International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:1, 2007

Enhance Performance of Secure Image Using Wavelet Compression

Goh Han Keat, Azman Samsudin and Zurinahni Zainol

***Abstract*—** The increase popularity of multimedia application especially in image processing places a great demand on efficient data storage and transmission techniques. Network communication such as wireless network can easily be intercepted and cause of confidential information leaked. Unfortunately, conventional compression and encryption methods are too slow; it is impossible to carry out real time secure image processing. In this research, Embedded Zerotree Wavelet (EZW) encoder which specially designs for wavelet compression is examined. With this algorithm, three methods are proposed to reduce the processing time, space and security protection that will be secured enough to protect the data.

Keywords—Embedded Zerotree Wavelet (EZW), Image compression, Wavelet encoder, Entropy encoder, Encryption.

I. Introduction

With the rapid expansion of computer technology, digital image processing place a great demand on efficient data storage and image content privacy. However, two fundamental problems need to be addressed before the digital image can safely send through the public network efficiently. There are to reduce the size of digital image and to protect the privacy of the image content.

With current technology, digital image compression and encryption will be the solution to solve the problems stated above. Compression technology such as JPEG, GIF, JPEG200, etc., is used to reduce the size of an image file by removing data redundancy acceptable degradation in image quality. Various encryption algorithms are used to encrypt the image content so the details are hidden when send through public network. Unfortunately, if compression and encryption are combining together, it leads to another problem. The overall processing performance will be slowed due to many computational involvements in these two algorithms.

This research is divided into five sections. The background explains basic concept on digital image processing. The explanation including what is wavelet transform, Embedded Zerotree Wavelet (EZW) [1] compression that is selected as a compression algorithm for this research, Arithmetic Entropy Encode algorithm that is used to further reduce the size of image file and finally streaming encryption algorithm RC4 that is used to encrypt the digital image content. The methodology section will explained the method proposed to enhance the overall performance of image processing. The discussion will include how to reduce image file size and to increase security protection. The third section discusses how the proposed methodology is being implemented and the forth section shows the results of the implementation. In the last section, the discussion on the future enhancement and conclusion is presented.

Manuscript received November 5, 2004. This work was supported by Universiti Sains Malaysia (USM).

II. Background

Uncompress image data requires considerable storage capacity and transmission bandwidth. Current solution is to compress the data to acceptable quality before it is transmitted and decompressed at the receiver side.

A. Principle of Image Compression

Most of the images have the common characteristic that is redundant information of neighbor pixels. Removing the redundancy information in a image is an important step during image compression. There are two fundamental components of compression. First, the redundancy reduction that aims to remove the duplication from signal source [2]. Second is irrelevancy reduction to omit parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System (HVS) [2].

Image compression consists of three closely connected components:

i. *Source Encoder* – source encoder transform the image from the space or time domain to a new domain where the inter-pixel redundancies are reduced.
ii. *Quantizer* – quantizers reduces the number of bits needed to store the transformed coefficient by reducing the redundancies of those values and represent it with appropriate method. Embedded Zerotree Wavelet (EZW) algorithm [3] is one of the famous quantizer that has the features of simplicity, efficiency, and low memory requirements.
iii. *Entropy Coder* – entropy coder further compresses the quantized values to give a better overall compression. Arithmetic Coder has been selected in this research because of the efficient model it has and the output is nearly optimum according to the Shannon Theorem.

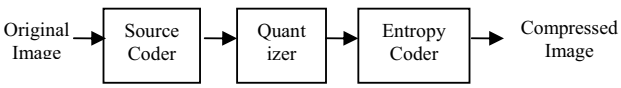

Figure 1: Principle of Image Compression System

B. Introduction to Wavelet Transform

The basic wavelet transform is similar to Fourier Transform. The coefficients are calculated by an inner product of the input signal with a set of orthonormal basic function. The different comes in the way these functions are constructed and more important in the types of analysis.

Multi-resolution transform is the key different of wavelet transform; it allows a form of time and frequency analysis. In wavelet transform, the result generated will shown when certain features occurred and about the scale characteristics of the signal. Scale is similar to frequency and is a measure of the amount of details in the signal. Small scale generally mean common details and large scale mean fine details.

1) Discrete Wavelet Transform

Discrete Wavelet Transform can be described as a recurrent process of filtering and sub-sampling perform on the input set data. In each level, a set of coefficients are calculated by applying a high-pass filter to the signal and down sampling the result by a factor of two. At the same level, low-pass filter is also performed followed by down sampling to produce the signal to next level. In the following level, the process of filtering and sub-sampling is performed on the signal output by the previous level, again produce the signal to the next level. The process is continued until it meets the requirement.

C. Quantizer: Embedded Zerotree Wavelet Encoder

Embedded Zerotree Wavelet encoder or EZW in short, is type of progressive encoding or known as embedded encoding that compress an image into bit stream with increasing accuracy. This algorithm was designed by J. M. Shapiro in 1993 specifically to be used with discrete wavelet transforms. Its development is based on two observations:

i. Natural images in general have a low pass spectrum. When an image is wavelet and the transformed energy in the subbands decrease, the scale decrease (low scale mean high resolution), so the wavelet coefficients will, on average, be smaller in the higher subbands than in the lower subbands. This show that progressive encoding is a very natural choice for compressing wavelet transformed images, since the higher subbands only added details [4].
ii. Large wavelet coefficients are more important than smaller wavelet coefficients [4]. Therefore, EZW algorithm encode larger wavelet coefficient first.

With the above two observation, EZW schema encode the larger coefficients to smaller coefficients with several pass in decreasing order. For very pass, a threshold is chosen against which all the coefficients are measured. If a wavelet coefficient is larger than the threshold it is encoded and removed from the image, if it is smaller it is save for the next pass. When the entire wavelet coefficient has been visited, the threshold is lowered and the image is scanned again to add more detail to the already encoded image. This process is repeated until all the wavelet coefficients have been encoded completely or another criterion has been satisfied.

D. Entropy Encoding

Entropy encoding is a coding schema that assigns codes to the symbols with the probabilities of the symbols occurs. Typically, it is used to compress data by replacing symbols with codes; therefore the most common use symbols will have the shortest codes. The process of entropy coding is split into two parts: modeling and coding. Modeling will assign probabilities to the symbols and coding will produces a bit sequence from these probabilities. An accurate probability need to be calculated so that the sequence bits will be assigned based on the probability of the symbols to achieve a better compression rate.

E. Cryptography

To make the information secure while sending via public network, encryption play an important role to protect the data confidentiality. Encryption process transformed the data into cipher text combine with a secret key or known as password, only authorized people that hold the secret key are allowed to decode it. Decryption is the reverse process of encryption by taking the cipher text combine with a secret key to reproduce the original plain text. If the same secret key is use in both encrypt and decrypt process, it is known as “symmetric”; if different key is used then it is known as “asymmetric” [6].

There are two main encryption methods: block cipher and stream cipher. Block cipher operates on a group of bits called as block that are typically 64 bits or 128 bits long. Examples of block ciphers are DES, triple-DES, AES, Blowfish etc. Stream ciphers are design to operate much faster than block ciphers. Normally, stream cipher [7] operates on smaller units in bits or sometimes in one byte at a time. It converts a key into a random key stream (a sequence of bit use as key) that combines with plaint text to encrypt it, usually with bit wise XOR operation. Examples of stream ciphers are RC4, SEAL, and SOBER. RC4 is selected as the encryption algorithm for this research.

RC4 is one of the fastest ciphers designed by Ron Rivest sometimes in the 1990's. It is essentially a pseudo-random number generator initialized from a secret key of up to 256 bytes. The RC4 algorithm generates a "keystream" which is simply XORed with the plain text to produce the cipher text stream. Decryption is exactly the same as encryption.

III. Methodology

A. To Increase Security Level

Encrypting image to hide the content is an important step to ensure image security. Besides the image content, few important parameters required for recovering the image should also be fully encrypted. Those parameters are:

i. Wavelet algorithm (if multiple type of wavelet algorithm are supported e.g. Haar, Daubechies, Antonini) – different wavelet algorithms will calculate the signal in different ways; the results produced will be definitely different from each other.
ii. Scale of compression – compression scale can be set

during compression process. To decode the image, the scale needs to be set correctly to prevent out of range error during decompression process.

iii. Initial threshold – initial threshold is set at the starting point to differentiate the significant level for each coefficient value. Decode process will identify incorrect significant level from the beginning if invalid threshold is provided.

iv. Scan order (if multiple type of scan order supported e.g. Raster scan, Morton scan) – different scan order algorithm scan the position in different way to produce the compression code. Invalid scan order may lead the algorithm to interpret the compress code incorrectly.

v. Size of image (width, height) – after image has been compressed, the width and height is unknown. Decoder needs the information to set the image size properly.

vi. Entropy algorithm (if multiple type of entropy algorithms are supported e.g. Range Coder, Arithmetic Coder [5], Huffman coder) – different entropy encoder algorithm have different ways in data representation. If wrong entropy algorithm is supplied during entropy decode process, invalid data will be produced.

The above parameters are important to recover the image. In order to increase the security level, the propose method will consist of the following step:

i. Collect all the required parameters.

ii. Encrypt the parameter with a secret key. A simple XOR method is enough to encrypt the value.

iii. Write the bits scatter around the image in different position. The position of the bits is based on calculation from secret key. Randomly insert the bits around the image is used to increase the difficulty of retrieving the parameters and protect the image with invalid bits.

B. To Reduce Space Consumption

Image compression is a process to reduce the image file size. In order not to violate the rule, integrating security feature should not increase the file size or causing any side effect to the compression process.

Most of the image encryption is done after performing EZW encoding. In this research, the method of encrypting image data after it has gone through entropy encoding process is introduced. The reason of proposing this design is to take advantage on the EZW encoding. In EZW encoding, only four symbols will be used to represent the image; positive significant (P), negative significant (N), Zero tree (T) and Isolated tree (Z). If encryption is done after EZW encoding, it may end up having more than four symbols. Increasing of symbols will cause entropy encoding to have a wider range of character possibility, thus it cannot reduce the file size efficiently. Whereas if encryption is performing after entropy encoding, only four symbols will be processed, the possibility appear other than P, N, T, Z is zero. Therefore, entropy encoding can further compress the size effectively. Moreover, the file size is smaller after performed entropy encode, the encryption process can save computational time to encrypt smaller file size compare to bigger file size.

C. Security Feature Integrate into Arithmetic Encoding

To make image secure to transmit over public channel, compression and encryption should be carried out simultaneously to save the computational power and time. In arithmetic encoding, the symbols are coded depending on the interval value. Interval value correlates with probability where higher probability will have wider interval. To make the image secure, the proposed method will encrypt the interval while the image is compressed. Failed to get the original value will lead to invalid symbol.

IV. Implementation

A. Implementation Architecture

To implement the secure image compression, it divided into three main steps.

First step: Discrete wavelet transform – with image coefficient value, it is transformed to multilevel level decomposition image by going through high pass filtering and low pass filtering.

Second step: Embedded zero tree encoding – with it own data structure representation, embedded zerotree transformed the image into bit stream and compress it.

Third step: Arithmetic entropy encoding – further compress the encoding bit stream with a secret key to protect the image confidentially.

V. Result

A. Result for Space Size Reduction

Encryption process done after compression shown a great space saving and a slight time saving for the whole process. Following result is tested with entropy process only. There are some assumptions made:

i. Non-encrypted image source image already encoded with Embedded Zerotree Wavelet encoding and come out with the 'P','N','T','Z' symbols. The parameters used are as below:
   - Wavelet decompose level – 4
   - Wavelet algorithm – Antonini algorithm
   - Scan order algorithm – Morton scan algorithm
   - Bit per pixel (bpp) – 1

ii. Encrypted image source image already encoded with embedded zerotree wavelet encoding and the symbols is encrypted with a secret key using RC4 algorithm.
   - Wavelet decompose level – 4
   - Wavelet algorithm – Antonini algorithm
   - Scan order algorithm – Morton scan algorithm
   - Bit per pixel (bpp) – 1
   - Password – 123456

iii. Original image size is the file sizes that already completed with embedded zerotree wavelet encoding and prepare to run arithmetic encoding

iv. Both non-encrypted image size and encrypted image size is the final file size that completed arithmetic encoding.

TABLE 1: The Disk Space Utilization Between Non-Encrypt Image And Encrypted Image Used For Compression

Image Title	Original Image Size (bytes)	Non-Encrypted Image Size (bytes)	Encrypted Image Size (bytes)
Dog	8,328	1,197	8,392
Frog	29,218	4,076	29,310
Bball	28,069	4,539	28,164
Bird	70,498	9,820	70,636
House	62,684	9,843	62,808
Lena	63,742	9,640	63,872
Barb	244,372	40,086	244,684
Goldhill	245,084	37,847	245,397
Peppers	259,682	37,312	260,005
Mountain	252,102	43,605	252,426

From the Table 1, the result shown most of the compression ratio is average on 80% and above for non-encrypted file. The reason behind is non-encrypted file have a lot of similarity, only 4 types of symbol ('P','N','T','Z') is used to represent the image. From the similarity, we can get a great compress ratio.

Whereas for the image which is encrypted, it contain 255 symbols (8 bits ASCII key). If the encryption algorithm is strong enough, it will try to generate the most random symbol to represent the image. Generating random symbol is tried to avoid cryptanalysis to launch attack using the similarity of the symbol. Since the similarity between symbols already been avoided, the image will fail to compress efficiently. This is the reason that most of the encrypted images having bigger file size compare to original image because the compression cannot perform well and some space required by arithmetic encoding algorithm to record how it compress the image. .

B. Result for Time Saving

There is a slight time save achieves with the method proposed (see Table 2). This is due to the symbol search while updating the symbol frequency when the algorithm is running. Search process for 4 symbols definitely will be faster than 256 symbols. That is the reason that few milliseconds will be save for 4 symbols searching.

TABLE 2: The Time Used Between Non-Encrypt Image and Encrypted Image Used for Compression

Image Title	Original Image Size (bytes)	Non-Encrypted Image (second)	Encrypted Image (second)
Dog	8,328	0.03	0.09
Frog	29,218	0.12	0.20
Bball	28,069	0.12	0.21
Bird	70,498	0.21	0.26
House	62,684	0.19	0.25
Lena	63,742	0.19	0.25
Barb	244,372	0.56	0.82
Goldhill	245,084	0.56	0.81
Peppers	259,682	0.56	0.87
Mountain	252,102	0.57	0.85

C. Result for Security Features Implementation

Security is one of the requirements for this project. To protect the image, 2 security features have been added. Secret key will play an important role to encrypt and recover the image.

Insert the bits into image stream

i. Collect all the required data to form a parameter with 128 bits.
ii. With secret key, generate 128 locations.
iii. Loop for 128 bits parameter
iv. For each location found in the image stream, insert one bit into the image stream.

Retrieve the bits from image stream

i. With secret key, generate 128 locations.
ii. Scan through image stream to find the location. Remove the bit from the image stream and record it. The process need continue until fully retrieve the 128 bits.
iii. Extract the data from the parameter.

1) Encrypt the arithmetic interval value of the image

During Arithmetic compression process, each of the symbols needed to go through an arithmetic formula to determine how many bits will be recorded. In the algorithm, interval value played an important role to determine the final result for each symbol.

If attacker failed to provide the correct password, 2 incidents will be happen:

i. Infinite loop that cause the program to hang. This is due to the calculation in arithmetic encoding did not actually meet the expected value. It will keep calculating with never ending result.
ii. Produce invalid image.

VI. CONCLUSION

Three approaches have been proposed to increase the compression efficiency and to integrate security features with minimum impact on the computation cost. First, the paper discussed on how to increase the security by encrypting the decoding parameter and randomly inserts each bit into the image stream. Second, the paper discussed on how to integrate the encryption into arithmetic algorithm to protect the image against compression. Finally, flow modification has been proposed to maximize the compression efficiency.

REFERENCES

[1] SHAPIRO, J. M., "Embedded Image Coding Using Zerotrees in the EZW Algorithm." IEEE Transactions on Signal Processing, December 1993, Vol. 41, No 12, pp. 3445-3462
[2] Subhasis Saha, Image Compression – from DCT to Wavelets: Review
[3] SHAPIRO, J. M., "A Fast Technique for Identifying Zerotrees in the EZW Algorithm" Proc. Of IEEE International Conference on Acoustics, Speech and Signal Processing, May 1996, Vol. 3, pp. 1455-1458
[4] C. Valens, "Embedded Zerotree Wavelet Encoding", 1999
[5] PAUL G.HOWARD and JEFFREY SCOTT VITTER, "Arithmetic Coding for Data Compression".
[6] William Stalling, "Cryptography and Network Security", Prentice Hall, 2003, Third Edition.
[7] M.J.B Robshaw, "Stream Ciphers", RSA Laboratories Technical Report TR-701, 25 July, 1995, Version 2.