www.ietdl.org

Published in IET Image Processing
Received on 18th December 2013
Revised on 11th June 2014
Accepted on 19th June 2014
doi: 10.1049/iet-ipr.2014.0321

ISSN 1751-9659

Semantic image compression based on data hiding

Xinpeng Zhang[1], *Weiming Zhang*[2]

[1]*School of Communication and Information Engineering, Shanghai University, Shanghai, People's Republic of China*
[2]*School of Information Science and Technology, University of Science and Technology of China, Hefei, People's Republic of China*
E-mail: xzhang@shu.edu.cn

Abstract: This study proposes a novel scheme of semantic image compression. A compressor firstly creates a compact image by gathering a part of pixels in an original image, and calculates estimation errors of the rest pixels. Then, a compressed image is produced by embedding the estimation errors into the compact image using data hiding techniques. This way, the compressed image are made up of a small number of pixel values, and the original content is still visible roughly through the compressed image without any decompression tool. If a decompression tool is available, a user may reconstruct a high quality image with original size by exploiting the embedded data. Because the proposed scheme is compatible with reversible and non-reversible data hiding techniques, either the lossy or lossless semantic compression can be performed. With different parameters, the qualities of compressed and decompressed images vary. Furthermore, the smoother the original image content, the better is the compression–decompression performance.

1 Introduction

The purpose of image compression methods, such as JPEG and JPEG 2000, is to reduce the data amounts of digital images when keeping satisfactory quality. In most compression applications, since the structure and representation of image data have been changed for removing the redundancy, that is, the image format has been altered, some corresponding decompression tools are necessary to reconstruct the image content from compressed data. In this work, we will focus on another type of image compression, called as semantic image compression, in which the compressed images with significantly smaller data amounts are also made up of pixel values. In other words, the formats of the original and compressed images are same, while the content of them are similar. That means the original content is still visible roughly through compressed images without any decompression tool. If the decompression tool is available, the decompressed images with high visual quality can be obtained.

A coding method of semantic image compression has been presented to convert an 8-bit grey-level image into a 7-bit grey-level version with acceptable visual quality, and the original content can be recovered without any error from the 7-bit version [1]. The theoretical bounds of lower-grey-level compression for both lossless and lossy manners have been also studied in [2]. However, since the pixel value is usually stored within one byte, that is, 8 bits, the applications of [1, 2] are limited. This work proposes a novel semantic image compression method, in which each pixel in compressed image is also represented as 8 bits and the image size is reduced. On decompression side, a high quality image with original size can be reconstructed. Fig. 1 gives a brief illustration of this work.

Data hiding techniques, which aim to embed additional data into host signals, are employed in this work to carry some data used for decompression within compressed images. In data hiding scenarios, the embedding operation will result in distortion in the host signals, so that it is always desired to lower the distortion with a given payload or to maximise the embedded payload with a given distortion level, in other words, to achieve a good rate-distortion performance. The theoretical bounds of rate-distortion performance have been studied in [3, 4], and a number of practical data hiding methods with good performance have been developed. When viewing the least significant bits (LSB) plane of host images as an available data space for accommodating additional data, the different patterns of cover data corresponding to a same syndrome can be used to represent an identical type of additional data, and a data-hider may modify the original cover data to the nearest pattern mapping the additional data to be hidden [5, 6]. This way, the rate-distortion performance is close to the theoretical bound of binary embedding method. In [7], Wet Paper Codes [8] and Hamming codes are introduced to derive a family of data hiding methods from an existing binary embedding method. If the binary embedding method is near optimal, the performance of the derived data hiding methods will still stay close to the theoretical limit [9].

When the distortion caused by data embedding, no matter how small it is, is unacceptable, it is imperative to embed the additional data in a reversible manner so that the original contents can be perfectly restored after extraction of the hidden data. The reversible data hiding (RDH) techniques can be roughly classified into three types:

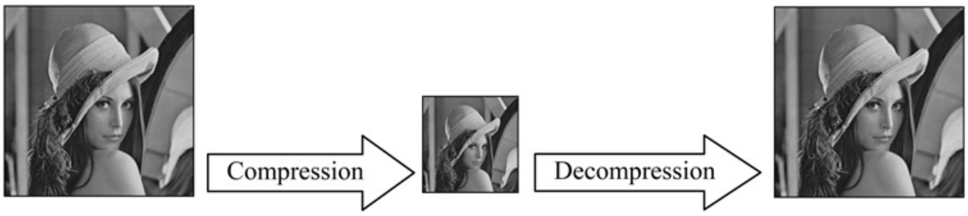

Fig. 1 *Brief illustration of semantic image compression in this work*

lossless compression-based methods, difference expansion (DE) methods and histogram shift (HS) methods. The lossless compression-based methods make use of statistical redundancy of the host signals by performing lossless compression in order to create a spare space to accommodate additional data [10, 11]. In the DE method [12], differences between two adjacent pixels are doubled so that a new LSB plane without carrying any information of the original is generated. The hidden data together with a compressed location map derived from the property of each pixel pair, but not the host information itself, is embedded into the generated LSB plane. Various techniques have been introduced into DE algorithm to improve its performance, including generalised integer transform [13, 14], pixel value prediction mechanism [15], histogram modification operation [16], prediction of location map [17], simplification of location map [18, 19] and improvement of compressibility of location map [20]. A data-hider can also employ HS mechanism to realise reversible data hiding. A typical HS method presented in [21] utilises the zero and peak points of the histogram of an image and slightly modifies the pixel grey scale values to embed data into the image. The HS mechanism can also be implemented in the differences between sub-sampled images [22] and the prediction errors of host pixels [23, 24]. In addition, several good prediction approaches [25, 26] and an optimal rule of value modification under a payload-distortion criterion [27] have been introduced to improve the performance of reversible data hiding.

This work employs the data hiding techniques to embed some data for decompression into a down-sampled version of original image to generate a compressed image. Then, the compressed image possesses similar content and its size is smaller then the original one. On decompression side, with the aid of extracted data, we may obtain a reconstructed image with original size and high quality. Since both the reversible and non-RDH techniques may be used, either the lossy or lossless semantic compression can be performed with the proposed scheme. The rest of this paper is organised as follows. The compression and decompression procedures are presented in Sections 2 and 3, respectively. Section 4 gives the experimental results and Section 5 concludes the paper.

2 Semantic compression procedure

In the semantic compression procedure, we firstly create a compact image by gathering a part of pixels in original image, and collect the estimation errors of the rest pixels. Then, the compressed data of the estimation errors are embedded into the compact image using the LSB replacement and/or RDH techniques to produce a compressed image with small size and similar content.

Assume an original image $\boldsymbol{P}$ is in uncompressed format and the pixel values are within [0, 255]. Denote the numbers of the rows and the columns in the original image as N_1 and N_2, and the pixels as $p(n_1, n_2)$ where $1 \le n_1 \le N_1$ and $1 \le n_2 \le N_2$. Firstly, a small image $\boldsymbol{G}$ is generated by employing the nearest neighbour interpolation method with a given scaling ratio r $(0 < r < 1)$. The number of rows and columns in $\boldsymbol{G}$ are

$$M_1 = \text{round}\,(N_1 \cdot r) \tag{1}$$

$$M_2 = \text{round}\,(N_2 \cdot r) \tag{2}$$

and the pixel values in $\boldsymbol{G}$ are

$$\begin{aligned} g(m_1, m_2) &= p\big(\text{round}(m_1/r), \text{round}(m_2/r)\big), \\ 1 \le m_1 &\le M_1, \quad 1 \le m_2 \le M_2 \end{aligned} \tag{3}$$

where the operation round $(\cdot)$ returns the nearest integer. In other words, a part of pixels in the original image $\boldsymbol{P}$ are picked to form the compact image $\boldsymbol{G}$. We call the picked pixels in $\boldsymbol{P}$ as seed pixels, and the pixels that are not picked as default pixels.

For each default pixel $p(n_1, n_2)$, calculate

$$m_1^U = \lfloor n_1 \cdot r \rfloor \tag{4}$$

$$m_1^D = \lceil n_1 \cdot r \rceil \tag{5}$$

$$m_2^L = \lfloor n_2 \cdot r \rfloor \tag{6}$$

$$m_2^R = \lceil n_2 \cdot r \rceil \tag{7}$$

where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ take the nearest integers towards infinity and negative infinity, respectively, and

$$n_1^U = \text{round}\big(m_1^U/r\big) \tag{8}$$

$$n_1^D = \text{round}\big(m_1^D/r\big) \tag{9}$$

$$n_2^L = \text{round}\big(m_2^L/r\big) \tag{10}$$

$$n_2^R = \text{round}\big(m_2^R/r\big) \tag{11}$$

That implies $p\big(n_1^U, n_2^L\big), p\big(n_1^U, n_2^R\big), p\big(n_1^D, n_2^L\big)$ and $p\big(n_1^D, n_2^R\big)$

are the seed pixels closest to the default pixel $p(n_1, n_2)$, and

$$g\left(m_1^U, m_2^L\right) = p\left(n_1^U, n_2^L\right) \quad (12)$$

$$g\left(m_1^U, m_2^R\right) = p\left(n_1^U, n_2^R\right) \quad (13)$$

$$g\left(m_1^D, m_2^L\right) = p\left(n_1^D, n_2^L\right) \quad (14)$$

$$g\left(m_1^D, m_2^R\right) = p\left(n_1^D, n_2^R\right) \quad (15)$$

We divide 8 bits of each pixel value into $(8 - T)$ highest bits and T lowest bits where T is an integer within [0, 7] and shared by both the compression and decompression sides, and denote the value of $(8 - T)$ highest bits of each pixel in $\boldsymbol{G}$ as

$$g_{\mathrm{H}}\left(m_1, m_2\right) = \left\lfloor g\left(m_1, m_2\right)/2^{\mathrm{T}} \right\rfloor \quad (16)$$

Then, estimate the default pixel $p(n_1, n_2)$ using the values of $g_{\mathrm{H}}\left(m_1^U, m_2^L\right)$, $g_{\mathrm{H}}\left(m_1^U, m_2^R\right)$, $g_{\mathrm{H}}\left(m_1^D, m_2^L\right)$, $g_{\mathrm{H}}\left(m_1^D, m_2^R\right)$ and the bilinear interpolation method (see (17))

where (see (18 and 19))

The estimation error is

$$e(n_1, n_2) = p(n_1, n_2) - \mathrm{round}\left[\bar{p}(n_1, n_2)\right] \quad (20)$$

In the following, we will compress the estimation errors and embed the compressed data into the compact image $\boldsymbol{G}$.

Denote the minimal and maximal values of all estimation errors as $V_{\min}$ and $V_{\max}$, and the number of estimation errors being i as h_i ($V_{\min} \le i \le V_{\max}$). Hence, the probability of estimation error being i is

$$\gamma_i = h_i / \sum_{j=V_{\min}}^{V_{\max}} h_j \quad (21)$$

We intend to divide the range $[V_{\min}, V_{\max}]$ into K sections: $[\alpha_1, \beta_1]$, $[\alpha_2, \beta_2]$, …, $[\alpha_K, \beta_K]$ where $\alpha_1 = V_{\min}$, $\beta_K = V_{\max}$, and $\alpha_{k+1} = \beta_k + 1$ ($k = 1, 2, \ldots, K - 1$), and use the index of each section to represent the estimation errors falling into the section. This way, the set of estimation errors is converted into a set of section indices, and the amount of required data for representing the estimation errors is

$$A = -\sum_{i=V_{\min}}^{V_{\max}} h_i \cdot \sum_{k=1}^{K} \left[\sum_{i=\alpha_k}^{\beta_k} \gamma_i \cdot \log_2 \left(\sum_{i=\alpha_k}^{\beta_k} \gamma_i \right) \right] \quad (22)$$

That implies the estimation errors are compressed. If we regard the estimation errors in a section $[\alpha_k, \beta_k]$ as a median $(\alpha_k + \beta_k)/2$, the distortion would be

$$D_k = \sum_{i=\alpha_k}^{\beta_k} h_i \cdot \left(i - \frac{\alpha_k + \beta_k}{2} \right)^2 \quad (23)$$

and the total distortion would be

$$D = \sum_{k=1}^{K} D_k \quad (24)$$

Clearly, A and D are determined by the section division $[\alpha_1, \beta_1]$, $[\alpha_2, \beta_2]$, …, $[\alpha_K, \beta_K]$. With the following steps, we may obtain a series of ways of section division to keep a low data amount with given distortion level or keep a low distortion with given data amount.

Step 1: At the beginning, we define $(V_{\max} - V_{\min} + 1)$ sections, each of which contains only one value. In other words, $\alpha_k = \beta_k = k + V_{\min} - 1$ $(1 \le k \le V_{\max} - V_{\min} + 1)$. And let

$$A = -\sum_{i=V_{\min}}^{V_{\max}} h_i \cdot \sum_{i=V_{\min}}^{V_{\max}} \left(\gamma_i \cdot \log_2 \frac{1}{\gamma_i} \right) \quad (25)$$

$$D = 0 \quad (26)$$

Step 2: Denote the number of current sections as K. Considering two adjacent sections $[\alpha_k, \beta_k]$ and $[\alpha_{k+1}, \beta_{k+1}]$

$$\bar{p}\left(n_1, n_2\right) = 2^{\mathrm{T}} \cdot \begin{cases} \bar{p}^L\left(n_1, n_2\right) & , \quad \text{if } n_2^L = n_2^R \\ \dfrac{n_2 - n_2^L}{n_2^R - n_2^L} \cdot \bar{p}^R\left(n_1, n_2\right) + \dfrac{n_2^R - n_2}{n_2^R - n_2^L} \cdot \bar{p}^L\left(n_1, n_2\right) & , \quad \text{if } n_2^L < n_2^R \end{cases} \quad (17)$$

$$\bar{p}^L\left(n_1, n_2\right) = \begin{cases} g_{\mathrm{H}}\left(m_1^U, m_2^L\right) & , \quad \text{if } n_1^U = n_1^D \\ \dfrac{n_1 - n_1^U}{n_1^D - n_1^U} \cdot g_{\mathrm{H}}\left(m_1^D, m_2^L\right) + \dfrac{n_1^D - n_1}{n_1^D - n_1^U} \cdot g_{\mathrm{H}}\left(m_1^U, m_2^L\right) & , \quad \text{if } n_1^U < n_1^D \end{cases} \quad (18)$$

$$\bar{p}^R\left(n_1, n_2\right) = \begin{cases} g_{\mathrm{H}}\left(m_1^U, m_2^R\right) & , \quad \text{if } n_1^U = n_1^D \\ \dfrac{n_1 - n_1^U}{n_1^D - n_1^U} \cdot g_H\left(m_1^D, m_2^R\right) + \dfrac{n_1^D - n_1}{n_1^D - n_1^U} \cdot g_{\mathrm{H}}\left(m_1^U, m_2^R\right) & , \quad \text{if } n_1^U < n_1^D \end{cases} \quad (19)$$

IET Image Process., 2015, Vol. 9, Iss. 1, pp. 54–61
doi: 10.1049/iet-ipr.2014.0321

$(1 \leq k \leq K-1)$, calculate

$$\varphi_k = \sum_{i=V_{\min}}^{V_{\max}} h_i \cdot \left[-\sum_{i=\alpha_k}^{\beta_{k+1}} \gamma_i \cdot \log_2\left(\sum_{i=\alpha_k}^{\beta_{k+1}} \gamma_i\right) + \sum_{i=\alpha_k}^{\beta_k} \gamma_i \cdot \log_2\left(\sum_{i=\alpha_k}^{\beta_k} \gamma_i\right) + \sum_{i=\alpha_{k+1}}^{\beta_{k+1}} \gamma_i \cdot \log_2\left(\sum_{i=\alpha_{k+1}}^{\beta_{k+1}} \gamma_i\right)\right] \tag{27}$$

$$\psi_k = \sum_{i=\alpha_k}^{\beta_{k+1}} h_i \cdot \left(i - \frac{\alpha_k + \beta_{k+1}}{2}\right)^2 - \sum_{i=\alpha_k}^{\beta_k} h_i \cdot \left(i - \frac{\alpha_k + \beta_k}{2}\right)^2 - \sum_{i=\alpha_{k+1}}^{\beta_{k+1}} h_i \cdot \left(i - \frac{\alpha_{k+1} + \beta_{k+1}}{2}\right)^2 \tag{28}$$

Here, ϕ_k and ψ_k are, respectively, the increments of data amount A and total distortion D if we combine the two adjacent sections $[\alpha_k, \beta_k]$ and $[\alpha_{k+1}, \beta_{k+1}]$ as a new section. That implies $\phi_k < 0$ and $\psi_k > 0$. After finding

$$k^* = \arg\max_k \frac{|\varphi_k|}{\psi_k} \tag{29}$$

combine the two sections $[a_k*, b_k*]$ and $[a_{k*+1}, b_{k*+1}]$ as a new section and decrease the number of sections K by 1.

Step 3: Update the values of A and D as in (22) and (24). Owing to the section combination in Step 2, the value of A decreases while the value of D increases, and a new pair of A and D has been produced.
Step 4: If $K > 1$, go to Step 2; otherwise, terminate the iterative procedure.

With the iterative procedure, a number pairs of A and D can be obtained in Step 3, and a rough division would result in a low A and a high D, while a fine division would result in a high A and a low D. In other words, a higher A corresponds to a lower D. After selecting a pair of (A, D) and the corresponding section division, the estimation errors are represented by the indices of their corresponding sections, and converted into a bit sequence using a lossless source coding method, such as arithmetic coding and Huffman coding.

Collect the values of r, N_1, N_2, α_k ($k = 1, 2, \ldots, K$), β_K, and the compressed data of estimation errors to form a data set $\boldsymbol{S}$, and denote the bit amount of $\boldsymbol{S}$ as L_S. If L_S is not larger than $M_1 \cdot M_2 \cdot T$, we keep the data in the $(8-T)$ highest bit-planes of $\boldsymbol{G}$ unchanged and replace the data in the T lowest bit-planes of $\boldsymbol{G}$ with the data of $\boldsymbol{S}$ to generate a new image $\boldsymbol{G}_C$. If L_S is larger than $M_1 \cdot M_2 \cdot T$, we divide $\boldsymbol{S}$ into two parts: $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, where the length of $\boldsymbol{S}_1$ is $M_1 \cdot M_2 \cdot T$ and the length of $\boldsymbol{S}_2$ is $(L_S - M_1 \cdot M_2 \cdot T)$. Then replace all data in the T lowest bit-planes of $\boldsymbol{G}$ with $\boldsymbol{S}_1$, and embed $\boldsymbol{S}_2$ into the $(8-T)$ highest bit-planes of $\boldsymbol{G}$, which are viewed as an $(8-T)$-bit image, by using a RDH technique. Here, the RDH technique ensures that the original $(8-T)$ highest bits of $\boldsymbol{G}$ can be retrieved without any error when extracting the embedded $\boldsymbol{S}_2$ in decompression phase, and a number of RDH methods with good payload-distortion performance are suitable for this purpose. By combining the new data in the T lowest bit-planes coming from $\boldsymbol{S}_1$ and the new data in $(8-T)$ highest bit-planes containing $\boldsymbol{S}_2$, a new image $\boldsymbol{G}_C$ with a size of $M_1 \times M_2$ is generated. The image $\boldsymbol{G}_C$ is just a final result of the semantic compression procedure. Then, the compressed image $\boldsymbol{G}_C$ is distributed broadly to the potential users or transmitted to some appointed receivers.

Since the principal content of original image is represented by the data in highest hit-planes and the affection of RDH technique is not serious as long as the length of $\boldsymbol{S}_2$ is not too large, the compressed image looks similar to the original one. Actually, the quality of $\boldsymbol{G}_C$ is dependent on the scaling ratio r, the value of T and the amount of embedded data L_S. A larger r and a smaller T imply the more original data reserved in the compressed image, while a smaller L_S implies the less data embedded into compressed image, leading to the better quality of the compressed image.

3 Decompression procedure

When a receiver or user with the compressed image $\boldsymbol{G}_C$ is interested in obtaining an image with better quality and higher resolution, he may employ the following procedure to generate a decompressed image.

Firstly, the receiver extracts the embedded data $\boldsymbol{S}$ from $\boldsymbol{G}_C$ and retrieve the original data in the $(8-T)$ highest bit-planes of $\boldsymbol{G}$. If $\boldsymbol{S}$ occupies only the T lowest bit-planes of $\boldsymbol{G}_C$, it is easy to obtain $\boldsymbol{S}$ from the lowest bit-planes, and the original $(8-T)$ highest bits of $\boldsymbol{G}$ are same as those in $\boldsymbol{G}_C$ since they are not changed in the compression procedure. On the other hand, if two parts of $\boldsymbol{S}$ are, respectively, carried by the T lowest and $(8-T)$ highest bit-planes of $\boldsymbol{G}_C$, the receiver may reorganise the data in the T lowest bit-planes as $\boldsymbol{S}_1$, and recover the original data in the $(8-T)$ highest bit-planes of $\boldsymbol{G}$ when extracting $\boldsymbol{S}_2$ from the bit-planes. In this case, $\boldsymbol{S}$ is obtained by combining $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. Then, the values of r, N_1, N_2, α_k ($k = 1, 2, \ldots, K$), β_K, and the compressed data of estimation errors are also obtained. That means the scaling rate, the size of original image, the way of section division and the section indices corresponding to the estimation errors of default pixels have been known by the receiver.

Then, a decompressed image $\boldsymbol{P}_D$ sized $N_1 \times N_2$ can be produced using the following method. Let the values of seed pixels in $\boldsymbol{P}_D$ to be equal to the corresponding pixel values in $\boldsymbol{G}_C$,

$$p_D(\text{round}(m_1/r), \text{round}(m_2/r)) = g_C(m_1, m_2), \quad 1 \leq m_1 \leq M_1, \ 1 \leq m_2 \leq M_2 \tag{30}$$

For the default pixels $p_D(n_1, n_2)$, assuming its estimation error falls into a section $[\alpha_k, \beta_k]$, we define

$$\bar{e}(n_1, n_2) = (\alpha_k + \beta_k)/2 \tag{31}$$

And the receiver can also calculate its estimated value $\bar{p}(n_1, n_2)$ using (17) since the original $(8-T)$ highest bits of $\boldsymbol{G}$ have been retrieved. So, the values of decompressed default pixels are

$$p_D(n_1, n_2) = \bar{e}(n_1, n_2) + \text{round}\left[\bar{p}(n_1, n_2)\right] \tag{32}$$

This way, a decompressed image is produced.

Fig. 2 *Original images*
a Lena
b Man

The distortion in the decompressed image is made up of two parts: the distortion in the seed pixels and the distortion in the default pixels. The first part is dependent on the value of T while the second part is dependent on the way of section division. Consider $T = 0$ and $[V_{\max}, V_{\min}]$ is divided into $(V_{\max} - V_{\min} + 1)$ sections, each of which contains only one value. In this case, the estimation errors of default pixels are compressed in a lossless manner and then embedded into the compact image $\boldsymbol{G}$ in a reversible manner to produce the image $\boldsymbol{G}_{\mathrm{C}}$. So, the original image $\boldsymbol{P}$ can be perfectly recovered from $\boldsymbol{G}_{\mathrm{C}}$ since no distortion is introduced in the compression–decompression procedures. In short, a lossless semantic compression is performed. On the other hand, if T is a positive integer, the data embedding results in a distortion in seed pixels of decompression image, and the section combination results in a distortion in default pixels. That implies a lossy semantic compression/decompression is performed.

4 Experimental results

Two 512 × 512 images, Lena and Man, shown in Fig. 2, were used as the original images. With a scaling ratio $r = 0.88$, we produced two compact version sized 450 × 450 as in Fig. 3. In this case, the image size was reduced to 77% of the original size. After assigning $T = 0$ and calculating estimation errors of default pixels, we let each section contain only one value to perform lossless semantic compression. That means the estimation errors of default pixels were compressed in a lossless manner and embedded into the compact images in a reversible manner. Here, the RDH method in [26] was employed. The final compressed images are given in Fig. 4, and the peak signal-to-noise ratio (PSNR) values are 26.4 dB and 23.1 dB when using the compact images in Fig. 3 as references. When implementing the decompression procedure, the original images can be perfectly recovered from the decompressed images.

We also performed lossy semantic compression using the proposed scheme. With $r = 0.50$, the compact Lena and Man sized 256 × 256 were obtained. We let $T = 3$ and compressed the estimation errors of default pixels in Lena and Man as 2.9×10^5 and 2.2×10^5 bits, respectively. Then, a part of data to be embedded was accommodated in the 3 LSB, and the rest data were embedded into the 5 MSB using the RDH method in [26] to generate the compressed images. The compact and compressed images are shown in

Fig. 3 *Compact images sized 450 × 450*
a Lena
b Man

doi: 10.1049/iet-ipr.2014.0321

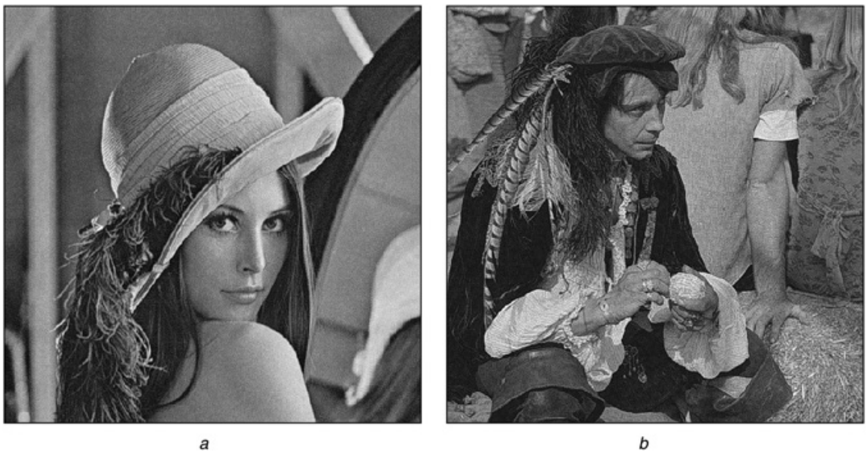

Fig. 4 *Lossless compressed images*
a Lena with PSNR 26.4 dB
b Man with PSNR 23.1 dB

Fig. 5 *Compact images sized 256 × 256*
a Lena
b Man

Fig. 6 *Lossy compressed images*
a Lena with PSNR 18.6 dB
b Man with PSNR 30.2 dB

Figs. 5 and 6, and the values of PSNR in the compressed Lena and Man are 18.6 and 30.2 dB when regarding the corresponding compact images as references. Fig. 7 gives two images decompressed from Fig. 6. The values of PSNR in the decompressed results are 38.8 dB and 34.1 dB when regarding the original Lena and Man as references.

Tables 1–4 list the compression/decompression performance with respect to different r, T and L_S when using four 512 × 512 images Lena, Man, Lake and Peppers as the original, while the last two images are given in Fig. 8. In these tables, PSNR_C is the PSNR value in compressed image when regarding the compact image as

Fig. 7 *Decompressed results*
a Lena with PSNR 38.8 dB
b Man with PSNR 34.1 dB

Table 1 Compression/decompression performance when using Lena as the original

r ($M_1 \times M_2$)	T	L_S	PSNR$_C$, dB	PSNR$_D$, dB
0.92 (470 × 470)	0	1.84×10^5	33.6	+∞
0.88 (450 × 450)	0	2.66×10^5	26.4	+∞
0.88 (450 × 450)	1	2.52×10^5	42.3	52.0
0.88 (450 × 450)	2	2.67×10^5	47.6	48.7
0.68 (350 × 350)	1	1.84×10^5	37.5	42.7
0.68 (350 × 350)	2	3.01×10^5	36.3	43.8
0.68 (350 × 350)	3	2.70×10^5	42.0	42.3
0.50 (256 × 256)	1	1.62×10^5	21.8	37.6
0.50 (256 × 256)	2	1.74×10^5	31.3	36.2
0.50 (256 × 256)	3	2.51×10^5	25.2	37.8
0.50 (256 × 256)	3	2.91×10^5	18.4	38.8

Table 2 Compression/decompression performance when using Man as the original

r ($M_1 \times M_2$)	T	L_S	PSNR$_C$, dB	PSNR$_D$, dB
0.92 (470 × 470)	0	2.02×10^5	31.9	+∞
0.88 (450 × 450)	0	2.93×10^5	23.1	+∞
0.88 (450 × 450)	1	2.93×10^5	37.5	52.3
0.88 (450 × 450)	2	2.94×10^5	46.6	47.7
0.68 (350 × 350)	1	3.03×10^5	21.0	44.2
0.68 (350 × 350)	1	1.76×10^5	37.3	39.0
0.68 (350 × 350)	2	2.61×10^5	36.1	41.9
0.68 (350 × 350)	2	3.65×10^5	26.4	43.8
0.68 (350 × 350)	3	3.55×10^5	38.2	40.2
0.68 (350 × 350)	3	4.82×10^5	22.2	40.7
0.50 (256 × 256)	1	1.51×10^5	20.4	32.9
0.50 (256 × 256)	2	2.20×10^5	19.6	32.9
0.50 (256 × 256)	3	2.21×10^5	30.2	34.1

Table 3 Compression/decompression performance when using Lake as the original

r ($M_1 \times M_2$)	T	L_S	PSNR$_C$, dB	PSNR$_D$, dB
0.92 (470 × 470)	0	2.20×10^5	25.5	+∞
0.88 (450 × 450)	1	3.18×10^5	32.0	52.3
0.88 (450 × 450)	1	2.78×10^5	37.0	51.4
0.88 (450 × 450)	2	3.18×10^5	46.0	47.1
0.68 (350 × 350)	1	2.35×10^5	26.2	38.7
0.68 (350 × 350)	2	3.01×10^5	31.4	40.2
0.68 (350 × 350)	2	3.62×10^5	24.4	42.7
0.68 (350 × 350)	3	3.82×10^5	30.2	39.8
0.68 (350 × 350)	3	4.73×10^5	22.2	40.6
0.50 (256 × 256)	1	1.23×10^5	26.1	29.6
0.50 (256 × 256)	2	1.66×10^5	31.0	32.6
0.50 (256 × 256)	3	2.66×10^5	20.5	32.7

Table 4 Compression/decompression performance when using Peppers as the original

r ($M_1 \times M_2$)	T	L_S	PSNR$_C$, dB	PSNR$_D$, dB
0.92 (470 × 470)	0	1.94×10^5	30.7	+∞
0.88 (450 × 450)	1	2.81×10^5	37.3	52.3
0.88 (450 × 450)	2	2.82×10^5	47.1	48.2
0.68 (350 × 350)	1	1.83×10^5	37.3	40.9
0.68 (350 × 350)	1	2.88×10^5	21.4	44.5
0.68 (350 × 350)	2	2.99×10^5	36.2	43.3
0.68 (350 × 350)	3	3.55×10^5	38.3	40.6
0.68 (350 × 350)	3	4.48×10^5	25.3	40.8
0.50 (256 × 256)	1	1.19×10^5	30.3	34.7
0.50 (256 × 256)	2	1.80×10^5	31.2	36.4
0.50 (256 × 256)	3	2.56×10^5	25.3	36.7
0.50 (256 × 256)	3	2.69×10^5	22.5	36.9

reference, while PSNR$_D$ is the PSNR value in decompressed image when regarding the original image as reference. When the scaling ratio r was close to 1, we could let $T = 0$ to perform lossless semantic compression. So, the original image content can be perfectly reconstructed, that is, PSNR$_D = \infty$. We also let $r < 0.9$ and $T > 0$ to perform lossy semantic compression, and, in these cases, both PSNR$_C$ and PSNR$_D$ are dependent on r, T and L_S. A lower r means a smaller size of compressed image and more default pixels, leading to lower PSNR$_C$ and PSNR$_D$. A smaller T means lower embedding capacity in compact version and more precise estimation on default pixels, so that a lower PSNR$_C$ and a higher PSNR$_D$ are caused. A larger L_S, which implies more data embedded for decompression, also results in a lower PSNR$_C$ and a higher PSNR$_D$. In other words, even though T is high, a very-large L_S can also result in a low PSNR$_C$ and a high

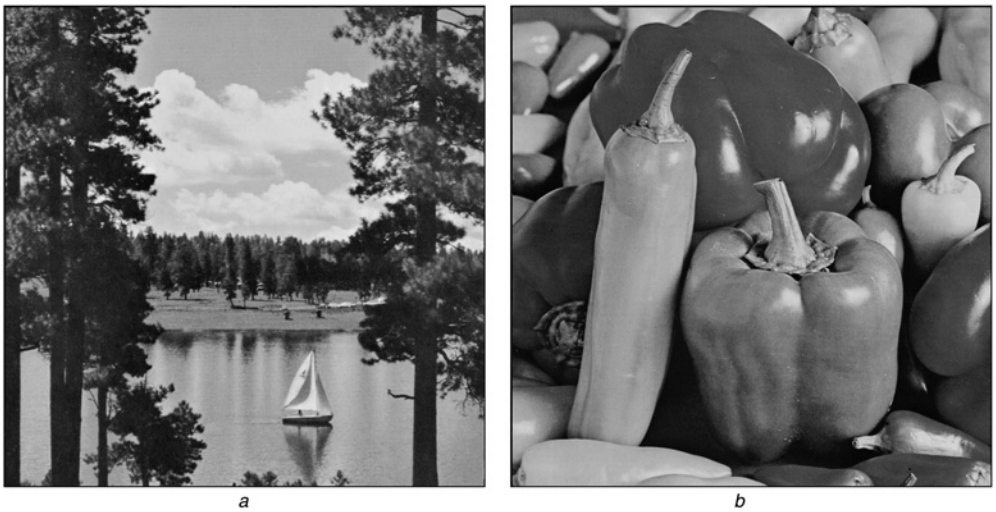

Fig. 8 *Original images*

a Lake
b Peppers

IET Image Process., 2015, Vol. 9, Iss. 1, pp. 54–61
doi: 10.1049/iet-ipr.2014.0321

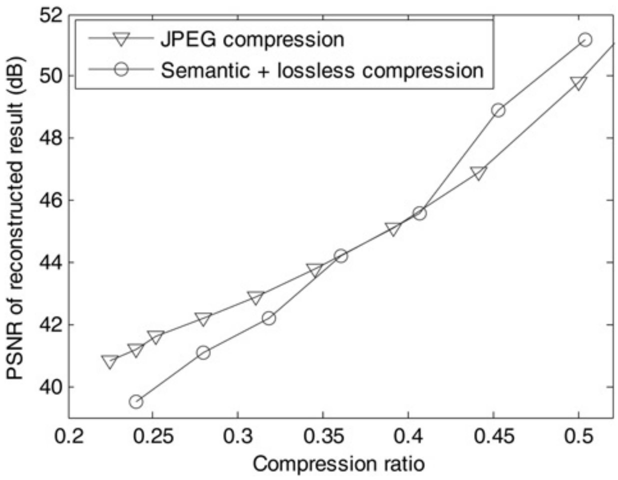

Fig. 9 *Comparison between the proposed scheme and JPEG compression when using Lena as original image*

$PSNR_D$. That means $PSNR_C/PSNR_D$ is not always obtaining worse/better with a decreasing *T*. In fact, $PSNR_C$ and $PSNR_D$ are also dependent on the original image content. Generally speaking, the smoother the original image content, the better are the qualities of compressed and decompressed images.

Fig. 9 compares the performance of the proposed scheme and lossy JPEG compression when using Lena as original image. With the proposed semantic compression scheme, since there is still redundancy in the compressed images, we employed the lossless JPEG compression to further reduce the data amount. A user may reconstruct high quality images using lossless and semantic decompression tools. It can be seen that the two curves intersect and the experiment results on other images were similar. That means the rate-distortion performance of the proposed semantic compression scheme is close to that of JPEG compression.

5 Conclusion

This work proposes a semantic image compression scheme based on data hiding techniques. A compact version is firstly generated by gathering a part of pixels in original image, and after embedding compressed data of estimation errors of the rest pixels into the compact version, a compressed image with smaller size and similar content is produced. With the aid of extracted data, a decompressed image with original size and high quality can be reconstructed. By employing reversible/non-RDH techniques, both the lossy and lossless semantic compression can be performed with the proposed scheme. The qualities of compressed and decompressed images are dependent on the parameter values and the original image content. In the future, the smarter semantic compression methods with better performance and the theoretical performance bounds deserve further investigation.

6 Acknowledgments

This work was supported by the Research Fund for the Doctoral Program of Higher Education of China under Grant 20113108110010, the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, and the Shanghai Pujiang Program under Grant 13PJ1403200.

7 References

1 Willems, F., Maas, D., Kalker, T.: ‘Semantic lossless source coding’. 42nd Annual Allerton Conf. on Communication, Control and Computing, Monticello, Illinois, USA, 2004, pp. 1411–1418
2 Willems, F., Kalker, T.: ‘Semantic compaction, transmission, and compression codes’. Proc. Int. Symp. on Information Theory, Adelaide, Australia, 2005, pp. 214–218
3 Willems, F., Dijk, M.: ‘Capacity and codes for embedding information in gray-scale signals’, *IEEE Trans. Inf. Theory*, 2005, **51**, (3), pp. 1209–1214
4 Fridrich, J.: ‘Steganography in digital media–principles, algorithms, and applications’ (Cambridge University Press, 2010)
5 Fridrich, J., Soukal, D.: ‘Matrix embedding for large payloads’, *IEEE Trans. Inf. Forens. Secur.*, 2006, **1**, (3), pp. 390–394
6 Fridrich, J., Filler, T.: ‘Practical methods for minimizing embedding impact in steganography’, *Proc. SPIE*, 2007, **6050**, pp. 650502.1–15
7 Fridrich, J., Goljan, M., Lisonek, P., Soukal, D.: ‘Writing on wet paper’, *IEEE Trans. Signal Process.*, 2005, **53**, (10), pp. 3923–3935
8 Zhang, W., Zhang, X., Wang, S.: ‘Near-optimal codes for information embedding in gray-scale signals’, *IEEE Trans. Inf. Theory*, 2010, **56**, (3), pp. 1262–1270
9 Fridrich, J.: ‘Asymptotic behavior of the ZZW embedding construction’, *IEEE Trans. Inf. Forens. Secur.*, 2009, **4**, (1), pp. 151–154
10 Goljan, M., Fridrich, J., Du, R.: ‘Distortion-free data embedding’. Proc. of 4th Int. Workshop on Information Hiding, Lecture Notes in Computer Science, Springer-Verlag, 2001, vol. 2137, pp. 27–41
11 Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: ‘Lossless generalized-LSB data embedding’, *IEEE Trans. Image Process.*, 2005, **14**, (2), pp. 253–266
12 Tian, J.: ‘Reversible data embedding using a difference expansion’, *IEEE Trans. Circuits Syst. Video Technol.*, 2003, **13**, (8), pp. 890–896
13 Alattar, A.M.: ‘Reversible watermark using the difference expansion of a generalized integer transform’, *IEEE Trans. Image Process.*, 2004, **13**, (8), pp. 1147–1156
14 Wang, X., Li, X., Yang, B., Guo, Z.: ‘Efficient generalized integer transform for reversible watermarking’, *IEEE Signal Process. Lett.*, 2010, **17**, (6), pp. 567–570
15 Wu, H.-C., Lee, C.-C., Tsai, C.-S., Chu, Y.-P., Chen, H.-R.: ‘A high capacity reversible data hiding scheme with edge prediction and difference expansion’, *J. Syst. Softw.*, 2009, **82**, pp. 1966–1973
16 Thodi, D.M., Rodríguez, J.J.: ‘Expansion embedding techniques for reversible watermarking’, *IEEE Trans. Image Process.*, 2007, **16**, (3), pp. 721–730
17 Kamstra, L., Heijmans, H.J.A.M.: ‘Reversible data embedding into images using wavelet techniques and sorting’, *IEEE Trans. Image Process.*, 2005, **14**, (12), pp. 2082–2090
18 Kim, H.J., Sachnev, V., Shi, Y.Q., Nam, J., Choo, H.-G.: ‘A novel difference expansion transform for reversible data embedding’, *IEEE Trans. Inf. Forens. Secur.*, 2008, **3**, (3), pp. 456–465
19 Weng, S., Zhao, Y., Pan, J.-S., Ni, R.: ‘Reversible watermarking based on invariability and adjustment on pixel pairs’, *IEEE Signal Process. Lett.*, 2008, **15**, pp. 721–724
20 Hu, Y., Lee, H.-K., Li, J.: ‘DE-based reversible data hiding with improved overflow location map’, *IEEE Trans. Circuits Syst. Video Technol.*, 2009, **19**, (2), pp. 250–260
21 Ni, Z., Shi, Y.-Q., Ansari, N., Su, W.: ‘Reversible data hiding’, *IEEE Trans. Circuits Syst. Video Technol.*, 2006, **16**, (3), pp. 354–362
22 Kim, K.-S., Lee, M.-J., Lee, H.-Y., Lee, H.-K.: ‘Reversible data hiding exploiting spatial correlation between sub-sampled images’, *Pattern Recognit.*, 2009, **42**, pp. 3083–3096
23 Chang, C.-C., Lin, C.-C., Chen, Y.-H.: ‘Reversible data-embedding scheme using differences between original and predicted pixel values’, *IET Inf. Secur.*, 2008, **2**, (2), pp. 35–46
24 Tsai, P., Hu, Y.-C., Yeh, H.-L.: ‘Reversible image hiding scheme using predictive coding and histogram shifting’, *Signal Process.*, 2009, **89**, pp. 1129–1143
25 Hong, W., Chen, T.-S., Chang, Y.-P., Shiu, C.-W.: ‘A high capacity reversible data hiding scheme using orthogonal projection and prediction error modification’, *Signal Process.*, 2010, **90**, pp. 2911–2922
26 Luo, L., Chen, Z., Chen, M., Zeng, X., Xiong, Z.: ‘Reversible image watermarking using interpolation technique’, *IEEE Trans. Inf. Forens. Secur.*, 2010, **5**, (1), pp. 187–193
27 Zhang, X.: ‘Reversible data hiding with optimal value transfer’, *IEEE Trans. Multimed.*, 2013, **15**, (2), pp. 316–325