Applied Mathematics and Computation 208 (2009) 284–292

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Capacity maximization for reversible data hiding based on dynamic programming approach ☆

Kuo-Liang Chung [a,*], Yong-Huai Huang [a], Wei-Ning Yang [b], Yu-Chiao Hsu [b], Chyou-Hwa Chen [a]

[a] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei 10672, Taiwan, ROC

[b] Department of Information Management, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei 10672, Taiwan, ROC

ARTICLE INFO

Keywords:
Embedding capacity maximization
Dynamic programming
Histogram modification
PSNR
Reversible data hiding

ABSTRACT

Recently, an efficient reversible lossless data hiding algorithm by Ni et al. was presented. Their fast algorithm can recover the original image without any distortion and its PSNR lower bound is higher than that of all existing reversible data hiding algorithms. Instead of selecting the peak-valley pairs in a greedy way, this paper presents a dynamic programming-based reversible data hiding algorithm to determine the most suitable peak-valley pairs such that the embedding capacity object can be maximized. Based on some artificial map images, experimental results demonstrate that our proposed algorithm has 9% embedding capacity improvement ratio and has the similar image quality performance when compared to Ni et al.'s algorithm although it has some execution-time degradation. For natural images, the embedding capacity of Ni et al.'s algorithm is very close to the maximal embedding capacity obtained by our proposed algorithm. Furthermore, the comparison between our proposed dynamic programming-based algorithm and the reversible data hiding algorithm by Chang et al. is investigated.

1. Introduction

Data hiding is an important technique for authentication, identification, annotation, and copyright protection [8]. In the past decades, many types of data hiding algorithms have been developed, such as the least significant bit plane (LSB)-based algorithm [14,16,24,5], the spread-spectrum-based data hiding algorithm [9,23,21,11,18,19], the singular value decomposition (SVD)-based algorithm [17,6,7,3], and the vector quantization (VQ)-based scheme [15,2]. Among these developed data hiding algorithms, the hiding data can be embedded into images without causing visual degradation. However, once the original image has been modified by the data hiding algorithm, it is very hard to recover the original image from the marked one when the receiver demands to use the original image.

In order to reverse the marked image back to the original one, some reversible data hiding algorithms have been developed to achieve the reversibility goal for the marked image. Based on the modulo 256 addition, Honsinger et al. presented the first reversible algorithm [13] to embed data into gray images. By using the lossless multiresolution transform, Macq and Deweyand [20] decomposed the original image into several subbands and applied the modulo 256 addition to embed data into these decomposed subbands. In [10], Fridrich et al. first compressed some selected bit planes of the original image, and then both the hidden data and the compressed bit planes were embedded into the selected bit planes. Due to concentrating

☆ Supported by National Council of Science of R.O.C. under contracts NSC96-2221-E-011-027 and NSC97-2221-E-011-128.
* Corresponding author.
E-mail address: k.l.chung@mail.ntust.edu.tw (K.-L. Chung).

0096-3003/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2008.10.066

on the authentication, the above three reversible data hiding algorithms only provide limited data hiding capacity, i.e. limited embedding capacity.

To increase the capacity of the reversible data hiding, Goljan et al. [12] segmented the original image into non-overlapping blocks and classified each block into two types. The hidding data is embedded by flipping a block from one type to the other type and the original type of each block is also embedded for reversibility. Based on the integer wavelet transform, Xuan et al. [25] embedded the data into the selected bit planes of wavelet coefficients in middle and high frequency subbands. The binary values of selected bit planes are first compressed losslessly, and then they are embedded into the original image together with the hiding data. Celik et al. [1] quantized gray values and then by using the predictive coding, these quantized gray values were used as the side information to losslessly compress the quantization residuals. Both the compressed residuals and the hiding data were embedded into the LSB of the original image. In the data extracting process, the original image can be recovered by adding decompressed quantization residuals to quantized gray values in the marked image. Chang et al. [4] presented a reversible data hiding algorithm based on the predictive coding. In the data embedding process, the predictive coding is first performed to predict the gray value of each pixel, and then the hiding data can be embedded by modifying the these predicted gray values. In the data extracting process, the predictive coding is performed again to extract the embedded data and restore the original gray value of each pixel. Although the above four reversible data hiding algorithms [12,25,1,4] have higher embedding capacity, the peak signal to noise ratios (PSNRs) between the original image and the marked one for the four algorithms are some degraded and they are time-consuming.

Recently, Ni et al. [22] presented a novel reversible data hiding algorithm based on the histogram modification technique. Using the greedy approach, the histogram of the original image is first segmented into several non-overlapped intervals and each interval is bounded by a peak-valley pair. The key idea in Ni et al. [22]'s algorithm is that the pixels contributed to the peak point of each peak-valley pair in the histogram can be used to embed data by slightly modifying their gray values. Experimental results showed that the capacity of the reversible data hiding can be significantly increased by embedding the data into peak points of the histogram. Especially, due to only modifying the gray values of concerned pixels slightly, the PSNR lower bound of Ni et al.'s algorithm is higher than any one of all existing reversible data hiding algorithms.

Instead of using the greedy approach in Ni et al.'s algorithm to select the set of peak-valley pairs heuristically, this paper presents a dynamic programming approach to determine the most suitable set of peak-valley pairs and the determined set can maximize the embedding capacity while has the similar image quality performance. Based on some artificial map images, experimental results demonstrate that our proposed algorithm has 9% embedding capacity improvement ratio and has the similar image quality performance when compared to Ni et al.'s algorithm although it has some execution-time degradation. For natural images, the embedding capacity of Ni et al.'s algorithm is very close to the maximal embedding capacity obtained by our proposed algorithm. Furthermore, the comparison between our proposed dynamic programming-based algorithm and the reversible data hiding algorithm by Chang et al. is investigated.

The rest of this paper is organized as follows. In Section 2, Ni et al.'s reversible data hiding algorithm is surveyed. In Section 3, our proposed dynamic programming-based algorithm is presented to maximize the embedding capacity of Ni et al.'s algorithm. In Section 4, some experiments are carried out to demonstrate the advantage of our proposed algorithm. Finally, some concluding remarks are addressed in Section 5.

2. Past work by Ni et al.

In Ni et al.'s algorithm, the data is embedded into the original image by slightly modifying its histogram. Assume that we have an original image I and its histogram $H(I)$ is shown in Fig. 1a. For ease of exposition, let $h(x)$ denote the number of pixels with gray value $x \in [0, 255]$. In Fig. 1a, it can be observed that the gray value p is the peak point in $H(I)$ since $h(p) > h(x)$ for all $x \neq p$; that is, p is the most frequent gray value appeared in the original image I. In addition, gray value v is the valley point in

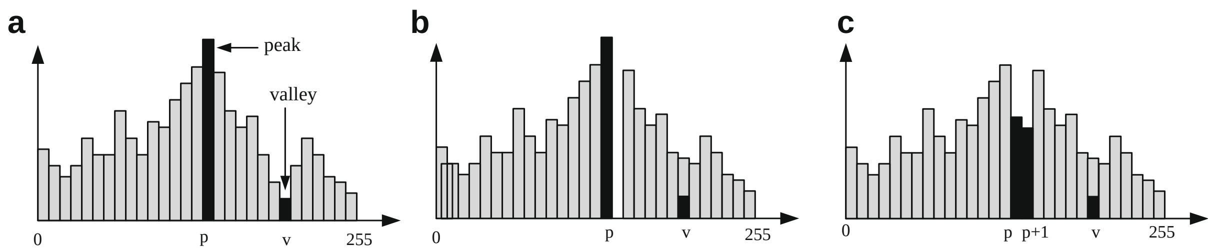

Fig. 1. The main idea in Ni et al.'s algorithm. (a) A histogram example with only one peak point and one valley point. (b) Shift the interval (p, v) to the right by one unit. (c) Embedding the data into the positions with gray levels p and $p + 1$.