WIKIPEDIA

Run-length encoding

Run-length encoding (**RLE**) is a very simple form of lossless data compression in which *runs* of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs. Consider, for example, simple graphic images such as icons, line drawings, and animations. It is not useful with files that don't have many runs as it could greatly increase the file size.

RLE may also be used to refer to an early graphics file format supported by CompuServe for compressing black and white images, but was widely supplanted by their later Graphics Interchange Format. RLE also refers to a little-used image format in Windows 3.x, with the extension **rle**, which is a Run Length Encoded Bitmap, used to compress the Windows 3.x startup screen.

Example

For example, consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text. A hypothetical scan line, with B representing a black pixel and W representing white, might read as follows:

```
WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWWWWWWWWWWWWWWWWBWWWWWWWWWWWWWW
```

With a run-length encoding (RLE) data compression algorithm applied to the above hypothetical scan line, it can be rendered as follows:

```
12W1B12W3B24W1B14W
```

This can be interpreted as a sequence of twelve Ws, one B, twelve Ws, three Bs, etc.

The run-length code represents the original 67 characters in only 18. While the actual format used for the storage of images is generally binary rather than ASCII characters like this, the principle remains the same. Even binary data files can be compressed with this method; file format specifications often dictate repeated bytes in files as padding space. However, newer compression methods such as DEFLATE often use LZ77-based algorithms, a generalization of run-length encoding that can take advantage of runs of strings of characters (such as `BWWBWWBWWBWW`).

Run-length encoding can be expressed in multiple ways to accommodate data properties as well as additional compression algorithms. For instance, one popular method encodes run lengths for runs of two or more characters only, using an "escape" symbol to identify runs, or using the character itself as the escape, so that any time a character appears twice it denotes a run. On the previous example, this would give the following:

```
WW12BWW12BB3WW24BWW14
```

This would be interpreted as a run of twelve Ws, a B, a run of twelve Ws, a run of three Bs, etc. In data where runs are less frequent, this can significantly improve the compression rate.

One other matter is the application of additional compression algorithms. Even with the runs extracted, the frequencies of different characters may be large, allowing for further compression; however, if the run lengths are written in the file in the locations where the runs occurred, the presence of these numbers interrupts the normal flow and makes it harder to compress. To overcome this, some run-length encoders separate the data and escape symbols from the run lengths, so that the two can be handled independently. For the example data, this would result in two outputs, the string "`WWBWWBBWWBWW`" and the numbers (`12,12,3,24,14`).

History and applications

Run-length encoding schemes were employed in the transmission of television signals as far back as 1967.[1] It is particularly well suited to palette-based bitmapped images such as computer icons, and was a popular image compression method on early online services such as CompuServe before the advent of more sophisticated formats such as GIF.[2] It does not work well at all on continuous-tone images such as photographs, although JPEG uses it quite effectively on the coefficients that remain after transforming and quantizing image blocks.

Common formats for run-length encoded data include Truevision TGA, PackBits, PCX and ILBM. The ITU also describes a standard to encode run-length-colour for fax machines, known as T.45.[3] The standard, which is combined with other techniques into Modified Huffman coding, is relatively efficient because most faxed documents are generally white space, with occasional interruptions of black.

See also

- Kolakoski sequence
- Look-and-say sequence
- Comparison of graphics file formats
- Golomb coding
- Burrows–Wheeler transform
- Recursive indexing
- Run-length limited
- Bitmap index
- Forsyth–Edwards Notation, which uses run-length-encoding for empty spaces in chess positions.
- DEFLATE

References

1. Robinson, A. H.; Cherry, C. (1967). "Results of a prototype television bandwidth compression scheme". *Proceedings of the IEEE*. IEEE. **55** (3): 356–364. doi:10.1109/PROC.1967.5493 (https://doi.org/10.1109%2FPROC.1967.5493).
2. Dunn, Christopher (1987). "Smile! You're on RLE!" (http://csbruce.com/cbm/transactor/pdfs/trans_v7_i06.pdf) (PDF). *The Transactor*. Transactor Publishing. **7** (6): 16–18. Retrieved 2015-12-06.
3. *Recommendation T.45 (02/00): Run-length colour encoding* (http://www.itu.int/rec/T-REC-T.45). International Telecommunication Union. 2000. Retrieved 2015-12-06.

External links

- Run-length encoding implemented in different programming languages (http://rosettacode.org/wiki/Run-length_encoding) (on Rosetta Code)