Two error resilient coding schemes for wavelet-based image transmission based on data embedding and genetic algorithms

Li-Wei Kang a, Jin-Jang Leou b,*

a Institute of Information Science, Academia Sinica, Taipei 115, Taiwan, Republic of China
b Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan, Republic of China

Received 27 November 2004; accepted 8 August 2006
Available online 7 November 2006
Communicated by Shipeng Li

Abstract

For an entropy-coded wavelet-based image, such as a JPEG-2000 image, a transmission error in a codeword will not only affect the underlying codeword but may also affect subsequent codewords, resulting in a great degradation of the received image. In this study, two error resilient coding schemes for wavelet-based image transmission based on data embedding and genetic algorithms (GA’s) are proposed.

In this study, using JPEG-2000, an image is decomposed into six wavelet levels (levels 0–5). At the encoder, for levels 0–2, some important data useful for error concealment performed at the decoder are extracted and embedded into the compressed JPEG-2000 image bitstream. At the decoder, the important (embedded) data for each corrupted code block are extracted and used to facilitate error concealment. For levels 3–5, the wavelet coefficients of each corrupted code block are simply replaced by zeros.

Based on the simulation results obtained in this study, the performances of the two proposed schemes are better than those of three comparison approaches, namely, the Zero-S, Mean-S, and Inter approaches. The proposed schemes can recover high-quality wavelet-based JPEG-2000 images from the corresponding corrupted images up to a bit error rate of 0.5%.

Keywords: Error resilient coding; Error concealment; Wavelet-based image; JPEG-2000 image; Data embedding

© 2006 Elsevier Inc. All rights reserved.

This work was supported in part by National Science Council and Ministry of Economic Affairs, Republic of China under Grant Nos. NSC 95-2221-E-194-020-MY3 and 95-EC-17-A-02-S1-032.

* Corresponding author. Fax: +886 5 2720859.
E-mail address: jjleou@cs.ccu.edu.tw (J.-J. Leou).

1047-3203/5 - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2006.08.003
1. Introduction

To reduce transmission bit rate or storage capacity, many compression techniques have been developed for various applications, such as videophones, videoconferencing, World Wide Web (WWW), and multimedia communications [1]. Transmission of compressed images/video over noisy channels, such as the Internet or wireless networks, becomes a challenging problem [2,3]. For an entropy-coded wavelet-based image, such as a JPEG-2000 image [4–6], a transmission error in a codeword will not only affect the underlying codeword but may also affect subsequent codewords, resulting in a great degradation of the received image [2,3]. To cope with the synchronization problem, JPEG-2000 decomposes an image into several subbands, which are clustered into successive packets. Because each packet is headed with a 3-byte Resync code (a synchronization codeword), after the decoder receives any synchronization codeword, the decoder will resynchronize the decoding operation regardless of the preceding slippage. Additionally, in JPEG-2000, each packet contains several code blocks and all the code blocks can be coded independently so that transmission errors in a JPEG-2000 compressed image bitstream can be restricted within a code block. Although the propagation effect of transmission errors can be restricted within either a synchronization unit between two successive synchronization codewords or a code block in JPEG-2000, transmission errors will affect the underlying codeword and its subsequent codewords within either a corrupted synchronization unit or a corrupted code block. This will greatly degrade the quality of a received image, as an illustrated example shown in Fig. 1.

In this study, two error resilient coding schemes for wavelet-based image transmission are proposed. In general, error handling approaches include three categories [2,3], namely: (1) the error resilient encoding approach [7–12], (2) the error concealment approach [13–18], and (3) the encoder–decoder interactive error control approach [19]. The error resilient encoding approach [7–12] includes: (1) the channel coding approach [7,8], (2) the unequal error protection approach [9–11], and (3) the multiple description coding approach [12].

For the channel coding approach, some error correction codes are added into compressed image bitstreams such that transmission errors can be detected and/or corrected by additional side information. Although the channel coding approach can detect and/or correct transmission errors to a small extent, it will moderately increase the transmission bit rate. Additionally, the channel coding approach cannot work well in very noisy channels. To improve the channel coding approach, Cosman et al. [7] proposed a hybrid approach combining the forward error control and packetization of the embedded zerotree bitstream. Banister et al. [8] combined source and channel coding for JPEG-2000 image transmission, in which the source and channel code rates are jointly optimized to produce a compressed bitstream of fixed-size channel packets, and punctured turbo codes are used for channel coding. For the unequal error protection approach [9–11], a bitstream is divided into several distinct subbitstreams with different importances and different error protections. The multiple description coding approach [12] divides a bitstream into several subbitstreams, known as descriptions. Any single description can provide a basic quality and more descriptions together will provide an improved quality.

The error concealment approach [13–18] conceals the corrupted (lost) information due to transmission errors in the transmitted images/video at the decoder. In terms of the information used for concealment, the error concealment approach can be classified into three categories: (1) spatial (spectral) [13–16], (2) tem-
poral [17], and (3) hybrid [18]. For the spatial (spectral) error concealment approach [13–16], the information from correctly reconstructed and/or previously concealed spatially neighboring blocks of a corrupted block is used to conceal the corrupted block. Hemami and Gray [13] presented two techniques for lost low-frequency and high-frequency subband coefficient reconstructions, respectively. Low-frequency subband coefficient reconstruction is based on inherent properties of hierarchical subband decomposition, whereas high-frequency subband coefficient reconstruction is performed using linear interpolation. In Atzori et al. [15], a wavelet patch repetition procedure is used to predict the similarity between the contour structure in a damaged area and its surroundings. The patch used to conceal the corrupted region is obtained by minimizing a correlation measure with the spatial structure extracted from an adjacent undamaged subband. Lee and Chen [16] proposed an algorithm for recovering the damaged bit plane data of a JPEG-2000 image according to the cross subband and undamaged bit plane information. For the temporal concealment approach [17], the information of the corresponding blocks and/or their neighboring blocks from the previous/successive frames of a corrupted block is used to conceal the corrupted block. For the hybrid error concealment approach [18], both spatial and temporal information are employed to conceal corrupted blocks. Note that both the temporal and hybrid error concealment approaches [17,18] are applicable to video transmission only.

On the other hand, if a feedback channel can be set up from the decoder to the encoder [19], the decoder can inform the encoder about which parts of the transmitted information are corrupted (due to transmission errors), and then the encoder can adjust its encoding operations accordingly to suppress or eliminate the effect of transmission errors. For example, the temporal error propagation effect can be completely terminated by intra refreshing the affected macroblocks (blocks). If the automatic repeat request (ARQ) function is supported, the corrupted (lost) packets can be retransmitted. But it is not suitable for real-time applications due to processing delays.

The foregoing error resilient approaches concentrate on limiting error propagation and using the correctly received information to conceal the corrupted data. Using the information of spatially and/or temporally neighboring blocks of a corrupted block to conceal the corrupted block may introduce some problems. First, the information of spatially and/or temporally neighboring blocks may be unavailable (may also be corrupted). Second, the contents of a corrupted block and that of its spatially and/or temporally neighboring blocks may be distinct. In those cases, the concealed results of the foregoing error concealment approaches are usually not good enough. Recently, several error resilient coding approaches based on data embedding were proposed [20–31], in which some important data useful for error concealment performed at the decoder can be embedded into compressed image/video bitstreams, when they are encoded at the encoder. The embedded data should be “almost” invisible and cannot degrade the quality of images/video greatly, just like digital watermarking [32–36]. At the decoder, if some corrupted blocks are detected and located, the important (embedded) data for the corrupted blocks can be extracted and used to facilitate error concealment performed at the decoder. Yu and Yin [21] proposed a multimedia data recovery approach, in which a content-associative signature of a block in an image is generated and inserted imperceptibly into another (remote) block of the image. At the decoder, the embedded content-associative signature for each corrupted block is extracted and employed to reconstruct (conceal) the corrupted block. Yin et al. [22] embedded the block type and the edge direction index of a block within an image into the DCT coefficients of another (remote) block within the image by the odd–even embedding scheme. At the decoder, the embedded data for each corrupted block are extracted and employed to conceal the corrupted block by bilinear interpolation. Kurosaki et al. [26] proposed an error concealment approach using the layer structure of JPEG-2000, in which the most significant layer is embedded into the least significant layer of the JPEG-2000 bitstream. When the most significant layer is corrupted by transmission errors, the corrupted data are concealed by the important data embedded in the least significant layer. Lu [27] proposed a data embedding technique to offer error resilience in wavelet-based image transmission. The authentication information generated from the structural digital signature is used for error detection, whereas the recovery information (an approximate version of the original image) is used for error concealment.

In this study, two error resilient coding schemes for wavelet-based image transmission based on data embedding and genetic algorithms (GA’s) are proposed. Using JPEG-2000, an image is decomposed into six wavelet levels (levels 0–5). For level 0, an error resilient coding scheme based on data embedding and side-match vector quantization [24] is employed. For level 1, in the proposed basic scheme, the important (embedded) data for a corrupted code block are used to conceal the corrupted code block, whereas in the
proposed enhanced scheme, a GA-based error concealment scheme is used to conceal the corrupted code block. For level 2, the important (embedded) data, i.e., the mean values of the subblocks, for a corrupted code block are used to conceal the corrupted code block. For levels 3–5, the wavelet coefficients of each corrupted code block are simply replaced by zeros.

In this study, a transmission error may be either a single-bit error or a burst error. However, to simplify the processing steps in the two proposed schemes, a burst error containing several error bit segments will be treated as several transmission errors, i.e., a transmission error may be either a single-bit error or a burst error (segment) containing $N$ successive error bits.

This paper is organized as follows: a brief overview of the JPEG-2000 image compression standard is given in Section 2. A brief overview of the genetic algorithms is given in Section 3. The two proposed error resilient coding schemes for wavelet-based image transmission are addressed in Section 4. Simulation results are included in Section 5, followed by concluding remarks.

2. JPEG-2000 Image compression standard

The JPEG-2000 image compression standard [4–6] represents advances in image compression technology where the image coding system is optimized not only for efficiency but also for scalability and interoperability in network and mobile environments. It provides a set of features that are of importance to many applications. At the encoder, the source image is first decomposed into components, such as $Y$, $C_b$, and $C_r$. The image components are optionally decomposed into rectangular tiles. A discrete wavelet transform is applied on each tile, in which each tile is decomposed into different resolution levels. Each resolution level is composed of several subbands of wavelet coefficients. The subbands of wavelet coefficients are quantized and each subband is divided into rectangular blocks, i.e., nonoverlapping rectangles. Three spatially consistent rectangles (one from each subband at each resolution level) comprise a packet partition location or precinct. Each precinct is further divided into nonoverlapping rectangles, namely, code blocks, which form the input to the entropy coder. Within each subband, the code blocks are visited in a raster scan order. These code blocks are then coded a bit plane at a time starting with the most significant bit plane with a nonzero element to the least significant bit plane. These code blocks can be coded independently.

The compressed bitstreams from the code blocks in a precinct comprise the body of a packet. The precinct is a spatial representation corresponding to the packet in the bitstream representation in JPEG-2000. A collection of packets, one from each precinct of each resolution level, comprises a layer. A packet could be interpreted as one quality increment for one resolution level at one spatial location, since precincts correspond roughly to spatial locations. Similarly, a layer could be interpreted as one quality increment for the entire full resolution image. Each layer successively and monotonically improves the image quality. For example, a bitstream consists of a base layer and one or several enhancement layer(s). The base layer provides a lower but acceptable image quality, whereas the enhancement layer(s) incrementally improves the image quality. However, only one layer is used in the default mode of JPEG-2000. The final bitstream is organized as a succession of layers. The basic organization of a JPEG-2000 image bitstream is shown in Fig. 2 [4–6].

In this study, using the default mode of JPEG-2000, an image has only one tile and is decomposed into six wavelet resolution levels. Level 0 contains only one subband denoted by $LL_0$. Levels 1 through 5 individually contain three subbands denoted by $LH_i$, $HL_i$, and $HH_i$, $i = 1, 2, \ldots, 5$, respectively. For level 0, only the subband $LL_0$ is included in the first precinct and equivalently forms the first packet. For level $i$ ($i \geq 1$), the three subbands $LH_i$, $HL_i$, and $HH_i$ are together included in the $(i + 1)$th precinct and equivalently form the $(i + 1)$th

![Fig. 2. Basic organization of a JPEG-2000 image bitstream.](image-url)
packet. For example, \( LH_4 \), \( HL_4 \), and \( HH_4 \) are included in the fifth precinct or packet. That is, a wavelet-based JPEG-2000 image bitstream will contain six precincts or packets in the default mode. A collection of the six packets forms a layer and only one layer is used here. Each subband is divided into code blocks with the default size of \( 64 \times 64 \). If the size of a subband is less than or equal to \( 64 \times 64 \), the subband is just a code block. A \( 512 \times 512 \) image decomposed into 6 levels will contain 16 separate subbands, or equivalently 70 separate code blocks, as shown in Fig. 3. For the 70 code blocks (numbered from 0 to 69), for example, \( LL_0 \) in level 0 contains code block 0, \( LH_1, HL_1, \) and \( HH_1 \) in level 1 contain code blocks 1–3, respectively, \( LH_4 \) in level 4 contains code blocks 10–13, \( LH_5 \) in level 5 contains code blocks 22–37, and so on.

3. Genetic algorithms

Genetic algorithms (GA’s) [18, 37], originally proposed by Holland, represent a class of parallel adaptive search algorithms based on the mechanics of natural selection and the natural genetic system. GA’s behave well in many application areas, such as searching, optimization, and machine learning. GA’s can find near-global optimal solutions in a large solution space quickly. Compared with conventional search methods, the characteristics of GA’s include: (1) they are population-based: GA’s search for the solution from a population of points (not a single point). Here, a population means a large solution space; (2) new populations are created from old populations by means of three genetic operations, namely, reproduction, crossover, and mutation; (3) individuals in a population represent possible solutions and are usually described as binary bit strings containing 1’s and/or 0’s; and (4) GA’s use probabilistic transition rules so that they will not fall into local optima.

In GA’s, reproduction is based on the Darwinian survival of the fittest among bit strings created. A bit string with a larger fitness function value should contribute to the next generation with the higher probability and vice versa. Crossover usually divides two parent bit strings into two or more segments and then combines the segments to generate two offspring bit strings. For example, for the two bit strings, “0111” and “1001,” the crossover operation is assumed to be performed by splitting them at the second bit and then two new bit strings, “0101” and “1011,” will be generated. Crossover can produce offspring that are radically different from their parents. Mutation usually performs random changes in the bit strings by some operation, such as bit shifting, inversion, and rotation. For example, for the bit string, “0101,” the mutation operation is
assumed to be performed by inverting the first bit and then the new bit string “1101” will be generated. Mutation can extend the scope of the solution space and reduce the possibility of falling into local optima. In general, mutation is applied with a very low probability.

Before running a GA, the following decisions should be made: (1) the choice of a chromosome of a solution; (2) the choice of a way to create the initial population of solutions; (3) the definition of the fitness function; (4) the definition of genetic operators that affect the composition of offspring bit strings during reproduction; and (5) the setting of system parameters, including the population size (i.e., the number of possible solutions considered in one iteration), possibilities with which genetic operators are applied, etc.

The main steps of a GA are listed as follows:

Step 1. Initialize a population of bit strings.
Step 2. Evaluate the fitness function values of bit strings in the population.
Step 3. Create new bit strings by mating current bit strings and applying crossover and mutation.
Step 4. Delete bit strings with the worse fitness function values in the population to make room for the new bit strings.
Step 5. Evaluate the new bit strings and insert them into the population.
Step 6. If the stopping criterion is satisfied, stop and output the best bit string (the best solution); otherwise, go to Step 3.

Note that the stopping criterion is either the percentage of the fitness function value improvement between two successive iterations is smaller than a threshold or the number of GA’s iterations is larger than another threshold.

4. Proposed two error resilient coding schemes for wavelet-based image transmission

In this study, two error resilient coding schemes based on data embedding and genetic algorithms for wavelet-based image transmission are proposed and the JPEG-2000 compression standard is used as the platform. Within the two proposed schemes, the following issues will be addressed: (1) what kind of important data for the code blocks within a wavelet-based JPEG-2000 image should be extracted and embedded, (2) how to embed the important data into the corresponding “masking” code blocks, (3) where should the important data be embedded, and (4) how to extract and use the important (embedded) data to conceal the corrupted code blocks at the decoder.

4.1. What kind of important data should be extracted and embedded

4.1.1. Data extraction for level 0 of JPEG-2000 images

In this study, for a 512×512 wavelet-based JPEG-2000 image, level 0 (LL₀) contains only one 16×16 code block. LL₀, a down-sampled version of the original image, should be protected carefully. Here, an error resilient coding scheme based on data embedding and side-match vector quantization (VQ) [24] is employed to encode LL₀. LL₀ is first divided into 16 4×4 subblocks. Similar to the VQ system, both the encoder and the decoder have an identical codebook, in which the dimension of each codeword is the same as the size, 4×4, of a subblock. For each subblock in LL₀, there exists a closest codeword in the codebook with the smallest mean-squared error (MSE) between the subblock and the codeword in the codebook. For a codebook of size $M$ (here, $M = 128$), only $b_{\text{index}} = \lceil \log_2 M \rceil$ bits are required to represent the index of each codeword, where $\lceil x \rceil$ denotes the ceiling of $x$. In the two proposed schemes, for each subblock in LL₀, the codebook index of each subblock is extracted as its important data and the codebook of size $M$ is trained by 2304 subblocks of the corresponding 144 code blocks (LL₀’s) within 144 training JPEG-2000 images. Based on the suggestion for VQ codebook training in [24] and the simulation results obtained in this study, 2304 training subblocks of size 4×4 are empirically enough. Here, VQ codebook training is performed by the fast VQ training algorithm developed in [24].
4.1.2. Data extraction for level 1 of JPEG-2000 images

Level 1 contains three subbands, $LH_1$, $HL_1$, and $HH_1$, and each subband contains only one $16 \times 16$ code block. Each code block is also divided into $16 \times 16$ subblocks. In the proposed basic scheme, the mean value ($b_{\text{mean}}$ bits) for each subblock is extracted as its important data. In the proposed enhanced scheme, the mean ($b_{\text{mean}}$ bits), variance ($b_{\text{var}}$ bits), average intersample difference (AID) ($b_{\text{AID}}$ bits), maximum wavelet coefficient (MAX) ($b_{\text{MAX}}$ bits), minimum wavelet coefficient (MIN) ($b_{\text{MIN}}$ bits), the position of MAX (4 bits), the position of MIN (4 bits), and four wavelet coefficients, $C_0$, $C_2$, $C_8$, and $C_{10}$, ($4 \times b_C$ bits) in positions 0, 2, 8, and 10 (as shown in Fig. 4) are extracted as its important data. Note that the positions for the 16 wavelet coefficients in a $4 \times 4$ subblock are numbered from 0 to 15 in a raster scan order. The average intersample difference (AID) for an $M \times N$ subblock, $B(x,y)$, $x = 0, 1, 2, \ldots, M - 1$, $y = 0, 1, 2, \ldots, N - 1$, is given by

$$AID = \frac{\sum_{y=0}^{N-1} \sum_{x=1}^{M-1} |B(x,y) - B(x-1,y)| + \sum_{x=0}^{M-1} \sum_{y=1}^{N-1} |B(x,y) - B(x,y-1)|}{[N \times (M - 1) + M \times (N - 1)]}.$$  

In the proposed enhanced scheme, the mean and variance for each subblock in a code block denote two global properties, namely, the global energy and variation, of the subblock, respectively. On the other hand, the AID for each subblock denotes a local property by considering the differences between successive wavelet coefficients in the subblock. The four wavelet coefficients, $C_0$, $C_2$, $C_8$, and $C_{10}$, in positions 0, 2, 8, and 10 in a subblock can be treated as the four “anchor” wavelet coefficients. Note that if more neighboring wavelet coefficients of a “corrupted” wavelet coefficient in the corrupted subblock can be first recovered perfectly, the “corrupted” wavelet coefficient will be probably well concealed. Due to the limited capacity for data embedding, it is impossible to embed all the wavelet coefficients in a subblock. Hence, four nonadjacent wavelet coefficients, $C_0$, $C_2$, $C_8$, and $C_{10}$ (or alternately $C_1$, $C_3$, $C_5$, and $C_7$), of a subblock are selected as its important data. Then, the rest of the coefficients in a corrupted subblock will be concealed more easily by the proposed GA-based error concealment scheme described in Section 4.4.

4.1.3. Data extraction for levels 2–5 of JPEG-2000 images

Level 2 contains three subbands, $LH_2$, $HL_2$, and $HH_2$, and each subband contains only one $32 \times 32$ code block. Each code block is divided into $16 \times 8 \times 8$ subblocks. In the two proposed schemes, the mean value ($b_{\text{mean}}$ bits) of each subblock is extracted as its important data. Due to the limited capacity for data embedding and the lower importance of levels 3–5, no important data are extracted from levels 3 to 5 for data embedding.

4.2. How to embed important data

Several data embedding schemes for wavelet-based and JPEG-2000 images have been proposed [33–36]. For some data embedding schemes, the original image is required to extract the embedded data, whereas for some data embedding scheme, it is not guaranteed that the embedded data can be always extracted precisely from the compressed JPEG-2000 image bitstream. Here, a simple data embedding scheme for wavelet-based JPEG-2000 images is employed.

In this study, for the three subbands of level 5 (i.e., $LH_5$, $HL_5$, and $HH_5$), if the absolute value of a quantized wavelet coefficient is either zero or a small value, the corresponding decoded coefficient will be zero. On the contrary, if the absolute value of a quantized wavelet coefficient is a large value, the corresponding decoded
For a quantized wavelet coefficient, $C_i$, for data embedding in the three subbands of level 5 and the data bit, $b_j (b_j = 0 \text{ or } 1)$, to be embedded, the data embedding scheme is

$$
C_i = \begin{cases} 
0 & \text{if } b_j = 0 \\
T_W & \text{if } b_j = 1 \text{ and } C_i \geq 0, \\
-T_W & \text{if } b_j = 1 \text{ and } C_i < 0,
\end{cases}
$$

where $T_W$ is a predefined large positive integer number. Here, $T_W$ is selected empirically. Because a quantized wavelet coefficient may not be completely encoded into the compressed JPEG-2000 image bitstream, if $T_W$ is not large enough, a quantized coefficient $C_i (= T_W)$ embedding a data bit “1” may be decoded to be “0” at the decoder, resulting in the loss of the embedded data bit “1.”

At the decoder, if a decoded wavelet coefficient in the three subbands of level 5 for data embedding is zero, the embedded data bit is “0,” whereas if the decoded wavelet coefficient is nonzero, the embedded data bit is “1.” However, in this study, if a data bit “1” is extracted, the decoded wavelet coefficient for embedding data bit “1” will be forced to be zero so that the image quality degradation induced by data embedding can be reduced. Note that because most of the quantized wavelet coefficients in level 5 are zeros, forcing a decoded wavelet coefficient to be zero after extracting a data bit “1” will not degrade the image quality greatly.

4.3. Where to embed important data

As the illustrated example shown in Fig. 3, level 5 contains three subbands, $LH_5$, $HL_5$, and $HH_5$, and each subband contains $16 \times 64 \times 64$ code blocks. In the two proposed schemes, the important data for each subblock in $LH_1$ and the corresponding subblock in $LH_2$ are concatenated into a bitstream and embedded into the corresponding “masking” code block in $LH_5$. For example, the important data (10 bits in the proposed basic scheme and 101 bits in the proposed enhanced scheme) for the upper-left subblock in code block 1 ($LH_1$) and those (identically 10 bits in the two proposed schemes) for the upper-left subblock in code block 4 ($LH_2$) are concatenated into a bitstream (total 20 bits in the proposed basic scheme and total 111 bits in the proposed enhanced scheme) and embedded into the corresponding “masking” code block, i.e., code block 22 (which can embed at most 4096 bits), in $LH_5$. For another example, the important data (10 bits in the proposed basic scheme and 101 bits in the proposed enhanced scheme) for the bottom-right subblock in code block 1 ($LH_1$) and those (identically 10 bits in the two proposed schemes) for the bottom-right subblock in code block 4 ($LH_2$) are concatenated into a bitstream (total 20 bits in the proposed basic scheme and total 111 bits in the proposed enhanced scheme) and embedded into the corresponding “masking” code block, i.e., code block 37 (which can embed at most 4096 bits), in $LH_5$. Similarly, the important data for each subblock in $HL_1$ (total 16 subblocks) and those for the corresponding subblock in $HL_5$ (total 16 subblocks) are concatenated into a bitstream and embedded into the corresponding “masking” code block in $HL_5$. The important data for each subblock in $LL_0$ (total 16 subblocks), those for the corresponding subblock in $HH_1$ (total 16 subblocks), and those for the corresponding subblock in $HH_5$ (total 16 subblocks) are concatenated into a bitstream and embedded into the corresponding “masking” code block in $HH_5$. For example, the important data (identically 7 bits in the two proposed schemes) for the upper-left subblock in code block 0 ($LL_0$), those (10 bits in the proposed basic scheme and 101 bits in the proposed enhanced scheme) for the upper-left subblock in code block 3 ($HH_1$), and those (identically 10 bits in the two proposed schemes) for the upper-left subblock in code block 6 ($HH_5$) are concatenated into a bitstream (total 27 bits in the proposed basic scheme and total 118 bits in the proposed enhanced scheme) and embedded into the corresponding “masking” code block, i.e., code block 54 (which can embed at most 4096 bits), in $HH_5$. For each concatenated bitstream, the data bits will be embedded into the quantized wavelet coefficients in the corresponding “masking” code block in level 5 in a raster scan order, in which each data bit will be embedded into one quantized wavelet coefficient by using Eq. (2). The important data (total 1072 bits in the proposed basic scheme and total 5440 bits in the proposed enhanced scheme) extracted from levels 0 to 2 can be completely embedded into level 5 (which can embed at most 196,608 bits). However, the embedded data will degrade the error-free image quality when the bit rate is kept or only slightly increased. Hence, the more the important data to be embedded are, the lower the error-free image quality will be.
4.4. How to use important data for error concealment

Because the code blocks in a wavelet-based JPEG-2000 image can be encoded independently [4–6], in this study, the corrupted code blocks in a wavelet-based JPEG-2000 image are detected and located first at the decoder. For each corrupted code block, if its corresponding masking code block is correctly received, the important data for the corrupted code block (or subblock) will be extracted from its corresponding masking code block.

4.4.1. Proposed error resilient coding scheme for level 0

For a corrupted code block in level 0 (LL0), the codebook index for each subblock of the corrupted code block is extracted and the closest codeword in the codebook is used to conceal the corrupted subblock. If the masking code block of a subblock is also corrupted, its four (four-connected) spatially neighboring subblocks will be concealed first if they can be concealed with their correctly extracted codebook indexes. Then, the codebook index of the corrupted subblock can be estimated by the side-match VQ technique [24], and the corrupted subblock can be concealed by the corresponding closest codeword in the codebook for the estimated codebook index. In side-match VQ [24], for a corrupted subblock, the four “boundary” wavelet coefficients are first recovered by using the “boundary” wavelet coefficients of the concealed spatially neighboring subblocks. The detailed processing steps of side-match VQ can be found in [24]. However, based on the simulation results, on the average, the important data for a subblock in a corrupted code block in level 0 are usually available.

4.4.2. Two proposed error resilient coding schemes for level 1

For the corrupted code blocks in level 1 (LH1, HL1, and HH1), in the proposed basic scheme, the mean value for each subblock in a corrupted block is extracted and used to conceal all the wavelet coefficients in the corrupted subblock, whereas in the proposed enhanced scheme, a GA-based error concealment scheme is proposed.

For each subblock in a corrupted code block in level 1, the corresponding important data, including MAX, MIN, the positions of MAX and MIN, and four wavelet coefficients, $C_0$, $C_2$, $C_8$, and $C_{10}$ (or alternately $C_1$, $C_3$, $C_5$, and $C_7$), are extracted and used to recover the corresponding wavelet coefficients first, and the remaining wavelet coefficients will be concealed by the proposed GA-based error concealment scheme.

As described in Section 3, before running a GA, the following five decisions should be made.

1. The choice of a chromosome of a solution. As shown in Fig. 4, the 10 wavelet coefficients denoted by “X” in a subblock totally can be represented as a binary bit string ($b_w$ bits per coefficient). Because the total number of bits within such a bit string is 10×$b_w$, the whole solution space of the 10 wavelet coefficients in a corrupted code block will contain $2^{10×b_w}$ possible solutions. In the worst case, if MAX and MIN are exactly equal to two among the four coefficients, $C_0$, $C_2$, $C_8$, and $C_{10}$, the whole solution space will contain $2^{12×b_w}$ possible solutions. For each subblock in a corrupted code block in level 1, GA’s are used to find the best candidate concealed subblock with the smallest fitness function value.

2. The choice of a way to create the initial population of solutions. Here, the initial population contains (i) randomly selected bit strings, in which each wavelet coefficient to be concealed (denoted by “X”) can be set to a random number between MIN and MAX, and (ii) user-specified bit strings, in which each wavelet coefficient to be concealed (denoted by “X”) can be set to “zero” or the mean value of the subblock.

3. The definition of the fitness function. Here, the proposed fitness function, $f(B)$, for evaluating the goodness of a candidate concealed subblock, $B$, is formed by linearly weighted combining the three statistical measures, namely, the mean, variance, and AID (Eq. (1)) of $B$. That is, $f(B)$ is given by

$$f(B) = w_M \times (M - \bar{M})^2 + w_V \times (V - \bar{V})^2 + w_A \times (A - \bar{A})^2$$

where $M$, $V$, and $A$ are the extracted mean, variance, and AID values, respectively, $\bar{M}$, $\bar{V}$, and $\bar{A}$ are the mean, variance, and AID values of the candidate concealed subblock, respectively, and $w_M$, $w_V$, and $w_A$ are three weighting coefficients with $w_M + w_V + w_A = 1$. In this study, the smaller the fitness function value is, the better the error concealment performance will be.
The definition of genetic operators. Here, the employed genetic operations can be defined as follows: for reproduction, half of the bit strings with the smaller fitness function values within the current population will be selected into the mating pool. For crossover, the two parent bit strings are crossed over via string segments. For each pair of string segments ($b_n$ bits per string segment), a randomly selected position will be used for crossover of the pair of string segments. Mutation is realized by performing a bit inversion operation on a randomly selected bit within each string segment.

The setting of system parameters. Here, the population size, the maximum number of iterations, and the probability of applying the mutation operation for GA’s are denoted by $MAXPOP$, $MAXIT$, and $P_{\text{mutation}}$, respectively, which will be selected empirically.

GA’s for error concealment will be iteratively performed until the stopping criterion is satisfied. The stopping criterion is either the percentage of the fitness function value improvement between two successive iterations is smaller than a threshold or the number of GA’s iterations is larger than $MAXIT$. The best binary bit string (the best candidate concealed subblock) with the smallest fitness function value will be finally used to conceal the subblock of the corrupted code block. Note that each wavelet coefficient will be normalized to a positive integer before performing the proposed GA-based scheme. The proposed GA-based error concealment scheme is listed as follows and summarized in Fig. 5.

The main steps of the proposed GA-based error concealment scheme for a corrupted subblock:

Step 1. Initialize a population of bit strings based on randomly selected bit strings and user-specified bit strings.

Step 2. Evaluate the fitness function values of bit strings in the population based on the proposed fitness function (Eq. (3)).

Step 3. Create new bit strings based on the three genetic operators, namely, reproduction, crossover, and mutation.

Step 4. Delete bit strings with the worse fitness function values in the population to make room for the new bit strings.

Step 5. Evaluate the new bit strings and insert them into the population.

Step 6. If the stopping criterion is satisfied, stop and output the best bit string (the best candidate concealed subblock) to conceal the corrupted subblock; otherwise, go to Step 3.

Fig. 5. The proposed GA-based error concealment scheme.
4.4.3. Proposed error resilient coding scheme for levels 2–5

For the corrupted code blocks in level 2 (LH$_2$, HL$_2$, and HH$_2$), the important data for a corrupted code block, i.e., the mean values of all subblocks in a corrupted code block, are extracted and used to conceal all wavelet coefficients of all subblocks in the corrupted code block. For levels 3–5, the wavelet coefficients of the corrupted blocks are simply replaced by zeros.

The two proposed schemes operate on the quantized wavelet coefficients in levels 1 and 2 so that the number of bits required to represent the extracted important data and quantized wavelet coefficients can be reduced. If the masking code block of a subblock in a corrupted code block in levels 1 or 2 is also corrupted, the subblock can be concealed by the simple error concealment scheme described as follows. Each corrupted code block in level 1 is concealed by “averaging” the other correctly received code blocks in level 1. Each corrupted code block in level 2 is concealed by the corresponding up-sampling code block in level 1. The simple error concealment scheme is more suitable for concealing a smooth area than an edge area. However, based on the simulation results, on the average, the important data for a corrupted code block in levels 1 or 2 are usually available.

5. Simulation results

The two proposed error resilient coding schemes have been implemented on a Pentium-III 1.14 GHz PC using the C++ programming language. Several test $512 \times 512$ color images “all outside” the VQ codebook training images for $LL_0$ with different bit error rates (BER) and average length of burst errors ($N_{\text{ave}}$) are used to evaluate the performances of the two proposed schemes (denoted by “proposed basic” and “proposed enhanced”), where $N_{\text{ave}}$ is given by $N_{\text{ave}} = \sum_{i=1}^{N} i \times P_i$, where $P_i$ is the probability of a burst error (segment) containing $i$ successive error bits with $\sum_{i=1}^{N} P_i = 1$. Here $N_{\text{ave}}$ is set to the range (1,5), i.e., the maximum number $N$ of successive error bits of a burst error (segment) is set to 5, which can be set to a larger number without any difficulty. The peak signal-to-noise ratio (PSNR) is employed in this study as the objective performance measure for the three components ($Y$, $C_B$, and $C_R$) of a wavelet-based JPEG-2000 image. The mean square error between an original image and the corresponding reconstructed (concealed) image, denoted by MSE, is given by

$$\text{MSE} = \frac{(4 \times \text{MSE}_Y + \text{MSE}_{CB} + \text{MSE}_{CR})}{6},$$

where $\text{MSE}_Y$, $\text{MSE}_{CB}$, and $\text{MSE}_{CR}$ are the corresponding MSE values of the $Y$, $C_B$, and $C_R$ components of a wavelet-based JPEG-2000 image, respectively. The PSNR of a wavelet-based JPEG-2000 image is given by

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right).$$

For the two proposed data embedding schemes, the numbers of bits for important data, including $b_{\text{index}}$, $b_{\text{mean}}$, $b_{\text{variance}}$, $b_{\text{AID}}$, $b_{\text{MIN}}$, $b_{\text{MAX}}$, and $b_{\text{y}}$, are set to 7, 10, 9, 9, 10, 11, and 11, respectively, the threshold $T_W$ is set to 90,000, and the number of bits, $b_n$, to represent a wavelet coefficient is set to 11. For GA’s, MAX-POP, MAXIT, and $P_{\text{mutation}}$ are set to 100, 20, and 0.01, respectively, and $w_M = w_Y = w_A = 1/3$, which are empirically selected and the significances of the three statistical measures, namely, the mean, variance, and AID are found to be almost equal.

To evaluate the performances of the two proposed schemes, three comparison approaches [13] are implemented in this study. They are (1) zero-substitution (denoted by Zero-S), which simply replaces all the wavelet coefficients in a corrupted code block by zeros; (2) mean-substitution (denoted by Mean-S), in which the mean values for all the 16 4 $\times$ 4 subblocks in level 0 are assumed to be available at the decoder and used to conceal the corrupted code blocks in level 0. Each corrupted code block in level 1 is concealed by “averaging” the other correctly received code blocks in level 1. Each corrupted code block in level 2 is concealed by the corresponding up-sampling code block in level 1 if it is correctly received. If the corresponding code block in level 1 of the corrupted code block in level 2 is also corrupted, the corrupted code block in level 2 is concealed by “averaging” the other correctly received code blocks in level 2. The wavelet coefficients of the corrupted code blocks in levels 3–5 are simply replaced by zeros; (3) interpolation for corrupted wavelet coefficients (denoted by Inter), in which a cubic interpolative surface function is used to interpolate the corrupted coefficients in level 0, whereas linear interpolation is used to interpolate the corrupted coefficients in all other subbands (levels 1–5) [13].
In terms of PSNR (dB), the simulation results for the “Lenna” and “Fruits” images with bit rate = 0.4 bpp and different BER and $N_{ave}$ of the three comparison approaches and the two proposed schemes are illustrated in Figs. 6–9, in which 50 runs followed by averaging the corresponding 50 PSNR values are performed for each experiment. As a subjective measure of the quality of the concealed images, the error-free, error-free with data embedding, corrupted, and concealed images of the “Lenna” and “Fruits” images with bit rate = 0.4 bpp by the three comparison approaches and the two proposed schemes are shown in Figs. 10 and 11. The simulation results for the “Lenna” image with bit rate = 0.4 bpp, BER = 0.1%, and $N_{ave} = 3$ by the proposed enhanced scheme with different numbers of GA iterations are shown in Fig. 12. The PSNR values (dB) for the error-free “Lenna” and “Fruits” images with bit rate = 0.4 bpp encoded by the original JPEG-

<table>
<thead>
<tr>
<th>BER (%)</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>35</td>
</tr>
<tr>
<td>0.03</td>
<td>30</td>
</tr>
<tr>
<td>0.05</td>
<td>25</td>
</tr>
<tr>
<td>0.1</td>
<td>20</td>
</tr>
<tr>
<td>0.5</td>
<td>10</td>
</tr>
</tbody>
</table>

Fig. 6. The simulation results, PSNR (dB), for the “Lenna” image with bit rate = 0.4 bpp, $N_{ave} = 3$, and different BER using the three comparison approaches and the two proposed schemes.

<table>
<thead>
<tr>
<th>BER (%)</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>0.03</td>
<td>25</td>
</tr>
<tr>
<td>0.05</td>
<td>20</td>
</tr>
<tr>
<td>0.1</td>
<td>15</td>
</tr>
<tr>
<td>0.5</td>
<td>10</td>
</tr>
</tbody>
</table>

Fig. 7. The simulation results, PSNR (dB), for the “Lenna” image with bit rate = 0.4 bpp, BER = 0.1%, and different $N_{ave}$ using the three comparison approaches and the two proposed schemes.

<table>
<thead>
<tr>
<th>BER (%)</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>35</td>
</tr>
<tr>
<td>0.03</td>
<td>30</td>
</tr>
<tr>
<td>0.05</td>
<td>25</td>
</tr>
<tr>
<td>0.1</td>
<td>20</td>
</tr>
<tr>
<td>0.5</td>
<td>10</td>
</tr>
</tbody>
</table>

Fig. 8. The simulation results, PSNR (dB), for the “Fruits” image with bit rate = 0.4 bpp, $N_{ave} = 4$, and different BER using the three comparison approaches and the two proposed schemes.
2000 and the two proposed schemes are illustrated in Table 1. The sizes (bytes) of the bitstreams for the “Lenna” and “Fruits” images with bit rate = 0.4 bpp encoded by the original JPEG-2000 and the two proposed schemes are illustrated in Table 2. In terms of PSNR (dB), the simulation results for the “Lenna” and “Fruits” images with bit rate = 0.4 bpp by the proposed basic scheme and the proposed enhanced scheme without/with GA’s are illustrated in Table 3. The decoding times (seconds) for the two images, “Lenna” and “Fruits,” of the original JPEG-2000 and the two proposed schemes are illustrated in Table 4. Note that for all the simulations,
the three comparison approaches operate on the bitstreams encoded by the original JPEG-2000, whereas the two proposed schemes operate on the bitstreams encoded by the two proposed (basic and enhanced) schemes, respectively.

6. Concluding remarks

Based on the simulation results obtained in this study, several observations can be found. (1) Based on the simulation results shown in Figs. 6–11, the concealment results of the two proposed schemes are better than those of the three comparison approaches. (2) Under the same BER, the concealed results of the corrupted images with the larger \( N_{\text{ave}} \) are usually better than those of the corrupted images with the smaller \( N_{\text{ave}} \). That is, because under the same BER, on the average, the case with the larger \( N_{\text{ave}} \) will produce fewer transmission errors than that for the case with the smaller \( N_{\text{ave}} \). However, because fewer transmission errors will not always produce fewer corrupted blocks, the above rule is not valid in a few cases. (3) Usually, the higher the BER is, the better the performance gains of the two proposed schemes over the three comparison approaches will be. (4) Based on the simulation results shown in Table 1, the average image quality degradation of the images with data embedding in the proposed basic and enhanced schemes are about 1 and 2 dB, respectively. However, the embedded important data are perceptually invisible in Figs. 10 and 11 because the important data are embedded into \( LH_5 \), \( HL_5 \), and \( HH_5 \) (i.e., the three highest wavelet subbands in level 5). (5) Based on the simulation results shown in Fig. 12 and Table 3, as the number of GA iterations increases, the PSNR values of the concealment results of the proposed enhanced scheme are improved. However, this type of improvement is gradually saturated when the number of iterations increases. When no GA iterations are performed, the proposed enhanced scheme still outperforms the proposed basic scheme because the extracted wavelet coefficients, such as MAX, MIN, \( C_{10} \), \( C_2 \), \( C_3 \), and \( C_4 \) (or alternately \( C_1 \), \( C_5 \), \( C_6 \), and \( C_7 \)), can be used to conceal the corresponding wavelet coefficients first. (6) Based on the simulation results shown in Table 2, the percentages of bit rate increment due to data embedding in the two proposed schemes are usually below 1%. (7) Based on the simulation results shown in Table 4, the decoding time of the proposed basic scheme is slightly larger than that of the original JPEG-2000 scheme, whereas the decoding time of the proposed enhanced scheme is larger than that of the original JPEG-2000 scheme. The proposed basic scheme is more suitable for real-time image transmission, whereas the proposed enhanced scheme is more suitable for non-real-time image transmission.

Based on the experimental results reported in three recent error concealment approaches for JPEG-2000 images [14–16] and the experimental results obtained in this study, some observations can be found as follows. In [15], it is reported that their proposed error concealment approach for JPEG-2000 images outperforms each corrupted image by about 2 dB, whereas the two proposed (basic and enhanced) schemes outperform each corrupted image by 4–10 and 5–12 dB, respectively. In [14], it is reported that their proposed error concealment approach for JPEG-2000 images outperforms the Zero-S approach by 1.5–3 dB and in [16], it is reported that their proposed error concealment approach for JPEG-2000 images outperforms the Zero-S approach by 0.6–3 dB, whereas the two proposed (basic and enhanced) schemes outperform the Zero-S approach by 3–9 and 4–11 dB, respectively. Therefore, in terms of PSNR (dB), on the average, the error concealment performances of the two proposed schemes are usually better than those of the three recent approaches developed in [14–16].

On the other hand, in the two proposed schemes, the encoder should perform the standard JPEG-2000 encoding process as well as important data extraction and embedding. In the proposed basic scheme, the decoder should perform the standard JPEG-2000 decoding process, important data extraction, and error concealment by a simple mean substitution concealment scheme (low complexity), whereas in the proposed enhanced scheme, including the standard JPEG-2000 decoding process and important data extraction, the iterative GA-based error concealment scheme (high complexity) will be performed at the decoder. In the three recent approaches developed in [14–16], either the encoder performs only the standard JPEG-2000 encoding process or the decoder performs the standard JPEG-2000 decoding process and a simple error concealment scheme (low complexity). Therefore, the computational complexities of the two proposed schemes are expensive than those of the three recent approaches developed in [14–16]. However, the three recent approaches [14–16] cannot deal with some possible error situations, such as \( L_{L_0} \) is corrupted or a complete wavelet coefficient is corrupted (instead of just some bit plane data). But the two proposed schemes can deal with the above-mentioned cases. Additionally the recent error concealment approach proposed in [25] is particularly designed for ROI
Fig. 11. The error-free and concealed JPEG-2000 images of the “Fruits” image with bit rate = 0.4 bpp, BER = 0.1%, and $N_{ave} = 3$: (a) the error-free image; (b and c) the error-free images with data embedding by the proposed basic and enhanced schemes, respectively; (d) the corrupted image; (e–i) the concealed images by Zero-S, Mean-S, Inter, and the two proposed (basic and enhanced) schemes, respectively.

Fig. 12. The simulation results for the “Lenna” image with bit rate = 0.4 bpp, BER = 0.1%, and $N_{ave} = 3$ by the proposed enhanced scheme with different numbers of GA iterations.

Table 1
The PSNR values (dB) for the two error-free images, “Lenna” and “Fruits,” with bit rate = 0.4 bpp encoded by the original JPEG-2000 and the two proposed schemes

<table>
<thead>
<tr>
<th>Images</th>
<th>JPEG-2000</th>
<th>Proposed basic</th>
<th>Proposed enhanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lenna</td>
<td>31.11</td>
<td>30.20</td>
<td>29.42</td>
</tr>
<tr>
<td>Fruits</td>
<td>31.02</td>
<td>30.03</td>
<td>29.01</td>
</tr>
</tbody>
</table>
In this study, two error resilient coding schemes for wavelet-based image transmission based on data embedding and genetic algorithms are proposed. Using JPEG-2000, an image is decomposed into six wavelet levels (levels 0–5). For level 0, an error resilient coding scheme based on data embedding and side-match vector quantization is employed. For level 1, the proposed basic scheme uses the embedded important data for a corrupted code block to conceal the corrupted code block, whereas the proposed enhanced scheme uses a GA-based error concealment scheme to conceal the corrupted code block. For level 2, the embedded important data for a corrupted code block are used to conceal the corrupted code block. For levels 3–5, the wavelet coefficients of each corrupted code block are simply replaced by zeros. Based on the simulation results obtained in this study, the performances of the two proposed schemes are better than those of the three comparison approaches. The two proposed schemes can recover high-quality wavelet-based JPEG-2000 images from the corresponding corrupted images up to a bit error rate of 0.5%.

References


---

Table 2
The sizes (bytes) of the bitstreams for the two images, “Lenna” and “Fruits,” with bit rate = 0.4 bpp encoded by the original JPEG-2000 and the two proposed schemes

<table>
<thead>
<tr>
<th>Images</th>
<th>JPEG-2000</th>
<th>Proposed basic</th>
<th>Percentage of increment (basic)</th>
<th>Proposed enhanced</th>
<th>Percentage of increment (enhanced)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lenna</td>
<td>12224</td>
<td>12267</td>
<td>0.35%</td>
<td>12284</td>
<td>0.49%</td>
</tr>
<tr>
<td>Fruits</td>
<td>12215</td>
<td>12260</td>
<td>0.37%</td>
<td>12279</td>
<td>0.52%</td>
</tr>
</tbody>
</table>

Table 3
The simulation results, PSNR (dB), for the two images, “Lenna” and “Fruits,” with bit rate = 0.4 bpp, BER = 0.1%, and Nave = 3 by the proposed basic scheme and the proposed enhanced scheme without/with GA’s

<table>
<thead>
<tr>
<th>Images</th>
<th>Corrupted</th>
<th>Proposed basic without GA’s</th>
<th>Proposed enhanced without GA’s</th>
<th>Gains contrast to proposed basic</th>
<th>Proposed enhanced with GA’s</th>
<th>Gains contrast to proposed basic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lenna</td>
<td>14.21</td>
<td>22.41</td>
<td>23.53</td>
<td>1.12</td>
<td>24.49</td>
<td>2.08</td>
</tr>
<tr>
<td>Fruits</td>
<td>15.06</td>
<td>23.26</td>
<td>24.29</td>
<td>1.03</td>
<td>25.05</td>
<td>1.79</td>
</tr>
</tbody>
</table>

Table 4
The decoding times (seconds) for the two images, “Lenna” and “Fruits,” with bit rate = 0.4 bpp, BER = 0.1%, and Nave = 3 by the original JPEG-2000 and the two proposed schemes

<table>
<thead>
<tr>
<th>Images</th>
<th>JPEG-2000</th>
<th>Proposed basic</th>
<th>Proposed enhanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lenna</td>
<td>1.31</td>
<td>1.71</td>
<td>6.05</td>
</tr>
<tr>
<td>Fruits</td>
<td>1.34</td>
<td>1.69</td>
<td>6.48</td>
</tr>
</tbody>
</table>
current research interests include image/video processing, image/video communication, and multimedia database systems.

Taiwan. Since March 2005, he joined the Institute of Information Science, Academia Sinica, Taipei, Taiwan as a postdoctoral fellow. His information engineering in June 1997, July 1999, and January 2005, respectively, all from National Chung Cheng University, Chiayi, Taiwan. Li-Wei Kang


Jin-Jang Leou was born in Chiayi, Taiwan, Republic of China on October 25, 1956. He received the B.S. degree in communication engineering in 1979, the M.S. degree in communication engineering in 1981, and the Ph.D. degree in electronics in 1989, all from National Chiao Tung University, Hsinchu, Taiwan.

From 1981 to 1983, he served in the Chinese Army as a Communication Officer. From 1983 to 1984, he was at National Chiao Tung University as a lecturer. Since August 1989, he has been on the faculty of the Department of Computer Science and Information Engineering at National Chung Cheng University, Chiayi, Taiwan. His current research interests include image/video processing, image/video communication, pattern recognition, and computer vision.