8 bits per sample in Europe (both including a sign bit) which, in turn, yields bit rates of 56 kbps and 64 kbps respectively. More modern systems have moved to using 8 bits per sample in each case, giving a much improved performance over early 7 bit systems. The digitization procedure is known as **pulse code modulation** or **PCM** and the international standard relating to this is defined in **ITU-T Recommendation G.711**. Figure 2.17 (a) shows the circuits that make up a PCM encoder and decoder.

As we can see, both the encoder and decoder, in addition to the circuits shown earlier in Figs. 2.2(a) and 2.5(a), consist of two additional circuits: a **compressor** (encoder) and an **expander** (decoder). The role of these circuits can best be described by reconsidering the quantization operation described earlier. This used equal – also known as linear – quantization intervals which means that, irrespective of the magnitude of the input signal, the same level of quantization noise is produced. The effect of this is that the noise level is the same for both low amplitude (quiet) signals and high amplitude (loud) signals. The ear, however, is more sensitive to noise on quiet signals than it is on loud signals. Hence to reduce the effect of quantization noise with just 8 bits per sample, in a PCM system the quantization intervals are made non-linear (unequal) with narrower intervals used for smaller amplitude signals than for larger signals. This is achieved by means of the compressor circuit and, at the destination, the reverse operation is performed by the expander circuit. The overall operation is known as **companding**. The input/output relationship of both circuits is shown in Figure 2.17(b) and (c) respectively; that shown in part (b) is known as the **compression characteristic** and that in part (c) the **expansion characteristic**. For clarity, just 5 bits per sample are used.

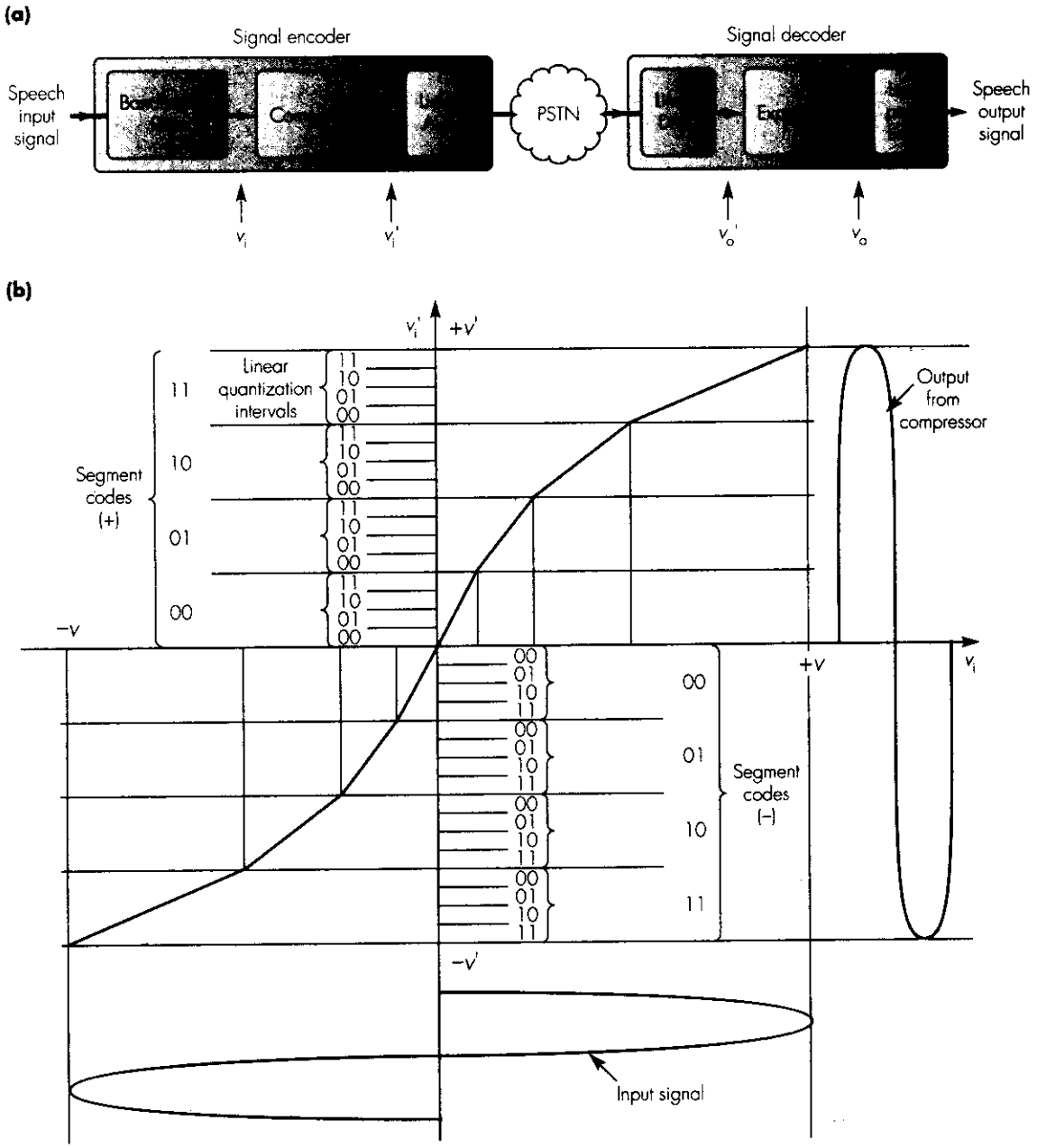
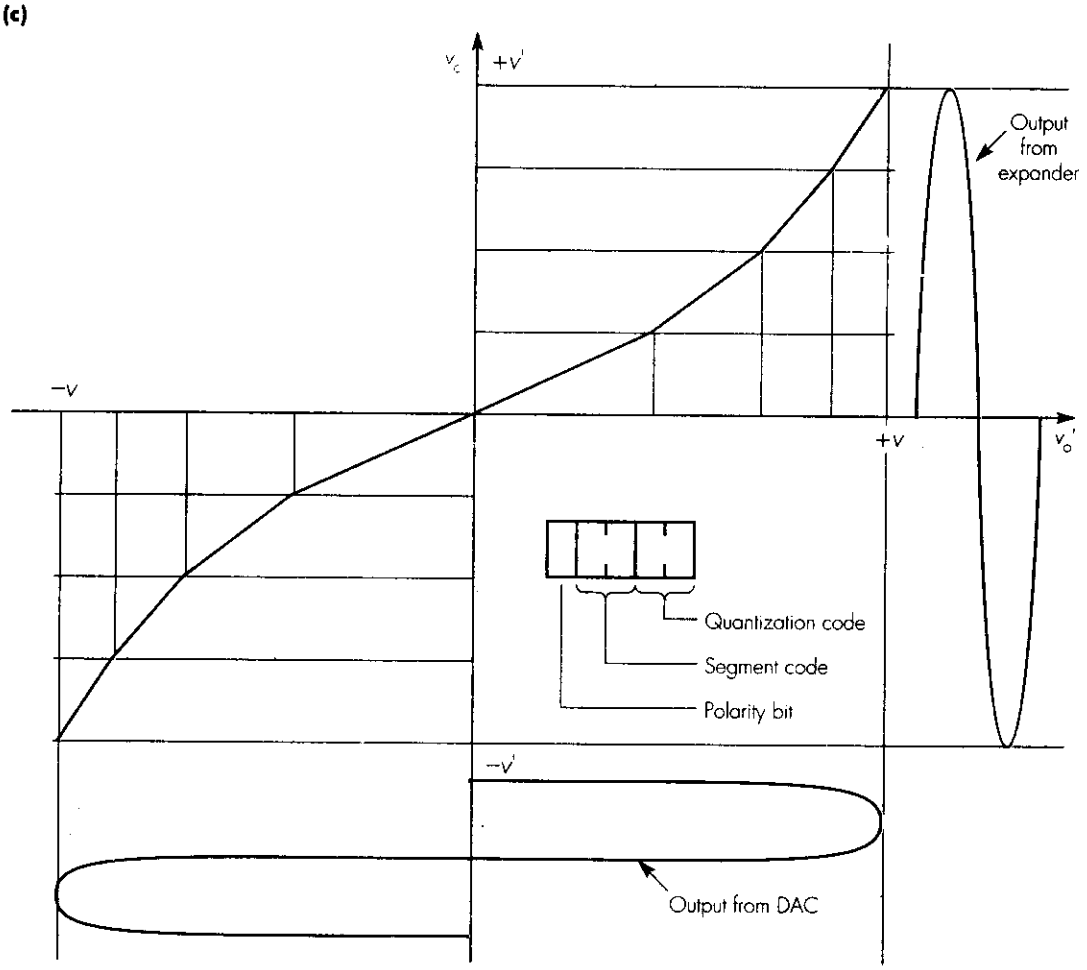


Figure 2.17 PCM principles: (a) signal encoding and decoding schematic; (b) compressor characteristic; (c) expander characteristic.

(c)



Note that in the G.711 standard a 3-bit segment code and 4-bit quantization code are used.

Figure 2.17 Continued

Prior to the input signal being sampled and converted into a digital form by the ADC, it is passed through the compressor circuit which effectively compresses the amplitude of the input signal. The level of compression – and hence the quantization intervals – increases as the amplitude of the input signal increases. The resulting compressed signal is then passed to the ADC which, in turn, performs a linear quantization on the compressed signal. Similarly, at the destination, each received codeword is first fed into a linear DAC. The analog output from the DAC is then passed to the expander circuit which performs the reverse operation of the compressor circuit. More modern systems perform both the compressor and expander operations digitally but the same principles apply.

In practice, for historical reasons, there are two different compression-expansion characteristics in use: **μ-law**, which is used in North America and Japan, and **A-law** which is used in Europe and some other countries. Hence, as we can deduce from Figure 2.17(a), it is necessary to carry out a conversion operation when communicating between the two systems. In both cases, however, the use of companding gives a perceived level of performance with 8 bits that is comparable with the performance obtained with 12 bits and uniform quantization intervals.

2.5.2 CD-quality audio

The discs used in CD players and CD-ROMs are digital storage devices for stereophonic music and more general multimedia information streams. There is a standard associated with these devices which is known as the **CD-digital audio** (**CD-DA**) standard. As indicated earlier, music has an audible bandwidth of from 15 Hz through to 20 kHz and hence the minimum sampling rate is 40 ksps. In the standard, however, the actual rate used is higher than this rate firstly, to allow for imperfections in the bandlimiting filter used and secondly, so that the resulting bit rate is then compatible with one of the higher transmission channel bit rates available with public networks.

One of the sampling rates used is 44.1 ksps which means the signal is sampled at 23 microsecond intervals. Since the bandwidth of a recording channel on a CD is large, a high number of bits per sample can be used. The standard defines 16 bits per sample which, as indicated earlier, tests have shown to be the minimum required with music to avoid the effect of quantization noise. With this number of bits, linear quantization can be used which yields 65 536 equal quantization intervals. The recording of stereophonic music requires two separate channels and hence the total bit rate required is double that for mono. Hence:

Bit rate per channel = sampling rate × bits per sample
$= 44.1 \times 10^3 \times 16 = 705.6$ kbps

and, Total bit rate $= 2 \times 705.6 = 1.411$ Mbps

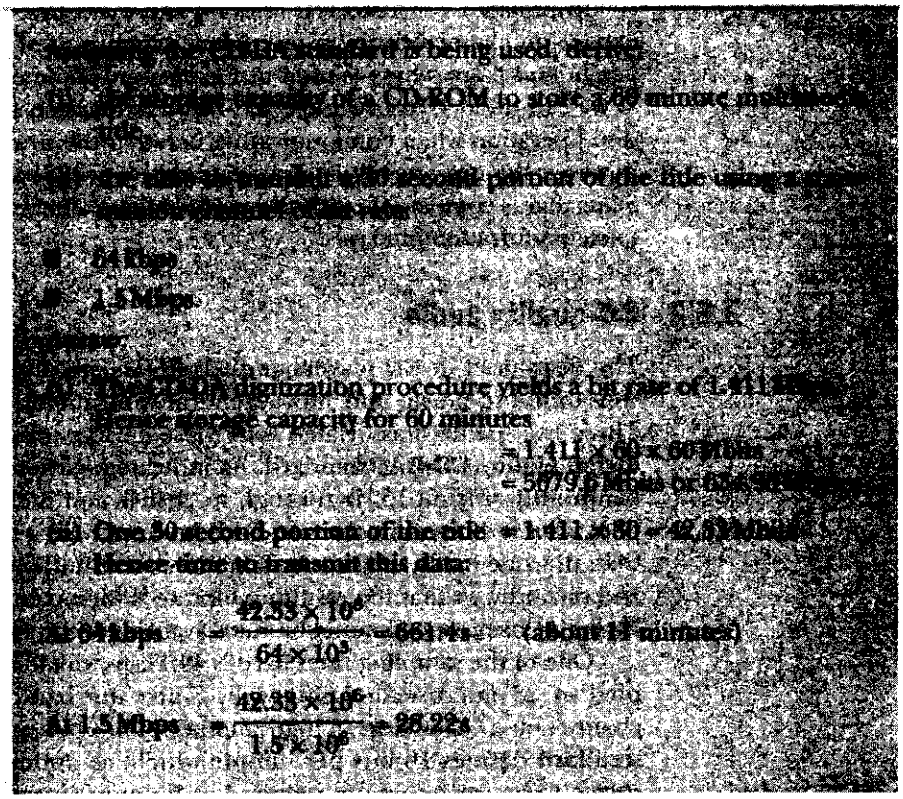
This is also the bit rate used with CD-ROMs which are widely used for the distribution of multimedia titles. Within a computer, however, in order to reduce the access delay, multiples of this rate are used.

As we can deduce from Example 2.5, it is not feasible to interactively access a 30s portion of a multimedia title over a 64 kbps channel. And with a 1.5 Mbps channel the time is still too high for interactive purposes.

2.5.3 Synthesized audio

Once digitized, any form of audio can be stored within a computer. However, as we can see from the results obtained in the next example, the amount of

Example 2.5



memory required to store a digitized audio waveform can be very large, even for relatively short passages. It is for this reason that synthesized audio is often used in multimedia applications since the amount of memory required can be between two and three orders of magnitude less than that required to store the equivalent digitized waveform version. In addition, it is much easier to edit synthesized audio and to mix several passages together. The main components that make up an audio synthesizer are shown in Figure 2.18.

The three main components are the computer (with various application programs), the keyboard (based on that of a piano), and the set of sound generators. Essentially, the computer takes input commands from the keyboard and outputs these to the sound generators which, in turn, produce the corresponding sound waveform – via DACs – to drive the speakers.

Pressing a key on the keyboard of a synthesizer has a similar effect to pressing a key on the keyboard of a computer inasmuch as, for each key that is pressed, a different codeword – known as a message with a synthesizer keyboard – is generated and read by the computer program. Essentially, the messages indicate such things as which key on the keyboard has been pressed and the pressure applied. The control panel contains a range of different

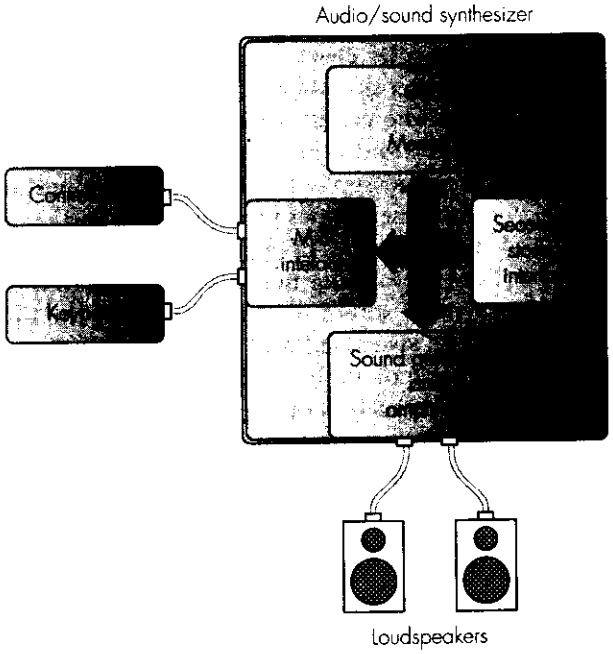


Figure 2.18 Audio/sound synthesizer schematic.

switches and sliders that collectively allow the user to indicate to the computer program additional information such as the volume of the generated output and selected sound effects to be associated with each key.

The secondary storage interface allows the sequence of messages – including those associated with the control panel – relating to a particular piece of audio to be saved on secondary storage such as a floppy disk. In addition, there are programs to allow the user to edit a previously entered passage and, if required, to mix several stored passages together. The sequencer program associated with the synthesizer then ensures that the resulting integrated sequence of messages are synchronized and output to the sound generators to create the merged passage.

As well as a (piano) keyboard, there is a range of other possible inputs from instruments such as an electric guitar, all of which generate messages similar to those produced by the keyboard. Hence in order to discriminate between the inputs from the different possible sources, a standardized set of messages have been defined for both input and for output to the corresponding set of sound generators. These are defined in a standard known as the **Music Instrument Digital Interface (MIDI)**. As the name implies, this does not just define the format of the standardized set of messages used by a synthesizer, but also the type of connectors, cables, and electrical signals that are used to connect any type of device to the synthesizer.

The format of a MIDI message consists of a *status byte*, which defines the particular event that has caused the message to be generated, followed by a number of *data bytes* which collectively define a set of parameters associated with the event. An example of an event is a key being pressed on the keyboard and typical parameters would then be the identity of the key, the pressure applied, and so on. As we indicated earlier, there can be a variety of instrument types used for input and output. Thus it is also necessary to identify the type of instrument that generated the event so that when the corresponding message is output to the sound generators, the appropriate type of sound is produced. Thus the different types of device have a MIDI code associated with them; a piano, for example, has a code of 0 and a violin a code of 40. In addition, some codes have been assigned for specific special effects such as the sound from a cannon and the applause from an audience.

As we can deduce from the above, a passage of audio produced by a synthesizer consists of a very compact sequence of messages – each comprising a string of bytes – which can either be played out by the sequencer program directly – and hence heard by the composer – or saved in a file on a floppy disk. Typically, in many interactive applications involving, say, multimedia pages comprising text and a passage of music, a synthesizer is first used to create the passage of music which is then saved in a file. The author of the pages then links the file contents to the text at the point where the music is to be played. Clearly, since the music is in the form of a sequence of MIDI messages, it is necessary to have a **sound card** in the client computer to interpret the sequence of messages and generate the appropriate sounds. The sound generators use either **FM synthesis** techniques or samples of sound produced by real instruments. The latter is known as **wavelet synthesis**.

2.6 Video

Video features in a range of multimedia applications:

- entertainment: broadcast television and VCR/DVD recordings;
- interpersonal: video telephony and videoconferencing;
- interactive: windows containing short video clips.

The quality of the video required, however, varies considerably from one type of application to another. For example, for video telephony, a small window on the screen of a PC is acceptable while for a movie, a large screen format is preferable. In practice, therefore, there is not just a single standard associated with video but rather a set of standards, each targeted at a particular application domain. Before describing a selection of these we must first acquire an understanding of the basic principles associated with broadcast television on which all the standards are based.

2.6.1 Broadcast television

We considered the basic principles of color television picture tubes earlier in Section 2.4.3 when the topic of digitized pictures was discussed. As you may recall, a color picture/image is produced from varying mixtures of the three primary colors red, green and blue. The screen of the picture tube is coated with a set of three different phosphors – one for each color – each of which is activated by a separate electron beam. The three electron beams are scanned in unison across the screen from left to right with a resolution of either 525 lines (NTSC) or 625 lines (PAL/CCIR/SECAM). The total screen contents are then refreshed at a rate of either 60 or 50 frames per second respectively, the rate being determined by the frequency of the mains electricity supply used in the different countries.

The computer monitors used with most personal computers use the same picture tubes as those in broadcast television receivers and hence operate in a similar way. The three digitized color signals that make up a stored picture/image are read from the computer memory in time-synchronism with the scanning operation of the display tube and, after each complete scan of the display, the procedure repeats so producing a flicker-free color image on the screen. In principle, broadcast television could operate in a similar way, but in practice it operates slightly differently both in terms of the scanning sequence used and in the choice of color signals. We shall look at each separately.

Scanning sequence

Although it is necessary to use a minimum refresh rate of 50 times per second to avoid flicker, to produce smooth motion, a refresh rate of 25 times per second is sufficient. Hence in order to minimize the amount of transmission bandwidth that is required to broadcast the television signal, this characteristic of the eye is exploited by transmitting the image/picture associated with each frame in two halves. Each is known as a **field**, the first comprising only the odd scan lines and the second the even scan lines. The two fields are then integrated together in the television receiver using a technique known as **interlaced scanning**, the principles of which are shown in Figure 2.19.

As we can see, in a 525-line system each field comprises 262.5 lines – 240 visible – while in a 625-line system each field comprises 312.5 lines – 288 visible, the remainder being used for other purposes. Each field is refreshed alternately at 60/50 fields per second and hence the resulting frame refresh rate is only 30/25 frames per second. As discussed, the higher field rate tricks the eye into thinking the frame rate is double what it is in practice. In this way, a refresh rate equivalent to 60/50 frames per second is achieved but with only half the transmission bandwidth.

Color signals

For historical reasons, the received signals associated with a color television broadcast had to be such that they could be used by an existing (unmodified) monochrome (black-and-white) television set to produce the same picture in high-quality monochrome. In addition, a color television had to be able to

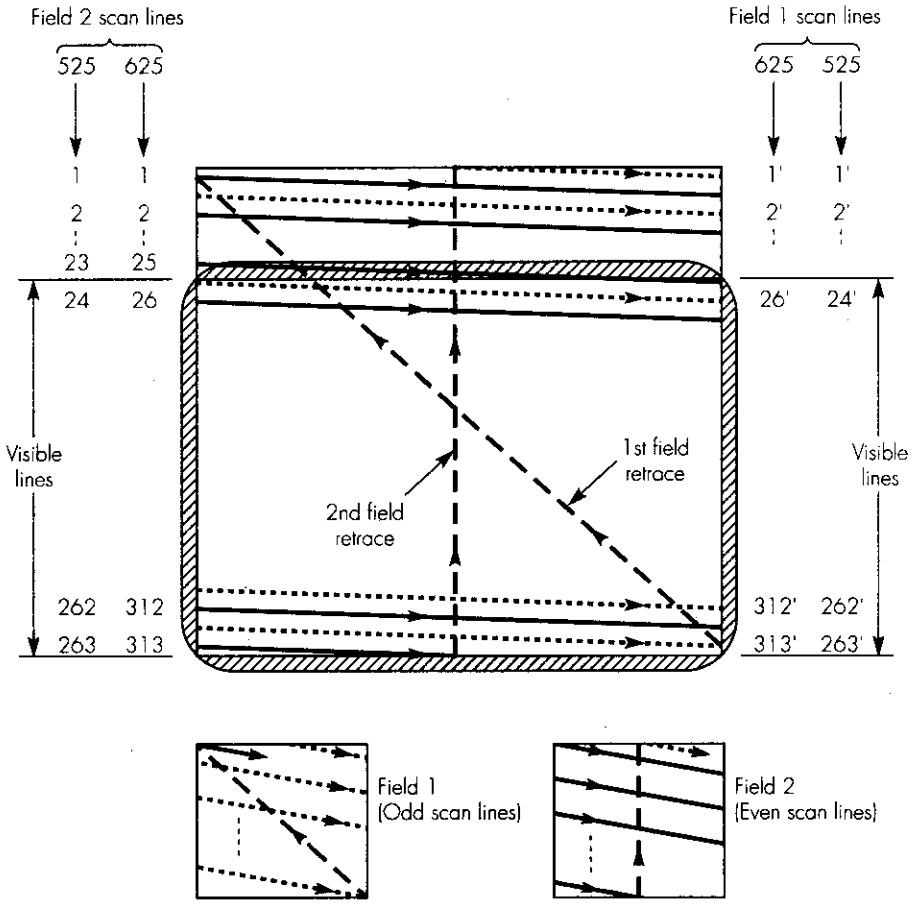


Figure 2.19 Interlaced scanning principles.

produce black-and-white pictures from monochrome broadcasts. For these reasons a different set of color signals from R, G, and B were selected for color television broadcasts.

The three main properties of a color source that the eye makes use of are:

- **brightness:** this represents the amount of energy that stimulates the eye and varies on a gray scale from black (lowest) through to white (highest). It is thus independent of the color of the source;
- **hue:** this represents the actual color of the source, each color has a different frequency/wavelength and the eye determines the color from this;

- **saturation:** this represents the strength or vividness of the color, a pastel color has a lower level of saturation than a color such as red. Also, a saturated color such as red has no white light in it.

The term **luminance** is used to refer to the brightness of a source, and the hue and saturation, because they are concerned with its color, are referred to as its **chrominance** characteristics.

As we saw in Section 2.4.3, a range of colors can be produced by mixing the three primary colors R, G, and B. In a similar way, a range of colors can be produced on a television display screen by varying the magnitude of the three electrical signals that energize the red, green, and blue phosphors. For example, if the magnitude of the three signals are in the proportion

$$0.299R + 0.587G + 0.114B$$

then the color white is produced on the display screen. Hence, since the luminance of a source is only a function of the amount of white light it contains, for any color source its luminance can be determined by summing together the three primary components that make up the color in this proportion. That is,

$$Y_s = 0.299\ R_s + 0.587\ G_s + 0.144\ B_s$$

where Y_s is the amplitude of the luminance signal and R_s, G_s, and B_s are the magnitudes of the three color component signals that make up the source. Thus, since the luminance signal is a measure of the amount of white light it contains, it is the same as the signal used by a monochrome television. Two other signals, the **blue chrominance** (C_b), and the **red chrominance** (C_r), – are then used to represent the coloration – hue and saturation – of the source. These are obtained from the two **color difference** signals:

$$C_b = B_s - Y_s \quad \text{and} \quad C_r = R_s - Y_s$$

which, since the Y signal has been subtracted in both cases, contain no brightness information. Also, since Y is a function of all three colors, then G can be readily computed from these two signals. In this way, the combination of the three signals Y, C_b, and C_r contains all the information that is needed to describe a color signal while at the same time being compatible with monochrome televisions which use the luminance signal only.

Chrominance components

In practice, although all color television systems use this same basic principle to represent the coloration of a source, there are some small differences between the two systems in terms of the magnitude used for the two chrominance signals. These arise from the constraint that the bandwidth of the transmission channel for color broadcasts must be the same as that used for

monochrome. As a result, in order to fit the Y, C_b, and C_r signals in the same bandwidth, the three signals must be combined together for transmission. The resulting signal is then known as the **composite video signal**. As a result of doing this, however, if the two color difference signals are transmitted at their original magnitudes, the amplitude of the luminance signal can become greater than that of the equivalent monochrome signal. This leads to a degradation in the quality of the monochrome picture and hence is unacceptable.

To overcome this effect, the magnitude of the two color difference signals are both scaled down. In addition, since they both have different levels of luminance associated with them, the scaling factor used for each signal is different. In practice, the two color difference signals are referred to by different symbols in each system. In the PAL system, for example, C_b and C_r are referred to as U and V respectively and the scaling factors used for the three signals are:

$$\begin{aligned} \text{PAL:}\quad Y &= 0.299\,R + 0.587\,G + 0.114\,B \\ U &= 0.493\,(B - Y) \\ V &= 0.877\,(R - Y) \end{aligned}$$

In the case of the NTSC system, the two color difference signals are combined to form two different signals referred to as I and Q. The scaling factors used are:

$$\begin{aligned} \text{NTSC:}\quad Y &= 0.299\,R + 0.587\,G + 0.114\,B \\ I &= 0.74\,(R - Y) - 0.27\,(B - Y) \\ Q &= 0.48\,(R - Y) + 0.41\,(B - Y) \end{aligned}$$

Example 2.6

Derive the scaling factors used for both the U and V (as used in PAL) and I and Q (as used in NTSC) color difference signals in terms of the three R, G, B color signals.

Answer:

PAL:
$$Y = 0.299R + 0.587G + 0.114B$$
$$U = 0.493\,(B - Y) \quad \text{and} \quad V = 0.877\,(R - Y)$$

Hence
$$\begin{aligned} U &= 0.493B - 0.493\,(0.299R + 0.587G + 0.114B) \\ &= -0.147R - 0.289G + 0.437B \end{aligned}$$

and
$$\begin{aligned} V &= 0.877R - 0.877\,(0.299R + 0.587G + 0.114B) \\ &= 0.615R - 0.515G - 0.100B \end{aligned}$$

NTSC:
$$\begin{aligned} I &= 0.74\,(R - Y) - 0.27\,(B - Y) \\ &= 0.74R - 0.27B - 0.47Y \\ &= 0.599R - 0.276G - 0.324B \end{aligned}$$
$$\begin{aligned} Q &= 0.48\,(R - Y) + 0.41\,(B - Y) \\ &= 0.48R + 0.41B - 0.89Y \\ &= 0.212R - 0.528G + 0.311B \end{aligned}$$

Signal bandwidth

As we saw in the last section, the bandwidth of the transmission channel used for color broadcasts must be the same as that used for a monochrome broadcast. As a result, for transmission, the two chrominance signals must occupy the same bandwidth as that of the luminance signal. The baseband spectrum of a color television signal in both the NTSC and PAL systems is shown in parts (a) and (b) of Figure 2.20 respectively.

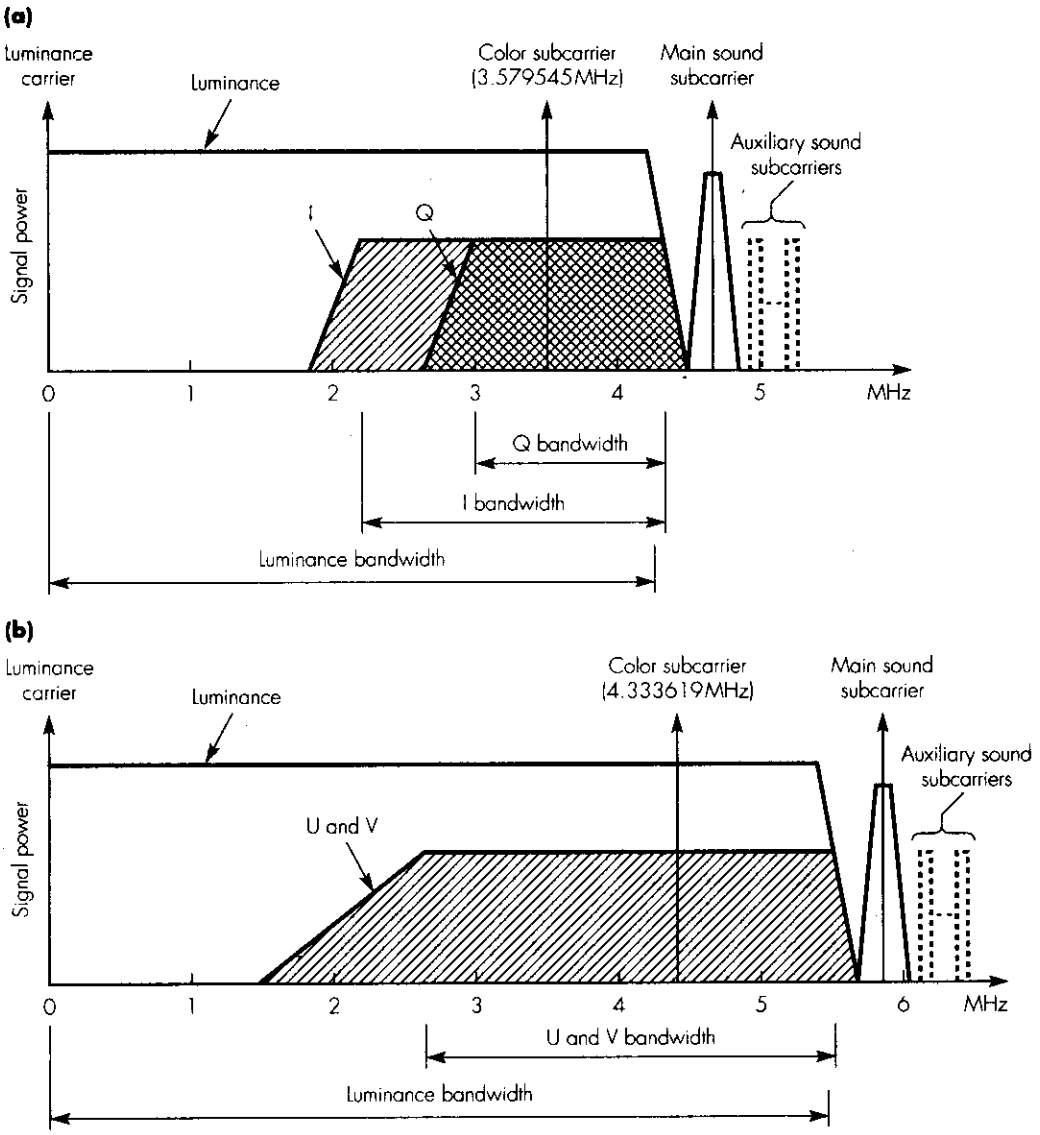


Figure 2.20 Baseband spectrum of color television signals: (a) NTSC system; (b) PAL system.

In practice, most of the energy associated with the luminance signal is in the lower frequency signals and hence the lower part of its frequency spectrum. Hence in order to minimize the level of interference between the luminance and two chrominance signals, firstly, the latter are transmitted in the upper part of the luminance frequency spectrum using two separate subcarriers and secondly, to restrict the bandwidth used to the upper part of the spectrum, a smaller bandwidth is used for both chrominance signals. In addition, both of the two chrominance subcarriers have the same frequency but they are 90 degrees out of phase with each other. Each is modulated independently in both amplitude and phase by the related chrominance signal. Using this technique, the two signals can use the same portion of the luminance frequency spectrum.

In the NTSC system the eye is more responsive to the I signal than the Q signal. Hence to maximize the use of the available bandwidth while at the same time minimizing the level of interference with the luminance signal, the I signal has a modulated bandwidth of about 2 MHz and the Q signal a bandwidth of about 1 MHz. With the PAL system, the larger luminance bandwidth – about 5.5 MHz relative to 4.2 MHz – allows both the U and V chrominance signals to have the same modulated bandwidth which is about 3 MHz. As we show in the figure, the audio/sound signal is transmitted using one or more separate subcarriers which are all just outside of the luminance signal bandwidth. Typically, the main audio subcarrier is for mono sound and the auxiliary subcarriers are used for stereo sound. When these are added to the baseband video signal, the composite signal is called the **complex baseband signal**.

2.6.2 Digital video

In the previous section we described the underlying principles of broadcast television and, in particular, the origin of the three component signals that are used. In most multimedia applications the video signals need to be in a digital form since it then becomes possible to store them in the memory of a computer and to readily edit and integrate them with other media types. In addition, although for transmission reasons the three component signals have to be combined for analog television broadcasts, with digital television it is more usual to digitize the three component signals separately prior to their transmission. Again, this is done to enable editing and other operations to be readily performed.

Since the three component signals are treated separately in digital television, in principle it is possible simply to digitize the three RGB signals that make up the picture. The disadvantage of this approach is that the same resolution – in terms of sampling rate and bits per sample – must be used for all three signals. Studies on the visual perception of the eye have shown that the resolution of the eye is less sensitive for color than it is for luminance. This means that the two chrominance signals can tolerate a reduced resolution relative to that used for the luminance signal. Hence a significant saving in

terms of the resulting bit rate – and hence transmission bandwidth – can be achieved by using the luminance and two color difference signals instead of the RGB signals directly.

Digitization of video signals has been carried out in television studios for many years in order, for example, to perform conversions from one video format into another. In order to standardize this process – and hence make the exchange of television programmes internationally easier – the international body for television standards, the **International Telecommunications Union – Radiocommunications Branch (ITU-R)** – formerly known as the **Consultative Committee for International Radiocommunications (CCIR)** – defined a standard for the digitization of video pictures known as **Recommendation CCIR-601**. In addition, a number of variants of this standard have been defined for use in other application domains such as digital television broadcasting, video telephony, and videoconferencing. Collectively these are known as **digitization formats** and, in practice, they all exploit the fact that the two chrominance signals can tolerate a reduced resolution relative to that used for the luminance signal. We shall now describe a selection of these.

4:2:2 format

This is the original digitization format used in Recommendation CCIR-601 for use in television studios. The three component (analog) video signals from a source in the studio can have bandwidths of up to 6 MHz for the luminance signal and less than half this for the two chrominance signals. To digitize these signals, as we described in Section 2.2, it is necessary to use band-limiting filters of 6 MHz for the luminance signal and 3 MHz for the two chrominance signals with a minimum sampling rate of 12 MHz (12 Msps) and 6 MHz respectively.

In the standard, however, a line sampling rate of 13.5 MHz for luminance and 6.75 MHz for the two chrominance signals was selected, both of which are independent of the particular scanning standard – NTSC, PAL and so on – being used. The 13.5 MHz rate was chosen since it is the nearest frequency to 12 MHz which results in a whole number of samples per line for both 525- and 625-line systems. The number of samples per line chosen is 702 and can be derived as follows:

In a 525-line system, the total line sweep time is 63.56 microseconds but, during this time, the beam(s) is (are) turned off – set to the black level – for retrace for 11.56 microseconds which yields an active line sweep time of 52 microseconds. Similarly, in a 625-line system, the total line sweep time is 64 microseconds with a blanking time of 12 microseconds which also yields an active line sweep time of 52 microseconds. Hence in both cases, a sampling rate of 13.5 MHz yields:

$$52 \times 10^{-6} \times 13.5 \times 10^{6} = 702 \text{ samples per line}$$

In practice, the number of samples per line is increased to 720 by taking a slightly longer active line time which results in a small number of black samples

at the beginning and end of each line for reference purposes. The corresponding number of samples for each of the two chrominance signals is then set at half this value; that is, 360 samples per active line. This results in $4Y$ samples for every $2C_b$ and $2C_r$ samples which is the origin of the term 4:2:2, the term 4:4:4 normally indicating the digitization is based on the R, G, B signals.

The number of bits per sample was chosen to be 8 for all three signals which corresponds to 256 quantization intervals. In addition, the vertical resolution for all three signals was also chosen to be the same, the precise number being determined by the scanning system in use; that is, 480 lines with a 525-line system and 576 lines with a 625-line system, the numbers 480 and 576 being the number of active (visible) lines in the respective system. Also, since the 4:2:2 format is intended for use in television studios, noninterlaced scanning is used at a frame refresh rate of either 60 Hz for a 525-line system or 50 Hz for a 625-line system.

Since each line is sampled at a constant rate (13.5 and 6.75 MHz) with a fixed number of samples per line (720 and 360), the samples for each line are in a fixed position which repeats from frame to frame. The samples are then said to be **orthogonal** and the sample method **orthogonal sampling**. Since each system (525 and 625) has a fixed number of (active) lines per frame, the sampling positions for each of the three signals relative to a rectangular grid are as shown in Figure 2.21.

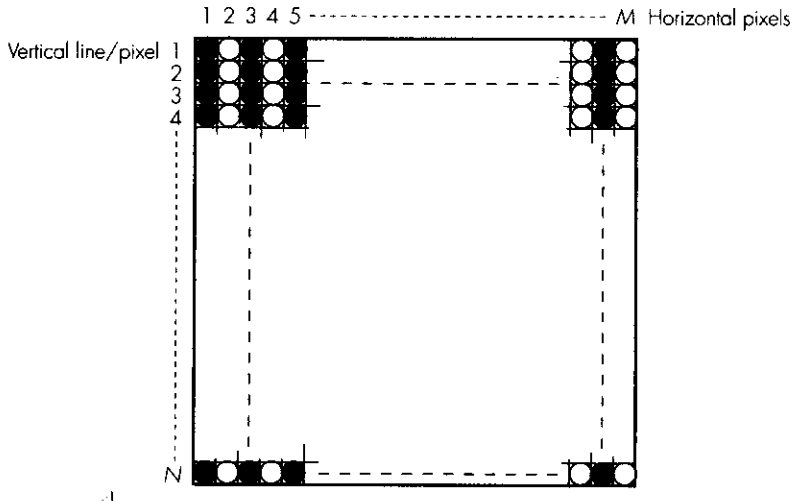


Figure 2.21 Sample positions with 4:2:2 digitization format.

Example 2.7

Derive the bit rate and the memory requirements of video frames that result from the digitization of both a 525-line and a 625-line system assuming a 4:2:2 format. Also find the total memory required to store a 1.5 hour movie/video.

Answer:

525-line system: The number of samples per line is 720 and the number of visible lines is 480. Hence the resolution of the luminance (Y) and the two chrominance (C_b and C_r) signals are:

$$Y = 720 \times 480$$
$$C_b = C_r = 360 \times 480$$

Bit rate: Line sampling rate is fixed at 13.5 MHz for Y and 6.75 MHz for both C_b and C_r, all with 8 bits per sample.

Hence: Bit rate = $13.5 \times 10^6 \times 8 + 2 (6.75 \times 10^6 \times 8) = 216$ Mbps

Memory required: Memory required per line = $720 \times 8 + 2(360 \times 8)$ = 11 520 bits or 1440 bytes

Hence memory per frame, each of 480 lines = $480 \times 11\,520$ = 5.5296 Mbits or 691.2 kbytes

and memory to store 1.5 hours, assuming 60 frames per second:

= $691.2 \times 60 \times 1.5 \times 3600$ kbytes
= 223.9488 Gbytes

625-line system: Resolution: $Y = 720 \times 576$
$C_b = C_r = 360 \times 576$

Bit rate = $13.5 \times 10^6 \times 8 + 2 (6.75 \times 10^6 \times 8) = 216$ Mbps

Memory per frame = $576 \times 11\,520 = 6.635\,55$ Mbits or 829.44 kbytes

and memory required for 1.5 hours assuming 50 frames per second:

= $829.44 \times 50 \times 1.5 \times 3600$ kbytes
= 223.9488 Gbytes

It should be noted that, in practice, the bit rate figures are less than the computed values since they include samples during the retrace times when the beam is switched off. Nevertheless, as we can deduce from the computed values, both the bit rate and the memory requirements are very large for both systems and it is for this reason that the various lower resolution formats have been defined.

4:2:0 format

This format is a derivative of the 4:2:2 format and is used in digital video broadcast applications. It has been found to give good picture quality and is derived by using the same set of chrominance samples for two consecutive lines. Since it is intended for broadcast applications, interlaced scanning is used and the absence of chrominance samples in alternative lines is the origin of the term 4:2:0. The position of the three sample instants per frame are as shown in Figure 2.22.

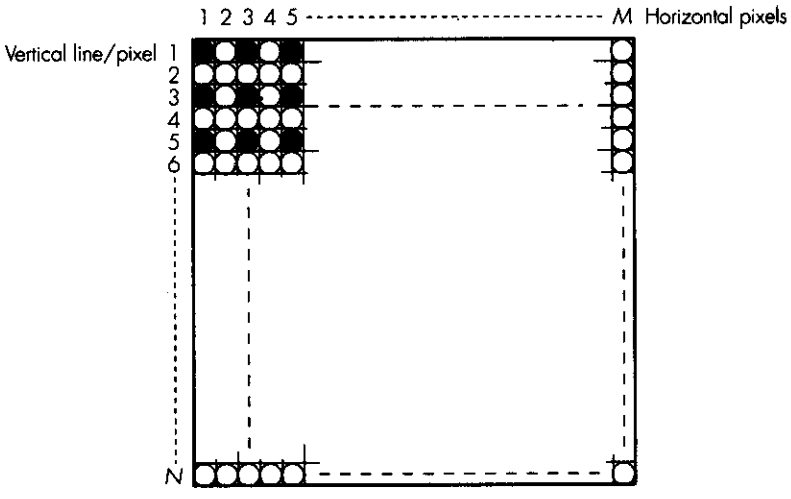
As we can see from the figure, this yields the same luminance resolution as the 4:2:2 format but half the chrominance resolution:

525-line system: $Y = 720 \times 480$
$C_b = C_r = 360 \times 240$

625-line system: $Y = 720 \times 576$
$C_b = C_r = 360 \times 288$

The bit rate in both systems with this format is:

$$13.5 \times 10^6 \times 8 + 2\,(3.375 \times 10^6 \times 8) = 162 \text{ Mbps}$$



O = Y, + = C_b, X = C_r sample positions

525-line systems: $M = 720$, $N = 480$, 60Hz refresh rate (interlaced)
$Y = 720 \times 480$, $C_b = C_r = 360 \times 240$

625-line systems: $M = 720$, $N = 576$, 50Hz refresh rate (interlaced)
$Y = 720 \times 576$, $C_b = C_r = 360 \times 288$

Figure 2.22 Sample positions in 4:2:0 digitization format.

It should be noted, however, that, as we pointed out in Example 2.7, this is the worst-case bit rate since it includes samples during the retrace times when the beam is switched off. Also, to avoid flicker effects with the chrominance signals, the receiver uses the same chrominance values from the sampled lines for the missing lines. With large-screen televisions, flicker effects are often reduced further by the receiver storing the incoming digitized signals of each field in a memory buffer. A refresh rate of double the normal rate – 100/120 Hz – is then used with the stored set of signals used for the second field.

HDTV formats

There are a number of alternative digitization formats associated with high-definition television (HDTV). The resolution of those which relate to the older 4/3 aspect ratio tubes can be up to 1440 × 1152 pixels and the resolution of those which relate to the newer 16/9 wide-screen tubes can be up to 1920 × 1152 pixels. In both cases, the number of visible lines per frame is 1080 which produces a square-pixel lattice structure with both tube types. Both use either the 4:2:2 digitization format for studio applications or the 4:2:0 format for broadcast applications. The corresponding frame refresh rate is either 50/60 Hz with the 4:2:2 format or 25/30 Hz with the 4:2:0 format. Hence in the case of the 1440 × 1152 resolution, the resulting worst-case bit rates are four times the values derived in the previous two sections and proportionally higher for the wide-screen format.

SIF

The **source intermediate format (SIF)** has been found to give a picture quality comparable with that obtained with video cassette recorders (VCRs). It uses half the spatial resolution in both horizontal and vertical directions as that used in the 4:2:0 format – a technique known as **subsampling** – and, in addition, uses half the refresh rate – also known as **temporal resolution**. This means that the frame refresh rate is 30 Hz for a 525-line system and 25 Hz for a 625-line system. Since the SIF is intended for storage applications, progressive (non-interlaced) scanning is used. The digitization format is known, therefore as 4:1:1. The positions of the three sampling instants per frame are as shown in Figure 2.23.

As we can deduce from the figure, this yields resolutions of:

525-line system: $Y = 360 \times 240$
$C_b = C_r = 180 \times 120$

625-line system: $Y = 360 \times 288$
$C_b = C_r = 180 \times 144$

The worst-case bit rate in both systems with this format is:

$$6.75 \times 10^6 \times 8 + 2(1.6875 \times 10^6 \times 8) = 81 \text{ Mbps}$$

At the receiver, the missing samples are estimated by interpolating between each pair of values that are sent.

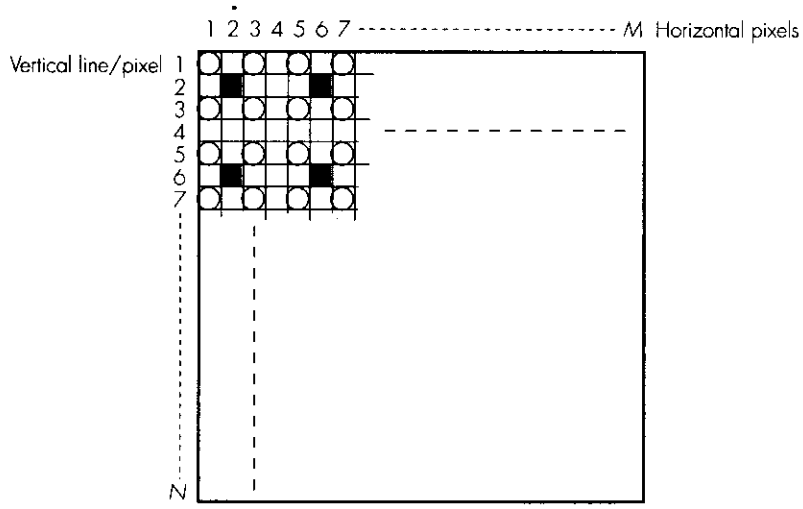


Figure 2.23 Sample positions for SIF and CIF.

CIF

The **common intermediate format (CIF)** has been defined for use in videoconferencing applications. It is derived from the SIF and uses a combination of the spatial resolution used for the SIF in the 625-line system and the temporal resolution used in the 525-line system. This yields a spatial resolution of:

$$Y = 360 \times 288$$
$$C_b = C_r = 180 \times 144$$

with a temporal resolution of 30 Hz using progressive scanning. The positions of the sampling instants per frame are the same as for SIF and hence the digitization format is 4:1:1. Similarly, the worst-case bit rate is the same and hence is 81 Mbps. As we can deduce from this, to convert to the CIF, a 525-line system needs a line-rate conversion and a 625-line system a frame-rate conversion.

In addition to the basic CIF, a number of higher-resolution derivatives of the CIF have been defined. As we described earlier in Section 1.4.1, there are a number of different types of videoconferencing applications including those that involve a linked set of desktop PCs and those that involve a linked

set of videoconferencing studios. In general, therefore, because most desktop applications use switched circuits, a typical bit rate used is a single 64 kbps ISDN channel. For linking videoconferencing studios, however, dedicated circuits are normally used that comprise multiple 64 kbps channels. Hence because the bit rate of these circuits is much higher – typically four or sixteen 64 kbps channels – then a higher-resolution version of the basic CIF can be used to improve the quality of the video. Two examples are:

$$\begin{aligned} \text{4CIF:} \quad & Y = 720 \times 576 \\ & C_b = C_r = 360 \times 288 \\ \text{16CIF:} \quad & Y = 1440 \times 1152 \\ & C_b = C_r = 720 \times 576 \end{aligned}$$

QCIF

The **quarter CIF** (**QCIF**) format has been defined for use in video telephony applications. It is derived from the CIF and uses half the spatial resolution of CIF in both horizontal and vertical directions and the temporal resolution is divided by either 2 or 4. This yields a spatial resolution of:

$$\begin{aligned} Y &= 180 \times 144 \\ C_b = C_r &= 90 \times 72 \end{aligned}$$

with a temporal resolution of either 15 or 7.5 Hz. The worst-case bit rate with this format is:

$$3.375 \times 10^6 \times 8 + 2\,(0.843\,75 \times 10^6 \times 8) = 40.5\,\text{Mbps}$$

The positions of the three sampling instants per frame are as shown in Figure 2.24 and, as we can see, it has the same 4:1:1 digitization format as CIF.

As we described in Section 1.4.1, a typical video telephony application involves a single switched 64 kbps channel and the QCIF is intended for use with such channels. In addition, there are lower-resolution versions of the QCIF which are intended for use in applications that use lower bit rate channels such as that provided by a modem and the PSTN. These lower-resolution versions are known as **sub-QCIF** or **S-QCIF** and an example is:

$$\begin{aligned} Y &= 128 \times 96 \\ C_b = C_r &= 64 \times 48 \end{aligned}$$

It should be noted, however, that although the sampling matrix appears sparse, in practice, only a small screen (or a small window of a larger screen) is normally used for video telephony and hence the total set of samples may occupy all the pixel positions on the screen or window.

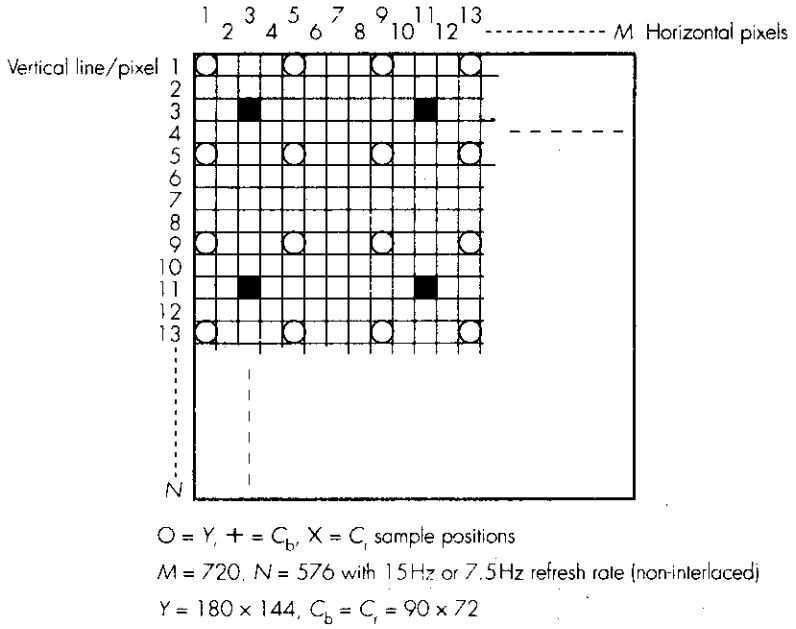


Figure 2.24 Sample positions for QCIF.

2.6.3 PC video

All the digitization formats described in Section 2.6.2 are intended for use with standard television receivers. However, as we discussed in Chapter 1, a number of multimedia applications that involve live video, use a window on the screen of a PC monitor for display purposes. Examples include desktop video telephony and videoconferencing, and also **video-in-a-window**.

As we discussed under the subheading "Aspect ratio" in Section 2.4.3, in order to avoid distortion on a PC screen – for example when displaying a square of $N \times N$ pixels – it is necessary to use a horizontal resolution of 640 ($480 \times 4/3$) pixels per line with a 525-line PC monitor and 768 ($576 \times 4/3$) pixels per line with a 625-line PC monitor. Hence for multimedia applications that involve mixing live video with other information on a PC screen, the line sampling rate is normally modified in order to obtain the required horizontal resolution.

To achieve the necessary resolution with a 525-line monitor, the line sampling rate is reduced from 13.5 MHz to 12.2727 MHz while for a 625-line monitor, the line sampling rate must be increased from 13.5 MHz to 14.75 MHz. In the case of desktop video telephony and videoconferencing, the video signals from the camera are sampled at this rate prior to transmission and hence can be displayed directly on a PC screen. In the case of a digital television broadcast a conversion is necessary before the video is displayed. The various digitization formats for use with PC video are as shown in Table 2.2. It should be remembered that all PC monitors use progressive (non-interlaced) scanning rather than interlaced scanning.

Table 2.2 PC video digitization formats.

Digitization format	System	Spatial resolution	Temporal resolution
4:2:0	525-line	$Y = 640 \times 480$ $C_b = C_r = 320 \times 240$	60 Hz
	625-line	$Y = 768 \times 576$ $C_b = C_r = 384 \times 288$	50 Hz
SIF	525-line	$Y = 320 \times 240$ $C_b = C_r = 160 \times 240$	30 Hz
	625-line	$Y = 384 \times 288$ $C_b = C_r = 192 \times 144$	25 Hz
CIF		$Y = 384 \times 288$ $C_b = C_r = 192 \times 144$	30 Hz
QCIF		$Y = 192 \times 144$ $C_b = C_r = 96 \times 72$	15/7.5 Hz

2.6.4 Video content

In the preceding section we described the various digitization formats that have been defined for use in different application domains of digital video. In terms of the actual video content, therefore, this depends on the particular application. For example, in entertainment applications, the content will be either a broadcast television program or, in a video-on-demand application, a digitized movie that is accessed from a suitable server. Similarly, in interpersonal applications such as video telephony and videoconferencing, the video source will be derived from a video camera and the digitized sequence of pixels relating to each frame are transmitted across the network. As the pixels are received at the destination, they are displayed directly on either a television screen or a computer monitor.

In addition, in many interactive applications that involve video, the short video clips associated with the application are obtained by plugging a video camera into a **video capture board** within the computer that is preparing the (interactive) page contents. Normally, the computer stores the digitized video produced by such boards into a file ready for linking to the other page contents.

In other applications the video may be generated by a computer program rather than a video camera. This type of video content is normally referred to as **computer animation** or sometimes, because of the way it is generated,

animated graphics. A range of special programming languages is available for creating computer animation. Hence in the same way that a graphical image produced by a graphics program can be represented in the form of either a high-level program or a pixel image, so a computer animation can be represented in the form of either an animation program or a digital video. As before, the form used depends on the application. In general the digital video form of representation of an animation requires considerably more memory and transmission bandwidth than the corresponding high-level program form.

The negative side of a high-level program form is that the low-level animation primitives that the program uses – move object, rotate object, object fill, and so on – have to be executed very fast in order to produce smooth motion on the display. Hence it is now common to have an additional (**3-D**) **graphics accelerator** processor to carry out these functions. Typically, the (host) processor simply passes the sequence of low-level primitives to the accelerator processor at the appropriate rate. The accelerator then, in turn, executes each set of primitives to produce the corresponding pixel image in the video RAM at the desired refresh rate.

Summary

In this chapter we have described the different ways that the range of media types associated with the various multimedia applications we identified in Chapter 1 are represented in their digital form. These included various types of text, images, digitized documents and pictures, audio – both speech and music – and video. In the case of audio and video, the conversion operations that are used to convert them from their source analog form into their corresponding digital form were also described. In practice, with these basic forms of representation, the amount of bandwidth that is required to transfer the total quantity of information associated with a particular application is considerably larger than that which is available with many of the communication networks used for these applications.

For example, the bandwidth available for digital television – in terms of bits per second – is in the order of 40 Mbps with cable and terrestrial broadcast systems and 60 Mbps with a satellite channel. Clearly, these are both still considerably less than the bit rate requirement of 162 Mbps that is generated using the 4:2:0 digitization format. Similarly, the bit rate available with a connection through the all-digital ISDN is between 64 kbps and 2 Mbps and hence again the bandwidth requirements for videoconferencing – 81 Mbps with the CIF – and video telephony – 40.5 Mbps with the QCIF – are both far in excess of these two values. Hence in order to provide such services over the related networks, it is necessary to reduce the bandwidth requirements of the source signals considerably. In addition, when using public networks such as a

PSTN or an ISDN in which call charges are based on the duration of a call, considerable cost savings can be made if the amount of data to be transmitted is reduced. For example, if the amount of data is reduced by a factor of two, then the cost of the call will be halved.

In most multimedia applications, therefore, a technique known as compression is first applied to the source information prior to its transmission. This is done either to reduce the volume of data to be transmitted – for example with text, fax, and images – or to reduce the bandwidth that is required for its transmission – for example with speech, audio, and video. In the next chapter we describe a range of compression algorithms that have been developed to reduce the volume of the data associated with text, fax, and images and, in Chapter 4, a range of compression algorithms associated with audio and video.

Exercises

Section 2.1

2.1 Explain the meaning of the following terms:
(i) codeword,
(ii) analog signal,
(iii) signal encoder,
(iv) signal decoder.

Section 2.2

2.2 With the aid of a set of signal waveforms, show the principles of how a time-varying analog signal is made up of a range of sinusoidal frequency components of differing amplitude and phase relative to one another.

2.3 Define the term "signal bandwidth". Hence show in graphical form the bandwidth of a speech signal and a music signal. Clearly show the dimensions of the horizontal and vertical axes.

2.4 Define the meaning of the term "channel bandwidth" in relation to a transmission channel. Hence with the aid of a diagram, explain the meaning of the term "bandlimiting channel".

2.5 Use a diagram to identify the main circuit components associated with a signal encoder. Hence by means of an associated set of signal waveforms, explain the meaning of the terms:
(i) bandlimiting filter,
(ii) ADC,
(iii) sample-and-hold,
(iv) quantizer.

2.6 Explain the meaning of the following terms relating to the sampling of an analog signal:
(i) Nyquist sampling theorem,
(ii) Nyquist rate.

2.7 Show by means of a diagram how sampling of an analog signal at a rate lower than the Nyquist rate can generate additional lower-frequency alias signals to those present in the original waveform. How can this be avoided?

2.8 Define the meaning of the term "quantization interval" and how this influences the accuracy of the sampling process of an analog signal. Hence with the aid of a diagram, explain the meaning of the terms "quantization error" and "quantization noise".

2.9 State the meaning of the term "dynamic range" as applied to an analog signal and show how this is expressed in decibels. How does this influence the number of bits to be used for the quantizer part of an ADC?

2.10 With the aid of a diagram and an associated waveform set, explain the function of the following components that make up a signal decoder:
(i) DAC,
(ii) low-pass filter.

Why is the latter also known as a recovery/reconstruction filter?

Section 2.3

2.11 State the meaning of the following types of text:
(i) unformatted/plaintext,
(ii) formatted/richtext,
(iii) hypertext.

2.12 By means of examples, show how the 7-bit ASCII character set can be extended to create additional characters and symbols. State one of the uses of an extended character set.

2.13 How is formatted text different from unformatted text? Hence describe the meaning of the term "text" and "document formatting commands". What is the origin of the acronym WYSIWYG?

2.14 Describe the terms "hypertext", "pages/documents", and "hyperlinks".

2.15 With the aid of diagrams where appropriate, describe the meaning of the following terms relating to HTML and the World Wide Web:
(i) browser,
(ii) home page,
(iii) URL,
(iv) page formatting commands.

Section 2.4

2.16 Explain the meaning of the following terms relating to graphical images:
(i) visual object,
(ii) freeform object,
(iii) clip-art,
(iv) 3-D objects.

2.17 With the aid of diagrams, explain the meaning of the following terms relating to graphical images:
(i) pixels,
(ii) video graphics array,
(iii) image object,
(iv) object attributes,
(v) open and closed object shapes,
(vi) rendering,
(vii) bit-map format.

2.18 With the aid of a diagram, explain the meaning of the following terms relating to facsimile machines:
(i) scanning,
(ii) pels,
(iii) digitization resolution.

2.19 What is the difference between a bitonal image and a continuous-tone image?

2.20 With the aid of diagrams where appropriate, explain the meaning of the terms:
(i) color gamut,
(ii) additive color mixing,
(iii) subtractive color mixing.

Give an application of both color mixing methods.

2.21 With the aid of diagrams, describe the raster-scan operation associated with TV/computer monitors. Include in your description the meaning of the terms:
(i) line scan,
(ii) horizontal and vertical retrace,
(iii) phosphor triad,
(iv) frame refresh rate,
(v) flicker,
(vi) pixel depth,
(vii) video RAM,
(viii) video controller.

2.22 Define the aspect ratio of a display screen. Give two examples for current widely used screen sizes.

2.23 What is the number of scan lines per frame associated with each of the following TV monitors:
(i) NTSC,
(ii) PAL?

In practice, the number of visible lines per frame are less than these values. State what these are for each type of monitor. In each case, derive the number of pixels per scan line that are used to obtain square pixels assuming a 4/3 aspect ratio.

2.24 Most high resolution computer monitors are not based on television picture tubes. What is the amount of memory that is required to

store an image with each of the following display sizes:
(i) 1024 × 768,
(ii) 1280 × 1024?

Derive the time to transmit an image with each type of display assuming a bit rate of
(i) 56 kbps,
(ii) 1.5 Mbps.

2.25 With the aid of a diagram, explain how a digital image produced by a scanner or digital camera is captured and stored within the memory of a computer.

2.26 With the aid of a diagram, explain how a color image is captured within a camera or scanner using each of the following methods:
(i) single image sensor,
(ii) a single image sensor with filters,
(iii) three separate image sensors. Include in your explanations the terms "photosites" and "CCDs" and the role of the readout register.

Section 2.5

2.27 With the aid of a diagram, explain the principle of operation of a PCM speech codec. Include in your diagram the operation of the compressor in the encoder and the expander in the decoder. Use for example purposes 5 bits per sample.

2.28 Identify the main features of the MIDI standard and its associated messages.

Section 2.6

2.29 With the aid of a diagram, explain the principles of interlaced of scanning as used in most TV broadcast applications. Include in your explanation the meaning of the terms "field", "odd scan lines", and "even scan lines". Show the number of scan lines per field with
(i) a 525-line system and
(ii) a 625-line system. Why do computer monitors not use interlaced scanning?

2.30 State and explain the three main properties of a color source that the eye makes of. Hence explain the meaning of the terms "luminance", "chrominance", and "color difference" and how the magnitude of each primary color present in the source is derived from these.

2.31 Why is the chrominance signal transmitted in the form of two color different signals? Identify the color difference signals associated with the NTSC and PAL systems.

2.32 State the meaning of the term "composite video signal" and, with the aid of a diagram, describe how the two color difference signals are transmitted within the same frequency band as that used for the luminance signal.

2.33 Explain why, for digital TV transmission, the three digitized signals used are the luminance and two color difference signals rather than the RGB signals. Why are a number of different digitization formats used?

2.34 With the aid of diagrams, describe the following digitization formats:
(i) 4:2:2,
(ii) 4:2:0,
(iii) SIF,
(iv) CIF,
(v) QCIF,
(vi) S-QCIF.

For each format, state the temporal resolution and the sampling rate used for the luminance and the two color difference signals. Give an example application of each format.

2.35 Derive the bit rate that results from the digitization of a 525-line and a 625-line system using the 4:2:0 digitization format and interlaced scanning. Hence derive the amount of memory required to store a 2-hour movie/ video.

2.36 Explain why modifications to the received (broadcast) TV signal have to be made if the signal is to be displayed in a window of a computer monitor. Hence assuming the SIF format, derive the spatial resolution required with
(i) a 525-line and
(ii) a 625-line system.

Text and image compression

3.1 Introduction

In the previous chapter we described the way the different types of media used in multimedia applications – text, fax, images, speech, audio, and video – are represented in a digital form. We derived the memory and bandwidth requirements for each type and, as we concluded in Section 2.7, in most cases, the bandwidths derived were greater than those that are available with the communication networks over which the related services are provided. In addition, when using a public network in which call charges are based on the duration of a call, considerable cost savings can be made if the volume of information to be transmitted is reduced.

In almost all multimedia applications, therefore, a technique known as **compression** is first applied to the source information prior to its transmission. This is done either to reduce the volume of information to be transmitted – text, fax, and images – or to reduce the bandwidth that is required for its transmission – speech, audio, and video. In this chapter we shall consider a selection of the compression algorithms which are used with text, fax, and images and, in Chapter 4, we shall describe a selection of the compression algorithms that are used with audio and video.