A New Block Data Hiding Method for the Binary Image

Jeanne Chen
HungKuang University, Taiwan

Tung-Shou Chen
National Taichung Institute of Technology, Taiwan

Meng-Wen Cheng
National Taichung Institute of Technology, Taiwan

INTRODUCTION

Great advancements in Web technology have resulted in increase activities on the Internet. Users from all walks of life — e-commerce traders, professionals and ordinary users — have become very dependent on the Internet for all sorts of data transfers, be it important data transactions or friendly exchanges. Therefore, data security measures on the Internet are very essential and important. Also, steganography plays a very important role for protecting the huge amount of data that pass through the internet daily.

Steganography (Artz, 2001; Chen, Chen & Chen, 2004, Qi, Snyder, & Sander, 2002) is hiding data into a host image or document as a way to provide protection by keeping the data secure and invisible to the human eyes. One popular technique is to hide data in the least significant bit (LSB) (Celik, Sharma, Tekalp, & Saber, 2002; Tseng, Chen, & Pan, 2002). Each pixel in a gray image takes up eight bits representation and any changes to its last three LSBs are less likely to be detected by the human visual system. However, an image in LSB hiding is not robust to attacks.

Other hiding techniques involve robust hiding (Lu, Kot, & Cheng, 2003), bit-plane slicing (Noda, Spaulding, Shirazi, Niimi, & Kawaguchi, 2002), hiding in compressed bitstreams (Sencar, Akansu, & Ramkumar, 2002) and more. Some applications require more data to be hidden but must not have visually detectable distortions. An example would be the medical records. More details on high capacity hiding could be found in Moulin and Mihcak (2002), Candan and Jayant (2001), Wang and Ji (2001), Rajendra Acharya, Acharya, Subbanna Bhat, and Niranjan (2001), and Kundur (2000).

Although much research had been done for data hiding, very little exists for hiding in binary images. The binary (or black and white) image is common and often appears as a cartoon in newspapers and magazines. Most are easy preys to piracy. Hiding is difficult for the binary image, since each of its black or white pixels requires only one bit representation. Any bit manipulation will reveal hiding activities, and the image is easily distorted. Indeed, the block data hiding method (BDHM) proposed in this paper is concentrated largely for hiding in binary images.

RELATED WORK: A NOVEL HIDING METHOD

Pan, Wu, and Wu (2001) partitioned an image into blocks where each block would be repartition into four overlapping sub-blocks. Next, all the white pixels in the sub-blocks will have a number assigned to each of them. Based on these numbers, the characteristic values of the sub-blocks were calculated and used to determine the suitable sub-blocks for hiding such that the hidden data will show as uniformly distributed on the block and have no visible distortions. Data will only be hidden in the center pixel. The bit for the center pixel in the selected sub-block will be toggled from white to black as hiding a bit.

For example, a 512×512 image which was partitioned into 16384 units of 4×4 blocks as in Figure 1(a). Each block was further repartitioned into four overlapping 3×3 sub-blocks as in Figure 1(b). From

the sub-blocks, it can then be easily determined that the possible sub-blocks for hiding are (1), (2) and (11). Also, after hiding the characteristic values should not be altered.

THE PROPOSED BLOCK DATA HIDING METHOD (BDHM)

A binary image is actually made up of black and white pixels. There is only one bit representation for each pixel. Each pixel is either 0 for black or 1 for white. There are two types of binary image: one is the simple black-white binary image, and the other is the complex binary image such as the natural or halftone images.

The Block Data Hiding Method (BDHM) proposed in this paper involves data hiding and data retrieving for the black-white binary image. First, the original image is partitioned into blocks. Next, characteristic values for individual blocks will be calculated and important data hidden in the blocks based on these values. The image with hidden data is called a stego-image. The data retrieval process is similar to data hiding with the exception that data will be retrieved. Figure 2 illustrates the flow for data hiding and retrieving.

Partitioning into Blocks and Sub-Blocks

First, the simple binary image will be partitioned into $N'N$ sized blocks. Next, each $N'N$ sized block will be repartitioned into four overlapping $n'n$ sized blocks. Suppose $N'N$ is a 4´4 sized block, then the overlapping sub-blocks will be 3´3 as shown in Figure 3. Pixels in columns two and three of sub-block (1) overlap pixels in columns one and two of (2). Pixels for sub-blocks (3) and (4) also overlap in this respect, too.

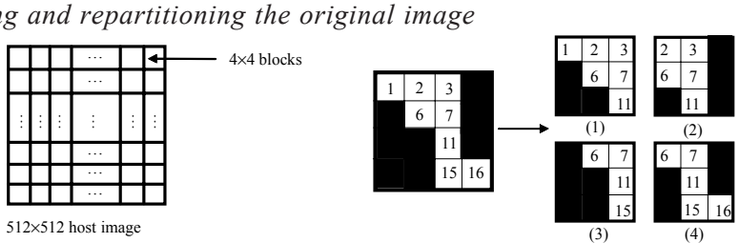
Calculating Characteristic Values

The characteristic values for the blocks are calculated based on the number of white pixels (each bit value is 1). Let R_j be the characteristic value for block j where j=1, 2, 3, …, M. M is the total number of partitioned blocks. Let r_i be the number of white pixels in sub-block i where i=1, 2, 3, 4 and T_j be the total number of sub-blocks with the same least number of white pixels. Then $R_j=r_i$ where r_i is the least number of white pixels. As illustrated in the example in Figure 4, r_i={3, 3, 4, 3} for sub-blocks (1), (2), (3) and (4), respectively. Since three sub-blocks; (1), (2), and (4) have the least number of white pixels, T_j=3. The least number of white pixels in the sub-blocks is 3; therefore R_j=3.

Hiding the Data

After characteristic values had been calculated for every block, the blocks will be sorted by ascending order of R_j. No data must be hidden in B_j if it contains exclusively black or white pixels. If a white pixel is hidden in an all black block (R_j=9) or a black pixel in an all white block (R_j=0), the contrast is too noticeable to the naked eyes. Therefore to avoid hiding in exclusively black or white blocks, a rule is set to allow hiding only in blocks with 3f"R_jf"6. The white and black pixels are usually distributed uniformly in the blocks with R_j within this range. Further information would also be needed on the distribution of the black and white pixels in each block before starting the hiding process. Figure 5 shows different distributions for two different blocks. Although the characteristic value R_j in Figure 5(a) is smaller than Figure 5(b), the distribution of the black and white pixels is more uniform in Figure 5(b). Data hidden in Figure 5(a) would stand out clearly rather than in Figure 5(b).

Figure 1. Partitioning and repartitioning the original image



(a) Partitioning into 4×4 blocks (b) Repartitioning into 3×3 sub-blocks