Asian Journal of Information Technology 5 (4) : 367-375, 2006

A Novel Video Coding Algorithm Using Wavelet Footprints and Nonlinear Approximation

R. Sudhakar, N. Vignesh and S. Jayaraman
Department of Electronics and Communication Engineering,
PSG College of Technology Peelamedu,Coimbatore-641 004, Tamilnadu, India

Abstract: The recent explosion in digital video storage and delivery has presented strong motivation for high performance video compression solutions. Video compression takes the advent of spatial, temporal and psycho-visual redundancies which can be compensated efficiently by transform, motion estimation and quantization respectively. An efficient video compression technique must preserve the trade-off between compression ratio and the quality of the video. The performance of existing image and video coding standards generally degrade at low bit-rates because of the underlying block based Discrete Cosine Transform (DCT) scheme. Over the past decade, the success of wavelets in solving many different problems has contributed to its unprecedented popularity. A new representation of these 2D (Image) signals called as wavelet footprints are the finite scale space vectors containing all the wavelet coefficients generated by particular polynomial discontinuities. The objective of this study is to develop an efficient compression scheme for the video stream by fast motion estimation algorithms up to half pixel accuracy as in newest international standard, ITU-T H.264/AVC and to obtain better quality and higher compression ratio through wavelet footprints.

Key words: Quantization, prediction, entropy coding, piecewise polynomial signal, wavelet footprints, cone influence

INTRODUCTION

Video compression is now essential for applications such as transmission and storage. Video coding has evolved through the development of the ISO/IEC MPEG-1, MPEG-2 and ITU-T H.261, H.262 and H.263 video coding standards[1,2] (and later enhancements of H.263 known as H.263+ and H.263++). Throughout this evolution, continued efforts have been made to maximize coding efficiency.

The performance of these coders generally degrades at low bit-rates mainly because of the underlying block-based Discrete Cosine Transform (DCT)[3] scheme. In the DCT the input image needs to be blocked. So correlation across the block boundaries is not eliminated resulting in noticeable and annoying blocking artifacts. More recently, the wavelet transform has emerged as a cutting edge technology, within the field of compression[4-6]. Wavelet-based coding[7] provides substantial improvements in picture quality at higher compression ratios. P.L. Dragotti and M. Vetterli proposed a new type of representation of 1D signals called footprints[8-9].They efficiently characterize the singular structures[6,9,10] of the signal which usually carry critical information. Here Matching Pursuit[11] was used in compressing the signal based on footprints which gives the sparser representation[12]. In this study we focus algorithm based on footprints for Video compression and the results are tabulated.

This study is organized as follows: In section 2 the new design of video coding Algorithm is explained. Section 3 discusses cone of influence and wavelet footprints.

In section 4. Video encoding and decoding using wavelet footprints are discussed. Results of video compression are shown in section 5 and finally conclusions are drawn in the section 6.

DESIGN OF NOVEL VIDEO CODING ALGORITHM

Multiple reference picture motion compensation: Prediction in MPEG-2 and its predecessors use only one previous picture to predict the values in an incoming picture. The new design extends upon the enhanced reference picture selection technique found in H.263++ and H.264 to enable efficient coding by allowing an encoder to select, for motion compensation purposes, among a three frames that have been decoded and stored in the decoder.

Variable block-size motion compensation with small block sizes: Our algorithm supports more flexibility in the selection of motion compensation block sizesthan any previous standard, with a minimum luma motion compensation block size, as small as, 4x4. Our algorithmsupports block sizes of 16×16, 8×8 and 4×4.

Half-sample-accurate motion compensation: Most prior standards enable integer sample motion vector accuracy at most. The new design improves up on this by adding half sample motion vector accuracy, as first found in an H.263 standard, but further reduces the complexity of the interpolation processing by fast algorithms compared to the prior design.

Fast algorithms for motion estimation: To reduce the time taken to find the best matching block among three reference frames fast estimation algorithm-new cross diamond search algorithm is used. To reduce the complexities involved in interpolations processing, Fast Two Step method for half pixel accuracy motion estimation is used.

Image based transform: All the existing standards for both image and video coding except JPEG2000 uses DCT based transform. In the Wavelet based transforms there is no need to block the image. Wavelet based coding provides substantial improvement in picture quality at higher compression ratios mainly due to the better energy compaction property of wavelet transforms.

Adaptive quantization method: The choice of a good quantizer depends on the transform that is selected, since the properties possessed by the coefficients from the transformation stage depend on the transform used. For evaluating the effectiveness of the coding images or videos at low bit rates, an effective Novel technique called wavelet footprints is applied.

Predictive coding of motion vectors: Encoding a motion vector for each partition can take a significant number of bits, especially if small partition sizes are chosen. So much of the compressed data will consist of motion vectors. The efficiency with which they are coded has a great impact on the compression ratio. To reduce the number of bits required to code, the motion vectors are predicted from the previously coded motion vectors and the difference is encoded using the variable bit rate coding.

CONE OF INFLUENCE AND WAVELET FOOTPRINTS

For every row of the image, wavelet transform is performed and then the wavelet coefficients are expressed in terms of footprints from the cone of influence as explained below. The discrete wavelet transform is given as

$$\psi_{m,n}(t)=\frac{1}{2^{m/2}}\psi(2^{-m}t-n)\ m,n\in z \qquad (1)$$

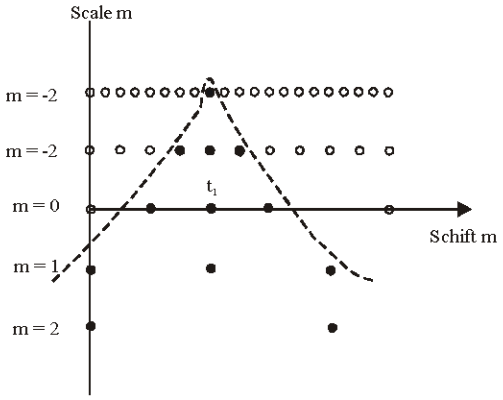

Fig. 1: Cone of influenceof t_1

The piecewise constant signal with one discontinuity at location t_1 and a wavelet series with one vanishing moment and compact support is considered for assumption. Wavelet transform of signal for scales m = -2 to 2 is shown in the Fig. 1.

At various scales the values are zero except on the positions where the discontinuity point t_1 influences. These non-zero coefficients constitute a cone. Thus the decomposition of this signal results in zero wavelet coefficients shown as white dots in Fig 1, which are not in the cone of influence of t_1.

The nonzero values come under the cone of influence which is shown as black dots. Cone of influence of t_1 in the scale space plane is the set of points (m, n) such that t_1 is included in the support of, $\psi_{m,n}(t)$, called as wavelet coefficients. The nonzero coefficient values that are extracted from the Cone of influence[6] of the polynomial discontinuity are known as wavelet footprints. The main property of the footprints is that they characterize efficiently the singular structures[6,9,10] of the signal, which usually carry the main information. Wavelet coefficients generated by a singularity are dependent across scales. Thus by representing any discontinuity with the combination of the footprints, we can get the sparser representation[12] of the signal under consideration.

NON LINEAR APPROXIMATION

A better scheme for approximation is to keep the best -M components in the signal expansion. This is referred to as non-linear approximation since it is adaptive and is based on the input signal[13,14]. Here the residue is approximated using NLA.

In general, the non-linear approximation can be written as

$$\hat{x}_M^{NLA} = \sum_{n\in I_M} c_n \psi_n$$

Where $\hat{x}_M^{NLA}$ is the approximated signal by non linear method with M retained coefficients, ψ_n is the wavelet coefficient and

$$c_n = \langle \hat{x}_M^{NLA}, \psi_n \rangle$$

The representation of the signal wavelet footprints explained in Fig. 2

Video Encoding and decoding Process Using Wavelet footprints: All luma and chroma samples are either encoded using transform coding or temporally predicted and the resulting prediction residual is encoded using transform coding. Each video sequence is dived into frames. In each frame Y,U, V components are separated and each component is subjected to wavelet footprint based based compression[15] and the bit streams obtained after compression are further compressed by using entropy coding methods.

Inter frame Prediction: The prediction signal for each predictive-coded M×N luma block is obtained by displacing an area of the corresponding reference picture, which is specified by a translational motion vector and a picture reference index. Thus, if the macroblock is coded using four 8x8 partitions and each 8x8 partition is further split into four 4×4 partitions, a maximum of sixteen motion vectors may be transmitted for a single P macroblock. The accuracy of motion compensation is in units of one half of the distance between luma samples. In case the motion vector points to an integer-sample position, the prediction signal consists of the corresponding samples of the reference picture; otherwise the corresponding sample is obtained using interpolation to generate non-integer positions. The prediction values at half-sample positions are obtained by applying a one-dimensional 6-tap FIR filter horizontally and vertically. The syntax supports multi-picture motion-compensated prediction. That is, more than one prior coded picture can be used as reference for motion-compensated prediction. Figure 3

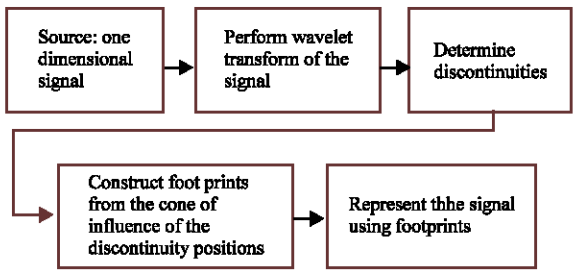

Fig. 2: Representation of signal using footprints

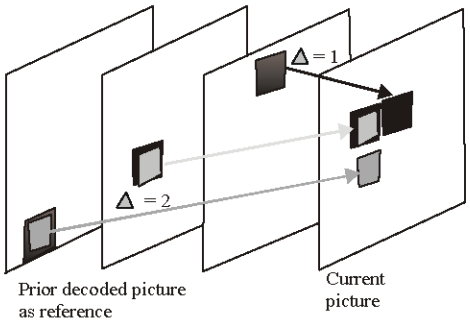

Fig. 3: Multi-refrence frame motion compensation

shows the concept. In addition to the motion vector, also picture reference parameters are transmitted. Multi-frame motion-compensated prediction requires both encoder and decoder to store the reference pictures used for inter prediction in a multi-picture buffer. The decoder replicates the multi-picture buffer of the encoder according to memory management control operations specified in the bit stream The reference index parameter is transmitted for each motion-compensated 16×16, 8×8, or 4×4 luma block. Motion compensation for 4×4 blocks uses the same reference index for prediction of all four blocks within the 8×8 region

Motion Estimation Algorithm: Full Search Algorithm (FSA) performs searching all the possible points in a search window exhaustively. Due to the high intensive computation required for the full search algorithm many fast motion algorithms has been proposed[16] over the last two decades to give a faster estimation with similar block distortion compared to the full search method. Many fast search algorithms such as Logarithmic Search Algorithm (LSA), Three Step Search (TSS), New Three Step Search (NTSS), Four Step Search (FSS), New Four Step Search (NFSS), Diamond Search (DS), Cross Search (CS), Cross Diamond Search (CDS), Predictive Search Algorithm (PSA) and Multilevel Successive Elimination Algorithm (MSEA) have been proposed.

The main aim of these fast search algorithms is to reduce the number of search points in the search window and hence the computations. One of the most important assumptions of all fast motion estimation algorithm is 'error surface is monotonic i.e., Block Distortion Measure (BDM) is the least at the center or the global minima of the search area and it increases monotonically as the checking point moves away from the global minima. New Cross Diamond Search (NCDS) algorithm (NCDS) which is based on the cross centre biased distribution characteristics[4] is employed here. The NCDS algorithm introduced half-way stop technique to achieve speed-up.

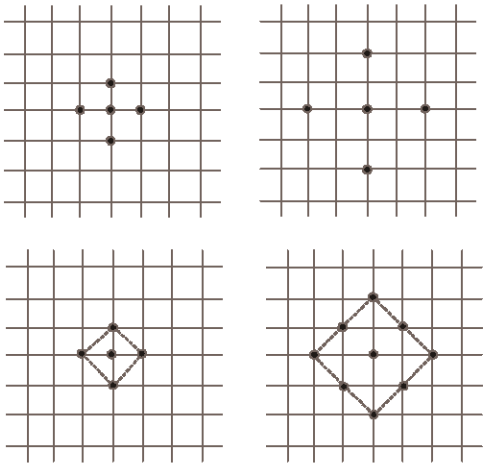

Fig. 4: Searching patterns used in NCDS algorithm (a) small cross shape (b) large cross shape (c) small diamond shape (d) large diamond shape

Instead of performing all the steps as in three Step Search or four step search, searching process is terminated at any step based on some criteria.

The NCDS Algorithm: The main difference between NCDS and Cross Diamond Search (CDS) is that the first 2 steps of NCDS are keeping a small cross shaped pattern which is saving the number of search point for stationary or quasi-stationary blocks. The various steps and the analysis of the algorithm are given below:

Step 1: A minimum BDM is found from the 5 search points of the Small Cross-Shaped Pattern (SCSP) shown in Error! Reference source not found..a located at the center of search window. If the minimum BDM point occurs at the center of the SCSP, the search stops and the minimum BDM point found is the final solution for the motion vector. Otherwise, go to next step.

Step 2: With the vertex i.e. minimum BDM point, in the first SCSP as the center, a new SCSP is formed. If the minimum BDM point occurs at the center of this SCSP, the search stops and the new minimum BDM point found is the final solution for the motion vector. Otherwise go to next step.

Step 3: The three unchecked outermost search points of the central – LCSP are checked, in which the step is trying to guide the possible correct direction for the subsequent steps. And then go to next step.

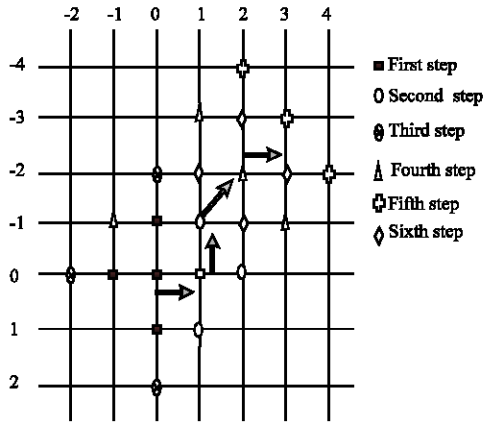

Fig. 5: Example of NCDS algorithm

Step 4: A new Large Diamond Search Pattern LDSP Error! Reference source not found..c is formed by repositioning the minimum BDM point 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 next step for converging the final solution; otherwise, this step is repeated.

Step 5: (Converging step): With the minimum BDM point in the previous step as the center, a SDSP is formed. The new minimum BDM point is found from 4 search points of the SDSP. The new minimum BDM point found is the final solution for the motion vector.

Example for the NCDS algorithm is shown in Fig. 5. The searching points in the each step are shown with different symbols. The minimal MAD point in each step is shown with grey color and also the direction of the convergence is shown by arrow marks. The final minimum MAD point which is the motion vector for the current block is (3, 2).

Half Pixel accuracy estimation: Candidate block at a fractional location usually gives better match than at an integer location. It can also reduce the degree of error between the original image and predicted image. The pixel values in these fractional candidate blocks are obtained by interpolating the nearest pixels at integer locations using 6-tap filtering.

Figure 6 shows the half sample interpolations for samples a, b and c. The sample values at half sample positions labeled a and b are derived by calculating intermediate values a1 and b1 respectively by applying the 6-tap filter as follows:

$$a1 = A-5B+20E+201-5J+$$
$$b1 = C-5D+20E+20F-5G+H$$

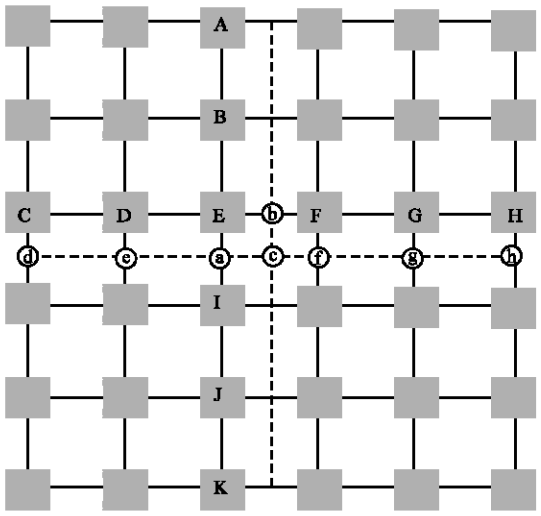

Fig. 6: Illustration of 6-tap filtering for interpolations

The final prediction values for locations a and b are obtained as follows and clipped to the range of 0 to 255.

$$a = (a1+16)>>5$$
$$a = (b1+16)>>5$$

The samples at half sample positions labeled as c is obtained from d, e, f, a, g and h in a manner similar to b.

However, this process increases the computational overhead for additional processes such as interpolation and MAD comparisons. Conventional encoders perform motion estimation in two steps to save computation: first they find the criterion minimum at an integer location. Then the candidate blocks corresponding to the eight nearest half-pixel displacements as shown in Fig. 7, to the best integer motion vector is interpolated and the motion vector is refined into sub-pixel accuracy by computing the criterion between the current block and the eight half-pixel candidate blocks. However, this requires much computation and may be difficult to perform in real-time encoders. Thus, faster methods have been investigated.

Fast two step method for half pixel accuracy motion estimation: In this method the direction of minimum error is predicted with reference to the predefined direction, so that it limits the number of candidate block to be searched. It reduces the computational overhead by limiting the number of interpolations of the candidate block. Here the mode of direction is to be chosen first. There are two modes

- Horizontal and vertical direction mode
- Two diagonal direction mode

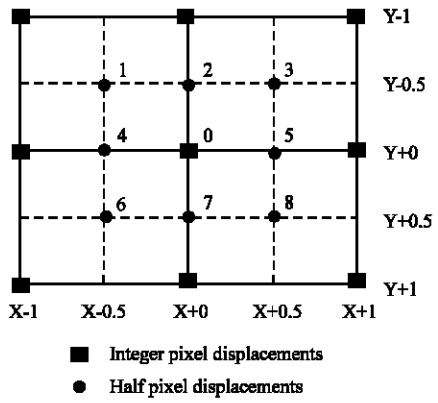

Fig.7: Integer and half pixel displacements

Then the MAD's of four reference half pixel candidates are calculated by interpolation. Then two neighboring reference candidates that have minimum MAD's among four possible candidate blocks is chosen and a motion block at a point between these two candidate block is interpolated. Finally a half pixel accuracy motion vector is obtained from a MAD comparison that is performed with an integer pixel point, two selected reference points and the point acquired in the previous step.

Horizontal and Vertical Directions as Reference (HVDR): Let the half pixel searching points include centre selected integer pixel points number 0-8 in Fig. 7 In this HVDR method points 2, 4, 5 and 7 shown in Fig. 7 are taken as reference directions. Points 2 and 7 are in vertical directions and points 4 and 5 are in horizontal directions. First these motion blocks are interpolated and MAD's of point 2 is compared with point 7 and point 4 with point 5. Then the candidate block with minimum MAD is chosen from the opposite pairs. Interpolation and MAD computation of the block point between chosen candidate pair are then performed to determine whether or not a minimum value is present. For example, consider that, point 2 has minimum MAD when compared with point 7 and point 4 has minimum MAD than point 5. So the minimum MAD pair is point 2 and point 4. Finally, the half pixel accuracy motion vector is determined by comparing the MAD's of centered integer block, the minimum candidate block and the most recently considered block.

Two Diagonal Directions as References (TDDR): The same procedure discussed above is applied here also but with different reference directions. Here the points 1, 3, 6 and 8 are used. The half pixel accuracy motion vector is obtained in similar way as discussed above.

1.5	1.0	-0.5	-1.5
1.0	1.5	2.0	1.5
0.5	-0.5	-0.5	-1.0
-0.5	-0.5	-1.0	-1.5

1.5	0.5	1.5	1.0
0.5	-0.5	-0.5	0.5
0.5	1.0	0.0	0.5
1.0	0.0	0.5	0.5

Fig. 8: (a) Actual motion vectors (b) Difference between the predicted and actual motion vectors

Predictive coding of Motion vector: The motion vector components are differentially coded using either median or directional prediction from neighboring blocks. No motion vector component prediction (or any other form of prediction) takes place across slice boundaries. A predicted vector, MVp, is formed based on previously calculated motion vectors. The method of forming the prediction MVp depends on the motion compensation partition size and on the availability of nearby vectors. MV_D, the difference between the current vector (MV) and the predicted vector MVp, is encoded and transmitted. The difference is encoded using variable bit length coding. Most frequently occurring motion vector are coded with short code. Figure 8 shows the actual motion vectors and the difference between the predicted one and the actual motion vectors.

Now the difference is encoded as:

- First bit represents the sign of the difference; negative difference is represented by 1 and positive as 0.
- Next to the sign bit is M ones followed by one zero; M is the absolute value of difference.
- Last bit represent the decimal value; 0.5 is represented as 1 and 0.0 is represented as 0
  For example, -1.0 and 0.5 are coded as

-1.0 → 1100
0.5 → 001

RESULTS AND DISCUSSION

Two videos namely Clarie (144×176) and Dancer (288×352) is taken for our experiment. The reason for selecting these videos are the first video Clarie contains less motion from frame to frame and the dancer video contains abrupt change between frame to frame. We have considered each frame as a still image and each frame is video has still background and has very less motion of only one object. The PSNR values for Claire image sequences for different thresholds are given in the Table 1 to 3. The first four reconstructed frames are shown in Fig. 9. A good compression ratio is achieved here. The average PSNR and CR is approximately same for all the three thresholds and it varies based on the percentage of NLA coefficients retained. From the above

Table 1: Comparison of PSNR and CR values of Claire video for Th:100

Claire-Th = 100 Frameno	3% NLA PSNR(dB)	7% NLA PSNR(dB)	12% NLA PSNR(dB)
0	35.74	39.41	41.64
1	35.28	38.47	39.96
2	35.55	38.64	39.70
3	37.92	41.09	42.94
4	37.93	40.65	42.35
5	37.71	40.30	41.70
6	38.22	41.13	42.93
7	38.08	40.87	42.14
8	37.94	40.34	41.63
9	38.27	41.13	42.88
10	38.17	40.80	42.25
11	38.15	40.69	41.91
AVG.PSNR	37.41	40.29	41.83
CR	27.32	15.44	9.97

Table 2:Comparison of PSNR and CR values of Claire video for Th: 150

Claire-Th = 150 Frame No.	3% NLA PSNR(dB)	7% NLA PSNR(dB)	12% NLA PSNR(dB)
0	36.13	39.90	42.11
1	36.14	39.03	40.40
2	36.36	39.11	40.04
3	38.17	41.00	42.91
4	38.13	40.78	42.27
5	37.86	40.50	41.67
6	38.45	41.04	42.90
7	38.32	40.76	42.13
8	38.21	40.39	41.60
9	38.49	41.04	42.88
10	38.37	40.79	42.20
11	38.36	40.73	41.86
AVG.PSNR	37.84	40.42	41.91
CR	27.61	15.48	9.98

Table 3: Comparison of PSNR and CR values of Claire video for Th: 200

Claire-Th = 150 Frame No.	3% NLA PSNR(dB)	7% NLA PSNR(dB)	12% NLA PSNR(dB)
0	PSNR	PSNR	PSNR
1	36.4165	40.1887	42.3753
2	36.6785	39.3578	40.9799
3	36.9015	39.2920	40.4892
4	38.2356	41.0198	42.9309
5	38.1895	40.7857	42.2672
6	37.8482	40.4515	41.8442
7	38.4241	41.0181	42.9105
8	38.2219	40.7724	42.2176
9	38.0395	40.3224	41.3959
10	38.4474	41.0300	42.8406
11	38.3667	40.6261	42.0632
AVG.PSNR	38.3608	40.46	41.85
CR	37.84	40.44	42.01

Fig. 9: Four compressed frames of CLAIRE video

Table 4: Comparison of PSNR and CR values of dancer video for Th: 100

Dancer Th = 100 FRAME NO	3%NLA PSNR(dB)	7% NLA PSNR(dB)	12% NLA PSNR(dB)
0	39.56	41.40	42.25
1	36.53	37.98	38.57
2	35.85	37.03	37.67
3	42.45	44.30	45.09
4	39.28	39.92	40.11
5	37.93	38.58	38.65
6	42.65	44.51	45.26
7	38.78	39.09	39.20
8	36.98	37.28	37.23
9	42.70	44.52	45.20
10	39.48	39.89	39.91
11	37.83	38.29	38.32
AVG.PSNR	39.16	40.23	40.62
CR	26.60	15.11	9.80

Table 5: Comparison of PSNR and CR values of Dancer video for Th: 150

DANCER Th=150 FRAME NO	3% NLA PSNR(dB)	7% NLA PSNR(dB)	12% NLA PSNR(dB)
0	41.12	43.15	44.05
1	38.01	39.40	39.79
2	37.20	38.19	38.37
3	42.79	44.46	45.18
4	39.74	40.03	40.03
5	38.25	38.63	38.66
6	42.78	44.50	45.34
7	38.74	39.11	39.32
8	36.90	37.31	37.34
9	42.67	44.56	45.41
10	39.39	39.89	40.12
11	37.99	38.32	38.47
AVG.PSNR	39.63	40.63	41.00
CR	26.71	15.13	9.81

mentioned Tables 1 to 3 it is seen that high compression ratio is achieved for less NLA coefficients with the expense of 4 dB reduction in PSNR compared to 12% of NLA coefficients for the same threshold.

Table 6: Comparison of PSNR and CR values of Dancer video for Th: 200

Dancer Th = 200 Frame No.	3% NLA PSNR(dB)	7% NLA PSNR(dB)	12% NLA PSNR(dB)
0	41.58	43.49	44.31
1	38.79	39.71	39.92
2	37.61	38.41	38.50
3	42.85	44.51	45.20
4	39.65	40.18	40.18
5	38.25	38.72	38.77
6	42.78	44.57	45.33
7	38.82	39.10	39.27
8	36.97	37.36	37.36
9	42.73	44.58	45.36
10	39.43	39.93	40.00
11	39.93	38.33	38.39
AVG.PSNR	39.95	40.74	41.04
CR	26.18	15.12	9.08

Fig. 10: Four compressed frames of dancer video

Dancer video has fast moving background and small motions of two moving objects. The PSNR values for Dancer image sequences for different thresholds are given in the Table 4 to Table 6. The first four reconstructed frames are shown in Fig 10. A good compression ratio of approximately 26 and PSNR of 39 db is achieved here. The average PSNR and CR is approximately same for all the three thresholds and it varies based on the percentage of NLA coefficients retained. From the mentioned Tables 4 to 6 it is seen that compression ratio gets increased by 17 times with the reduction of 2 db in PSNR if we go from 3 to 12% NLA coefficients for the same threshold.

Figure 11 and 12 represents the comparison of CR values for both the videos under different NLA Coefficients retained. The CR for both the videos is almost same when the threshold is low and it is

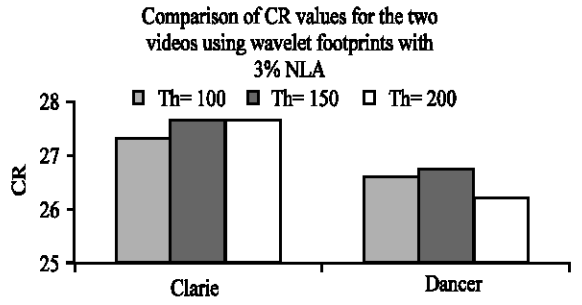

Fig. 11: Comparison of Compression Ratios for the two videos with 3% NLA coefficients retained

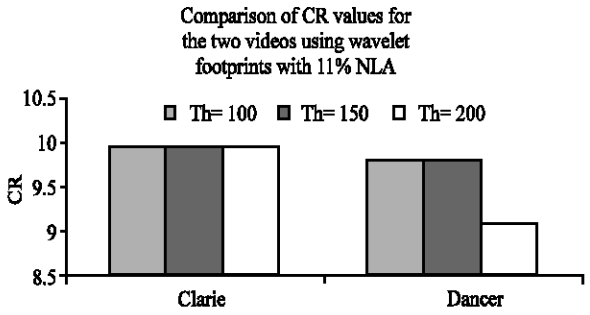

Fig. 12: Comparison of Comparison ratios for the two videos with 11% NLA coefficients retained

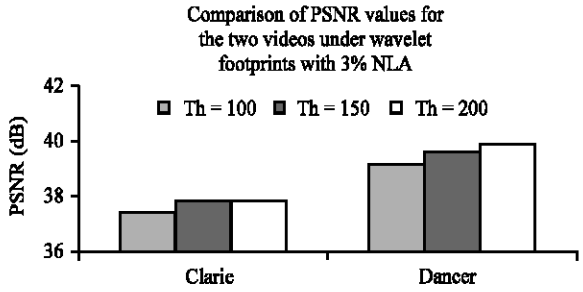

Fig. 13: Comparison of PSNR for the two videos with 3% NLA coefficients retained

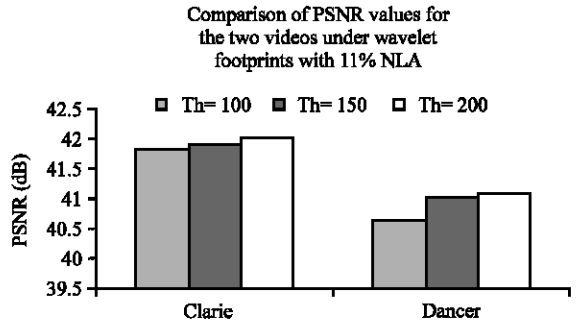

Fig. 14: Comparison of PSNR for the two videos with 11% NLA coefficients retained

small for Dancer video as the threshold increases. Fig. 13 and 14 shows the corresponding comparison for PSNR values.

CONCLUSION

This design is based on conventional block-based motion-compensated hybrid video coding concepts, but with some important differences relative to prior standards. Some of the important differences are summarized below

- Enhanced motion prediction capability
- Use of Wavelet footprints
- Adaptive quantization Scheme

When used well together, the features of the new design provide a compression ratio of approximately 25 to 30 for equivalent perceptual quality relative to the performance of prior standards.

REFERENCES

1. Rao, K.R. and J.J. Hwang, 1996. Techniques, standards for image, video and audio coding, Prentice Hall, PTR.
2. Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard and Ajay Luthra, 2003. Overview of the H.264/AVC video coding standard, IEEE Trans. Circuits Syst. Video Technol., pp: 243-250
3. Rao, K.R. and P. Yip, 1990. Discrete Cosine Transforms-Algorithms, Advantages, Applications, Academic Press.
4. Chi-Wai La, Lai-Man Po and Chun Ho Cheung, 2000. A new cross-diamond search algorithm for fast block matching motion estimation. IEEE Trans. On Image Processing, 9: 287-290.
5. Soman, K.P. and K.I. Ramachandran, 2004. Insight into Wavelets from Theory to Practice. Prentice Hall of India, New Delhi.
6. Mallat, S., 1998. A Wavelet Tour of Signal Processing Academic Press, New York.
7. Vetterli, M. and J. Kovacevic, 1995. Wavelets and Subband Coding Englewood Cliffs, N.J., Prentice Hall,
8. Dragotti, P.L. and M. Vetterli, 2003. Wavelet Footprints: Theory Algorithms and Applications. IEEE Transactions on Signal Processing, 51.
9. Dragotti, P.L. and M. Vetterli, 2000. Wavelet Transform Footprints: Catching Singularities for compression and Denoising. IEEE Intl. Conf. on Image Processing, Vancouver, Canada.

10. Hwang, W.L. and S. Mallet, 1992. Singularity detection and processing with waveletes. IEEE Trans. Inform. Theory, 38: 617-643
11. Mallat, S. and Z. Zhang, 1993. Matching pursuit with time-frequency dictionaries. IEEE Trans. Signal Processing, 41: 3397-3415
12. Elad, M. and A.M. Bruckstein, 2001. Bruckstein On sparse signal representations. IEEE Intl. Conf. Image Processing Thessaloniki, Greece.
13. DeVore, R.A., 1998. Nonlinear approximation. Acta Number, 7: 51-150.
14. DeVore, R.A., B. Jawerth and B.J. Lucier, 1992. Image compression through wavelet transforms coding. IEEE Transactions on Information Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis, 38: 719-746.
15. Sudhakar, R., N. Vignesh, S. Jayaraman, 2005. Application of Wavelet footprints for Fingerprint Compression. International Journal on Graphics Vision and Image processing, 5: 39-45
16. Archana, Rao and Manasa Raghavan, 2003. Fast Motion Estimation Algorithms computation and performance Trade-Offs. Course project fall.