Chapter 2

The Compact Disc

2.1 History of the Compact Disc

CD started life as an audio-only competitor to vinyl discs. Outwardly the main changes were smaller size (hence *compact!*) and the lack of a visible groove. Other features are: data is recorded from the middle outwards, so that the start of the disk can contain index information with reduced risk of it being soiled; recording on the under surface, allowing fancy design work on the top surface; and constant linear replay rate, so that the disk could be used to maximum capacity (in fact 74 minutes, longer than vinyl LPs but still not as long as an old C90 tape). After some early problems with quality control, CDs rapidly came to dominate the mass audio market as well as finding use in studio work where their robust construction and high audio quality tolerated rough handling.

In fact there were other more significant changes which, in due course, made it the interchangeable medium of choice for a variety of computer-related applications. The most important was that the data was digitally-encoded, although the audio CD standard does not guarantee perfect reproduction of the source digital data (error-correction is included to repair any gaps).

There are now numerous formats for differing applications. CD-ROM (Compact Disc – Read Only Memory) has the capacity to store all types of digital information in ways that allow interactive control over the information. One disc can hold around 650MB of data and can be reproduced very cheaply in volume. DVD is the most recent extension of the CD idea: we will deal with that separately a little later. First we summarise the early developments.

2.1.1 CD Standards

CDs evolved and it was therefore necessary for the main patent holders to publish standards, so that many manufacturers could use the technology (and of course pay royalties for doing so). These standards became known as the coloured books because each had a different colur cover.

CD-Audio: Red Book Format Standard The compact disc industry started in 1982 when Sony and Philips created the Compact Disc Digital Audio Standard, commonly known as Red

Book in the trade. This format describes the CDs found in music shops and is the foundation for other compact disc formats. One disc can hold up to 74 minutes of digitally-encoded stereo. The Red Book standard allows the audio to be organized into one or more tracks with each track normally being one song or movement. The tracks are further divided into sectors with a fixed size and duration, though this is not apparent to the user. There is an index at the start (centre), for quick access to the tracks.

The system is designed for continuous replay at constant linear speed (i.e. the rotational speed varies, unlike with vinyl), starting from the middle, with the tracks being read from underneath (i.e. opposite side to the artwork). This is the longest-established CD form.

CD-I: Green Book Format Standard Introduced by Philips in 1986, this was a specialised format called Compact Disc Interactive or CD-I. CD-I titles required a special CD-I player, a stand-alone player connected to a TV set. The CD-I market was limited by the special CD-I development environment and the relatively small number of manufacturers of CD-I players. However, it is historically interesting because the standard allowed for interleaved data; which is the interleaving of pictures, sound, and movies in one track on a disk.

CD-ROM: Yellow Book Format Standard Also published by Sony and Philips, this defines the physical format for storing computer data on the CD-ROM disc. It takes the basic Red Book (audio) definition and defines two new track types – computer data and compressed audio or video/picture data. The first type became the main use and the basis for further development. The format also added better error correction (necessary for computer data) and better random access capabilities (because it was not intended to be played sequentially).

When a CD has data tracks and audio tracks, it is called a Mixed Mode disc. Unfortunately, data cannot be read while sound is playing, so computer applications must use Mixed Mode in a staged manner. To get around these limits and to obtain synchronization of multiple tracks, an interleaving style was added to the format, just like Green Book. So, in 1988, CD-ROM/XA (extended architecture) was proposed by Philips, Sony and Microsoft as an extension to Yellow Book. The CD-ROM/XA drive has to separate the interleaved channels. When it does, it gives the appearance of simultaneous synchronized media. This is used (for example) in Sony Playstation game disks.

CD-R: Orange Book Format Standard The more recently published Orange Book standard defines recordable CD formats. In particular it covers CD-R (Compact Disc Recordable) for compact disc, write-once media and defines the multi-session format. This allows you to add the material in several sessions, rather than having to write the whole disk in one go. Once you have finished writing to the disk, an index is created to finish the recording: after that, you cannot write any more information to it.

CD-R drives have been readily available for some years and prices are low. We lso have CD-E/CD-RW (same thing, two names for a rewritable disk).

2.1.2 Other Standards

As is obvious from the above list, the development of CD has been driven largely by commercial interests, with the effect that the standards are perhaps not as coherent as would ideally be wished. There are other proprietary CD-ROM formats, the common ones being those associated with games machines. These are non-standard, in the formal sense, precisely because each manufacturer wishes to lock consumers into their players and games.

DVD is the main standardised consumer development. This is important enough that we will deal with it separately, later.

Filing system: ISO 9660 While the Yellow Book standard provided a low-level definition of encoded bits on a disc, it was necessary to provide some degree of standard organization so that computer systems could read them in an operating-system-independent manner. This general *file system* for CD-ROM encoding was first proposed in 1985, when it was called the High Sierra format. This proposal was modified slightly to become the ISO 9660 file system recognized by the International Standards Organisation. ISO 9660 standardizes such things as directories, subdirectories, the length of filenames and the characters that can be used in them. It looks most like an MS-DOS file system in design, largely due to the significant part Microsoft played in defining the standard.

Kodak Photo-CD Photo-CD holds multi-resolution colour images, recorded from negative or transparency originals. A transfer service is offered to the public by local film processors. The image is held digitally, in a Kodak-proprietary compressed format. In fact a range of resolutions is included.

Name	Resolution (pixels)	Applications
Base/16	128 x 192	Index print of image (thumbnail view)
Base/4	256 x 384	Displaying rotated image on television (Base image is too large when rotated)
Base	512 x 768	Television or computer monitor display
4 Base	1024 x 1536	High-definition television display (HDTV)
16 Base	2048 x 3072	High res. for 35mm film original highest resolution on a Photo CD Master Disc
64 Base (optional)	4096 x 6144	Full resolution scans of 35 mm and larger format films (for pre press); available only on a Pro Photo CD Master Disc

The highest level is equivalent to 4400 pixels/inch from 35mm film.

Each disk typically holds around 100 images but this depends on resolution and compression. Each image requires about 1 Mbyte for Base-level storage, 3-6 Mbytes for five-level storage. The basic system offers five levels (all except 64 Base).

To ensure proper colour representation, Kodak use their own colour space known as PhotoYCC. There is also now available an encryption and/or watermark scheme, such that only users to whom you have given a key will be able to read your disks. Full motion video cannot be achieved on Photo CD; it is only intended for static images.

Kodak Photo CD format is an example of a *bridge* disc. A bridge CD-ROM can run in either a CD-ROM drive or CD-I drive. The CD-Bridge specification defines a way to put additional information in a CD-ROM/XA track. This lets a CD-I player read a CD-I application from the disc.

2.2 How a CD Player Works

As this was the first compact disk format, we will now examine in detail how a digital disk works. We will take the basic audio CD as the model but the more-recent formats are similar physically. Other mechanisms are used but we describe just one, to cover the technical issues of replay.

A CD consists of a small disk with data encoded on the underneath surface. This consists of microscopic pits, separated by “islands”, starting at the centre of the disk and spiralling outwards. These physical indentations are protected by an optically transparent layer. CDs usually have track, time and index indications, as well as the basic audio data, this being encoded in a header at the start (i.e. near the centre) of the disk.

During playback, the rotational speed of an audio disc decreases from 500 to 200 rpm (revolutions per minute), to maintain a constant scanning speed. CD-ROM discs usually work at higher speeds, to increase the rate of data delivery, which is why you see drives described as 24x (or whatever).

A low-power diode laser is used to read the track. There is no physical contact, so keeping the beam at the correct position on the rotating disk requires some ingenuity. There are three servo mechanisms ensuring correct tracking, together with an ingenious optical system. The laser diode is mounted below the disk and so fires upwards. Its light passes in turn through a diffraction grating, then a beam splitter. The beam continues upwards through a focusing lens system and a polariser (to enhance contrast against scattered light) and then is reflected from the CD. The reflected light passes back through the same optics until it hits the beam splitter, when it is diverted through a cylindrical lens onto an array of six photocells. The original ‘split’ beam also arrives here. The cells are arranged as follows:

```
       [1]
  [2][3]
  [4][5]
[6]
```

So how does this all work? Let’s look at each main component.

Focus of the laser beam

The position of the disc can vary by 1mm because of the physical limitation of the mechanism holding it. The lens is moved by a servo system, to put it at the right distance from the disk. This is done by projecting a circular spot of light onto the disk. When the distance is correct, a circular spot is reflected back to photocells 2, 3, 4 and 5. When the distance is incorrect, the cylindrical lens produces an elliptical spot, which covers 2 and 5 *or* 3 and 4, depending on whether the lens is too close or too far from the disk. The servo system responds by moving the lens system up or down accordingly.

Tracking the disk

The laser also has to be kept on the track being read. Typical CD players use *three-beam scanning* for correct tracking. The diffraction grating splits the beam into three. It shines the middle one exactly on the track, and the two other “control” beams are projected one slightly to the left and one slightly to the right of the track. These two reflect back, one to near photocell 1 and one to near photocell 6. When correctly on track, neither 1 nor 6 “see” their reflected beams. However if the alignment drifts slightly, then one of them will, so a servo moves the lens sideways slightly to compensate.

Reading the track

The laser’s light is reflected from an island to a photo-electric cell (in fact to cells 2, 3, 4 and 5 but we can just pretend there is a single cell from now on). When the beam shines on a pit, reflection still occurs, so we cannot be sure if the beam is on an island or a pit. Either way, the photoelectric cell will see a bright light. However, if the spot is wide enough that half of it is in the pit and half is still on the land, we can get cancellation as follows.

For the pit, the light has to travel two pit depths further (one on the way in and one as it is reflected back out of the pit). The depth of the pit is a quarter wavelength of the laser beam light. Two pit-depths is exactly a half a wavelength. There is thus a half wavelength path difference between the beam reflected from the surface of the disc and the beam reflected from the pit.

If we have a beam which is half in the pit and half on the island, then about half the light reflected will be from the pit, half from the island. These two halves will differ by a half wavelength. Therefore cancellation occurs and the photo-electric cell emits no current. So we use the *change* between island and pit as a logic 1 and a steady state as 0, as we will see.

A digital filter then removes noise. The DAC (Digital to Analogue Converter) converts the digital data to an analogue audio signal.

2.2.1 How Audio CD Works

Audio sample rate: 44.1 kHz

Sample size: 16 bits of audio for each stereo channel

Sampled data rate: 44.1 at 32 kbits/s= 1.4112 Mbits/s

This seemingly straightforward exercise in digital recording is faced with a number of practical problems before a domestic CD can be made reliable.

Problem

16 bits implies 65,536 different audio levels can be recorded. This is more than adequate in general but the quantization error is large enough to be heard when the signal is quiet. This is because the error is fixed, so it is a high proportion of a quiet signal.

(Recall that quantization error is the error introduced by approximating an analogue value with a digital one: the digital representation, being in a fixed number of bits, only has a finite accuracy.)

Solution

Dither the signal: superimpose a high frequency noise signal just large enough to ensure that quiet parts of the signal are forced to cross quantization levels. The result is that even a steady, quiet signal will be represented as a sequence of slightly varying samples, almost as though pulse-width modulation had been used. Interestingly, this also means that a CD can record signals which are quieter than the lowest quantum. Of course we have added a small amount of noise (the dither signal) but the perceptual benefits – reduced distortion – outweigh this.

Problem

CDs get scratches and finger-prints on them; inevitably some small areas become unreadable. If this causes a complete drop-out then the result is all too obvious to the listener.

Solution

Instead of coding the data in strict time order, spread the information across a longer section of the track (and therefore of time). Data can still get lost but a particular disk sample now contributes small amounts to several audio samples, rather than being all of one sample. Any error is thus far less audible.

In practice, CD recording is grouped into logical frames. A frame contains the data for six sample periods; each sample period contains 4 bytes (2 channels for stereo; 16 bits = 2 bytes per sample). One frame thus corresponds to 24 bytes of original audio samples. The bytes are not laid down sequentially in time order; rather they are shuffled in a particular way (using *Cross Interleaved Reed-Solomon Code*: CIRC) to achieve the required spread. Here is an example of naive cross-interleaving:

Original position:	1	2	3	4	5	6	7	8
New position:	3	6	8	2	4	5	1	7

This example shows some places where it works well (2 is next to 4 and 8 in row 2) and some where it is less successful (5 is still next to 4). Here is a better version where no entry has the same neighbour it had before.

Original position:	1	2	3	4	5	6	7	8
New position:	1	4	7	2	5	8	3	6

Using CIRC means that if a short sequence is damaged, the damage may not be audible. Instead of a large error in one place, there are small errors in a few separate sections. If the damage is too great for CIRC, then a further system repairs the damage by estimation from nearby good samples.

Philips claim:

The effectiveness of this system can be demonstrated by pasting a piece of paper several millimetres wide across the CD, and playing the CD. A good correction system will produce normal sound. The CIRC system will first try to correct the signal errors produced by the paper. If the error is too large (if too many bits have dropped out), the system will calculate and insert the missing parts. If this approach fails because of the size of the interruption, the system suppresses disturbance. The music will skip.

I offer the above without warranty!

Problem

We have to maintain constant linear speed of the disk, so the rotational speed must vary. This means the player needs to know the rate at which the data is flying past the laser, so that it can stay locked to the correct sampling rate and make the necessary adjustments to the rotational speed of the disk. If we recorded the signal directly (pit = 1, island = 0 say) then the laser beam could not distinguish a long series of zeroes from a blank disk (and the same is true of a long series of ones) and we would have no way of doing this.

Solution

Code the data such that the *change* from a pit to an island (or vice versa) on the surface of the disk corresponds to a logical 1. The *length* of the pit or island corresponds to the number of zeroes. Note carefully: zeroes are represented by pits or islands; ones are represented by the cliffs between the pits and islands. Furthermore, encode in such a way that there are never more than 10 zeroes in succession and that every sample has at least one transition. The scheme guarantees a maximum time (10 bits) before a transition occurs, which means that we can use a reference oscillator to check the linear speed of the disk and adjust it as necessary. When we get a transition from the disk, we expect it to match a "tick" from the oscillator. If it is a bit early, the disk is rotating too quickly; if late, too slowly. So we can adjust the rotational speed to keep it at the correct speed. This means we need a coding scheme which guarantees the transition: this comes about when solving the next Problem.

Problem

A CD track is purely sequential, yet we have to record 32 bits (stereo) for each of 44.1 ksamples/s. If the signal happened to change bit by bit, the pits and the lands between them would be very small on the disk, making manufacture harder.

Solution

First, consider the whole process in abstract. We have a physical disk, which delivers its data as a stream of *bits*, which in due course are converted into *bytes*, which are then converted into *sound*. In order to guarantee continuous sound, the rate at which the bytes are delivered to the digital-to-analogue convertor has to be constant. In other words, we can determine the time interval in which each byte has to be delivered (it does not matter what it is, though we can in fact calculate it from the Red Book).

Within this same time interval, the disk will have delivered a number of bits but there is no reason for this number to be 8: it could be more. In fact, CD delivers 17 bits in that time. Three of these bits carry no information about the audio, so that leaves 14 bits in which to encode the 8 bits of useful data.

[NOT EXAMINED: For the curious, the additional 3 bits include 2 bits to ensure that successive samples obey the the rule below; and 1 parity-like bit to allow an additional transition if needed, to ensure that the dc level of the recovered signal is zero, regardless of the recorded data, keeping the player's electronics simple.]

Therefore we call this *8-to-14 modulation (EFM)*. As the name suggests, each 8-bit byte is encoded as a 14 bit pattern. These patterns are chosen such that there are always at least two and at most 10 zeroes (see above for the reason for this 10) in succession. 14 bits is the smallest number of bits to give at least 256 codes with these properties (in fact it gives 267 codes). Although we have expanded the data from 8 to 14 bits, the encoding used ensures that the size and rate at which the pits change is lower.

To see why this is so, let me first make the calculation simpler. Suppose we were to use 8-to-16 modulation. In the time that the audio needs 8 bits, we have 16 bits arriving from the disk. So, if the 8 bit bytes change every bit, we can determine the highest digital frequency to be f say. If every bit of our 16 bits changes in the same time interval, then this would require a frequency of $2f$, which of course is higher than f. We are trying to make it lower. But if every interval with a one is required to have two intervals of zero after it, then the maximum frequency is one change every 3 intervals. So the maximum frequency will be $2f/3$, which is lower than f. This is why we insist that our encoding method has a shortest run of zeroes of 2 bits (and never has runs of 1s).

In fact we only need 14 bits to guarantee that we have 256 codes (one for each possible byte value) which meet the above conditions, so we actually use EFM and the frequency is therefore a little lower still ($14f/(8 \times 3)$).

If you are not comfortable with calculating frequencies, or do not find it helpful, then just consider this: 100100100... gives longer pits than 101010... and so these must be easier for the laser to "see"!

2.2.2 Writable CDs

Writable CD do not have pits. Instead they are manufactured with a "pre-groove", which is simply a groove pattern that the optical system can use to track. The active surface is dye-coated and the die absorbs energy at the frequency of a relatively high power laser used to write the disk. The die gets hot, deforms and leaves pit-like marks. The pits can be read by a laser on a lower power setting, because of the different absorption. The process is not reversible.

2.2.3 Rewritable CDs and DVDs

Rewritable CD (CD-RW) became available in 1997, supported by Philips, Ricoh, Sony, Yamahah and Hewlett-Packard. These discs also have a pre-groove and use a higher power laser. The discs are coated with a metal alloy (silver, indium, antimony and tellurium) which is in a crystalline state and reflects light well. The laser generates a spot hot enough to melt the alloy locally. It cools too quickly to regain its crystalline form however and so does not reflect light as well. This difference is what allows binary data to be recorded. However, if the laser heats the spot to a lower temperature, it is possible to regain the crystalline state. In effect, the disk has been erased. This can be done around a 1000 times. As with ordinary CDs, there is a protective layer covering this metal layer. Rewritable DVDs use the same method.

In practice these disks did not read reliably on early CD players. This is because conventional CDs reflect around 70% of light from the surface, 30% from the pits. Rewritable CDs manage 20% from the crystalline state and only 5% from the amorphous state and so needs a much more sensitive sensor. Later players incorporate an automatic gain control which provides additional amplification when needed by CD-RW discs. A revised standard has been agreed to allow this.

2.2.4 Mini disk

Mini disk is a physically smaller disk, designed to be recorded and edited. It appeared long ago, as a replacement for compact tape. It did not originally do well here but found a strong market in Japan. After a slow start, it now has a market in the West, for applications such as audio 'Walkmans', where its physically-small size is important. The disks are cheap and you can make a digital to digital copy from a CD, though with some loss of quality. Newer solid-state devices are now replacing this technology.

It uses Sony's own selective perceptual compression technique (Adaptive Transform Acoustic Coding: ATRAC) to achieve as much time as ordinary audio CD. Audio quality is not as good as CD but you can erase and re-use any section of the CD. Of course this is true of tape too but the Mini disk is more robust and permits editing, not just direct over-recording. This is achieved by having a Table of Contents, which is kept up to date to show which parts of the disk contain the information for any given track.

Philips and Sony developed the technology of magneto-optical recording. Recording is achieved by sandwiching the disk between a magnetic recording head (above the disk) and a laser (below). It is thus similar to CD-RW in relying on a phase change. However this is brought about as a magnetic realignment of the particles of material. The laser's job is to heat the material to just above the point where the magnetism is lost. The electromagnet's job is to align the particles in one of two directions as the material cools. Hence it can operate at lower spot temperatures (about 200 degrees C instead of 600 for CD-RW). The laser is used at lower power to distinguish optically between these two states. Strong magnetic fields can corrupt the disk.

2.3 Digital Versatile Disk

(The disk formerly known as Digital Video Disk)

How is DVD different from CD? For greater data density, there are smaller pits (half the length), a more closely-spaced track (half the spacing) and a shorter-wavelength red laser. The error correction is more robust. The modulation scheme is more efficient.

Hence a standard DVD can hold 4.7 gigabytes of data: seven times the data capacity of a Compact Disc. It is also possible to make dual-layer DVDs. This is achieved by the player having a focus control which is selective in depth. They can hold more than twelve times the information of a CD, without the need to turn the disc over.

Using MPEG-2 compression, a single-layer DVD can hold movies over two hours long – with room for Dolby Digital sound in three languages. Dual-layer discs can hold movies about four hours long.

The quality of the movie image depends on the MPEG coder used to create the disk. Problems with tones blocking up can sometimes be seen and are a symptom of poor coding producing too low a bit rate.

Data transfer rates are higher than even the fastest current CD-ROM. The drives can play existing CD-ROMs. Both DVD-Write Once and DVD-Rewritable are available.

Here is some technical data to give you a feel for the technology. However you don't need to memorise this, except that it is useful to remember the basic capacity (4.7 gigabytes) in comparison to that of CD-ROM (650 Mbytes).

Physical Characteristics (DVD5	
Disc Diameter	120 mm
Disc Thickness	1.2 mm
Track Pitch	0.74 um
Minimum pit length	0.40 um
Wavelength of Laser	635/650 nm
Linear Speed	4.0 m/s
Data Capacity	4.7/8.5 gigabytes

Video Specification	
Data Transfer Rate	Variable-speed at an average rate of 4.69 megabits per second (10.7 megabits per second maximum)
Image Compression	MPEG-2 standard
Audio Standard	Dolby Digital (AC-3) (United States); MPEG Musicam (Europe, in principle!)
Number of Channels	Upto 8 audio channels and 32 subtitle channels
Running Time	133 minutes per side with 3 audio channels and 4 subtitle channels

Physical Disc Configurations	
DVD5	Playback-only, single layer, 4.7 GB (133 minutes) capacity
DVD9	Playback-only, dual layer, 8.5 GB (240 minutes) capacity
DVD10	Playback-only, single layer, double-sided, 9.4 GB (266 minutes) capacity
DVD18	Playback-only, dual layer, double-sided, 17 GB (481 minutes) capacity
DVD-R	Record-once, organic dye process, double-sided, 7.6 GB (215 minutes) capacity
DVD-RAM	Record-many, phase change process, double-sided, 5.2 GB (147 minutes) capacity

In everyday use, single layer (DVD5) and double layer (DVD9) disks are common for the prerecorded films you buy in high street shops. DVD-R disks are very cheap and readily available for computer use.

As its name DVD suggests, DVD stores all information digitally. The component format stores pictures in 3 separate channels: the black-and-white image (luminance) and two channels of colour information (chrominance). DVD stores pictures at 720 by 480 lines of digital resolution (3:2 aspect ratio: other aspect ratios are also offered), which exceeds the old laserdisc.

European TVs have long accepted either component or composite video signals, so getting full quality was never a problem. In America, early DVD players had to downgrade these high-quality signals into an NTSC S-video signal or, worse, into an NTSC composite signal or worse still, into radio frequency modulation to mimic a broadcast signal. In fact most early sales were of DVD drives for PCs. Christmas 2000 was the first year in which domestic stand-alone players sold very heavily in the UK. Christmas 2005 was the first year in which some major UK high street shops ceased stocking video tape recorders.

The home DVD player is backward compatible with CD (Compact Disc) and the computer DVD-ROM player is backward compatible with CD and CD-ROM standards.

2.3.1 High definition DVD

Standard DVD players use a red laser and the frequency of this limits the capacity of the disk. More recently, blue (higher frequency) lasers have become available at a reasonable price. Naturally there has been interest in developing a higher-density disk format, with the possibility of higher resolution pictures or greater storage as a ROM format.

In fact several proposals were floated and two of them, Toshiba-backed HD-DVD and Sony-backed Blu-Ray, were signing up Hollywood studios. Most studios initially opted for HD-DVD, which was first out of the blocks by a few months, but Disney and then others chose Blu-Ray. Sony put a blu-ray player in their games machine and thereby achieved more sales of players. After several years of hype but no disks, players are now around, TVs and projectors with the required resolution are also around and disks are in the high street shops. It now seems TV is driving the sales of hi-def, more than DVDs, though there are signs that consumers do not fully appreciate how much better hi-def really is. The need to define the market means there is intense pressure to have products as soon as possible. The format war ended in February 2008, when Toshiba announced they were pulling out of HD-DVD, leaving Blu-ray as the default standard.

HD-DVDs could be produced by relatively modest updates to existing DVD plants. Blu-ray requires bigger investment, so initially appeared less attractive to manufacturers of pre-recorded disks. However, Hollywood and its supply chain were not the only influences on DVD formats. PC users will find the capacity of Blu-ray DVD ROMs is greater than HD-DVD ROMs and the Sony Playstation 3 uses Blu-Ray disks.

2.3.2 Audio-only DVD

DVD can offer upto 24 bits of audio resolution, which is beyond the full dynamic range of the human ear (unlike CD, with 16 bits). Naturally there has been interest in developing the DVD as a kind of super CD. Unfortunately, two competing versions have appeared. These are called DVD-Audio and Super Audio CD (SA-CD).

Sony and Philips offer SA-CD, which is a two layer hybrid disk. One layer contains a conventional encoding, so it will play on existing equipment. The other layer contains the high definition version. This uses a completely different encoding (delta-sigma modulation, as 1-bit direct stream digital (DSD)).

The other suppliers offer DVD-Audio, which can encode 16/20/24 bit sound, up to 96 kHz, using PCM just as with CD. In principle they too could offer a second layer with an ‘old’ style CD version but the patents for this are held by Sony and Philips.

Neither seems to be finding a major market demand.

2.4 MPEG

MPEG is the way of compressing digital movies used by DVDs. Even with the capacity of a DVD, compression is still necessary. In fact, going from analogue to digital typically *increases* the bandwidth needed by a factor of 10. Compression by a factor of 10 is therefore necessary just to get back where we started. We want to do better than this. MPEG permits compression by about 20:1. It can however be made to give higher compression, by trading off picture quality as we will see.

MPEG arose from an informal process to become a Draft standard for digitally-encoded movies in late 1990. It has evolved in a controlled way and MPEG-2 is the current result. We will just use the term “MPEG” to mean MPEG-2.

MPEG is not strictly a complete standard, rather it is a generic standard, independent of a particular application. What is specified is the format and the meaning of the compressed stream, rather than how you are to go about compressing it. This is all you need to decompress it. Thus one company might produce a better MPEG system than another. As a result, you can design a decompressor without knowing in detail how the compressor worked.

The original system, now called MPEG-1, had no support for sound and worked on a field-by-field basis, rather than frame-by-frame. The typical result is only half the resolution of standard TV.

MPEG-2 adds sound and frame-by-frame full TV resolution, as well as various aspect ratios. MPEG-2 decoders will also decode MPEG-1.

MPEG claims to support:

- random access to any frame
- fast forward/reverse
- reverse playback
- audio-visual synchronisation
- robustness to errors
- short decoding delay (in fact asymmetric: slow to encode)
- editable
- format independent (raster size, frame rate)
- small chip-set

2.4.1 Overview of MPEG coding

MPEG achieves high compression by working on two levels: image compression of each frame (exploiting *spatial redundancy*) and motion-compensated compression (exploiting *temporal redundancy* between frames). The video thus goes through a number of processes, as follows. (The first two are not needed if the camera is digital or the images are computer generated.)

1. Analogue video from camera, video tape, film etc.
2. Analogue to digital conversion.
3. Inter-frame compression: frame-to-frame encoding to exploit *temporal coherence.*
4. Intra-frame compression: within-frame encoding to exploit *spatial coherence.*

To do the compression, MPEG makes use of the Discrete Cosine Transform (DCT), quantisation (to fewer bits) and variable length coding (to ensure that the most common features are encoded with fewer bits). (We will explain DCT in detail, later.)

MPEG-2 organises the picture data into *blocks* and *macroblocks*, so let's explain those terms before we go any further. A block consists of 8×8 samples of luminance plus 4×4 samples of chroma. That is, there are 2×2 luminance samples associated with each chroma sample. The luminance (brightness) information is more important to the human eye than the chroma (colour) information because we can resolve finer detail in luminance that we can in chroma.

A macroblock consists of four blocks (i.e. 2×2 blocks) and thus represents 16 lines of 16 pixels. The macroblocks are the basis of the compression and these are the blocks that you sometimes see on screen when the compression is poor or when data goes missing in digital TV transmission. The macroblocks used to reconstruct the frame are arranged in a grid over the whole frame.

Let's look at the two forms of compression (time and space) at the heart of MPEG. Compression is about *coherence*: finding something which stays much the same, so we can record the data once but use it several times over. We can assume that each frame is just an array of pixels to start with; our job is to encode that in a more compact way. Note that this means we can read any of the pixels we choose; we don't have to confine ourselves to the pixels in a given macroblock but we do have to output a coded form of each macroblock.

2.4.2 Temporal compression

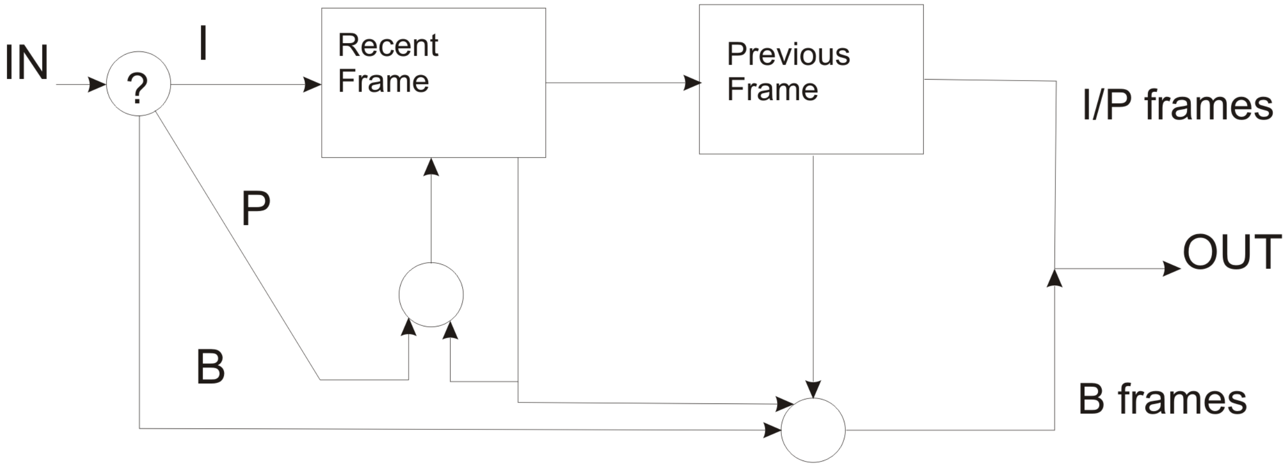

Figure 2.1: MPEG decoder

By "temporal", we mean compression between frames which are adjacent in time, exploiting the fact that some parts of the second frame will look very similar to some parts of the first frame. We use the term *temporal coherence* for this idea.

The basis is the *macroblock* of 16×16 pixels. What MPEG does is to look at two adjacent frames and try to encode only what has changed. This is called *interframe* coding. Once this is complete, MPEG will perform spatial compression, within the blocks (i.e. 8×8 pixels). This *intraframe* coding will be explained in a later section; for now, we concentrate only on the temporal coding.

Call two time-adjacent frames the "new" frame and the "old" frame, where "old" appears first in time order. For each macroblock in the new frame, we try to find a similar region in the old frame. This previous region is 16×16 pixels but it does not have to be aligned with the macroblocks of the old image: it can be centred on any pixel because the decoder will have access to all of the pixels of the old frame. Suppose it finds a good match. Since the decoder already has all of the old picture, the MPEG encoder sends a reference to that old macroblock instead of sending the whole macroblock. In fact all it has to do is send the coordinates of the first pixel in the old macroblock. We thus get compression.

Prediction is made independently for each macroblock of the new picture. If we find a satisfactory match, we send the (x, y) translation between the position in the new picture and the position in the old picture. These (x, y) pairs are called *translation vectors*. In this way, the decoder can start with the picture of frame1, which we can assume it successfully decoded into pixels. It then receives frame2 encoded as translation vectors. It uses these vectors to fill its macro-blocks from frame1, essentially just cutting and pasting from the array of pixels.

Any macroblocks without a good match are spatially compressed (see next section) and sent in their entirety. We get compression because it takes far less data to send (x, y) than it does to send a complete 16×16 block of pixels. Furthermore, this means that spatial compression will only have to be applied to the remaining blocks.

When the image is moving, the close match is likely to be nearby but not at exactly the same place, which is why we need translation vectors. For the same reason this method is called *motion-compensated* prediction. So MPEG achieves temporal compression by sending references to blocks in an earlier frame.

Where to look for a match is not defined, so this is an area where the implementor has freedom. We also need to know when we have a good match, so we need a measure of error. Again, this is not defined by the MPEG standard. Here is a commonly used error measure.

$$Error(dx, dy) = \sum_{i=0}^{15} \sum_{j=0}^{15} \mid f(i, j) - g(i - dx, j - dy) \mid$$

Here, f represents the macroblock from the new picture and g is the corresponding macroblock in the previous picture. The displacement of the reference macroblock is (dx, dy), measured in pixels.

In plain English, this formula says measure the error as the sum of the differences of corresponding pixel values in f and g, over a 16×16 area. There are thus 256 values to be calculated each time the error measure is made. Doing this for lots of positions can be slow but note this is part of the coding, not the decoding. For creating DVD masters, we do not need to work in real time and can tweak for best quality. For live TV, we can't do this.

Now we can try to find good matches. All we are really trying to do is find a 16×16 block which looks very similar to the correct one. One approach is brute-force: compare the macroblock from the new picture against all positions of a block in the old picture; compute the error measure in each case. Choose the best fit (i.e. least error) and output its translation vector. It isn't feasible to do this for a whole movie; there would be far too many comparisons.

To reduce the computational load, a Three-Step-Search (TSS) is sometimes used. In the original version of this algorithm, the evaluation was performed at the centre (i.e. the same position in the old frame as the block we are trying to match in the new frame) and eight other positions. The eight positions are those which are located three pixels to the north, south, east, west and diagonally. The position that produces the smallest error then becomes the centre of the next stage, but now eight positions two pixels away are used. The method is repeated a third time, now looking only one pixel away. The position of this best match gives the required translation.

A TSS variant starts with a 32×32 search area. The eight positions are half way to the edge of the 32×32, in the same directions as before. That is, the test points are eight pixels horizontally and/or vertically from the centre pixel. The algorithm proceeds in much the same way except that the range is halved each time (rather than reducing by one). The next test points are thus four pixels away, then two. After this third step a single position has been identified and this defines the translation, as with the first method.

What both methods assume is that a closely-matching block is likely to be found nearby (because any movement will not be great). The error measure tells us how good the match is, so we have the option of rejecting the best match found if we think it is too poor. Instead of sending a translation vector for that block, we send the macro-block itself. In fact we use DCT and variable-length coding on the macro-block, the same method used for spatial encoding, so even this is compressed. Of course we are making this decision block by block in a given frame, so some macro-blocks will be sent as translation vectors, some as compressed macro-blocks.

In fact MPEG improves on the reconstructed visual quality of translation vector blocks by also transmitting the difference between the new macro-block and the old area pointed-to by the vector. This correction will be added to the new macroblock. For a good choice of translation, the image difference (error) will be small, so it is possible to encode it with relatively little data. The difference image is once again encoded with DCT and variable-length coding, to compress further.

2.4.3 Spatial compression

By "spatial", we mean compression within a single frame. Typically we would exploit the fact that some parts of the frame will look very similar to other parts. We use the term *spatial coherence* to express this idea. As we will see however, we need to be more subtle for movie frames.

In computer graphics, especially with block diagrams or flat-coloured cartoons, run-length encoding (RLE) is simple and effective. RLE is any of various methods using the same core idea: each item is a record of two values, namely the data value and the number of consecutive samples (the run-length) which have that data value. Although the run-value pair requires more bits than a single value, the average amount of data will be reduced if the data has spatial coherence, meaning simply that the next value is likely to be the same as the previous one.

However, with live-action images, such as one frame of a movie, RLE performs very badly. Typically every pixel is different from its neighbours. Even if the picture shows a flat, single-coloured surface, there will be subtle variation in lighting, texture and even electronic noise. These characteristics prevent suitably large run-lengths. In fact in most live action pictures, run-length encoding would take more data than the raw pixel values. Since we are trying not only to reduce the data but also to contain it within the fixed capacity of a DVD disk, RLE would be a poor choice for the image coding. It does have a particular use in MPEG however, as we will see later.

So, how are we to compress a single frame where every pixel is different to every other? To address this, we need to find a method which exploits, not pixel to pixel similarity, but some feature of the whole image. The key to this is to use a *perceptual encoding* scheme, one which exploits the limitations of the human eye. We can reduce the information in a frame without the eye realising; this will be a lossy scheme.

We have already met this idea when talking about audio CD. There we filtered out high audio frequencies – above the Nyquist limit – to ensure the reconstruction produced no aliasing. We can extend this idea to filtering images, to remove high spatial frequencies that we do not easily see. We have to do this filtering anyway when we digitise an image, for exactly the same reason we do it in audio: if we don't, we get visual aliasing in the form of Moiré patterns. Now we will choose to filter more than that.

Something very important emerges from this. If we work with spatial frequencies, instead of pixel values, we have a representation which applies to the whole image. This allow us to reduce or omit frequencies which the eye cannot easily see. The key is thus working with spatial frequencies. This is where the discrete cosine transform comes in.

2.4.4 The Discrete Cosine Transform

Fourier theory tells us that any repeating waveform can be represented as a sum of basic sine (or cosine) waves of various frequencies. In particular, if we analyse a waveform to find the amplitude of its component sine waves, then we can reconstruct the original. Typically this requires an infinite sum of sine waves of ever-increasing frequency. However, we can get an approximation to the original by using just the first few terms: the fewer terms, the worse the approximation but the less information we need to encode it.

This is why DCT is useful for perceptual encoding. We can choose just enough terms for the reconstructed picture to be good enough for the human eye. The practical consequence of leaving out the higher frequency terms is that edges are not quite as sharp: the picture will “soften”. But for a given image size and viewing distance, the eye cannot see detail finer than some limit anyway, so you won’t notice. In movies, we can sometimes be more brutal in dropping the high frequencies: you won’t be able to see detail on a fast moving object.

For MPEG-2 spatial compression, DCT is applied to the blocks, each of 8 × 8 pixels. Recall that the whole frame is divided into blocks.

In general terms, the DCT computes 64 coefficients, from the block pixel values. These values are measures of different *spatial frequencies* in the block: high frequency corresponds to small detail and can typically be left out if maximum compression is needed.

Although DCT works in two dimensions, x and y, it is a feature of the DCT that each dimension is independent of the other. Let’s make it easier by showing the DCT in one dimension, for x is 0 to $N-1$ (we will want $N=8$ for MPEG-2). [THIS FORMULA IS NOT EXAMINED]

$$C(u) = \sum_{x=0}^{N-1} f(x) \cos\left(\frac{(2x+1)u\pi}{2N}\right)$$

The C values are the requisite DCT coefficients of pixel values, f. They tell us the amount (amplitude) of each frequency that is needed to make the picture.

If we fix the value of u, we calculate a single value of the transform. This will depend on all the pixel values in the block. If we choose another u, then we will get a different answer, still dependent on all the pixel values in that row of the block. In fact the transform is defined for any real number u: this is the coordinate in frequency space (note how it affects the frequency of the cosine term). However we are only interested in the discrete frequencies which our pixels can generate; that is, in the integer values 0 to 7 in the MPEG example, so we calculate these 8 values.

Let’s tease this out a bit further. For MPEG we have $N=8$. We will stay with one dimension, x say. Imagine a row of 8 pixel values, representing the brightness of a picture and sampled from a real (continuous) scene by our digitising process. We have something like the following:

```
brightness
     ^
     |    o         o
     |      o     o
     |  o     o o
     |
     |o
     --------------------->
      0 1 2 3 4 5 6 7     x
```

Of course the original waveform would have been continuously changing across all values of x, not just at the integer positions, but we have now digitised it, so we only have these 8 values.

Our MPEG-specific formula becomes:

$$C(u) = \sum_{x=0}^{7} f(x) \cos\left(\frac{(2x+1)u\pi}{16}\right)$$

So, if I give you the above 8 brightness values $f(0), f(1), \ldots, f(7)$ and tell you a value for u, you can evaluate the above formula to get $C(u)$. Note what we just said: to get one value of

C you have to know all 8 values of brightness. Now the values of u that I am interested in are in fact $0, 1, 2, \ldots, 7$; producing $C(0)$, $C(1)$, $C(2)$, $\ldots$, $C(7)$. I'll tell you why in a minute: just understand that for each of these different $C(u)$ I will still need all 8 pixel values. If I change a single pixel value, then all values of C will change, because the frequencies will have changed.

In fact, there is a very similar formula to tell us f (the picture), given values of C (the DCT coefficients). So the reverse is also true: if any value of C changes, then all values of f will change.

So much for the mechanics. This still does not give you much insight into what is happening. Let's have a go at that. Concentrate on the cosine term in our formula. What is the effect of varying u? To simplify the formula, we can replace everything except the u with $A(x)$, giving us the much more digestible:

$$\cos(A(x).u)$$

If we ask what this evaluates to as u varies, we get:

u	
0	1
1	$\cos(A(x).1)$
2	$\cos(A(x).2)$
3	$\cos(A(x).3)$
4	$\cos(A(x).4)$
5	$\cos(A(x).5)$
6	$\cos(A(x).6)$
7	$\cos(A(x).7)$

This makes it clear that these are cosines of linearly increasing frequency. Our formula requires us to use just one of these frequencies to calculate one value of C; and it is u which determines which C and hence which frequency. But we always use all 8 pixels.

An analogy might make this clearer; let's work with sound. If you play a loud sound at a single frequency, you can sometimes get a sympathetic vibration from a nearby object. A wine glass might "sing" for example, or a window pane might rattle. What that tells you is that the sound is hitting a natural resonant frequency of the object. What DCT does is "play" a series of spatial frequencies (the cosines) and look for the spatial resonances in the picture. A high value of the transform $C(u)$ says there is a lot of frequency u present.

In the formula, the frequency is multiplied by $f(x)$ (to be more precise, by the 8 pixel values standing in for the continuous $f(x)$). Now what that means is, if the picture strongly varies in brightness at the *same* frequency, then we will get a large value of C: the picture brightness variation and the cosine frequency will be much the same everywhere and so will multiply together and add up to give a big result. If the picture hardly varies at all *at that particular frequency*, then we will get a much lower value of C: the picture brightness will vary largely independently of the cosine frequency, so some bits will add up and some bits will be subtracted.

This is the key to whole transform game: it says we can identify how much of that frequency is present. Of course, if we can do it for one value of frequency, we can do it for more: so we in fact do it for all 8 frequencies and the resulting 8 coefficients, $C(0 - 7)$, tell us how much of each of those 8 frequencies are present. We can also do the sums backwards, as it were, to deduce the pixel values from the C values.

This is why you need all 8 pixel values for each C value; and vice versa. It is also why 8 frequencies are enough: if we go any higher, we are looking for brightness variation which is more closely-spaced that the pixels allow.

Are you wondering why the first 'frequency' was just a constant value, 1? This came from $\cos 0$. It must be telling us the amount of zero frequency, because $u = 0$. So it determines how much the brightness is constant across all 8 pixels. Put more directly, if all 8 pixels are of value 3 (say), then the constant component is 3. The higher components will be zero in this case (because there is no change in brightness). In general, this first coefficient is the average brightness of the 8 pixels and the higher coefficients say how much variation takes place about this average.

Now when we move into 2D, there is very little new. We can have vertical spatial frequencies as well as horizontal ones but otherwise the argument is exactly the same. The 2D transform creates $C(u, v)$ from $f(x, y)$. As the two dimensions are independent, the formula has a double sum on the right (one for x and one for y) and two cosine functions (one for x and one for y). It will thus generate $8 \times 8 = 64$ values of $C(u, v)$ when $N = 8$.

$$C(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

NOT EXAMINED: If you are familiar with Fourier theory, you probably know that the use of periodic sines/cosines requires us to assume that the signal being encoded repeats forever. The same is true of our DCT. However the Fourier transform operates on the basis that we simply repeat the interval of interest. This means that the resulting infinitely-long signal will usually have discontinuities at every interval boundary. In turn this means we will need lots of high frequencies to represent it properly. In the form that we use it, the DCT assumes that these repeats are *reflections* of the signal in the original sample. In other words, the infinite signal is guaranteed to be continuous. This reduces the high frequencies needed to represent it. Finally we note that the use of cosine (rather than sine) ensures that the first term is the DC (average) component, the one of zero frequency.

Compression

All we have done so far is replace every pixel value with a DCT coefficient: this is the same amount of data, so no compression.

MPEG therefore uses different numbers of bits for each coefficient, so it can use fewer bits for the high-frequency information, the *variable length encoding* mentioned. This means coarser steps for the less visible information. Now we have some compression.

We can however do better than that. The eye's sensitivity to noise varies with frequency. We have just decided that higher frequencies are more coarsely represented. A useful side-effect is that this will usually increase the number of zero coefficients when we quantize: small values will round to zero. Small values tend to be associated with the high spatial frequencies, it is only the first few low frequency terms that have big coefficients. MPEG arranges the coefficients in increasing frequency order (which is done by reading the matrix of coefficients diagonally), so the zeroes are likely to occur in runs. MPEG therefore uses an RLE-style method to encode these zeroes. The method uses *run-value* pairs: the *run* is the number of zeroes found before the next non-zero value appears; and *value* is its value. This gives further compression.

Finally we could deliberately increase the compression further, by choosing to force all the frequencies above some level to zero, degrading the picture further. We might do this if, for example, we know the movie will be delivered over the internet and we are more interested in high compression than good visual quality.

The reconstruction process is straightforward: it does not need to know anything other than the coefficients to generate the pixels.

We have now covered both spatial and temporal compression. How are these related to an MPEG movie, with many frames?

2.4.5 The MPEG picture types

The requirement for random access means that MPEG cannot simply compress a complete sequence to the point where frames cannot be identified and reconstructed quickly. Three types of picture are therefore used:

- Intrapictures (I). These provide random access points and are encoded to ensure accurate reconstruction. They are compressed only spatially (intra-frame; hence the name) and will generate a high quality image.
- Predicted pictures (P). These are coded with respect to the most recent I or P picture. They use both spatial and temporal compression, typically being three times more compressed than an I picture.
- Bidirectional pictures (B). These too are spatially and temporally compressed, typically being more compressed than a P picture. They differ from P pictures in that each translation vector can refer either to the previous reference frame or to the next one. These are never used as a reference for other pictures.

B pictures will be the lowest quality because they are created from a pair of pictures, I or P. The prediction in a B picture may be forward, backward, or a combination of the two (selected in the macroblock layer). If the original reference pair are P pictures, the resulting B pictures will be lower quality than either: in effect, they are predicted from predicted pictures. Even if the original pair are both I pictures we usually construct several B pictures in between the same pair, whereas a P picture is a single picture constructed from one original picture. A more subtle point is that once the two reference pictures have been constructed, the B pictures in between have no choice but to use them: there is no way to select "better" references. It is this lower quality which means we never use them as reference pictures.

However, using B pictures allows us to have a sequence in which something is revealed for the first time. Imagine a van covering part of the image at the start of a sequence. It moves off, revealing our heroine on the far side of the road. There is no image data available to allow us to construct her from the earlier frames. However she appears in the following frames, so we can use translation vectors from those. The van however is disappearing, so we can use translation vectors to earlier frames for the part of the image where it is still visible.

2.4.6 The MPEG picture stream

Precisely when to use I, B or P is not defined, as this is likely to be application-specific. For example, if editing and random access are not issues, then the application could use fewer I-type pictures in a given stream.

A simple MPEG encoder will use a repeating group of pictures (GOP). Here is a (realistic) example, with the frames shown in the order they appear in the movie:

```
I B B B P B B B I
```

We start with a good-quality I picture, use it to predict the P picture, and the decoder holds on to both these frames. Then it uses the vectors etc to interpolate these two to get the three intervening B pictures. For the second half of the sequence we use the P picture and the final I picture to interpolate the second group of three B pictures. If we assume this pattern repeats across the movie, we will average 8 frames of output for the cost of one spatially-compressed I picture and enough vector information to predict one P picture and six B pictures.

This use of B pictures means that the frames in an MPEG stream are not necessarily in time order. For the receiver (a DVD player, for example) to generate B pictures, it needs the previous

I/P as well as the future I/P; so these must both be sent *before* the B data for the frames which fall between them. So the receiver has to be able to hold in its own buffers two frames. The frames are therefore re-ordered according to their dependencies, such that we always send an I or P picture before the other pictures which depend on it.

Let's think about the time order in which the above frames would be sent. We know we want the outputs to appear in the above order. First we send the I frame. We next have to send the P frame, otherwise we do not the data to work out what the B frames look like.

```
I P B B B ...
```

The receiver loads the first I frame into buffer-1; then it uses incoming information about the P frame to construct the P frame in buffer-2. Now it can generate and immediately output the first three B frames from these. Then it can output the P frame.

Before it can generate any more, it needs the second I frame and must also hold on to the P frame. So the P frame data becomes buffer-1 and the next thing sent must be the I frame, which goes in to buffer-2. Now the receiver can generate the remaining three B frames as their data arrives:

```
I P B B B I B B B
```

Finally it can output the second I frame.

2.4.7 Summary of MPEG encoding

The easiest way to think about the whole process is to imagine the encoder has access to all the pixels of all the frames. It then decides how often to send I pictures, P pictures and B pictures (this may be a dynamic decision based on the nature of the images). Then it calculates the required translation vectors to generate the P pictures. Finally it compresses whichever images it is really going to send, with the discrete cosine transform.

What gets sent is therefore the DCT-compressed I images and the relevant translation vectors for P and B images in the time-order which allows reconstruction to take place.

So why do we need P pictures? After all, you can think of a P picture as a special case of a B picture in which all the vectors happen to refer to an earlier picture. The reason is that the decoder holds on to the two most recent I or P pictures, so that it can decode any P and B pictures when they arrive. It never holds on to B pictures (they are poor quality) but outputs them immediately. The purpose of B pictures is therefore to permit several frames to be generated without the need to update the two picture buffers. A P picture (or of course an I) is sent *exactly when* we want to make such an update, which will typically be when we have generated all the B pictures that the current buffered pair of images can sensibly support. We send a P when the quality will be good enough; otherwise we send an I.

Advantages and disadvantages

These techniques give a number of advantages. For examples:

- the use of future frames makes it possible to 'predict' a newly-uncovered background.
- error propagation is curtailed by inserting I pictures
- you can trade-off compression (more B pictures) against quality (more I/P pictures); and you can do this second by second. (Typical systems might have a P picture about every 0.1s, I every 0.4s, so IBBPBB etc)
- can use all I pictures, for high quality if need be.

A disadvantage is that decoders cannot work in time sequence (B pictures are produced from a forward and a backward picture), increasing their complexity. This doesn't matter much when running a movie (the picture stream arrives in the order the decoder needs), although it does require a second buffer. However it makes finding a particular frame more complicated.

2.4.8 MPEG Performance

DVD is limited to an average bandwidth of 4.69 MBytes per second, so MPEG-2 must work hard to stay within this. It must work even harder on CD-ROM and other media, where rates of under 0.2 MBytes per second are used.

An extreme MPEG-2 compression may result in combinations of the following artifacts:

- Lack of detail (jagged or dull edges)
- Mosaic image patterns (pixelation)
- Blockiness in the shadows
- Bouncy (or jumpy) moving images
- Losing some important frames

Pictures with large areas of constant colour, such as cartoon films, may show such artifacts if they are compressed with an MPEG designed for live action films.

2.5 When is MPEG Useful for What?

We have mainly been talking about DVD, where the deliverable data rate is predetermined. In other words, the MPEG encoder needs to keep the rate just below some fixed threshold. Reducing it much below that threshold achieves nothing except a reduction in quality.

MPEG is also used for satellite and cable television. This shares with DVD the need to limit the data rate to the amount that the transmission channel can support. Although the overall data rate from a satellite is fixed, the allocation of data rates to the various channels is under the control of the supplier. A satellite can easily be reconfigured to send extra channels by reducing the data rate on each. So, when they need a new channel, they do not necessarily have to launch a new satellite immediately. They might also give a higher data rate to a premium movie channel, lower to a home shopping channel. MPEG allows these decisions to stay flexible.

If you sent the complete movie over the web, the data rate is important but not under our control. We need to reduce the total data sent, typically by reducing the size of the frames. This is the one case considered where we can make the image smaller. Only then do we need to consider further action, perhaps by forcing the DCT to use only the first few frequencies.

2.6 DVD Software

"Software" is what the industry calls that which they record on DVDs, CDs or tapes. It is what most of us would think of as content: films, speech or music.

DVD was slow to get going because, although the technology has been around for some time, it has been necessary to get the big Hollywood studios and others to commit their films to the

DVD catalogues. One result is that DVDs are usually given a Region Code, to allow selective release.

Region 0	Region-free: plays anywhere
Region 1	USA and Canada
Region 2	Europe, Japan, Middle East, Rep. of South Africa
Region 3	SE Asia, Taiwan
Region 4	Mexico, Central America, South America, Australia, New Zealand
Region 5	Russian Federation, Africa, India, Pakistan
Region 6	China

The disk carries the code and the intention is that your DVD player will only play disks from your region. In practice it is quite common to find players which can easily be modified to accept other regions. It is not in the interests of player manufacturers to restrict the attraction of their products! Even so there is no guarantee that this will continue to be the case.

At the moment, most disks are available first in Region 1. It is quite legal to buy and import Region 1 disks and, even with import duty and postage, it is often cheaper. It is at least possible that region coding will fail because of consumer pressure. However, with a move towards digital projection in cinemas, the issue of moving films around the world will be replaced with one of moving data around the world, so studio pressure for region encoding may reduce. Even so, there is usually only one set of stars to appear at each premier!

Some disks are supplied code-free, working in any region: Region 0, above.

Finally, do not confuse region-coding with PAL/NTSC coding. The latter is a technical consideration concerned with the precise number of scanlines per frame and the way colour is encoded. Europe uses PAL and America uses NTSC, for example. It applies to all TV equipment. The differences are small and most modern equipment will handle either. Region-coding is a feature only of DVD disks and is an arbitrary system designed to prevent export of disks between regions.

2.7 Digital Sound Standards

Since Audio CDs arrived, there has been a lot of development of digital sound for other applications. A movie has digital sound as well as digital pictures, so DVDs need to record that sound to a standard specification, which is part of the overall MPEG standards package. You will see references to the MPEG Audio Layer. You will also see that DVDs use Dolby sound. So what is going on here?

Dolby Labs were founded in the UK, in 1965. Their early products were concerned with improving the quality of domestic cassette tapes, of studio recordings and of cinema equipment. Cinema film was a prime target because movies used optical recording of sound: you can literally see the soundtrack on such a movie, as a width-modulated stripe running along the film edge. Sound quality was very poor. Having introduced Dolby Stereo, an analogue system with noise reduction and four channels, the company moved to the USA in 1976. Their commercial position all along has been to develop effective solutions and then licence them to manufacturers.

MPEG-1 Audio Layer II made an early impact as a digital sound standard. The unwieldy title is often reduced to MP2 and this is usually the file extension of audio-only files having this encoding method. An extension to this produced MPEG-2 Audio Layer II. Development of this international audio standard was entirely within Europe. It became part of MPEG-1 in 1992.

An alternative with higher compression appeared at almost the same time: MPEG Audio Layer III: MP3. MP3 is widely used for computer and internet applications. However MP2 performs better at high bit rates and is the *de facto* standard for digital audio broadcasts (DAB).

MP2

MP2 works by dividing the audio into 32 frequency bands, compression being in time. The basic perceptual observation is that a loud sound masks a quiet one when the loud sound is the lower in frequency and the quiet one is not much higher. If a given channel is quiet and an adjacent one is loud, then the quiet channel will be omitted, improving compression. This is an example of perceptual compression (Dolby, MP2 and MP3 are all perceptual systems).

European (PAL) DVD players contain stereo MP2 decoders, in competition to Dolby Digital. DVD players in NTSC countries are not required to decode MP2 audio, though most do. Some DVD recorders record in MP2. In practice almost all DVDs use Dolby, regardless of in which region they are sold.

MP3

Recall that DCT is a transform encoder, working in the (spatial) frequency domain. MP3 is also a transform encoder, so that compression takes place in the frequency domain. MP3 generates 576 frequency components, many more components than the 32 frequency bands used by MP2. Each band is then sampled (the rate is defined but varies with the application: 32kHz, 44.1kHz, 48kHz). The coder/decoder includes a perceptual model which permits it to modify the data, in accordance with perceptual principles. Typical rates achieved are 64-256 kbits/s. MP3 uses a more sophisticated psycho-acoustic model than MP2, which is why the compression is usually better. As with MPEG video, the decoding model is standardised but the way perceptual information is used by the encoder is not. All MP3 files should therefore play on a given decoder, but the quality depends on the encoder used. MP3 is of course a lossy format.

MP3 was widely used by peer-to-peer web sites such as Napster. Commercial online music does not use it because it has no means of preventing endless digital copying. There have been patent issues which have discouraged its use. Microsoft in particular has moved away from MP3 to its own format. As with gif format for image files (see later Chapter), the patent issue has led the open source community to develop an alternative, Vorbis, which is both free and seems to give better results.

Dolby

[As fans of Spinal Tap will know: it is not “Dobly”]

The proprietary, perceptual system for DVD, modern cinema theatres, digital satellite, TV and HDTV is Dolby Digital. It produces 5.1 channels: left, middle, right, left rear, right rear and a low-frequency effects channel (the “0.1”. The last carries little information but produces the thumps and bangs that shake you out of your chair and, at home, causes the neighbours to complain). The format supports other numbers of channels and you do occasionally see more (it also supports fewer). Dolby Digital Plus is a 7.1 format, required for HD-DVD, optional for Blu-Ray.