Turk J Elec Eng & Comp Sci, Vol.20, No.1, 2012, © TÜBİTAK
doi:10.3906/elk-1101-1014

A video encoder design combining edge-adaptive lifting and scalable block-matching with motion compensation

Sedat TELÇEKEN[1,*], **Ömer Nezih GEREK**[2]
[1]*Department of Computer Engineering, Anadolu University, 26470 Eskişehir-TURKEY*
e-mail: stelceken@anadolu.edu.tr
[2]*Department of Electrical and Electronics Engineering, Anadolu University, 26470 Eskişehir-TURKEY*
e-mail: ongerek@anadolu.edu.tr

Received: 13.01.2011

Abstract

This study aimed to achieve video compression by using a novel lifting-based hybrid encoder that also uses motion compensation. The proposed encoder separates video frames into temporal groups, within which certain frames are selected for producing temporal and spatial predictions over the rest of the frames. The predictions utilize spatiotemporal lifting together with motion compensation. The combination of spatial information with temporal changes is inspired from the idea of edge-adaptive lifting, which alters prediction directions in images. A further incorporation of well-known block-matching methods with different levels was observed to improve the performance. To provide the first compression results, unpredicted frames and residues between the predicted frames and their originals were quantized and entropy was encoded. Peak signal-to-noise ratio (PSNR) measurements were used to measure the performance analysis of the proposed encoding method. Experimental results revealed that higher compression ratios were obtained as the block-matching level was increased, while the PSNR values remained within reasonable ranges. PSNR values of around 29 dB were obtained with a 1000:27 compression ratio.

Key Words: *Video signal processing, video coding, video compression, motion compensation, encoding*

1. Introduction

In this study, a video signal decomposition method that combines motion compensation (MC) and 3-dimensional (3D) lifting-based prediction techniques was explored. Following brief descriptions of these techniques, 2 alternative methods for the incorporation of motion compensation are proposed here. These techniques decompose the frames of video sequences in the temporal domain. Some (intratype) frames are encoded directly and the rest of the frames within the same group are predicted from intraframes. The predicted frames are then encoded by the 3-dimensional set partitioning in hierarchical trees (3D-SPIHT) algorithm [1,2]. The peak signal-to-noise ratio (PSNR) values of the experimental video sequences are presented and the results are discussed.

*Corresponding author: Department of Computer Engineering, Anadolu University, 26470 Eskişehir-TURKEY

The aim of a video encoder is to decrease the information in all video signals by removing the redundant information, which cannot be perceived by the human visual system [3,4]. Compression of video data without significant distortion of the visual quality is usually possible, since video sequences contain a high degree of redundancy. These redundancies can be classified as:

- spectral redundancy, which is related with correlation among the color components;
- spatial components, which are related between the neighboring pixels;
- temporal redundancy, which is related with the correlation among the group of frames; and
- statistical redundancy, which is related with correlation among the symbols in bit planes [5].

The motivation for the present work was the fact that both MC and temporal lifting predictions seek the minimum residual signal variance for predicted frames [6-8]. Clearly, the direct application of temporal lifting does not take into account any kind of motion for the consecutive frames. Recently, for the temporal subband decomposition of video signals, Pesquet-Popescu and Bottreau improved the compression efficiency by using a lifting scheme with MC [6,9]. In those studies, both adaptive subband decomposition and linear subband decomposition methods were tested. In this study, a third decomposition method is proposed, in which the temporal decomposition is combined with spatial information through the edge-adaptation strategy of the 2D lifting method, which was proposed in [10]. Since the aim of the paper is to introduce the idea of combining spatio-temporal prediction within a lifting framework with motion compensation, explanations are provided and only simple coding results are presented.

Specifically, in Section 2, we focus on the proposed frame prediction methods. Experimental works and initial results are given in Section 3. Some concluding remarks are given in Section 4.

2. The proposed frame prediction methods

The 3D lifting scheme used in this study and 2 different included MC methods are described in this section. Instead of focusing on the center pixels of blocks after block-matching in MC, as was described in [9], the entire block is dealt with after block-matching for directional spatio-temporal prediction.

2.1. Edge-adaptive lifting method with motion compensation

In a temporal lifting strategy, one stage of lifting starts by assigning even- and odd-numbered frames as predicted frames and intraframes. In a complete system, the prediction error is also encoded according to a desired compression ratio. The SPIHT algorithm is used for compressing the prediction residual, due to its flexibility in achieving a target bit rate. Clearly, more dedicated 2D codecs or entropy coders could be achieved in a more detailed study, but that detail level is outside the scope of this work.

Let $x_i[m,n]$ represent the pixel located at coordinates $[m,n]$ of frame i of video signal x. Let the center pixel of any cubic block (over the 3D video stream) of $3\times3\times3$ be estimated. The front 3×3 side of this cubic block belongs to frame $(2i-1)$, and the rear 3×3 side belongs to frame $(2i+1)$. Hence, between those sides, there exists an intermediate 3×3 layer that includes the center pixel to be estimated. These are illustrated in Figures 1-3.

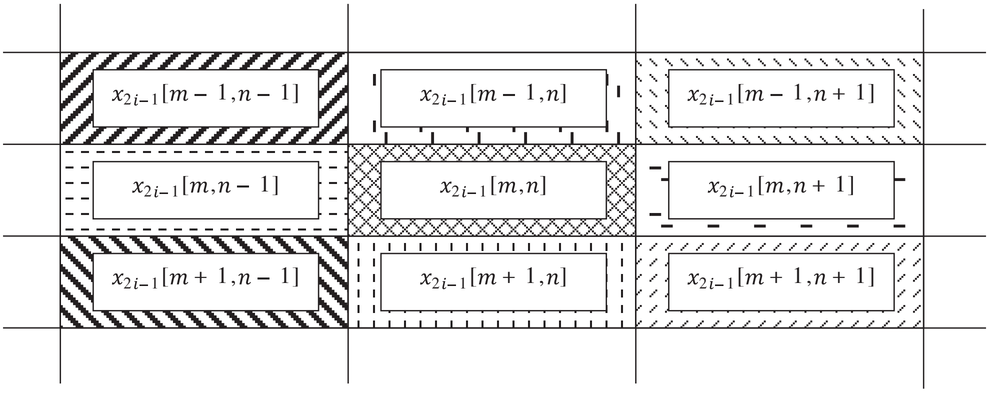

Figure 1. 3×3 front block side in frame $(2i - 1)$.

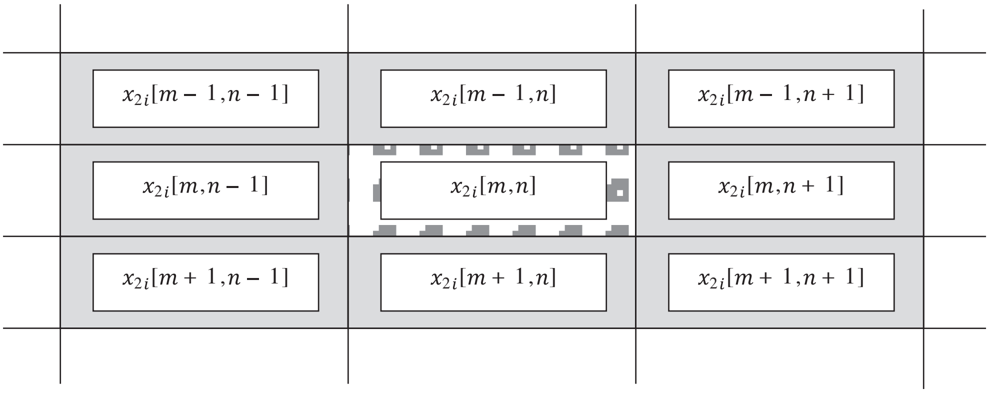

Figure 2. 3×3 intermediate block layer in frame $(2i)$.

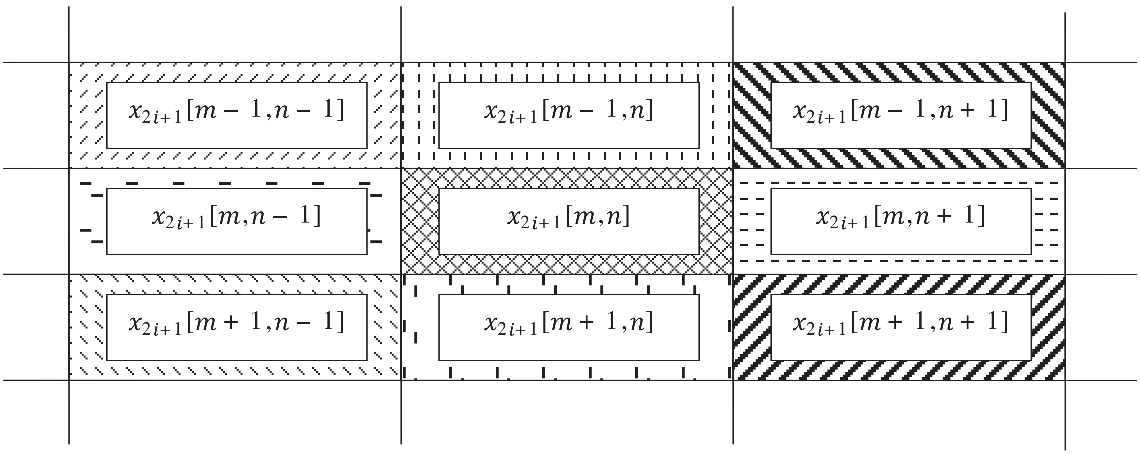

Figure 3. 3×3 rear block side in frame $(2i + 1)$.

The value of $x_{2i}[m, n]$ is estimated by the previous layer's and next layer's pixel couples, which are centered around the $x_{2i}[m, n]$ pixel.

We define 17 different gradient approximations around $x_{2i}[m, n]$. Eight of these that are within the same frame can be omitted, because they do not use the intraframes for prediction, which would spoil the symmetry of the decoder part. The remaining valid gradient approximations are given below.

$$\Delta_1 = |x_{2i-1}[m-1, n-1] - x_{2i+1}[m+1, n+1]|$$

$$\Delta_2 = |x_{2i-1}[m-1, n] - x_{2i+1}[m+1, n]|$$

$$\Delta_3 = |x_{2i-1}[m-1, n+1] - x_{2i+1}[m+1, n-1]|$$

$$\Delta_4 = |x_{2i-1}[m, n-1] - x_{2i+1}[m, n+1]|$$

$$\Delta_5 = |x_{2i-1}[m, n] - x_{2i+1}[m, n]|$$

$$\Delta_6 = |x_{2i-1}[m, n+1] - x_{2i+1}[m, n-1]|$$

$$\Delta_7 = |x_{2i-1}[m+1, n-1] - x_{2i+1}[m-1, n+1]|$$

$$\Delta_8 = |x_{2i-1}[m+1, n] - x_{2i+1}[m-1, n]|$$

$$\Delta_9 = |x_{2i-1}[m+1, n+1] - x_{2i+1}[m-1, n-1]|$$

These gradients are illustrated in Figure 4.

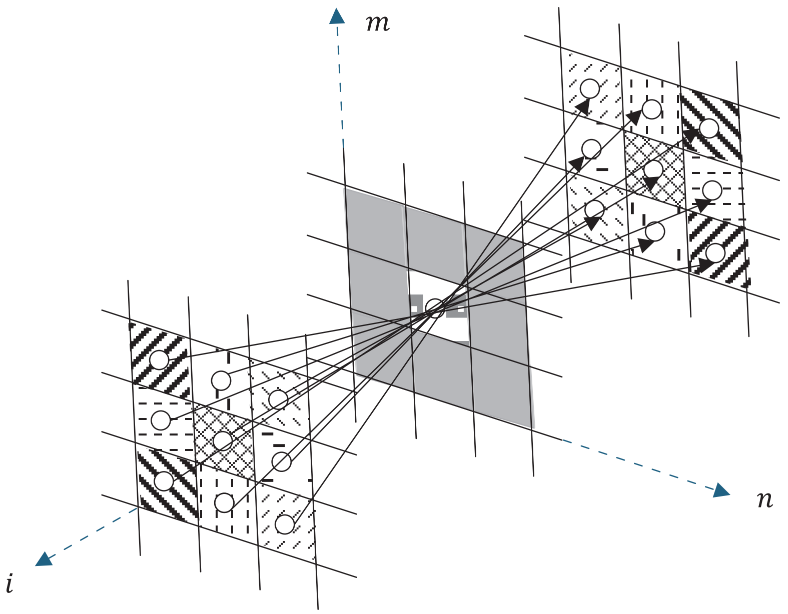

Figure 4. Gradient approach on consecutive 3×3 block of odd frames.

Nine different $x_{2i}[m, n]$ estimation values can be obtained from these expressions, as listed below.

$$\hat{x}_{2i}^{(1)}[m, n] = (x_{2i-1}[m-1, n-1] + x_{2i+1}[m+1, n+1])/2$$

$$\hat{x}_{2i}^{(2)}[m, n] = (x_{2i-1}[m-1, n] + x_{2i+1}[m+1, n])/2$$

$$\hat{x}_{2i}^{(3)}[m, n] = (x_{2i-1}[m-1, n+1] + x_{2i+1}[m+1, n-1])/2$$

$$\hat{x}_{2i}^{(4)}[m, n] = (x_{2i-1}[m, n-1] + x_{2i+1}[m, n+1])/2$$

$$\hat{x}_{2i}^{(5)}[m,n] = \left(x_{2i-1}\left[m,n\right] + x_{2i+1}\left[m,n\right]\right)/2$$

$$\hat{x}_{2i}^{(6)}[m,n] = \left(x_{2i-1}\left[m,n+1\right] + x_{2i+1}\left[m,n-1\right]\right)/2$$

$$\hat{x}_{2i}^{(7)}[m,n] = \left(x_{2i-1}\left[m+1,n-1\right] + x_{2i+1}\left[m-1,n+1\right]\right)/2$$

$$\hat{x}_{2i}^{(8)}[m,n] = \left(x_{2i-1}\left[m+1,n\right] + x_{2i+1}\left[m-1,n\right]\right)/2$$

$$\hat{x}_{2i}^{(9)}[m,n] = \left(x_{2i-1}\left[m+1,n+1\right] + x_{2i+1}\left[m-1,n-1\right]\right)/2$$

The straight temporal prediction value (i.e., $\hat{x}_{2i}^{(5)}[m,n] = \left(x_{2i-1}\left[m,n\right] + x_{2i+1}\left[m,n\right]\right)/2$) is the classical estimation, which is obtained by the lifting prediction on the $3 \times 3 \times 3$ cubic block along the temporal direction. In the case of no motion gradient, this value would give us the minimum difference between the estimated and actual pixel value. However, if any motion gradient occurs near the centered pixel on the video sequence and if this causes a 3D edge (a tilted optical flow) within the cube boundaries, then another cross pixel prediction value may be closer to the center pixel's actual value. Hence, pixel couples in the corresponding direction will give better results. This selection method introduces a direction adaptive filter, just like the 2D version described in [10]. This adaptive filter still uses pixel couples for estimation and finds the direction that holds the pixel couples with the minimum difference. It must be noted that the minimum gradient direction is selected according to the "decoded" frames; therefore, no side information needs to be sent to the decoder, meaning that the operation induces no extra cost.

2.2. Block-matching method in motion compensation

In the gradient estimation method described in Section 2.1, the internal 3×3 side of a $3 \times 3 \times 3$ cubic block is located by matching the front side in frame $(2i-1)$ to the rear side in frame $(2i+1)$, and the value of central pixel $x_{2i}[m,n]$ of this internal layer is estimated as illustrated in Figure 5. Due to the complications that arise when a central coordinate is never rendered by this method, an alternative method was developed, which makes sure that each and every pixel of the prediction frame is processed. In this alternative method, the location of the internal layer is kept as a reference point and, during the block-matching search, matched blocks are moved in symmetrically opposite directions, making the center location fixed. During the matching operation, the sum of squared differences within a block is used as the matching metric. The symmetric block-wise search operation is performed for each and every pixel of the prediction frame $(2i)$. Despite the analogy of the method to classical (MPEG-like) block-matching, keeping the center frame's position constant is a substantially different operation.

Several search block sizes $(3 \times 3,\ 5 \times 5,\ 7 \times 7,..)$ were experimentally tested to achieve an empirically good performance. Results of different block sizes were given in [6].

2.3. Block-matching methods with motion compensation in different levels

The even frames of a video sequence were predicted by the odd frames using the block-matching method that was described in Section 2.2. This method is denoted as *Level1 Block Matching Method*.

Let us extend the temporal decomposition described in Section 2.2 to a 3-level temporal decomposition. In Level2, frames 1 and 5 are used to predict frame 3, frames 5 and 9 are used to predict frame 7, and so on. At the end, all odd frames are obtained, and then even frames are predicted from the odd frames.

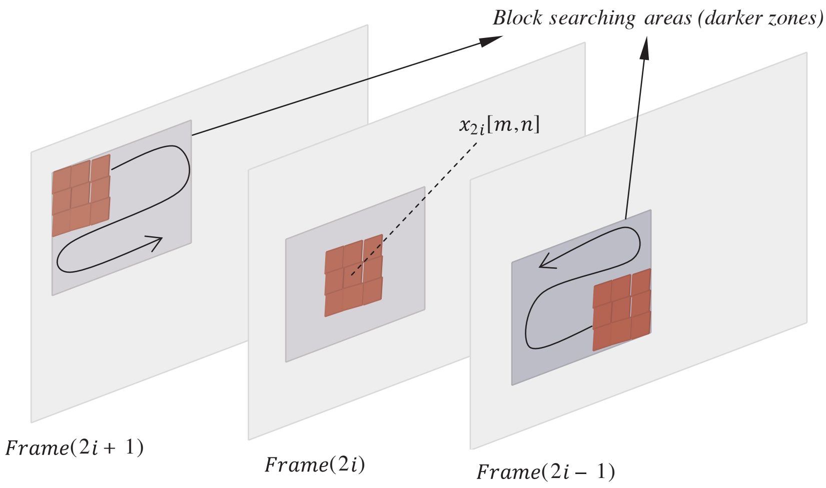

Figure 5. Locating source blocks in block-matching.

In Level3, frames 1 and 9 are used to predict frame 5, frames 9 and 17 are used to predict frame 13, and so on. In Step 2, frames 1 and 5 are used to predict frame 3, frames 5 and 9 are used to predict frame 7, and so on. In the last step, the even frames are predicted from the odd frames (as in Section 2.2). These levels of decomposition are illustrated in Figure 6.

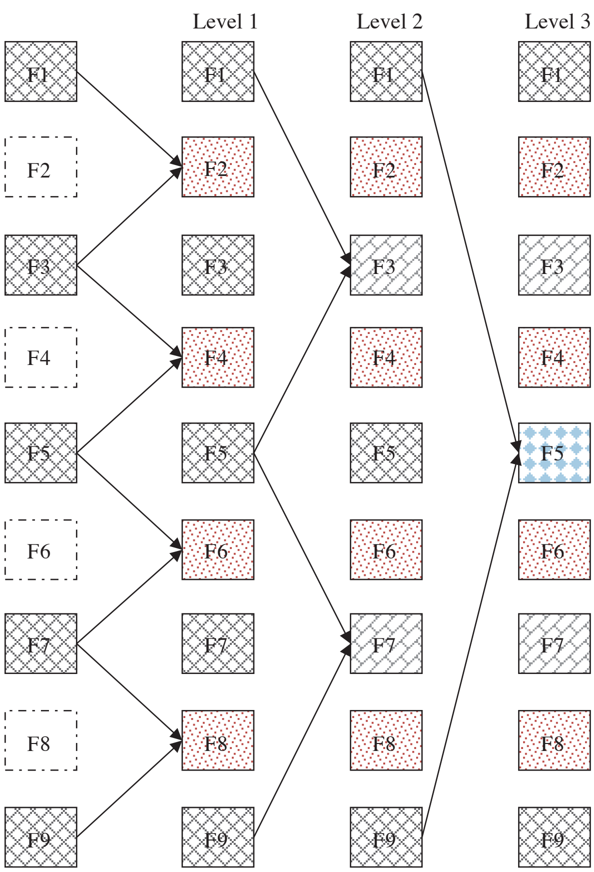

Figure 6. Level1, Level2 and Level3 prediction.

3. Experimental work and results

A significant portion of the compression success can be achieved by the selection of the quantizer and the entropy coding method. Several coding standards were achieved as a result of long fine-tuning processes. Since the aim of this work was to introduce a hybrid processing idea, extensive experimentation on the quantization and entropy coding of the residual frames was left outside of the current scope. Instead, a fair image coding method, namely SPIHT, was adopted [2]. The reason for choosing SPIHT was mainly its ability to meet any arbitrary target bit rate (up to lossless), making it very suitable for experimenting with different overall bit rates in the proposed video codec. Consequently, comparative rate/distortion experiments could be conducted with the celebrated MPEG2 and H.264 standards.

Several experiments were performed on 145-frame "bus," "coastguard," and "container" YUV-formatted and CIF (352×288)-sized videos.

3.1. Experimental results of edge-adaptive lifting method with motion compensation

The Level1 decomposition and reconstruction is illustrated in Figure 7. Odd frames of video sequences are SPIHT-coded by a certain bit per pixel value, "bpp1." After this coding stage, reconstructed frames $ReC(2i-1)$ are obtanied. This makes the coder and decoder sides symmetric, making it possible to avoid transmission of any extra side information. Even frames are predicted with the proposed method by using consecutive reconstructed frames. Residual frames, which correspond to the difference between original even frames and predicted even frames, are encoded at the target rate of the "bpp2" value. The output of this process gives us $ReC(2i)$ values.

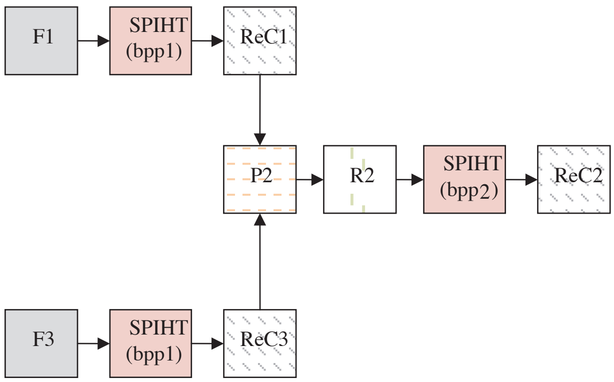

Figure 7. Decomposition and reconstruction of Level1.

Classical estimation of the pixel value obtained by the lifting prediction and the edge-adaptive lifting, which is essentially the proposed method, was tested on 3 different video sequences. Performances of these methods in terms of mean PSNR values were evaluated over the Y channels, and hence the name YSNR. In Table 1, a total of 145 frames of 3 sequences of 30 FPS are used. Different "bpp" values for different temporally polyphase frames were tested for each experiment. In other words, intraframes, or odd frames, were SPIHT-coded to the target bit rate of bpp1, whereas prediction residuals (even frame residuals) were SPIHT-coded to a target bit rate of bpp2. The overall corresponding compression ratio (CR) values were analyzed. A similar strategy of altering target bit rates via tuning the SPIHT coder was adopted for comparing different hybrid combinations at various decomposition levels. Notice that, for 2 levels, 3 different polyphase frames are encoded at bpp1, bpp2, and bpp3, whereas for 3 levels, 4 different polyphase frames are tested with 4 different bpp variations. In this table, YSNR1 corresponds to the application of temporal lifting without 3D edge-adaptive gradient consideration, and YSNR2 corresponds to the incorporation of the proposed 3D edge gradient.

Table 1. YSNR values (in dB) of 3 different videos.

				bus		coastguard		container	
Test No.	bpp1	bpp2	CR	YSNR1	YSNR2	YSNR1	YSNR2	YSNR1	YSNR2
1	1	1	8:1	19.30	19.79	21.24	21.70	24.04	24.51
2	1	0.75	9.09:1	18.69	19.14	20.68	21.29	23.65	24.33
3	1	0.5	10.64:1	17.44	17.91	20.00	20.71	23.28	24.12
4	1	0.25	12.66:1	16.11	16.87	19.11	19.94	22.38	23.83
5	1	0.1	14.49:1	14.93	15.61	18.34	19.18	21.31	23.71

3.2. Experimental results of block-matching method in motion compensation

In this alternative hybrid method, test video frames were coded with SPIHT at target bit rates of bpp1 and bpp2. This time, the block-matching method was applied for the incorporation of MC. Consequently, Tables 2-4 provide results for YSNR1 and YSNR2 together with MC. The obtained YSNR2 values for the corresponding bpp combinations are given in Figure 8. Notice that the slowly varying nature of the "container" video does not benefit from MC.

The Level2 decomposition and reconstruction algorithm used in the SPIHT encoder are illustrated in Figure 9. In this level, video frames were coded by 3 different bpp values (bpp1, bpp2, and bpp3), according to their temporal polyphase situation. The methods were then applied to test videos. Test results are given in Table 3. YSNR1 and YSNR2 values were calculated similar to those in the method in 3.1, but the MC operation was added. YSNR3 corresponds to the direct application of the SPIHT codec and YSNR4 corresponds to the MPEG2 coding results at a close overall bit rate. The YSNR2 graph is given in Figure 10 with an axis corresponding to different bpp combinations, as indicated in Table 3. It is noteworthy that the distribution of bits differently to polyphase frames may produce somewhat different performances.

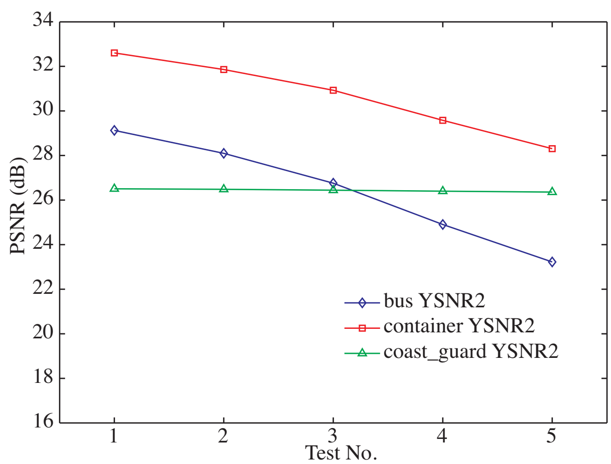

Figure 8. YSNR2 graphics of Level1 test.

Table 2. YSNR values of block matching method in Level1.

				bus				coastguard				container			
Test No.	bpp1	bpp2	CR	YSNR1	YSNR2	YSNR3	YSNR4	YSNR1	YSNR2	YSNR3	YSNR4	YSNR1	YSNR2	YSNR3	YSNR4
1	1	1	8:1	21.63	29.13	21.79	30.86	24.10	32.61	24.81	33.16	24.61	26.51	25.30	34.95
2	1	0.75	9.09:1	20.77	28.09	21.49	29.99	23.53	31.86	24.50	31.99	23.59	26.48	24.91	33.74
3	1	0.5	10.64:1	19.78	26.76	21.07	27.60	22.81	30.93	24.15	30.52	23.35	26.44	24.52	31.99
4	1	0.25	12.66:1	18.40	24.90	20.68	25.80	21.79	29.58	23.76	28.73	22.01	26.40	24.09	29.90
5	1	0.1	14.49:1	17.05	23.22	20.41	24.59	20.64	28.31	23.45	27.45	21.47	26.36	23.70	28.62

Table 3. YSNR values of block matching method in Level2.

					bus				coastguard				container			
Test No.	bpp1	bpp2	bpp3	CR	YSNR1	YSNR2	YSNR3	YSNR4	YSNR1	YSNR2	YSNR3	YSNR4	YSNR1	YSNR2	YSNR3	YSNR4
1	1	0.75	0.75	9.43:1	21.09	26.91	21.40	29.35	23.81	30.59	24.42	34.27	25.85	28.24	24.84	34.72
2	1	0.75	0.5	10.64:1	20.21	25.81	21.07	28.01	23.14	29.73	24.15	30.99	25.80	28.19	24.53	33.59
3	1	0.75	0.25	12.35:1	18.94	24.18	20.73	26.44	22.19	28.52	23.82	30.43	25.74	28.13	24.16	32.06
4	1	0.75	0.1	13.51:1	17.77	22.70	20.54	25.71	20.75	26.68	23.62	28.80	25.67	28.06	23.90	30.99
5	1	0.5	0.5	11.36:1	19.75	25.21	20.91	27.12	22.50	28.92	24.01	30.25	25.48	27.85	24.39	33.06
6	1	0.5	0.25	13.33:1	18.58	23.73	20.57	25.74	21.64	27.79	23.65	29.01	25.42	27.79	23.94	31.14
7	1	0.5	0.1	14.71:1	17.21	22.37	20.37	25.02	20.82	26.77	23.42	28.21	25.36	27.72	23.65	30.00
8	1	0.25	0.25	14.29:1	17.90	22.86	20.43	25.25	20.81	26.75	23.49	28.39	24.40	26.67	23.74	30.43
9	1	0.25	0.1	16.39:1	16.93	21.64	20.19	24.21	20.06	25.80	23.17	27.12	26.13	28.56	23.24	28.79
10	1	0.1	0.1	17.24:1	16.34	20.88	20.11	23.75	19.26	24.78	23.06	26.73	25.43	27.80	23.01	28.19

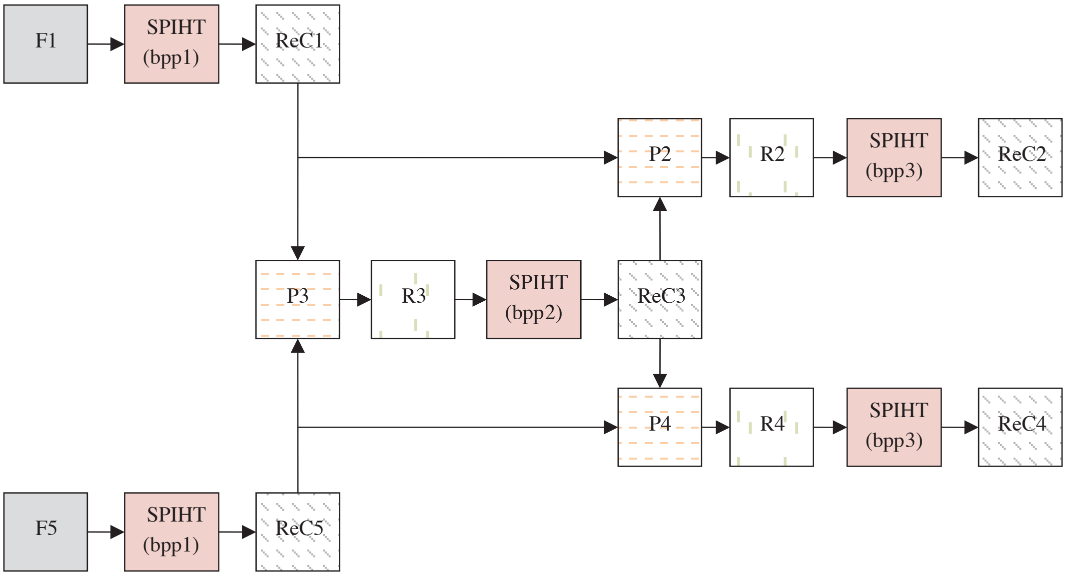

Figure 9. Level2 Decomposition and reconstruction algorithm.

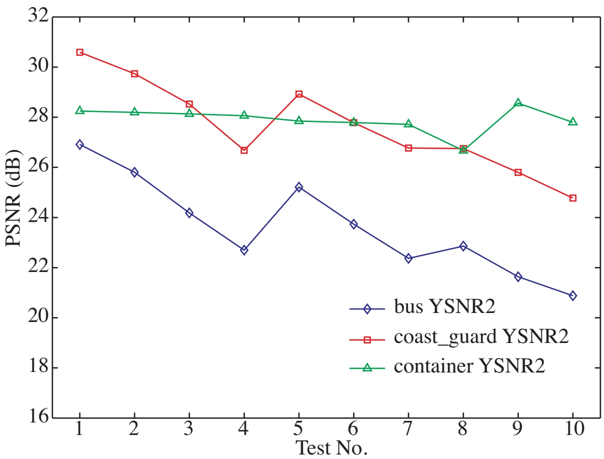

Figure 10. YSNR2 graphics of Level2 test.

The last test, for Level3 decomposition, used 4 different bpp combinations for 4 different temporal polyphase type frames. Several combination test results for the test videos are given in Table 4. YSNR2 values of the tests are plotted in Figure 11 with different bpp combinations given in accordance with the order presented in Table 4.

In order to provide a comparative study, MPEG2 and H.264 compression results of the same test sequences are provided at available bit rates in Table 5. Results from H.264 are given for comparison with the state-of-the-art standards. It must be noted that the overall encoder of the proposed method is nowhere near as optimized as the overly studied and fine-tuned MPEG and H.264 standards. The results indicate that the proposed method is relatively strong at lower bit rates and fairly promising in general.

Table 4. YSNR-values-of-block-matching-method-in-Level3.

						bus				coastguard				container			
Test No.	bpp1	bpp2	bpp3	bpp4	CR	YSNR1	YSNR2	YSNR3	YSNR4	YSNR1	YSNR2	YSNR3	YSNR4	YSNR1	YSNR2	YSNR3	YSNR4
1	1	0.75	0.75	0.75	10.20:1	20.64	25.98	21.17	28.85	23.27	28.70	24.24	31.02	26.38	27.90	24.62	33.21
2	1	0.75	0.75	0.5	12.20:1	19.69	25.03	20.76	26.51	22.76	28.07	23.84	29.92	26.17	27.85	24.19	31.70
3	1	0.75	0.75	0.25	14.93:1	18.56	23.60	20.35	25.03	22.02	27.16	23.38	27.68	26.12	27.80	23.58	29.25
4	1	0.75	0.75	0.1	17.24:1	17.30	22.28	20.11	24.12	21.33	26.31	23.06	26.36	26.06	27.73	23.01	27.75
5	1	0.75	0.5	0.5	13.33:1	19.26	24.49	20.57	25.99	22.45	27.69	23.64	29.00	26.13	27.81	23.94	30.35
6	1	0.75	0.5	0.25	16.95:1	18.12	23.17	20.14	24.24	21.69	26.75	23.09	26.52	26.08	27.75	23.09	27.83
7	1	0.75	0.5	0.1	20:1	17.34	21.95	19.85	22.81	21.00	25.90	22.76	25.10	26.01	27.67	22.54	25.98
8	1	0.75	0.25	0.25	19.61:1	17.69	22.37	19.88	22.73	21.27	26.24	22.79	25.31	26.04	27.70	22.58	26.13
9	1	0.75	0.25	0.1	23.81:1	16.89	21.25	19.59	21.23	20.56	25.36	22.42	23.77	26.46	28.16	22.17	23.87
10	1	0.75	0.1	0.1	27.03:1	16.32	20.53	19.44	20.49	19.78	24.39	22.19	22.89	26.62	28.32	21.88	22.85
11	1	0.5	0.5	0.5	14.09:1	19.22	24.18	20.46	25.57	22.06	27.21	23.52	28.42	26.69	28.40	23.78	30.09
12	1	0.5	0.5	0.25	18.18:1	18.02	22.93	20.01	23.58	21.19	26.13	22.95	25.95	26.62	28.33	22.80	26.99
13	1	0.5	0.5	0.1	21.74:1	17.19	21.76	19.67	22.17	20.55	25.35	22.60	24.46	26.54	28.24	22.38	24.75
14	1	0.5	0.25	0.25	20.83:1	17.51	22.16	19.78	22.36	20.59	25.40	22.68	24.82	26.24	27.92	22.47	25.34
15	1	0.5	0.25	0.1	26.32:1	16.66	21.09	19.47	20.53	19.97	24.63	22.23	23.12	26.15	27.82	21.95	22.66
16	1	0.5	0.1	0.1	29.41:1	16.02	20.41	19.34	20.01	19.41	23.94	22.05	22.22	26.02	27.69	21.71	22.07
17	1	0.25	0.25	0.25	22.73:1	17.14	21.69	19.58	21.46	20.57	25.37	22.51	23.93	26.77	28.49	22.27	24.43
18	1	0.25	0.25	0.1	29.41:1	16.05	20.69	19.34	20.01	19.91	24.56	22.05	22.22	26.67	28.38	21.71	22.07
19	1	0.25	0.1	0.1	33.33:1	15.50	20.01	19.20	19.42	19.15	23.61	21.86	21.45	26.22	27.90	21.53	21.02
20	1	0.1	0.1	0.1	37.04:1	14.58	19.61	18.87	18.76	18.75	23.13	21.70	20.82	27.04	28.78	21.38	20.19

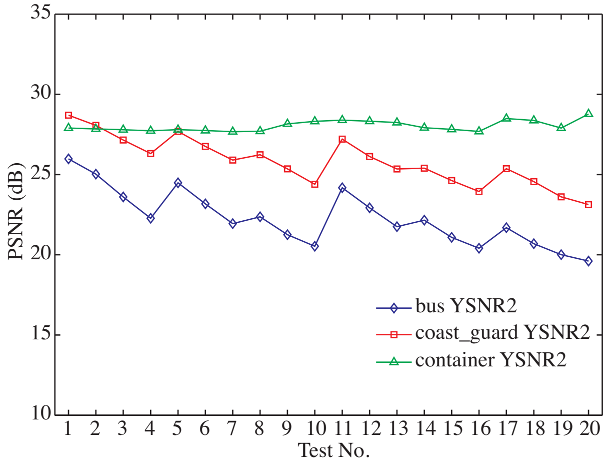

Figure 11. YSNR2 graphics of Level3 test.

Table 5. PSNR value comparison with 2 different video codecs.

	bus			coastguard			container		
CR	PSNR of proposed method	PSNR of MPEG2	PSNR of H.264	PSNR of proposed method	PSNR of MPEG2	PSNR of H.264	PSNR of proposed method	PSNR of MPEG2	PSNR of H.264
8:1	29.13	30.86	33.1	32.61	33.16	36.61	26.51	34.95	32.29
15:1	24.49	25.99	28.2	27.69	29.00	31.20	27.81	30.35	31.04
27:1	21.09	20.53	24.8	24.63	23.12	26.97	27.82	22.66	29.02
37:1	19.61	18.76	20.72	23.13	20.82	25.58	28.78	20.19	28.96

4. Conclusion

A coder structure, which incorporates 3D gradient-based adaptive temporal lifting together with polyphase symmetric motion compensation, was proposed. The effects of each of these components in the hybrid coder were examined. It was observed that putting a 3D edge alteration within the prediction in the form of temporal lifting improved the prediction efficiency as compared to direct temporal prediction without any 3D gradient considerations. This indicates that the proposed gradient approach better approximates the optical flow within the video sequence. Since the gradient estimation utilizes decoded intraframes as the domain polyphase frames, no side information is needed for transmission. It was also observed that with localization of the prediction frame block according to the motion flow within a symmetric block-based motion compensation, the performance was further improved. In essence, the symmetric block-matching in the decoded frames with a specified center block is a 3×3 extended version of the more highly detailed 3D gradient applied within the block. Therefore, there is still no extra motion vector necessary for transmission. The overall method is scalable in the sense of decomposition levels in the temporal decomposition. We experimented with up to 3 levels of decomposition and compared the results to MPEG2, which was selected due to the fact that the open source implementations enabled us to match the group of pictures (GOP) structure to that of our decomposition levels, i.e., 1, 2, and 3. As an example, for Level3 decomposition, the analogous GOP corresponded to IPBBBBBB, whose PSNR values at compatible compression ratios were presented. In order to exactly match the bit rate obtained from the compared methods, the intra- and prediction residual-type frames were encoded by the SPIHT algorithm, and no particular interest was paid to the optimization of quantization or entropy coding. Since the only motivation was to introduce the proposed method, more extensive comparisons with other state-of-the-art techniques were also kept out of the scope of this paper. For a quick comparison, the H.264 encoder produces about 2 dB of better results as compared to MPEG2 at its specific bit rate, meaning that the proposed algorithm stands well with the motivated observation of performance improvements by combining 3D gradient-based temporal lifting together with MC. It was observed that Level1 decomposition with the block-matching method gave PSNR values close to those of the MPEG2 encoder, whereas higher levels of the method gave better results than the MPEG2 encoder for most of the tested sequences. In the third level, a plausible PSNR value (29 dB) was obtained with a high compression ratio (1000:27) on the "container" test video, as compared to the MPEG2 PSNR value of 20 dB. Within our simulation environment, using MATLAB combined with the run-time of SPIHT, the computational time of the combined hybrid coder was, on the average, only 1.2 times more than the MPEG execution in MATLAB. The set of experiments indicate that the proposed hybrid method is a promising methodology for video compression.

References

[1] A. Said, A.W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6, pp. 243-250, 1996.

[2] B.J. Kim, Z. Xiong, A.W. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10, pp. 1374-1387, 2000.

[3] A. Aksay, Motion Wavelet Video Compression, Master Thesis, Electrical and Electronics Engineering Department, Middle East Technical University, Ankara, 2001.

[4] E. Elbaşı, "Robust multimedia watermarking: hidden Markov model approach for video sequences", Turkish Journal of Electrical Engineering & Computer Science, Vol. 18, pp. 159-170, 2010.

[5] M.J. Chen, C.S. Chen, M.C. Chi, "Recursive block-matching principle for error concealment algorithm", Proceedings of the 2003 International Symposium on Circuits and Systems, Vol. 2, pp. 528-531, 2003.

[6] S. Telceken, S. Gorgulu, O.N. Gerek, "A 3D lifting based method augmented by motion compensation for video coding", 9th International Conference on Intelligent Systems Design and Applications, Pisa, Italy, pp. 1324-1328, 2009.

[7] J.R. Ohm, "Temporal domain subband video coding with motion compensation", IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 3, pp. 229-232, 1992.

[8] J.R. Ohm, "Three-dimensional subband coding with motion compensation", IEEE Transactions on Image Processing, Vol. 3, pp. 559-571, 1994.

[9] B. Pesquet-Popescu, V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 3, pp. 1793-1796, 2001.

[10] Ö.N. Gerek, A.E. Çetin, "A 2-D orientation-adaptive prediction filter in lifting structures for image coding", IEEE Transactions on Image Processing, Vol. 15, pp. 106-108, 2006.