IACSIT International Journal of Engineering and Technology, Vol.2, No.3, June 2010
ISSN: 1793-8236

Video coding Technique using 3-D Dual tree Complex Wavelet Transform with Improved Particle Swarm Optimization

M. Thamarai and Dr. R. Shanmugalakshmi

***Abstract*—Video compression plays an important role in video signal processing, transmission and storage. Since the available bandwidth for transmission is very limited, Multimedia Applications such as video conferencing, video on demand, video telephony and remote sensing are not possible without compression. A lot of video compression techniques have been developed and the video signal transmission has followed at data rates below 64kbps.Wavelet transform based motion compensated video codec performs better compression in order to meet the rate and distortion constraint in video transmission for the available bandwidth than the block based techniques, which are followed in standard video transmissions such as H.261 and H.263.But the efficiency of those technique's depends on the way in which it estimates and compensates the object motions in the video sequence. Wavelet based embedded image coder is quite attractive in modern multimedia applications. Wavelet transform, bit plane coding and other techniques make embedded image coder practical and also provide efficient compression. In this paper, we have proposed a novel video coding using swarm intelligence in dual tree complex wavelet transform for video coding. The 3-D DDWT is an attractive video representation because it isolates motion along different directions in separate subbands. However, it is an over-complete transform with redundancy, which is going to be eliminated by choosing optimal subbands with the help of Noise shaping followed by an Improved Particle Swarm Optimization (PSO). The proposed video codec does not require motion compensation and provides better performance than the 3D SPIHT (Embedded type) codec, both objectively and subjectively, and the coder allows full scalability in spatial, temporal and quality dimensions.**

***Index Terms*—Dualtree Discrete Wavelet Transform, NoiseShaping, Particle Swarm Optimization.**

I. Introduction

Most of the lossy compression technique use signal transforms those produce a hierarchical or layered representation of the input content of video frame or image in space and time [10-13]. In this representation, the significant visual information tends to be clustered in a small percentage of transform coefficients, while the remaining coefficients tend to constitute a sparse representation. Such transform representations can be efficiently quantized and coded with a variety of techniques depending on the available bandwidth, while a percentage of the transform-coefficient information is ignored. It is important to note that, in the case of video, the sparseness in the transform domain representation is significantly increased with the use of motion estimation and compensation techniques those exploit temporal similarities among neighboring frames [2]. In many cases, depending on the performance of the utilized motion estimation model, as well as the transform and coding techniques, a visually near-lossless representation of the input video can be obtained after decoding. For example, modern state-of-the-art video coders can achieve compression ratios of more than 100:1 with little loss of visual quality. This comes as a result of more than thirty years of research.

The standard separable discrete wavelet transform (DWT) provides a multi-resolution representation of a signal and has established an impressive reputation for video compression. Several recently proposed DWT-based video coders have achieved coding efficiency similar to or slightly better than block-based hybrid video coders [1]. But the poor directional selectivity of the multidimensional DWT can lead to checkerboard artifacts at the low bit rate range. An important recent development in wavelet-related research is the design and implementation of 2-D multiscale transforms that represent edges more efficiently than does the DWT. Kingsbury's complex dual-tree wavelet transform (DT-CWT) is an outstanding example [2]. The DT-CWT is an overcomplete transform with limited redundancy (2^m:1 for m-dimensional signals). This transform has good directional selectivity[4] and its subband responses are approximately shift-invariant. The 2-D DT-CWT has given superior results for image processing applications compared to the DWT [2, 3].

Recently, Selesnick and Li introduced a 3-D version of the dual-tree wavelet transform and showed that it has superior motion selectivity [4]. The major challenge to apply the 3-D complex DDWT for video coding is it is overcompleteness transform with 8:1 redundancy. By choosing the real parts of the wavelet coefficients, perfect reconstruction is obtained with the motion selectivity retained. This reduces the redundancy to 4:1 [4]. To reduce the number of coefficients, Kingsbury proposed an iterative projection-based noise shaping (NS) scheme [3], which modifies previously chosen large coefficients to compensate for the loss of small coefficients. Wang et al., [5, 16] found that noise shaping applied to 3-D DDWT can yield a more compact set of coefficients than from the 3-D DWT. The fact that noise

M. Thamarai, Research Scholar, GCT, Coimbatore, Assistant Professor/ECE Dept., Karpagam College of Engineering, Coimbatore ,India. Email:sreethamarai2000@yahoo.co.in.

Dr. R. Shanmugalakshmi, Assistant Professor, Department OF CSE, GCT, Coimbatore, India(email:shanmuga_lakshmi@yahoo.co.in).

shaping can reduce the number of coefficients below that required by DWT (for the same video quality) is very encouraging.

In [5], the vector entropy study we understand that only a few bases have significant energy for an object feature. The relatively low entropy of the significant vector across subbands suggests that the where about of significant coefficients may be coded efficiently by coding the significance bits across subbands jointly. Based on the above investigation, we proposed a video codec in this paper that doesn't require motion compensation and allows full spatial, temporal and quality salability by using 3D-DDWT hybrid with Swarm Improved Particle Swarm Optimization (IPSO). PSO has been implemented in a wide range of research areas such as functional optimization, pattern recognition, neural network training, fuzzy system control and obtained significant success.

PSO is widely used to solve nonlinear and multiobjective problems such as optimization of weights of neural networks (NN), electrical utility, computer games, and mobile robot path planning, etc. As a recursive algorithm, the PSO algorithm simulates social behavior among individuals (particles) "flying" through a multidimensional search space, where each particle represents a point at the intersection of all search dimensions. The particles evaluate their positions according to certain fitness functions at each iteration, and particles in a local neighborhood share memories of their "best" positions, then use those memories to adjust their own velocities and positions. In PSO, a particle is an independent intelligence agent, which searches the problem space based on its own experience and the experience of peer particles. It leads to the problem of prematurity and easily trapping in local optimum, a modified PSO algorithm is proposed, that the only global best particle is perturbed in every iteration of the algorithm and other particles are updated according to original updating method. And this may increase the time for reaching the best positions. A number of variations in standard PSO have been presented in the literature to avoid local minimum problem. In our Improved PSO, at each iteration we are choosing 'n' number of best particles ($1< n < N/2$, where N is the total number of particles) and share this best solutions with the rest of the particles based on crossover operation. Thus, the other particles can also travel with the best solution instead of performing independent search.

In this paper first the signal is decomposed by dual-tree real discrete wavelet transform. Noise shaping scheme is used to select the significant coefficients from the vast ddwt coefficients. Next using the improved version of PSO algorithm, the dual tree subband coefficients which contain high energy are identified. The identified subband coefficients in each plane are encoded using significance map coding like in SPHIT algorithm.

This paper is organized as follows. Section 2 briefly introduces the 3-D DDWT and its properties for video representation. Section 3 describes the proposed codec in detail. Section 4 presents the coding results of the video codec using 3-D DDWT-IPSO. The final section summarizes our work.

II. 3D-DUAL-TREE WAVELET TRANSFORM

The design and the motion-selectivity of dual-tree filters are described in [6] and [2]. The dual-tree wavelet transform is implemented by first applying separable transforms and then combining subband signals with simple linear operations. Even though it is non-separable, it inherits the computational efficiency of separable transforms.

Figure 1 illustrates the difference between the standard 3-D DWT and the 3-D DDWT. The figure depicts the wavelets (i.e. the basis functions) associated with the 3-D DWT and the 3-D DDWT respectively. As illustrated, the 3-D DWT mixes different orientations in one wavelet basis, but the 3-D DDWT is free of this effect. The 3-D DDWT has many more subbands than DWT. while decomposing the signal it contains 28 High subbands and 4 low subbands. Each subband represents the coefficient with one wavelet basis. The 28 high subbands isolate 2-D edges with different orientations that are moving in different directions. Because of this motion selectivity, the 3-D DDWT does not require separate motion compensation procedure for video coding.

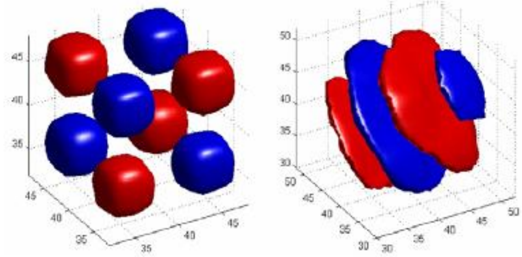

Figure.1 Iso surfaces of a typical 3-D DWT (left) and typical 3-D DDWT (right). For the 3-D DDWT, each subband corresponds to motion in a specific direction.

The analysis and synthesis Filter Banks (FB) [2] used to implement the dual-tree CWT and its inverse are illustrated in Figures 2 and 3.The two real wavelet transforms use two different sets of filters, with each satisfying the PR conditions [2].

The $h0(n)$, $h1(n)$ denote the low-pass/high-pass filter pair for the upper FB, and let $g0(n)$, $g1(n)$ denote the low-pass/high-pass filter pair for the lower FB. $\psi h(t)$ and $\psi g(t)$ are the two real wavelets associated with each of the two real wavelet transforms. In addition to satisfying the PR conditions, the filters are designed so that the complex wavelet

$$\psi(t) = \psi h(t) + j\psi g(t) \quad (1)$$

is approximately analytic. They are designed such as $\psi g(t)$ is approximately the Hilbert transform of $\psi h(t)$

$$\psi g(t) \approx H\{\psi h(t)\} \quad (2)$$

2D Complex DWT is constructed as

$$\psi(x,y) = [\psi h(x) + j\psi g(x)][\psi h(y) + j\psi g(y)] \quad (3)$$

Real part of $\psi(x,y)$ is given by

$$\text{Real part}\{\psi(x,y)\} = \psi h(x)\,\psi h(y) - \psi g(x)\,\psi g(y) \quad (4)$$

Thus the real wavelet filters are implemented in Dual tree CWT. Though 2D DWT uses two separable real filters, it introduces check board artifact problem, But Complex Dual tree doesn't have such problem [17] and the filters are not separable one. It is two times expansive in 1-D because the total output data rate is exactly twice the input data rate.

The inverse of the dual-tree CWT is as simple as the forward transform. To invert the transform, the real part and the imaginary part are each inverted the inverse of each of the two real DWTs are used to obtain two real signals. These two real signals are then averaged to obtain the final output.

One major obstacle for applying the 3-D DDWT for video coding is that it is an over complete transform by a factor of eight or four (if only the real parts of the coefficients are retained).The drawback can be eliminated by the suitable selection of subband coefficients using swarm intelligence. Moreover in Particle swarm intelligence, particles are real values. Here the DDWT coefficients are also real and a memory constrained implementation (Video coding) can be achieved by the suitable subband selection using IPSO.

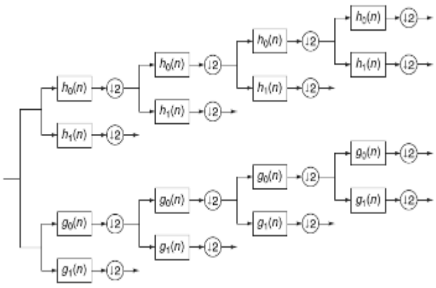

Figure 2. DDWT Analysis filter bank

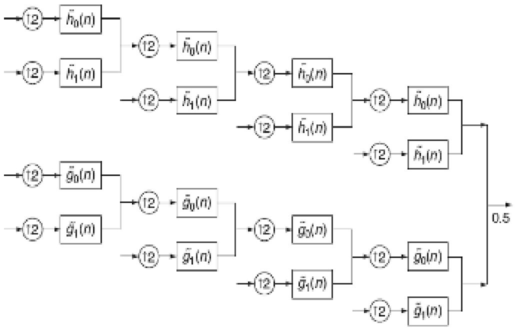

Figure 3. DDWT Synthesis Filter Bank

III. VIDEO CODING USING IPSO IN 3D-DDWT

The proposed coder (DDWT-Swarm Intelligence) first applies 3D DTWT to the input video sequence. The video sequence is subjected to temporal decomposition which constructs the sequence into group of frames and then it is subjected to it spatial decomposition. The DDWT coefficients are vast in number since it is an expansive type transform. Instead of directly applying PSO algorithm to find the significant coefficients we go for any one of the coefficient reduction method. Noise shaping scheme is used to select the significant coefficients. Among the significant coefficients, IPSO algorithm selects the 3D DTWT coefficients those have to be retained, and then a bit plane coder is applied to code the retained coefficients. The low subbands and high sub bands are coded separately.

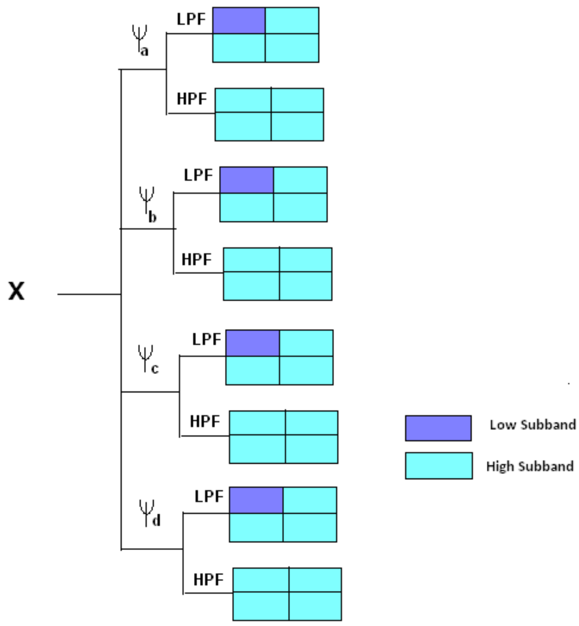

Figure 4. 1 level decomposition of input signal using dual tree discrete wavelet transform

Figure 4 shows the 1 level decomposition of the input signal.The signal is decomposed into 4 low subbands and 28 high subbands. Here Low pass filters are h(n) and high pass filters are g(n).The four wavelets bases(Ψa, Ψb, Ψc and Ψd) of dual tree real discrete transform is obtained by the combination of four three dimensional wavelets $\psi_1(x,y,z)$, $\psi_2(x,y,z)$, $\psi_3(x,y,z)$, and $\psi_4(x,y,z)$.

$$\Psi_1(x, y, z) := \Psi_h(x)\,\Psi_h(y)\,\Psi_h(z) \qquad (5)$$

$$\Psi_2(x, y, z) := \Psi_g(x)\,\Psi_g(y)\,\Psi_h(z) \qquad (6)$$

$$\Psi_3(x, y, z) := \Psi_g(x)\,\Psi_h(y)\,\Psi_g(z) \qquad (7)$$

$$\Psi_4(x, y, z) := \Psi_h(x)\,\Psi_g(y)\,\Psi_g(z) \qquad (8)$$

$$\Psi a\,(x; y; z) = 0{:}5\,(\Psi 1(x; y; z) - \Psi 2(x; y; z) - \Psi 3(x; y; z) - \Psi 4(x; y; z)) \qquad (9)$$

$$\Psi b(x; y; z) = 0{:}5\,(\Psi 1(x; y; z) - \Psi 2(x; y; z) + \Psi 3(x; y; z) + \Psi 4(x; y; z)) \qquad (10)$$

$$\Psi c(x; y; z) = 0{:}5\,(\Psi 1(x; y; z) + \Psi 2(x; y; z) - \Psi 3(x; y; z) + \Psi 4(x; y; z)) \qquad (11)$$

$$\Psi d(x; y; z) = 0{:}5\,(\Psi 1(x; y; z) + \Psi 2(x; y; z) + \Psi 3(x; y; z) - \Psi 4(x; y; z)) \qquad (12)$$

A. *Noise Shaping*

Noise shaping is an iterative projection based algorithm. The dual tree wavelet coefficients are subjected to this NS Scheme in order to select the significant coefficients from vast DT DWT coefficients. It is faster when comparing to the matching pursuit algorithm. More than one coefficients are selected in an iteration. In this algorithm initially simple thresholding method is applied and the significant coefficients are retained. But the simple thresholding method is not suitable for non orthogonal transforms like DT DWT. So the retained coefficients magnitudes are modified by multiplying those by a small constant k. This is done for the compensation of the loss of small coefficients due to thresholding. The overall block diagram is shown below in figure 5.

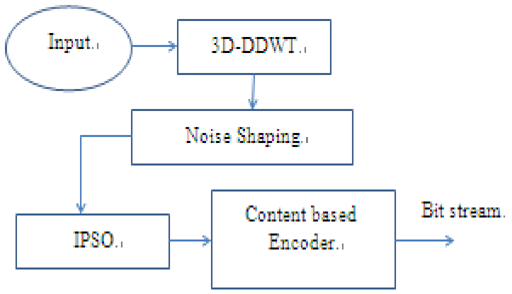

Figure 5. Block diagram of the proposed coder

B. *Coding of the Significance Mas*

Although 3-D DDWT has 28 high subbands, only a few subbands have significant energy for an object feature and the typical combination of significant subbands at the same spatio-temporal location is quite predictable. In order to achieve high compression ratio, with good quality of the reconstructed video, the context based[8] arithmetic vector coding is used. In our proposed coder, the optimal subbands are selected. using Swarm Intelligence, which includes several heuristics methods such as Particle Swarm Optimization and Ant Colony Optimization. Particle Swarm Optimization is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995, inspired by social behavior of bird flocking. [14, 15].

C. *Particle Swarm Optimization*

Particle swarm optimization uses a population of particles. Each particle is a potential solution. The system is initialized with a population of random solutions and searches for optima, according to some fitness function, by updating particles over generations; that is, particles “fly” through the N-dimensional problem search space by following the current better-performing particles.

Particle swarm algorithm is a kind of evolutionary algorithm based on swarm intelligence. Each potential solution is considered as one particle, and these particles are distributed stochastically in the high-dimensional solution space in the initialization period of the algorithm. Through following the optimum discovered by itself and the entire group, each particle periodically updates its own velocity and position.

$$V_{id}(t+1) = w \times v_{id}(t) + c_1 \times rand_1(.) \times (p_{id} - x_{id}) + c_2 \times rand_2(.) \times (p_{gd} - x_{id}) \quad (13)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1),\ 1 \le i \le N,\ 1 \le d \le D \quad (14)$$

Where, N is the number of particles and D is the dimensionality; $Vi = (v_{i1}, v_{i2},\ldots, v_{iD})$, $v_{id} \in [-v_{max}, v_{max}]$ is the velocity vector of particle i which decides the particle’s displacement in each iteration. Similarly, $Xi = (x_{i1}, x_{i2}, \ldots, x_{iD})$, $x_{id} \in [-x_{max}, x_{max}]$ is the position vector of particle i which is a potential solution in the solution space. The quality of the solution is measured by a fitness function; w is the inertia weight which decreases linearly during a run; c_1, c_2 are both positive constants, called the acceleration factors which are generally set to 2.0; $rand_1(.)$ and $rand_2(.)$ are two independent random number distributed uniformly over the range [0, 1]; and p_g, p_i are the best solutions discovered so far by the group and itself respectively.

In the t + 1 time iteration, particle i uses p_g and p_i as the heuristic information to updates its own velocity and position. The first term in the above equation represents the diversification, while the second and third intensification. The second and third terms should be understood as the trustworthiness towards itself and the entire social system respectively. Therefore, a balance between the diversification and intensification is achieved based on which the optimization progress is possible. As widely adopted, we measure the amount of computation and the quality of compensated video sequence by Computation and Peak Signal-to-Noise Ratio (PSNR). Computation is defined as the average number of the error function evaluations per MV generation. Due to the minimum computational cost, we choose Summed Absolute Difference (SAD) as the error function which is defined as follows:

$$SAD = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}(|I_k(i,j) - I_{k-1}(i,j)|) \quad (16)$$

where the size of a MB is N × N. The motion estimate quality between the original I_{ogn} and the compensated video sequences I_{cmp} is measured in PSNR which is defined as:

$$PSNR = 10\log_{10}\frac{I_{max}^2}{\sigma_e^2} \quad (17)$$

$$\sigma_e^2 = MSE = \frac{1}{N}\sum_{k=0}^{K}\sum_{i=0}^{N}\sum_{j=0}^{N}(I_{ogn}(i,j,k) - I_{cmp}(i,j,k))^2 \quad (18)$$

where K is the number of frames in the video sequence. In standard PSO, at each iteration a best position is chosen from the particles known as ‘Xpbest’. This best solution is compared with the best solution so far (Xgbest). If current is the best then the global best is interchanged with Xpbest, otherwise the procedure is continued with previous best. In this case, the particles are independent each other, they are not sharing the information about their travel. This leads to local minimum and may be takes long time for convergence. In our proposed method, we are choosing ‘n’ number of best solutions at each iteration. The value of should satisfy the following condition:

$1 < n < N/2$, where N = total number of particles

And these ‘n’ best solutions are compared with previous best solutions. Finally ‘n’ best solutions are chosen globally. And with these ‘n’ best solutions are matted with each other at random to fill the population size. Here, we are performing single-point crossover operation to perform matting. For example, if n=3, and the population size is 5, consider the following 3 best solutions.

1 0 1 0 0 1 0 1 0 1 0 1 0
1 0 0 0 0 1 1 1 0 0 0 1 1
1 0 1 0 0 1 0 1 1 1 0 0 1

With these 3 best solutions, to fill the population we need 2 more series that can be generated by performing single point crossover between 2 & 3 as given below.

(2nd best) 1 0 0 0 0 1 1 <u>1 0 0 0 1 1</u>
(3rd best) 1 0 1 0 0 1 0 <u>1 1 1 0 0 1</u>
(Crossover at 5th bit)
(4th solution) 1 0 1 0 0 1 0 1 0 0 0 1 1 (5th solution) 1 0 0 0 0 1 1 1 1 1 0 0 1

Thus, in our proposed method, the particles can communicate and share the best solutions with each other. Based on this the positions and the velocity changed. This process leads local minimum and minimize the time for convergence.

Generally, there are two widely adopted stopping criteria. One is Fixed-iteration, that is, given a certain iteration time, saying N, the search stops after N times of iteration. The other is Specified-threshold. During a PSO run, the most-fitted value found by the entire group p_g, called the "best so far" value will be updated by the particles. For minimization problems, we specify a very small threshold ε, and if the change of pg during t times of 4 iteration is smaller than the threshold, we consider the group best value very near to the global optimum, thus the matching procedure stops. Due to the center-biased characteristics of real-world motion fields, we adopt the fixed-iteration method in this paper for reducing the computational cost.

Step 1. Generate initial population of Swarm (Xi) and Velocity (Vi).
Step 2. Each swarm represents subset of blocks.
Step 3. Calculate the fitness value of each swarm.
Step 4. Calculate 'n' number of Xgbest for each particle.
Step 5. Change the position and velocity of each particle based on crossover operation.
Step 6. Again calculate the fitness value of each swarm.
Step 7. Find out 'n' number of Xpbest for each particle.
Step 8. Compare Xgbest and Xpbest, hold best as Xgbest.
Step 9. Repeat from Step 5 for maximum number of cycles.

IV. Experiments & Results

In this section, we evaluate the coding performance of the proposed video codec using 3-D DDWT-PSO. The comparisons are made to DDWTVC and 3-D SPIHT [7], which also does not use motion compensation. DTWTVC explores the spatio-temporal correlation among the subbands. Only the comparisons of luminance component Y are presented here. All experimental results are obtained by actually running the codec software. For compression, 3-level wavelet decompositions are applied. The 3-D SPIHT uses the Daubechies (9, 7)-tap filters. For DDWT+PSO, the Daubechies (9, 7)-tap filters are used at the first level, and Qshift filters in [4] are used below level 1.

Two video sequences "Foreman" and "Rhinos" are used for testing. Both sequences have 80 frames with a frame rate of 30 fps. Table 1 lists the average PSNR of the two sequences at different bit rates. For a video sequence which has many edges and motions, such like "Foreman", DDWTVC outperforms 3-D SPIHT more than 1 dB. For sequence "Rhinos", DDWTVC offers around 1 dB better PSNR results. Whereas our proposed codec DDWT-PSO outperforms better than DDWTVC and 3-D SPIHT with more than 3 dB for both the sequences. Subjectively, DDWT-PSO has better performance than the existing and has the redundancy caused by symmetric extension, the coding results are very promising. Figures 6 and 7 show the graph between the number of frames and the PSNR value, as shown, the more frame reaches maximum compression.

TABLE-1: Average PSNR Comparision at Different Bit Rates

Video Sequences	Foreman			Rhinos		
Compression Ratio	1:4	1:3	1:2	1:4	1:3	1:2
Bit-rate (kbps)	730	1000	1424	730	1000	1424
DDWT + IPSO	33.98	37.43	39.35	28.96	33.71	35.37
DDWT	31.79	32.45	34.51	26.95	29.62	31.15
3D-SPIHT	29.32	31.47	33.86	26.42	27.29	30.93

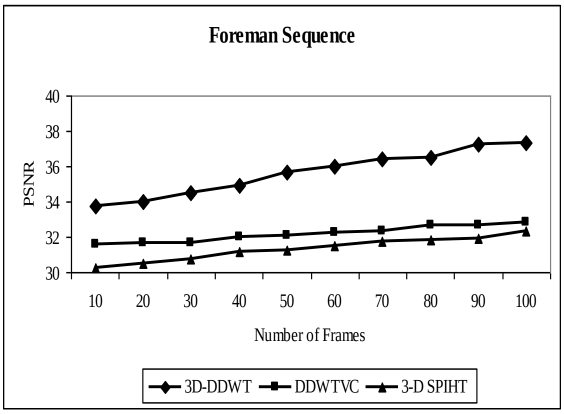

Figure 6. Performance Analysis of Foreman Sequence

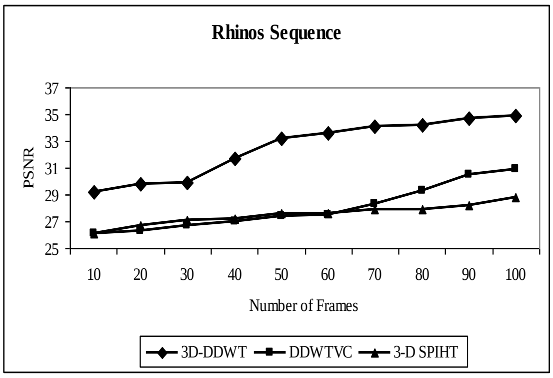

Figure 7. Performance Analysis of Rhinos Sequence

V. Conclusion

In this paper, a new video codec using the novel 3-D Dual-tree wavelet transform hybrid with Particle Swarm Optimization (PSO) is proposed and tested on standard video sequences foreman and Rhinos. The DDWT-PSO video codec applies adaptive vector arithmetic coding across subbands to efficiently code the significance bits jointly. At the time of coding the subbands, the optimal subbands are chosen by using the PSO algorithm. In terms of future work, the spatial dependence in each subband needs to be further explored to improve the coding efficiency [5]. For DDWT, the lifting steps [9] which map integers to integers may further reduce the coded data and improve the computation speed. Another challenging open research problem is to design a rate-distortion optimized scalable video coder, so that each additional coefficient offers a maximum reduction in distortion without modifying the previous coefficients. The result shows that our proposed codec outperforms the existing method with better quality.

REFERENCES

[1] S-T Hsiang and J. W. Woods, "Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank", Signal Processing: Image Communications, vol. 16, pp. 705-724, May 2001.
[2] N.G. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals", Applied Computational Harmonic Anal, vol. 10, no. 3, pp. 234-253, May 2001.
[3] T. H. Reeves and N. G. Kingsbury, "Over complete image coding using iterative projection-based noise shaping", ICIP 02, Rochester, NY, Sept 2002.
[4] I. W. Selesnick, and K. Y. Li, "Video denoising using 2D and 3D dual-tree complex wavelet transforms", Wavelet Appl. Signal Image Proc. X (Proc. SPIE 5207), Aug 2003.
[5] B. Wang, Y. Wang, I. W. Selesnick and A. Vetro, "An investigation of 3D Dual-Tree wavelet transform for video coding", ICIP 2004.
[6] I. W. Selesnick, "The design of approximate Hilbert transform pairs of wavelet bases", IEEE Trans. on Signal Processing, 50(5): 1144-1152, May 2002.
[7] B-J Kim and W.A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)", Data Compression Conference, 1997. DCC '97. Proceedings.
[8] J. Hua; Z. Xiong; X. Wu, "High-performance 3-D embedded wavelet video (EWV) coding", Multimedia Signal Processing, 2001 IEEE Fourth Workshop on, 3-5 Oct. 2001.
[9] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps", J. Fourier Anal. Appl., 4 (no. 3), pp. 247-269, 1998.
[10] A. Puri, J. Ribas-Corbera, W. Husak, and G. W. (eds), "Special issue on Technologies enabling Movies on Internet, HD DVD, and DCinema," Signal Processing: Image Communication, vol. 19, 2004.
[11] T. Sikora, "Trends and perspectives in image and video coding," Proceedings of the IEEE, vol. 93, pp. 6-18, 2005.
[12] G. Sullivan and T. Wiegand, "Video compression - from concepts to the H.264 standard," Proceedings of the IEEE, vol. 93, pp. 18-31, 2005.
[13] J.-R. Ohm, "Advances in scalable video coding," Proceedings of the IEEE, vol. 93, pp. 42-56, 2005.
[14] Kennedy, J., Eberhart, R. C.: "Particle swarm optimization," in IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.
[15] Kennedy, J., Eberhart, R. C.: "A discrete binary version of the particle swarm optimization algorithm," in IEEE International Conference on Neural Networks, Perth, Australia, 1997, pp. 4104-4108.
[16] Beibei Wang,Yao Wang, Ivan Selenick and Antony Vetro"Video coding using Dual tree wavelet transform"EURASIP Journal on Image and video processing Hundawi Publishing Corporation, Jan 2007