

Péter SERFŐZŐ*, József VÁSÁRHELYI*, Attila VARGA*

ANALYSIS OF HARDWARE RESOURCES OF MOJETTE TRANSFORM IMPLEMENTATION

ANALÝZA HARDWAROVÝCH ZDROJŮ IMPLEMENTACÍ MOJETTE TRANSFORMACE

Abstract

Mojette transform (MoT) is an exact discrete Radon transform defined for a set of specific projections. The MoT is used for image processing applications. The paper analyzes the implementation of Mojette transform and Inverse Mojette transform (IMoT) in Field Programmable Gate Arrays. Tries to outline the development work in order to create an embedded reconfigurable hardware based on Xilinx ML310 board.

Abstract

Mojette transformace (MoT) je diskretní Radanova transformace definovaná pro specifický soubor předpokladů. MoT se používá pro zpracování obrazu. Tento příspěvek analyzuje implementaci Radanovy transformace a Inverzní Radanovy transformace (IMoT) v poli logických členů programovatelných uživatelem (FPGA). Pokouší se vymezit provedený vývoj v závislosti na tvorbu vložené hardwarové Xilinx ML310 desky.

1 INTRODUCTION

Inscribing invisible marks (watermarking) into an image has different applications such as copyright, steganography or data integrity checking. Many different techniques have been employed for the last years in different spaces (Fourier, wavelet, Mojette domains, etc.) [1-9]. The paper describe development work related to create the functional block scheme of Mojette transform and Inverse Mojette transform embedded hardware.

2 DIRECT AND INVERSE MOJETTE TRANSFORM

The Mojette transform (MoT) [6, 7, 8] projects the original digital 2D image $F = \{ F(i,j); i = 1, \dots, N; j = 1, \dots, M \}$ onto a set of K discrete 1D projections with $M = \{ M_k(l); k = 1, \dots, K; l = 1, \dots, l_k \}$.

$$F \xleftarrow{MoT} M \quad (1)$$

MoT is an exact discrete Radon transform defined for a set $S = \{(p_k, q_k), k = 1, \dots, K\}$ specific projections angles.

$$M_K(l) = proj(p_k, q_k, b_l) = \sum_{(i,j) \in L} F(i, j) \delta(b_l - iq_k - jp_k) \quad (2)$$

* University of Miskolc, Department of Automation, Miskolc, Hungary, {serfozo1, vajo, avarga}@mazzola.iit.uni-miskolc.hu

$$\text{where } \delta(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{if } x = 1 \end{cases} \quad (3)$$

$$\text{and } L = \{(i, j); b_l - iq_k - jp_k = 0\} \quad (4)$$

is a digital bin in the direction θ_k and on set b_l . So the projection operator sums up all pixels value which centers are intersected by the discrete projection line L . The restriction of angle θ_k leads both to a different sampling and to a different number of bins in each projection (p_k, q_k).

The direct MoT is depicted in Figure 1 for a 3x3 image and for the set of three directions $S = \{(-1, 1), (1, 1), (1, 0)\}$, giving the thirteen bins for real valued and unary image.

The MoT can be either performed by directly addition of the image gray values or by addition-modulus (to preserve the magnitude of the computed spectral coefficients). For binary images operation XOR may be used [6, 7].

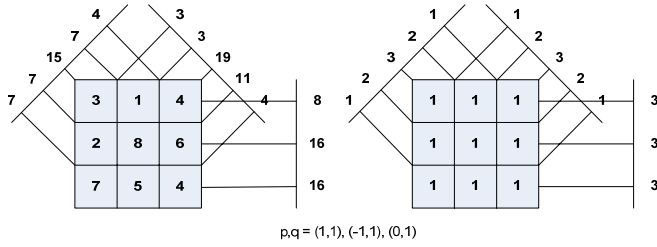


Figure 1. The direct Mojette transform for real valued (left) and unary (right) image

The inverse Mojette transform (IMoT) back-projects the bins of the different projections onto the reconstructed image. Two direct MoT modules are needed for the IMoT: a direct MoT of the reconstructed image F and a direct MoT for the unary image (Figure 1), in order to know for each bin, the number of crossed pixels. This implies that a single pixel-bin correspondence must be found in order to reconstruct a pixel value. Once this has been done, the value of the pixel is subtracted from each projection. The reconstruction is achieved where the total number of pixels has been treated. The first one to one correspondence of the image reconstruction algorithm is shown in Figure 2 for the MoT computed in Figure 1.

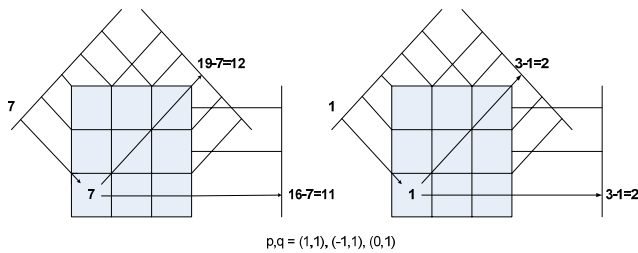


Figure 2. Inverse Mojette Transform: (left) reconstructed image, (right) correspondence finding in unary image

According to Katz's lemma [8], an image F of dimension $N \times M$ pixels can be reconstructed with the set of K projections $M = \{M_k\}$ if

$$N \leq P_K = \sum_{k=1}^K |p_k| \quad \text{or} \quad M \leq Q_K = \sum_{k=1}^K |q_k| \quad (5)$$

The order of complexity of the MoT is linear both in the number of projections and the number of pixels. In order to obtain the same complexity for the IMoT, a specific list of single pixel-bin correspondence has to be created and managed [6, 7].

The main characteristic of the MoT is the redundancy, which can be measured by the following parameter:

$$R = \frac{\sum bins}{\sum pixels} - 1 \quad (6)$$

Due to the ill-posed nature of the IMoT important degradation occurs when the bins have been slightly modified during the transfer (i.e. before the IMoT process start). The MoT is widely used for several image processing applications: data transmissions, image watermarking, cryptography and steganography [6, 7, 8].

3 “MoTIMoT” processor implementation

To construct a new hardware for a special task is difficult and expensive both in time and costs. Estimating the hardware and software needs to implement a MoTIMoT processing system there is necessary to map the tasks what such a system should implement. These tasks are not limited to the direct inverse Mojette transformation, but also posses embedded computer functions with or without (subject of the analyses) real time operating system kernel. The images are received thru an Ethernet connection and after processing are returned to the sender.

The MoT and IMoT functions are implemented as separate hardware coprocessors of the main processor. This is possible in two ways. The main processor can be an off-chip or an on-chip solution. The coprocessors are implemented in an ASIC chip or they are in Field Programmable Gate Array (FPGA). The implementation is motivated by this two solutions as in this way the algorithm is implemented in parallel processes and using relatively low frequencies (100-300MHz) we can obtain very high processing speeds.

The ASIC solution it is not usable for an academic research. While using FPGA we have the advantage of using embedded on-chip processors with on-chip memory and on chip system bus. For this reason and for the flexibility the Virtex FPGA chip will be used. Also for the design process the ML310 board will be used. This board is a Xilinx embedded system board and for details see [9].

Just to summarize the ML310 board hardware possibilities some characteristics are mentioned:

In addition to the Virtex-II Pro™ FPGA with the embedded PPC405, the ML310 board features the following:

- ATX form factor motherboard,
- 256 MB DDR DIMM,
- System ACE™ CF controller,
- 512 MB CompactFlash card,
- Onboard 10/100 Ethernet network interface card (NIC),
 - 4 PCI slots (3.3V and 5V),
 - LCD character display and cable,
 - FPGA serial port connection,
 - RS-232 mini-cable,
- Personality module interface for RocketIO MGT and LVDS access,
 - Standard JTAG connectivity,
 - ALi South Bridge Super I/O controller:

- 1 parallel and 2 serial ports,
 - 2 Universal Serial Bus (USB) ports,
 - 2 IDE connectors,
 - General purpose I/O (GPIO),
 - System Management Bus (SMBus) interface,
 - AC'97 audio codec,
 - PS/2 keyboard and mouse ports,
- ATX power supply.

One can conclude that the above characterised board is a good solution for the implementation of the described image processing system.

The System ACE™ CF controller is the power on configuration manager with multiple configuration possibilities. System ACE support multiple-bitstream management, FPGA microprocessor cores, and system reconfiguration/update over a network. They also provide an easily scalable and reusable configuration platform, a built in interface to a system processor, and the ability to centralize configuration for the entire system, simplifying debug and minimizing board space.

The main advantage using FPGAs is the flexibility of development tools, the hardware-software co-design solution, and hardware in the loop simulation. The development work is concentrated to implement the functional block schema of the planned embedded system instead of difficult and expensive PCB design. The proposed MoT/IMoT starts from the block schema given in [9]. Figure 3 shows the internal block schema of a MoT/IMoT computing hardware.

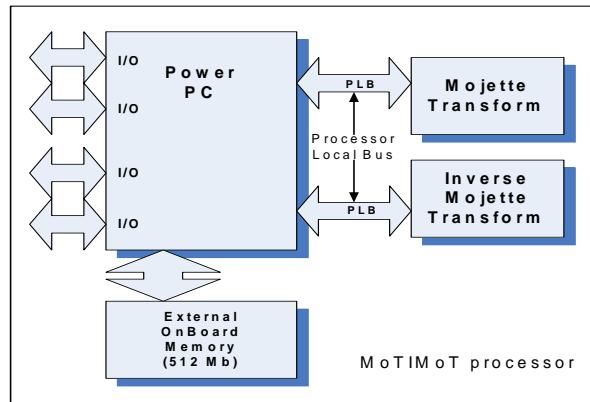


Figure 3. MoTIMoT Processor Block scheme

Figure 3. summarise only the main parts of the image processing system. This are: the PowerPC as the main processing unit, the MoT unit and the IMoT unit. Both Mot and IMoT processors are connected to the PowerPC via the internal PLB bus.

The image processing algorithms are memory costly therefore the onboard external memory is very useful.

The PowerPC is the embedded hard core of the Virtex II chip and provides the tasks which are needed by the system. If it will be used an operating system kernel (such as Linux), then this will be ported to the PowerPC. Also the main processor will supply the data the MoT and IMoT units by transferring the 64x64 pixel size images from external memory to the coprocessors.

3.1 MoTIMoT Block Scheme

Figure 4 shows the block scheme of the MoT computing unit. The memory unit with the size of 64x64 byte enables to decrease the I/O operation between the onboard external memory and the MoT-IMoT processor. This size of memory not large enough for a 256x256 pixel size image, of course. A part of the image, a 64x64 pixel size window will be transformed, so we have to make the transform sixteen times.

An advantage of this method can be the smaller damage in the picture when the bins get crashed. When we are working with the 256x256 pixel size image and a bin get crashed, the reconstructed image contain only distortions in the columns of the bin and in the close neighbourhood. Using only a 64x64 pixel size part of the image, the vertical size of the damaged bin area will be decreased to 25%.

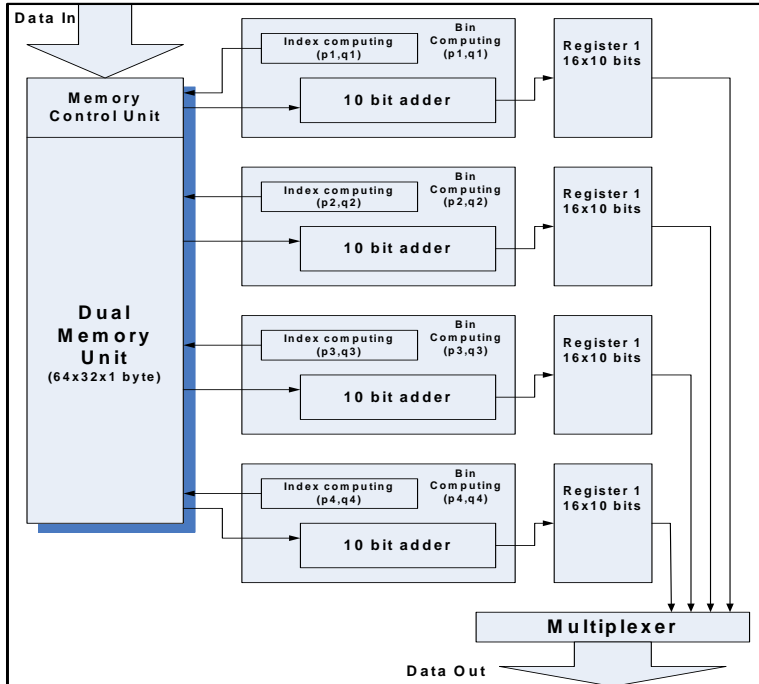


Figure 4. MoT Computing Unit

The index computing units has a starting address (increased by a counter) and computes the memory address of the corresponding pixels. After that the pixel values are summed up by the 10-bit adder, the output result the bin (ten bits long) is stored in a 16x10 bits size register and when the registers are full, the bins are sent to the external memory for storage.

4 CONCLUSIONS

The paper outlined the hardware requirements for the implementation of Mojette direct and inverse transformation in the embedded system using FPGA.

ACKNOWLEDGMENT

The authors gratefully acknowledge the donations of Xilinx Inc. and Celoxica Inc. which made possible the start of this research.

The research is part of the Hungarian-Research Project GVOP 3.1.1-2004-05-0333/3.0.

REFERENCES

- [1] Major, R. – Deac, V. – Borda, M.: An application of the Hash Functions in Digital Watermarking, *Proc. Inf. Workshop, Trend and Recent Achievements in Inf. Technology*, Cluj – Napoca, Romania, 2002.
- [2] Hartung, F. – Kutter, M.: Multimedia Watermarking Techniques, *Proc. IEEE*, vol. 87, No7, 1999.
- [3] Turán, J.: Fast Translation Invariant Transforms and Their Applications. *Elfa*, Kosice, 1999.
- [4] Normand, N. – Guedon, J. P.: La transformee Mojette: une representation recordante pour l'image, *Comptes Rendus Academie des sciences de Paris*, theoretical comp. sci. section, 1998, 124 – 127.
- [5] Guedon, J. P. – Parrein, B. – Normand, N.: Internet Distributed Image Databases. *Int. Comp. Aided Eng.*, Vol. 8, 2001, 205 – 214.
- [6] Katz, M.: Questions of uniqueness and resolution in reconstruction from projections. *Springer Verlag*, Berlin, 1977.
- [7] Autrusseau, F. – Guedon, J. P.: Image Watermarking for Copyright Protection and Data Hiding via the Mojette Transform, *Proc SPIE*, VOL. 4675, 2002, 378 – 386.
- [8] Turán, J. – Ovsenik, L. – Benca, M. – Turán, J. Jr: Implementation of CT and IHT Processors for invariant Object Recognition System, *Radioengineering*, Vol. 13, No 4 2004. dec. page 65-71
- [9] Xilinx, ML310 User Guide, pp73, <http://xilinx.com>

Reviewer: prof. Dr. RNDr. Lubomír Smutný, VŠB – Technical University of Ostrava