COVER SHEET

System Architecture Analysis of a Hybrid Watermarking Method

Chaw-Seng Woo[1], Jiang Du[1], Binh Pham[2], and Hamud Ali Abdulkadir[2]

1 Information Security Institute, Faculty of Information Technology
Queensland University of Technology, GPO Box 2434, Brisbane, QLD4001, Australia
cs.woo@student.qut.edu.au, j2.du@qut.edu.au

2 Faculty of Information Technology, Queensland University of Technology
GPO Box 2434, Brisbane, QLD4001, Australia
b.pham@qut.edu.au, h.abdulkadir@student.qut.edu.au

Abstract. A hybrid watermark that consists of a robust part and a fragile part can be used to serve multiple purposes. The robust part can protect copyright information, the fragile part can detect tampering, and their combination enables identification of attacks encountered. This paper analyses an overlap and a non-overlap implementation of the robust and fragile parts in a hybrid system. The difference between the two implementation methods lies in the robust and fragile watermarks embedding positions. Embedding capacity, computational costs, watermark robustness, and tamper detection localization of the two implementations are analyzed. In addition, optimization issues of block size in the hybrid system are discussed.

1 Introduction

To date, hybrid watermarking methods published are limited compared to other multimedia content watermarking methods. Some of the recent works are [1], [2], and [3]. A hybrid watermarking system that consists of a robust part and a fragile part can be used to serve multiple purposes. For example, robust watermarks are suitable for copyright protection because they remain intact with the protected content under various manipulative attacks. Fragile watermarks are good for tamper detection since it is sensitive to changes. The combination of robust and fragile watermarks offers some advantages over single watermarks. For example, it can identify copy attack and substitution attack [2]. In addition, watermark detection results of a hybrid system can be used to deduce whether a malicious tampering or a common image processing operation has taken place [1].

The performance factors of a watermarking method are mutually exclusive. For instance, increasing watermark robustness normally degrades its imperceptibility and limits its embedding capacity. To achieve a desirable balance among the performance factors, a designer must understand the influence of one factor on another. Therefore, it is important to evaluate the effects of system architecture on a hybrid system.

We analyzed the system architecture of a hybrid digital image watermarking system, and compared two implementation methods of its robust and fragile parts. The first method ensures that robust and fragile watermarks are embedded in non overlapping positions [2], and will be called .non-overlap. implementation. The second method overlaps both watermarks, and will be called .overlap. implementation.

The overlap implementation has the advantage of full embedding capacity and higher localization in tamper detection. However, the compromise in its robustness and computational cost need to be investigated. The comparison include watermark embedding capacity, computational cost, robustness of the robust part, and tamper detection effectiveness of the fragile part. In addition, the effects of block size on the hybrid system.s performance are also studied.

2 Overview of the Hybrid Watermarking System

The hybrid system chosen in our analysis [2] embeds a periodic robust watermark pattern in the discrete wavelet transform (DWT) domain. It uses robust estimation technique with superior performance to enable watermark detection. Thus, it is suitable for real life application where the cover image may not be available during watermark detection. It also has a state-of-the-art fragile part that embeds watermark blocks in the least significant bits (LSB) of pixels. The following two paragraphs describe the robust and fragile parts. Detail steps can be found in [2].

The robust part uses a self-reference method to recover from geometrical distortions. Firstly, the watermark message is encoded using an error correction code (ECC) for reliable decoding, and encrypted for confidentiality. Secondly, the message bits are spread in a symmetric pattern to cover the whole image size. This provides regularly-spaced peaks in geometrical re-synchronization for watermark detection.

Finally, the watermark is embedded in DWT domain for robustness. Instead of employing human visual system (HVS) masking for imperceptibility, we simplified it with constant energy embedding. To detect a watermark in an attacked image, it exploits the periodic peaks of magnitude spectrum for image re-synchronization. These steps are detailed in [4]. We applied thresholding on the magnitude spectrum to extract the peaks for simplicity. Fig. 1 (Left) depicts peaks obtained from the magnitude spectrum of the embedded watermark. Fig. 1 (Right) shows peaks extracted from a rotated stego image. Assuming local distortions are restricted by the acceptable image quality change, local and non-linear transformations can be recovered using the same approach at the local level. Details of such approaches are described in [5]. After that,

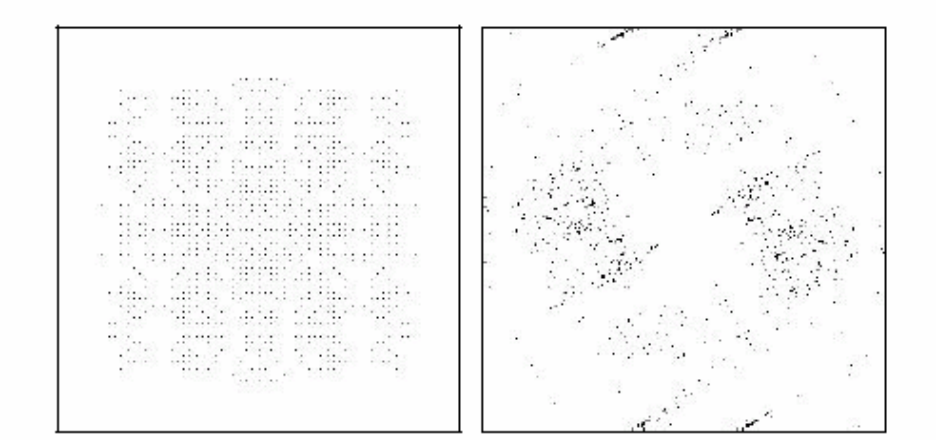

Fig. 1. (Left) Peaks obtained from the magnitude spectrum of the embedded watermark. (Right) Peaks extracted from a stego image with 30 degree rotation and auto-crop

a watermark estimation based on *Maximum a Posteriori* (MAP) probability is applied on the re-synchronized image. Then, a correlator detector is used in watermark decoding with a threshold value.

The fragile part uses a block-wise scheme to locate tampered regions. It computes a key-dependant hash value for overlapping blocks of an image and embeds the value into the LSB of pixels inside that block. By comparing the estimated signatures of the fragile blocks, tampered regions can be highlighted.

3 Non-overlap Implementation of the Hybrid System

In the non-overlap implementation, the embedding of its robust and fragile parts is performed simultaneously as described in Section 2 above. The robust positions do not overlap with the fragile positions within each block. Hence it was named .orthogonal . in [2]. The detection of watermarks in the non-overlap implementation is the same as its embedding part where the robust and fragile parts are processed independently. In our implementation, the robust blocks and fragile blocks are chosen to have the same size.

The block-wise hashing of fragile part takes the current block with its eight neighboring blocks as input. The computed hash code is then embedded into the current block. This provides local contextual dependency. However, this approach not only detects modifications within the block but also modifications in its neighboring blocks. Compensation steps mentioned in [2] are not implemented at this stage.

4 Overlap Implementation of the Hybrid System

In this implementation, the robust part is embedded prior to the fragile part. By definition, the robust part must survive distortions caused by the fragile part. Therefore, we proposed to embed the fragile part in all positions, overwriting the LSBs of the robust stego image. As a result, both the robust and fragile parts can be embedded into all positions, achieving maximum watermarking capacity. Furthermore, it gives the highest possible localization for tamper detection. In addition, it also reduces computation by eliminating position tracking of the robust and fragile parts.

The watermark detection in the overlap implementation is similar to those in the non-overlap implementation. However, all pixel locations in overlap implementation are processed because both the robust and fragile parts are embedded in every position. This requires an examination of the compromise in computational cost.

5 Experimental Results Analysis

To compare the overlap and non-overlap implementations in a hybrid system, the parameters listed in Table 1 were applied. Two test images with 256 gray levels were used. They are *Lena* and *Cameraman* of 256·256 pixels. A set of general image manipulation operations listed in Table 2 was used to evaluate the performance of the robust watermark. Three types of attacks were used in fragile watermark evaluation: local tampering modify the pixel values of a small area, copy attack copies a small region from an image and pastes it onto the same image whereas collage attack pastes it onto another image.

Table 1. Parameter values for non-overlap and overlap implementations

Parameter	Non-overlap implementation	Overlap implementation
Block size $t_1 \times t_2$	16×16 = 256 pixels	16×16 = 256 pixels
Robust positions	178	256
Fragile positions	42	256
Empty positions	36	0

Table 2. Attacks for robustness evaluation

Attack	Descriptions
Rotation followed by cropping	Rotate 30 degree with auto-crop
Scaling followed by shearing	Uniform scaling factor 0.98; Shear 2%
JPEG compression	Quality factors at 50%
Gaussian noise insertion	Zero mean; variance 0.02
Contrast adjustment	Gamma value 0.6
Median filtering	2×2 smoothening kernel

5.1 Analysis of Robust Part Results

Using block size of 32·32, the stego images of non-overlap and overlap implementations give PSNR of 37.02dB and 37.29dB respectively. This indicates image qualities of both implementations are very close. Such observation can be explained by the small difference of un-marked positions between the two implementations, i.e. 256 . 178 = 78 bits ≈ 30.47% in each block.

Regularly-spaced peaks can be observed after thresholding the magnitude spectrum of the non-attacked stego images. These patterns are very similar to those of the embedded watermark. Therefore, the peak patterns can be used in geometrical resynchronization, and the robust watermark message can be extracted successfully in both non-overlap and overlap implementations. The non-attacked stego image of overlap implementation gives better peak patterns compared to those of the nonoverlap implementation because it has full embedding capacity. To improve the robustness of the implementation modes, the watermark embedding energy can be increased to warrant better peak patterns, but it will degrade the visual qualities. In the non-overlap implementation, compromise must be made between the densities of robust part and fragile part. Increasing fragile watermark positions to enhance its localization in tamper detection will reduce those of the robust watermark, thus degrade its robustness.

To evaluate the robustness of the watermark, the attacks listed in Table 2 were carried out. With the obvious axes in the peak patterns, distortions can be compensated with a re-synchronization step to enable successful watermark detection. This is done using Hough transform to estimate the rotation angle, and Maximum Likelihood (ML) to estimate peak periods. Details of the recovery steps can be found in [4].

There are two items worth description here. Firstly, the estimation outcome of Hough transform may deviate one degree. Therefore, brute force search need to be applied in finding the correct parallel lines for period estimation. Secondly, a predefined period range must be specified in the estimation of period between peaks as mentioned in [2]. Overall, both of the implementation modes are equally robust to the attacks. The robust watermark was detected in both non-overlap and overlap implementations after re-synchronization.

Computational costs for the implementation methods are listed in Table 3 for block size 16·16 pixels. The overlap implementation requires more processing time because it embeds robust watermark into every pixel in each block whereas the nonoverlap implementation only need to process about 70% of the pixels in each block.
The savings of not tracking robust and fragile watermark positions in an overlap implementation does not offset the overall computational costs.

Table 3. Comparison of hybrid watermarking time (seconds)

	Non-overlap implementation	Overlap implementation
Robust embedding	3.10	3.12
Robust detection	0.49	0.50
Fragile embedding	5.70	6.35
Fragile detection	5.60	6.34

5.2 Analysis of Fragile Part Results

The fragile watermark evaluation for both implementation modes were done using local tampering, copy attack, and collage attack. Local tampering was easily detected and highlighted as shown in Fig. 2.
A copy attack on *Cameraman* stego image and its fragile watermark detection results are given in Fig. 3. In the test, a dark color region is copied and pasted onto another region on the cloth of the same image. A similar operation is performed on a textured region, i.e. the lawn. The results of a collage attack involving *Lena* and *Cameraman* stego images gave similar results. The fragile watermark in both implementations highlighted tampered regions correctly.

Since the overlap implementation employs full capacity embedding, it was able to highlight modifications at each pixel. Conversely, the non-overlap implementation embedded its fragile watermark in about 30% pixels of each block. As a result, it was not as accurate as the overlap implementation.

Fig. 2. (Left) Tampered Lena stego image. (Right) Tampered regions highlighted by fragile watermark detection

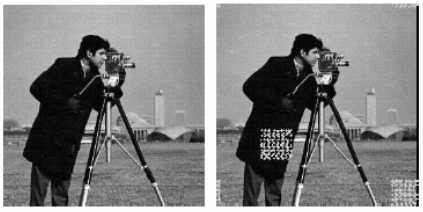

Fig. 3. (Left) Copy-attacked Cameraman stego image. (Right) Tampered regions highlighted by fragile watermark detection

Besides the three types of attacks above, the effects of block size on the fragile watermark are also examined on the non-overlap implementation. As tabulated in Table 4, larger block size requires less processing time. This is due to the convolution operation in hashing neighboring blocks. Also, large block size allows high security with long signatures. On the other hand, the smaller the block size, the more blocks are involved. Thus, the more computing cycles are needed. Nevertheless, smaller block size offers better localization in tamper detection.

Table 4. Effects of block size on fragile watermarking time (seconds)

Block size	Embed time	Detect time
4×4	15.60	15.30
8×8	7.53	7.51
16×16	5.70	5.60
32×32	5.00	4.90

6 Conclusions

We have analyzed and compared the overlap and non-overlap implementations of a hybrid system and have found that both implementations generally produce similar results. This is due to the fact that the robust part in the overlap implementation resisted distortions introduced by the fragile part. Although the overlap implementation reduces computational by not tracking robust and fragile watermark positions, its embedding time and detection time is slightly longer compared to those of the nonoverlap implementation. This is caused by the extra processing load of embedding fragile and robust watermarks in all pixel positions. The overlap implementation offers higher watermark capacity for both the robust and fragile watermarks compared to the non-overlap implementation. Hence, the overlap implementation gives better peak patterns than the non-overlap implementation in robust watermark extraction.

Due to the same reason, the overlap implementation has better localization in tamper detection compared to the non-overlap implementation. Finally, a balance between tamper detection localization and computational cost must be determined when selecting an optimum block size for both implementation modes. In summary, the overlap implementation can meet high integrity requirements in digital contents while the non-overlap implementation is suitable for commercial applications where processing speed is a preference.

Acknowledgement

This work is supported by the Strategic Collaborative Grant on Digital Rights Management (DRM) awarded by Queensland University of Technology (QUT), Australia.

References

1. Jiri Fridrich, .A hybrid watermark for tamper detection in digital images., Proceedings of the Fifth International Symposium on Signal Processing and Its Applications (ISSPA '99), 22-25 Aug. 1999. 1 (1999) 301.304
2. F. Deguillaume, S. Voloshynovskiy, T. Pun, .Secure hybrid robust watermarking resistant against tampering and copy attack., Signal Processing, Elsevier. 83 (2003) 2133.2170
3. Chaw-Seng Woo, Jiang Du, Binh Pham, .Multiple Watermark Method for Privacy Control and Tamper Detection in Medical Images., APRS Workshop on Digital Image Computing (WDIC2005), Brisbane, Australia, 21 Feb. 2005 (2005) 43.48
4. F. Deguillaume, S. Voloshynovskiy, T. Pun, .Method for the estimation and recovering of general affine transforms in digital watermarking applications., IS& T/SPIE.s 14th Annual Symposium, Electronic Imaging 2002: Security and Watermarking of Multimedia Contents IV, San Jose, CA, USA, 20.25 January 2002. 4675 (2002) 313.322
5. S. Voloshynovskiy, F. Deguillaume, T. Pun, .Multibit digital watermarking robust against

local nonlinear geometrical distortions., IEEE International Conference on Image Processing (ICIP2001), Thessaloniki, Greece, October 2001 (2001) 999.1002