Pattern Recognition 42 (2009) 1604 – 1611

Contents lists available at ScienceDirect

Pattern Recognition

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

Data hiding in grayscale images by dynamic programming based on a human visual model☆

I-Shi Lee[a,1], Wen-Hsiang Tsai[a,b,*]

[a]Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan
[b]Department of Information Communication, Asia University, Taichung 41354, Taiwan

ARTICLE INFO

Article history:
Received 7 September 2006
Received in revised form 22 June 2008
Accepted 11 January 2009

Keywords:
Data hiding
Grayscale image
Human visual system
Block pattern encoding
Dynamic programming

ABSTRACT

A new method for data hiding in grayscale images based on a human vision model with distortion-minimizing capabilities is proposed. Each of the eight bit planes of an input grayscale image is viewed as a binary image, into which message data are embedded horizontally. Two optimization techniques, namely, block pattern coding and dynamic programming, are proposed for image distortion minimization. Experimental results show good performs of the proposed method.

1. Introduction

Data hiding in images is a useful technique for secret communication. Many data hiding techniques have been proposed recently [1–3]. A common approach is least-significant-bit (LSB) replacement, which embeds message data in the LSB planes of an image. The image into which a message is hidden is called a *cover image*, and the result a *stego-image*. Wang et al. [4] embedded a binary image in the fifth LSB plane of a cover image using a genetic algorithm and a local pixel adjustment method to lower the distortion in the stego-image. Chang et al. [5] used dynamic programming to obtain an optimal solution for the LSB substitution method. Chan and Cheng [6,7] presented an optimal pixel adjustment process to improve the quality of the stego-image acquired by Wang's schemes. Thien and Lin [8] embedded data in images digit by digit using a modulus function, which improves LSB substitution not only in eliminating false contours but also in reducing image distortion. Lee and Chen [9] applied variable-sized LSB insertion to estimate the maximum embedding capacity by a human visual system (HVS) property, and to maintain image fidelity by removing false contours in smooth image regions. Liu et al. [10] presented a novel bit plane-wise data hiding scheme using variable-depth LSB substitution and employed post-processing to eliminate the resulting noticeable artifacts.

Most of the above methods lack consideration of using precise human visual models in improving the data hiding effect. Instead, Wu and Tsai [11] presented a method based on the HVS by modifying quantization scales according to variation insensitivity from smooth to contrastive to improve stego-image quality. And Lie and Chang [12] presented an adjusted LSB technique with the number of LSBs adapting to the pixels of different grayscales.

On the other hand, some steganalysis techniques were developed to detect secret messages among stego-images. Lyu and Farid [13] developed a universal blind detection scheme to detect hidden messages in stego-images, which uses wavelet-like decomposition to build higher-order statistical models of natural images and adopts the support vector machine as an optimal classifier to separate stego-images from cover images. The method demonstrates good performance on JPEG images and the selected statistics is rich enough to detect hidden data in the results yielded by a very wide range of steganographic methods. In addition, to detect data hidden in LSBs in the spatial domain, it is observed that the basic LSB substitution method changes pixel values only between $2i$ and $2i$+1 in the i-th bit plane of the pixel value. This leads to an effective

☆This work was supported partially by the NSC Advanced Technologies and Applications for Next Generation Information Networks (II)—Sub-Project 5: Network Security, Project no. NSC-96-2752-E-009-006-PAE and partially by the NSC Project NSC96-2422-H-009-001.

* Corresponding author at: Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan.
Tel.: +886 3 5728368; fax: +886 3 5734935.
E-mail addresses: gis87809@cis.nctu.edu.tw (I.-S. Lee), whtsai@cis.nctu.edu.tw (W.-H. Tsai).
[1] Also with Department of Management Information at Northern Taiwan Institute of Science and Technology.

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.01.014

steganalytic technique, the RS method proposed by Fridrich et al. [14], which not only can expose the presence of secret data but also can estimate the length of the embedded data.

In this study we propose a method to embed data into a grayscale image, based on the use of a new HVS model to estimate the number of usable bits of each pixel in the cover image. Furthermore, a block pattern encoding method is proposed to embed up to three data bits in a 2×2 block of the bit planes without yielding visible image quality degrading. This is achieved by using two optimization techniques. The first technique utilizes multiple block pattern encoding tables, from which an optimal one is chosen for each input image; and the second technique uses dynamic programming to divide the message data stream into appropriate bit segments for optimal data bit embedding in the image blocks to minimize a cost function. The proposed method can extract embedded data without referencing the original image.

In the remainder of this paper, we introduce the idea behind the proposed method in Section 2. In Section 3, we describe the adopted HVS model and the corresponding cost function. In Section 4, the proposed data hiding method is described. The corresponding data recovery process is proposed in Section 5. Some experimental results are given in Section 6, followed by discussions and conclusions in Section 7.

2. Embedding data in bit planes of grayscale images

Eight bits represent a pixel's intensity in a grayscale image. The *bit plane* formed by the same bit of each pixel in the grayscale image is a binary image. Fig. 1 shows the eight bit planes of each of three 128×128 grayscale images. The LSB plane bp_0 is almost fully randomized. If a message is embedded in bp_0, the result will appear almost unaltered. On the contrary, random noise is less in a more-significant-bit plane. The most-significant-bit plane bp_7 contains almost no noise, and data cannot be embedded easily in it without causing significant visual changes. Embedding data into bit planes in the order of $bp_0, bp_1, \ldots, bp_7$ is called *horizontal data hiding*, contrastive with traditional *vertical data hiding* which embeds data into the bits $b_7, b_6, \ldots, b_0$ of each pixel in the order of b_0 through b_7, where b_0 is the LSB of the pixel. Comparatively, horizontal data hiding can reduce more distortion in the stego-image, as revealed in the results of this study.

On the other hand, embedding data directly in bit planes will cause visible damages to the edges in the bit planes. To overcome this difficulty, in this study we design a new cost function which considers certain perception characteristics of the HVS, and adopt a method proposed in Lee and Tsai [15] for data embedding. Each bit plane is regarded to have a different weight in its capability for data hiding, and the new cost function is designed accordingly to measure the degree of distortion resulting from pixel value changes, as described next.

3. Cost function for distortion measurement

Two HVS characteristics may be exploited for reducing image distortion in stego-images. First, human perception is more sensitive to grayscale changes in smooth areas than in texture areas in a grayscale image. Second, human perception is sensitive to relative luminance rather than absolute one. Designing the cost function for distortion measurement for data embedding must take these two characteristics into consideration.

For the first consideration, assume that a pixel P with grayscale value g is to be used to embed message data. Let MAX denote the maximum grayscale value, and MIN the minimum, in a 3×3 block with P as the center, which we call the *neighborhood* of P. Then, the maximum *between-pixel grayscale range* in this block is $\Delta = \text{MAX}-\text{MIN}$. To avoid a significant change of smoothness with respect to the neighborhood of P, the new grayscale value g' resulting from data embedding should be restricted in a certain range, which is taken to be $g \pm \Delta/2$ in this study. Then, we define a *maximum number D* of *data-embeddable bits* at P as

$$D = \lfloor log_2(\Delta/2) \rfloor = \lfloor (\log_2 \Delta) - 1 \rfloor = \lfloor \log_2(\text{MAX} - \text{MIN}) - 1 \rfloor. \quad (1)$$

For the second consideration, let f denote the luminance of a pixel P with grayscale value g where $1 \leqslant f \leqslant 100$. According to the Fechner law [16], the relative luminance property perceived by the HVS may be expressed as a contrast value c computed by $c = 50 \times \log_{10} f$ where $0 \leqslant c \leqslant 100$. Moreover, according to the Weber law [16], the maximum allowable change Δc of the contrast value c, according to the principle of "just noticeable difference (JND)" about the pixel's luminance change, is about 2. That is, if the luminance of a pixel is changed too much so that Δc is larger than 2, the change will be noticeable to the HVS. Accordingly, we can compute in another way a maximum number of data-embeddable bits in the 8 bits of a pixel's grayscale value, as described next.

First, we compute the maximum luminance change $(\Delta f)_{max}$ in accordance with the maximum allowable contrast change $(\Delta c)_{max} = 2$. With c being the contrast of pixel P, let c_{max} denote the maximum possible contrast value. Then, we have

$$2 = (\Delta c)_{max} = c_{max} - c = 50 \times \log_{10} f_{max} - 50 \times \log_{10} f = 50 \times \log_{10} \frac{f_{max}}{f},$$

which can be reduced to be

$$\frac{f_{max}}{f} = 10^{(2/50)} = 10^{0.04}.$$

So, the maximum allowable luminance change can be expressed as

$$(\Delta f)_{max} = f_{max} - f = \left(\frac{f_{max}}{f} - 1\right) f = (10^{0.04} - 1) \times f \approx 0.0965 \times f.$$

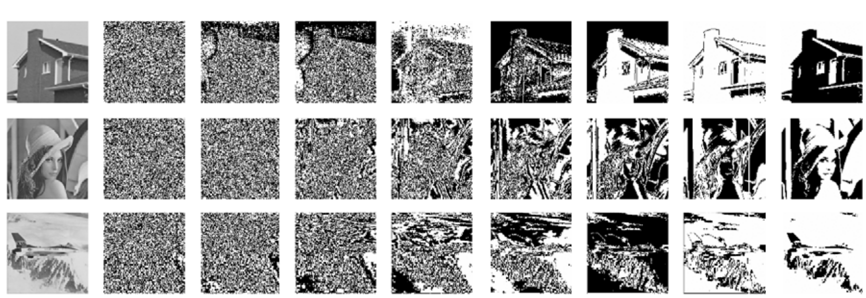

Fig. 1. Three grayscale images and their 8 corresponding bit planes (from left to right, original images, bp_0, bp_1, $bp_2, \ldots$, and bp_7, respectively).