# A Novel Image Restoration Scheme Based on Structured Side Information and Its Application to Image Watermarking[☆]

Hui Wang, Anthony TS Ho*, Shujun Li*

*Department of Computing, University of Surrey, Guildford, Surrey, GU2 7XH, UK*

**Abstract**

This paper presents a new image restoration method based on a linear optimization model which restores part of the image from structured side information (SSI). The SSI can be transmitted to the receiver or embedded into the image itself by digital watermarking technique. In this paper we focus on a special type of SSI for digital watermarking where the SSI is composed of mean values of $4 \times 4$ image blocks which can be used to restore manipulated blocks. Different from existing image restoration methods for similar types of SSI, the proposed method minimizes image discontinuity according to a relaxed definition of smoothness based on a $3 \times 3$ averaging filter of four adjacent pixel value differences, and the objective function of the optimization model has a second regularization term corresponding to a 2nd-order smoothness criterion. Our experiments on 100 test images showed that given complete information of the SSI, the proposed image restoration technique can outperform the state-of-the-art model based on a simple linear optimization model by around 2 dB in terms of average Peak Signal-to-Noise Ratio (PSNR) value and around 0.04 in terms of Structural Similarity Index (SSIM) value. We also tested the robustness of the image restoration method when it is applied to a self-restoration watermarking scheme and the experimental results showed that it is moderately robust to errors in SSI (which is embedded as a watermark) caused by JPEG compression (the average PSNR value remains above 16.5 dB even when the JPEG QF is 50), additive Gaussian white noises (the average PSNR value is approximately 18.4 dB when the noise variance $\sigma^2$ is 10) and image rescaling assuming the original image size known at the receiver side (*e.g.* the average PSNR value is approximately 19.6 dB when the scaling ratio is 1.4).

*Keywords:* Image Restoration, Digital Watermarking, Structured Side Information, Linear Programming, Optimization

## 1. Introduction

When it comes to image restoration, there are different scenarios with different problems to be solved. The classical image restoration problem refers to the operation of mitigating the visual quality degradation caused by some image processing steps [1]. Corruption may come in many forms like blur, noise, geometrical deformation, and camera misfocus. There are many existing algorithms for handling this type of image restoration problems, often based on an optimization model linked to some known information or model of the distortions.

There is another image restoration scenario where the image content is partly lost. Digital image inpainting is one of the image restoration techniques for handling this scenario. Many image inpainting approaches have been proposed in the literature [2, 3, 4, 5]. The aim of image inpainting is often to restore a natural-looking image, but the authenticity and accuracy of the image content in the missing region are not necessarily guaranteed. There is normally no available side information about the missing region.

The above two image restoration scenarios have been well studied in the literature and many different ap-

[☆] This is the author's version of a work that was accepted to *Signal Processing: Image Communication* in 2014. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version has been published in *Signal Processing: Image Communication*, vol. 29, no. 7, pp. 773-787, 2014, Elsevier. DOI: 10.1016/j.image.2014.05.001.

*Corresponding authors.

*Email addresses:* h.wang@surrey.ac.uk (Hui Wang), a.ho@surrey.ac.uk (Anthony TS Ho), shujun.li@surrey.ac.uk (Shujun Li)

*URL:* http://www.hooklee.com (Shujun Li)

proaches have been proposed. Many approaches are built upon an energy minimization model involving a smoothness criterion and a regularization (data energy) term which counts some known statistical information about the signal and/or the distortion.

Yet another class of image restoration problems are those with deterministic structured side information (SSI) of the missing region. By structured, we mean some well defined side information about missing pixels is available somewhere. One typical problem is Discrete Cosine Transform (DCT) coefficients restoration where one or more DCT coefficients of each block are missing in DCT-transformed images, so they have to be restored from other available DCT coefficients. A special case of this problem is DC coefficients estimation from AC coefficients, and has been studied as early as in 1980s by Cham and Clarke [6] and new methods have also been proposed very recently [7, 8] which can find applications in cryptanalysis of selective multimedia encryption. The state of the art approach of solving this problem is to use a linear optimization model where the objective is to minimize the total discontinuity between all pixel pairs in the image (*i.e.* one form of total variation minimization). Compared with the traditional image restoration scenarios, in these models there are more additional constraints representing some SSI which cannot be easily incorporated into the objective function and could potentially make the problem harder to solve.

Th SSI based image restoration problems can also find applications in digital watermarking systems when the capability of recovering manipulated regions of a watermarked image is required (*e.g.* for forensic investigation purposes). In these applications, the SSI can be transmitted to the receiver or embedded into the image itself, so it is possible to choose the structure of the side information. In this paper, we focus on the application scenario where side information with a special structure is self-embedded as a digital watermark to assist image restoration at the receiver's end. In this application, the self-restoration watermark will work with an authentication watermark to localize manipulated regions that need restoring. In the following we briefly overview some related work on self-restoration watermarking.

Ho *et al.* proposed a semi-fragile authentication watermarking algorithm with self-restoration capability using the Pinned Sine Transform (PST) in [9]. The tamper detection accuracy rate of this scheme was shown higher than 98% even with light non-malicious image processing operations. The bits as restoration watermark were generated from the image coefficients after PST, and embedded into the least significant bits (LSBs) plane of the original image. However, the LSB based self-restoration watermark is fragile and could be easily distorted. To reduce the size of the restoration watermark and to improve the robustness, Region of Interest (ROI) self-restoration schemes were proposed in [10, 11]. The two restoration schemes were shown more robust against JPEG compression. In addition, the quality of the restored image in ROI was improved compared with other Non-ROI restoration methods. However the size of the ROI is limited if any region of the image should be protected for potential manipulation.

If any region of the image needs protecting, much more information is required to be hidden in the restoration watermark. On the other hand, semi-fragile watermarking requires more robustness against common signal processing operations, which will normally limit the embedding capacity. How to balance the requirements for both higher capacity and more robustness is an open research question. Lin and Lin [12] introduced a semi-fragile watermarking scheme with self-restoration ability based on $(t,n)$ secret sharing and Reed-Solomon (RS) code. The restoration watermark is generated from the four lowest DCT coefficients and encoded as an RS code. This watermarking scheme was shown moderately robust to image processing operations including JPEG compression, noise and brightness adjustment. One major issue of this scheme is that the percentage of the tampered region is limited to around 16.66%. Phadikar *et al.* [13] proposed a semi-fragile watermarking scheme for image restoration based on halftoning. The restoration watermark is generated by halftoning the down-sampled image, and then embedded in Low-High Level 1 (LH1) and High-Low Level 1 (HL1) subbands of Discrete Wavelet Transform (DWT) coefficients by the Quantization Index Modulation (QIM) method. Experiments showed that the tampered region could be restored if its size is less than 40% of the whole image. Wang *et al.* [14] proposed a semi-fragile self-restoration watermarking scheme based on linear regression. In this scheme, the mean value of each $4 \times 4$ block is used as the restoration watermark. After a tampered $8 \times 8$ block is identified, its four lowest DCT coefficients are restored from linear combinations of mean values of its four $4 \times 4$ sub-blocks, where the weights of linear combinations are obtained via linear regression. The algorithm was shown moderately robust to JPEG compression and the percentage of the tampered region can go up to 50%.

In this paper, we propose an enhanced image self-restoration method based on a linear optimization model with mean values of $4 \times 4$ blocks as SSI (*i.e.* the self-restoration information used in the semi-fragile watermarking scheme proposed in [14]). Our experimental results on 100 test images showed that the proposed restoration method can recover the original image with better overall visual quality than the original linear regression based method and the simpler linear model in [8]. The image quality is assessed by both objective metrics (via two image quality assessment (IQA) metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [15]) and subjective means (by manual inspection of the authors).

The rest of the paper is organized as follows. In Section 2, a detailed description of our proposed restoration method is given. Then, the self-restoration watermarking system, from which the structured side information comes and to which the proposed restoration model is applied, is briefly introduced in Section 3. The experimental results of image restoration technique based on 100% correct SSI (*i.e.* ideal watermarking or a side channel for transmitting SSI) are reported and analyzed in detail in Section 4. Then the experimental results on watermarking-related performance of a self-restoration watermarking system equipped with the proposed image restoration method, especially the robustness of the image restoration performance against some common image processing operations, are exhibited and analyzed in detail in Section 5. Finally, Section 6 concludes the paper with future work.

## 2. Proposed Image Restoration Method

In this section we describe our proposed image restoration method when the SSI is composed of mean values of 4×4 blocks. For this particular type of SSI, the existing model for solving similar problems proposed in [8] seems to be a good choice but our experiments showed that this model does not work as well as expected – with high probability a restored $4 \times 4$ block is a uniform block whose pixel values are all equal to the mean value, which is the most trivial solution. The problem of the existing model revealed that a different model is required to handle this special type of SSI. Our proposed model is derived from the above simple model by making two major changes: 1) the discontinuity criterion is changed from the simple difference of each adjacent pixel pair to the result of averaging four adjacent pixel value differences centered in a $3 \times 3$ window; 2) an additional regularization term is added to the objective function to overcome side effects introduced by the generalized discontinuity criterion. In the remaining part of this section, we first explain how the model in [8] can be generalized to handle SSI especially this special type of SSI, highlighting problems to be solved. Then, we focus on our proposed new model and explain why the two introduced changes can help to improve the image restoration performance.

### 2.1. Original Model with SSI

The model in [8] is based on a well-known property of most natural images: the difference between adjacent pixels follows a Laplacian distribution with a zero mean and a small variance. Based on this property, the objective of the optimization model is set to minimize the sum of absolute values of differences between all adjacent pixel pairs. For the Laplacian distribution $f(z) = \frac{1}{\sigma\sqrt{2}} e^{-\frac{\sqrt{2}}{\sigma}|z|}$, the maximum likelihood estimate (MLE) of the standard deviation $\sigma$ is $\frac{1}{S}\sum_{i=1}^{S}|z_i|$ when there are $S$ observations of the distribution [16]. Therefore, minimizing the sum of absolute values of adjacent pixel differences is equivalent to minimizing the MLE of the underlying Laplacian distribution's standard deviation $\sigma$. This objective has a clear physical meaning of minimizing the global discontinuity (or maximizing global smoothness) in the image, and it can also be explained as a variant of the total variation (TV) minimization followed by many image restoration/enhancement methods. While this model was proposed to study a particular problem of recovering DCT coefficients in [8], it is actually a universal model and can be used to solve many problems with SSI where the side information is represented as a group of constraints. This model has been proved to perform well to restore missing DCT coefficients from known ones in [8]. The model can be described mathematically as follows:

$$\begin{aligned} \min \quad & \sum_{(i,j) \text{ and } (i',j') \text{ are adjacent}} |x_{i,j} - x_{i',j'}| \\ \text{s.t.} \quad & x_{i,j} = x^*_{i,j}, \text{ if } x_{i,j} \text{ is known}, \\ & x_{\min} < x_{i,j} < x_{\max}, \text{ if } x_{i,j} \text{ is unknown}, \\ & A \leqslant f_{\text{SSI}}(\{x_{i,j}\}) \leqslant B, \end{aligned} \qquad (1)$$

where $x_{i,j}$ and $x_{i',j'}$ are two adjacent pixels, $x^*_{i,j}$ is a known pixel value, $x_{\min}$ and $x_{\max}$ are the lower and upper bounds of unknown pixel values, and the last group of constraints represents the SSI related to a set of pixel values $\{x_{i,j}\}$. For the DCT coefficients recovery problem studied in [8], the SSI is the relationship between pixel values and DCT coefficients. In this paper we focus on a different type of SSI, for which the last group of constraints becomes

$$\forall k, \frac{1}{16} \sum_{(i,j) \in B_k} x_{i,j} = m_k, \qquad (2)$$

where $B_k$ is the $k$-th $4 \times 4$ block in the image and $m_k$ is the known mean value of $B_k$.

The above model may look difficult to solve due to the involvement of the nonlinear function $|\cdot|$, but it can be easily linearized by introducing auxiliary variables $h_{i,j,i',j'}$ and two more linear constraints:

$$x_{i,j} - x_{i',j'} \leqslant h_{i,j,i',j'}, \tag{3}$$

$$x_{i',j'} - x_{i,j} \leqslant h_{i,j,i',j'}, \tag{4}$$

and then change the objective to

$$\min \sum_{(i,j) \text{ and } (i',j') \text{ are adjacent}} h_{i,j,i',j'}. \tag{5}$$

By making the above changes, the linearized model will have the same optimal solution as the nonlinear model (1) because each $h_{i,j,i',j'}$ will be tight at $|x_{i,j} - x_{i',j'}|$ of the optimal solution of the model Eq. (1). This can be explained by the fact that Eqs. (3) and (4) are equivalent to $0 \leqslant \max(x_{i,j} - x_{i',j'}, x_{i',j'} - x_{i,j}) = |x_{i,j} - x_{i',j'}| \leqslant h_{i,j,i',j'}$. Since the objective is to minimize the sum of $h_{i,j,i',j'}$, the optimal solution will be achieved at $h_{i,j,i',j'}$'s lower bound which is $|x_{i,j} - x_{i',j'}|$.

### 2.2. *Our Proposed New Model*

While the above model performs very well to recover missing DCT coefficients as reported in [8], our experimental results revealed that it does not work equivalently well for the SSI shown in Eq. (2): it tends to produce more uniform $4 \times 4$ block whose pixel values are all close to the mean $m_k$. While this does help to reduce the adjacent pixel differences within each $4 \times 4$ block, it brings blocking artifacts around block boundaries. An example about recovering the 10% right bottom corner of the test image "Lenna" can be found in Fig. 1(e) in which the blocking artifact is clearly visible along $4 \times 4$ block boundaries.

To overcome the problem of the model (1), we will need to modify the model so that it can look for solutions in a larger solution space. To achieve this, we can either relax the objective or remove some constraint(s). Since no any constraint is redundant, we propose to relax the objective by generalizing the smoothness criterion from the simple absolute difference of two adjacent pixel values to the result of a 3×3 averaging filter of four neighbouring adjacent pixel value differences. Mathematically, for each pixel $x(i, j)$ we define a generalized smoothness term $f(x_{i,j})$ as follows:[1]

$$f(x_{i,j}) = \sum_{d_i \in \{-1,1\}} \sum_{d_j \in \{-1,1\}} (x_{i,j} - x_{i+d_i,j+d_j}), \tag{6}$$

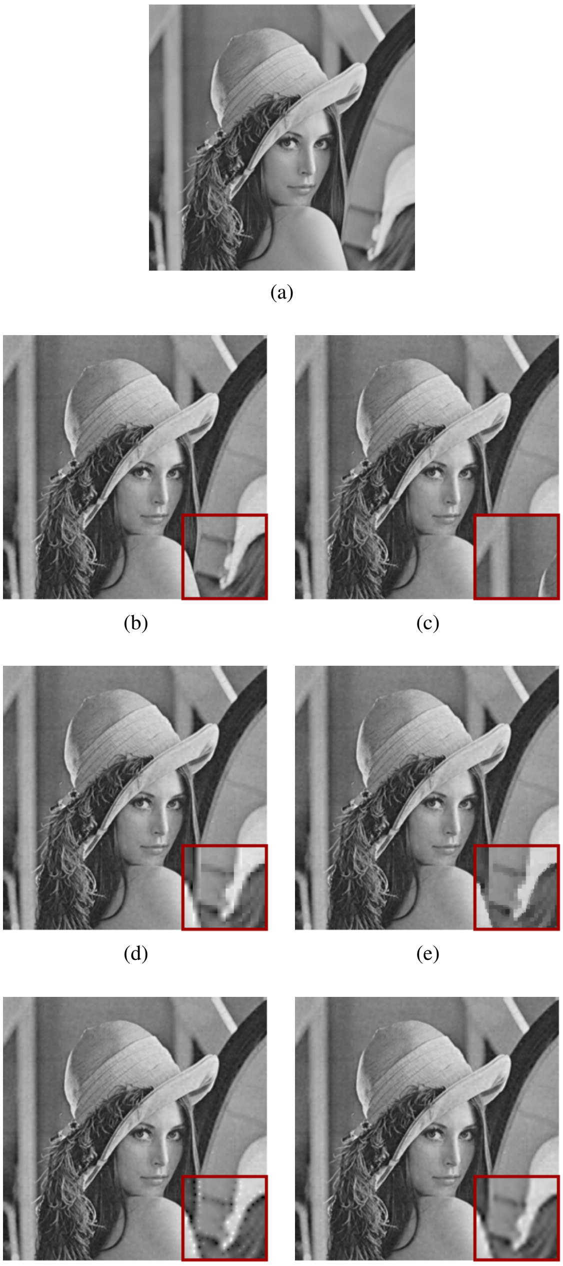


Figure 1: Experimental results of the three restoration methods on the test image "Lenna": (a) the original image; (b) the watermarked image; (c) the watermarked image after its 10% right bottom corner was manipulated; (d) the image recovered by the linear regression based model in [14]; (e) the image recovered by the model (1); (f) the image recovered by applying the relaxed smoothness criterion to the objective function of the model (1); (g) the image recovered by the proposed model (9).

which changes the original smoothness term $x_{i,j} - x_{i',j'}$ (a single pixel value difference) to a sum of four pixel

[1] It is possible to define $f(x_{i,j})$ as a weighted sum and the weights can be made locally content-adaptive. We leave this for future study.

value differences. Considering the fact that

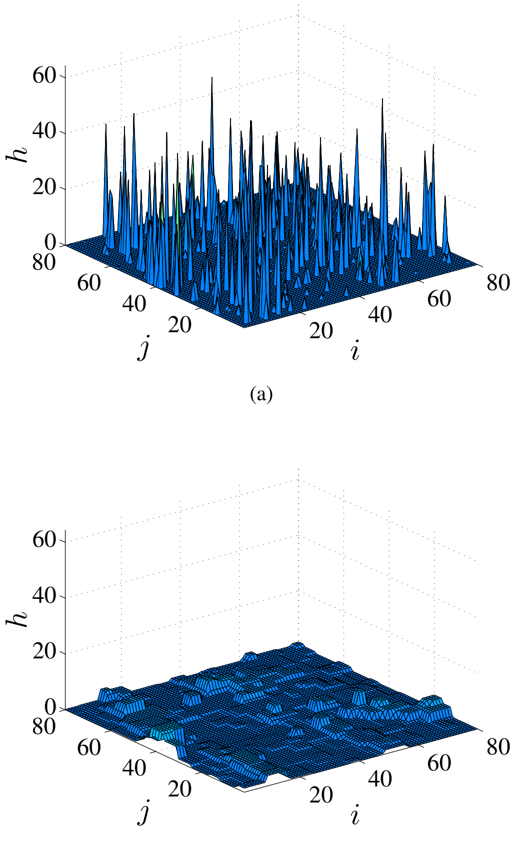
$$|f(x_{i,j})| \leqslant \sum_{d_i \in \{-1,1\}} \sum_{d_j \in \{-1,1\}} |x_{i,j} - x_{i+d_i,j+d_j}|, \qquad (7)$$

the new smoothness criterion does relax the objective to allow *large* adjacent pixel value differences. In other words, when $f(x_{i,j})$ as a sum of four terms has a small amplitude, each term inside may have a either small or large one. This enlarges the solution space by allowing local structures to appear in each $3 \times 3$ window if we change the model objective function to

$$\min \sum_{i,j} |f(x_{i,j})|. \qquad (8)$$

Our experiments showed that making the above change to the model (1) does help to reduce blocking artifacts, but it also introduces some sparse but large local changes. An example of the recovered image is shown in Fig. 1(f) in which some white and black spots are clearly visible. Figure 2(a) shows the recovered $\{h_{i,j}\}$ (i.e., $\{|f(x_{i,j})|\}$) with the sparse but rather high peaks here and there in the recovered image region. The side effect suggests that the relaxed smoothness criterion *alone* is not sufficient to guarantee good visual quality. Since the side effect is about the unwanted sharp local changes of $\{h_{i,j}\}$, we proposed to apply the same relaxed smoothness criterion $f(\cdot)$ to each $h_{i,j}$ to get a regularization term $h'_{i,j}$ which is added to the objective function in order to make the distribution of $\{h_{i,j}\}$ more uniform. This has been proved effective as shown in Fig. 2(b) from which one can see that all the values of the recovered $\{h_{i,j}\}$ become much smoother after the regularization term is added to the objective function. The maximum value of $\{h_{i,j}\}$ drops from more than 64 to below 8. As a consequence, the overall visual quality of the recovered image is improved (see Fig. 1(g)).

With the relaxed smoothness criterion and the additional regularization term, our proposed optimization model becomes



(a)

(b)

Figure 2: The 3-D views of recovered $\{h_{i,j}\}$ variables of the manipulated region of the test image "Lenna" (a) when the relaxed smoothness criterion is used in the new objective function and (b) after the second regularization term $h'_{i,j}$ is added.

$$\begin{aligned}
\min \quad & \sum_{i,j} \Big( (1-\lambda) \cdot h_{i,j} + \lambda \cdot h'_{i,j} \Big) \\
\text{s.t.} \quad & x_{i,j} = x^*_{i,j}, \text{ if } x_{i,j} \text{ is known}, \\
& x_{\min} \leqslant x_{i,j} \leqslant x_{\max}, \text{ if } x_{i,j} \text{ is unknown}, \\
& \tfrac{1}{16} \sum_{(i,j) \in B_k} x_{i,j} = m_k, \\
& f(x_{i,j}) \leqslant h_{i,j}, \\
& -f(x_{i,j}) \leqslant h_{i,j}, \\
& f(h_{i,j}) \leqslant h'_{i,j}, \\
& -f(h_{i,j}) \leqslant h'_{i,j}.
\end{aligned} \qquad (9)$$

where $f(\cdot)$ is the new smoothness criterion function defined by Eq. (6). In the above model the objective function is a weighted sum of two terms $h_{i,j}$ and $h'_{i,j}$ where the weights are $(1 - \lambda)$ and $\lambda$, respectively and $\lambda$ is a value in the range of $[0, 1]$. Our experimental results revealed that there is an optimal value of $\lambda$ which is around 0.5 (meaning equal contributions of both terms to the objective). More further analysis and discussions on this issue can be found in Sec. 4.1. The first three groups of constraints are the same as those in the model (1), and the last two groups of constraints are about the two auxiliary variables $h_{i,j}$ and $h'_{i,j}$ for linearizing the actual terms $|f(x_{i,j})|$ and $|f(h_{i,j})|$.

## 3. Watermarking System

Since the structured side information comes from the self-restoration watermarking scheme proposed in [14], we demonstrate the performance of the proposed restoration model working with the watermarking scheme. In this section we briefly explain how this watermarking scheme works. We also made some changes to the original scheme to improve its performance. All the changes applied are independent of the image restoration phase and have no direct impact on the image restoration step. However, they could contribute indirectly to the image restoration results by providing more accurate SSI.

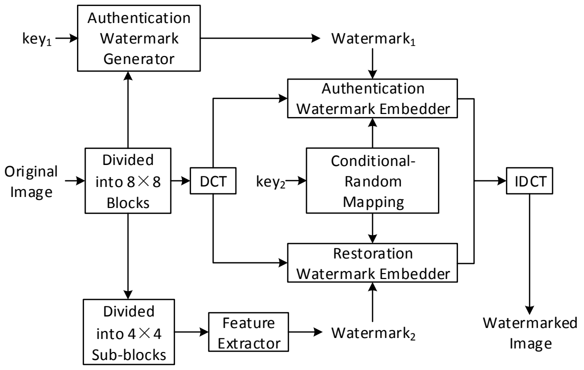

key1
Authentication Watermark Generator
Watermark1
Authentication Watermark Embedder
Original Image
Divided into 8×8 Blocks
DCT
key2
Conditional-Random Mapping
IDCT
Restoration Watermark Embedder
Divided into 4×4 Sub-blocks
Feature Extractor
Watermark2
Watermarked Image

Figure 3: The process of watermark embedding.

As with all other digital watermarking systems, the semi-fragile algorithm contains two parts: the sender side for watermark embedding, and the receiver side for image authentication and restoration using extracted watermarks. In the watermark embedding process as shown in Fig. 3, two types of watermarks are generated for each 8×8 block: a 6-bit authentication watermark (a pseudo-random number linked to the location of the block which are controlled by $key_1$) and four restoration watermarks (mean values of four 4×4 sub-blocks of the 8×8 block, normalized to be in the range $[0, th \times 255]$ where $th = 0.13$). Then the authentication watermark is embedded bitwise in 6 selected low-medium frequency DCT coefficients of a *different* 8×8 block using a one-to-one mapping linking two $8 \times 8$ blocks together under the control of $key_2$. The standard binary QIM method is applied as embedding method for each selected DCT coefficient. Each of the four restoration watermarks is embedded in one selected mid-frequency DCT coefficient of the same corresponding 8×8 block using the same mapping function under the control of $key_2$. The block mapping can be represented by $g : \mathbb{A} \to \mathbb{A}$ where $\mathbb{A} = \{1, \ldots., M\} \times \{1, \ldots., N\}$ for an image of size $M \times N$. Given such a mapping, the restoration watermark of Block $(i, j)$ is embedded in Block $(i', j') = g(i, j)$. The actual embedding process works as follows. Let $x$ denote the selected DCT coefficient for embedding and $w$ denote the restoration watermark, calculate $r = \left\lfloor \frac{x}{T_2} + \frac{1}{2} \right\rfloor$, then the DCT coefficient is changed to

$$y = \begin{cases} w + rT_2 - \frac{T_2}{2}, & \text{if } r \text{ is odd,} \\ (T_2 - w) + rT_2 - \frac{T_2}{2}, & \text{if } r \text{ is even,} \end{cases} \tag{10}$$

where $T_2 > 0$ is the quantization step size. After both watermarks are embedded, inverse DCT is applied to each block and the watermarked image is obtained.

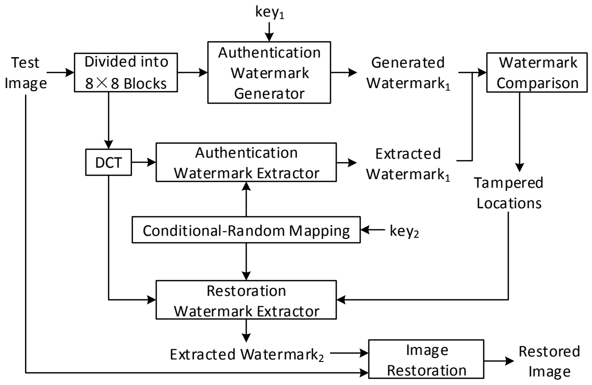


Figure 4: The process of image content authentication and self-restoration.

The image authentication and self-restoration process is illustrated in Fig. 4. The image authentication is firstly achieved by comparing the re-calculated watermark and the authentication watermark extracted. If the two watermarks do not match with each other (in terms of the number of different bits which should be greater than one), the block is pre-marked as possibly manipulated. After all the blocks in the whole image are pre-marked, for each block $(i, j)$ pre-marked as "possibly manipulated", we check the pre-mark of the block $g^{-1}(i, j)$. If $g^{-1}(i, j)$ is also marked as "possibly manipulated", we keep block $(i, j)$'s mark as it is; otherwise block $(i, j)$'s mark is changed to "not manipulated".[2] At the end of the authentication process, we apply the mathematical morphology area opening operation twice to remove isolated blocks which are considered as false positives and false negatives respectively. Finally, if the block is authenticated as manipulated, the restoration watermark is extracted from the block where the watermark of the tampered block is saved using the following method. Let $\tilde{w}$ denotes the extracted watermark and $\tilde{x}$ denotes the DCT coefficient from which $\tilde{w}$ is extracted, calculate $\tilde{r} = \left\lfloor \frac{\tilde{x}}{T_2} + \frac{1}{2} \right\rfloor$, then

$$\tilde{w} = \begin{cases} \operatorname{mod}(\tilde{x} + \frac{T_2}{2}, T_2), & \text{if } \tilde{r} \text{ is odd}, \\ T_2 - \operatorname{mod}(\tilde{x} + \frac{T_2}{2}, T_2), & \text{if } \tilde{r} \text{ is even}, \end{cases} \tag{11}$$

where $T_2 > 0$ is the same quantization step size used in the embedding process. There are two processing strategies for image restoration on how to handle a manipulated (as detected) block $(i, j)$ whose corresponding block at location $g(i, j)$ is also marked as manipulated. The first strategy is to still use the extracted restoration watermark from Block $g(i, j)$ as the SSI was still correct and the other one is to regard the extracted restoration watermark from Block $g(i, j)$ as wrong information and delete the constraint corresponding to the "wrong" SSI from the model (9). After all restoration watermarks are extracted, the restoration model takes the tampered image and the restoration watermarks as the input to produce the restored image.

## 4. Experimental Results and Analysis for Image Restoration

To validate the performance of our proposed restoration model, we developed a software implementation based on the optimization software package IBM ILOG CPLEX [17], the same software used in [8]. The whole system was implemented in MATLAB and the linear optimization model is solved via the MATLAB interface of CPLEX.

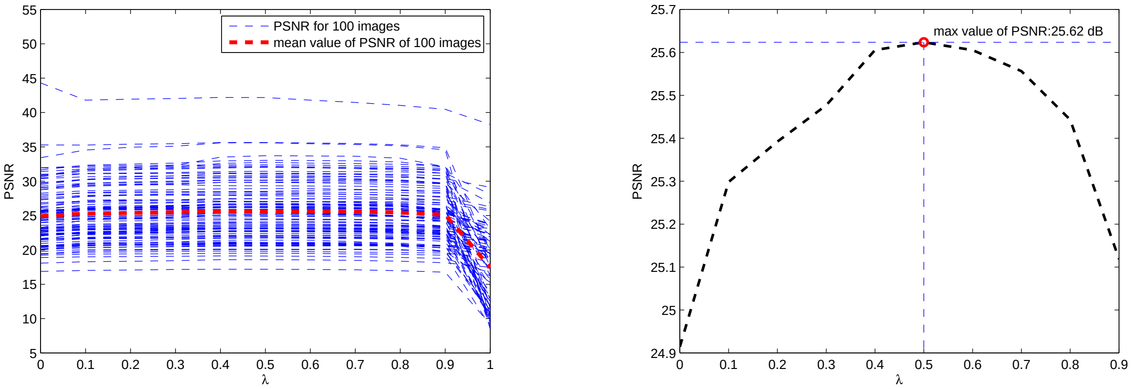
We used 100 gray-scale test images of size 256 × 256 for assessing the image restoration model, which are all widely used standard test images gathered from different public sources such as the CVG-UGR database [18] and the USC-SIPI database [19]. When the original images in the public sources are not 256 × 256, we rescaled them or cropped them to be of size 256 × 256. When the original images are true-color ones, we converted them to gray-scale editions using MATLAB's `rgb2gray()` function. To judge the visual quality of recovered images, we used two objective IQA metrics, PSNR and SSIM, and also manually inspected all recovered images to confirm their subjective quality. The PSNR and SSIM values are calculated for the recovered region only because the other part does not contribute to the performance evaluation of the image restoration models. In this section, we assume the SSI is 100% accurate to the receiver (which corresponds to an ideal watermarking algorithm) so that we can focus on the performance of the image restoration itself independent of the underlying watermarking scheme.

In this section we report experimental results on three aspects. Firstly, in Sec. 4.1, experimental results are given on the relationship between the regularization term's weight $\lambda$ and the quality of restored image in order to clarify how the two terms (the relaxed smoothness criterion and the regularization term) influence the overall image restoration results. The main conclusion of our study on this aspect is that the optimal value of $\lambda$ is 0.5, meaning that both terms in the objective function contribute equally to the final result at the optimum point. Secondly, Section 4.2 shows experimental results on performance evaluation and comparison of the following three image restoration methods: the linear regression based method proposed in [14] (M1), the simpler linear optimization model (1) with mean values of 4 × 4 blocks as the SSI (M2), and the proposed new model in (9) (M3). Finally, the computational complexity of the proposed image restoration method is analyzed in Sec. 4.3.

### *4.1. Impact of Weights on Performance*

The proposed model's objective contains two terms now: the relaxed smoothness criterion and a regularization term. Both terms are weighted by a value in [0,1]. The weights $\lambda$ and $1 - \lambda$ actually define the relative contributions of the two terms to the final objective. To better understand how the weights influence the final image restoration results, we conducted some experiments with different values of $\lambda$. Note that both weights are summed to 1, so there is only one independent weight and results on one weight covers another one naturally.

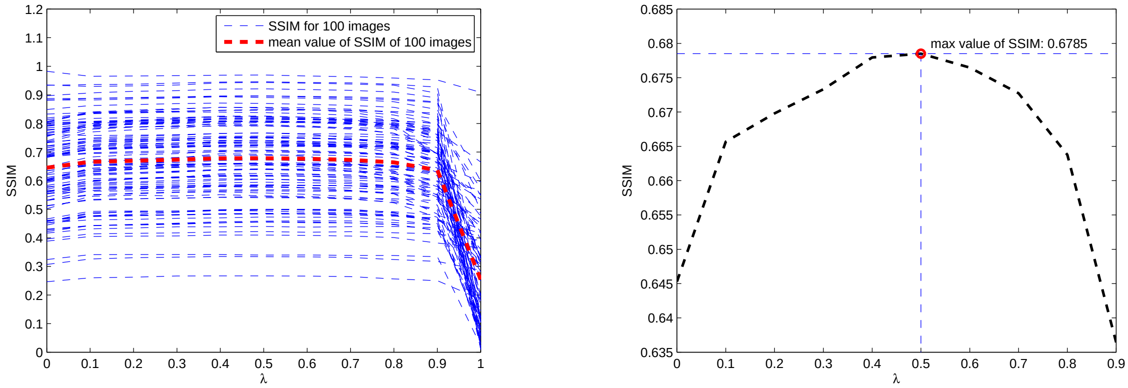
[2] The mathematics behind this arrangement is rather subtle. Since this paper's main focus is not the watermarking scheme, we do not cover the math here. The main idea is that when the manipulation rate is relatively low, the selected arrangement has a higher probability to match the ground truth.



(a)

(b)

Figure 5: The influence of the weight $\lambda$ on image restoration performance: (a) the relationship between $\lambda$ and the PSNR of 100 recovered images; (b) the mean values of PSNR with different values of $\lambda$ in rage of $[0, 1)$.



(a)

(b)

Figure 6: The influence of the weight $\lambda$ on image restoration performance: (a) the relationship between $\lambda$ and the SSIM of 100 recovered images; (b) the mean values of SSIM with different values of $\lambda$ in rage of $[0, 1)$.

Figures 5 and 6 present the PSNR and SSIM values of 100 recovered images and the mean values across all images when the value of $\lambda$ changes from 0 to 1 with a step size 0.1. As shown in Figs. 5(a) and 6(a), for all test images one can observe the following general pattern: the image quality remains rather consistent before a specific value of $\lambda$ but drop more significantly after $\lambda$ goes beyond that value. A closer look at the results when $\lambda \in [0, 0.9]$ revealed that there is actually an optimum value of $\lambda$ at the point of 0.5, as shown in Figs. 6(b) and 6(b). This implies that both terms in the objective function contribute to the final results equally so they complement to each other in an exactly balanced manner. The sharp drop of performance at the right end of the curve (corresponding to $\lambda = 1$) is not surprising because this value means complete removal of the smoothness term which is against the general principle of the image restoration model. On the other hand, when $\lambda = 0$ the results are still reasonably acceptable because without the regularization term the model can still work although with worsened performance. As a whole, the experimental results suggest the following two facts: 1) the first term (the relaxed smoothness criterion) is more important than the regularization term; 2) the regularization term helps to suppress side effects of the relaxed smoothness criterion to improve the overall image restoration quality.

Figure 7 shows the experimental results of recovering the test image “Lenna” using the proposed image restoration model with different values of $\lambda$. As illustrated in Fig. 7(a) when $\lambda = 0$, there are some sparse but obvious “pepper and salt” like noises in the recovered region. In Fig. 7(k) when $\lambda = 1$, one can see strong random texture like noises. When the value of $\lambda$ moves from both sides to the middle value 0.5, the quality of the recovered image improves steadily.

### 4.2. Comparison of Different Restoration Methods

In this subsection, we compare our proposed the three image restoration models (M1, M2 and M3) against each other, leading to the conclusion that M3 (the proposed new model) is the best among the three. To compare the performance of each pair methods, we copied the 10% left upper corner to the 10% bottom right corner of each test image to see how well the manipulated region can be recovered by the two methods.

#### 4.2.1. Performance Comparison: M2 vs. M1

The performance of M2 is shown in Fig. 8 in terms of image quality improvement it provides over M1. While for the majority of tested images the PSNR and SSIM values are improved, our manual inspection of the recovered images revealed that the images recovered by M2 still have clearly visible blocking artifacts along the boundary of many $4 \times 4$ blocks. Although there are a few individual images looking like "dominating" the PSNR results, our manual inspection revealed that these high peaks in PSNR results do not represent perceptually much better quality. Instead, we consider SSIM results more accurate since it is well known in the field that SSIM matches subjective quality better than PSNR [20] and our observations match with what we saw in the SSIM results. A closer analysis into the recovered pixel values further revealed that M2 often produces rather smooth $4 \times 4$ blocks so there is no finer structure in those blocks. Figures 1(d) and 1(e) show the "Lenna" images recovered by M1 and M2, respectively. One can see that both methods suffer from blocking artifacts although M2 at a lower level.

In order to verify the reliability of the inference above statistically, we applied paired $t$-tests to the PSNR and SSIM values obtained with M2 and M1 (as shown in Fig. 8) where the null hypothesis is that M2 has the same performance as M1 in terms of the PSNR/SSIM value recovered. The results showed that the null hypothesis is rejected for both PSNR and SSIM results ($p$-values are $1.5 \times 10^{-11}$ and $7.4 \times 10^{-24}$, respectively) in favor of the alternative hypothesis, *i.e.*, M2 outperforms M1 statistically significantly in terms of both PSNR and SSIM.

#### 4.2.2. Performance Comparison: M3 vs. M2 and M1

The performance of M3 is shown in Fig. 9 in terms of image quality improvement it provides over M1. It is clear that M3 outperforms M1 consistently, with an average gain of 1.92 dB for PSNR and an average increase of approximately 0.04 for SSIM. There are only two images whose SSIM values are lower when M3 is used, but our manual inspection did not reveal any noticeable visual difference. We also manually inspected all other 98 images recovered by each of the three methods, and confirmed that M3 was indeed able to remove most blocking artifacts and could produce images with better visual quality than M1 and M2. See Figs. 1(d), 1(e) and 1(g) for example recovery results of the three methods.

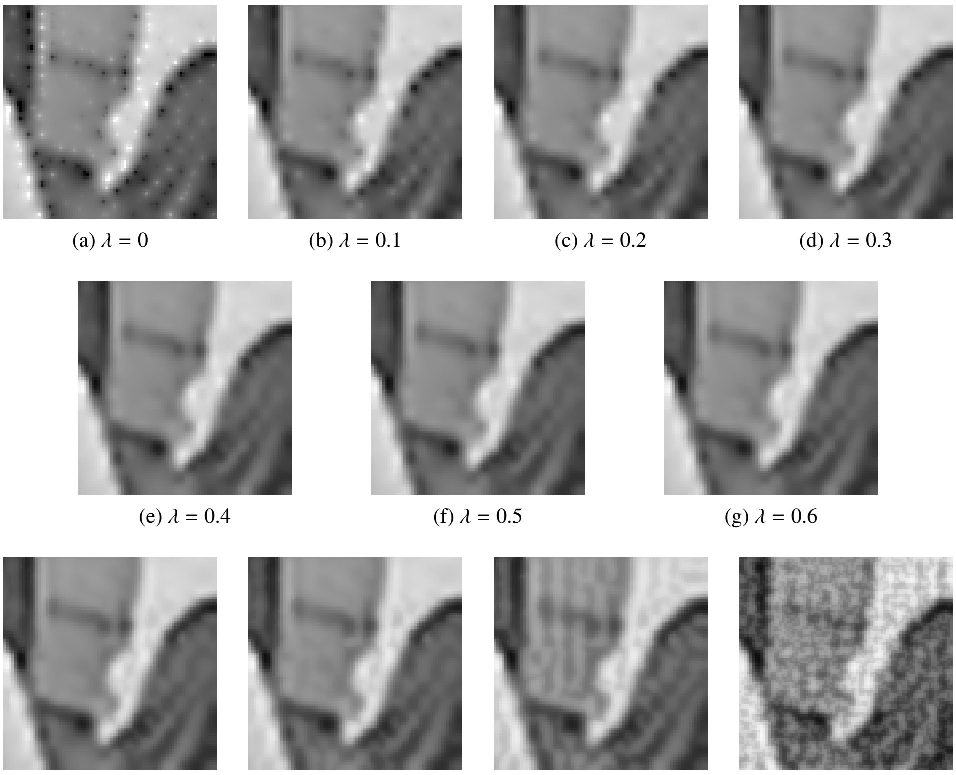
In order to verify the the above performance comparison results statistically, we applied paired $t$-tests to the PSNR and SSIM values obtained with M3, M2 and M1. The results showed that the improvements of M3 over M1 and M2 are statistically significant for both PSNR ($p$-values are $7.9 \times 10^{-18}$ and $5.9 \times 10^{-16}$ for "M3-M1" and "M3-M2", respectively) and SSIM results ($p$-values are $3.4 \times 10^{-40}$ and $5.9 \times 10^{-21}$ for "M3-M1" and "M3-M2", respectively).

### 4.3. Computational Complexity of M3

We also analyzed the time complexity of the proposed image restoration method to investigate its computational efficiency. Given an $M \times N$ image with $r$ percentage of pixels modified, the mathematical model presents an optimization problem with $3MN$ variables. For our problem, the pixels in the unmodified region are actually known variables. Thus, the amount of time taken by running our restoration algorithm is dependent on $3rMN$ variables only. We tested different size of images with different percentage of modification, and recorded the running time consumed by the restoration process of each test image. The relationship between the time $t$ (seconds) and the number of restored pixels $x = rMN$ can be fit well as a polynomial function $y = a \cdot x^{2.4836}$, where $a = 7.6435 \times 10^{-10}$. It is thus likely the time complexity of M3 is $O((rMN)^{2.5})$. Taking the image "Lenna" of size $256 \times 256$ as an example, the running time for recovering 10% content is approximately 3.46 seconds. While the time consumed is relatively long (which is not surprising for a model based on optimization), it is still acceptable for the target application where no real-time processing is required.

## 5. Experimental Results on Watermarking-related Performance

In this section, we report experimental results when the proposed image restoration model and other two



(a) $\lambda = 0$

(b) $\lambda = 0.1$

(c) $\lambda = 0.2$

(d) $\lambda = 0.3$

(e) $\lambda = 0.4$

(f) $\lambda = 0.5$

(g) $\lambda = 0.6$

(h) $\lambda = 0.7$

(i) $\lambda = 0.8$

(j) $\lambda = 0.9$

(k) $\lambda = 1$

Figure 7: Experimental results of recovered image on the test image“Lenna” with different weight values of $\lambda$.

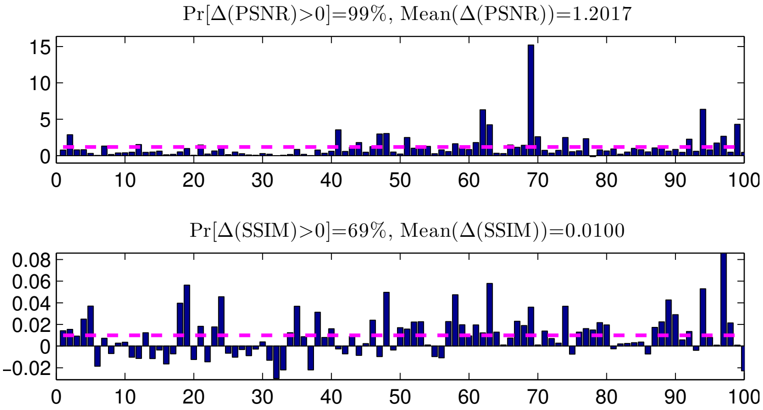


Figure 8: The performance improvement of M2 over M1, in terms of visual quality improvement defined as the difference of PSNR and SSIM values of the image recovered by M2 and that recovered by M1. The dashed line shows the mean of the visual quality differences across all the 100 tested images and the equation Pr[Δ(PSNR) > 0] shows the percentage of recovered image with visual quality improvement.

competitive models work with “realistic” (*i.e.* not 100% correct) SSI recovered by the underlying watermarking algorithm. As we mentioned in Section 3, there are two restoration processing strategies in the watermarking algorithm for handling blocks whose restoration watermarks extracted from blocks marked as “manipulated”. In our experiments, we chose the first strategy because 1) the blocks storing the restoration watermarks may be false positives; 2) our experiments showed the first strategy outperforms the second slightly. After the wa-

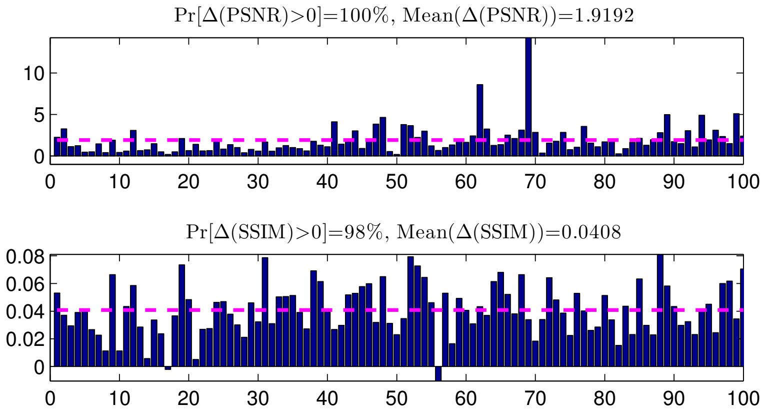


Figure 9: The performance improvement of M3 over M1, in terms of visual quality improvement defined as the difference of PSNR and SSIM values of the image recovered by M3 and that recovered by M1.

termarking algorithm is implemented, we applied the three image restoration methods to it in order to test the overall performance of the watermarking system and the three image restoration methods working with the watermarking system. In the remaining part of this section, we first give some results on the general properties of the underlying watermarking system without covering robustness. Then, we present experimental results about robustness of the three image restoration methods in the context of the underlying watermarking algorithms.

### 5.1. *Performance of Underlying Watermarking Algorithm*

Since our proposed image restoration method is simulated with a specific semi-fragile watermarking algorithm, here we present some results to show different properties of the watermarking algorithm excluding the robustness against various image processing operations. These properties include capacity, imperceptibility, authentication accuracy and restored image quality without any attacks when the SSI is embedded using the underlying watermarking algorithm (which may not produce 100% correct SSI at the receiver side). The same 100 256×256 gray-scale images were used to test the watermarking algorithm.

As we introduced in Section 3, the capacities of the two authentication and restoration watermarking embedding methods are 6 bits per $8 \times 8$ block for authentication watermark and 4 scaled mean values of $4 \times 4$ blocks per $8 \times 8$ block for restoration watermark, respectively. The imperceptibility of the watermarking algorithm is measured by both PSNR and SSIM. Note that the above are not maximum capacities of the corresponding watermark embedding methods. On average, the quality of watermarked image achieves PSNR≈37.41 dB and SSIM≈0.9510. The authentication accuracy of the watermarking algorithm is measured in terms of the false positive rate (FPR) and the false negative rate (FNR). False positive (FP) refers to blocks in the non-manipulated region which are falsely detected as manipulated ones, and false negative (FN) refers to blocks in the manipulated region which are falsely detected as non-manipulated ones. Our experimental results showed that the watermarking system was able to achieve a nearly perfect authentication rate when the watermarked image does not go through any lossy processing: the average FPR is just 0.3% and the FNR is 0 for all the 100 test images. The quality of recovered image achieves nearly same results as analyzed in Section 4 since the manipulated region can be 100% correctly authenticated (*i.e.* FN probability is 0) and the restoration watermark can be extracted without lossy information (*i.e.* the SSI is nearly 100% correct) when the watermarked images does not go through any lossy processing.

Figure 10 illustrates an example of the whole watermarking process in the case that the image is maliciously manipulated by collage attack proposed by Fridrich in [21]. As demonstrated here, Fig. 10(a) and 10(b) are two watermarked images produced by using the same key. Then, as shown in Fig. 10(c), the boat in Fig. 10(b) is copied-and-pasted to the same place in Fig. 10(a). At the receiver side the authentication watermark is extracted by the previously mentioned watermarking algorithm and Figure 10(d) illustrates the attacked regions as detected by the image authentication process. Finally, for each $8 \times 8$ block in the detected regions, the corresponding restoration watermark is extracted and Figs. 10(e)-10(f) show the restored image and a close-up look of the attacked region which has a

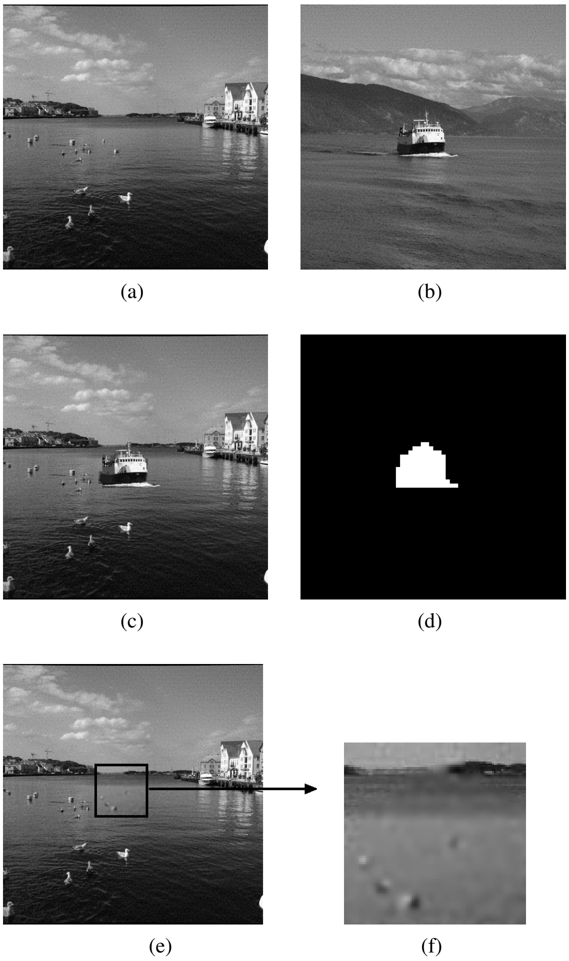


Figure 10: Sample results of a collage attack, image authentication and self-restoration: (a) the first watermarked background image “Shore”; (b) a second watermarked image “Boat”; (c) collage attacked image with the boat in (b) is copied and pasted to the same place in (a); (d) localization of the image content modification; (e) the recovered image; (f) the close-up version of the recovered area of the image (e).

good subjective visual quality.

### *5.2. Robustness Comparison*

We also conducted some experiments to see if the proposed method M3 is more robust against some image processing operations than the other two methods M1 and M2. To this end, various non-malicious attacks including lossy JPEG compression, additive Gaussian white noise and rescaling were applied to the same 100 test images with 10% content modified. Table 1 shows the average values of FPR and FNR of authentication rate, and average values of PSNR and SSIM of the recovered images after the test images were JPEG compressed with QF from 100 to 50, additive Gaussian white noise was added with variance $\sigma^2$ from 1 to 10,[3] and the image were rescaled with a ratio from 0.7 to 1.5. For each row the boldfaced results show the best PSNR and SSIM values across the three compared image restoration methods.

For JPEG compression, the authentication false rate remains quite low (*i.e.* FPR$\leqslant$ 2.43% and FNR $\leqslant$ 0.01%) until JPEG QF reaches 50. The results for recovered image quality show that M3 keeps performing better than M1 and M2 until QF decreases to around 70-75 after which M2 produces slightly better results. This suggests that M3 is slightly less robust than M2. However, this should not be seen as a real drawback because the quality of recovered images becomes very close when QF drops below 75, so all three methods have similar performance.

For additive Gaussian white noise, the authentication rate of the underlying semi-fragile watermarking algorithm represented good robustness for all the tested image operation (FPR$\leqslant$ 4.78% and FNR$\leqslant$ 0.08%). The results for recovered image quality show that M3 performs better than M1 and M2 for all the variance values when the image quality is evaluated by SSIM, although the PSNR of M3 produces slightly worse results when the variance $\sigma^2 \geqslant 4$. Since it is well known in the field that SSIM matches subjective quality better than PSNR, we consider SSIM results to be more accurate.

For rescaling test we assume that the information about the original image size is either fixed (e.g. in CCTV applications) or the original size is transmitted to the receiver via a side channel (which could be a third watermark embedded). Under this assumption, we first converted the rescaled image back to its original image size, and then extracted watermark by the same method for image content authentication and self-restoration. From Table 1 we can see that the performance is rather bad for both authentication and restoration when the scaling ratio down to 0.7, and relatively poor when the scaling ratio up to 1.5. When the scaling ratio goes closer to 1, there is a general trend for the performance to become better both in terms of PSNR and SSIM. Generally speaking, judging from PSNR results M2 is the best among the three methods but from SSIM results M3 is the best. Since SSIM is a more accurate objective IQA metric than PSNR as we mentioned before, we believe M3 is more capable of producing perceptually better results than M1 and M2. As a whole the

[3] We actually used the normalized values of the variance ($\sigma^2/255^2$) in our MATLAB code since they are what the MATLAB function `imnoise()` requires.

Table 1: Robustness against some image processing operations of different self-restoration watermarking methods.

| | | Authentication Rate | | Recovered Image Quality | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | PSNR (dB) | | | SSIM | | |
| | | FPR (%) | FNR (%) | M1 | M2 | M3 | M1 | M2 | M3 |
| JPEG Compression | | | | | | | | | |
| QF = | 100 | 0.35 | 0.00 | 23.567 | 24.685 | **25.373** | 0.622 | 0.635 | **0.665** |
| | 95 | 0.44 | 0.00 | 23.270 | 24.302 | **24.878** | 0.592 | 0.609 | **0.636** |
| | 90 | 0.48 | 0.00 | 22.654 | 23.587 | **23.949** | 0.545 | 0.565 | **0.585** |
| | 85 | 0.67 | 0.00 | 21.805 | 22.597 | **22.777** | 0.471 | 0.492 | **0.505** |
| | 80 | 0.71 | 0.00 | 20.955 | 21.622 | **21.687** | 0.447 | 0.465 | **0.477** |
| | 75 | 1.14 | 0.00 | 20.212 | **20.796** | 20.740 | 0.402 | 0.420 | **0.425** |
| | 70 | 1.07 | 0.00 | 19.351 | **19.938** | 19.742 | 0.343 | 0.358 | **0.361** |
| | 65 | 1.26 | 0.00 | 18.584 | **19.136** | 18.950 | 0.347 | **0.367** | 0.364 |
| | 60 | 1.64 | 0.01 | 17.847 | **18.327** | 18.097 | 0.313 | **0.330** | 0.326 |
| | 55 | 2.43 | 0.00 | 17.146 | **17.643** | 17.308 | 0.291 | **0.307** | 0.300 |
| | 50 | 99.99 | 0.00 | 16.597 | **16.876** | 16.502 | 0.276 | **0.281** | 0.273 |
| Additive Gaussian White Noise | | | | | | | | | |
| $\sigma^2$ = | 1 | 0.41 | 0.00 | 22.444 | 23.245 | **23.615** | 0.513 | 0.529 | **0.552** |
| | 2 | 0.48 | 0.01 | 21.617 | 22.311 | **22.526** | 0.455 | 0.469 | **0.487** |
| | 3 | 0.56 | 0.00 | 21.005 | 21.643 | **21.695** | 0.412 | 0.424 | **0.437** |
| | 4 | 0.65 | 0.00 | 20.386 | **20.995** | 20.952 | 0.379 | 0.390 | **0.401** |
| | 5 | 0.71 | 0.00 | 19.965 | **20.557** | 20.450 | 0.356 | 0.366 | **0.376** |
| | 6 | 0.95 | 0.03 | 19.522 | **20.069** | 19.923 | 0.334 | 0.343 | **0.351** |
| | 7 | 1.36 | 0.01 | 19.056 | **19.591** | 19.365 | 0.312 | 0.320 | **0.326** |
| | 8 | 1.96 | 0.01 | 18.799 | **19.332** | 19.062 | 0.301 | 0.309 | **0.314** |
| | 9 | 2.89 | 0.08 | 18.466 | **18.991** | 18.708 | 0.288 | 0.296 | **0.300** |
| | 10 | 4.78 | 0.03 | 18.210 | **18.708** | 18.378 | 0.277 | 0.283 | **0.286** |
| Image Rescaling | | | | | | | | | |
| scaling | 0.7 | 83.60 | 0.08 | 11.374 | **11.513** | 10.955 | 0.094 | **0.098** | 0.091 |
| ratio = | 0.8 | 14.42 | 0.00 | 17.911 | **18.350** | 18.027 | 0.311 | 0.324 | **0.329** |
| | 0.9 | 27.73 | 0.00 | 17.420 | **17.846** | 17.464 | 0.295 | 0.307 | **0.309** |
| | 1.1 | 5.71 | 0.00 | 19.78 | **20.334** | 20.216 | 0.390 | 0.405 | **0.416** |
| | 1.2 | 5.15 | 0.02 | 19.853 | **20.405** | 20.300 | 0.393 | 0.407 | **0.418** |
| | 1.3 | 14.41 | 0.02 | 19.020 | **19.544** | 19.307 | 0.358 | 0.372 | **0.381** |
| | 1.4 | 11.23 | 0.02 | 19.269 | **19.804** | 19.597 | 0.368 | 0.382 | **0.391** |
| | 1.5 | 54.84 | 0.13 | 14.922 | **15.207** | 14.666 | 0.212 | **0.222** | 0.218 |

image restoration scheme is moderately robust to scaling especially when the scaling ratio is close to 1.

In order to verify the above results statistically, we also applied paired $t$-tests to the PSNR and SSIM values obtained in our experiments. The results with the $p$-values are shown in Table 2 where "$H_0$" means that the null hypothesis $M_{latter} = M_{former}$ cannot be rejected at a significance level of 5% and "$H_1$" means that the null hypothesis is rejected in favour of the alternative hypothesis. According to the mean values shown in Table 1, we marked the two different cases of the alternative hypothesis as "$H_{1+}$" when the mean difference (the variable on the right minus the one on the left) is positive, and "$H_{1-}$" when the mean difference is negative. The data in Table 2 shows that the improvement of M2 over M1 is statistically significant for nearly all attacks we applied in our experiments except one SSIM result for JPEG QF=50 with a small $p$-value of 0.069. The performance difference between M3 and M1 is also statistically significant except one SSIM result for JPEG QF=50 and one PSNR result for rescaling with the scaling ratio of 0.9, and it is clear that M3 outperforms

Table 2: The results of *t*-test: the test decision at the 5% significance level and its *p*-value.

|  | M1 *vs.* M2 |  | M1 *vs.* M3 |  | M2 *vs.* M3 |  |
|---|---|---|---|---|---|---|
|  | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| JPEG Compression |  |  |  |  |  |  |
| QF = 100 | $H_{1+}$ $(4.2\times10^{-11})$ | $H_{1+}$ $(1.7\times10^{-08})$ | $H_{1+}$ $(7.6\times10^{-21})$ | $H_{1+}$ $(1.2\times10^{-41})$ | $H_{1+}$ $(5.8\times10^{-17})$ | $H_{1+}$ $(8.6\times10^{-22})$ |
| 95 | $H_{1+}$ $(1.0\times10^{-13})$ | $H_{1+}$ $(1.1\times10^{-11})$ | $H_{1+}$ $(2.2\times10^{-25})$ | $H_{1+}$ $(1.5\times10^{-44})$ | $H_{1+}$ $(2.8\times10^{-15})$ | $H_{1+}$ $(2.3\times10^{-19})$ |
| 90 | $H_{1+}$ $(1.4\times10^{-14})$ | $H_{1+}$ $(5.8\times10^{-15})$ | $H_{1+}$ $(8.2\times10^{-25})$ | $H_{1+}$ $(3.2\times10^{-43})$ | $H_{1+}$ $(3.8\times10^{-09})$ | $H_{1+}$ $(2.2\times10^{-13})$ |
| 85 | $H_{1+}$ $(4.2\times10^{-18})$ | $H_{1+}$ $(1.3\times10^{-16})$ | $H_{1+}$ $(3.1\times10^{-27})$ | $H_{1+}$ $(1.4\times10^{-35})$ | $H_{1+}$ $(1.6\times10^{-04})$ | $H_{1+}$ $(4.8\times10^{-07})$ |
| 80 | $H_{1+}$ $(2.4\times10^{-24})$ | $H_{1+}$ $(2.5\times10^{-16})$ | $H_{1+}$ $(1.2\times10^{-24})$ | $H_{1+}$ $(4.9\times10^{-35})$ | $H_0$ $(0.13)$ | $H_{1+}$ $(1.3\times10^{-05})$ |
| 75 | $H_{1+}$ $(3.4\times10^{-24})$ | $H_{1+}$ $(6.0\times10^{-12})$ | $H_{1+}$ $(1.4\times10^{-20})$ | $H_{1+}$ $(4.7\times10^{-20})$ | $H_0$ $(0.091)$ | $H_{1+}$ $(0.020)$ |
| 70 | $H_{1+}$ $(1.1\times10^{-19})$ | $H_{1+}$ $(4.6\times10^{-12})$ | $H_{1+}$ $(1.5\times10^{-12})$ | $H_{1+}$ $(1.7\times10^{-18})$ | $H_{1-}$ $(1.8\times10^{-09})$ | $H_0$ $(0.25)$ |
| 65 | $H_{1+}$ $(2.2\times10^{-15})$ | $H_{1+}$ $(1.0\times10^{-16})$ | $H_{1+}$ $(1.1\times10^{-08})$ | $H_{1+}$ $(4.3\times10^{-25})$ | $H_{1-}$ $(1.1\times10^{-10})$ | $H_0$ $(0.15)$ |
| 60 | $H_{1+}$ $(3.7\times10^{-14})$ | $H_{1+}$ $(5.8\times10^{-16})$ | $H_{1+}$ $(1.3\times10^{-05})$ | $H_{1+}$ $(3.5\times10^{-19})$ | $H_{1-}$ $(5.6\times10^{-17})$ | $H_{1-}$ $(0.049)$ |
| 55 | $H_{1+}$ $(1.2\times10^{-21})$ | $H_{1+}$ $(3.7\times10^{-14})$ | $H_{1+}$ $(1.0\times10^{-04})$ | $H_{1+}$ $(2.2\times10^{-11})$ | $H_{1-}$ $(1.2\times10^{-26})$ | $H_{1-}$ $(5.9\times10^{-04})$ |
| 50 | $H_{1+}$ $(4.4\times10^{-11})$ | $H_0$ $(0.069)$ | $H_{1-}$ $(8.0\times10^{-03})$ | $H_0$ $(0.22)$ | $H_{1-}$ $(2.0\times10^{-31})$ | $H_{1-}$ $(1.6\times10^{-04})$ |
| Additive Gaussian White Noise |  |  |  |  |  |  |
| $\sigma^2$ = 1 | $H_{1+}$ $(1.6\times10^{-19})$ | $H_{1+}$ $(6.8\times10^{-13})$ | $H_{1+}$ $(2.7\times10^{-34})$ | $H_{1+}$ $(3.1\times10^{-40})$ | $H_{1+}$ $(1.7\times10^{-12})$ | $H_{1+}$ $(3.8\times10^{-18})$ |
| 2 | $H_{1+}$ $(4.6\times10^{-22})$ | $H_{1+}$ $(7.8\times10^{-12})$ | $H_{1+}$ $(1.5\times10^{-35})$ | $H_{1+}$ $(9.6\times10^{-37})$ | $H_{1+}$ $(7.1\times10^{-06})$ | $H_{1+}$ $(2.3\times10^{-12})$ |
| 3 | $H_{1+}$ $(2.2\times10^{-26})$ | $H_{1+}$ $(1.7\times10^{-10})$ | $H_{1+}$ $(3.4\times10^{-38})$ | $H_{1+}$ $(8.1\times10^{-29})$ | $H_0$ $(0.18)$ | $H_{1+}$ $(3.6\times10^{-08})$ |
| 4 | $H_{1+}$ $(2.4\times10^{-28})$ | $H_{1+}$ $(3.7\times10^{-09})$ | $H_{1+}$ $(7.8\times10^{-38})$ | $H_{1+}$ $(2.9\times10^{-23})$ | $H_0$ $(0.25)$ | $H_{1+}$ $(1.4\times10^{-06})$ |
| 5 | $H_{1+}$ $(1.3\times10^{-32})$ | $H_{1+}$ $(1.2\times10^{-08})$ | $H_{1+}$ $(3.4\times10^{-34})$ | $H_{1+}$ $(6.2\times10^{-19})$ | $H_{1-}$ $(4.2\times10^{-03})$ | $H_{1+}$ $(5.8\times10^{-05})$ |
| 6 | $H_{1+}$ $(2.1\times10^{-32})$ | $H_{1+}$ $(8.3\times10^{-08})$ | $H_{1+}$ $(5.7\times10^{-27})$ | $H_{1+}$ $(2.7\times10^{-17})$ | $H_{1-}$ $(1.9\times10^{-04})$ | $H_{1+}$ $(2.1\times10^{-04})$ |
| 7 | $H_{1+}$ $(5.3\times10^{-34})$ | $H_{1+}$ $(1.6\times10^{-08})$ | $H_{1+}$ $(3.2\times10^{-25})$ | $H_{1+}$ $(2.7\times10^{-15})$ | $H_{1-}$ $(8.8\times10^{-10})$ | $H_{1+}$ $(6.1\times10^{-03})$ |
| 8 | $H_{1+}$ $(1.1\times10^{-36})$ | $H_{1+}$ $(2.0\times10^{-09})$ | $H_{1+}$ $(7.6\times10^{-26})$ | $H_{1+}$ $(4.6\times10^{-12})$ | $H_{1-}$ $(3.9\times10^{-14})$ | $H_{1+}$ $(0.036)$ |
| 9 | $H_{1+}$ $(2.5\times10^{-37})$ | $H_{1+}$ $(2.9\times10^{-08})$ | $H_{1+}$ $(3.8\times10^{-20})$ | $H_{1+}$ $(2.0\times10^{-11})$ | $H_{1-}$ $(1.2\times10^{-13})$ | $H_{1+}$ $(0.023)$ |
| 10 | $H_{1+}$ $(3.4\times10^{-37})$ | $H_{1+}$ $(7.4\times10^{-07})$ | $H_{1+}$ $(9.1\times10^{-16})$ | $H_{1+}$ $(8.1\times10^{-09})$ | $H_{1-}$ $(2.4\times10^{-19})$ | $H_0$ $(0.12)$ |
| Image Rescaling |  |  |  |  |  |  |
| ratio= 0.7 | $H_{1+}$ $(1.2\times10^{-08})$ | $H_{1+}$ $(3.2\times10^{-07})$ | $H_{1+}$ $(2.0\times10^{-30})$ | $H_{1-}$ $(1.7\times10^{-06})$ | $H_{1-}$ $(4.5\times10^{-64})$ | $H_{1-}$ $(7.6\times10^{-14})$ |
| 0.8 | $H_{1+}$ $(4.1\times10^{-19})$ | $H_{1+}$ $(6.1\times10^{-15})$ | $H_{1+}$ $(5.7\times10^{-03})$ | $H_{1+}$ $(1.0\times10^{-23})$ | $H_{1-}$ $(6.1\times10^{-26})$ | $H_{1+}$ $(0.017)$ |
| 0.9 | $H_{1+}$ $(1.3\times10^{-20})$ | $H_{1+}$ $(2.0\times10^{-13})$ | $H_0$ $(0.24)$ | $H_{1+}$ $(1.6\times10^{-19})$ | $H_{1-}$ $(7.8\times10^{-32})$ | $H_0$ $(0.17)$ |
| 1.1 | $H_{1+}$ $(4.8\times10^{-24})$ | $H_{1+}$ $(6.5\times10^{-15})$ | $H_{1+}$ $(5.0\times10^{-17})$ | $H_{1+}$ $(3.9\times10^{-31})$ | $H_{1-}$ $(3.3\times10^{-04})$ | $H_{1+}$ $(7.3\times10^{-08})$ |
| 1.2 | $H_{1+}$ $(5.7\times10^{-24})$ | $H_{1+}$ $(6.2\times10^{-15})$ | $H_{1+}$ $(3.1\times10^{-17})$ | $H_{1+}$ $(5.6\times10^{-31})$ | $H_{1-}$ $(7.6\times10^{-04})$ | $H_{1+}$ $(2.5\times10^{-07})$ |
| 1.3 | $H_{1+}$ $(1.9\times10^{-24})$ | $H_{1+}$ $(7.5\times10^{-15})$ | $H_{1+}$ $(5.4\times10^{-11})$ | $H_{1+}$ $(2.9\times10^{-30})$ | $H_{1-}$ $(4.4\times10^{-14})$ | $H_{1+}$ $(2.2\times10^{-05})$ |
| 1.4 | $H_{1+}$ $(1.8\times10^{-24})$ | $H_{1+}$ $(1.2\times10^{-14})$ | $H_{1+}$ $(1.3\times10^{-12})$ | $H_{1+}$ $(3.6\times10^{-29})$ | $H_{1-}$ $(8.3\times10^{-11})$ | $H_{1+}$ $(4.5\times10^{-06})$ |
| 1.5 | $H_{1+}$ $(1.6\times10^{-18})$ | $H_{1+}$ $(7.0\times10^{-13})$ | $H_{1-}$ $(3.8\times10^{-16})$ | $H_{1+}$ $(3.3\times10^{-06})$ | $H_{1-}$ $(6.1\times10^{-56})$ | $H_{1-}$ $(6.6\times10^{-04})$ |

M1 significantly for JPEG compression with QF from 100 down to 55, additive Gaussian white noise with variance from 1 to 10, and image rescaling with a ratio of 0.8 or between 1.1 and 1.5. When M3 and M2 are compared, the t-test results suggest that M3 outperforms M2 for a smaller parameter region for each type of attack and each quality metric. As a whole, the advantage of M3 over M1 and M2 decreases while the attack becomes stronger (in terms of distortion) and M2 performs slightly better than M3 (but statistically significantly) when the attack is sufficiently strong, which suggests that M3 and M2 can be used in different application scenarios depending on the expected level of attacks.

Since the recovered image quality in watermarking application depends on both the authentication rate and the accuracy of restoration watermarks, its value drops quickly when the watermarked images go through some image processing operations against which the underlying watermarking schemes are not robust. For instance, the watermarking system we used in this paper to demonstrate the proposed image restoration method is not designed to be robust against geometric distortions (rotation, scaling and translation (RST)) which can be seen in Table 1 (the system fails when a too small or too large scaling factor is used). Despite this RST robustness issue, we would like to highlight that the main focus of this paper is the image restoration model which is actually watermarking independent even though it came from and can be applied to watermarking applications. To improve the whole watermarking system's robustness against RST operations, one needs to switch to more advanced RST-resistant digital watermarking algorithms such as those proposed in [22, 23].

## 6. Conclusion and Future Work

This paper presents a novel image restoration method based on a linear optimization model. The performances of the optimization model was demonstrated by comparing with two other models when applied to a recently proposed self-restoration watermarking scheme [14]. Experimental results have shown that the proposed model outperforms similar models including the one in [8] and the original image restoration method used in [14].

In our future work, we will try to find a rigorous proof of the failure of the simpler optimization model (1) and the success of the proposed model (9). We plan to investigate if the model proposed in this paper can be further improved *e.g.* by further generalizing the smoothness criterion to be a weighted sum of the four adjacent pixel value differences. We will also investigate other types of structured side information to see if there is a better structure supporting an even more efficient image restoration model which can then be used to design a better self-restoration watermarking scheme. In addition, how to make the underlying watermarking system and thus the proposed image restoration model more robust against RST is another direction in future research.

[1] M. R. Banham, A. K. Katsaggelos, Digital image restoration, IEEE Signal Processing Magazine 14 (2) (1997) 24–41.

[2] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: Proceedings of 27th ACM Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2000), 2000, pp. 417–424.

[3] N. Komodakis, G. Tziritas, Image completion using efficient belief propagation via priorityscheduling and dynamic pruning, IEEE Transactions on Image Processing 16 (11) (2007) 2649–2661.

[4] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, C. Rother, A comparative study of energy minimization methods for markov randomfields with smoothness-based priors, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (6) (2008) 1068–1080.

[5] M.-S. Koh, E. Rodriguez-Marek, Turbo inpainting: Iterative *k*-SVD with a new dictionary, in: Proceedings of 2009 IEEE International Workshop on Multimedia Signal Processing (MMSP 2009), IEEE, 2009, pp. 1–6.

[6] W. K. Cham, R. J. Clarke, DC coefficient restoration in transform image coding, IEE Proceedings Part F 131 (7) (1984) 709–713.

[7] T. Uehara, R. Safavi-Naini, P. Ogunbona, Recovering DC coefficients in block-based DCT, IEEE Transactions on Image Processing 15 (11) (2006) 3592–3596.

[8] S. Li, A. Karrenbauer, D. Saupe, C.-C. J. Kuo, Recovering missing coefficients in DCT-transformed images, in: Proceedings of 18th IEEE International Conference on Image Processing (ICIP 2011), IEEE, 2011, pp. 1537–1540.

[9] A. T. S. Ho, X. Zhu, Y. L. Guan, Image content authentication using Pinned Sine Transform, EURASIP Journal of Applied Signal Processing 2004 (11) (2004) 2174–2184.

[10] Y. M. Y. Hasan, A. M. Hassan, Tamper detection with self-correction hybrid spatial-DCT domains image authentication technique, in: Proceedings of 2007 IEEE International Symposium on Signal Processing and Information Technology, IEEE, 2007, pp. 369–374.

[11] C. Cruz, J. A. Mendoza, M. N. Miyatake, H. P. Meana, B. Kurkoski, Semi-fragile watermarking based image authentication with recovery capability, in: Proceedings of 2009 International Conference on Information Engineering and Computer Science (ICIECS 2009), IEEE, 2009, pp. 1–4.

[12] S. Lin, J. Lin, Authentication and recovery of an image by sharing and lattice-embedding, Journal of Electronic Imaging 19 (4) (2010) art. no. 043008.

[13] A. Phadikar, S. P. Maity, M. Mandal, Novel wavelet-based QIM data hiding technique for tamper detectionand correction of digital images, Journal of Visual Communication and Image Representation 23 (3) (2012) 454–466.

[14] H. Wang, A. T. S. Ho, X. Zhao, A novel fast self-restoration semi-fragile watermarking algorithm for image content authentication resistant to JPEG compression, in: Digital Forensics and Watermarking: 10th International Workshop,IWDW 2011, Atlantic City, NY, October 23-26, 2011, Revised SelectedPapers, Vol. 7128 of Lecture Notes in Computer Science, Springer, 2012, pp. 72–85.

[15] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612.

[16] J. K. Ord, Laplace Distribution, Vol. 6, John Wiley & Sons, Inc., 2004, pp. 3961–3962.

[17] IBM, IBM ILOG CPLEX optimizer, `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`.

[18] Computer Vision Group, University of Granada, Spain, CVG-UGR image database, `http://decsai.ugr.es/cvg/dbimagenes/`.

[19] Signal and Image Processing Institute, University of South Califonia, USA, The USC-SIPI image database, `http://sipi.usc.edu/database/`.

[20] Z. Wang, A. C. Bovik, Modern Image Quality Assessment, Synthesis lectures on image, Morgan & Claypool Publishers, 2006.

[21] J. Fridrich, M. Goljan, N. Memon, cryptanalysis of the Yeung-Mintzer fragile watermarking technique, Journal of Electronic Imaging 11 (2) (2002) 262–274.

[22] C. Deng, X. Gao, X. Li, D. Tao, A local Tchibichef moments-based robust image watermarking, Signal Processing 89 (8) (2009) 1531–1539.

[23] X. Gao, C. Deng, X. Li, D. Tao, Geometric distortion insensitive image watermarking in affine covariant regions, IEEE Transactions on Systems Man and Cybernetics, Part C: Applications and Reviews 40 (3) (2010) 278–286.