PERGAMON

Pattern Recognition 36 (2003) 1583–1595

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy

Chin-Chen Chang[a], Ju-Yuan Hsiao[b,*], Chi-Shiang Chan[a]

[a] *Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan 621, ROC*
[b] *Department of Information Management, National Changhua University of Education, Changhua, Taiwan 500, ROC*

Received 27 July 2002; received in revised form 19 August 2002; accepted 19 August 2002

Abstract

The processing of simple least-significant-bit (LSB) substitution embeds the secret image in the least significant bits of the pixels in the host image. This processing may degrade the host image quality so significantly that grabbers can detect that there is something going on in the image that interests them. To overcome this drawback, an exhaustive least-significant-bit substitution scheme was proposed by Wang et al. but it takes huge computation time. Wang et al. then proposed another method that uses a genetic algorithm to search "approximate" optimal solutions and computation time is no longer so huge. In this paper, we shall use the dynamic programming strategy to get the optimal solution. The experimental results will show that our method consumes less computation time and also gets the optimal solution.

Keywords: Image hiding; Least-significant-bit substitution; Dynamic programming; Genetic algorithm; Image embedding

1. Introduction

In the Internet, tons of various kinds of data are sent, transmitted and received every single moment. Some of them may be secret information of commerce and others confidential messages from the government, both of which are candidate preys for grabbers to access. In order to keep the grabbers away, a variety of techniques have been proposed. One of the most famous methods is data encryption [1,2], which uses a certain algorithm to transform data into cipher texts. Only the user that has keys can decrypt the secret data from the cipher texts. For any grabber who does not have a key, the cipher texts will look like nothing but streams of meaningless codes. Although data encryption is a good way to prevent grabbers from accessing secret data, it still has some weaknesses. The appearance of cipher texts would give grabbers an impulse to recover them. Moreover, grabbers might even simply destroy the cipher texts out of rage when they have trouble recovering them so that the legal receivers cannot get the data in time. That is the reason why data hiding [3,4] has been researched recently. Data hiding techniques embed the important data into multimedia data such as images, videos or movies. In this paper, we take an image for a carrier of secret data and name it as a host image. After embedding the secret data into the host image, the output image of this hiding process is called a stego- image. When we discuss image hiding, one of the most important things is that the quality of the stego-image must not be degraded too much after embedding. The goal of data hiding is to make the stego-image invisible to grabbers. Thus, if the process of embedding degrades the quality of the stego-image too much, any grabber will easily take notice of it. Techniques of least-significant-bit-based substitution are simple ones to achieve data hiding. A least-significant-bit-based substitution method replaces some least significant bits of the host image with the secret data. After least-significant-bit-based data hiding, the researches of data hiding focus on the

* Corresponding author. Fax:+886-4-7211162.
E-mail addresses: ccc@cs.ccu.edu.tw, ccs88@cs.ccu.edu.tw (C.-C. Chang), hsiaojy@cc.ncue.edu.tw (J.-Y. Hsiao).

0031-3203/03/$30.00 © 2003 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.
PII: S0031-3203(02)00289-3

difficulty of removing the embedded data through operations of images. Most researches concerned are concentrated upon watermarking [5–7]; the capacity of embedding is not the key point in those methods. However, in our opinion, one of the important things in data hiding is the embedding capacity. When we want to transmit important secret data via a certain technique of data hiding, it is obviously not practical if we have no choice but to break the secret data into pieces and put each piece in a large host image due to the limited capacity. On the other hand, the simple least-significant-bit substitution method directly replaces some least significant bits of each pixel value in the host image with the secret data; it does not focus on the difficulty of removing secret data through images operations but rather on the capacity of embedding. When talking about the capacity of embedding, the major concern is to make stego-image extremely hard for grabbers to sense the existence of the secret data. There exists one method to achieve this aim. The method first sets a bijective mapping function and transforms values of secret data into another set of values according to a bijective mapping function. It then replaces the rightmost least significant bits of each pixel in host image with transformed values to form the stego-image. It goes without saying that if we can find an optimal solution of bijective mapping function, we can use a bijective mapping function to make the stego-image have the best quality. In order to find the optimal solution of a bijective mapping function, the procedure of the exhaustive least-significant-bit substitution must exhaustively go through all possible bijective mapping functions. When the rightmost k least significant bits of the host image are replaced by the secret data, k being large, it takes huge computation time and makes the method impractical. In 2001, Wang et al. [8] proposed a method that uses a genetic algorithm to search for an approximate optimal solution of a bijective mapping function, and the computation time is no longer so huge as that of the exhaustive least-significant-bit substitution. Although Wang et al.'s method reduces the computation time, the solution it comes up with is not an optimal one of the bijective mapping function but an "approximate" optimal solution instead. In order to get an optimal solution of a bijective mapping function, we propose the method that uses a dynamic programming strategy to achieve this purpose. Our method not only gets an optimal solution of a bijective mapping function but also significantly reduces the computation time.

The rest of this paper is organized as follows. We shall describe the related works first in Section 2. We shall introduce the image hiding in detail by using the least-significant-bit substitution in Section 2.1 and Wang et al.'s genetic algorithm in Section 2.2. After that, we shall describe our method in Section 3. Details of our method for finding an optimal solution are first introduced in Section 3.1, and then some properties about our solution are discussed in Section 3.2. In Section 4, the experimental results show how our method performs. Finally, the conclusions are presented in Section 5.

2. Related works

In this section, the related works are introduced. We will first look at the image hiding technique using least-significant-bit substitution. After least-significant-bit substitution, we will continue to describe exhaustive least-significant-bit substitution. Wang et al.'s genetic algorithm to search for an optimal or approximate optimal solution is described in Section 2.2.

2.1. Least-significant-bit substitution

In this subsection, the simple least-significant-bit substitution and exhaustive least-significant-bit substitution are introduced. Suppose our embedded data is an image called the secret image S, and the host multimedia data is also an image called the host image. Both the images are 8-bit gray-scale images. We assume that the size of the host image is a multiple of the secret image S.

There exist two steps to the method of simple least-significant-bit substitution. In the first step, the method gathers all the 8-bit pixel values in the secret image S sequentially. It, then, decomposes the bit streams into several k-bit unit and takes each single k-bit unit as a single k-bit pixel. Let this k-bit pixel secret image be called S'. The value of k is decided on the basis of a calculation to make the height and width of S' be the same as that of the host image. The second step replaces each k least significant bit of the 8-bit pixel value in the host image with the corresponding k-bit in S' to get the stego-image. According to our description above, the quality of the stego-image produced by simple least-significant-bit substitution may not be acceptable. It means that the method degrades the image quality so significantly that the stego-image may probably attract grabbers' attention. Once a grabber notices the stego-image, she/he can simply extract and analyze the least significant bits to get the secret image. To improve the security and the quality of the stego-image, a method of exhaustive least-significant-bit substitution is presented in [8].

For convenience, let the k-bit residual image denote the k-bit gray-scale image which is derived by extracting the rightmost k least significant bits from each pixel in a host image. With regard to exhaustive least-significant-bit substitution, some additional works will be done for S' to increase the security and image quality. In the first step, a bijective (i.e., one-to-one and onto) mapping function is set up that transforms the positions of each pixel in S' into random positions. More specifically, we first number all pixels in S' sequentially from left to right and top to bottom. After numbering, the original position, x, a pixel is transformed into a new position, $F(x)$, according to the following bijective mapping function [8]:

$$F(x) = (k_0 + k_1 \times x) \bmod N. \tag{1}$$