# LO8 – Data Compression

The transmission and storage of multimedia material requires compression. All compression schemes require two algorithms: one for compression (encoding) and one for decompression (decoding).

Considerations regarding compression:

* The algorithms for encoding and decoding may be **asymmetrical**. For example, when compressing a movie, it may be acceptable to have a slow/expensive system for encoding and a fast/cheap system for decoding. The reason is that the movie is only encoded once, but decoded every time the movie is viewed. On the other hand, video conferencing would require fast/cheap algorithms for encoding/decoding (**symmetrical**).
* The encode/decode process might not be **invertible**. For example, when encoding an image, the image might not be 100% identical to the original. In some cases, this is acceptable and the system is said to be **lossy** (for example, JPEG, MP3). If input and output are a 100% match, the system is said to be **lossless** (for example, PNG).

Compression schemes can be divided into general categories: **Entropy** encoding and **Source** encoding.

## Entropy encoding

Entropy encoding techniques generally manipulate the bit stream **without** regard to what the bits mean. It is usually lossless (fully reversible).

Example:

1. **Run-length encoding**. In many forms of multimedia data, repeated symbols are common. In audio, silence is represented as a string of 0’s. In video, runs of the same color are common in things such as the sky. These repeated symbols can be replaced by a special marker (not otherwise allowed in the data), followed by the number of times that it occurred. Consider the string of data:

21200000000000065711111111111111

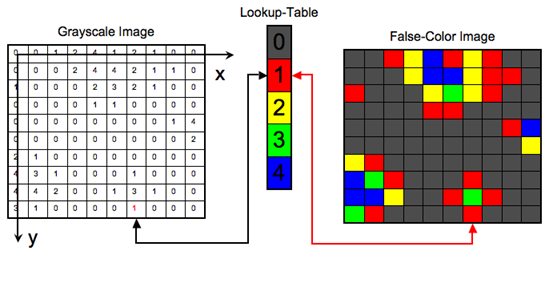
212A0012657A1014

Special marker A is in red.

Repeating character is in blue.

Number of occurrences is in green and consists of 3 digits in this case (padded with 0’s if necessary).

1. **Color lookup table**. Consider an RGB image encoded with 3 bytes per pixel (24-bit color). In theory, there are as many as 224 colors (approximately 16 million), but in practice, there may be fewer values in a specific image. For example, suppose a picture is made up of 256 actual colors. Each pixel is then represented as an index into the color table (requiring only 8 bits, since 28 = 256).



(color-by-number for computers)

## Source Encoding

Source encoding takes advantage of the properties of the data to produce compression. It is usually lossy. Example: MP3 files or other audio compression techniques will eliminate parts of the sound that humans can’t usually hear to reduce storage (using psychoacoustics).

Examples of source encoding:

1. Differential encoding – a sequence of values (audio signal) is encoded by representing each value as the difference from the previous value. It is lossy since the signal might jump so much between consecutive values that the difference does not fit in the field for expressing differences(The filed must cost less space). This is an example of source encoding because large jumps between consecutive data points are unlikely. Assume the following example stores differences in a single digit:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Raw data | 112 | 98 | 105 | 110 |
| Compressed data | 112 | -9 | 2 | 5 |
| Reconstructed data | 112 | 103  (some error) | 105 | 110 |

1. Transformations – signals are transformed from one domain to another to achieve compression. For example, consider the case of a two-dimensional grayscale image of size 4×4 pixels.

|  |  |  |  |
| --- | --- | --- | --- |
| 160 | 160 | 161 | 160 |
| 161 | 165 | 166 | 158 |
| 160 | 167 | 165 | 161 |
| 159 | 160 | 161 | 171 |

These data points can be transformed by subtracting the value in the upper-left corner from all pixels except itself.

|  |  |  |  |
| --- | --- | --- | --- |
| 160 (8 bits) | 0 (4 bits) | 1 (4 bits) | 0 (4 bits) |
| 1 | 5 | 6 | -2 |
| 0 | 7 | 5 | 1 |
| -1 | 0 | 1 | 11 (4 + 8 = 12 bits) |

This type of transformation would be useful especially if **variable length encoding** is used. For example, difference from -7 to +7 could be encoded with 4-bit numbers. Differences outside this range could be stored in 8 bits. The value of -8 would be used as a special marker indicating that an 8-bit difference follows.

1. Vector Quantization – source encoding directly related to image data. In this case, the image is divided up into rectangles. In addition to the image itself, a table of rectangles is also used. This is called a **code book**. Each rectangle is transmitted by looking it up in the code book and just sending the index into the table rather than the rectangle itself. If a code book is created dynamically (per some set of images), then the code book is transmitted as well. If a small number of rectangles dominates the image, a large compression factor can be achieved.