IMPROVING EMBEDDING PAYLOAD IN BINARY IMAGES WITH “SUPER-PIXELS”

Hongmei Gou and Min Wu

ECE Department, University of Maryland, College Park, USA

ABSTRACT

Hiding data in binary images can facilitate authentication of important documents in the digital domain, which generally requires a high embedding payload. Recently, a steganography framework known as the wet paper coding has been employed in binary image watermarking to achieve high embedding payload. In this paper, we introduce a new concept of *super-pixels*, and study how to incorporate them in the framework of wet paper coding to further improve the embedding payload in binary images. Using binary text documents as an example, we demonstrate the effectiveness of the proposed super-pixel technique.

Index Terms— Data hiding in binary images, wet paper coding, pixel flippability.

1. INTRODUCTION

In this information era, an increasing number of binary images become widely used in our everyday lives. Some examples include digitized signatures, writings/drawing from pen-based devices, and text/graphic documents from scanning. The capability of hiding data in binary images can facilitate their authentication. A pre-determined pattern or some content feature is taken as the watermark and then embedded into the original image. When the watermarked image is altered, the embedded watermark will change accordingly, providing evidence that the image in question has been tampered [1].

Hiding data in binary images for authentication is a challenging problem. The embedding of patterns or content features generally requires a high embedding payload. Meanwhile, blind detection without using the original image must be supported since there is no “true” image before the authentication. When manipulating a binary image, the only operation available is black-white or white-black flipping, and pixels to be flipped for carrying watermarks must be carefully chosen to preserve perceptual quality of marked binary images. Such flippable pixels are also shown to have an uneven spatial distribution in most non-dithered binary images [1].

A common framework of data embedding in binary images with blind detection is to first identify flippable pixels based on human visual systems, and then embed watermarks by locally manipulating flippables to enforce certain pixel properties [1, 3–5]. The uneven distribution of flippables can be handled by random shuffling [1], where pixels in the original binary image are randomly shuffled before a block-based parity enforcement embedding. The shuffling method can embed a moderate amount of data, but the number of hidden bits is still quite smaller than the total number of flippable pixels. To achieve high utilization of flippables, a recent steganography framework known as wet paper coding (WPC) [6] has been employed for data embedding in binary images [7]. By jointly considering the embedding of multiple bits, the WPC-based scheme can achieve an embedding payload approaching the total number of flippables.

The authors can be contacted via {hmgou, minwu}@eng.umd.edu.

When evaluating pixel flippability in binary images, the prior art generally focuses on the flipping of individual pixels. However, in some types of binary images, such as text documents, a group of pixels that may not be individually flippable can be changed together without introducing visible artifacts, as shown in Fig. 1. After being grouped as a single unit, a set of individually non-flippable pixels may take a few number of patterns satisfying the perceptual quality requirement. This provides the embedder more room for carrying the hidden data. In our work, we refer to such a pixel set as a *super-pixel*, and propose to incorporate it in the framework of wet paper coding to further increase the embedding payload in binary images.

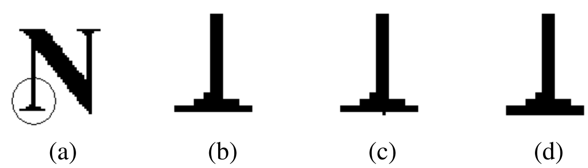

Fig. 1. Super-pixel example: (a) original text image; (b) zoomed-in view for the lower part of the leftmost stroke in (a); (c) flipping an individually non-flippable pixel below the bottom horizontal line in (b); (d) flipping a set of individually non-flippables below the same horizontal line as in (c).

The rest of the paper is organized as follows. In Section 2, we briefly review the data hiding system for binary images using wet paper coding. Section 3 discusses the formulation of super-pixels and how to incorporate them in the framework of wet paper coding for binary image watermarking. In Section 4, we present experimental results on applying the proposed technique to watermarking binary text documents. Finally, conclusions are drawn in Section 5.

2. BINARY IMAGE WATERMARKING USING WET PAPER CODES

The wet paper coding (WPC) was proposed as a solution to a scenario called "writing on wet paper" [6,8] which frequently occurs in steganography. To explain this metaphor, imagine that the cover object $\mathbf{x}$ is an image that was exposed to rain and the embedder can only slightly modify the dry spots of $\mathbf{x}$ but not the wet spots. During transmission, the marked image $\mathbf{y}$ dries out and thus the detector does not know which pixels were used by the embedder for data hiding. The task of wet paper coding is to enable both parties to exchange secret messages under the above scenario. The problem of data embedding in binary images fits this "writing on wet paper" paradigm quite well, with flippables viewed as dry spots and non-flippables as wet spots [7]. Since flipping a pixel in the embedding process may modify the flippability of its neighboring pixels, the detector will not be able to correctly identify the pixels that were used in the embedding.

For data embedding, let us assume that the cover binary image $\mathbf{x}$ consists of n elements $\{x_i\}, i = 1, ..., n, x_i \in \{0, 1\}$. Among them, there are k flippable pixels $\{x_j\}, j \in C \subset \{1, 2, ..., n\}$ and $|C| = k$. The marked image $\mathbf{y}$ also consists of n pixels. The embedding may flip a flippable pixel (i.e., $y_j = 1 - x_j$), or leave it unmodified (i.e., $y_j = x_j$). Consider the case of embedding q-bit data $\mathbf{m} = \{m_1, ..., m_q\}^\top$ in $\mathbf{x}$. A secret key is used to generate a pseudo-random binary matrix D of dimensions $q \times n$. The flippable pixels $\{x_j\}, j \in C$ are then modified if needed so that the watermarked binary image $\mathbf{y}$ satisfies $D\mathbf{y} = \mathbf{m}$. Thus, the embedder needs to solve a system of linear equations in GF(2) (i.e., binary arithmetic). For detection, the hidden data $\mathbf{m}$ can be extracted by simply performing a matrix multiplication $\mathbf{m} = D\mathbf{y}$ using the shared matrix D.

The maximal length of the data that can be embedded using wet paper coding is related to the expected rank of the matrix D, which determines if the system $D\mathbf{y} = \mathbf{m}$ has a solution or not for an arbitrary message $\mathbf{m}$. Assume that we always try to embed as many bits as possible by adding rows to D while maintaining that $D\mathbf{y} = \mathbf{m}$ still has a solution. Given k flippable pixels, it has been shown in [6] that the expected maximum number of bits that can be embedded in this manner approaches the number of flippables.

3. INCORPORATING SUPER-PIXELS IN WPC-BASED BINARY IMAGE WATERMARKING

For many binary images such as text documents, a set of neighboring pixels that are individually non-flippable can be changed together without introducing visible artifacts. Combining these pixels into a logical unit of "super-pixel", we have more resources to manipulate for carrying the hidden data. Since the wet paper coding scheme can achieve a high utilization of changeable resources, in this section, we discuss how to incorporate super-pixels in the WPC framework to further increase the embedding payload while satisfying the perceptual quality requirement on watermarked binary images.

3.1. Problem Formulation

Given a cover binary image $\mathbf{x}$ consisting of n elements $\{x_i\}, i = 1, \cdots, n, x_i \in \{0, 1\}$, in addition to the k individually flippable pixels $\{x_j\}, j \in C$, we now have u super-pixels coming from those individually non-flippable pixels $\{x_j\}, j \notin C$. In the traditional wet paper coding, the embedder determines values of the k individually flippable pixels so that the marked binary image $\mathbf{y}$ satisfies $D\mathbf{y} = \mathbf{m}$, as discussed in Section 2. In order to leverage our new concept of super-pixels to further increase the embedding payload, additionally the embedder needs to identify one of the allowed patterns for each of the u super-pixels. On the detector side, since blind detection is required, the hidden data $\mathbf{m}$ shall still be extracted via the matrix multiplication $\mathbf{m} = D\mathbf{y}$, without using the original image and any information about both individually flippable pixels and super-pixels.

3.2. Representing Super-Pixels as Regular Flippables

From the above problem formulation, we can see that the main issue remains at the embedder side, which needs to determine one of several patterns for each super-pixel. We solve this problem by representing super-pixels as regular individual flippables as follows.

Assume that the i^{th} super-pixel consists of $p_i \geq 2$ black-white pixels and allows $s_i = 2^{t_i}$ patterns satisfying the fidelity requirement, where $t_i < p_i$. To represent the s_i patterns, we now can use t_i bits, each of which can take either 0 or 1, just like a regular flippable pixel. In this way, through the introduction of the i^{th} super-pixel, we transform p_i individually non-flippable pixels to t_i regular individual flippables. With all the u super-pixels, we can increase the total number of flippables from k to $k + \sum_{i=1}^{u} t_i$.

In the framework of wet paper coding, the pseudo-random matrix D is the only information shared between the embedder and the detector, and the detector shall be able to extract the q-bit hidden data $\mathbf{m}$ from the watermarked image $\mathbf{y}$ via matrix multiplication $\mathbf{m} = D\mathbf{y}$. When incorporating super-pixels in the WPC-based scheme, such blind detection is also required, meaning that the detector shall still be able to extract the hidden data from the watermarked image $\mathbf{y}$ using only the shared pseudo-random matrix D. However, at the embedder side, the introduction of the i^{th} super-pixel shall reduce p_i individually non-flippable pixels to t_i flippables. Correspondingly, the p_i columns in the pseudo-random matrix D for the p_i non-flippables shall be reduced to t_i columns for the t_i bits representing the i^{th} super-pixel. In order to support blind detection at the detector side, such pixel and matrix reduction should be appropriately executed. Denote the above p_i columns in D as a $q \times p_i$ sub-matrix D_{p_i}, and their reduced t_i columns as a $q \times t_i$ matrix D_{t_i}. Also organize the s_i patterns that the i^{th} super-pixel can take as a $p_i \times s_i$ matrix $P^{(i)} = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \ldots \quad \mathbf{p}_{s_i}]$, and their corresponding t_i-bit super-pixel representation as a $t_i \times s_i$ matrix $T^{(i)} = [\mathbf{t}_1 \quad \mathbf{t}_2 \quad \ldots \quad \mathbf{t}_{s_i}]$. It is required that

$$D_{p_i} P^{(i)} = D_{t_i} T^{(i)}, \tag{1}$$

and this should hold true for each of the u super-pixels. After applying the above reduction to all the u super-pixels, the original cover image $\mathbf{x}$ with n pixels becomes a vector $\mathbf{x}^{(r)}$ of $n^{(r)} = n - \sum_{i=1}^{u}(p_i - t_i)$ elements, and the matrix D of size $q \times n$ becomes $D^{(r)}$ of size $q \times n^{(r)}$.

In (1), the pseudo-random sub-matrix D_{p_i} and the pattern set $P^{(i)}$ are given, while the reduced matrix columns D_{t_i} and the t_i-bit pattern indices $T^{(i)}$ are to be determined. Since the pattern set $P^{(i)}$ is designed according to perceptual properties of binary images, there may not be a solution to (1) in some cases. One special situation is that $T^{(i)}$ can be chosen to satisfy a linear relationship with $P^{(i)}$, i.e., there exists a binary matrix $G^{(i)}$ such that $P^{(i)} = G^{(i)} T^{(i)}$. By plugging $P^{(i)} = G^{(i)} T^{(i)}$ into (1), we can easily identify $D_{t_i} = D_{p_i} G^{(i)}$. A simple example of this linear case is a super-pixel allowing two patterns of $\mathbf{p}_1 = [0, 0, \cdots, 0]$ and $\mathbf{p}_2 = [1, 1, \cdots, 1]$. The 1-bit indices of $\mathbf{p}_1$ and $\mathbf{p}_2$ can be determined as $\mathbf{t}_1 = 0$ and $\mathbf{t}_2 = 1$, respectively, while the $G^{(i)}$ matrix is an all 1 vector.

3.3. WPC-based Embedding with Super-Pixels

As shown in Fig. 2, there are three main steps when implementing WPC-based embedding with super-pixels. In Step-1, by representing super-pixels as regular flippables, we reduce the original binary image $\mathbf{x}$ to $\mathbf{x}^{(r)}$, and the pseudo-random matrix D to $D^{(r)}$, as discussed above in Section 3.2. In Step-2, a set of equations are established over the reduced $\mathbf{x}^{(r)}$ and $D^{(r)}$ and solved to obtain a reduced marked vector $\mathbf{y}^{(r)}$ satisfying $D^{(r)} \mathbf{y}^{(r)} = \mathbf{m}$, using wet paper coding as reviewed in Section 2. In Step-3, from the reduced marked vector $\mathbf{y}^{(r)}$ carrying the hidden data $\mathbf{m}$, we reconstruct the marked binary image $\mathbf{y}$ of the same size as the original image $\mathbf{x}$. In the following, we present some implementation details of Step-1 reduction and Step-3 reconstruction. To facilitate these two operations, we establish a locating vector of size $(p_i, 2)$ to record the row and column indices of the p_i individual black-white pixels that are contained in the i^{th} super-pixel, and this is performed for all the u super-pixels.

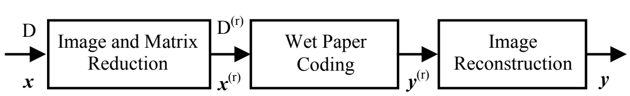

Fig. 2. Diagram of WPC-based embedding with super-pixels.

Step-1 performs image and matrix reduction as follows. Assume we have established the t_i-bit representation for each of the $s_i = 2^{t_i}$ patterns that the i^{th} super-pixel can take, i.e., we have determined the $t_i \times s_i$ index matrix $T^{(i)} = [\mathbf{t}_1 \quad \mathbf{t}_2 \quad \ldots \quad \mathbf{t}_{s_i}]$ corresponding to the $p_i \times s_i$ pattern matrix $P^{(i)} = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \ldots \quad \mathbf{p}_{s_i}]$ as discussed in Section 3.2. For the i^{th} super-pixel consisting of p_i individual pixels in the original image $\mathbf{x}$, its corresponding t_i-bit index can be identified via looking up the index-pattern mapping $T^{(i)}$-$P^{(i)}$. Aided with the locating vector, we put this t_i-bit index at the first t_i locations of the super-pixel, and mark the remaining $p_i - t_i$ locations as "invalid". To perform matrix reduction, according to the locating vector, we first organize the p_i columns in D to obtain D_{p_i}. Then, as discussed in Section 3.2, we reduce D_{p_i} to D_{t_i} of t_i columns. After that, we replace the first t_i columns of D_{p_i} with D_{t_i}, and mark the remaining $p_i - t_i$ columns as "invalid". After the above processing is finished for all the u super-pixels, "invalid" pixels in $\mathbf{x}$ and "invalid" columns in D are repudiated to obtain $\mathbf{x}^{(r)}$ and $D^{(r)}$, respectively. The reduced image vector $\mathbf{x}^{(r)}$ has $n - \sum_{i=1}^{u}(p_i - t_i)$ pixels, among which $k + \sum_{i=1}^{u} t_i$ are flippables, ready for the subsequent WPC-based embedding.

In Step-3, we have a reduced $\mathbf{y}^{(r)}$ of $n^{(r)} = n - \sum_{i=1}^{u}(p_i - t_i)$ pixels and satisfying $D^{(r)} \mathbf{y}^{(r)} = \mathbf{m}$, and we need to reconstruct a marked image $\mathbf{y}$ with n pixels and satisfying $D\mathbf{y} = \mathbf{m}$. Among the $n^{(r)}$ pixels in $\mathbf{y}^{(r)}$, WPC embedding may have flipped some of the $k + \sum_{i=1}^{u} t_i$ flippables. Comparing $\mathbf{y}^{(r)}$ and the reduced cover image $\mathbf{x}^{(r)}$, we can easily identify the actual flippings, each of which originally may be an individually flippable pixel or one element of a super-pixel index. For the former, we identify the pixel location j in the original image $\mathbf{x}$ and then flip it to obtain its value in the marked image $\mathbf{y}$, i.e., $\mathbf{y}(j) = 1 - \mathbf{x}(j)$. For the latter, aided with the locating vector, we first retrieve the whole index with t_i bits. Then, we find its corresponding p_i-pixel group via the established index-pattern mapping, and place the pixel group at its corresponding locations in $\mathbf{y}$. Such reconstruction is performed for all actual flippings, while the untouched pixels in $\mathbf{y}$ will be the same as their counterparts in $\mathbf{x}$.

4. EXPERIMENTAL RESULTS

In this section, we present experimental results on applying the proposed super-pixel technique to watermarking binary images. Using binary text documents as an example, we first evaluate the embedding payload with super-pixels identified and incorporated in the framework of wet paper coding. We then examine the perceptual quality of watermarked binary images. We will see that the incorporation of super-pixels in the WPC-based embedding can effectively increase the embedding payload while preserving the perceptual quality of marked binary images.

Embedding Payload with Super-Pixels: Taking the 200×410 binary text document in Fig. 3(a) as the cover image, we first identify its individually flippable pixels. Using the method in [1], which assigns a flippability score to individual pixels, we identify 792 pixels having the highest flippability

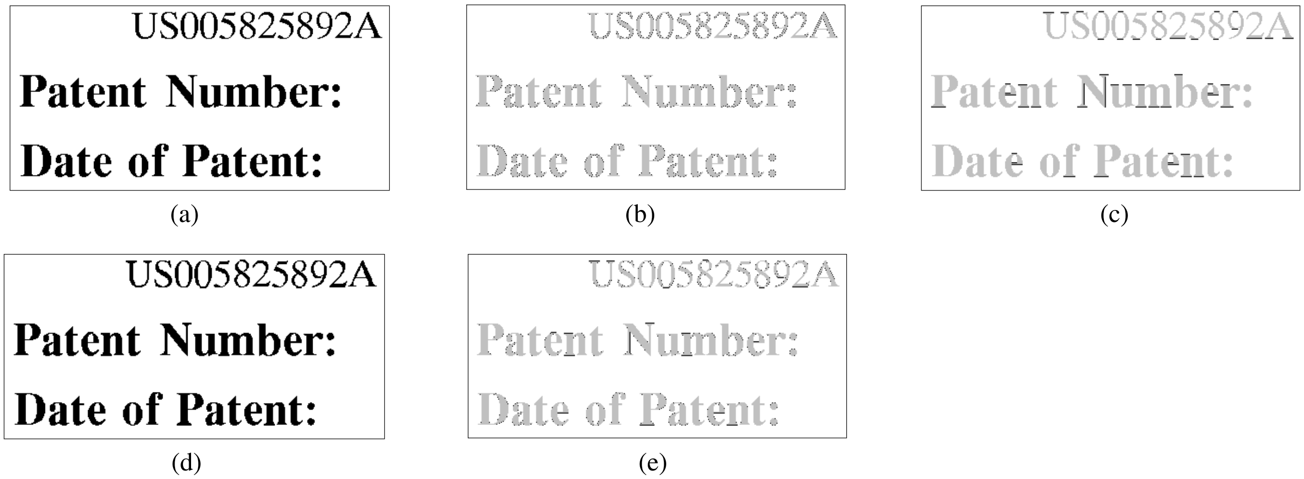

Fig. 3. Experimental results: (a) original image; (b) individually flippable pixels shown as dark points; (c) straight boundaries as super-pixels shown as dark lines; (d) marked image; (e) pixel-wise difference between (a) and (d) shown as dark points/lines.

score 0.625, 174 pixels with score 0.375, 35 pixels with score 0.25, and 196 pixels with score 0.125. Flipping a pixel with a higher flippability score shall bring less noticeable artifacts. To obtain watermarked images with high perceptual quality, we select all pixels with the highest score as the $k = 792$ individually flippable pixels in the embedding, as shown in Fig. 3(b). We can see that these individually flippable pixels are located on the non-straight boundaries.

Next, we identify super-pixels for the cover text document. As discussed in Section 1, individually non-flippable pixels on a straight boundary can be flipped together without affecting the perceptual quality of text documents. Hence, as shown in Fig. 3(c), we select 53 straight boundaries as our 53 super-pixels, with appropriate trimming to separate each super-pixel from individually flippable pixels as well as other super-pixels. Each of these 53 super-pixels can take two patterns, either all 1's or all 0's. As discussed in Section 3.2, with such a pattern design, we can successfully implement the image and matrix reduction before the wet paper coding.

Incorporating super-pixels in the wet paper coding in addition to individually flippable pixels, we can further improve the embedding payload of binary image watermarking. As each of the $u = 53$ super-pixels takes two patterns that can be indexed by 1 bit, the total number of effective flippables is increased from $k = 792$ to $k + u \times 1 = 792 + 53 = 845$, so is the embedding payload. This is an increase of 6.7%.

Perceptual Quality of Marked Images: In Fig. 3(d), we show a watermarked text document for the cover image in Fig. 3(a), with $q = 842$ bits embedded using both the 792 individually flippable pixels and the 53 super-pixels. Visually comparing these two images, we can see that the embedding does not introduce annoying artifacts. The pixel-wise difference between the original image and the marked image is shown in Fig. 3(e). Given the marked image in Fig. 3(d), the 842-bit hidden data can be accurately extracted via matrix multiplication by using only the shared matrix D.

5. CONCLUSIONS

In this paper, we have proposed a novel method to improve the embedding payload in binary images. We have introduced a new concept of super-pixels to address dependencies among the flippability of multiple pixels, and studied how to incorporate super-pixels under the framework of wet paper coding. Before the WPC embedding, our method performs appropriate reduction by representing super-pixels as regular flippables. After that, reconstruction is applied to obtain a watermarked binary image of the same size as the original image. At the detector side, the same procedure as in the wet paper coding is employed for blind watermark extraction.

6. REFERENCES

[1] M. Wu and B. Liu, "Data hiding in binary image for authentication and annotation," *IEEE Trans. on Multimedia*, Aug. 2004.

[2] M. Wu and B. Liu, "Data Hiding in Image and Video: Part-I - Fundamental Issues and Solutions," IEEE Trans. on Image Proc., vol. 12, no. 6, pp.685–695, 2003.

[3] K. Matsui and K. Tanaka, "Video-steganography: how to secretly embed a signature in a picture," *IMA Intellectural Property Project*, 1994.

[4] Q. Mei, E.K. Wong, and N. Memon, "Data Hiding in Binary Text Documents," *SPIE Conference on Security, Watermarking and Steganography*, 2001.

[5] E. Koch and J. Zhao, "Embedding robust labels into images for copyright protection," *Int. Congr. Intellectural Property Rights for Specicalized Info., Knowledge and New Technologies*, 1995.

[6] J. Fridrich, M. Goljan, P. Lisonek, D. Soukal, "Writing on Wet Paper," *IEEE Trans. on Signal Processing*, Oct. 2005.

[7] M. Wu, J. Fridrich, M. Goljan, and H. Gou, "Handling Uneven Embedding Capacity in Binary Images: A Revisit," *SPIE Conference on Security, Watermarking and Steganography*, 2005.

[8] J. Fridrich, M. Goljan, and D. Soukal, "Perturbed Quantization Steganography with Wet Paper Codes," ACM Multimedia and Security Workshop, 2004.