PAPER *Special Section on Digital Signal Processing*

A Method of Extracting Embedded Binary Data from JPEG Bitstreams Using Standard JPEG Decoder

Yoshihiro NOGUCHI†, ***Student Member***, **Hiroyuki KOBAYASHI**††, ***and*** **Hitoshi KIYA**†††, ***Regular Members***

SUMMARY We proposed a method for embedding binary data into JPEG bitstreams and extracting embedded data from JPEG bitstreams using the standard JPEG decoder. In the proposed method, we can decode the image from JPEG bitstreams into which the binary data is embedded first using the traditional standard JPEG decoder, and then we can extract the embedded binary data perfectly by the post-processing from the decoded JPEG image. For the post-processing, we use only the decoded image data to extract the embedded binary data. Namely, we do not need any kind of particular parameters, which are used for JPEG decoding, such as quantization table value. Thus, we can use the traditional standard JPEG decoder for the pre-processing of extracting binary data. Furthermore, we address the effect of the calculation bit accuracy of discrete cosine transform (DCT) and inverse discrete cosine transform (IDCT) for extracting embedded binary data perfectly as post-processing. Simulations using extracting embedded binary data as post-processing are presented to quantify some performance factors concerned. And we confirmed that the proposed method could be of practical use.
key words: *digital watermark, JPEG, embedding binary data, standard JPEG decoder*

1. Introduction

In many multimedia applications such as digital TV, DVD and Internet, video data is often compressed with JPEG [1] and MPEG [2]–[4]. In these applications, the embedding binary data into the compressed images is often required. These purposes of embedding binary data are for the index information of the image, the BIFS (Binary Format for Scene) information of MPEG-4 and broadcast monitoring of digital TV satellite link [5]. On the other hand, the embedded information for these purposes is the binary data, and the embedded data into lossy compressed image such as JPEG or MPEG must be extracted perfectly. Two ways of embedding data technique in the discrete cosine transform (DCT) domain have been proposed [8]–[13]. In these both methods, we had to prepare an exclusive decoder for extracting embedded data use only.

However, using traditional standard decoder is more suitable for extracting embedded data when we think of the applications of this technique. We first decode the image using standard decoder, then we extract the embedded data from decoded image by post-processing. The computational complexity of the post-processing for extracting data from the decoded image is required to be as small as we realize with software processing of CPU or DSP. Namely, we can decode the compressed image using traditional standard decoder. And furthermore, we can extract the embedded data perfectly by post-processing of decoded image data when we need the embedded data. So we can extend the real application of this technique.

We can realize the method that is satisfied those conditions by expanding the methods we have proposed before [10]–[13]. Moreover, we consider that any particular parameters for JPEG decoding use only such as a quantization table value should not be used for post-processing to use a standard JPEG decoder. We address the effect of the calculation bit size of DCT and IDCT for extracting embedded binary data by post-processing. In this paper, we propose the method of embedding binary data into JPEG bitstreams and extracting embedded data using standard JPEG decoder.

The technique of watermark which inserts image information into MPEG video has been studied [6], [7]. This technique is the method for inserting the signature information for copyright protection. There are some different points between the proposed insertion technique of the object information and the watermarking technique. The technique of watermarking prepare to deal with alter by others. And the technique of watermarking is required for the identification of the signature and we do not have to extract the inserted image of signature perfectly in such as the lossy JPEG or MPEG environment. On the other hand, we have to extract all of the inserted object information data correctly in the insertion technique of the object information.

The paper is organized as follows. Section 2 reviews the JPEG encoding algorithms and outlines the former proposed data embedding methods. In Sect. 3, we describe how we can insert binary data into JPEG bitstreams and extract the embedded binary data with post processing. Section 4 shows the results of some

Manuscript received December 2, 1999.
Manuscript revised February 23, 2000.
†The author is with the Central Technology Laboratory, Asahi Chemical Industry Co., Ltd., Atsugi-shi, 243-0021 Japan.
††The author is with the Department of Electrical Engineering, Tokyo Metropolitan College of Technology, Tokyo, 140-0011 Japan.
†††The author is with the Department of Electrical Engineering, Graduate School of Engineering, Tokyo Metropolitan University, Hachioji-shi, 192-0397 Japan.

simulations of the proposed method.

2. Former Methods for Embedding Binary Data into JPEG Bitstreams

In this section, we describe the former methods for embedding binary data into JPEG bitstreams, which have been proposed so far. Two ways of the technique which binary data is embedded into the DCT domain, which can extract embedded binary data perfectly, have been proposed [8]–[13]. We describe these two methods on each to the following.

2.1 Decoding Procedure Former Proposed Method-A [8], [9]

Figure 1 shows the block diagram for the JPEG encoder which can embed binary data proposed in [8], [9]. This processing flow is the structure which the binary data embedding block is added into the standard JPEG encoder. A input image data is divided into 8×8 blocks, and each 8×8 blocks are performed DCT, then each DCT coefficients as a result of DCT are quantized by qantization table. After that, 1-bit binary data is embedded into the least significant bit (LSB) of the quantized DCT coefficient at some particular position of 8×8 block. To be more concrete, the LSB of the quantized DCT coefficient is set to an even number in case we embed binary data '0' into JPEG bitstreams. On the other hand, the LSB of the quantized DCT coefficient is set to an odd number in case we embed binary data '1'.

Figure 2 shows the block diagram for the JPEG decoder which can extract the embedded binary data.

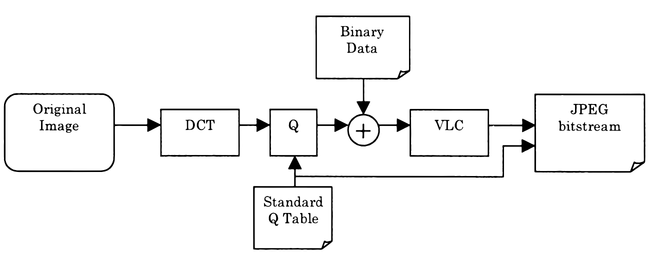

Fig. 1 Former encoding method-A for embedding data into JPEG bitstreams.

At the decoder side, we can decode the embedded binary data perfectly by extracting the LSB of that quantized DCT coefficient. In this method, the LSB of the quantized DCT coefficient must be extracted before inverse quantization to decode the embedded binary data. Therefore, we have to prepare a particular JPEG decoder for extracting data use only. Moreover, the quantization table must be needed when we extract the embedded binary data from the JPEG decoded image by post-processing as described later.

2.2 Decoding Procedure Former Proposed Method-B [10]–[13]

On the other hand, we outline the binary data embedding method which we have proposed so far [10]–[13]. Figure 3 shows the block diagram for the JPEG encoder which can embed binary data. In our proposed method, we replaced a specified quantized DCT coefficient data to embedded binary data. In other words, we change the quantized DCT coefficient itself instead of the LSB of quantized DCT coefficient when we embed the binary data into JPEG bitstreams. Thus, for the reason of the changing the whole bits of quantized DCT coefficient, we can modify the quantization table value which corresponds to the position of embedded data. And the modification of the quantization table allows to reduce the degradation of the decoded image quality. This encoding processing is also used in our proposed method, we describe it in detail in following Sect. 3.

We can extract the embedded binary data perfectly at JPEG decoding side by extracting the quantized DCT coefficient. However, we also have to prepare a particular JPEG decoder for extracting data use only.

We do not discuss the effect of embedding binary data in these two ways for the degradation of JPEG decoded image in this paper, because we already discussed about it in detail in [13]. And furthermore, we described the comparison of image quality under the condition of the same file size of the encoded compressed image data [13].

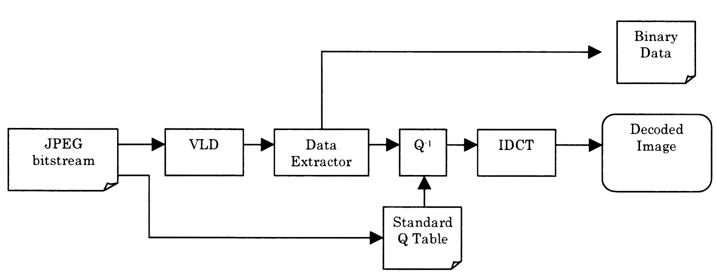

Fig. 2 Former decoding method-A for extracting data from JPEG bitstreams.

3. Proposed Method

In this section, we describe our proposed method for embedding binary data into JPEG bitstreams which we can extract the embedded binary data from decoded image by standard JPEG decoder. In our proposed method, the value of the quantization table does not need for the extracting embedded binary data by post-processing. Therefore, we can use the traditional standard JPEG decoder to decode images as pre-processing for extracting binary data. Thus, we do not have to prepare a particular JPEG decoder for extracting embedded data use only. We realize the post- processing which can extract the embedded binary data from the decoded JPEG image by the traditional standard JPEG decoder. Moreover, we consider that particular parameters for JPEG decoding use only such as the quantization table should not be used for post-processing.

We replace a k-th quantized DCT coefficient by zigzag scanning: $QDCT(k)$ to 1 bit embedding binary data (0 or 1) of the information. For example, 5,280 bits can be embedded in maximum into the gray scale image which size is 704×480 pixels.

3.1 Embedding Binary Data at JPEG Encoder

Figure 3 shows the block diagram of the proposed JPEG encoder which can embed the binary data. As shown in Fig. 4, the embedded one bit binary data of the information is replaced to the k-th quantized DCT coefficient by zigzag scanning: $QDCT(k)$. Where k is a constant, and the quantized DCT coefficients at same position of 8×8 block is replaced. So the JPEG decoder knows the position of $QDCT(k)$ where we embed the data in 8×8 block, the embedded binary data can be extracted perfectly. Even though the JPEG algorithm is non-information preserving coding, the JPEG encoding process after quantization is the information preserving coding. So we can extract the embedded binary data perfectly. In this method, we just replace the embedding binary data from the quantized DCT coefficient. Therefore, the embedded information JPEG bitstream is still suited to standard JPEG bitstream, and the standard JPEG decoder can decode image from that embedded data JPEG bitstream.

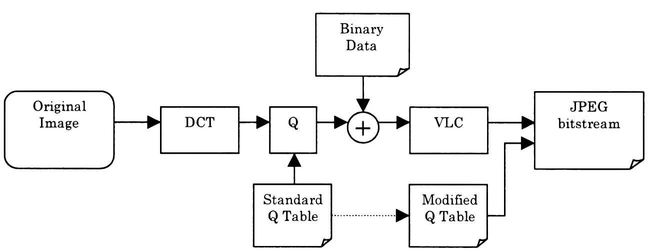

Fig. 3 Proposed data embedding JPEG encoder.

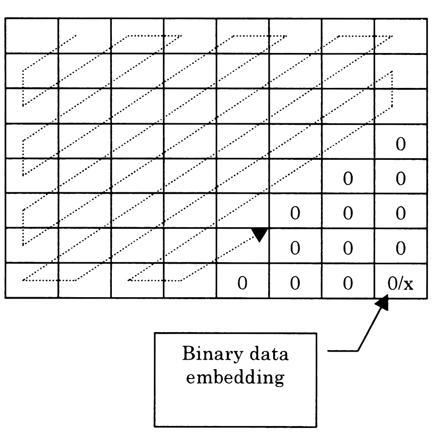

Fig. 4 Embedding binary data into quantized DCT coefficient.

(1) The Choice of Embedding Position: $QDCT(k)$

The coefficient area of DCT high frequency is better than low frequency for the position of inserted information data in 8×8 block. There are two reasons.

The first reason is the high frequency coefficients in 8×8 block are often become '0' after quantization. When we embed the data at that position where the original quantized coefficient is '0', we can reset the coefficient value to '0' after extracting the embedded binary data. As a result, we can embed the binary data without changing value of coefficients. Therefore we can reduce the degradation of image quality by replacing coefficient values.

And the second reason is the high frequency noise is less visible than low frequency noise in the DCT domain. Then we can reduce the degradation of decoded image quality when we decode embedded data JPEG bitstreams by traditional standard JPEG decoder. We can not extract and remove the embedded data, and this embedded data cause the degradation of image. It is better to choose a high frequency coefficient of quantized DCT coefficients for embedding data from the view point of image noise visibility. We have described about the detail of the image quality comparison on condition the position of embedded binary data in [13].

(2) The Modification of Quantization Table

In traditional standard JPEG decoding, we use a same qantization table as which is used for encoding. However, in our proposed method, we prepare a different quantization table for decoding from it for encoding, and send it to the decoder in the standard JPEG bitstream header.

The first purpose of preparing different quantization table is to reduce the noise of addition information data. As shown in Fig. 5, the values of a quantization table in high frequency area are so big. Therefore, even though the value which we replace for inserting information after quantization of DCT coefficients is so

Standard Q table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	66
14	13	16	24	40	57	69	57
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	36	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	110	103	99

Modified Q table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	66
14	13	16	24	40	57	69	57
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	36	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	110	103	①

Fig. 5 Modified quantization table.

small, it change to big value with inverse quantization and such big value cause image degradation so much. For preventing this image degradation, we change the value of quantization table where we insert the information data. Then we send this modified quantization table, and we use it for inverse quantization at decoding. Using this proposed method, we can make small value for quantization step where we embed the binary data, and we prevent the image degradation by embedding data.

Figure 5 shows a sample of the quantization table which we send to JPEG decoder in case the binary data is embedded into $QDCT(63)$. In the JPEG bit format, the area for a user-defined quantization table is prepared in a frame header field. Then the standard JPEG video decoder can decode image using our proposed modified qantization table.

And second purpose is to remove the quantization process in the post-processing for extracting embedded binary data from decoded JPEG image. The value of quantization table does not need for post-processing when we modify the quantization value to '1' as shown in Fig. 5. We only have to perform DCT the decoded JPEG image data at post-processing to extract the embedded binary data. It is not necessary to hand any kind of parameters except for decoded JPEG image data to post-processing. So the traditional standard JPEG decoder can be used. If we do not change the quantization table value to '1', the quantization processing is required at post-processing and also have to prepare the particular JPEG decoder from which the post-processing can obtain the quantization table value.

When the embedded data value is (0 or 1), the degradation of JPEG decoded image is smallest. However, we can choose the larger embedded data value such as (0 or 2) or (0 or 4) when we need expand the robustness for the calculation accuracy of DCT or IDCT as described later.

3.2 Extracting Binary Data at JPEG Decoder

Figure 6 shows the block diagram of proposed JPEG decoder and the post-processor which can extract embedded binary data perfectly. At the JPEG decoder side, the compressed JPEG bitstream is put into a variable length decoder (VLD) and decoded to the quantized DCT coefficient. And it is performed the inverse quantization (Q^{-1}) and the inverse DCT (IDCT), then we can obtain the decoded image. This procedure is just normal traditional standard JPEG decoder. To extract the embedded binary data, we perform DCT on the decoded pixel data at which we embedded. In the applications of this method, we do not assume the environment which some more image processing will be held on the decoded image before the embedded data is extracted.

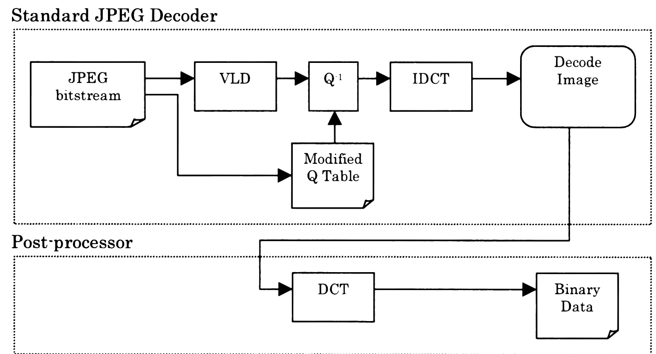

Fig. 6 Proposed data extracting JPEG decoder and post-processor.

In our proposed method, the traditional standard JPEG decoder can be used to extract the embedded binary data from JPEG bitstreams. We do not have to prepare the particular JPEG decoder for extracting embedded data use only. At the post-processor, the only input decoded image data is performed DCT, then we can extract the embedded binary data. In this proposed method, the post-processor does not need the value of the quantization table, because the value of the quantization table at embedded data position was changed to '1' by the proposed encoder.

Furthermore, we do not have to calculate all 8x8 DCT coefficients in this DCT processing. Only one DCT coefficient which position corresponds to the embedded data place is needed, so the computational complexity is approximately reduced to 1/64 compare with normal 8×8 DCT.

In our proposed method, one DCT and one IDCT are performed to extract the embedded binary data from decoded JPEG images. Therefore, we have to consider the calculation accuracy of DCT and IDCT to extract the embedded binary data perfectly. The effect of the calculation accuracy of DCT and IDCT will be described in Sect. 4.

3.3 Comparison with Former Methods [8], [9]

We compare our proposed method and the former method [8], [9] we described before. The first difference is that out proposed method changes the whole bits of the quantized DCT coefficient itself instead of the LSB of quantized DCT coefficient in case the binary data is embedded.

In the former method [8], [9], the all more significant bits than LSB of the quantized DCT coefficient still remain even after embedding binary data. Therefore, if we change the quantization table value, the original quantized DCT coefficient at the embedding position must be change the value. So the modified quantization table must not be used. Therefore, the quantization table value which corresponds to the position the data embedded is not ‘1’, so the quantization table value must to be handed to the post- processor for extracting embedded binary data.

Furthermore, it is impossible that we can extend the robustness for the calculation accuracy of DCT and IDCT using the former method [8], [9] as described later, because we can embed 1-bit binary data (0 or 1) only into the LSB of the quantized DCT coefficient. On the other hand, our proposed method allows us to extend the robustness for the calculation accuracy by embedding larger value such as (0 or 2) or (0 or 4) than (0 or 1), because we can embed whole bits binary data.

4. Simulation of Embedding Binary Data

We have simulated the proposed method with embedding binary data into JPEG bitstreams. We used the 704×480 grayscale image: Barbara and embedded 4,832 bits binary data as some character sentence which consists of 604 characters into the each $63rd$ quantized DCT coefficient by zigzag scanning: $QDCT(63)$ from upper left 8×8 block of the image. The quantization table we used was based on that as shown in figure 5 and the each quantization value is obtained by $Q-factor/50$ times. We changed the quantization scale to change the $Q-factor$ value. According to increase the value of $Q-factor$, the quantization scale becomes large and the compression ratio is increasing, then the quality of image is degrading.

We have simulated the degradation of decoded image using standard JPEG decoder when the binary data was embedded using our proposed method. Table 1 shows the PSNR comparison of JPEG decoded image between no- embedding binary data and embedding binary data. The left column shows the PSNR of no-embedding binary data and the right column shows the PSNR of embedding binary data into $QDCT(63)$ using our proposed method. There is no clear difference of PSNR between no-embedding data and embedding data, and the degradation of decoded image by embedded binary data do not show clear subjective difference.

Table 1 Evaluation of decoded image distortion with standard JPEG decoder: $PSNR$[dB].

Q-factor	Without Data-Embedding	With Data-Embedding using Proposed Method $QDCT(63)$
5	44.2404	44.2042
10	40.7913	40.7849
20	37.5403	37.5388
40	34.3546	34.3541
60	32.4469	32.4456
80	31.0561	31.0558

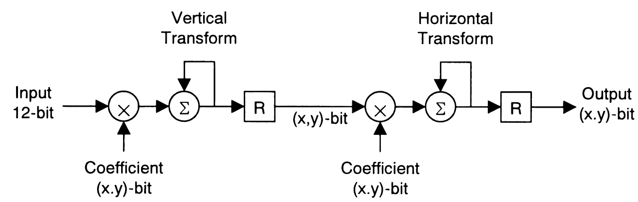

Fig. 7 Fixed-point calculation model of DCT and IDCT. R: Rounding processing, (x, y): x is bit width of integer and y is bit width of decimal.

5. Required DCT/IDCT Accuracy

In this section, we discuss about the required bit size of DCT and IDCT calculation for extracting the embedded binary data perfectly by post-processing. As shown in Fig. 6, both one DCT and one IDCT are needed to extract the embedded information from the decoded JPEG image data of Standard JPEG decoder. Therefore, the DCT and IDCT calculation accuracy is required to extract binary data perfectly.

We have led the theoretical maximum calculation error of DCT and IDCT under the fixed point calculation with several bit width conditions. Figure 7 shows the bit width model of DCT and IDCT calculation we used for the simulation of calculation error. Where R is the rounding processing, and x is the bit width for integer and y is the bit width for decimal of (x, y). We round off to each decimal places at each R block diagram point. In this evaluation, we use the model that perform 8×8 DCT of two dimensions as one 8×1 DCT of one dimension for column and one 1×8 DCT of one dimension for raw.

Table 2 shows the theoretical maximum error of

Table 2 Calculation accuracy of DCT/IDCT.

DCT/IDCT Bit Size	Theoretical Maximum Calculation Error
Floating point 32-*bit*	0.110914
Fixed point (12.4)-*bit*	0.155048
Fixed point (12.3)-*bit*	0.308387
Fixed point (12.2)-*bit*	0.510252
Fixed point (12.1)-*bit*	0.770879

both one DCT and one IDCT calculation on each bit width model. The actual error of limited bit width for DCT and IDCT depends on the input data value. However, we evaluated the theoretical maximum error of both one DCT and one IDCT as shown in Table 2. In case that fixed point bit width is larger than (12.2), the calculation error of both one DCT and one IDCT is smaller than 0.5, so we can extract the embedded binary data (0 or 1) perfectly. In the standard JPEG decoder, the bit width of IDCT processing is usually larger that 16-bit, and the required IDCT calculation accuracy of our proposed method is satisfied. Furthermore, the procedure of a fixed point calculation is the same as an integer calculation from the view point of a calculating data flow. And the accuracy of the DCT and IDCT with an integer calculation could be the same as a fixed point calculation on some condition. For example, the integer 14-bit calculation could have same accuracy of the fixed point (12.2)-bit calculation. Therefore, we can extend the fixed point calculation procedure to the integer calculation procedure.

Even though the bit width of DCT and IDCT is available only for fixed point (12.1)-bit for example, it will be no problem to choose the (0 or 2) as embedding binary value. Thus, our proposed method can be matched off against the variation of the calculation accuracy of DCT and IDCT. However, the former method [8], [9] can not extend the robustness for the calculation accuracy. Moreover, we do not need the quantization table value for extracting the embedded binary data by post-processor in our proposed method as compared with the former method [8], [9]. We can extract the embedded binary data from the JPEG decoded image data only.

6. Conclusions

We have proposed the method for embedding the binary data into JPEG bitstreams and for extracting the embedded binary data from JPEG decoded image. This proposed method allows us to use the traditional standard JPEG decoder to decode image which we use at post-processor for extracting the embedded binary data. We embed the binary data into a certain position of quantized DCT coefficients, and we can extract the embedded binary data perfectly. We confirmed that the required bit width of DCT and IDCT for our proposed method is smaller than the bit width of the usual standard JPEG decoder.

We choose a high frequency coefficient in the quantized DCT plane for inserting data, and we can prevent the degradation of image decoded quality. Furthermore, we modify the value of quantization table small, we success to keep image quality when we decode data inserted JPEG bitstreams by traditional standard decoder and also to remove the quantization processing in post-processing for extracting embedded binary data.

References

[1] ISO/IEC International Standard 10918-1, "Digital compression and coding of continuous-tone still images," 1991.
[2] ISO/IEC 11172-2, "Information technology-coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbits/s: Part 2 video," Aug. 1993.
[3] ISO/IEC 13818-2, "Information technology-generic coding of moving pictures and associated audio: Part 2 video," Nov. 1995.
[4] MPEG-4 Video Group, "MPEG-4 video verification model version 7.0," ISO/IEC JTC1/SC29/WG11, MPEG98/N1642, Bristol, England, April 1997.
[5] G. Depovere, T. Kalker, J. Haitsma, M. Maes, L.D. Strycker, P. Termont, J. Vandewege, A. Langell, C. Alm, P. Norman, G.O. Reilly, B. Howes, H. Vaanholt, R. Hintzen, P. Donnely, and A. Hudson, "The VIVA project: Digital watermarking for broadcast monitoring," Proc. IEEE Int. Conf. Image Proc., Kobe, Oct. 1999.
[6] F. Boland, J.O. Ruanaidh, and C. Dautzenberg, "Watermarking digital images for copyright protection," Proc. IEE Int. Conf. Image Proc. Applicat., pp.321–326, 1995.
[7] R. Wolfgang and E. Delp, "A watermark for digital images," Proc. 1996 Int. Conf. Image Proc., vol.III, pp.219–222, 1996.
[8] K. Matsui, Image hidden code? A method and application, pp.139–147, Morikita-publisher, 1993.
[9] K. Matsui, A basic of digital watermarking? A new protection technique of multimedia, Morikita-publisher, 1998.
[10] H. Kiya, Y. Noguchi, A. Takagi, and H. Kobayashi, "A method of inserting binary data into MPEG bitstreams," Proc. 1999 Int. Symp. Circuits and Systems, Orlando, May 1999.
[11] H. Kiya, Y. Noguchi, A. Takagi, and H. Kobayashi, "A method of inserting binary data into MPEG video in the compressed domain," IEICE Trans. Fundamentals, vol.E82-A, no.8, pp.1485–1492, Aug. 1999.
[12] H. Kiya, Y. Noguchi, A. Takagi, and H. Kobayashi, "A method of inserting binary data into MPEG bitstreams for video index labeling," Proc. IEEE Int. Conf. Image Proc., Kobe, Oct. 1999.
[13] H. Kobayashi, Y. Noguchi, and H. Kiya, "A method of embedding binary data into JPEG bitstreams," IEICE Trans., vol.J83-D-II, no.6, pp.1469–1476, 1999.

Yoshihiro Noguchi was born in Okayama, Japan, on March 2, 1962. He received the B.E. degree in electrical engineering from Kobe University, Japan, in 1985. In 1985, he joined Asahi Chemical Ind., where he is currently research engineer of Central Technology Laboratory, and also he is now a candidate for D.E. degree at Tokyo Metropolitan University. He was a visiting scholar of the University of California at Berkeley from 1994 to 1996. His research interests are in image processing, VLSI inplementation, and home networking. He is a member of the IEEE.

Hiroyuki Kobayashi received the B.E. and the D.E. degrees in electrical engineering from Tokyo Metropolitan University in 1992 and 1997 respectively. In 1997, he joined Tokyo Metropolitan College of Technology, where he is currently an Associate Professor of Electrical Engineering. His research interests are in multirate signal processing and image processing.

Hitoshi Kiya was born in Yamagata, Japan, on November 16, 1957. He received the B.E. and M.E. degrees in electrical engineering from Nagaoka University of Technology, Niigata, Japan, and the D.E. degree in electrical engineering from Tokyo Metropolitan University, Tokyo, Japan, in 1980, 1982, and 1987, respectively. In 1982, he joined Tokyo Metropolitan University, where he is currently a Professor of Electrical Engineering, Graduate School of Engineering. He was a visiting researcher of the University of Sydney in Australia from Oct. 1995 to March 1996. His research interests are in digital signal processing, multirate systems, adaptive filtering, image processing, and efficient algorithms for VLSI implementation. He is an Associate Editor of Trans. on Signal Processing of IEEE and Trans. of IEICE. He is a member of the IEEE, the Image Electronics Engineers of Japan and the Institute of Television Engineers of Japan.