ISSN No. 0976-5697

Volume 6, No. 1, Jan-Feb 2015

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

An efficient HW/SW implementation for Wavelet-based t+2D Video Coding

Mehrez Marzougui
Department of Computer Engineering
King Khalid University, P.O Box 3236 CP 61413 KSA
Kingdom of Saudi Arabia

Jamel Baili
Department of Computer Engineering
King Khalid University, P.O Box 3236 CP 61413 KSA
Kingdom of Saudi Arabia

Mohamed Atri
Electronic and Micro-electronic Laboratory
Faculty of Sciences at Monastir
5000 Monastir, Tunisia

Abstract: In recent years, video compression has emerged as an effective technique to reduce spatial and temporal redundancy in video sequence. Temporal redundancy reduction deals with motion estimation and motion compensation algorithm with the matching technique to produce the next encoded video frame with motion vector. However, computational complexity and resource sharing of the motion estimation algorithm poses great challenges for real time applications. Dealing with these issues, the first part of this paper focuses on a system-level implementation of a wavelet-based t+2D video coding. Three blocks matching algorithms are implemented in order to evaluate the overall system performance. In fact, since the chip design and layout process is time consuming and expensive, it is very important to be able to predict the overall system performance in a high-level implementation before its circuit layout is deployed. The aim is to discuss the impact of different block matching motion estimation algorithms on video coding performance. Because of the complexity of the entire motion estimation system, decision in choosing one algorithm versus the other algorithms is often empirical and crucial. The paper presents also an efficient HW/SW codesign architecture for the proposed video encoder and its FPGA implementation. Each module of the encoder is investigated to find which approach between HW and SW is better to achieve overall performances such as real-time processing speed and flexibility.

Keywords: motion estimation, block matching algorithm, t+2D wavelet transform, video compression, HW/SW codesign, MPSoC.

I. INTRODUCTION

With the advance of multimedia systems and the internet, digital networking and video storage systems have been gaining a lot of popularity. Video compression becomes necessary for an efficient data storage as well as transmission of internet video through a limited bandwidth channel. Compression is useful because it helps reduce the consumption of resources such as data space or transmission capacity. The design of data compression schemes involve trade-offs among various factors, including the degree of compression, the amount of distortion introduced when using lossy data compression, and the computational resources required to compress and uncompress the data. Recent video coding are based on predicting a new frame from a previous frame and then predicting the current block from previous block in the same frame. The process of finding a match of pixel blocks in inter-frame coding is known as motion estimation (ME).

Video coding standards such as H.261, H.262, H.263,MPEG-1, MPEG-2 and MPEG-4 use motion estimation to reduce the temporal redundancy in the video sequence and a 2D transform, such as 2D wavelet or 2D DCT transform, is performed in the spatial domain. Finally, entropy coding is realized. In this paper, we focus on the t+2D analysis where temporal redundancy is first exploited through a motion compensated multiresolution decomposition and the resulting temporal subband frames are then spatially decomposed with a wavelet transform. Thus, temporal and spatial scalability are achieved, in particular when also the motion information is properly managed between resolution levels. Block matching algorithm (BMA) is the most popular used motion estimation method for its simplicity and regularity [1]. The basic operation of a block matching algorithm is picking up the best candidate image block in the reference image frame by calculating and comparing the matching functions between the current image block and all the candidate blocks inside a confined area in the reference frame [2]. The acceptance criterion is based on minimizing the mean square error or mean absolute difference between the two sets of pixels, and the relative displacement between the two blocks is taken to be the motion vector.

A straightforward block matching algorithm is the full search in which a measure of the difference between every block in a search window from the previous frame and the current block is calculated. This algorithm searches all locations in a specific search range and select the position where the cost function of block matching is minimized.

The heavy computational load of the full search can be significant problem in real time video coding. To reduce the computation time needed for an exhaustive search, many fast search algorithms have been proposed. However, many of them decrease the coding time at the expense of coding quality.

This paper is organized as follows. Section II presents t+2D video coding and motion compensated temporal filtering (MCTF) as well as the lifting based MCTF used to implement motion compensated wavelet transform. We introduce block matching algorithms in general and their simulation methodology. We choose three well-known algorithms and analyze them in depth in this paper. They are the three step search, the new three step search and the four step search. Section III presents an overview about a system-level implementation of a wavelet-based t+2D scheme used in most powerful video coding. In section IV, simulation results for the different block matching algorithms in terms of computational complexity and PSNR are compared and presented. Section V presents the HW/SW codesign platform, the design environment and FPGA implementation of the encoder. The last section concludes the paper.

II. T+2D VIDEO CODING AND MOTION COMPENSATED TEMPORAL FILTERING

Wavelet-based video coding began with the work of Ohm [3] who presented in 1994 the first application of the filtering scheme t+2D with a motion compensated filter allowing to take into account motion in the decorrelation of video frames. This model became the template for all wavelet-based coders.

As shown in Fig. 1, the principle of video coding scheme based on motion compensated temporal filter consists of applying a wavelet transform in the direction of frames motion to take advantage of temporal interframe redundancy. The resulting temporal subbands are then spatially decomposed to exploit their spatial redundancy. They are then quantized and encoded by a scalable embedded encoder.

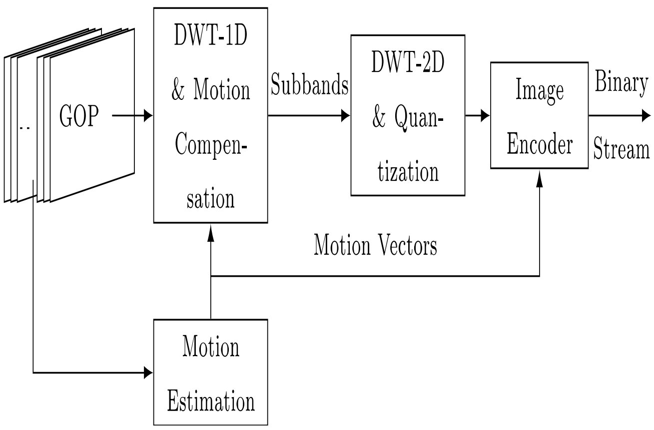


Figure 1. t+2D video coding block diagram.

A. Motion Estimation:

The aim of motion estimation algorithms is to explore the temporal redundancy in a video sequence. It consists in determining the movement of blocks between adjacent video frames to predict the most similar region in the neighbor reference image. Vectors representing the estimated motion are encoded in the bit stream. Motion estimation represents a basic criterion for evaluating a coder especially in terms of decoding time. According to the used algorithm, the motion estimation step is the most significant challenge to be addressed in the coding time. In the other hand, motion vectors inserted in the bit stream must occupy a minimum size. To address this challenge, several algorithms and methods have been implemented. Among these methods, the block matching algorithm is the most commonly used.

Block matching algorithms are emerged as a standard for encoding motion in video sequences. The basic technique of block matching is dividing the current image in macro block as shown in Fig. 2. These are then compared with the corresponding block and its immediate neighbors in the reference image (the previous or the next) to create a vector that provides the displacement of a macro-block from one location to another in the reference image [4].

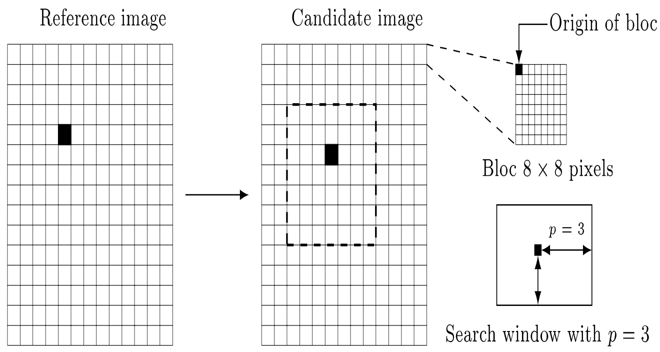


Figure 2. Principle of Block-matching algorithm.

The movement calculated for all the macro-blocks constitutes the estimated motion in the current image. The search zone also known as search window of more similar macro block, is limited to a $p$number of pixels known as a search parameter or search step on all four sides. The macro-block is considered a square of side 8 by 8 pixels or 16 by 16 pixels and the search parameter p is generally 7 pixels. There are several comparison criteria and the most popular and least expensive in terms of calculation is the mean absolute difference (MAD) given by the equation (1).

$$MAD = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|C_{ij} - R_{ij}\right| \qquad (1)$$

Where$N$ is the size of the macro-block, $C_{ij}$and $R_{ij}$are respectively the pixels corresponding to the current and reference macro-blocks. To reduce the computational complexity of full search algorithm, the number of search location or the number of operations should be decreased. This is the motivation for developing fast search algorithms over full search method that restricts the search to a few points. Many fast algorithms of block-matching have been proposed such as the Three Step Search (TSS) [5], [6], Four Step Search (4SS) [7], New Three Step Search (NTSS) [8], and Diamond Search [9], [10]. All the mentioned block matching algorithms use the equation (1) as cost function.

a. Three Step Search algorithm (TSS): The three step search algorithm (TSS) has been widely used in block matching motion estimation due to its simplicity and effectiveness [11], [5], [6], [8]. It searches for the best motion vectors in a coarse to fine search pattern. It starts with selecting initial step size in the search area of current frame. Furthermore, it sets the step size $S = 4$ and then eight blocks at a distance of step size from the center location $(0,0)$ (around the block center) are considered for comparison. At this level, cost function is calculated at each block and new origin is assigned to the pixel giving the minimal cost function. The step size is then halved and the same process is repeated until the step size becomes equal to 1. At that stage, it finds the location with the least cost function and the macro block at that location is the best match. Fig. 3 shows a particular path for the convergence of this algorithm where the calculated motion vector is $(5, -7)$. One problem that occurs with the three step search is that it uses a uniformly allocated checking point pattern in the first step, which becomes inefficient for small motion estimation.

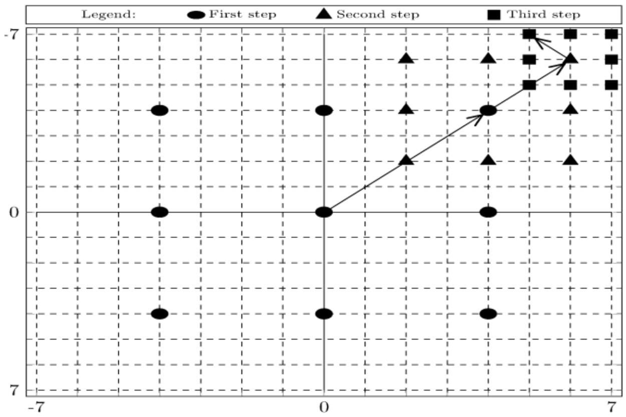


Figure 3. Three Step Search algorithm procedure.

b. New Three Step Search algorithm (NTSS): The threestep search algorithm has been widely used as the motionestimation technique in some low bit-rate video compressionapplications due to its simplicity and effectiveness. However,it is inefficient in terms of computation for the image withsmall motions since it uses a uniformly allocated checkingpoint pattern in its first step. That is, TSS has unnecessarilymany checking points when the motion is small for the block.New Three Step Search (NTSS) algorithm was proposed forsmall motioned images by R. Li et al. [12], [13]. The NTSSuses a center biased searching scheme and having provisionsfor half way stop to reduce computational cost. Unlike TSSalgorithm, NTSS employs 17 checking points in the first stepfor lowest weight using a cost function, and makes the searchadaptive to the distribution of the motion vector. This adaptivesearch is based on using 8 search locations at a distance of $S = 4$similar to TSS and 8 other at $S = 1$ away from search origin.If the lowest cost function is at the origin then the search isstopped right here and the motion vector is set as $(0,0)$. Ifthe lowest cost function is at any one of the 8 locations at$S = 1$, then we change the origin of the search to that point and check for weights adjacent to it. This means that half-way stop is possible in the first or second step when motion vector is the range of $(\pm 1, \pm 1)$. Fig. 4 shows the search procedure of NTSS where the motion vector is $(7, -3)$. Computational complexity of the NTSS algorithm may increase compared to the TSS one in the worst case [14].

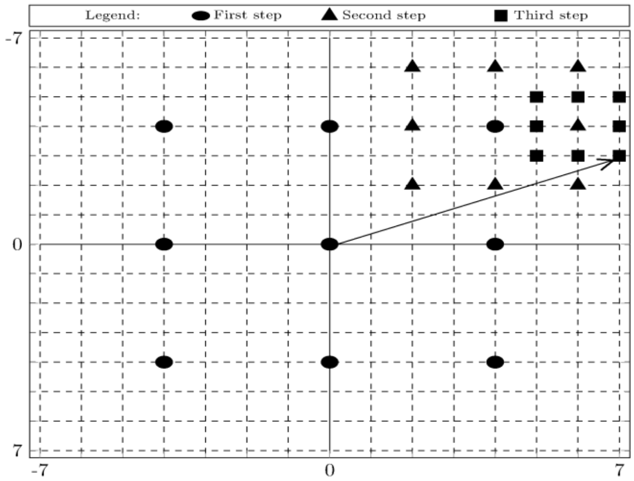


Figure 4. New Three Step Search algorithm procedure.

c. Four Step Search algorithm (4SS): Four step search algorithm (4SS) based on the center biased motion vector distribution characteristic is proposed in [7], [15]. This algorithm has been developed for real time video encoding where computational complexity in the worst case should be considered. Similar to NTSS algorithm, 4SS has a half-way stop provision. It starts with selecting initial step in the $15x15$. searching area as shown in Fig. 5. Furthermore, it sets the step size $S = 2$ and then eight blocks on a $5x5$ window located at the center of the searching area are considered for comparison. The minimum cost function is calculated at each block. If the minimum is found at the center of the search window, the search jumps to the fourth step, otherwise, the search jumps to the second step.

In the second step, the search window is sill maintained as 5x5 pixels wide. Depending on the calculated minimum cost function, we might end up checking at 3 locations or 5 locations. If the previous minimum cost function is located at the corner of the previous search window, 5 additional checking points are used. If the previous minimum cost function is located at the middle of horizontal or vertical axis of the previous search window, 3 additional checking points are used. If the minimum cost function is found at the center of the search window, the search jumps to the fourth step; otherwise to the third step.

The third step uses the same searching pattern strategy as the second step but finally it will go to the fourth step. In the fourth step the window size is dropped to 3x3(step size $S = 1$) and the direction of the overall motion vector is considered as the minimum cost function among these 9 searching points. The motion vector illustrated by Fig. 5 is $(3, -7)$. In terms of checking points for the worst case, 4SS requires 27 checking points instead of 33 candidates for NTSS.

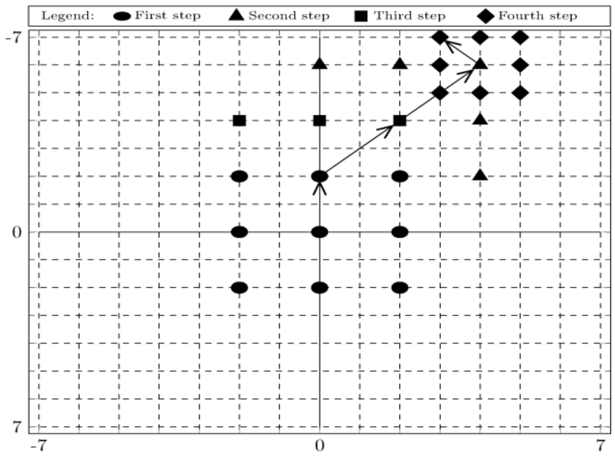


Figure 5. Four Step Search algorithm procedure.

d. Lifting based motion compensated temporal filtering:

The discrete wavelet transform (DWT) has gained widespread acceptance in signal processing and image compression. Because of their inherent multiresolution nature, wavelet coding schemes are especially suitable for applications where scalability and tolerable degradation are important. It converts an input signal xi into a sequence of low-pass frame frequency (L) "approximations" and several sequences of high-pass frames (H) that represent the details as shown in Fig.6. In practice, such transformation will be applied recursively on the low-pass series until the desired number of iterations is reached. However, the use of classical filters bank for temporal filtering using wavelet transform does not allow non linear operation on the frames in a reversible manner, such as motion compensation (MC). This limitation has been acknowledged using the MCTF lifting scheme to implement the motion compensated wavelet transform [16], [17].

Motion compensated temporal filtering is a revolution in the context of video coding standards since it provides a filtering scheme t+2D. It not only allows considering motion in video sequence but also an efficient temporal decorrelation in wavelet based video coding. In a lifting-based MCTF coder, a wavelet transform is applied along the motion trajectories with an open loop structure [18] to take advantage from temporal redundancy between neighboring frames. While the application of a linear transformation such as DWT in the temporal direction of a video sequence is not very effective if there is a significant movement, a combination of a linear transformation and motion compensation seems promising for effective compression.

Lifting scheme of DWT has been recognized as a faster approach compared to classical wavelet transform with mirror filters. Lifting decomposition is easily invertible, any type of operation, linear or non-linear, can be incorporated into the prediction and update steps. This lifting scheme consists of three basic steps: polyphase operation (SPLIT) repeatedly applied on the samples, prediction and update. Input samples are first split into odd (x_{2i+1}) and even (x_{2i}) samples and then apply a prediction operator Pand an update operator U to get the low-pass subband (L) and high-pass subband (H) frames. Motion estimation is performed between the input frames and generated motion vectors are used for motion compensation operation in the prediction and update steps as shown in Fig. 7.

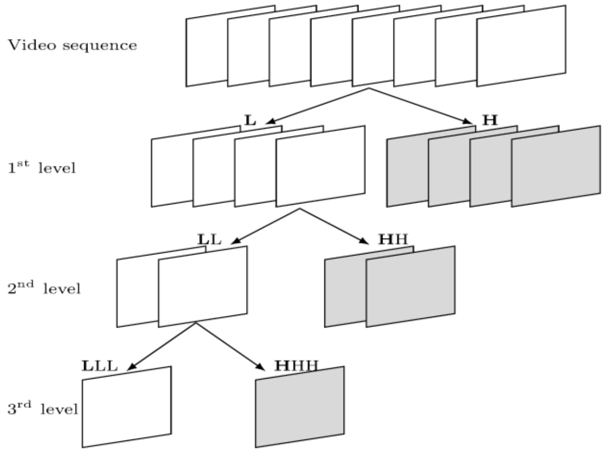


Figure 6. Temporal wavelet transform with three levels.

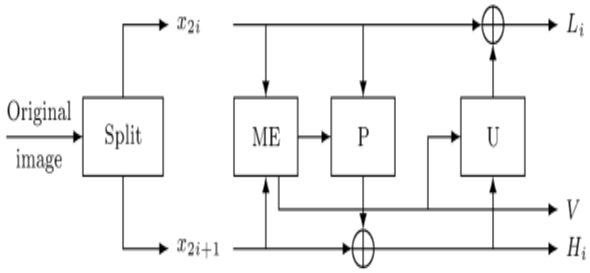


Figure 7. Lifting-based motion compensated temporal filtering analysis.

In the prediction step, odd frames are predicted by even frames and replaced by the prediction errors to form the corresponding high-pass temporal subband. In the update step, the low-pass temporal subband is obtained by updating even frames with the normalized high-pass subband samples. This also involves that motion vectors are used to match corresponding positions. Consequently, prediction and update operators become effectively spatio-temporal operators as illustrated by the equation (2).

$$\begin{cases} H_t = x_{2t+1} - P\left(\left\{x_{2(t-k)}, v_{2(t-k)}^{2t+1}\right\}_{k \in T_k^p}\right) \\ L_t = x_{2t} + U\left(\left\{H_{t-k}, v_{2(t-k)+1}^{2t}\right\}_{k \in T_k^u}\right) \end{cases} \quad (2)$$

Where$V_i^j$is the motion vector used to predict the current frame j from reference frame i . T_k^p and T_k^u respectively represent thesupport prediction time and the support update time.The lifting implementation of the wavelet transform allowsfor a motion compensated temporal transform, based on anywavelet kernel and any motion model, without sacrificing the perfect reconstruction property. The MCTF schemes mainly used is the 5/3 wavelet [19]. It employs bidirectional motion estimation and thus produces one H-frame from three input frames and one L-frame from five input frames. However, longer filters such as 9/7 [20] take better use of the temporal redundancies, but the quality on the lower bit-rates is influenced by the number of motion vectors that are transmitted negatively.

In addition to better temporal prediction introduced by bidirectional motion estimation as shown in Fig. 8, the 5/3 wavelet transform constitutes an ideal candidate to ensure the temporal filtering in a coding scheme t + 2D. The 5/3 lifting scheme can be implemented by the lifting steps illustrated by the equation 3.

$$\begin{cases} H_t(p) = x_{2t+1}(p) - x_{2t}(p - v_{2t}^{2t+1}) \\ L_t(p) = x_{2t}(p) + \frac{1}{2} H_t(p + v_{2t+1}^{2t}) \end{cases} \quad (3)$$

The low-pass and high-pass resultant frames are transformed using 2D spatial wavelet transform for the removal of spatial redundancy, by applying the DWT in both directions rows and columns as shown in Fig. 9(a).

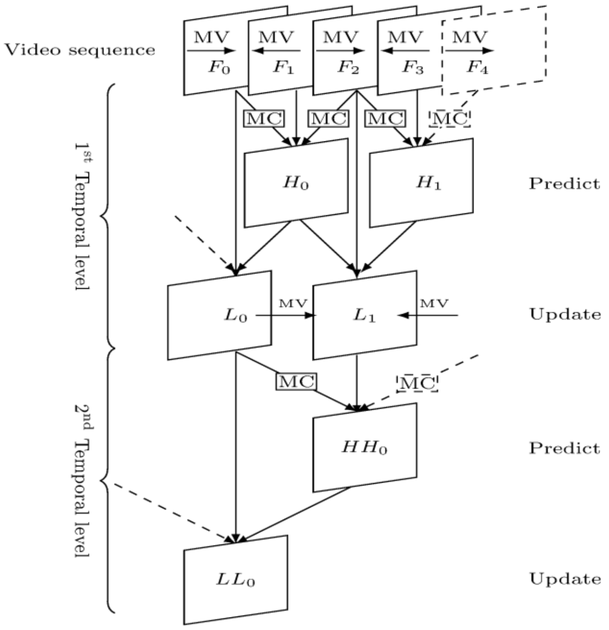
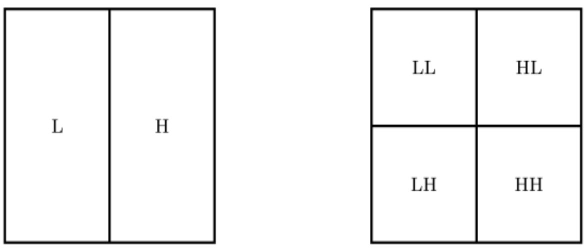
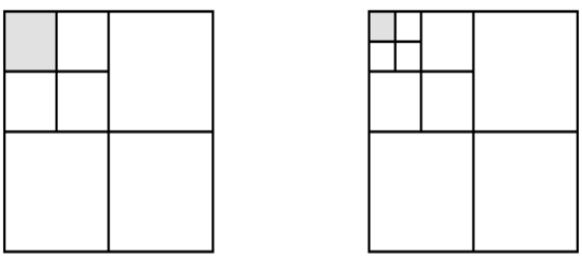


Figure 8. 5/3 Temporal wavelet transform with tow levels.

To increase the level of decomposition, we apply again the DWT on the low frequency band (subband LL) as shown in Fig. 9(b). The transformed frames will be quantized and coded using embedded coding, such as Motion-Compensated Embedded Zero blocks Coder (MC-EZBC) [21]. Motion vectors are also coded using lossless entropy coding and integrated throughout the bitstream.



(a) One level DWT.



(b) Two levels DWT (left side) and three levels DWT (right side).

Figure 9. Spatial Wavelet Transform 2D-DWT.

III. SYSTEM LEVEL IMPLEMENTATION OF A WAVELET-BASED T+2D VIDEO CODING

Based on the general structure of t+2D video coding, we have implemented the encoder shown in Fig. 10, which essentially composed of two blocks: one block for the temporal processing based on the lifting scheme MCTF and a spatial block processing applied on the-subbands generated by the temporal wavelet transform. Sequence of frames is given as input to the DWT and motion estimation and compensation is performed on the low frequency coefficients obtained from DWT. The motion estimation and compensation are directly applied on the coefficients resulting from the discrete wavelet transform, rather than on natural images. It will reduce the number of computations for the motion estimation. We use the GCC compiler with the standard SystemC library as a modeling language for the design.

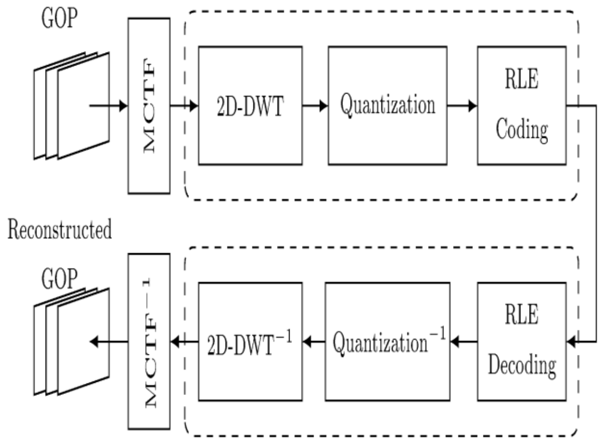


Figure 10. Proposed t+2D video coding approach.

In spatial redundancies reduction, the wavelet transform has been shown to give better results than the DCT based schemes. The wavelet-based scheme not only eliminates the visual artifacts due to DCT coding, but also allows for a multiresolution approach. In other words, the bit stream can be partially decoded in order to reconstruct a low resolution image. The DWT has several advantages of multiresolution analysis and subband decomposition, which has been successfully used in image processing.

The quality of the MCTF plays an essential role in motion compensated t+2D subband wavelet coding. It will influence not only the coding efficiency, but also the quality of the resulting low frame-rate video simultaneously. We use the 5/3 temporal filter to demonstrate the MCTF process [22] but more complex wavelet filter banks can improve potential PSNR.

Motion estimation algorithms in our video coding design have been subjected to rigorous test vectors. In fact, the computational complexity of these algorithms is the one of critical challenges for real time applications [23].

IV. COMPARATIVE STUDY OF MOTION ESTIMATION ALGORITHMS: SIMULATION AND DISCUSSION

Since different block matching algorithms are used, their image qualities are not identical. Peak signal-to-noise ratio (PSNR) is used as an indicator for quality comparison.

Another measure of the effectiveness of a motion estimation algorithm is the number of tested blocks to provide motion vectors also called computation.

To choose the adequate block matching algorithm for the proposed video coding, we used 31 frames of "Tennis Player", "Susie", "Claire", "Call Train" and "Garden" video sequences. Experiments were performed with the Tree Step Search (TSS), the New Three Step Search (NTSS) and the Four Step Search (4SS) algorithms. We note that the search parameter P influences the coder performances. Indeed, the more the search zone the number of compared blocks increase. The macro block was a square of side 16 by 16 pixels and the search parameter was ± 7 pixels. Image size was SIF (352 by 240) for each sequence. MAD (Mean Absolute Difference) as error criterion for finding optimal motion vector was used. The simulation results are presented in terms of PSNR, and also MSE (Mean Squared Error), and computational performance defined as treated macro blocks per frame while computing the motion vectors with the same sequences. In fact, the higher the PSNR, the closer the distorted image is to the original. In general, a higher PSNR value should correlate to a higher quality image.

These criteria for error and performance are described by the equations (1), (4) and (5).

$$MSE = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(C_{ij} - R_{ij}\right)^2 \qquad (4)$$

$$PSNR = 10 \log \left(\frac{255^2}{\frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(C_{ij} - R_{ij}\right)^2} \right) \qquad (5)$$

Where N is the size parameter of the matching block andi, j are related to position information in the matching block.C_{ij} points to original pixel of current frame and R_{ij} showspredicted pixel from past frame. Different experimentations for the cited above conventional algorithms, have been done and are presented in Table 1.

Table I. Average PSNR and Computations for Different Search Algorithms and Different Sequences.

Sequences		***NTSS***	***TSS***	***4SS***
Tennis Player	***PSNR***	28,727	28,514	28,519
	Computations	17,610	23,088	16,759
Call train	***PSNR***	30,269	29,426	29,804
	Computations	19,783	23,637	18.765
Claire	***PSNR***	41,755	41,644	41,673
	Computations	16,325	23,239	16,291
Garden	***PSNR***	23,552	23,056	23,310
	Computations	22,067	23,165	19,034
Susie	***PSNR***	36,242	35,621	35,744
	Computations	17,260	23,063	16,176

The aim of this study is to choose the adequate search algorithm that reduces the amount of computations without serious degradation of predicted image. According to our study, their PSNR values in all five sequences are very close to each other. In general, all the three search algorithm shave a better PSNR performance on slowly moving pictures(such as Tennis Player and Claire) but have a poorer PSNR performance on fast moving pictures (such as Garden). As shown in Table I, the average PSNR value is still quite high at the acceptable quality of 27dB expects for the fast moving pictures where the average PSNR is about 23dB. On the other hand, it is clear that the four step search algorithm outperforms all the other algorithms in terms of computation. The average computation of the four step search algorithm is about 16compared blocks while it is about 23 compared blocks for the three step search algorithm. Overall the four step search algorithm is giving the acceptable PSNR with reduced number of computations for slow and fast video sequences.

If the picture quality is our major concern, the new three step search may be the best choice. In fact, The new three step search is higher by roughly 1dB in PSNR. However, for large size pictures and/or large search ranges, the three step search and the new three step search algorithms can be costly and time-consuming.

For reasons of memory management, latency and to ensure the continuity of the produced subbands we have implemented the update and prediction steps in the temporal processing block by the MCTF-lifting steps presented by the equation 6 to generate the temporal subbands. The reconstruction of input GOP is done by the equation 7.

$$\begin{cases} H_t(p) = x_{2t+1}(p) - \frac{1}{2}\left(x_{2t}(p + v_{2t+1}^{2t}) + x_{2t+2}(p + v_{2t+1}^{2t+2})\right) \\ L_t = x_{2t}(p) \end{cases} \qquad (6)$$

$$\begin{cases} x_{2t}(p) = L_t(p) \\ x_{2t+1}(p) = H_t(p) + \frac{1}{2}\left(x_{2t}(p + v_{2t+1}^{2t}) + x_{2t+2}(p + v_{2t+1}^{2t+2})\right) \end{cases}$$
(7)

To generate the temporal subbands, we have applied the MCTF-temporal filtering to 8 original frames of the "Foreman" sequence. At the end of the decomposition, we get seven high-pass frames and one low-pass frame. The result of the temporal filtering analysis of the input GOP obtained is shown in Fig. 11.

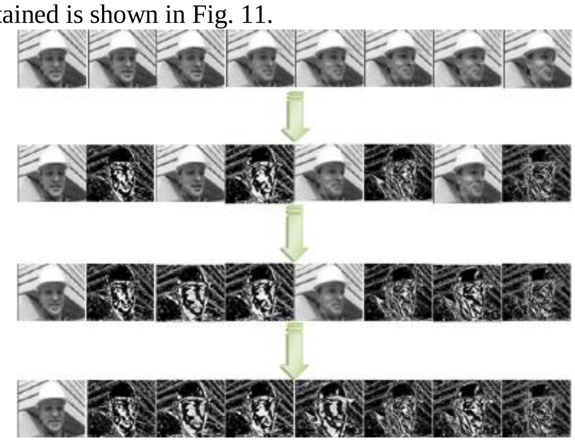


Figure 11. Temporal filtering analysis of the input GOP.

After the temporal analysis, a spatial analysis is performed on the obtained temporal subbands. The spatial analysis is done by applying 2D-DWT on each temporal subbands followed by applying a quantization and soft threshold steps to the high-pass spatial subbands, then they are reordered in Zig-Zag and coded using run length encoding (RLE).

The quality of wavelet-based compressed image depends essentially on the choice of the wavelet itself. Indeed, this choice is based on finding filter banks that compacts the maximum energy, while ensuring a good quality of image reconstruction. To choose a wavelet for spatial analysis in our encoder, we made a comparative study between the two types of wavelets (5/3 and 9/7) [20] known by their effectiveness in image compression.

The compression ratio and PSNR are inversely proportional. Indeed, increasing the compression ratio may decrease the quality of the reconstructed image. For the validation of our encoder, it is necessary to study the quality of the reconstructed images based on the compression ratio. Table II shows the PSNRs of the reconstructed GOP of the "Foreman" sequence according to the compression ratio using a spatial analysis based on the 9/7 and 5/3 wavelet with a single level of decomposition.

Table II. PSNR of Reconstructed "Foreman" GOP based on the Compression Ratio with the Spatial Analysis based on Wavelet 9/7 and 5/3.

Reconst. GOP	*Spatial analysis based on 9/7 wavelet*		*Spatial analysis based on 5/3 wavelet*	
	RATE (%)	*PSNR (db)*	*RATE (%)*	*PSNR (db)*
Frame 0	7	40,181	10,3	42,577
Frame 1	27	32,792	21	30,801
Frame 2	26,7	32,369	20,9	31,619
Frame 3	29,7	30,59	23,6	31,331
Frame 4	15,3	31,099	12,8	32,439
Frame 5	19,7	30,996	16,5	30.416
Frame 6	17,5	32,414	15,6	31,065
Frame 7	12,2	31,347	11	30,156
Average	19,387	32,723	16,462	32,425

With spatial analysis based on 9/7 wavelet we note that we can achieve an average compression ratio close to 20% while maintaining a good quality of the reconstructed frame with PSNR close to 34db. Analysis with the 5/3 wavelet also shows good performance but still less accurate than the analysis by 9/7 wavelet in terms of quality and specially in terms of compression ratio. We also notice the great difference between the compression ratio of the first reconstructed frame (frame 0) compared to others which may be explained by two reasons. First, after temporal filtering, the Frame0 represents the key frame (approximation) of the filtered GOP. The margin of error is very small compared to other frames that represent the details or the least significant coefficients. Second, in the temporally filtered GOP, the nature of the seven high-pass frames is close to being black and white. After quantification, the result is a significant redundancy that increases the compression ratio.

This test was performed on the sequence "Foreman" characterized by a complex motion and with a single level of spatial decomposition, the proposed chain coding achieve a compression ratio close to 20% with a good quality of reconstructed GOP ($\approx$ 32dB). The proposed encoder is characterized by low computational complexity and good management of the occupied memory. Indeed, in consequence of the MCTF-lifting used on temporal filtering (equation 6), the temporal low-pass subbands no longer dependent on height-pass subbands. This can greatly minimizes the time of temporal filtering process compared to classical temporal 5/3-MCTF filtering (equation 3).

Moreover, to complete the temporal filtering with tree levels of a GOP made of 8 frames; we just need to store the 8 frames of current GOP plus the first frame of the next GOP. This may significantly decrease the occupied memory during the encoding/decoding process and the GOP is shifted frame by frame to ensure the continuity of the produced temporal subbands.

V. FPGA-BASED IMPLEMENTATION

A. Hardware/Software partitioning:

The main idea of FPGA-Based implementation of HW/SW architecture for t+2D encoder is to exploit advantages of the parallel structures that can be efficiently implemented in hardware. Hardware implementation of t+2D wavelet transform promise better results compared to software based algorithms. The benefits of a parallel hardware structure are a reduced number of operation and ability to function in a parallel manner. However, there is still a good chance to reduce the complexity of the ME in software using fast motion estimation algorithms mentioned above.

To respect the system overall performances such as real-time processing speed and flexibility, we have chosen to implement the Motion Estimation, 2D-DWT, 2D-IDWT modules in hardware. The quantization and inverse quantization are regular formula and use multi-cycle to code data with Microblaze soft processor integrated with the Virtex 5 ML 507 development board. To improve performance of our encoder, we can use single-cycle combinatorial Microblaze instruction logic to implement these equations.

B. Implementation results:

Results in Table III have been obtained with separate implementation on Virtex 5 ML 705 of the particular modules (Microblaze soft core, motion estimation, 2D-DWT and 2D-IDWTcoprocessor core). The HW custom instruction for quantization and inverse quantization use only 1% of theLUTs. The entire t+2D encoder utilizes 25% of the LUTs.We can see that there is a sufficient free space for otherapplications. The whole design work at 128 MHz system clock.

Table III. Slices Occupation of the Proposed Architecture

	Microblaze	*T+2D DWT*	*T+2D IDWT*	*Q & IQ*
ALUTS	11%	3%	3%	1%
Fmax (Mhz)	255	120	130	255

The implementation of the t+2D encoder on the FPGA constitutes a multiprocessor system on a chip (MSoC) as shown in Fig. 12.

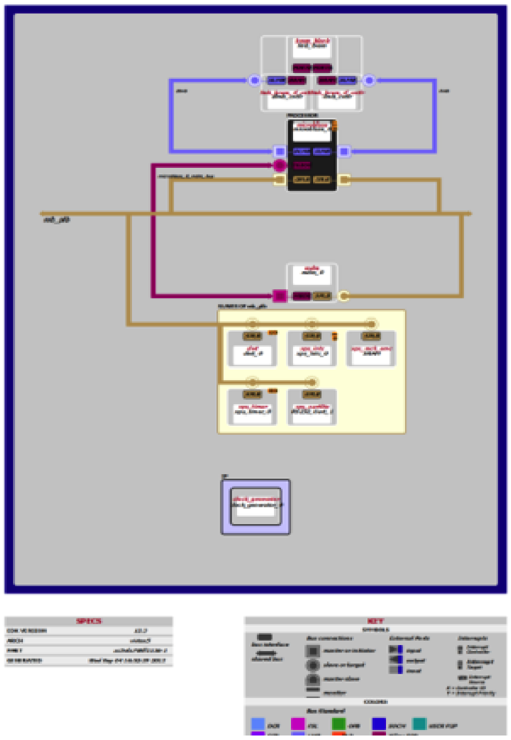


Figure 12. Circuit layout of the implemented t+2D video coding.

The power consumption of the design was evaluated on the same boards by using the Xpower tool and is presented in Fig. 13. As it was shown in the slice utilization result only one quarter of slices available was utilized. This means that FPGA leakage (static power) is going to overshadow the design's power consumption. After evaluating the results and observing the above diagram, some clear conclusions come out.

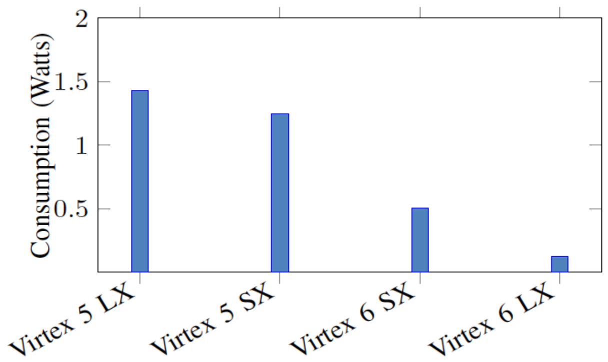


Figure 13. Power estimation of the proposed MPSoC architecture on FPGA.

First, Virtex-5 SX consumes less power than Virtex-5 LX. This is important, because the Virtex-5 LX device that was selected for our implementation (XC5VLX110T) is the "smallest" device from the LX sub-family, on which the design can fit when instantiated great image sizes. Furthermore, experiment shows that the maximum frequencies achieved are the same for these devices. Therefore, Virtex-5 SX sub-family proves to be a better solution in comparison to Virtex-5 LX sub-family, especially when we wish to exploit large image sizes. On the other hand, between Virtex-6 LX and Virtex- 6 SX, the LX device proves to be a better solution as it combines lower power consumption (75% power reduction) in comparison to SX, while the maximum frequencies achieved are approximately the same in both devices. For the purposes of estimating power consumption, changing the synthesizer's goals as far as area, speed and power is concerned, did not have any impact on the final estimation. This is because, as stated before, the 2D-DWT and 2D-IDWT cores use only 3%each of the total available slices in each device; therefore FPGA leakage is literally the only factor affecting the total power consumption estimation.

VI. CONCLUSION

The Motion estimation is a process that determines the motion between two or more frames of video sequence. Block matching algorithms are the most popular and efficient of the various motion estimation techniques. Comparative study on results, in a high level implementation, in terms of reconstructed image quality and search speed of block matching algorithms may result in an optimized hardware implementation. In this paper, a brief description and a system-level implementation of motion compensation based video compression are presented. Three block matching motion estimation algorithms, namely Three Step Search (TSS), New Three Step Search (NTSS), and Four Step Search (4SS) algorithms are compared and implemented. Experimental studies have proved that Four Step Search algorithm is the best matching motion estimation algorithm that achieve best tradeoff between search speed (number of computations) and reconstructed picture. The proposed architecture is implemented in a Multiprocessor System on Chip using Virtex 5 FPGA with the integrated Microblaze soft processor. The design is verified to work at 128 MHz.

VII. REFERENCES

[1] M. Marzougui, A. Zoghlami, M. Atri and R. Tourki, "Preliminary Studyof Block Matching Algorithms for Wavelet-based t+2D Video Coding,"International Journal Transactions on systems, signals & devices, Issueson communication & signal processing, vol. 10, no. 3, 2014.

[2] A. Ahmadi and M. M. Azadfar,"Implementation of fast motion estimationalgorithms & comparison with full search method in H.264," InternationalJournal of Computer Science & Network Security, 8(3):139–143, 2008.

[3] J. Ohm,"Three dimensional subband coding with motion compensation,"IEEE Transactions on Image Processing, 3(5):559–571, 1994.

[4] V. S. Kumar Reddy and S. Sengupta,"A Fast Predictive Algorithmand Architecture for Block Matching Motion Estimation," ICGST-GVIPJournal, 8(1):9–16, 2008.

[5] Z. Chen,"Efficient Block Matching Algorithm for Motion Estimation,"International Journal of Signal Processing, 5(2):133–137, 2009.

[6] L. Tao, Y. S. Ying, S. Z. Feng and G. Peng,"A improved threestep search algorithm with Zero Detection and vector filter for motion estimation," International Conference on Computer Science and Software Engineering, 2:967–970, Wuhan-China, December 2008.

[7] L. M. Po and W. C. Ma,"A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," IEEE Transactions on Circuits and Systems for Video Technology, 6(3):313–317, 1996.

[8] R. Li, B. Zeng and M. L. Liou,"A new three-step search algorithm for block motion estimation," IEEE Transaction, Circuits and systems for video Technology, 4(4):438–442, 1994.

[9] S. Zhu and K. K. Ma,"A new diamond search algorithm for fast block matching motion estimation," IEEE Transactions on Image Processing, 9(2):287–290, 2000.

[10] A. Barjatya,"Block Matching Algorithms for Motion Estimation," DIP 6620 Final Project Paper, spring 2004.

[11] T. Koga, K. Inuma, A. Hirano, Y. Ijima and T. Ishiguro,"Motion Compensated Interframe coding for video conferencing," Proceedings National Telecommunication, G.5.3.1-G.5.5.5, New Orleans, LA, 1981.

[12] R. Li, B. Zeng and M. L. Liou,"A new three-step search algorithm for block motion estimation," IEEE Transactions on Circuits and Systems for Video Technology, 4(4):438–442, 1994.

[13] T. Liang and P. Kuo,"A novel fast block-matching algorithm for motion estimation using adaptively asymmetric patterns," International Journal of Innovative Computing, Information and Control, 4(8):2011–2024, 2008.

[14] S. C. Cheng and H. M. Hang,"A Comparison of Block-Matching Algorithms Mapped to Systolic-Array Implementation," IEEE Transactions on Circuits and Systems for Video Technology, 7(5):741–757, 1997.

[15] D. H Lee and S. H. Cho,"An efficient VLSI architecture for motion estimation with 4-step search algorithm," Proc. International Technical Conference on Circuits Systems, Computers and Communications, 1:153– 155, Okinawa, Japan, 1997.

[16] S. Choi and J. Woods,"Motion-compensated 3-D subband coding of video," IEEE Transactions on Image Processing, 8(2):156–167, 1999.

[17] F. Li, N. Ling and S. Chiappari,"The effect of MCTF update stepon coding efficiency for H.264/AVC scalable extension a theoreticalapproach," Proc. Pictures Coding Symposium, Beijing, China, 2006.

[18] J. R. Ohm,"Multimedia Communication Technology," Springer-Verlag,Germany, 2005.

[19] Cho-Chun Cheng, Guan-Ju Peng and Wen-Liang Hwang,"SubbandWeighting with Pixel Connectivity for 3-D Wavelet Coding," IEEETransactions on Image Processing, 18(1):52-62, 2009.

[20] B. E. Reddy and K. V. Narayana,"A Lossless Image Compression usingTraditional and Lifting based Wavelets," Signal & Image Processing : AnInternational Journal (SIPIJ), 3(2):213–222, 2012.

[21] P. Chen and J. W. Woods,"Bidirectional MC-EZBC with LiftingImplementation," IEEE Transactions on Circuits and Systems for VideoTechnology, 14(10):1183–1194, 2004.

[22] M. Flierl and B. Girod,"Investigation of motion-compensated liftedwavelet transforms," Proc. Picture Coding Symposium, 59–62, Saint-Malo,France, 2003.

[23] K. L. Chung and L. C. Chang,"A new predictive search area approach forfast block motion estimation," IEEE Transactions on Image Processing,12(6):648-652, 2003.

