Imperial College of Science Technology and Medicine

Department of Electrical and Electronic Engineering

Digital Image Processing

PART 4

IMAGE COMPRESSION

FUNDAMENTALS

Academic responsible

Dr. Tania STATHAKI

Room 812

Ext. 46229

Email: t.stathaki@ic.ac.uk

COMPRESSION FUNDAMENTALS

Introduction

In recent years, there have been significant advancements in algorithms and architectures for the processing of image, video, and audio signals. These advancements have proceeded along several directions. On the algorithmic front, new techniques have led to the development of robust methods to reduce the size of the image, video, or audio data. Such methods are extremely vital in many applications that manipulate and store digital data. Informally, we refer to the process of size reduction as a compression process. We will define this process in a more formal way later. On the architecture front, it is now feasible to put sophisticated compression processes on a relatively low-cost single chip; this has spurred a great deal of activity in developing multimedia systems for the large consumer market.

One of the exciting prospects of such advancements is that multimedia information comprising image, video, and audio has the potential to become just another data type. This usually implies that multimedia information will be digitally encoded so that it can be manipulated, stored, and transmitted along with other digital data types. For such data usage to be pervasive, it is essential that the data encoding is standard across different platforms and applications. This will foster widespread development of applications and will also promote interoperability among systems from different vendors. Furthermore, standardisation can lead to the development of cost-effective implementations, which in turn will promote the widespread use of multimedia information. This is the primary motivation behind the emergence of image and video compression standards.

Background

Compression is a process intended to yield a compact digital representation of a signal. In the literature, the terms *source coding*, *data compression, bandwidth compression*, and *signal compression* are all used to refer to the process of compression. In the cases where the signal is defined as an image, a video stream, or an audio signal, the generic problem of compression is to minimise the bit rate of their digital representation. There are many applications that benefit when image, video, and audio signals are available in compressed form. **Without compression, most of these applications would not be feasible!**

Example 1: Let us consider **facsimile image transmission**. In most facsimile machines, the document is scanned and digitised. Typically, an 8.5x11 inches page is scanned at 200 dpi; thus, resulting in 3.74 Mbits. Transmitting this data over a low-cost 14.4 kbits/s modem would require 5.62 minutes. With compression, the transmission time can be reduced to 17 seconds. This results in substantial savings in transmission costs.

Example 2: Let us consider a video-based CD-ROM application. **Full-motion video**, at 30 fps and a 720 x 480 resolution, generates data at 20.736 Mbytes/s. At this rate, only 31 seconds of video can be stored on a 650 MByte CD-ROM. Compression technology can increase the storage capacity to 74 minutes, for VHS-grade video quality.

Image, video, and audio signals are amenable to compression due to the factors below.

- **There is considerable statistical redundancy in the signal.**
  1. Within a single image or a single video frame, there exists significant correlation among neighbour samples. This correlation is referred to as *spatial correlation.*
  2. For data acquired from multiple sensors (such as satellite images), there exists significant correlation amongst samples from these sensors. This correlation is referred to as *spectral correlation.*
  3. For temporal data (such as video), there is significant correlation amongst samples in different segments of time. This is referred to as *temporal correlation.*

- **There is considerable information in the signal that is irrelevant from a perceptual point of view.**

- **Some data tends to have high-level features that are redundant across space and time; that is, the data is of a fractal nature.**

For a given application, compression schemes may exploit any one or all of the above factors to achieve the desired compression data rate.

There are many applications that benefit from data compression technology. Table 1.1 lists a representative set of such applications for image, video, and audio data, as well as typical data rates of the corresponding compressed bit streams. Typical data rates for the uncompressed bit streams are also shown.

Application	Data Rate	
	Uncompressed	**Compressed**
Voice 8 ksamples/s, 8 bits/sample	64 kbps	2-4 kbps
Slow motion video (10fps) framesize 176x120, 8bits/pixel	5.07 Mbps	8-16 kbps
Audio conference 8 ksamples/s, 8 bits/sample	64 kbps	16-64 kbps
Video conference (15fps) framesize 352x240, 8bits/pixel	30.41 Mbps	64-768 kbps
Digital audio 44.1 ksamples/s, 16 bits/sample	1.5 Mbps	1.28-1.5 Mbps
Video file transfer (15fps) framesize 352x240, 8bits/pixel	30.41 Mbps	384 kbps
Digital video on CD-ROM (30fps) framesize 352x240, 8bits/pixel	60.83 Mbps	1.5-4 Mbps
Broadcast video (30fps) framesize 720x480, 8bits/pixel	248.83 Mbps	3-8 Mbps
HDTV (59.94 fps) framesize 1280x720, 8bits/pixel	1.33 Gbps	20 Mbps

Table 1.1: Applications for image, video, and audio compression.

In the following figure, a systems view of the compression process is depicted.

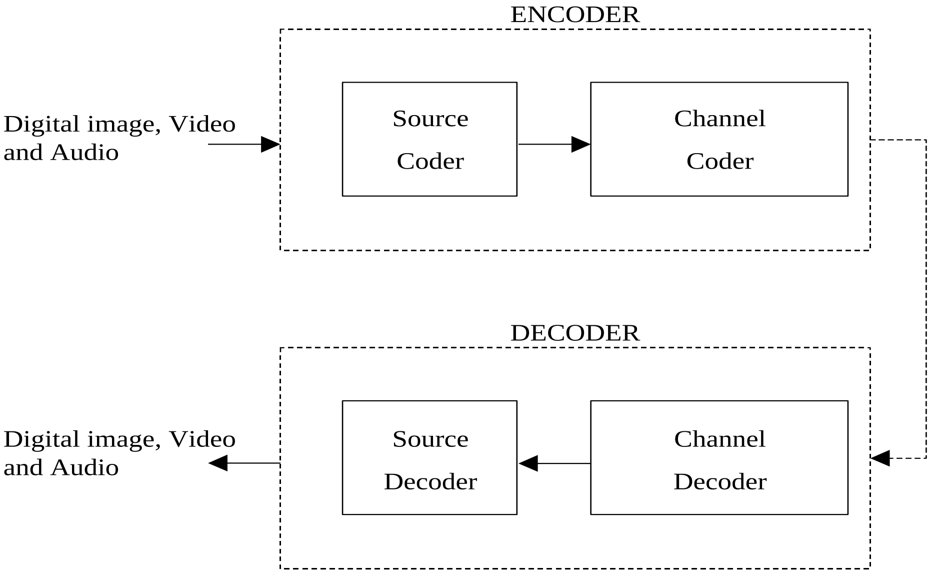


Figure 1.1 Generic compression system

The core of the encoder is the source coder. The source coder performs the compression process by reducing the input data rate to a level that can be supported by the storage or transmission medium. The bit rate output of the encoder is measured in bits per sample or bits per second. For image or video data, a pixel is the basic element; thus, bits per sample is also referred to as bits per pixel or bits per pel. In the literature, the term *compression ratio,* denoted as c_r, is also used instead of *bit rate* to characterise the capability of the compression system. An intuitive definition of c_r is

$$c_r = \frac{source\,coder\,input\,size}{source\,coder\,output\,size}$$

This definition is somewhat ambiguous and depends on the data type and the specific compression method that is employed. For a still-image, size could refer to the bits needed to represent the entire image. For video, size could refer to the bits needed to represent one frame of video. Many compression methods for video do not process each frame of video, hence, a more commonly used notion for size is the bits needed to represent one second of video.

In a practical system, the source coder is usually followed by a second level of coding: the channel coder (Figure 1.1). The channel coder translates the compressed bit stream into a signal suitable for either storage or transmission. In most systems, source coding and channel coding are distinct processes. In recent years, methods to perform combined source and channel coding have also been developed. Note that, in order to reconstruct the image, video, or audio signal, one needs to reverse the processes of channel coding and source coding. This is usually performed at the decoder.

From a system design viewpoint, one can restate the compression problem as a bit rate minimisation problem, where several constraints may have to be met, including the following:

- **Specified level of signal quality**. This constraint is usually applied at the decoder.
- **Implementation complexity**. This constraint is often applied at the decoder, and in some instances at both the encoder and the decoder.
- **Communication delay**. This constraint refers to the end to end delay, and is measured from the start of encoding a sample to the complete decoding of that sample.

Note that, these constraints have different importance in different applications. For example, in a two-way teleconferencing system, the communication delay might be the major constraint, whereas, in a television broadcasting system, signal quality and decoder complexity might be the main constraints.

Lossless versus lossy compression

Lossless compression

In many applications, the decoder has to reconstruct without any loss the original data. For a lossless compression process, the reconstructed data and the original data must be identical in value for each and every data sample. This is also referred to as a reversible process. In lossless compression, for a specific application, the choice of a compression method involves a trade-off along the three dimensions depicted in Figure 1.2; that is, coding efficiency, coding complexity, and coding delay.

Coding Efficiency

This is usually measured in bits per sample or bits per second (bps). Coding efficiency is usually limited by the information content or *entropy* of the source. In intuitive terms, the entropy of a source X provides a measure for the "randomness" of X. From a compression theory point of view, sources with large entropy are more difficult to compress (for example, random noise is very hard to compress).

Coding Complexity

The complexity of a compression process is analogous to the computational effort needed to implement the encoder and decoder functions. The computational effort is usually measured in terms of memory requirements and number of arithmetic operations. The operations count is characterised by the term millions of operations per second and is often referred to as MOPS. Here, by operation, we imply a basic arithmetic operation that is supported by the computational engine. In the compression literature, the term MIPS (millions of instructions per second) is sometimes used. This is specific to a computational engine's architecture; thus, in this text we refer to coding complexity in terms of MOPS. In some applications, such as portable devices, coding complexity may be characterised by the power requirements of a hardware implementation.

Coding Delay

A complex compression process often leads to increased coding delays at the encoder and the decoder. Coding delays can be alleviated by increasing the processing power of the computational engine; however, this may be impractical in environments where there is a power constraint or when the underlying computational engine cannot be improved. Furthermore, in many applications, coding delays have to be constrained; for example, in interactive communications. The need to constrain the coding delay often forces the compression system designer to use a less sophisticated algorithm for the compression processes.

From this discussion, it can be concluded that these trade-offs in coding complexity, delay, and efficiency are usually limited to a small set of choices along these axes. In a subsequent section, we will briefly describe the trade-offs within the context of specific lossless compression methods.

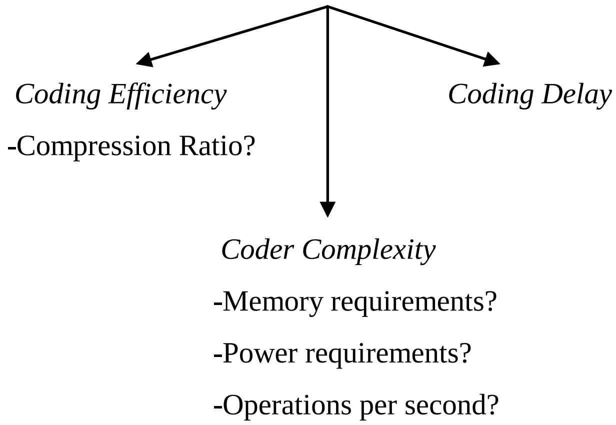


Figure 1.2 Trade-offs in lossless compression.

Lossy compression

The majority of the applications in image or video data processing do not require that the reconstructed data and the original data are identical in value. Thus, some amount of loss is permitted in the reconstructed data. A compression process that results in an imperfect reconstruction is referred to as a lossy compression process. This compression process is irreversible. In practice, most irreversible compression processes degrade rapidly the signal quality when they are repeatedly applied on previously decompressed data.

The choice of a specific lossy compression method involves trade-offs along the four dimensions shown in Figure 1.3. Due to the additional degree of freedom, namely, in the signal quality, a lossy compression process can yield higher compression ratios than a lossless compression scheme.

Signal Quality This term is often used to characterise the signal at the output of the decoder. There is no universally accepted measure for signal quality.

One measure that is often cited is the signal to noise ratio SNR, which can be expressed as

$$SNR = 10\log_{10} \frac{encoder\,input\,signal\,energy}{noise\,signal\,energy}$$

The noise signal energy is defined as the energy measured for a hypothetical signal that is the difference between the encoder input signal and the decoder output signal. Note that, SNR as defined here is given in decibels (dB). In the case of images or video, $PSNR$ (peak signal-to-noise ratio) is used instead of SNR. The calculations are essentially the same as in the case of SNR, however, in the numerator, instead of using the encoder input signal one uses a hypothetical signal with a signal strength of 255 (the maximum decimal value of an unsigned 8-bit number, such as in a pixel).

High SNR or $PSNR$ values do not always correspond to signals with perceptually high quality. Another measure of signal quality is the mean opinion score, where the performance of a compression process is characterised by the subjective quality of the decoded signal. For instance, a five point scale such as *very annoying, annoying, slightly annoying, perceptible but not annoying,* and *imperceptible* might be used to characterise the impairments in the decoder output.

In either lossless or lossy compression schemes, the quality of the input data affects the compression ratio. For instance, acquisition noise, data sampling timing errors, and even the analogue-to-digital conversion process affects the signal quality and reduces the spatial and temporal correlation. Some compression schemes are quite sensitive to the loss in correlation and may yield significantly worse compression in the presence of noise.

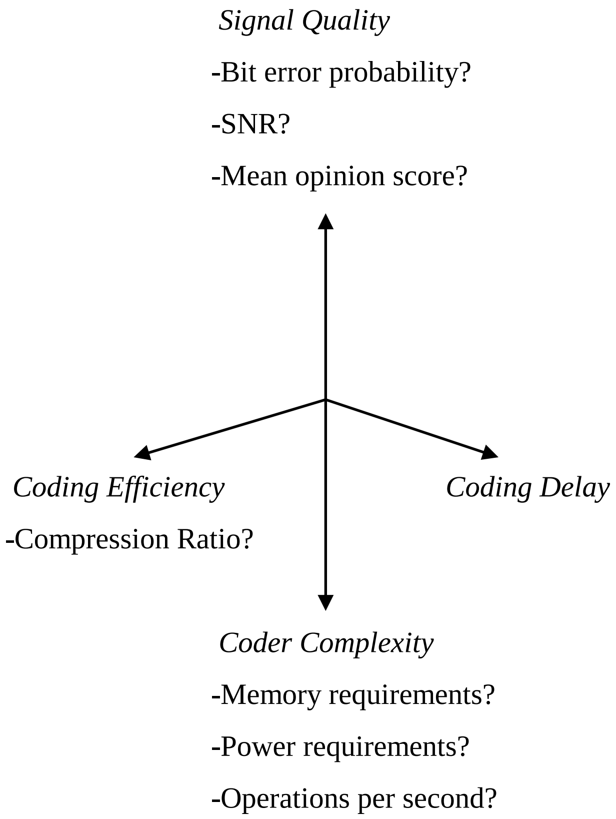


Figure 1.3 Trade-offs in lossy compression.

Issues in compression method selection

In this chapter, we have introduced some fundamental concepts related to image, video, and audio compression. When choosing a specific compression method, one should consider the following issues:

- Lossless or lossy. This is usually dictated by the coding efficiency requirements.

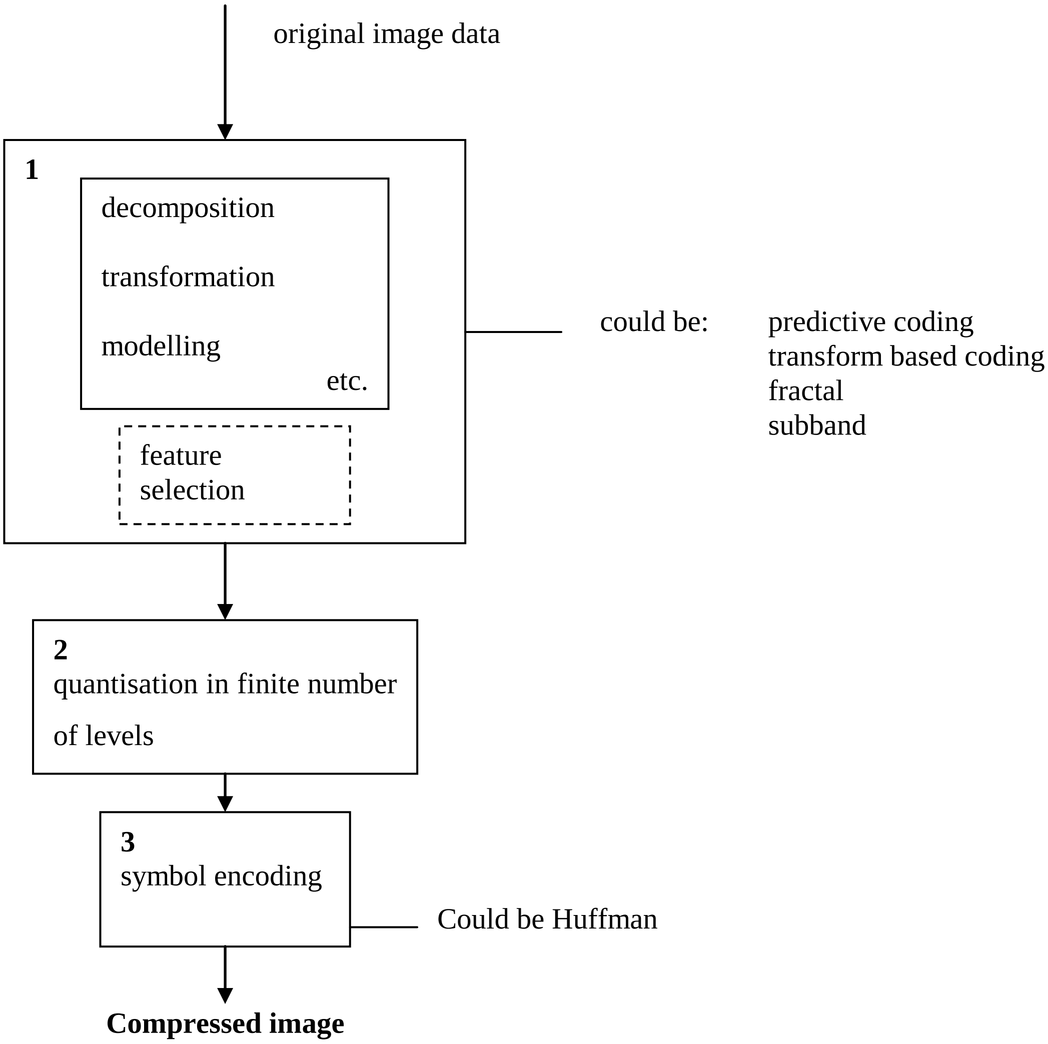
- Coding efficiency. Even in a lossy compression process, the desirable coding efficiency might not be achievable. This is especially the case when there are specific constraints on output signal quality.
- Variability in coding efficiency. In some applications, large variations in coding efficiency among different data sets may not be acceptable.
- Resilience to transmission errors. Some compression methods are more robust to transmission errors than others. If retransmissions are not permitted, then this requirement may impact on the overall encoder- decoder design.
- Complexity trade-offs. In most implementations, it is important to keep the overall encoder-decoder complexity low. However, certain applications may require only a low decoding complexity.
- Nature of degradations in decoder output. Lossy compression methods introduce artifacts in the decoded signal. The nature of artifacts depends on the compression method that is employed. The degree to which these artifacts are judged also varies from application to application. In communication systems, there is often an interplay between the transmission errors and the coding artifacts introduced by the coder. Thus, it is important to consider all types of error in a system design.
- Data representation. In many applications, there is a need to support two decoding phases. In the first phase, decoding is performed to derive an intelligible signal; this is the case in data browsing. In the second phase, decoding is performed to derive a higher quality signal. One can generalise this notion to suggest that some applications require a hierarchical representation of the data. In the compression context, we refer to such compression schemes as *scalable compression methods.* The notion of scalability has been adopted in the compression standards.
- Multiple usage of the encoding-decoding tandem. In many applications, such as video editing, there is a need to perform multiple encode-decode operations using results from a previous encode-decode operation. This is not an issue for lossless compression; however, for lossy schemes, resilience to multiple encoding-decoding cycles is essential.
- Interplay with other data modalities, such as audio and video. In a system where several data modalities have to be supported, the compression methods for each modality should have some common elements. For instance, in an interactive videophone system, the audio compression method should have a frame structure that is consistent with the video frame structure. Otherwise, there will be unnecessary requirements on buffers at the decoder and a reduced tolerance to timing errors.
- Interworking with other systems. In a mass-market environment, there will be multiple data modalities and multiple compression systems. In such an environment, transcoding from one

compression method to another may be needed. For instance, video editing might be done on a frame by frame basis; hence, a compression method that does not exploit temporal redundancies might be used here. After video editing, there might be a need to broadcast this video. In this case, temporal redundancies can be exploited to achieve a higher coding efficiency. In such a scenario, it is important to select compression methods that support transcoding from one compressed stream format to another. Interworking is important in many communications environments as well.

THE SOURCE CODER

In this course we are interested in exploring various compression techniques referring to the **source coder** only, where the image compression takes place.

A general procedure for image data compression is shown in the following block diagram (Figure 1.4). **This is the source coder of the previous diagram!**



ELEMENTS OF INFORMATION THEORY

- Any information generating process can be viewed as a source that emits a sequence of symbols chosen from a finite alphabet (for example, text: ASCII symbols; n -bit images: 2^n symbols).
- Simplest form of an information source: *discrete memoryless source* (DMS). Successive symbols produced by such a source are statistically independent.
- A DMS is completely specified by the source alphabet $S = \{s_1, s_2, \ldots, s_n\}$ and the associated probabilities $\{p_1, p_2, \ldots, p_n\}$.
- *Self Information:*

$$I(s_i) = \log_2 \frac{1}{p_i} = -\log_2 p_i$$

  - the occurrence of a less probable event provides more information
  - the information of independent events taken as a single event equals the sum of the information

- *Average Information* per Symbol or *Entropy* of a DMS:

$$H(S) = \sum_{i=1}^{n} p_i I(s_i) = -\sum_{i=1}^{n} p_i \log_2 p_i \text{ bits/symbol}$$

- Interpretation of Entropy:
  - Average amount of information per symbol provided by the source (definition)
  - Average amount of information per symbol an observer needs to spend to remove the uncertainty in the source
- $N-$th extention of the DMS: Given a DMS of size n, group the source into blocks of N symbols. Each block can now be considered as a single source symbol generated by a source S^N with alphabet size n^N. In this case

$$H(S^N) = N \times H(s)$$

Noiseless Source Coding Theorem

Let S be a source with alphabet size n and entropy $H(S)$. Consider coding blocks of N source symbols into binary codewords. For any $\delta > 0$, it is possible by choosing N large enough to construct a code in such a way that the average number of bits per original source symbol l_{avg} satisfies

$$H(s) \le l_{avg} \le H(s) + \delta$$