Fast Encoding Method for Image Vector Quantization by Using Partial Sum Concept in Walsh Domain

Zhibin Pan[1)], Koji Kotani[2)], and Tadahiro Ohmi[1)]

1) New Industry Creation Hatchery Center, Tohoku University, Japan
2) Department of Electronic Engineering, Graduate School of Engineering, Tohoku University, Japan
Aza-aoba 10, Aramaki, Aoba-ku, Sendai, 980-8579, Japan
E-mail: pzb@fff.niche.tohoku.ac.jp

ABSTRACT

In a framework of image vector quantization (VQ), the encoding speed is a key issue for its practical applications. To speed up the VQ encoding process, it is benificial to use lower dimensional features of k-dimensional vectors (k-D) first to check the difference between the input vector and a candidate codeword so as to reject a lot of unlikely codewords. As the features of a k-D vector in spatial domain, sum (1-D) and partial sums (2-D) are already proved to be very effective for codeword rejections in the previous works. However, because the energy of image vectors or codewords distributes almost equivalently at each dimension in spatial domain, the search efficiency through using sum and patial sums in spatial domain is still not very high.

By exploiting the energy compaction property of Walsh transform, this paper proposes to use partial sum concept in Walsh domain instead of spatial domain to improve search efficiency further. Unlike to compute two partial sums over the first half of [1, k/2] dimensions of a vector and the second half of [k/2+1, k] dimensions in spatial domain, partial sums are computed over [1, $K_{Wal}/2$] and [$K_{Wal}/2+1$, K_{Wal}] dimensions ($K_{Wal}<<k$) of the transformed vector in Walsh domain. In this way, the rejection power by using partial sums in Walsh domain can be enhanced obviously. Experimental results confirmed that the proposed method in Walsh domain can reduce the computational cost obviously compared to the previous works in spatial domain.

1. INTRODUCTION

VQ is a popular signal compression method. In a conventional VQ [1] method, VQ encoding is conducted block by block sequentially. The real distortion between an input image block and a codeword is a difference vector $D_i = x - y_i \quad i = 1,2,\cdots,N_c$, where $x=(x_1, x_2, \ldots, x_k)^T$ is an input, $y_i=(y_{i,1}, y_{i,2}, \ldots, y_{i,k})^T$ is the i^{th} codeword in the codebook $Y=\{y_i \mid i=1, 2, \ldots, N_c\}$, k is the vector dimension and "^{T}" refers to a transpose. For simplicity, D_i is usually measured by squared Euclidean distance or its energy as

$$d^2(x, y_i) = \|D_i\|^2 \overset{Def}{=} \sum_{j=1}^{k} (x_j - y_{i,j})^2 \qquad i = 1,2,\cdots,N_c \quad (1)$$

where j is the dimension of a vector, N_c is the codebook size.

Then, a best-matched codeword with minimum distortion can be determined straightforwardly by

$$d^2(x, y_w) = \min_{y_i \in Y} \left[d^2(x, y_i) \right] \qquad i = 1,2,\cdots,N_c \quad (2)$$

where y_w means the winner and subscript "w" is the winner index. This is a full search (FS) process. Clearly, FS method can achieve the best PSNR for a fixed codebook but it is computationally very expensive. Once "w" has been found, which uses much less bits than y_w, VQ only transmits this index "w" instead of "y_w" to the receiver to reduce data amount so as to realize image compression.

Obviously, the principle of VQ encoding implies that only the sole winner y_w has to be found by an exact Euclidean distance computation but all other y_i ($i \neq w$) has to be rejected actually. This means that an exact Euclidean distance for each y_i ($i \neq w$) is redundant. Instead, it is sufficient to just know whether Euclidean distance from x to y_i ($i \neq w$) is "larger" than the minimum Euclidean distance from x to y_w or not. In other words, VQ encoding can also be viewed as a process for rejecting all non-best-matched codewords rather than finding a best-matched codeword. This property of VQ provides a possibility of estimating Euclidean distance by a rough but lighter computation to see whether it is really "large" enough for rejecting a candidate codeword.

In order to make an estimation for Euclidean distance by just a little computational cost, lower dimensional features such as sum (1-D) [2] and partial sums (2-D) [3]-[5] of a vector in spatial domain are already proposed. This paper aims at using partial sum concept in Walsh domain so as to achieve a higher encoding performance.

2. WALSH TRANSFORM

It is well-known that among all kinds of orthogonal transforms such as KLT (Karhunen-Loeve transform), Haar transform, Slant transform, and DCT transform, Walsh transform is most computational inexpensive because the basis vectors in a Walsh transform kernel only consist of the value of "+1" or "−1". Therefore, it actually just needs to use addition (±) but not multiplication (×) operations to conduct Walsh transform on a vector. This property of not using multiplications (×) makes Walsh transform very suitable to fast VQ encoding. In addition, the transform performance of Walsh transform is rather good compared to other complicated orthogonal transforms [6].

Then, the most popularly-used sequency-ordered Walsh transform kernel for k=16 is given as below

$$W_{seq,16} = \frac{1}{4} \times \begin{matrix} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \end{matrix} \overset{Sequency}{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix}} = \begin{bmatrix} W_{seq,16,1} \\ W_{seq,16,2} \\ W_{seq,16,3} \\ W_{seq,16,4} \\ W_{seq,16,5} \\ W_{seq,16,6} \\ W_{seq,16,7} \\ W_{seq,16,8} \\ W_{seq,16,9} \\ W_{seq,16,10} \\ W_{seq,16,11} \\ W_{seq,16,12} \\ W_{seq,16,13} \\ W_{seq,16,14} \\ W_{seq,16,15} \\ W_{seq,16,16} \end{bmatrix}$$

where for a basis vector of $W_{seq,16,j}$ $j\in[1,k]$ in the kernel $W_{seq,16}$, the number of zero-crossings is defined as its sequency, which is an equivalent concept of the conventional frequency in Walsh domain.

Obviously, $W_{seq,16}W_{seq,16}^{T} = I_{16}$ is true, where I_{16} is an 16×16 identity matrix. This is the unitary property of Walsh transform. For VQ encoding, suppose $z=W_{seq,16}x$ and $w_i=W_{seq,16}y_i$ are the corresponding Walsh transformed vectors of x and y_i, it concludes that

$$d^2(z,w_i) \overset{Def}{=} \sum_{j=1}^{k}(z_j - w_{i,j})^2 = d^2(x,y_i) \tag{3}$$

This is the energy conservation property of Walsh transform. Eq.3 is an essential fact because the problem for finding the winner in spatial domain by Eq.1 can be equivalently converted to finding the winner in Walsh domain by Eq.3.

After conducting Walsh transform, the energy of a vector can be compacted into its first several dimensions. Then, a transform gain is used to measure how good this energy compaction process is. In a VQ encoding application, because the input image is always changing but codebook is fixed and they are independent to each other, the transform gain is usually computed only for codebook. Then, the transform gain $G_{Wal}(J)$ for a codebook till J^{th} dimension ($J\le k$) can be defined as

$$\begin{aligned} &\overline{w}_j^2 = \sum_{i=1}^{N_c} w_{i,j}^2 / N_c \\ &G_{Wal}(J) = 100\times \sum_{j=1}^{J}\overline{w}_j^2 \Big/ \sum_{j=1}^{k}\overline{w}_j^2 \qquad J=1,2,\cdots,k \end{aligned} \tag{4}$$

$G_{Wal}(J)$ gives out the energy accumulated from the first dimension to J^{th} dimension. A larger $G_{Wal}(J)$ at a smaller J value is preferred because it implies a better energy compaction effect. Correspondingly, the energy distribution $E_{Spt}(J)$ of a codebook in spatial domain is defined as

$$\begin{aligned} &\overline{y}_j^2 = \sum_{i=1}^{N_c} y_{i,j}^2 / N_c \\ &E_{Spt}(J) = 100\times \sum_{j=1}^{J}\overline{y}_j^2 \Big/ \sum_{j=1}^{k}\overline{y}_j^2 \qquad J=1,2,\cdots,k \end{aligned} \tag{5}$$

As an example, the energy distribution in spatial domain and Walsh domain for the codebook of size CB=256 is plotted in Fig.1. From Fig.1, it is clear that (1) in a k-D spatial domain, energy almost equally distributes at each dimension because $E_{Spt}(J)$ increases linearly and (2) in a k-D Walsh domain, about 90% energy has been compacted into its first four dimensions because $G_{Wal}(4)$=89.3%.

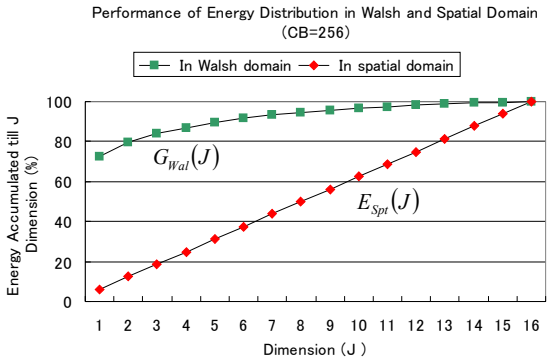

Fig.1. Comparison of energy distribution performance in Walsh domain and spatial domain.

3. PREVIOUS WORK

During a winner search process, suppose the "so far" minimum Euclidean distance is d_{min}. In spatial domain, the previous work [2] proposed a codeword rejection rule by using the sum information as

$$kd^2(x,y_i) \ge (S_x - S_{y_i})^2 \tag{6}$$

where $S_x = \sum_{j=1}^{k} x_j$ is the sum of an input image block x and $S_{y_i} = \sum_{j=1}^{k} y_{i,j}$ means the same for y_i. If $(S_x - S_{y_i})^2 \ge kd_{\min}^2$ holds, then reject y_i safely. This is ENNS (i.e. equal-average nearest neighbor search) method. To use Eq.6, it needs one "±", one "×" and one "cmp" (comparison) operation for a rejection test.

Apparently, as the feature to approximately represent a k-D vector, sum is still rather coarse. In order to reject candidate codewords more efficiently, the previous work [3]-[5] proposed to divide a k-D vector in half to generate two partial vectors and then to apply ENNS method to each partial vector separately again to construct a new rejection test. Let the first and the second partial vector of x and y_i be $px_1=[x_1, x_2, \ldots, x_{k/2}]^T$, $px_2=[x_{k/2+1}, x_{k/2+2}, \ldots, x_k]^T$, $py_{i,1}=[y_{i,1}, y_{i,2}, \ldots, y_{i,k/2}]^T$ and $py_{i,2}=[y_{i,k/2+1}, y_{i,k/2+2}, \ldots, y_{i,k}]^T$, respectively. Therefore, partial sums in spatial domain can be defined as

$$\begin{aligned} &PS_{x,1} = \sum_{j=1}^{k/2} x_j \quad , \quad PS_{x,2} = \sum_{j=k/2+1}^{k} x_j \\ &PS_{y_i,1} = \sum_{j=1}^{k/2} y_{i,j} \quad , \quad PS_{y_i,2} = \sum_{j=k/2+1}^{k} y_{i,j} \end{aligned} \tag{7}$$

Then, the previous work [3]-[5] proposed a codeword rejection rule by using the partial sum information as

$$\frac{k}{2}d^2(x,y_i) \ge (PS_{x,1} - PS_{y_i,1})^2 + (PS_{x,2} - PS_{y_i,2})^2 \tag{8}$$

If $(PS_{x,1} - PS_{y_i,1})^2 + (PS_{x,2} - PS_{y_i,2})^2 \ge \frac{k}{2}d_{\min}^2$ becomes true, then reject y_i safely. This is PENNS (i.e. partial-vector based equal-average nearest neighbor search) method. To use Eq.8, it needs three "±", two "×" and one "cmp" operation for a rejection test. It has been confirmed in [3]-[5] that PENNS method is much powerful compared to ENNS method.

Proof of Eq.8

Obviously, for each pair of partial vectors of (px_1, $py_{i,1}$) and (px_2, $py_{i,2}$), Eq.6 can be directly applied as

$$\begin{aligned} &\frac{k}{2}d_1^2(x,y_i) \overset{Def}{=} \frac{k}{2}\times\sum_{j=1}^{k/2}(x_j - y_{i,j})^2 \quad \ge (PS_{x,1} - PS_{y_i,1})^2 \\ &\frac{k}{2}d_2^2(x,y_i) \overset{Def}{=} \frac{k}{2}\times\sum_{j=k/2+1}^{k}(x_j - y_{i,j})^2 \ge (PS_{x,2} - PS_{y_i,2})^2 \end{aligned}$$

Then, Eq.8 can be obtained easily as

$$\begin{aligned} d^2(x,y_i) &\equiv d_1^2(x,y_i) + d_2^2(x,y_i) \\ &\ge [1/(k/2)]\times\left[(PS_{x,1} - PS_{y_i,1})^2 + (PS_{x,2} - PS_{y_i,2})^2\right] \end{aligned}$$ #

Clearly, a larger $[1/(k/2)]\times\left[\left(PS_{x,1}-PS_{y_i,1}\right)^2+\left(PS_{x,2}-PS_{y_i,2}\right)^2\right]$ value will be very helpful to the rejection test of Eq.8. As a result, it becomes important to make this value as large as possible.

Then, there exit two possible ways to make this value larger. First, it is beneficial to make the denominator of [(k/2)] smaller, which requires to use less dimensions of K_{Spt} (K_{Spt}<k) for rejection test or to discard some of unimportant (K_{Spt}+1, k) dimensions. Second, it is also beneficial to make the numerator of $\left[\left(PS_{x,1}-PS_{y_i,1}\right)^2+\left(PS_{x,2}-PS_{y_i,2}\right)^2\right]$ larger, which requires to use more dimensions (the maximum=k/2). According to Eq.7, the ($PS_{x,1}$, $PS_{yi,1}$) pair represents the DC components of (px_1, $py_{i,\,1}$) pair. From the viewpoint of statistics, in order to have a larger $\left(PS_{x,1}-PS_{y_i,1}\right)^2$, it is better to have larger values of ($PS_{x,1}$, $PS_{yi,1}$) pair. For example, let ($PS_{x,1}$, $PS_{yi,1}$) change around (1, 2) or (100, 200), it is obvious that it is much easier for the latter choice to achieve a larger value of $\left(PS_{x,1}-PS_{y_i,1}\right)^2$. It is similar for ($PS_{x,2}$, $PS_{yi,2}$) pair.

But according to Fig. 1, because all dimensions are almost the same important in spatial domain, it is very difficult to determine which dimensions can be discarded in practice to obtain a smaller K_{Spt} value. Meanwhile, if some dimensions are forcefully discarded, the values of ($PS_{x,1}$, $PS_{yi,1}$) pair and ($PS_{x,2}$, $PS_{yi,2}$) pair will surely become smaller according to Eq.7. Therefore, it is a conflict requirement in order to use a smaller K_{Spt} value and larger values of ($PS_{x,1}$, $PS_{yi,1}$) pair and ($PS_{x,2}$, $PS_{yi,2}$) pair in spatial domain.

4. PROPOSED METHOD

From the analysis in Section 3, it is necessary to satisfy the two requirements of a smaller K_{Spt} value and larger ($PS_{x,1}$, $PS_{yi,1}$), ($PS_{x,2}$, $PS_{yi,2}$) values in order to enhance the power of PENNS method of Eq.8, which are impossible to be realized simultaneously in spatial domain.

However, because the winner can be found equivalently in Walsh domain as well and Walsh transform itself has a good energy compaction property, it becomes possible to satisfy these two requirements in Walsh domain.

In principle, Walsh transform has a property of $z_1=\sum_{j=1}^{k}x_j\Big/\sqrt{k}$ and $w_{i,1}=\sum_{j=1}^{k}y_{i,j}\Big/\sqrt{k}$ so that Eq.6 using the sum information in spatial domain becomes

$$d^2(x,y_i)=d^2(z,w_i)\ge\left[z_1-w_{i,1}\right]^2 \tag{9}$$

If $\left[z_1-w_{i,1}\right]^2\ge d_{\min}^2$ is true, then reject w_i safely. Eq.9 in Walsh domain has the same rejection power as Eq.6.

Then, we try to use less dimensions of K_{Wal} (K_{Wal}<<k) to construct two new partial sums in Walsh domain as defined below

$$\begin{aligned} PS_{z,1}&=\sum_{j=1}^{K_{Wal}/2}z_j \quad , \quad PS_{z,2}=\sum_{j=K_{Wal}/2+1}^{K_{Wal}}z_j \\ PS_{w_i,1}&=\sum_{j=1}^{K_{Wal}/2}w_{i,j} \quad , \quad PS_{w_i,2}=\sum_{K_{Wal}/2+1}^{K_{Wal}}w_{i,j} \end{aligned} \tag{10}$$

Therefore, PENNS method in Walsh domain becomes

$$\frac{K_{Wal}}{2}d(x,y_i)=\frac{K_{Wal}}{2}d(z,w_i)\ge\left(PS_{z,1}-PS_{w_i,1}\right)^2+\left(PS_{z,2}-PS_{w_i,2}\right)^2 \tag{11}$$

If $\left(PS_{z,1}-PS_{w_i,1}\right)^2+\left(PS_{z,2}-PS_{w_i,2}\right)^2\ge\frac{K_{Wal}}{2}d_{\min}^2$ becomes true, then reject w_i safely. Because current comparison baseline of $\frac{K_{Wal}}{2}d_{\min}^2$ becomes much smaller in Eq.11 and the ($PS_{z,1}$, $PS_{wi,1}$), ($PS_{z,2}$, $PS_{wi,2}$) values become rather large due to energy compaction property in Walsh domain, Eq.11 can achieve a higher rejection power than Eq.8 in spatial domain. In other words, Walsh transform provides a possibility to simultaneously satisfy the two requirements of (1) using less dimensions to construct partial sums and (2) obtaining larger values of partial sums. Therefore, it is more promising to use partial sum concept in Walsh domain for fast VQ encoding.

5. FURTHER DISCUSSIONS

In order to search the winner in Walsh domain by using Eq.9 and Eq.11, some extra computational costs are necessary for conducting Walsh transform on the input vector x or codewords y_i. Because Walsh transform on each y_i is performed off-line, it does not affect search efficiency. However, Walsh transform on x must be performed on-line so that it would affect search efficiency to some extent. By using fast Walsh transform algorithm [7], it only needs a small amount of $k\times\log_2(k)$ additions (±) for transforming the input x compared to $k\times(k-1)$ additions (±) by the definition. For a typical 16-D input vector, it just needs $16\times\log_2(16)=64$ additions (±). Of course, only once on-line Walsh transform for each x is sufficient.

On the other hand, if the Walsh transformed codebook $W=\{w_i|\ i=1,2,\ldots,N_c\}$ is stored at the receiver, it is necessary to use inverse Walsh transform to reconstruct an image. Therefore, the codebook $Y=\{y_i|\ i=1,2,\ldots,N_c\}$ in spatial domain is preferred at the receiver. Because $W=\{w_i|\ i=1,2,\ldots,N_c\}$ at the transmitter that is sorted by "$w_{i,1}$" in ascending order has a *one-to-one* mapping relation with $Y=\{y_i|\ i=1,2,\ldots,N_c\}$ at the receiver that is sorted by "S_{yi}" in ascending order (Note: $w_{i,1}=\sum_{j=1}^{k}y_{i,j}\Big/\sqrt{k}$), it is practical to store $Y=\{y_i|\ i=1,2,\ldots,N_c\}$ that is sorted by "S_{yi}" in ascending order at the receiver. It is guaranteed that the index "w" of the winner w_w found in Walsh domain at the transmitter definitely points to the same codeword y_w at the receiver. As a result, "w" can be sent directly to the receiver to retrieval y_w so as to avoid using inverse Walsh transform.

6. EXPERIMENTAL RESULTS

To compare the performance of VQ encoding, the latest previous work [5] that is based on partial sum concept in spatial domain is used as a benchmark. Codebooks of size 256, 512 and 1024 are generated using 512×512, 8-bit Lena image as a training set. Ascending-order sorted $W=\{w_i|\ i=1,2,\ldots,N_c\}$ and $Y=\{y_i|\ i=1,2,\ldots,N_c\}$ are respectively stored at the transmitter and receiver. Block size is 4×4. Based on Fig.1, the parameter K_{Wal} is selected as 4, which is a possible minimum value for K_{Wal}. And these four dimensions can compact approximately 90% energy.

The winner search process consist of (1) performing on-line Walsh transform for the input x; (2) finding the initial best-matched codeword w_N in $W=\{w_i|\ i=1,2,\ldots,N_c\}$ by using a binary search to let $\left(z_1-w_{N,1}\right)^2\Rightarrow\min$; (3) computing two test conditions based on Eq.9 and Eq.11 up and down around w_N for realizing a possible rejection until $\left[z_1-w_{i,1}\right]^2>d_{\min}^2$ becomes true; (4) if two rejection tests in Step (3) failed, computing real Euclidean distance and updating "so far" $d^2_{\min}$ again if current w_i is a better-matched codeword.

Encoding performance is firstly evaluated by the reduced search space or remaining Euclidean distance computations.

A smaller value is better. This is because the reduced search space is a key fact in fast VQ encoding algorithms. The results are summarized in Table 1.

From Table 1, it is clear that the proposed method can reduce the search space to 65.1% ~ 93.2% depending on the details of an input image compared to the previous work [5]. It is a rather lager reduction, especially for the low-detailed images such as Lena image. This is because Eq.11 is more powerful than Eq.8 by using less dimensions and exploiting the energy compaction property of Walsh transform.

On the other hand, because (1) it requires different computational cost for on-line constructing the features of (S_x, $PS_{x,1}$, $PS_{x,2}$) and (S_z, $PS_{z,1}$, $PS_{z,2}$); (2) it requires different times of computation for the test condition by using Eq.8 or Eq.11, the overall encoding performance is secondly evaluated by the total computational cost in terms of the number of addition (±), multiplication (×) and comparison (cmp) operations per input vector with FS as a relative baseline. The results are summarized in Table 2. It is clear that the overall computation cost can also be reduced remarkably, especially multiplication (×) operations.

7. CONCLUSION

In this paper, partial sum concept is applied to Walsh domain in order to improve the existing partial-sum-based fast VQ encoding method in spatial domain. Three issues on why it is useful to use partial sum concept in Walsh domain are discussed in detail as (1) in principle, it is beneficial to use a smaller K_{Spt} value and larger ($PS_{x,1}$, $PS_{yi,1}$), ($PS_{x,2}$, $PS_{yi,2}$) values for a partial-sum-based search method; (2) it is impossible to realize this purpose in spatial domain because energy distributes at each dimension almost equivalently. However, it becomes practical to realize this purpose in Walsh domain because Walsh transform has a good energy compaction property; (3) although VQ encoding is conducted in Walsh domain, it is preferred to conduct VQ decoding in spatial domain so as to avoid inverse Walsh transform at the receiver. It becomes possible by respectively storing codebook $W=\{w_i|\ i=1,2, \ldots, N_c\}$ at the transmitter and $Y=\{y_i|\ i=1,2, \ldots, N_c\}$ at the receiver. Experimental results confirmed that it is indeed more efficient for fast VQ encoding by using partial sum concept in Walsh domain.

TABLE 1 COMPARISON OF REDUCED SEARCH SPACE OR REMAINING EUCLIDEAN DISTANCES PER INPUT VECTOR

Size	Method	Test	Lena	F-16	Pepper	Baboon
256	Previous work [5]	Eq.6	16.27	14.17	18.58	49.60
		Eq.8	9.04	7.65	10.29	34.18
	This work	Eq.9	16.27	14.17	18.58	49.60
		Eq.11	6.55	5.9	8.08	31.86
512	Previous work [5]	Eq.6	29.81	27.40	35.83	98.29
		Eq.8	15.27	14.35	19.05	67.92
	This work	Eq.9	29.81	27.40	35.83	98.29
		Eq.11	10.90	11.01	15.20	64.99
1024	Previous work [5]	Eq.6	52.06	52.55	68.86	189.97
		Eq.8	23.33	25.21	33.33	124.67
	This work	Eq.9	52.06	52.55	68.86	189.97
		Eq.11	15.19	18.86	24.88	119.58

TABLE 2 COMPARISON OF TOTAL COMPUTATIONAL COST PER INPUT VECTOR

Size	Method	Op	Lena	F-16	Pepper	Baboon
256	Full search	±	7936	7936	7936	7936
		×	4096	4096	4096	4096
		cmp	256	256	256	256
	Previous work [5]	±	376.4	324.9	424.4	1289.0
		×	209.5	181.0	236.4	711.7
		cmp	45.6	40.0	51.4	137.4
	This work	±	299.3	270.8	356.1	1217.1
		×	169.8	153.1	201.2	674.6
		cmp	43.2	38.3	49.3	135.1
512	Full search	±	15872	15872	15872	15872
		×	8192	8192	8192	8192
		cmp	512	512	512	512
	Previous work [5]	±	623.5	585.6	764.9	2529.5
		×	349.7	327.9	428.3	1397.5
		cmp	78.9	73.2	94.7	268.5
	This work	±	488.6	481.9	645.9	2439.0
		×	280.1	274.4	366.9	1350.8
		cmp	74.7	69.9	91.0	265.6
1024	Full search	±	31744	31744	31744	31744
		×	16384	16384	16384	16384
		cmp	1024	1024	1024	1024
	Previous work [5]	±	962.6	1022.8	1339.7	4655.5
		×	545.5	577.1	755.9	2580.6
		cmp	131.5	134.3	175.1	508.6
	This work	±	710.9	826.2	1078.7	4498.0
		×	415.8	475.7	621.4	2499.3
		cmp	123.7	128.2	167.0	503.6

8. REFERENCES

[1] P.C. Cosman, R.M. Gray and M. Vetterli, "Vector quantization of image subbands: a survey," *IEEE Trans. on Image Processing*, vol. 5, pp.202-225, 1996.

[2] Guan, L and Kamel, M, "Equal-average hyperplane partitioning method for vector quantization of image data," *Pattern Recognition Letters*, vol.13, pp.693-699, 1992.

[3] Zhibin Pan, Koji Kotani, and Tadahiro Ohmi, "A hierarchical fast encoding algorithm for vector quantization with PSNR equivalent to full search," *2002 IEEE International Symposium on Circuits and Systems (ISCAS 2002)*, pp.I-797~I-800, May 2002.

[4] Zhibin Pan, Koji Kotani, and Tadahiro Ohmi, "A fast full search equivalent encoding method for vector quantization by using appropriate features," *2003 IEEE International Conference on Multimedia and Expo (ICME 2003)*, Vol. II of III, pp.261-264, July 2003.

[5] Zhibin Pan, Koji Kotani, and Tadahiro Ohmi, "An improved fast encoding method for vector quantization based on memory-efficient data structure," *2004 IEEE International Conference on Multimedia and Expo (ICME 2004)*, June 2004.

[6] R.Costantini, J.Bracamonte, G.Ramponi, J.L. Nagel, M.Ansorge and F.Pellandini, "A low-complexity video coder based on the discrete Walsh-Hadamard transform," *Proc. of 2000 European Signal Processing Conference (EUSIPCO 2000)*, pp.5-8, 2000.

[7] K.G. Beauchamp, *Application of Walsh and related functions*, Academic Press, London, 1984.