Volume 5, Issue 1, January 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Message Communication System for Smart Phones Using Modified Huffman Compression

Miss Poonam P. Bhamare, Prof. Kiran. P. Gaikwad

Department of Computer Engineering SND COE & RC,

Yeola, Savitribai Phule Pune University, Maharashtra, India

Abstract— Data compression is a common requirement for most of the computerized applications. There are number of data compression algorithms, which are dedicated to compress different data formats. Even for a single data type there are number of different compression algorithms, which use different approaches. Mobile communication devices have become popular tools for gathering and disseminating information and data. This paper proposes an efficient data compression technique by doing some modification to Huffman Coding. The aim of this paper is to compress data of any size to send through Android based mobiles to optimize bandwidth .

Keywords— Huffman Coding, compression, android, mobile communication

I. INTRODUCTION

Compression implies reducing the quantity of data used to represent a file, image or video content without excessively reducing the quality of the original data. It also reduces the number of bits required to store and/or transmit digital media. To compress something means that you have a piece of data and you decrease its size.

Huffman coding is a type of lossless data compression technique. Huffman coding is based on the frequency of occurrence of a data item i.e. frequency of occurrence of characters in text message. The technique is to use a lower number of bits to encode the data in to binary codes that occurs more frequently.

Now a days mobile communication is becoming very popular for getting and giving information and data. In normal Huffman coding the tree generation and coding or decoding both the processes are done at the same side. In this paper we are going to do some modification in the normal Huffman code such that the tree will get generated at the server side and compression and decompression will be done at the client side with the help of tree generated by server.

II. LITERATURE SURVEY

Detlev Marpe, *Member, Ieee*, Heiko Schwarz, And Thomas Wiegand (July 2003) have reviewed on "context-based adaptive binary arithmetic coding in the h.264/avc video compression standard" and they conclude that the CABAC entropy coding method is part of the main profile of h.264/avc and may find its way into video streaming, broadcast, or storage applications within this profile. Experimental results have shown the superior performance of CABAC in comparison to the baseline entropy coding method of vlc/cavlc. For typical test sequences in broadcast applications, averaged bit-rate savings of 9% to 14% corresponding to a range of acceptable video quality of about 30–38 db were obtained.

Abu Shamim Mohammad Arif, Asif Mahamud & Rashedul Islam (2009) have studied on "an enhanced static data compression scheme of bengali short message". The prime objective of this research technique is to establish a low complexity compression scheme suitable for small devices having small memory and relatively lower processing speed. The basic aim is not to compress text of any size up to its maximum level without having any constraint on space and time; rather than the main target is to compress short messages up to an optimal level which needs minimum space, consume less time and the processor requirement is lower. We have implemented character masking, dictionary matching, associative rule of data mining and hyphenation algorithm for syllable based compression in hierarchical steps to achieve low complexity lossless compression of text message for any mobile devices.The scheme to choose the diagrams are performed on the basis of extensive statistical model and the static huffman coding is done through the same context.

Radu Radescu (November 2009) have studied on "transform methods used in lossless compression of text files". He concludes the compression technique based on BWT provides good results in comparison with the general-purpose compressors. The algorithm has a high degree of generality and could be applied on the majority of file types (text, image or other files). The text files used for star transform and LIPT are either extracted from the calgary corpus or specially conceived (in the case of romanian text). The size of the compressed files was compared with the original files size. The performed tests with and without star transform and LIPT pointed out an important increasing in compression performance (especially) in the case of the Huffman-based compressor. The Huffman compression encodes the most frequent character (*) with a one-bit codeword. Improvements of up to 33% for star transform and up to 7% for LIPT were obtained for the huffman algorithm. As an average result, star transform and LIPT prove their utility as lossless text compression preprocessing methods.

Ibrahim Akman, Hakan Bayindir, Serkan Ozleme, Zehra Akin, And Sanjay Misra(August 2010) have worked on "lossless text compression technique using syllable based morphology". In this paper, they present a new lossless text compression technique which utilizes syllable-based morphology of multi-syllabic languages. The proposed algorithm is designed to partition words into its syllables and then to produce their shorter bit representations for compression. The method has six main components namely source file, filtering unit, syllable unit, compression unit, dictionary file and target file. The number of bits in coding syllables depends on the number of entries in the dictionary file. The proposed algorithm is implemented and tested using 20 different texts of different lengths collected from different fields. The results indicated a compression of up to 43%.

Ahmed S. Musa, Ayman Al-Dmour And Mansour I. Irshid (2010) had worked on "an efficient text compression technique based on using bitwise lempel-ziv algorithm" and they get analysis and simulation results obtained based on using the modified compression technique show that the bitwise lz-78 encoder of the fourth-order extended binary source, which includes 16 symbols, achieves compression efficiency close to that of the conventional lz-78 encoder, which includes 256 symbols.

Parul Bhanarkar & Nikhil Jha had worked on "SMS text compression through IDBE (intelligent dictionary based encoding) for effective mobile storage utilization" they given the objective involved is designing a semantic dictionary based on intelligent dictionary based encoding (IDBE) which provides a high text compression ratio to utilize the space in phone's memory. When SMS file will be received, english words present in the text will be replaced by the respective short words in the designed semantic dictionary. Thus replacing english words by the respective short forms reduces the space occupied by the SMS file. The paper describes the IDBE compression techniques for SMS text compression.

III. IMPLEMENTATION DETAILS

A. Relevent Mathematics:

Input:

Alphabet $A=\{a_1, a_2,...,a_n\}$, which is the symbol alphabet of size n.

Set $W=\{w_1,w_2,...,w_n\}$, which is the set of the symbol weights (usually proportional to probabilities), i.e. $w_i=\text{weight}(a_i)$, $1\leq i\leq n$.

Output:

Code $C(A,W)=\{c_1, c_2,...,c_n\}$, which is the set of codeword , where c_i is the codeword for a_i, $1\leq i\leq n$

Goal:

Let $L(C)=\sum w_i \times \text{length}(c_i)$ be the weighted path length of code C . Condition: $L(C) \leq L(T)$ for any code T(A,W) .

B. System Architecture

This section introduces system architecture. The block diagram of system is shown in Fig 3.1 which reveals the details of the system components.

1) Server:

Server will

a. Generate Huffman's Tree.
b. Authentiate Useres.
c. Store Messages if user is not online.
d. check the content of messages to update huffman's tree.

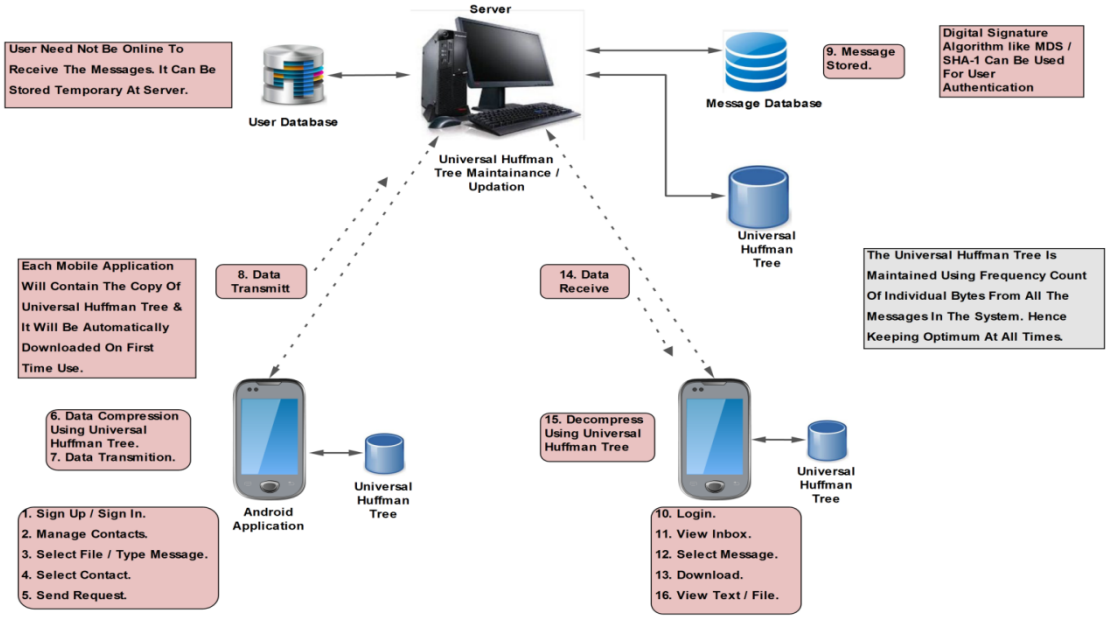

Fig 3.1. Proposed System Architecture.

2) Sender:
Sender will
   a. Sing up/ Sign in.
   b. Download huffmans tree form server and compress message using it.
   c. Transmit the message.

3) Receiver:
Receiver will
   a. Sing up/ Sign in.
   b. Download the message form server intended for it.
   c. Download huffmans tree form server and decompress message using it.

C. Modification in Huffman Coding

- Generation of Huffman tree is at server.
- No need of transmitting Huffman tree along with compressed data.
- Tree will be downloaded at each user when user wants to communicate.
- Tree will be generated and updated as per frequency of character change.

IV. CONCLUSION

We are developing a mobile application for communicating all types of file by modifying the Huffman code as generating tree at server side and by coding and decoding data to be transferred at client side. Because of such modification the bit required to transfer message will be reduced as tree will not get transferred at each send and receive. There will be reduction in time complexity as tree is generated at once and will updated with each version. Arithmetic coding can be viewed as a generalization of Huffman coding; indeed, in practice arithmetic coding is often preceded by Huffman coding, as it is easier to find an arithmetic code for a binary input than for a nonbinary input. Huffman coding is in wide use because of its simplicity, high speed and lack of encumbrance by patents. Huffman coding today is often used as a "back-end" to some other compression method. DEFLATE (PKZIP's algorithm) and multimedia codes such as JPEG and MP3 have a front-end model and quantization followed by Huffman coding. This modification will help to transfer all types of files with less numbers of bits and time complexity will get reduced.

REFERANCES

[1] Witten, Ian H. Neal, Radford M. Cleary, John G. 1987. Arithmetic Coding for Data Compression.
[2] Lelewer, Debra A. Hirschberg, Daniel S. *Data Compression.*
[3] Mahmoud, Tarek M et al. *Hybride* Compression Encryptions Technique for Securing SMS.
[4] Sameh Ghwanmeh,Riyad Al-Shalabi and Ghassan Kanaan "Efficient Data Compression Scheme using Dynamic Huffman code Applied on Arabic Language", Journal of Computer Science,2006
[5] Tomas Kuthan and Jan Lansky "Genetic algorithms in syllable based text compression", Dateso 2007.
[6] Abdel-Rahman Jaradat and Mansour I. Irshid, 2000, "An efficient bit-wise source encoding technique based on source mapping", Devices, Circuits and Systems, Proceedings of the 2000 Third IEEE International Caracas Conference.
[7] Abdel-Rahman Jaradat and Mansour I. Irshid, 2001, "A simple binary run-length compression technique for non-binary sources based on source mapping", Active and Passive Electronic Components, 24(4):211-221.doi:10.1155/2001/23505.
[8] Bell, T., J. Cleary and I. Witten, 1990, "Text compression", Prentice-Hall, Englewood Cliffs NJ.
[9] Elabdalla, A. and M. Irshid, 2001, "An efficient bitwise Huffman coding technique based on source mapping", Journal of Computer and Electrical Engineering, 27: 265–272.
[10] Ghwanmeh, S., R. Al-Shalabi, and G. Kanaan, 2006, " Efficient data compression scheme using dynamic Huffman code applied on Arabic language", Journal of Computer Science, pp: 885-888.
[11] Held, G. and T. Marshall, 1991, " Data compression", New York: John Wileyand Sons.
[12] Langdon, G. and J. Rissanen, 1981, " Compression of black-white images with arithmetic coding.", IEEE Trans. Communication, 29: 858-867.
[13] Langdon, G. and J. Rissanen, 1982, "A simple general binary source code", IEEE Trans. Information Theory,28: 800-803.
[14] Mano, M., 1984. Digital design. Prentice-Hall, New Jersey: Englewood Cliffs. Nelson, M., 1991. The data compression book. Prentice-Hall.

