Motion Estimation in H.264/AVC by 2D Logarithmic Search Algorithm

Alpana S Uppin[1] Prof. Sandhya S Bevoor[2]
[1]P.G. Student [2]Assistant Professor
[1,2]Department of Electronics and Communication Engineering
[1,2]Maratha Mandal Engineering College Belgaum Karnataka India

Abstract— The H.264 which is also named Advanced Video Coding compression standard in video processing which results in good quality of video. The H.264 standard in video compression allows fast block matching algorithm to estimate the displacement of pixels in a search area limit which reduces the complexity of computing the motion vectors. In this paper we discussed on a 2D logarithmic search algorithm which gives the best match when compared to other block matching algorithm. Next, we check quality of video of predicted frames by computing PSNR. Thus using SP-line interpolation we increase the PSNR value for a set of specified frames.

Key words: H.264/AVC, Motion estimation, 2D logarithmic search, interpolation, Peak signal-to-noise ratio (PSNR)

I. INTRODUCTION

H.264/AVC is a video compression licensed standard format which is highly efficient in recent days. As per the ITU standard, H.264 is frequently used than other formats for downloading a video, compression etc,. In video compression, we can perform motion estimation operation by suitable block matching algorithm which gives best match compared to other. In Block matching algorithm we often divide the current frame in smaller blocks. Then we examine the best match for individual block in reference frame. Further we detect the motion vector from current and previous frame. Block size is typically of 16*16 pixel.Motion estimation simply determines the estimated motion in form of vector representation. Motion compensation uses thi**s result to get compressed data.**In this paper we illustrate fast and efficient block matching method for motion estimation by 2D logarithmic search algorithm by taking a foreman video where each frame size is (288*352). The maximum number of frames taken is upto 300. User can enter any number less than 300 and play the video to the given number. The video is played by movie player that is in built on MATLAB software and run it.There are many accurate and exact approaches for block matching algorithms such as Full search where the time is being wasted which is applicable to whole window and rest are 3 step search which takes total of 9 points with center distance of w/2, after finding the best match it takes that as center and the distance is halved and select 8new points. In conjugate direction search, begin from center point, then we find for best match with its neighbouring horizontal points. Further we hunt for 1 point along with best match. If there is no change from horizontal best match, we repeat for vertical points. Again if we note there is no change then we select that point as best. In case of diamond search we assume 9 points to form a big diamond with center in search region. This best match becomes a center. If this best match occupies in center, consider 5 points that forms a small shape diamond, then search for best match. Further we improve the PSNR results for both frames and a video by showing the PSNR plot. The detailed explanation of our proposed algorithm is described in upcoming sections.

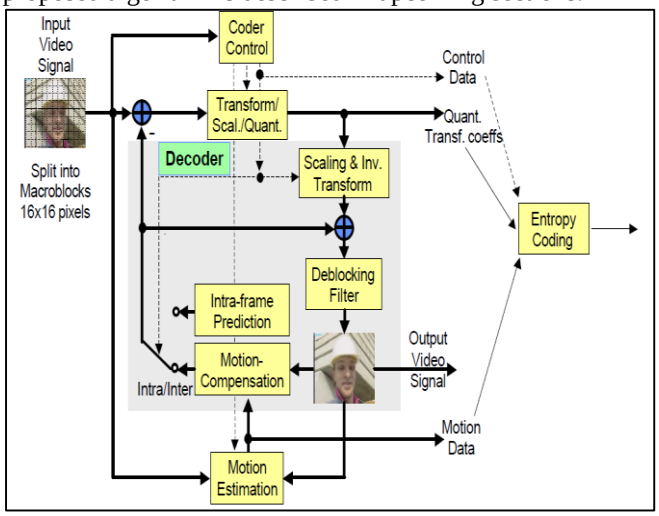

Fig 1: Architecture of H.264/AVC video coder

Fig1 shows the complete architecture of H.264/AVC video coding scheme.

Fig2 shows a picture of block matching within a specified search limit.

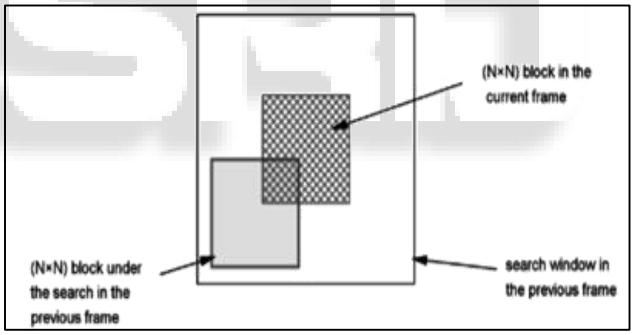

Fig. 2: Block matching in search limit

II. IMPLEMENTATION

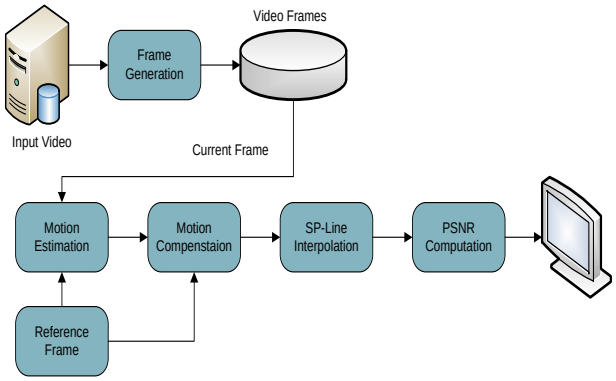

Fig. 3: Architectural proposed system.

The architectural set up of our proposed system is arranged as shown in above fig3. Accordingly, our very first step is video as an input. There are several frames which are collectively used to form a single video that are usually done

by frame generation. These frames are stored in a memory buffer for performing important operations in a video compression based techniques. Our next idea is to consider a reference frame, usually we can call it as a frame with zero amount of motion. Further the first frame are considered as reference frame and second frame as current and so on till the frame number. In our application we are using foreman frames which is of (288*352) size where rows=288(height), columns=352(width). The original frames are then stored in a dataset and a query is selected that is a file of .yuv extension since this format allow us to divide the video into frames easily. In existing days, yuv is used to mention the file formats that are encoded using YCbCr. After selecting this file of yuv, enter the number of frames from the keyboard and the software uses some delay for processing these frames. Once it is being processed we play the video in video player where it continuously keeps on changing the motion of object upto specified frames in the form of vector representation which is done by logarithmic search algorithm where x-axis denotes columns and y-axis indicate the rows. Finally, when done with motion estimation by a fast block matching algorithm followed by motion compensation we get a reduced PSNR value. Only by one of the interpolation method that is SP- Line gives a better and increased PSNR value. In order to have better vision we plot a graph of percentage versus the PSNR values that is got after interpolation.

III. FAST BLOCK MATCHING ALGORITHM

This algorithm has certain steps to be followed theoretically which is mentioned below

- This method was first developed by Jain & Jain.
- Mark the centre point and its neighbouring 4 points.
- Distance from centre – w/2.
- If centre position becomes best match decrease the step size by 2.
- Examine 5 other points which is previous best.
- When distance from centre becomes unity, consider all 9 points, find best match. Stop.

IV. FLOW PROCESS OF PROPOSED ALGORITHM

The flow of our proposed work in systematic and serial order as per in section II is followed in terms of flow diagram as shown below in fig4.

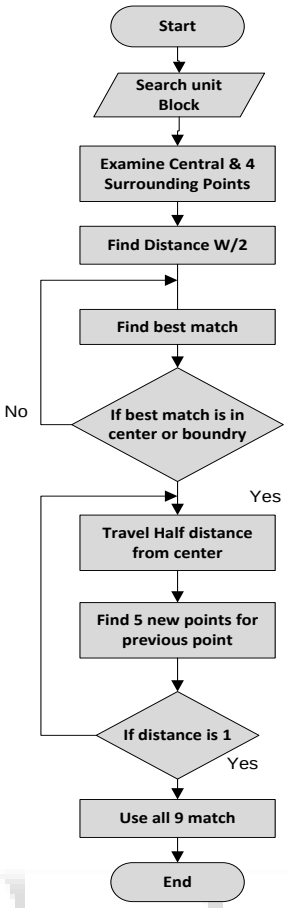

Fig. 4: Flow of logarithmic search

V. RESULT ANALYSIS

As seen from fig5 we have taken nearly 10 frames from reference to current which is of RGB colour type format of (288*352) pixel size. Further we compensate the motion to original frame by approximating it. PSNR for compensation and interpolation is calculated. In fig5 we get the clear idea of how motion vectors are represented in form of vector fields in adjacent pixels. PSNR of video without interpolation is 31.6077 and with interpolation is 36.3754

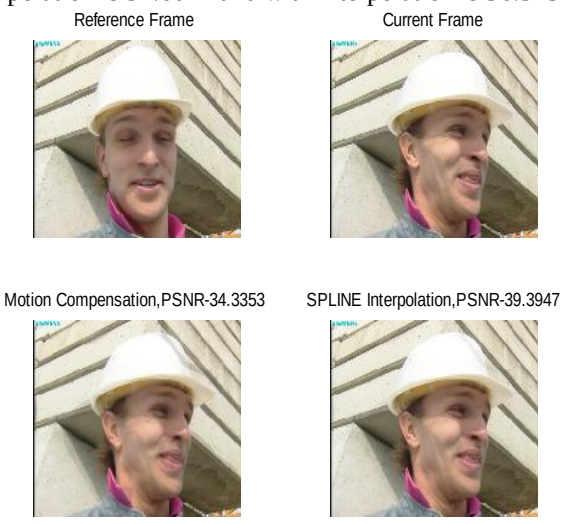

Fig. 5: Simulation results of first 10 frames of a foreman video

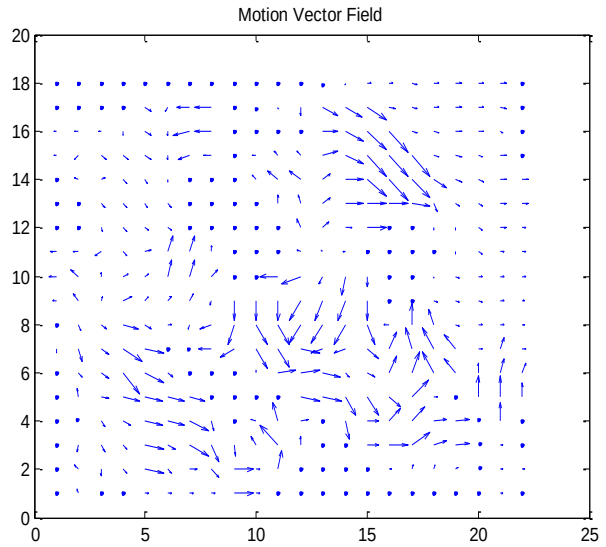

Fig. 5: Logarithmic Search Result

VI. MOTIVATION

Motion estimation is one of the most consequential as well as computationally exhaustive block of a video encoder. Block matching algorithm is used for motion estimation in various video coding. Full search is highly computational so we use fast block matching algorithm techniques that utilize less search points and complexity if finding the best match. Logarithmic search algorithm is most efficient in terms of the computational speed & achieves good PSNR by increasing the video quality.

VII. FUTURE SCOPES

With this advantages of all fast block matching algorithm over full search estimation, we can also compress the video that requires more amount of space to reduce the size of a video file. Further encoding and decoding of frames can be done by differencing and adding of these frames as per the architectural block diagram of video coder of H.64/AVC. Work can also be done on the extension video compression standards to have HD quality video with reduced computing time.

VIII. CONCLUSION

Various block matching algorithms are analysed. This paper mainly focuses on 2D logarithmic search algorithm which gives off best match. In addition, Motion estimation is an important phenomenon in video processing. Thus from previous section we observe and the simulation results are clear that the quality of predicted video frames are greatly improved is observed through PSNR & also increase PSNR by performing interpolation. This is simulated on a MATLAB tool which includes video sequences, search limit and algorithm used. This algorithm also reduces computation time as compared to Full search algorithm & is also known as Fast Block matching algorithm that is widely accepted. Further the idea can be extended to compress a video file which occupies less storage of memory for future scope.

REFERENCES

[1] Hussain Abo Surrah, Mohd. Junedul Haque " A Comparative Approcah for Block Matching Algorithms used for Motion Estimation". IJCSI International Journal of Computer Science Issues, Vol.11, Issue 3,No 2, May 2014.
[2] Saloni R Mistry, Heni S Modi, Rahul N Gonnwala " Logarithmic search for motion estimation" International journal of Industrial Electronics & Electrical Engineering ISSN:2347-6982, Vol-2, Issue-6, June 2014.
[3] Stanley H Chan, Dung T Vo & Truong Q Nguyen " Subpixel Motion estimation without interpolation" ICASSP 2010.
[4] S. Ashwin, S. Jayanthi Sree, S. Aravind Kumar "Study of the contemporary motion estimation techniques for video coding" IJRTE ISSN: 2277-3878, Vol 2, Issue-1, March 2013.
[5] P. Muralidhar & C.B. Rama Rao, "Efficient block matching algorithms for motion estimation in H.264/AVC " ICTACT Journal on image & video processing, Feb 2015, Vol 5, Issue 3, ISSN: 0976-9102.
[6] M.Usha, "Motion detection in compressed video using macro block classification", Advanced computing international journal (ACIJ), Vol5, No 2/3,May 2014.
[7] Deepa Mary Thomas, Subha Varier, "A Novel based approach for finding motion estimation in video compression" , International journal of advanced research in couter & communication Engg, Vol 1, Issue 8, Oct 2012, ISSN: 2319-5940.
[8] D.V. Manjunatha, Dr. Sainarayanan, "Comparison and implementation of fast block matching motion estimation algorithms for video compression", D.V.Manjunatha et al./ International Journal of Engg. Science & technology (IJEST), ISSN: 0975-5462, Vol 3, No 10, Oct 2013.
[9] Piyush N Korat, Dr. Dhaval R Bhojani, "A New hybrid block based motion estimation algorithm for video compression", IJAREETIE, Vol 3, Issue 5, ISSN: 2320-3765, May 2014.
[10] A. Ahmadi, M.M. Azadfar, "Implementation of fast motion estimation algorithms and comparsion with full search method in H.264", IJCSNS, Vol8, No 3, March 2008.

