Journal of Electronic Imaging 23(4), 043015 (Jul/Aug 2014)

Index compression for vector quantization using principal index-pattern coding algorithm

Yung-Chih Liu,[a,b] Gwo-Her Lee,[b] Jan-Ray Liao,[a,*] Li-Pin Chi,[b] and Jinshiuh Taur[a]
[a]National Chung Hsing University, Department of Electrical Engineering, Taichung 402, Taiwan
[b]National Chung-Shan Institute of Science and Technology, Aeronautical Systems Research Division, Taichung 407, Taiwan

Abstract. This paper presents an efficient lossless compression algorithm, the coding tree assignment scheme with principal index-pattern coding algorithm (CTAS-PIPCA), to encode image vector quantization (VQ). The coding model is designed on the basis of the schemes proposed in the previous works to further improve the coding performance of coding tree assignment scheme with improved search-order coding algorithm (CTAS-ISOC) by PIPCA. The PIPCA technique exploits the correlation of neighboring index pairs not in the original vector-quantized index map but in the principal index-pattern table which is generated from the two-dimensional histogram of index patterns in the training stage. The CTAS-PIPCA method is evaluated via extensive experiments. The searching matched index in the principal index-pattern table results in lower time complexity than CTAS-ISOC. The results also show that the proposed technique apparently reduces the bit rate as compared to the conventional VQ and other existing popular lossless index coding schemes, such as SOC and CTAS-ISOC. *© 2014 SPIE and IS&T* [DOI: 10.1117/1.JEI.23.4.043015]

Keywords: vector quantization; index coding; image compression.

Paper 13517 received Sep. 21, 2013; revised manuscript received May 27, 2014; accepted for publication Jun. 16, 2014; published online Jul. 31, 2014.

1 Introduction

Vector quantization (VQ) has been dominantly employed for image and audio compression systems.[1,2] VQ has research value due to its envisioned applications, such as voice compression,[3–5] data hiding,[6,7] digital watermarking,[8–10] etc.

The ordinary VQ encoder first partitions the input image into nonoverlapping equal-size blocks (vectors) whose sizes are equal to the code vector size of the codebook, and then compares each vector with a set of predetermined vectors (codebook) to find the closest code vector. Since each input vector is encoded with the index of the nearest code vector, the encoder transmits the index of the best matched code vector instead of the image block. Therefore, the image compression is achieved. At the decoder, the image can be reconstructed by a table look-up operation from the identical codebook according to the received indices.

In image VQ, by taking advantage of the high correlations existing among the neighboring blocks of an image, the bit rate can be further reduced if the interblock correlation in index domain is appropriately exploited. Some efficient coding schemes[11–17] are proposed to encode the output index map of ordinary VQ. Hsieh and Tsai propose search-order coding (SOC) method[11] to encode the output index map of VQ by searching the previous indices and sending the number of the passed different indices in the search path rather than the full binary word of the current index; therefore, the storage of the index is compressed. Hu and Chang propose a low complexity index compressed (LCIC) algorithm[12] that explores the interblock correlation existing among the neighboring indices and the mean-distance-ordered codebook. In Ref. 13, the index coding scheme (ICS) has proven that it can get better performance in reducing bit rates than LCIC. The ICS technique also achieves better compression efficiency than some variable-length coding (VLC) coding schemes, such as Huffman, arithmetic, and Lempel–Ziv–Welch (LZW). Somasundaram and Domnic design an adaptive index coding scheme[14] based on Refs. 12 and 13, and it gives a better performance in bit rate. In Ref. 15, Sun et al. modify the LCIC (Ref. 12) method called index searching algorithm with index associated list (ISAIAL) to encode VQ indices by utilizing a relative index table and to obtain better performances. Moreover, they propose the modified ISAIAL (Ref. 16) based on each row of the relative index table with different sizes. The modified ISAIAL with M2 exploits the interblock correlation in the index domain and decreases the bit rate apparently as compared to the conventional VQ, SOC, and LCIC with relative index tables. In Ref. 17, Taur et al. propose a coding tree assignment scheme with improved search-order coding algorithm (CTAS-ISOC) by the CTAS and index-pair coding scheme to get better performance in bit rate than others. In this paper, we propose the coding tree assignment scheme with principal index-pattern coding algorithm (CTAS-PIPCA) by extending frequencies of occurrence of index patterns in ISAIAL (Ref. 15). The new PIPCA scheme, instead of ISOC in CTAS-ISOC, reduces the time complexity by encoding the current index in a predefined principal index-pattern table that is computed from the two-dimensional (2-D) histogram of the adjacent index pairs. Experiments show the new proposed algorithm has better compression rate than other existing algorithms.

The rest of this paper is organized as follows. In Sec. 2, the CTAS-ISOC is introduced. Also, the proposed algorithms are explained in Sec. 3. In Sec. 4, the experimental results are given. Finally, the conclusion is given in Sec. 5.

*Address all correspondence to: Jan-Ray Liao, E-mail: jrliao@mail.nchu.edu.tw

I_{UL}	I_U	I_{UR}
I_L	I_C	

Fig. 1 Neighboring indices of the current index I_C.

2 CTAS-ISOC Algorithm

In the section, we will briefly introduce the CTAS-ISOC Algorithm in Ref. 17.

2.1 *Coding Tree Assignment Scheme*

CTAS is used to select a proper coding scheme according to the distribution of the surrounding indices of the current index. The notations and the positions of the current index and its neighboring indices are shown in Fig. 1. They are the current index (I_C), the left index (I_L), the upper index (I_U), the upper right index (I_{UR}), and the upper left index (I_{UL}). Since the distributions of the neighboring indices are different from each coding case, the conditions for assigning the short index code are different. The current index (I_C) can be encoded by the coding scheme using Table 1. According to the coding tree, the current index (I_C) can be encoded by one of the neighboring index code assignment (NICA), the coding scheme [the coding scheme represents the method which is used to implement the left-index (upper-index) pattern coding; in CTAS-ISOC, the coding scheme is ISOC. In CTAS-PIPCA, the coding scheme is PIPCA], or the original index coding algorithm. Note that the rules and code vectors

Table 1 Coding methods and code vectors for the current index I_C.

Rules	Conditions	Coding trees	Code vectors
1	Three or all of four indices (I_L, I_{UL}, I_U, I_{UR}) having the same index value called I_m	NICA	1 (if $I_C = I_m$)
		Coding scheme with the left index pair	01 + n-bit counter
		Coding scheme with the upper index pair	001 + n-bit counter
		Original coding	000 + original index
2	Four indices (I_L, I_{UL}, I_U, I_{UR}) consisting of two index pairs with different index values called I_{P1} and I_{P2} (Note that if the input index is at the last column, we assume $I_{UR} = I_U$)	NICA	00 (if $I_C = I_L$, $I_L = I_{P1}$)
			01 (if $I_C = I_{P2}$)
		Coding scheme with the left index pair	10 + n-bit counter
		Coding scheme with the upper index pair	110 + n-bit counter
		Original coding	111 + original index
3	Four indices (I_L, I_{UL}, I_U, I_{UR}) containing only one index pair called I_{1P} and the other two different indices belonging to $I_{others} = \{I_{O1}, I_{O2}\}$	NICA	00 (if $I_C = I_{1P}$)
			01 (if $I_C = I_{O1}$)
		Coding scheme with the left index pair	10 + n-bit counter
		Coding scheme with the upper index pair	110 + n-bit counter
		Original coding	111 + original index
4	Four indices (I_L, I_{UL}, I_U, I_{UR}) with different index values	NICA	0000 (if $I_C = I_L$)
			0001 (if $I_C = I_U$)
		Coding scheme with the left index pair	01 + n-bit counter
		Coding scheme with the upper index pair	001 + n-bit counter
		Original coding	1 + original index
5	First row (first column)	NICA	1 (if $I_C = I_L(I_U)$)
		Coding scheme with the left (upper) index pair	01 + n-bit counter
		Original coding	00 + original index

in Table 1 are *ad hoc*. The flag settings obtained by Shannon-Fano coding algorithm[18] in Table 1 are based on the statistical results of training images. Flag of Rules 1, 4, and 5 are adopted for distinguishable occurrences of patterns of coding methods. Due to undistinguishable occurrences of patterns of coding methods, the flag settings in Rules 2 and 3 of Table 1 are 2 or 3 bits. The code vectors in Rule 3 of Table 1 are different from CTAS-ISOC (Ref. 17) for coding schemes of the left-pair-search-order coding (LSOC) and the original index coding. According to our statistics, the current index and its neighbors have the same index value with a high probability. Therefore, in each coding tree, the NICA scheme first checks if the current index is the same as one of its neighbors. In order to improve coding efficiency, the code vector of PIPCA with the left index pair is modified as 2 bits plus an n-bit counter, and the code vector of the original coding scheme is modified as 3 bits plus the original index in Rule 3 of Table 1.

2.2 *Improved Search-Order Coding*

The ISOC algorithm includes the LSOC scheme and the upper-pair-search-order coding (USOC); the ISOC technique extends the idea of the search-order coding algorithm in Ref. 11 by matching indices pair by pair. Each index pair consists of two neighboring indices in the horizontal direction (left index pair) or the vertical direction (upper index pair).

The example in Fig. 2 illustrates how LSOC works in which a 2-bit search-order counter is used. The grids represent the quantized block of an image, and the number marked in the upper area of each grid is the corresponding quantized index. The gray grid at position (3, 4) denotes the input quantized index, and it is regarded as the search center. In principle, the search order is clockwise and outward from the search center. The thick dotted rectangle and the thin dotted rectangles indicate the matching pair and qualified search pairs, respectively. The mark "×" means that the search pair at this search point is not a qualified search pair or it is a repetition search pair. If a matched search pair can be found within a limited counter in the search path, the value of the counter will be encoded instead of the quantized index. Finally, the matching pair is equal to the search pair at (1, 3) and LSOC returns "11," and therefore the length of the input index is reduced.

3 Proposed Coding Algorithm

The proposed lossless coding scheme, CTAS-PIPCA, consists of one CTAS with two essential coding techniques. CTAS is designed to select a coding method based on the distributions of the neighboring indices. The first coding technique is NICA, which encodes the current index by the direct comparison with its neighbors and has been successfully applied to the VQ index coding.[17] The second coding technique is PIPCA, which encodes the current index in a predefined principal index-pattern table that is computed from the 2-D histogram of the adjacent index pair $[I_L\, I_C]$ or $[I_U\, I_C]$. The coding procedure is constructed by cascading a conventional VQ and the proposed coding unit as shown in Fig. 3.

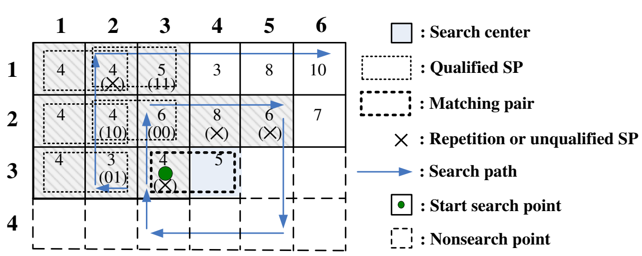

Fig. 2 Example for illustrating the left-pair-search-order coding algorithm.

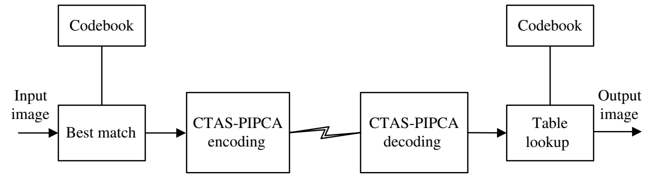

Fig. 3 The proposed system.

3.1 *Principal Index-Pattern Coding Algorithm*

Since a small block size is usually used in VQ and similar patterns may appear in different shapes, the index pair also has a high probability to appear in the neighboring regions repeatedly. Thus, the index-pattern coding scheme is adopted to further encode the current index (I_C) when NICA failed. The proposed PIPCA scheme is developed to improve the computational complexity of ISOC (the index-pattern coding scheme in CTAS-ISOC).[17] It exploits candidate indices in the principal index-pattern table which stores the first several high occurrences of index patterns. Thus, the encoding work of searching a matched index is not in the index map but in the principal index-pattern table constructed in the training stage. The operations of searching the matched index of ISOC are conducted in the index map shown in Fig. 2. In the PIPCA technique, searching the candidate is performed in a previously principal index-pattern table which has limited number of candidate indices shown in Fig. 4. The experimental result shows that the operations of searching the matched index in the table are less than that in the index map. Thus, the proposed PIPCA scheme improves the coding efficiency. Since the coding algorithms

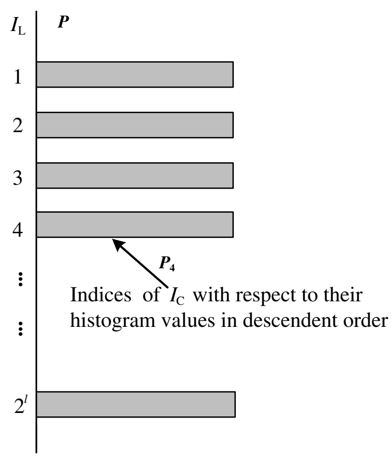

Fig. 4 The proposed principal index-pattern table ***P***.

for PIPCA with the left index pair and the upper index pair are the same, only the coding scheme for the left index pair is depicted in this paper.

The approach under consideration is based on the probabilistic arguments stemming from the statistical nature of the distribution of index patterns. In general, image VQ has a high correlation among the neighboring index blocks in the output index map. If two adjacent index blocks are designed as a pattern, the pattern can be found recurrently in the index map. Due to the raster scan order of VQ, every two adjacent index blocks, the left pair $[I_L I_C]$, are treated as the index patterns in our scheme. In ordinary VQ, if l bits are used to represent the index value, the codebook size is 2^l. For a given I_L, its possible index value of I_C in the left pair $[I_L\, I_{Ci}]$ is one of the values of $1, 2, \ldots, i, \ldots,$ and 2^l denoted as events of $I_{C_1}, I_{C_2}, \ldots, I_{C_i}, \ldots,$ and, $I_{C_{2^l}}$, respectively. Thus, for the given I_L, the finite set can be generated from training samples

$$\boldsymbol{W} = \{\boldsymbol{w}_i : i = 1,2,\ldots,2^N | \boldsymbol{w}_i \in R^2\}, \tag{1}$$

where $\boldsymbol{w}_i = [I_L\, I_{Ci}]$ and $N \le l$. 2^N is the total number of the index patterns found in training samples. According to the classical definition of probability, the conditional probability of I_{Ci} given I_L is defined as $[P(I_L) > 0]$

$$P(I_{Ci}|I_L) = \frac{P(I_{Ci} \cap I_L)}{P(I_L)} = \frac{n(I_{Ci} \cap I_L)}{n(I_L)} \propto n(I_{Ci} \cap I_L). \tag{2}$$

It is obvious that the probability of I_{Ci} given I_L is proportional to its joint probability. The more the number of the left pair $[I_L\, I_C]$ is found in training samples, the higher the conditional probability $P(I_{Ci}|I_L)$ is, and the higher probability that the matched index pattern can be found for the current input left pair. Therefore, the conditional probability $P(I_{Ci}|I_L)$ can be directly estimated from statistical quantities of training samples in the training stage. In our design, for the given current left pair $[I_L\, I_C]$, I_C is compared with each entry of a principal index-pattern table. Also, with the use of a n-bit counter which is less than or equal to N, the compression can be achieved. If the index pattern is found in the principal index-pattern table at the encoder, the current index I_C is assigned a flag code with the value of the n-bit counter and transmitted to the decoder. In the decoding unit, the index map is also reconstructed in a raster scan order. The search procedure is the same as that used in the encoding unit. Also, the identical principal index-pattern table is used to decode the receiving coding indices.

The essential technique of the proposed PIPCA scheme consists of the training stage and the coding stage. In the training stage, in order to get the statistical distribution of occurrences $n(I_{Ci} \cap I_L)$ of each respective bin I_L [cf. Eq. (2)], the 2-D histogram of the left index pair in the I_L and I_C space are first constructed by exploiting training samples. If l bits are used to represent the index value, a 2-D histogram $\mathbb{S}_{LC}$ of the left index pair is generated with the size of $2^l \times 2^l$. Figure 5 shows an example of the 2-D histogram of the left index pair with images Lena, F-16, and Peppers. If the 2-D histograms are rearranged by the frequencies of occurrences (histogram values) in descending order for each bin of I_L, the first several principal histogram values dominate the I_C space shown in Fig. 6. Subsequently, the index-pattern set for the left index value I_L is given as

$$\bar{\boldsymbol{P}}_{I_L} = [I_C^1\, I_C^2 \ldots I_C^i \ldots I_C^{2^l}], \tag{3}$$

where I_C^1, I_C^2, I_C^i, and $I_C^{2^l}$ are the bin values in the I_C space with respect to the histogram values in descending order. In case the input index value I_C^i is the same as the left neighboring index I_L, the current index will be encoded by NICA. Thus, $I_C^i = I_L$ is not required as a candidate of PIPCA. If a n-bit counter is adopted for PIPCA, the principal index-pattern set for the left index value I_L can be defined as

$$\boldsymbol{P}_{I_L} = [I_C^1\, I_C^2 \ldots I_C^{2^n}], \tag{4}$$

where $\boldsymbol{P}_{I_L}$ consists of bin values with respect to the first 2^n higher histogram values. Finally, the principal index-pattern table in the training stage is obtained as

$$\boldsymbol{P} = [\boldsymbol{P}_1\, \boldsymbol{P}_2 \ldots \boldsymbol{P}_{I_L} \ldots \boldsymbol{P}_{2^l}]^T, \tag{5}$$

where $\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_{I_L}$, and $\boldsymbol{P}_{2^l}$ are the principal index-pattern sets for $I_L = 1, 2, I_L$, and 2^l, respectively. Figure 4 illustrates the principal index-pattern table $\boldsymbol{P}$ where each subset is arranged by index bin value with respect to the histogram values in descending order.

In the coding stage, a left search index pattern $[I_L\, I_C]$ is first composed of the input current index I_C and its left adjacent index I_L. Then, the candidates in the subset $\boldsymbol{P}_{I_L}$ of the principal index-pattern table $\boldsymbol{P}$ can be located by the left index I_L. After that, comparisons between elements of $\boldsymbol{P}_{I_L}$

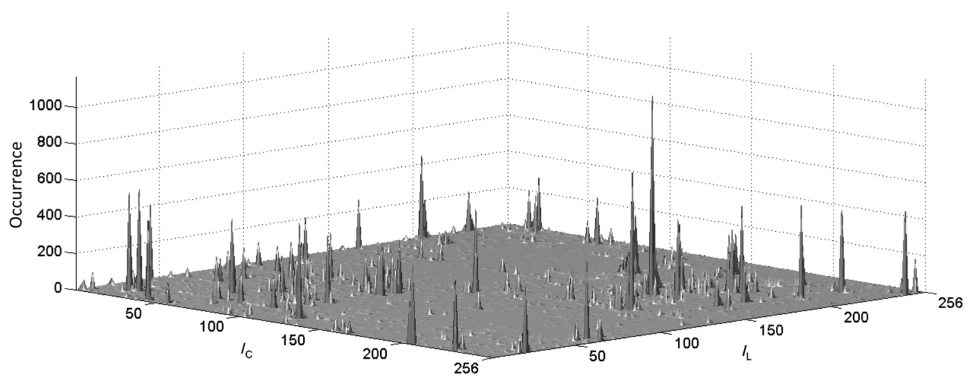

Fig. 5 Two-dimensional histogram of the left index pair from training images (Lena, F-16, and Peppers).

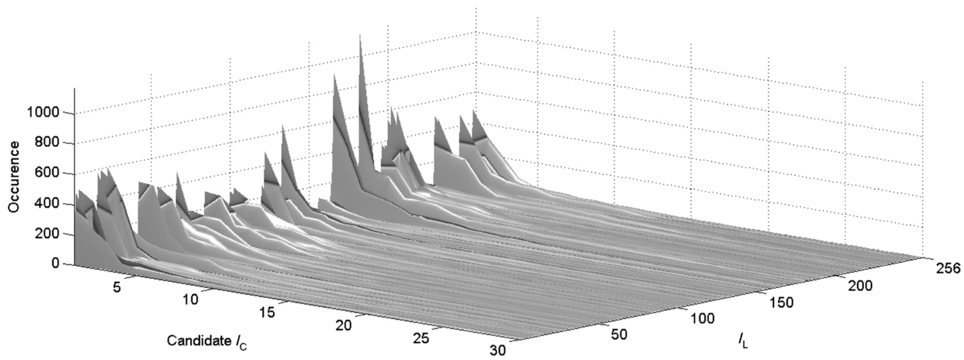

Fig. 6 Rearrangement of each bin of I_L with the first several principal histogram values in descending order.

and I_C are conducted to encode the current index. For an n-bit counter, elements of $\boldsymbol{P}_{I_L}$ are sequentially compared with $\boldsymbol{P}_{I_L}$ from the beginning. If the entry of $\boldsymbol{P}_{I_L}$ is not the same as I_C, the n-bit counter is increased by 1. Then, the next entry of $\boldsymbol{P}_{I_L}$ is extracted for searching the matched index. If the element is the same as I_C, the n-bit counter is increased by 1 and the current index I_C is encoded with a flag code plus the value of the n-bit counter (cf. Table 1). In order to enhance the coding efficiency, a repetition array $\boldsymbol{D}$ is adopted to temporarily store elements of $\boldsymbol{P}_{I_L}$ whose index values are not the same as I_C. In the coding process, if the entry of $\boldsymbol{P}_{I_L}$ is coincidentally the same as one element of $\boldsymbol{D}$, the n-bit counter need not be changed. Thus, it is more likely for the PIPCA encoding scheme to find the matched index without the overflow of the n-bit counter. In summary, the PIPCA coding scheme with left index pair is described in the following steps and its flow chart is depicted in Fig. 7.

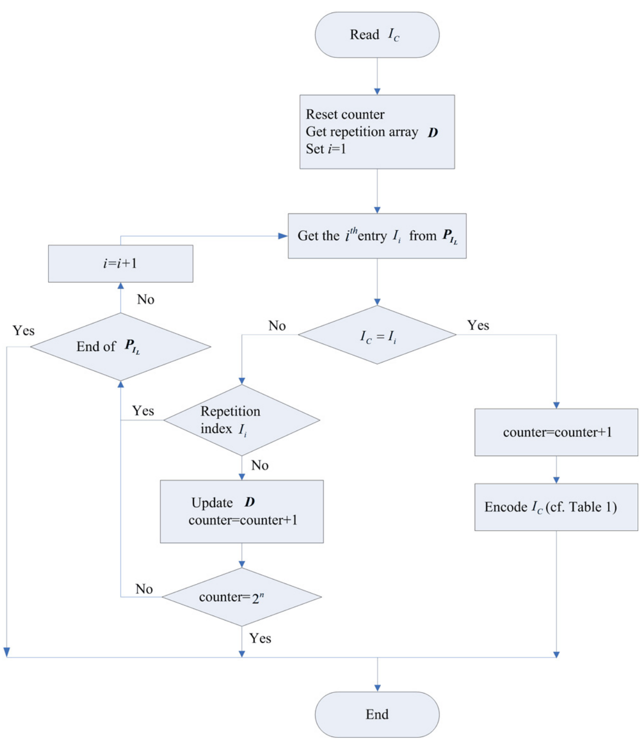

Fig. 7 The coding flow chart of the principal index-pattern coding algorithm scheme.

The PIPCA coding algorithm for the left index pair

1) Training stage

1. Set l bits to represent the index value, and let each I_C with its adjacent left index I_L be the left index pair $[I_L I_C]$.
2. Get the $2^l \times 2^l$ 2-D histogram of the left index pair in the I_L and I_C space from training samples.
3. Rearrange occurrences in descending order for each bin I_L, and construct the $2^l \times 2^n$ principal index-pattern table $\boldsymbol{P}$ with the first 2^n higher histogram values if the n-bit counter is adopted for PIPCA.

2) Coding stage

Step 1. Let $\boldsymbol{D}$ be the repetition array.
Step 2. Get an input quantized index I_C, and clear the n-bit counter.
Step 3. Locate $\boldsymbol{P}_{I_L}$ from $\boldsymbol{P}$ according to the left neighboring index I_L.
Step 4. Compare entries of $\boldsymbol{P}_{I_L}$ with the current index I_C and encode the matched index.

Repeat

(a) Read a candidate index I in $\boldsymbol{P}_{I_L}$ and compare it with I_C.
(b) If the n-bit counter overflows or $I = I_C$ is not found in $\boldsymbol{P}_{I_L}$, then go to step 5.
(c) If $I \neq I_C$.

(1) If I is not a repetition index value, increment the n-bit counter by 1, append I into the repetition array $\boldsymbol{D}$. If the n-bit counter is equal to 2^n, jump to Step 5, otherwise, go to (a).
(2) If I is a repetition index value, the n-bit counter need not be changed, and go to (a).

d) If $I = I_C$, increment the n-bit counter, encode the current index I_C with a flag code plus the value of the n-bit counter (cf. Table 1), and stop.

Until there is no other candidate index in $\boldsymbol{P}_{I_L}$.

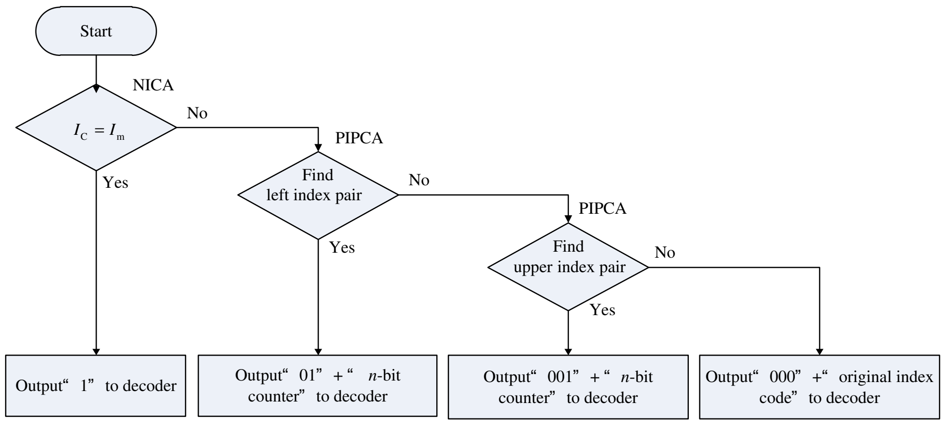

Fig. 8 Illustration of code vector generation for Rule 1 of Table 1.

Step 5. Adopt PIPCA with the upper index pair to further encode the current index I_C.

3.2 *Coding Tree Assignment Scheme with Principal Index-Pattern Coding Scheme*

When an output quantized index is sent to the proposed encoding unit, the relationships of its surrounding indices are first compared for coding tree selection. For example, if the surrounding indices (I_L, I_{UL}, I_U, I_{UR}) are with the same index value, Rule 1 of Table 1 is selected for generating code vector, and its corresponding illustration of code vector generation is shown in Fig. 8. If $I_C = I_m$ holds, I_C is encoded by NICA with a bit 1. Otherwise, the left index pair is applied to encode the current index by the PIPCA coding scheme. If the left index pair is found, I_C is encoded with the flag code 01 plus n-bit counter for the match. If there is not a match, the upper index pair is adopted for code vector generation. Again, if there is a match, I_C is encoded with the flag code 001 plus n-bit counter for the match. I_C is encoded with the flag code 000 plus the original index if all proposed coding techniques fail. The coding rules for the other conditions of the surrounding indices (I_L, I_{UL}, I_U, I_{UR}) of the current index I_C are shown from Rules 2 to 5 of Table 1.

4 Experimental Results and Discussion

In this section, two kinds of experiments are conducted to evaluate the proposed CTAS-PIPCA scheme. One is the comparison with VQ-based lossless index coding algorithms and the other is the comparison with JPEG2000 technique. The former comparison is used to illustrate the improvement of the proposed method over the existing VQ techniques and the later experiment is adopted to address the differences between the proposed scheme and the JPEG2000 technique.

The bit rate of VQ is the ratio of the bits required to represent the index of a vector to the vector dimension. For example, an image with the size of 512×512, if we use an image block with the size of 4×4, and codebooks with the sizes of 7, 8, and 9 bits to encode each image block, the bit rates of VQ are 0.4375, 0.5, and 0.5625 bpp, respectively. With the same evaluation on the CTAS-PIPCA scheme, the bit rate of the proposed system is given by

$$\mathrm{BR}_{\mathrm{VQ}} = \frac{1}{N_f}\sum_{i=1}^{5}[N_i^{\mathrm{NICA}} \times l_i^{\mathrm{NICA}} + N_i^{\mathrm{L}} \times (l_i^{\mathrm{L}} + n) + N_i^{\mathrm{U}} \times (l_i^{\mathrm{U}} + n) + N_i^{\mathrm{Ori}} \times (l_i^{\mathrm{Ori}} + p)], \tag{6}$$

where N_f is the number of pixels in the image. N_i^{NICA}, N_i^{L}, N_i^{U}, and N_i^{Ori} are the number of indices encoded by NICA, PIPCA with the left index pair, PIPCA with the upper index pair, and the original indices adopted by CTAS-PIPCA using the i'th rule of Table 1, respectively. l_i^{NICA}, l_i^{L}, l_i^{U}, and l_i^{Ori} are the flag bits for NICA, PIPCA with the left index pair, PIPCA with the upper index pair, and the original index coding using the i'th rule of Table 1, respectively. n is the number of bits used by the searching counter of PIPCA. p represents the number of bits needed for storing the original index of each code vector in the codebook. For comparison, the reduction of the average bit rate of a compression method with respect to the ordinary VQ is defined as

$$(b_{\mathrm{VQ}} - b_{\mathrm{CM}})/b_{\mathrm{VQ}} \times 100\%, \tag{7}$$

where b_{VQ} is the bit rate of the ordinary VQ and b_{CM} is the average bit rate of a compression method.

4.1 *Comparison with Lossless Index Coding Algorithms*

To evaluate the performance of the CTAS-PIPCA scheme, the USC-SIPI (Ref. 19) standard images are adopted for bit rate comparison. The images are monochrome with the size of 512×512 and brightness resolution of 8 bits. In this experiment, the search counter of the PIPCA n is 2, the images are vector-quantized with the blocks of 4×4 pixels, and the codebooks with the sizes of 7, 8, and 9 bits are used to represent the index value. In the training stage, three images (Lena, F-16, and Peppers) are trained by the popular Linde–Buzo–Gray (LBG) algorithm with a random initialization technique[20] to generate index maps with the codebook sizes of 128, 256, and 512. By treating the left pair $[I_L\, I_C]$ as the index pattern, three principal index-pattern tables with the sizes of $2^7 \times 2^2$, $2^8 \times 2^2$, and $2^9 \times 2^2$ are obtained by the use of codebook with the sizes of 7, 8, and 9 bits, respectively. Similarly, principal index-pattern tables for the upper pair $[I_U\, I_C]$ can be made.

In the coding stage, the USC-SIPI (Ref. 19) standard images are adopted for the evaluation of the CTAS-PIPCA scheme. There are eight test images: Couple, Man, House and Car, Cameraman, Bridge, Boat, Elaine, and Lake, which are labeled as I_{T1} to I_{T8}, respectively. It should be noted that the compression scheme of CTAS-PIPCA that does not adopt repetition candidates (the repetition array $\boldsymbol{D}$) is denoted as CTAS-PIPCA M1, and CTAS-PIPCA with the repetition candidates is denoted as CTAS-PIPCA M2. Also, the size of each row of principal index-pattern table should be larger than 2^n in CTAS-PIPCA M2. To demonstrate the effectiveness of the proposed two techniques, some popular existing lossless index compression algorithms, including SOC, modified ISAIAL, and CTAS-ISOC, are conducted on the same conditions. Tables 2–4 show the results of the proposed scheme and other methods. The average bit rates of CTAS-PIPCA M2 are 0.230, 0.293, and 0.367 bits/pixel for codebook sizes of 128, 256, and 512

Table 2 Bit rates (bits/pixel) of the test images using LBG with the random initialization (codebook size = 128, 0.4375 bits/pixel for VQ).

	I_{T1}	I_{T2}	I_{T3}	I_{T4}	I_{T5}	I_{T6}	I_{T7}	I_{T8}	Avg	Red (%)
SOC	0.284	0.293	0.297	0.246	0.339	0.291	0.257	0.271	0.285	35
Modified ISAIAL-M2	0.257	0.272	0.279	0.213	0.328	0.266	0.226	0.246	0.261	40
CTAS-ISOC	0.246	0.254	0.248	0.179	0.316	0.251	0.215	0.223	0.242	45
CTAS-PIPCA M1	0.238	0.244	0.245	0.177	0.306	0.244	0.206	0.222	0.235	46
CTAS-PIPCA M2	0.233	0.238	0.240	0.173	0.296	0.238	0.202	0.218	0.230	47
PSNR (dB)	26.33	25.12	33.33	28.62	23.22	26.72	28.27	25.59	—	—

Table 3 Bit rates (bits/pixel) of the test images using LBG with the random initialization (codebook size = 256, 0.5 bits/pixel for VQ).

	I_{T1}	I_{T2}	I_{T3}	I_{T4}	I_{T5}	I_{T6}	I_{T7}	I_{T8}	Avg	Red (%)
SOC	0.345	0.360	0.356	0.282	0.422	0.362	0.319	0.331	0.347	31
Modified ISAIAL-M2	0.329	0.349	0.347	0.261	0.426	0.345	0.296	0.316	0.334	33
CTAS-ISOC	0.311	0.324	0.309	0.227	0.396	0.325	0.275	0.292	0.308	38
CTAS-PIPCA M1	0.301	0.310	0.304	0.229	0.389	0.313	0.262	0.288	0.300	40
CTAS-PIPCA M2	0.295	0.303	0.298	0.225	0.380	0.306	0.254	0.282	0.293	41
PSNR (dB)	27.10	26.05	26.93	29.60	23.90	27.81	28.93	26.61	—	—

Table 4 Bit rates (bits/pixel) of the test images using LBG with the random initialization (codebook size = 512, 0.5625 bits/pixel for VQ).

	I_{T1}	I_{T2}	I_{T3}	I_{T4}	I_{T5}	I_{T6}	I_{T7}	I_{T8}	Avg	Red (%)
SOC	0.415	0.436	0.434	0.347	0.510	0.439	0.410	0.413	0.425	24
Modified ISAIAL-M2	0.406	0.429	0.433	0.334	0.527	0.430	0.399	0.416	0.422	25
CTAS-ISOC	0.381	0.400	0.396	0.300	0.485	0.399	0.364	0.376	0.387	31
CTAS-PIPCA M1	0.369	0.375	0.389	0.299	0.476	0.387	0.343	0.368	0.376	33
CTAS-PIPCA M2	0.361	0.366	0.384	0.290	0.468	0.379	0.331	0.360	0.367	35
PSNR(dB)	27.71	26.61	27.73	30.29	24.69	28.51	30.28	27.23	—	—

Table 5 The average number of searching indices required to find a matched index for CTAS-ISOC and CTAS-PIPCA.

	Codebook size		
Method	128	256	512
CTAS-ISOC	334.4	705.6	1347.7
CTAS-PIPCA M1	2.3	3.2	4.1
CTAS-PIPCA M2	12.8	18.6	24.6

code vectors, respectively. Obviously, the results show that the proposed method achieves lower bit rate than other index compression schemes for the three different codebooks.

The proposed PIPCA scheme is developed to improve the computational complexity of ISOC (the index-pattern coding scheme in CTAS-ISOC).[17] It exploits candidate indices in the principal index-pattern table which stores the first several high occurrences of index patterns. Thus, the encoding work of searching a matched index is not in the index map but in the principal index-pattern table constructed in the training stage. Table 5 shows the average number of indices required to find the matched index. It is obvious that the operations of searching the matched index in the principal index-pattern table are fewer than that in the index map. With CTAS-PIPCA M2, the performances of the average number of searching indices are less than 26 to 54 times than the previous work.

Figure 9 shows the relationships between the codebook size and the average percentage of bit rate reduction of different index compression algorithms over ordinary VQ. The proposed algorithm consists of CTAS, NICA, and PIPCA techniques, taking the neighboring indices and index patterns (left pair and upper pair) into account, and achieves better performance compared with other methods. The PIPCA technique searches the matched index in a principal index-pattern table that has stored the principal candidate indices and limited their number. Therefore, the operations of searching the matched index in the table are fewer than that in the index map, and the proposed PIPCA scheme improves the searching operations of ISOC.

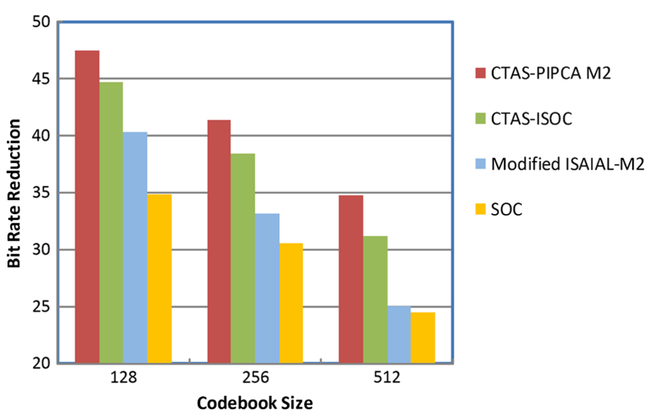

Fig. 9 Bit rate reduction over ordinary vector quantization (%).

In comparison with CTAS-PIPCA M1, CTAS-PIPCA M2 need not increment the n-bit counter when an index in the principal index-pattern table is found in the repetition array $\boldsymbol{D}$. It means CTAS-PIPCA M2 is more possible to find the matched index before the overflow of the n-bit counter. Since the memory of the repetition array is small, CTAS-PIPCA M2 outperforms CTAS-PIPCA M1 at a cost of extra memory. In summary, the proposed methods reach lower bit rate than CTAS-ISOC.

4.2 *Comparison with JPEG2000*

Since JPEG2000 offers significant new features and capabilities that reduce the bit rate of an image while increasing its peak signal-to-noise ratio (PSNR), it is an interesting question for us to know the difference between the proposed coding scheme and the JPEG2000 technique. To conduct the comparison, the free open source code (openjpeg-2.0.0-win32-x86.zip) of JPEG2000 is adopted for evaluation.[21] In this experiment, codebooks with the sizes of 128, 256, and 512 are generated from the popular LBG algorithm with the random initialization technique[19] in training stage, and the same eight USC-SIPI standard images are adopted for the evaluation of the PSNR quality at the same bit rate between the CTAS-PIPCA M2 scheme and the JPEG2000 technique. Comparisons of VQ, CTAS-PIPCA M2, and JPEG2000 are shown in Table 6. It can be observed that the PSNR qualities between the VQ and the proposed scheme are the same but the proposed CTAS-PIPCS M2 scheme reduces to 47%, 41%, and 35% average bit rates of the output index maps of the test VQ images for codebook sizes of 128, 256, and 512, respectively. The main reason is that the proposed CTAS-PIPCA scheme encodes the output index map of ordinary VQ to further reduce the bit rate, but does not influence the original PSNR quality. Once the codebook is generated, its PSNR quality of the output index map of ordinary VQ is determined.

Table 6 also shows that the PSNR quality of JPEG2000 is superior to that of CTAS-PIPCA at the equal bit rate obtained by CTAS-PIPCA M2. The average PSNR qualities of CTAS-PIPCA M2 are 26.22, 27.12, and 27.88 dB for codebook sizes of 128, 256, and 512, respectively, but the average PSNR qualities of JPEG2000 reach 28.81, 29.76, and 30.74 dB for codebook sizes of 128, 256, and 512, respectively. Obviously, the results show that the JPEG2000 scheme outperforms the proposed method in the PSNR quality for the test images. Since the CTAS-PIPCA M2 system is constructed by cascading a conventional VQ and the proposed coding unit that is a simple encoder in comparison with JPEG2000, the proposed coding unit just encodes the output index map of ordinary VQ to reduce the bit rate without influencing the original PSNR quality. In the proposed system, the PSNR quality is determined by the training method in training stage. If the classes of training vectors of images are trained properly, the higher PSNR quality of the index map of VQ can be achieved. Table 7 shows that the codebooks with the sizes of 128, 256, and 512 are obtained from the LBG algorithm with the splitting method.[22] It can be observed that the PSNR quality of the output index map of ordinary VQ is improved. In our future works, the performance of the higher PSNR quality and lower bit rate will be studied by cascading a variant VQ (Refs. 23–27) and the proposed coding unit.

Table 6 Performance of VQ, CTAS-PIPCA M2, and JPEG2000 (codebooks generated by LBG with the random initialization).

Codebook size		128			256			512		
Images		VQ	CTAS-PIPCA M2	JPEG2000	VQ	CTAS-PIPCA M2	JPEG2000	VQ	CTAS-PIPCA M2	JPEG2000
I_{T1}	Bit rate	0.438	0.233	0.233	0.500	0.295	0.295	0.563	0.361	0.361
	PSNR	26.33	26.33	28.62	27.10	27.10	29.63	27.71	27.71	30.52
I_{T2}	Bit rate	0.438	0.238	0.238	0.500	0.303	0.303	0.563	0.366	0.366
	PSNR	25.12	25.12	27.22	26.05	26.05	27.97	26.61	26.61	29.03
I_{T3}	Bit rate	0.438	0.240	0.240	0.500	0.299	0.299	0.563	0.384	0.384
	PSNR	25.88	25.88	28.87	26.93	26.93	29.91	27.73	27.73	31.06
I_{T4}	Bit rate	0.438	0.173	0.173	0.500	0.225	0.225	0.563	0.290	0.290
	PSNR	28.62	28.62	32.82	29.60	29.60	34.59	30.29	30.29	36.10
I_{T5}	Bit rate	0.438	0.296	0.296	0.500	0.380	0.380	0.563	0.469	0.469
	PSNR	23.22	23.22	24.47	23.90	23.90	25.01	24.69	24.69	25.62
I_{T6}	Bit rate	0.438	0.238	0.238	0.500	0.306	0.306	0.563	0.379	0.379
	PSNR	26.72	26.72	29.25	27.81	27.81	30.52	28.51	28.51	31.48
I_{T7}	Bit rate	0.438	0.202	0.202	0.500	0.254	0.254	0.563	0.331	0.331
	PSNR	28.27	28.27	31.56	28.93	28.93	31.91	30.28	30.28	32.39
I_{T8}	Bit rate	0.438	0.218	0.218	0.500	0.282	0.282	0.563	0.360	0.360
	PSNR	25.59	25.59	27.63	26.61	26.61	28.54	27.23	27.23	29.74
Avg	Bit rate	0.438	0.230	0.230	0.500	0.293	0.293	0.563	0.368	0.368
	PSNR	26.22	26.22	28.81	27.12	27.12	29.76	27.88	27.88	30.74

Table 7 The PSNR quality of the CTAS-PIPCA obtained by using LBG with the splitting method.

	Images							
Codebook size	I_{T1}	I_{T2}	I_{T3}	I_{T4}	I_{T5}	I_{T6}	I_{T7}	I_{T8}
128	29.59	27.00	25.99	26.84	29.45	24.12	27.34	29.58
256	30.36	27.67	26.64	27.49	30.31	24.65	27.98	30.29
512	31.08	28.29	27.23	28.22	31.14	25.21	28.61	30.87

5 Conclusions

In this paper, we present the CTAS-PIPCA algorithm. It is an efficient online lossless coding algorithm for the quantized indices of monochrome images. The experimental results show that the CTAS-PIPCA outperforms other schemes such as SOC, modified ISAIAL, and CTAS-ISOC in reducing the bit rate. The method can be applied to the VQ system to improve the coding efficiency of the basic image VQ and does not introduce any extra distortion. Due to the low computational complexity and small memory requirements of CTAS-PIPCA, the proposed algorithm is suitable for real-time implementation.

References

1. R. M. Gray and A. Gersho, *Vector Quantization and Signal Compression*, Kluwer, Boston, MA, USA (1992).
2. N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: a review," *IEEE Trans. Commun.* **36**(8), 957–971 (1998).
3. U. Haskar and K. Swaminathan, "Low bit-rate voice compression based on frequency domain interpolative techniques," *IEEE Trans. Audio Speech Lang. Process.* **14**(2), 558–576 (2006).
4. C. H. Lee, S. K. Jung, and H. G. Kang, "Applying a speaker-dependent speech compression technique to concatenative TTS synthesizers," *IEEE Trans. Audio Speech Lang. Process.* **15**(2), 632–640 (2007).
5. S. Kandadai and C. D. Creusere, "Scalable audio compression at low bitrates," *IEEE Trans. Audio Speech Lang. Process.* **16**(5), 969–979 (2008).
6. C. C. Chang, W. Tai, and C. C. Lin, "A reversible data hiding scheme based on side match vector quantization," *IEEE Trans. Circuits Syst. Video Technol.* **16**(10), 1301–1308 (2006).
7. P. Tsai, "Histogram-based reversible data hiding for vector quantisation-compressed images," *IET Image Process.* **3**(2), 100–114 (2009).
8. C. Lin, J. S. Pan, and B. Y. Liao, "Robust VQ-based digital image watermarking for mobile wireless channel," in *ICSMC '06 IEEE Int. Conf. on Systems, Man and Cybernetic*, Vol. 3, pp. 2380–2384, IEEE, Taipei, Taiwan (2006).
9. F. H. Wang, L. C. Jain, and J. S. Pan, "VQ-based watermarking scheme with genetic codebook partition," *J. Network Comput. Appl.* **30**(1), 4–23 (2007).
10. G. Wu, E. H. Yang, and D. K. He, "Joint watermarking and compression for Gaussian and Laplacian sources using uniform vector quantization," in *ICASSP '09. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 1493–1496, IEEE, Taipei, Taiwan (2009).
11. C. H. Hsieh and J. C. Tsai, "Lossless compression of VQ index with search-order coding," *IEEE Trans. Image Process.* **5**(11), 1579–1582 (1996).
12. Y. Hu and C. C. Chang, "Low complexity index-compressed vector quantization for image compression," *IEEE Trans. Consum. Electron.* **45**(1), 219–224 (1999).
13. P. Y. Chen and C. T. Yu, "Lossless vector-quantised index coding design and Implementation," *IEE Proc. Circuits Devices Syst.* **152**(2), 109–117 (2005).
14. K. Somasundaram and S. Domnic, "Adaptive index coding scheme for VQ indices compression," in *ICCTA '07. Int. Conf. Theory and Applications*, pp. 639–644, IEEE, Kolkata (2007).
15. H. M. Sun et al., "Improvement of VQ index compression with relative index tables," in *ICCSIT '08. Int. Conf. Computer Science and Information Technology*, pp. 492–496, IEEE, Singapore (2008).
16. H. M. Sun et al., "Improving VQ index compression with modified relative index table," in *ACIS Int. Symp. on Cryptography and Network Security, Data Mining and Knowledge Discovery, E-Commerce & Its Applications and Embedded Systems (CDEE)*, pp. 221–225, IEEE, Qinhuangdao (2010).
17. J. S. Taur et al., "Vector quantization index compression based on a coding tree assignment scheme with improved search-order coding algorithms," *IET Image Proc.* **6**(4), 318–326 (2012).
18. R. M. Fano, "The transmission of information," Technical Report 65, M. I. T., Res. Lab. of Electronics, Cambridge, Massachusetts, USA (1949).
19. University of Southern California Signal and Image Processing Institute, "The USC-SIPI Image Database," http://sipi.usc.edu/database/.
20. Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.* **28**(1), 84–95 (1980).
21. Communications and Remote Sensing Lab (TELE), "Welcome to the OpenJPEG homepage," (3 July 2014) http://www.openjpeg.org/.
22. P. Franti, T. Kaukoranta, and O. Nevalainen, "On the splitting method for VQ codebook generation," *Opt. Eng.* **36**(11), 3043–3051 (1997).
23. P. Y. Chen and C. T. Yu, "New clustering algorithm for vector quantization using rotation of error vector," *Int. J. Comput. Sci. Inf. Secur.* **7**(3), 159–165 (2010).
24. S. Vimala, K. Kowsalya Devi, and M. Sathya, "Codebook generation for vector quantization using interpolations to compress gray scale images," *Int. J. Comput. Appl.* **42**(9), 14–19 (2012).
25. M. H. Al-Otum, "Qualitative and quantitative image quality assessment of vector quantization, JPEG, and JPEG2000 compressed images," *J. Electron. Imaging* **12**(3), 511–521 (2003).
26. S.-H. Honga et al., "Image interpolation using interpolative classified vector quantization," *Image Vision Comput.* **26**(2), 228–239 (2008).
27. A. D. Setiawan and A. B. Suksmono, "An indicator elimination method for side-match vector quantization," *J. Inf. Hiding Multimedia Signal Process.* **4**(4), 233–249 (2013).

Yung-Chih Liu received an MS degree in electrical engineering from Chung-Cheng Institute of Technology, Taoyuan, Taiwan, in 1997. He was awarded the PhD degree in the Department of Electrical Engineering at National Chung Hsing University, Taichung, Taiwan, in 2014. Since 2000, he has been an engineer in the Chung-Shan Institute of Science and Technology. His research interests include image processing, fuzzy logic systems, and computer vision.

Gwo-Her Lee received BS and MS degrees in electrical engineering from Chung-Cheng Institute of Technology, Taoyuan, Taiwan, in 1989 and 1992, respectively. He received a PhD degree in the Department of Electrical Engineering at National Chung Hsing University, Taichung, Taiwan, in 2006. Currently, he is a senior researcher at Chung-Shan Institute of Science and Technology. His research interests include flight simulation, 3-D graphics, neural networks, fuzzy logic systems, and machine learning.

Jan-Ray Liao received a BS degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1989 and MS and PhD degrees in electrical engineering from Stanford University, Palo Alto, California, USA, in 1993 and 1997, respectively. From 1997 to 2001, he was an assistant professor in the Department of Electrical Engineering, National Chung-Hsing University, Taichung, Taiwan. Since 2001, he has been an associate professor in the same department.

Li-Pin Chi received a BS degree from the Chung-Cheng Institute of Technology, Taoyuan, Taiwan, in 1991. He obtained an MS degree from the Department of Electrical Engineering of Da-Yeh University, Taiwan, in 1999. He was awarded the PhD degree in the Department of Electrical Engineering of Feng Chia University, Taichung, Taiwan, in 2009. Currently, he is working at Chung-Shan Institute of Science and Technology. His current research interests are in antenna design and microwave engineering.

Jinshiuh Taur received his BS and MS degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1987 and 1989, respectively, and his PhD degree in electrical engineering from Princeton University, United States, in 1993. He was a member of technical staff at Siemens Corporate Research, Inc. From 2001 to 2013, he was a professor at National Chung Hsing University in Taiwan. His primary research interests include neural networks, pattern recognition, and fuzzy logic systems.