2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing

Embedding Secret Messages Using Modified Huffman Coding

Kuo-Nan Chen
Department of Computer Science and Information Engineering,
National Chung Cheng University
Chiayi, Taiwan
kuonan.chen@gmail.com

Chin-Feng Lee
Department of Information Management,
Chaoyang University of Technology
Taichung, Taiwan
lcf@cyut.edu.tw

Chin-Chen Chang, Huang-Ching Lin
Department of Information Engineering and Computer Science,
Feng Chia University
Taichung, Taiwan
alan3c@gmail.com,
hata9871@yahoo.com.tw

***Abstract*—**This paper demonstrates an effective lossless data hiding scheme using modified Huffman coding. The binary secret message is concurrently embedded and encoded with a cover medium such as a video file, an audio file, or even a text file. The proposed scheme not only provides good data hiding capacity and data recovery capability, but also being efficient in space saving (the stego medium is much smaller than the cover medium). Each symbol in a cover medium can carry one secret bit, and the cover medium can be reversed. And the experimental results show that the stego code can saves 30% to 35% of space compared with the cover medium**.**

Keywords-lossless data compression; Huffman code; data hiding

I. INTRODUCTION

Server data formats are used to be the cover medium in data hiding, e.g. audio files, video files, image files, text files, and so on. Although the data structure of text files is similar to image files than the other data format mentioned above, most of image data hiding schemes are not suitable for text files. The main reason is that most image data hiding schemes embed secret information into cover image by slightly perturbing the pixel values. Since gray-scale or color images can tolerant a small amount modifications of pixel values, it will cause no perceptible distortions. On the contrary, any changes in the text file might lead to meaningless content.

Few studies have referred to hiding secret messages in text files. In [1], the data was embedded by modifying the inter-character space, but it resulted in some distortions in the shape of words. In [3], a technique was proposed for copyright protection that marks the text file by shifting lines up or down and words right or left; however, the technique might change the typesetting of the text file accordingly.

In addition to the security problem, bandwidth consumption is also an important concern. The size of transmitted files can be reduced by either of two categories of data compression technology: lossless and lossy technologies. The lossy data compression technology is widely used in images, but it may be unsuitable for text files because any loss of data may lead the content meaningless. In that point of view, we use an efficient lossless data compression scheme, Huffman coding, to compress the transmitted files. In general, it is a compromise between hiding capacity and detectability in data hiding. In the experimental results, we will show that our proposed scheme is not only efficient in both data hiding capacity and visual concealment of stego files, but also efficient in space saving. In data hiding capacity, each symbol in the cover medium can carry one secret bit. In visual concealment, the stego file has no any distortion compared with the cover medium. Finally, compared with the totally size of cover medium and binary secret message, the space savings with stego code is 30% to 35%. In addition to text files, the proposed scheme is also suitable for most cover media, including images and video files.

The remainder of this paper is organized as follows. Section 2 first defines the notations that will be used in our proposed scheme; then the pre-process of the encoding procedure for constructing a modified Huffman tree is presented followed by the embedding and extracting procedures. Section 3 reviews the experimental results, which show that our proposed scheme is not only efficient in data hiding capacity and data concealment, but also efficient in space saving with stego code. Finally, some conclusions are stated in Section 4.

II. THE PROPOSED SCHEME

In the proposed scheme, secret messages can be embedded with cover media based on Huffman coding. Here, the secret message might be an image, a document file, a video file, or other kind of documentation, which is transformed into its binary presentation first. The cover media here might be images, document files, video files, and so on. The word "symbol" used in the following description might refer to a pixel of an image, a character, a white space or a tab of a document file, and so on.

There are two procedures in the proposed scheme: an embedding procedure and an extracting procedure. In the embedding procedure, each symbol is encoded with one secret bit based on the modified Huffman tree, so each symbol can carry one secret bit in our proposed scheme. In the extracting procedure, the stego code can be decoded into

978-0-7695-3762-7/09 $26.00 © 2009 IEEE
DOI 10.1109/IIH-MSP.2009.53

its uncompressed format and the secret message can be completely extracted according to the modified Huffman tree that was built in a bottom-up manner during the embedding procedure.

A. Notations

A: an alphabet set, $A = \{a_i \mid i = 1, 2, \ldots, n\}$, where a_1 denotes the first element in A, a_2 denotes the second element in A, and so on.

B: a binary secret message, $B = \{b_i \mid i = 1, 2, \ldots, m\}$, where b1 denotes the first bit of B , b_2 denotes the second bit of B, and so on.

C: a stego code which is encoded by the modified Huffman tree H with secret message S embedded within.

CW: the codeword of a symbol, e.g., $CW(a)$ is the codeword of the symbol "a."

ES: a symbol used to indicate the end of secret bits; its frequency is set to be 0.

F: a frequency set, $F = \{f_i \mid i = 1, 2, \ldots, n\}$, where f_1 is the first element standing for the frequency of a_1, f_2 is the second element which represents the frequency of a_2, and so on.

MH: a modified Huffman tree.

Mx: a symbol whose frequency is set to be ∞.

S: a rearranged binary secret message, $S = \{s_i \mid i = 1, 2, \ldots, m\}$, where s_1 denotes the first bit of S , s_2 denotes the second bit of S, and so on.

SK: a secret key kept by the owner.

T: a cover medium, $T = \{t_i \mid i = 1, 2, \ldots, p\}$ represents all symbols in the cover medium, where t_1 denotes the first symbol, t_2 denotes the second symbol, and so on.

Following is an example for the relationship between A and F.

B. Pre-process

The pre-process of the embedding procedure tends to build the modified Huffman tree according to [2] as follows.

Procedure Modified_Huffman_tree( )

Input: A cover medium T; two preserved symbols, i.e., Mx and ES (End of Secret bits), whose frequencies are set to be ∞ and 0, respectively.

Output: A modified Huffman tree MH.

1) Statistically analyze the cover medium T to get the corresponding alphabet set A and the frequency set F. Add the two preserved symbols, Mx and ES, and their corresponding frequencies to A and F, respectively.

2) Create a leaf node for each element in A.

3) Sort all elements in A in increasing order according to their frequencies in F and extract the two least frequent elements from A. Merge them into a new node whose frequency equals their sum. Label the edge to the left child as 0, and 1 for the edge to the right child. Add this new node to A and its corresponding frequency to F.

4) If there is more than one element in A, repeat 3) until there are no more elements.

5) The last merged node is assigned as the root of MH.

6) Duplicate the left sub-tree MHL of the root in MH to the right side. In the right sub-tree MHR, replace the parent node of ES with the sibling node of ES. Next, remove ES and its sibling at MHR.

7) Return MH as the modified Huffman tree.

C. Embedding procedure

The embedding procedure is shown in Figure 1.

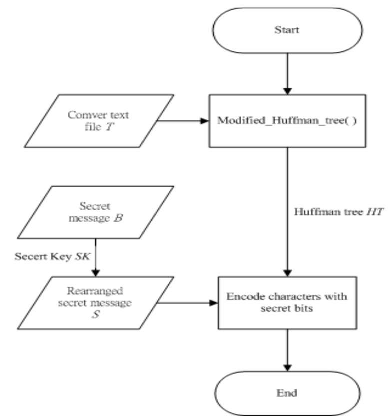

Figure 1. The embedding procedure.

The embedding procedure is as follows.

Procedure Embedding ( )

Input: A cover medium T; a binary secret message B.

Output: The stego code C and the modified Huffman tree MH.

1) Rearrange B by a pseudorandom number generator with secret key SK, the rearranged result is named S.

2) Call the pre-procedure Modified_Huffman_tree (T, Mx, ES) to obtain MH.

3) For each secret bit s_i, for $i = 1, 2, \ldots, m$, encode the symbol t_i, for $i = 1, 2, \ldots, m$, with si according to MH.

a) If $s_i = 0$, add the codeword of t_i in MHL to C.

b) If $s_i = 1$, add the codeword of t_i in MHR to C.

4) Add the corresponding codeword of ES to C.

5) For the remaining un-encoded symbols t_i, for $i = m+1, m+2, \ldots, p$, in T, add their corresponding codewords to C.

6) Output the stego code C and the modified Huffman tree MH.

D. Extracting procedure

The extracting procedure is shown in Figure 2.

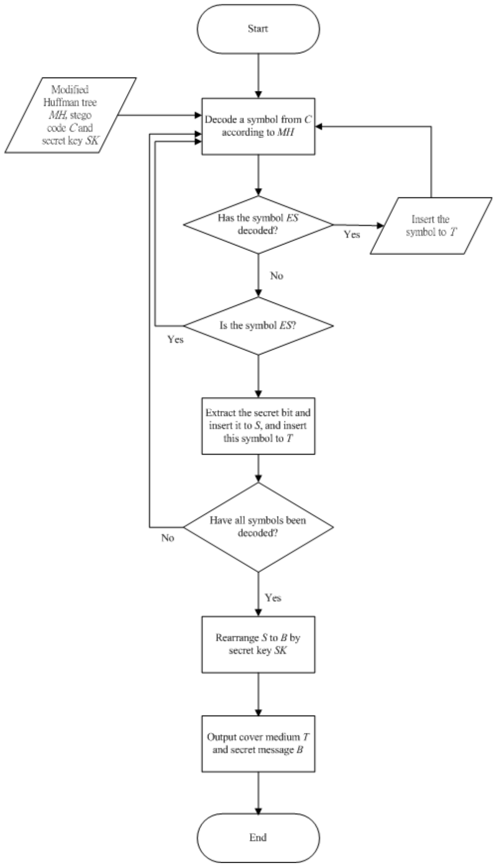

Figure 2. The extracting procedure.

When C and MH are received, a symbol can be decoded by matching each bit in C with the edge labeling in MH from the root whenever a leaf node is reached. If a symbol is decoded and the decoded symbol is not ES, some secret bits need to be extracted. Before the ES symbol is encountered, every secret bit can be extracted according to the first bit of each symbol's codeword. Then the original secret message B can be recovered by rearranged S with secret SK. The extracting procedure can be stepped as follows.

Procedure Extracting( )

Input: The stego code C, the modified Huffman tree MH and the secret key SK.

Output: The restored cover medium T and the secret message B.

1) Set $i = 1$ and both T and S are initially empty.

2) Extract a codeword $CW(t_i)$ associated with symbol t_i according to MH when a leaf is reached. Decode $CW(t_i)$ to get t_i.

a) Before ES is decoded:

- extract the first bit of $CW(t_i)$ to be the secret bit s_i. concatenate s_i to S such that $S = S \cdot s_i$.
- concatenate t_i to T such that $T = T \cdot t_i$.
- set $i = i + 1$.

b) If $t_i = ES$, ignore the symbol ES and set $i = i + 1$. Repeat *2)*.

c) After ES is decoded, concatenate t_i to T such that $T = T \cdot t_i$. Set $i = i + 1$.

3) Repeat *2)* until all elements in C are completely decoded.

4) Rearrange S with secret key SK to obtain B.

5) Output cover medium T and secret message B.

III. EXPERIMENTAL RESULTS

In our designed experiment, binary images are used as secret messages, and articles written in English are treated as cover media. Three steps are used to implement and guarantee data consistency in the proposed scheme. In the first step, binary secret messages are embedded and encoded into stego codes with cover media. In the second step, binary images are decoded from stego codes. In the final step, the extracted secret messages are compared bit-by-bit with the original ones to ensure data accuracy.

Five articles [4-5] written in English are used as cover media, and a segment of one article is shown in Figure 3.

The experimental results are listed in Table 1: The first column has the file size of each cover medium; the second column has the size of the Huffman code of each cover medium; the third column has the total number of symbols in each cover medium; the fourth column has the capacity of each cover medium; and the fifth column has the size of each stego code.

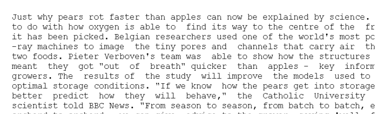
Just why pears rot faster than apples can now be explained by science.
to do with how oxygen is able to find its way to the centre of the fr
it has been picked. Belgian researchers used one of the world's most pc
-ray machines to image the tiny pores and channels that carry air th
two foods. Pieter Verboven's team was able to show how the structures
meant they got "out of breath" quicker than apples - key inform
growers. The results of the study will improve the models used to
optimal storage conditions. "If we know how the pears get into storage
better predict how they will behave," the Catholic University
scientist told BBC News. "From season to season, from batch to batch, e
orchard to orchard - we can give advice to the grower, saying 'well, f

Figure 3. A segment of one article.

TABLE I. THE SIMULATION RESULTS

Size / media	1st File size (Bits)	2nd Huffman code (Bits)	3rd Symbol number	4th Secret bits (Bits)	5th Stego code (Bits)
(1)	131832	63456	16479	16479	96664
(2)	32896	18040	4112	4112	26584
(3)	121120	66368	15140	15140	96912
(4)	65112	32408	8139	8139	48920
(5)	133128	72224	16641	16641	10584

In the proposed scheme, each symbol in the cover media can carry one secret bit, so the embedding capacity is decided by the number of symbols in the cover media, which is apparent from the equal quantity in the third and fourth columns in Table 1. Compared with the size of the cover medium added to the binary secret message, the space savings with the stego code is 30% to 35%. Moreover, anyone can recover the original cover media by decoding the stego code with our proposed modified Huffman coding, but only the legal receiver can get both the original cover media and secret message by decoding the stego code with the proposed scheme.

IV. CONCLUSIONS

This paper proposed an efficient data hiding scheme based on a modified Huffman coding algorithm. In the proposed scheme, each symbol of the cover medium can carry one secret bit. While anyone can get the original cover medium by decoding the stego code with Huffman coding, only the legal receiver can extract the binary secret message precisely. According to the experimental results, the space savings of the stego code is 30% to 35% of the size of the cover medium added to the binary secret message. Thus, the proposed scheme not only takes advantage of Huffman coding for space saving but also successfully embeds the binary secret message in text files.

REFERENCES

[1] N. Chotikakamthorn, "Electronic document data hiding technique using inter-character space," Proceeding of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems, pp. 419-422, Chiangmai, Thailand , Nov. 1998.

[2] D. A. Huffman, "A method for the construction of minimum redundancy codes," Proceedings of the Institute of Radio Engineers, vol. 40, pp. 1098-1101, Sept. 1952.

[3] S. H. Low, N. F. Maxemchuk and A. M. Lapone, "Document identification for copyright protection using centroid detection," IEEE Transactions on Communications, vol. 46, no. 3, pp. 372-383, Mar. 1998.

[4] Http://news.bbc.co.uk/, Jul. 2008.

[5] Http://www.cnn.com/, Jul. 2008.