Video Compression Using Multiwavelet and Multistage Vector Quantization

Esakkirajan Sankaralingam[1], Veerakumar Thangaraj[2], Senthil Vijayamani[3], and Navaneethan Palaniswamy[1]
[1]Department of Electrical and Electronics Engineering, PSG College of Technology, India
[2]Department of Electronics and Communication Engineering, PSG College of Technology, India
[3]Department of Electrical and Electronics Engineering Nasik, Mahindra and Mahindra, India

Abstract: *This paper presents a new video coding technique using multiwavelet transform and multi-stage vector quantization. Three types of redundancies that are common in video sequence are spatial, temporal and psycho visual redundancies. In this work, the spatial redundancy in the video is minimized using multiwavelet transform. The transform coefficients are then quantized using multi-stage vector quantization scheme. Motion estimation/ compensation reduce temporal redundancy by exploiting interpicture correlation. Kite cross diamond search is the block matching algorithm used for motion estimation. The objective is to develop low bit rate video coder with acceptable visual quality. The performance of the proposed method is compared with wavelet based video coder.*

Keywords: *Multiwavelet, multistage vector quantization, kites cross diamond search.*

Received November 30, 2007; accepted May 12, 2008

1. Introduction

Video compression plays an important role in many digital video applications, such as digital libraries, video on demand and high definition television. The objective of a video compression [14] algorithm is to exploit both the spatial and temporal redundancy of a video sequence such that fewer bits can be used to represent a video sequence at an acceptable visual distortion. An NTSC digital video sequence which conforms to the ITU recommendation (720 x 486 pixels per luminance frame, 4:2:2 chrominance sub sampling, 8 bits per pixel per color) has an uncompressed data rate of 168 mega-bits per second. A typical two-hour movie would occupy approximately 150 gigabytes of disk space. This fact implies that there is a need for compression. Compression can be achieved by minimizing the spatial and temporal redundancies present in the video. The spatial redundancy within a frame is minimized by multiwavelet transform [16] followed by Multi-Stage Vector Quantization (MSVQ). The tool which is widely used to minimize spatial redundancy is wavelet transform. Wavelet based coding [1] provides substantial improvements in picture quality. For better performance in compression, filters used in wavelet transform should have the property of orthogonality, symmetry, short support and higher approximation order. Due to implementation constraints scalar wavelets do not satisfy all these properties simultaneously. New class of wavelets called ‘Multiwavelets’ which posse's more than one scaling function overcomes this problem. Multiwavelets can achieve better level of performance [12] than scalar wavelets with similar computational complexity. Vector Quantization (VQ) has been shown to be effective in image compression. Full search VQ enjoys small quantization distortion. However, it has long compression time and may not be well-suited for real-time signal compression systems. Tree-Structured Vector Quantization (TSVQ) although can significantly reduce the compression time, has the disadvantage that the storage size required for the VQ is usually very large and cannot be controlled during the design process. It is not convenient to use TSVQ for applications where the storage size is a major concern. In MSVQ [8], the compression time is short and the required storage size is small and has extensive applications than full search VQ and TSVQ. In addition to being used as a tool for signal compression, MSVQ can also be used to implement progressive information transmission system [4]. MSVQ offers potential for progressive coding and avoids the complexity obstacle of excessively large codebook sizes but it suffers from the need for a high bit-rate for each additional stage that is added. In our scheme, we have employed two stages in MSVQ. By exploiting the temporal redundancies present in the image sequence, video-compression techniques are able to achieve a high quality of the reconstructed image at low bit rates. Temporal redundancy can be minimized by efficient motion estimation/compensation algorithm. The estimation of motion vector is done through block matching algorithm. In this work we have employed Kite Cross Diamond Search (KCDS) [3] algorithm for block matching.

This paper is organized as follows. Section 2 highlights some key points on multiwavelets. Section 3 gives brief introduction to multi-stage vector quantization scheme. Section 4 explains the concept of motion estimation and compensation. The proposed algorithm is explained in section 5. Results and discussions are presented in section 6 and finally conclusions are drawn in section 7.

2. Multiwavelet Transform

The spatial redundancy which is present between the image pixels can be reduced by taking transforms which decorrelates the similarities among the pixels. The choice of the transforms depends upon a number of factors, in particular, computational complexity and coding gain. Coding gain is a measure of how well the transformation compacts the energy into a small number of coefficients. The predicted error frames are usually encoded using either block-based transforms, such as DCT [13], or non-block-based coding, such as subband coding or the wavelet transform. A major problem with a block-based transform coding algorithm is the existence of the visually unpleasant block artifacts, especially at low data rates. This problem is eliminated using wavelet transform, which is usually applied over the entire image. The wavelet transform has been used in video coding for the compression of motion predicted error frames [11].

In the context of image coding application, the following three properties are important (1) Orthogonality to ensure the decorrelation of subband coefficients (2) Symmetry in order to obtain linear phase so that finite length signals can be processed without redundancy and artifacts and (3) Finite length filters for computational efficiency. However, most real scalar wavelet transforms fail to possess these properties simultaneously. To circumvent these limitations, multiwavelets have been proposed in this work where orthogonality and symmetry are allowed to co-exist by relaxing the time-invariance constraint [10]. Multiwavelets can be considered as generalization of scalar wavelets. Scalar wavelets have a single scaling function $\phi(t)$ and wavelet function $\psi(t)$. Multiwavelets have two or more scaling and wavelet functions. In general r scaling functions can be written using the vector notation $\Phi(t) = [\phi_1(t), \phi_2(t), ...\phi_r(t)]^T$, where $\Phi(t)$ is called the multiscaling function. Similarly the multiwavelet function using r wavelet functions as $\Psi(t) = [\psi_1(t), \psi_2(t), ...\psi_r(t)]^T$. The scalar case is represented by r = 1. The two scale relationship for $r = 2$ is given by

$$\phi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} H_k \phi(2t - k) \quad (1)$$

$$\psi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} G_k \phi(2t - k) \quad (2)$$

where $\{H_K\}$ and $\{G_K\}$ are 2 x 2 matrix filters defined as

$$H_k = \begin{bmatrix} h_0(2k) & h_0(2k+1) \\ h_1(2k) & h_1(2k+1) \end{bmatrix} \quad (3)$$

$$G_k = \begin{bmatrix} g_0(2k) & g_0(2k+1) \\ g_1(2k) & g_1(2k+1) \end{bmatrix} \quad (4)$$

where $\{h_k(n)\}$and $\{g_k(n)\}$are the scaling and wavelet filter sequences such that $\sum_n h_k^2(n) = 1$ and $\sum_n g_k^2(n) = 1$ for k = 1, 2. The matrix elements in the filter given in equations 3 and 4 provide more degrees of freedom than a traditional scalar wavelet. Due to these extra degrees of freedom, multiwavelets can simultaneously achieve orthogonality, symmetry and high order of approximation.

Each color video frame with the color triplets [*R, G, B*] is first transformed into the YUV 4:1:1 color domain, and all subsequent processing is performed in this transformed color domain. The subsampling ratio 4:1:1 denotes subsampling of the chrominance channels, *U* and *V*, by a factor of two in both horizontal and vertical directions. Figure 1 illustrates the organization and labeling of the multiwavelet subbands of *Y, U*, and *V* channels. The *Y* channel is decomposed into '*L*' multiwavelet octave scales, the *U* and *V* channels are decomposed into '*L*-1' scales as a result of the corresponding subsampling. The symbols *V, H, D*, denote the vertical, horizontal and diagonal orientation subbands of the decomposition at each resolution scale, while the symbol *L* represents the low pass-filtered sub images.

3. Multi Stage Vector Quantization

VQ collects the transformation coefficients into blocks and assigns one symbol to each block. VQ is a powerful tool for data compression. Vector quantization extends scalar quantization to higher dimensional space. The superiority of VQ lies in the block coding gain, the flexibility in partitioning the vector space, and the ability to exploit intra-vector correlations. MSVQ is an error refinement scheme in which inputs to a stage are residual vectors from the previous stage and they tend to be less and less correlated as the process proceeds. MSVQ is a non-uniform vector quantizer, which exploits the correlation between vector components and allocates quantization centroids in accordance with the probability distribution density. According to information theory, the overall rate distortion performance of a uniform quantizer is superior to a non-uniform quantizer in higher bit rates and the non-uniform quantizer can be superior in the medium and low bit ranges. It justifies the use of MSVQ in the lower bit range. Also MSVQ is a structured vector quantization scheme in which substantial complexity such as search time and code book storage reduction with respect to optimal VQ is obtainable [5].

$L_{2,1}$	$L_{2,2}$	$V_{2,1}$	$V_{2,2}$	$V_{1,1}$	$V_{1,2}$
$L_{2,3}$	$L_{2,4}$	$V_{2,3}$	$V_{2,4}$		
$H_{2,1}$	$H_{2,2}$	$D_{2,1}$	$D_{2,2}$	$V_{1,3}$	$V_{1,4}$
$H_{2,3}$	$H_{2,4}$	$D_{2,3}$	$D_{2,4}$		
$H_{1,1}$		$H_{1,2}$		$D_{1,1}$	$D_{1,2}$
$H_{1,3}$		$H_{1,4}$		$D_{1,3}$	$D_{1,4}$

(a) Luminance channel.

$L_{1,1}$	$L_{1,2}$	$V_{1,1}$	$V_{1,2}$
$L_{1,3}$	$L_{1,4}$	$V_{1,3}$	$V_{1,4}$
$H_{1,1}$	$H_{1,2}$	$D_{1,1}$	$D_{1,2}$
$H_{1,3}$	$H_{1,4}$	$D_{1,3}$	$D_{1,4}$

(b) Chrominance channel U.

$L_{1,1}$	$L_{1,2}$	$V_{1,1}$	$V_{1,2}$
$L_{1,3}$	$L_{1,4}$	$V_{1,3}$	$V_{1,4}$
$H_{1,1}$	$H_{1,2}$	$D_{1,1}$	$D_{1,2}$
$H_{1,3}$	$H_{1,4}$	$D_{1,3}$	$D_{1,4}$

(c) Chrominance channel V.

Figure 1. Multiwavelet subband structure.

In this paper, we adopt a two-stage vector quantizer as illustrated in Figure 2. The input vector '*X*' is quantized by the first-stage vector quantizer denoted by VQ_1 whose code book is $C_1 = \{C_{10}, C_{11,\ldots\ldots} C_{1(N2-1)}\}$ with size N_1. The quantized approximation $\hat{X}_1$ is then subtracted from X producing the error vector e_2. This error vector is then applied to the second vector quantizer VQ_2 whose codebook is $C_2 = \{C_{20}, C_{21,\ldots\ldots} C_{2(N2-1)}\}$ with size N_2 yielding the quantized output $\hat{e}_2$. The encoder for this VQ simply transmits a pair of indices specifying the selected codewords for each stage and the task of the decoder is to perform two table look ups to generate and then sum the two codewords. In the Figure 2, I_1, I_2 indicates the indices from the first, second stages respectively. The decoder receives the indices and reconstructs the image.

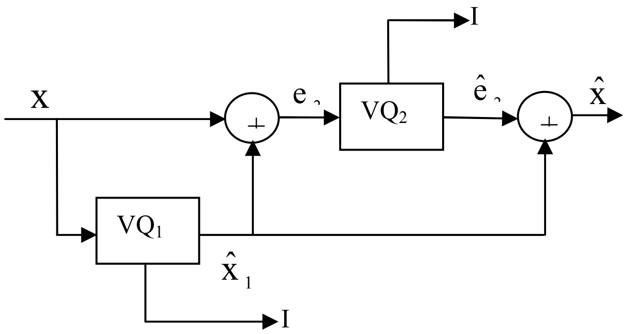


Figure 2. Encoder of two-stage VQ.

4. Motion Estimation and Compensation

The main objective of any motion estimation algorithm is to exploit the strong frame to frame correlation along the temporal dimension. Motion estimation examines the movement of objects in an image sequence to obtain vectors representing the estimated motion. Motion is described by a two-dimensional vector, known as motion vector that specifies where to retrieve a macro-block from the reference frames. The motion vector can be found using matching criterion. The motion vector that minimizes some cost measure involving the candidate and the target macro blocks is usually selected. Although many cost measures can be used, the most widely used measure is the Sum of Absolute Difference (SAD). There are two mainstream techniques of estimation of motion vectors: Pel-Recursive Algorithm (PRA) and Block-Matching Algorithm (BMA). Block matching algorithms are suitable for a simple hardware realization because of their regularity and simplicity. In block matching, the current image is divided into blocks and each block is matched with a reference frame. The best matched block of pixels from the reference frame is then used in the current block. The simplest BMA, known as the full search or exhaustive search BMA, evaluates the SAD at every possible pixel location in the search area. For software implementations, the computational burden of the full search algorithm is usually comparable to or greater than that of all remaining encoding steps combined. Consequently, many algorithms with a reduced number of search locations have been proposed in the literature namely logarithmic search [7], the three-step search [9], the cross-search [6], the one at a time search [15], and the hierarchical search [2] algorithms. However most of these algorithms can quickly get trapped in local minima, resulting in a significant loss in estimation accuracy. To overcome this problem, KCDS is used as a block matching algorithm in this work. Motion compensation is usually block based, that is the current image is divided into blocks and each block is matched with a reference frame. The best matched block of pixels from the reference frame is then used in the current block. The prediction error frame is obtained by taking the difference between the current frame and the motion predicted frame. The predicted error frame is then encoded using multiwavelet transform.

4.1. Kite Cross Diamond Search Algorithm

In this section the search patterns used in the algorithm is described first, and later the search path strategy will be explained. Search patterns: the search-point configuration used in the KCDS is divided in 4 different shapes: cross-shaped pattern, diamond-shaped pattern, Kite-Shaped Pattern (KSP) and Biased-Corner Pattern (BCP). Figure 3 shows the Small Cross-Shaped Pattern (SCSP) and the Large Cross-Shaped Pattern (LCSP). The same search pattern for diamond shape is Small Diamond-Shaped Pattern (SDSP) and Large Diamond Shaped Pattern (LDSP). Unlike traditional search pattern, such as square, diamond, cross – all are vertically and horizontally symmetry, in the KSP, only the diagonal that connects the longer ends of the kite is the line of symmetry. Figure 4 (a) shows the vertical-kite described as up-kite in which the dart (the most outer vertex) is pointing up. Figure 4 (b) is called left-kite. For another 2 KSP –down-kite and right-kite are shown in Figure 4 (c) and (d). BCP [3], is sharing the same center of the KSP and it depends on the direction of the dart to indicate the biased point of searching.

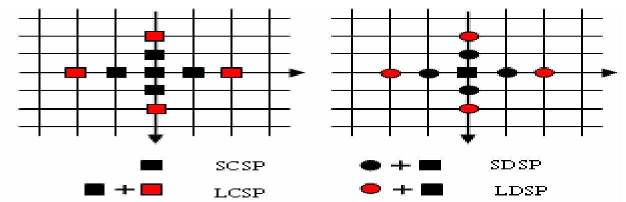


Figure 3. Search patterns for KCDS.

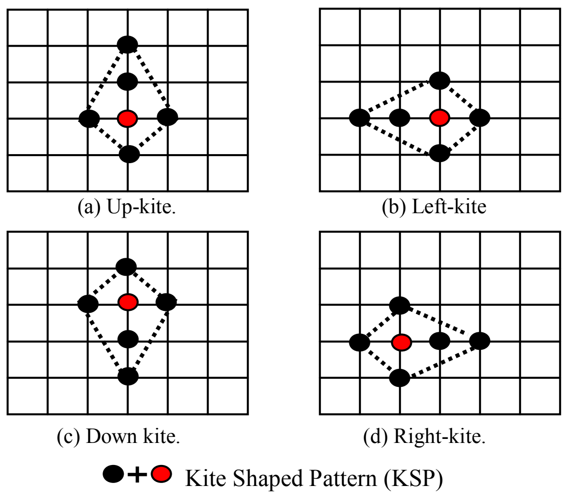


Figure 4. Kite search patterns: a, b with vertical symmetry c, d with horizontal symmetry.

The algorithm: the difference between KCDS and Diamond Search (DS) is that the first step of KCDS is a SDSP, which is saving the number of search point for stationary. The details of the algorithm are given below:

- Starting SCSP: a minimum Block Distortion Measure (BDM) is found from the 5 search points of the SCSP as shown in Figure 3 (a), located at the center of search window. If the minimum BDM point occurs at the center of the SCSP (0, 0), the search stops (First Step Stop); otherwise, go to step 2.
- KSP: with the vertex (minimum BDM point) in the first SCSP as the center, a particular KSP is formed based on the motion direction in previous step. For example, if the minimum BDM is located in upper vertex in first step, the new KSP will be an up-kite shape (the dart is pointing up) described as Figure 4 (a). Thus, depending on the MV direction in step 1, there are 4 cases of newly formed KSP in this step: up-kite, down-kite, right-kite and left-kite, if the minimum BDM point occurs at the center of this KSP, then go to step 3; otherwise go to step 4.
- Checking the two BCP points by following the biased of KCP of previous step. For example, if there is a up-kite in previous step, the BCP will be the up-biased corner against the center. If the minimum BDM point is still unchanged, then the search stop (third-step stop). Otherwise go to Step 4.
- Diamond Searching: a new Large-Diamond-Shaped Pattern (LDSP) is formed by repositioning the minimum BDM found in previous step as the center of the LDSP. If the new minimum BDM point is at the center of the newly formed LDSP, then go to Step 5 for converging the final solution; otherwise, this step is repeated.
- Ending: with the minimum BDM point in the previous step as the center, a SDSP is formed. Identify the new minimum BDM point from the SDSP, which is the final solution.

5. Proposed Algorithm

The block diagram of the proposed scheme is shown in Figure 5. The proposed scheme shows both the video encoder and decoder part. The image encoder is basically used to minimize spatial redundancy. The detail block diagram of the image encoder block in Figure 5 is shown in Figure 6. From the Figure 6, it is clear that multiwavelet transform is taken for the input frame, the resultant coefficients are grouped into vectors. The vectors are mapped into one of the code vectors through a search engine. The number of code vectors in the codebook is decided by rate and dimension. In this work we have fixed the dimension as two and varied the rate from 0.125 bpd to 1.00 bpd.

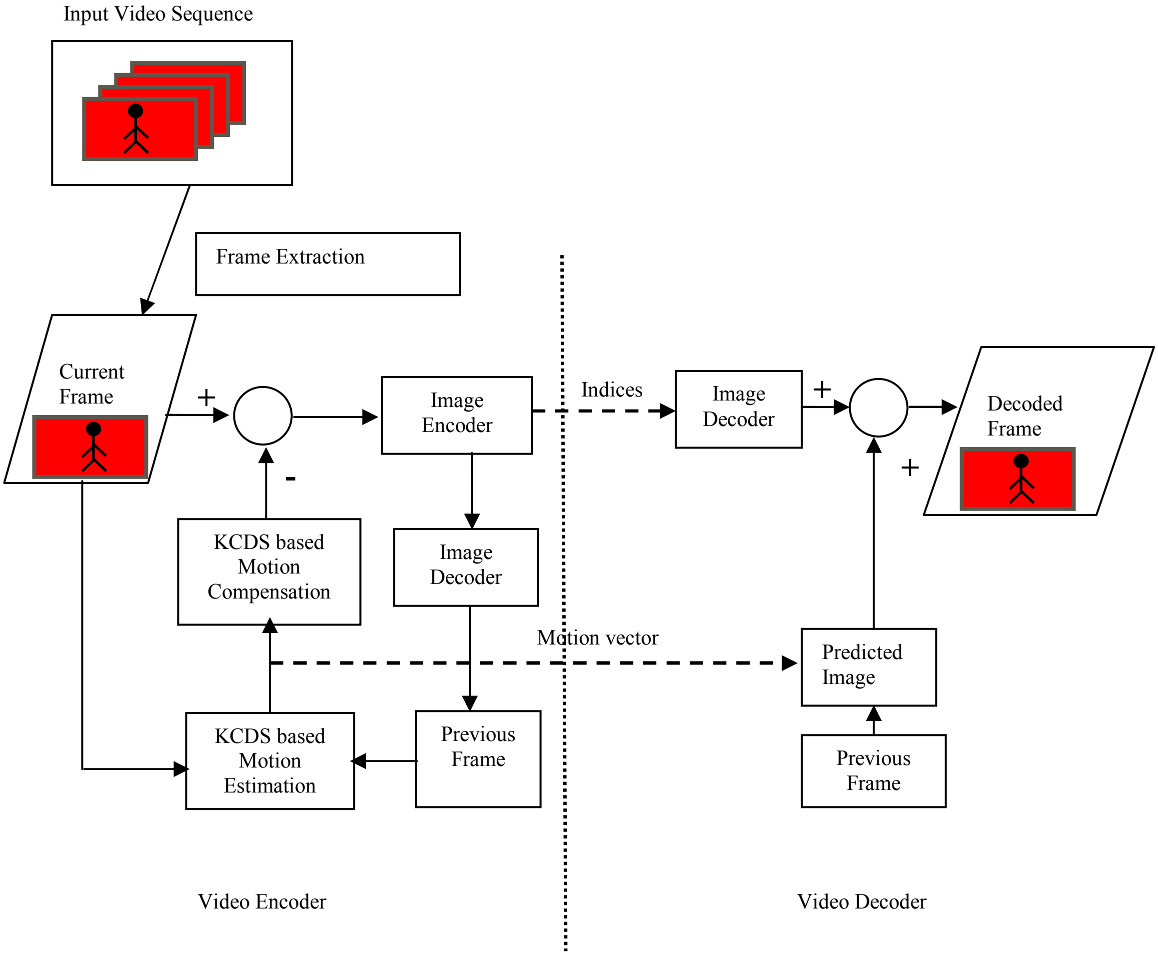


Figure 5. Proposed scheme.

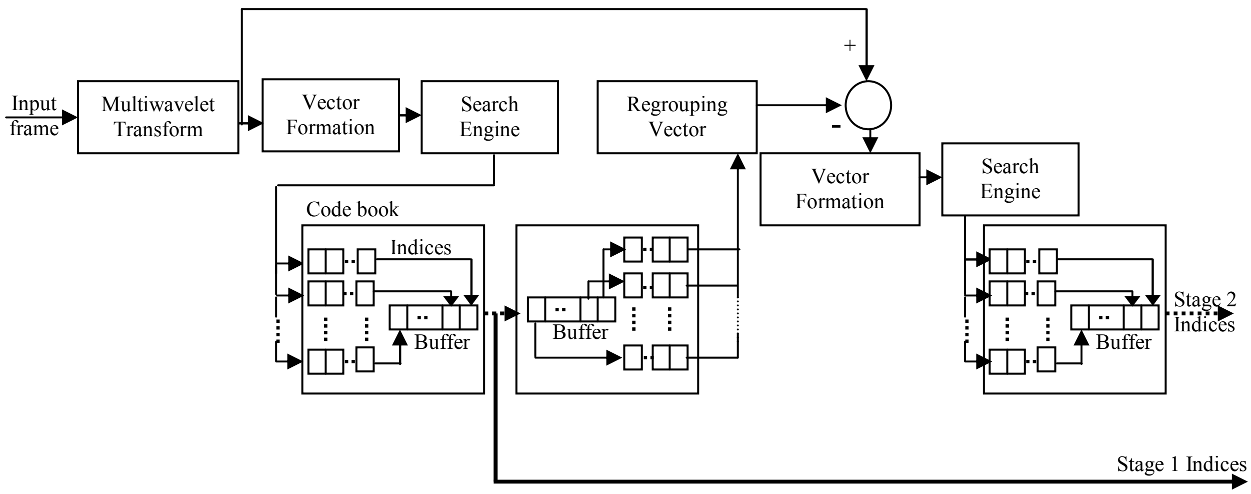


Figure 6. Image encoder block.

The number of stages incorporated in MSVQ is two. The input to the second stage is the error from the previous stage. The output of MSVQ is the indices from stage 1 and stage 2. From the indices, the image is reconstructed using image decoder which performs the reverse operation of image encoder. The correlation between successive frames is minimized by motion estimation/compensation. The block matching algorithm used in the motion estimation is KCDS. The block size chosen is 8 x 8 in the block matching algorithm. The estimated motion vector and the indices from image encoder are transmitted to the receiver.

6. Results and Discussion

In this work, two sets of video sequences are used. First set is CIF (352 x 288) which includes 'Dancer', 'Football' video sequences. The other set is QCIF (176 x 144) with the video sequences 'Claire', 'Foreman', 'Trevor' and 'Miss America'. To evaluate the performance of the proposed scheme, the peak signal to noise ratio (PSNR) based on mean square error is used as a quality measure, and its value can be determined by the following equation

$$PSNR = 10\log_{10}\left(\frac{255^2}{\frac{1}{N}\sum\left(P_{ref}(x,y) - P_{prc}(x,y)\right)^2}\right) \quad (5)$$

where N the total is number of pixels within the image, P_{ref} (x, y) and P_{prc} (x, y) are the pixel values of the reference and the processed images respectively. The summation of PSNR, over the image frames, will then be divided by the total number of frames to obtain the average value. The performance of the proposed scheme is compared with wavelet based scheme using the average PSNR value over fifty frames obtained from the experiment. The codes are run on Pentium dual core processor with 512 Mb RAM.

Table 1. Results of 'miss America' video sequence.

Rate (bpd)	Average PSNR (dB)			
	Wavelet		Multiwavelet	
	Haar	LA8	CLAP	SA4AP
0.125	29.60	29.43	30.17	29.70
0.25	36.54	36.32	36.37	36.24
0.5	40.73	40.51	38.05	37.79
0.75	42.34	42.23	38.71	38.48
1.0	43.53	43.41	39.22	38.97

Table 2. Execution time comparison in 'miss America' video sequence.

Rate (bpd)	Execution time (Sec)			
	Wavelet		Multiwavelet	
	Haar	LA8	CLAP	SA4AP
0.125	317.04	720.96	227.78	247.12
0.25	370.02	772.95	273.34	290.81
0.5	531.48	935.60	437.18	473.45
0.75	582.93	1042.09	519.92	549.15
1.0	1189.85	1556.03	1048.12	1061.56

The proposed algorithm is applied to 'Miss America' video sequence. This sequence is a typical video conferencing sequence with a talking head-and-shoulder person and static background. There is only small local motion and no global motion. The proposed algorithm is compared with wavelet based multistage quantization scheme, the parameters taken into consideration are average PSNR and the execution time and the results are given in Tables 1 and 2, respectively. From Table 1, it is obvious that at low bit rate, the performance of multiwavelet is better than wavelet. Also the performance of the multiwavelet varies with the choice of the prefilter and multifilter chosen. The multiwavelet filters taken for comparison are symmetric/anti symmetric multifilter ('SA4'), Chui-Lian orthogonal multifilter ('CL'). The corresponding prefilters used are 'SA4AP' and 'CLAP'. The scalar wavelet filters taken for comparison are 'Haar' and 'LA8'. It is found that among multiwavelets 'CL' performs better than 'SA4'. The computation time for 'CL' is lower than 'SA4' and other scalar wavelets chosen which is evident from Table 2. At high bit rate, the performance of scalar wavelet is better than multiwavelet at the expense of increased execution time. With respect to execution time, the performance of multiwavelet is superior to that of the scalar wavelets.

The performance of the proposed algorithm is tested for 'dancer' video sequence. In the dancer sequence, there is a fast motion of human bodies. At low bit rate, the performance of multiwavelet is better than wavelet, which is evident from Table 3. At high bit rate, the performance of wavelets dominates the performance of multiwavelets at the expense of increased computation time which is evident from Table 4.

Table 3. Results of 'dancer' video sequence.

Rate (bpd)	Average PSNR (dB)			
	Wavelet		Multiwavelet	
	Haar	DB4	CLAP	SA4AP
0.125	22.96	23.98	23.41	23.63
0.25	32.24	32.22	33.12	33.12
0.5	36.96	36.89	36.78	36.72
0.75	38.81	38.68	38.15	38.08
1.0	39.92	39.77	38.94	38.85

Table 4. Execution time comparison of 'dancer 'sequence.

Rate (bpd)	Execution time (Sec)			
	Wavelet		Multiwavelet	
	Haar	DB4	CLAP	SA4AP
0.125	2192.41	3103.67	1545.02	1672.97
0.25	2801.06	3481.78	2254.20	2352.09
0.5	3699.45	4154.97	3068.36	3029.25
0.75	4208.94	5377.04	3396.13	3278.53
1.0	6685.39	6866.27	5760.42	5323.20

From Tables 3 and 4, it is obvious that the choice of scalar filter and multifilter has an impact both in the quality of the reconstructed sequence as well as the computation time. From Table 4, it is clear that at low bit rate, the computation time taken by multiwavelet is almost half that of wavelet. This implies that multiwavelet is superior to wavelet with respect to computation time especially at low bit rate. The performance of wavelet is compared with that of multiwavelet for different test video sequences at the rate of 0.5 bits per dimension and the results are given in Tables 5 and 6 respectively. Table 5 shows the average PSNR results for different video sequences. From Table 5, it is obvious that except 'Claire' video sequence, the performance of multiwavelet matches the performance of wavelets, with reduced computation time. From Table 6, it is obvious that irrespective of the test sequence, the execution time of multiwavelet is less than that of scalar wavelet.

Table 5. Comparison of wavelet and multiwavelet for different video sequence.

Video sequence	Average PSNR (dB)			
	Wavelet		Multiwavelet	
	Haar	DB4	CLAP	SA4AP
Miss_am.yuv	40.74	40.51	38.06	37.79
Trevor.yuv	41.73	42.13	34.52	35.25
Foreman.yuv	36.70	38.02	34.71	34.51
Claire.yuv	41.05	41.27	28.51	28.75

Table 6. Execution time of wavelet against multiwavelet for different test sequence.

Video sequence	Execution time (Sec)			
	Wavelet		Multiwavelet	
	Haar	DB4	CLAP	SA4AP
Miss_am.yuv	531.48	637.52	437.18	473.45
Trevor.yuv	534.84	675.14	465.84	457.86
Foreman.yuv	539.69	700.72	425.14	433.05
Claire.yuv	532.55	678.86	432.05	451.69

The average PSNR and the execution time results of 'Miss America' video sequence for different scalar and multifilters at a fixed rate of 0.5 bits per dimension is given in Table 7. It is found that among multiwavelets 'CL' performs better than other multiwavelets chosen and among wavelets 'Haar' performs better than other wavelets chosen.

The first, twenty fifth and fiftieth reconstructed frame of 'Dancer' and 'Miss America' video sequence using multiwavelet ('CL') are presented in Figure 7 and 8, respectively. The Haar wavelet performance is better than all other scalar wavelet chosen, similarly, among multiwavelet 'CL' performance is better than all other multiwavelet chosen. It is worth to compare the performance of 'Haar' with respect to 'CL'. The execution time and the rate distortion curve comparison of 'Haar' and 'CL' multiwavelet is given in Figures 9 (a) and (b) respectively. From the figure, the following conclusions can be drawn, with respect to execution time, the performance of 'CL' is better than that of 'Haar' wavelet. With respect to rate-distortion behavior, at low bit rate, the performance of 'CL' is better than 'Haar', at high bit rate, the performance of 'Haar' is far better than 'CL' at the expense of increased execution time.

Table 7. Comparison of different scalar and multifilters.

	Wavelet				Multiwavelet			
	Haar	DB4	Bi9/7	Bi5/3	CLAP	SA4AP	GHMAP	ID
Avg. PSNR (dB)	36.54	36.29	36.25	36.63	36.37	36.24	31.68	34.91
Exec. time (Sec)	370.01	498.52	773.03	500.75	273.34	290.81	307.72	414.97

Figure 7. Reconstructed 1, 25 and 50^{th} frames in 'dancer' video.

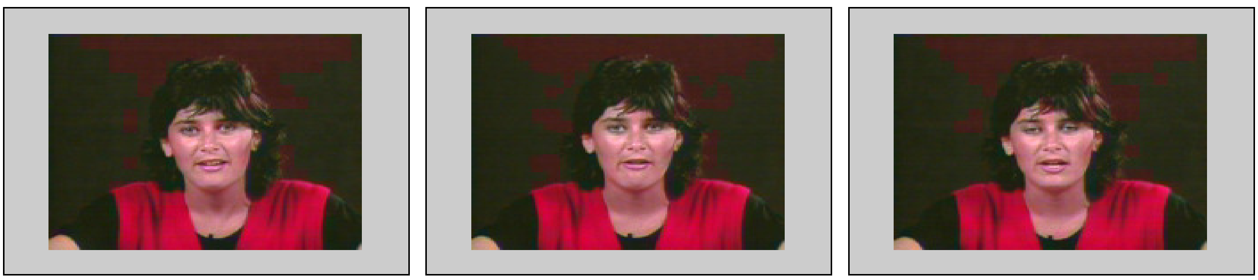
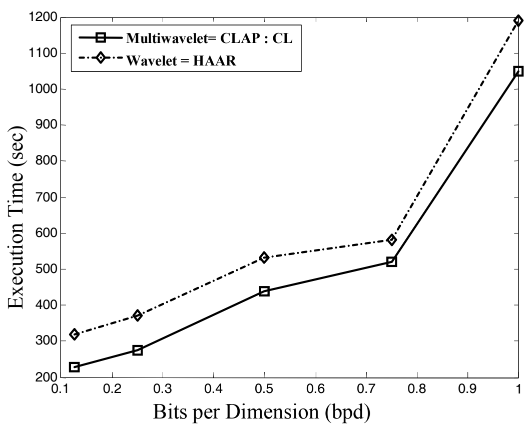
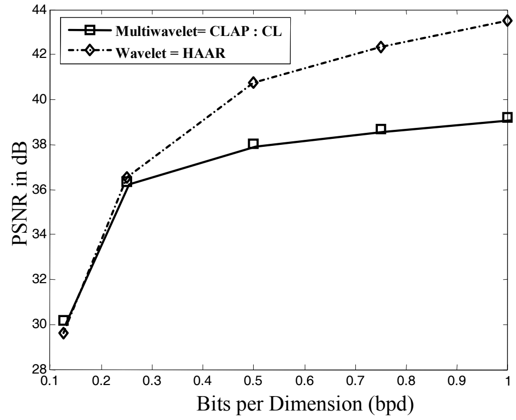


Figure 8. Reconstructed 1, 25 and 50^{th} frames in 'miss America' video sequence.



(a) Comparison of execution time of multiwavelet vs wavelet.



(b) Rate-distortion curve of multiwavelet vs wavelet.

Figure 9. Essential analysis.

7. Conclusion

The computation complexity of the algorithm, along with the compressed data rate, and the visual quality of the decoded video are the three major factors used to evaluate a video compression algorithm. An ideal algorithm should have a low computation complexity, a low compressed data rate and a high visual quality for the decoded video. However, these three factors cannot be achieved simultaneously. The computation complexity directly affects the time needed to compress a video sequence. The multiwavelet outperforms wavelet in terms of execution time. Hence the proposed video compression algorithm with multiwavelet as the transform, multi-stage vector quantization as the quantization scheme and kite cross diamond search as the block matching algorithm achieves visually acceptable video quality and also it is computationally efficient which is evident from the reduced execution time. The visual quality can be enhanced by including more stages in multistage vector quantization, but it will result in increased execution time.

Acknowledgements

The authors wish to thank their teachers Dr. S. Jayaraman, Dr. N. Malmurugan, and Mr. R. Sudhakar for their continued support and encouragement. They also thank their present institution where they are working.

References

[1] Antonini M., Barlaud M., Mathieu P., and Daubechies I., "Image Coding Using Wavelet Transform," *Computer Journal of IEEE Transaction Image Processing*, vol. 1, no. 8, pp. 205-220, 1992.

[2] Bhaskaran V. and Konstantinides K., *Image and Video Compression Standards: Algorithms and Architecture*, Boston, Kluwer, 1995.

[3] Chi-Wai L., Lai-Man P., and Chun Ho C., "A Novel Kite-Cross Diamond Search Algorithm for Fast Block Matching Motion Estimation," *in Proceedings of IEEE International Symposium on Circuits and Systems*, Canada, pp. 94-98, 2004.

[4] Equitz W. and Cover T., "Successive Refinement of Information," *Computer Journal of IEEE Transaction Information Theory*, vol. 37, no. 2, pp. 269-275, 1991.

[5] Gersho A. and Gray R., *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1995.

[6] Ghanbari M., "The Cross-Search Algorithm for Motion Estimation," *Computer Journal of IEEE Transaction Communcation*, vol. 38, no. 3, pp. 950-953, 1990.

[7] Jain J. and Jain A., "Displacement Measurement and its Application in Interframe Image Coding," *Computer Journal of IEEE Transactions on Communication*, vol. 29, no. 4, pp. 1799-1808, 1981.

[8] Juang H. and Gray A., "Multiple Stage Vector Quantization for Speech Coding," *in Proceedings of ICASSP*, France, pp. 597-600, 1982.

[9] Koga K., Hirano A., Iijima Y., and Ishiguro T., "Motion-Compensated Interframe Coding for Video Conferencing," *in Proceedings of NTC'81*, New Orleans, pp. 239-251, 1992.

[10] Lebrun J. and Vetterli M., "Balanced Multiwavelets: Theory and Design," *Computer Journal of IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1119-1125, 1998.

[11] Martucci S., Sodagar I., Chiang T., and Zhang Y., "A Zerotree Wavelet Video Coder," *Computer Journal of IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 109-118, 1997.

[12] Micheal B. and Amy E., "New Image Compression Techniques using Multiwavelets

and Multiwavelet Packets," *Computer Journal of IEEE on Transactions Image Processing*, vol. 10, no. 4, pp. 500-511, 2001.

[13] Rao K. and Yip P., *Discrete Cosine Transform: Algorithms, Advantages and Applications*, Academic Press, 1990.

[14] Iain G., *Video Codec Design*, John Wiley & Sons, New Jersey, 2002.

[15] Srinivasan R. and Rao R., "Predictive Coding Based on Efficient Motion Estimation," *Computer Journal of IEEE Transactions on Communication*, vol. 33, no. 5, pp. 888-896, 1985.

[16] Strela P., Strang G., Topiwala P., and Heil C., "Application of Multiwavelet Filter Banks to Image Processing," *Computer Journal of IEEE Transactions on Image Processing*, vol. 8, no. 4, pp. 548-563, 1999.

Esakkirajan Sankaralingam received his Bachelor of science degree from Sadakathullah Appa College, Palayamkottai, Bachelor of technology degree from Cochin University of Science and Technology. He received his postgraduate degree from PSG College of Technology. He is currently working as a lecturer in Instrumentation and Control Systems Engineering Department, PSG College of Technology, Coimbatore.

Veerakumar Thangaraj received his Bachelor of engineering degree from RVS College of Engineering and Technology, Dindugal. He received his Master of engineering degree from PSG College of Technology, Coimbatore. He is currently working as a Lecturer in Electronics and Communication Engineering Department, PSG College of Technology, Coimbatore.

Senthil Vijayamani is currently working as a professor in Electrical and Electronics Engineering Department, PSG College of technology, Coimbatore. He received his Bachelor of engineering degree from Coimbatore Institute of Technology, Master of engineering from Anna University and Doctorate degree from Indian Institute of Science, India.

Navaneethan Palaniswamy received his Bachelor of engineering degree from PSG College of Technology. He is currently working as a design engineer in Mahindra and Mahindra, Chennai, India.