A Near-Lossless Image Compression System with Data Hiding Capability

Masaaki FUJIYOSHI and Hitoshi KIYA
Tokyo Metropolitan University, Hino, Tokyo 191–0065, Japan
E-mail: mfujiyoshi@ieee.org, kiya@tmu.ac.jp Tel: +81–42–585–8454

***Abstract*—This paper proposes a near-lossless compression method for images where the method can conceals data to an image before compression, i.e., a data hiding-then-compression system. The proposed method first preprocesses an image near-losslessly, i.e., the difference between a pixel of the original image and that of the preprocessed image is not more than the user given maximum allowed error. This method, then, embeds data to the preprocessed image and reversibly post-processes the image conveying data to improve the compression efficiency. Finally, the method losslessly compresses the post-processed image. The framework of the proposed system accepts any arbitrary lossless compression technique, whereas ordinary joint compression-and-data hiding and compression-then-data hiding systems do not have such flexibility. In addition, the near-lossless condition is satisfied between the preprocessed image and that with hidden data and between the original image and the preprocessed image with hidden data. For low bitrate compression, the proposed method serves better decompressed image than lossy compression of the original image in terms of the image quality. Experimental results show the effectiveness of the proposed method.**

I. Introduction

Efficient transmission of images over bandwidth-constrained communication channels and/or effective storage of images on finite-sized storage systems mandatorily require compression of images [1]–[4]. In addition, it is desired to multiplex supplementary data, or to hide data, to images with the wide variety of multimedia applications [5]–[7] such as copyright protection [8], error control [9], broadcast monitoring [10], and so on. To this end, compression systems with data hiding capability have been studied [9]–[12].

A data hiding scheme distorts an original image to hide data in the image. Distorted images generally deteriorates the compression efficiency and compression noise distorts hidden data in images. To overcome these problems, joint compression-and-data hiding [11] and compression-then-data hiding [9] systems are widely employed. The former, however, requires a proprietary compression encoder and the latter needs a data hiding scheme specially customized for the used compression technique, viz., systems are with no flexibility.

This paper proposes a near-lossless compression system enabling data hiding for images. The proposed method consists of near-lossless preprocessing including data hiding and lossless compression. This separated realization achieves a data hiding-then-compression system which accepts any arbitrary lossless compression technique. By its design, the proposed method meets the near-lossless condition for data hiding as well as for image compression. The method achieves the better compression efficiency than directly lossy compression of an original image under low bitrate compression, or high compression ratio.

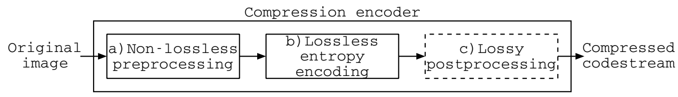


Fig. 1. An ordinary non-lossless compression system for images. The system is exactly the same as the compression encoder.

II. Preliminaries

This section mentions an ordinary lossy or near lossless image compression system and another realization. In addition, several systems consisting of data hiding and image compression are briefly described.

A. Non-Lossless Compression Systems

Figure 1 shows an ordinary system for non-lossless compression, i.e., lossy or near-lossless compression. This system is exactly the same as an image compression encoder, c.f., the whole system in Fig. 1 is in the box indicated as compression encoder. The system, or the encoder, is mainly compounded of a) a non-lossless preprocessing block and b) a lossless entropy encoding block, and some encoder optionally have c) a lossy post-processing block, where symbols a), b), and so on represent different block types hereafter. For instance, in JPEG [13] lossy compression, the system is formed by two blocks; a) discrete cosine transformation (DCT) of an original image and quantization of DCT coefficients and b) Huffman encoding of quantized DCT coefficients, while JPEG 2000 [14] lossy compression has three blocks; a) discrete wavelet transformation (DWT) of an original image and quantization of DWT coefficients, b) arithmetic encoding of quantized DWT coefficients, and c) bit truncation of the compressed codestream. In JPEG-LS [15] near-lossless compression, the system consists of two blocks; a) prediction error derivation of an original image and quantization of prediction error and b) Golomb encoding of quantized prediction error.

Another realization of non-lossless compression system [16] is shown in Fig. 2 where this system is *separable*. That is, a compression encoder is just a part of the system and any compression standard can be used. This realization has the following blocks; it has a) a non-lossless preprocessing block

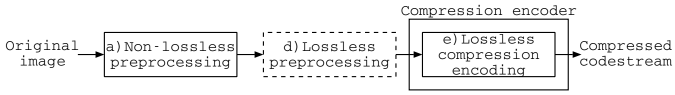


Fig. 2. Another realization of non-lossless compression system for images. The compression encoder is a part of the system.

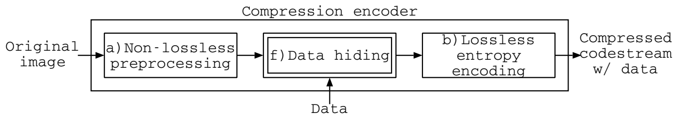


Fig. 3. Joint compression-and-data hiding system for images. A proprietary compression encoder containing a data hiding process.

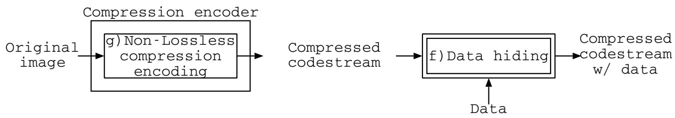


Fig. 4. Compression-then-data hiding system for images. The data hiding technique is specially designed for the used compression technique.

as well as ordinary systems, but it uses e) a lossless compression encoding block, i.e., a lossless compression encoder, and it optionally uses d) a lossless post-processing block before e). Conventional works which do not embed data to images are compounded of a) zero skip quantization [16], [17] of an original image, d) histogram packing [18]–[20], and e) JPEG 2000 lossless compression or JPEG-LS lossless compression.

B. Image Compression and Data Hiding

Simultaneous applying compression and data hiding to images have been studied for several purposes, and two approaches exist; One is joint image compression-and-data hiding and the other is image compression-then-data hiding.

Figure 3 shows a joint image compression-and-data hiding system, where data are inserted to images in the compression encoder. The system consists of three blocks; a) a non-lossless preprocessing block, f) a data hiding block, and b) a lossless entropy encoding block, so the proprietary compression encoder is required. To protect hidden data from compression noise, f) follows a). For example, a JPEG-based system [11] is compound of the following blocks; a) DCT of an original image and DCT coefficient quantization, f) hiding data to quantized DCT coefficients, and b) Huffman encoding of DCT coefficients carrying hidden data.

An image compression-then-data hiding system is shown in Fig. 4 where lossy image compression and data hiding are separable. In this system, data are concealed in a lossily compressed codestream for protecting hidden data from compression noise where the system consists of g) a lossy compression encoding block and f) a data hiding block. The data hiding scheme is required to parse and reconstruct codestreams, i.e., the scheme is specially designed for the used compression technique. The JPEG 2000-based system [9] is compound of blocks g) a standard JPEG 2000 lossy compression encoder and f) a data hiding scheme in codestream domain.

III. Proposed Method

This section proposes a near-lossless compression method for images where a compressed image conceals data in it. First, strategies of the proposed method are summarized. Then, the algorithms in the proposed method are described.

A. Strategies

1) Near-Lossless Compression: The proposed method decompresses a compressed codestream which conveys hidden data in it and the method further removes concealed data from the decompressed image. In this method, the difference between a pixel in the decompressed image and that in the original image is not more than user given maximum allowed error δ;

$$\left|\tilde{\mathbf{f}}-\mathbf{f}\right|_\infty = \max_{x,y}\left|\tilde{f}(x,y)-f(x,y)\right| \le \delta, \tag{1}$$

where $\mathbf{f}=\{f(x,y)\}$ is the $X \times Y$-sized K-bit grayscale original image, $\tilde{\mathbf{f}}=\{\tilde{f}(x,y)\}$ is the $X \times Y$-sized K-bit grayscale decompressed image without hidden data, $x=0,1,\ldots,X-1$, $y=0,1,\ldots,Y-1$, and $f(x,y),\tilde{f}(x,y)\in[0\ ..\ 2^K-1]$. In addition, a pixel in the image with hidden data differs from that in $\mathbf{f}$ by not more than δ;

$$\left|\hat{\mathbf{f}}-\mathbf{f}\right|_\infty \le \delta, \tag{2}$$

where $\hat{\mathbf{f}}=\{\hat{f}(x,y)\}$ is the $X \times Y$-sized K-bit grayscale decompressed image with hidden data and $\hat{f}(x,y)\in[0\ ..\ 2^K-1]$. Furthermore, this method meets

$$\left|\hat{\mathbf{f}}-\tilde{\mathbf{f}}\right|_\infty \le \delta. \tag{3}$$

Consequently, the proposed method meets three different near-lossless conditions.

2) Separable: The proposed method is based on the compression system shown in Fig. 2. Joint image compression-and-data hiding systems and compression-then-data hiding systems strongly depend on the used compression technique as mentioned in Section II-B. A proprietary compression encoder or a data hiding scheme specially designed for the compression technique is required. On the other hand, a separable compression system can employ any arbitrary lossless image compression technique and it can serve a data hiding-then-compression system.

3) Reversible Data Hiding: The proposed method hides data to an image in addition to compression of the image. That is, this method introduces data hiding noise to images as well as compression noise, even data hiding noise is limited as Eq. (3). To produce $\tilde{\mathbf{f}}$, this method should remove hidden data from the image. In irreversible data hiding that is the most well-known data hiding class, data can be removed but the distortion of the image caused by data hiding cannot be compensated [10]. On the other hand, *reversible* data hiding [21]–[23] removes the distortion of the image as well as concealed data. Thus, this method uses reversible data hiding.

According to the above strategies, the proposed method shown in Fig. 5 consists of four blocks, namely, a) a non-lossless preprocessing block, f) a reversible data hiding block,

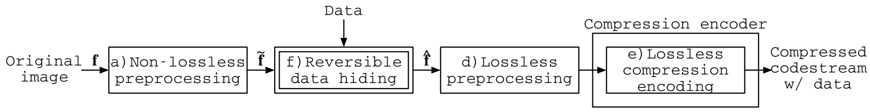
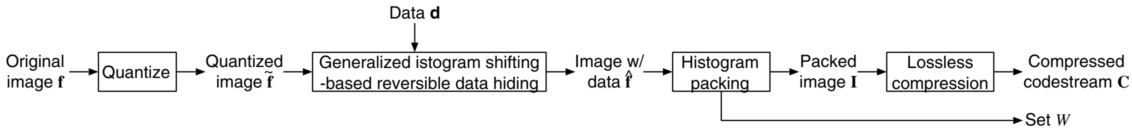
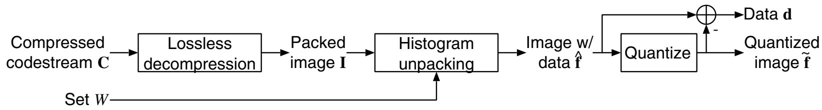


Fig. 5. Proposed method.



(a) Data hiding and compression.



(b) Decoding and data extraction.

Fig. 6. Block diagrams of the proposed method.

d) a lossless post-processing block, and e) a lossless compression encoding block. To simultaneously meet Eqs. (1), (2), and (3), the proposed method uses a floor function-based quantizer as a) and generalized histogram shifting-based reversible data hiding [24]–[27] as f) as described in the next section.

B. Algorithms

The successive sections describe the algorithm for data hiding and compression shown in Fig. 6 (a) and that for decompression and data extraction shown in Fig. 6 (b).

1) Data Hiding and Compression: The following algorithm compresses $X \times Y$-sized K-bit grayscale original image $\mathbf{f}$ and simultaneously hides L-length q-ary data $\mathbf{d} = \{d(l)\}$ to $\mathbf{f}$ where $d(l) \in [0 \; .. \; q-1]$ and $l = 0, 1, \ldots, L-1$.

1. **Near-Lossless Preprocessing**
   Original image $\mathbf{f}$ is first quantized as
   $$\tilde{f}(x,y) = q \left\lfloor \frac{f(x,y)}{q} \right\rfloor, \quad (4)$$
   where $\tilde{\mathbf{f}}$ is the quantized image and becomes the decompressed image in the *Decompression and Data Extraction* algorithm described in the next section, $q = \delta + 1$ is the quantization step, and $\lfloor \cdot \rfloor$ is the floor function which returns the largest integer not larger than the input. This step makes room for data hiding and for improving the compression efficiency.
2. **Reversible Data Hiding**
   Data $\mathbf{d}$ are hidden to quantized image $\tilde{\mathbf{f}}$ based on generalized histogram shifting-based reversible data hiding [24]–[27];
   $$\hat{f}(x,y) = \tilde{f}(x,y) + d\left(L_{a-1} + l_a\right), \quad (x,y) = m_a\left(l_a\right), \quad (5)$$
   where $\hat{\mathbf{f}}$ is the quantized image with hidden data, $m_a(l_a)$ is the l_a-th element of set $M_a = \left\{(x,y) \mid \tilde{f}(x,y) = qa\right\}$, $l_a = 1, 2, \ldots, |M_a|$, $L_{-1} = 0$, $L_{a-1} = \sum_{b=0}^{a-1} |M_b|$, $a = 0, 1, \ldots, A-1$, and $A \leq Z = \left\lfloor \left(2^K - (q-1)\right)/q \right\rfloor + 1$. Note $\sum_{b=0}^{Z-1} |M_b| \leq XY$.
3. **Lossless Preprocessing**
   Histogram packing [18]–[20] which can improve the lossless compression efficiency of images having a sparse histogram is applied to $\hat{\mathbf{f}}$;
   $$I(x,y) = n, \quad \hat{f}(x,y) = w(n), \quad (6)$$
   where $\mathbf{I} = \{I(x,y)\}$ is the N-level histogram packed image, $w(n)$ is the n-th element of set $W = \left\{v \mid \hat{h}(v) \neq 0\right\}$, $\hat{h}(v) = \left|\left\{(x,y) \mid \hat{f}(x,y) = v\right\}\right|$, $v = 0, 1, \ldots, 2^K - 1$, $n = 0, 1, \ldots, N-1$, and $I(x,y) \in [0 \; .. \; N-1]$. This step outputs W for histogram unpacking at decompression.
4. **Lossless Encoding**
   Any arbitrary lossless image compression technique can be applied to $\mathbf{I}$ to produce a compressed codestream with hidden data where the codestream is represented as $\mathbf{C}$.

The above mentioned algorithm outputs compressed codestream $\mathbf{C}$ concealing $\mathbf{d}$ in it and set W.

2) Decompression and Data Extraction: The following algorithm is applied to codestream $\mathbf{C}$ to take $\mathbf{d}$ out and to reconstruct quantized image $\tilde{\mathbf{f}}$ from $\mathbf{C}$.

1. **Lossless Decoding**
   Lossless image decompression technique corresponding to the compression technique in Step 4 of the *Data Hiding and Compression* algorithm is applied to $\mathbf{C}$ to obtain $\mathbf{I}$.
2. **Lossless Postprocessing**
   Histogram unpacking with W is applied to $\mathbf{I}$ to reconstruct $\hat{\mathbf{f}}$ as
   $$\hat{f}(x,y) = w(n), \quad I(x,y) = n. \quad (7)$$
3. **Data Extraction and Quantized Image Recovery**
   Reconstructed image $\hat{\mathbf{f}}$ is re-quantized as
   $$\tilde{f}(x,y) = q \left\lfloor \frac{\hat{f}(x,y)}{q} \right\rfloor, \quad (8)$$

0	1	2	0
1	2	2	5
2	1	3	3
4	5	7	7

(a) Original **f**.

0	0	2	0
0	2	2	4
2	0	2	2
4	4	6	6

(b) Quantized $\tilde{\mathbf{f}}$.

0	1	1	1
0	0	0	0
1	1	0	1
0	0	0	1

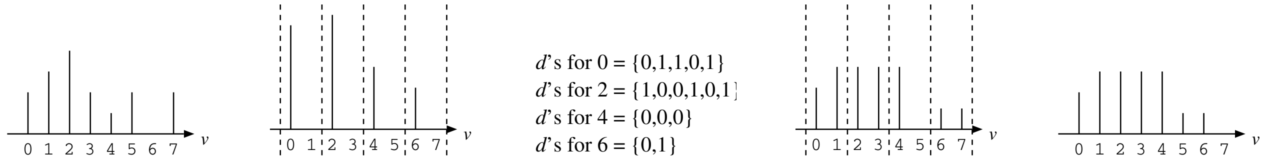
(c) Data **d**.

0	1	3	1
0	2	2	4
3	1	2	3
4	4	6	7

(d) With data $\hat{\mathbf{f}}$.

0	1	3	1
0	2	2	4
3	1	2	3
4	4	5	6

(e) Packed **I**.



(f) Original **h**. (g) Quantized $\tilde{\mathbf{h}}$. (h) Data for qa. (i) With data $\hat{\mathbf{h}}$. (j) Packed $\mathbf{h_I}$.

Fig. 7. Tangible example with 4×4-sized 3-bit image ($X = Y = 4$ and $K = 3$). Data **d** are displayed in two-dimensional form, and **h**, $\tilde{\mathbf{h}}$, $\hat{\mathbf{h}}$, and $\mathbf{h_I}$ are the histogram of **f**, $\tilde{\mathbf{f}}$, $\hat{\mathbf{f}}$, and **I**, respectively.

and hidden data **d** are extracted based on generalized least significant bitplane substitution-based data hiding [28] as

$$d\left(L_{a-1}+l_a\right) = \hat{f}(x,y) - \tilde{f}(x,y), \quad (x,y) = m_a\left(l_a\right), \quad (9)$$

where L_{a-1}, l_a, and $m_a\left(l_a\right)$ are given by using $\tilde{\mathbf{f}}$ as described in Step 2 of the *Data Hiding and Compression* algorithm.

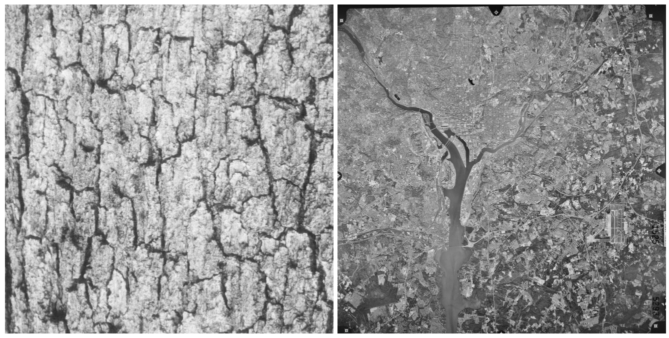
The above mentioned algorithm outputs quantized image $\tilde{\mathbf{f}}$ and L-length q-ary data **d** where Eq. (1) is satisfied.

Figure 7 shows a simple example of the proposed method with 4×4-sized 3-bit grayscale images, where tonal distribution **h** consists of $h(v) = |\{(x,y) \mid f(x,y) = v\}|$ is the histogram of original image **f**.

IV. Experimental Results

All images from a well-known image database [29] consisting of several characteristics (textures, aerials, sequences, and miscellaneous) and several sizes (256×256, 512×512, 1024×1024, and 2250×2250) are used for performance evaluation here. Color images are converted to grayscale images before evaluation by $f(x,y) = 0.2989R(x,y) + 0.5870G(x,y) + 0.1140B(x,y)$ where $R(x,y)$, $G(x,y)$, and $B(x,y)$ are the pixel values of R, G, and B channels, respectively, so all images for evaluation are 8-bit grayscale images, i.e., $K = 8$. Four images from evaluated images are shown in Fig. 8. The number of M_a's for data hiding, A's, are 1, 2, 4, 8, 16, 32, 64, and 128 where M_a's having the A of largest $|M_a|$ are used, and ten different data consisting of uniformly distributed q-ary random symbols are used for each condition.

Figure 9 (a) shows the compression efficiency of image '1.3.02,' where the peak signal-to-noise ratios (PSNR's) of decompressed images are shown. First, the proposed method with a JPEG 2000 standard encoder (JasPer 1.900.1 [30]) compresses image '1.3.02' where $\delta = 1$, viz., $q = 2$ and ten different random equiprobable data are hidden to the image. Then, the compression ratio averaged over ten different codestreams was derived including the bzip2-compressed set for histogram unpacking. The blue curve in the figure indicated 'w/ data' shows the PSNR between $\hat{\mathbf{f}}$ and **f**, while the red curve indicated 'w/o data' shows the PSNR between $\tilde{\mathbf{f}}$ and **f**. The

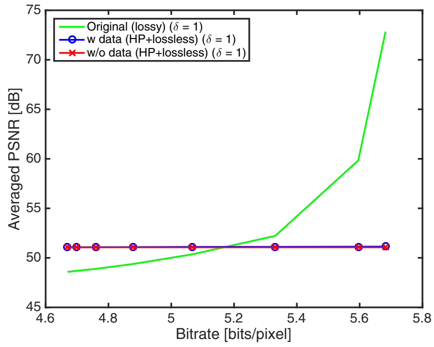


(a) 1.3.02 (texture, 1024×1024). (b) wash-ir (aerial, 2250×2250).

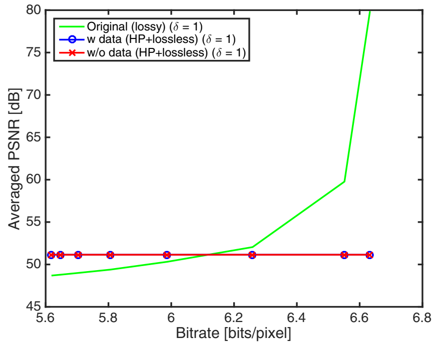
(c) 6.2.01 (sequences, 256×256). (d) 4.2.04 (miscellaneous, 512×512).

Fig. 8. Four examples from 215 of 8-bit grayscale images for evaluation.

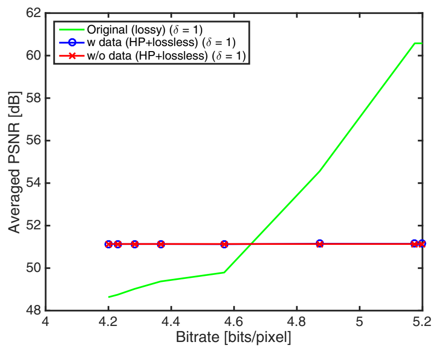
green curve in the figure indicated 'Lossy' shows the PSNR between **f** and the lossily JPEG 2000 compressed original image where the compression rate is the same as the above derived rate. It is found that the proposed method can give images with higher PSNR than those lossily compressed by the standard JPEG 2000 encoder for low bitrate conditions (under 5.2 bpp for image '1.3.02'), whereas distorted images generally deteriorate the compression efficiency. It is noted that the almost similar result, i.e., the proposed method achieves better compression efficiency than lossily compression of the original image, is confirmed for other 200 and more images as shown in Figs. 9 (b), (c), and (d).



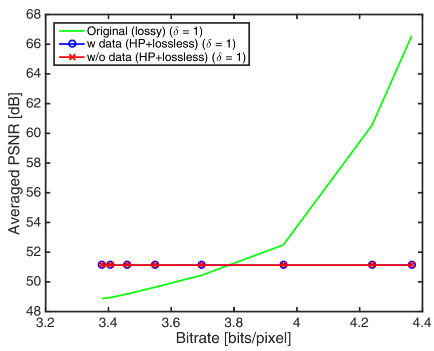
(a) 1.3.02.



(b) wash-ir.

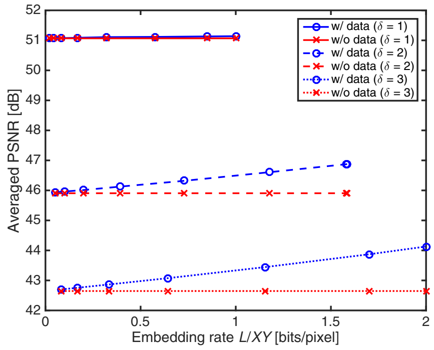


(c) 6.2.01.

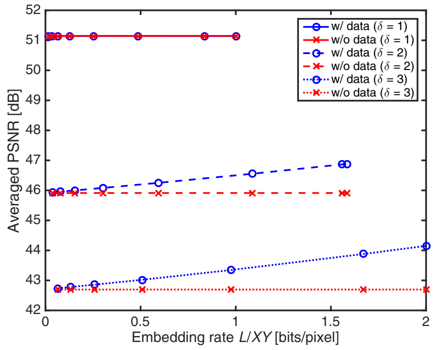


(d) 4.2.04.

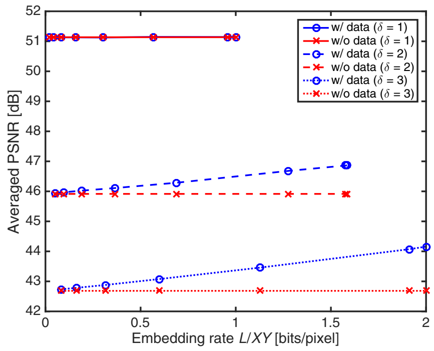
Fig. 9. Compression efficiency improvement by comparing the PSNR of decompressed images ($\delta = 1$ for the proposed method, dividing the codestream length in bits by the number of pixels of the image gives the bitrate in bits/pixel, and dividing $K = 8$ by the bitrate gives the compression ratio). The proposed method achieves better PSNR than lossy compression of the original image for lower bitrates.



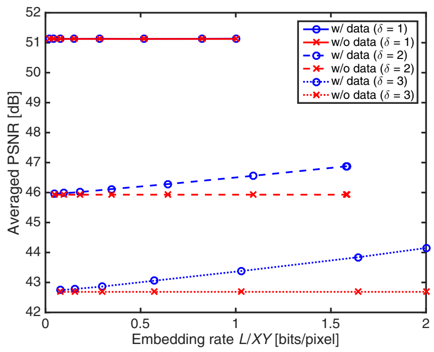
(a) 1.3.02.



(b) wash-ir.

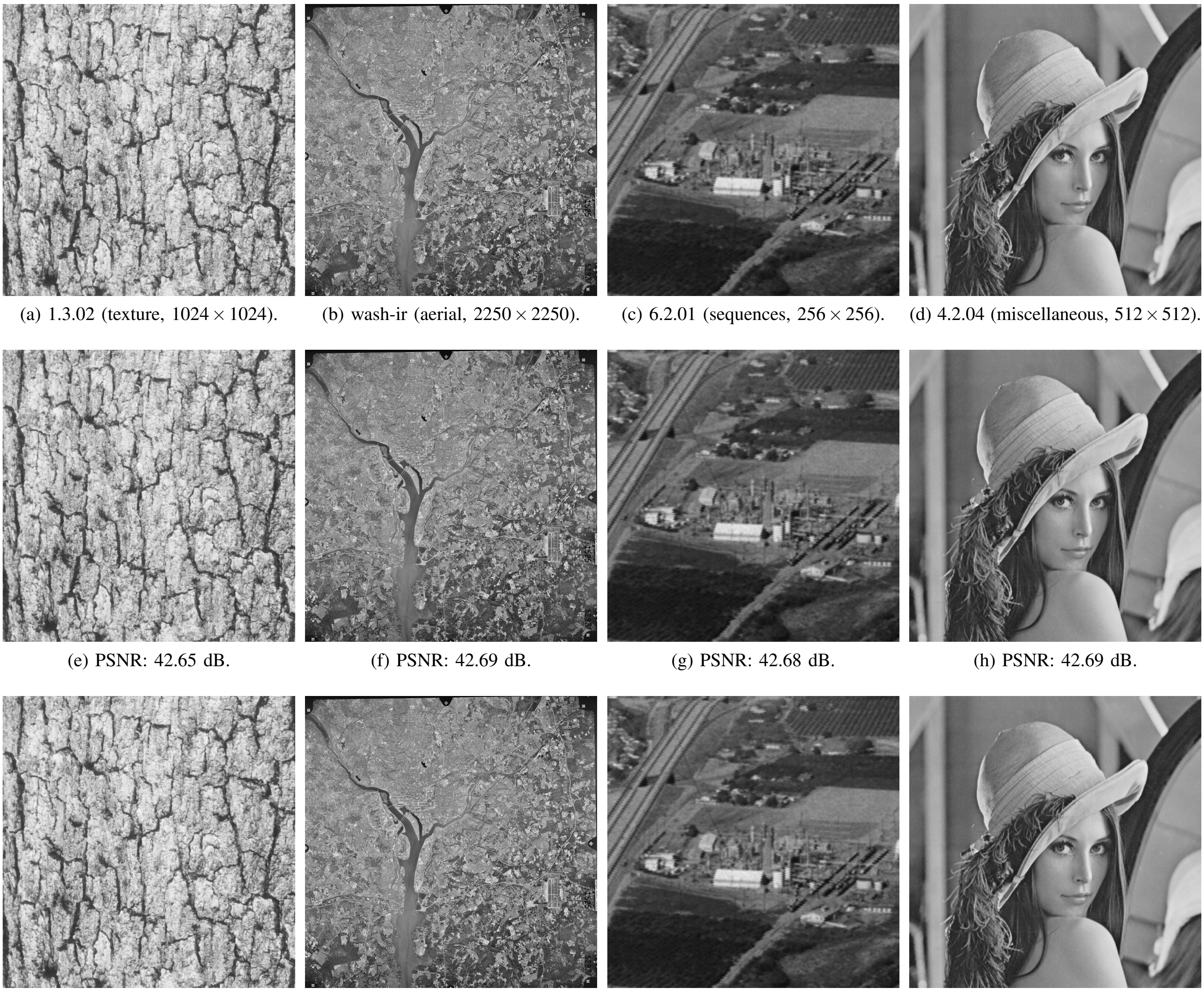


(c) 6.2.01.



(d) 4.2.04.

Fig. 10. PSNR versus embedding rate. The floor function-based quantization in the proposed method once gathers pixel values to qa as shown in Fig. 7 (g), adding q-ary symbols, however, moves several pixel values towards to the original values or brings several pixel values back to the original values as shown in Fig. 7 (i). It results in the PSNR improvement.



(a) 1.3.02 (texture, 1024×1024). (b) wash-ir (aerial, 2250×2250). (c) 6.2.01 (sequences, 256×256). (d) 4.2.04 (miscellaneous, 512×512).

(e) PSNR: 42.65 dB. (f) PSNR: 42.69 dB. (g) PSNR: 42.68 dB. (h) PSNR: 42.69 dB.

(i) PSNR: 44.13 dB. (j) PSNR: 44.15 dB. (k) PSNR: 44.16 dB. (l) PSNR: 44.15 dB.

Fig. 11. Image examples by the proposed method. The top row shows the original images, the middle row displays the decompressed images without hidden data, and the bottom row lists the decompressed images with hidden data ($\delta = 3$ and the embedding rate is 2). As images have the PSNR over 40 dB, it is difficult to tell the decompressed images from the original images.

Figure 10 shows the averaged PSNR between $\tilde{\mathbf{f}}$ and $\mathbf{f}$ and that between $\hat{\mathbf{f}}$ and $\mathbf{f}$ by the proposed method. It was found that the PSNR between $\hat{\mathbf{f}}$ and $\mathbf{f}$ tends to be higher than that between $\tilde{\mathbf{f}}$ and $\mathbf{f}$. Eq. (4) maps pixel values $\{qa, qa+1, \ldots, qa+(q-1)\}$ to qa, as shown in Fig. 7 (g). That is, quantization errors $\{0, 1, \ldots, q-1\}$ are introduced to the image, respectively. Adding q-ary data element $d(l) \in [0 .. q-1]$ to quantized pixel $\tilde{f}(x,y)$ by Eq. (5) brings qa back to $\{qa, qa+1, \ldots, qa+(q-1)\}$ as shown in Fig. 7 (i). In other words, $\hat{\mathbf{f}}$ could be closer to $\mathbf{f}$ than $\tilde{\mathbf{f}}$ as the number of pixels with hidden data (embedding rate L/XY) gets larger. Thus, the PSNR between $\hat{\mathbf{f}}$ and $\mathbf{f}$ tends to be higher than that between $\tilde{\mathbf{f}}$ and $\mathbf{f}$. It is noted that almost similar results are derived for other images. Figure 11 shows example images by the proposed method. Since images have the PSNR over 40 dB, it is difficult to tell the decompressed images from the original images.

V. Conclusions

This paper has proposed a system for near-lossless compression of images where the method can conceal data in images. The proposed method is a separable system where near-lossless preprocessing of images and lossless compression of preprocessed images are separated. This feature can employ any arbitrary lossless compression encoding technique and can serve data hiding-then-compression, whereas conventional methods are joint compression-and-data hiding or compression-then-data hiding systems. Furthermore, by utilizing the framework and choosing preprocessing and reversible data hiding technique carefully, the proposed method meets three different near-lossless conditions and it tends to improve the PSNR of decompressed images by data hiding. In addition, the method achieves the better compression efficiency than lossy compression for higher compression ratios.

Further works include the performance evaluation of the proposed method with zero skip quantization [16], [17].

References

[1] Khairul Munadi, Fitri Arnia, Mohd Syaryadhi, M. Fujiyoshi, and H. Kiya, "A secure online image trading system for untrusted cloud environments," *SpringerPlus*, vol.4, pp.277:1–277:12, Jun. 2015.

[2] S. Liu, W. Sae-Tang, M. Fujiyoshi, and H. Kiya, "An efficient compression of amplitude-only images for the image trading system," *IEICE Trans. Inf. Syst.*, vol.E97-D, pp.378–379, Feb. 2014.

[3] M. Fujiyoshi, Y. Tachizaki, and H. Kiya, "A moving object detection scheme in codestream domain for Motion JPEG encoded movies," in *Advances in Multimedia Information Processing – PCM 2010*, LNCS, G. Qiu, K.M. Lam, H. Kiya, X.-Y. Xue, C.-C.J. Kuo, and M.S. Lew, Eds. Springer Berlin Heidelberg, Sep. 2010, vol.6298, pp.595–606.

[4] Fitri Arnia, I. Iizuka, M. Fujiyoshi, and H. Kiya, "Fast and robust identification methods for JPEG images with various compression ratios," in *Proc. IEEE ICASSP*, 2006.

[5] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, 2nd ed. Morgan Kaufmann Publishers, 2008.

[6] M. Barni, "What is the future for watermarking? (Part I)," *IEEE Signal Process. Mag.*, vol.20, no.5, pp.55–59, Sept. 2003.

[7] ——, "What is the future for watermarking? (Part II)," *IEEE Signal Process. Mag.*, vol.20, no.6, pp.53–59, Nov. 2003.

[8] C.-C.J. Kuo, T. Kalker, and W. Zhou, eds., "Digital rights management," *IEEE Signal Process. Mag.*, vol.21, no.2, pp.11–117, Mar. 2004.

[9] M. Kurosaki, Khairul Munadi, and H. Kiya, "Error correction using data hiding technique for JPEG2000 images," in *Proc. IEEE ICIP*, 2003.

[10] T. Tachibana, M. Fujiyoshi, and H. Kiya, "A removable watermarking scheme retaining the desired image quality," in *Proc. IEEE ISPACS*, 2003, pp.538–542.

[11] Y. Seki, H. Kobayashi, M. Fujiyoshi, and H. Kiya, "Quantization-based image steganography without data hiding position memorization," in *Proc. IEEE ISCAS*, 2005, pp.5-4987–5-4990.

[12] W. Sae-Tang, S. Liu, M. Fujiyoshi, H. Kiya, "A copyright- and privacy-protected image trading system using fingerprinting in discrete wavelet domain with JPEG 2000," *IEICE Trans. Fundamentals*, vol.E97-A, pp.2107–2113, Nov. 2014.

[13] Information technology — Digital compression and coding of continuous-tone still image: Requirements and guidelines. ISO/IEC IS-10918-1, 1994.

[14] Information technology — JPEG 2000 image coding system – Part 1: Core coding system. ISO/IEC IS-15444-1, 2000.

[15] Information technology — Lossless and near-lossless compression of continuous-tone still image — Baseline. ISO/IEC IS-14495-1, 1994.

[16] M. Iwahashi, H. Kobayashi, and H. Kiya, "Lossy compression of sparse histogram images," in *Proc. IEEE ICASSP*, 2012, pp.1361–1364.

[17] M. Iwahashi, T. Yoshida, and H. Kiya, "Range reduction of HDR images for backward compatibility with LDR image processing," in *Proc. APSIPA ASC*, 2014.

[18] A.J. Pinho, "An online preprocessing technique for improving the lossless compression of images with sparse histograms," *IEEE Signal Process. Lett.*, vol.9, pp.5–7, Jan. 2002.

[19] P.J.S.G. Ferreira and A.J. Pinho, "Why does histogram packing improve lossless compression rates?" *IEEE Signal Process. Lett.*, vol.9, pp.259–261, Aug. 2002.

[20] T. Odaka, W. Sae-Tang, M. Fujiyoshi, H. Kobayashi, M. Iwahashi, and H. Kiya, "An efficient lossless compression method using histogram packing for HDR images in OpenEXR format," *IEICE Trans. Fundamentals*, vol.E97-A, pp.2181–2183, Nov. 2014.

[21] R. Caldelli, F. Filippini, and R. Becarelli, "Reversible watermarking techniques: An overview and a classification," *EURASIP J. Info. Security*, vol.2010, 2010.

[22] S. Han, M. Fujiyoshi, and H. Kiya, "An efficient reversible image authentication method," *IEICE Trans. Fundamentals*, vol.E91-A, pp.1907–1914, Aug. 2008.

[23] M. Fujiyoshi, S. Sato, H.L. Jin, H. Kiya, "A location-map free reversible data hiding method using block-based single parameter," in *Proc. IEEE ICIP*, 2007, pp.III-257–III-260.

[24] M. Arabzadeh, M. S. Helfroush, H. Danyali, and K. Kasiri, "Reversible watermarking based on generalized histogram shifting," in *Proc. IEEE ICIP*, 2011, pp.2797–2800.

[25] M. Fujiyoshi and H. Kiya, "Generalized histogram shifting-based reversible data hiding with an adaptive binary-to-q-ary converter," in *Proc. APSIPA ASC*, 2012.

[26] M. Fujiyoshi, "A blind lossless information embedding scheme based on generalized histogram shifting," in *Proc. APSIPA ASC*, 2013.

[27] ——, "Generalized histogram shifting-based blind reversible data hiding with balanced and guarded double side modification," in *Digital-Forensics and Watermarking*, LNCS, Y.Q. Shi, H.-J. Kim, and F. Pérez-González, Eds. Springer Berlin Heidelberg, Oct. 2014, vol.8389, pp.488–502.

[28] M.U. Celik, G. Sharma, A.M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol.14, pp.253–266, Feb. 2005.

[29] Signal & Image Processing Institute, University of Southern California, "The USC-SIPI image database," http://sipi.usc.edu/database/

[30] Information technology — JPEG 2000 image coding system: Reference software. ISO/IEC IS-15444-5, 2003.