DSP-ACCELERATED COMPRESSION JPEG 2000

Marek Kváš
Doctoral Degree Programme (1), FEEC BUT
E-mail: marek.kvas@phd.feec.vutbr.cz

Supervised by: František Zezulka
E-mail: zezulka@feec.vutbr.cz

ABSTRACT

This paper deals with the adaptation of an open source JPEG 2000 compression library for using PCI card with a digital signal processor as an accelerator. The main principles of compression JPEG 2000, differences between standard JPEG and JPEG 2000 and advantages and disadvantages of using hardware accelerators are discussed. Paper describes problems which had to be solved like PCI communication and optimization. PCI card based on a digital signal processor TMS320C6415 (the clock frequency 600MHz) was used as an accelerator.

1. INTRODUCTION

Field of image processing systems is being developed at a very fast rate. Sensors have better and better resolution, color depth and frame rates are increasing as well. Increase of all these parameters results in a huge increase of data volume which sensors produce and which has to be processed. Mainly for real-time image processing are often used specialized types of processors with architecture optimized for fast processing of big data volumes. These processors are called digital signal processors (DSP). Solutions based on DSP are very efficient in connection with personal computers (PC). PC offers comfortable user interface and development tools and DSP in a form of DSP based card connected to PC through PCI bus is used as very powerful computation accelerator. Because DSP perform only very specialized and well-defined tasks they can be very efficient and powerful.

Main target of work described in this paper was to adapt OpenJPEG library for work in system described above. OpenJPEG library allows compression of bitmap pictures into format JPEG 2000. OpenJPEG library is open source library and is written in C. Adapted library works under MS Windows XP

2. COMPRESSION JPEG 2000

JPEG 2000 compression is direct successor of well known and widely used compression standard JPEG. JPEG 2000 as well as JPEG was developed by Joint Photographic Expert Group.

2.1. COMPARISON BETWEEN JPEG AND JPEG 2000

We can do a comparison between JPEG and JPEG 2000 in three ways. We can compare formats defined by those standards, structure and principles of compression or quality of results of compressions.

Format JPEG 2000 is much more complex in comparison with JPEG. JPEG 2000 brings many new features. For example: possibility to work with any color model (even with models with more than three colors), possibility of inserting meta-data into files, ability to display picture in different resolutions and sizes (e.g. from not complete data during downloading or because of showing preview), optional data order, region of interests and many other. JPEG 2000 is prepared even for video compression.

The main change in principles is using discrete wavelet transformation (DWT) instead of discrete cosine transformation (DCT). This change brings the possibility of easily implement lossless compression. Lossless compression is not possible in old JPEG. Arithmetic entropic coding process (called MQ-coder) was improved as well. These changes together significantly increase visual quality of lossy compressed pictures.

The third point of view can be the visual quality of pictures after compression, or compression ratio. These two ways of comparison are connected. The better vision quality by the same file size determines the better compression ratio by the same quality. Figure 1 shows quality difference by the same compression ratio.

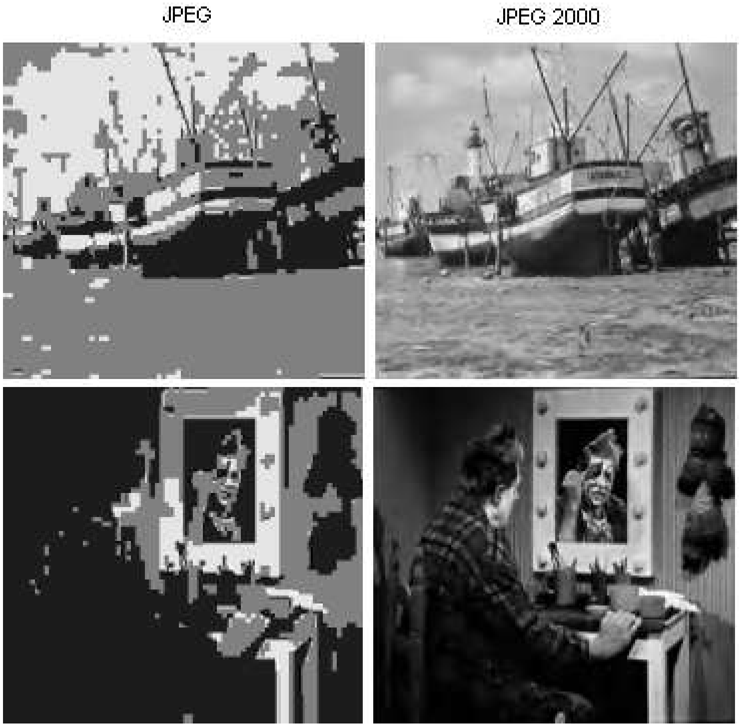

Figure 1: Comparison between JPEG (left Picture) and JPEG 2000 (right pictures) with the same compression ratio 1:40.

2.2. MAIN STRUCTRURE OF JPEG 2000 COMPRESSION

Figure 2 shows the chain of main procedures applied on picture data during compression. The first step is DC level shifting. Data are converted from unsigned format to signed format in this step. It is very simple operation of addition (subtraction). The second step is called multicomponent color transformation. Original color model (mostly RGB) is converted to YCbCr model because of reduction of correlation. Next step is DWT. To concentrate information in picture is main aim of DWT. Quantization is used for lossy compression only. Precision of data is decreased in this block which means reduction of information and loss of quality. Previous steps are for compression very important and they have significant influence on compression features but they don't reduce data value. Real compression is performed in next two blocks, Coefficient Bit Modeling and Arithmetic Coding. These two processes together are in literature called MQ-Coder. MQ-Coder is very complex adaptive process offering big efficiency in data reduction (but without information loss). This process is the most computationally demanding part of all compression. Processing of MQ-Coder takes nearly 90 % of all compression time. The last step is called data ordering. Data are ordered according to user option. Sometimes user wants to see picture first all in small resolution and with increasing resolution during downloading for example and at another time to see first region of interest in full resolution and then step by step other parts of image is better choice.

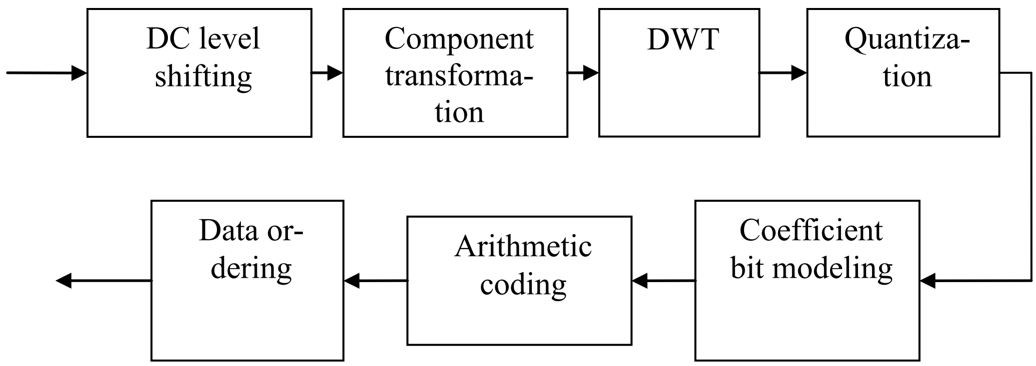

Figure 2: Block diagram of compression JPEG 2000.

2.3. IMPLEMENTATION

Implementation can be divided in two parts. Implementation of interface between DSP and PC is the first one. Communication through PCI on MS Windows platform, fast data transfer through PCI, synchronization DSP and PC processes are main problems of this part. The way how PC processes and DSP processes are synchronized is shown in the figure 3. At the beginning data and control structures (contain parameters of next step) are transferred from PC into DSP memory – this operation starts processing in DSP. Then PC interrupt system is set and compression task is put to sleep – PC processor is free for processing other tasks. As soon as DSP finishes processing or needs next data DSP triggers interrupt in PC. Interrupt is served by an interrupt routine and compression task in PC is woken up.

Implementation of real compression algorithms is the other part of implementation problem. OpenJPEG library developed by Communications and Remote Sensing Lab (TELE), in the Université Catholique de Louvain (UCL) as a JPEG 2000 codec for PC, available as an open source on [2] is used as a basic of project. Library structure was changed according to scheme on figure 3. Main algorithms were replaced with the code for setting and transferring control structures and data in code for PC and were changed for working on DSP. For example access to memory had to be significantly changed and some techniques that can't be used on PC like extended direct memory access (EDMA) were used.

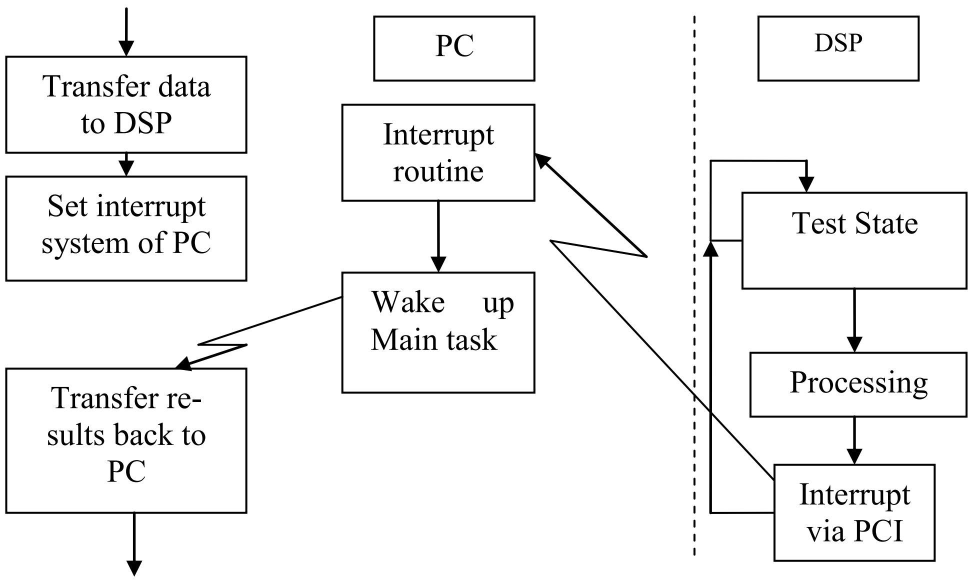

Figure 3: Synchronization of processes in DSP and PC.

2.4. OPTIMISATION

OpenJPEG library source code was written considering mainly functionality, simplicity, transparency and understandability. Speed of code wasn't considered very much; therefore algorithms transferred into DSP were optimized. Optimization was performed on the language C level only. Memory access, cycles and the way how equation was programmed was changed in order to improve efficiency of compiler optimizations. Possibilities of Extended Direct Memory Access (EDMA) subsystem was used in maximal extent and mainly in first three steps of compression helped to make a big performance improvement.

Table 1 shows optimization results for DC level shifting and Component Transformation together. Using EDMA for transferring and separation color data increases speed of those algorithms nearly three times. In most efficient variant EDMA subsystem works autonomously in background and the other task is processing simultaneously. The result is hundred times better than without optimizing.

Modification	Processing time	
	[CPU cycles]	[ms]
Without optimization	8 952 896	14.2
Using DMA for separation	3 505 848	5.84
Computation simultaneously with DMA transfers	116 704	0.19

Table 1: Optimization of DC Level Shifting and Component Transformation.

But optimization efficiency has dependence on the complexity of the algorithm. The more complex algorithm, the more difficult optimization is. MQ-Coder is most complex algorithm in this project. Its behavior is dependent on processing data itself. It cause that data access (and memory access as well) is really unsystematic. Because of this optimization is very difficult and wasn't very successful. Next effect of unsystematic data accesses is memory subsystem inefficiency which causes low speed of MQ-Coder. In spite of this basic optimization, the speed improved by 20 %.

Velikost obrázku	**64 x 64**		**128 x 128**		**256 x 200**	
	DSP [µs]	**PC [µs]**	**DSP [µs]**	**PC [µs]**	**DSP [µs]**	**PC [µs]**
Data transfer PC → DSP	113		508		1 603	
DC Level Shifting + komponent transformation	277	360	470	8370	973	3349
DWT	738	1 812	2 954	6 817	8 147	20 955
MQ-Coder	160 772	55 706	273 840	198 110	627 829	522 578
In total	**161 920**	**57 878**	**277 772**	**212 297**	**638 552**	**546 882**
PC is faster	**64%**		**24%**		**15%**	

Table 2: Comparison between system with and without acceleration.

3. CONCLUSION

In table 2 is the comparison of compression speed with and without DSP acceleration. The PC without acceleration was faster for all shown examples. However we must note PC with doublecore processor on frequency 2.8 GHz was used for tests and DSP was working on frequency 600 MHz. Inspite of big power difference speed was nearly the same. In case of DSP accelerated system the PC processor is nearly free for other tasks. This is advantage in comparison with not accelerated system where processor must very intensively process compression task. Result of this work is only a first small step to practical applicable system but experiments and measurements heve shown that this way is possible.

ACKNOWLEDGEMENT

This work has been supported in part by Ministry of Education, Youth and Sports of the Czech Republic (Research Intent MSM0021630529 Intelligent systems in automation)

REFERENCES

[1] JPEG: JPEG 2000 Image Coding System: Part 1 Final Committee Draft Version 1.0, 2000, Available on <www.jpeg.org> (1.2.2007)

[2] Vershueren, J.: Open JPEG, 2003. Available on <www.openjpeg.org> (1.2.2007)

[3] Impoco, G.: JPEG 2000 – A Short Tutorial, 2004. Available on <www.dmi.unict.it/~impoco/files/tutorial_JPEG2000.pdf> (1.2.2007)

[4] Kváš Marek: 2-D Signal Processing Based on TMS320Cxx structures [Diploma thesis], BUT Brno, Department of Control and Instrumentation