Abstract

Arithmetic coding is a very efficient and most popular entropy coding technique. Arithmetic coding method is based on the fact that the cumulative probability of a symbol sequence corresponds to a unique subinterval of the initial interval $[0, 1)$. In this method, when encoding a symbol, it first computes new interval $[\text{low}, \text{high})$ based on cumulative probability segment of the symbol. Thereafter it iterates in a loop to output code bits till the interval becomes $2^b-2$ wide, where $b$ is number of bits used to store range of an interval. In conventional implementation of arithmetic coding algorithm, in single loop iteration, only one bit is processed at a time. When most significant bit (msb) of low and high of a subinterval matches, it writes this msb in coded message and doubles the interval by extracting msb. When underflow occurs, it extracts second msb to expand an interval. Processing such single bit and expanding an interval is also called renormalization in a loop. In this paper, an upgradation of this conventional arithmetic coding algorithm is proposed, wherein more than one bit is processed at a time instead of just single bit in single iteration. This proposed multi-bit processing arithmetic coding algorithm is implemented here to reduce the iterations needed in renormalizing an interval. It is observed that processing multiple output bits at a time leads to big improvement in execution time. To determine the number of maximum possible matching most significant bits to output, two alternatives are used here; (i) Using shift operation in a loop (ii) Using log function. It is found
that first technique is far better than second one with respect to execution time. As compared to conventional implementations processing single bit at a time, about 52% overall saving in execution time is observed when processing multi-bits using shift operation in a loop; whereas about 31% overall loss in performance is observed with the technique of using log function. We have also tried these two alternative ways to determine the number of consecutive occurrences of underflow and process them all in single iteration; but it has not shown any significant gain in speed. As expected, in using any of the above methods, there is no compromise in compression ratio.

References

- Ranjan Bose, Saumitr Pathak, "A Novel Compression and Encryption Scheme
- Boris Ryabko and Jorma Rissanen, &quot;Fast Adaptive Arithmetic Code for Large Alphabet Sources With Asymmetrical Distributions;&quot;, IEEE COMMUNICATIONS LETTERS, VOL. 7, NO. 1, JANUARY 2003 pp. 33-35
- I. Mengyi Pu, Fundamental Data Compression, Butterworth-Heinemann, 2006
- A. Drozek, Elements of data compression, Brooks/Cole, 2002
- Jyotika Doshi, Savita Gandhi, &quot;Improved Performance Of Arithmetic Coding By Extracting Multiple Bits At A Time;&quot;, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181, Vol. 1 Issue 8, October – 2012

Index Terms

Computer Science
Algorithms

Keywords
Arithmetic coding   computing number of output bits  computing number of consecutive occurrences of underflow

faster execution

lossless data compression

multi-bit processing

renormalizing interval