7.4-5

Motion Estimation and Temporal Up-Conversion on the TM3270 Media-Processor

Jan-Willem van de Waerdt[1,2], Stamatis Vassiliadis[2], Erwin Bellers[1] and Johan G.W.M. Janssen[1]
[1]Philips Semiconductors, San Jose, CA, USA
[2]Delft University of Technology, Delft, The Netherlands

***Abstract*—We present a quantative performance evaluation of several components of a video format conversion algorithm (referred to as Natural Motion (NM)). The implementation platform is a new programmable media-processor, the TM3270, combined with dedicated hardware support. The performance of two compute-intense NM components, Motion Estimation (ME) and Temporal Up-conversion (TU), is evaluated. The impact of new TM3270 features, such as new video-processing operations and data prefetching, is quantified. We show that a real-time implementation of the ME and TU algorithms is achievable in a fraction of the available compute performance, when operating on standard definition video.**

I. Introduction

The Natural Motion (NM) video format conversion algorithm addresses the increasing need for video format conversion (Figure 1). Both spatial and temporal format conversions may be required to convert the source video stream format to that of the display. Until recently, these conversions had a dedicated hardware implementation. However, increased performance of media-processors has enabled software implementations [1][2]. Software implementations provide flexibility, which can be exploited in different ways; e.g. multiple algorithms can be mapped onto the same implementation platform, algorithmic changes without modifications to the implementation platform, a higher level of algorithmic adaptation to video content, etc. These advantages have led to a co-existence of dedicated hardware and software implementation platforms.

In this paper we evaluate the performance of two compute-intensive components of the NM algorithm on the new TM3270 media-processor.

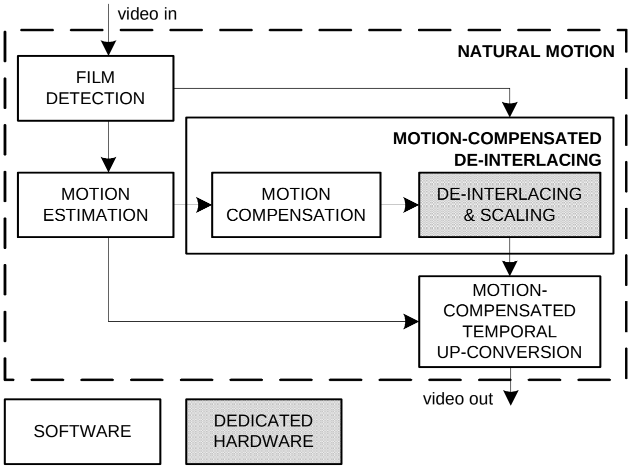

Fig. 1. Natural Motion (NM) algorithm, and its main components.

II. Algorithms

Motion Estimation (ME) and Temporal Up-conversion (TU) are two components of the NM algorithm that can benefit from a software implementation: algorithmic innovations in both areas have led to continuous quality improvements, and a software implementation allows us to apply these innovations with a short time-to-market.

It is not the intend of this paper to present best-in-class ME and TU algorithms, but rather to evaluate the performance of an algorithm that is representative for its class. For ME, we use the 3-D Recursive Search (3DRS) block-based motion estimator [3]. Our version of the 3DRS estimator performs a motion search using a set of 11 candidate motion vectors for each 8x8 block of image pixels. It provides a high quality result at a low computational complexity. The motion search range has a horizontal search range of the video image width, and a vertical range of [-40, 39 ¾] pixels. For TU, we use an enhanced version of the motion-compensated cascaded median up-converter [4]. Our version performs up-conversion on 4x4 blocks of image pixels. The 4x4 block motion vectors are derived through block erosion from the 8x8 block motion vectors as determined by the ME algorithm. Both algorithms operate on ¼ pixel accuracy; non-integer image pixels are calculated through bi-linear interpolation using the fractional pixel position offsets as weighing factors.

III. Software implementation platform

The ME and TU algorithms were implemented on the TM3270 media-processor. The TM3270 is a five issue slot VLIW processor; i.e. every cycle, up to five independent operations can be started. Using the SIMD capabilities of the 32-bit datapaths, a single operation is performed on either one 32-bit, two 16-bit, or four 8-bit sub-operands. In other words each cycle, up to 5 * 4 = 20 8-bit operations can be performed (five issue slots, 8-bit SIMD parallelism). At a 450 MHz operating frequency, this results in a maximum computational performance of 450 MHz * 20 = 9 Gops/sec (note that these operations can be relative complex, e.g. 3-taps median operations). The processor has a 64 Kbyte instruction and a 128 Kbyte data cache.

The TM3270 has several new features, when compared to other media-processors. Most notable are data prefetching and new three- and four-input video-processing operations. Data prefetching anticipates the future use of data by retrieving it from main memory into the processor's data cache, before its actual use by the processor. As a result, the data is available when needed by the processor, which eliminates processor stall cycles due to cache misses. As an example of new operations, consider the *QUADUMEDIAN* and *LD_FRAC8* operations (Table I).

A single *QUADMEDIAN* SIMD operation performs four 8-bit 3-taps median operations on three inputs, whereas the TM3260 required two *QUADUMIN* and two *QUADMAX* operations to calculate the same result. Since the 3-taps median is a basic building block of the TU algorithm, a significant performance improvement is observed when this new operation is used.

The *LD_FRAC8* operation retrieves five 8-bit data elements from memory and performs a weighted average on neighboring data elements to produce a four 8-bit result. This operation is used in both the ME and TU algorithms for the calculation of horizontal non-integer image pixels.

IV. PERFORMANCE EVALUATION RESULTS

We measured the performance of the ME and TU algorithms in the following environment: the TM3270 and its caches operate at 450 MHz and video stream data is retrieved from a 200 MHz 32-bit DDR main memory. For both algorithms we have two (functionally equivalent) implementations: one that uses the new TM3270 operations and one that doesn't. Furthermore, we measured the performance with and without data prefetching (Table II).

For both algorithms, the impact of new TM3270 operations is more than a factor two, as illustrated by the VLIW instructions count. The ME algorithm was measured to consume 3,075,097 processor cycles per NTSC image. At a frame rate of 25 Hz, this translates into a processor load of 76.9 MHz. The TU algorithm was measured to consume 963,766 processor cycles per image. At a LCD display frame rate of 60 Hz, this translates into a processor load of 57.8 MHz. Prefetching is most beneficial for the TU algorithm: the implementation that uses the new TM3270 operations has its stall cycle percentage reduced from 31.3% to 8.0%

V. CONCLUSIONS

Software implementation platforms provide flexibility, which can even be exploited by very demanding video processing systems. The ME and TU algorithms have a combined load of 137.4 MHz, at a 450 MHz processor frequency, this leaves enough available performance for algorithmic improvements. Furthermore, available processor performance can be utilized to perform other media-processing algorithms, such as video and audio standards, e.g. MPEG2, DivX, H.264/AVC, MP3, etc.

REFERENCES

[1] R.J. Schutten and G. de Haan, "Real-time 2-3 pull-down elimination applying motion estimation / compensation on a programmable device", *IEEE Tr. on Consumer Electronics*, vol. 44, no. 3, pp. 930-938, August 1998.

[2] E.B. Bellers, J.G.W.M. Janssen, and R.J. Schutten, "A software approach to high-quality video format conversion", *ICCE Digest of Technical Papers*, pp. 441-442, January 2005.

[3] G. de Haan et al., "True-motion estimation with 3-D recursive search block matching", *ICCE Tr. on Circuits and Systems for Video Technology*, vol. 3, pp. 368-379, October 1993.

[4] O. Ojo and G. de Haan, "Robust motion-compensated video up-conversion", *IEEE Tr. on Consumer Electronics*, vol. 43, no. 4, pp. 1045-1056, November 1997.

TABLE I. SOME OF THE NEW TM3270 OPERATIONS.

Operation	Description

QUADUMEDIAN
src1 src2 src3 -> dest;
 	
Semantics: 3-taps median.
	dest[31:24] = Min (Max (Min (src1[31:24], src2[31:24]), src3[31:24]), Max (src1[31:24], src2[31:24]));
	dest[23:16] = Min (Max (Min (src1[23:16], src2[23:16]), src3[23:16]), Max (src1[23:16], src2[23:16]));
	dest[15:8] = Min (Max (Min (src1[15:8], src2[15:8]), src3[15:8]), Max (src1[15:8], src2[15:8]));
	dest[7:0] = Min (Max (Min (src1[7:0], src2[7:0]), src3[7:0]), Max (src1[7:0], src2[7:0])) ;

LD_FRAC8 src1 src2 -> dest;
 	
Semantics: load combined with weighted average.
	data0 = Mem[rsrc1]; data1 = Mem[rsrc1 + 1]; data2 = Mem[rsrc1 + 2]; data3 = Mem[rsrc1 + 3];
	data4 = Mem[rsrc1 + 4];
	dest[31:24] = (data0*(16-src2[3:0]) + data1*src2[3:0] + 8) / 16;
	dest[23:16] = (data1*(16-src2[3:0]) + data2*src2[3:0] + 8) / 16;
	dest[15:8] = (data2*(16-src2[3:0]) + data3*src2[3:0] + 8) / 16;
	dest[7:0] = (data3*(16-src2[3:0]) + data4*src2[3:0] + 8) / 16;

TABLE II. MOTION ESTIMATION (ME) AND TEMPORAL UP-CONVERSION (TU) PERFORMANCE EVALUATION: WITH AND WITHOUT NEW TM3270 OPERATIONS, WITH AND WITHOUT PREFETCHING. CYCLE COUNTS ARE FOR A SINGLE NTSC STANDARD DEFINITION 720*480 IMAGE.

Algorithm	New operations?	Prefetching?	VLIW instructions (% of total cycles)	Stall cycles (% of total cycles)	Total cycles
ME	no	no	6,427,255 (95.5 %)	300,980 (4.5 %)	6,728,235
		yes	6,427,255 (99.0 %)	68,032 (1.0 %)	6,495,287
	yes	no	2,924,205 (89.8 %)	333,862 (10.2 %)	3,258,067
		yes	2,924,205 (95.1 %)	150,892 (4.9 %)	3,075,097
TU	no	no	1,918,957 (82.7 %)	400,087 (17.3 %)	2,319,044
		yes	1,918,957 (96.3 %)	74,265 (3.7 %)	1,993,222
	yes	no	886,829 (68.7 %)	403,117 (31.3 %)	1,289,946
		yes	886,829 (92.0 %)	76,937 (8.0 %)	963,766