An Effective Implementation of Configurable Motion Estimation Architecture for Block Matching Algorithm

C. Mohan\textsuperscript{1} V. Viswanadha\textsuperscript{2}
\textsuperscript{1}M. Tech, \textsuperscript{2}Associate Professor
\textsuperscript{1,2}SIETK, Puttur, A.P.

Abstract— This project introduces configurable motion estimation architecture for a wide range of fast block-matching algorithms (BMAs). Contemporary motion estimation architectures are either too rigid for multiple BMAs or the flexibility in them is implemented at the cost of reduced performance. In block-based motion estimation, a block-matching algorithm (BMA) searches the best matching block for the current macro block from the reference frame. During the searching procedure, the checking point yielding the minimum block distortion (MBD) determines the displacement of the best matching block.

Keywords: Block-matching algorithms (BMA's), BMA framework, motion estimation.

I. INTRODUCTION

BLOCK-BASED motion estimation has been widely adopted by the current video compression standards such as MPEG-1/2/4 and H.261/263/264. In block-based motion estimation, a block-matching algorithm (BMA) searches for the best matching block for the current macro block from the reference frame. During the searching procedure, the checking point yielding the minimum block distortion (MBD) determines the displacement of the best matching block.

For the block distortion computation, the sum of absolute differences (SAD) is one of the most frequently employed criteria. After finding the MBD point, motion estimation delivers a motion vector (MV) of the current block and prediction residues. The MV of the current block equals the displacement of the best matching block.

II. LITERATURE SURVEY

The pixel based motion estimation approach seeks to determine motion vectors for every pixel in the image. This is also referred to as the optical flow method, which works on the fundamental assumption of brightness constancy that is the intensity of a pixel remains constant, when it is displaced. However, no unique match for a pixel in the reference frame is found in the direction normal to the intensity gradient. It is for this reason that an additional constraint is also introduced in terms of the smoothness of velocity (or displacement) vectors in the neighbourhood. The smoothness constraint makes the algorithm interactive and requires excessively large computation time, making it unsuitable for practical and real time implementation for this reason I go for BMA. In BMA a single motion vector is computed for the entire block, whereby we make an inherent assumption that the entire block undergoes translational motion. This assumption is reasonably valid, except for the object boundaries and smaller block size leads to better motion estimation and Compression. Block based motion estimation is accepted in all the video coding standards proposed till date.

By observing all these below algorithm like TSS, BS, FSS, and TDL, introducing fast full search algorithm method.

III. ABOUT BMA

In a typical Block Matching Algorithm, each frame is divided into blocks, each of which consists of luminance and chrominance blocks. Usually, for coding efficiency, motion estimation is performed only on the luminance block. Each luminance block in the present frame is matched against candidate blocks in a search area on the reference frame. These candidate blocks are just the displaced versions of original block. The best candidate block is found and its displacement (motion vector) is recorded. In a typical inter frame coder the input frame is subtracted from the prediction of the reference frame. Consequently the motion vector and the resulting error can be transmitted instead of the original luminance block thus inter frame redundancy is removed and data compression is achieved.

A Block Matching Algorithm (BMA) is a way of locating matching blocks in a sequence of digital video frames for the purposes of motion estimation. The purpose of a block matching algorithm is to find a matching block from a frame \(i\) in some other frame \(j\), which may appear before or after \(i\). This can be used to discover temporal redundancy in the video sequence, increasing the effectiveness of interface video compression. Block matching algorithms make use of criteria to determine whether a given block in frame \(i\) matches the search block in frame \(j\). Motion estimation is the process of determining motion vectors.
IV. PROPOSED ALGORITHM

Fast full search is an algorithm that significantly reduces the number of computations required to carry out template matching and yields exactly the same result as the full search algorithm. The algorithm relies on the concept of bounding the matching function. Finding an efficiently computable upper bound of candidates, that can provide a better score with respect to the current best match. In this framework, we apply a succession of increasingly tighter upper bounding functions. Moreover, by including a parameter prediction step, we obtain a parameter free algorithm that, in most cases, affords computational advantages very similar to those attainable by optimal parameter tuning. Experimental results show that the proposed algorithm can significantly accelerate a full-search equivalent template matching process and performs state-of-the-art methods.

V. SYNTHESIS RESULTS

According to the above analysis, FS is well suited. Hence, the proposed BMA framework is configured to support TDL, BS, and TSS operating modes. The area and timing results based on logic synthesis as well as other characteristics. The proposed architecture outperforms the reference architectures in terms of performance because of its efficient memory system MAD and SAD unit.

VI. CONCLUSION

During the searching procedure, the checking point yielding the minimum block distortion (MBD) determines the displacement of the best matching block. This project introduces configurable motion estimation architecture for a wide range of fast block-matching algorithms (BMAs). The simulation results show that the time permitting for performing of searching within a block is less when compare to the full search. So we are achieved both speed and performance factor for searching the block without any trade off factor.

REFERENCES


