http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (5): 776-779, 2007
ISSN 1812-5638

A Huffman Decoding Algorithm in Mobile Robot Platform

[1]R. Ponalagusamy, [2]E. Kannan and [3]Michael Arock
[1]Department of Mathematics, National Institute of Technology, Trichy 620 015, Tamilnadu, India
[2]Department of Computer Science and Engineering,
Vel Shri Rengarajan Shaugunthala High Tech Engineering College, Chennai 600 062, Tamilnadu, India
[3]Department of Computer Applications, National Institute of Technology, Trichy 620 015,
Tamilnadu, India

Abstract: In the field of mobile robotics, data is transmitted from the robot over low bandwidth channels or incrementally in short bursts to a host, where it can be further processed for visualization. JPEG-LS is the one of the image and file compressions, in which the Huffman decoding is used to compress the result of quantization stage. In the recent production and management scenario, advanced factories require first an increasing degree of automation to reduce the product cost and, second, availability of greater flexibility of output and product types. To meet such requirements in near future, the factories need to link the flow of material, energy and information together more efficiently. In order to link the flow of information with other entities, we have to involve data compression, coding and decoding techniques. The speed is very demanding for such applications which in turn motivate to develop a fast decoding algorithm to meet the challenges arising in the field of predicting collision free path for robot. Huffman decoding has been widely used in data, image and video compression. This motivates the present work. For a Concurrent Read and Exclusive Write (CREW), Parallel Random Access Machine (PRAM) model with N processors, we propose a parallel algorithm for Huffman decoding in this paper. The algorithm employs (N+1)-ary search, which is parallel version of binary search. Its time complexity amounts to O (log N+1 (n+1)), where n is the number of symbols in a Huffman tree.

Key words: Huffman code, huffman decoding, mobile robot, jpeg, parallel algorithm, CREW PRAM model

INTRODUCTION

Considerable efforts have been made in the development of new techniques and sensing devices to provide inputs from environment for robotic manipulators. To gather such information, tactile sensors, scene analyzers, proximity sensors and range finders have been introduced and tested in operation. Range finders are now able to provide sensing information required for functions such as object grasping, obstacle avoidance and moving parallel to production table.

Mobile robots can be used to acquire 3D models of the environment for a variety of military and commercial applications. Stereo vision or laser range can be used for acquiring the data. Laser sensors provide more accurate 3D information, but vision sensors are smaller and cheaper and additionally provide texture maps for more realistic 3D models. Processing vision data, however, is usually very resource consuming and often not be done in a real-time, while laser data directly yields a 3D model. At present, several loseless image and file compression schemes are available for data compression and JPEG-LS (Pennebaker and Mitchel, 1993; Marcellin *et al*., 2000) is the one of effective scheme in which Huffman decoding is used to compress the result of quantization stage. Hence Huffman decoding is still one of the popular compression techniques and is widely used in different fields of Science (Bell *et al*., 1990; Huffman, 1952; Roman, 1992) The field of data compression constantly attracts the interest of many researchers both in theoretical foundations of computing and in application oriented areas. In the last two decades, the fundamentals of data compressions have been laid (Lei and Sun, 1991) and efficiently applied to text and image compression. Currently, data compression is of increasing interest again because of the growing amount of data processed in applications and transferred over the internet. In particular compression of geometric data is currently an active research area (Mandal, 2000) and it is also important to perform operations on the compressed data directly working in the compressed domain, instead of decompressing prior to any processing. However, fast

Corresponding Author: R. Ponalagusamy, Department of Mathematics, National Institute of Technology, Trichy 620 015, Tamilnadu, India

decoding and scanning through compressed data are more important than code construction. Huffman decoding is subjected to numerous investigations in the past 52 years. Huffman coding creates minimal redundancy codes for a given set of symbols and their respective occurrence frequencies. Each code, in this technique, is a binary string, encoded form of the original message, which is decoded into actual message at the receiving end, again, using a decode tree. A decode tree is a binary tree in which external nodes represent messages. An efficient decoding algorithm that comprises of an ordering and clustering scheme had already been developed (Chen *et al*., 1999), which was later enhanced by Chung (1997) with the help of memory-efficient array data structure developed by himself. Further (Chen *et al*., 1999), a more efficient sequential algorithm to save memory space has been given. A recent space-efficient algorithm is found (Lin and Chung, 2000) and new data structure is proposed for Huffman Decoding (Klein, 2000). Also, we find a parallel algorithm for Robot path planning proposed (Michael and Ponalagusamy, 2006). This motivates us to design a parallel algorithm for Huffman decoding.

This research employs a parallel algorithm for Huffman decoding for CREW PRAM model, which is a Concurrent Read and Exclusive Write, Parallel Random Access Memory consisting of N processors.

DATA STRUCTURE

Consider a Huffman tree T with n symbols (that represent messages). The symbols corresponding to leaves are labeled as m_1, m_2... m_n from left to right. The root is said to be at level 0. The level of any other node is 1 more than the level of its father. The level of the Huffman tree is denoted by 'p'. The largest level is the depth 'd' of the Huffman tree. The weight of a symbol is defined to be 2^{d-p}. Let w_i be the weight of the symbol m_i for I = 1, 2...n. Let us define the variable $cum_1 = w_1$ and

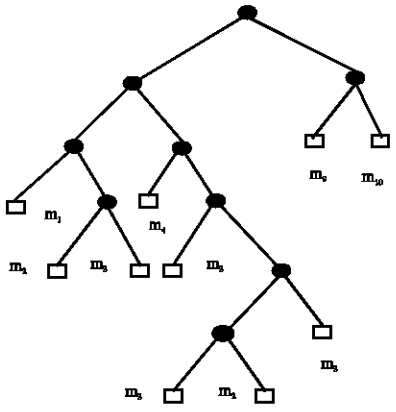

Fig. 1: Huffman tree with 10 leaves denoting 10 messages

Table 1: The values of m_i, w_i, cum_i

Index	m_i	w_i	cum_i
1	m_1	8	8
2	m_2	4	12
3	m_3	4	16
4	m_4	8	24
5	m_5	4	28
6	m_6	1	29
7	m_7	1	30
8	m_8	2	32
9	m^9	16	48
10	m_{10}	16	64

$cum_i = cum_{i-1} + w_i$ for i = 2, 3...n. The Huffman tree with 10 leaves is shown in Fig. 1 and the corresponding values of m_i, w_i and cum_i are shown in Table 1.

DESCRIPTION OF THE ALGORITHM

First, we compute value t using a formula $t = (c + 1) * 2^{d-b}$ from the codeword c given. Then t value is searched in 'cum' array and when a match is found (let the index be k), we check if w_k equals 2^{d-b}. If it is so, the corresponding message m_k is the message for the codeword c. Now, notice that 'cum' is the cumulative function of 'w' and it is strictly increasing function, since w_i can't be less than 1. This implies that the elements of array 'cum are distinct and in non-decreasing order. This is the requisite for using (N+1)-ary search (Parallel version of Binary Search), where N is the number of processors. Hence, in our parallel Huffman decoding algorithm, we employ the aforementioned search procedure in which, at each stage, the 'cum' array is splits into N+1 subsequences of equal length and the N processors simultaneously search the elements at the boundary between successive subsequences. Once it is searched, we check the corresponding weight and based on the flag, we get the message for the corresponding code word.

OUR PARALLEL ALGORITHM

Here we present a Huffman decoding parallel algorithm using CREW PRAM model with N processors.

Algorithm:

Input: The values of m_i, w_i, cum_i where i = 1, 2...n of a Huffman tree T which contains symbols with depth d and a binary codeword c.

Output: The corresponding symbol m_k of c.

Method:

Step 1: compute $t \leftarrow (c+1) * 2^{d-b}$ where b is the number of binary digits in c.

Step 2: {Search t in the array 'cum'}

2.1) $q \leftarrow 1$

2.2) $r \leftarrow n$

2.3) $k \leftarrow 0$

2.4) stages $\leftarrow \lceil \log(n+1)/\log(N+1) \rceil$

2.5) while ($q \leq r$ and $k = 0$) do

$j_0 \leftarrow q - 1$

For $i \leq 1$ to N do in parallel

$x \leftarrow j_1 \leftarrow (q-1) + i * (N+1)^{stages-1}$

{Processor P_i compares t to cum_j and decides on the part to be retained}

if $j_i = r$

then if $cum_x = t$

then $k \leftarrow j_i$

else

if $cum_x > t$

then $flag_i \leftarrow$ left

else

$flag_i \leftarrow$ right

endif

endif

else

$j_i \leftarrow r + 1$

$flag_i \leftarrow$ left

endif

{This indices of the subsequence to be searched in the next iteration are computed}

if $flag_i <> flag_{i+1}$

then $q \leftarrow j_{i-1} + 1$

$r \leftarrow j_i - 1$

endif

if (i = N and $flag_i <> flag_{i+1}$)

then $q \leftarrow j_i + 1$

endif

endfor

stages $\leftarrow$ stages -1

endwhile

Step 3: if $k <> 0$ and $w_k = 2^{d-b}$ then

m_k is the corresponding symbol of c

else

C is not a code word of T

endif

Step 4: stop

TIME COMPLEXITY ANALYSIS

Theorem 5.1: The proposed parallel algorithm correctly finds the symbol for a given code on CREW PRAM model with $O(\log_{N+1}(n+1))$-time using N processors.

Proof: In step-1, the value of t can be calculated in constant time. In step-3, we check the value of k, weight (w_i) and based on the flag, we find the message corresponding to the symbol. So, this step also takes constant time. In step-2, we employed (N+1)-ary search, which is parallel version of binary search and in the worst case, the number of stages needed is $O(\log_{N+1}(n+1))$. Hence the time complexity of our Huffman decoding algorithm amounts to $O(\log_{N+1}(n+1))$, where $1 < N \leq n$. Hence the proof.

ILLUSTRATION

We illustrate our algorithm with the following examples.

Example 1:

Let c = 01011 and N be 2.

In step-1, $x = (11+1) * 2^{6-5}$

$= 24$

In step-2, the sequence cum_i is split into (8,12,16), (24,28,29), (30,32,48,64). and in the next iteration, the subsequence (24,28,29) is split into (24), (28), (29). Step-2 terminates on giving $cum_4 = 24$ (k being 4).

In step-3, w_4 (= 8) <> 2^{6-5}. Thus 01011 is not a codeword of T.

Example 2:

Let c = 0110 and N be 2.

In step-1, $x = (6+1) * 2^{6-4} = 28$.

In step-2, the sequence cum_i is split into (8,12,16),(24,28,29), (30,32,48,64). In the next iteration of step-2, the subsequence (24,28,29) is split into (24), (28), (29). Step-2 terminates on giving $cum_5 = 28$ (k being 5).

In step-3, $w_5 (=4) = 2^{6-4}$. Thus, 0110 is a codeword of T corresponding to message m_5.

DISCUSSION

In order to cope with the challenge of the future factories that provide higher productivity at lower cost and compatible environments, automation of future industries and plant is unavoidable. Such advanced factories require first an increasing degree of automation to reduce the product cost and, second, availability of greater flexibility of output and product types. To meet such requirements, the factories of the future need to link the flow of material, energy and information together in more efficient way. Hence it is pertinent to point out here that the development of an exact or more efficient parallel algorithm for Huffman decoding is deemed of importance

from the view of wide practical applications on mobile robot platform for compressing the result of quantization stage in JPEG. Mobile robots are the key elements for the integration of transport and handling function. It is of interest to state that to determine collision free path for robot's movement without hitting the obstacles, we are in need of fast (parallel) algorithm to compress the data obtained from the picture of obstacles or objects lying in the robot platform.

It is observed that the sequential time complexity of this algorithm is O (log n) time. The application of sequential algorithm for data compression in JPEG-LS, cause the sluggish movement of robot in its collision free path, which in turn affects the rate of production. Keeping this in view, our attention has been focused to present a parallel algorithm for Huffman Decoding using CREW PRAM model with time complexity $O(\log_{N+1}(n+1))$, where $1< N \leq n$. When $N = n$, the algorithm runs in constant time, since the elements of 'cum' array are distinct (which is the required condition for achieving this constant time).

REFERENCES

Bell, T.C., J.G. Cleary and I.H. Witten, 1990. Text Compression, Prentice Hall, Englewood Cliffs, Reading, New Jersey.

Chen, H., Y. Wag and Y. Lan, 1999. A memory-efficient and fast Huffman decoding algorithm. Inform. Process. Lett., 69: 119-122.

Chung, K.L., 1997. Efficient Huffman decoding. Inform. Process. Lett., 61: 97-99.

Huffman, A., 1952. A method for the construction of minimum redundancy codes. Inform. Process. IRE, 40: 1098-1101.

Klein, S.T., 2000. Skeleton trees for the efficient decoding of Huffman encoded texts. Kluwer J. Inform. Retrieval, 3: 7-23.

Lei, S.M. and M.T. Sun, 1991. An entropy coding system for digital HDTV applications. IEEE Trans. Circuit Systems Video Tech., 1: 147-155.

Lin, Y. and K.L. Chung, 2000. A Space Efficient Huffman decoding algorithm and its Parallelism. Theor. Comput. Sci., 246: 227-238.

Mandal, J.N., 2000. An approach towards development of efficient data compression algorithms and correction techniques. Ph.D Thesis, Jadavpur University, India.

Marcellin, M.W., M.J. Gormish, A. Bilgin and M.P. Boliek, 2000. An overview of JPEG-2000. In: Proc. Data Compression Conf. DCC-2000, Snowbird, Utah, 523-541, IEEE Computer Society, New Jersey.

Michael, A. and R. Ponalagusamy, 2006. Parallel algorithms for robot path planning with simpler VLSI architecture. Int. J. Comput. Applns. Tech., 26: 157-163.

Pennebaker, W.B. and J.L. Mitchel, 1993. JPEG: Still Image Date Compression Standard, Van Nostrant Reinhold, New York.

Roman, S., 1992. Coding and Information Theory, Springer, Reading, New York.