International Journal of Innovative Computing, Information and Control
Volume **14**, Number **3**, June **2018**
ICIC International ©2018 ISSN 1349-4198
pp. 767–779

ENHANCING THE PERFORMANCE OF AUDIO DATA HIDING METHOD BY SMOOTHING INTERPOLATED SAMPLES

Tohari Ahmad[1] and Tegar Palyus Fiqar[2]

[1]Department of Informatics
Institut Teknologi Sepuluh Nopember
Kampus ITS Surabaya, Jawa Timur 60111, Indonesia
tohari@if.its.ac.id

[2]Department of Informatics
Institut Teknologi Kalimantan
Kampus ITK Balikpapan, Kalimantan Timur 76127, Indonesia
tegar@itk.ac.id

Received September 2017; revised January 2018

Abstract. *Protecting sensitive data such as military and medical data has played a significant role. Many methods have been introduced to secure and prevent the data from unauthorized users, including hiding secret data in other data such as audio, which is called data hiding methods. However, its ability to accommodate larger secret data is very limited. Moreover, the quality of the resulted stego data is not high enough. In this paper, we intend to work on these issues. Here, audio samples are firstly interpolated before being embedded by the secret. Next, the samples are smoothed to preserve their quality. In order to recover the audio and extract the secret data, the reverse steps are performed. The evaluation is done on various genres of audio musics. The experimental results show that this proposed method is able to increase the performance. Furthermore, the quality of the stego data has significantly raised in certain music genres and instruments.*

Keywords: Data hiding, Data protection, Data security, Information security, Interpolation

1. **Introduction.** In this digital era, computer networks have become a popular media for people to communicate. The ease of data transfer leads to the increasing of the communication traffic. As a result, the number of internet users worldwide has significantly grown for last decades [1], from 6.8% to 46.1% of the world population.

Overall, the transmitted data can be text, image, audio or video which can also be categorized as public and private/confidential data. Since this second type of data must not be disclosed to unauthorized users, there must be mechanisms to protect them. For this reason, various techniques have been introduced, such as cryptography, watermarking and steganography, such as in [2, 3]. Those second and third methods are almost similar that both embed secret data in the cover/carrier data, which can be any type of files. The result of this embedding process is stego data, which are the cover with the secret in it. Different from watermarking, however, steganography which is also often called as data hiding focuses on the payload (secret data) instead of the cover. While watermarking provides the payload either visible or invisible to the public, steganography makes the payload totally invisible. So, the payload can only be obtained by authorized users. In the practical implementation, cryptography is often used to encrypt the secret before being embedded to the cover. Nevertheless, this may lead to increasing the cost of the system.

In terms of data hiding, some methods have been proposed, from simple to complex ones. Examples of those which work on audio files are least significant bit (LSB) [4] along with its variations such as [5], echo hiding [6] and its extension like [7], phase coding [8], spread spectrum [9], fragile watermarking [10], modulus function [11], compressed domain [12] and LSD [13]. The LSB method itself is considered as an early and simple method for concealing confidential data. This works by replacing some least significant bits (LSB) of respective pixels in images or samples in audio, with the payload. This process is relatively fast. Yet, the payload whose size is not too high, may be detected easily. In further development, this method has been used as the main reference of more complex and modern techniques, such as in [14, 15].

In general, data hiding methods have the same challenging problems that much research works on. Those are the capacity of the payload and the quality of the stego data. Here, quality is defined as the similarity level between before and after the payload is embedded. Higher capacity and quality mean that the method is better. This also means that more secret data can be hidden in a media without much distorting its appearance. However, these two factors are inversely proportional which may not be achieved at the same time. In this paper, we propose an interpolation-based technique in order to increase the capacity while still maintaining the quality of the stego data, and vice versa. This is intended to improve the performance of the existing method, especially that in [15, 16]. For this research, we use audio files to carry the secret, which have been expanded before the embedding process takes place.

This paper is organized as follows. Section 2 describes some previous research which relates to ours. The proposed method is explained in Section 3, followed by its experimental results in Section 4. Finally, the conclusion of the paper is drawn in Section 5.

2. **Related Works.** This section describes some previous research which relates to the proposed method. This covers recent development which has been able to improve the performance of the existing research. As depicted in the previous discussion, a data hiding method requires a media to carry the secret. Once this secret has been embedded in it, the stego data is obtained as illustrated in Figure 1. This stego data is then sent to the respective destination through any communication network. Next, the receiver extracts the payload to obtain the secret data. Sometimes, the original cover is also recovered, but this is not a main requirement.

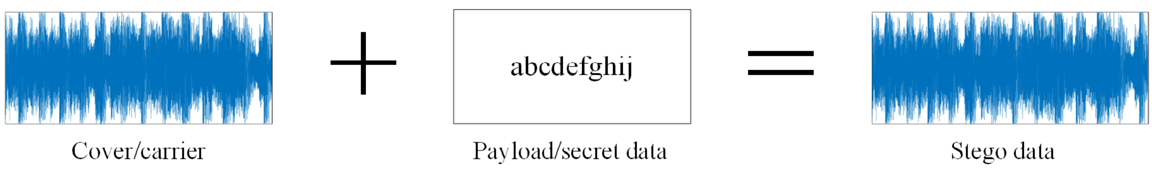


FIGURE 1. General concept of data hiding

Due to its characteristic, audio is often used as a carrier. Since it is an analog wave, it must be digitized which needs pulse code modulation (PCM) for the quantizing process. This results in various numbers of samples, depending on its sampling rate. Additionally, each sample has bit depth (e.g., 16 bit) which also affects the quality of the audio. Among existing audio-based data hiding methods, there are some techniques which are often used.

In phase coding [18], phases of the resulted components of signal are selected. This component is where the secret resides. This technique is able to defend against distortion and compression, but its embedding capacity is relatively low. Moreover, it is susceptible to low pass filtering. In spread spectrum [9], the secret is put along the spectrum of the signal frequency. In addition to the carrier, this method needs chip signals for differentiating the secret from its carrier. There are two modulation steps in the embedding process: modulating chip signal with the carrier and modulating that resulted signal with the secret. Therefore, this technique is more complex.

While the previous data hiding mechanisms employ normal voice, fragile watermarking [10] is intended to work on audio forensic. In this research, the payload is distributed along the signal. Although it is not for directly hiding the secret, the mechanism of how the embedding process to be done can be considered.

Recently, least significant digit (LSD) is introduced [19] as an improvement of LSB method. Different from LSB, here the secret is not represented as bit anymore, but in digit. However, this data format may have an effect on the quality of the resulted stego data. In other research, Fujimoto et al. [20] propose a mechanism which implements spline interpolation. It has only limited capacity since a frame can only be used by 1 bit secret data. In more details, some other research can be found in [21].

In the next research, Jung and Yoo [15, 16] develop an interpolation-based method on image which is combined with LSB. An embedded synthesizing scheme is designed with some new functions. The neighbor mean interpolation is used to obtain new pixels used as insertion point. This interpolation process is provided in Figure 2. It is shown that 2×2 block of pixels is expanded to 3×3 before the embedding is performed. Pixels of the original image are $P(0,0)$, $P(0,2)$, $P(2,0)$, $P(2,2)$ and the interpolated pixels are $P(0,1)$, $P(1,0)$, $P(1,1)$, $P(1,2)$, $P(2,1)$. The number of bits which is statically defined, however, has become its weakness. This has an effect on its overall performance.

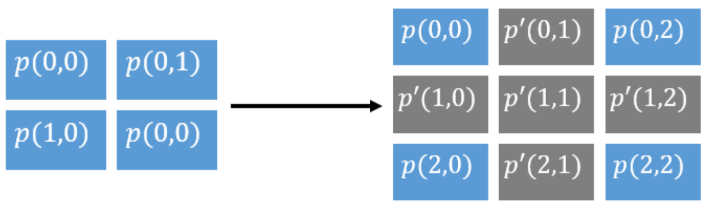


FIGURE 2. Interpolation process of Jung and Yoo's method [16]

3. **Proposed Method.** As the description in the previous section, there are still challenges in the existing data hiding methods: the capacity of the payload and the quality of the stego data. In this research, we intend to work on this issue according to those research in Section 2, especially [16] as the main reference. However, in this research we focus on the use of audio file for carrying the secret, instead of image files. For this purpose, we adopt Newton's divided difference interpolation (NDDI) polynomials [17] to create a pivoting scheme that generates a new signal. This is where the secret to be hidden. So, the secret is not embedded into the original signal anymore. Furthermore, the NDDI signal which is the pivoting point parameter is also to be the location maps.

This data hiding method consists of two processes: embedding/encoding and extraction/decoding. Here, the overall process of the proposed method is provided in Figure 3, where the research contribution is within the dash-line. It is shown that the maximum number of bits which can be hidden in a sample is specified. Once this value is embedded, the resulted signal is smoothed for reducing the difference. Finally, the stego audio is obtained which should be as similar as possible to the original audio. Meanwhile, the input

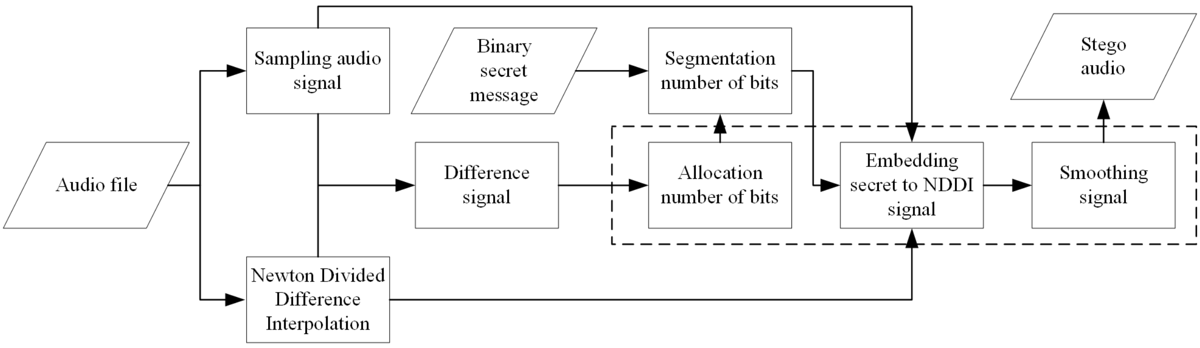


FIGURE 3. The embedding process

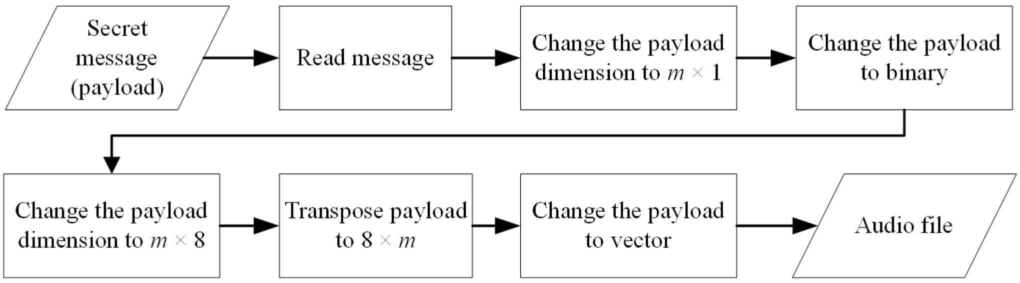


FIGURE 4. Processing the payload

(secret data) is converted to binary digits (bits). This is performed by transforming each character of secret into ASCII codes before changing it to bits. This resulting bits are stored in a vector s which contains $8m$ elements, where m is the number of payload characters whose step is depicted in Figure 4. In more details, the embedding and extraction processes are provided in the following subsections.

3.1. **Embedding process.** The input of this proposed technique is *.wav audio files. This is selected because it is not in the compressed format which means that its quality is relatively good. Here, each audio sample is normalized to unsigned 16 bit whose range is from 0 to 65535. An example of this sampling process is depicted in Figure 5(a).

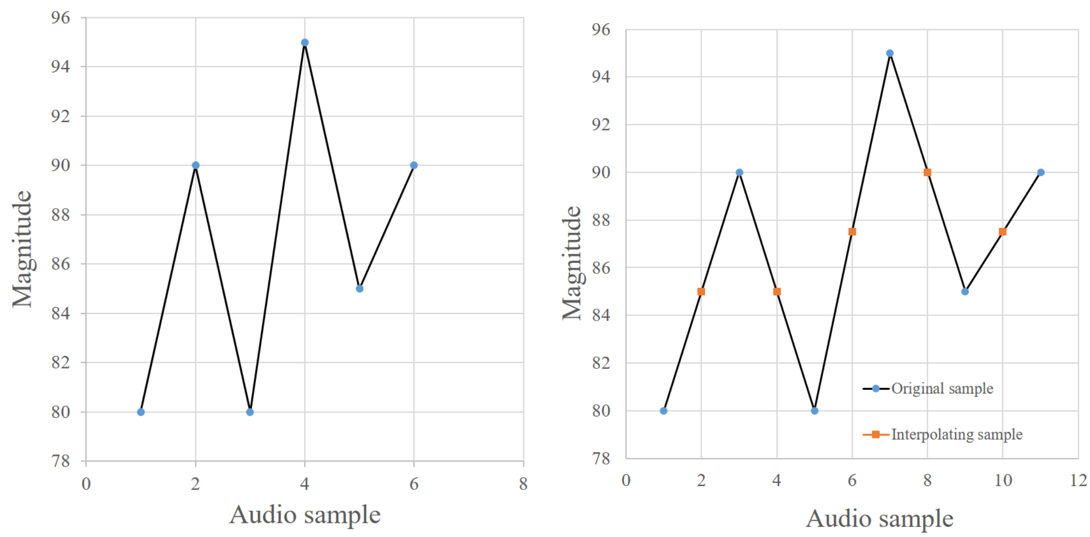
By using Newton's divided difference interpolating (NDDI) polynomials system [17] especially the linear interpolation, the resulted samples are interpolated whose rule is defined in (1) where $y_{(n-1)\_(n+1)}$ represents the new generated value from a sampling with index n. This new value is put between sample in $n-1$ and $n+1$ whose illustration is shown in Figure 5(b). So, after the interpolation process has finished, the total number of samples has become $2i-1$, where i is the number of original samples.

$$y_{(n-1)\_(n+1)} = \frac{n-(n+1)}{(n-1)-(n+1)} y_{n-1} + \frac{n-(n-1)}{(n+1)-(n-1)} y_{n+1} \quad (1)$$

An example of this process is illustrated as follows. Let $y_1 = 80$, $y_2 = 90$, $y_3 = 80$ be the values of the first, second and third original audio samples, respectively (see Figure 5(a)). According to (1), the value of NDDI interpolation is as follows:

$$y_{1\_2} = \frac{2-3}{1-3}80 + \frac{2-1}{3-1}90 \Rightarrow y_{1\_2} = 85$$

$$y_{2\_3} = \frac{4-5}{3-5}90 + \frac{4-3}{5-3}80 \Rightarrow y_{2\_3} = 85$$



(a) Audio sampling (b) Interpolated sample

Figure 5. Examples of sampling and interpolating processes

It is depicted that these new interpolation values have even sequent numbers of samples, while the original samples have odd numbers. This characteristic implies that the location map is not needed anymore, because the sequence itself specifies the mapping. These new interpolating and original samples are put in the same vector, such that we have $y = [y_1, \overline{y_2}, y_3, \overline{y_4}, \ldots, \overline{y_n}, y_{n+1}]$, where $\overline{y_n}$ represents the interpolation samples. Therefore, in this example, the vector is $y = [80, 85, 90, 85, 80]$ as depicted in Figure 5(b).

Next, the space for hiding the secret is created within those samples. For this purpose, the difference between samples is calculated by using (2), similar to that of [15].

$$d_n = \lfloor |\overline{y_n} - y_{n-1}| \rfloor \tag{2}$$

In these examples, we have two difference values as follows.

$$d_1 = \lfloor |85 - 80| \rfloor \Rightarrow d_1 = 5$$
$$d_2 = \lfloor |85 - 90| \rfloor \Rightarrow d_2 = 5$$

Some bits are then embedded in the obtained interpolating samples. In this research, we define (3) to specify the number of bits which can be carried. It is depicted that each interpolating sample may hide different amount of secret bits. It is worth noting that this is more than that of [15] which defines two bits for the embedding block.

$$N_i = 2^{\lfloor \log_2 |d_n| \rfloor} \tag{3}$$

At this stage, the number of bits is:

$$N_1 = 2^{\lfloor \log_2 |5| \rfloor} \Rightarrow N_1 = 4$$
$$N_2 = 2^{\lfloor \log_2 |5| \rfloor} \Rightarrow N_2 = 4$$

Let $s = (00100111)_2$ be the secret to be hidden. According to (3), the first 4 bit (b_1) is allocated to d_1, and the second 4 bit (b_2) is for d_2. This follows (4), where b_i is the payload whose index is i.

$$b_i = N_i(s) \tag{4}$$

So, it can be determined that:

$$b_1 = (0010)_2$$
$$b_2 = (0111)_2$$

These blocks of bits are then converted to decimal values, such that:

$$b'_1 = bit2dec(b_1) \Rightarrow b'_1 = 2$$

$$b'_2 = bit2dec(b_2) \Rightarrow b'_2 = 7$$

The hiding process is done by embedding those decimal values. There are three possible conditions which determine how the embedding should be performed, depending on the trend of the sample values, as depicted in (5). Firstly, the interpolating sample ($\overline{y_n}$) is higher than the prior original sample (y_{n-1}) such as sample 2 in Figure 5(b); secondly, the interpolating sample is lower than the prior original sample such as sample 4 in Figure 5(b); thirdly, the interpolating sample is equal to the prior original sample.

$$\overline{y_n}' = \begin{cases} \overline{y_n} - b'_i, & \text{if } y_{n-1} > \overline{y_n} \\ \overline{y_n} + b'_i, & \text{if } y_{n-1} < \overline{y_n} \\ \overline{y_n}, & \text{if } y_{n-1} = \overline{y_n} \end{cases} \tag{5}$$

Therefore, from the example we obtain:

$$\overline{y_2}' = 85 + 2 \Rightarrow \overline{y_2}' = 87$$

$$\overline{y_4}' = 85 - 7 \Rightarrow \overline{y_4}' = 78$$

This is because $y_1 < \overline{y_2}$, that is: $\Rightarrow 80 < 85$. Also, $y_3 > \overline{y_4}$, that is $\Rightarrow 90 > 85$. After this stage, the resulted signal is $y' = [y_1, \overline{y_2}, y_3, \overline{y_4}, \ldots, \overline{y_n}, y_{n+1}] \Rightarrow y' = [80, 87, 90, 78, 80]$. These values, however, are relatively considerable different from those of before being embedded. This condition, in some cases, may result in significant distortion which lowers the quality of the stego audio. Therefore, it needs to decrease this difference as low as possible without reducing the number of payloads to carry. For this purpose, we deduct it to half of the difference, as depicted in (6), where $\overline{y_n}''$ represents the n^{th}-reduced embedded sample. We call this as reduced difference (RD).

$$\overline{y_n}'' = \frac{\overline{y_n}' + \overline{y_n}}{2} \tag{6}$$

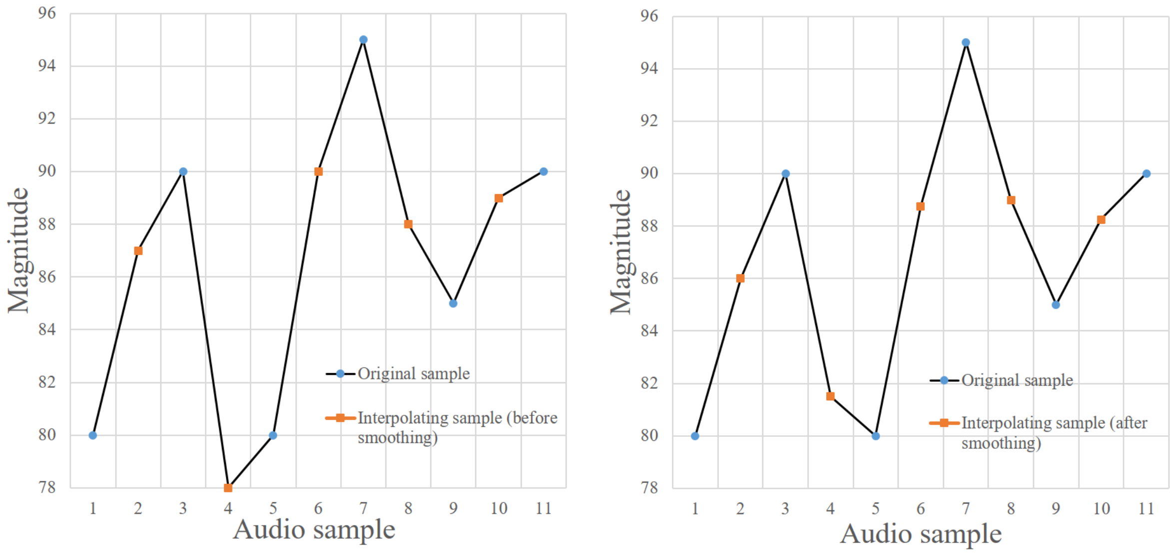
According to the example, the process of (6) leads to $\overline{y_2}'' = \frac{87+85}{2} = 86$, and $\overline{y_4}'' = \frac{78+85}{2} = 81.5$. It is shown that the difference has become lower, from 2 to 1, and from 7 to 3.5, respectively. So, at this stage, the produced stego data is $y'' = [80\ 86\ 90\ 81.5\ 80]$.

The illustration of this process is provided in Figure 6. Here, Figures 6(a) and 6(b) depict before and after the RD process is applied to the embedded signal. For further reduction, this RD process can be repeatedly performed several times. So, this stage can be formulated as (7) where $\overline{y_{i,n}}''$ is the i^{th} level of RD which is used for n^{th} reduced embedded sample.

$$\overline{y_{i,n}}'' = \frac{\overline{y_{(i-1),n}}'' + \overline{y_{(i-2),n}}''}{2} \tag{7}$$

In this case, higher level of i means that the more smoothing to the embedded signal is applied. This also means that the respective signal is more smooth. That is, $i = 1, 2, 3, \ldots$ produces 50%, 75%, 87%, ... of reduction of the difference, respectively. An illustration of this smoothing level is presented in Figure 7. Nevertheless, higher i leads to higher computation, too. Therefore, it is important to consider what level the RD is used. In this paper, we set $i = 1$.

It is worth noting that the number of samples is almost double ($2\times(i-1)$). Consequently, the sampling rate has modified from the original. The length of the voice, nevertheless, does not change because the rate has been adjusted before being converted back to an audio file.



(a) After embedding (before smoothing)

(b) After embedding (after smoothing)

FIGURE 6. Examples of smoothing processes

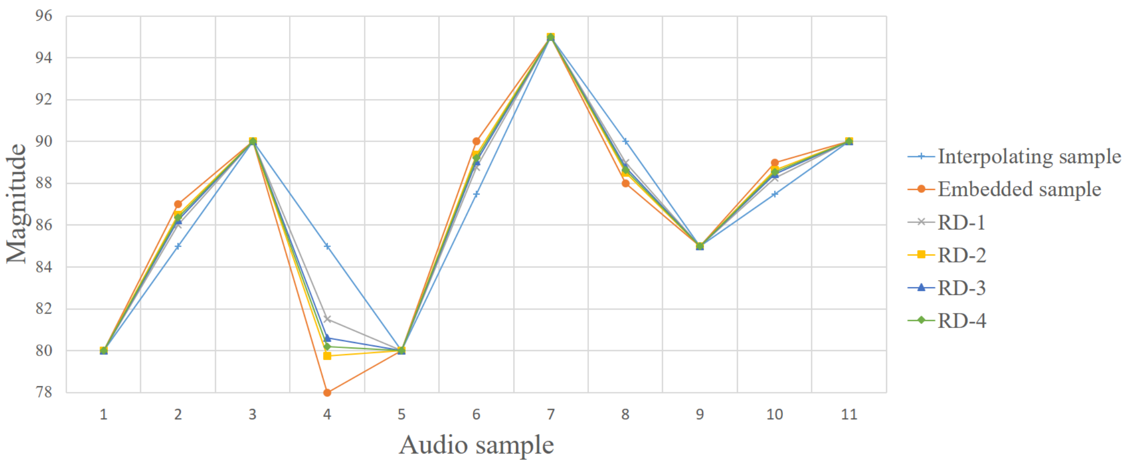


FIGURE 7. Comparison of smoothing levels

3.2. **Extraction process.** Overall, the extraction process is the reverse of the embedding one, as provided in Figure 8. It takes the stego audio as the only input. This is different from other methods such as [14] which require both the stego audio and a location map. Accordingly, this stage produces both the original cover (audio file) and the payload as the output.

Similar to that in the embedding stage, the input signal is sampled. By following the previous example, we have the input data $y'' = [80\ 86\ 90\ 81.5\ 80]$. These samples are then split according to their index number to find both the original and embedded samples, as it has been described in Section 3.1. From this process, we have two vectors: [80 90 80] and [86 81.5]. The first vector which comprises original samples is used for extracting the payload. Normalization is also required at this step in order to make the split signals readable to the audio encoder. At this stage, the sampling rate is changed back to the initial rate, which is about half of the existing. Once it is done, the original signal is fully recovered without any noise since there is no change on it.

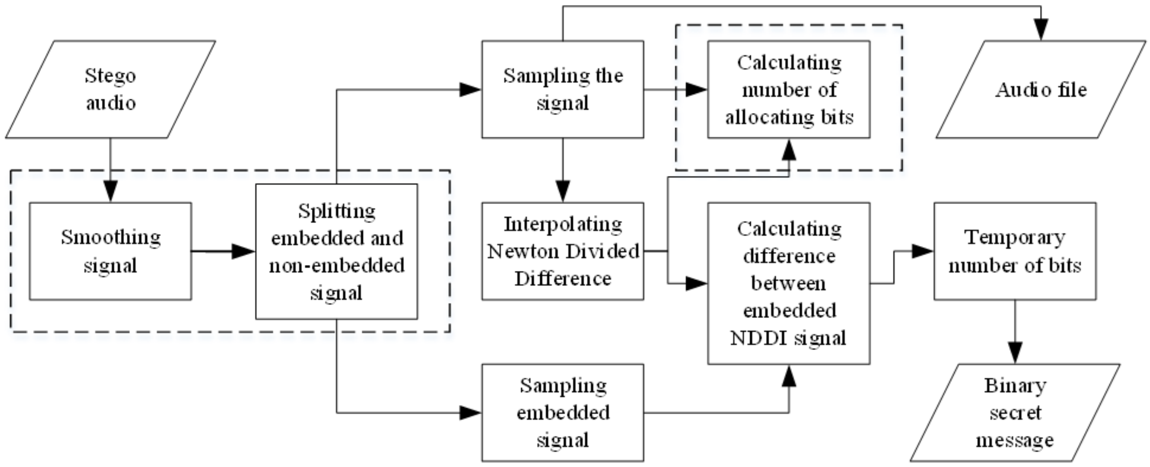


FIGURE 8. Extraction process

In order to extract the payload, the original signal is interpolated similar to that in the embedding step by using (1). This results in $y = [80\ 85\ 90\ 85\ 80]$. Because it needs to find the value of the embedded samples, the pre-smoothing of this signal must be firstly recovered. This is performed by employing (6) or (7) in case the smoothing level is 1 or n, respectively as in the previous example. Here, we find $\overline{y_n}'$ from known $\overline{y_n}''$ and $\overline{y_n}$. This can be formulated consecutively as in (8) or (9).

$$\overline{y_n}' = 2 \times \overline{y_n}'' - \overline{y_n} \tag{8}$$

$$\overline{y_{(i-1),n}}'' = 2 \times \overline{y_{i,n}}'' - \overline{y_{(i-2),n}}'' \tag{9}$$

Since the example has $n = 1$, we use (8) for the further processing step. In this case, the generated values are $\overline{y_2}' = 2 \times 86 - 85 \Rightarrow \overline{y_2}' = 87$ and $\overline{y_4}' = 2 \times 81.5 - 85 \Rightarrow \overline{y_4}' = 78$. Therefore, we have $y' = [80\ 87\ 90\ 78\ 80]$. According to the embedding process which has been explained in Section 3.1, there are three conditions of how the embedding is carried out (see (5)). By following those requirements, this extraction leads to:

$$b_1 = 87 - 85 \Rightarrow b_1 = 2$$

$$b_2 = 85 - 78 \Rightarrow b_2 = 7$$

At this step, the payload has been recovered, which is $(2)_{10}$ and $(7)_{10}$. Nevertheless, the decoder has not known how many bits to be used to represent each of this payload. This value is obtained by using (3) the same as that in the embedding step. From this, we have $N_1 = 4$ and $N_2 = 4$. So, the binary representation of the payload is $b_1 = (0010)_2$ and $b_2 = (0110)_2$. These parts of payload are then combined into a bit stream to construct $s = (00100110)_2$ which is the original data which have been hidden into the audio file. So, at the end of this process, both the original cover and the payload can be fully reconstructed.

Overall, the comparison between the proposed method and [15, 16] is provided in Table 1. It is worth noting that the embedding process in this table affects how the recovery of the audio and the extraction of the secret are carried out.

4. **Experimental Results.** Similar to other research such as [14], we use audio file in *.wav format for the experiment. This is because this type of file is not compressed, so there are no missing components. In more details, the covers for experimental data are collected from IRMAS [22]. There are 15 audio files comprising 5 instruments which consists of 3 different genres, as provided in Table 2. In this evaluation, each audio file plays 3 seconds whose sampling rate and bit depth are 44100 Hz and 16 bit, correspondingly.

TABLE 1. Comparison between [15, 16] and the proposed method

Parameters	Method of [16]	Method of [15]	Proposed method
Medium	Image	Image	Audio
Interpolation method	Neighbor mean interpolation (NMI)	$C' = C^T \times F_x \times F_y$	NDDI (taken from [17])
Number of bits which can be embedded	Dynamic $N_i = \lfloor \log_2 \lvert d_n \rvert \rfloor$	Static	Dynamic $N_i = 2^{\lfloor \log_2 \lvert d_n \rvert \rfloor}$ (explained in (3))
Embedding step	$\overline{y_{ij}}' = y_{ij} + b$	$\overline{y_{ij}}' = y_j^T - y_{ij}^T \bmod 2^k + b_l'$	$\overline{y_n}' = \begin{cases} \overline{y_n} - b_i', & \text{if } y_{n-1} > \overline{y_n} \\ \overline{y_n} + b_i', & \text{if } y_{n-1} < \overline{y_n} \\ \overline{y_n}, & \text{if } y_{n-1} = \overline{y_n} \end{cases}$ (explained in (5))
Reduce difference	Not available	Not available	Embedding (explained in (7)): $\overline{y_{i,n}}'' = \frac{\overline{y_{(i-1),n}}'' + \overline{y_{(i-2),n}}''}{2}$ Extraction (explained in (9)): $\overline{y_{(i-1),n}}'' = 2 \times \overline{y_{i,n}}'' - \overline{y_{(i-2),n}}''$

TABLE 2. Audio data set for the experiment

No	Audio Data	Instrument	Genre
1	Audio1	Cello	Country-Folk
2	Audio2	Cello	Classical
3	Audio3	Cello	Pop-Rock
4	Audio4	Acoustic guitar	Country-Folk
5	Audio5	Acoustic guitar	Classical
6	Audio6	Acoustic guitar	Pop-Rock
7	Audio7	Piano	Country-Folk
8	Audio8	Piano	Classical
9	Audio9	Piano	Pop-Rock
10	Audio10	Saxophone	Country-Folk
11	Audio11	Saxophone	Classical
12	Audio12	Saxophone	Pop-Rock
13	Audio13	Voice	Country-Folk
14	Audio14	Voice	Classical
15	Audio15	Voice	Pop-Rock

The payload is obtained from a text generator [23] whose size is varied, in order to analyze the effect of the payload size. In more details, it consists of 1kb, 5kb, 10kb, 15kb, 20kb, 25kb, 30kb, 35kb, 40kb, 45kb, 50kb and 55kb. For simplicity, this payload is stored in a text file.

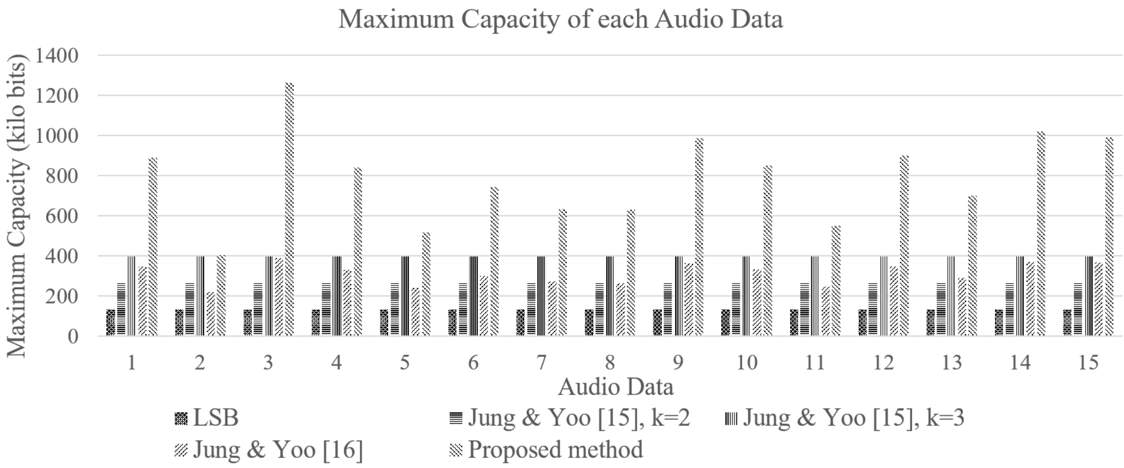


FIGURE 9. Comparison of the embedding capacity between LSB, Jung and Yoo [16], Jung and Yoo [15] with $k = 2$ and $k = 3$, and the proposed method

Overall, there are two scenarios which are used to evaluate the method. The first scenario is intended to measure the capacity (size) of the payload which can be carried by the cover whose result is represented in bits. The second scenario is to evaluate the quality of the stego audio which is obtained after the payload is embedded to the cover. In this case, this quality is calculated by using signal to noise ratio (SNR) whose result is represented in dB. This is to measure the similarity between before and after the payload is embedded to the cover.

4.1. **Capacity of the payload.** The capacity of the cover (Audio1-Audio15) which can be used to hide the payload is depicted in Figure 9. Here, the number in the x-axis represents the audio data according to Table 2, while y-axis represents the amount of bits which can be hidden. For a comparison purpose, other methods are also evaluated whose results are also plotted in Figure 9. It is shown that the LSB method has the lowest capacity. This is followed by that of [15] with $k = 2$ and $k = 3$, respectively. In [15], the value of k means the amount of bits which is carried by the cover. These three methods have stable capacity, regardless of the genre. This is because the number of bits is fixedly allocated before the embedding process starts. In this case, increasing the capacity is achieved by either lengthening the audio duration or raising the sampling rate which leads to increasing the number of samples.

Differently, the method in [16] is adaptive to the characteristic of the audio carrier. Its capacity, however, is between that of $k = 2$ and $k = 3$ of [15]. In this evaluation, the proposed method has been able to deliver the highest value. In more details, this capacity value, actually, has the same pattern with [16]; that is, the capacity follows the instrument and genre of the audio. In this proposed method, the value of the sample affects the number of bits which can be carried by the sample. The higher the difference between two consecutive samples is, the more bits can be accommodated. It is shown that in general, pop-rock is the most suitable audio to use. Contrarily, if two successive samples have relatively small difference, then only a few bit can be embedded. This happens to classical music, and some cases in country-folk as depicted in Figure 9. This also shows that the combination between cello and pop-rock is the highest, while that of cello and classical is the lowest. In this case, the instrument (i.e., cello) does not have much effect on the result.

4.2. **Quality of the stego data.** The smoothing process which is designed in the embedding step is analyzed whose experimental result is presented in Figure 10. This is the average of the SNR values after the cover is embedded by various sizes of payload. It is shown that the smoothing step has been able to increase around 3 dB of the quality of the stego data. It is predicted that this increases even higher if the smoothing step is applied several times (multi-smoothing as described in Section 3.1). It is also presented that although without using smoothing step, the quality is still higher than that of [16] in many cases. Nevertheless, the proposed method produces lower level of quality in cello-pop-rock, whose difference between successive samples is relatively high. This condition is inversely proportional to the embedding capacity.

An example measurement of the quality and capacity of various instruments with the pop-rock genre is depicted in Figure 11. It is shown that, the proposed method is able to achieve higher quality level while maintaining the capacity than the method of [16]. In general, similar results happen to country-folk and classical genres.

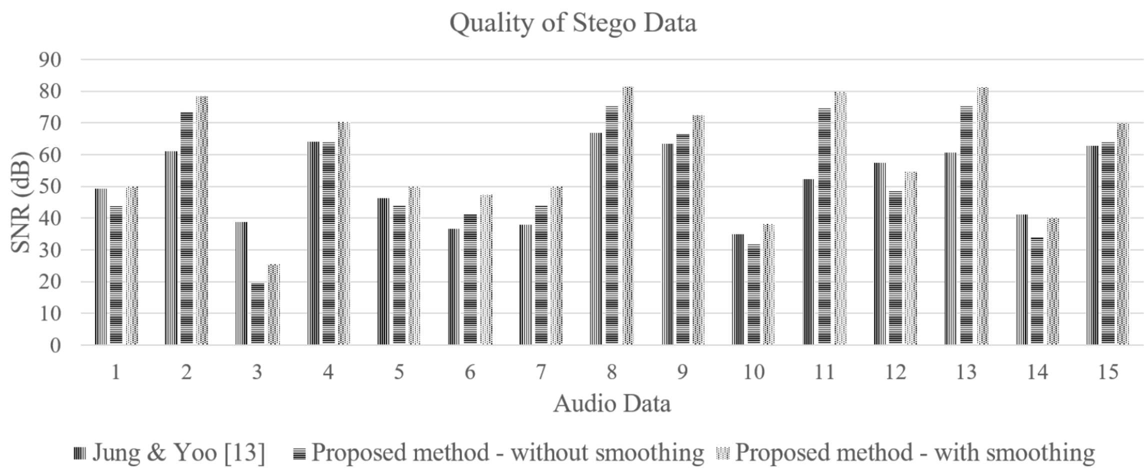
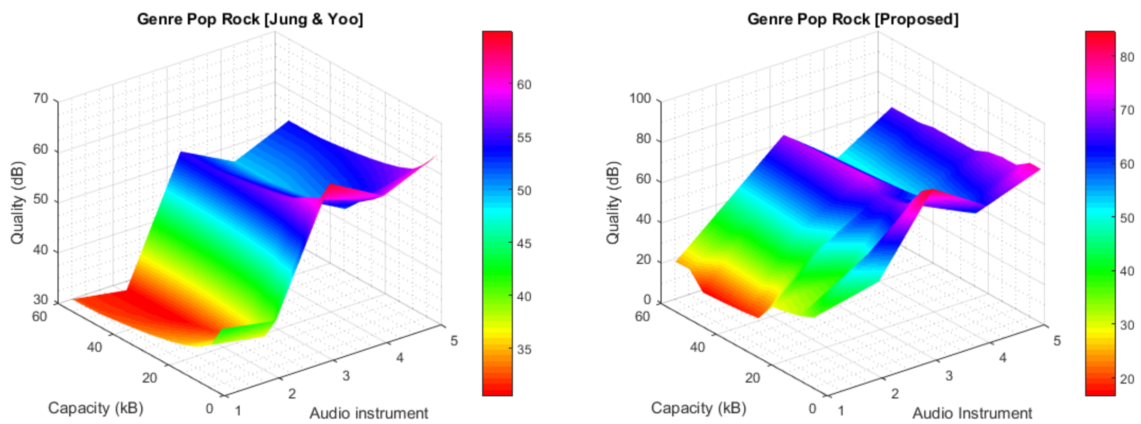


FIGURE 10. Comparison of the effect of smoothing step to the quality of the stego data between method of Jung and Yoo [16] and the proposed method

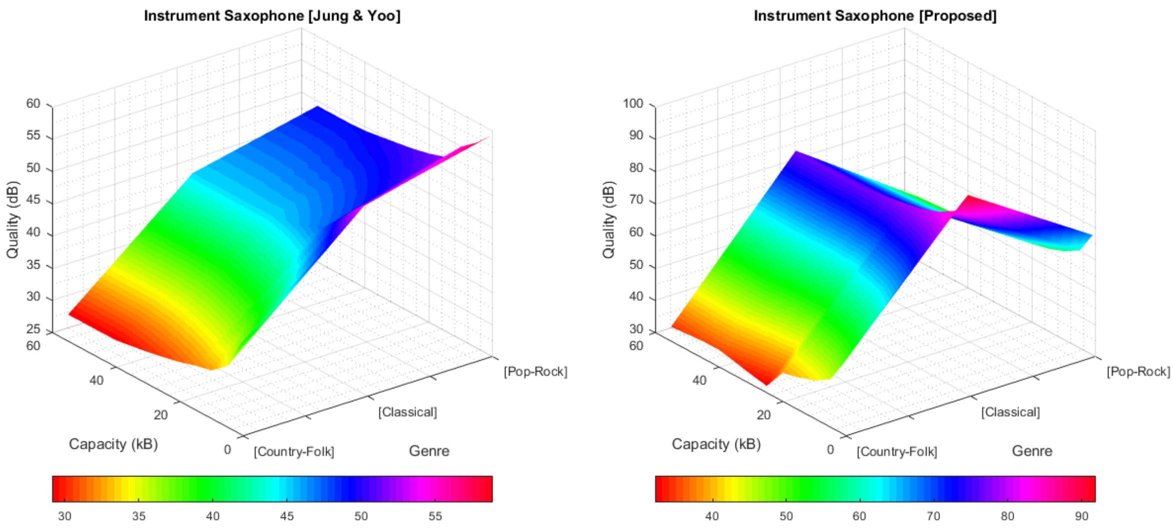


(a) SNR value of Jung and Yoo [16]

(b) SNR value of the proposed method

FIGURE 11. Comparison of SNR value of various instruments with pop-rock genre

Similar to the genre-based evaluation, the instrument-based also shows the same trend. That is, the proposed method delivers better performance than [16]. An example of this evaluation is provided in Figure 12, where saxophone is used. It can be inferred that the smoothing step which reduces the difference, works on both various genres and instruments. In the classical genre, moreover, the best quality is achieved; meanwhile, the quality of pop-rock is lower than that of classical. This is because the classical genre has relatively low difference. As in the previous description, in general, the amount of differences is proportional to the number of bits which can be embedded, and inversely proportional to the quality.



(a) SNR value of Jung and Yoo [16] (b) SNR value of the proposed method

FIGURE 12. Comparison of SNR value of various genres with saxophone instrument

5. **Conclusion.** In this paper, an adaptive data hiding method has been proposed. It has been able to increase the quality of the stego data and at the same time preserving the capacity, and vice versa. A high difference between two consecutive samples raises the capacity of the payload. This is because more spaces are available to accommodate the data. In this case, a combination between pop-rock and cello produces high capacity of payload. The smoothing technique has been able to reduce the noise caused by this higher space of payload. Furthermore, the proposed method does not need a location map which is often needed by data hiding techniques. This has made the method simpler than before.

In the future, this research can be expanded by combining it with other methods. Also, the smoothing step can be applied several times to increasing the quality by still considering the complexity. Additionally, various interpolation algorithms can be implemented to raise the size of the payload.

REFERENCES

[1] *Internet Live Stats*, http://www.internetlivestats.com/internet-users/, 2017.
[2] H. Zhang, X. Liu and C. Chen, An online mobile signature verification system based on homomorphic encryption, *International Journal of Innovative Computing, Information and Control*, vol.13, no.5, pp.1623-1635, 2017.

[3] X. Jiang, Z. Lu and X. Ding, A semi-fragile blind watermarking scheme for color images based on visual cryptography and discrete cosine transform, *International Journal of Innovative Computing, Information and Control*, vol.13, no.5, pp.1709-1719, 2017.

[4] P. Malathi and T. Gireeshkumar, Relating the embedding efficiency of LSB steganography techniques in spatial and transform domains, *Procedia Computer Science*, vol.93, pp.878-885, 2016.

[5] H. Dadgostar and F. Afsari, Image steganography based on interval-valued intuitionistic fuzzy edge detection and modified LSB, *Journal of Information Security and Applications*, vol.30, pp.94-104, 2016.

[6] D. Gruhl, A. Lu and W. Bender, Echo hiding, *International Workshop on Information Hiding*, pp.295-315, 1996.

[7] Y. Kang, K. Yang, J. Wang and Y. Liu, Multiple delayed position of echo hiding algorithm research and development, *IEEE International Conference on Signal and Image Processing (ICSIP)*, Beijing, China, 2016.

[8] L. Gang, A. N. Akansu and M. Ramkumar, MP3 resistant oblivious steganography, *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, 2001.

[9] F. Djebbar, B. Ayad, H. Hamam and K. Abed-Meraim, A view on latest audio steganography techniques, *International Conference on Innovations in Information Technology*, Abu Dhabi, United Arab Emirates, 2011.

[10] D. Renza, D. M. Ballesteros L. and C. Lemus, Authenticity verification of audio signals based on fragile watermarking for audio forensics, *Expert Systems with Applications*, vol.91, pp.211-222, 2018.

[11] M. H. A. Al-Hooti, S. Djanali and T. Ahmad, Audio data hiding based on sample value modification using modulus function, *Journal of Information Processing Systems*, vol.12, no.3, 2016.

[12] H.-Y. Fan and Z.-M. Lu, A BTC-compressed domain information hiding method based on histogram modification and visual cryptography, *International Journal of Innovative Computing, Information and Control*, vol.12, no.2, pp.395-405, 2016.

[13] J. Liu, K. Zhou and H. Tian, Least-significant-digit steganography in low bit rate speech, *IEEE ICC 2012 – Communication and Information Systems Security Symposium*, 2012.

[14] M. B. Andra, T. Ahmad and T. Usagawa, Medical record protection with improved GRDE data hiding method on audio files, *Engineering Letters*, vol.25, no.2, pp.112-124, 2017.

[15] K. Jung and K. Yoo, Steganographic method based on interpolation and LSB substitution of digital images, *Multimedia Tools and Applications*, vol.74, no.6, pp.2143-2155, 2015.

[16] K. Jung and K. Yoo, Data hiding method using image interpolation, *Computer Standards & Interfaces*, vol.31, no.2, pp.465-470, 2009.

[17] *Newton's Divided Difference Interpolation Formula*, http://mathworld.wolfram.com/.

[18] R. Ansari, H. Malik and A. Khokhar, Data-hiding in audio using frequency-selective phase alteration, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Quebec, Canada, 2004.

[19] D. Yadav, M. Agrawal and A. Arora, Performance evaluation of LSB and LSD in steganography, *IEEE the 5th International Conference the Next Generation Information Technology Summit (Confluence)*, Nioda, India, 2014.

[20] R. Fujimoto, M. Iwaki and T. Kiryu, A method of high bit-rate data hiding in music using spline interpolation, *Proc. of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2006.

[21] Y. Shi, X. Li, X. Zhang, H. Wu and B. Ma, Reversible data hiding: Advances in the past two decades, *IEEE Access*, vol.4, pp.3210-3237, 2016.

[22] *IRMAS: A Dataset for Instrument Recognition in Musical Audio Signals*, https://www.upf.edu/web/mtg/irmas/, 2017.

[23] *Lorem Ipsum*, http://www.id.lipsum.com/, 2017.