The Journal of Systems and Software 93 (2014) 152–162

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A high capacity data hiding scheme for binary images based on block patterns

Chung-Chuan Wang[e], Ya-Fen Chang[d], Chin-Chen Chang[c], Jinn-Ke Jan[a], Chia-Chen Lin[b,*]

[a] Department of Computer Science, National Chung Hsing University, Taichung 402, Taiwan, ROC
[b] Department of Computer Science and Information Management, Providence University, Taichung 43301, Taiwan, ROC
[c] Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan, ROC
[d] Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung, Taiwan, ROC
[e] Department of Multimedia and Game Science, Chung Chou University of Science and Technology, Changhua, Taiwan, ROC

ARTICLE INFO

Article history:
Received 8 February 2013
Received in revised form
24 December 2013
Accepted 18 February 2014
Available online 5 March 2014

Keywords:
Data hiding
Binary image
Watermarking
Authentication

ABSTRACT

This paper proposes a high capacity data hiding scheme for binary images based on block patterns, which can facilitate the authentication and annotation of scanned images. The scheme proposes block patterns for a 2×2 block to enforce specific block-based relationship in order to embed a significant amount of data without causing noticeable artifacts. In addition, two kinds of matching pair (MP) methods, internal adjustment MP and external adjustment MP, are designed to decrease the embedding changes. Shuffling is applied before embedding to reduce the distortion and improve the security. Experimental results show that the proposed scheme gives a significantly improved embedding capacity than previous approaches in the same level of embedding distortion. We also analyze the perceptual impact and discuss the robustness and security issues.

1. Introduction

Digital watermarking techniques have been proposed for a wide range of applications, including ownership protection, copy control, annotation, and authentication. Most past works are designed for color and grayscale images where the pixels take on a wide range of colors or brightness levels. Nowadays, there are many digital binary images are increasingly common used in our life, such as signatures, drawings, and scanned documents. Having the capability of hiding data in binary images can facilitate the authentication, annotation, and tracking of these binary contents. However, hiding data in binary images is much more difficult than in color or grayscale images. For binary images in which the pixels take value from only two possibilities: black or white, and they are drastically different to our eyes. As a result, hiding data in binary images without causing visible artifacts becomes more difficult.

Several block-based methods for hiding data in specific types of binary images have been proposed in literature. Tseng and Pan (2001) proposed a data hiding scheme for binary image blocks by using an integer-weight matrix. These matrices are operated with bitwise exclusive-OR and pair-wise multiplication to hide and protect data. The scheme achieves reasonable hiding capacity but poor visual effects. In 2002, they further improved their scheme by sacrificing some data hiding capacity to suppress visual distortion (Tseng and Pan, 2002). However, the "hole" phenomenon still occurs on two adjacent flippable bits. Chen et al. (2003) proposed another data embedding scheme based on an $N \times N$ block pattern, called block data hiding method (BDHM). However, in their scheme each embeddable block only hides one bit. Tzeng and Tsai (2003) proposed a block-based data hiding scheme in which special codes are embedded into the blocks of given images and verified to accomplish the authentication purpose.

Wu and Liu (2004) first identified those places in a binary image where a white pixel can be flipped to black or vice versa without introducing noticeable artifacts. Most of these so-called flippable pixels are on the edge of characters or along the border of a stroke. However, the embeddability of binary images is unevenly distributed. The uneven embedding capacity problem has been addressed in (Wu et al., 2005) in which a random shuffling key is used to shuffle the image so that the flippable pixels are distributed uniformly throughout the image. With different flippability criteria, Yang and Kot (2007) proposed another pattern-based data embedding scheme based on the connectivity-preserving in a local neighborhood. A window of size 3×3 is employed to assess

* Corresponding author. Tel.: +886 4 26328001x18108; fax: +886 4 26321161.
E-mail addresses: ccwang@dragon.ccut.edu.tw (C.-C. Wang), cyf@nutc.erdu.tw (Y.-F. Chang), ccc@cs.ccu.edu.tw (C.-C. Chang), jkjan@cs.nchu.edu.tw (J.-K. Jan), mhlin3@pu.edu.tw (C.-C. Lin).

http://dx.doi.org/10.1016/j.jss.2014.02.023
0164-1212/

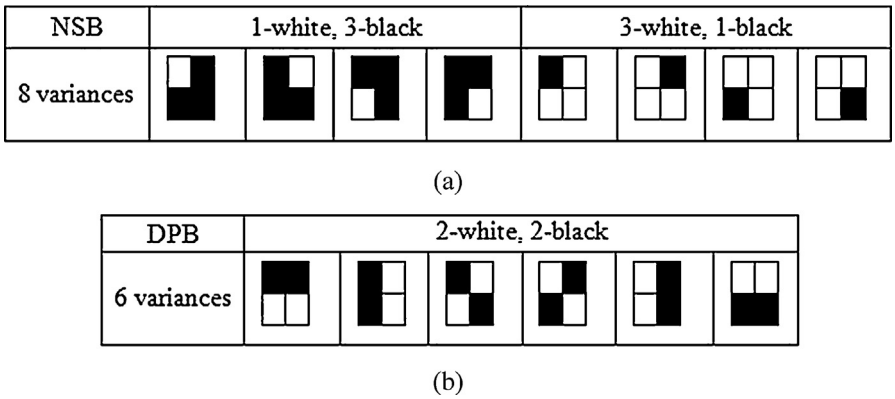

Fig. 1. Two embedding block patterns: (a) NSB with 8 variances and (b) DPB with 6 variances.

the flippability of a pixel in a block. However, it requires 0.5 pixel flipping on average to embed a bit of authentication information. To reduce the high pixel-flipping rate, Lee et al. (2007, 2009) used Hamming codes to embed authentication information into the cover image with flipping only a small number of pixels. Moreover, their proposed schemes are more resilient against some known steganalysis attacks (Cheng et al., 2005).

Many watermarking methods are to embed the watermark by modulating the coefficients of a properly chosen transform domain such as discrete wavelet transform (Yang et al., 2008), gyrator domain (Liu et al., 2010a,b), and discrete fractional Fourier transform (Guo et al., 2011). Yang et al. (2008) presented a high capacity data hiding scheme for binary images authentication based on the interlaced morphological binary wavelet transforms. The relationship between the coefficients obtained from different transforms is utilized to identify the suitable locations for watermark embedding such that blind watermark extraction can be achieved.

Some of data hiding techniques utilize the LSB embedding applied either directly to pixel values, or to quantized the coefficients for image format (Wu and Lee, 1998). To achieve least visual quality reduction, Pan et al. (2001, 2002) used the prioritized pattern matching scheme to embed the invisible data in the pixels those are close to the boundaries. Venkatesan et al. (2007) proposed a data hiding scheme using the parity of blocks to ensure that for any bit that is modified in the host image, the bit must be adjacent to another bit that has the same value as the former's new value. In addition, recently there are many reversible data hiding schemes (Chang et al., 2013; Ou et al., 2013; Huang et al., 2013) have been proposed for authentication applications.

Unfortunately, the robustness is low in some existing methods, which may result in inauthenticity and unintegrity of the original image. Besides, some methods require a large number of pixels to be flipped to embed reasonable authentication information. Therefore, we propose a high capacity data hiding for binary images based on block patterns. To achieve high hiding capacity, we use two block patterns for a 2×2 block, dual-pair block (DPB) and non-symmetrical block (NSB), to hide secret data in a binary image. In addition, two kinds of matching-pair (MP) methods: internal adjustment (MP_IA) and external adjustment (MP_EA), are proposed to suppress the embedding changes. Shuffling is applied before embedding to reduce the distortion and improve the security.

To make this paper self-contained, in Section 2, we describe the concepts from block pattern and matching pair that will be needed for the rest of the paper. Section 3 contains a detailed exposition of the proposed algorithm. In Section 4, we experimentally investigate the relationship between the capacity and distortion, and the influence of variant images on the capacity. We also give compare performance with existing schemes and analyze the robustness and security issues in the same section. Finally, we conclude the paper in Section 5.

2. Block pattern and matching pair

We first present two new embedding block patterns: non-symmetrical block (NSB) and dual-pair block (DPB) to improve the hiding capacity. Besides, in order to reduce the embedding changes, two kinds of matching pair methods: MP_IA and MP_EA are also proposed in this section.

2.1. NSB and DPB

Two new embedding block patterns designed for the proposed data hiding scheme: non-symmetrical block (NSB) and dual-pair block (DPB), are shown in Fig. 1(a) and (b), respectively. NSB is a 2×2 block set with 1-white, 3-black or 3-white, 1-black pixels, while DPB is a 2×2 block set with 2-white and 2-black pixels. We further extend NSB and DPB to complement NSB (CNSB) and complement DPB (CDPB), which are illustrated in Fig. 2.

For simplicity, we use a binary bit 0 to represent black, bit 1 to represent white, and four binary bits to represent the pixels in a block. Each class comprises a complement pair of blocks. For example, white-black-black-black and black-white-white-white blocks represent class 1 with two complement (1000, 0111) blocks. Any two complement blocks are considered as a class. Seven classes are generated from CNSB and CDPB. The CNSB pattern comprises four classes, (1000, 0111), (0100, 1011), (0010, 1101), and (0001, 1110) as classes 1, 2, 3, and 4, respectively. Similarly, the CDPB pattern has two-white and two-black pixels with three classes, (0011, 1100), (0101, 1010), (0110, 1001) as classes 5, 6 and 7, respectively. We use the seven classes to denote septenary *se*. Here, we define classes 1, 2, 3, and 4 of CNSB in septenary value as $se = i - 1$, where i is the class number or the position where its pixel value is different from others in the block. In other words, two binary bit streams, (1000) and (0111), are mapped to the septenary value 0. And classes 5, 6 and 7 of CDPB in the septenary value can be determined according to the clustering directions of their pixels: horizontal (0011, 1100 as *se* 4), vertical (0101, 1010 as *se* 5) and skew (0110, 1001 as *se* 6). Take class 5 for example, two pixels groups, 00 and 11, can be separated horizontally as (0011), (1100). The same idea can be easily applied to classes 6 and 7. To give a clear explanation, an example is given in Fig. 3. From Fig. 3(b), 19 septenary values are generated according to two block patterns: CNSB and CDPB defined in Fig. 2. Note that the 2×2 gray area including four white pixels does not fit any block patterns, hence, there is an empty cell occurred in Fig. 3(b).

Furthermore, we use a mapping relationship based on variable length transformation to increase the embedding bit rate. From Fig. 4, we can see our average hiding capacity is about 2.857 bits