Signal Processing 108 (2015) 56–68

Contents lists available at ScienceDirect

Signal Processing

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

Data embedding in random domain

Mustafa S. Abdul Karim*, KokSheik Wong

Faculty of Computer Science & Information Technology, University of Malaya, Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 14 May 2014
Received in revised form
9 August 2014
Accepted 12 August 2014
Available online 3 September 2014

Keywords:
DeRand
Random domain
Universal data embedding
Universal parser

ABSTRACT

A universal data embedding method based on histogram mapping called DeRand (Data embedding in Random domain) is proposed. DeRand theoretically defines redundancy in any digital signal by applying the universal parser such that high entropy random signals can certainly be utilized for data embedding. First, DeRand recursively parses a random signal into a set of tuples each of certain length until there exist some tuples of zero occurrences in the histogram. Then, tuples that occur in the histogram are associated with those of zero occurrences. Next, a tuple (of non-zero occurrence) is mapped to its corresponding associated tuple to embed "1", while the tuple is left unmodified to embed "0". DeRand is universal, reversible, applicable to any random signal and scalable in terms of embedding capacity and signal quality. Experimental results show that DeRand achieves an embedding capacity up to 4909 bits in random signal of size 256 Kbytes. In addition, the quality of the processed signal ranges from 0.0075 to 395.67 in terms of MSE.

1. Introduction

Multimedia data is massively generated nowadays thanks to the advanced yet low cost capturing and storage technologies. In addition to the conventional network traffic such as web accesses, email communication, and transmission of e-commerce data, these multimedia data are communicated and shared among various users across continents. The advent of innovative social network services at no cost and the deployment of online content store further multiplied the utilization of network. For example, it is reported that around 28,000 photos and 72 h of videos are uploaded each minute to Instagram and Youtube [1], respectively. Thus, the communication channels, including Internet and cellular phone networks, convey a mixed streams of information (appearing in various formats and coding structures) sent from multiple sources. These streams appear as random data when being transmitted over the communication channels. Generally, any unintelligible data can be considered as a random data (or random signal hereinafter) from the perspective of a third party who has no access to its original (intelligible) form. These random data include encrypted signal, records in database, and data uploaded to cloud-storage, which need to be managed for efficient utilization of the communication bandwidth and storage space. Here, for digital data management purposes, data embedding technologies provide conveniences to achieve annotation, authentication, watermarking, etc. However, fundamentally, data embedding is a feature-dependent process, where features of a host are modified in certain domain and coding structure in order to embed data [2–6]. For example: (a) [7–9] manipulate the features of the host image in the spatial domain; (b) [10–12] process the host image in the frequency domain; (c) [13] is restricted to compressed video; (d) [14] is applicable only to audio; (e) [15] embeds data in text, and; (f) [16–19] are restricted to encrypted signals. Therefore, the interchangeability among most data embedding methods is generally restricted by its domain or media with certain features. In other words, the definition of feature is necessary to achieve

* Corresponding author. Tel.: +603 7967 6417; fax: +603 7957 9249.
E-mail addresses: mustafa.s.abdulkarim@ieee.org, mustafa.alwahaib@siswa.um.edu.my (M.S. Abdul Karim), koksheik@um.edu.my (K. Wong).

http://dx.doi.org/10.1016/j.sigpro.2014.08.037
0165-1684/

data embedding by these methods. On the other hand, it is technically challenging to extract features from a random signal, which is unintelligible. To the best of our knowledge, data embedding in the random domain has not been considered in the literature.

Karim and Wong [20] proposes a reversible data embedding method that can be universally applied to any encrypted signal. This method operates solely in the universal domain [20] and hence it is not dependent on the underlying features of the encrypted host. For that, this method can be applied, theoretically, to any signal. However, this method completely changes the format of the host due to the design of the data embedding mechanism. In particular, all segments of the host are mapped to the Golomb–Rice Codewords [21,22], which are then modified to accommodate a payload while maintaining the bitstream size. Thus, it is impossible to preserve the semantic of the host after achieving data embedding. As a result, the original encrypted host must be completely restored prior to decryption. In addition, there is no control on the embedding capacity and host distortion (viz., not scalable) in [20].

On the other hand, Ong et al. [23] put forward a reversible data embedding method that offers scalability in terms of embedding capacity and perceptual quality in image. In this method, the ability to control the perceptual quality (in the host image) is utilized to achieve perceptual encryption (viz., image scrambling), where the semantic of the image is intentionally masked by the designed substitution operation. Here, the payload is embedded into the image, along with the side information required to restore the original image. The substitution operation is achieved by modifying the histogram of the image. In particular, the probability $P(x)$ of each intensity level x in the image is analyzed. Basically, x is classified into two sets based on its probability, namely, $G_1 = \{x: P(x) > 0\}$ and $G_2 = \{x: P(x) = 0\}$. Hence, to embed “1”, $x \in G_1$ is mapped to $y \in G_2$, and to embed “0”, the value $x \in G_1$ is left unmodified, i.e., no mapping is performed. This method successfully overcomes the underflow and overflow problems in the conventional histogram shifting methods [24,25]. However, it fails to embed data when the entire range of intensity levels is occupied, i.e., $P(x) > 0$ for all $x \in [0, 2^{\lambda} - 1]$, where λ is a positive integer greater than zero. In such case, [23] partitions the image into non-overlapping blocks and handles each block individually. Furthermore, this method is verified only with images, and it depends on the statistical features of the image in the spatial domain. Hence, it is not feasible to apply [23] for handling a random signal, which consists of high entropy data.

This paper proposes a universal, reversible and scalable data embedding method that is applicable to any random signal based on histogram mapping. Unlike the traditional histogram mapping methods, the proposed method can *certainly* define redundancy in any given signal by applying the universal parser [20], even when all bins in the histogram are occupied, i.e., $P(x) > 0$ for all $x \in [0, 2^{\lambda} - 1]$. In particular, the universal parser recursively partitions the signal into segments using increasing length. Theoretically, we prove that as the length of the segments increases, the probability of defining redundancy (i.e., $\exists x' | P(x') = 0$) increases. In other words, the amount of redundancy changes as the length of the segments changes. This change in redundancy is exploited to embed a payload into the random signal by histogram mapping. The proposed data embedding method achieves the following properties: (1) applicable to any random signal; (2) reversible, and; (3) scalable in terms of carrier capacity and progressive quality degradation. To the best of our knowledge, the proposed method is the first of its kind to achieve reversible data embedding in random signal and the first universal data embedding method that offers scalability in carrier capacity and quality degradation.

The rest of this paper is organized as follows: Section 2 describes the possible applications of data embedding in the random domain. Section 3 presents the theoretical study of data embedding using histogram mapping in the random domain. Section 4 proposes DeRand (Data embedding in Random Domain) based on histogram mapping. The results of empirical study of are detailed in Section 5, and Section 6 concludes this paper.

2. Applications of data embedding in random domain

It is assumed that any unintelligible signal which losses its semantic due to intentional encryption or un-intentional loss of access to the original format is called a random signal. This includes signals generated by a random number generator. Hence, random signals occupy a significant percentage among all digital signals exchanged through the communication channels and those stored in the storage systems. Unfortunately, the random signals are, by and large, still left unexplored in the applications of information processing. For that, data embedding in the random domain contributes in filling up this deficiency through some applications, such as data hiding for covert communication (i.e., steganography) in random signals. On the other hand, reversible data embedding enables the insertion of external information while preserving the file size of the original signal. This feature can be exploited to reduce the bandwidth required to communicate/store the random signal by embedding (or hosting) one segment of a random signal into another random signal. Hence, compression is gained and the total size of the random signals is reduced. The reduction in total signal size is a significant contribution in bandwidth utilization, especially for massive data transmission and storage purposes. As another application, a third party (e.g., a cloud administrator who has no access to the original format of the data) needs to embed data in random signals for annotation and management purposes, which include the insertion of a hash value in its corresponding random signal for integrity checking. The aforementioned potential applications justify the need to consider data embedding in the random domain.

3. Theoretical study on histogram mapping

In this section, data embedding by histogram mapping is studied theoretically.

Definition 1. Let $X = \{x_i\}_{i=1}^{N}$ be a finite set of discrete uniformly distributed ordered elements where each element $x_i \in X$ is derived from a finite set of alphabets $\mathbb{A}$, i.e.,