HARDWARE IMPLEMENTATION OF CONDITIONAL MOTION ESTIMATION IN VIDEO CODING

Avinash Kakarala

Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

December 2011

APPROVED:

Gayatri Mehta, Major Professor
Kamesh Namuduri, Committee member
Parthsarathy Guturu, Committee member
Murali Varanasi, Chair of the Department of
Electrical Engineering
Costas Tsatsoulis, Dean of College of
Engineering
James D. Meernik, Acting Dean of the
Toulouse Graduate School

Kakarala, Avinash. Hardware Implementation of Conditional Motion Estimation in Video Coding. Master of Science (Electrical Engineering), December 2011, 33 pp., 2 tables, 15 figures, references, 34 titles.

This thesis presents the rate distortion analysis of conditional motion estimation, a process in which motion computation is restricted to only active pixels in the video. We model active pixels as independent and identically distributed Gaussian process and inactive pixels as Gaussian-Markov process and derive the rate distortion function based on conditional motion estimation. Rate-Distortion curves for the conditional motion estimation scheme are also presented. In addition this thesis also presents the hardware implementation of a block based motion estimation algorithm. Block matching algorithms are difficult to implement on FPGA chip due to its complexity. We implement 2D-Logarithmic search algorithm to estimate the motion vectors for the image. The matching criterion used in the algorithm is Sum of Absolute Differences (SAD). VHDL code for the motion estimation algorithm is verified using ISim and is implemented using Xilinx ISE Design tool. Synthesis results for the algorithm are also presented.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my major advisor Dr. Gayatri Mehta and Dr. Kamesh Namuduri for their guidance, support and constant mentoring throughout my thesis. I would also like to thank Dr. Parthasarathy Guturu for his support as a committee member for my thesis. I am thankful to the faculty and staff of the electrical engineering department for their support. Lastly, I am grateful to my family and friends for their love and support.

TABLE OF CONTENTS

Page

Chapters

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

The effectiveness of a video encoding technique depends on the rate distortion tradeoffs that it offers. Therefore, there is a need to investigate the tradeoffs between rate and distortion. Rate distortion analysis determines these tradeoffs.

The minimum rate required to encode a source (one symbol at a time) without any loss is given by the first order entropy H_1 of this source. If a source is denoted by a random variable X, then H_1 is given by [1],

$$H_1(X) = -\sum_{x \in X} p(x) \log p(x)$$

where x is an instance of X and p(x) is the probability that event x occurs. The entropy reduces if more number of symbols are encoded together. If a source has N symbols, it is well known that

$$H_1(X) > H_2(X) > \ldots H_i(X) \ldots > H_N(X) \; i = 1,2, \ldots N,$$

where i is the order of entropy and also the number of symbols coded together. As N tends to infinity, $H_N(X)$ tends to entropy rate, $\overline{H}(X)$. Entropy rate is the least required to encode the source. Thus,

$$H_1(X) > H_2(X) > \ldots H_i(X) \ldots > \overline{H}(X) \; i = 1,2, \ldots \infty.$$

Rate distortion analysis is appropriate for lossy coding schemes. If an attempt to encode all source symbols together is made, the complexity of the video encoding technique increases. Further, the process of deriving an *R(D)* function for such a technique becomes more complex. Therefore, a better approach is needed to investigate rate distortion. In video, there exist spatial as well as temporal correlations. If these two type of correlations are eliminated such that the resulting source can be modeled as an ensemble of independent and identically distributed (i.i.d) samples, then the entropy rate can be achieved by encoding these samples independently. Motion estimation is one way to achieve this.

Conditional motion estimation is a method that estimates motion only for certain regions selected based on a motion activity criterion [2], [3]. In this technique, the image frame is first divided into blocks. The blocks are classified into active and inactive blocks based on the level of motion activity. Once classified the active blocks are coded using displaced frame difference (DFD) and the inactive blocks are coded using frame difference (FD) method. If the parameters involved in the decision criterion are chosen appropriately, conditional motion estimation offers significantly better rate distortion tradeoffs compared to classical motion estimation methods. A conditional motion estimation scheme is developed and applied to derive the rate distortion function for video encoding.

Video coding standards such as H.26x, use motion estimation algorithm to achieve high compression efficiency. Block matching algorithms are used to estimate the motion for the video. But block matching algorithms are computationally complex and takes most of the processing time for video encoding. Block matching algorithms are very difficult to implement on FPGA chip because of its computational complexity. The simplest and most accurate algorithm is the exhaustive block matching algorithm (EBMA). EBMA exhaustively evaluates the best matched block in the reference frame. EBMA requires intense computation hence it is difficult to implement this algorithm in hardware. Hence various fast algorithms such as 2D-log search method, three-step search method (TSS), the new three step search (NTSS), diamond search methods are proposed. We use 2D Log search method to estimate the motion vectors and implement it in FPGA. Sum of absolute differences (SAD) is the matching criterion used in 2D-log search algorithm. The code for motion estimation is written in VHDL and implemented it on Virtex-6 XC6VLX130T FPGA device. The design is simulated and verified using ISim.

1.1 Thesis Outline

The thesis report is organized as follows. Chapter 2 presents a background and related work. Chapter 3 presents the proposed rate distortion approach and its results. Chapter 4 presents the Hardware implementation of 2D-logarithmic search algorithm. Chapter 5 presents the conclusions and the future scope.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Rate Distortion Theory and Its Applications

The foundation for rate distortion theory was first laid out by Shannon in [4]. The measure of distortion that can be allowed by the communication system is termed as fidelity criterion [5], and is related to the rate by a function R(D) called rate distortion function. The significance of R(D) is that it determines the rate R required by a communication system for a fidelity criterion D. The early contributions to rate distortion theory are presented in [6].

One potential application is its use in pattern classification problem [1]. In this application, features belonging to different classes are assumed as outputs of a source and an equivalent data compression problem is designed. The R(D) for such a data compression problem explains the tradeoffs between the number of features selected and the resulting error in classification.

A study on the rate distortion function and its applicability in designing a practical communication system for video sources with bounded performance is discussed in [7]. The application of rate distortion theory in intrusion detection is discussed in [34]. Numerous other applications of rate distortion theory are extensively discussed in [8] and [9]. In most applications of rate distortion theory, be in video processing or in chemical processing, a basic rate distortion problem analogous to the application is designed and the implications of rate distortion theory are used to solve it.

2.2 Basics of Video Coding

The basic components in a video coding system are shown in the Fig. 2.1. The digital video sequence in the encoder is described using the parameters of source model. The source symbols are quantized. The quantization parameters depend on the desired trade-off between the bit rate and distortion. The quantized symbols are mapped into binary code words using lossless coding techniques. The resulting bit stream is transmitted through the communication channel. The decoder retrieves the quantized version of the source model by reversing the binary encoding and quantization process of the encoder. The synthesis block computes the decoded video frame using the quantized version of the source model. Analysis involves the motion estimation and compression process for the video frame.

In video there exist spatial as well as temporal correlations. Transform and predictive coding are used to eliminate these correlations. Transform coding is effective for compression of still images and video frames. Generally, a transform should be applied to an entire image or video frame, to fully exploit the spatial correlation among pixels. But to reduce computational complexity, block based transform coding is used. The popular transforms used in video coding techniques are Karhunen-Loeve transform (KLT), discrete Fourier transform (DFT), discrete cosine transform (DCT) and wavelets. KLT is optimal in sense of energy compaction but it requires more computational resources when compared to DCT and DFT. DCT provides better energy compaction when compared to DFT, therefore, it is widely used in image and video coding standards.

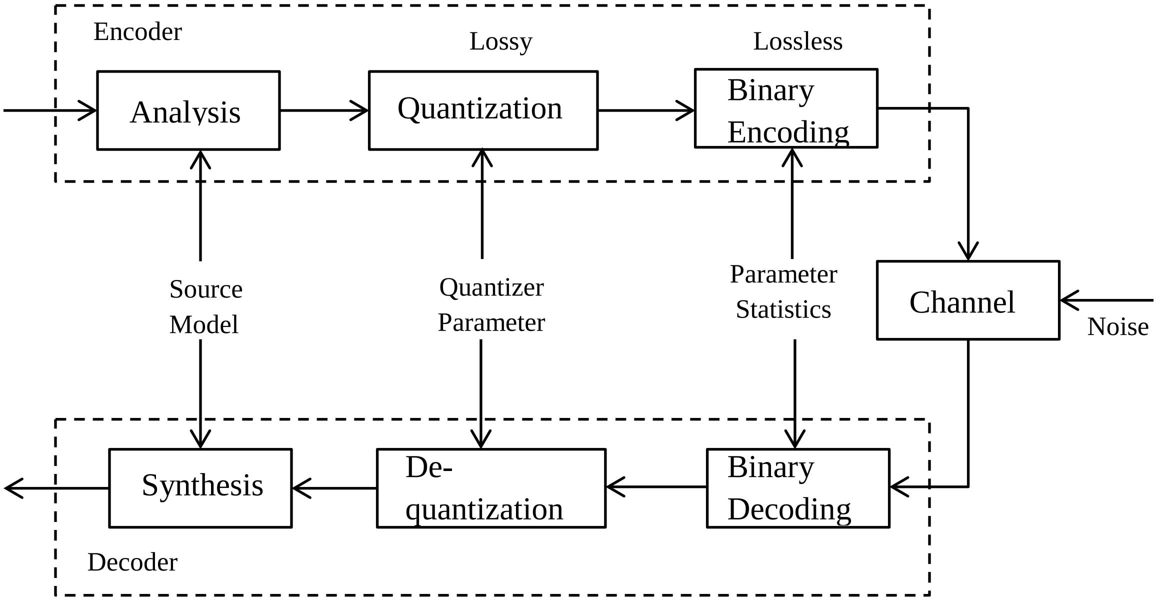


Fig. 2.1 Overview of a video coding system

Predictive coding is another important technique for image and video coding. Temporal predictive coding using motion compensated prediction is the key to success of modern video coding standards. In predictive coding, a pixel is not coded directly, it is predicted from those of adjacent pixels in the same frame or in a previous frame. Fig. 2.2 shows the block diagrams of the encoder and decoder for a general lossy predictive coding system. In the encoder, an input sample is first predicted from some previously reconstructed samples stored in the memory, and then the prediction error is quantized and then coded using variable length coder. The reconstructed value at decoder is the predicted value plus quantized error. The encoder must repeat the same process as the decoder to reproduce reconstructed samples to guarantee that encoder and decoder use exactly the same prediction value. In the fig. 2.2, s and s_p represent the original and predicted sample, e_p and $\hat{e}_\mathrm{p}$ represent the original and quantized prediction error and $\hat{s}$ represent the reconstructed sample.

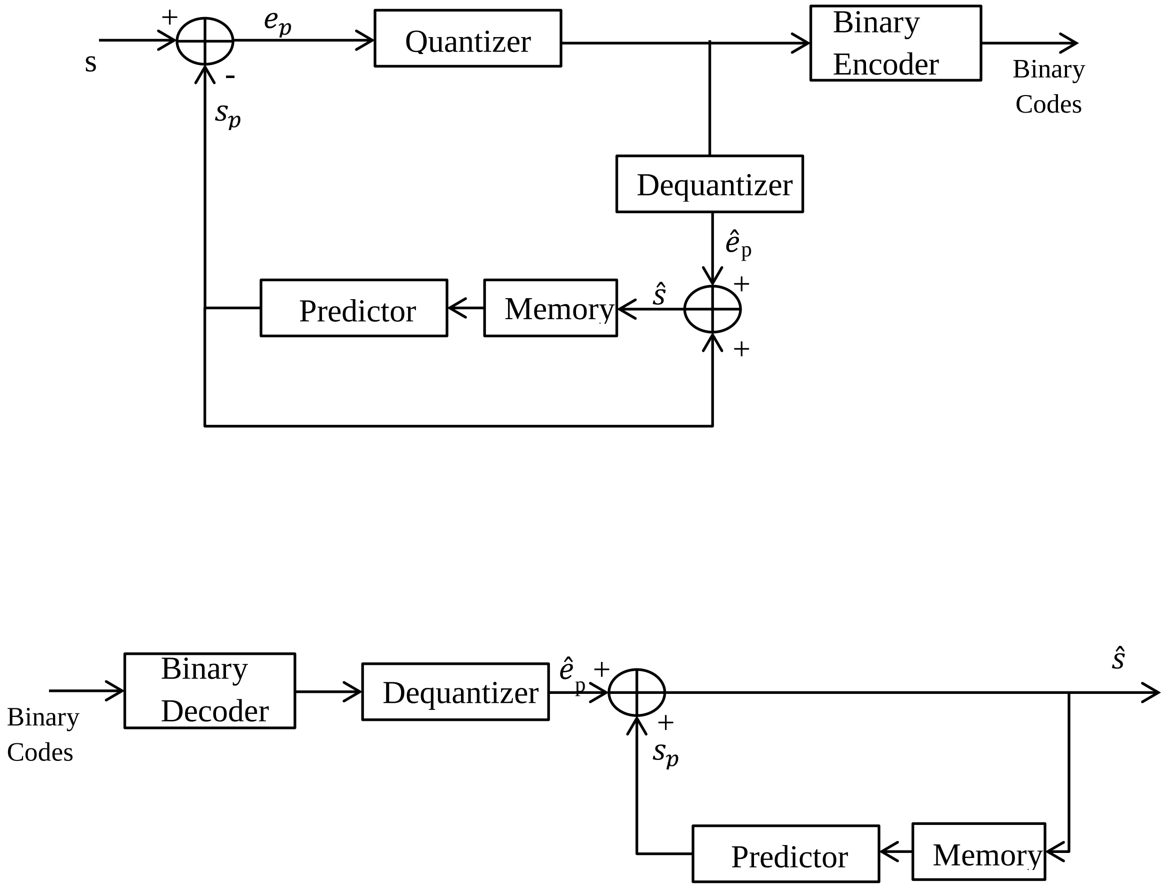


Fig. 2.2 Block diagrams of encoder and decoder for lossy predictive coding systems

Transform coding and entropy coding in a video coding system are generally lossless and information loss is generated by quantization. Quantization is done to represent a continuous source with finite number of bits. Scalar quantization quantizes each sample in a source signal to one of the reconstruction values in a predesigned codebook. The simplest scalar quantizer is the uniform quantizer, which has equal distances between adjacent boundary values and between adjacent reconstruction values. To improve the quantization efficiency, optimal scalar quantizer and minimal mean square error (MMSE) error quantizer are introduced. Rather than quantizing one sample at a time, one can quantize a group of N samples together. Each sample group is known as a vector and the process of quantizing a vector at a time is called vector quantization. The simplest vector quantizer is lattice vector quantizer in which all the partition regions have same shape and size. Similar to scalar quantizer, to improve the quantization efficiency Lloyd algorithm for designing optimal vector quantizer are introduced.

Binary encoding is a process of representing each symbol from a finite alphabet source by a sequence of binary bits, called code word. The code words for all possible symbols form a code book. A symbol may correspond to one or several original or quantized pixel values or model parameters. The simplest binary code is the fixed length binary representation of all symbols. If the number of symbols is L, then the bit rate will be $\log_2 L$ bit/symbol. The fixed length coding will be inefficient i.e. bit rate is much higher than the source entropy, unless source have uniform distribution. To reduce the bit rate, variable length coding (VLC) is needed, which codes a symbol with a high probability using shorter code word. VLC is also referred to as entropy coding.

Huffman coding, Lempel-Ziv-Welch (LZW) method and arithmetic coding are three popular VLC methods. Huffman coding converts a fixed number of symbols into a variable length code word. LZW method converts a variable number of symbols into a fixed length code word. Arithmetic coding converts a variable number of symbols into a variable length code word. Huffman and arithmetic methods are probability model based and both can reach entropy bound asymptotically. Arithmetic coding can more easily achieve the asymptotic performance but it is more complex than Huffman coding. LZW method is universally applicable, but it is less efficient than the other two methods. Huffman and arithmetic methods have been used in various video coding standards.

2.3 Video Coding Techniques

Conventional video coding techniques perform motion estimation on the sender side. Motion in video is represented using different methods including pixel based representation, block based representation, and mesh based representation etc [10]. Motion is estimated using a criterion such as DFD or Bayesian. A tutorial on estimating two dimensional motion is presented in [11].

The complexity of an encoder increases as the complexity of motion estimation method increases. An encoder of this kind is not suitable to be used in resource constrained application such as wireless sensor networks. A better way of encoding in resource constrained situations is distributed source coding which is built on Slepian-Wolf coding, [12], Wyner-Ziv coding, [13] and channel coding principles. One of the coding techniques built on distributed source coding principles, PRISM which stands for Power-efficient, Robust, high compression, syndrome based Multimedia coding”, is described in [14].

2.4 RD Analysis of Video Coding Techniques

Rate distortion analysis for video coding has recently gained more interest due to paradigm shift in video coding. The principles of distributed source coding [12] are extended to lossy-compression in [15]. The rate distortion analysis for Wyner-Ziv video coding has been proposed recently in [16]. A rate-distortion function for distributed source coding with L+1 correlated memory less Gaussian sources in which L sources are assumed to provide partial side information at the decoder side to construct the L+1th source is proposed in [17]. In general, a rate distortion function for any source that can be modeled as an Nth order Gaussian-Markov process is derived in [18].

2.5 Conditional Motion Estimation

Conditional motion estimation methods first divide a image into blocks of equal size, and then, classify the blocks into active or inactive. Block-based motion estimation is then performed for active pixels. Block matching motion estimation process is shown in fig. 2.3.

The blocks are classified into active and inactive blocks based on two thresholds, one at pixel level (T_g) and one at block level (T_p) [2]. If a pixel value in the difference image is greater than the threshold T_g, then that pixel is classified as an active pixel, otherwise it is classified as an inactive pixel [19]. A block in a difference image and the classification of active and inactive pixels is illustrated in Fig. 2.4. The number of active pixels in every block is counted, and if this count is greater than T_p, it is classified as an active block, else, it is classified as an inactive block.

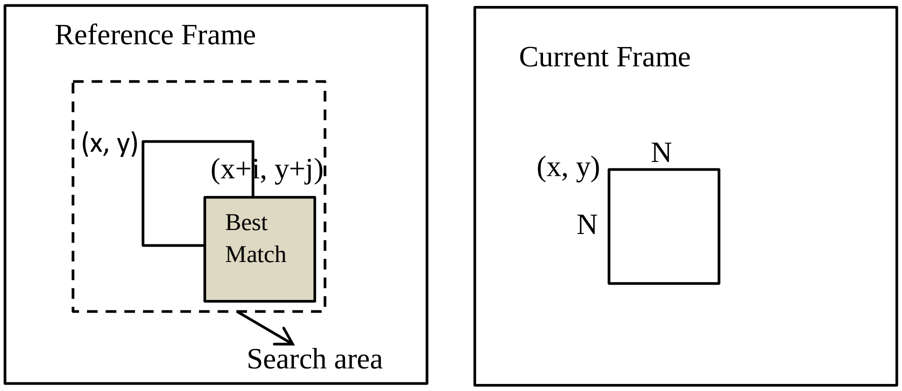
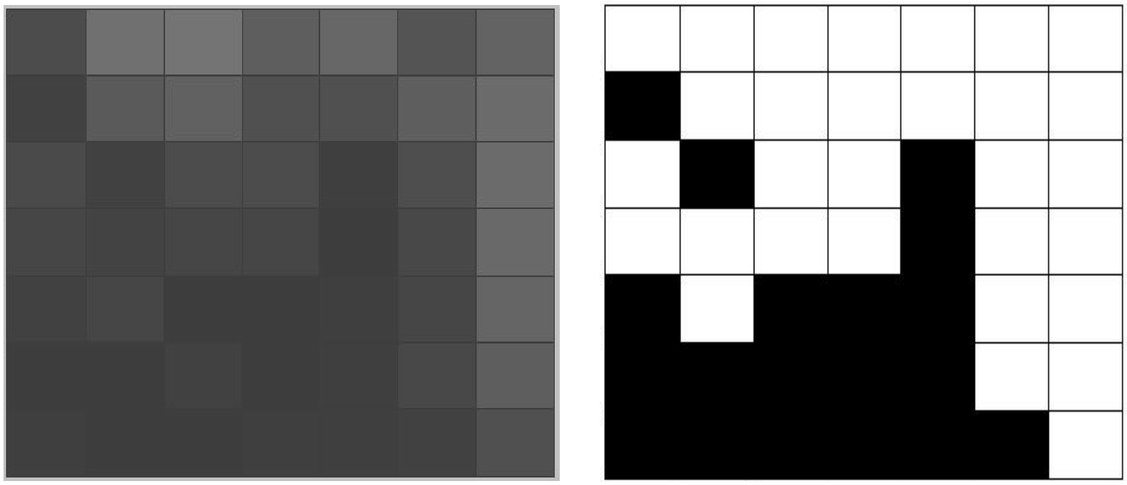


Fig. 2.3 Block based motion estimation process where (x, y) is the block for which best matched block is to be found in the reference frame and (i, j) is the motion vector for the current block.

The selection of the two thresholds T_g and T_p is an important task in conditional motion estimation process. The selection of T_g is crucial since it directly decides if a pixel is active or not. It is known that in a frame there exists a correlation between intensities of adjacent groups of pixels. Taking this fact into account, the pixel level threshold T_g is adaptively selected using Bayesian criterion in [20]. In this research, we use the same concept and estimate T_g adaptively for all pixels in a frame.

The selection of T_p also significantly impacts the performance of the proposed method. For example, if T_p is increased, then the number of active blocks decreases, resulting in low bit rate and high distortion. On the other hand, if T_p is decreased, then the number of active blocks increases, resulting in high bit rate and low distortion.



(a) (b)

Fig. 2.4 The figure (a) depicts a block in a difference image and figure (b) depicts the active pixels in that block. The gray pixels in (a) indicate the intensity values and the dark and bright pixels in (b) indicate the pixels with intensities above T_g and below T_g respectively

2.6 2D-Logarithmic Search

EBMA requires a very large amount of computation. To speed up the search, various fast algorithms for block matching have been developed. The key to reduce the computation is by reducing the number of search candidates. One popular fast search algorithm is the 2D-log search method. It starts from the position corresponding to zero displacement. Each step tests five search points in a diamond arrangement. In the next step, the diamond search is repeated with the center moved to the best matching point resulting from the previous step. Search step size is reduced if the best matching point is the center point or on the border of maximum search range. Otherwise step size remains the same. The final step is reached when the step size reduces to 1 pel and nine search points are examined at this last step. The initial step size is usually set to half of the maximum search range. With this method one cannot pre-determine the total number of steps and total number of search points as it depends on actual motion vectors. This algorithm requires more steps when compared to three step search, but it can be more accurate when the search window is large. Step by step procedure for the 2D-Logarithmic search is given below.

Step 1: Initial step size is set as half of the search range. The block is located at the center of the search range.

Step 2: Test the four blocks in diamond arrangement around the center block for the best matched block.

Step 3: If the best matched block is the center block, halve the step size. If any one of the four blocks is the best match, then it becomes the center and step 2 is repeated.

Step 4: If the step size is 1, then all the eight blocks around the center block are tested and the best matched block is the required block.

2.7 Sum of Absolute Difference

Sum of absolute differences (SAD) is the function used to estimate the differences or similarities between the two blocks. Smaller the value returned by the SAD, the more the two blocks are similar. MSD (mean squared difference) and MAD (mean absolute difference) are the two other functions used to estimate the similarity between the two blocks. SAD is the simplest function to find the similarities between the two blocks when compared to other functions. The complexity of motion estimation increases with the complexity of these functions. So we use SAD function as it can be easily implemented in VLSI design. SAD for two blocks of size N x N is given by,

$$SAD(x,y,i,j) = \sum_{p=0}^{N}\sum_{q=0}^{N} |A(x+p, y+q) - B((x+p)+i, (y+q)+j)|$$

where (x,y) is the position of the current block and (i,j) denotes the motion vector i.e. the displacement of the current block A with respect to the block in reference frame B.

2.8 FPGA Implementation of Motion Estimation Algorithms

As the complexity of the motion estimation algorithms increases, its hardware implementation also becomes more complex. Different VLSI architectures are designed to speed up and reduce the computational complexity of motion estimation algorithms. The hardware design and implementation of block based motion estimation algorithms for various real time applications is described in [28]. A hardware architecture for fast search block matching motion estimation algorithm using Line Diamond Parallel Search (LDPS) for H.264 coding system is proposed in [29]. A novel block matching algorithm for motion estimation in a video sequence well suited for high performance FPGA is given in [30]. An algorithm developed to provide automatic motion detection and object tracking functionality embedded within intelligent systems is implemented on Altera stratix FPGA [31]. A motion estimation hardware architecture which can be used for frame rate up conversion is presented in [32]. An efficient implementation of H.264/AVC motion estimation algorithm in hardware and software is proposed in [33].

CHAPTER 3

CONDITIONAL MOTION ESTIMATION

This section discusses the rate distortion analysis of video coding based on conditional motion estimation technique.

3.1 Video Encoding

Let $F_1(\bar{x})$ be the anchor frame and $F_2(\bar{x})$ be the target frame. If D($\bar{x}$) represents the difference image, then,

$$D(\bar{x}) = |F_2(\bar{x}) - F_1(\bar{x})|$$

Where $\bar{x}$ is a vector representing pixel locations. Every pixel in D($\bar{x}$) is compared to its corresponding T_g and classified as an active pixel if it is greater or inactive if it is lesser. The number of active pixels in a block is then counted and if the count is greater than T_p that block is classified as an active block else it is classified as an inactive block. Once active blocks in target frame are determined, the next step is to perform block based motion estimation for all those blocks. We assume that anchor frame is already available at the decoding side and encode the target frame using conditional motion estimation.

Block based motion estimation is a motion compensated video technique used in various video coding standards including H.26X and MPEG-X. A block based motion estimation technique uses a block matching algorithm to test each block in the anchor frame with every block in the target frame to find the block that matches the most.(The matching criteria is usually the mean square difference between the blocks compared.) In this work, a fast block-matching search algorithm called diamond search algorithm is used to estimate motion vectors for all blocks in the anchor frame. A motion vector represents the displacements of a block in x and y directions.

Let $\bar{a}$ be a motion vector whose parameters a_h and a_v represent the horizontal and vertical displacements that a block in an anchor frame undergoes to reach its position in the target frame. If every block is identified by the first pixel in it, then the set of motion vectors for the frame can be represented as d($\bar{x}$;$\bar{a}$). In conditional motion estimation, we find motion vectors for active blocks only. The inactive blocks are assumed not to have moved and are represented with zero motion vectors. Once the motion vectors are found the displaced frame difference,E_D, which is the difference between the target frame and the motion compensated anchor frame, is generated. This can be written as,

$$E_D = F_2 - C(F_1; d)$$

where $C(F_1; d)$ is the motion compensated frame constructed from $F_1(\bar{x})$ and motion vectors set d($\bar{x}$;$\bar{a}$). The displaced frame difference thus evaluated, is scalar quantized to obtain $Q_D(\bar{x})$

3.2 Rate Distortion Analysis

A given video encoded using conditional motion estimation method consists of the motion vectors and quantized displaced frame differences for the active blocks and the quantized frame differences for inactive blocks. In order to derive the rate-distortion function, these encoded variables need to be modeled using standard probability density functions. We know that samples of the displaced frame difference follows i.i.d Gaussian distribution. We assume that samples of frame difference follow Gauss Markov process. This can be shown in the Fig. 3.1.

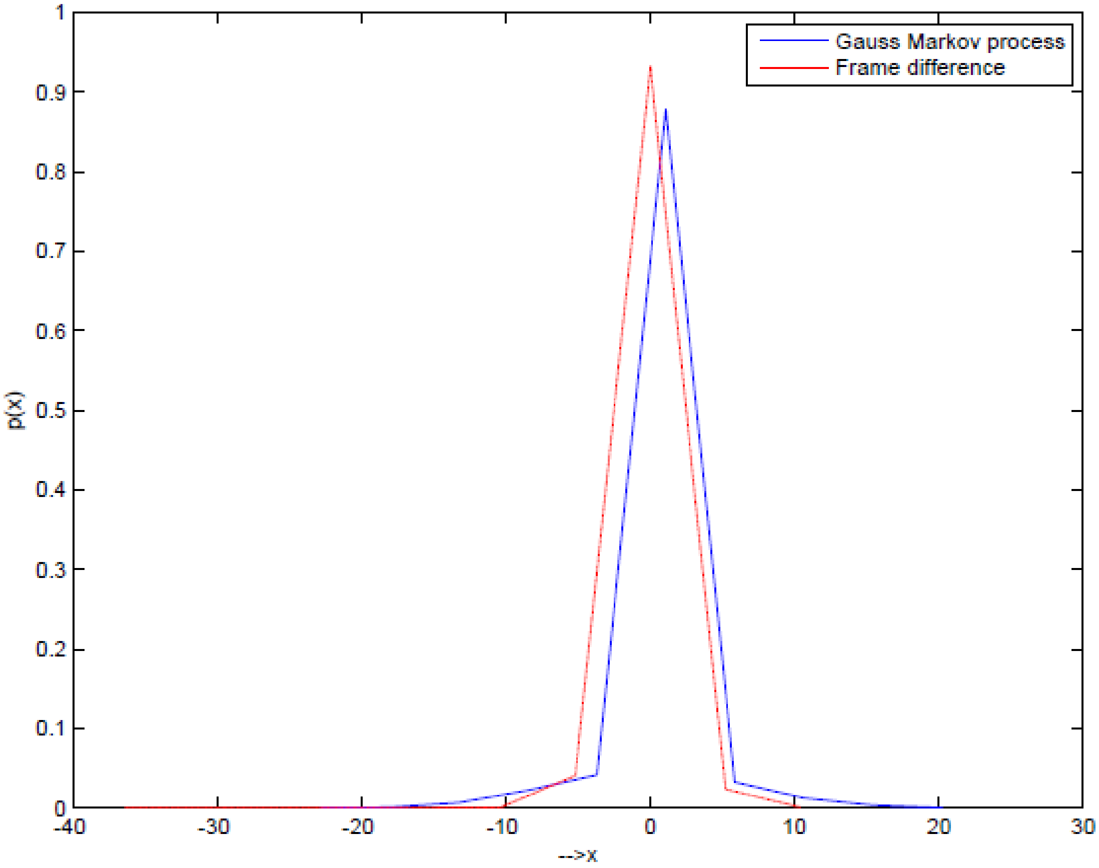


Fig. 3.1 Probability density functions of Gauss-Markov process and frame difference

3.2.1 RD Analysis for Active Pixels

E_D is the displaced frame difference image formed after motion compensation. It is well known that samples of the displaced frame difference follow i.i.d. Gaussian distribution.The rate distortion relationship of i.i.d. Gaussian source is given by [1],

$$R_A = \frac{1}{2}\log_2 \frac{\sigma_D^2}{D_A}$$

where σ_D^2 is the variance, D_A is the distortion of the active pixels.The bit rate allocation for motion vectors R_M is given by,

$$R_M = \frac{b_M}{N_{br}N_{bc}}$$

where b_M is the number of bits for each motion vector and N_{br} and N_{bc} are the dimensions of the block.

3.2.2 RD Analysis for Inactive Pixels

There exist spatial and temporal correlations between consecutive frames of a video. An image block in a given frame may be correlated with some other blocks in its previous and consecutive frame. We assumed that the inactive pixels in each image frame follow Gauss Markov process, then the bit rate R_I incurred in encoding samples of such a source is related to distortion D_I by [27],

$$R_I = \frac{1}{2}\log_2 \frac{(1-\rho_I^2)\sigma_I^2}{D_I}$$

where ρ_I^2 is the correlation between the inactive pixels in the frame, D_I and σ_I^2 are the distortion and variance of all inactive pixels in the frame. Therefore, the sum rate and distortion for the video coding scheme is given by,

$$R = \lambda(R_A + R_M) + (1-\lambda)\, R_I$$

$$D = \lambda D_A + (1-\lambda)\, D_I$$

where λ represents the amount of motion activity defines fraction of active pixels in the video.

3.3 Results

A tennis video, football video and flower garden video, each 19 frames long, are chosen for experiments. We plot the RD curves based on the results obtained for these 3 video sequences when they are encoded using the conditional motion estimation scheme discussed in section using Matlab. RD curves for the tennis and flower garden videos can be seen in the Fig. 3.2, 3.3, 3.4, 3.5, 3.6 and 3.7. The two thresholds T_g and T_p used in conditional motion estimation scheme provide the ability to vary rate needed to encode the input video sequences.

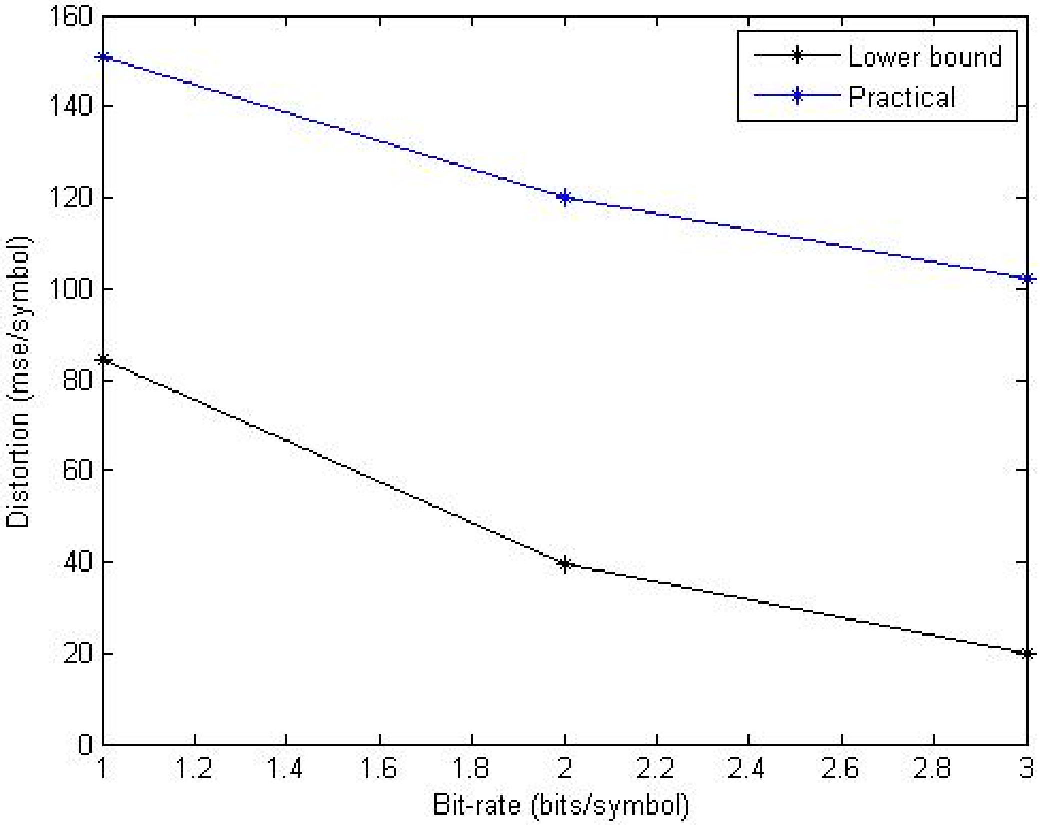


Fig. 3.2 Practical RD curve and the lower bound for the tennis video

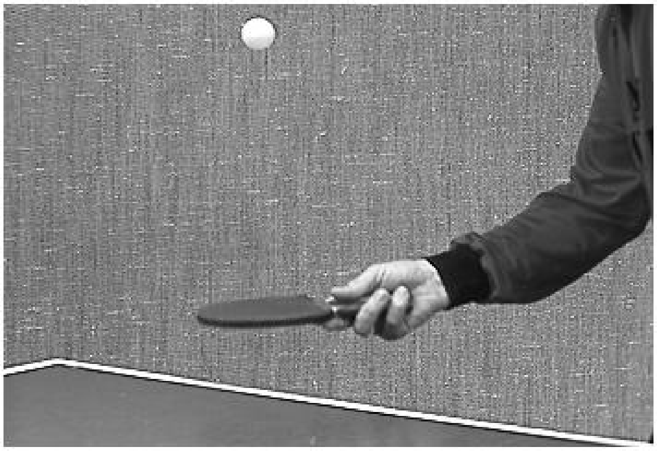


Fig. 3.3 Target tennis frame used for conditiona1 motion estimation

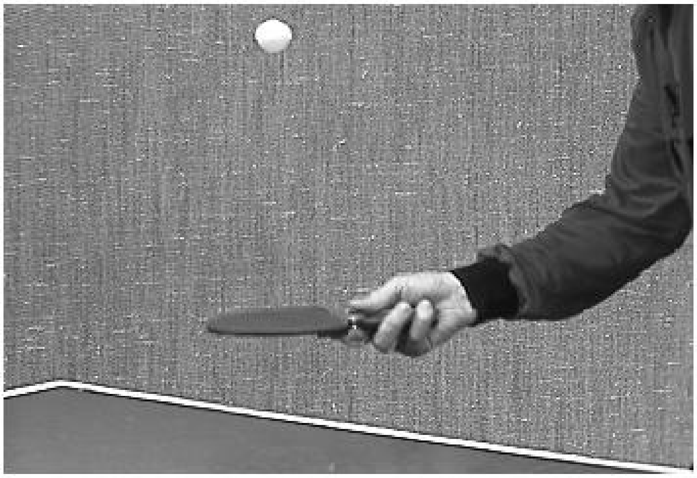


Fig. 3.4 Reconstructed tennis image using conditional motion estimation scheme at Tp =20

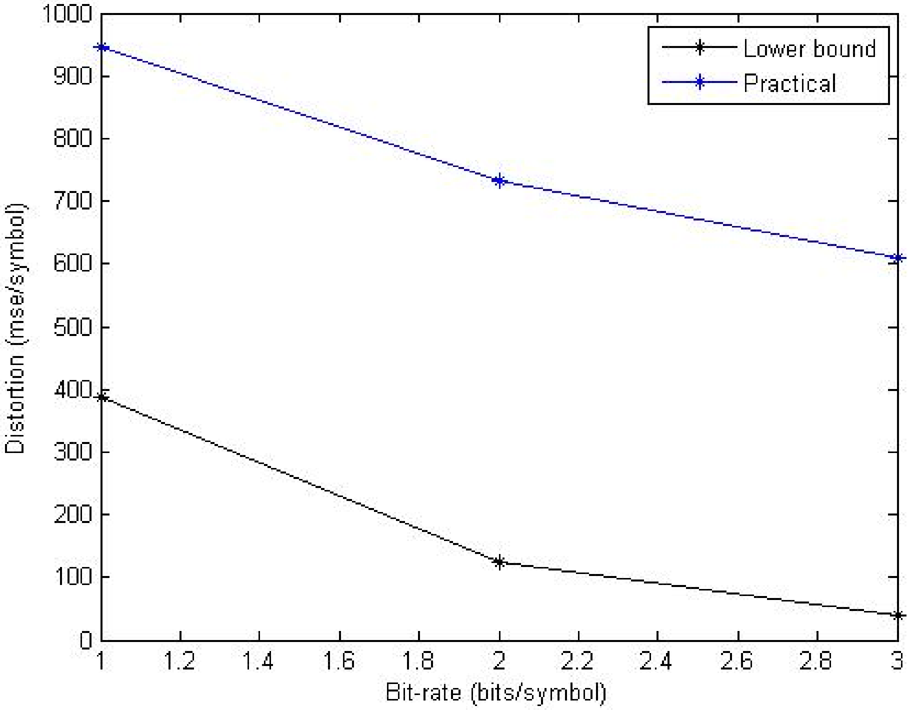


Fig. 3.5 Practical RD curve and the lower bound for the flower garden video

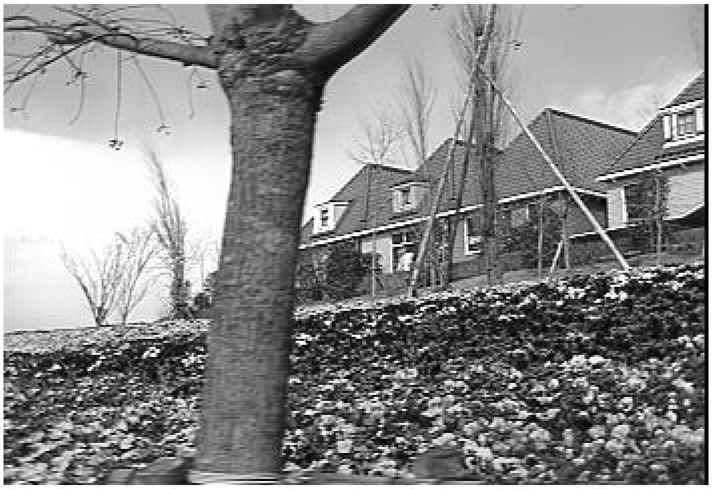


Fig. 3.6 Target flower garden frame used for conditional motion estimation

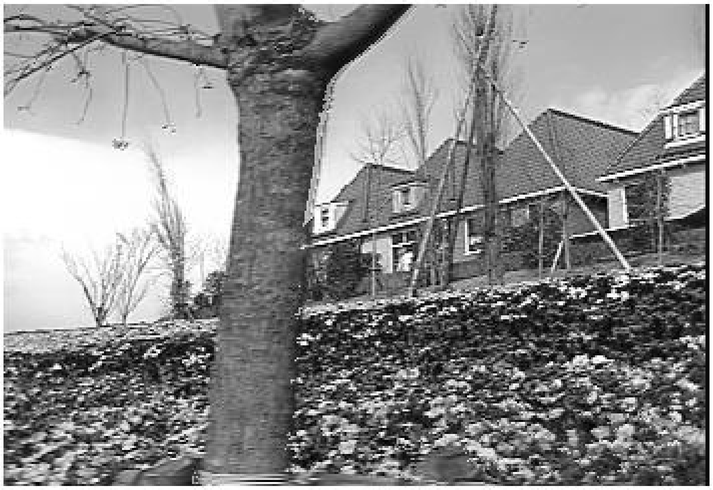


Fig. 3.7 Reconstructed flower garden image using conditional motion estimation scheme at Tp =20

CHAPTER 4

HARDWARE IMPLEMENTATION

Hardware implementation of block matching algorithms is very difficult because of the complexity of algorithms. We consider a simple 2D-Logarithmic search and implement it using Xilinx ISE Design Tool Suite. Block diagram for the hardware implementation is shown in the figure 4.1.

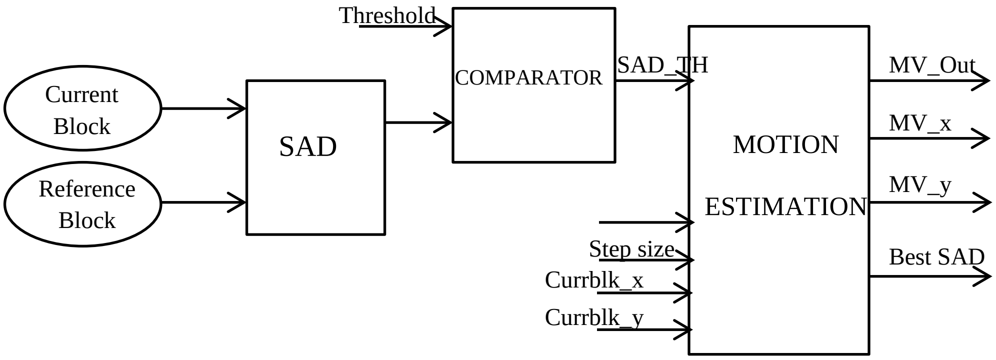


Fig. 4.1 Block diagram for the hardware implementation

The SAD block computes SAD between current and reference blocks and passes that value to the comparator. The comparator compares the SAD value with a threshold value and gives the result to the motion estimation block. Threshold value is chosen randomly and it depends on the type of application. If the SAD is less than the threshold value, we assume that the current block didn't move and we assign zero motion vectors for that block. If the SAD value is greater than the threshold value then the motion estimation block finds the motion vectors for the current block using 2D-logarithmic search algorithm explained in chapter 2. Motion vectors are given to the motion compensation block to get the motion compensated frame.

4.1 SAD Module

The hardware architecture for the SAD module is shown in the figure 4.2. The SAD architecture shown in Fig 4.2 computes the sum of absolute difference value of two blocks of size 4 x 4. It computes the difference of two image blocks pixel by pixel simultaneously, and then computes the absolute value of the differences. The SAD architecture has 16 difference blocks and 16 absolute blocks. The 16 absolute difference values are then added using the adder to get the sum of absolute difference. The accumulator is used for storing the intermediate results.

4.2 Motion Estimation Module

The block diagram for the motion estimation module is shown in Fig. 4.3. The motion estimation module implements the 2D-logarithmic search algorithm. This module has six sub modules, the diamond generate unit, the square generate unit, search range unit, SAD unit, sad5 comparator and sad9 comparator units. The search range unit determines the search area in which the best matched block is to be searched in the reference frame. The SAD unit gives the sum of absolute difference value between the current block and reference blocks.

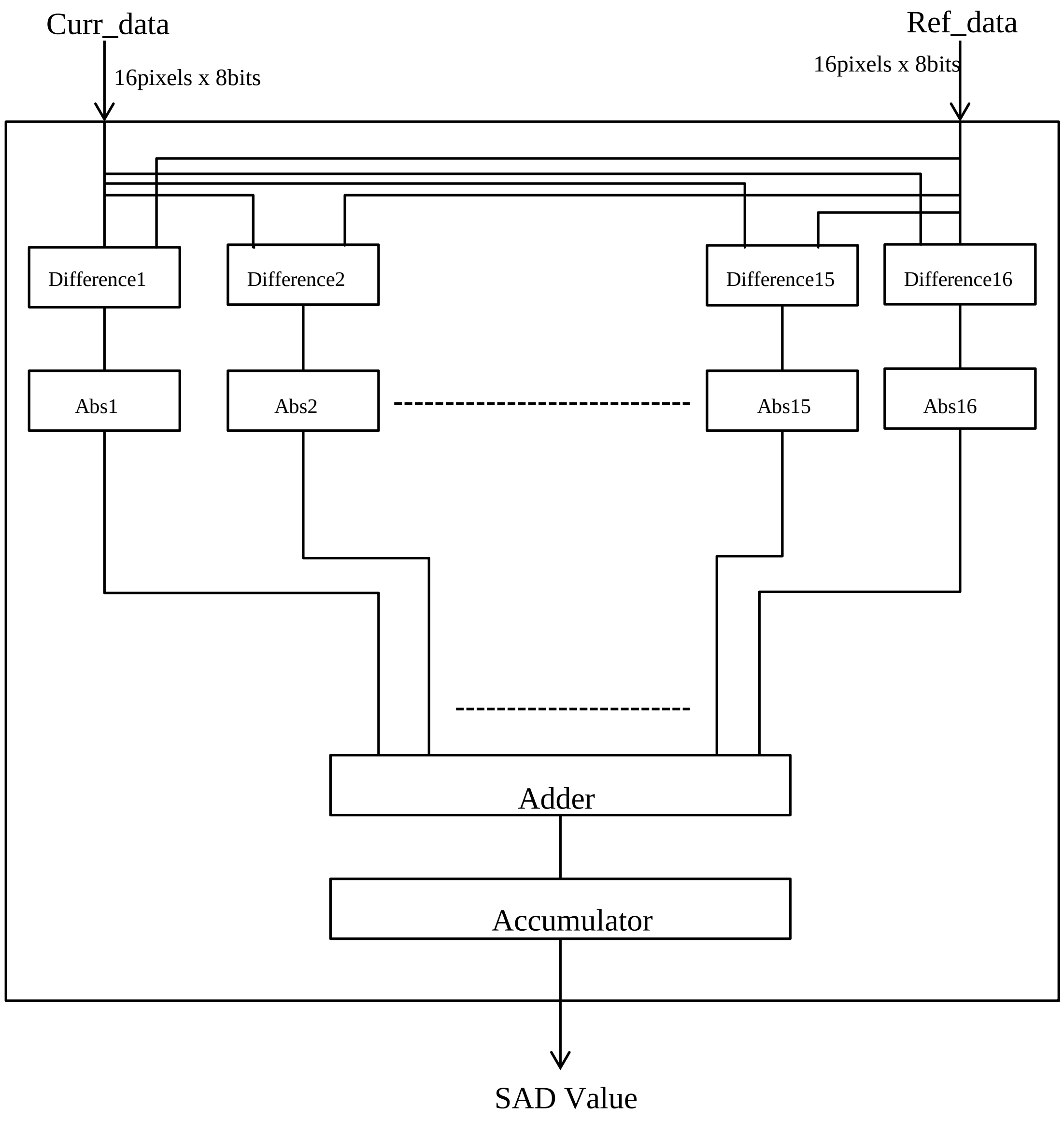


Fig. 4.2 Architecture for the SAD module

The diamond generate unit generates five blocks from the search range of reference frame. The five blocks are generated based on the step size and the position of the current block. SAD for the five blocks from reference frame are calculated with respect to the current block for which motion vectors need to be found. The SAD values

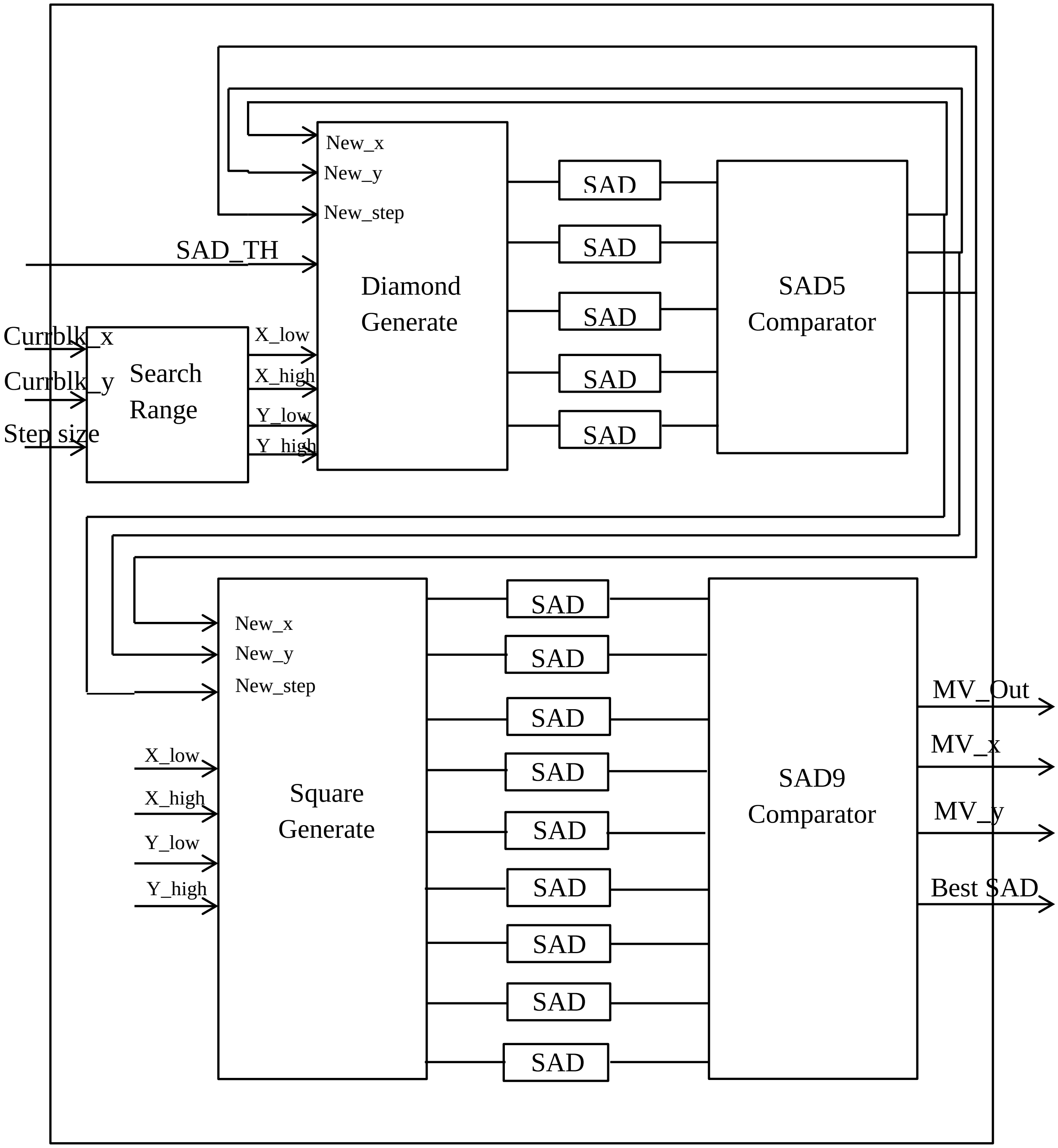


Fig. 4.3 Architecture for the 2D-logarithmic search algorithm

are then given to the SAD5 comparator to find the best matched block. SAD5 comparator compares the five SAD values and gives the new position of the better matched block and new step size to continue the process. If the new step size value is one, then the square generate unit is enabled, else the position of best matched block and step size are fed back to diamond generate unit and this process is repeated until the step size becomes one.

The square generate unit is enabled when the step size is one and it generates nine blocks from the reference frame. The eight blocks around the better matched block are chosen to find the best matching block for the current block. SAD for all the 9 blocks are calculated and are compared by the SAD9 comparator. SAD9 comparator gives the block with the minimum SAD value, which is the best matched block. It gives the motion vectors for the current block by subtracting the position of current block from the position of the best matched block.

4.3 Results

The proposed architecture has been implemented in VHDL, simulated and verified using Isim 12.1 tool. The VHDL code has been synthesized, placed and routed using Xilinx ISE 12.1 tool. The FPGA chosen is Xilinx Virtex-6 XC6VLX130T. Synthesis results are shown in the Table 4.1. Number of logic devices used in architecture is shown in Table 4.2.

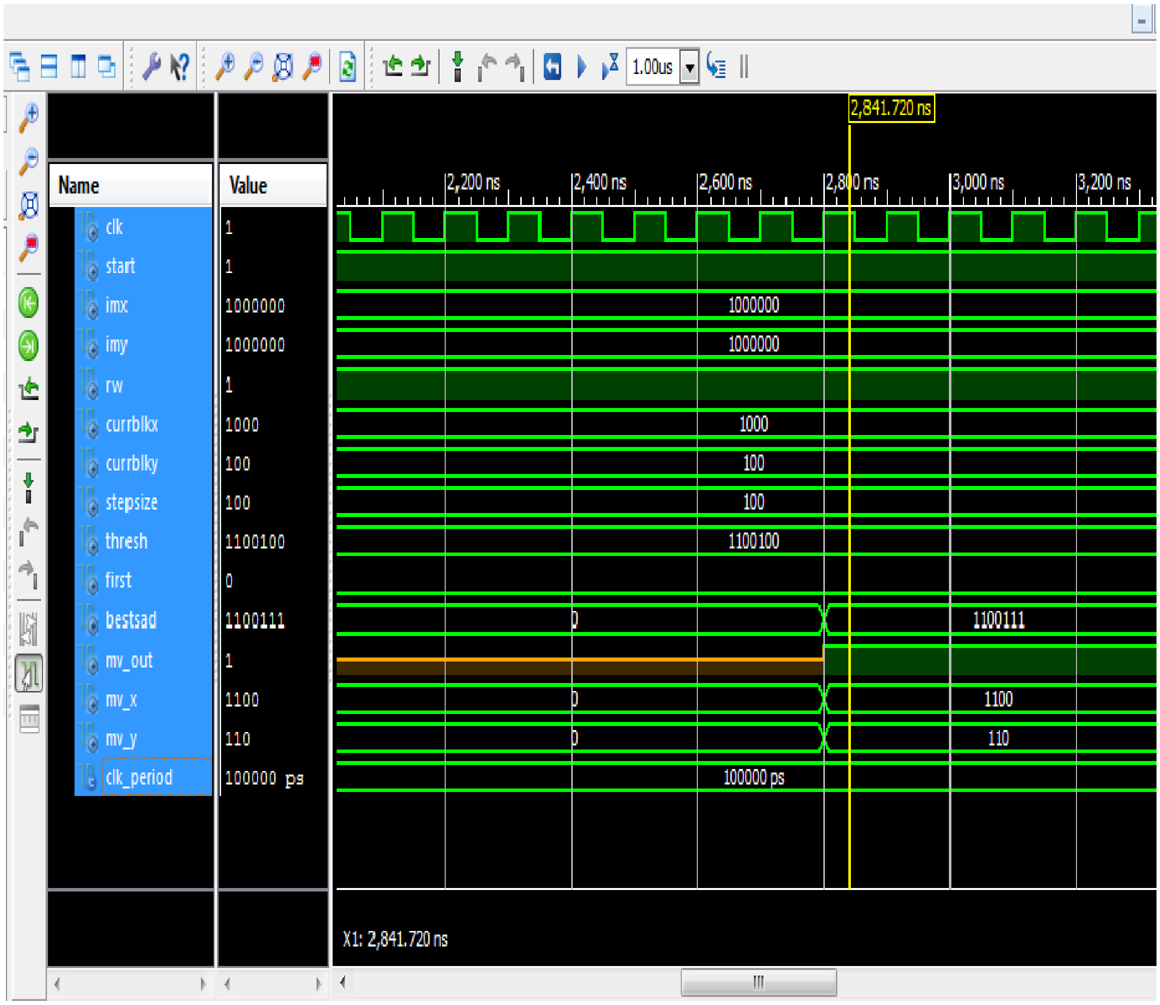
Resource Type	Usage	% Utilization
Slice Registers	4336	2
Slice LUTs	32767	40
Occupied Slices	11148	55
Bonded IOBs	63	26
BUFG/BUFGCTRLs	1	3

Table 4.1 Synthesis results for the 2D-Log search algorithm

Logic Type	Used
RAMs	4352
Adders/Subtractors	816
Registers	538
Comparators	43
Multiplexers	337
FSMs	1
Xors	1792

Table 4.2 Logic devices used in architecture

The proposed architecture has been simulated using ISIM tool and snapshot of the simulated waveform is show in the Fig 4.4. The gray level pixel values of current and reference frame images are stored in two different RAMs and are used when required. A test bench has been written to verify and simulate the operation of the VHDL code. The test bench provides the position of the block for which motion vectors are to be found, step size and threshold values. If the SAD of the current block is greater than the threshold value, then these inputs are provided to the motion estimation block. Block based motion estimation is done with block size of 4x4 pixels and step size of 4. Search range for the block is of size 32 x 32 pixels.



Name	Value
clk	1
start	1
imx	1000000
imy	1000000
rw	1
currblkx	1000
currblky	100
stepsize	100
thresh	1100100
first	0
bestsad	1100111
mv_out	1
mv_x	1100
mv_y	110
clk_period	100000 ps

X1: 2,841.720 ns

Fig. 4.4 Simulation of the motion estimation unit

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this thesis, the rate distortion function for the conditional motion estimation scheme is derived by modeling the encoded variables as standard probability density functions. Encoded variables DFD and frame difference are modeled as Gaussian i.i.d and Gauss markov process respectively. Rate distortion plot for the conditional motion estimation scheme is presented along with its lower bound. The plots suggest that while the practical R-D curve and its lower bound exhibit similar trend, there is a significant gap between them. The practical curve includes the cost of implementation such as quantization and the penalty associated with incorrect modeling of the PDFs associated with active and inactive pixels.

The hardware architecture for the implementation of 2D-logarithmic search algorithm is also proposed. The proposed architecture has been implemented in VHDL, simulated and verified using Xilinx ISE Design Tool Suite. The architecture has been synthesized and synthesis results are also presented.

The future work includes the accurate modeling of the probability density functions and including the effects of quantization of the residuals to reduce the gap between the theoretical and practical rate distortion curves. Area, power and performance tradeoffs for the proposed hardware architecture can be studied. The proposed hardware architecture can be extended for the conditional motion estimation process.

REFERENCES

[1] T.Berger. *Rate distortion theory*. Englewood Cliffs, NJ: Prentice Hall, 1971.

[2] G. B. Rath and A. Makur, “Sub block matching based conditional motion estimation with automatic threshold selection for video compression,” IEEE Transactions on circuits and systems for video Tech, vol. 13, no. 9 (2003): 914-924.

[3] R. Yarlagadda, “Rate Distortion analysis for adaptive threshold based conditional motion estimation schemes,” MS Thesis Wichita State University, May 2005.

[4] C. Shannon and W. Weaver. *The mathematical theory of communication*. University of Illinois, 1949.

[5] C. Shannon. *Coding theorems for a discrete source with fidelity criterion, Information and Decision Process*. McGraw-Hill,1960.

[6] H. C. Andrews, “Bibliography on rate distortion theory,” IEEE Transactions on Information Theory, vol. IT-17 (1971): 198-199.

[7] L. D. Davisson, “Rate-Distortion theory and application,” Proceedings of the IEEE, vol. 60, no. 7 (1972): 800-808.

[8] R. G. Gallager. *Information theory and reliable communication*. New York: Wiley, 1968.

[9] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley Series in Telecommunications, 2004.

[10] Y. Wang, J. Ostermann, and Y. Q. Zhang. *Video processing and communication*. Prentice Hall, Signal processing series, Alan V. Oppenheim,Series Editor, 2002.

[11] C. Stiller and J. Konard, “Estimating motion in image sequences,” IEEE Signal Processing Magazine, vol. 16, no. 4 (1999): 70-91.

[12] D. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” IEEE Transactions on Information Theory, vol. IT-19, no. 7 (1973): 471-480.

[13] A. Wyner, “Recent results in the Shannon theory,” IEEE Transactions on Information Theory, vol. IT-20, no. 1 (1974): 2-9.

[14] A. M. Rohit Puri and K. Ramchandran, “PRISM:A Video Coding Paradigm with Motion Estimation at the Decoder,” IEEE transactions on Image Processing, vol. 16, no. 10 (2007): 2436-2448.

[15] A. Wyner and J. ziv, “The rate distortion function for source with side information at the decoder,” IEEE Transactions on Information Theory, vol. 22, no. 2 (1976): 1-10.

[16] L. L. Zhen Li and E. J. Delp , “Rate distortion analysis of motion side estimation in Wyner-ziv video coding,” IEEE Transactions on Image Processing, vol. 16, no. 1 (2007): 98-113.

[17] Y. Oohama, “Rate-distortion theory for gaussian multiterminal source coding systems with several side informations at the decoder,” IEEE Transactions on Information Theory, vol. 51, no. 7 (2005): 2577-2593.

[18] S. Ghoneimy and S. F. Bahgat, “Rate-distortion function for nth order gaussian-markov process,” Circuits Systems Signal Process, vol. 12, no. 4(1993): 567-578.

[19] G. B. Rath and A.Makur, “Iterative least squares and compression based estimations for a 4-parameter linear global motion model and global motion compensation,” IEEE Transactions on Circuits and Systems for Video Tech, vol. 9 (1999): 1075-1099.

[20] S. Payyavula, “Automatic threshold selection using bayesian decision model for block based conditional motion estimation,” MS Thesis Wichita State University, May 2004.

[21] CCITT, Recommendation H.261: Video Codec for audiovisual services at p x 64 kbits/s, COM XV-R 37-E,1989.

[22] ITU-T Recommendation H.263, Video Coding of Narrow Telecommunication Channel at 64 Kbit/s, 1995.

[23] ISO/IEC JTC1 IS 11172-2(MPEG-1),Information Technology –Coding of Moving pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s, 1993.

[24] ISO/IEC JTC1 IS 13818-2 (MPEG-2), Information Technology – Generic Coding of Moving pictures and Associated Audio Information, 1996.

[25] ISO/IEC JTC1/SC29/WG11, MPEG-4 version 2 Visual Working Draft Revision 2.0. N1993, 1998.

[26] S. Zhu and K. K. Ma, “A new diamond search algorithm for fast block matching motion estimation,” IEEE transactions on Image Processing, vol. 9, no. 2 (2000): 287-290.

[27] B. J. Bunin, “Rate distortion functions for Gaussian Markov process,” The Bell System technical journal, (1969): 3059-3075.

[28] Maher E. Rizkalla, Paul Salama, Mohamed El-Sharkawy and Modukuri Sushmitha, “Hardware Implementation of Block-based Motion Estimation for Real Time Applications,” Journal of VLSI Signal Processing 49 (2007): 139–159.

[29] M. Kthiri, P. Kadionik, H. Lévi, H. Loukil, A. Ben Atitallah, N.Masmoudi, "An FPGA implementation of motion estimation algorithm for H.264/AVC," I/V Communications and Mobile Network (ISVC), (2010): 1-4.

[30] S. Ramachandran, S. Srinivasan, "FPGA implementation of a novel, fast motion estimation algorithm for real-time video compression," FPGA '01 Proceedings of the 2001 ACM/SIGDA ninth international symposium on Field programmable gate arrays.

[31] Michael McErlean, "An FPGA Implementation of Hierarchical Motion Estimation for Embedded Object Tracking," IEEE International Symposium on Signal Processing and Information Technology, (2006): 242-247.

[32] Suk-Ju Kang, Dong-Gon Yoo, Sung-Kyu Lee, Young Hwan Kim, "Hardware implementation of motion estimation using a sub-sampled block for frame rate up-conversion," Soc Design Conference(ISOCC), (2008): 101-104.

[33] R. R. Colenbrander, A. S. Damstra, C. W. Korevaar, C. A. Verhaar, A. Molderink, "Co-design and Implementation of the H.264/AVC Motion Estimation Algorithm Using Co-simulation," Digital System Design Architectures, Methods and Tools, (2008): 210-215.

[34] Junzhong Zhao, Lin You, Shanli Sun, Maozhi Xu, "Rate Distortion Theory and its application in Intrusion Detection," International Conference on Neural Networks and Brain, (2005): 759-762.