**ACROPOLIS INSTITUTE OF TECHNOLOGY AND RESEARCH INDORE**



SUBJECT: Computer graphics and multimedia

TOPIC: how lossy and lossless compression techniques are work for multimedia compression

Submitted to: Submitted By:

Mr. Anil Patidar Vartika Khaddar(0827CS161239)

Varsha chawla(0827CS161238)

**Data Compression**

Data compression is a way of saving more data in a certain amount of computer storage, it is trying to save a quart in a pint pot! This is possible in computer systems as most data has redundancy which can be exploited to reduce its size.

The principal disadvantage is that it requires the computer to compress and decompress the data which takes time and resource. Another disadvantage is that if part of a compressed file becomes damaged it will be virtually impossible to recover any part of the file, whereas with an uncompressed file some portion of it can usually be recovered.

**Lossy and Lossless Compression**

Data compression techniques come in two forms: lossy and lossless.

Generally a lossy technique means that data is saved approximately rather than exactly. If two types of data look roughly alike they are deemed the same. They are then given the same code and therefore require less space to store. However, when they are recovered the original differences between the versions are lost. This type of technique is mainly employed for multimedia files. Examples of the standards are JPEG for pictures and MP3 for sounds.

In contrast lossless techniques save data exactly. They look for sequences that are identical and code these. This type of compression has a lower compression rate then a lossy technique, but when the file is recovered it is identical to the original. This type of technique is used for text and database files which must be exact although techniques do exist for multimedia files. The most used lossless standard for data is the ZIP format.

**Lossy vs Lossless Compression**

In an ideal world you would not wish to compress files at all, and with the plummeting cost of storage this may be the way forward as you can keep the original data pristine. With uncompressed files if you lose some part of the file the rest of it can still be seen.

Lossless files also preserve the original data but anyone who has used the ZIP format to pack files will observe that the levels of compression are generally modest. Lossless formats such as PNG for pictures generally produce big files.

Sending uncompressed and lossless files over the internet is very expensive. Even with fast broadband there is still a finite amount of bandwidth available. So although storage is cheap the cost of sending the files is very high.

The lossy techniques such as JPEG for pictures can achieve high levels of compression with virtually no noticeable loss of quality. Figures vary but compression down to 10% of the original size has virtually no loss. However there are two problems:

* If an image contains very high contrast such as black lines against a white background, then JPEG artifacts - or distortion - can be seen,
* If an image is to be edited then each time it is saved in JPEG format more distortion will occur. It is better to edit the picture in lossless formats such as PNG and then save it as a JPEG when it has completed.

**Understanding How the Techniques Work**

**Simple Lossless Compression**

Data compression conceptually follows three principles:

1. Find repeating patterns in a file.
2. Replace these patterns with a reference to a dictionary entry.
3. Create a dictionary of the repeating patterns.

The reduction in file size comes about due to replacing repeating sequences by the dictionary reference. However, the references add back an overhead by increasing the file size.

Lossless and lossy data compression both use this basic structure where they differ is in the methods they use to implement it. To illustrate this and the next three articles will deal with two lossless methods and two lossy ones.

# Complex Lossless Compression (Pt 1)

The article on ["Simple Lossless Data Compression"](https://www.coleyconsulting.co.uk/simple_lossless_compression.htm) showed that using straightforward rules the sample document could be compressed down to 73% of its size. However, within the dictionary there are several character strings which are repeated making for a large dictionary.

The most noticeable repeated long phrase is "we shall fight" which has 3 entries in the dictionary. Among the smaller words "the" appears in 4 entries in the dictionary. If these could be reduced then a smaller dictionary would result.

Another characteristic to check is just because a string is repeated does it result in a saving in file size. An example of this is the sequence "once" which appears twice. If it was kept in the file it would use 6 characters. However, putting it in the dictionary require 3 characters for the phrases, 1 character for the dictionary code and 2 characters as code tags making a total of 6 and thus no reduction in size. This has an overhead of creating a bigger dictionary and requiring more processing for no benefit.

**Lossy Compression (Pt 1)**

An example of lossy data compression is the JPEG standard for storing pictures. The reason this standard is called "lossy" is because a picture can be saved into smaller and smaller files with on each occasion the image degrading with the structure still visible but the details getting lost. This means that when the file is recreated it is not identical to the original.

How does this happen? It is done by making similar types of pattern identical. The more aggressive the rules for making patterns similar then the smaller the file, but the larger the differences.

The JPEG standards and other standards are mathematically complex but use underlying basic principles. These principles can be understood by looking at how a text file would appear if logically similar techniques to lossy file compression are applied to it.

This article will emulate this by producing a lossy version of the Churchill war time speech "**We shall fight on the beaches**".

**Lossy Repeating Patterns Rules**

The key difference between Lossless and Lossy file compression is that in Lossy the exact sequence is not retained by near enough is used instead. For a computer this is done by applying rules in a similar manner to the two Lossless techniques. In fact Lossy techniques will retain exact patterns where possible.

The rules I am using for the Lossy technique are similar but different from the Complex Lossless.

1. Start by finding the largest sequence of characters that nearly repeats at least once using these guidelines:
   * Ignore capitalisation when comparing strings.
   * Ignore changes in punctuation.
   * Ignore additional spaces.
   * Allow up to one character difference between sequences.
2. Compare smaller sequences with larger ones to see if the larger one can be split following the same guidelines.
3. Choose the most common sequence as the substitute for all the similar sequences.
4. Look at smaller sequences until you reach three character sequences. (as before)
5. Only accept a sequence if there it will result in an at least two character reduction in file size.

The fifth rule means that to be viable there must be 3 repetitions of a 3 or 4 character sequence. For all greater sequence sizes 2 repetitions are enough.