Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **3**, May **2015**

Tree-Structured Vector Quantization with Flexible Multipath Searching Method Based on Side Match Prediction

Li Liu[1], An-Hong Wang[1,†], Chin-Chen Chang[2,3,*], Qiang Jin[1], Bin-Bin Xia[1]

[1]College of Electronic Information and Engineering,
Taiyuan University of Science and Technology, Taiyuan 030024, China.
†Correspondence Autor
wah_ty@163.com

[2,*]Department of Information Engineering and Computer Science,
Feng Chia University, Taichung 40724, Taiwan.
∗Correspondence Autor
alan3c@gmail.com

[3]Department of Computer Science and Information Engineering,
Asia University, Taichung 41354, Taiwan.

Received September, 2014; revised December, 2014

Abstract. *Although multipath tree-structured vector quantization (MP-TSVQ) has a higher probability of determining the closest codeword for each input vector than single-path TSVQ(SP-TSVQ), it lacks flexibility in some cases because of the fixed number of search paths. To overcome this drawback, we propose the side-match prediction TSVQ (SMP-TSVQ) to make the number of search paths adjustable. SMP-TSVQ uses the correlation of adjacent blocks, and the smoothness of the upper block and the left block relative to the current block to determine the number of search paths. For seed blocks that play an important role, 8-path is always used to guarantee the quality of the image. For residual blocks, there are three cases to be considered, i.e., 1) when the upper block and left block are both smooth, 2-path is used; 2) when either the upper block or the left block is smooth, 4-path can be used; 3) when neither block is smooth, 8-path must be used to ensure the quality of the image. Therefore, compared to the previous multipath search algorithm, SMP-TSVQ requires flexible search paths to reduce computational complexity and encoding time. Experimental results have proved the expected merits of the proposed scheme. Moreover, when the codebook sizes are 128, 256, 512 and 1024, the encoding time of SMP-TSVQ is only 34.9%, 20.3%, 13.9%, 6.7% of that of FSVQ, respectively.*

Keywords: Codebook tree, TSVQ; Multipath TSVQ, Path selection, The computational complexity

1. **Introduction.** With the development of information technology, a large number of multimedia files (e.g., audio, image, and video files) often must be stored and transmitted in many fields, including spaceflight, the military, weather forecasting and reporting, and medicine. To save storage space and improve transmission efficiency, encoding algorithms are used to remove redundant information in multimedia files, and the quality of recovered versions must be guaranteed. Vector quantization (VQ) [1, 2, 3, 4] is an extensively-used, effective lossy method in the area of image compression, and its main merits are its high

compression ratio and simple decoding procedures. A good codebook is generated from a training image by using the Linde-Buzo-Gray (LBG) algorithm [5] or the K-means algorithm [6]. The codebook that is generated is a finite set of representative codewords that are used in the encoding and decoding procedures. In the image-encoding procedure, first, the image is divided into several k-dimensional image vectors, and second, each image vector is quantized to the closest codeword in the codebook and is finially replaced by the index of its closest codeword. In image decoding, the received indices are used to determine the corresponding codewords in the codebook simply and quickly by using a look-up table. By filling the corresponding codewords in the original location, an image that is similar to the original image can be reconstructed.

In the VQ algorithm, a major problem is determining the closest codeword in the codebook for an input vector by an efficient search algorithm. The original search algorithm is the Full Search (FS) algorithm, which is uesd to calculate the distortion between the input vector and all codewords and find out the minimally-distorted codeword. However, when the codebook size is large or the dimensionality of the vector is large, the computational complexity and the time loss are both very large. Thus, many kinds of VQ algorithms have been proposed to resolve these two issuses. One approach is to efficiently construct a representative codebook in order to improve the quality of the image, and the other is to reduce the computational complexity of the search process in order to reduce the time loss. For the latter, the tree-structured VQ (TSVQ) [7, 8, 9, 10] is an efficient method. It uses a hierarchical binary tree to represent the codebook tree, in which each codeword on level l is either a leaf node as the final encoded codeword or an internal node with two sons in level $l+1$. In the encoding process, the encoder follows a path from the root node of the tree to the leaf node, and selects the nearest codeword for each node. Obviously, this single-path search TSVQ (SP-TSVQ) reduces the computational complexity because only two nodes in each level must be calculated. But it usually fails to determine the closest codeword, so the quality of the image is poor. To improve the quality of the image, the multipath TSVQ (MP-TSVQ) [8, 10, 11, 12, 13] often has been proposed by scholars. Different from the single-path TSVQ, MP-TSVQ follows m paths to search for the closest codeword, therefore, the probability of determining the closest codeword is improved. Cardinal [12] presented two design algorithms for TSVQ. For the first algorithm, the number of paths was fixed, and the encoding was similar to the M-algorithm for delayed decision coders; for the second algorithm, the paths are chosen adaptively by the $(1+\varepsilon)$ nearest-neighbor rule in each node. These methods obtained near full search performances at a fraction of the complexity cost. Chang et al. [8] proposed a dynamic path TSVQ (DP-TSVQ), that used a critical function to judge whether the number of search paths were altered. And the experimental results showed that DP-TSVQ is always faster than the multipath TSVQ for images that have the same quality. The closest-coupled TSVQ (CC-TSVQ), proposed by Chang et al. [11], also is an efficient multipath searching method in TSVQ. This method used the closest-neighboring pointers to enlarge the search range of each search path and to guide all search paths into more appropriate codewords, thereby improving the quality of the image.

In addition, side-match VQ [14, 15, 16, 17] also is a well-known algorithm that enhances the quality of VQ and improves the compression ratio. It uses the previously-coded blocks, which are the block that is above and the block that is to the left of the current block to predict the current block. Inspired by this side-match idea, in this paper, we proposed a new method called the side-match prediction TSVQ (SMP-TSVQ) to improve the image quality and the encoding time. In this method, we first designed three different path selections, i.e., 2-path, 4-path and 8-path, respectively. Then we used side-match prediction to judge which path is applied to the current block. That is to say, 8-path

is always used for the seed blocks (i.e., the blocks in the top row and leftmost column of the image). However, path selection for residual blocks must be judged based on the smoothness of the current blocks upper block and left block. When the upper block and left block are both smooth, the 2-path is used. And when either the upper block or the left block is smooth, we used 4-path. If neither of the blocks is smooth, we must use 8-path to ensure the images quality. Therefore, compared to the previous multipath search algorithm, SMP-TSVQ requires flexible search paths to improve the images quality and reduce the encoding time. Our experimental results proved the expected merits of the proposed scheme.

2. **Background.**

2.1. **TSVQ.** TSVQ is an efficient method for reducing the computational complexity of searching for a closest codeword by constructing a binary codebook tree. Here, a codebook tree T was trained from a set of training images and was used in the encoding and decoding processes. Assume that the number of nodes in the l^{th} level of tree T is 2^l and that the level of the root node is 0. Each image in a set of training images is first divided into non-overlapping blocks of size $B \times B$, and each block can be regarded as a k-dimensional (where $k = B^2$) vector. All of the k-dimensional vectors constitute a training sequence (TS) set. Next, we used the K-means algorithm with the splitting method to build codebook tree T. The steps in using the algorithm are as follows.

Step 1. Calculate the centroid of the TS set as the initial codeword W_0 (i.e., the root node).

Step 2. Split W_0 into two vectors $W_0^{(1)} = W_0(1+\varepsilon)$ and $W_0^{(2)} = W_0(1-\varepsilon)$, where $(1 \pm \varepsilon)$ is the disturbing factor.

Step 3. Divide the TS set into two cells TS_1 and TS_2 ,

$$TS_1 = \{X | d(X, W_0^{(1)}) \leq d(X, W_0^{(2)}), X \in TS)\}, \tag{1}$$

$$TS_2 = \{X | d(X, W_0^{(2)}) \leq d(X, W_0^{(1)}), X \in TS)\}, \tag{2}$$

where the distortion $d(X, W_0^{(i)})$ between each vector X in TS set and $W_0^{(i)}$ is defined as

$$d(X, W_0^{(i)}) = \|X - W_0^{(i)}\|^2, i = 1 or 2. \tag{3}$$

Step 4. Calculate the centroids of cells TS_1 and TS_2 as two codewords W_1 and W_2. That is to say, W_1 and W_2 are the left child and the right child of the root node W_0 , respectively.

Step 5. Repeat Steps 2 to 4 to deal with each node of the l^{th} level until the number of leaf nodes is equal to the desired number of codewords.

Then, the codebook tree T is generated. It is well known that SP-TSVQ only compares two nodes in every level and that m-path TSVQ compares $2m$ nodes in every level. But the amount of storage of the codebook increases because all nodes must be stored, including leaf nodes and internal nodes.

2.2. **MP-TSVQ.** Although SP-TSVQ greatly reduces the computational complexity, it usually fails to determine the closest codeword. Increasing the number of search paths is an effective way to solve this problem. That is to say, the multipath searching algorithm is able to match the closer codeword more effectively than SP-TSVQ in the codebook tree T. Obviously, the computational complexity increases as the number of search paths increases. Yet, at the same time, the quality of the image gets better and better. In MP-TSVQ, assume that m is the number of search paths and that S is the set of m nodes currently being considered. Initially, m nodes on level $log_2 m$ of the codebook T

are selected into the set S. Let W_i^{left} and W_i^{right} denote the left child and right child, respectively, of internal node W_i. This multipath searching algorithm, which matches the closest leaf node P in the codebook tree T for an input vector X, is given as follows:

Step 1. Consider each internal node W_i in S.

Step 2. Calculate the distortion $d(X, W_i^{left})$ between X and W_i^{left}, and the distortion $d(X, W_i^{right})$ between X and W_i^{right}.

Step 3. If $d(X, W_i^{left}) < d(X, W_i^{right})$, select W_i^{left} instead of W_i into the set S; otherwise, select W_i^{right} instead of W_i.

Step 4. Repeat Steps 2 and 3 until all nodes in S are leaf nodes.

Step 5. Determine the closest codeword of input vector X by comparing all nodes in set S.

3. **Proposed Scheme.** Based on the above description, we know that increasing the number of search paths can improve the probability of determining the closest codeword. However, the previous MP-TSVQ uses a fixed number of search paths. In other words, this algorithm always processes m nodes on each level when the level is greater than or equal to $log_2 m$. In practice, this is unnecessary and time-consuming. We observed that there is a certain correlation between adjacent image blocks, meaning that the current coded block can be predicted by its adjacent blocks. So, based on this idea, we proposed a new method in this paper, called the side-match prediction TSVQ (SMP-TSVQ). This algorithm takes advange of the correlation of neighboring blocks to determine the number of search paths m for every coded block. That is, the number of search paths m for every coded block changes according to the state of the adjacent blocks.

In our scheme, both the sender and the receiver have the same codebook tree T. The original uncompressed image I is firstly divided into non-overlapping $B \times B$ blocks. For simplicity, we assumed that image I can be divided by B with no remainder. In Figure 1(a), the blue regions represent the blocks in the leftmost and topmost of image I, and they are called seed blocks; the white regions represent the residual blocks. These seed blocks are important because they can be used to predict the residual blocks. We designated the current coded block as $B_{x,y}$ and its left and upper blocks as $B_{x,y-1}$ and $B_{x-1,y}$, respectively, as shown in Figure 1(b). The difference σ between the maximum pixel and the minimum pixel in each of the image blocks is calcualted to characterize the smoothness of the image blocks. Here, we need to set an important, pre-determined threshold Th. When $\sigma \leq Th$, the image block is smooth; otherwise, this image block is not smooth. To improve the quality of the reconstructed image and limit the computational complexity, the value of searching paths m should not be too high. So, we designed three different path selections, i.e., 2-path, 4-path, and 8-path. Usually, 8-path is used for seed blocks because of their importance. Nevertheless, for residual blocks, there are three different cases to be considered in SMP-TSVQ. Here, let σ_l and σ_u denote the difference of the left block and the upper block for the current block.

Case 1: If the values of σ_l and σ_u are smaller than or equal to a pre-determined threshold Th, (i.e., both the upper block and the left block for the current block are smooth), it implies that the currrent residual block $B_{x,y}$ is located in a relatively smooth region and that it has a high correlation with its neighboring blocks. So, we use the 2-path searching algorithm.

Case 2: If the values of σ_l and σ_u are greater than a pre-determined threshold Th, (i.e., the upper block and the left block for the current block are not smooth), it implies that the current block $B_{x,y}$ is located in a relatively complex region and that it has low correlation with its neighboring blocks. So, we use the 8-path searching algorithm.

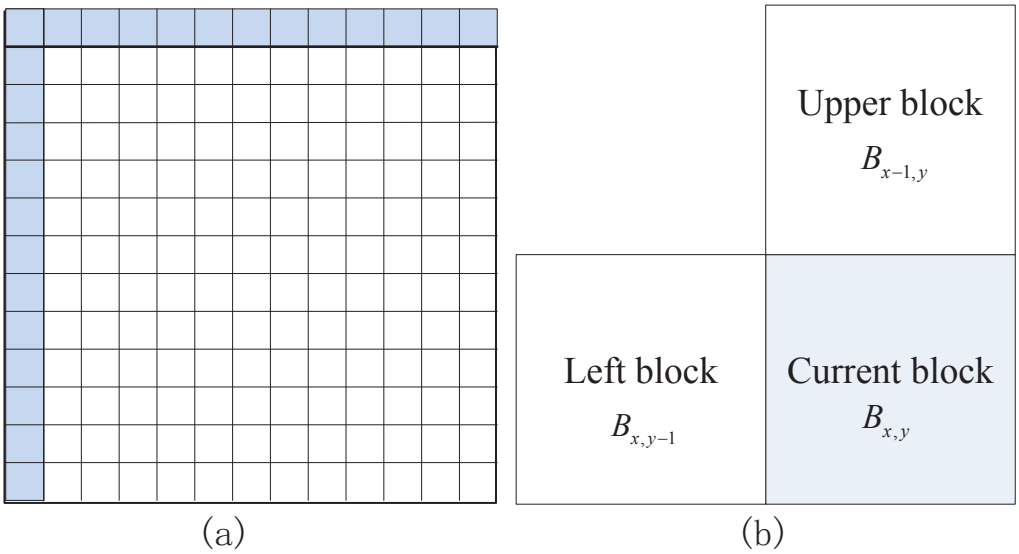

FIGURE 1. (a) Diagram of image blocks; (b) Location illustration of upper block and left block

Case 3: One of the values σ_l and σ_u is smaller than or equal to the pre-determined threshold Th, and the other is greater than Th. In this case, we developed a compromise for selecting the 4-path searching algorithm.

The major difference between MP-TSVQ and the SMP-TSVQ is that the former uses a fixed number of searching paths, while the latter flexibly switches among three different path-selection modes. In the process of image compression, switching modes can reduce the number of processing nodes and consequently reduce the computational complexity and the time loss.

4. **Experimental Results.** Six 8-bit, 512×512 gray-level test images (Lena, Frog, Baboon, Peppers, Boat and Zelda) are shown in Figure 2. The sizes of the image blocks were 4×4, i.e., each block was a 16-dimensional vector. We used four of the test images, i.e., "Lena," "Frog," "Baboon" and "Peppers" as training images to generate the codebook tree T. "Boat" and "Zelda" were not training data, and they were used to evaluate the performance of our scheme. It is worth noting that the codebook trees T were always identical in the algorithms that we compared. The quality of the rebuilt image is measured by the peak signal-to-noise ratio (PSNR) in our experiments, which is defined as

$$PSNR = 10log_{10}(\frac{255^2}{MSE})dB, \tag{4}$$

where MSE is the mean square error for an $m \times n$ image between the original images and the rebuilt image, and x_{ij} and y_{ij} are the pixels of the original image and the rebuilt image, respectively.

$$MSE = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (x_{ij} - y_{ij})^2. \tag{5}$$

In our scheme, the pre-determined threshold Th also is important because the choice of threshold Th determines the number of smooth blocks and, affects the computational complexity, encoding time, and image quality. Figure 3 shows the average experimental results for six test images when the thresholds were different. In this experiment, we considered four kinds of codebook sizes (128, 256, 512, 1024). Figure 3 shows that the

FIGURE 2. Six test images, "Lena," "Frog," "Baboon," "Peppers," "Boat," and "Zelda"

quality of the rebuilt image was reduced as the threshold Th increased and as the encoding time (the value shown within brackets in Figure 3) decreased. That is because the number of smooth blocks also increased correspondingly when the threshold Th increased, and more and more blocks used the 2-path or 4-path searching algorithms. Table 1 shows that the number of blocks that used the different path-selection modes for the different thresholds values when the codebook size is 256. We know from these data that the number of blocks that used the 2-path mode and the 8-path mode obviously changed, but the number of blocks using the 4-path mode showed little change. Through the statistical analysis of the six test images, we also found that the difference was concentrated mainly in the range of 10 to 15. Figure 3 also shows that the quality of the image and encoding time had obvious decreases within 10 to 15 and that the decreases leveled off as the threshold Th increased. Similarly, the data in Table 1 also verified this observation.
Table 2 compares the average results for the six test images in the different schemes, and there are four different codebook sizes. In MP-TSVQ, we set paths m to 2, 4, and 8, i.e., 2-path TSVQ, 4-path TSVQ, and 8-path TSVQ. In SMP-TSVQ, the threshold Th was set to 20. Table 2 shows that the encoding time of SMP-TSVQ was slightly larger than that of 2-path TSVQ, close to that of 4-path TSVQ, and less than that of 8-path TSVQ. The quality of the image of SMP-TSVQ fell between 2-path TSVQ and 8-path TSVQ, and it was slightly larger than that of 4-path TSVQ. So, SMP-TSVQ had greater flexibility than the algorithm that had a fixed number of search paths. In FSVQ, the codebook of VQ is a set of codewords located on the leaf nodes in the codebook tree T. Although the images quality of FSVQ was slightly superior to that of SMP-TSVQ, its encoding time was three times greater than that of SMP-TSVQ when the codebook size was 128. Thus, the larger the codebook size, the more obvious advantages SMP-TSVQ provides with respect to the

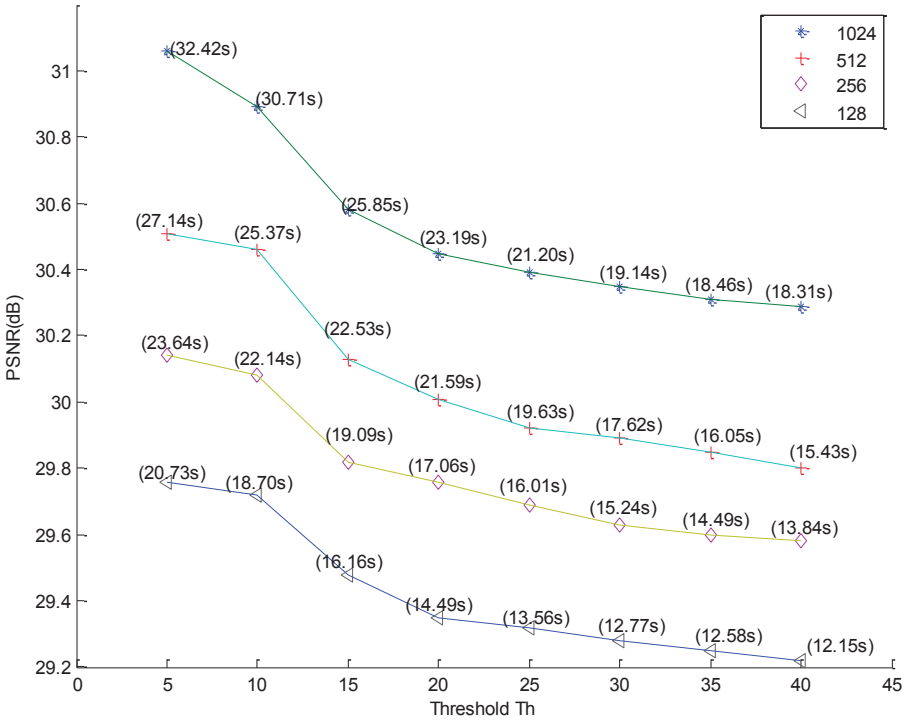

FIGURE 3. Average experimental result for six test images when the thresholds were different

TABLE 1. Number of blocks that used different path selection modes when thresholds were different

Image	Path Selection Mode	Threshold Th				
		10	15	20	25	30
Lena	2-path	3498	6719	8499	9723	10672
	4-path	2430	2773	2886	2854	2654
	8-path	10456	6892	4999	3807	3058
Peppers	2-path	1598	5292	7943	9614	10708
	4-path	2325	3802	3817	3540	3215
	8-path	12461	7290	4624	3230	2461
Zelda	2-path	2110	7061	10242	11695	12508
	4-path	2704	3504	2613	2310	2136
	8-path	11570	5819	3529	2379	1740

encoding time. When the codebook size was 256, the encoding time of SMP-TSVQ was only 19.8% of that of FSVQ, but the images quality of SMP-TSVQ was 2.33 dB lower than that of FSVQ. And when the codebook sizes were 512 and 1024, the encoding time of SMP-TSVQ were only 13.9% and 6.7% of that of FSVQ, respectively.

5. **Conclusions.** In this paper, a new SMP-TSVQ was proposed. The scheme is based on the idea of side-match prediction to judge which path selection mode should be applied to the current block. That is to say, judging by the smoothness of the upper block and the left block, the path selection of the current block can be flexibly switched among

TABLE 2. Comparison of different schemes with different codebook sizes for six test images

Codebook Size	2-path TSVQ		4-path TSVQ		8-path TSVQ		SMP-TSVQ		FSVQ	
	Time(s)	$PSNR$	Time(s)	$PSNR$	Time(s)	$PSNR$	Time(s)	$PSNR$	Time(s)	$PSNR$
128	8.93	28.84	14.27	29.17	22.80	29.65	14.49	29.35	41.46	31.15
256	10.38	29.09	17.58	29.38	27.69	30.12	17.06	29.76	86.11	32.09
512	11.31	29.23	19.25	29.45	33.10	30.41	21.59	30.01	154.46	33.67
1024	12.86	29.51	21.82	29.76	37.81	30.57	23.19	30.45	341.67	33.94

2-path, 4-path and 8-path to reduce the computational complexity and time loss. The experimental results also demonstrated that SMP-TSVQ had greater flexibility than the algorithms that have a fixed number of search paths. In addition, when the codebook sizes were 128, 256, 512, and 1024, the encoding times of SMP-TSVQ were only 34.9%, 20.3%, 13.9%, and 6.7% of that of FSVQ, respectively. So SMP-TSVQ can compress an image faster than FSVQ.

Acknowledgment. We would like acknowledge the supports from the National Natural Science Foundation of China (No. 61073142, No. 61272262, No. 61210006), International Cooperative Program of Shanxi Province (No. 2011081055), The Shanxi Provincial Foundation for Leaders of Disciplines in Science (20111022), Shanxi province Talent Introduction and Development Fund (2011), Shanxi Provincial Natural Science Foundation (2012011014-3), YSTRF of TYUST (No.20123004).

REFERENCES

[1] N. M. Nasrabadi and R. A. King, Image coding using vector quantization: A review, *IEEE Trans. on Communications*, vol. 36, no. 8, pp. 957-971, 1988.

[2] A. Gersho and R. M. Gray, *Vector quantization and signal compression*, Springer, 1992.

[3] H. C. Huang, S. C. Chu, J. S. Pan and Z. M. Lu, A tabu search based maximum descent algorithm for VQ codebook design, *Journal of Information Science and Engineering*, vol. 17, no. 5, pp. 753-762, 2001.

[4] H. C. Huang, J. S. Pan, Z. M. Lu, Z. H. Sun and H. M. Hang, Vector quantization based on genetic simulated annealing, *Signal Processing*, vol. 87, no. 7, pp. 1513-1523, 2001.

[5] Y. Linde, A. Buzo, and R. M. Gray, An algorithm for vector quantizer design, *IEEE Trans. on commucations*, vol. 28, no. 1, pp. 84-95, 1980.

[6] J. T. Tou and R. C. Gonzalez, Pattern recognition principles, *Applied Mathematics and Computation, Reading, MA: Addison-Wesley*, 1974.

[7] A. Buzo, Jr A. Gray and R. M. Gray, Speech coding based upon vector quantization, *IEEE Trans. on Acoustics Speech and Signal Processing*, vol.28, no. 5, pp. 562-574, 1980.

[8] C. C. Chang, F. C. Shiue, and T. S. Chen, Tree structured vector quantization with dynamic path search, *International Workshops on Parallel Processing*, pp. 536-541, 1999

[9] C. C. Chang, Y. C. Li, and J. B. Yeh, Fast codebook search algorithms based on tree-structured vector quantization, *Pattern Recognition Letters*, vol. 27, no. 10, pp. 1077-1086, 2006.

[10] R. F. Chang, W. T. Chen, and J. S. Wang, Image sequence coding using adaptive tree-structured vector quantisation with multipath searching, *Communications, Speech and Vision, IEE Proceedings I*, vol. 139, no. 1, pp. 9-14, 1992.

[11] C. C. Chang and T. S. Chen, New tree-structured vector quantization with closest-coupled multipath searching method, *Optical Engineering*, vol. 36, no. 6, pp. 1713-1720, 1997.

[12] J. Cardinal, Multipath tree-structured vector quantizers, *Proc. European Signal Processing Conference*, pp. 817-820, 2000.

[13] Y. H. Yu, C. C. Chang and Y. C. Hu, A genetic-based adaptive threshold selection method for dynamic path tree structured vector quantization, *Image and Vision Computing*, vol. 26, no. 6, pp. 597-609, 2005.

[14] T. Kim, Side match and overlap match vector quantizers for images, *IEEE Trans. on Image Processing*, vol. 1, no. 2, pp. 170-185, 1992.
[15] F. H. Hsu, M. H. Wu, and S. J. Wang, Reversible data hiding using side-match predictions on steganogra phic images, *Multimedia Tools and Applications*, vol. 67, no. 3, pp. 571-591, 2013.
[16] S. B. Yang, Side-match tree-structured vector quantiser for image progressive coding, *IEE Proceedings-Vision, Image and Signal Processing*, vol. 150, no. 1, pp. 6-13, 2003.
[17] H. C. Huang, S. C. Chu, J. S. Pan, C. Y. Huang and B. Y. Liao, Tabu search based multi-watermarks embedding algorithm with multiple description coding, *Information Sciences*, vol. 181, no. 16, pp. 3379-3396, 2011.