An Unified Data Embedding and Scrambling Method

Reza Moradi Rad, Student Member, IEEE, KokSheik Wong, Member, IEEE, and Jing-Ming Guo, Senior Member, IEEE

Abstract—Conventionally, data embedding techniques aim at maintaining high output image quality so that the difference between the original and the embedded images is imperceptible to the naked eye. Recently, as a new trend, some researchers exploited reversible data embedding techniques to deliberately degrade image quality to a desirable level of distortion. In this work, an unified data embedding-scrambling technique called UES is proposed to achieve two objectives simultaneously, namely, high payload, and adaptive scalable quality degradation. First, a pixel intensity value prediction method called checkerboard based prediction (CBP) is proposed to accurately predict 75% of the pixels in the image based on the information obtained from 25% of the image. Then, the locations of the predicted pixels are vacated to embed information while degrading the image quality. Given a desirable quality (quantified in SSIM) for the output image, UES guides the embedding-scrambling algorithm to handle the exact number of pixels, i.e., the perceptual quality of the embedded-scrambled image can be controlled. In addition, the prediction errors are stored at a predetermined precision using the structure side information to perfectly reconstruct or approximate the original image. In particular, given a desirable SSIM value, the precision of the stored prediction errors can be adjusted to control the perceptual quality of the reconstructed image. Experimental results confirmed that UES is able to perfectly reconstruct or approximate the original image with SSIM value greater than 0.99 after completely degrading its perceptual quality while embedding at 7.001bpp on average.

Index Terms—Data Embedding, Scrambling, Unified, Prediction, Reversible.

I. INTRODUCTION

Recent advances in Internet and mobile technologies have made the storage, access, and transmission of huge amount of multimedia information more convenient. However, the prevalence of these technologies has led to serious security concerns and handling needs. These issues have driven the research community to invent data hiding techniques. In general, a data hiding technique for digital content can be classified into two disciplines, namely: (1) data embedding, and; (2) perceptual encryption. Data embedding aims to utilize a content (e.g., image) as a venue to host external information. On the contrary, purpose of perceptual encryption (hereinafter refer to scrambling) is to make a content imperceptible by converting it into a severely distorted or meaningless form [1].

Data embedding methods can be further classified into two main categories, namely, irreversible and reversible [2]. Conventionally, both irreversible and reversible embedding techniques try to maintain the perceptual quality of the output image (i.e., embedded with data) at the highest possible level while embedding as many external information as possible into the image. For irreversible techniques, the loss of information due to the embedding process is permanent and the original image is not completely recoverable. For reversible methods, in addition to above mentioned objectives, the method must be able to perfectly reconstruct the original image. Reversibility is an attractive and beneficial feature particularly for those applications dealing with crucial and sensitive information such as medical images, military images, forensic, and valuable artwork.

Traditionally, data embedding and scrambling are explored independently. However, the availability of contents in large number and the computational power to manipulate them leads to a growing interest in integrating the features from both fields to manage and handle the contents more efficiently. Recently, some researchers attempted to join both disciplines under a single framework [3], [4], [5], [7], [8], [9]. While imperceptibility and output image quality is a matter of interest in the conventional reversible data embedding methods, it is no longer a concern in the joint approach. On the contrary, joint approach should be able to severely degrade the perceptual quality of the image by embedding external information into it while being able to reconstruct the original content. However, due to the huge number of modifications in the structure of the original content, the reconstruction process is more technically challenging in the joint approach when compared to that of the conventional data embedding methods. Despite the possible obstacles, some new interesting applications can take the advantage of the integration of data embedding and scrambling, particularly for visual contents. For example, patients information can be embedded into his/her scrambled medical image to avoid the unnecessary exposure of confidential information. A nurse (with lower access rights compared to a doctor) can administrate the embedded-scrambled image by referring to the embedded information even without knowing the actual perceptual meaning of the plaintext image, while the doctor can access both the medical image and patients information [4], [5]. In addition, embedding-scrambling holds

Manuscript received Sept. 3, 2013; revised Nov. 8, 2013; accepted Jan. 20, 2014. This paper was written under the project title Unified Scalable Information Hiding and entirely supported by the University of Malaya HIR Grant UM.C/625/1/HIR/MOHE/FSIT/01, B000001-22001.

Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

R. M. Rad and K. Wong are with the Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia. e-mail: {rcarce@siswa., koksheik}@um.edu.my

J. M. Guo is with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. e-mail: jinguo@seed.net.tw
significant importance in the ubiquitous network environment where nowadays users tend to store data in large and online using cloud storage, which is hosted by a third party [10]. In such scenario, the users privacy can be ensured through scrambling while the administrator can manage the scrambled files more efficiently using the embedded data even without knowing the actual contents of the files. Furthermore, scalability in payload and the ability to control the quality of the embedded-scrambled image as well as the reconstructed image are desirable features. These features can be deployed by the content/service provider to deliver low quality version or reveal partial detail of their products, i.e., in the embedded-scrambled form, to attract potential customers. In addition, quality of the reconstructed image can be controlled so that only the provider has the highest quality version of the image. Therefore, the scrambled-embedded image as well as the reconstructed image can be adjusted based on nature of the application in question.

In this work, a novel unified data embedding and scrambling (UES) method is proposed to degrade image quality of an image by inserting external information into it. Instead of transforming to new domain (such as Universal Domain in [10]) or finding some features suitable for data embedding, external information are inserted into selected pixel locations by direct replacement. Pixel intensity values are first predicted by using the proposed novel prediction method, and then the selected locations among the predicted pixels are vacated and replaced by the external information. Given a desirable SSIM value, UES is able to control the distortion of the output image to the targeted level. In addition, UES can operate in two modes, namely, lossless to allow perfect restoration of the original host image, or lossy to accommodate high payload. This work can be considered as an image encryption method with added feature of reversible and irreversible data embedding. Nonetheless, from the perceptive of data embedding, the proposed method is blind where the original image is not needed to extract the embedded information.

The rest of the paper is structure as follows: Section II surveys the conventional methods in reversible data embedding and scrambling. The proposed method and the discussion on achieving desirable output image quality are presented in Section III and IV, respectively. Robustness of UES against unauthorized viewing is discussed in Section V and experimental results are presented in Section VI. Section VII concludes this paper and presents some future research directions.

II. RELATED WORK

In this section, reversible data embedding methods in the literature are reviewed. In particular, reversible data embedding methods are usually proposed in the spatial domain where an image is losslessly stored as an array of raw pixel values, but also occasionally in the compressed domain and frequency domain. The reversible methods in the spatial domain can be further classified into three sub-categories, namely, compress-and-append, expansion based (EB) and histogram shifting (HS).

The first generation of reversible data embedding method was based on compress-and-append technique [11], [12], [13], [14]. The main idea of these methods is to find a losslessly compressible subset A (e.g., LSB bit-plane) in the host content (e.g., image, video and audio file), then replaces A by its compressed representation A′, and exploits the vacated space for data embedding. However, the main limitation of this approach is its limited payload (i.e., the number of embeddable bits), which is generally lower than that of DE and HS.

On the other hand, the essence of EB technique is to create some features that are representable by small magnitudes using a decorrelation function [11]. The first method of this kind, namely difference expansion (DE), was proposed by Tian [15], where the difference between two adjacent pixels is doubled. Then, if ‘0’ is to be embedded, the difference remains unchanged as an even value. Otherwise, the difference is increased by unity, changing it into an odd value. Since the LSB of the difference between every pair of adjacent pixels contains the external information, the maximum possible payload of DE is 0.5 bpp (bits per pixel). In 2007, Thodi et al. [16] proposed a prediction error expansion (PEE) based method to embed external information. They employed a predictive function to predict pixel intensity values and embed external information by expanding the prediction errors. The maximum possible payload is 1 bpp because the prediction error of every pixel of host image is exploited to embed data. Other reversible data embedding techniques that fall under this category can be found in [15], [16], [17], [18], [19]. EB methods suffer mainly from severe distortion especially when the magnitude of the features are relatively large. Furthermore, due to the expansion for data embedding purposes, the resulting pixels may be out of the range, i.e., overflow or underflow. To prevent the aforementioned problems, these methods must store a location map along with the external information to explicitly record the expandability of each pixel. As such, the effective payload is significantly used by the need to store the location map [19].

Last but not least, histogram shifting consists of three steps: 1) identify peak and zero points in the pixel value histogram of the host image; 2) vacate space for data insertion by shifting the bins between the zero and peak points, and; 3) embed information by using adjacent bin values [20]. The embedded image by HS has higher quality than EB, but its effective payload is usually lower than EB. The performance of HS technique is highly dependent on the occurrence of the most frequent pixel value and the local characteristics of the host image (i.e., peak and zero points). Moreover, HS suffers from the underflow or overflow problem where side information is needed to record if a pixel is embeddable. HS is also of high computational complexity due to preprocessing. Huang et al. [21] proposed a new histogram shifting based method by applying quadtree decomposition prior to histogram shifting, where the block size and luminance value of the maximum point are stored in block map. Next, HS is applied to each decomposed block in two rounds to embed the external information and the block map. They achieved higher payload and higher output image quality when compared to conventional HS methods.

Recently, the effort in consolidating features of both data embedding and scrambling has drawn attention of the research community. In particular, data embedding is exploited to deliberately degrade the perceptual quality of an image. In the joint
approach, quality of the embedded image and imperceptibility of the embedded data is not of one's concern, which differs from the conventional reversible data embedding methods. The relaxation in image quality requirement in the joint approach allows high payload data embedding. However, it makes the reconstruction process of the original image more technically challenging because of the huge number of modifications in the structure of the host content. Wong et al. achieved scalable scrambling effect by permuting DC coefficients within window of various sizes while the external information is embedded by exploiting the cardinality of AC coefficients in two adjacent DCT blocks [3]. Ong et al. achieved scalable visual quality degradation by mapping all pixels in the predetermined block to their corresponding mirror values for data embedding purposes [4]. Quality of the embedded-scrambled image is further degraded by shuffling pixels in blocks of predefined size. Zhang proposed a reversible data hiding method in an encrypted image by dividing the image into non-overlapping blocks where each block is further divided into two sets [5]. Three least significant bitplanes of all pixels in either set are flipped to embed external information. For other approaches exploiting the cardinality of AC coefficients in two adjacent blocks where each block is further divided into two sets [5]. Three least significant bitplanes of all pixels in either set are flipped to embed external information. For other approaches combining data embedding and scrambling, we refer the interested reader to [7], [8], [9].

III. PROPOSED METHOD

First, an efficient pixel intensity value prediction method, called Checkerboard Based Prediction (CBP), is proposed to accurately predict 75% of the pixels in the host image prior to data embedding. Then, data embedding is achieved by directly replacing the pixel values by the external information. The proposed method consists of three main processes, namely, pixel value prediction, information embedding, and reconstruction. These processes are elaborated in the following sub-sections.

A. Checkerboard Based Prediction (CBP)

Pixel intensity value prediction has been widely employed as an image processing technique for pixel decorrelation purpose in various applications including compression. Many predictors have been proposed to estimate pixel values in spatial domain. Notably, the predictor used in JPEG-LS [22] and MED (Median Edge Detection) which is also known as LOCO-I (Low-Complexity Lossless Compression) [23], [24] are the two earliest predictors proposed in 1992 and 1996, respectively. Over the years, more prediction methods are devised, such as GAP (Gradient Adjusted Prediction) [25] in CALIC (Context Adaptive Lossless Image Compression) compression system, Graham [26], DARC (Differential Adaptable Run Coding) [27], GBSW (Gradient-Based Selection and Weighting) [28], and GBTA (Gradient-Based Tracking and Adapting) [29].

Since the latter predictors are significantly more sophisticated when compared to the primitive ones such as MED, they are expected to outperform MED. However, the results in Table I suggests that the newer predictors only gain marginally in terms of prediction accuracy. As such, many researchers are still exploiting MED in their data hiding methods [11], [16], [30]. These observations have inspired us to further investigate into the potentials of the MED predictor.

MED uses three neighboring pixels, namely, N (North), NW (North West), and W (West), to predict the intensity value of the target pixel. It is able to detect horizontal and vertical edges by switching among three conditions captured by the following expression:

\[
\hat{X} = \begin{cases} 
\min(N, W), & \text{if } NW \geq \max(N, W) \\
\max(N, W), & \text{if } NW \leq \min(N, W) \\
N + W - NW, & \text{otherwise,}
\end{cases}
\]

(1)

where \(\max(N, W)\) and \(\min(N, W)\) return the maximum and minimum values between \(N\) and \(W\), respectively. In other words, MED can be regarded as a median selector where it outputs the median value among \(N, W,\) and \(N + W - NW\) [29]. One intrinsic limitation of MED is that, by the first two conditions in Eq. (1), the pixels are duplicated because the predicted value is either exactly \(N\) or \(W\), and it always falls within the range of \(\min(N, W)\) and \(\max(N, W)\) as reported in [29].

Based on empirical results, we discovered that two parameters are essential and crucial for accurate pixel value estimation. In particular, they are: (a) the distance from the target pixel - since closer neighboring pixels have higher correlation with the target pixel (refer to Fig. 1(a)), and; (b) the bidirectional information around the target pixel - since by knowing the value of both the previous and next pixels in various directions, the pattern of the target pixel can be accurately approximated (refer to Fig. 1(b)). Therefore, CBP (Checkerboard Based Prediction) is proposed as an efficient pixel estimation technique which can be independently ex-

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication. The final version of record is available at http://dx.doi.org/10.1109/TIP.2014.2302681
exploited in various applications, including decorrelation, compression, data embedding, etc.

First, every other row and column of pixels are stored to be utilized as the reference points to predict the rest of the pixels (i.e., store 25% of the pixels in raw values to predict the remaining 75%) as shown in Fig. 2(a). Then, pixel value estimation is invoked in two passes. In the first pass, pixels marked as ‘X’ in Fig. 2(b) are predicted. Next, those slightly shaded pixels marked as ‘O’ in Fig. 2(c) are predicted in the second pass. The pixel values are estimated using Eq. (2) and Eq. (3) in the first and second passes, respectively, where \( \text{rnd}() \) denotes the rounding operation. Here, the symbols W, E, N, NW, NE, S, SW, and SE denote the neighboring pixels at west, east, north, north-west, north-east, south, south-west, and south-east of target pixel ‘X’, respectively, as shown in Fig. 3. Then, the prediction errors are captured as side information and the locations of the predicted pixels are vacated for data embedding purposes.

\[
X = \begin{cases} 
\text{rnd}((NW + SE)/2), & \text{if } ||NE - SW|| > ||NW - SE|| \\
\text{rnd}((NE + SW)/2), & \text{if } ||NE - SW|| < ||NW - SE|| \\
\text{rnd}((NW + NE + SW + SE)/4), & \text{otherwise.} 
\end{cases}
\] (2)

\[
X = \begin{cases} 
\text{rnd}((W + E)/2), & \text{if } ||N - S|| > ||W - E|| \\
\text{rnd}((N + S)/2), & \text{if } ||N - S|| < ||W - E|| \\
\text{rnd}((W + N + E + S)/4), & \text{otherwise.} 
\end{cases}
\] (3)

**B. Unified Embedding-Scrambling (UES)**

The process flow of the proposed embedding-scrambling method is summarized in Fig. 4. First, the proposed CBP is utilized to predict pixel values in the image. Next, each prediction error, denoted as \( e_p \) for the rest of the presentation, is computed as \( e_p = x - x_p \) where \( x \) and \( x_p \) are the original and predicted value by CBP, respectively. Then, \( e_p \) is analyzed to decide if the corresponding pixel location is suitable for data embedding. In particular, if \( e_p \) falls within a predefined range as expressed in Eq. (4),

\[-\varepsilon \leq e_p \leq \varepsilon\] (4)

where \( \varepsilon \in \mathbb{N} \), then it is utilized for data embedding purpose. Otherwise it will be left unmodified.

In particular, we classify all pixels in an image into three categories, namely, (a) not-predicted (NP), (b) predicted but not embedded (PN) and, predicted-and-embedded (PE). Here, the set of NP consists of all the reference points in every other column and row, which are utilized to predict the rest of the pixels using the proposed CBP method. Next, PN refers to a pixel whose \( e_p \) fails the condition in Eq. (4). In other words, PN is a pixel that cannot be predicted accurately by the proposed CBP method, and it is not considered for data embedding. Thus, PN holds the original pixel value. Finally, PE refers to a pixel that satisfies Eq. (4), and it is utilized for data embedding. Here, the prediction errors \( e_p \) are stored as side information to reconstruct the original image.

In order to correctly flag the category of each pixel, an \( \alpha \)-bit structured side information per predicted pixel value is considered. The category information plays a crucial part in the extraction and reconstruction procedures as detailed in Section III-C. Note that, by the design of CBP, the location of NP pixels are known and we do not handle them. However, these locations can be shifted randomly to complicate unauthorized viewing and reconstruction, if required. To ease the discussion, we set \( \alpha = 2 \) and hence \( 2^\alpha = 4 \) cases are available to encode the category to which a prediction error belongs. In particular, ‘00’ is reserved to signify those predicted pixels which are not vacated for data embedding (i.e., PN). The remaining three cases (i.e., ‘01’, ‘10’ and ‘11’) are considered to encode three categories of the prediction error, and four possible encodings are shown in Table II using various ranges of \( \varepsilon \). Here, the median of each interval is utilized as the representative value during the reconstruction process. From another perspective, the category information can be considered as the output index of a quantization process. Therefore, the proposed UES can be utilized either as a reversible or irreversible data hiding
method, depending on the encoder/decoder design and the choice of $e_p$. In particular, if
\[ \alpha \geq \left\lceil \log_2 \left( (2 \times \varepsilon) + 1 \right) \right\rceil \] (5)
holds true, the original host image can be perfectly reconstructed. It is because, by setting $\alpha$ to a larger number, more cases are available to encode the category information of the prediction errors. Therefore, prediction errors can be captured at higher accuracy to improve the quality of the reconstructed image. Specifically, perfect reconstruction is possible whenever the length of each interval is 1. However, the effective payload uses as $\alpha$ increases since more bits are spent on coding the prediction error category. Nonetheless, to improve the storage efficiency, entropy coding can be exploited to store the side information.

Since the reference points (i.e., every other row and column of pixels) are stored directly without prediction, they can be extracted to form the subsampled image, which is of half the dimension of the original image in both the vertical and horizontal directions. In other words, the subsampled image is the down-sampled image obtained by skipping every other row and column of the original image. To further degrade quality of the output image and to increase the payload, the embedding process can be repeatedly applied to the subsampled version of the scrambled-embedded image (i.e., one layer down) to realize multiple layer embedding as shown in Fig. 5. In particular, by repetitive embedding, the values of the reference pixels in the immediate upper layer are considered and replaced by the external information. Here, the term ‘level’ also captures the number of times the embedding process is repeatedly applied to the subsampled image. In the example shown in Fig. 5, the image size at layer $L = 1$ is $16 \times 16$ pixels, and the image is subsampled by selecting the shaded locations (i.e., reference pixels). In other words, for $L > 1$, the image in the $L^{th}$ layer is formed by the shaded pixels in the $(L - 1)^{th}$ layer. This process continue until a $2 \times j$ or $j \times 2$ sub-image is achieved regardless of the value of $j$. Consequently, the combination of $\varepsilon$, $\alpha$, and $L$ constitutes the embedding-scrambling keys in the proposed method because they are all needed for scrambling, embedding, extraction, and reconstruction.

It is worth emphasizing that due to the nature of CBP, unlike expansion based and histogram shifting methods, the proposed reversible embedding-scrambling method never faces the over/under flow issue. Therefore, location map is not needed to mark the usability of each pixel.

### C. Extraction and Reconstruction

Side information is required to extract the embedded information and to reconstruct the original host image. The most important role of our side information is to tell apart if a pixel is the embedded information or it contains the actual pixel intensity value. In the case where the actual intensity value is stored (i.e., NP and PN), no information is embedded. In other words, the pixel value is either not predicted (NP) or it is predicted but not vacated for data embedding (PN) because the error fails Eq. (4). Therefore, there is no embedded information to be extracted and there is no modification on the host image which needs to be restored.

In the case where the external information is embedded (PE), the original pixel value will be estimated based on CBP and the encoded prediction error. Based on the structure of the side information, $2^\alpha - 1$ cases were utilized to capture the category of prediction error. Here, 1 is subtracted from $2^\alpha$ because one case is reserved for PE to distinguish it from NP and PN. In particular, $\alpha$ and $\varepsilon$ together determine the precision of the prediction error. For the examples given in Table II, three cases (denoted by ‘01’, ‘10’ and ‘11’) were available to encode the prediction errors where the corresponding (representative) error values are the medians.

### Table II

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>0</th>
<th>0.0</th>
<th>1.0</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\leq 4$</td>
<td>$\leq 2$</td>
<td>$\leq 0$</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>$\leq 13$</td>
<td>$\leq 0$</td>
<td>$\leq 0$</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>$\leq 22$</td>
<td>$\leq 0$</td>
<td>$\leq 0$</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>$\leq 31$</td>
<td>$\leq 0$</td>
<td>$\leq 0$</td>
<td>0</td>
</tr>
</tbody>
</table>
of the prediction error intervals in which they belong. These values will be added to the values predicted by CBP to estimate the original pixel values. Therefore, the encoding is irreversible when the ranges of values represented by each category are greater than one. For instance, in case of \(\alpha = 2\) and \(\varepsilon = 12\), if the actual prediction error is equal to 11, it will be replaced/represented by the median value of the interval, i.e., \(e_p = 7\), during reconstruction. Note that for \(\alpha = 2\), reversibility is only achieved when \(\varepsilon \leq 1\). Nonetheless, the embedded external information can be completely extracted regardless of \(\varepsilon\) and \(\alpha\).

IV. SCALABILITY

For reversible [15] and rewritable [31] data embedding methods, the quality of three images, namely, original, embedded and reconstructed, are commonly considered to gauge their performances. For the rest of this paper, these three images are refer to as \(O\), \(E\), and \(R\), respectively. One of the most important features of UES is its ability to control image quality degradation for both \(E\) and \(R\). In particular, UES is able to take a desirable level of degradation as its input query and suggest the corresponding set of parameters to produce \(E\) so that its quality is close to that of the desirable quality. Note that UES does this discriminately and not by trial and error. To quantify the distortion in visual quality, the SSIM (Structural Similarity Index Measure) [32] between the original and processed image is considered and it is computed using Eq. (6) as follows:

\[
\text{SSIM}(O, E) = \frac{(2\mu_O \mu_E + C_1)(2\sigma_{O,E} + C_2)}{(\mu_O^2 + \mu_E^2 + C_1)(\sigma_O^2 + \sigma_E^2 + C_2)},
\]

where \(\mu\), \(\sigma\), and \(\sigma^2\) denote average, variance, and covariance respectively. To derive the set of parameters, Eq. (6) is first used to one unknown variable. Suppose every pixel in \(E\) can be classified as either embedded or not-embedded pixel. Since the original host image is available, \(\mu_O\) and \(\sigma^2_O\) are known. On other hand, \(\mu_E\), \(\sigma^2_E\) and \(\sigma_{O,E}\) can be estimated from the characteristics of the external information and the original image \(O\). In particular, suppose that \(n_1\) is number of pixels in \(E\) containing external information (i.e., embedded) and \(n_2 = n - n_1\) is number of pixels in \(E\) containing the original pixel values (i.e., not-embedded). Therefore, it can be assumed that \(n_1\) pixels inherit the characteristics of the external information, while the remaining \(n_2\) pixels inherit the characteristics of \(O\). For the rest of the discussion, these two sub-images are refer to as \(E_1\) and \(E_2\), respectively.

To estimate \(\mu_E\), the weighted average of \(E_1\) and \(E_2\) is considered as follows:

\[
\mu_E = \frac{(n_1 \times \mu_{E_1}) + (n_2 \times \mu_{E_2})}{n_1 + n_2}.
\]

(7)

By the aforementioned assumptions, Eq. (7) can be expressed as:

\[
\mu_E = \frac{(n_1 \times \mu_X) + (n - n_1) \times \mu_O}{n},
\]

where \(\mu_X\) denotes the average value of the external information when it is represented in segments of byte. To estimate \(\sigma^2_E\) using the same assumptions, the following equation is considered:

\[
\sigma^2_E = \frac{(n_1 \times \sigma^2_X) + ((n - n_1) \times \sigma^2_O)}{n}.
\]

(9)

For estimating the covariance between \(O\) and \(E\) (denoted as \(\sigma(O, E)\)), we consider two sub-covariances which are denoted by \(\sigma(O, E_1)\) and \(\sigma(O, E_2)\). In particular, \(\sigma(O, E_1)\) is calculated as follows:

\[
\sigma(O, E_1) = \frac{n_1}{n_1 - 1} \sum_{i=1}^{n_1} \frac{(O_i - \mu_{O_1})(E_i - \mu_{E_1})}{\sigma_{O,E}^2},
\]

(10)

where \(O_i\) and \(E_i\) is the \(i\)th pixel in \(O\) and \(E\), respectively. Since the not-embedded pixels are identical in both \(O\) and \(E\), the term \((E_i - \mu_{E_1})\) is replaced by \((O_i - \mu_{O_1})\) and Eq. (10) becomes:

\[
\sigma(O, E_1) = \frac{n_1}{n_1 - 1} \sum_{i=1}^{n_1} \frac{(O_i - \mu_{O_1})^2}{\sigma_{O,E}^2},
\]

(11)

As a result, \(\sigma(O, E_1)\) is estimated as \(\sigma^2_O\). Similarly, \(\sigma(O, E_2)\) is estimated as follows:

\[
\sigma(O, E_2) = \frac{n_2}{(n - n_1) - 1} \sum_{i=1}^{n_2} \frac{(O_i - \mu_{O_2})(E_i - \mu_{E_2})}{\sigma_{O,E}^2},
\]

(12)

Since the external information and \(O\) are independent, \(\sigma(O, E_2) = 0\) and therefore \(\sigma(O, E)\) is approximately the same as \(\sigma^2_O\). Finally, by replacing the constants \(C_1\) and \(C_2\) and the total number of pixels \(n\), there is only one unknown variable in the equation, i.e., \(n_1\), that represents the number of embedded pixels. Therefore, \(n_1\) of the PE pixels will be replaced (by the external information) to achieve the desirable image quality.

As the order of operation, the external information is embedded into the host image by first considering the PE pixels with the smallest prediction errors, i.e., \(e_p = 0\) and \(L = 1\). If \(e_p = 0\) and \(L = 1\) do not offer enough space (i.e., locations) to embed the external information, the embedding process proceeds to the next layer, i.e., \(L = 2\), and so forth. This process is repeated with the increasing value of \(\varepsilon\) until enough locations are attained.

V. ROBUSTNESS AGAINST UNAUTHORIZED VIEWING AND DECODING

In this section, the robustness of UES against brute force attack is discussed. Here, blind and targeted attacks are considered separately. In the case of blind brute force attack, a scrambled image of \(m_1 \times m_2\) pixels is given while no information about UES is available. Let \(x \in [1, m_1 \times m_2]\) denote the number of embedded pixels, thus the length of the embedded external information is \(x \times 8\) bits for an 8-bit image. Given an embedded-scrambled image, the total number of combinations to consider in obtaining the original image is

\[
\binom{m_1 \times m_2}{x} \times 2^{(m_1 \times m_2)} \times 2^8,
\]

(13)
which is a relative huge number where there are
\[
\binom{m_1 \times m_2}{x} = \frac{(m_1 \times m_2)!}{(m_1 \times m_2 - x)! \times x!}
\]  
combinations to guess \(x\) from \(m_1 \times m_2, 2^x\) combinations to guess whether \(x\) pixels are embedded; and \(2^x\) possible values for each pixel. Therefore, to perform the blind brute force attack, the attacker needs to consider a large number of combinations to reconstruct the original image as expressed by Eq. (13). As an example, suppose an image of size \(512 \times 512\) pixels is exploited to host 40,000 bits external information. In this case, \(40,000/8 = 5000\) pixels (i.e., less than 2\% of the total pixels) are modified for embedding purposes. The attacker must consider the following number of combinations to reconstruct the original image in case of brute force attack:
\[
\binom{512 \times 512}{5000} \times 2^{(512 \times 512)} \times 2^8 \simeq 2^{622422}.
\]
Since the value obtained is too large even when modifying only 2\% of the total pixels and the number of combinations increases proportionally as the length of external information increases, it can be considered that reconstructing the original image perfectly is difficult to be practically achieved by an unauthorized party. In the case of targeted brute force attack, additional information about UES is available to the attacker on top of the embedded-scrambled image. Here, we assume that the attacker is aware that every other column and row of pixels are left unmodified at each layer. Thus, the total number of possible combinations to guess is expressed as
\[
3 \times \sum_{i=1}^{L} \binom{m_1}{2^{i-1}} \times \binom{m_2}{2^{i-1}} \times 2^9,
\]
where there are \(2^9\) possibilities for each predicted pixel value\(^1\) for each \(L\), and \(L \in [1, \log_2(\min(m_1, m_2))]\) needs to be guessed correctly. While the number of combinations captured by Eq. (16) may appear to be relative small, there are still rooms to further strengthen the proposed UES against brute force attack for reconstructing the original image. In particular, as explained in Section III-A, pixels from every other row and column are stored as the reference points for prediction purposes. By default, in each layer, the upper left pixel in every \(2 \times 2\) block is used but any of those 4 pixels can be utilized interchangeably for prediction purposes. This adds another factor of \(4!/3! \times 1!) = 4\) possibilities per \(2 \times 2\) block, or \(\prod_{l=1}^{L} 4^{\tau(l)}\) to Eq. (16) where \(\tau(l) = m_1 \times m_2/4^l\). Note that taking the block average to generate an image of lower resolution will not reveal additional information about the original image because 3 pixels in each \(2 \times 2\) block are modified for data embedding purposes.

As for the robustness against unauthorized viewing of the embedded data, the external data and side information can be encrypted prior to data embedding. In addition, instead of embedding the encrypted external data and side information in sequential manner, dummy values can be introduced or selected PE’s can be skipped to further complicate unauthorized viewing of the embedded information at the expense of lower payload.

Therefore, it can be considered that decoding the embedded data as well as reconstructing the original image perfectly are difficult to be practically achieved by an unauthorized party.

VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

As a proof of concept, the proposed method is implemented using Matlab Version 7.14.0.739 on a Intel Core i7 3.40 Mhz with 4GB of memory. Two benchmark image datasets, namely: 7 images (each of dimension \(512 \times 512\) pixels) from the USC-SIPI standard test image dataset [33] and 1338 uncompressed images (each of dimension \(384 \times 512\) pixels and converted to grayscale) in UCID (An Uncompressed Colour Image Database) [34] are considered for experimental purposes. It is verified by visual inspection that our method is able to severely degrade quality of the host image by embedding external information into it and the distortion level can be controlled by changing \(\varepsilon\) as well as \(L\) (i.e., level of processing). It is also verified that the embedded information can be completely extracted, and quality of the reconstructed image is controllable.

A. Effective Payload

First, the effective payload of UES is considered in unit of bpp (bits per pixel), which is calculated as
\[
\frac{8}{m_1 \times m_2} \times ||\{\text{PE}(\varepsilon)\}|| - ||\{\varepsilon_p\}||,
\]
where \(\{\text{PE}(\varepsilon)\}\) and \(\{\varepsilon_p\}\) denote the number pixels classified as PE using \(\varepsilon\) and the number of bits spent in storing the prediction errors \(\varepsilon_p\), respectively. While \(\{\text{PE}(\varepsilon)\}\) depends on the characteristics of the image, \(\{\varepsilon_p\}\) depends solely on the image size and \(\alpha\), i.e.,
\[
\{\varepsilon_p\} = \alpha \times \frac{3}{4} \times \sum_{i=1}^{L} \binom{m_1}{2^{i-1}} \times \binom{m_2}{2^{i-1}}.
\]
In the best scenario where all the predicted pixels are categorized as PE (i.e., qualified for data embedding), the effective payload is
\[
\frac{3}{4} \times \frac{(8 - \alpha)}{m_1 \times m_2} \times \sum_{i=1}^{L} \binom{m_1}{2^{i-1}} \times \binom{m_2}{2^{i-1}},
\]
which translates to \(~5.906\) bpp when \(\alpha = 2\) and \(L = 3\). In this example, the side information is not compressed and stored using \(\alpha = 2\) bits per predicted pixel.

To improve the effective payload, the prediction errors are packed in bytes and losslessly compressed using JPEG2000 [35] as \(\{\varepsilon_p\}_{j2k}\) where \(\{\varepsilon_p\}_{j2k} > \{\varepsilon_p\}_{j2k}\) and \(\alpha = 2\) is set. When \(\varepsilon = 1\), the average effective payload is 1.75, 2.16 and 2.26 bpp for \(L = 1, 2\) and 3, respectively. Note that in this case, the original image can be perfectly reconstructed from the embedded-scrambled image. Based on the results recorded in Table III, the effective payload steadily increases

---

\(^1\)The prediction error assumes any integer in the range \([-256, 255]\) but the actual output value lies in the range of \([0, 255]\). Hence we consider the smaller number.
as $\varepsilon$ and $L$ increase for all test images considered. This is because by employing a larger value of $\varepsilon$, more locations can be considered for data embedding, which leads to higher effective payload. For example, for $\varepsilon = 15$ and $L = 3$, the effective payload is, on average, 6.84 bpp. However, when $\varepsilon > 1$, the original image cannot be reconstructed perfectly because Eq. (5) is not satisfied. Nonetheless, the approximated image is of high similarity to the original image as reported in the next section. As expected, for image of higher spatial activity (e.g., Baboon), the prediction accuracy decreases (as reported in Table I and detailed in Section VI-B) and hence using the effective payload. On the other hand, image of lower spatial activity (e.g., Camera-man) achieves higher effective payload due higher pixel prediction accuracy.

For completion of discussion, the percentage of the gross payload occupied by the side information is also recorded in Table III. The gross payload of UES can be calculated by

$$\text{EP}(\varepsilon) \times \frac{100}{(100 - \gamma)},$$

(20)

where EP($\varepsilon$) and $\gamma$ denote the effective payload and the percentage of the side information, respectively. First, it is observed that $\gamma$ increases when Level $L$ increases regardless of $\varepsilon$. In other words, the effect of increasing side information is more dominant when compared to that of increasing the space for embedding. This agrees with the intuition that pixel prediction is less precise as the resolution decreases, and hence the ratio of PE to PN decreases. Secondly, $\gamma$ decreases as $\varepsilon$ increases when $L$ is kept constant. For example, in case Baboon, $\gamma = 16.6\%$ for $L = 1$, but $\gamma$ reduces to 3.1% for $L = 3$ and $\varepsilon = 15$. This is an expected outcome because more predicted pixels satisfy Eq. (4) and hence increasing the ratio of PE to PN, while the size of the side information remains unchanged. Last but not least, the results suggest that image of higher spatial activities (e.g., Baboon) results in larger $\gamma$ because there are more uncertainties in such image, and vice versa (e.g., Camera-Man).

### B. Image Quality

In this section, the image quality of the embedded-scrambled and reconstructed images are considered objectively using SSIM [32] and subjectively by visual inspection. The SSIM values of the embedded-scrambled images (denoted as $ES$) and the reconstructed images (denoted as $R$) from the USC-SIPI and UCID databases for various combinations of $\varepsilon$ and $L$ are reported in Table IV. In addition, Fig. 6 and 7 show the embedded-scrambled image and the corresponding reconstructed image, respectively, for various parameter settings by using Lenna as the representative example.

First, we consider the quality of the embedded-scrambled image. When $\varepsilon = 0$ and $L = 1$, the reported SSIM values in Table IV is relatively high (i.e., $\max = 0.887$ by Baboon, and $\min = 0.4143$ by Camera-Man) where the general appearance of the original image is still perceivable as suggested by Fig. 6(a). By increasing the value of $\varepsilon$ and $L$, more distortion is introduced to host image as more pixels are exploited for data embedding. Hence, the SSIM value and the perceptual quality of the embedded-scrambled image decrease. When $\varepsilon = 25$ and $L = 3$, the resulting SSIM values reach their lowest points as reported in Table IV. The appearance of the original image also becomes imperceivable as depicted in Fig. 6(f). In particular, when $\varepsilon$ increases, Eq. (4) becomes less restrictive, hence more pixels become PE. On the other hand, when $L$ increases, more pixels (in the sub-images) are considered and tested against Eq. (4), hence possibly increasing the number of PE’s. Nevertheless, the change in $\varepsilon$ has greater impact on quality of the embedded-scrambled image when compared to $L$ due to the sharp distribution of the prediction errors. In addition, the impact is more obvious for smaller $\varepsilon$. It is worth mentioning that, with the current arrangement of parameters, the quality of the embedded-scrambled image decreases (almost all the time) from left to right as $\varepsilon$ and $L$ increase.

Secondly, quality of the reconstructed image (directly from the embedded-scrambled image) is investigated. As expected, when $\varepsilon \leq 1$, the original image can be perfectly reconstructed since the reported SSIM value is unity, i.e., UES operates in lossless mode. Similar to the trend observed for the embedded-scrambled image, quality of the reconstructed image decreases as $\varepsilon$ and $L$ increase. Nonetheless, the quality of reconstructed host image is high as suggested by the SSIM values recorded in Table IV. For the same reasons in the embedded-scrambled image scenario, the impact of the change in $\varepsilon$ on image quality of the reconstructed image is greater than that of $L$. It can be observed from Table IV that quality of the reconstructed image decreases monotonically as $\varepsilon$ and $L$ increase. Fig. 7 shows the reconstructed Lenna image for various combinations of $\varepsilon$ and $L$. Although the SSIM decreases as $\varepsilon$ and $L$ increase, the perceptual quality of the reconstructed image are high and they appear identical regardless of the parameters in use. Note that the average SSIM of the reconstructed host image from the UCID dataset is always greater than 0.99 for all the combinations of parameter values considered.

### TABLE III

<table>
<thead>
<tr>
<th>Image</th>
<th>$\varepsilon = 1$</th>
<th>$\varepsilon = 3$</th>
<th>$\varepsilon = 6$</th>
<th>$\varepsilon = 15$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Lenna</td>
<td>1.98 (5.9)</td>
<td>2.61 (7.7)</td>
<td>2.80 (9.4)</td>
<td>3.78 (5.2)</td>
</tr>
<tr>
<td>Baboon</td>
<td>0.63 (16.6)</td>
<td>0.75 (22.6)</td>
<td>0.77 (27.3)</td>
<td>1.40 (8.2)</td>
</tr>
<tr>
<td>Airplane</td>
<td>1.94 (6.1)</td>
<td>2.56 (7.9)</td>
<td>2.74 (9.5)</td>
<td>3.69 (5.3)</td>
</tr>
<tr>
<td>Camera-Man</td>
<td>3.52 (3.4)</td>
<td>4.16 (5.0)</td>
<td>4.29 (6.3)</td>
<td>4.70 (2.6)</td>
</tr>
<tr>
<td>Gold-Hill</td>
<td>1.35 (8.5)</td>
<td>1.61 (12.0)</td>
<td>1.66 (14.8)</td>
<td>2.75 (4.4)</td>
</tr>
<tr>
<td>Boat</td>
<td>1.31 (8.7)</td>
<td>1.54 (12.5)</td>
<td>1.59 (15.4)</td>
<td>2.72 (4.4)</td>
</tr>
<tr>
<td>Pepper</td>
<td>1.55 (7.5)</td>
<td>1.92 (10.2)</td>
<td>1.98 (12.7)</td>
<td>3.22 (5.2)</td>
</tr>
<tr>
<td>UCID (average)</td>
<td>2.24 (18.1)</td>
<td>2.71 (22.1)</td>
<td>2.80 (36.6)</td>
<td>3.59 (4.3)</td>
</tr>
</tbody>
</table>

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication. The final version of record is available at http://dx.doi.org/10.1109/TIP.2014.2302681

Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.
C. Scalability

Next, scalability in terms of quality of the embedded-scrambled image is verified. Specifically, the ability of the proposed method in generating an output (i.e., embedded-scrambled) image with the desirable level of image quality is verified by using 13 SSIM values as the queries, which decreases from 0.65 to 0.05 with a constant step size of 0.05. In other words, given a desirable SSIM value, UES estimates the number of pixels to be replaced in order to achieve the specified quality as described in Section IV. Specifically, a combination of $\varepsilon$ and $L$ which offers at least $n_1$ pixels is selected by UES. By embedding external information into exactly $n_1$ pixels, approximately the desirable amount of distortion is introduced into the host image.

Fig. 8 plots the query number against the average SSIM of the embedded-scrambled images from UCID. For reference purposes, the graph of query number against 13 query SSIM values are also plotted on the same graph. Results indicate that the proposed method successfully generates image with SSIM which is, on average, $\sim 0.025$ off from the queried SSIM value. In addition, proposed method allows the user to decide on lossy or lossless mode of operation by tuning $\varepsilon$, where it guarantees complete reversibility in lossless mode.

These observations verified that the proposed UES can control image quality of both the embedded-scrambled and the reconstructed images.

D. Comparison with Existing Methods

1) Pixel Value Predictor: First, the performance of the proposed CBP is compared to the six conventional prediction methods considered, namely, MED [23], [24], GAP [25], Graham [26], DARC [27], GBSW [28], and GBTA [29]. Table I records the prediction accuracy in terms of MAE (mean absolute error) for predicting 7 standard test images
and images from the UCID dataset [34]. Here,

\[
\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \|x_i' - x_i\|, \tag{21}
\]

where \(x_i\) and \(x_i'\) denote the original and predicted pixel values, respectively. It is observed that, regardless of the prediction method, the prediction accuracy decreases when handling image of high spatial activity (e.g., Baboon), and vice versa (e.g., Camera-man). However, results indicate that the prediction accuracy of the proposed CBP is, on average, two times better than (i.e., MSE is half of) the widely used MED. Furthermore, CBP always outperforms all the conventional prediction methods considered regardless of the test image in question. Therefore, it can be concluded that the proposed CBP achieves high prediction accuracy and hence it is a viable prediction method.

Secondly, the degree of parallelism is discussed. Due to the design of the conventional predictors, the earlier predicted pixels are required to carry out the (immediate) next prediction process, i.e., the dependency is cascaded from the last pixel to the first pixel in the image in the worst case scenario. Therefore, the architecture of the conventional predictors has a low degree of parallelism. In contrast, the prediction process in CBP does not rely on the previously predicted pixel values, and hence the prediction process can be executed in parallel. Nonetheless, the prediction process in the first pass must be carried out prior to the second pass. Therefore, the predictions in each pass can be executed in parallel.

Next, the processing time required to process an image by each prediction method is reported in Table V. It is observed that CBP requires, on average, \(\sim 0.221\) and \(\sim 0.229\) seconds to process an image of dimension \(512 \times 512\) (USC-SIPI) and \(384 \times 512\) (UCID) pixels, respectively. In addition, thanks to high degree of parallelism in CBP, a noticeable speedup by a factor of 10 in processing time is achieved when CBP is implemented in parallel mode. When executing in parallel mode, CBP requires, on average, \(\sim 0.020\) and \(\sim 0.024\) seconds to process an image in the USC-SIPI and UCID datasets, respectively. Therefore, these results suggest that CBP is even slightly faster than those low complexity methods such as Graham, GAP, and DARC, and significantly faster when compared to complex methods such as GBSW and GBTA.

For completion of discussion, we show the efficiency of the proposed CBP by reimplementing PEE (Prediction Error Expansion) once again using CBP instead of MED, and the results are depicted in Fig. 9. Results indicate that the performance of PEE method can be improved when CBP is utilized in place of MED. In particular, for the same SSIM, PEE⊕CBP archives higher payload, or in other words, when achieving the same payload, PEE⊕CBP produces image of higher quality than that of PEE⊕MED. Thus, these observations justify the need and superiority of the proposed CBP, which is a general pixel value prediction method applicable for various purposes.

2) Data Embedding Method: For completion of discussion, we compare UES to the conventional reversible and irreversible methods. Note that the results shown in Table IV for \(\varepsilon = 0\) and 1 regardless of \(L\) reflect the performance of the proposed UES as a reversible data hiding method where the original host image can be perfectly reconstructed (because \(\alpha = 2 \geq \lceil \log_2((2 \times \varepsilon) + 1) \rceil = 2\)). The rest of the cases (i.e., \(\varepsilon > 1\)) suggest the performance of the proposed UES as an irreversible method.

For reversible data embedding, the achievable payload (bpp) by UES and the conventional methods [15], [16], [20], [21] are recorded in Table VI. It is observed that when UES is employed as a reversible method, its effective payload is significantly higher (at least 4 folds) than the conventional methods considered. However, it should be noted that the conventional methods aim at maintaining image quality while the proposed
method is designed to distort image quality. From a different perspective, the results also indicate how much payload can be gained when the imperceptibility criterion is relaxed. In other words, the recorded values show the payload gained by the proposed reversible scrambling-embedding method (i.e., ignoring imperceptibility) over the conventional reversible data embedding methods (i.e., emphasizing imperceptibility). It is concluded that a gain of more than 2.1 bpp is achieved when the imperceptibility criterion is relaxed. Note that the payload can be further increased significantly by sacrificing the reversibility property.

On the other hand, for irreversible data embedding, the maximum theoretical payload for an 8-bit image is 8bpp, which is achieved by replacing all bit planes by the information to be embedded. However, neither the original image nor its approximation can be reconstructed. Nonetheless, the proposed UES can achieve, on average, 7.001bpp for the UCID image dataset (when \( \varepsilon = 25 \) and \( L = 3 \)) while being able to approximate the original image with high similarity (i.e., SSIM \( \geq 0.99 \)). Therefore, by relaxing the condition to maintain image quality as required in the conventional data embedding methods, the proposed UES can achieve high effective payload while severely distorting the image quality, thus achieving the purpose of embedding-scrambling simultaneously.

3) Embedding-Scrambling Method: Here, we consider the performance of the proposed method against conventional embedding-scrambling methods. Zhang proposed a separable reversible data hiding in encrypted images [5]. Zhang’s method does not guarantee complete reversibility because it relies on the correlation among neighboring pixels. Furthermore, [5] suffers from very limited payload, i.e., 0.0001bpp. Nonetheless, it should be noted that [5] is separable, where encryption can be carried out prior to data embedding and vice versa, to generate the same embedded-scrambled image.

Fujiyoshi also proposed a separable reversible data hiding in encrypted images [6]. While ensuring perfect reconstruction of the host image, [6] allows the user to take seven different actions, including: (a) payload extraction; (b) partially decrypt the image; (c) completely decrypt the image; (d) recover the original images, and; their combinations. However, reversibility and the aforementioned features in [6] are achieved at the expense of very limited payload, i.e., 0.019 bpp. On the other hand, although UES is not separable and has less features when compared to [6], it is able to embed up to 2.80 bitplanes on average into the host image while ensuring complete reversibility.

Ong et al. [4] proposed a unified data embedding and scrambling method while achieving scalable visual quality degradation. However, it suffers from unbalanced distortion due to the uneven distribution of reflective and non-reflective blocks as well as the block-type dependent operations. This leads to possible leakage of perceptual information about the original host image when the parameters are not chosen appropriately. To overcome the aforementioned problem, pixels are permuted within each block and the final output image is depicted in Fig. 10. The main advantage of UES over [4] is the degree of scalability in controlling quality of the embedded-scrambled image as well as the reconstructed image. Given a desirable SSIM for the output image, UES automatically constructs the output image with similar SSIM value by estimating how many pixels are needed to be replaced by the external information. However, in [4], the desirable quality is achieved by trial and error. Although [4] can introduce a wide range of distortion levels to the host image, it is not practical because it lacks the control on quality of the embedded-scrambled image. Furthermore, UES has better scalability in terms of payload (i.e., up to 7.31bpp) while insignificantly degrading quality of the reconstructed image. In contrast, [4] only achieves 2.56bpp on average and it cannot control quality of the reconstructed image. A functional comparison of UES and the existing data embedding-scrambling are provided in Table VII.

### Table VII

<table>
<thead>
<tr>
<th>Method</th>
<th>Effective payload (bpp)</th>
<th>Embedding-Scrambling</th>
<th>Reversibility</th>
<th>Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang [5]</td>
<td>0.0001</td>
<td>Separable</td>
<td>Irreversible*</td>
<td>No</td>
</tr>
<tr>
<td>Fujiyoshi [6]</td>
<td>0.019</td>
<td>Separable</td>
<td>Reversible only</td>
<td>No</td>
</tr>
<tr>
<td>Ong et al. [4]</td>
<td>2.564</td>
<td>Unified</td>
<td>Reversible only</td>
<td>Yes</td>
</tr>
<tr>
<td>UES</td>
<td>2.801</td>
<td>Unified</td>
<td>Both (controllable)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*There is no guarantee for reversibility

### Table VI

<table>
<thead>
<tr>
<th>Method</th>
<th>Payload (bpp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE [5]</td>
<td>0.285</td>
</tr>
<tr>
<td>PEE [6]</td>
<td>0.669</td>
</tr>
<tr>
<td>HS1 [20]</td>
<td>0.031</td>
</tr>
<tr>
<td>HS2 [21]</td>
<td>0.107</td>
</tr>
<tr>
<td>UES</td>
<td>2.801</td>
</tr>
</tbody>
</table>

Fig. 10. A comparison between UES and Ong et al. [4]. Here, the quantitative result for Ong et al. [4] is presented for the case of block size = \( 8 \times 8 \) for embedding and scrambling. For UES, \( \varepsilon = 14 \) and \( L = 2 \).


VII. CONCLUSION

In this work, an unified data embedding and image scrambling method called UES was proposed. UES is able to severely distortion perceptual quality of the host image by means of data embedding. A pixel value predictor was proposed to predict pixel values, where the predicted pixels were replaced by the external information to be embedded. The proposed pixel prediction method achieved accurate prediction, up to twice the accuracy of the conventional methods considered. Then, the predicted pixels were selected based on their prediction errors to embed external information while their prediction errors were stored as side information. Experimental results confirmed that the proposed method is able to completely degrade the perceptual quality of the host image by embedding external information into it. It was also verified that high payload of $> 6.39$ bpp was achieved (e.g., effective payload for Lena is 7.26 bpp when ε = 15 and ‘level’ = 3) when operating in the lossy mode. Experimental results also indicated that the proposed method is able to recover the host image after imposing severe degradation by embedding huge amount of external information, with an average reconstruction quality of SSIM $\geq 0.99$ for the UCID image dataset.

As future work, we want to apply the proposed method to the frequency domain in order to predict frequency coefficients for embedding external information [36]. We also want to apply CBP to the compressed domain.

REFERENCES


Copyright (c) 2014 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.
Reza Moradi Rad received the B.S. degree in computer engineering from University of Guilan, Rasht, Iran, in 2012. He is currently pursuing the master degree in computer science at Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia, under the supervision of Dr. KokSheik Wong. His current research interests include information hiding, data compression and multimedia signal processing. Reza is a student member of the IEEE.

KokSheik Wong received the B.S. and M.S. degrees in both computer science and mathematics from Utah State University, USA, in 2002 and 2005, respectively. In 2009, he received the Doctor of Engineering degree from Shinshu University, Japan, under the scholarship of Monbukagakusho. In 2010, he joined the Faculty of Computer Science and Information Technology, University of Malaya, Malaysia, where he is currently a senior lecturer. He is a member of Centre for Image and Signal Processing (CISP), University of Malaya, where he leads the Multimedia Signal Processing and Information Hiding (MSPHIH) group. His research interests include information hiding, steganography, watermarking, multimedia perceptual encryption, multimedia signal processing, and their applications. Dr. Wong is a member of IEEE.

Jing-Ming Guo (M06-SM10) received the Ph.D. degree from the Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan, in 2004. He is currently a Professor with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include multimedia signal processing, biometrics, computer vision, and digital halftoning. Dr. Guo has acted as the Technical program Chair for IEEE International Symposium on Intelligent Signal Processing and Communication Systems in 2012, IEEE International Symposium on Consumer Electronics in 2013, and IEEE International Conference on Consumer Electronics in Taiwan in 2014. He has been invited as a lecturer for the IEEE Signal Processing Society summer school on Signal and Information Processing in 2012 and 2013. He has been elected as the Chair of the IEEE Taipei Section GOLD group in 2012. He has served as a Guest Co-Editor of two special issues for Journal of the Chinese Institute of Engineers and Journal of Applied Science and Engineering. He serves on the Editorial Board of the Journal of Engineering, The Scientific World Journal, International Journal of Advanced Engineering Applications, and Open Journal of Information Security and Applications. Currently, he is Associate Editor of the IEEE Transactions on Multimedia, IEEE Signal Processing Letters, the Information Sciences, and the Signal Processing.

Dr. Guo is a senior member of the IEEE, and a Fellow of IET. He has been promoted as a Distinguished Professor in 2012 for his significant research contributions. He received the Outstanding youth Electrical Engineer Award from Chinese Institute of Electrical Engineering in 2011, the Outstanding young Investigator Award from the Institute of System Engineering in 2011, the Best Paper Award from the IEEE International Conference on System Science and Engineering in 2011, the Excellence Teaching Award in 2009, the Research Excellence Award in 2008, the Acer Dragon Thesis Award in 2005, the Outstanding Paper Awards from IPPR, Computer Vision and Graphic Image Processing in 2005-2006 and 2013, and the Outstanding Faculty Award in 2002 and 2003.