

« Motion Estimation Accelerator with User Search Strategy in an RVC Context »

Julien Dubois¹, Richard Thavot², Romuald Mosqueron²,
Johel Miteran¹ and Christophe Lucarz²

¹Laboratoire LE2I, Université de Bourgogne

jdubois@u-bourgogne.fr

²LSM Laboratory, EPFL

ABSTRACT

Motion estimation represents a key module in video compression. The RVC context requires proposing a flexible solution for motion estimation. According to the nature of the application, a full search is sometimes not suitable, hence, alternative fast/reduced solutions should be considered. This paper proposes a model and implementation of a flexible motion estimation engine, which can be configured to support any user-defined search strategy. Typically, the computational requirements of the search strategy can be traded with the RD-performance of the obtained video encoder. A CAL dataflow description of the accelerator is proposed so that it can be easily handled in the RVC context. An automatic translation of the proposed CAL module to HDL is performed, and a comparison between the generated HDL with handwritten HDL code of the same model is performed. This helps to evaluate the influences of the CAL model refinements.

Index Terms— Reduced search strategy, motion estimation, RVC, CAL.

1. INTRODUCTION

The Motion Picture Expert Group (MPEG) has initiated the definition of the “Reconfigurable Video Coding” (RVC) framework which proposes a new way to describe and therefore to implement video coder/decoder [1] [2]. The idea is to specify a reconfigurable multi-format video codec, which supports the desired coding or decoding standard with small computational and control overhead. The new video decoders must be described in a flexible way at the higher level of the RVC video coding tool library. This flexible way requires adapting the decoder performances to the video data flow. Therefore the coder should be modeled with a flexible description. A CAL description [3] has been proposed by MPEG group for the following reasons. Moreover, a prototyping tool provides efficient HW performances [4].

The motion estimation is well known to be the most computation-intensive stage of video coding process and

has been the subject of many research works (on both algorithmic and architecture sides) which aim to reduce the implementation complexity. For brevity, we cannot here provide an accurate review of the wide literature on the subject, we suggest referring for instance to [5] [6] and their references, for an updated review of the state of the art. This paper focuses one family called “Reduced Search Algorithms” which aim to reduce the complexity by limiting the measure of the matching criterion to only a (small) subset of candidate vectors in the search window. The key element here is the “intelligence” of the search algorithm that may completely change depending on the requirements of the video coding application (portable video telephony, HDTV for sport events, etc...). The RVC context requires proposing a flexible solution for motion estimation. Therefore a motion estimation based on reduced search strategy represents an innovative solution. This paper is organized as follows. The reduced search strategy and its interests in the RVC context are presented in section 2. The two descriptions (CAL and hand written VHDL versions) of the motion estimator and their advantages, are presented in section 3. The two resulting hardware implementations are presented and discussed in section 4.

2. A REDUCED SEARCH STRATEGY IN A RVC CONTEXT

The most probable matching position is obtained with a classical Maximum Absolute Difference (MAD) criterion. With a full search strategy, the motion detection process is regular, as described in Figure 1 all possible positions of the pattern in the search window are scanned. Therefore the number of matching depends on the sizes of the pattern and of the search window (i.e for 16x16 pattern and a 56x40 search window, 1025 matching are requested). Obviously, with the high number of matching (i.e >1000 matching) the motion estimation represents then the most computation intensive stage of the video encoding. To obtain high-speed performances, the number of MAD blocks can be multiplied and therefore a specific architecture should be designed to provide the required input-data bandwidth. In any case, with reference sizes for pattern and search window (receptively

16x16 and 56x40) only few matching can be processed simultaneously and the input-data bandwidth becomes quickly a bottleneck, the high number of MAD becomes useless. Therefore the motion estimation processing time remains high.

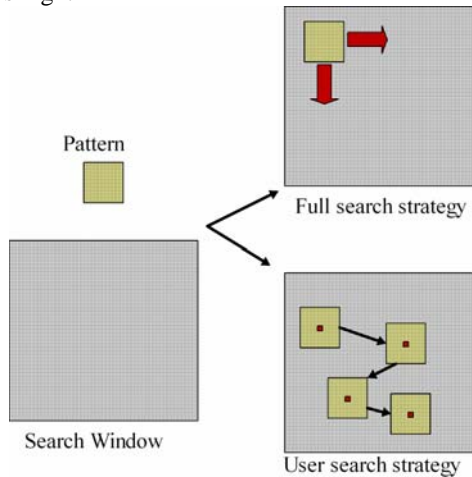


Figure 1: Full Search and Reduced Search strategy

Contrary to the full-search, the reduced search aims to minimize the number of matching by using previous information (such as the vectors previously computed). Therefore using a reduced search strategy enables the number of matching and then the computation time to be reduced. As mentioned previously, using appropriate (for the video application) reduced search algorithms, it is possible to achieve optimal coding results [5]. The Figure 1 represents a reduced search (4 matching represented) versus a full-search. The list of matching to process is reduced according to the user's search algorithm. The motion estimation accelerator receives this list and can process only the concerned positions, due to its specificity of randomly matching any area in a search window without any other limitation on the access sequence. In other terms, even with a random access, the performance per matching is not changed. The address-generation is processed by a flexible unit (FPGA-embedded DMA or processor) to enable the translation of the user search strategy into the hardware set-up.

The user strategy is based on the fact that the motion vector is normally highly-correlated with the previous vectors. The algorithm can be split in several phases. An implementation in two phases is presented. We presented here a simplified version of the technique described in [5]. The first set of "candidate" vectors is mainly obtained with neighbour vectors and random vectors. In a second phase, the resulting score of each candidate vector enables them to be ranked (Figure 2). Scalar combinations of these vectors are done to propose a second set of "candidate vectors". Once again the number of combinations depends of the user "budget".

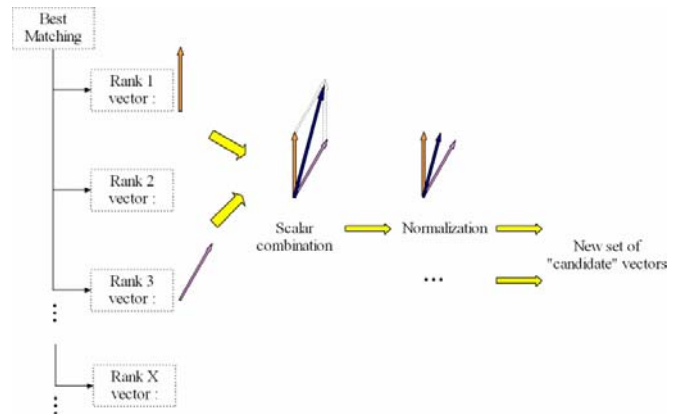


Figure 2: Determination of second set of candidate vectors

Moreover, the system does not only provide flexibility, the performances can also be increased. With this system, the user can scale the resource budget in terms of available "candidate" vectors for each search in view of the required performances and the best configuration for the video encoding. For instance, the user can:

- Adjust the number of vectors to obtain optimal coding results or in general the trade-off between the coding performances and the processing time,
- Adjust/modify the search algorithm complexity,
- Adjust/increase the size of the search window.

The user can define with precision the different parameters required for a reduced search strategy in a video context (coding performances, search algorithm, window size). All these possible trade-off are possible. This type of approach can match RVC constraints. In this context, we propose a CAL description of a motion estimator component which supports user search strategy.

3. MODELISATION AND ARCHITECTURE DESCRIPTION

This data flow modelisation describes this system at high level of abstraction. CAL Actor Language is a language based on the Actor model of computation for dataflow systems. It provides many natural concepts to facilitate modeling of those systems [3]. A dataflow model expressed in CAL is composed by a set of independent "actors" and their connection structure. It makes a network of actors. An actor is a stand alone entity which has its own internal state represented by a set of state variables and it performs computations by firing actions.

The major challenges of the architecture design of the co-processor are to guarantee the random access of any search strategy to be user defined and to allow the setting of the size of the search-window hence providing sufficient processing resources for the different standard versions and profiles. So to guarantee the random access in any position

of a search window, the search window pixels have to be accessible to the matching engine and need to be stored in the FPGA to reduce the number of access to the external memory, so as not to exceed the available bandwidth..

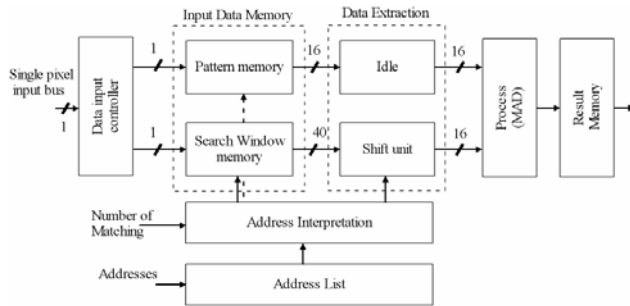


Figure 3. Data flow of the proposed architecture

The Figure 3 shows the motion estimation accelerator which is constituted of:

- the block that includes the internal data memory and the MAD estimation core,
- Two data internal cache memory to store the pattern (16x16 pixels) and of the search window (56x40 pixels),
- Specific units to handle with the matching addresses generation and the system control.

The MAD matching evaluation criterion has been used in the current version of the implementation. The implemented pipelined architecture enables to match a block and a portion of search window, column by column. At each cycle, the two internal memories permit access of two columns, one extract from the block and the corresponding one in the search window. The data-flow bandwidth is relative to a 16x16 pattern (one macroblock vector estimation). An input data memory and the associated control have been designed in order to provide the column of 16 pixels at each cycle. An address allows the extraction of one column from the search window and one from the block, and the programmable shift to select 16 pixels from the search window column according to the selected matching. The address and shift control are both generated by the control blocks. The resulting model allows to access, in one clock cycle, any matching evaluations do not need to be adjacent in terms of memory locations.

The VHDL handwritten model has slightly adapted. The pixels are coded on a byte on most of the video profiles, so to improve the system bandwidth efficiency, the input data has been mapped with 32-bits bus width. The internal search window memory has been directly implemented on the FPGA internal memory (BRAMs). At the initialisation it is possible to reconfigure the search window size. Any search-window width between 20 and 256, can be set according to the available processing. The size of search-window can only be configured at the system initialization.

Rectangular search windows, wider in the horizontal dimension have been shown to be very efficient for most of the video applications. Nevertheless, the search-window high can only be modified without important time-consuming design changes on the proposed architecture. Considering the internal architecture of memory blocks (BRAMs) depth, an optional element in charge of the image transposition can be inserted into the data flow whenever the search window height is lower than its width. Such component yields a reduction of the number of memory blocks (BRAMs) usage.

The CAL model is the most compact description. The **code size is around three times smaller** than for the VHDL model. The resulting CAL description is more adaptable than the VHDL model. For instance, the memory size can be defined as a parameter and the connection can be easily modified. In the RVC context, the CAL description represents therefore a real advantage as it can be easily modified in function of the video constraints. Another main advantage of the CAL model is the low conception time. **This time has been estimated at four times lower** than that with the hand written VHDL description.

The CAL code has been processed to produce automatically a VHDL code using a CAL2HDL translator. The performances of the hand written VHDL represent our coding reference in terms of coding performance. Moreover with this reference description, the hardware resources required have been minimized by a previous study [7]. The performances of the two resulting implementations are compared in the following section.

4. HARDWARE IMPLEMENTATION

The two implementations reach high throughput performances. The hand written VHDL version provides the highest performances. For instance, with a 16x16 pattern and 56x40 search window and without considerations of the data transfer time, a matching can be achieved in 92 ns. The performance of the translated CAL version enables a matching to be processed in 144 ns. With the translated CAL version, a 4CIF image can be therefore processed a full search in less than 229 μ s. The reduced search strategy can obviously decreased the processing time. In the full search configuration and with these block image sizes, 1025 matching are required. The reduced search algorithm, defined in [5], shows optimal coding performances can be achieved with that less than 200 matching (two set of 100 matching). The global processing time depends on time required for the determination of the second set of vectors. Nevertheless, in the reduced search mode, this design has a high potential of coding performances.

The input data storage can be done simultaneously with the processing. The input data bandwidth is currently limited at 110 Mpixels per second, can be improved up to 300.

	Number of occupied slices	Number of Flip-flops	Number of BRAMs (18x2s)	Number of 4 inputs LUT	Max Frequency (MHz)
Data extraction unit	318	657	0	673	175
MAD unit	365	354	0	576	350
Motion estimation unit	1 940	1 077	7	1 412	175

Table I: Hand written VHDL implementation (based on xc5vlx110 speed 3 FPGA)

	Number of occupied slices	Number of Flip-flops	Number of BRAMs (18x2s)	Number of 4 inputs LUT	Max Frequency (MHz)
Data extraction unit	1 648	152	0	2 036	110
MAD unit	378	452	0	1 075	320
Motion estimation unit	2 961	2 961	2	5 407	110

Table II: Hardware implementation obtained with the translated CAL description (based on xc5vlx110 speed 3 FPGA)

Table I and Table II report the results of the two hardware implementations. The hardware resources required are higher with the second implementation. The overhead is around 50 % in terms of slices used. A higher overhead is observed for flips-flops and specially LUT tables. Nevertheless the memory blocks are reduced from 7 to 2. The resource overhead is quite different depending on the tasks to process and the data format. For instance, a main part of the resource overhead, especially noticeable in the MAD and the data extraction units, is due to large data bus handling. The maximum bus format is 32 bits with the current CAL descriptors. The data should be slip in several tokens that increased the require resources. Moreover, it can be noticed that supplying CAL with a library of basic functions similar to what in VHDL are Concatenation(), Bitselect() and similar low level library functions, such compactness factor can largely further improve. A reduction of the resources is currently investigated by writing the CAL description and the potential fusion of actors (especially the input data memory and the data extraction actors).

5. CONCLUSION

A CAL description has been obtained of a motion estimator with user/reduced search strategy has been implemented which is fully adapted to RVC context. The automatic VHDL translation has permit to produce a hardware implementation with high matching performances (**matching to be processed in less than 144 ns**). The hardware implementation requires a resource overhead which is around 50 per cent higher than with an optimized hand-written VHDL description. Optimizations are currently investigated to reduce this overhead.

6. REFERENCES

- [1] Gwo Giun Lee, Euee S. Jang, Marco Mattavelli, Chun-Jen Tsai, Christophe Lucarz, Mickaël Raulet, Dandan Ding, "Text of ISO/IEC FCD 23001-4: Codec Configuration Representation," ISO/IEC JCT1/SC29/WG11 MPEG w9772, Archamps, France, Apr. 2008.
- [2] Yi-Shin Tung, Gwo-Giun Lee, Euee S. Jang, Sunyoung Lee, Kohtaro Asai, Yoshihisa Yamada, Marco Mattavelli, "FCD ISO/IEC 23002-4: Video Tool Library," ISO/IEC JCT1/SC29/WG11 MPEG w9774, Archamps, France, Apr. 2008
- [3] Johan Eker and Jorn Janneck, "CAL Language Report", Tech.Rep.ERL Technical Memo UCB/ERL M03/48, University of California at Berkeley, Dec. 2003.
- [4] Jörn W. Janneck, Ian D. Miller, and all, "Translating dataflow programs to efficient hardware: an MPEG-4 simple profile decoder case study", in *Design Automation and Test in Europe (DATE)*, Munich, Germany, 2008.
- [5] M. Mattavelli, G. Zoia: "Vector tracing algorithms for motion estimation in large search windows", *IEEE Trans. on Circuits and Systems for Video Technology*, December 2000, Vol 10, Nr. 8, pp 1426-1437.
- [6] J-H Luo, C-N Wang, T. Chiang: "A novel all binary motion estimation (ABME) with optimized hardware architecture", *IEEE Trans. on Circuits and Systems for Video Technology*, August 2002, Vol 12, Nr. 8, pp 700-712.
- [7] J. Dubois, M. Mattavelli, L. Pierrefeu, J. Miteran, "Configurable Motion-Estimation Hardware Accelerator Module for the MPEG-4 Reference Hardware Description Platform", *IEEE International Conference On Image Processing (ICIP05)*, Genova, Italy, Sept. 2005.