International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 09 | Dec-2015 www.irjet.net p-ISSN: 2395-0072

A SURVEY ON DIFFERENT COMPRESSION TECHNIQUES FOR EFFICIENT IMAGE TRANSFER

Remya G R[1], Smitha J C[2]

[1] *Student, Computer Science and Engineering, Lourdes Matha College of Science and Technology, Kerala, India*
[2] *Assistant Professor, Computer Science and Engineering, Lourdes Matha College of Science and Technology, Kerala, India*

Abstract - ***Efficient image compression is essential for applications such as transmission and storage in databases. In many practical applications, image encryption has to be performed before image compression. Various compression and encryption methods are used here. This paper compares different compression methods. In the first method, compression takes place by wavelet transform. Second method describes an image compression algorithm based on VQ. Third method describes the compression of encrypted images. In the fourth method, compression takes place by discarding the excessively rough and fine information of coefficients generated from orthogonal transform. The fifth method describes the compression of data encrypted with block ciphers. Finally, in the sixth method an arithmetic coding-based approach is exploited to efficiently compress the encrypted images.***

Key Words: *Encryption-Then-Compression, Compression-Then-Encryption, Low density parity check*

1. INTRODUCTION

In imaging science, image processing is processing of images using mathematical operations by using any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two- dimensional signal and applying standard signal-processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing also are possible [2]. This article is about general techniques that apply to all of them. The acquisition of images (producing the input image in the first place) is referred to as imaging. Closely related to image processing are computer graphics and computer vision.

In computer graphics, images are manually made from physical models of objects, environments, and lighting, instead of being acquired (via imaging devices such as cameras) from natural scenes, as in most animated movies. Computer vision, on the other hand, is often considered high-level image processing out of which a machine/ computer / software intends to decipher the physical contents of an image or a sequence of images (e.g., videos or 3D full-body magnetic resonance scans) [2].In modern sciences and technologies, images also gain much broader scopes due to the ever growing importance of scientific visualization (of often large-scale complex scientific/experimental data)[2] . Image processing is a technique applied for processing an input image and to get the output as either improved form of the same image or original input image. Now a day's use of a multimedia data is increased, due to this multimedia data security come into picture. Images are transmitted over networks on large scale, we need to have a reliable technique to prevent data getting leaked or attacked. The security mechanism should be a reliable method to protect the images [3]. To ensure the security of electronic data cryptographic techniques are used. Image encryption is one of them and larger images are difficult to process hence image compression can be done after encryption process. Even though the Compression-then-Encryption (CTE) paradigm meets the requirements in many secure transmission scenarios, the sequence of applying the compression and encryption needs to be reversed in some other situations. As the content owner, ie, sender is always interested in protecting the privacy of the image data through encryption. But, A has no incentive to compress her data and hence, will not use her limited computational resources to run a compression algorithm before encrypting the data. This is especially true when sender uses a resource-deprived mobile device. In contrast, the channel provider has an overriding interest in compressing all the network traffic so as to maximize the network utilization. And channel provider typically has abundant computational resources. A very big challenge within such Encryption-then-Compression (ETC) [1] framework is that compression has to be conducted in the encrypted domain.

To promote faster transmission and prevent data loss during transmission, different compression algorithms are used to reduce the size of the data during transmission. If a compressed file is encrypted, it has better security and faster transfer rate across the network than encrypting and transferring uncompressed file. But in some cases, compression also increases the overhead, so that there is a need to analyze different cryptographic algorithms for various parameters so as to understand the factors that can affect the performance of the cryptographic algorithms. If compression is needed then identify the best suitable compression algorithm that should be used for compressing the file according to data type and data size to reduce the overhead of time for compression and increase the efficiency and security to data that is being transferred. The existing system addresses the problem in compression performance after encrypting an image. In this scheme compressing the encrypted images is almost equally efficient as compressing their original image. The objective of the system is to improve the existing systems that are able to transfer an image securely and efficiently. This approach designs the image encryption and then compression which is suitable for both lossy and lossless images. Now a day's transmitting an image efficiently through the secure channels in secure manner is a big problem. In image security the cryptography algorithm should be applied to every pixels of image and in the receiver side all of the pixels must be decoded. Most of the times while any image is transferring across the network for security reasons they are normally encrypted directly to make user visibly unreadable.

Nowadays attackers are too intelligent to break the encrypted images and obtain original contents. Different techniques are designed to combine both encryption and compression in a single system which provides greater security. This paper discusses the comparison of five different compression methods.

2. IMAGE COMPRESSION

Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level [4]. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from Web pages. Here describes different methods for compression techniques. Image compression may be lossy or lossless. Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art, or comics. Lossy compression methods, especially when used at low rates, introduce compression artifacts. Lossy methods are especially suitable for natural images such as photographs in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences may be called visually lossless.

3. COMPRESSION TECHNIQUES

3.1 Image compression by wavelet transform

Wavelets are functions defined over a finite interval. Wavelet has an average value of zero[7]. The basic idea of the wavelet transform is to represent any arbitrary function (t) as a superposition of a set of such wavelets or basis functions. These basis functions or baby wavelets are obtained from a single prototype wavelet called the mother wavelet, by dilations or contractions (scaling) and translations (shifts). The Discrete Wavelet Transform of a finite length signal x (n) having N components, for example, is expressed by an N x N matrix[5].

3.2 VQ Compression

A vector quantizer is composed of two operations. The first is the encoder, and the second is the decoder[7]. The encoder takes an input vector and outputs the index of the codeword that offers the lowest distortion. In this case the lowest distortion is found by evaluating the Euclidean distance between the input vector and each codeword in the codebook. Once the closest codeword is found, the index of that codeword is sent through a channel. When the encoder receives the index of the codeword, it replaces the index with the associated codeword. The fundamental idea of VQ[6] for image compression is to establish a codebook consisting of code vectors such that each code vector can represent a group of image blocks of size m × m, (m=4 is always used). An image or a set of images is first partitioned into m × m non overlapping blocks which are represented as m2-tuple vectors, called training vectors. The size of training vectors can be very large.

3.3 Compression of Encrypted Images

This technique describes how to decode images using a model designed to capture the underlying 2-D structure of images. The result is more efficient compression of encrypted images and implements a practical scheme, based on LDPC (Low Density Parity Check) codes (for compressing encrypted images). Also describe how to apply this scheme to binary images [8].

Here present a practical encoder and decoder for compressing encrypted images. Begin by assuming that full knowledge of the source statistics (p, h_0, h_1, v_0, v_1) is available to both encoder and decoder. Compress the encrypted source using a sparse linear transformation implemented with a matrix multiplication. The design of the transform matrix is based on LDPC codes.

The decoder operates by running belief propagation over the factor graph [6]. Thus proceed by describing the appropriate factor graph. The graphical model consists of three components connected together; the models for the source, the encryption, and the code. Form the encryption model and attach it to the source model as shown in the Fig -1: , Since we consider only stream ciphers here, model the encryption process as $y_{i,j} = x_{i,j} \oplus k_{i,j}$, where $y_{i,j}$ is the cipher-text, $k_{i,j}$ is the bits of the key, and $\oplus$ indicates the exclusive-OR operation. Then represent the constraint between these three variables in the graphical model with a square node labeled $f_{ei,j}$. The circles representing the variables $x_{i,j}$, $k_{i,j}$, and $y_{i,j}$ are all connected to the encryption constraint $f_{ei,j}$. The code model consists of a representation of the linear transformation matrix H, the cipher bits $y_{i,j}$, and the compressed bits s_i . It was shown that good performance can be achieved when the transformation matrix is designed as an LDPC code. The squares labeled f_{si} represent the linear transformation H, and the results of that transformation are represented by the circles labeled s_i (i.e., the compressed bits).

Decoding is achieved using the sum-product algorithm on the factor graph. The sum-product algorithm is an inference algorithm designed to be exact on trees. Empirical performance is very good. The first half of each iteration, source being smooth. The algorithm is iteratively updates an estimate of the distribution for each of the variables. Typically between 30% and 50% of the compressed bits are doped bits. These bits are used in two ways. First, since these doped bits are known unambiguously at the decoder they anchor the iterative decoding process by catalyzing the process. Second, they provide a mechanism for estimating the statistics of the masked source. By selecting the doped bits to come in adjacent pairs, the decoder can empirically estimate the source statistics and use the estimates for decoding.

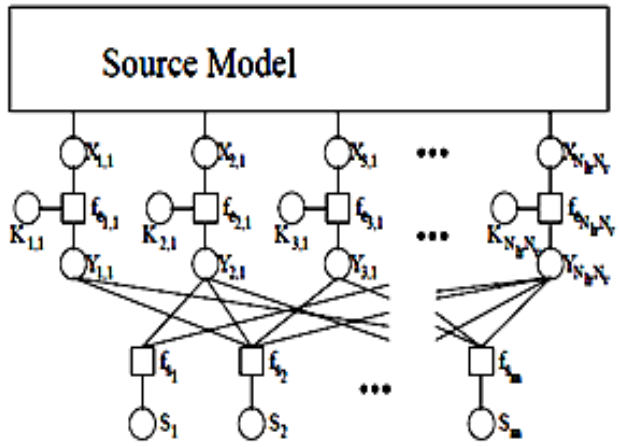

Fig -1: The full graphical model for compressing encrypted spatially correlated sources. The model consists of the source model on top.

3.4 Compression via Orthogonal Transform

The number of pixels in the rows and the columns of the original image is N_1 and N_2, and the number of all pixels as ($N = N_1 \times N_2$). In the compression method, a majority of pixels is converted to a series of coefficients using an orthogonal transform. Then the excessively rough and fine information in the coefficients is discarded, which leads to a reduced the data amount.

The network provider divides it into two parts: the first part made up of α. N pixels and the second one containing the rest of the (1- α).N pixels. Denote the pixels in the first part as $p_1, p_2, \ldots p_{\alpha. N}$ and the pixels in the second part as $q_1, q_2, \ldots q_{(1-\alpha). N}$. The value of α is within (0, 1). Here, the data in the first part will be reserved while the data redundancy in the second part will be reduced. The pixels in the first part are called rigid pixels and the second part is called elastic pixels. Perform an orthogonal transform [9] in the elastic pixels to calculate the coefficients $Q_1, Q_2 \ldots Q_{(1-\alpha). N}$. Here, H is a public orthogonal matrix with a size of (1-α). N × (1-α).N , and it can be generated from orthogonalizing a random matrix.

Then for each elastic pixel calculate,

$$s_k = \mathrm{mod}\left[\mathrm{round}\left(\frac{Q_k}{\frac{\Delta}{M}}\right), M\right], k = 1,2,\ldots,(1-\alpha).N \qquad (1)$$

Here Δ and M are system parameters.

The round operation returns the nearest integer and the mod operation gets the remainder. Q_k is converted into an integer s_k within [0, M-1] .With a small M, the data amount for representing the elastic pixels is reduced. Then Q_k can be rewritten as

$$Q_k = r_k . \Delta + s_k . \frac{\Delta}{M} + t_k \qquad (2)$$

The rough information r_k and the fine information t_k are discarded, and then only the information on the medium level s_k remains. The rough information r_k will be retrieved by an iterative image reconstruction procedure, and the loss of the fine information t_k cannot seriously affect the quality of the reconstructed image. Since s_k are within [0, M-1]. Segment the set of s_k into many pieces with L_1 digits. Then, convert each decimal value into L_2 bits in a binary notational system, where,

$$L_2 = \lceil L_1 . \log_2 M \rceil \qquad (3)$$

the total length of bits generated from all pieces of s_k is,

$$L = (1- \alpha). N. \frac{L_2}{L_1} \approx (1 - \alpha). N. \log_2 M \qquad (4)$$

Collect the data of rigid pixels, the bits generated from all pieces of s_k , and the values of parameters including N_1, N_2

, α , Δ, M, and L_1 to produce the compressed data of encrypted image.

3.5 Compression of Data Encrypted With Block Ciphers

Here describes compression of data encrypted with block ciphers, such as the Advanced Encryption Standard. As opposed to stream ciphers, such as the one-time pad, block ciphers are highly nonlinear and the correlation between the key and the cipher text is, by design, hard to characterize. If a block cipher operates on each block of data individually, two identical inputs will produce two identical outputs. While this weakness does not necessarily enable an unauthorized user to decipher an individual block.

It can reveal valuable information; for example, about frequently occurring data patterns. To solve this problem, various chaining modes, also called modes of operation, are used with block ciphers. The idea is to randomize each plaintext block by using a randomization vector derived from previous encrypted inputs or outputs. This randomization prevents identical plaintext blocks from being encrypted into identical cipher text blocks. Here focus on the cipher block chaining (CBC) mode [10]. The CBC mode is interesting because it is the most common mode of operation used with block ciphers. The CBC mode is interesting because it is the most common mode of operation used with block ciphers.

3.6 Compression via Arithmatic Coding

Assume the encrypted file is I_e. In Fig -2: show the diagram of lossless compression of I_e. Using a de-assembler, the encrypted image Ie is divided into L segments $\tilde{C}_0, \tilde{C}_1 \ldots, \tilde{C}_{L-1}$. An adaptive AC is used to losslessly encode each prediction error sequence $\tilde{C}_k$ into a binary bit stream B_k. The generation of all B_k can be carried out in a parallel manner to improve the throughput. Then an assembler concatenates all B_k to produce the final compressed and encrypted bit stream B. Here, the random permutation only changes the locations, not changing the values of the prediction errors. This leads to the preservation of the probability mass function (PMF) [1] of prediction error sequence, which drives the adaptive AC. The length of the resulting compressed bit stream can compute by,

$$L_c = |\mathbf{B}| + (L-1)\lceil \log_2 |\boldsymbol{B}| \rceil \quad (5)$$

Where |B| is measured by bits, and the second term denotes the overhead induced by sending the side information $|B_k|$, for $0 \le k \le L-2$.

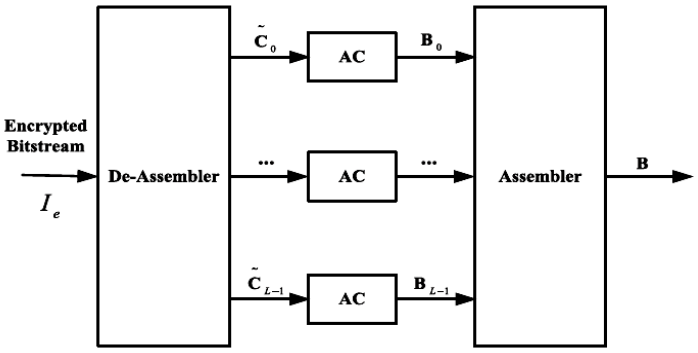

Fig -2: Schematic diagram of compressing the encrypted data

Table -1: Comparison between Compression Techniques

Method	Advantages	Disadvantages
Wavelet	High compression ratio	Coefficient quantization,Bit allocation
VQ	Simple decoder No-coefficient quantization	Slow codebook generation, Small bpp
Compression Via Finding Orthogonal Transform	Better Compression Efficiency	Complex Method
Compression Of Data Encrypted With Block Ciphers	Better Compression Efficiency	Fundamental Compression
Compression Of Encrypted Images	Simple	Less Compression Gain is available

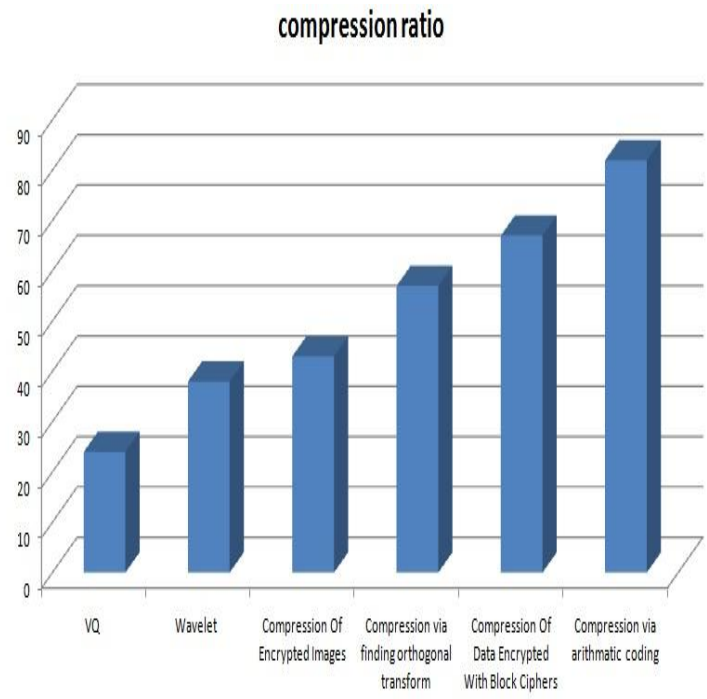

Chart -1: Compression Ratio Analysis

4. CONCLUSION

This paper compares five different compression methods. In the first method, compression takes place by wavelet transform. Second method describes an image compression algorithm based on VQ. Third method describes the compression of encrypted images. In the fourth approach, the image is compressed by discarding the excessively rough and fine information of coefficients in the transform domain.

The fifth method describes compression of data encrypted with block ciphers. Finally, in the sixth method, highly efficient compression of the encrypted data has been realized by a context-adaptive arithmetic coding approach. Advantages and disadvantages of the different compression methods are in Table 1 and compression ratio analysis is shown in Chart No.1.

sixth method is better and efficient than all others. It is highly efficient. Efficient compression is achieved and security level is also high. It is used in many applications like media application and cloud server data storage.

REFERENCES

[1] Jiantao Zhou," Designing an Efficient Image Encryption-Then-Compression system via Prediction Error Clustering and Random Permutation",in IEEE Transactions On Information Forensics And Security, Vol. 9, No. 1, January 2014.
[2] https://en.wikipedia.org/wiki/Image_processing.
[3] Riyaz Sikandar Kazi, Prof. Navnath Pokale," Secure Image Transfer Using Clustering and Permutation Based Approach",IJARCET,Vol.4, June 2015.
[4] http://whatis.techtarget.com/definition/image-compression
[5] Chan, Y. T., "Wavelet Basics", Kluwer Academic Publishers, Norwell, MA, 1995.
[6] Gersho, A., Gray, R.M., "Vector Quantization and Signal Compression", Kluwer Academic Publishers, 1991.
[7] Sachin Dhawan, "A Review of Image Compression and Comparison of its Algorithms", in IJECT Vol. 2, Issue 1, March 2011
[8] D. Schonberg, S. C. Draper, and K. Ramchandran, "On compression of encrypted images," in Proc. IEEE Int. Conf. Image Process., Oct. 2006, pp. 269–272.
[9] X. Zhang,"Lossy compression and iterative reconstruction for encrypted image," IEEE Trans. Inf. Forensics Security, vol. 6, no. 1, pp. 53–58, Mar. 2011.
[10] D. Klinc, C. Hazay, A. Jagmohan, H. Krawczyk, and T. Rabin, "On compression of data encrypted with block ciphers," IEEE Trans. Inf. Theory, vol. 58, no. 11, pp. 6989–7001, Nov. 2012.
[11] J. Zhou, X. Liu, and O. C. Au, "On the design of an efficient encryption-then-compression system," in Proc. ICASSP, 2013, pp. 2872–2876.
[12] W. Liu, W. J. Zeng, L. Dong, and Q. M. Yao, "Efficient compression of encrypted grayscale images," IEEE Trans. Imag. Process., vol. 19, no. 4, pp. 1097–1102, Apr. 2010.
[13] X. Zhang, G. Feng, Y. Ren, and Z. Qian, "Scalable coding of encrypted images," IEEE Trans. Imag. Process., vol. 21, no. 6, pp. 3108–3114, Jun. 2012.
[14] A. Kumar and A. Makur, "Lossy compression of encrypted image by compressing sensing technique," in Proc. IEEE Region 10 Conf. TENCON, Jan. 2009, pp. 1–6.