IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 4, Issue 5, May 2016

Image Tampering and Protection Using Watermark Algorithm

Biju V.G.[1], Anith Mohan[2], Nisha V.[3], Hebsa Thomas[4], Reshma C.Raj[5], Adarsh D.S.[6]

Associate Professor, Dept. of E.C.E, College of Engineering, Munnar, Munnar, India[1]

Assistant Professor, Dept. of E.C.E, College of Engineering, Munnar, Munnar, India[2]

U.G. Scholar, Dept. of E.C.E, College of Engineering, Munnar, Munnar, India[3,4, 5, 6]

Abstract: In the current scenario image tampering remains as a challenging issue. In this paper water marking algorithm for the protection of images against tampering is used. The method can detect and localize alterations of the original image. A secure key and a random number generator are used to hide the information in a secret, undetectable and unambiguous way. Here Set Partitioning in Hierarchical Trees (SPIHT) is the source channel coding algorithm for image recovery and Reed Solomon code for channel coding. The encoded image is decoded at the receiver end for the recovery. The recovery of tampering efficiently without any distortion is accomplished here.

Keywords: Watermarking, SPIHT, Reed-Solomon, Permutation, Hash detection.

I. INTRODUCTION

For the easier editing and perfect reproduction, the protection of ownership and the prevention of unauthorized manipulation of digital audio, image, and video materials are the important aspects which is to be considered. Digital watermarking has made considerable progress in recent years. There are several categories of watermarking schemes. Among them, fragile watermarking is a technique that is used for both authentication of the received image and localization of the tampered zone. The receiver declares the image as unaltered if the hash output is the same as the one transmitted from the original image. The image transfer by means of hash requires a secure channel that must be reused for each image transmission. Since such a channel is unavailable, a more applicable approach is to embed the verification data into image. This is fragile watermarking. But this method is not preferred since it does not provide 100% localization and the tampering recovery.

In this paper, the focus is on watermarking for digital image authentication. The scheme used must possess some key characteristics like- to determine whether an image has been tampered or not, to locate the tampered area and also to recover the original image and the water marking algorithm focuses on the above three characteristics as a result of which this algorithm became a widely popular one.

Watermarking exist in today's hierarchy basically in two types; visible and invisible watermarking. In visible watermarking the watermark is spatially embedded into the host image to create a watermarked image. Visible watermarking algorithms are generally less complex compared to invisible. The digital eras have begun and there is high possibility to change the ownership details and visible watermarking alone cannot withstand the image processing attacks. This lead to the introduction of invisible watermarking. In this technique watermark is embed into the original content in spatial or frequency domain using DCT, FFT, and DWT.

In this method the total watermark bits can be divided into mainly three; they are source encoder bits, channel code parity bits and check bits. In this algorithm the input bits are source coded and the output bits stream is protected using channel encoder. The image recovery of the erasure location can be done by checking the check bits to retrieve the original image. This technique does not affect the clarity of the image.

This paper basically gives an idea on the proposed watermarking technique used as well as the different steps involved in channel coding and steps taken for tampering detection and image recovery. The section following the introduction is methodology which gives us a brief idea on how the image is watermarked and sends through the channel. The source coding algorithm and the steps involved are mentioned later on which is followed by the channel coding which reed solomn code is also presented in this section. The hash generation using XORing technique is described after that. Tampering detection is the next step which has to be followed after which the tampered location is recovered and the complete image recovery is explained. The third section includes the result attained from this project. The last section concludes the topic by quoting the advantages of the topic and also the possibilities for expansion in the area.

II. METHODOLOGY

The goal of this algorithm is to embed a watermark into original image to protect it against tampering. The MSB bits of each pixel are kept unchanged while the LSB are used for watermark embedding. The image is first compressed using a source coding algorithm. In this embedding phase the original image is represented as 8 bit gray scale pixel values and this 8 bit is divided into 4 parts. MSB (which remains unchanged and used for hash generation and image reconstruction), check bits, source code bits and parity bits.

IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 4, Issue 5, May 2016

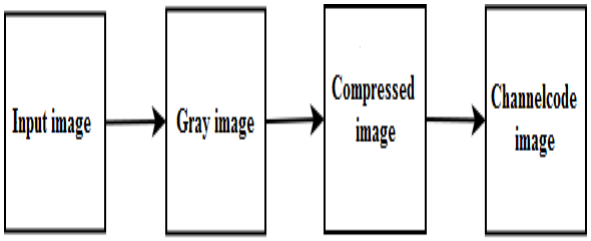

Fig1: Image compression and channel coding

The source coding algorithm used here is (SPIHT) Set Partitioning in Hierarchical Trees algorithm. By using this technique the extract an estimation of the original image by truncating its output in every desired rate. Hence it is a compression algorithm. Wavelet transform is employed in this algorithm, So that the sorting order from the encoder section can be made available even at the decoder section.

STEPS IN SOURCE CODING SPIHT ALGORITHM:

STEP 1: In the sorting pass, the List of Insignificant Pixel (LIP) is scanned to determine whether an entry is significant at the current threshold. If an entry is found to be significant, output a bit '1' and another bit for the sign of the coefficient, which is marked by either '1' for positive or '0' for negative. Now the significant entry is moved to the list of significant pixel (LSP). If an entry in LIP is insignificant, a bit '0' is output.

STEP 2: Entries in List of Insignificant Set (LIS) are processed. When an entry is the set of all descendants of a coefficient, named 'type A', magnitude tests for all descendants of the current entry are carried out to decide whether they are significant or not. If the entry is found to be as significant, the direct offspring's of the entry undergoes magnitude tests. If direct offspring is significant, it is moved into LIP; otherwise it is moved into LSP. If the entry is deemed to be insignificant, this spatial orientation tree rooted by the current entry was a zero-tree, so a bit '0' is output and no further processing is needed. Finally, this entry is moved to the end of LIS as 'type B', which is the set of all descendants except for the immediate offspring of a coefficient. If the entry in LIS is type B, significance test is performed on the descendants of its direct offspring. If significance test is true, the spatial orientation tree with root of type B entry is split into four sub-trees that are rooted by the direct offspring and these direct offspring's are added in the end of LIS as type A entries. The important thing in LIS sorting is that entire sets of insignificant coefficients, zero-trees, are represented with a single zero. The purpose behind defining spatial parent-children relationships is to increase the possibility of finding these zero-trees.

STEP 3: Finally, refinement pass is used to output the refinement bits (nth bit) of the coefficients in LSP at current threshold. Before the algorithm proceeds to the next round, the current threshold is halved.

Reed Solomon code (RS) is used as channel coding algorithm which is used for tampering detection and image recovery. A channel coding in a particular rate 'R' is applied to the permuted compressed image by knowing the site of tampered block at the receiver end. The original image is divided into different blocks of size N*N. Each block has $b_c=n_c*N^2$channel code bits. These bc bits whose row and column are turned into a binary stream of brc bits called position bits, brc along with $b_m=n_m*N^2$ MSB of each block are used as input to a hash generator algorithm to produce $b_h=n_h*N^2$ hash bits. A binary key of length bh is generated at the embedding phase which remains fixed over the whole image. Check bits are generated by XORing key with hash bits, bh and bc bits are spread over the block and as a result last $n_w=n_h+n_c$ are replaced (ie, LSB of original image), where nw is the number of LSB per pixel used for watermark embedding.

After the LSB detection the image is decomposed into blocks of size N*N.

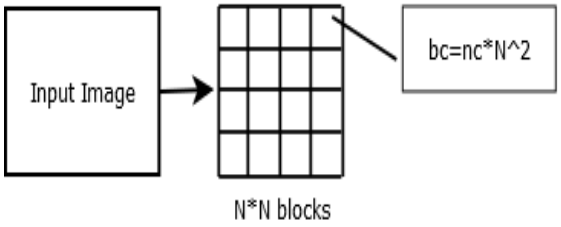

Fig.2: bc=channel code bit, included brc position bits of every location

Watermarked image is produced when all blocks are processed.MD5 (Message-Digest algorithm 5): is a widely used cryptographic function with a 128- bit hash value. MD5 has been employed in a wide variety of security applications, and is also commonly used to check the integrity of files. An MD5 hash is typically expressed as a 32-digit hexadecimal number.

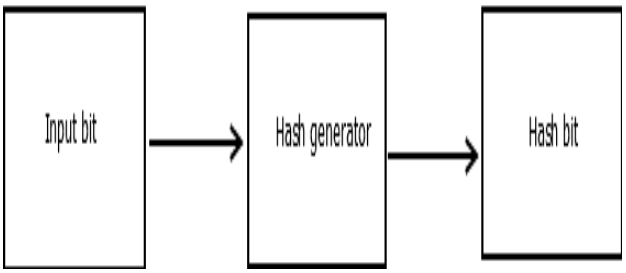

Fig 3: Hash generation

The received image which is tampered is decomposed into N*N blocks. Position bits are found using k2 (security key) and block bits are decomposed to nm MSB and nw watermark LSB per pixel. That is, $b_m=n_m*N^2$ MSB and $b_w=n_w*N^2$ watermark bits. Watermark bits are decomposed to $b_h=n_c*N^2$ check bits and $b_c=n_c*N^2$ channel code bits, brc position along with b_m MSB are used to generate b_h hash bits. XOR of hash bits and extracted check bits are recorded. The key in the embedding phase is equal to the bit stream in the case of an unaltered block. Tampered block location can be identified by the comparison of bit stream and he key. Channel code bits are collected evenly from the image after identifying n_c.

Fig. 4: Check bit formation

IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 4, Issue 5, May 2016

The channel code bits are then inversely permuted and provided to RS decoder along with the tampered location. This compressed output is given to the source decoder after this. The output of this source decoder is the reconstructed image.

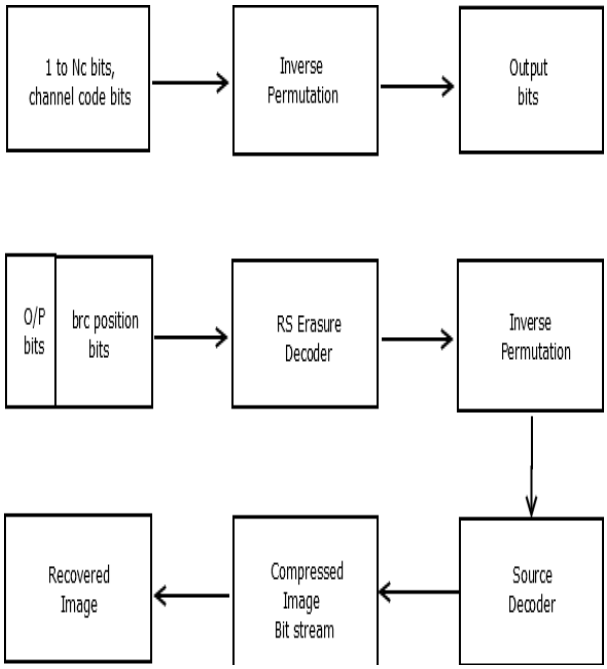

Fig.5: Image recovery

III. RESULTS AND DISCUSSION

In this paper we have implemented the watermark embedding of a scenery image monarch.png for the purpose of protection. The data is embedded on different LSB values and the corresponding variations shown in the image are verified. From the method we will be able to show that the noticeable distortion happening in the original image can be avoided by watermark embedding, since this technique preserves the quality of the original image. The recovery of the tampered image is also implemented. The result is as shown below.

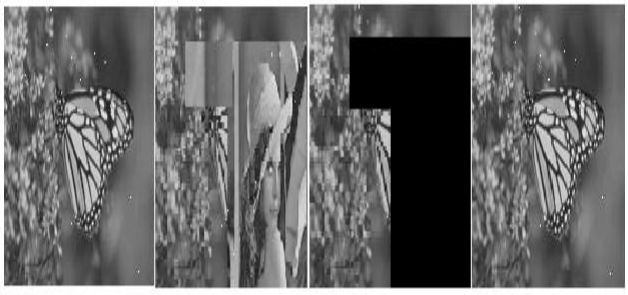

Fig.6:(a).Original image (b).Tampered image (c).Tampered area detected (d).Reconstructed image.

The PSNR analysis is done based on the equation,

PSNR=10*log(255*255/MSE) / log(10)

where, MSE is the Mean Square Error. The PSNR value of the image monarch.png is 37.762.

IV. CONCLUSION

In this paper, a general source-channel coding framework is proposed to solve the image tampering protection problem. The original image is compressed using an efficient source encoder (SPIHT), and the output bit stream is protected against tampering through RS channel codes. For each block, check bits are calculated and embedded. These bits are used to locate the tampered blocks. If the tampering rate is below a certain limit, the channel erasure decoder succeeds, and the compressed version of the original image is recovered.

REFERENCES

[1] M. Wu and B. Liu, Watermarking for image authentication, in Proc. Int.Conf. Image Process. (ICIP), vol. 2. 1998, pp. 437441.
[2] J. Fridrich, Image watermarking for tamper detection, in Proc. Int. Conf.Image Process. (ICIP), vol. 2. Oct. 1998, pp. 404408.
[3] M. Tagliasacchi, G. Valenzise, and S. Tubaro, Hash-based identification of sparse image tampering, IEEE Trans. Image Process., vol.18.no.11,pp. 24912504, Nov. 2009.
[4] S. Roy and Q. Sun, Robust hash for detecting and localizing imagetampering, in Proc. IEEE Int. Conf. Image Process. (ICIP), vol. 6. Sep./Oct. 2007, pp. VI-117VI-120.
[5] A. Swaminathan, Y. Mao, and M. Wu, Robust and secure image hashing,IEEE Trans. Inf. Forensics Security, vol. 1, no. 2, pp. 215230,Jun.2006.
[6] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," IEEE Trans. Circuits Syst. Video Technol., vol. 6, no. 3, pp. 243–250, Jun. 1996.
[7] J. Lee and C. S. Won, "Authentication and correction of digital watermarking images," Electron. Lett.,vol. 35, no. 11, pp. 886–887, 1999.
[8] A. Cheddad, J. Condell, K. Curran, and P. McKevitt, "A secure and improved self-embedding algorithm to combat digital document forgery," Signal Process., vol. 89, no. 12, pp. 2324–2332, 2009.
[9] X. Zhang, Z. Qian, Y. Ren, and G. Feng, "Watermarking with flexible self-recovery quality based on compressive sensing and compositive reconstruction," IEEE Trans. Inf. Forensics Security, vol. 6, no. 4, pp. 1223–1232, Dec. 2011.
[10] T.-Y. Lee and S. D. Lin, "Dual watermark for image tamper detection and recovery," Pattern Recognit., vol. 41, no. 11, pp. 3497–3506, 2008.
[11] P. Korus and A. Dziech, "Efficient method for content reconstruction with self-embedding," IEEE Trans. Image Process., vol. 22, no. 3, pp. 1134–1147,Mar. 2013.
[12] S. B. Wicker, Reed–Solomon Codes and Their Applications. Piscataway, NJ, USA: IEEE Press, 1994.