Full Search Algorithm for Movie Files

Sangeeta Mishra
TCET,Mumbai
Shyamnaran Thakur Marg
Thakur Village, Kandivli(E)-101

Sudhir Sawarkar
DMCE
Airoli

ABSTRACT
In this paper, we describe the search algorithm for movie files .The algorithm is first divides the video clip into different frames. Full search algorithm in video compression. The algorithm takes 16 pixels window left ,right ,upper and lower. Find out the best possible matching block of current frame to reference frame ,then it find out the motion vectors. Since algorithm works pixel by pixel basis,for each window it performs 33X33X8X8 multiplications and 33X33X8X8x2 subtractions so speed is low.

Keywords
BMA, Motion estimation, Frames , MSE.

1. INTRODUCTION

Video compression is vital for efficient storage and transmission of digital signal. The hybrid video coding techniques based on predictive and transform coding are adopted by many video coding standards such as ISO MPEG-1/2 and ITU-T H.261/263. Motion estimation and motion compensation is a predictive technique for exploiting the temporal redundancy between successive frames of video sequence. Block matching techniques are widely used motion estimation method to obtain the motion compensated prediction. By splitting each frame into macroblocks, motion vector of each macroblock is obtained by using block matching algorithm (or motion estimation algorithm). In order to get motion vector of each macroblock, the most obvious and simplistic method is full search algorithm. All possible displacements in the search window are evaluated using block-matching criteria (cost function). The advantage of full search is that we can find the absolute optimal solution. However, its high computational complexity makes it impossible for real-time implementation. Because the computational complexity of video compression, the compression efficiency and the compression quality is determined by the motion estimation algorithm, development of Fast Motion Estimation Algorithm for real-time application becomes compelling.

The computational complexity of a motion estimation technique can then be determined by three factors: 1. search algorithm. 2. cost function/evaluate function. 3.search range parameter p. We can reduce the complexity of the motion estimation algorithms by reducing the complexity of the applied search algorithm and/or the complexity of the selected cost function. A full search algorithm evaluates all the weights in the search window, and a more efficient, less complex search algorithm will decrease the search space. We will identify and evaluate the recent and widely used fast estimation algorithms, especially in hybrid estimation algorithms: how they get the trade-off between
video quality and compression efficiency? Based on this, we will propose our new motion estimation algorithm produced our new video encoder. The Full Search or exhaustive search algorithm (FS) acts as a benchmark for evaluating the efficiency of all existing fast block-matching motion estimation algorithms .

MOTION ESTIMATION

MOTION estimation using a block-matching algorithm (BMA) is widely used in many motion-compensated video coding systems, such as those recommended by the H.261 and MPEG standards [1], [2], to remove interframe redundancy and thus achieve high data compression. In a typical BMA, the current frame of a video sequence is divided into non overlapping square blocks of pixels, say, of size 16 by 16.For each reference block in the current frame, BMA searches for the best matched block within a search window of size in the previous frame, where stands for the maximum allowed displacement. Then the relative position between the reference and its best matched block is represented as the motion vector of the reference block. A nonnegative matching error function is defined over all the positions to be searched.

Block-based matching method is the most widely used motion estimation method for video coding since pictures are normally rectangular in shape and block-division can be easily done. Usually, standards bodies, e.g. MPEG, defines the standard block sizes for motion estimation. This can be 16 by 16, 8 by 8, etc, depending on the target application of the video codec. In the latest codec standards such as MPEG-4 or H.264/AVC, variable block sizes are supported which can be 4 by 4, 8 by 8 and 16 by 16. The goal of motion estimation is to predict the next frame from the current frame by associating the motion vector to picture macro-blocks as accurately as possible. The block size determines the quality of prediction and thus the accuracy. The prediction block in the reference frame is not necessarily in the same coordinates than the block in the current frame. Because of motion in the image sequence, the most suitable predictor for the current block may exist anywhere in the reference frame. The *motion estimation* specifies where the best prediction (best match) is found, whereas *motion compensation* merely consists of calculating the difference between the reference and the current block.

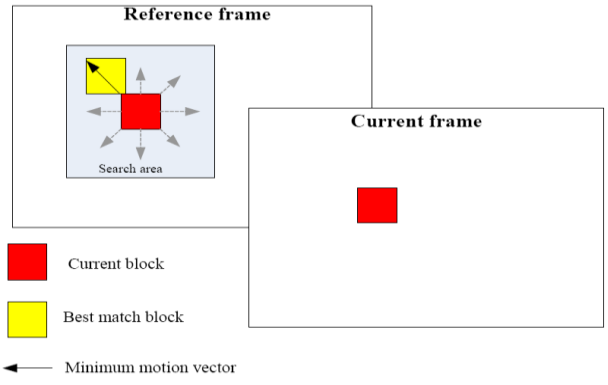


Fig1 .Motion estimation and motion vector

2.1 Steps involved in motion estimation,

• Frame Segmentation

The Actual frame is divided into non-overlapping blocks (macro blocks) usually 8x8 or 16x16 pixels. The smaller the block sizes are chosen, the more vectors need to be calculated; the block size therefore is a critical factor in terms of time performance, but also in terms of quality: if the blocks are too large, the motion matching is most likely less correlated. If the blocks are too small, it is probably, that the algorithm will try to match noise. MPEG uses usually block sizes of 16x16 pixels.

• Search Threshold

In order to minimize the number of expensive motion estimation calculations, they are only calculated if the difference between two blocks at the same position is higher than a threshold, otherwise the whole block is transmitted.

• Block Matching

In general block matching tries, to stitch together an actual predicted frame by using snippets (blocks) from previous frames. The process of block matching is the most time consuming one during encoding. In order to find a matching block, each block of the current frame is compared with a past frame within a search area. Only the luminance information is used to compare the blocks, but obviously the color information will be included in the encoding. The search area is a critical factor for the quality of the matching. It is more likely that the algorithm finds a matching block, if it searches a larger area. Obviously the number of search operations increases quadratically, when extending the search area. Therefore too large search areas slow down the encoding process dramatically. To reduce these problems often rectangular search areas are used, which take into account, that horizontal movements are more likely than vertical ones.

• Prediction Error Coding

Video motions are often more complex, and a simple shifting in 2D is not a perfectly suitable description of the motion in the actual scene, causing so called prediction errors. The MPEG stream contains a matrix for compensating this error. After prediction, the predicted and the original frame are compared, and their differences are coded. Obviously less data is needed to store only the differences.

• Vector Coding

After determining the motion vectors and evaluating the correction, these can be compressed. Large parts of MPEG videos consist of B-frames and P-frames as seen before, and most of them have mainly stored motion vectors. Therefore an efficient compression of motion vector data, which has usually high correlation, is desired.

To represent the motion of each block, a motion vector is defined as the relative displacement between the current candidate block and the best matching block within the search window in the reference frame. It is a directional pair representing the displacement in horizontal (x-axis) direction and vertical (y-axis) direction. The maximum value of motion vector is determined by the search range. The larger the search range, the more bits needed to code the motion vector.

Inter-frame predictive coding is used to eliminate the large amount of temporal and spatial redundancy that exists in video sequences and helps in compressing them. In conventional predictive coding the difference between the current frame and the predicted frame (based on the previous frame) is coded and transmitted. The better the prediction, the smaller the error and hence the transmission bit rate. If a scene is still, then a good prediction for a particular pel in the current frame is the same pel in the previous frame and the error is zero. However, when there is motion in a sequence, then a pel on the same part of the moving object is a better prediction for the current pel. The use of the knowledge of the displacement of an object in successive frames is called Motion Compensation. There are a large number of motion compensation algorithms for inter-frame predictive coding. In this study, however, we have focused only on one class of such algorithms, called the Block Matching Algorithms. These
algorithms estimate the amount of motion on a block by block basis, i.e. for each block in the current frame, a block from the previous frame is found, that is said to match this block based on a certain criterion.

2. Full Search Algorithm

After motion estimation, a picture residue and a set of motion vectors are produced. The following procedure is executed for each block (16x16, 8x8 or 4x4) in the current frame.

a. For the reference frame, a search area is defined for each block in the current frame. The search area is typically sized at 2 to 3 times the macro-block size (16x16). Using the fact that the motion between consecutive frames is statistically small, the search range is confined to this area. After the search process, a best match will be found within the area. The best matching usually means having lowest energy in the sum of residual formed by subtracting the candidate block in search region from the current block located in current frame. The process of finding best match block by block is called block-based motion estimation.

b. When the best match is found, the motion vectors and residues between the current block and reference block are computed. The process of getting the residues and motion vectors is known as motion compensation.

c. The residues and motion vectors of best match are encoded by the transform unit and entropy unit and transmitted to the decoder side.

d. At decoder side, the process is reversed to reconstruct the original picture.

3. Algorithm

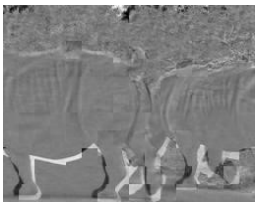
1. Divide the video into frames
2. Get the two consecutive frames.
3. Calculate the mean square error.
4. Calculate minimum of MSE.
5. Store all the values to a matrix i.e. the residue image.
6. Collect all the residue images or frames and convert it back to video clip.

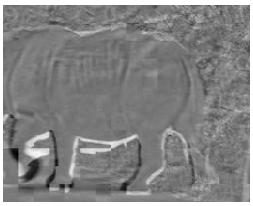
In selecting a suitably matched block, the FS algorithm searches the entire search region for a block such that the BDM is a global minimum. If more than one block generates a minimum BDM, the FS algorithm selects the block whose motion vector has the smallest magnitude, in order to exploit the centre-biased motion-vector distribution characteristics of a real-world video sequence. To achieve this, checking points are used in a spiral trajectory starting at the centre of the search region. If the maximum displacement of a motion vector in both the horizontal and vertical directions is + d or -d pixels, the total number of search points used to locate the motion vector for each block can be as high as (2d+1)2. The spiral trajectory of the checking points used by the FS algorithm with the maximum displacement, d = 7.

4. Results & Discussion

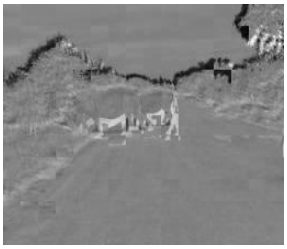
Frame No.1 **Residue of Frame No.1**

Frame No. 25 **Frame No. 37**





ResidueFrameNo.25 **ResidueFrameNo.37**



Frame No.98 **Residue of Frame No.98**

The algorithm takes 16 pixels window left ,right ,upper and lower. Find out the best possible matching block of current frame to reference frame , ten it find out the motion vectors. Since algorithm works pixel by pixel basis ,for each window it performs 33X33X8X8 multiplications and 33X33X8X8 x2 subtractions speed is low.

In modern video coding standards, the reference frame can be a previous frame, a future frame or a combination of two or more previously coded frames. The number of reference frames needed depends on the required accuracy. The more reference frames referenced by current block, the more accurate the prediction is.

5. Conclusion

Even though more commonly linked to lossy video compression, motion estimation is in fact a technique that goes beyond and allows for video processing and computational vision algorithms and applications. It allows a computer to detect movement as well as to perform comprehensive video sequence analysis, identifying scenes, and camera and object movements. Motion estimation is one technique that allows for a simple, yet effective, object identification scheme.

6. Future Scope

In future this algorithm can be further improved by developing new search technique so that it can take less time to search macro-blocks and by changing some size of macro-blocks, quality of motion can also be further improved..

7. References

[1]. Gary J. Sullivan,Thomas Wiegand "Video Compression—From Concepts to the H.264/AVC Standard" Proceeding of the IEEE, VOL. 93, NO. 1, JAN 2005.

[2]. Milind Phadtare, "Motion estimation techniques in video processing" Electronic Engineering Times India August 2007 eetindia.com.

[3] Xiaomin Wu, Weizhang Xu, Nanhao Zhu, Zhanxin Yang, "A Fast Motion Estimation Algorithm for H.264"IEEE 2010 International Conference on Signal Acquisition and Processing.

[4] Wei-feng Liu, Ji-li Lu, Zeng-fu Wang, Hua-jun Song , Xue-zheng ,"High Compression Algorithm for Video Stream" IEEE 2008 Conference on Image and Signal Processing.

[5] Hui Gu, Yan-Shanli "The Research Of Block Motion Estimation Algorithm In Video Compression" Proceedings of the Third International Conference on Machine Leg and Cybernetics, Shanghai, 26-29 August 2004.

[6] Marc Alzina, Wojciech Szpankowski,and Ananth Grama, "2D-Pattern Matching Image and Video Compression: Theory, Algorithms, and Experiments", IEEE Transaction On Image Processing, Vol. 11, No. 3, Mar 2002