HARDWARE ACCELERATION ARCHITECTURES FOR MPEG-BASED MOBILE VIDEO PLATFORMS: A BRIEF OVERVIEW

N. O'CONNOR, V. MURESAN, A. KINANE, D. LARKIN, S. MARLOW, N. MURPHY

Visual Media Processing Group, Dublin City University,
Glasnevin, Dublin 9, IRELAND.E-mail: Noel.OConnor@dcu.ie

This paper presents a brief overview of past and current hardware acceleration (HwA) approaches that have been proposed for the most computationally intensive compression tools of the MPEG-4 standard. These approaches are classified based on their historical evolution and architectural approach. An analysis of both evolutionary and functional classifications is carried out in order to speculate on the possible trends of the HwA architectures to be employed in mobile video platforms.

1. Introduction

There is an ongoing global trend to shift multimedia applications from more traditional delivery platforms, such as a set-top boxes or desktop PCs, to mobile platforms (e.g. PDAs and smart-phones). MPEG-4's improved compression efficiency will drive the first wave of mobile multimedia applications, while its content-based features will underpin future applications. MPEG-4 pays for its compression efficiency and content-based advantages with computationally expensive algorithms for motion estimation (ME), shape adaptive DCT/IDCT and context-based arithmetic encoding (CAE). Generic HW limitations of mobile devices include: low computational power, low memory capacity, short battery life and miniaturization requirements. These limitations are exacerbated by real-time multimedia applications that require complex but low power HwAs.

Several HW solutions have been proposed for the video compression domain [1]: Application Specific Integrated Circuit (ASIC) solutions, Digital Signal Processing (DSP) architectures, systolic arrays, with Single Instruction Multiple Data (SIMD) or Multiple Instruction Multiple Data (MIMD) control paradigms and Field Programmable Gate Array (FPGA) implementations. Unfortunately, these are not intrinsically power efficient. The HwA solutions developed to date have been high-throughput driven. Hence, they use more power than their SW counterparts for the simple reason that power is not an issue on desktops. However, the short battery life issue of mobile devices has become the biggest HW design constraint facing truly mobile multimedia. Power efficiency can be achieved by maximally exploiting the possibility for

The support of the Research Innovation Fund of Enterprise Ireland is gratefully acknowledged.

MPEG-4 tools to be HW accelerated and re-modeled, in conjunction with other state-of-the-art techniques such as dynamic power management (DPM). Thus, a hybrid HwA design paradigm has evolved, where computationally intensive tasks are implemented on more flexible dedicated HW architectures while their adaptive control is committed to SW RISC processor-based solutions. The following sections outline the evolution of this paradigm shift.

2. Motion Estimation Hardware Acceleration

Motion estimation (ME), complicated by object representation, is by far the most computationally expensive MPEG-4 tool, requiring over 50% of the computational power [2]. A block-matching algorithm (BMA) approach consists of two tasks: a *block-matching* (BM) task carrying out distance calculations and a *search task* specifying the sequence of candidate blocks where the distance criterion is calculated. Many of the distance criteria for BMA are based on square root, multiplication and division operations, which are not efficient in HW. Those deemed to be feasible in HW from the performance/complexity ratio point of view are [2]: different pel count (DPC) and sum of absolute differences (SAD). Of these, SAD was proved to deliver the best accuracy/complexity ratio.

2.1. *Processing Datapath Approaches*

Systolic arrays (SA) were the first approaches used for BMAs. They were meant to maximally exploit BM operations' regularity in a full search (FS) strategy. Implementations can be classified as 1-D or 2-D approaches, with global or local accumulation [3]. Clock rate, picture size, search range, and block size are the parameters used to decide on the number of processing elements (PEs) in the systolic structure [4]. SA implementations have been proposed for both full search (FS) and fast heuristical (FH) search strategies. Usually a SA does not require significant control circuitry overhead [4]. However, for FH strategies, the complexity of the controller needed to generate data addresses and flow control signals increases considerably along with the power inefficiency. In order to avoid this, a *tree-architecture* BM is proposed in [5]. Although suitable for irregular search strategies, this requires unfeasible high memory bandwidth.

Recent categories of BMA are the *reduced pel information* approaches. They firstly reduce the pel information (usually by edge extraction, frame processing or pel subsampling) and then apply a search strategy on reduced-bit frame representations. Some adaptive reduced-bit BMA vary the size of the pel information so that acceptable compressed image quality is maintained [6]. Another important category is *binary search algorithms* [7] that are also employed in shape coding. The short battery life issue has steered ME research

towards the so-called *fast exhaustive* (FE) search strategies that employ conservative SAD estimations and SAD cancellation mechanisms [8]. These approaches achieve the same results as the FS ones, but reduce computation by skipping irrelevant candidate blocks.

2.2. *ME Memory Optimization Approaches*

Recently, a lot of ME optimization approaches have been proposed to tackle memory efficiency. They target memory data flow rather than traditional memory banking optimization. They re-arrange and remap the content of the on-chip memory in order to achieve the highest memory access efficiency. This is achieved by a high degree of on-chip memory content re-use, parallel pel information access, memory access interleaving [9].

3. DCT/IDCT Hardware Acceleration

This section briefly surveys the range of DCT algorithms and architectures in the literature and evaluates them in terms of viability for a low power HwA implementation.

3.1. *DCT / IDCT Algorithms and Architectures*

The algorithm proposed in [10] is a very common baseline DCT implementation, and many algorithms and architectures are based upon it. The algorithm exploits the fact that a unitary matrix can theoretically be factorised into products of relatively sparse matrices. There are faster algorithms known, but they typically translate to very complex signal-flow graphs with irregular routing, complex architectures and numerous I/O pins. The large number of butterfly stages can make a unified DCT/IDCT block difficult to implement.

More recently, research attention has shifted towards developing more regular DCT architectures with less emphasis on developing the fastest possible algorithm. There are numerous techniques used, and these include systolic arrays, recursive structures, and Distributed Arithmetic (DA). Some specify the lower level architecture such as the adders and memory architectures. There are also architectures that are targeted directly for low power applications. Some current low power techniques exploit the mathematical properties of the DCT/IDCT such as DCT pruning algorithms [11] (which only transforms a subset of the whole block), low SAD macroblock skipping, skipping all-zero IDCT input and truncated multiplication. These architectures exploit the fact that the DCT coefficients will be subsequently quantized and adaptively adjust the precision of the calculations accordingly. Other techniques exploit the fact

that the DCT/IDCT are essentially large multiply-accumulate operations and use DA [12] as an alternative to power consuming multiplications. There are also some general low-level techniques proposed [13] such as clock gating, low-transition data paths and voltage scaling.

The first generally accepted SA-DCT algorithm [14] can be easily incorporated with existing block-based techniques and is backward compatible with existing standards. A new architecture [15] has been proposed that can be configured to evaluate the DCT or the SA-DCT as appropriate. The architecture trades off scalability, modularity and regularity. It is a feed-forward architecture, which avoids numerical inaccuracy or bit-width explosion that can occur in some of the fast algorithms more suited a SW implementation.

4. Shape Encoding Hardware Acceleration

It is generally accepted that shape coding is the second most computational expensive process in a MPEG-4 encoder after motion estimation, while shape decoding is the most complex task in a MPEG-4 decoder. Context Based Arithmetic Coding (CAE), the shape coding approach used in MPEG-4, is discussed in detail in [16][17].

4.1. *HwA Evolution*

To date, there has only been one complete HW implementation of the Shape Encoder [18] and a number of proposed architectures for the decoder. This approach achieves real-time core profile level 2 processing at 23.5 Mhz. Its performance is optimized for throughput. There are no architectural or implementation attempts for efficient power consumption. Profiling has revealed that 95% of the computational load of shape coding is consumed by the binary motion estimation/compensation, CAE and size conversion sub-blocks. Optimised HW solutions are proposed in [18] to allow these functions operate in real time. The main architectural features proposed included: bit data parallelism processing, bit addressing scheme, efficient reuse of windowed pel data.

Whilst no low power shape coder/decoder exists, the literature outlines power efficient techniques and architectures for blocks common to both. There are also lower level optimizations for discrete elements, such as low power Barrel shifters, adders etc. However the potential for power saving with these is relatively small unless coupled with an efficient architectural implementation. Some of the shape coding/decoding sub blocks, such as Mode Decision Logic, Accepted Quality comparisons etc, are potentially more suited to a SW implementation in a hybrid HW/SW shape coding solution.

5. Architectural Trends

A multicore SOC architecture has recently been proposed in [19] to exploit the many different modes, options, and switches that are provided within the MPEG-4 standard. This is achievable only if early design decisions are made on MPEG-4 tools' specifications and enough flexibility is left in the dedicated HW blocks. Thus, a heterogeneous SOC concept is proposed to employ multiple cores that are adapted to different classes of algorithms.

Architectural solutions are the main topic of this overview. However, one important remark is that since power consumption is directly proportional to the computational complexity, it is important to tackle this issue at a high-level where specific behavioral aspects of the compression tools can be exploited in order to reduce computation. Architectural-level memory optimization in order to meet the best area-speed-power trade-off can be achieved by reducing the amount of memory accesses through multiple memory splitting into sub-banks, selective line activation, bit-line segmentation, on-chip memory, and application specific data-flow transformations (e.g. memory interleaving). In the context of memory bandwidth, an efficient balance between on-chip and off-chip memory has to be obtained to meet the best power/bandwidth trade off. Since video compression tools are memory intensive, local memory architectures are also used to also avoid system bus conflicts and congestion.

6. Conclusions

The technology shift from desktop platforms to mobile platforms requires a change in focus in the ME HwA design trade-off problem and a substitution of the area/performance design space for a performance/power one. Hybrid power-efficient HwA architectures are required to meet real-time mobile multimedia applications' needs. This paper gives a very brief overview on this recent design paradigm shift. The Visual Media Processing Group at DCU carries out research in the area of low-power hardware acceleration for mobile multimedia platforms. The objective of this paper is to help researchers in this field to focus on low-power enhancements of the HW solutions proposed so far for MPEG-4's computationally intensive video compression tools.

References

[1] Pirsch *et al*., VLSI Architectures for Video Compression - A Survey, Proc. IEEE, Vol. 83, No. 2, Feb 1995.

[2] P. Kuhn, Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation, Kluwer Academic Publishers, 1999.

[3] E. Chan *et al*., Motion Estimation Architecture for Video Compression, IEEE Trans. Consumer Electronics, Vol. 39, No. 3, Aug 1993.

[4] S.C. Cheng *et al*., A Comparison of Block-Matching Algorithms Mapped to Systolic-Array Implementation, IEEE Trans. Circuits and Systems for Video Technology (CSVT), Vol. 7, No. 5, Oct 1997.

[5] Y.S. Jehng, *et al*., An Efficient and Simple VLSI Tree Architecture for Motion Estimation Algorithms, IEEE Trans. Signal Processing, Vol. 41, No. 2, Feb 1993.

[6] Z.L. He *et al*., Low-Power VLSI Design for Motion Estimation Using Adaptive Pixel Truncation, IEEE Trans. CSVT, Vol. 10, No. 5, Aug 2000.

[7] J.H. Luo *et al*., A Novel All-Binary Motion Estimation (ABME) with Optimized Hardware Architectures, IEEE Trans. CSVT, Vol. 12, No. 8, Aug 2002.

[8] V.L. Do *et al*., A Low-Power VLSI Architecture for Full-Search Block-Matching Motion Estimation, IEEE Trans. CSVT, Vol. 8, No. 4, Aug 1998.

[9] Y.K. Lai *et al*., A Data-Interlacing Architecture with Two-Dimensional Data-Reuse for Full-Search Block-Matching Algorithm, IEEE Trans. CSVT, Vol. 8, No. 2, Apr 1998.

[10] W. Chen *et al*., A Fast Computational Algorithm for the Discrete Cosine Transform, IEEE Trans. Communications, Vol. 25, No. 9, Sep 1977.

[11] J. Astola *et al*., Architecture-Oriented Regular Algorithms for Discrete Sine and Cosine Transforms, IEEE Trans. Signal Processing. Vol. 47, No.4, Apr 1999.

[12] T. Xanthopoulos *et al*., A Low-Power DCT Core Using Adaptive Bitwidth and Arithmetic Activity Exploiting Signal Correlations and Quantization, IEEE Journal of Solid-State Circuits, Vol. 35, No. 5, May 2000.

[13] N. August *et al*., On the Low-Power Design of DCT and IDCT for Low Bit-Rate Video Codecs, Intn'l ASIC/SoC Conf., Sep 2001, Arlington VA, USA.

[14] T. Sikora *et al*., Shape-Adaptive DCT for Generic Coding of Video, IEEE Trans. CSVT, Vol. 5, No. 1, Feb 1995.

[15] T. Le *et al*., Flexible Architectures for DCT of Variable-Length Targeting Shape-Adaptive Transform, IEEE Trans. CSVT, Vol. 10, No. 8, Feb1999.

[16] A. Katsaggelos, *et al*., "MPEG-4 and Rate Distortion Based Shape Coding Techniques", Proc. IEEE, Vol. 86, NO. 6, Jun 1998

[17] N. Brady, "MPEG-4 Standardized Methods for the Compression of Arbitrarily Shaped Video Objects", IEEE Trans. CSVT, Vol. 9, No. 8, Dec 1999

[18] H.C. Chang *et al*., VLSI Architecture Design of MPEG-4 Shape Coding, IEEE Trans. CSVT, Vol. 12, No. 9, Sep 2002

[19] M. Berekovici *et al*., Multicore System-On-Chip Architecture for MPEG-4 Streaming Video, IEEE Trans. CSVT, Vol. 12, No. 8, Aug 2002.