A Review Paper on Implementation & Comparative Analysis of Motion Estimation Algorithm in Video Compression

N.K. Nakum, A.M.Kothari

Abstract— This paper is a review of the block matching algorithms. The motion estimation algorithm is one of the most important issues in the video coding standards. To achieve a high compression ratio in coding video data, a method known as Motion Estimation (ME) is often applied to reduce the temporal redundancy between successive frames of a video sequence. This paper shows implementations and comparison of different types of block matching algorithms that range from the very basic Exhaustive Search to the recent fast adaptive algorithms.

Index Terms— Block matching, motion estimation, video compression, H.261.

I. INTRODUCTION

Video has huge redundant information which must be exploited to be stored and transmitted efficiently. The common technique to achieve this goal is known as motion estimation. Video compression is a critical component when transmitting or storing digital video sequences. Hybrid video compression, exploits temporal redundancy by motion compensation and spatial redundancy by DCT transformation, and it has been widely adopted many international standards. Video has wide redundant information which must be removed to transmit efficiently. Knowledge of the motion is not available from a video data and must be deduced using computationally intensive algorithms. For efficient handling of motions with variety of contents, the need for adaptive motion estimation methods is inevitable.

The underlying supposition behind motion estimation is that the patterns corresponding to objects and background in a frame of video sequence move within the frame to form corresponding objects on the subsequent frame.

The idea behind block matching is to divide the current frame into a matrix of 'macro blocks' that are then compared with corresponding block and its adjacent neighbors in the previous frame to create a vector that stipulates the movement of a macro block from one location to another in the previous frame.

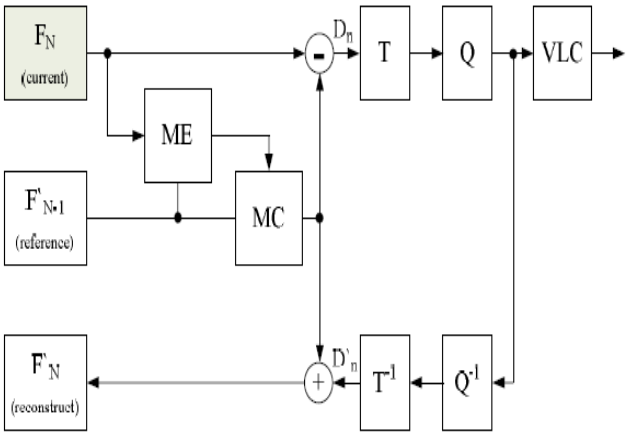

Figure 1. Video compression process flow

The basic flow of the entire compression-decompression process is depicted in Fig. 1. This includes the motion estimation (ME) and motion compensation (MC). The hybrid video encoder can be divided into the forward and backward paths. In the forward path, estimates the motion in the current frame with respect to a previous frame. A motion compensated image for the current frame is then created that is built of blocks of image from the previous frame. Here for each macro block (MB), motion estimation (ME) is firstly conducted to find the best matching position. Then the motion compensation (MC) is scheduled to generate the matching reference block. Thirdly, the original MB is subtracted by the reference block to get the residuals Dn. Fourthly, the obtained residuals are transformed (T) and quantized (Q), and the calculated coefficients are encoded by the variable length code (VLC) and ready for transmission. In the backward path, the generated coefficients are firstly inverse quantized (Q^{-1}) and inverse transformed (T-1) to get the reconstructed residuals D`n, which is then added with the reference block obtained by MC to generate the reconstruct MB, which will be used for future ME process.

H.261 (last modified in 1993) is the video compression standard included under the H.320 umbrella (and others) for videoconferencing standards. H.261 is a motion compression algorithm developed specifically for videoconferencing, though it may be employed for any motion video compression task. H.261 allows for use with communication channels that are multiples of 64 kbps (P=1,2,3...30.), the same data structure as ISDN. H.261 is sometimes called Px64.

An overview of the H.261 CODEC, taken from the ITU reference documentation, shows the major components used to code and decode the bitstreams.

Manuscript Received on November, 2012.

N.K.Nakum Student, Master of Engineering in Electronics and Communication at Atmiya Institute of Technology and Science, (GTU). Rajkot, India.

A.M.Kothari, Assistant Professor in Electronics and communication department at Atmiya Institute of Technology and Science, (GTU). Rajkot, India.

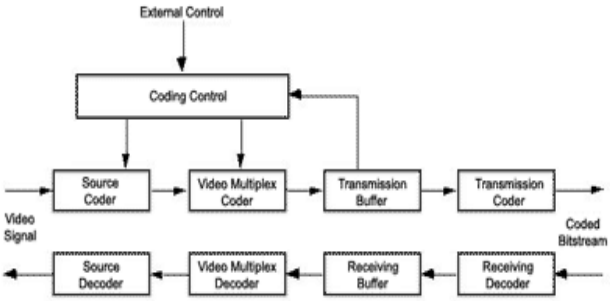

Figure 2: H.261 block diagram from ITU recommendation

In essence, this has the effect of standardizing the bit stream in and the bit stream out of the decompression functional block. The encoder design is unspecified, except of course that it must be compatible with the recommended decoder.

II. MOTION ESTIMATION ALGORITHMS

Motion estimation is the process which generates the motion vectors that determines how each motion compensated prediction frame is created from the previous frame. It examines the movement of objects in an image sequence to try to obtain vectors representing the estimated motion. Motion compensation uses the knowledge of object motion obtained to achieve data compression.

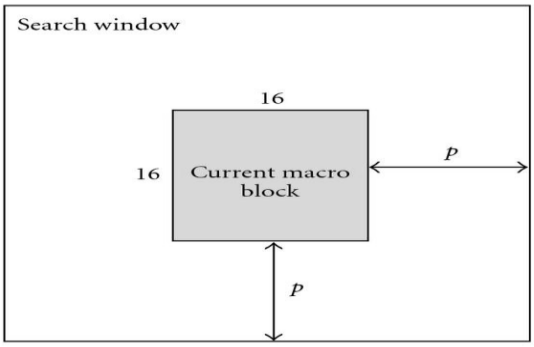

Figure 3. Block Matching a macro block of side 16 pixels and a search parameter p of size 3 pixels.

Considering a video sequence containing moving objects, we estimate the displacement vector of each object in the image plane by the technique of Full Search Block Matching Algorithm. The current frame is divided into a matrix of "macro-blocks" that are then compared with the corresponding block and its adjacent neighbours in the previous frame. This enables to create a vector that stipulates the movement of a macro-block from one location to another in the previous frame. This movement, calculated for all the macro-blocks included in a frame, represents the motion estimated in the current frame. The search area for a good macro-block match is constrained up to p pixels on all fours sides of the corresponding macro-block in the previous frame. This "P" stands for the search parameter. Larger motions require a larger "P"; the larger the search parameter is, the more computationally expensive the process of motion estimation becomes. The idea is depicted in Figure 3. The matching of one macro-block with another is based on the output of a cost function. The macro-block that results in the least cost is the one that closely matches to current block.

A. Full Search or Exhaustive Search (ES) Algorithms

The root node of this tree is labeled s0 to indicate that this is the starting state for our search. The node in the center that is three levels down and which has a g to its right is our goal node. For purposes of illustration we will pretend that this is the total search tree. This is, of course, pretty unrealistic. Most search trees will have hundreds if not millions of nodes. Our little example has only 23 nodes or 22 states that can be reached from the start node; the branching factor from any node varies between one and three, and, the goal node is only three moves away from the start node.

This algorithm, also known as Full Search, is the most computationally expensive block matching algorithm of all. This algorithm calculates the cost function at each possible location in the search window. As a result of which it finds the best possible match and gives the highest PSNR amongst any block matching algorithm. Fast block matching algorithms try to achieve the same PSNR doing as little computation as possible. The obvious disadvantage to ES is that the larger the search window gets the more computations it requires.

B. DWP Algorithm

The DWP method proposed in. A. Jukan et al (2004) as capable of:

1. Handling Multiple constraints without usage of weights.
2. Enabling services differentiated routing and wavelength reallocation.
3. Finding of multiple candidate paths among which the best one can be chosen based on routing objectives. Usage of decentralized instead on centralized global network state update.

C. Enhanced DWP Algorithm

The MDWP Algorithm overcomes the disadvantages of DWP algorithm by implementing the following concepts Predefined Routes & Dynamic wavelength Creating Check Points at Each Node Buffering for Future use.

Predefined Routes and Dynamic Wavelength:

The Route between the source and the destination is going to be specified in that particular source node itself but the wavelength is going to be selected dynamically. So by using this method the number of packets that reach the destination can be much reduced and thus the complexity can also be reduced. By doing so we are able to reduce the time required for the convergence of a path.

Creating Check Points at Each Node:

Here we are going to create check points at each node, so that the packets that do not satisfy the constraints are going to be blocked from reaching the destination. This concept also reduces the time required for convergence of a path.

Buffering for future Use:

Here if our network state is going to vary after x time units and a service between source a and destination b has taken only y time units, where x>>y. Then immediately a service request for the same source and destination arrives it will be time consuming to select a path once again. Since the network state has not changed it is better to allocate the same path that has been used earlier

Concept of Wavelength Rerouting:

In a wavelength routed WDM network, a light path needs to be

wavelength continuous this constraint results in inefficient utilization of wavelength channels (G.Mohan et al 1996).improving channel utilization is an important problem in this type of network. Wavelength rerouting is one possible solution to this problem. .wavelength rerouting accommodates a new connection request by migrating a few existing light paths to new wavelengths while maintaining their path. In addition to this we are also going to consider only the constraints required for that particular service i.e. if a particular service have only a constraint for delay then only the delay parameter is going to be updated at each node so that the computational complexity at each node is considerably reduced.

Wavelength rerouting

Some basic operations that can be used for migrating a light path have been presented in. K.C.Lee et al (1996).Move to vacant wavelength retuning (MTV-WR) moves a light path to a vacant wavelength on the same path. It can greatly reduce the disruption period. MTV-WR operation has advantages of both MTV and WR operations while overcoming their drawbacks. Now, we briefly explain the implementation of this operation. A central controller is used for sending control messages to set up, migrate, and release light paths. The following steps are used for light path migration C.Siva Ram Murthy et al (2002).

1. The controller sends control messages to the intermediate switches(routing nodes) on the path of the rerouted light path. These messages are used to set the state of a switch such that the new wavelength is switched from an inbound link to an appropriate outbound link. Then, the source node prepares to switch data transmission from the old wavelength to the new wavelength.
2. The source node appends an end-of-transmission (EOT) control packet after the last packet on the old wavelength and holds the first packet on the new wavelength for a guard time. The EOT packet is used to inform the destination node that the data transmission via the old wavelength has ended and data will soon arrive via the new wavelength. The guard time prevents data from being lost during the transient period of light path migration.
3. The source node tunes its transmitter to the new wavelength and, after the end of the guard time, starts transmission via the new wavelength. Upon detecting the EOT packet, the destination node tunes its receiver to the new wavelength and becomes ready for receiving data via the new wavelength.

Rerouting and minimization of incurred disruption due to rerouting in a wide area all optical wavelength division multiplexed (WDM) network with random circuit arrivals and departures. One limitation of such a network is the wavelength constraint imposed by the all-optical cross-connect switches which do not allow a circuit to be placed on a no wavelength-continuous route. Wavelength rerouting is proposed to rearrange certain existing circuits to create a wavelength-continuous route in order to accommodate a new circuit. To reduce the disruption period, move-to-vacant wavelength retuning (MTV_WR) is used as the basic operation of circuit migration.(Siva Ram Murthy et al (2002)), and K.C.Lee et al (1996) in which a circuit is moved to a vacant wavelength on the same path, and parallel MTV_WR rerouting is used to reroute multiple circuits.

Scheme used is MTV_WR

Guard time depends on three factors

- The switching time of optical Tx and Rx
- The processing time of detecting the End-of- transmission at destination
- The differential propagation delay of two wavelength W_2 and W_3.

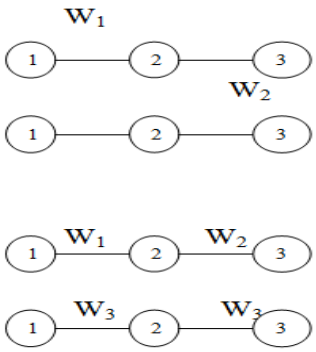

Figure 4. An Example Showing the Benefit of Wavelength Rerouting

D. Three Step Search (TSS)

This is one of the earliest attempts at fast block matching algorithms and dates back to mid 1980s.

At about the same time as Jain & Jain's introduction of the 2-D Logarithmic (TDL) search Koga et al. introduced a similar algorithm known as the Three Step Search (TSS). As with the TDL search, a number of positions around a Centre are tested and the position of minimum distortion becomes the Centre of the next stage. The Three Step Search, however, tests eight points around the Centre instead of four

The general idea is represented in Figure 4. It starts with the search location at the center and sets the 'step size' S = 4, for a usual search parameter value of 7. It then searches at eight locations +/- S pixels around location (0,0). From these nine locations searched so far it picks the one giving least cost and makes it the new search origin. It then sets the new step size S = S/2, and repeats similar search for two more iterations until S = 1. At that point it finds the location with the least cost function and the macro block at that location is the best match. The calculated motion vector is then saved for transmission. It gives a flat reduction in computation by a factor of 9. So that for p = 7, ES will compute cost for 225 macro blocks whereas TSS computes cost for 25 macro blocks.

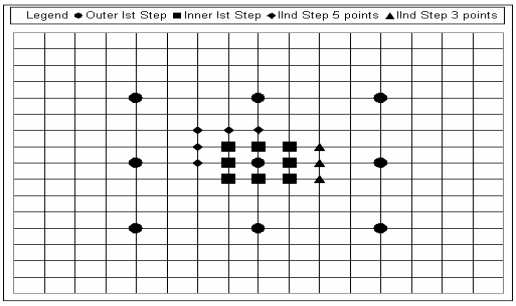

Fig. 4. New Three Step Search block matching

The idea behind TSS is that error surface due to motion in every macro block is unimodal. A unimodal surface is a bowl shaped surface such that the weights generated by the cost function increase monotonically from the global minimum.

E. Diamond Search (DS)

DS [7] algorithm is exactly the same as 4SS, but the search point pattern is changed from a square to a diamond, and also there is no limit on the number of steps that the algorithm can take. DS uses two different types of fixed patterns, one is Large Diamond Search Pattern (LDSP) and the other is Small Diamond Search Pattern (SDSP). These two patterns and the DS procedure are illustrated in Fig. 5.

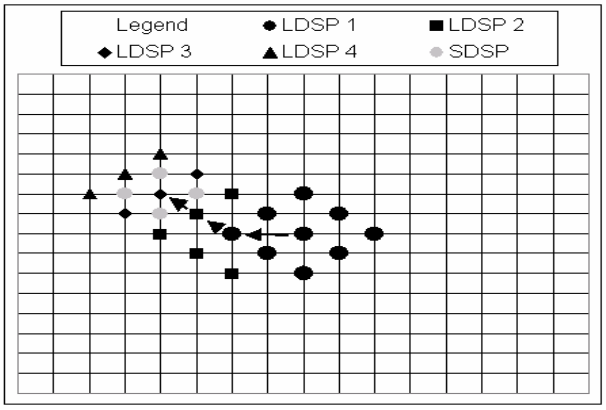

Fig. 5. Diamond Search procedure.

Just like in FSS, the first step uses LDSP and if the least weight is at the center location we jump to fourth step. The consequent steps, except the last step, are also similar and use LDSP, but the number of points where cost function is checked are either 3 or 5 and are illustrated in second and third steps of procedure shown in Fig.9. The last step uses SDSP around the new search origin and the location with the least weight is the best match. As the search pattern is neither too small nor too big and the fact that there is no limit to the number of steps, this algorithm can find global minimum very accurately. The end result should see a PSNR close to that of ES while computational expense should be significantly less.

F. Four Step Search (4SS)

Similar to NTSS, 4SS [6] also employs center biased searching and has a halfway stop provision. 4SS sets a fixed pattern size of S = 2 for the first step, no matter what the search parameter *p* value is. Thus it looks at 9 locations in a 5x5 window. If the least weight is found at the center of search window the search jumps to fourth step. If the least weight is at one of the eight locations except the center, then we make it the search origin and move to the second step. The search window is still maintained as 5x5 pixels wide.

Depending on where the least weight location was, we might end up checking weights at 3 locations or 5 locations. The patterns are shown in Fig 6.

Once again if the least weight location is at the center of the 5x5 search window we jump to fourth step or else we move on to third step.

The third is exactly the same as the second step. In the fourth step the window size is dropped to 3x3, i.e. S = 1. The location with the least weight is the best matching macro block and the motion vector is set to point o that location. A sample procedure is shown in Fig 6. This search algorithm has the best case of 17 checking points and worst case of 27 checking points.

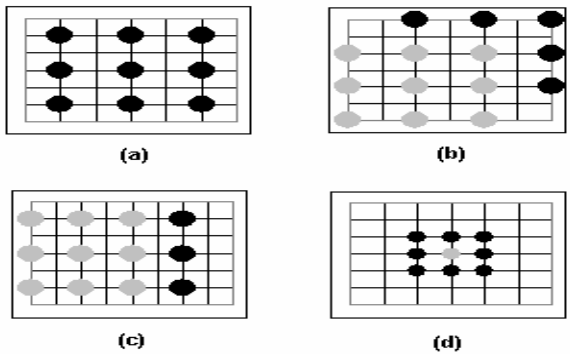

Fig. 6. Search patterns of the FSS. (a) First step (b) Second/Third step (c)Second/Third Step (d) Fourth Step

III. SUMMARY

Since last so many years of technology up gradation we have seen the growth of wide acceptance of multimedia. Video compression plays an important role in archival of entertainment based video as well as real-time reconnaissance / video conferencing applications. The three-step search (TSS) algorithm for block-matching motion estimation, due to its simplicity, significant computational reduction and good performance, has been widely used in real-time video applications.

IV. CONCLUSION

Video technology promises to be the key for the transmission of motion video. A number of video compression techniques and standards have been introduced in the past few years. The algorithms explained in this paper gives us the overview of most widely used algorithms for video compression. Motion Estimation Algorithms for Video Compression develops the theoretical framework for a new fast search algorithm.

V. ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their comments and suggestions.

REFERENCES

[1] Aroh Barjatya, *Student Member, IEEE* "Block Matching Algorithms For Motion Estimation",DIP 6620 Spring 2004 Final Project Paper 2.

[2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proc. NTC 81, pp. C9.6.1-9.6.5, New Orleans, LA, Nov./Dec. 1981.

[3] Amish Tankariya , Prof. Mukesh Tiwari and Prof. Jaikaran Singh Department of Electronics & Communication Engineering, SSSIST- Sehore, Bhopal,, (M.P), "International Journal on Emerging Technologies" (IJET)(0975-8364).

[3] 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, pp.287-290, Feb. 2000.

[4] K. H.-K. Chow and M. L. Liou, "Genetic motion search algorithm for video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 440–445, Dec. 1993.

[5] Liang-Wei Lee, Jhing-Fa Wang, Jau-Yien Lee, andJung-Dar Shie," Dynamic Search-Window Adjustment and Interlaced Search for Block-Matching Algorithm" IEEE Transactions on Circuits and Systems for video Technology. VOL. 3. NO I . FEBRUARY 1093.

AUTHOR PROFILE

Mr.N.K.Nakum, Student, Master of Engineering in Electronics and Communication at Atmiya Institute of Technology and Science, (GTU).

Prof.A.M.Kothari, Assistant professor in Electronics and communication department at Atmiya Institute of Technology and Science, (GTU).