Research Journal of Applied Sciences 1(1-4): 26-32, 2006

Implementation of a Video Cut Detector on a Mixed Architecture DSP/FPGA

[1,2]Abdellatif Mtibaa, [3]Lotfi Boussaid and [1]Houda Ben Mansour
[1]Laboratoire d'EμE, Faculté des Sciences de Monastir, Avenue de l'Environnement-5019 Monastir, Tunisie
[2]Ecole Nationale d'ingénieurs de Monastir, Avenue Ibn ElJazzar 5019 Monastir, Tunisie
[3]Laboratoire CES, École Nationale d'ingénieurs de Sfax, Route Sokkra, Sfax, Tunisie

Abstract: In this study, we present a hardware/Software design approach for multimedia applications. The studied application concerns the automatic video segmentation based on the block histograms algorithm. Our design study consists in exploring the different solutions of the conception flow on a mixed platform based on a Texas DSP (TMS320C6201) and a Virtex FPGA of Xilinx.

Key words: Video segmentation, block histograms, hardware implementation, codesign, DSP, FPGA

INTRODUCTION

Digital signal treatment applications and especially those concerning image processing become more complex and require important performances in terms of time computing, power consuming and surfaces. This aspect is more crucial in the presence of real-time constraints in the case of image and video applications.

Nowadays, the new solutions consist in using parallel and mixed architectures often constituted by a software component (processor) and a hardware component (VLSI device). In fact, the ever-increasing embedded system and the tight time-to-market window has revolutionized the system on chip design process. The concurrent design of hardware and software has displaced traditional sequential design. Designer has to define a system architecture consisting of creating system functions that form the basis of concurrent hardware and software design[1].

On the other hand, the development of the hardware technology and communications infrastructure has made applications such as automatic analyzing of video content very challenging. In this study, our target application concerns video segmentation which is also known as Shot Boundary Detection (SBD). The used approach is the one based on block histograms which was proposed by Nagasaka and Tanaka[2]. They have divided each frame into 16 blocks and computed local histogram before evaluating a difference metric. Histogram based methods have been shown a good performance for shot cut detection[3,4].

In produced video such as television or movies, shots are separated by different types of transitions, or boundaries. The simplest transition is a cut, an abrupt shot change that occurs between two consecutive frames. Gradual transition such as fades and dissolves are more complex. Shot boundaries are fades when the frames of the shot gradually change from or to black and can be dissolves when the frames of the first shot are gradually morphed into the frames of the second[5]. Fig. 1 shows an example of transition effects. The study of the state of the art related to video segmentation shows that several methods for SBD were proposed. These methods can operate in different environments such as temporal, frequency, uncompressed and compressed domains.

DESIGN PROCESS

Block histograms specification: The color histogram of an image is constructed by counting the number of pixels of each color. More formally, the color histogram is defined as the probability mass function of the image intensities. To increase the quality of shot change detection block-based methods were proposed[2,6-8]. The main advantage of these methods is their relative insensitivity to noise and camera or object motion. In this work we have used the approach proposed by Nagasaka[2] who divided each frame into 16 blocks and computed the distance of dissimilarity between consecutive frames fn and fn+1 as follows:

$$D(f_n,f_n+1)=\sum_{c\in RGB}\sum_{b=1}^{16}\sum_{j=0}^{N-1}|H(f_{n+1},c,b,j)-H(f_n,c,b,j)|$$

Where c, b and N are respectively the luminance of color components of the picture (red, green and blue), the number of blocks and the number of bins of the pixel value.

Corresponding Author: Abdellatif Mtibaa, Laboratoire d'EμE, Faculté des Sciences de Monastir, Avenue de l'Environnement-5019 Monastir, Tunisie

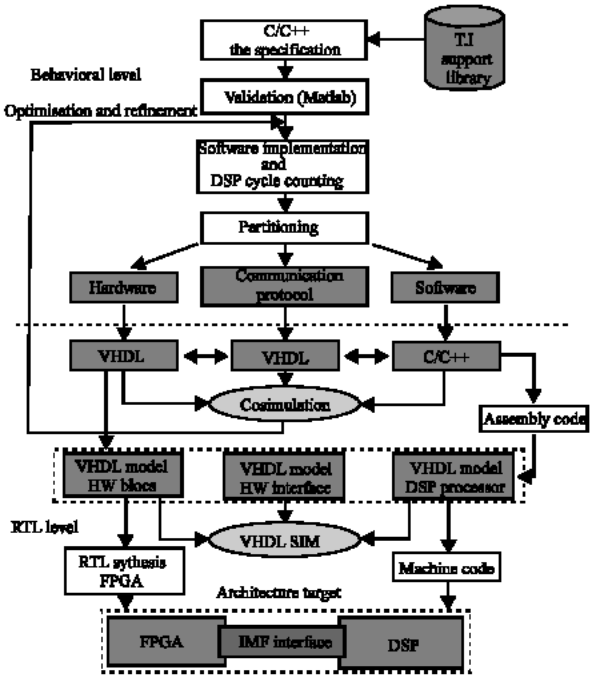

Fig. 2: Conception

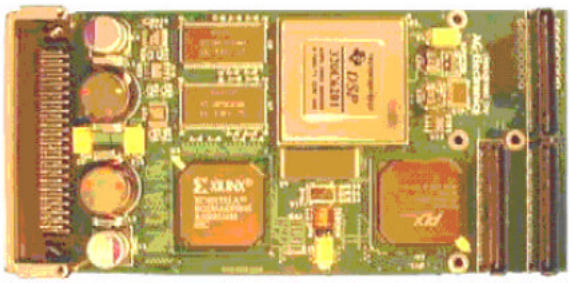

Fig. 3: Platform target (DM11 Agelectroincs)

$H(f_n,c,b,j)$ is the histogram value for frame f_n and for the discrete value of the intensity k. The value of k is in the range [0,N-1], where N is the number of discrete values a pixel can have. To detect break shots, the metric $D(f_n,f_{n+1})$ is compared to a global threshold. When this metric exceeds the value of threshold, it indicates that a shot transition has occurred. The experiments done in Boussaid[9] have shown that working in the gray space and uniform quantization at 4 levels (bins) present reliable results and relatively low computation time. The distance of dissimilarity becomes:

$$D(f_n,f_{n+1})=\sum_{b=1}^{16}\sum_{j=0}^{3}|H(f_{n+1},b,j)-H(f_n,b,j)|$$

Design methodology and conception flow: An electronic system is often formed by a collection of hardware circuits and software executable code in narrow interaction.

The present tendency, in the case of complex systems (ex. SOC), is to take into account the interaction between ware and hardware components since the specification phase. Thus, the separation is done as late as possible in the design cycle. This permits to unify the specification in order to include both the hardware and the software. This approach is named "hardware/software codesign", or briefly, "codesign"[10]. The adopted strategy of design is presented in the Fig. 2. The point of entries is described in C language. Then what to be achieved is a functional validation whose results would be compared with a description in Matlab.

The Partitioning is realised manually by starting with a 100% software description and 0% hardware description. We migrate, then, more and more toward the hardware until reaching an optimal step which could satisfy the constraints imposed by the application. The hardware part and the communication interface are then transformed in VHDL RTL. The software part remains in C language. At this level, the co-simulation is ensured. In the synthesis stage of the hardware part, the Xilinx Foundation Environment[11] has been adopted. This environment assured different tasks of development such as the creation of the circuit source, the programs verification, the simulation, the synthesis, the implantation and the control of all relative aspects in the whole design. For the software, the "Code to Composer Studio[12]" has been adopted as a unified environment for all Texas Instrument DSP. It permits the control of the different steps of development and the simulation tests. Starting with a source file written in C or in C++, we get, thanks to Code Composer Studio, the assembly file and the executable file which will be implemented on DSP.

Target architecture: The adopted target architecture, which is presented in Fig. 3, is the DM11 card of Agelectronics. It has been basically conceived for real time applications. It is constituted around a processor software DSP, the TMS320C6201 of Texas Instrument, a Virtex XCV50 FPGA of Xilinx and the "EMIF: standard communication interface" which assure the communication between the DSP and the FPGA circuit[12].

EXPERIMENTAL RESULTS

Functional validation: The block histograms approach in the gray space was tested with 256, 8 and 4 levels of

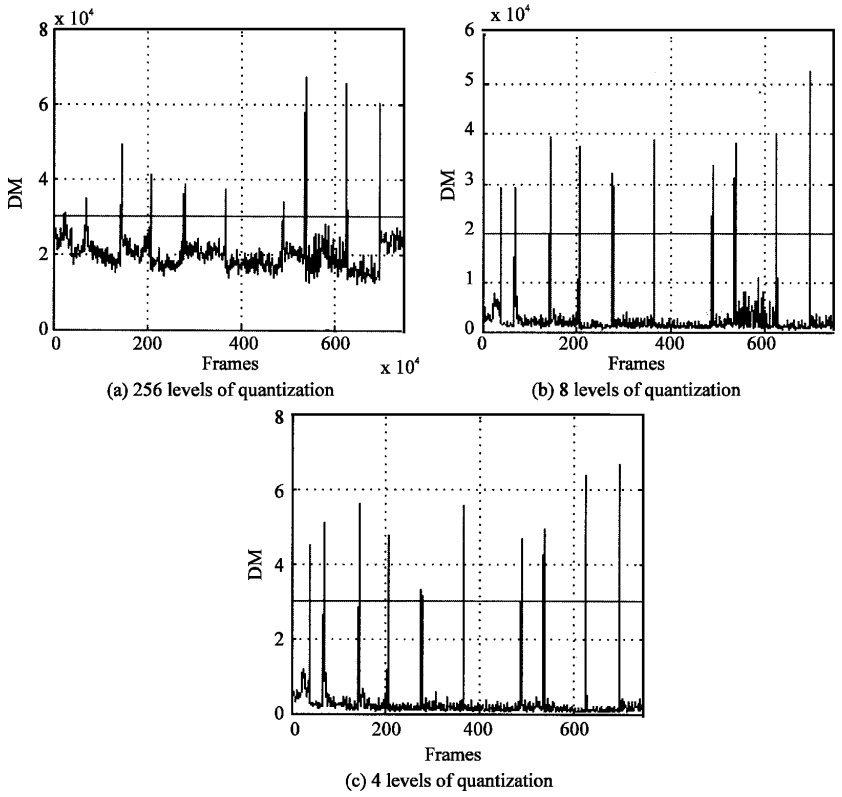

Fig. 5: Scene change detection

quantization on a 30 seconds video sequence from scorpion king movie. This video sequence has a 320 x 240 image size and a rate of 25 frames per second. It is composed of 15 shots mounted with 11 cuts and 3 dissolves.

The process of cut detection based on block histograms includes five tasks:

- Quantization;
- Pixel localization;
- Histograms calculation;

- Metric of dissimilarity computing;
- Comparison to a predetermined threshold value.

If the metric of dissimilarity $D(f_n,f_{n+1})$ exceeds the threshold, it indicates so a scene change occurrence. The first frame of a shot is usually called key- frame. Extracted key-frames of video sequence test are presented in Fig. 4. The effect of quantization on scene change detection is described by Fig. 5. In fact, Fig. 5a shows an important sensitivity to light and rapid colour changing. This sensibility to illumination was enormously reduced with a quantization of 8 or 4 levels (Fig. 5b and 5c).

The 4 levels quantization permits not only to reduce the false detection due to flash effects, but also, it permits to reduce the computational time (Histogram Vectors of dimension 4). The time computational of the block histograms approach completely implemented on DSP (100% software), for different levels of quantization, is shown in Fig. 6.

Hardware implementation: To work in real-time conditions, signals of synchronization have to be taken into account. In fact, for analog PAL TV, images are constituted by 625 lines of 64 µs. During a line screening, beam of light must be turned off 2 µs before the horizontal synchronization pulse and almost 10 µs after. Visible pixels remain effectively about 52 µs. In the same way, only 575 lines were really used in the screen. The 50 lines have been used for the vertical synchronization of interlaced video (odd and even frames). The inter-image time is about 3.2 ms (2 x 1.6 ms per frame).

In digital TV, the streaming video at 25 Mb/s is composed by DCT based compression video two or four tracks of audio, a Time Code (TC) track and CRC redundancy check track. Audio, TC and CRC tracks represent the inter-image time presented for the analog TV[13]. For digital video, we digitalize only 720 pixels instead of 856 per line. The rest of the entire line is replaced by sound, TC and CRC. For audio we have to choose between:

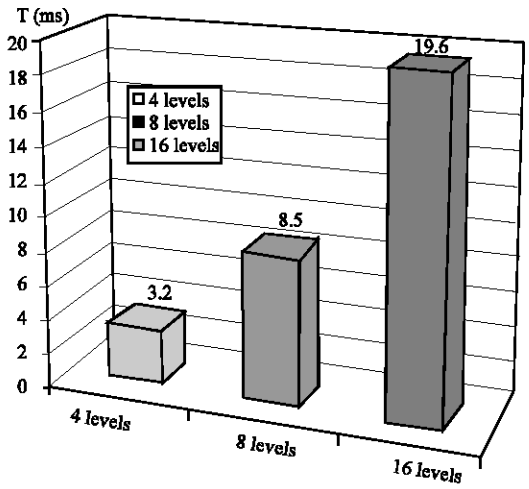

Fig. 6: Time computational of DM for different levels

Table 1: Rate of used resources

Number of slices	127 of 768 (16%)
Number of flip-Flops	105 of 1536 (6%)
Number of "LUTs"	153 of 1536 (9%)
Number of O/I	53 of 180 (29%)
Number of "GCLK"	1 of 4 (25%)

Table 2: Signals propagation latency

Minimum period (Max. Frequency)	11,781ns (84,882MHz)
Maximum combinational path delay	13,126ns
Maximum net delay	7,821ns

- 2 tracks at 48Khz 16 bits = 1536 kb/s
- 4 tracks at 32Khz 12 bits = 1536 kb/s

In digital, the inter-image time is about 1.9 ms. This dead-time must be sufficient to compute our study in real-time condition. Since the visible pixels in a line remain 52 μs and the video sequence is of 320 x 240 pixels, video clock is therefore of 162.5 ns. To respect this condition, pixel value must be converted and affected to corresponding histogram vector before the coming of the next pixel. After the exploration of the space of solutions, we have decided to compute the three first steps on DSP and implement the dissimilarity metric and comparison to threshold on FPGA. In the experiments, the elapsed time for the three first tasks (quantization, localization and histogram vector incrementation) computed on DSP was estimated to 70.68ns. For hardware part, FPGA resources were used for two purposes:

- Compute the dissimilarity metric and compare to threshold.
- Implement the interface of communication between hardware and software parts (FPGA/DSP).

The interface between DSP and FPGA is constructed by Agelectronics in order to facilitate the communication and data changes between hardware and software parts.

The hardware implementation on FPGA was simulated and synthesized with the Xilinx "PROJECT MANAGER TOOL". Fig. 6 describes the simulation process. The logical resources used for hardware implementation are presented in Table 1. The propagation signals are given in Table 2.

CONCLUSION

In this study, we have presented a case study of multimedia application which concerns the automatic video segmentation based on block histograms. The use of Codesign approach and the Hardware/Software platform design kit have satisfied the real-time constraints imposed by the application. Our future work consists in integrating DSP and hardware parts of the design on an embedded System on Chip (SOPC).

REFERENCES

1. Ernst, 1998. Codesign of Embedded Systems: Status and Trends. IEEE Design an Test of Computers Journal.
2. Nagasaka, A. and Y. Tanaka, 1992. Automatic Video Indexing and Full-Video Search for Object Appearances. Video Database Systems, pp: 113-127.
3. Dailianas, A. and al, 1995. Comparison of Automatic Video Segmentation Algorithms. International Issues in Large Commercial Media Delivery Systems, Proc. SPIE, 2615: 2-16.
4. Lefèvre, S. and al, 2003. A Review of Real-Time Segmentation of Uncompressed Video Sequences for Content-Based Search and Retrieval. Real Time Imaging, 9: 73-98.
5. Porter, S.V. and al, 2001. Detection and classification of shot transitions. Proceedings of the 12th British Machine Vision Conference, T. Cootes and C. Taylor, Editors, BMVA Press, pp: 73-82.
6. Kasturi, R. and R.C. Jain, 1991. Dynamic Vision. Computer Vision: Principles. IEEE Computer Society Press, Washington, pp: 469-480.
7. Ren, W. and al, 2001. Automated Video Segmentation. International Conference in Information, Communication and Signal processing, Singapore.
8. Swanmberg, D. and al, 1991. Knowledge Guided Parsing and Retrieval in Video Database Systems. SPIE Conference on Storage and Retrieval for Image and Video Databases, San Jose, CA, 1908: 13-24.

9. Boussaid, L., A. Mtibaa, M. Abid and M. Pandavoine, 2005. Real-Time Shot Boundary Detection Algorithm Based on Local Histogram. SSD'05, IEEE International Conference on Signals Systems Decision and Information Technology, Sousse, Tunisia.
10. Abid, M., 2000. Contribution to design of mixed electronic systems software-hardware Codesign. Thesis degree of ENIT High School, Electrical, Tunis-Tunisia.
11. www.xilinx.com
12. http://focus-webapps.ti.com
13. Glossary for Professional Video http://bhphotovideo.com