Encoding onto a CD

Source coding The amplitude of the sound wave is sampled at 44.1kHz (i.e. 44100 times per second) and measured on a scale of -2^{15} to $+2^{15}$, on two channels (left and right). Thus at each time interval, four bytes of information are gathered, and stored as a list of four elements of $GF(2^8)$. Six time intervals are grouped together into a *frame* of 24 bytes.

(For a 74-minute CD, the amount of information required to be stored on the disk is therefore $74 \times 60 \times 44100 \times 4 = 78321600$ bytes, i.e. about 747 megabytes.)

First encoding Next, these 24 bytes are encoded using a (shortened) Reed–Solomon code of length 28 (and dimension 24) over $GF(2^8)$. The result is that the frame now occupies 28 bytes.

Interleaving Now we write the frames as row vectors of length 28, and write 28 frames one underneath the other. We output the information in the order given by reading down the columns, instead of across the rows. (Actually, it is more complicated than this, as the successive rows are *delayed* by four frames, so that the whole process is continuous, rather than being done in one 28×28 block at a time.)

Second encoding We encode each column of length 28 now as a vector of length 32, by using another (shortened) Reed–Solomon code, of dimension 28 and length 32, again over $GF(2^8)$. Thus a frame now occupies 32 bytes, although the information has been spread around so that each input frame influences a large number of output frames.

Second interleaving We now interleave the odd-numbered bytes from one frame with the even-numbered bytes from the next frame.

Further information An extra byte is added to each frame, which can be used for control information, e.g. track numbering, etc.

The hardware The CD is written as a series of *pits* of varying lengths, with *lands* of varying lengths between them. The lengths of these pits and lands correspond to a string of 0s, while the transition between a pit and a land or vice versa corresponds to a 1. The engineering constraints are that each pit or land must be between 3 and 11 units long (inclusive).

Channel encoding Thus we must re-write our bytes (each now thought of as 8 bits, or elements of $GF(2)$) so that the 1's are always separated by between 3 and 11 0's. This is done by a look-up table which converts each byte into 14 bits, followed by 3 bits between it and the next byte. Finally, each frame of 32 bytes is followed by a 24-bit *synchronization pattern* and 3 bits to separate it from the next frame.

Decoding a CD

Most of the processes involved in encoding the CD are just reversed in the obvious way. The two Reed–Solomon codes however are used for error-correction as follows.

Decoding the inner code The first code we encounter in the decoding process is the last to be used in the encoding. It is a Reed–Solomon code of minimal distance 5, and it is used for simultaneously correcting single errors, and erasing all other errors. This means that the entire codeword is destroyed in the latter case. The interleaving, however, now ensures that the bytes which are destroyed are distributed into 28 different codewords (frames) at the next level up.

Decoding the outer code The outer code now receives codewords possibly with some bytes erased. It is used simply to reinstate these missing bytes, with no error correction at all. Since the minimal distance of the code is again 5, it can restore up to 4 erasures in each codeword.

MSMYP7
R.A.Wilson
21.03.02