International Journal of Recent Trends in Engineering & Research (IJRTER)
Conference on Recent Trends in Electrical and Electronics Engineering (RTEEE '19)
Special Issue; March- 2019 [ISSN: 2455-1457]
DOI: 10.23883/IJRTER.CONF.20190304.026.POKSD

Source-Channel Coding Approach to Generate Tamper-Proof Images

[1] D Nishma, [2] S Jeyavinotha, [3] D Nibin, [4] N.V.Cibin, [5] I Anugraka

[1,2]Electronics and Communication Engineering, Arunachala College of Engineering for Women
[3]Electronics and Communication Engineering, Nehru Institute of Engineering and Technology
[4]Senior Engineer , Ericson Ind Pvt Ltd.,Ernakulam
[5] Senior Engineer, CTS, Ernakulam

Abstract: The privacy is a major concern in big data mining approach.Recent works in this field offer algorithms that not only localize the tampering, but also recover the original content in the lost area. The self-restoration problem can be modeled as a source-channel coding problem. The original image is compressed using an efficient source encoder. The output is then channel coded. At the receiver, decoder reveals the encoder output bit stream. The output of the source decoder is then used to replace the content of the tampered area. Watermarking the original images to protect them against tampering has recently attracted an overgrowing interest. The performance is significantly improved in terms of the quality of watermarked image, quality of the restored content, and tolerable tampering rate.

Keywords: Image tampering protection, self-recovery, and water marking.

I. INTRODUCTION

Digital imaging has been rapidly developing in last two decades, and digital multimedia products are utilized in countless applications nowadays. As a consequence of this expansive development, popular and low-cost access to image editing applications challenges the integrity of digital images. On the other hand, sophisticated techniques are required to guarantee the integrity of an image or protect it against malicious modifications. One common approach is to use the hash of the original image. The receiver declares the image as unaltered if the hash output is the same as the one transmitted from the original image [1]–[3]. Image integrity verification through hash requires a secure channel that must be reused for each image transmission.

Since such a channel might be unavailable, a more applicable approach is to embed the verification data into image itself, which is referred to as fragile watermarking. Fragile watermarks can be used for both authentication of the received image and localization of tampered zone in case of malicious modifications (tampering localization), and recovering the image information in the lost area (error concealment). Inceptive fragile watermarking techniques aim only to verify the integrity of image or locate the tampered area with limited robustness against image processing modifications [4]–[8]. More recent methods in the field of tampering detection achieve the perfect 100% localization using watermarks robust against wide variety of attacks[9]-[12]. This self-recovery watermarking trend, initiated by [13], has recently attracted growing interest. The problem of image self-recovery has been approached in numerous ways. In [14], conventional error control coding schemes are adopted for localization and restoration. Several methods embed a representation of an original image into itself for the sake of self-recovery. In [13], discrete cosine transform (DCT) coefficients or reduced color-depth version of the host image is embedded in the least significant bits (LSB) of the original image.

This representation of the original image can also be the first few DCT coefficients of each block [15], a binary image generated from the difference between the host image and its chaotic pattern [16], the hash of the original image [17], watermark derived from approximation coefficients of its wavelet transform [18], a vector quantized [19] or halftone [20] version of the original image. Fragile watermarks may also be designed for specific purposes, such as binary images [21], JPEG compressed images [22], colored images [28], [23], compression-resistant [24] orcropping resistant applications [25]. Watermark bits in self-recovery methods are conventionally fallen into two categories, namely check bits and reference bits. The check bits are used to localize the tampered blocks, while the reference bits are employed to restore the original. Therefore, the watermark consists of three parts in our algorithm: source code bits, channel code parity bits and check bits. Source code bits which act as the reference bits are the bit stream of the SPIHT-compressed original image at a desired rate. In order to survive tampering erasure, the reference bits are channel coded to produce channel code bits. Check bits are used at the receiver to determine the erasure location for the channel erasure decoder. The output of channel decoder is source decoded to find the compressed version of the original image. This work shows that by choosing appropriate parameters for source and channel encoding, our algorithm output performs existing methods in the same watermark payload of three bits per pixel (bpp). Nevertheless, since the watermark artifacts are significant for embedding in three LSB, we would recommend two-LSB version of our algorithm and show that its performance is still remarkable. The Tampered image is protected against high-rate tampering by watermark generated. Next the image is processed for encryption and decryption. Password is generated for both encrypted and decrypted images. After that image is decrypted using decryption key. Following that image is also recovered using efficient source decoder.

This paper proceeds as follows. Section II existing self-significant *nw* bits (LSB) are comprised of both channel coded bits and check bits. Having tampered blocks known using the check bits, tampering can be modeled as an erasure error. Therefore, compressed bit stream is channel coded using a code capable of resistance against certain level of erasure. At the receiver, the check bits locate

embedding schemes and its drawbacks. Section III presents our image self-recovery algorithm in general, while its components are explained in details in the subsequent sections. Section IV Block diagram for encoding to combat the channel erasure. Section V describes an example of parameter selection based on required performance. Experimental results are presented and discussed in Section VI, and Section VII, concludes the paper.

II. EXISTING METHOD

In this method, the total watermark bit-budget is dedicated to three groups: 1) source encoder output bits; 2) channel code parity bits; and 3) check bits. Erasure locations detected by check bits help channel erasure decoder to retrieve the original source encoded image. This scheme significantly outperforms recent techniques in terms of image quality for both watermarked and recovered image. Existing algorithm, reference bits are the source coded image. This data is derived from and then scattered over the whole image to overcome both tampering and waste problems.This algorithm is to embed a watermark into original image to protect it against tampering. It means that the watermark must be capable of both finding the tampered areas of the received image, and recovering the content of the original image in those zones. In order to achieve this goal, we keep nm most significant bits of each pixel unchanged, and use the remaining (nw) bits for the watermark embedding. For the purpose of image recovery, we compress the image using a source encoding algorithm, and embed the result as watermark. However, some of compressed image information might be lost because of image tampering; hence the com-pressed image bit stream must be channel coded to exhibit robustness against a certain level of tampering.

The existing algorithm is to embed a watermark into original image to protect it against tampering. It means that the watermark must be capable of both finding the tampered areas of the received image, and recovering the content of the original image in those zones. In order to achieve this goal, we keep *nm* most significant bits of each pixel unchanged, and use the remaining *nw* bits for the watermark embedding. For the purpose of image recovery, we compress the image using a source encoding algorithm, and embed the result as watermark. However, some of compressed image information might be lost because of image tampering; hence the compressed image bit stream must be channel coded to exhibit robustness against a certain level of tampering. In order to detect tampered blocks at the receiver, some check bits are generated from those parts of image which remain unchanged during watermark embedding procedure. These check bits are inserted as a part of total watermark. As a result, the least

documents to prevent counterfeiting. Consider the original image I represented by 8-bit gray-scale pixel values. These eight bits are divided into four parts: The most significant bits that will not change at the watermark embedding phase, check bits, source code bits, and channel code parity bits, denoted by nm, nh, ns and np respectively. Then m MSB of each pixel are remained unchanged during watermark embedding and will be used later for hash generation and image reconstruction. The remaining bits are used for the purpose of watermark embedding. Implementation of the set partitioning in hierarchical trees (SPIHT)image compression algorithm, as the source encoder. SPIHT is an embedded compression algorithm, i. e., one can extract an estimation of the original image by truncating its output in every desired rate. This property which fits our design of a general framework, together with the high compression gain when applied over the whole image, have been our main motivations to employ tampered blocks. The list of tampered blocks identifies erasure locations and helps the channel erasure decoder to find the compressed image bit stream despite the occurring erasure. Then source encoded image would be decoded and the estimation of the original image is recovered. The general description of our watermark insertion and image recovery procedures.

In order to detect tampered blocks at the receiver, some check bits are generated from those parts of image which remain unchanged during watermark embedding procedure. These check bits are inserted as a part of total watermark. As a result, the least significant nw bits (LSB) are comprised of both channel coded bits and check bits. Having tampered blocks known using the check bits, tampering can be modeled as an erasure error. Therefore, compressed bit stream is channel coded using a code capable of resistance against certain level of erasure. At the receiver, the check bits locate tampered blocks. The list of tampered blocks identifies erasure locations and helps the channel erasure decoder to find the compressed image bit stream despite the occurring erasure. Then source encoded image would be decoded and the estimation of the original image is recovered.

[1] **Drawbacks**

It has strong disadvantages for some applications

- It is complex
- It has poor energy compaction
- Energy compaction is the ability to pack the energy of the spatial sequence into as few frequency coefficients as possible ,this is very important for image compression , we represent the signal in the frequency domain if compaction is high we only have to transmit a few coefficients 2 if compaction is high, we only have to transmit a few coefficients instead of the whole set of pixels.

III. PROPOSED METHOD

[1] **Algorithm**

A **watermark** is an identifying image or pattern in paper that appears as various shades of lightness/darkness when viewed by transmitted light (or when viewed by reflected light, atop a dark background), caused by thickness or density variations in the paper.[1]

There are two main ways of producing watermarks in paper; the dandy roll process, and the more complex cylinder mould process. Watermarks vary greatly in their visibility; while some are obvious on casual inspection, others require some study to pick out. Various aids have been developed, such as watermark fluid that wets the paper without damaging

the SPIHT. Channel coding algorithm of rate is applied to the permuted compressed image bit stream.

Knowing the location of a tampered block at the receiver, all of its watermark bits are marked as erased. Therefore, wean integrate these lost bits into a few symbols by setting up the channel code over large fields. The other demand of our application is to implement a channel encoder and decoder that work on long blocks as input and output. In this case, the best performance of the channel code in terms of TTR is achieved, when the whole input bit stream is channel encoded using only a single block. Reed-Solomon (RS) codes can be implemented on the large fields, and automatically can be applied to a very long block of the symbols. Therefore, RS is our choice as the channel code. In the next Section, we show that the whole image can be channel encoded by only applying a single iteration of the channel code.

reconstructed image. The compressed image bit stream available at the output of the decider is passed through the source decoder after undergoing proper inverse permutation. The reconstructed image is made by replacing the tamper blocks by their corresponding blocks at the output of the source decoder. Obviously, the content of the received image in preserved blocks will be replaced with the corresponding information derived from the restored image.

[2] **Tamper proof and Image Recovery**

The received image which is probably tampered is decomposed into blocks of size. For each block, position bits are found, derived from shared secret key. Block bits are decomposed to nm MSB and nm watermark LSB per pixel, which results watermark bits. The watermark bit stream itself is decomposed into channel code bits. Position bits along with MSB are used to generate hash bits. The XOR of calculated hash bits and extracted check bits is recorded for each block. For unaltered blocks, this bit stream equals the random key used in the embedding phase. Therefore, comparing these results and spotting the different ones lead to locating the tampered blocks. The probability of missing a tampered block equals, which is almost zero for sufficiently large. After locating the tampered blocks, the Nc channel code bits are collected through the whole image.

Channel code bits are undergoing proper inverse permutation. Then, they are delivered as input to RS erasure decoder along with the erasure locations calculated from the list of tampered blocks. The compressed image bit stream available at the output of the decoder is passed through the source decoder after undergoing proper inverse permutation. The output of source decoder is the

the sake of watermark insertion, our algorithm is called *nw*-LSB. Block diagram of watermark embedding for 2-LSB algorithm is shown in Fig. 1. In this case, *nw*, *ns* , *n p* and *nh* are equal to 2, 1, 0.5 and 0.5, respectively.

To recover the tamper images, received image which is probably tampered is decomposed into blocks of size $B \times B$. For each block, position bits are found using $k2$, derived from shared secret key. Block bits are decomposed to *nm* MSB bits and *nw* watermark LSB bits per pixel (bpp), which results in $bm = nm \times B2$ MSB bits and $bw = nw \times B2$ watermark bits. The watermark bit stream itself is decomposed into $bh = nh \times B2$ check bits and $bc = nc \times B2$ channel code bits. *brc* position bits along

IV. BLOCK DIAGRAM

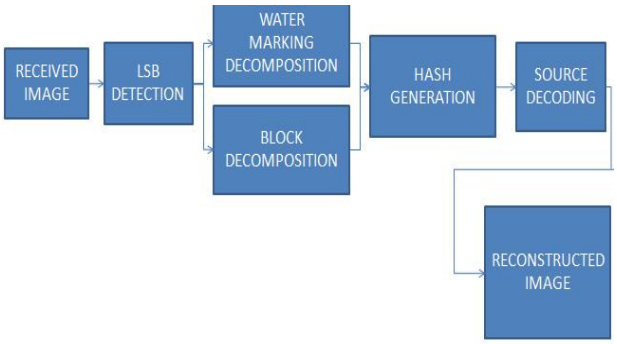
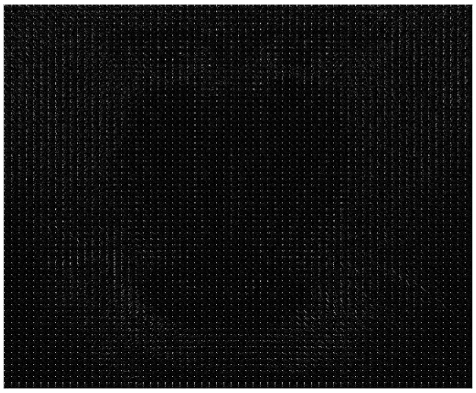


Fig1. The generic block diagram of watermarking embedding and tamper detection and image recovery.

Consider the original image *I* represented as 8-bit gray-scale pixel values. These eight bits are divided into four parts: The most significant bits (MSB) that will not change at the watermark embedding phase, check bits, source code bits, and channel code parity bits, denoted by *nm*, *nh*, *ns* and *n p*, respectively. The *nm* MSB bits of each pixel remain unchanged during watermark embedding and will be used later for hash generation and image reconstruction. The remaining bits are used for the purpose of watermark embedding. The original image is also divided into blocks of size $B \times B$, thus each block will host $bc = nc \times B2$ channel code bits. These *bc* bits originally belonged to some other blocks, whose rows and indices are turned into a binary stream of *brc* bits called position bits. These *brc* position bits along with $bm = nm \times B2$ MSB bits of each block are used as input to a hash generator algorithm (MD5 here), to produce $bh = nh \times B2$ hash bits. A random binary key of length *bh* fixed over the whole image is generated at the embedding phase. This key is *XOR*ed with hash bits to generate *bh* check bits. These *bh* check bits along with *bc* channel code bits of each block are spread over the block which results in replacing last $nw = nc + nh$ least significant bits of each pixel of the original image, where *nw* is the number of LSB per pixel used for watermark embedding. After having all blocks processed, watermarked image is produced.

To summarize, *nm* MSB of each pixel are preserved and $nw = 8 - nm$ LSB are replaced with watermark bits during embedding process. These *nw* bits consist of *ns* source code bits, *n p* channel code parity bits, and *nh* check bits. *nw* is not necessarily an integer. For instance, one may use two or three LSB bits in each block for watermark insertion alternatively. In this case, we have $nw = 2.5$. For the sake of simplicity, we assume integer *nw* (*nm*) hereafter. In the case that *nw* LSB of each pixel is used for



DOI: 10.23883/IJRTER.CONF.20190304.026.POKSD

with *bm* MSB bits are used to generate *bh* hash bits. The *XOR* of calculated hash bits and extracted check bits is recorded for each block. For unaltered blocks, this bit used in the embedding phase.

V. EXPERIMENTAL RESULTS

First, the Tampered image is protected against high-rate tampering by watermark generated. Next the image is processed for encryption and decryption. Password is generated for both encrypted and decrypted images. After that image is decrypted using decryption key. Following that image is also recovered using efficient source decoder.

(a) (b)

Fig 2.(a) orginal image, 2.(b) water mark image

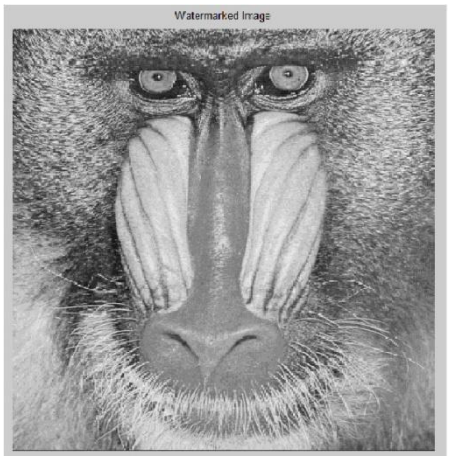


Fig 3: The watermarked image

Next the image is processed for encryption and decryption. Password is generated for both encrypted and decrypted images. After that image is decrypted using decryption key.

stream equals the random key

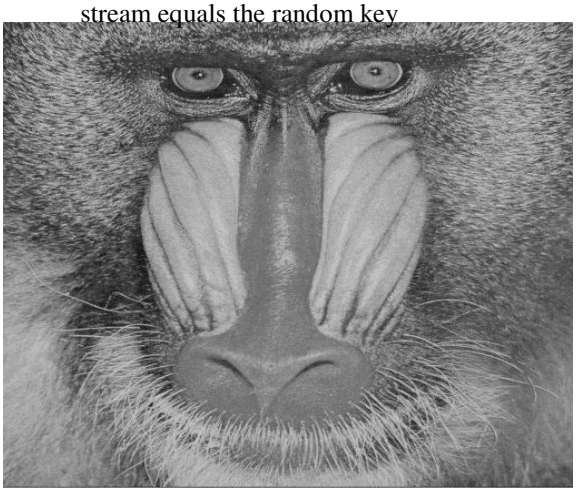


Fig 4.2: Decrypt The Original Message

To recovery the watermarking from the original image.

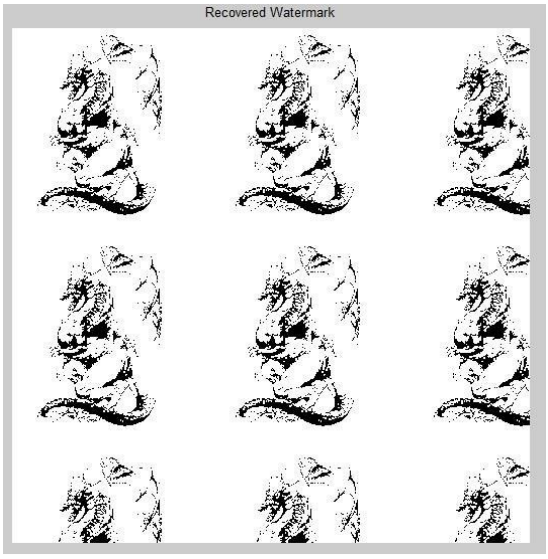


Fig 5: Recovered Watermark

VI. CONCLUSION

The original image is compressed using an efficient source en-coder (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.

ACKNOWLEDGMENT

The heading of the Acknowledgment section and the References section must be number. Causal Productions wishes to acknowledge and other contributors for developing and maintaining the IJREST.

REFERENCES

[1] S. Roy and Q. Sun, "Robust hash for detecting and localizing image tampering," in Proc. IEEE Int. Conf. Image Process. (ICIP), vol. 6. Sep./Oct. 2007, pp. VI-117–VI-120.

[2] M. Tagliasacchi, G. Valenzise, and S. Tubaro, "Hash-based identification of sparse image tampering," IEEE Trans. Image Process., vol. 18, no. 11, pp. 2491–2504, Nov. 2009.

[3] M. Wu and B. Liu, "Watermarking for image authentication," in Proc. Int. Conf. Image Process. (ICIP), vol. 2. 1998, pp. 437–441.

[4] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digitalto-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.

[5] Fridrich, "Image watermarking for tamper detection," in Proc. Int. Conf. Image Process. (ICIP), vol. 2. Oct. 1998, pp. 404–408.

[6] D. Kundur and D. Hatzinakos, "Digital watermarking for telltale tamper proofing and authentication," Proc. IEEE, vol. 87, no. 7, pp. 1167–1180, Jul. 1999.

[7] C.-S. Lu, S.-K. Huang, C.-J. Sze, and H.-Y. M. Liao, "Cocktail watermarking for digital image protection," IEEE Trans. Multimedia, vol. 2, no. 4, pp. 209–224, Dec. 2000.

[8] P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," IEEE Trans. Image Process., vol. 10, no. 10, pp. 1593–1601, Oct. 2001.

[9] M. U. Celik, G. Sharma, E. Saber, and A. M. Tekalp, "Hierarchical watermarking for secure image authentication with localization," IEEE Trans. Image Process., vol. 11, no. 6, pp.

585–595, Jun. 2002. S. Suthaharan, "Fragile image watermarking using a gradient image for improved localization and security," Pattern Recognit. Lett., vol. 25,no. 16, pp. 1893–1903, 2004.

[10] D. Zou, Y. Q. Shi, Z. Ni, and W. Su, "A semi-fragile lossless digital watermarking scheme based on integer wavelet transform," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 10, pp. 1294–1300, Oct. 2006.

[11] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99 -02, 1999.

[12] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1997.

[13] Swaminathan, M. Wu, and K. J. R. Liu, "Digital image forensics viaintrinsic fingerprints," IEEE Trans. Inf. Forensics Security, vol. 3, no. 1, pp. 101–117, Mar. 2008.

[14] J. Fridrich and M. Goljan, "Images with self-correcting capabilities," in Proc. Int. Conf. Image Process. (ICIP), vol. 3. 1999, pp. 792–796.

[15] J. Lee and C. S. Won, "Authentication and correction of digital watermarking images," Electron. Lett., vol. 35, no. 11, pp. 886–887, 1999.

[16] H.-J. He, J.-S. Zhang, and F. Chen, "Adjacent-block based statistical detection method for self-embedding watermarking techniques," Signal Process., vol. 89, no. 8, pp. 1557–1566, 2009.

[17] S.-H. Liu, H.-X. Yao, W. Gao, and Y.-L. Liu, "An image fragile watermark scheme based on chaotic image pattern and pixel-pairs," Appl. Math. Comput., vol. 185, no. 2, pp. 869–882, 2007.

[18] V. Mall, K. Bhatt, S. K. Mitra, and A. K. Roy, "Exposing structural tampering in digital images," in Proc. IEEE Int. Conf. Signal Process., Comput. Control (ISPCC), Mar. 2012, pp. 1–6.

[19] R. Chamlawi, A. Khan, and I. Usman, "Authentication and recovery of images using multiple watermarks," Comput. Elect. Eng., vol. 36, no. 3, pp. 578–584, 2010.

[20] C.-W. Yang and J.-J. Shen, "Recover the tampered image based on VQ indexing," Signal Process., vol. 90, no. 1, pp. 331–343, 2010.

[21] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "A secure and improved self-embedding algorithm to combat digital document forgery," Signal Process., vol. 89, no. 12, pp. 2324–2332, 2009.

[22] M. Wu and B. Liu, "Data hiding in binary image for authentication and annotation," IEEE Trans. Multimedia, vol. 6, no. 4, pp. 528–538, Aug. 2004.

[23] X. Zhang, S. Wang, Z. Qian, and G. Feng, "Reversible fragile watermarking for locating tampered blocks in JPEG images," Signal Process., vol. 90, no. 12, pp. 3026–3036, 2010.

[24] N. Wang and C.-H. Kim, "Tamper detection and self-recovery algorithm of color image based on robust embedding of dual visual watermarks using DWT-SVD," in Proc. 9th Int. Symp. Commun. Inf. Technol. (ISCIT), Sep. 2009, pp. 157–162.

[25] K.-C. Liu, "Colour image watermarking for tamper proofing and pattern based recovery," IET Image Process., vol. 6, no. 5, pp. 445–454, Jul. 2012.

