N-Square Approach For Lossless Image Compression And Decompression

G. Srinivas* Prasad Reddy P.V.G.D* K. Gayatri Devi*

GJCST Computing Classification
I.4.2

***Abstract*- There are several lossy and lossless coding techniques developed all through the last two decades. Although very high compression can be achieved with lossy compression techniques, they are deficient in obtaining the original image. While lossless compression technique recovers the image exactly. In applications related to medical imaging lossless techniques are required, as the loss of information is deplorable. The objective of image compression is to symbolize an image with a handful number of bits as possible while preserving the quality required for the given application. In this paper we are introducing a new lossless encoding and decoding technique which even better reduces the entropy there by reducing the average number of bits with the utility of Non Binary Huffman coding through the use of N-Square approach and fasten the process of searching for a codeword in a N-Square tree, we exploit the property of the encoded image pixels, and propose a memory efficient data structure to represent a decoding N-Square tree. Our extensive experimental results demonstrate that the proposed scheme is very competitive and this addresses the limitations of D value in the existing system by proposing a pattern called N-Square approach for it. The newly proposed algorithm provides a good means for lossless image compression and decompression.**

Keywords- Minimum Redundancy Code, Image compression, Non Binary Huffman, Redundancy, N^2, Interval generation, N2 Tree.

I. Introduction

In many military and security applications, there is a need to compress image data to reduce the bandwidth requirements of wireless and networked-based systems. By reducing the amount of data associated with these images, the amount of storage space required can also be minimized, allowing long image sequences to be stored on a single disk. With the advanced development in Internet, teleconferencing, multimedia and high definition television technologies, the amount of information that is handled by computers has grown exponentially over the past decades. Hence, storage and transmission of the digital image component of multimedia systems is a major problem. The amount of data required to present images at an acceptable level of quality is extremely large. High quality image data requires large amounts of storage space and transmission band width [1]. One of the possible solutions is to compress the information so that the storage space and transmission time can be reduced. Image compression address the problem of reducing the amount of data required to represent a digital image with no significant loss of information. The goal of image compression is to represent an image with a few number of bits as possible while preserving the quality required for the given application [2].

G. SRINIVAS PRASAD REDDY P.V.G.D** K. GAYATRI DEVI**
**Dept. of IT, GITAM University, AP, srinivas.gitam@gmail.com, gayatridevi07@gmail.com **Dept. of CS & SE, Andhra University, AP.*

In this paper, as per the conventions of Huffman, the sequence of symbols associated with a given message will be called the ―message code". The entire number of messages to be transmitted will be called the ―message ensemble". The amiability between the sender and receiver about the meaning of the code for each message of the ensemble will be called the ―ensemble code". In the way to formulate the requirements of an ensemble code, the coding symbols will be represented by numbers. Therefore, if there are D different types of symbols to be used in coding, they will be represented by the digits 0, 1, 2... D-1 [4].
The number of messages in the ensemble is denoted by N. Let P(i) be the probability of the its message. Then

$$\sum P(i) = 1 \tag{3}$$

The length of message, L(i) is the number of coding digits assigned to it. Thus the average length is

$$L.avg=\sum P(i)\, L(i) \tag{4}$$

This paper presents a new array data structure to represent the N-Square tree. The memory required in the proposed data structure is $nd = O(n)$, which is less than the previous ones. We then address an efficient N-Square decoding algorithm based on the proposed data structure; given a N-Square code, the search time for finding the source symbol is $O(\log n)$[5].In section 2 of the paper, we present the overview of proposed data structure. In section 3, we propose a N-Square approach for compression. In section 4, we present the decoding technique and elucidate how the proposed approach is better than the existing one. In section 5, we conclude our paper and summarize our recommendation.

II. The Proposed Data Structure

The occurrences of probability of pixels $p = \{p_0, p_1, \ldots, p_{n-1}\}$ with frequencies $f = \{f_0, f_1, \ldots, f_{n-1}\}$ for $f_0 \geq f_1 \geq \ldots \geq f_{n-1}$, where fi is the frequency of probability pi [5]. Using the N-Square algorithm to erect the N-Square tree T, to obtain the codeword ci, $0 \leq i \leq n-1$, for probability pi determined by traversing the path from root to the leaf node allied with probability pi, where the left branch is corresponding to ‗0' and the right one is corresponding to

‗1‘,‗2‘, etc. Let the level of the root be zero and the level of the other node is equal to summing up its parent‘s level and one. Codeword length l_i for p_i can be known as the level of p_i. Then the weighted external path length $\sum_{i=0}^{N-1} f_i l_i$ is minimum. For example, the N-Square tree corresponding to the pixels {p_0, p_1, . . . , p_{22}} with the probabilities {0.3, 0.2, 0.1, 0.1,0.065, 0.05,0.04, 0.02, 0.02, 0.02, 0.01, 0.01, 0.01,0.01, 0.009, 0.008, 0.007, 0.006, 0.005, 0.004, 0.003, 0.002, 0.001} is shown in Fig. 1. The codeword set C {c_0, c_2, . . . , c_{22}} is derived as {0,2,3,4,11,12,14,101,102,103,104,130,131,132,133,134,1000,1001,1002,1003,1004,10000,10011}, where the length set L{l_0, l_1,…..l_{22}} is {1, 1, 1,1, 2, 2, 2,3, 3, 3, 3, 3, 3, 3, 3, 3, 4,4,4,4,4,5,5}. In addition, the codeword set composes of a space with 2^d addresses, where d = 5 is the depth of the N-Square tree [5].

III. N-SQUARE ENCODING

The erraticism of Non binary Huffman coding stems from the fact that there is no definite pattern for the value of D. For the same number of samples the method shows different results for different D values. There are no limitations expressed on the value of D to be taken. Therefore one can take the maximum value of D for which the message can be sent in a one step auxiliary ensemble which is quite disgusting as the burden of adding large number of messages is an overhead to the encoder. So in our approach we follow a pattern or structure for fixing the value of D and eradicating the erraticism of the existing approach. The number of initial messages to be added has also been used differently which ultimately reduces the average no of bits per message.

A. N-Square Approach For D Value

Here the number of samples N plays a major role i.e. the value of D will be fixed depending on the number of samples.

Step 1-Find the range of N-Square in which our N lies where N takes values from 1 through n.

Step2-Assign D to the upper boundary of N of the range.

Let us examine an example, for suppose our N=20. Then N lies between 4^2 and 5^2. So the range is [4 5]. The upper boundary of the range is 5. Therefore the value of D=5.

Table 2: shows N values.

N^2	N
1^2	1
2^2	4
3^2	9
4^2	16
5^2	25
6^2	36
.	.
N^2	N

Thus in this way we are following a pattern instead of taking arbitrary values for D. As we got all the required elements i.e. N, D lets start our ensemble coding. And as usual the first step will be finding the initial no of messages to be added in the original ensemble (say k) and is given by the following formula.

$$J=2 + [(N-2) \bmod (D-1)] \quad K=J+D-1 \qquad (7)$$

Here K indicates the least number of messages above D to be added in the initial step in order to make the penultimate ensemble to contain D messages. And from the later steps we keep on adding D least probable messages to form a composite message. It will be noted that the terminating auxiliary ensemble always has one unity probability message. Each preceding ensemble is increased in number by D-1 until the fist auxiliary ensemble is reached. Therefore if N1 is the number of messages in the first auxiliary ensemble, then (N1-1) / (D-1) must be an integer. However N1=N-K+1.Therefore K must be of such a value that (N-K) / (D-1) is an integer. We are calculating K and all these just to ensure that in the penultimate ensemble the number of messages should be equal to D. If in our problem when J=D then no need to calculate K. We continue by adding D bits in each pass. The decoding is done by assigning bits 0 to D-1 in order, to each of the brackets. But since the initial number of messages to be added i.e. K is more than D we repeat the cycle 0 to D-1 again for the remaining extra K-D messages. The whole method would get cleared with the following illustration of an example.

Table 1. An Example of N-Square Encoding

Gray level value	Probability P_i	Code word C_i
G4	0.3	0
G7	0.2	2
G1	0.1	3
G17	0.1	4
G6	0.065	11
G2	0.05	12
G8	0.04	14
G14	0.02	101
G20	0.02	102
G11	0.02	103
G22	0.01	104
G13	0.01	130
G9	0.01	131
G15	0.01	132
G16	0.009	133
G5	0.008	134
G18	0.007	1000
G25	0.006	1001
G10	0.005	1002
G21	0.004	1003
G12	0.003	1004
G3	0.002	10000
G19	0.001	10011

The codeword‘s generated from the N-Square tree could be treated as suffixes, which obey the suffix property, i.e., no codeword is the suffix, or start of the code for another codeword. For example, c_0 could be treated as an interval from address 00000 to 11110, and c_{20} starts from 00102 to 11102. Since there is no empty branch in the N-Square tree, each address is occupied by exactly one interval of the suffix [5].

IV. DECODING ALGORITHM

We can achieve the decoding procedure by N-square tree. The detailed algorithm is listed below. The time complexity for decoding is O(log n)[5].

Step 1-Interval Generation Algorithm:

Table 3 : Neighboring Interval Representation of symbols in Table 1 [5]

Interval	Gray level value	Starting Address
int_o	G4	00000
Int_1	G7	00002
Int_2	G1	00003
Int_3	G17	00004
Int_4	G6	00011
Int_5	G2	00012
Int_6	G8	00014
Int_7	G14	00101
Int_8	G20	00102
Int_9	G11	00103
Int_{10}	G22	00104
Int_{11}	G13	00130
Int_{12}	G9	00131
Int_{13}	G15	00132
Int_{14}	G16	00133
Int_{15}	G5	00134
Int_{16}	G18	01000
Int_{17}	G25	01001
Int_{18}	G10	01002
Int_{19}	G21	01003
Int_{20}	G12	01004
Int_{21}	G3	010000
Int_{22}	G19	010011

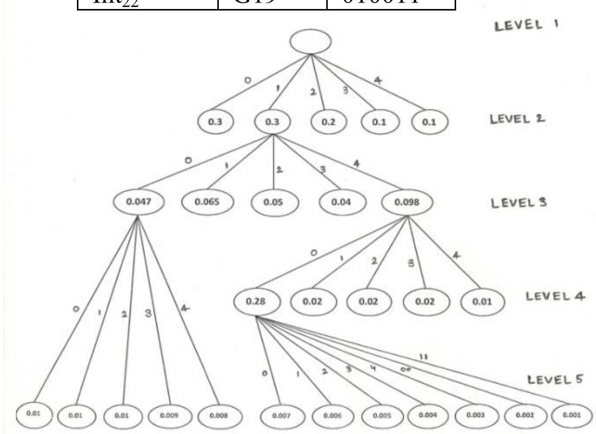

Figure 1: An Example of the N-Square tree

Input-Construct d level N-Square tree

Output-sort N intervals.

Step 2- Generate a tree equivalent to the encoding tree.

Step 3-Scan the input characters or string from left to right and go left on a 0 and go to right sub-tree on reading 1, 2, and 3 until a leaf is reached.

Step 4-Output the probability of pixels encoded in the leaf node and return to root.

Step 5-Continue with step2 until the input is empty.

V. CONCLUSION

In this paper we introduced a method for lossless compression of images as an efficient one. We saw that the proposed approach eliminates the erraticism of non binary Huffman coding regarding the value of D. It introduced a pattern for finding the D value as per the value of N through N-Square approach. We also saw that the proposed method outperformed the existing one in terms of entropy as well. The implication of the N-Square is due to its acknowledged use in image and data compression. The main rationale of this paper is to simplify the representation of the N-Square tree and the decoding procedure by the property obscure in the N-Square tree.

VI. REFERENCES

1) ―Digital Image Processing”, S.Jayaram, S.Esakkirajan, T.VeeraKumar, Tata Mc grawHill Publications.
2) ―Digital Image processing”, Second Edition, Rafel c. Gonzalez, Richard E.Woods, 6Th Impression 2009, ISBN 978-81-7758-168-3.
3) ―Fundamentals of Digital Image processing”, S. Annadurai, R.ShanmughaLakshmi, 3rd Impression ,ISBN 978-7758-479-0.
4) DAVID A. HUFFMAN, ASSOCIATE, IRE,” A Method for the Construction of Minimum-Redundancy Codes”, 1952.
5) Pi-Chung Wang, Yuan-Rung Yang, Chun-Liang Lee, Hung-Yi Chang, ―A Memory-efficient Huffman Decoding Algorithm”, (AINA‘05), 1550-445X/05, 2005 IEEE
6) Cebrail TAŞKIN, Serdar Kürşat SARIKOZ,” An Overview of Image Compression Approaches”, ICDT.2008.
7) Lu G., ―Advances in digital image compression techniques”, 1993.
8) Bei Chen, Hong-Cai Zhang, Wen-Lun CAO, Jian-Hu Feng, ―Huffman Coding Method based on Number Character”, ICMLC 19-22 Aug-2007 IEEE.
9) Reza Hashemian ,” Direct Huffman Coding and Decoding using the table of Code-Lengths”,ITCC‘03 IEEE.
10) Jia-Yu Lin, Ying Liu and Ke-Chu Yi,” Balance of 0,1 Bits for Huffman and Reversible Variable-engt Coding”, IEEE Transactions on Communications, VOL.52, NO.3, March 2004