Data Compression

4th Stage

Department of Computer Engineering

University of Diyala

References:

1. Sayood, Khalid. *Introduction to data compression*. Newnes, 2012.
2. Salomon, David. *Data compression: the complete reference*. Springer Science & Business Media, 2004.

Introduction

The last decade we have been witnessing a transformation—some call it a revolution—in the way we communicate, and the process is still under way. This transformation includes the ever-growing Internet; the explosive development of mobile communications; and the ever-increasing importance of video communication. Data compression is one of the enabling technologies for each of these aspects of the multimedia revolution.

So, what is data compression, and why do we need it? Most of you have heard of JPEG and MPEG, which are standards for representing images, video, and audio. Data compression algorithms are used in these standards to reduce the number of bits required to represent an image or a video sequence or music.

Data Compression

Data compression for (image, audio, video, text, or graphic) is the art or science of representing information in a compact form to reducing the size of the data. The primary goal of data compression is to reduce the file size. Compression is used just about everywhere for several reasons:

1) To save space when storing it.
2) To save time when transmitting it.

The task of compression consists of two components, an encoding algorithm that takes a message and generates a “compressed” representation and a decoding algorithm that reconstructs the original message or some approximation of it from the compressed representation.

Types of Data Compression

There are two major families of compression techniques in terms of the possibility of reconstructing the original source. They are called Lossless and lossy compression.

1. Lossless Compression

A compression approach is lossless only if it is possible to exactly reconstruct the original data from the compressed version. There is no loss of any information during the compression1 process. Lossless compression techniques are mostly applied to symbolic data such as character text, numeric data, computer source code and executable graphics and icons. Lossless compression techniques are also used when the original data of a source are so important that we cannot afford to lose any details. For example, medical images, text and images preserved for legal reasons; some computer executable files.

2. Lossy Compression

A compression method is lossy compression only if it is not possible to reconstruct the original exactly from the compressed version. There are some insignificant details that may get lost during the process of compression. Data such as multimedia images, video and audio are more easily compressed by lossy compression techniques.

Compression System Model

The compression system model consists of two parts: the compressor (Encoding) and the decompressor (Decoding). The compressor consists of a preprocessing stage and encoding stage, whereas the decompressor consists of a decoding stage followed by a post-processing stage Figure (1.1). Before encoding, preprocessing is performed to prepare the data for the encoding process and consists of a number of operations that are application specific. After the compressed file has been decoded, post processing can be performed to eliminate some of the potentially undesirable artifacts brought about by the compression process. Often, many practical compression algorithms are a combination of a number of different individual compression techniques.

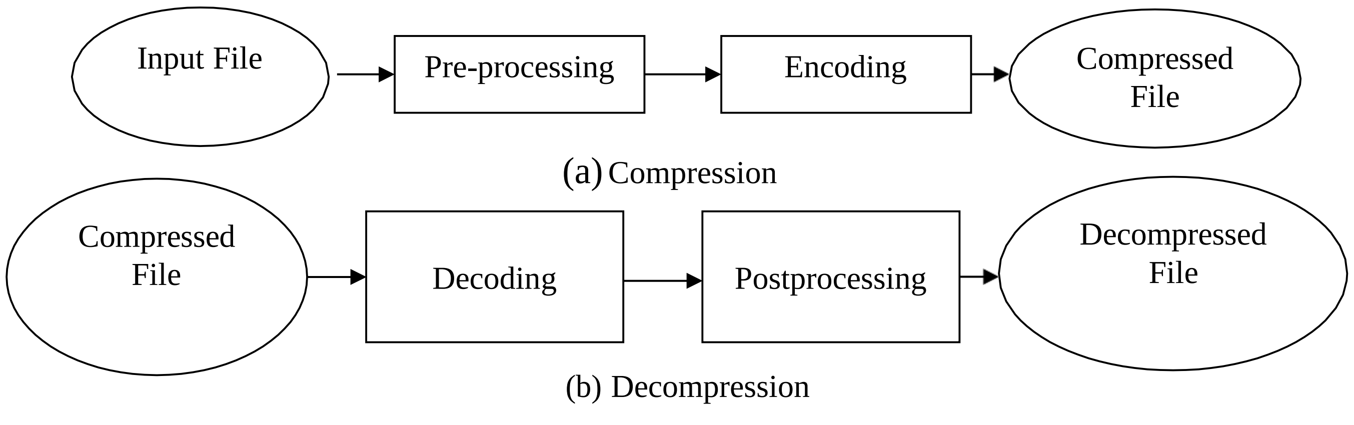

Figure 1: Compression System Model

Fidelity Criteria

The key in data compression algorithm development is to determine the minimal data required retaining the necessary information. This is achieved by taking advantage of the redundancy that exists in data. To determine exactly what information is important and to be able to measure data fidelity, we need to define data fidelity criterion. Note that the information required is application specific, and that, with lossless schemes, there is no need for a fidelity criterion. Fidelity Criteria can be divided into two classes:

1. **Objective fidelity criteria:** this fidelity is borrowed from digital signal processing and information theory and provides us with equations that can be used to measure the amount of error in the reconstructed (decompressed) data. Commonly used objective measures are the root-mean-square error (RMSE), the root-mean-square signal-to-noise ratio (SNRRMS), and the peak signal-to-noise ratio (SNRPEAK). We can define the error between an original, uncompressed pixel value and the reconstructed (decompressed) value. These objective measures areoften used in the research because they are easy to generate andseemingly unbiased, these metrics are not necessarily correlated to ourperception of data.
2. **Subjective fidelity criteria:** these criteria require the definition of a qualitative scale to assess data quality. This scale can then be used by human test subjects to determine data fidelity. In order to provide unbiased results, evaluation with subjective measures requires careful selection of the test subjects and carefully designed evaluation experiments. The subjective measures are better method for comparison of compression algorithms, if the goal is to achieve high-quality data as defined by visual perception.

Compression Performance

The performance of a compression algorithm can be measured by variouscriteria. It depends on what is our priority concern. We could measure the relative complexity of the algorithm, the memory required to implement the algorithm, how fast the algorithm performs on a given machine, and how closely the reconstruction resembles the original. A very logical way of measuring how well a compression algorithm is to compresses a given set of data and look at the difference in size of the data before the compression and size of the data after the compression. There are several ways of measuring the compression effect:

- ❖ **Compression Ratio:** This is simply the ratio of size after compression to size before compression. Values greater than 1 imply an output stream bigger than the input stream (negative compression). The compression ratio can also be called bpb (bit per bit)

Compression Ratio = size after compression / size before compression

- ❖ **Compression Factor:** This is the reverse of compression ratio. In this case, values greater than 1 indicate compression and values less than 1 imply expansion. This measure seems natural to many people, since the bigger the factor, the better the compression.

Compression Factor = size before compression / size after compression

- ❖ **Saving Percentage:** This shows the shrinkage as a percentage

Saving Percentage = size before comp. - size after comp. / size before comp. %

Example: Source image file (256 × 256) with 65,536 bytes is compressed into a file with 16,384 bytes. The compression ratio is 1/4 and the compression factor is 4. The saving percentage is: 75%.