LOW COST DATA COMPRESSION

MARTIN E. HELLMAN
Stanford University

Summary. The large volumes of raw data involved in telemetry often require compression prior to transmission over a limited data rate channel. If the transmitter is located in a remote location, power, reliability and cost considerations add the requirement that the encoder be fairly simple, while the decoder can be much more complex. A similar encoder-decoder complexity tradeoff exists if there are many sensing stations (and encoders) but only one receiving station (and decoder). This paper describes a technique which is well suited to such applications in that it places almost all of the computational burden on the decoder, and requires only an extremely simple encoder.

Introduction. This paper describes a method of source coding (also known as data compression) which is ideally suited to communication problems where the encoding devices must be extremely simple while the decoding devices may be much more complex. Examples of such problems include space and satellite communications, remote telemetry, and remotely piloted vehicles.

The purpose of source coding is to remove redundancy from the output of a source so that less information need be stored or transmitted. In noiseless source coding, the operation must be invertible so that the decoded data is equivalent to the original data. As a simple example, dropping the letter u after the letter q in English text is invertible since the recipient of the message can add it back in.

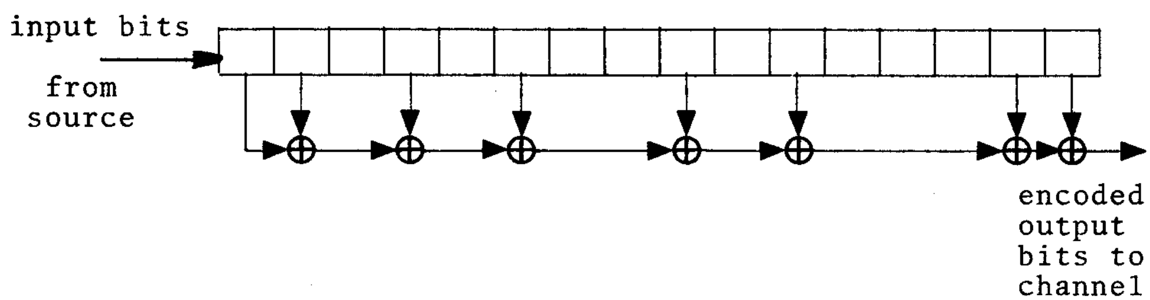
Although our primary interest is in source coding, it is easiest to understand the technique when it is used for joint source and channel encoding. Also for ease of understanding, we use written English as the data to be compressed. Channel coding adds redundancy to data prior to transmission over a noisy channel. The redundancy is added in a controlled manner so that most errors which occur on the channel can be corrected. Repetition of a message is a crude form of channel coding.

If source coding removes redundancy and channel coding adds it back in, why not dispense with both operations and use the natural redundancy of the source for error correction? A simple example shows why this cannot be done. Suppose the message “I

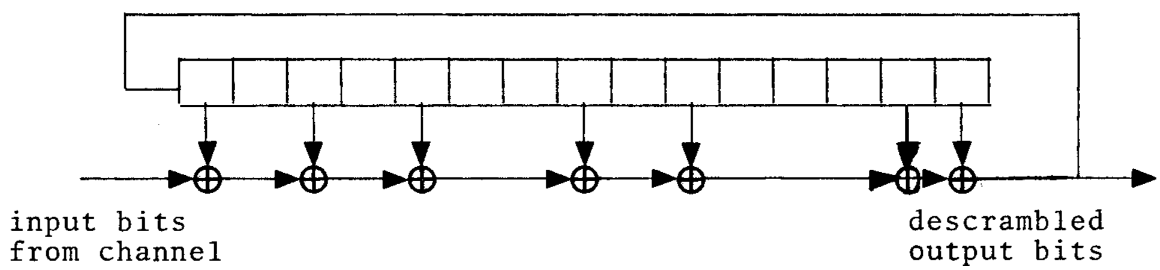
This work was supported by the U.S. Air Force Office of Scientific Research under Contract F44620-73-C-0065.

AM NOT ABLE TO PROVIDE SUPPORT." is transmitted directly and a single character is received in error: "I AM NOW ABLE TO PROVIDE SUPPORT." is received. Not only is the error overlooked, but the meaning of the message has been reversed. This is in spite of the fact that the amount of redundancy is more than sufficient to correct single errors. Although the redundancy is large it is not evenly distributed

We have developed a very simple device to transform the redundancy of the source into a usable form. This joint source and channel encoder is a convolutional encoder or scrambler. It consists of a shift register and a number of mod-2 adders (EXCLUSIVE OR gates). A typical encoder is shown below: This encoder has sixteen stages, and typically



25-100 stages will be more than adequate. The output bits are transmitted over the noisy channel and placed through a descrambler which performs the inverse operation of the encoder. The descrambler which corresponds to the above encoder is shown below.



Because of the feedback in the descrambler, even a single transmission error will cause many errors in the descrambled information. If there are enough stages in the encoder then the output following an error is complete gibberish. For example a single character error in the message used above can cause the output: "I AM NOWJ.NXAAVWM,E TYROVBGZ, RI". It is easily seen that the characters most likely to be in error are the W and the J of NOWJ... . A first attempt at decoding would be to change the J of NOWJ ... to a blank, making the text "I AM NOW...". This puts two errors in the descrambler, the first being due to the channel, the second being due to our attempted correction. Thus the output is again gibberish: "I AM NOW HU.CVKIWXRORBHUWTZHUIGK*". When we try correcting the W we find the output is meaningless except when the proper correction to a T is tried. This example used a 5 bit code with star = 00000, A 00001,... Z = 11010, blank = 11011, period = 11100, comma 11101, quote = 11110, question mark = 11111. Low order bits are encoded first. This code has lower redundancy than most codes in common use (e.g., ASCII) and therefore we expect even greater correction capabilities with these other codes.

Note the simplicity of both the encoder and the feedback descrambler, each consisting of a shift register and several EXCLUSIVE OR gates. The only complexity involved is in recognizing what is and what is not a meaningful message and in deciding which errors are most likely. It is the necessity of having such a capability which makes the receiver more complex than the transmitter. However this requirement can be met by having a sequential decoder available at the receiver. The sequential decoder need not have a complete description of the source characteristics, but of course a less complete description doesn’t allow as great an error correction capability.

If the channel is noiseless, or if separate source and channel coding operations are desired, then the convolutional code can be used at rates above 1 to effect optimal, noiseless data compression. To obtain a rate 2 code, for example, every other bit which is emitted by the rate 1 code is not transmitted. Surprisingly, the compressed data allows reliable reconstruction of the source output provided that a 2:1 compression ratio is possible with any other form of noiseless source code. The details of the proof are contained in reference [1]. Koshelev [2] and Blizard [3] independently realized the applicability of convolutional codes to data compression. Reference [4] also provides useful insights.

Koshelev [2] indicates that the decoder’s computational burden will be excessive, and efforts are under way to minimize this problem. There appears to be a gradual tradeoff between encoder and decoder complexity and we hope to be able to find a reasonable operating point for applications, such as remote telemetry, where a premium is placed on keeping the encoder’s complexity small.

1. M. E. Hellman, “Convolutional source encoding,” IEEE Trans. on Info. Theory, vol. IT-21, pp. 6Sl-656; November, 1975.

2. V. N. Koshelev, “Direct sequential encoding and decoding for discrete sources,” IEEE Trans, on Info. Theory, vol. IT-19, pp. 340-343; May, 1973.

3. R. B. Blizard, “Convolutional source coding for data compression,” Martin Marietta Corp. Denver Division, Report R-69-17, 1969.

4. M. E. Hellman, “On using natural redundancy for error detection.” IEEE Trans, on Comm, Tech., vol. COM-22, pp. 1690-1693; October, 1974.