EUSIPCO 2013 1569744413

HIGH-QUALITY SELF-EMBEDDING FOR JPEG-COMPRESSED DIGITAL IMAGES

Paweł Korus, Jarosław Białas and Andrzej Dziech

Department of Telecommunications, AGH University of Science and Technology,
al. Mickiewicza 30, 30-059 Krakow, Poland, E-mail: {pkorus,bialas}@agh.edu.pl

ABSTRACT

This paper deals with the design of a self-embedding scheme for JPEG-compressed images. Most of existing schemes are compatible only with loss-less images. Few of them are capable of handling lossy compression, but deliver very low restoration fidelity, and support only small amounts of tampered content. In this study, we extend a recently proposed self-embedding model, and perform theoretical analysis of the impact of watermark extraction and block classification errors on the achievable reconstruction performance. The theoretical results are verified experimentally with a new scheme dedicated to JPEG. Our scheme achieves the average reconstruction quality between 28 dB and 33 dB, for the maximum allowed tampering rates of 50% and 20%, respectively.

Index Terms— content reconstruction, self-embedding, image authentication, digital watermarking, fountain codes

1. INTRODUCTION

Self-embedding is a pro-active digital image protection technique, which allows for the reconstruction of maliciously tampered image fragments. In addition to content hashes, an encoder embeds in the image a reconstruction reference, which is exploited by a decoder to aid content restoration [1].

Due to high requirements towards watermarking capacity, most of existing schemes use least significant bit substitution for information embedding. Hence, they are not compatible with lossy-compressed image formats. There exist only a few schemes, which can tolerate lossy compression, yet still only to a limited extent. The typical approach is to use a restoration technique, which is robust against minor watermark recovery errors [2, 3]. In [3] the watermark is a binary halftone image, and the emerging errors introduce noise into the restoration result, which quickly becomes indiscernible. Such schemes typically feature low reconstruction quality, ranging from 22 dB to 28 dB in terms of peak signal to noise ratio (PSNR), even when no compression is actually observed.

There also exist format-specific schemes, e.g., [4, 5] for JPEG. The former uses a down-sampled gray-scale version of the original image, compressed to an equivalent of JPEG quality level 25, which severely limits the restoration fidelity. Reconstruction is possible if the tampering affects only a small area. A specific bound on the tampering rate is not reported. The scheme from [5] uses linear regression to predict first 4 DCT coefficients of the tampered blocks from the embedded reference information. Again, the maximum achievable reconstruction quality is low. The PSNR averages at 25 dB, and drops even further with the JPEG quality level. Again, the scheme tolerates only limited tampering, and no specific bound on the tampering rate is given.

For practical use of self-embedding with JPEG, it is necessary to develop a dedicated scheme, capable of high-quality reconstruction, even under extensive tampering. In this paper, we present such a scheme. For this purpose, we adopt a recently proposed content reconstruction model based on digital fountain codes [6]. We extend the analysis to address problems, which are specific to lossy-compressed images. Based on the performed analysis, we propose a new scheme, which can efficiently handle block classification errors caused by prospective image editing, or re-compression.

Based on both theoretical analysis, and experimental validation, we show that such an approach can deliver an efficient self-embedding mechanism. Even when the tampering affects large image areas, it is possible to achieve high-quality reconstruction. The desired fidelity is controlled by a single parameter, and in contrast to existing schemes, is not affected by the watermark recovery errors. For the highest considered setting, the average PSNR reaches over 33 dB on a test-set of 10,000 natural images.

2. PROPOSED SCHEME

The considered application scenario is illustrated in Fig. 1. The encoder yields a protected JPEG image with quality factor Q_1. As a result of malicious tampering, the attacker yields a JPEG with quality Q_2, potentially different than Q_1.

Let I be the input image of size $w \times h$ px, divided into $4N$ blocks of size 8×8 px. Due to limited embedding capacity, the blocks are grouped into 16×16 px macro-blocks, which serve as authentication, and reconstruction units. The i-th macro-block is denoted as I_i. The embedding capacity is $4B + 2H$ bits per macro-block. The number of reference bits is $4\lambda B$ for all macro-blocks. Hence, the reconstruction quality is controlled by $\lambda \in \mathbf{N}^+$.

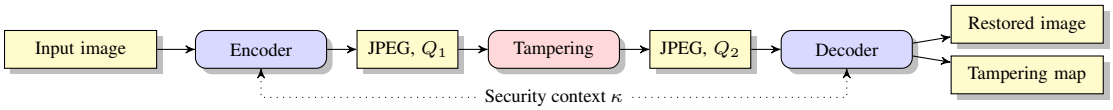


Fig. 1: Operation of the considered self-embedding scenario with prospective recompression to a quality factor $Q_2 \neq Q_1$.

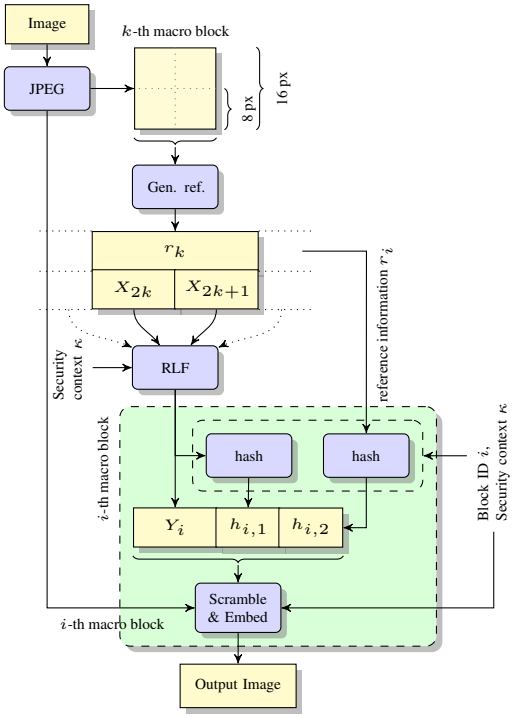


Fig. 2: Operation of the encoder for $\lambda = 2$.

2.1. Encoder

Operation of the encoder is shown in Fig. 2. The first step is to perform a standard JPEG compression with quality factor Q_1. The resulting JPEG is then used to generate the reconstruction reference. The reference information for the i-th macro-block is denoted as r_i, and consists of concatenated bit-streams for its corresponding image blocks. Each block is described by λB bits, allocated to individual coefficients according to an allocation matrix $\mathbf{P}_\lambda$. The component corresponding to the i-th coefficient is denoted as $\mathbf{P}_\lambda[i]$.

Let $\mathbf{E}_{Q_1}$ be a matrix of embedding capacity, and $\mathbf{D}_{Q_1}$ a matrix of maximal coefficient precision for Q_1. Then, the reference information for the i-th coefficient c_i can be extracted as a sign and $\mathbf{P}_\lambda[i]-1$ most significant bits from its $\mathbf{D}_{Q_1}[i]$-bit representation:

$$\text{round}\left(c_i \cdot 2^{\mathbf{P}_\lambda[i]-\mathbf{D}_{Q_1}[i]}\right). \tag{1}$$

Coefficients' magnitudes exceeding the precision defined by $\mathbf{D}_{Q_1}$ are saturated to $2^{\mathbf{D}_{Q_1}[i]}-1$. In order to ensure that the extractable reference information is identical after watermark embedding, the following condition needs to be satisfied:

$$\forall_i\ \mathbf{P}_\lambda[i] + \mathbf{E}_{Q_1}[i] \leq \mathbf{D}_{Q_1}[i]. \tag{2}$$

The complete reconstruction reference is then divided into $4B$-bit symbols $X_k : k \in \{1, \ldots, \lambda N\}$. A random linear fountain (RLF) code produces same-length embedding symbols $Y_i : i \in \{1, \ldots, N\}$ for N macro-blocks. Watermark symbols are then obtained by appending two H-bit hashes:

$$h_{i,1} = \text{hash}(Y_i, \kappa, i), \tag{3a}$$
$$h_{i,2} = \text{hash}(r_i, \kappa, i). \tag{3b}$$

The dual-hash mechanism improves the reconstruction performance by enabling discrimination between corrupted block payload and content (Section 3).

The final step is to scramble and embed the individual watermark symbols into their corresponding macro-blocks. In order to embed a message $m \in \{0, \ldots, 2^{\mathbf{E}_{Q_1}[i]}-1\}$, the coefficients of the originally produced JPEG file are modified according to:

$$\hat{c}_i = \text{round}\left(c_i \cdot 2^{-\mathbf{E}_{Q_1}[i]}\right) 2^{\mathbf{E}_{Q_1}[i]} - 2^{\mathbf{E}_{Q_1}[i]-1} + m, \tag{4}$$

which can be seen as a variant of bit substitution or quantization index modulation. Depending on the desired embedding strength, the coefficients might be bit-wise shifted before and after embedding. We use a 1-bit shift for $Q_1 \geq 93$. The embedding locations are defined individually for various quality levels Q_1. Based on the desired embedding capacity, the matrices $\mathbf{E}_{Q_1}$ are derived from a base embedding map $\mathbf{E}$ by discarding the coefficients most vulnerable to rounding errors.

2.2. Decoder

Operation of the decoder is shown in Fig. 3. The first step is to extract the watermark. For each watermarked coefficient c_i the embedded message m is extracted according to:

$$m = \hat{c}_i - \text{round}\left(\hat{c}_i \cdot 2^{-\mathbf{E}_{Q_1}[i]}\right) 2^{\mathbf{E}_{Q_1}[i]} - 2^{\mathbf{E}_{Q_1}[i]-1}. \tag{5}$$

The extracted symbols are then demultiplexed to yield the embedding payload $\hat{Y}_i$, and the hashes $\hat{h}_{i,1-2}$. Simultaneously, macro-blocks' reference information is regenerated. Both hashes are then recalculated, and compared with their extracted counterparts. The resulting erasure and tampering maps identify image blocks which need to be restored, and watermark symbols which can be used for the restoration.

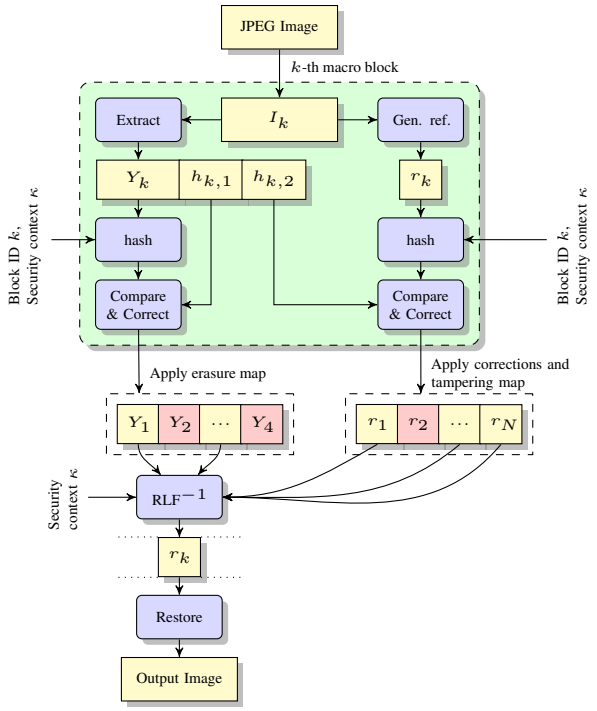


Fig. 3: Operation of the self-embedding decoder.

Due to prospective coefficient rounding errors resulting from re-compression, a compensation step is employed. A pre-calculated error map indicates the coefficients, which are the most vulnerable during each possible re-compression. If a block is deemed tampered, the decoder attempts to match the hashes for a number of most likely rounding errors. We allow for ± 1 changes in the coefficient magnitudes. Analogous compensation is used for the watermark payload, where various bit-flip combinations are considered. Additionally, since the hashes differ significantly even for the slightest change in the input data, it is beneficial to allow for a certain number of erroneous bits during hash comparison, provided that their locations match the most probable rounding errors.

The number of trails should be chosen accordingly to the desired false negative classification rate. The issue is addressed in detail in Section 3. Alternatively, the most vulnerable coefficients can be skipped from watermark embedding at the cost of limiting the available watermark capacity.

The corrected reference information of authentic image blocks is then used to remove these dependencies from the correctly extracted embedding symbols. The resulting simplified RLF code is then decoded to yield the reference information of the tampered image fragments. Approximate original appearance is then restored using the recovered DCT coefficients.

3. THEORETICAL ANALYSIS

Due to coefficient rounding errors during prospective image editing, unintentional bit flips either in the blocks' reference information, or the embedded payload, make it possible for authentic image blocks to be misclassified as tampered. Such blocks would be restored in the decoder, and would unnecessarily limit the achievable tampering rates. Let $\tilde{\gamma} = 1 - \gamma$ denote the tampering rate, i.e., the number of maliciously modified authentication units. Given false positive probability f_p, the restoration condition becomes:

$$(1 - f_p)\gamma \geq \lambda(1 - \gamma) + \lambda\gamma f_p \tag{6a}$$

$$\gamma \geq \lambda(1 - f_p + \lambda(1 - f_p))^{-1}. \tag{6b}$$

The introduced dual-hash mechanism can distinguish tampered blocks from erased embedding symbols. If a blocks is authentic, yet contains invalid payload, it will not be reconstructed. Let p_e denote the watermark symbol error rate. Then, the reconstruction condition becomes:

$$(1 - p_e)\gamma \geq \lambda(1 - \gamma) + \lambda\gamma f_p \tag{7a}$$

$$\gamma \geq \lambda(1 - p_e + \lambda(1 - f_p))^{-1}. \tag{7b}$$

The false positive classification errors are significantly less frequent than watermark symbol errors, i.e., $f_p \ll p_e$. Hence, the dual-hash mechanism allows for higher tampering rates. Table 1 collects the theoretical tampering rate bounds for both a single and a dual-hash configuration. This theoretical result will be experimentally validated in Section 4.2.

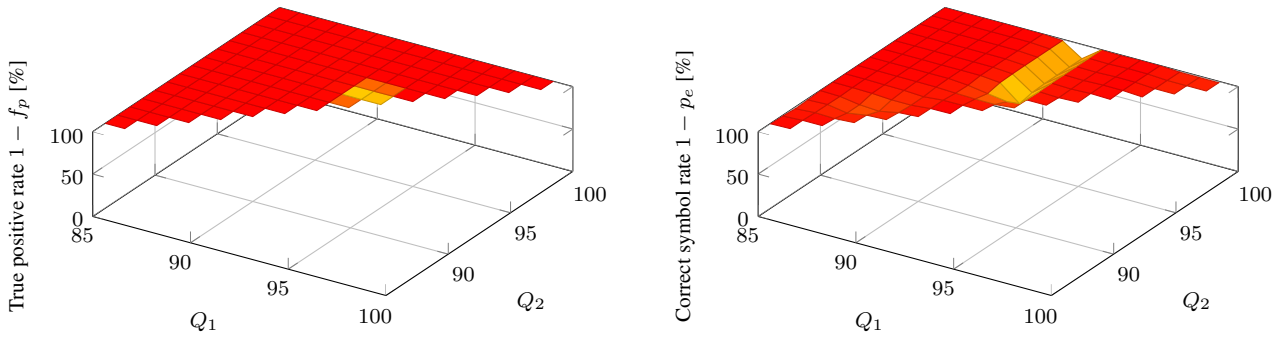
Table 1: Success bounds with block classification errors.

		$\tilde{\gamma}_{\max}$ **[%] for symbol error rate** p_e				
Mode	λ	0.0	0.01	0.05	0.10	0.15
Single-hash	1	50.0	49.5	47.4	44.4	41.2
Single-hash	2	33.3	32.7	29.8	25.9	21.6
Single-hash	3	25.0	24.2	21.1	16.7	11.8
Single-hash	4	20.0	19.2	15.8	11.1	5.9
False positive rate f_p =0.05						
Dual-hash	1	48.7	48.5	47.4	45.9	44.4
Dual-hash	2	31.0	30.8	29.8	28.6	27.3
Dual-hash	3	22.1	21.9	21.1	20.0	18.9
Dual-hash	4	16.7	16.5	15.8	14.9	14.0

False negative classification errors occur when a tampered block is by chance deemed authentic. The primary factor, which influences the collision probability f_0 is the length of the hash, i.e., $f_0 \approx 2^{-H}$. The introduced hash tolerance and compensation mechanism increases the effective collision probability. By proper selection of the compensation parameters it is possible to maintain the desired error rate.

The compensation mechanism attempts to perform the most likely ± 1 adjustments of the coefficients' values. Given that up to d_c coefficients out of d_m most probable ones can be corrected at once, the number of tested combinations is:

$$n_l = \sum_{i=1}^{d_c} \binom{d_m}{i} 2^i \tag{8}$$



(a) average rate of true positives

(b) average rate of correctly extracted symbols

Fig. 4: Impact of re-compression on the authentication and watermark extraction performance ($\lambda = 1$).

Once the compensation attempts fail, the decoder compares the Hamming distance between the hashes against a threshold d_h. The number of possible valid hashes is:

$$\sum_{i=1}^{d_h} \left(\begin{array}{c} H \\ i \end{array} \right) \tag{9}$$

Finally, the false negative probability can be estimated from the Bernoulli trials:

$$f_n \approx 1 - (1 - f_0)^{n_l} + (1 - f_0)^{n_l} \sum_{i=1}^{d_h} \left(\begin{array}{c} H \\ i \end{array} \right) f_0 \tag{10}$$

4. EXPERIMENTAL EVALUATION

For the purpose of experimental evaluation of the proposed approach, we implemented the described scheme in Matlab. The experiments were performed on 512×512 px natural gray-scale images from the BOWS2 data set [7].

We use $H = 24$ bit hashes, and embedding symbols of length $4B = 96$. In each 8×8 px block, we embed 36 bits, divided into $6 + 6$ bits for the hashes $h_{i,1-2}$, and 24 bits for the reconstruction reference. Hence, the amount of reference information per macro-block is 96λ bits.

4.1. Block Classification Errors

The goal of this experiment is to assess the rate of false positive classification errors, i.e., how often authentic, but re-compressed image blocks are deemed as tampered. The first step is to produce a protected image with quality factor $Q_1 \in [85; 100]$. After re-compression to $Q_2 \in [Q_1; 100]$, the decoder attempts to authenticate and reconstruct the image. The experiment is repeated with various seeds for the PRNG, and 45 representative natural images. Fig. 4 shows the average rates of correctly classified blocks and extracted watermark symbols. The highest observed false classification rate f_p is 0.15%. The highest observed symbol error rate p_e is 21.5%.

To validate the theoretical estimate of the false negative probability (10), we tested 6,243 unwatermarked images. Since $H = 24$, the rank of the principal false negative rate is $\log_{10} f_0 = -7.22$. Compensation of the reconstruction reference uses $d_m = 24; d_c = 2$, and from (10) the rank of the false negative rate increases to $\log_{10} f_n = -4.16$. The compensation has dominant influence on f_n, and on the basis of (10), we can allow for $d_h = 2$ without significantly deteriorating f_n. Then, the expected $\log_{10} f_n = -4.06$. From the total of 6,392,832 blocks, exactly 608 were classified as authentic. Hence, the empirical false negative classification rate falls into range $\log_{10} f_n \in [-4.058; -3.989]$ with 95% confidence.

The payload compensation mechanism allows to consider $d_m = 32$ most probable coefficients. Hence, after allowing for up to $d_h = 2$ different bits in the hash vectors, the rank of the false negative rate increases to $\log_{10} f_n = -3.854$. Exactly 834 blocks were identified to carry a valid watermark payload. Hence, the empirical false negative classification rate falls into range $\log_{10} f_n \in [-3.915; -3.856]$ with 95% confidence.

4.2. Reconstruction Success Bound Validation

The goal of this experiment is to confirm the theoretical success bound (7b). The encoder produces a protected image with $Q_1 = 87$ and $\lambda = 2$. Then, the image is randomly tampered, and re-compressed to $Q_2 = 90$. We measure the number of successful reconstruction attempts for increasing tampering rates. The experiment is repeated 600 times for each tampering rate; each time with a different PRNG seed. By performing only re-compression, we obtain a more accurate estimate of the applicable error rates: $f_p = 0.003$, and $p_e = 0.104$. Hence, from (7) the expected success bound is $\tilde{\gamma} = 0.308$. Fig. 5 shows the measured reconstruction success rate vs. the tampering rate.

4.3. Image Quality

The embedding distortion depends mainly on Q_1, since the JPEG quantization table determines the embedding strength.

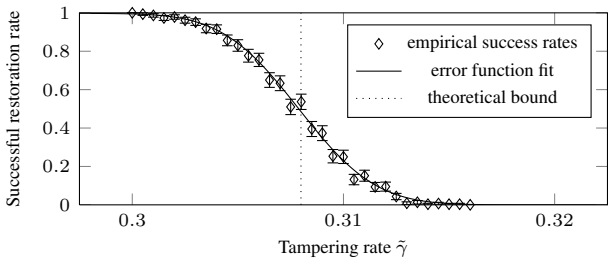


Fig. 5: Successful reconstruction attempts for various tampering rates in the presence of recompression; error bars correspond to 95% confidence intervals; theoretical success bound is shown with a dotted line.

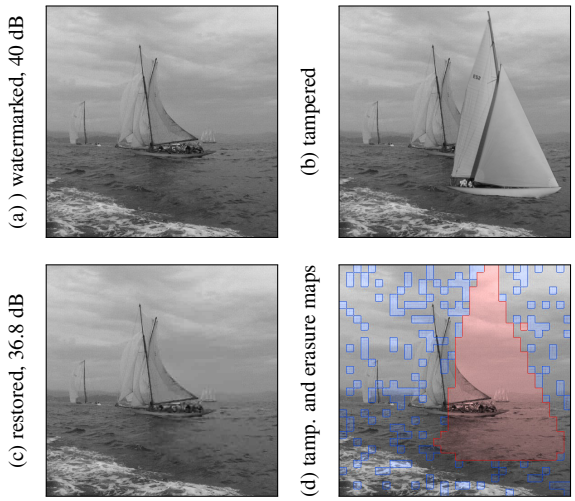


Fig. 6: Reconstruction example with $Q_1 = 90, \lambda = 2$; tampering rate $\tilde{\gamma} = 0.23$, re-compression to $Q_2 = 92$.

The PSNR, with respect to an unwatermarked Q_1 JPEG grows from 34 dB to 40 dB with increasing Q_1. Due to higher embedding strength, the behavior repeats for $Q_1 \geq 93$.

The reconstruction quality depends mainly on the reference rate λ. Table 2 shows the reconstruction PSNR scores for 10,000 natural images from the BOWS2 data set. Fig. 6 shows an example reconstruction result. The original image is encoded with $\lambda = 2$ to $Q_1 = 90$. The resulting watermarked image (a) has PSNR=40 dB, compared to a generic Q_1 JPEG. The image is then tampered, and recompressed to $Q_2 = 92$ (b). The decoder can successfully recover the approximate appearance of the tampered fragments. The restoration result (c) has PSNR=36.8 dB, and 32.7 dB in the restored regions only, compared to a $Q = 100$ JPEG. The detected tampering and erasure maps are shown in (d) in red and blue, respectively.

Table 2: Reconstruction PSNR for 10,000 natural images.

PSNR	λ =1	λ =2	λ =3	λ =4
Mean	27.8 dB	30.0 dB	31.7 dB	33.2 dB
Quantile 0.9	32.4 dB	35.2 dB	36.9 dB	38.0 dB
Quantile 0.1	23.5 dB	25.4 dB	27.0 dB	28.6 dB

5. CONCLUSIONS

In conclusion, by introducing additional mechanisms for dealing with block classification and watermark recovery errors, the content reconstruction model proposed in [6] allows for constructing efficient self-recovery schemes for lossy-compressed images. Existence of both error types can be easily incorporated into the theoretical model, and the expected reconstruction performance can be calculated analytically.

Compared to existing JPEG-compatible schemes, our approach behaves differently. Erroneous portions of the watermark are not used for reconstruction. Instead of introducing restoration artifacts, emerging errors limit the supported tampering rates without affecting the restoration fidelity.

6. ACKNOWLEDGMENT

The research leading to these results has received funding from the European Regional Development Fund under INSIGMA project no. POIG.01.01.02-00-062/09.

7. REFERENCES

[1] J. Fridrich and M. Goljan, "Images with self-correcting capabilities," in *Proc. of IEEE Int. Conf. on Image Processing*, 1999.

[2] X. Zhu et al., "A new semi fragile image watermarking with robust tampering restoration using irregular sampling," *Signal Processing : Image Communication*, vol. 22, no. 5, 2007.

[3] A. Cheddad et al., "A secure and improved self-embedding algorithm to combat digital document forgery," *Signal Process.*, vol. 89, pp. 2324–2332, December 2009.

[4] C-Y Lin and S-F Chang, "Sari: self-authentication-and-recovery image watermarking system," in *Proc. ACM Int. Conf. on Multimedia*, 2001, pp. 628–629.

[5] H. Wang et al., "A novel fast self-restoration semi-fragile watermarking algorithm for image content authentication resistant to jpeg compression," in *Proc. of Int. Conf. on Digital-Forensics and Watermarking*, 2012, pp. 72–85.

[6] P. Korus and A. Dziech, "Efficient method for content reconstruction with self-embedding," *IEEE Trans. on Image Process.*, vol. 22, no. 3, 2013.

[7] "Bows2 dataset," `http://bows2.ec-lille.fr/`.