

Design an Algorithm for Data Compression using Pentaoctagesimal SNS

Debasis Das
Assistant Professor
VMV Com, JMT Arts & JJP Science College
Wardhaman Nagar, Nagpur

U. A. Lanjewar, PhD.
Professor
VMV Com, JMT Arts & JJP Science College
Wardhaman Nagar, Nagpur

ABSTRACT

Compression is the science and art of representing the information in a compact form rather than its original or uncompressed form in data storage or transmission. Data compression is storing data in a format that requires less space than usual. Nowadays, data compression is a common requirement in telecommunication industry. Since last few decades, the research is continuing in the field of compression and various data compression techniques are already developed to compress different data formats. But, a data compression program using strange number system (specially using pentaoctagesimal SNS) tries to compact the data in some special format, such that the data occupies less disk space or that it can be transmitted in less time. A decompression program restores the information again. This paper mainly describes a simple compression and decompression process which is free from time complexity.

Keywords

Strange Number System, Pentaoctagesimal SNS, Data Compression.

1. INTRODUCTION

Data compression is one of the most important fields and tools in modern computing. The development of data compression procedures starts immediately after the appearance of the first computer systems. At any time available resources like memory or data transfer capabilities were limited or an adequate investment was required. This circumstance has not changed up to now. Moreover, the need for data compression has been strengthened due to larger and more complex contents. Nowadays almost every multimedia data as audio or video will be used in a compressed form. This trend is supported by more powerful hardware components being available.

In the meantime a variety of procedures is available specially designed to fulfill the different requirements. Selecting a proper compression procedure is always a compromise that tries to reconcile contrary characteristics.

Data compression is a technique used on computer systems and in modern database to compress data, such that the data occupies less storage space, or can be transmitted in shorter time. It is the process of encoding data so that it takes less storage space or less transmission time than it would if it were not compressed. The compressor has to compact the data in such a way that a decompressor can restore the compacted data to its original form, without any distortion. One characteristic of data compression is that the compressed information cannot be used directly again: a decompression technique is required to restore the information in its original form [4].

Data compression is classified in two broad categories: Lossless Compression and Lossy Compression. Lossless compression reduces bits by identifying and eliminating statistical redundancy without lost any information. Lossless compression is used in cases where it is important that the original and the decompressed data be identical, or where deviations from the original data could be deleterious. Another type of data compression is *lossy* compression, where a small amount of distortion is allowed between the decompressed data and the original data. Lossy compression is most commonly used to compress multimedia data (audio, video, and still images), especially in applications such as streaming and internet telephony. Lossy algorithms achieve better compression effectiveness than lossless algorithms, but lossy compression is limited to audio, images, and video, where some loss is acceptable. There are quite a few lossless compression techniques nowadays, and most of them are based on dictionary or probability and entropy [2].

Finally, data compression techniques have to be universal. This means that they should be able to process all types of data found on computer systems. It is impossible for a data compression program to compress all possible computer files, but at least it should be able to compress a wide range of different types of files.

This is a new data compression algorithm using pentaoctagesimal strange number system. It is dependent about the repetition of base conversion. Though the compression ratio of this algorithm is not so much but the differential technique makes the compression ratio to a constant value. The main advantage of this algorithm is that it can compress the output file which is produced after applying certain compression techniques on a file. So this algorithm will play a significant role in data compression when it is used for hybridization with other algorithm.

2. RELATED WORK

Data Compression is a method using which the storage space required for storing data is reduced with the help of mathematical techniques. Data compression is also referred to as source coding. This is the process of encoding data information using as few bits as possible compared to the unencoded data.

In the more technical and mathematical sense, data compression is applying certain algorithms in order to reduce bits in a data file. Most computer software applications for compressing data use a variation of the LZ adaptive dictionary-based algorithm in reducing file sizes without changing the meaning of the data. "LZ" refers to the name of the creators of the algorithm, Lempel and Ziv.

There two main types of data compression namely lossless compression and lossy compression. Lossless compression

guarantees that what is compressed can be recovered without any data loss. As a developer, you wouldn't be too impressed if your backup archive corrupted your files. Lossy compression though affects photographs and video where some loss of image quality is unnoticeable and so acceptable. An uncompressed jpg can easily be compressed to half its size or smaller without it being noticed. Too much compression though and it starts pixellating with small square blobs appearing.

Some examples of lossless data compression include entropy encoding, Burrows-Wheeler Transform, Prediction by Partial Matching (also known as PPM), Dictionary Coders (LZ77 & LZ78 and LZW), Dynamic Markov Compression (DMC) [8], Run-length encoding and context mixing. Examples of lossy data compression include vector quantization, A-law Compander, Mu-law Compander, Distributed Source Coding Using Syndromes (for correlated data), Discrete Cosine Transform, Fractal compression, Wavelet compression, Modulo-N code for correlated data and linear predictive coding [1].

There have been efforts earlier to apply different compression techniques to compress text on servers; but only a few of the compression techniques like deflate and LZ algorithm [7] have been successful. The latest release of IIS 5.0 (Internet Information servers) on Windows 2000 by Microsoft has integrated the above methods. It has the capacity to compress the HTML stream statically and on demand. These methods have reported an increase in performance of the server by 400% and have reduced network traffic significantly. Other services like Email have embedded this compression technique [5] into the POP3 and SMTP servers. News and FTP servers use none of the above method during transmission. These algorithms were successful because of the fact that, they are fast and achieve a substantial compression ratio without using significant amount of resources (memory, I/O and time) on the server. The compression achieved by these algorithms is not high compared to the best available in the literature, they are faster and they utilize fewer resources on the server. But, all these systems lack the capability to know when to compress the data and when not to and what methods should be used to compress the data [10].

Data compression is particularly useful in communications because it enables devices to transmit or store the same amount of data in fewer bits. There are a variety of data compression techniques, but only a few have been standardized. The CCITT has defined a standard data compression technique for transmitting faxes (Group 3 standard) and a compression standard for data communications through modems (CCITT V.42bis). In addition, there are file compression formats, such as ARC and ZIP [6].

3. SNS AND IT'S IMPORTANCE

Presently in computer science and technology, number system is based on some traditional number system viz. decimal (base 10), binary (base-2), octal (base-8) and hexadecimal (base 16). However, except the traditional number system, there are also some other number systems, those are not as widely known or widely used as traditional number system in computing, but they have charms all their own having a genuine mathematical distinction in its favour. The numbers in **strange number system (SNS)** are those numbers which poses some extra features than the numbers of traditional number system (TNS) viz. decimal (base 10), binary (base-2), octal (base-8) and hexadecimal (base 16). Some of the strange

numbers are unary, ternary, ..., Nonary, ..., unodecimal, ..., vigesimal... sexagesimal, etc.

As we have seen that, not only traditional numbers are used in digital world, but there are some strange numbers, which are also very common and frequently used in most of the digital technologies and devices. Due to the benefits of strange number representation, which include greater speed of arithmetic operations realization, greater density of memorized information, better usage of transmission paths and decreasing of pin number of integrated circuits, this paper concludes that strange number system even though they are not yet more commercially available, remain a viable field for research, and have a promising future as a replacement for traditional number system.

Today, the complexity of traditional number system is steadily increasing in computing. Due to this fact, strange number system is investigated for efficiently describing and implementing in digital systems. Purpose to review those research papers are to understand the strange number system in depth and their awareness and detailed explanation is necessary for understanding various digital aspects. A basic motivation of this paper is to implement various applications using strange number system in the field of Computer Science and technology.

Although many researcher and knowledge seeker know only the traditional number system such as decimal, binary, octal and hexadecimal and are very comfortable with performing operations using this system, it is important for them to understand that traditional number system is not the only system. By studying other number system such as quadrovigesimal (base-24), hexavigesimal (base-26), heptavigesimal (base-27), trigesimal (base-30), duotrigesimal (base-32), hexatrigesimal (base-36), quadragesimal (base-40), pentagesimal (base-50), sexagesimal (base-60), duosexagesimal (base-62), quadroxagesimal (base-60), pentaogesimal (base-85) and octovicesimal (base-128), researcher will gain a better understanding of how strange number systems work in general [3].

The revolution of digital technologies has changed the way human beings number representation and application from traditional number system to strange number system. In this day and age, most people tend to use strange number system in the various field of computing to perform a variety of tasks. The invention of ternary computer is the best example of such changes to science and technology.

Since last few decades strange number system has been possible alternative to binary logic. Unfortunately, the development of strange number system was not keeping up with the speed of the binary counterparts. But the strange number system (SNS) does have a genuine mathematical distinction in its favor. By one plausible measure, it is the most efficient of all integer bases; it offers the most economical way of representing numbers. The potential advantages of strange number system distinguish them from the traditional number system and make them worth an extra look; some of these features include:

- Greater speed of arithmetic operations realization
- Greater density of memorized information
- Better usage of transmission paths
- Decreasing of interconnections complexity and interconnections area
- Decreasing of pin number of integrated circuits and printed boards
- Avoid sign problem and zero redundancy problem

The modern digital computer normally deals with the traditional number (i.e. binary, octal, decimal and hexadecimal) as per as computer science and information technology is concern. Apart from these traditional number systems, the strange number system also plays a significant role in computing. The strange number system poses some extra features which distinguish them from the traditional number systems and make them worth an extra look. In this paper, we mainly focused to develop a novel concept for data encoding and decoding using strange number system.

3.1 Pentaoctagesimal SNS

The number system with base eighty five is known as the pentaoctagesimal number system. In this system eighty five symbols are used to represent numbers and these are numerals 0 through 9, capital alphabets A through Z, small alphabets a through z and special symbol !, #, \$, %, &, (,), *, +, -, ;, <, =, >, ?, @, ^, _', {, |, }, and ~. It is also a positional number system that each bit position corresponds to a power of 85. It has two parts the integral part or integers and the fractional part or fractions, set a part by radix point. For example $(4&d.P3)_{85}$

In pentaoctagesimal number system the leftmost bit is known as most significant bit (MSB) and the right most bit is known as least significant bit (LSB). The following expression shows the position and the power of the base 85 [3]:

$$\dots 85^3 85^2 85^1 85^0 . 85^{-1} 85^{-2} 85^{-3} \dots$$

The arithmetic operations like addition, subtraction, multiplication and division operations of decimal numbers can be also performed on pentaoctagesimal numbers. This number system is used in ASCII 85 encoding system to encode binary data to text format.

4. DATA COMPRESSION USING PENTAOGTAGESIMAL SNS

In computer science and information theory, data compression, involves encoding information using fewer bits than the original representation. Data compression is useful because it helps reduce the consumption of expensive resources, such as hard disk space or transmission bandwidth. On the downside, compressed data must be decompressed to be used, and this extra processing may be detrimental to some applications. The goal of data compression is to represent an information source (e.g. a data file, a speech signal, an image, or a video signal) as accurately as possible using the fewest number of bits.

4.1 Methodology

At first calculate the total length of the text and read four characters at a time. Then calculate ASCII value of each four character individually and concatenated them. Now, convert concatenated result number of times into pentaoctagesimal

number system whenever the converted result is greater than pentaoctagesimal number system and also calculated, how many times pentaoctagesimal base conversion is performed. Finally represent these two results using the symbol of pentaoctagesimal number system as a compressed data. To get the best compression higher base is very necessary.

4.2 Algorithm

4.2.1 Steps for Compression

1. Start
2. Calculate the length of the text and store it into a variable
3. Set count=0
4. Do until (end of the file)
5. Read the text and pick the four characters from the text
6. Calculate the ASCII value of each character and concatenate them and store into a variable b
7. Set count=0
8. Do until (b > 85)
9. Set s = ""
10. Do until (b > 0)
11. r = b % 85
12. Concatenate s and r and store it into s
13. Perform integer division between b and 85 and store the result into b
14. End of the loop
15. count = count + 1
16. End of the loop
17. Convert the value of count into string type and concatenate it with s and store the result into str
18. Calculate the length of str and store it into L
19. Set I = 1
20. Do until (I <= L)
21. Read two digit from str and symbolized them using Base 85 number system
22. I = I + 2
23. End of the loop
24. End of the loop
25. Result file is compressed file
26. End

4.2.2 Steps for Decompression

1. Start
2. Calculate the length of the compressed text and store it into L
3. Set s=""
4. Set I = 1
5. Do until (I<=L)

6. Read one symbol from compressed text and convert them using base 85 and store it into str
7. Concatenate s and str and store the result into s
8. Set $n = \text{str}$
9. $I=I+2$
10. End of the loop
11. Set $k=1$
12. $s1=s$
13. Set $S2=0$
14. Set $t=0$
15. Do until ($k \leq n$)
16. Calculate the length of s1 and store it into p
17. Set $q=1$
18. Do until ($q \leq p$)
19. Read two digit from s and store it into r
20. $s2 = s2 + (\text{Math.Pow}(85, t)) * r$
21. $q=q+2$
22. $t=t+1$
23. End of the loop
24. $s1=s2$
25. $k=k+1$
26. End of the loop
27. Calculate the length of s1 and store it into p
28. Set $m=1$
29. Do until ($m \leq p$)
30. Read the digit and converted it into character
31. $m=m+1$
32. End of the loop
33. Extract will be decompressed file
34. End

4.3 Algorithm Illustration

For example consider a text like 'EARTH IS ROUND'. Then four characters from the beginning is 'E', 'A', 'R', 'T' and their ascii value is 69, 65, 82, 84 respectively. After concatenation the number is 69658284. Now convert this number into pentaoctagesimal number system whenever converted number is not less than 85. For this above number you have to convert 20 times to the number less than 85 i.e., 54. Then represent these two numbers using pentaoctagesimal number system, the compressed text is 'sK'. So by representing in this way the total character size of compressed file will be 2, whereas the original size of the text was 4. Now in the decompression process, convert the compressed text into the decimal value twenty times. So, after conversion the original number is 69658284. Now separate two bits and represent them into character to get back the original text. In such a way decompression process can be done.

5. USE OF DATA COMPRESSION IN TODAY'S CONTEXT

Data compression is used just about everywhere especially in modern database systems. Compression can be utilized in database systems for different reasons including: 1) Reducing storage/archival costs, which is particularly important for large data warehouses 2) Improving query workload performance by reducing the I/O costs 3) Reducing manageability costs by reducing the time taken and storage costs for backup, recovery and log shipping.

Data compression in dynamic systems has several applications in the real world. It is very useful in two main areas: resource management and data transmission. With data compression, consumption of expensive resources like hard disk can be greatly reduced. But the downside to this is that compressed data often needs extra processing for decompressing so extra hardware may be needed [9].

In terms of transmission, compressed data will help save bandwidth and as result; a company may not need to spend extra money for bandwidth. But as with any communication, a protocol need exists between the sender and receiver to get the message across.

Data compression is also widely used in backup utilities, spreadsheet applications, and database management systems. Certain types of data, such as bit-mapped graphics, can be compressed to a small fraction of their normal size.

6. CONCLUSION

In this paper a new algorithm is used for data compression using strange number system. The potential advantage of this algorithm is its simplicity. It is a simple compression and decompression process, is used to reduce the size of text files. The technique of 'reduce bits' is used in this algorithm. Since every character is taken care of, so the output codes do not depend upon the repetition, like most of the other compression algorithms. Data compression using strange number system is reduced to reduce a larger combination of characters by a fewer numbers of bits. After the code formation, ASCII replaces the pentaoctagesimal numbers, which finally reduces the text size. Though a lot of research and findings are already happened in the field of data compression but there are no such algorithms in data compression that lay emphasis on differential compression based on strange number system and bit reduction.

7. ACKNOWLEDGMENTS

I would like to thanks Professor Ujwal A. Lanjewar sir for his valuable advice, support and constant encouragement. His constant criticisms and reviews gave me the conceptual clarity.

8. REFERENCES

- [1] Senthil Shanmugasundaram, Robert Lourdasamy, "A Comparative Study Of Text Compression Algorithms", International Journal of Wisdom Based Computing, Vol. 1 (3), December 2011.
- [2] Debashish Chakraborty, Sandipan Bera, Anil Kumar Gupta, Soujit Mondal, "Simple Data Compression by Differential Analysis using Bit Reduction and Number System Theory", ACEEE Int. J. on Information Technology, Vol. 01, No. 03, Dec 2011.

- [3] Debasis Das, Dr U A Lanjewar, “ Exploring Strange Number System: Latent Talent to be used in place of Traditional Number System”, *International Journal of Advances in Science and Technology*, Vol. 3, No.1, 2012.
- [4] Debasis Das, Dr U A Lanjewar, “Strange Number System: An Enhancing Tool for Data Encryption and Decryption”, *International Journal of Advanced Research in Computer Science*, Volume 3, No. 2, March-April 2012.
- [5] T.SubhamastanRao, M.Soujanya, T.Hemalatha, T.Revathi, “Simultaneous Data Compression and Encryption”, *International Journal of Computer Science and Information Technologies*, Vol. 2 (5) , 2011, 2369-2374.
- [6] H. Kruse and A. Mukherjee. “Data Compression Using Text Encryption”, *Proc. Data Compression Conference*, 1997, IEEE Computer Society Press, 1997, p. 447.
- [7] Dr. V. K. Govindan, B. S. Shajee mohan: An Intelligent Text Data Encryption and Compression for High Speed and Secure Data Transmission Over Internet. NIT Calicut, Kerala.
- [8] Mohammad Al-laham, Ibrahiem M. M. El Emary: Comparative Study between Various Algorithms of Data Compression Techniques. *IJCSNS*, Vol. 7 No. 4, April 2007.
- [9] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2007.
- [10] Demijan Klinc, Carmit Hazay, Ashish Jagmohan, Hugo Krawczyk, and Tal Rabin, “On Compression of Data Encrypted with Block Ciphers”, Draft, September 22, 2009.