Towards a Mobile Implementation of Waaves for Certified Medical Image Compression in E-Health Applications

Imen Mhedhbi[1], Khalil Hachicha[1], Patrick Garda[1], Yuhui Bai[2], Bertrand Granado[2], Sébastien Topin[3], and Sylvain Hochberg[3]

[1] UPMC, LIP6, CNRS UMR 7606; 4 Place Jussieu, 75252 PARIS Cedex 05, France
[2] ETIS, CNRS UMR 8051, ENSEA, Université Cergy Pontoise; 6 avenue du Ponceau, 95014 CERGY Cedex, France
[3] CIRA, 38 Boulevard Henri Sellier, 92156 SURESNES, France

Abstract. In this article, we present two studies that pave the way towards a mobile implementation of the WAAVES certified medical image compression encoder. On the algorithmic side, we compared three techniques to increase the compression rate. The obtained results show a significant bit-rate reduction, around 40% with respect to the WAAVES encoder, while keeping the same visual quality. On the architectural side, we describe the HW/SW co-design of an architecture implemented in a FPGA platform. By using code profiling, critical portions of the code were identified, then two methods for hardware acceleration were used to implement the critical part of the coder. The tests were done on a StratixIVGX230 FPGA and the results showed that HW/SW co-design could achieve up to 20x performance gain in the critical portion. The combination of these results demonstrates the feasibility of a mobile implementation of the WAAVES certified medical image coder suitable for e-health applications.

Keywords: Medical images, Image compression, Motion detection, Markov Models, SSIM, FPGAs, NIOS II, HW/SW co-design, Medical devices.

1 Introduction

Nowadays, medical images are becoming an important source of information for a medical expert to realize a diagnosis. But their size is growing continuously to provide more accurate information. In addition, some examination such as endoscopy or angiography, use video imaging, thus increasing tremendously the amount of data and hence the need for their compression. Fortunately, there is a clinically validated compression algorithm that reduces with a big ratio the volume of images while ensuring sufficient quality for medical diagnosis: this is the WAAVES coder developed by CIRA [1]. The basic steps used in its compression scheme include Discrete Wavelet Transformation (DWT) and quantization followed by entropy coding to encode the resulting coefficients. This solution is compatible with DICOM (Digital Communications in Medicine). It was certified as a Medical Device.

B. Godara and K.S. Nikita (Eds.): MobiHealth 2012, LNICST 61, pp. 79–87, 2013.

The challenge today is to give access to these medical images remotely on embedded terminals with low computing power through low bandwidth networks. On the one hand, to address the network bandwidth limitation, we propose to improve the compression performance of the original WAAVES algorithm. Specifically, in the case of video imaging, our approach is to divide the medical images flow into images of reference and images of difference that are then compressed by the WAAVES coder. On the other hand, to address the low computing power of embedded platforms, we propose a dedicated architecture and we developed an efficient HW/SW co-design technique for real time execution of MMWAVES on embedded terminals.

This article is organized as follow: section 2 presents our masking algorithm principle. Next, we describe MMWAVES and give the compression performances results. In section 3, we describe the real-time image acquisition hardware interface and the HW/SW co-design architecture for WAAVES encoder. Next, we present the optimization of performance by using NIOS Custom Instructions and Hardware Accelerator IPs. The implementation results in terms of time partition are discussed in section 8. Finally, we conclude the paper and present future work.

2 MMWaaves

To increase the compression rate for video sequences, our approach is to combine a motion detection algorithm with WAAVES to create the MMWaaves coder (Motion Mask WAAVES) as shown in Figure 1. More specifically, we divide the medical images flow into images of reference and images of difference and we use a masking technique based on Markov models to reduce the difference images noise. This technique was introduced, patented and validated in previous works [2][3] with different video coders (MJPEG2000, H264) and it achieved good results for generic video benchmarks. We describe the motion detection in section 2.1.

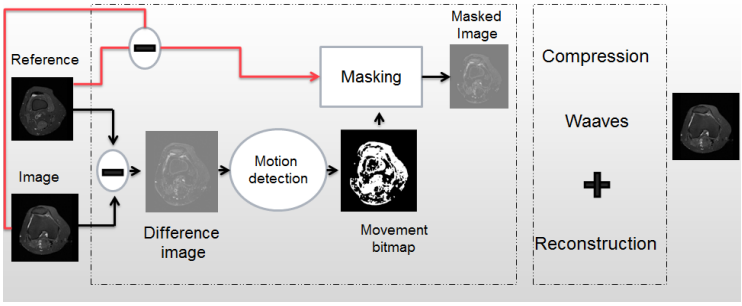


Fig. 1. Mask Motion Waaves

For the masking algorithm, three techniques were studied: the *difference*, *substitution* and *Exclusive-or*. The *difference* means that, when we get a "1" pixel in the binary mask map, we write in the output image the difference between the same pixel in the reference and in the current image. The *substitution* means that, when we get a "1" pixel

in the binary mask map, we write in the output image the same pixel in the current image. The *Exclusive-or* means that, when we get a "1" pixel in the binary mask map, we write in the output image the exclusive-or between the same pixel in the reference and in the current image. The performances achieved for these three techniques on a set of medical images will be given in the following.

At first, we reworked a model using the potential functions foreseen by the detection of motion combining the spatial and temporal information [5][6][7]. It is composed of two distinct steps: the first consists of a preprocessing phase through which the variance is determined. The absolute value of the difference matrix is then calculated and binarized by setting a threshold.

The second grouping algorithm of the ICM to update the binary state of the pixels of difference (moving or not) which is made site by site in the sense that every change in state is taken immediately into account in the relaxation of the neighboring site. In this way, it will allow the convergence to the first minimum of the energy function. In order to calculate the energy, one must know the state of the pixels belonging to a neighborhood defined by eight spatial neighbors and two temporal neighbors. The principle of this algorithm is presented in Figure 2.

To choose one of these three techniques, we proposed a multi-criteria performance evaluation based on an objective measure (PSNR, SNR) and a psycho visual measure (SSIM index). Tests were carried out using 512x512 images. We applied a subtraction

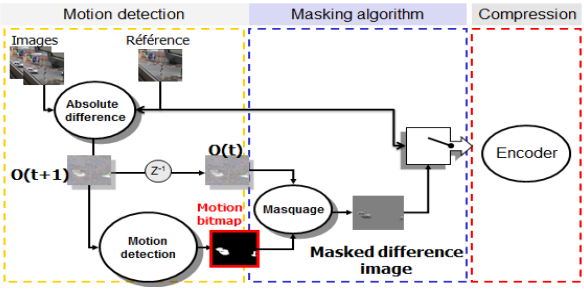


Fig. 2. Masking technique

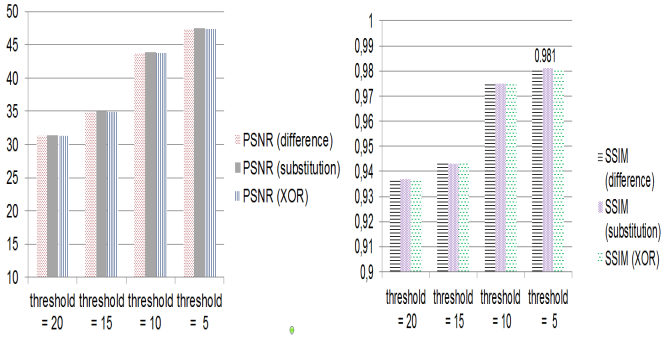


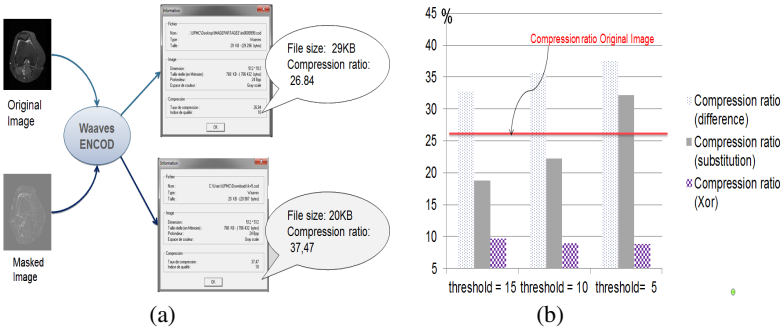
Fig. 3. (a) Threshold impact on PSNR (b) Threshold impact on SSIM

operation between the reference image and the current image and we computed the mask using the Markov model. To measure the performance of the quality of the image, we applied these two metrics.

Figure 3(a) summarizes the impact of the binarization on the PSNR evolution. Firstly, we notice that we find the best PSNR, equal to 46, using a threshold equal to 5. Secondly, we remark that PSNR's values are almost identical for the various techniques. This is due to the use of the same motion bitmap.

However, for medical images, PSNR metric does not accurately reflect image quality as perceived by a clinician. Thus we used the index of structural similarity SSIM[8][9] as a metric measuring the psycho visual quality of image. SSIM has a value between 0 and 1 that indicates the correlation with respect to the source, where 1 indicates a perfect correlation. The diagram in Figure 3(b) summarizes the results giving the impact of the binarization threshold on the evolution of the SSIM index. We showed that the reconstructed images using a threshold equal to 5 got the best quality. This confirms the measurements obtained previously using the PSNR metric.

Finally, we made a comparison between the compression of masked images (difference, substitution, Or Exclusive) and the compression of the original images (Figure 4(a)).



(a) (b)

Fig. 4. (a) Waaves compression (b) Threshold Influence on evolution of compression ratio

The diagram in Figure 4(b) shows the impact of the binarization threshold on the compression ratio of the different images. We note that we got a loss of compression performances using the exclusive-or technique. However, we got a gain that varied between 21% for the substitution technique and 42% for the difference technique using a threshold equal to 5. The gain could reach 50% for other sets of test images.

In conclusion, the results demonstrate that a significant gain in compression rate can be achieved while preserving the image quality performances.

3 Embedded Encoding System

To embed the Waaves and MMWaaves algorithms, we need to accelerate their execution on an embedded terminal. To realize this task, we preliminary use an

experiment platform with a camera and a FPGA Altera StratixIV GX230 that was chosen for its outstanding benchmarking in terms of density, performance and power consumption among 40-nm FPGA family.

The diagram of our camera system platform is briefly presented in Figure 5(a), the FPGA is connected with off-chip devices trough GPIO, such as a 5-megapixel CMOS camera that provides digital raw image data and a VGA monitor to display the compressed then uncompressed images. The camera controller component provides an interface between the Cmos camera and the FPGA to send the acquired real time image data to the memory buffer and further to an external DDR, while the VGA controller allows transferring data from the memory buffer to the VGA monitor for visualization. The DDR controller is composed of two DMA modules which allow circular transfer between the on chip FIFO and the external DDR memory.

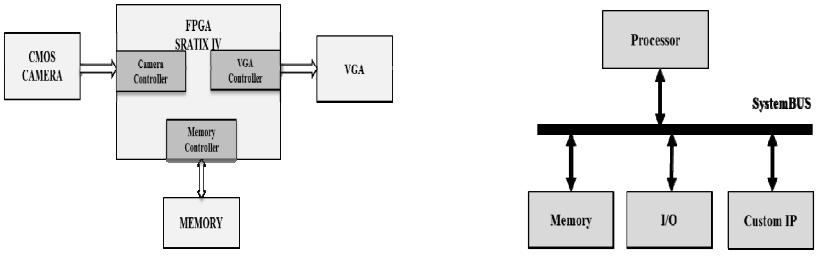


Fig. 5. (a) Camera system (b) A generic SOC system

4 HW/SW Co-design Encoding System

A generic system on chip is illustrated in Figure 5(b) in order to develop hardware and software in parallel. The processor, memory and peripherals, along with hardware IPs, are connected with a NoC (Network on Chip) interface to ease function re-use and system co-design.

Our SoC co-design is conducted using Altera's design flow with the SOPC builder tool. Instead of using fixed hardware cores, the NIOS II Soft-core processors are designed to fit and run inside an FPGA. The core is a 32-bits scalar RISC Processor that allows the designer to add user-defined instructions [11] in addition to pre-specified features to give a highly focused, programmable processor solution. Here, NIOS II soft-core processor is used as an experiment processor to validate the HW/SW codesign and the acceleration of both algorithms with techniques like user defined instruction and IP based accelerator.

The Waaves wavelet-based image encoding algorithms were first coded in C++ and validated on a PC host. The tested code was then compiled for the Nios II Integrated Device Electronics (IDE) and deployed in the development board. The standard HW/SW co-design process can consist of three main steps [10], which include the implementation of the Algorithm into software, followed by an analysis of the program with profiling to detect the critical parts of Algorithm, then efficiently

implement the algorithm in hardware. The detection of critical part of the software and the optimization will be discussed in the next section.

4.1 Timing Analysis of Application

In order to optimize and achieve the best performance in terms of real-time operation of the image encoder, we identified the critical portions in the software by using Altera Performance Counters [16], which allow measuring the number of processor clock cycles for the execution time.

As illustrated in Figure 6, DWT (Discrete Wavelet Transform) and Encoder (Entropy Encoder), which use respectively 60 % and 30 % of the execution time, are the two primary computationally intensive components. Thus, two methodologies were adopted to accelerate the critical component.

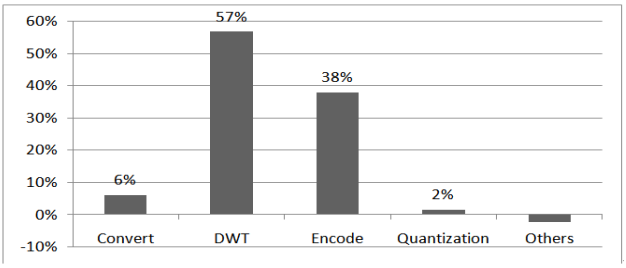


Fig. 6. Execution time distribution for image encoder

4.2 Optimization of the DWT Module with Custom Instructions

The 2D DWT module is used for decomposition of the images with high-pass and low-pass filtering and a factor of two sub sampling, the operations are applied in row and column wise direction and are implemented in floating point. Since Nios II does not have a floating point unit (FPU), like many processors in embedded systems, the processing of DWT becomes expensive. Altera provides an efficient way of adding small custom instructions in the Nios II processor using its Sopc/Qsys tools. Custom instructions is a straightforward option for accelerating software in FPGAs, which allows increasing system performance by offloading portions of the software code to hardware functions [11][12]. The custom instruction presents a data path in parallel to the CPU's arithmetical logical unit (ALU). During system generation, special assembler instructions are generated to access the additional component. Nios Embedded Design Suites directly generates a macro in to simplify the access to the hardware component; each custom component can transfer at most 64 bits of data. Thus the simple operations can be replaced by several custom instructions. With the Nios II floating point custom instructions, we can accelerate arithmetic functions executed on float variable types and are able to take full advantage of the flexibility of the FPGA to optimize the system performance.

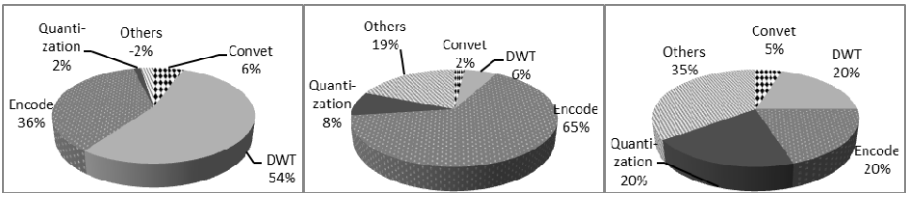
4.3 Optimization of Encoder Module with Hardware Accelerator IP

The wavelet based coding scheme is used in several image coding methods, among which the state of art coding algorithm EZW [13] and SPIHT [14] both proposed the progressive sub-band coding based on the frequency and spatial correlation after sub-band transform, to encode separately each resolution level of the image in an progressive way. In our application, the Encoder module, that takes 30% of total execution time, consists of an adaptive scanning algorithm which reorganizes the wavelet coefficients of each sub-band to get a better compression rate [15].

This algorithm is inefficient in software because the manipulated coefficients are accessed with two linked list and a memory block. The linked list needs to be updated frequently to select the wavelet coefficients. The utilization of linked list requires extremely random memory access. In order to reduce the random memory access, our hardware accelerator is designed by using a structure of grouped linked for accelerating the adaptive scanning algorithm. Instead of using 3 memory blocks to separately store the indexes, one memory block, which is three times larger, is used to store the coefficients. Thus the DMA could use memory burst to reduce the number of random memory access, which leads to more time efficiency. The IP is designed in Verilog and it is connected to the processor via the Avalon bus interface. Due to the stand-alone design of the hardware component, a simple re-use of the design files in different architectures is supported.

5 Implementation Results

Results were obtained with the implementation of the wavelet-based image encoder in FPGA. The DWT module was optimized with floating point custom instruction set and the Encoder module was optimized with hardware accelerator IP. The prototyping board operated at 100 MHz. The system received images from the camera and restituted processed images on the VGA monitor. Our implementation on the prototype showed up to 20 times speed improvement for the DWT module and 10 times speed improvement for the Encoder module compared to the software based solution. The results before and after timing optimization are presented in Figure 7.



(a) (b) (c)

Fig. 7. CPU time partition before and after optimization : (a) Average coding time before using Custom Instructions (b) Average coding time after using Custom Instructions (c) Average coding time after using Hardware Accelerator

6 Conclusion

In this work, we proposed to combine the WAAVES coder with a mask motion detection algorithm based on Markov Model (MMWAAVES). We developed three techniques to create masked images and we evaluated the image quality based on two measures: objective (PSNR, SNR) and Psycho visual (SSIM index). We demonstrated that masking the image differences gives the best results and allows a compression gain up to 42%. We also described the HW/SW co-design architecture of WAAVES based on FPGA platform. We proposed two methods for hardware acceleration, which led to 20x speedup for the critical part of the encoder compared to pure software based solution. Our future work will focus on the improvement of the masking process and building hardware acceleration IP for the critical parts.

Acknowledgments. This work takes part in the WARM project with the support from the FEDER/FUI funds. The authors thank the HEGP and PARTELEC partners of the project for fruitful exchanges.

References

1. Created in 1998, CIRA has developed WAAVES, a digital imaging compression technology offering a paradigm shift in compression, transfer and recovery quality performance. Based on a major professional and social potential impact the technology may have in public healthcare, CIRA has developed an initial strategic focus in medical imaging. WAAVES has been partnered with a growing number of medical applications, such as Apicrypt, MacDent, Medistory, PDB, SantNet and, recently, Dentalvia. A number of French University hospitals have adopted WAAVES as their imaging technology standard (1998), `http://www.waaves.com`,
2. Bouthemy, P., Lalande, P.: Recovery of moving object masks in an image sequence using local spatiotemporal contextual information. Optical Engineering 32, 1205–1212 (1993)
3. Lohier, F., Garda, P., Lacassagne, L.: Procédé et dispositif de traitement de séquences d'images avec masquage. Brevet Français UPMC FR2804777 (Août 10, 2001), Brevet Européen EP1297494 (Avril 2, 2003)
4. Hachicha, K., Garda, P.: Accelerating the multiple reference frames compensation in the H.264 video coder. Journal of Real-Time Image Processing 4(1) (March 2009) ISSN1861-8200
5. Luthon, F., Caplier, A.: Motion detection and segmentation in image sequences using Markov Random Field Modeling. In: 4th Eurographics Animation and Simulation Workshop, pp. 265–275 (September 1993)
6. Hachicha, K., Faura, D., Romain, O., Garda, P.: Noise-robustness improvement of the H.264 video coder. Journal of Electronic Imaging, SPIE and IS&T 17(03), 033019 (2008)
7. Lohier, F., Garda, P., Lacassagne, L.: Procédé et dispositif de traitement de séquences d'images avec masquage. Brevet Français UPMC FR2804777 (Août 10, 2001), Brevet Européen EP1297494 (Avril 2, 2003)
8. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)

9. Taubman, Marcellin, M.W.: JPEG 2000: Image Compression Fundamentals, Standards, and Practice. Kluwer Academic Publishers (November 2001)
10. Atitallah, A.B., Kadionik, P., Ghozzi, F., Nouel, P., Masmoudi, N., Levi, H.: An FPGA implementation of HW/SW codesign architecture for H.263 video coding. AEU – International Journal of Electronics and Communications (December 2006)
11. Altera nios custom instructions User Guide – Altera
12. Altera SOPC Builder User Guide – Altera
13. Shapiro, J.M.: Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. Signal Process. 41, 3445–3462 (1993)
14. Said, A., Pearlman, W.A.: A new fast and efficient image codec bsed on set partitioning in hierarchical trees. IEEE Trans. Circuits Syst. Video Technol. 6, 243–250 (1996)
15. Haapala, K., Lappalainen, V., Hämäläinen, T.D.: Experimental parallel implementation of a wavelet-based still image encoder. Microprocessors and Microsystems 29(4), 155–167 (2005)
16. Profiling Nios II Systems – Altera