# Errorless Robust JPEG Steganography for Pixel Images

Jan Butora, Etienne Levecque, Patrick Bas

# Errorless Robust JPEG Steganography for Pixel Images

Jan Butora
*Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL* Lille, France
jan.butora@cnrs.fr

Etienne Levecque
*Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL* Lille, France
etienne.levecque@cnrs.fr

Patrick Bas
*Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL* Lille, France
patrick.bas@cnrs.fr

***Abstract*—An antecedent of a JPEG image is any pixel-valued image that gets compressed to this JPEG image. This paper presents a novel robust JPEG steganography method based on a JPEG antecedent search of a given stego image. This method can be used whenever Alice is able to upload on an online platform a pixel-valued stego image and when the platform compresses the image. Since the antecedent search can be in some cases rather long for specific JPEG blocks, we heuristically restrict the number of search steps per block to a given threshold. Nevertheless, we show that even with this limitation, we can achieve errorless robustness against JPEG compression for payloads as big as 0.5 bpnzac for various quality factors. Not only can we ensure errorless robustness for uncompressed and compressed images, but we also show that the proposed robustness update of a given steganographic algorithm increases its undetectability. We then verify that this security improvement comes from preventing embedding changes in DCT blocks with saturated pixels. Finally, we demonstrate that the proposed method outperforms previous state-of-the-art robust steganography in terms of security with feature-based and deep-learning detectors.**

***Index Terms*—JPEG robustness, errorless, steganography, JPEG antecedent**

## I. Introduction

Communicating privately is becoming increasingly important not only for governmental entities but also for individuals across various spheres. Steganography, the art of concealing messages within other media, is one suitable tool for this purpose. While significant emphasis has been placed in recent decades on making steganography undetectable, its major drawback is the absolute lack of robustness. In the context of digital images, this means that the steganographic message cannot be retrieved after typical post-processing operations, such as downscaling or JPEG compression, which severely limits the practical applicability of steganography in real-world scenarios, necessitating greater efforts to enhance its robustness. In this paper, we propose a modification of an existing steganographic algorithm, which makes it robust against subsequent JPEG compression without any additional post-processing operations.

### A. Contrast with prior art

Recent years have witnessed some advancements in the field of steganography resilient against JPEG compression. In [1], Cleaves and Ker have studied the impact of lossy transmission combined with syndrome trellis code (STC) [2], demonstrating that the STC replicates the errors associated with the channel on the decoded payload. They propose a dual-STC scheme combined with Reed-Solomon Error Correcting Codes to reduce the error rate while minimizing the embedding distortion, although this scheme has not been benchmarked against steganalysis.

Other methods such as Sign Steganography Revisited (SSR) [3] or MINImizing Channel Error Rate (MINICER) [4] prioritize robustness over security, making steganography much more detectable. Moreover, while both SSR and MINICER, could potentially provide small bit error rates in some limited scenarios, it was shown [5] that in a practical setting, they are, in fact, not robust. Given the critical nature of steganography, the scheme must exhibit extremely high robustness, as a single erroneous bit can jeopardize the entire transmission since the payload is most of the time encrypted. Therefore, errorless steganographic methods are recommended.

To the best of the authors' knowledge, the only errorless robust technique has been proposed by Butora *et al.* [5]. This method divides the image into 64 non-overlapping lattices and iteratively calls the JPEG compressor as an oracle to assess which embedding changes are robust. Although this approach provides errorless robustness, it is limited since the embedding must be done into a JPEG image compressed with the same quantization table as the one the robustness is targeted against. In other words, Bob can receive only images compressed twice with the same quantization table.

In contrast, our proposed method takes a different approach, where a pixel-valued image is created and mapped through the compression to the desired stego image. This is achieved by embedding into the compressed cover image and searching for its JPEG antecedent (see Section II-A), which is a slightly modified version of the decompressed stego image. This fundamentally distinguishes our method from [5], which allows the steganographer to only send JPEG files. The scheme of the proposed method is illustrated in Figure 1. Additionally, the authors note that the precover image does not need to be uploaded to an online platform if the JPEG compressor is known, as the upload can be simulated offline, avoiding unnecessary suspicion. Having access to the JPEG compressor

*WIFS'2024, December, 2-5, 2024, Rome, Italy. XXX-X-XXXX-XXXX-X/XX/$XX.00* 

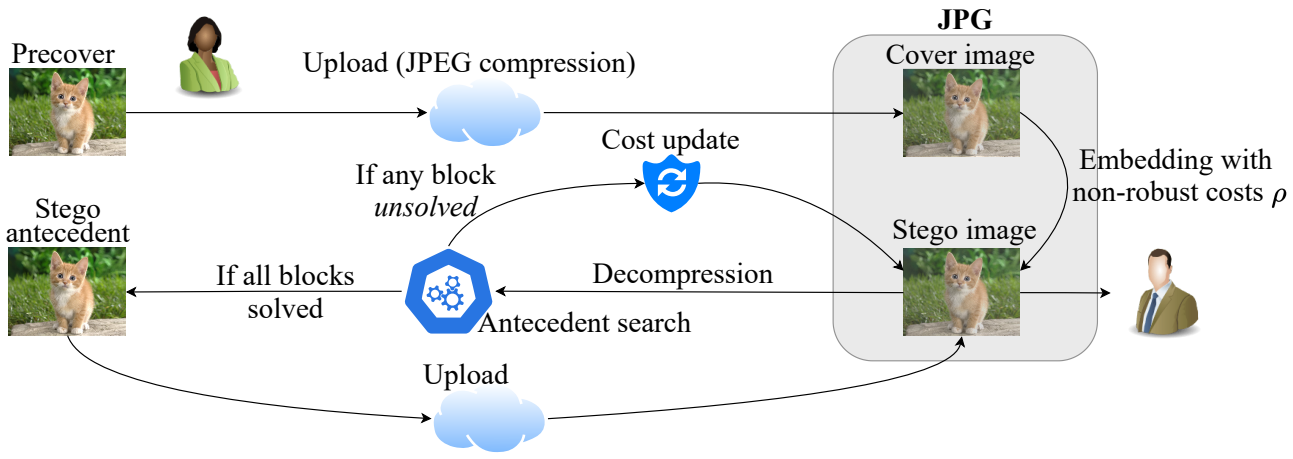


Fig. 1: The proposed errorless JPEG robust steganography scheme. Alice uploads her precover image (uncompressed or compressed) to an online platform, which JPEG compresses the image to the cover image. Alice downloads the cover image and embeds into it a message with given steganographic embedding costs, thus creating the stego image. She then finds the stego antecedent and uploads it to the platform, which compresses it to the desired stego image. Note that in the scenario considered in the paper, Alice has access to the JPEG compressor and thus does not have to upload the image online.

is the main requirement of our method, as searching for the antecedent requires potentially many calls to the compressor.

### B. Outline of the paper

In the next section, we introduce the JPEG antecedents and how they are used to build a robust steganographic scheme. Section III describes the experimental setup, Section IV lays out the paper's main results, and the paper is concluded in Section V.

## II. Proposed method

In the following, we introduce the JPEG antecedents, how to search for them, and how to use them to create an errorless robust steganography method from existing steganographic costs.

### A. JPEG Antecedent

A *cover* image is a natural image compressed with a specific JPEG Quality Factor (QF). Note that the cover image can be a single-compressed JPEG or an image compressed multiple times. This is the image that gets uploaded to the platform when Alice does not use steganography. A *stego* image is a modified version of the cover image carrying the steganographic message aimed for Bob. A *stego antecedent* is then a pixel-valued image (uncompressed or decompressed JPEG) that gets JPEG compressed to the desired stego image. The pixel-valued image that created the cover image is referred to as the *precover*, which is one specific antecedent of the cover image.

Note that in the current application scenario, Alice does not have to practically upload her cover image, which could create suspicion if Eve is able to monitor the platform. Instead, having access to the JPEG compressor, she can simulate the JPEG compression on her side. This is the main requirement of the proposed method, but it is not too constraining as it is reasonable to assume that many online platforms use standard JPEG compression libraries, such as `libjpeg`.

The JPEG antecedents were originally studied for steganalysis of JPEG compressed images at QF 100 with no false positives [6], where it was shown that some DCT blocks have no pixel antecedents and are thus manipulated.

### B. Antecedent Search

The antecedent search algorithm is explained in detail in [7] but for better clarity, we will provide a brief overview of the method in this paragraph. The main requirement for finding a correct antecedent is to have access to the compressor, even if only in a black-box framework. Let $\mathcal{C}and\mathcal{D}$ denote respectively the JPEG compressor of interest and a decompressor function, where the decompression is done with the mathematical definition of the 2D Inverse Discrete Cosine Transform (IDCT) and the result is rounded and clipped to the dynamic range of 8-bit images $[0, 255]$. Furthermore, let $\mathbf{q}$ be the quantization table used during the compression. The algorithm described in Algorithm 1 is applied independently to each $8 \times 8$ block of the image.

The algorithm can be divided into two parts: the initialization and the exploration loop. In the initialization phase, we decompress the stego block $\mathbf{c}$ to obtain a first pixel-block candidate $\mathbf{x}_s = \mathcal{D}(\mathbf{c}, \mathbf{q})$. To verify if this initial candidate is a correct antecedent, we apply the compressor to it. If the result equals the stego block, $\mathcal{C}(\mathbf{x}_s, \mathbf{q}) = \mathbf{c}$ the algorithm has found an antecedent that will be compressed to the stego block and the search terminates.

If the output does not match the stego block, we have an initial pixel-block candidate that is not an antecedent, and

**Algorithm 1** Local search to find antecedent

**Require:** $\mathbf{c}, \mathbf{q}$ {Stego block and quantization table}
**Require:** $\mathcal{C}, \mathcal{D}$ {Compressor and decompressor}
**Require:** $N > 0$ {Max iteration}
  $\mathbf{x}_s \leftarrow \mathcal{D}(\mathbf{c})$ {Starting pixel-block of the search}
  add $\mathbf{x}_s$ to $P$ with cost 0 {Priority queue initialization}
  $k \leftarrow 0$ {Search step initialization}
  **while** $P$ not empty **and** $k \leq N$ **do**
    $k \leftarrow k + 1$
    $\mathbf{x} \leftarrow$ remove first element of $P$
    **for** $\mathbf{x}_n$ in $neighbors(\mathbf{x})$ **do**
      **if** $\mathbf{x}_n$ has not been visited **then**
        **if** $\mathcal{C}(\mathbf{x}_n; \mathbf{q}) = \mathbf{c}$ **then**
          **return** $\mathbf{x}_n$ {$\mathbf{x}_n$ is an antecedent of $\mathbf{c}$}
        **end if**
        $\gamma_n \leftarrow \|\mathbf{c} - \mathcal{C}(\mathbf{x}_n; \mathbf{q})\|_1$
        add $\mathbf{x}_n$ to $P$ with cost $\gamma_n$ {$\mathbf{x}_n$ is a new candidate}
      **end if**
    **end for**
  **end while**
  **return** *unsolved* {Antecedent was not found in $N$ steps}

we proceed to the exploration loop. For a given number of iterations, we make $\pm 1$ changes to every position of the candidate. This creates 128 new candidates (2 changes for each 64 position in the block), and each one is compressed using the compressor. Then, for each candidate, we compute the $\ell 1$-norm between the compressed candidate and the stego block: $\gamma = \|\mathbf{c} - \mathcal{C}(\mathbf{x}_n; \mathcal{Q})\|_1$. If this distance is 0, we have found an antecedent and terminate the search. Otherwise, we select the candidate with the smallest distance $\gamma$ and go to the next iteration. The algorithm either finds an antecedent within the prescribed number of search steps $N$ or returns *unsolved* if the antecedent was not found.

Note that the algorithm is very similar to a path-finding algorithm such as A-star but differs from it because of the cross-domain search: we modify the pixel domain but the target is in the DCT domain.

Note also that although unsolvable blocks exist for JPEG Quality Factor (QF) 100 [7], the algorithm should be able to find an antecedent for any block compressed with lower QF given enough time (search steps). However, such a search can be potentially very time and energy-consuming, and we thus limit ourselves to $N = 10$ in this work, which can result in a larger number of *unsolved* blocks, even for lower QFs. This heuristic provided an acceptable trade-off between robustness and embedding speed.

### C. Errorless Robust embedding

Let $\boldsymbol{\rho}$ denote an $8 \times 8$ block of given embedding costs of some JPEG steganography algorithm associated with a stego DCT block $\mathbf{c}$. We describe in the following an efficient way of constructing the stego antecedent. Three distinct scenarios can arise for a given block:

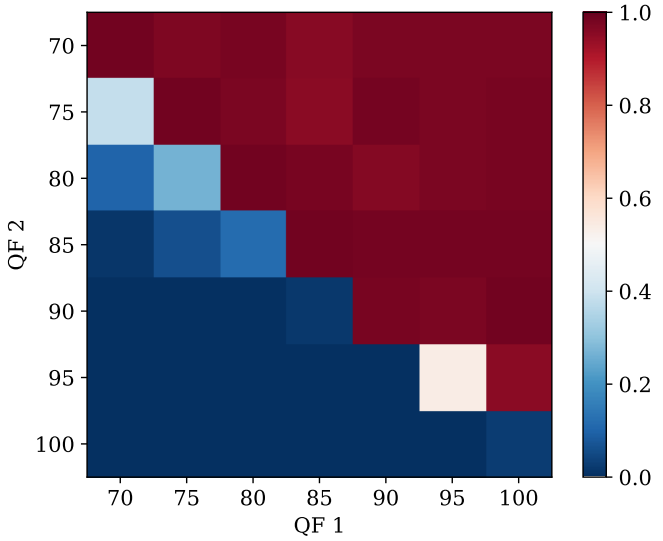


Fig. 2: Ratio of antecedents found with 10 steps of search in double-compressed JPEG images. Only embedded blocks are considered. The values are averaged over 100 randomly selected images, each of which was embedded with 5 payloads: 0.1,0.2,0.3,0.4, and 0.5 bpnzac.

1) The block was not modified during the embedding. In this case, we simply use the precover block, as it is already an antecedent.
2) The block has been modified and an antecedent was found. In this case, we use the found antecedent.
3) The block has been modified and antecedent was not found. In this case, we set all the embedding changes to wet costs [8], *i.e.* $\boldsymbol{\rho} \leftarrow \infty$.

If an antecedent is found for every block, then we have successfully found the stego antecedent and this is Alice's image she will be sending to the compressor (online platform). However, if any of the antecedents were not found, the image has to be re-embedded with the updated wet costs to avoid embedding into blocks for which it requires more than 10 steps of the search to find an antecedent. The described procedure is shown visually in Figure 1.

Even though this may seem like a severe constraint of the embedding algorithm, we observed in practice that the number of *unsolved* blocks decreases rapidly during every re-embedding, making up to 5 re-embeddings for every image. Moreover, due to the content-adaptive nature of modern steganography, in many blocks, the same exact embedding changes are preferred (and performed), which further speeds up the embedding process, as their antecedents have already been found. This also holds when embedding with the STCs since freezing some coefficients is equivalent to preferring certain paths through the trellis over others in a small local neighborhood of the embedding path.

It is worth mentioning that the embedding capacity consequently decreases and a desired message could potentially not be possible to embed with the updated costs, but we did not observe this behavior for the considered embedding payloads.

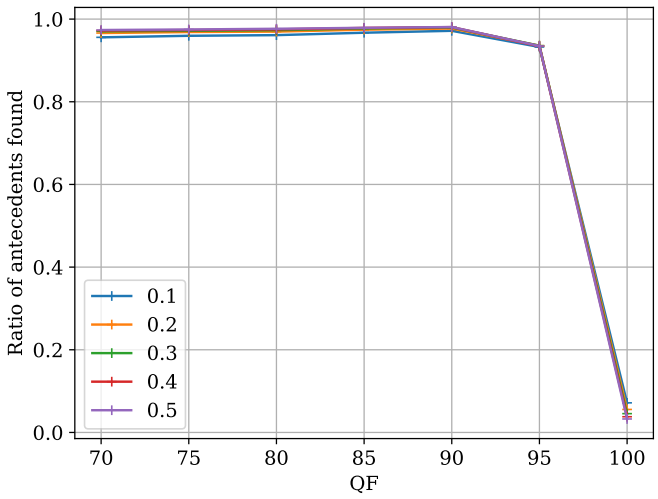


Fig. 3: Robust set size with 10 steps of antecedent search in single-compressed JPEG images. Only embedded blocks are considered. The values are averaged over 100 randomly selected images and results for 5 payloads (bpnzac) are shown.

## III. Experimental Setup

To experimentally verify our robust methodology and its impact on steganographic security, we use $10,000$ uncompressed grayscale images of size $512 \times 512$ from the BOSSBase dataset [9]. To generate single-compressed and double-compressed cover JPEG images, we use a publicly available python implementation[1] of `libjpeg`.

For the steganographic algorithm, we picked UERD [10] due to its embedding speed. We will refer to its robust version using the proposed methodology as R-UERD. We simulate the embedding with optimal coding for different payloads 0.1, 0.2 0.3, 0.4, and 0.5 bits per non-zero AC DCT coefficient (bpnzac).

To evaluate security, we report the probability of error under equal priors $P_{\text{E}}$. On one hand, we use the DCTR features [11] with low-complexity linear classifier [12] using balanced training and testing image splits. On the other hand, we use the SRNet [13] pre-trained on steganalysis in the ImageNet database [14]. The training, validation, and testing splits for the deep learning detector are of size 7k, 1k, and 2k, respectively. The detector has been refined for every QF and payload separately by training for 50 epochs with the rest of the hyper-parameters set as in [14].

## IV. Results

### A. Antecedents and Robustness

First, we investigate the robustness in double-compressed images, re-compressed with the same quality, as a direct comparison can be made to the robust method in [5]. We only investigate the robustness in embedded blocks, as for the non-embedded blocks, the antecedent is trivially found from the precover image.

[1]https://gitlab.cristal.univ-lille.fr/elevecqu/incompatible-jpeg-blocks

| QF | Method | Emb. rate (bpnzAC) 0.1 | 0.2 | 0.3 | 0.4 |
|---|---|---|---|---|---|
| 75 | R-UERD | **0.5** | **0.4138** | **0.3258** | 0.1693 |
| | UERD | 0.4998 | 0.4074 | 0.3165 | **0.2274** |
| 95 | R-UERD | **0.5** | 0.4995 | **0.4311** | **0.3125** |
| | UERD | **0.5** | **0.5** | 0.4077 | 0.2983 |

TABLE I: Detection error of DCTR features on single compressed images.

| QF | Method | Emb. rate (bpnzAC) 0.1 | 0.2 | 0.3 | 0.4 |
|---|---|---|---|---|---|
| 75 | R-UERD | **0.1322** | **0.0503** | **0.0198** | **0.0098** |
| | UERD | 0.1055 | 0.0370 | 0.0155 | 0.0065 |
| 95 | R-UERD | **0.2800** | **0.1450** | **0.0870** | **0.0470** |
| | UERD | 0.2290 | 0.1225 | 0.0702 | 0.0330 |

TABLE II: Detection error of SRNet on single compressed images.

We can observe on the diagonal in Figure 2 that the robust set size for double-compressed images drops to roughly 50% at QF 95, which is in line with the results obtained in [5] (see Fig. 7). Moreover, we see that the robust set size is very close to 100% whenever the quality factor of the first compression is higher than the quality of the subsequent compression, $QF_1 > QF_2$. We explain this by higher granularity in a high-quality, single-compressed image which allows us to find correct pixel modifications in the antecedent causing desired effect in the DCT domain of the re-compressed image.

On the contrary, the small robust set size for cases with $QF_1 < QF_2$ are of no practical interest due to their extreme detectability with the Reverse JPEG compatibility attack [15].

For the single compressed images, we see in Figure 3 that the robust set sizes are very similar to the double-compressed case (re-compression with the same QF) with an exception at QF 95 where the robust set size is around 93.5%. Once again, we conclude that higher-quality images are more suitable for finding the antecedents.

These results are likely susceptible to the maximum number of the antecedent search steps $N$, and we plan to further study its effect in the future.

### B. Security Evaluation

The detection results of DCTR for the robust R-UERD and non-robust UERD in single-compressed images are shown in Table I. Surprisingly, we can notice that the robust version is, in fact, in many cases less detectable than the original UERD. This is even more obvious in Table II with the results of SRNet, where the robust version is always more secure, up to 5% in terms of $P_{\text{E}}$ for 0.1 bpnzac at QF 95.

A similar conclusion can be made for double-compressed images after inspecting the results in Tables III and IV, where the security gain is negligible with DCTR and reaches up to 3% with SRNet. Moreover, we see that compared to the other robust method [5], our method is consistently less detectable with SRNet, up to 2.8% for the highest payloads and QF 95. This is an interesting result, as our method is faster and allows

| QF | Method | Emb. rate (bpnzAC) 0.1 | 0.2 | 0.3 | 0.4 |
|---|---|---|---|---|---|
| 75 | R-UERD | 0.4658 | **0.4006** | 0.3101 | **0.1669** |
| | [5] | **0.4990** | 0.3968 | **0.3200** | 0.1620 |
| | UERD | 0.5000 | 0.4116 | 0.3362 | 0.1612 |
| 95 | R-UERD | **0.5000** | 0.4761 | **0.4392** | **0.3102** |
| | [5] | 0.4999 | **0.4956** | 0.3426 | 0.2141 |
| | UERD | 0.5000 | 0.4705 | 0.4329 | 0.3780 |

TABLE III: Detection error of DCTR features on double-compressed images with the same quality factor. The best values for robust algorithms are put in bold.

| QF | Method | Emb. rate (bpnzAC) 0.1 | 0.2 | 0.3 | 0.4 |
|---|---|---|---|---|---|
| 75 | R-UERD | 0.1273 | **0.0470** | **0.0193** | **0.0070** |
| | [5] | **0.1328** | 0.0445 | 0.0180 | 0.0068 |
| | UERD | 0.1163 | 0.0415 | 0.0173 | 0.0050 |
| 95 | R-UERD | **0.2767** | **0.1470** | **0.0810** | **0.0453** |
| | [5] | 0.2745 | 0.1283 | 0.0593 | 0.0170 |
| | UERD | 0.2438 | 0.1268 | 0.0715 | 0.0390 |

TABLE IV: Detection error of SRNet on double compressed images with the same quality factor. The best values for robust algorithms are put in bold.

Alice to send uncompressed images, but also to combine different quality factors for double-compressed images.

### C. Why robustness increases security?

In this section, we are going to explain the peculiar security gain through robustness. First, let us remind the reader that given enough antecedent search steps, the R-UERD should be equal to UERD, as antecedents exist, at least for single-compressed images with a quality factor below 100. This means, that by avoiding embedding into blocks for which we have not found antecedents fast enough, we improve security. We hypothesize that these blocks are containing saturated pixels, as the search algorithm clips the saturated pixels into the range $[0, 255]$, as explained in Section II-B, which potentially creates complications for the search algorithm.

To verify this hypothesis, we embedded 10 randomly selected images at QF 95 with payload 0.1 bpnzac and inspected the blocks that were tagged by the search algorithm as *unsolved*. We obtained this way 883 blocks and computed a per-block minimum and maximum of their 64 pixels. The results, visualized in Figure 4, show that 50% of all these blocks contain at least one saturated pixel (whether at 0 or 255). This is in line with our hypothesis that the increased detectability of UERD is due to the embedding into blocks with saturated pixels.

To verify this hypothesis even further, we embedded our database with UERD but set all the embedding costs in all blocks that have any saturated pixels (0 or 255) to wet costs to prevent embedding in them. We only conducted this experiment on single compressed images with QF 95 embedded with payloads 0.1, 0.2, 0.3, and 0.4 bpnzac. The detection error rates with SRNet are shown in Table V.

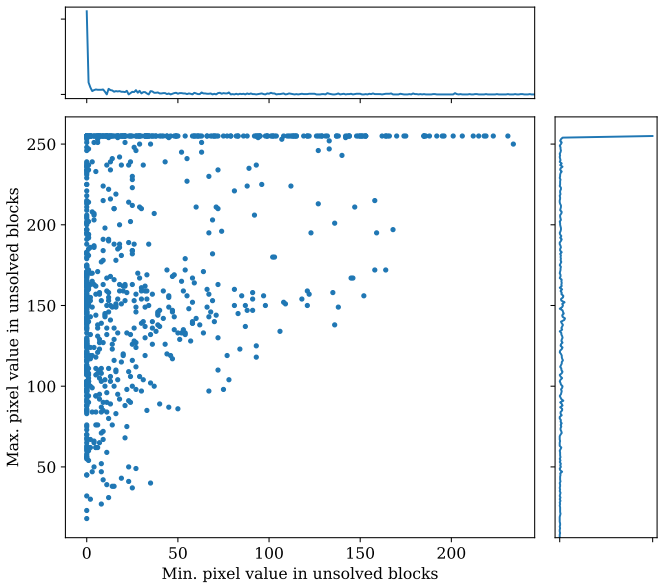


Fig. 4: Minimum and maximum pixel value of 883 *unsolved* blocks across 10 randomly selected image. Embedding was done on QF 95 compressed images with 0.1 bpnzac. Roughly 50% of all the blocks contain at least one saturated pixel.

Comparing these to the results on R-UERD, we can observe two things. One, this updated UERD algorithm improves the security even further, outperforming it by more than $1\%$ in terms of $P_{\mathrm{E}}$ for the two smaller payloads, providing overall benefit over original UERD of up to $6.1\%$. Secondly, for the two larger payloads, R-UERD is as detectable as UERD without embedding into saturated blocks. We can thus safely conclude that this is indeed the main reason for the security improvement. Nevertheless, this method remains non-robust.

We find it surprising that avoiding embedding into saturated pixels has been already mentioned in the context of natural steganography [16], however modern spatial and JPEG domain steganographic algorithms do not prohibit embedding into saturated pixels. We plan to investigate further the effect of these saturated pixels on steganographic security in the future.

Note that while we used only a simulation of optimal coding instead of practical codes, such as the STCs [2], the proposed method does not rely on a specific coding algorithm as we only modify a subset of the embedding costs to wet costs and look for the antecedent after the stego modifications are performed. Consequently, our method is immediately usable with practical codes.

| QF | Method | Emb. rate (bpnzAC) 0.1 | 0.2 | 0.3 | 0.4 |
|---|---|---|---|---|---|
| 95 | UERD (no saturation) | **0.2900** | **0.1615** | 0.0848 | **0.0480** |
| | R-UERD | 0.2800 | 0.1450 | **0.0870** | 0.0470 |

TABLE V: Detection error of SRNet on single-compressed images embedded with UERD without embedding into blocks with saturated pixels and the robust UERD.

## V. Conclusions and perspectives

We have presented a novel errorless robust JPEG steganography method based on JPEG antecedents. The main requirement is that the JPEG compressor for which we want to be robust has to be available to the steganographer, even if only as a black box. Any JPEG steganographic method can be used to create a stego image and an antecedent search is then performed to find the antecedent - an image that gets compressed to the desired stego image. If this is not possible within an allocated time, embedding into blocks for which antecedents were not found is forbidden and the process is repeated.

Unlike previous robust methods, our approach allows embedding into single-compressed images but also combining different quality factors in the double-compression pipeline, as long as the second compression is not of higher quality than the first one, without suffering from any undesirable errors during the JPEG lossy compression.

It was observed that the proposed method is more secure than its non-robust version, which we then attributed to embedding into blocks with saturated pixels which is highly detectable.

In our future work, we plan to study the effect of saturated pixels on steganographic security in more detail for JPEG and uncompressed images. We will further expand the proposed method by using the side-information in the form of compression rounding errors, which requires white-box access to the JPEG compressor. Furthermore, we want to investigate the effect of different quality factors in the double-compression pipeline on steganographic security. Next, the effect of the maximum number of antecedent search steps on the robust set size and on the embedding time will be studied. Finally, we believe that the JPEG antecedents can be used to construct a steganographic method for double-compressed images recompressed with the same quality, resilient against the Reverse JPEG compatibility attack.

The code to reproduce the results in this paper is made available on https://janbutora.github.io/downloads/

## Acknowledgment

This work was granted access to the HPC resources of IDRIS under the allocation 2024-AD011012855 made by GENCI. This work was also supported by a French government grant managed by the Agence National de la Recherche under the France 2030 program, reference ANR-22-PECY-0011.

## References

[1] C. Kin-Cleaves and A. D. Ker, "Adaptive steganography in the noisy channel with dual-syndrome trellis codes," in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2018, pp. 1–7.

[2] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, September 2011.

[3] X. Wu, T. Qiao, Y. Chen, M. Xu, N. Zheng, and X. Luo, "Sign steganography revisited with robust domain selection," *Signal Processing*, vol. 196, p. 108522, 2022.

[4] K. Zeng, K. Chen, W. Zhang, Y. Wang, and N. Yu, "Improving robust adaptive steganography via minimizing channel errors," *Signal Processing*, vol. 195, p. 108498, 2022.

[5] J. Butora, P. Puteaux, and P. Bas, "Errorless robust JPEG steganography using outputs of JPEG coders," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–13, 2023.

[6] E. Levecque, P. Bas, and J. Butora, "Compatibility and timing attacks for jpeg steganalysis," in *Proceedings of the 2023 ACM Workshop on Information Hiding and Multimedia Security*, ser. IH&MMSec '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 29–35. [Online]. Available: https://doi.org/10.1145/3577163.3595093

[7] E. Levecque, J. Butora, and P. Bas, "Finding incompatible blocks for reliable JPEG steganalysis," *IEEE Transactions on Information Forensics and Security*, 2024, under review.

[8] J. Fridrich, M. Goljan, and D. Soukal, "Wet paper codes with improved embedding efficiency," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 1, pp. 102–110, 2006.

[9] P. Bas, T. Pevny, and T. Filler, "Bossbase," http://agents.fel.cvut.cz/stegodata/, May 2011.

[10] L. Guo, J. Ni, W. Su, C. Tang, and Y. Shi, "Using statistical image model for JPEG steganography: Uniform embedding revisited," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2669–2680, 2015.

[11] V. Holub and J. Fridrich, "Low-complexity features for JPEG steganalysis using undecimated DCT," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219–228, February 2015.

[12] R. Cogranne, V. Sedighi, T. Pevný, and J. Fridrich, "Is ensemble classifier needed for steganalysis in high-dimensional feature spaces?" in *IEEE International Workshop on Information Forensics and Security*, Rome, Italy, November 16–19, 2015.

[13] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, May 2019.

[14] J. Butora, Y. Yousfi, and J. Fridrich, "How to pretrain for steganalysis," in *The 9th ACM Workshop on Information Hiding and Multimedia Security*, Brussels, Belgium, June 21–25, 2021.

[15] J. Butora and J. Fridrich, "Extending the reverse JPEG compatibility attack to double compressed images," in *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Canada, June 6–11, 2021.

[16] P. Bas, "Steganography via cover-source switching," in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016, pp. 1–6.