Volume : 2 | Issue : 2 | Feb 2013 • ISSN No 2277 - 8179

Research Paper

VLSI Motion Estimation Architecture for Full Search Block Matching Algorithm

Engineering

KEYWORDS : Block-matching algorithms (BMAs), BMA framework, motion estimation

Bhagavanreddy Thokala VLSI System design, Anurag group of institutions

K.Haripriya Electronics and Comm. Engg

ABSTRACT

This paper introduces a 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. The proposed architecture overcomes both of these limitations. The configurability of the proposed architecture is based on a new BMA framework that can be adjusted to support the desired set of BMAs. The chosen framework configuration is implemented by an intelligent control logic which is integrated to an efficient parallel memory system and distortion computation unit. The flexibility of the framework is demonstrated by mapping five different BMAs (BBGDS, DS, CDS, HEXBS, and TSS) to the architecture. The total execution time of the mapped BMAs is shown to be almost directly proportional to the number of tested checking points in the search area, so the architecture is very tolerant of different BMA-specific search strategies and search patterns. In addition, a run-time switching between supported BMAs can be done without performance compromises. With a 0.25-µm CMOS technology, the proposed architecture configured for FULLSEARCH, BBGDS, and TSS requires only 14.2 kgates and 2.5 KB of memory at 200 MHz operating frequency. A performance comparison to the reference programmable architecture sreveals that only the proposed implementation is able to process real-time (30 fps) fixed block-size motion estimation (1 reference frame) at full HDTV resolution (1920× 1080).

1.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.

The proposal one is full-search (FS) is the simplest, but the most computation-intensive BMA, which exhaustively tests all the checking points in the search area. Numerous fast BMAs have been developed to reduce huge computation of FS . They can be roughly classified into lossy and lossless BMAs. Lossless BMAs such as successive elimination-based algorithms have the same search quality as FS, but they speed up FS by eliminating unnecessary checking points as early as possible. Techniques utilized by lossy BMAs include simplification of matching criterion and decimation of checking points. The contemporary architectures supporting multiple BMAs are either fixed to a limited set of BMAs [1], [24]–[29] or the programmability [30]–[32] is implemented at a cost of performance and area. Particularly, their performance is a limiting factor when processing high-resolution video sequences. This is the motivation for designing a motion estimation architecture that provides real-time (30 fps) processing speed for a specific BMA up to HDTV resolution, but is still flexible enough to be used with a wide set of BMAs Our proposal presents a new BMA framework that is compatible with a predefined set of BMAs at run time. In addition, it is parametrizable to new BMAs at design time. The proposed framework is dedicated to fast BMAs that utilize decimation of checking points, so it is not suited for BMAs that rely on successive elimination and sub sampling. Although this paper considers only single reference frame motion estimation with fixed block size (16 × 16 pixels), the basic principles presented within the framework are also applicable in variable block-size motion estimation with multiple reference frames. flexibility of the proposed framework is based on an introduced separable search path generation and parametrizable search strategy control. Together they enable that an execution time of each BMA is almost directly proportional to the number of tested checking points in the search area.

2.RELATED WORK

Numerous VLSI architectures have been introduced for motion estimation during the last decades. The presented architectures can be classified as BMA-specific, flexible, and programmable

A. BMA-Specific Architectures:BMA-specific architectures support only a single BMA. The majority of contemporary BMA-specific architectures implement FS because of its regular data flow and low control overhead. FS is typically implemented with a systolic mesh-connected array which offers high throughput through parallel processing, pipelining, and data reuse. Due to the huge inherent complexity of FS, a large array of processing elements (PEs) is needed to achieve high performance.

B. Flexible Architectures: Flexible architectures implement a specific set of BMAs.Present Introduced architectures that support both FS and TSS.The architecture is designed to receive a control sequence for an executed BMA in advance through simulations. The flexibility of the architecture is demonstrated by mapping, Although flexible architectures are more generic than BMAspecific architectures, they only support a limited set of BMAs. Their control and computational structures tend to be incompatible with new BMAs or additional BMA features.

C. Programmable Architectures: Programmable architectures are designed to support a practically unlimited set of BMAs. However, all the reference architectures consider only BMAs that utilize pixel decimation techniquesa programmable motion estimation architecture that applies macro-commands to execute BMAs. The architecture processes macro-commands interactively instead of executing fixed search patterns in batches. As in part of the BMA control is managed by a host processor.

3. PROPOSED BMA FRAMEWORK.

When a BMA is executed, the parameters of the proposed.BMA framework control the search path generation in the search area. In addition, the parameters adjust search strategy control which is responsible for BMA-specific search directions and search patterns.

A. Proposed Search Path Generation

Typically, the search of the best matching block is restricted to a search area around the original location of the block .Let a size of a two-dimensional search area be w × h pixels, where $w \in \{48, 80, 112, 144\}$ and $h \in \{48, 80, 112, 144\}$ are the practical dimensions of the area in i and j directions, respectively. For video sequences with small motion content, w and h are usually 48, whereas an enlarged search area is beneficial for larger frame sizes and for sequences having rapid motion content. In the search area, the accessed 4×4 base block is indicated by a scanning point (rRR(riRR, rjRR)) , where riRR ∈ [0,w − 1] and rjRR ∈ [0, h − 1]. The first partial offset of rRR is an initial offset (_rα(_riα,_rjα)) that points to the center of the search area. _riα is incremented so that the successive searching procedures can partially reuse the same search area data. _riα ∈ [0,w − 1] and _rjα∈ [0, h − 1] are multiples of 16.

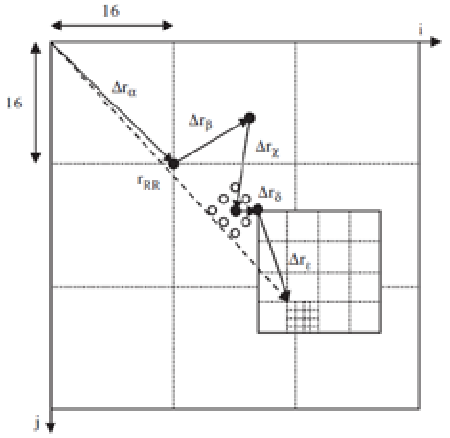

Fig. 1. Proposed separable composition of r_{RR} in the 48 × 48 pixel search area.

B. Proposed Search Strategy Control

A flexible architecture having a hardwired control for each of the supported BMAs achieves high performance, but even a diminutive BMA modification requires redesigning the control. Since the search strategies and patterns tend to vary between BMAs, a parametrizable search strategy control is a necessity for the proposed configurable architecture. The general BMA execution flow is summarized for a specific BMA as:

Step 1: The first search pattern is positioned in the search center and only the search center is tested and selected as the initial MBD point. The searching procedure proceeds to Step 2.

Step 2:All the valid checking points surrounding the pattern center are tested. If no new MBD point is found, i.e., the MBD point is still found in the pattern center or if a single-pass pattern is used, the searching procedure proceeds to Step 3. Otherwise, the recursive search pattern is repositioned so that the new MBD point is at the center of the pattern and Step 2 is recursively repeated.

Step 3: If all the available search patterns have already been executed, the MBD point found in Step 2 is selected and the searching procedure is stopped. Otherwise, the search pattern is switched to a next search pattern which is positioned so that the MBD point found in Step 2 is in the center of it and the searching procedure proceeds to Step 2.

4. PERFORMANCE ANALYSIS

The performance of the proposed motion estimation architecture is evaluated with TSS , FS ,DS , and CDS . All the five BMAs follow the proposed BMA execution flow, but the search strategies and patterns vary between them. TSS uses single pass search patterns and a coarse-to-fine search strategy in which search step size is hierarchically converged. The amount of repeated steps is determined by the size of the search area.full search utilizes a square pattern of 3 × 3 checking points which recursively advances towards the MBD point

Synthesis Results and Comparison

According to the above analysis, full search is well suited for sequences with small or moderate motion content, TSS for lo bit rate purposes due to its moderately good search quality, and FS for fast execution. Hence, the proposed BMA framework is configured to support FS, BBGDS, and TSS operating modes. The area and timing results based on logic synthesis as well as other characteristics are listed for the proposed configuration A 0.25-μm LVCMOS.The proposed architecture clocked at 200 MHz is implementable with 10%IOBS and 5%RAM memory when the 4%Gclks .

CONCLUSION

The architecture supports a predefined set of BMAs at run time and provides low latency as well as high throughput for all the mapped BMAs independent of the search strategy, search path, or search patterns. The total execution time of each BMA is almost directly proportional to the number of checking points tested. Hardware utilization is also moderately high since all the processing elements always participate in distortion computation. In addition, the silicon area cost is close to BMA-specific architectures. Mapping FULL SEARCH, BBGDS, and TSS to the architecture clocked at 10%, configuration A 0.25-μm LVCMOS. The proposed architecture implementable with 10%IOBS and 5%RAM memory when the 4%Gclks,and the cost 30% reduced.

Acknowledgment

The satisfaction that accompanies the successful completion of my project. I Express my sincere gratitude to prof.j.v. Sharma (Ph.D), K.HARIPRIYA (Ph.D), department of electronics and communication engineering.

REFERENCE

[1] P. Kuhn, Algorithms, Complexity Analysis and VLSI Architectures forMPEG-4 Motion Estimation. Boston, MA: Kluwer, 1999, p. 239. | [2] Y. W. Huang, C. Y. Chen, C. H. Tsai, C. F. Shen, and L. G. Chen, "Surveyon block matching motion estimation algorithms and architectureswith new results," J. VLSI Signal Process., vol. 42, no. 3, pp. 297–320,Mar. 2006. | [3] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," IEEE Trans. Image Process., vol. 4, no. 1, pp. 105–107,Jan. 1995. | [4] X. Q. Gao, C. J. Duanmu, and C. R. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," IEEETrans. Image Process., vol. 9, no. 3, pp. 501–504, Mar. 2000. | [5] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," IEEE Trans. Circuits Syst. Video Technol., vol. 3, no. 2,pp. 148–157, Apr.1993. | [6] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proc. Nat.Telecommunication Conf., New Orleans, LA, 1981, pp. G5.3.1–5.3.5.