Asynchronous DCT-Based Data-Hiding Robust to Cropping

Jean-Luc Toutant, William Puech, Christophe Fiorio

Asynchronous DCT-Based Data-Hiding Robust to Cropping

J.L. Toutant[1], W.Puech[1] and C. Fiorio[1],
[1] LIRMM Laboratory, UMR CNRS 5506, University of Montpellier II, France

ABSRACT

In this paper, we present a DCT-based data-hiding method robust to cropping. As it works in the DCT-domain, it is also robust to JPEG compression.

A cropping operation on a JPEG image leads to the application of an additionnal compression. So, we achieve robustness to several JPEG compressions by an original data-hiding process. The cropping robustness is then obtained by localizing the information in a particular region of interest (ROI): whereas no area of the ROI is removed, the integrity of the hidden data is preserved. As we choice the ROI centered on the main focus, removing the hidden-data leads to removing the interest of the image. The data-hiding process is finally not dependant on the image coordinates but on the ROI. It is applied on other blocks than those used by the JPEG compression and leads to an asynchronous method instead of a whole integrated one: firstly a data-hiding step and secondly a compression step.

1 INTRODUCTION

The steganography aims at hiding information in numerical documents like pictures. Until now, this was made by a global process over the entire image. It allows good results for invisibility since all the space is used, but not for the robustness. All transformations that alter or change the image space destroy the embedded data. Cropping operations is one of these, all the data remaining in the removed part are lost.

This paper presents a contextual data-hiding that overcomes such limitations: the information is localized in a ROI of the image and as long as its integrity is respected, so, the one of the message is.

JPEG compression is a standard for image file format. It seems to be necessary to withstand it. The combination of this requirement and the contextual data-hiding leads to an asynchronous method. Section 2 focus on the defintion of an appropriate DCT-based data-hiding method and Section 3 on the croppping robustness.

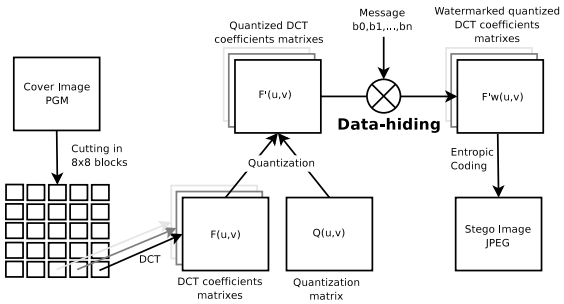


Figure 1: Integration of the data-hiding in the JPEG compression process.

2 DCT-BASED DATA-HIDING

With the development of network, compression became an important point for fast transmission. The JPEG compression format, based on the Discrete Cosine Transform (DCT), could be considered as the current compression standard for the images. The data-hiding methods should resist such transformations. Some of them do it, but our purpose need furthermore to be robust to cropping.

2.1 Existing methods

Existing methods, like JPEG/JSTEG [2] or data-hiding by modification of the quantization matrix [1, 3], work already in the DCT-based domain. They allow to obtain a JPEG stego-image for almost all the quality factors. The idea behind is to substitute one or several less significant bits (LSBs) of several quantized DCT coefficients for the bits of the secret message in the JPEG compression process as shown in Figure 1 (data-hiding step after the quantization).

A cropping operation over a JPEG Image leads to an additionnal JPEG compression. So we need to preserve the secret message even if the stego-image undergoes several successive JPEG compressions. Existing methods do not allow that.

Such requirement is hard to obtain with a low quality factor. Our main purpose is not the robustness to high compression rate, but the robustness to image cropping, so we can firstly take care only of a quality factor of 100%.

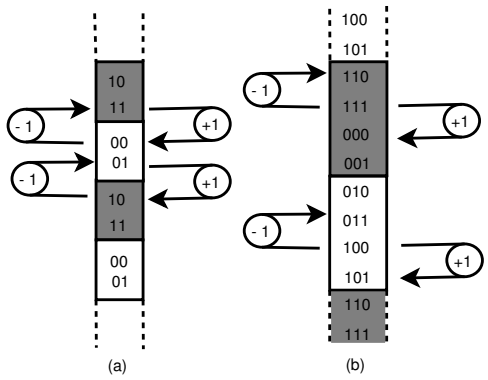


Figure 2: (a) Unstability of a secret bit coded by two bits, (b) Stability of a secret bit coding by three bits.

2.2 Substitution of the three less significant bits

The substitution of one bit of a quantized DCT coefficient by a secret bit leads to a state where a ± 1 variation of this coefficient alters the hidden information. With an additionnal compression such a variation takes place and this substitution does not allow the required robustness.

The idea is to encode the secret bit not only in one bit, but in several bits of the DCT coefficient as already done in the spacial domain [4]. The stego-image loose quality relatively to the cover image, but gain robustness against several JPEG compressions.

We still work on the LSBs to avoid too much visible change in the image. To use two bits for coding a secret bit does not resolve the problem. We have four possible values, so a small modification still leads to change the secret bit recovered value. To use three bits, eight values are possible. If we well choose, among these, the two values encoding the possible secret bit values, we can differenciate them, even if small variations take place, and deduce the hidden bit as show in Figure 2. There is a stability of the embedded bit relatively to a variation of ± 1.

With at least three bits for coding a secret bit, it resists small variations. If each coefficient of a block embed a bit, it leads in the worst case to a variation of above thirty grayscales for a particular pixel of the block. These changes are important and grow with the number of bits used, so we work only on three bits.

The presence of null coefficients is useful for compression but also for invisibility. They are less visible than low coefficient values in the DCT matrices as they play a role in the JPEG compression. So, we choose to substitute by 000 the three LSBs to code a secret bit 0 and by 100 to code a secret bit 1.

This can be expressed by:

$$F'(u,v) = \left[\frac{F'(u,v)}{8}\right] * 8 \pm 4 * b_w, \qquad (1)$$

and the extraction by:

$$\begin{cases} b_w = 0 & \text{if} \quad F'_w(u,v)\%8 \in [0,2[\cup[6,8[\\ b_w = 1 & \text{if} \quad F'_w(u,v)\%8 \in [2,6[\end{cases},$$

where $[x]$ is the closest integer value of x, $x\%y$ the rest of the integer division of x by y and b_w the value of the bit to be embedded.

The sign in Equation (1) depends on the value of the first right term:

- if $\left[\frac{F'(u,v)}{8}\right] > \frac{F'(u,v)}{8}$ then the sign is a minus,
- if $\left[\frac{F'(u,v)}{8}\right] < \frac{F'(u,v)}{8}$ then the sign is a plus,
- if $\left[\frac{F'(u,v)}{8}\right] = \frac{F'(u,v)}{8}$ then the sign is determinated randomly to preserve the variations mean close to zero.

The number of coefficients used by the data-hiding varies according to the size of the secret message and the available space. The quality of the stego-image relatively to the original one stays in an acceptable range even if a bit per coefficient is embedded (the PSNR comes down to 40 dB in this case). The selection of frequencies (low, high, or randomly chose frequencies) which embed bit does not really change the results.

Thus, we have achieved a necessary condition for the robustness to cropping of the data-hiding method, the conservation of the message integrity over several JPEG compressions.

3 ASYNCHRONOUS DATA-HIDING

In order to solve the problem of cropping robustness, we work with a contextual data-hiding: instead of applying the data-hiding to the whole image, we just take care of its revelant object. Thus, as long as no part of the ROI bounding this revelant object is removed, the secret message keeps its integrity.

The embedded data localization leads to two problems:

- the ROI could be too small to embed the message and need to be extended,
- the data-hiding can no more be integrated in the JPEG process, the method should be asynchronous.

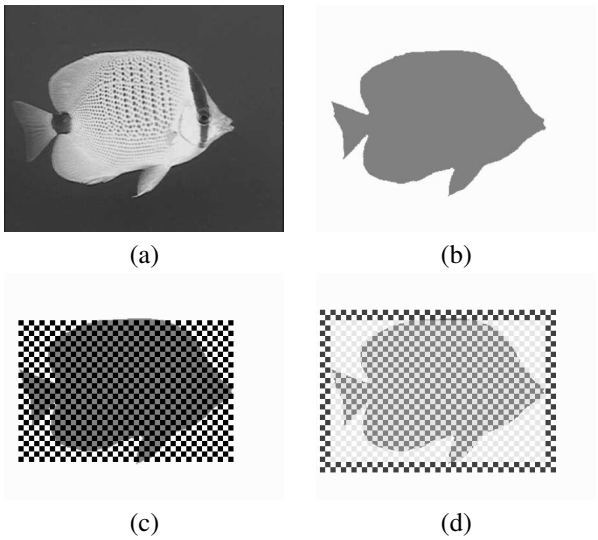


Figure 3: ROI definition and enlarging for embedding data: (a) original image, (b) segmentation, (c) ROI, (d) extended ROI.

3.1 Cropping Robustness

By segmentation, ROIs are obtained. In order to apply the data-hiding method previously presented, we need 8x8 blocks. So, each original ROI is reduced to a tiling of 8x8 blocks.

The ROI has a limited capacity for data-hiding, depending on its size. An extended ROI can be defined if the message size exceeds the original capacity as in the example of Figure 3. Then, the data-hiding method have to agree with the following points:

- The message recovery needs the message size to decide if the original ROI has enough space for the embedding. As no relation exists between the message size and the ROI space, the number of bits to be hidden should be embedded in the original ROI,
- In case of enlarging, the used strategy should preserve the original ROI blocks to retrieve the message size. Thus, we will work by adding column and/or row of 8x8 blocks,
- the embedding of the message itself should take care of the previously embedded data, the message size.

When the stego-image in JPEG format undergoes cropping operation, it is compressed again, but a desynchronization could also be induced. Indeed, the origin of the image could change and so blocks defined by the JPEG compression. It leads to new variations of the pixels and can disturb the secret message. Nevertheless when this phenomenon happens, only few embedded bits are lost (less than 1%), as we could see in the example of Figure 6. These errors could be easily avoid by adding redundancy.

3.2 Data-hiding strategy

In order to garantee the secrecy of the message, we will spread randomly the bits over the blocks and over the coefficients. After the definition of the extended ROI, we define where they will be located in the following way:

- random choice of a block of the enlarged box among those not already containing the maximum number of possible embedded bits,
- random choice of a coefficient of this block among those not already selected for embedding a secret bit.

The seed of the Pseudo-Random Number Generator (PRNG) will be used as key for the data-hiding method.

Even if we work in the same way as JPEG to embed data, the two steps could not be integrated due to the use of a ROI. JPEG blocks are based on the image coordinates whereas data-hiding blocks are relative to ROI ones. Working in an asynchronous way is the solution: the data-hiding step takes place before the JPEG compression one. As the method withstands several compressions, the two successive JPEG compressions applied over the ROI do not destroy the secret message.

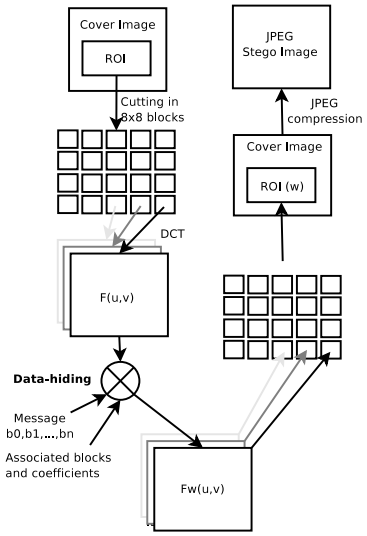


Figure 4: Data-hiding of the message.

Figure 4 shows the data-hiding process and Figure 5 the extraction.

We apply our data-hiding method on an image of a fish, Figure 6(a). We use as embedded data a PGM file, Figure 6(b). It represents around 1.5% of the cover-image, the same capacity as a data-hiding of one bit per blocks over the whole image. In the stego-image, Figure 6(c), the secret information, and so the modification of pixels, is confined to the fish itself, Figure 6(d). After a cropping operation, Figure 6(e), we notice that only four recovered secret pixels (in

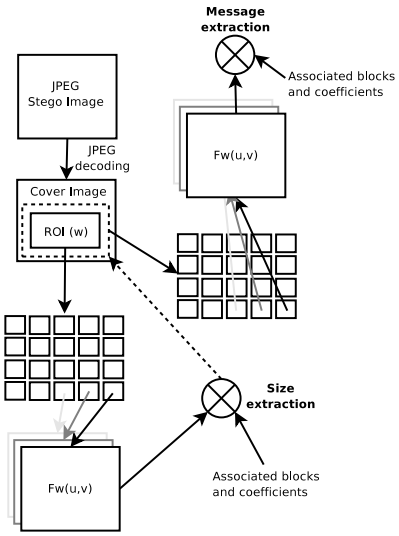


Figure 5: Extraction of the message.

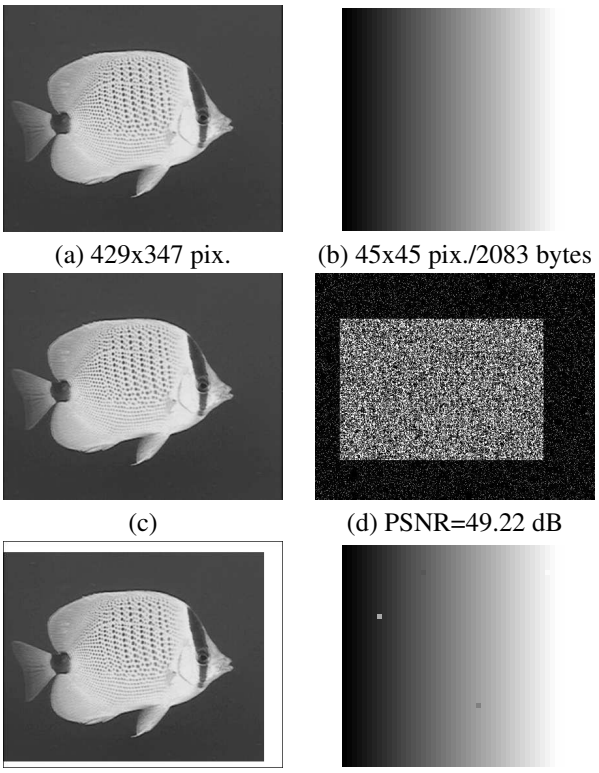
fact, four bits) are differents from original ones, Figure 6(f). With redundancy, if the message is embedded three times, it leads to use 48 coefficients per block instead of 16 and there is no more error in the extracted data.

4 CONCLUSION

In this article, we have presented a data-hiding method robust to cropping and to JPEG compression. As the message is not embedded in the whole image but in a particular ROI, the method is contextual. This leads to an asynchronous application between the data-hiding step and the compression one. The proposed JPEG robustness is a bit different from existing ones: our main goal is not the robustness to high compression rates, but rather the robustness to successive JPEG compression. The capacity of such data-hiding is important even if we work in the frequential domain, one bit can be embedded in each coefficient DCT.

Using contextual data-hiding should allow to develop robustness against transformations not yet well studied. We have presented a method robust to cropping, but basic geometric transformations, like scaling and rotation, could be treat on the same way. The difference takes place in the choice of the invariant. For the cropping, the information is located in a ROI. For the rotation, we could for example put the secret message along the main axis of the object.

Robustness to one transformation in this way seems not to be hard to obtain. Nevertheless, robustness to different kinds of transformations in a whole leads to more complex approach: the different invariants would have to be taken into account simultaneously.



(a) 429x347 pix. (b) 45x45 pix./2083 bytes

(c) (d) PSNR=49.22 dB

(e) (f)

Figure 6: Application over an image of fish: (a) original image, (b) secret message (PGM file), (c) stego-image, (d) difference between original and stego-image, (e) cropping operation, (f) message extracted from the cropped stego-image.

References

[1] C.-C. CHANG, T.-S. CHEN AND L.-Z. CHUNG. A Steganographic Method Based upon JPEG and Quantization Table Modification. In *Information Sciences* (2002), vol. 141, pp. 123–138.

[2] C.-T. HSU AND J.-L. WU. Hidden Digital Watermarks in Images. In *IEEE Transaction on Image Processing* (1999), vol. 8, pp. 58–68.

[3] H.-W. TSENG AND C.-C. CHANG. High Capacity Data Hiding in JPEG-Compressed Images. In *Informatica* (2004), vol. 15, pp. 127–142.

[4] J.M. RODRIGUES, J.R. RIOS AND W. PUECH. SSB-4 System of Steganography using Bit 4. In *WIAMIS 2004, Lisboa, Portugal* (2004).