The Theory Behind Mp3

Rassol Raissi

December 2002

Abstract

Since the MPEG-1 Layer III encoding technology is nowadays widely used it might be interesting to gain knowledge of how this powerful compression/decompression scheme actually functions. How come the MPEG-1 Layer III is capable of reducing the bit rate with a factor of 12 without almost any audible degradation? Would it be fairly easy to implement this encoding algorithm? This paper will answer these questions and give further additional detailed information.

Table of Contents

List of Figures

List of tables

1 Introduction

Uncompressed digital CD-quality audio signals consume a large amount of data and are therefore not suited for storage and transmission. The need to reduce this amount without any noticeable quality loss was stated in the late 80ies by the International Organization for Standardization (ISO). A working group within the ISO referred to as the Moving Pictures Experts Group (MPEG), developed a standard that contained several techniques for both audio and video compression. The audio part of the standard included three modes with increasing complexity and performance. The third mode, called Layer III, manages to compress CD music from 1.4 Mbit/s to 128 kbit/s with almost no audible degradation. This technique, also known as MP3, has become very popular and is widely used in applications today.

Since the MPEG-1 Layer III is a complex audio compression method it may be quite complicated to get hold of all different components and to get a full overview of the technique. The purpose of this project is to provide an in depth introduction to the theory behind the MPEG-1 Layer III standard, which is useful before an implementation of an MP3 encoder/decoder. Note that this paper will not provide all information needed to actually start working with an implementation, nor will it provide mathematical descriptions of algorithms, algorithm analysis and other implementation issues.

2 Introduction To Data Compression

The theory of data compression was first formulated by Claud E. Shannon in 1949 when he released his paper: “A Mathematical Theory of Communication”. He proved that there is a limit to how much you can compress data without losing any information. This means that when the compressed data is decompressed the bitstream will be identical to the original bitstream. This type of data compression is called *lossless*. This limit, the entropy rate, depends on the probabilities of certain bit sequences in the data. It is possible to compress data with a compression rata close to the entropy rate and mathematically impossible to do better. Note that entropy coding only applies to lossless compression.

In addition to lossless compression there is also *lossy* compression. Here the decompressed data does not have to be exactly the same as the original data. Instead some amount of distortion (approximation) is tolerated. Lossy compression can be applied to sources like speech and images where you do not need all details to understand.

Lossless compression is required when no data loss is acceptable, for example when compressing data programs or text documents. Three basic lossless compression techniques are described below.

Runlength Encoding (RLE)
Figure 2.1 demonstrates an example of RLE.

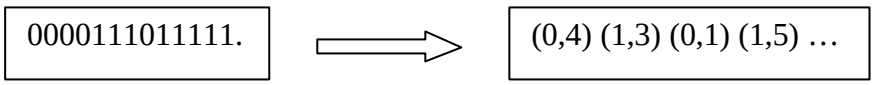

Figure 2.1: Runlength Encoding

Instead of using four bits for the first consecutive zeros the idea is to simply specify that there are four consecutive zeros next. This will only be efficient when the bitstreams are non random, i.e. when there are a lot of consecutive bits.

Move To Front Encoding (MTF)
This is a technique that is ideal for sequences with the property that the occurrence of a character indicates it is more likely to occur immediately afterwards. A table as the one shown in Table 2.1 is used. The initial table is built up by the positions of the symbols about to be compressed. So if the data starts with symbols 'AEHTN...' the N will initially be encoded with 5. The next procedure will move N to the top of the table. Assuming the following symbol to be N it will now be represented by 1, which is a shorter value. This is the root of Entropy coding; more frequent symbols should be coded with a smaller value.

increasing values ↓

encoding	symbol
1	A
2	E
3	H
4	T
5	N
…	…

↑ probability

Table 2.1: Move To Front Encoding

RLE and MTF are often used as subprocedures in other methods.

Huffman Coding
The entropy concept is also applied to Huffman hence common symbols will be represented with shorter codes. The probability of the symbols has to be determined prior to compression (see Figure 2.2).

symbol	probability
A	0.13
B	0.05
C	0.33
D	0.08
E	0.18
F	0.23

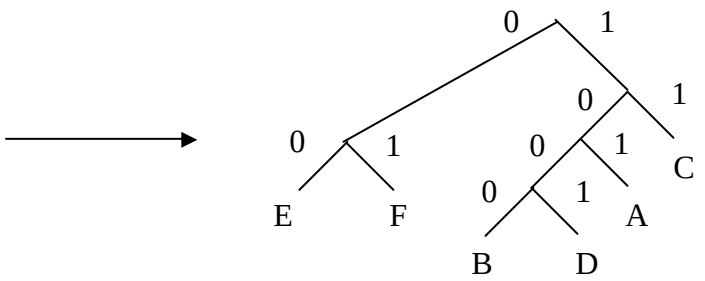

Figure 2.2: Huffman Coding

A binary tree is constructed with respect to the probability of each symbol. The coding for a certain symbol is the sequence from the root to the leaf containing that symbol. A greedy algorithm for building the optimal tree:

1. Find the two symbols with the lowest probability.
2. Create a new symbol by merging the two and adding their respective probability. It has to be how to treat symbols with an equal probability (see Figure 2.3).
3. Repeat steps 1 and 2 until all symbols are included.

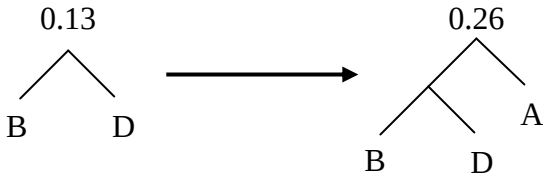

Figure 2.3: Greedy Huffman algorithm

When decoding the probability table must first be retrieved. To know when each representation of a symbol ends simply follow the tree from the root until a symbol is found. This is possible since no encoding is a subset of another (prefix coding).

3 Background

3.1 Psychoacoustics & Perceptual Coding

Psychoacoustics is the research where you aim to understand how the ear and brain interact as various sounds enter the ear.

Humans are constantly exposed to an extreme quantity of radiation. These waves are within a frequency spectrum consisting of zillions of different frequencies. Only a small fraction of all waves are perceptible by our sense organs; the light we see and the sound we hear. Infrared and ultraviolet light are examples of light waves we cannot percept. Regarding our hearing, most humans can not sense frequencies below 20 Hz nor above 20 kHz. This bandwidth tends to narrow as we age. A middle aged man will not hear much above 16 kHz. Frequencies ranging from 2 kHz to 4 kHz are easiest to perceive, they are detectable at a relatively low volume. As the frequencies changes towards the ends of the audible bandwidth, the volume must also be increased for us to detect them (see Figure 3.1). That is why we usually set the equalizer on our stereo in a certain symmetric way. As we are more sensitive to midrange frequencies these are reduced whereas the high and low frequencies are increased. This makes the music more comfortable to listen to since we become equal sensitive to all frequencies.

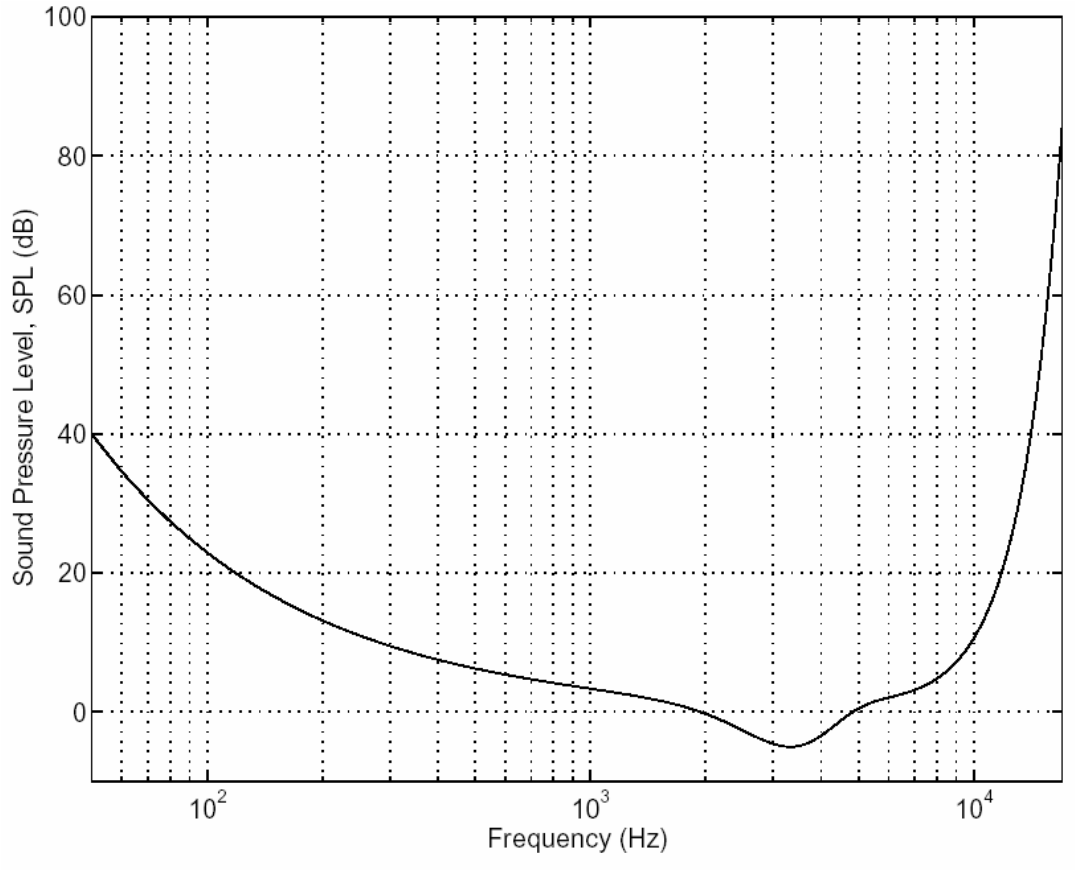

Figure 3.1: The absolute threshold of hearing (Source [1])

As our brain cannot process all the data available to our five senses at a given time, it can be considered as a mental filter of the data reaching us. A perceptual audio codec is a codec that takes advantage of this human characteristic. While playing a CD it is impossible to percept all data reaching your ears, so there is no point in storing the part of the music that will be inaudible. The process that makes certain samples inaudible is called *masking*. There are two masking effects that the perceptual codec need to be aware of; *simultaneous* masking and *temporal* masking.

Experiments have shown that the human ear has 24 frequency bands. Frequencies in these so called *critical bands* are harder to distinguish by the human ear. Suppose there is a dominant tonal component present in an audio signal. The dominant noise will introduce a masking threshold that will mask out frequencies in the same critical band (see Figure 3.2). This frequency-domain phenomenon is known as simultaneous masking, which has been observed within critical bands.

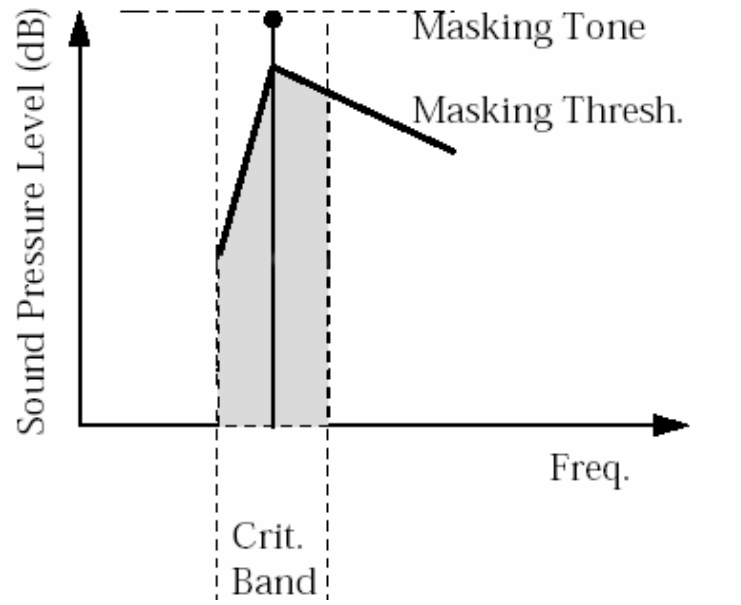

Figure 3.2: Simultaneous masking (Source [1])

Temporal masking occurs in the time-domain. A stronger tonal component (masker) will mask a weaker one (maskee) if they appear within a small interval of time. The masking threshold will mask weaker signals pre and post to the masker. Premasking usually lasts about 50 ms while postmasking will last from 50 to 300 ms, depending on the strength and duration of the masker as shown in Figure 3.3.

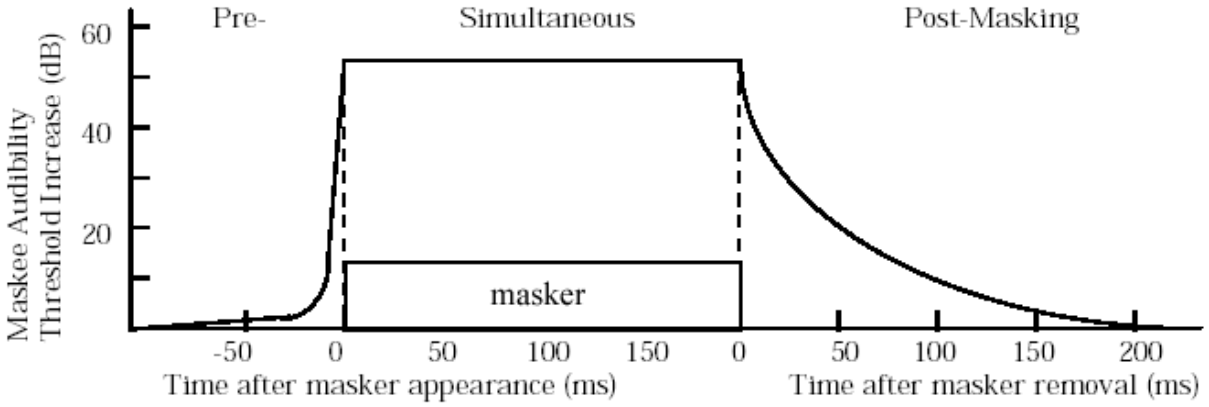

Figure 3.3: Temporal Masking (Source [1])

3.2 PCM

Pulse Code Modulation is a standard format for storing or transmitting uncompressed digital audio. CDs, DATs are some examples of media that adapts the PCM format. There are two variables for PCM; sample rate [Hz] and bitrate [Bit]. The sample rate describes how many samples per second the recording consists of. A high sample rate implies that higher frequencies will be included. The bitrate describes how big the digital word is that will hold the sample value. A higher bitrate gives a better audio resolution and lower noise since the sample can be determined more exactly using more bits. CD audio is 44,100 Hz and 16 Bit.

A crude way of compressing audio would be to simple record at a lower sample rate or bitrate. Using a bitrate of 8 bits instead of 16 bits will reduce the amount of data to only 50% but the quality loss in doing this is unacceptable.

4 An Overview of the MPEG-1 Layer III standard

4.1 The MPEG-1 Standard

The International Organization for Standardization (ISO) is an international federation that aims to facilitate the international exchange of goods and services by publishing international standards. Working within ISO, the Moving Picture Experts Group was assigned to initiate the development of a common standard for coding/compressing a representation of moving pictures, audio and their combination. This standard had to be generic, meaning that any decoder using the standard had to be capable of decoding a bitstream generated by a random encoder using the same standard. Furthermore, trying to preserve both the video and audio quality was obviously very essential.

The development began in 1988 and was finalized in 1992 given the name MPEG-1. The standard consisted of three different parts:

- An audio part
- A video part
- A System part

The system part was a description of how to transmit multiple audio and video signals on a single distribution media. Using the MPEG-1 standard it was possible to transmit video and associated audio at a bitrate of between 1-2 Mbit/s.

For the audio part there were three levels of compression and complexity defined; Layer I, Layer II and Layer III. Increased complexity requires less transmission bandwidth since the compression scheme becomes more effective. Table 4.1 gives the transmission rates needed from each layer to transmit CD quality audio.

	Coding	Ratio	Required bitrate
	PCM CD Quality	1:1	1.4 Mbps
Complexity ↓	Layer I	4:1	384 kbps
	Layer II	8:1	192 kbps
	Layer III (MP3)	12:1	128 kbps

Table 4.1: Bitrates required to transmit a CD quality stereo signal

The third layer compresses the original PCM audio file by a factor of 12 without any noticeable quality loss, making this layer the most efficient and complex layer of the three. The MPEG-1 Layer III standard is normally referred to as MP3.

What is quite easy to misunderstand at this point is that the primary developers of the MP3 algorithm were not the MPEG but the Fraunhofer Institute, who began their work in 1987

together with the German University of Erlangen. ISO then codified the work into the MPEG-1 Layer III standard. This is usually the way standards are created.

Nevertheless, the work continued and MPEG-2 was finalized in 1994, introducing a lot of new video coding concepts. The main application area for MPEG-2 was digital television. The audio part of MPEG-2 consisted of two extensions to MPEG-1 audio:

- Multichannel audio encoding, including the 5.1 configuration. (Backward compatible)
- Coding at lower sample frequencies (see chapter 4.5)

More standards (MPEG-4, MPEG-7) have been developed since then but this paper will only mention the two first phases of this research.

4.2 Reducing the data by a factor of 12

Since MP3 is a perceptual codec it takes advantage of the human system to filter unnecessary information. Perceptual coding is a lossy process and therefore it is not possible to regain this information when decompressing. This is fully acceptable since the filtered audio data cannot be perceptible to us anyway. There is no point in dealing with inaudible sounds.

Each human critical band is approximated by *scalefactor bands*. For every scalefactor band a masking threshold is calculated. Depending on the threshold the scalefactor bands are scaled with a suited scalefactor to reduce quantization noise caused by a later quantization of the frequency lines contained in each band.

But merely lossless compression will not be efficient enough. For further compression the Layer III part of the MPEG-1 standard applies Huffman Coding. As the codec is rather complex there are additional steps to trim the compression. For a more detailed description on the encoding algorithm consult chapter 6.

4.3 Freedom of Implementation

The MP3 specification (ISO 11172-3) defines how the encoded/decoded bitstream should be structured/interpreted. The output of an encoder developed according to this specification will be recognizable to any MP3 decoder and vice versa. This is of course necessary for it to be a standard specification. But the specification does not exactly specify the steps of how to encode an uncompressed stream to a coded bitstream. This means that the encoders can function quite differently and still produce a compliant to the standard. It is up to the developer to decide how to implement certain parts of the encoder. For instance, it is not specified how to deal with the frequencies over 16 kHz. Since it is quite hard to detect audio signals in that spectrum a developer might choose to discard these frequencies, which will leave bits available to encode more audible signals.

Two important aspects when developing an encoder are speed and quality. Unfortunately, the implementations given by the standard do not always apply the most efficient algorithms. This leads to huge differences in the operating speed of various encoders. The quality of the output may also vary depending on the encoder.

Regarding the decoding, all transformations needed to produce the PCM samples are defined. However, details for some parts are missing and the emphasis lies on the interpretation of the encoded bitstream, without using the most efficient algorithms in some cases.

This freedom of implementation given by the MPEG-1 Layer III standard should be carefully considered in order to find a good application solution. It is also important to always optimize the encoding and decoding procedures since they are not optimized in the standard definition.

4.4 Bitrate

The bitrate is a user option that has to be set prior to encoding. It will inform the encoder of the amount of data allowed to be stored for every second of uncompressed audio. This gives the user the opportunity to choose the quality of the encoded stream. The Layer III standard defines bitrates from 8 kbit/s up to 320 kbit/s, default is usually 128 kbit/s. A higher bitrate implies that the samples will be measured more precisely giving an improved audio resolution.

Note that a stereo file with a certain bitrate divides the bitrate between the two channels, allocating a larger portion of the bitrate to the channel which for the moment is more complex.

The standard specifies two different types of bitrates; *Constant Bitrate* (CBR) and *Variable Bitrate* (VBR). When encoding using CBR (usually default) every part of a song is encoded with the same amount of bits. But most songs will vary in complexity. Some parts might use a lot of different instruments and effects while other parts are more simply composed. CBR encoding causes the complex parts of a song, which require more bits, to be encoded using the same amount of bits as the simple parts, which require less bits. VBR is a solution to this problem allowing the bitrate to vary depending on the dynamics of the signal. As you will see in chapter 5, the encoded stream is divided into several frames. Using VBR makes it possible for the encoder to encode frames using different bitrates. The quality is set using a threshold specified by the user to inform the encoder of the maximum bitrate allowed. Unfortunately there are some drawbacks of using VBR. Firstly, VBR might cause timing difficulties for some decoders, i.e. the MP3 player might display incorrect timing information or non at all. Secondly, CBR is often required for broadcasting, which initially was an important purpose of the MP3 format.

4.5 Sampling frequency

The audio resolution is mainly depending on the sampling frequency, which can be defined as the number of times per second the signal is stored. A high bitrate will give a better precision of a sampled value whereas a high sampling frequency gives the ability to store more values, which in turn gives a broader frequency spectrum. MPEG-1 defines audio compression at 32 kHz, 44.1 kHz and 48 kHz.

4.6 Channel Modes

There are four different channel modes defined:

- Single Channel
- Dual Channel (channels are encoded independently of each other)
- Stereo
- Joint Stereo

Note: Dual channel files are made of two independent mono channels. Each one uses exactly half the bitrate of the file. Most decoders output them as stereo, but it might not always be the case. One example of use would be some speech in two different languages carried in the same bitstream, and then an appropriate decoder would decode only the chosen language.

4.6.1 Joint Stereo

The Joint Stereo mode considers the redundancy between left and right channels to optimize coding. There are two techniques here; *middle/side stereo (*MS stereo*)* and *Intensity Stereo*.

MS stereo is useful when two channels are highly correlated. The left and right channels are transmitted as the sum and difference of the two channels, respectively. Since the two channels are reasonably alike most of the time the sum signal will contain more information than the difference signal. This enables a more efficiently compressing compared to transmitting the two channels independently. MS stereo is a lossless encoding.

In intensity stereo mode the upper frequency subbands are encoded into a single summed signal with corresponding intensity positions for the scalefactor bands encoded. In this mode the stereo information is contained within the intensity positions because only a single channel is transmitted. Unfortunately stereo inconsistencies will appear for this model since audio restricted to one channel will be present in both channels. The inconsistencies will not be conceivable by the human ear if they are kept small.

Some encodings might use a combination of these two methods.

5 The Anatomy of an MP3 file

All MP3 files are divided into smaller fragments called *frames*. Each frame stores 1152 audio samples and lasts for 26 ms. This means that the frame rate will be around 38 fps. In addition a frame is subdivided into two *granules* each containing 576 samples. Since the bitrate determines the size of each sample, increasing the bitrate will also increase the size of the frame. The size is also depending on the sampling frequency according to following formula:

$$\frac{144 * bitrate}{samplefrequency} + Padding \text{ [bytes]}$$

Padding refers to a special bit allocated in the beginning of the frame. It is used in some frames to exactly satisfy the bitrate requirements. If the padding bit is set the frame is padded with 1 byte. Note that the frame size is an integer: Ex: 144*128000/44100 = 417

5.1 The Frame Layout

A frame consists of five parts; header, CRC, side information, main data and ancillary data, as shown in Figure 5.1.

Header	CRC	Side Information	Main Data	Ancillary Data

Figure 5.1: The frame layout

5.1.1 Frame header

The header is 32 bits long and contains a synchronization word together with a description of the frame. The synchronization word found in the beginning of each frame enables MP3 receivers to lock onto the signal at any point in the stream. This makes it possible to broadcast any MP3 file. A receiver tuning in at any point of the broadcast just have to search for the synchronization word and then start playing. A problem here is that spurious synchronization words might appear in other parts of the frame. A decoder should instead check for valid sync words in two consecutive frames, or check for valid data in the side information, which could be more difficult.

Figure 5.2 shows an illustration of the header.

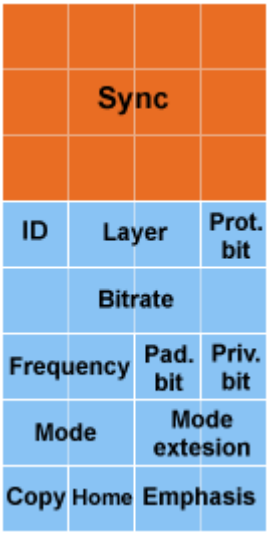

Figure 5.2: The MP3 frame header (Source [7])

Sync (12 bits)
This is the synchronization word described above. All 12 bits must be set, i.e. '1111 1111 1111'.

Id (1 bit)
Specifies the MPEG version. A set bit means that the frame is encoded with the MPEG-1 standard, if not MPEG-2 is used.

Some add-on standards only use 11 bits for the sync word in order to dedicate 2 bits for the id. In this case Table 5.1 is applied.

00	MPEG-2.5 (Later extension of MPEG-2)
01	Reserved
10	MPEG-2
11	MPEG-1

Table 5.1: Bitvalues when using two id bits

Layer (2 bits)
See Table 5.2

00	reserved
01	Layer III
10	Layer II
11	Layer I

Table 5.2: Definition of layer bits

Protection Bit (1 bit)
If the protection bit is set, the CRC field will be used.

Bitrate (4 bits)
These four bits tells the decoder in what bitrate the frame is encoded. This value will be the same for all frames if the stream is encoded using CBR. Table 5.3 shows the defined bit values.

bits	MPEG-1, layer I	MPEG-1, layer II	MPEG-1, layer III	MPEG-2, layer I	MPEG-2, layer II	MPEG-2, layer III
0 0 0 0						
0 0 0 1	32	32	32	32	32	8
0 0 1 0	64	48	40	64	48	16
0 0 1 1	96	56	48	96	56	24
0 1 0 0	128	64	56	128	64	32
0 1 0 1	160	80	64	160	80	64
0 1 1 0	192	96	80	192	96	80
0 1 1 1	224	112	96	224	112	56
1 0 0 0	256	128	112	256	128	64
1 0 0 1	288	160	128	288	160	128
1 0 1 0	320	192	160	320	192	160
1 0 1 1	352	224	192	352	224	112
1 1 0 0	384	256	224	384	256	128
1 1 0 1	416	320	256	416	320	256
1 1 1 0	448	384	320	448	384	320
1 1 1 1						

Table 5.3: Bitrate definitions (Source [7])

Frequency (2 bits)
2 bits that give the sampling frequency, see Table 5.4.

Bits	MPEG1	MPEG2	MPEG2.5
00	44100 Hz	22050 Hz	11025 Hz
01	48000 Hz	24000 Hz	12000 Hz
10	32000 Hz	16000 Hz	8000 Hz
11	reserv.	reserv.	reserv.

Table 5.4: Definition of accepted sampling frequencies

Padding bit (1 bit)
An encoded stream with bitrate 128 kbit/s and sampling frequency of 44100 Hz will create frames of size 417 bytes. To exactly fit the bitrate some of these frames will have to be 418 bytes. These frames set the padding bit.

Private bit (1 bit)
One bit for application-specific triggers.

Mode (2 bits)
Specifies what channel mode is used according to Table 5.5.

00	Stereo
01	Joint Sereo
10	Dual Channel
11	Single Channel

Table 5.5: Channel Modes and respective bitvalues

Mode Extension (2 bits)
These 2 bits are only usable in joint stereo mode and they specify which methods to use. The joint stereo mode can be changed from one frame to another, or even switched on or off. To interpret the mode extension bits the encoder needs the information in Table 5.6.

Bits	Intensity stereo	MS stereo
00	Off	Off
01	On	Off
10	Off	On
11	On	On

Table 5.6: Definition of mode extension bits

Copyright Bit (1 bit)
If this bit is set it means that it is illegal to copy the contents.

Home (Original Bit) (1 bit)
The original bit indicates, if it is set, that the frame is located on its original media.

Emphasis (2 bits)
The emphasis indication is used to tell the decoder that the file must be de-emphasized, i.e. the decoder must 're-equalize' the sound after a Dolby-like noise supression. It is rarely used.

00	None
01	50/15 ms
10	Reserved
11	CCITT J.17

Table 5.7: Noise supression model

CRC *(0 bytes, 16 bytes)*
This field will only exist if the protection bit in the header is set and makes it possible check the most sensitive data for transmission errors. Sensitive data is defined by the standard to be bit 16 to 31 in both the header and the side information. If these values are incorrect they will corrupt the whole frame whereas an error in the main data only distorts a part of the frame. A corrupted frame can either be muted or replaced by the previous frame.

5.1.2 Side Information

The side information part of the frame consists of information needed to decode the main data. The size depends on the encoded channel mode. If it is a single channel bitstream the size will be 17 bytes, if not, 32 bytes are allocated. The different parts of the side information are presented in Table 5.8 and described in detail below.

The length of each field will be specified in parenthesis together with the fieldname above the actual description. If one length value is written the field size is constant. If two values are specified the first value will be used in mono mode and the second will be used for all other modes, thus these fields are of variable length. All tables below will assume a mono mode. The tables will change depending on mode since separate values are needed for each channel.

main_data_begin	private_bits	scfsi	Side_info gr. 0	Side_info gr. 1

Table 5.8: Side information

main_data_begin (9 bits)
Using the layer III format there is a technique called the *bit reservoir* which enables the left over free space in the main data area of a frame to be used by consecutive frames. To be able to find where the main data of a certain frame begins the decoder has to read the main_data_begin value. The value is as a negative offset from the first byte of the synchronization word. Since it is 9 bits long it can point (2^9 – 1) * 8 = 4088 bits. This means that data for one frame can be found several previous frames. Note that static parts of a frame like the header, which is always 32 bytes, are not included in the offset. If main_data_begin = 0 the main data starts directly after the side information.

private_bits (5 bits, 3 bits)
Bits for private use, these will not be used in the future by ISO.

scfsi (4 bits, 8 bits)
The ScaleFactor Selection Information determines weather the same scalefactors are transferred for both granules or not. Here the scalefactor bands are divided into 4 groups according to Table 5.9.

group	scalefactor bands
0	0,1,2,3,4,5
1	6,7,8,9,10
2	11,12,13,14,15
3	16,17,18,19,20

Table 5.9: Scalefactor groups

4 bits per channel are transmitted, one for each scalefactor band. If a bit belonging to a scalefactor band is zero the scalefactors for that particular band are transmitted for each granule. A set bit indicates that the scalefactors for granule0 are also valid for granule1. This means that the scalefactors only need to be transmitted in granule0, the gained bits can be used for the Huffman coding.

If short windows are used (block_type = 10) in any granule/channel, the scalefactors are always sent for each granule for that channel.

Side info for each granule

The last two parts of a frame have the same anatomy and consists of several subparts as shown in . These two parts store particular information for each granule respectively.

part2_3_length	big_values	global_gain	scalefac_compress
windows_switching_flag	block_type	mixed_block_flag	table_select
subblock_gain	region0_count	region1_count	preflag
scalefac_scale	count1table_select		

Table 5.10: Fields for side information for each granule

par2_3_length (12 bits, 24 bits)
States the number of bits allocated in the main data part of the frame for scalefactors (part2) and Huffman encoded data (part3). 12 bits will be used in a single channel mode whereas in stereo modes the double is needed. This field can be used to calculate the location of the next granule and the ancillary information (if used).

big_values (9 bits, 18 bits)
The 576 frequency lines of each granule are not coded with the same Huffmancode table. These frequencies range from zero to the Nyquist frequency and are divided into five *regions* (see
Figure 5.3). The purpose of this partitioning is to allow different Huffman tables to different parts of the spectrum in order to enhance the performance of the Huffman encoder.

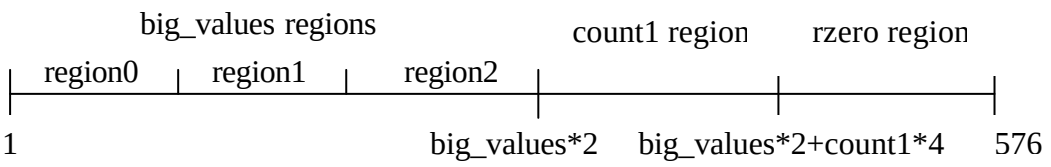

Figure 5.3: Regions of the frequency spectrum

Partitioning is done according to the maximum quantized values. This is done with the assumption that values at higher frequencies are expected to have lower amplitudes or does not need to be coded at all.

The rzero region represents the highest frequencies and contains pairs of quantized values equal to zero. In the count1 region quadruples of quantized values equal to -1, 0 or 1 reside. Finally, the big_values region contains pairs of values in representing the region of the spectrum which extends down to zero. The maximum absolute value in this range is constrained to 8191. The big_values field indicates the size of the big_values partition hence the maximum value is 288.

global_gain (8 bits, 16 bits)
Specifies the quantization step size, this is needed in the requantization block of the decoder.

scalefac_compress (4 bits, 8 bits)
Determines the number of bits used for the transmission of scalefactors. A granule can be divided into 12 or 21 scalefactor bands. If long windows are used (block_type = {0,1,3})the granule will be partitioned into 21 scalefactor bands. Using short windows (block_type = 2) will partition the granule into 12 scalefactor bands. The scale factors are then further divided into two groups, 0-10, 11-20 for long windows and 0-6, 7-11 for short windows.

The scalefac_compress variable is an index to a defined table (see Table 5.11). slen1 and slen2 gives the number of bits assigned to the first and second group of scalefactor bands respectively.

scalefac_compress	slen1	slen2
0	0	0
1	0	1
2	0	2
3	0	3
4	3	0
5	1	1
6	1	2
7	1	3
8	2	1
9	2	2
10	2	3
11	3	1
12	3	2
13	3	3
14	4	2
15	4	3

Table 5.11: scalefac_compress table

windows_switching_flag (1 bit, 2 bits)
Indicates that another window than the normal is used (Ch 6.2). block_type, mixed_block_flag and subblock_gain are only used if windows_switching_flag is set.

Also when windows_switching_flag is set all remaining values not being in region0 are contained in region1 thus region2 is not used.

block_type (2 bits, 4 bits)
This field is only used when windows_switching_flag is set and indicates the type of window used for the particular granule (see Table 5.12, all values but 3 are long windows). The value 00 is forbidden since block_type is only used when other than normal windows are used.

block_type	window type
00	forbidden
01	start
10	3 short windows
11	end

Table 5.12: block_type definition

mixed_blockflag (1 bit, 2 bits)
This field is only used when windows_switching_flag is set.

The mixed_block_flag indicates that different types of windows are used in the lower and higher frequencies. If mixed_block_flag is set the two lowest subbands (see 6.1) are transformed using a normal window and the remaining 30 subbands are transformed using the window specified by the block_type variable.

table_select ((10 bits, 20 bits) or (15 bits, 30 bits))
There are 32 possible Huffman code tables available in the standard. The value of this field gives the Huffman table to use when decoding and its size is 5 bits, i.e. 32 different values, for each region, granule and channel. The table_select only specifies the tables to use when decoding the big_values partition. The table specified is dependent on the local statistics of the signal and by the maximum quantization allowed to quantize the 576 frequency lines in the granule.

As stated above, when the windows_switching_flag is set region2 is empty so only two regions are coded. This implies that in mono mode 5*2*1 = 10 bits are needed and in stereo mode5*2*2 = 20 bits are needed if windows_switching_flag = 1. Using all regions (windows_switching_flag = 0) the bits needed will be 5*3*1 = 15 and 5*3*2 = 30 respectively.

subblock_gain (9 bits, 18 bits)
This field is only used when windows_switching_flag is set and when block_type = 10, although it is transmitted independently of block_type. This 3 bit variable indicates the gain offset from global_gain for each short block.

region0_count (4 bits, 8 bits), region1_count (3 bits, 6 bits)
region0_count and region1_count contains one less than the number of scalefactor bands in region0 and region1 respectively. The region boundaries are adjusted to the partitioning of the frequency spectrum into scalefactor bands. If short windows are used the number of each windows is counted. For instance if region0 = 8 there are 9/3 = 3 scalefactor bands in region0.

preflag (1 bit, 2bits)
This is a shortcut for additional high frequency amplification of the quantized values. If preflag is set, the values of a defined table are added to the scalefactors. If block_type = 10, i.e short blocks, preflag is never used.

scalfac_scale (1 bit, 2bits)
The scalefactors are logarithmically quantized with a step size of 2 or v2 according to Table 5.13.

scalfac_scale	step size
0	v2
1	2

Table 5.13: Quantization step size applied to scalefactors

count1table_select (1 bit, 2bits)
Two possible Huffman code tables are available for the count1 region. This field specifies which table to apply.

5.1.3 Main Data

The main data part of the frame consists of scalefactors, Huffman coded bits and ancillary data.

scalefactors
The purpose of scalefactors is to reduce the quantization noise. If the samples in a particular scalefactor band are scaled the right way the quantization noise will be completely masked. One scalefactor for each scalefactor band is transmitted. The scfsi field determines if the scalefactors are shared between the granules or not. The actual bits allocated for each scalefactor depends on the scalefac_compress field.

The subdivision of the spectrum into scalefactor bands is fixed for every window length and sampling frequency and stored in tables in the coder and decoder (a table of scalefactor bands used for long windows with a sampling frequency of 44.1 kHz is presented in appendix C).

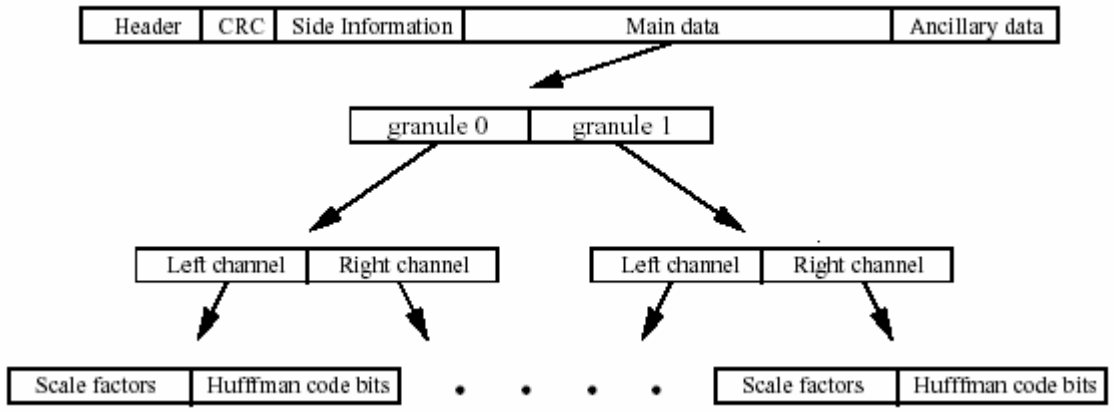

Figure 5.4: Organization of scalefactors in granules and channels

Huffman code bits
This part of the frame contains the Huffman code bits. Information on how to decode these is found in the side information. For the three subregions in the big_values region always pairs are encoded. For instance, the Huffman table nr 7 (Appendix C) may be applied to big_values. *x* and *y* are the pair of values to be coded, *hlen* specifies the length of the Huffman code for x and y and hcode is the actual Huffman code representing x and y. In the case of count1 values they are encoded in quadruples, *v*,*w*,*x*,*y*. Only tables A and B support coding in this region. The rzero region is not Huffman coded but runlength coded since all values are zero.

Depending on whether long or short blocks are used, the order of the Huffman data differs. If long blocks are used, the Huffman encoded data is ordered in terms of increasing frequency.

5.1.4 Ancillary Data

The ancillary data is optional and the number of bits available is not explicitly given. The ancillary data is located after the Huffman code bits and ranges to where the next frame's main_data_begin points to.

5.2 ID3

Although the MP3 format did manage to compress audio files very effectively without any noticeable quality degradation, there was no possibility to store textual information. For this purpose a fixed-size 128-byte tag at the end of the file was introduced. The tag was called ID3 and contained fields for title (30 bytes), artist (30 bytes), album (30 bytes), year (4 bytes), comment (30 bytes) and genre (1 byte). The byte value in the genre field corresponds to a value in a predefined list. An MP3 file using the ID3 tag should write 'TAG' at the end of the ordinary encoding. Counting these 3 bytes needed to write 'TAG' the *ID3* tag will be 128 bytes. Unfilled space in each field should be filled with the binary value 0.

The comment field was later reduced by two bytes in order to include a one byte track field telling the decoder which track number on the CD this music came from. The byte left over should be a binary 0 and written between the comment field and the track field. This variant of the ID3 tag was named *ID3v1.1* (see Figure 5.5).

'TAG' (3)	Title (30)	Artist (30)	Album (30)	Year (4)	Comment (28)	'0'	Track (1)	Genre (1)

Figure 5.5: ID3v1.1

Unfortunately ID3 was not a clever way to store textual information. It only supported a few fields of information and these were limited to 30 characters. An additional drawback was that since the tag was put at the end the information could not be retrieved when streaming the file. With respect to these drawbacks a second more complex version was released, ID3v2 [7]. If this tag is used 'ID3' will be written at the very beginning of the file. ID3v2 introduced a lot of new fields and mostly every field can store information of any length since this tag is dynamic in size. The tag is located in the beginning of the file to facilitate streaming. The current informal standard is ID3v2.4. [7] will provide a detailed description of the ID3v2 tag.

6 Encoding

The encoding process is very complex and not fully described here.

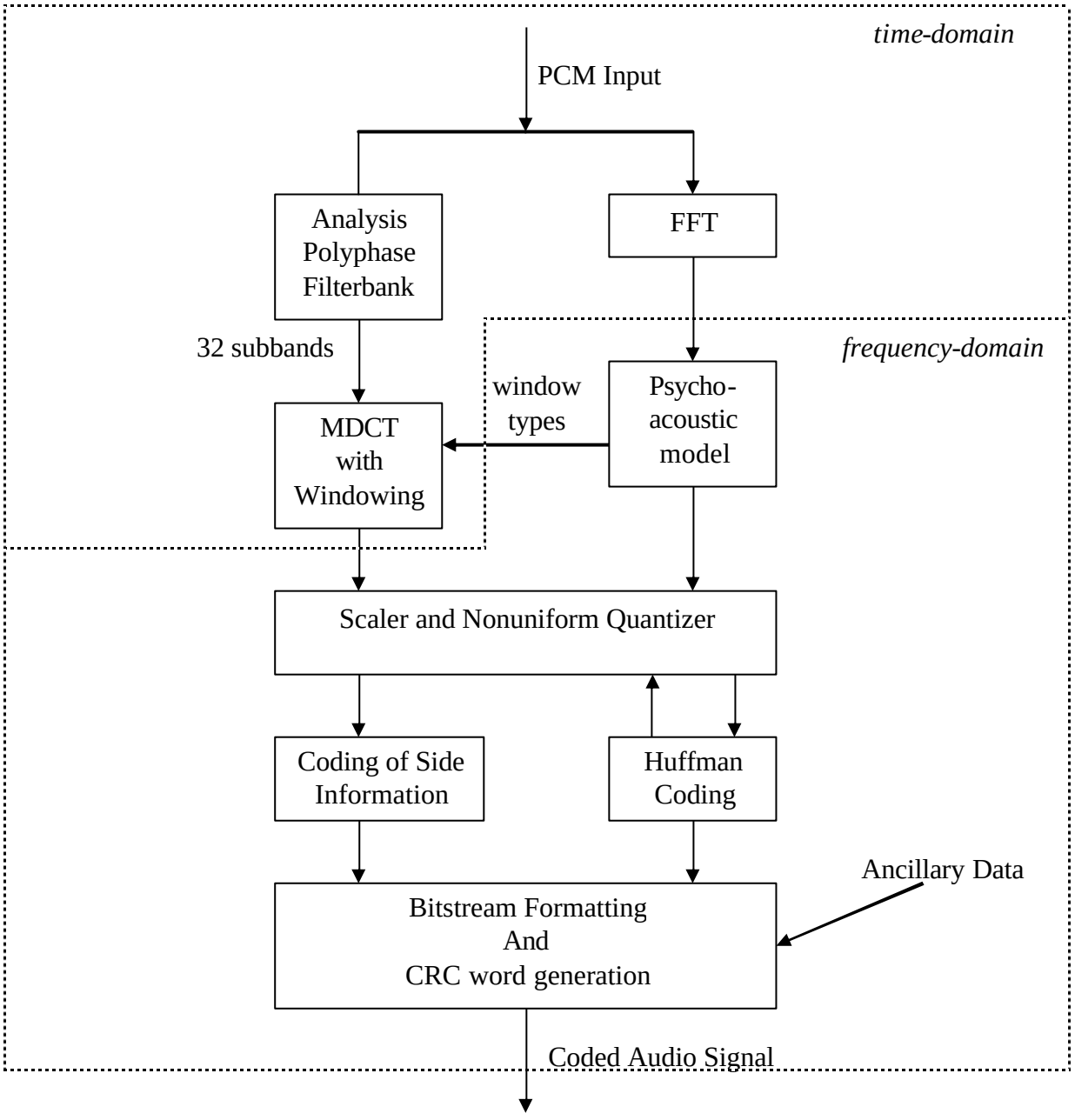

Figure 6.1: MPEG-1 Layer III encoding scheme

6.1 Analysis Polyphase Filterbank

A sequence of 1152 PCM samples are filtered into 32 equally spaced frequency subbands depending of the Nyquist frequency of the PCM signal. If the sample frequency of the PCM signal is 44.1 kHz the Nyquist frequency will be 22.05 kHz. Each subband will be approximately 22050/32 = 689 Hz wide. The lowest subband will have a range from 0-689

Hz, the next subband 689 – 1378 Hz, etc. Every sample (might) contain signal components from 0 – 22.05 kHz that will be filtered into appropriate subband. This means that the number of samples has increased by a factor 32 since every subband now stores a subspectra of the sample. For example, having filtered 100 samples increases the number of samples to 3200. The 100 samples in every subband will then be decimated by a factor 32, hence only every thirty-second sample is retained. The number of samples is now reduced from 3200 back to 100. But note that there has been a data reduction since a sample in a subband does not include the whole frequency spectra since it has been filtered.

Since it is not possible to construct bandpass filters with a perfectely square frequency response, some aliasing will be introduced by the decimation.

6.2 Modified discrete cosine transform (MDCT)

By applying a modified discrete cosine transform to each time frame of subband samples the 32 subbands will be split into 18 finer subbands creating a granule with a total of 576 frequency lines. But prior to the MDCT each subband signal has to be windowed.

Windowing is done to reduce artefacts caused by the edges of the time-limited signal segment. There are four different window types defined in the MPEG standard (see Figure 6.2). Depending on the degree of stationarity the psychoacoustic model determines which window type to apply and forwards the information to this block.

If the psychoacoustic model decides that the subband signal at the present time frame shows little difference from the previous time frame, then the long window type is applied, which will enhance the spectral resolution given by the MDCT. Alternatively, if the subband signal shows considerable difference from the previous time frame, then the short windows is applied. This type of window consists of three short overlapped windows and will improve the time resolution given by the MDCT. A higher time resolution is necessary in order to control time artifacts, for instance pre-echoes. In order to obtain a better adaptation when windows transitions are required, two windows referred to as start windows and stop windows, are defined.

A long window becomes a start window if it is immediately followed by a short window. Similarly, a long window becomes a stop window if it is immediately preceded by a short window. The start and stop windows are skewed to account for the steeper sides of the adjacent short window.

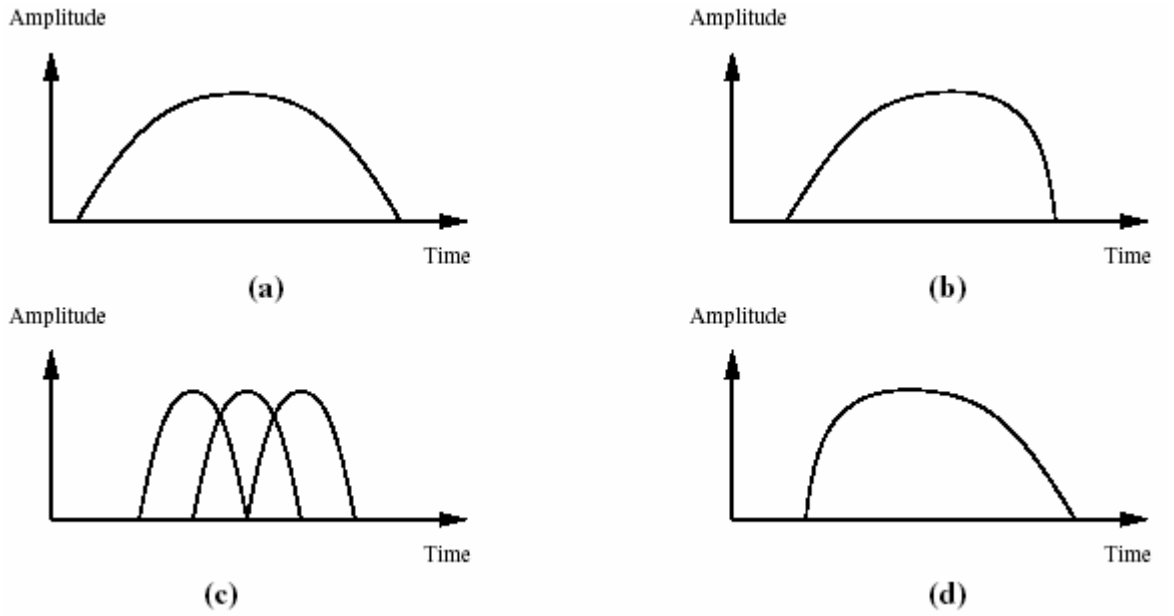

Figure 6.2: Window types

(a) normal window, (b) start window, (c) short windows, (d) stop window

The aliasing introduced by the polyphase filter bank is now removed to reduce the amount of information that needs to be transmitted. This is achieved using a series of butterfly computations that add weighted, mirrored versions of adjacent subbands to each other (see chapter 7: Alias Reduction).

6.3 FFT

Simultaneously as the signal is processed by the polyphase filterbank it is also transformed to the frequency domain by a Fast Fourier Transform. Both a 1024 and a 256 point FFT are performed on 1152 PCM samples at the time to give higher frequency resolution and information on the spectral changes over time.

6.4 Psychoacoustic Model

This block retrieves the input data from the FFT output. Since the samples are in the frequency domain they can be applied to a set of algorithms. These algorithms will model the human sound perception and hence they can provide information about which parts of the audio signals that is audible and which parts are not. This information is useful to decide which window types the MDCT should apply and also to provide the Nonuniform Quantization block with information on how to quantize the frequency lines.

To know which window type to send to the MDCT block the two presently FFT spectra and the two previous spectra are compared. If certain differences are found a change to short windows requested. As soon as the differences fades away the MDCT block will be informed to change back to long (normal) windows (see Figure 6.3).

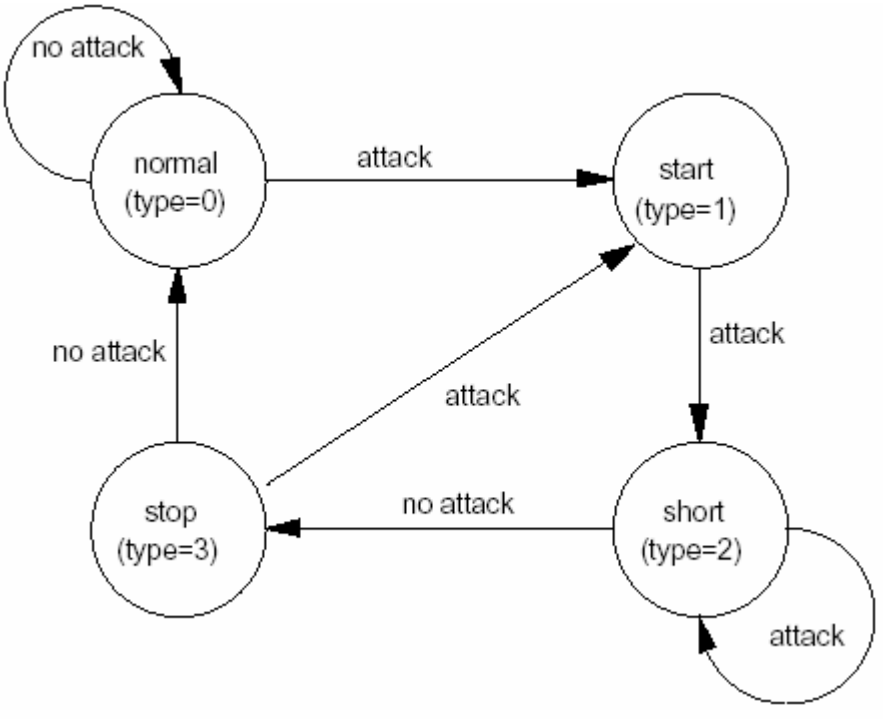

Figure 6.3: Window switching decision (Source [8])

The Psychoacoustic Model also analyzes the FFT spectrum to detect dominant tonal components and for each critical band masking thresholds are calculated. Frequency components below this threshold are masked out. Recall that the scalefactor bands are roughly equivalent to the critical bands of human hearing. The thresholds limits for each scalefactor band are used by the quantization block to keep the quantization noise below these limits.

6.5 Nonuniform Quantization

In these two blocks the scaling, quantization and Huffman coding are applied to 576 spectral values at a time. This is done iteratively in two nested loops, a *distortion control loop* (outer loop) and a *rate control loop* (inner loop).

rate control loop
The rate loop does the quantization of the frequency domain samples and thus also determines the required quantization step size. Furthermore the subdivision of the big_values into regions, the Huffman table selection decision for each region and the calculation of the boundaries between the regions take place here.

To begin with the samples are quantized with an increasing step size until the quantized values can be coded using one of the available Huffman code tables. A larger step size leads to smaller quantized values. Then the overall Huffman coded bit sum is calculated and compared with the number of bits available. If the calculated bit sum exceeds the number of bits available the quantization step size is further increased and the entire procedure is repeated until the available bits are sufficient.

The nonlinearity is achieved raising each sample to the power of 3/4.

distortion control loop
This loop controls the quantization noise which is produced by the quantization of the frequency domain lines within the rate control loop. The aim is to keep the quantization noise below the masking threshold (allowed noise given by the psychoacoustic model) for each scalefactor band.

To shape the quantization noise scalefactors are applied to the frequency lines within each scalefactor band. The scalefactors of all scalefactor bands and the quantization step size are then saved before the rate control loop in called. After the inner loop the quantization noise is calculated. This is repeated until there is no more scalefactor band with more noise than allowed by the threshold. The values of the scalefactors belonging to bands that are too noisy are increased for each iteration loop. Finally the noise caused by the quantization will not be audible by a human and the loop will exit.

There are still situations where both loops can go on forever depending on the calculated threshold. To avoid this there are several conditions in the distortion control loop that can be checked to stop the iterations more early.

Loops input:

- vector of the magnitudes of the 576 spectral values
- the allowed distortion of the scalefactor bands
- the number of scalefactor bands
- bits available for the Huffman coding and the coding of the scalefactors
- the number of bits in addition to the average number of bits, as demanded by the value of the psychoacoustic entropy for the granule.

Loops output:

- vector of 576 quantized values
- the scalefactors
- quantizer step size information
- number of unused bit available for later use
- preflag (loops preemphasis on/off)
- Huffman code related side information
    - big_values (number of pairs of Huffman coded values, excluding "count1")
    - count1table_select (Huffman code table of absolut values <= 1 at the upper end of the spectrum
    - table_select (Huffman code table of regions)
    - region0_cound and region1_count (used to calculate boundaries between regions)
    - part2_3_length

6.6 Huffman Encoding

The quantized values are Huffman coded. Each division of the frequency spectrum can be coded using different tables. The Huffman coding is one of the major reasons why the MPEG-1 Layer III retains a high quality at low bitrates.

6.7 Coding of Side Information

All parameters generated by the encoder are collected to enable the decoder to reproduce the audio signal. These are the parameters that reside in the side information part of the frame.

6.8 Bitstream Formatting CRC word generation

In this final block the defined bitstream is generated (see Chapter 5.1). The frame header, side information, CRC, Huffman coded frequency lines etc are put together to form frames. Each one of these frames represents 1152 encoded PCM samples.

7 Decoding

The encoding process is quite complex and not fully described here.

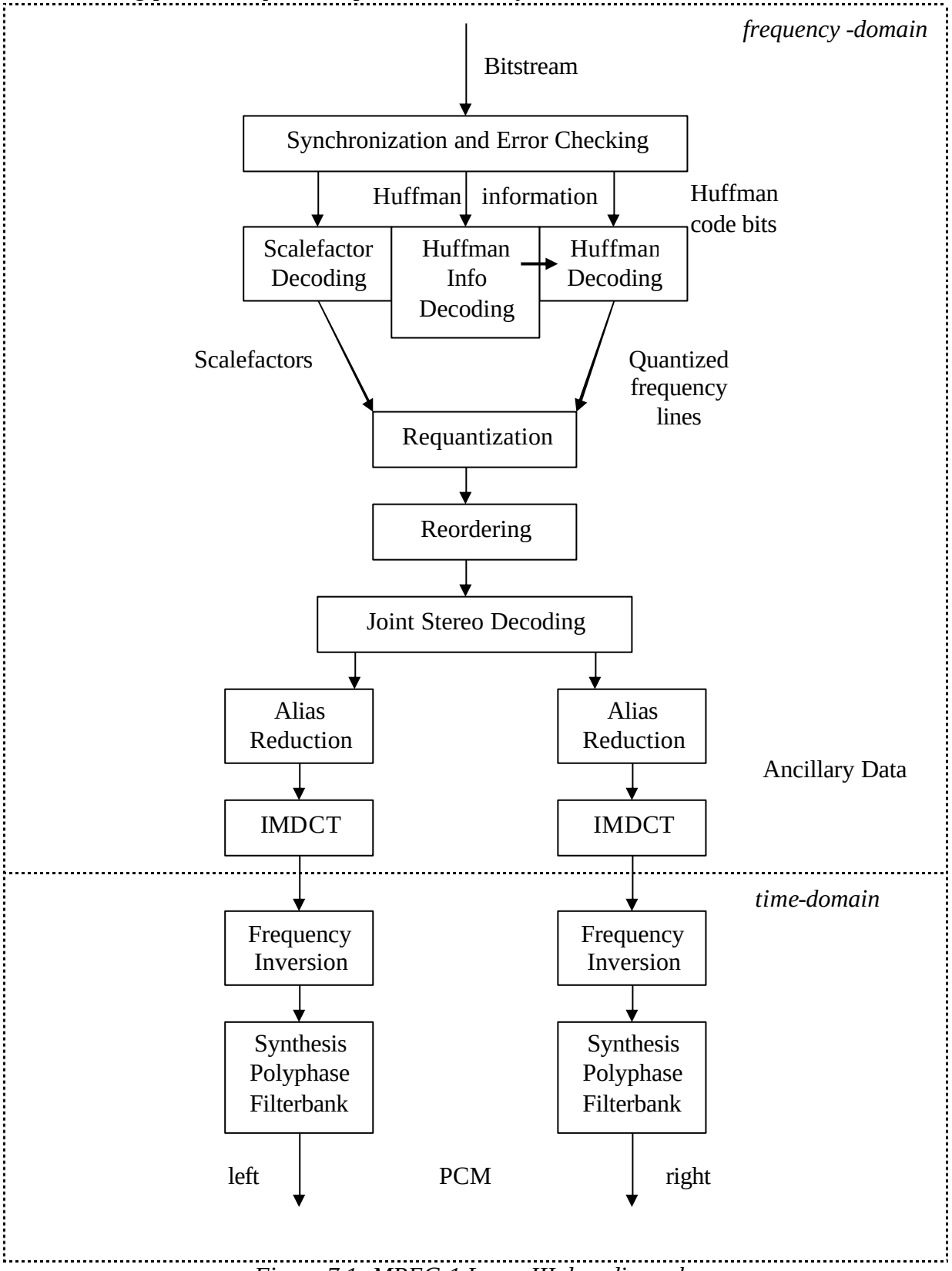

Figure 7.1: MPEG-1 Layer III decoding scheme

7.1 Sync and Error Checking

This block receives the incoming bitstream. Every frame within the stream must be identified by searching for the synchronization word. It is not possible for the following blocks to extract the correct information needed if no frames are located.

7.2 Huffman Decoding & Huffman info decoding

Since Huffman coding is a variable length coding method a single codeword in the middle of the Huffman code bits cannot be identified. The decoding must start where the codeword starts. This information is given by the Huffman info decoding block. The purpose of this block is to provide all necessary parameters by the Huffman decoding block to perform a correct decoding.

Moreover, the Huffman info decoder block must insure that 576 frequency lines are generated regardless of how many frequency lines are described in the Huffman code bits. When fewer than 576 frequency lines appear the Huffman info decoding block must initiate a zero padding to compensate for the lack of data.

7.3 Scalefactor decoding

This block decodes the coded scalefactors, i.e. the first part of the main data. The scalefactor info needed to do this is fetched from the side information. The decoded scalefactors are later used when requantizing.

7.4 Requantizer

Here the global_gain, scalefactor_scale, preflag fields in the side information contributes to restoring the frequency lines as they were generated by the MDCT block in the encoder. The decoded scaled and quantized frequency lines output from the Huffman decoder block are requantized using the scalefactors reconstructed in the Scalefactor decoding block together with some or all fields mentioned. Two equations are used depending on the window used. Both these equations are raised to the power of 4/3, which is the invers power used in the quantizer.

7.5 Reordering

The frequency lines generated by the Requantization block are not always ordered in the same way. In the MDCT block the use of long windows prior to the transformation, would generate frequency lines ordered first by subband and then by frequency. Using short windows instead, would generate frequency lines ordered first by subband, then by window and at last by frequency. In order to increase the efficiency of the Huffman coding the frequency lines for the short windows case were reordered into subbands first, then frequency and at last by window, since the samples close in frequency are more likely to have similar values.

The reordering block will search for short windows in each of the 36 subbands. If short windows are found they are reordered.

7.6 Stereo Decoding

The purpose of the Stereo Processing block is to perform the necessary processing to convert the encoded stereo signal into separate left/right stereo signals. The method used for encoding the stereo signal can be read from the mode and mode_extension in the header of each frame.

7.7 Alias Reduction

In the MDCT block within the encoder it was described that an alias reduction was applied. In order to obtain a correct reconstruction of the audio signal in the algorithms to come the aliasing artifacts must be added to the signal again. The alias reconstruction calculation consists of eight butterfly calculations for each subband, as illustrated in Figure 7.2. The constants in the figure are in the specified standard [8]. Aliasing is only applied to granules unsing short blocks.

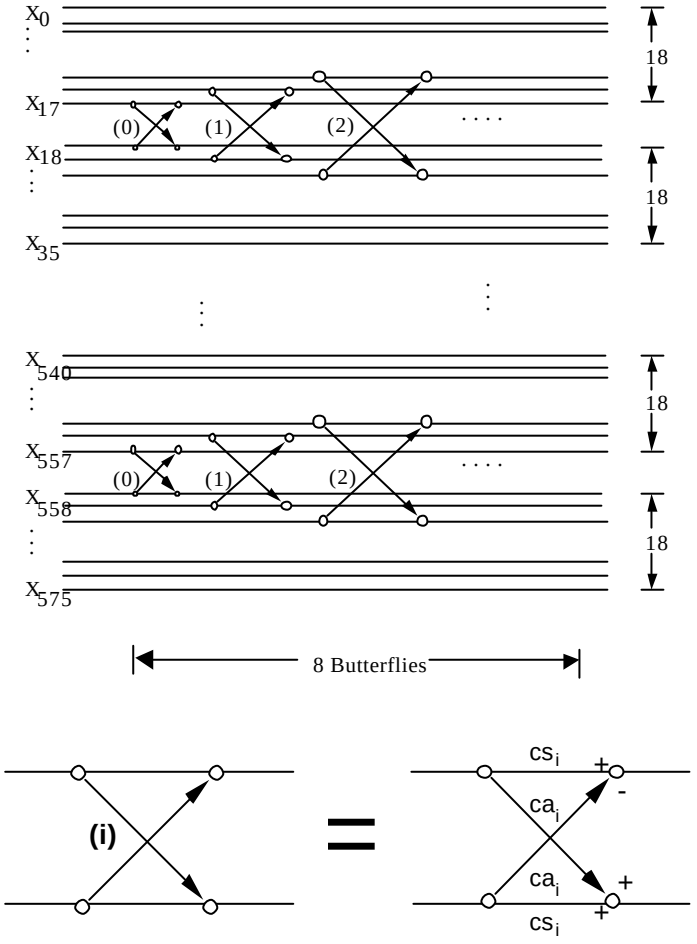

Figure 7.2: Alias reduction butterflies (source [8])

7.8 Inverse Modified Discrete Cosine Transform (IMDCT)

The frequency lines from the Alias reduction block are mapped to 32 Polyphase filter subbands. The IMDC will output 18 time domain samples for each of the 32 subbands.

7.9 Frequency Inversion

In order to compensate for frequency inversions in the synthesis polyphase filter bank, every odd time sample of every odd subband is multiplied with -1.

7.10 Synthesis Polyphase Filterbank

The synthesis Polyphase filterbank transforms the 32 subbands of 18 time domain samples in each granule to 18 blocks of 32 PCM samples, which is the final decoding result.

8 Conclusions

As assumed the audio part of the ISO MPEG-1 standard is very complex. It contains several subprocedures to achieve the optimal compression. These include the psychoacoustic model, which determines non perceptible signals, the filterbanks and cosine transforms, which effectively handles the mapping between the frequency- and the time-domains, scaling and quantization of the sample values and finally the Huffman coding. Both lossy and lossless compression has to be combined in the process. Neither of the two alone will be able to reduce the data to meet the compression demands.

This paper has provided the reader with an insight to the MPEG-1 Layer III standard and thus given a good preparation for a future encoder/decoder implementation. A decoder would be simpler to implement bearing in mind that it only has to reconstruct the bitstream and does not need to be concerned about psychoacoustics or the quality of the encoded data. Although this paper has given enough information on the frame header and side information to be able to locate the frames, read the parameters and browse through them, more detailed information regarding the subprocedures has to be examined to put this theory into practice. The MPEG-1 specification [8] is a good place to start but additional papers will probably be useful since [8] is ambiguous in some parts and lacks details in other parts. The newest version of [8] is recommended where previous typo errors have been corrected.

List of Abbreviations

ISO – International Organization for Standardization
MPEG – Moving Pictures Experts Group
PCM – Pulse Code Modulation digital audio
CD – Compact Disc
DAT – Digital Audio Tape
CRC – Cyclic Redundancy Code

References

[1] Ted Painter, Andreas Spanias, “A Review of Algorithms for Perceptual Coding of Digital Audio Signals”
[2] K. Salomonsen, “Design and Implementation of an MPEG/Audio Layer III Bitstream Processor”, Master´s thesis, Aalborg University, Denmark, 1997

[3] International Organization for Standardization webpage, http://www.iso.ch

[4] Scot Hacker, *Mp3: The Definitive Guide*, O´REILLY 2000

[5] K. Brandenburg, G. Stoll, “ISO-MPEG-1 Audio: A Generic Standard for Coding of High Quality Digital Audio”
[6] MP3´ Tech, http://www.mp3-tech.org/
[7] ID3.org, http://www.id3.org
[8] ISO/IEC 11172-3 ”Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3”
[9] M. Sieler, R Sperschneider, ” MPEG-Layer3 Bitstream Syntax and Decoding”

A Definitions (taken from the ISO 11173-2 specification)

For the purposes of this International Standard, the following definitions apply.

AC coefficient: Any DCT coefficient for which the frequency in one or both dimensions is non-zero.
access unit: in the case of compressed audio an access unit is an audio access unit. In the case of compressed video an access unit is the coded representation of a picture.
Adaptive segmentation: A subdivision of the digital representation of an audio signal in variable segments of time.
adaptive bit allocation: The assignment of bits to subbands in a time and frequency varying fashion according to a psychoacoustic model.
adaptive noise allocation: The assignment of coding noise to frequency bands in a time and frequency varying fashion according to a psychoacoustic model.
Alias: Mirrored signal component resulting from sub-Nyquist sampling.
Analysis filterbank: Filterbank in the encoder that transforms a broadband PCM audio signal into a set of subsampled subband samples.
Audio Access Unit: An Audio Access Unit is defined as the smallest part of the encoded bitstream which can be decoded by itself, where decoded means "fully reconstructed sound".
audio buffer: A buffer in the system target decoder for storage of compressed audio data.
backward motion vector: A motion vector that is used for motion compensation from a reference picture at a later time in display order.
Bark: Unit of critical band rate.
bidirectionally predictive-coded picture; B-picture: A picture that is coded using motion compensated prediction from a past and/or future reference picture.
bitrate: The rate at which the compressed bitstream is delivered from the storage medium to the input of a decoder.
Block companding: Normalizing of the digital representation of an audio signal within a certain time period.
block: An 8-row by 8-column orthogonal block of pels.
Bound: The lowest subband in which intensity stereo coding is used.
byte aligned: A bit in a coded bitstream is byte-aligned if its position is a multiple of 8-bits from the first bit in the stream.
channel: A digital medium that stores or transports an ISO 11172 stream.
chrominance (component): A matrix, block or sample of pels representing one of the two colour difference signals related to the primary colours in the manner defined in CCIR Rec 601. The symbols used for the colour difference signals are Cr and Cb.
coded audio bitstream: A coded representation of an audio signal as specified in this International Standard.
coded video bitstream: A coded representation of a series of one or more pictures as specified in this International Standard.
coded order: The order in which the pictures are stored and decoded. This order is not necessarily the same as the display order.
coded representation: A data element as represented in its encoded form.
coding parameters: The set of user-definable parameters that characterise a coded video bitstream. Bit-streams are characterised by coding parameters. Decoders are characterised by the bitstreams that they are capable of decoding.
component: A matrix, block or sample of pel data from one of the three matrices (luminance and two chrominance) that make up a picture.

compression: Reduction in the number of bits used to represent an item of data.
constant bitrate coded video: A compressed video bitstream with a constant average bitrate.
constant bitrate: Operation where the bitrate is constant from start to finish of the compressed bitstream.
Constrained Parameters: In the case of the video specification, the values of the set of coding parameters defined in Part 2 Clause 2.4.4.4.
constrained system parameter stream (CSPS): An ISO 11172 multiplexed stream for which the constraints defined in Part 1 Clause 2.4.6 apply.
CRC: Cyclic redundancy code.
Critical Band Rate: Psychoacoustic measure in the spectral domain which corresponds to the frequency selectivity of the human ear.
Critical Band: Part of the spectral domain which corresponds to a width of one Bark.
data element: An item of data as represented before encoding and after decoding.
DC-coefficient: The DCT coefficient for which the frequency is zero in both dimensions.
DC-coded picture; D-picture: A picture that is coded using only information from itself. Of the DCT coefficients in the coded representation, only the DC-coefficients are present.
DCT coefficient: The amplitude of a specific cosine basis function.
decoded stream: The decoded reconstruction of a compressed bit stream.
decoder input buffer: The first-in first-out (FIFO) buffer specified in the video buffering verifier.
decoder input rate: The data rate specified in the video buffering verifier and encoded in the coded video bitstream.
decoder: An embodiment of a decoding process.
decoding process: The process defined in this International Standard that reads an input coded bitstream and outputs decoded pictures or audio samples.
decoding time-stamp; DTS: A field that may be present in a packet header that indicates the time that an access unit is decoded in the system target decoder.
Dequantization [Audio]: Decoding of coded subband samples in order to recover the original quantized values.
dequantization: The process of rescaling the quantized DCT coefficients after their representation in the bitstream has been decoded and before they are presented to the inverse DCT.
digital storage media; DSM: A digital storage or transmission device or system.
discrete cosine transform; DCT: Either the forward discrete cosine transform or the inverse discrete cosine transform. The DCT is an invertible, discrete orthogonal transformation. The inverse DCT is defined in 2-Annex A of Part 2.
display order: The order in which the decoded pictures should be displayed. Normally this is the same order in which they were presented at the input of the encoder.
editing: The process by which one or more compressed bitstreams are manipulated to produce a new compressed bitstream. Conforming editted bitstreams must meet the requirements defined in this International Standard.
elementary stream: A generic term for one of the coded video, coded audio or other coded bit streams.
encoder: An embodiment of an encoding process.
encoding process: A process, not specified in this International Standard, that reads a stream of input pictures or audio samples and produces a valid coded bitstream as defined in this International Standard.
Entropy coding: Variable length noiseless coding of the digital representation of a signal to reduce redundancy.

fast forward: The process of displaying a sequence, or parts of a sequence, of pictures in display-order faster than real-time.
FFT: Fast Fourier Transformation. A fast algorithm for performing a discrete Fourier transform (an orthogonal transform).
Filterbank [audio]: A set of band-pass filters covering the entire audio frequency range.
Fixed segmentation: A subdivision of the digital representation of an audio signal in to fixed segments of time.
forbidden: The term 'forbidden" when used in the clauses defining the coded bitstream indicates that the value shall never be used. This is usually to avoid emulation of start codes.
forced updating: The process by which macroblocks are intra-coded from time-to-time to ensure that mismatch errors between the inverse DCT processes in encoders and decoders cannot build up excessively.
forward motion vector: A motion vector that is used for motion compensation from a reference picture at an earlier time in display order.
Frame [audio]: A part of the audio signal that corresponds to a fixed number of audio PCM samples.
future reference picture: The future reference picture is the reference picture that occurs at a later time than the current picture in display order.
Granules: 576 frequency lines that carry their own side information.
group of pictures: A series of one or more pictures intended to assist random access. The group of pictures is one of the layers in the coding syntax defined in Part 2 of this International Standard.
Hann window: A time function applied sample-by-sample to a block of audio samples before Fourier transformation.
Huffman coding: A specific method for entropy coding.
Hybrid filterbank [audio]: A serial combination of subband filterbank and MDCT.
IMDCT: Inverse Modified Discrete Cosine Transform.
Intensity stereo: A method of exploiting stereo irrelevance or redundancy in stereophonic audio programmes based on retaining at high frequencies only the energy envelope of the right and left channels.
interlace: The property of conventional television pictures where alternating lines of the picture represent different instances in time.
intra coding: Compression coding of a block or picture that uses information only from that block or picture.
intra-coded picture; I-picture: A picture coded using information only from itself.
ISO 11172 (multiplexed) stream: A bitstream composed of zero or more elementary streams combined in the manner defined in Part 1 of this International Standard.
Joint stereo coding: Any method that exploits stereophonic irrelevance or stereophonic redundancy.
Joint stereo mode: A mode of the audio coding algorithm using joint stereo coding.
layer [audio]: One of the levels in the coding hierarchy of the audio system defined in this International Standard.
layer [video and systems]: One of the levels in the data hierarchy of the video and system specifications defined in Parts 1 and 2 of this International Standard.
luminance (component): A matrix, block or sample of pels representing a monochrome representation of the signal and related to the primary colours in the manner defined in CCIR Rec 601. The symbol used for luminance is Y.
macroblock: The four 8 by 8 blocks of luminance data and the two corresponding 8 by 8 blocks of chrominance data coming from a 16 by 16 section of the luminance component of the picture. Macroblock is sometimes used to refer to the pel data and sometimes to the coded

representation of the pel and other data elements defined in the macroblock layer of the syntax defined in Part 2 of this International Standard. The usage is clear from the context.
Mapping [audio]: Conversion of an audio signal from time to frequency domain by subband filtering and/or by MDCT.
Masking threshold [audio]: A function in frequency and time below which an audio signal cannot be perceived by the human auditory system.
Masking: property of the human auditory system by which an audio signal cannot be perceived in the presence of another audio signal .
MDCT: Modified Discrete Cosine Transform.
motion compensation: The use of motion vectors to improve the efficiency of the prediction of pel values. The prediction uses motion vectors to provide offsets into the past and/or future reference frames containing previously decoded pels that are used to form the prediction.
motion vector estimation: The process of estimating motion vectors during the encoding process.
motion vector: A two-dimensional vector used for motion compensation that provides an offset from the coordinate position in the current picture to the coordinates in a reference picture.
MS stereo: A method of exploiting stereo irrelevance or redundancy in stereophonic audio programmes based on coding the sum and difference signal instead of the left and right channels.
non-intra coding: Coding of a block or picture that uses information both from itself and from blocks and pictures occurring at other times.
Non-tonal component: A noise-like component of an audio signal.
Nyquist sampling: Sampling at or above twice the maximum bandwidth of a signal.
pack: A pack consists of a pack header followed by one or more packets. It is a layer in the system coding syntax described in Part 1 of this standard.
packet data: Contiguous bytes of data from an elementary stream present in a packet.
packet header: The data structure used to convey information about the elementary stream data contained in the packet data.
packet: A packet consists of a header followed by a number of contiguous bytes from an elementary data stream. It is a layer in the system coding syntax described in Part 1 of this International Standard.
Padding: A method to adjust the average length of an audio frame in time to the duration of the corresponding PCM samples, by conditionally adding a slot to the audio frame.
past reference picture: The past reference picture is the reference picture that occurs at an earlier time than the current picture in display order.
pel aspect ratio: The ratio of the nominal vertical height of pel on the display to its nominal horizontal width.
pel: An 8-bit sample of luminance or chrominance data.
picture period: The reciprocal of the picture rate.
picture rate: The nominal rate at which pictures should be output from the decoding process.
picture: Source or reconstructed image data. A picture consists of three rectangular matrices of 8-bit numbers representing the luminance and two chrominance signals. The Picture layer is one of the layers in the coding syntax defined in Part 2 of this International Standard. NOTE: the term "picture" is always used in this standard in preference to the terms field or frame.
Polyphase filterbank: A set of equal bandwidth filters with special phase interrelationships, allowing for an efficient implementation of the filterbank.
prediction: The use of predictor to provide an estimate of the pel or data element currently being decoded.

predictive-coded picture; P-picture: A picture that is coded using motion compensated prediction from the past reference picture.
predictor: A linear combination of previously decoded pels or data elements.
presentation time-stamp; PTS: A field that may be present in a packet header that indicates the time that a presentation unit is presented in the system target decoder.
presentation unit: A decoded audio access unit or a decoded picture.
Psychoacoustic model: A mathematical model of the masking behaviour of the human auditory system.
quantization matrix: A set of sixty-four 8-bit scaling values used by the dequantizer.
quantized DCT coefficients: DCT coefficients before dequantization. A variable length coded representation of quantized DCT coefficients is stored as part of the compressed video bitstream.
quantizer scale factor: A data element represented in the bitstream and used by the decoding process to scale the dequantization.
random access: The process of beginning to read and decode the coded bitstream at an arbitrary point.
reference picture: Reference pictures are the nearest adjacent I- or P-pictures to the current picture in display order.
reorder buffer: A buffer in the system target decoder for storage of a reconstructed I-picture or a reconstructed P-picture.
reserved: The term "reserved" when used in the clauses defining the coded bitstream indicates that the value may be used in the future for ISO defined extensions.
reverse play: The process of displaying the picture sequence in the reverse of display order.
Scalefactor band: A set of frequency lines in Layer III which are scaled by one scalefactor.
Scalefactor index: A numerical code for a scalefactor.
Scalefactor: Factor by which a set of values is scaled before quantization in order to reduce quantization noise
sequence header: A block of data in the coded bitstream containing the coded representation of a number of data elements. It is one of the layers of the coding syntax defined in Part 2 of this International Standard.
Side information: Information in the bitstream necessary for controlling the decoder.
skipped macroblock: A macroblock for which no data is stored.
slice: A series of macroblocks. It is one of the layers of the coding syntax defined in Part 2 of this International Standard.
Slot [audio]: A slot is an elementary part in the bitstream. In Layer I a slot equals four bytes, in Layers II and III one byte.
source stream: A single non-multiplexed stream of samples before compression coding.
Spreading function: A function that describes the frequency spread of masking.
start codes: 32-bit codes embedded in that coded bitstream that are unique. They are used for several purposes including identifying some of the layers in the coding syntax.
STD input buffer: A first-in first-out buffer at the input of system target decoder for storage of compressed data from elementary streams before decoding.
stuffing (bits); stuffing (bytes): Code-words that may be inserted into the compressed bitstream that are discarded in the decoding process. Their purpose is to increase the bitrate of the stream.
Subband [audio]: Subdivision of the audio frequency band.
Subband filterbank: A set of band filters covering the entire audio frequency range. In Part 3 of this International Standard the subband filterbank is a polyphase filterbank.
Syncword: A 12-bit code embedded in the audio bitstream that identifies the start of a frame.

Synthesis filterbank: Filterbank in the decoder that reconstructs a PCM audio signal from subband samples.
system header: The system header is a data structure defined in Part 1 of this International Standard that carries information summarising the system characteristics of the ISO 11172 multiplexed stream.
system target decoder; STD: A hypothetical reference model of a decoding process used to describe the semantics of an ISO 11172 multiplexed bitstream.
time-stamp: A term that indicates the time of an event.
Tonal component: A sinusoid-like component of an audio signal.
variable bitrate: Operation where the bitrate varies with time during the decoding of a compressed bitstream.
variable length coding; VLC: A reversible procedure for coding that assigns shorter code-words to frequent events and longer code-words to less frequent events.
video buffering verifier; VBV: A hypothetical decoder that is conceptually connected to the output of the encoder. Its purpose is to provide a constraint on the variability of the data rate that an encoder or editing process may produce.
video sequence: A series of one or more groups of pictures.
zig-zag scanning order: A specific sequential ordering of the DCT coefficients from (approximately) the lowest spatial frequency to the highest.

B Scalefactors for 44.1 kHz, long windows (576 frequency lines)

scalefactor band	width	start index	end index
0	4	0	3
1	4	4	7
2	4	8	11
3	4	12	15
4	4	16	19
5	4	20	23
6	6	24	29
7	6	30	35
8	8	36	43
9	8	44	51
10	10	52	61
11	12	62	73
12	16	74	89
13	20	90	109
14	24	110	133
15	28	134	161
16	34	162	195
17	42	196	237
18	50	238	287
19	54	288	341
20	76	342	417

C Huffman code table 7

x y	hlen	hcod
0 0	1	1
0 1	3	010
0 2	6	001010
0 3	8	00010011
0 4	8	00010000
0 5	9	000001010
1 0	3	011
1 1	4	0011
1 2	6	000111
1 3	7	0001010
1 4	7	0000101
1 5	8	00000011
2 0	6	001011
2 1	5	00100
2 2	7	0001101
2 3	8	00010001
2 4	8	00001000
2 5	9	000000100
3 0	7	0001100
3 1	7	0001011
3 2	8	00010010
3 3	9	000001111
3 4	9	000001011
3 5	9	000000010
4 0	7	0000111
4 1	7	0000110
4 2	8	00001001
4 3	9	000001110
4 4	9	000000011
4 5	10	0000000001
5 0	8	00000110
5 1	8	00000100
5 2	9	000000101
5 3	10	0000000011
5 4	10	0000000010
5 5	10	0000000000