# Color Mathematical Morphology In A FPGA

Emerson Carlos Pedrino[1], José Hiroki Saito[1,2], Hermes Senger[1]
[1]Department of Computer Science
Federal University of São Carlos, UFSCar
Sao Carlos, SP, Brazil
[2]Faculdade Campo Limpo Paulista, FACCAMP
Campo Limpo Paulista, SP, Brazil
{emerson, saito, hermes}@dc.ufscar.br

Valentin Obac Roda
Department of Electrical Engineering
University of São Paulo
Sao Carlos, SP, Brazil
valentin@sel.eesc.usp.br

*Abstract* — In this paper we present the development of a hardware algorithm for fast morphological processing of color images using a FPGA. The developed system forms a complete lattice environment for color mathematical morphology on the RGB color space. This color space is derived from a digitalization process of a video signal supplied by a CCD camera. The basic operators of mathematical morphology (dilation and erosion) are implemented. The processed images can be shown by a RGB monitor. Examples of application show the good performance of the system compared with an implementation by means of a conventional processor.

***Keywords- PLDs; color mathematical morphology.***

## I. INTRODUCTION

Morphological image processing is a nonlinear branch in image processing developed by Matheron and Serra in the 1960´s, based on geometry and the mathematical theory of order [1-2]. Morphological image processing has proved to be a powerful tool for binary and grayscale image computer vision processing tasks, such as edge detection, noise suppression, skeletonization, segmentation, pattern recognition and enhancement. The basic operations in mathematical morphology are the dilation and the erosion, and these operations can be described by logical and arithmetic operators. Dilation and erosion morphological operators can be represented respectively by the sum and subtraction of Minkowski sets [1]:

$$A \oplus B = \cup\{B + a \,|a \in A\}, \qquad (1)$$

and

$$A\Theta - B = \bigcap\{A + b \,|b \in B\}, \qquad (2)$$

In Eq. (1), A is the original binary image, B is the structuring element of the morphological operation and B+a is the B displacement by a. Therefore, the dilation operation is obtained by the union of all B displacements in relation to the valid A elements. In equation 2, -B is the $180^o$ rotation of B in relation to its origin. Therefore, the erosion operation corresponds to intersection of the A displacements by the valid points of -B.

In color images, pixels are represented by vector values:

$$P(x, y) = [P1(x, y), P2(x, y), P3(x, y)]^T, \qquad (3)$$

The application of mathematical morphology to color images is difficult due to the vector nature of the color data. The extension of concepts from binary and grayscale morphology to color morphology must first choose an appropriate color ordering, a color space that determines the way in which colors are represented and an infimum and a supremum operator in the selected color space should also be defined. There are several techniques for ordering vectors. The two main approaches are marginal ordering and vector ordering. In the marginal ordering, each component P1, P2 or P3 is ordered independently and the operations are applied to each channel; unfortunately, this procedure has some drawbacks, e.g., producing new colors that are not contained in the original image and may be unacceptable in applications that use color for object recognition. The vector ordering method for morphological processing is more advisable. Only one processing over the three dimensional data is performed using this method. There are several ways of establishing the order, e.g., ordering by one component, canonical ordering, ordering by distance and lexicographical order [3]. Once these orders are defined, then the morphological operators are defined in the standard way. The vector erosion of color image f at pixel x by the structuring element B of size n is:

$$EnB(f)(x) = \{\inf[f(z)], z \in n(B(x)\}, \qquad (4)$$

The corresponding dilation DnB is obtained by replacing the inf by sup:

$$DnB(f)(x) = \{\sup[f(z)], z \in n(B(x)\}, \qquad (5)$$

An opening is an erosion followed by a dilation, and a closing is a dilation followed by an erosion.

Hardware solutions for morphological image processing have been proposed in the literature to reduce the processing time per pixel [4-6], but only some works have been used dedicated architectures to handle color images for morphological image processing. The advantage and power of these solutions come from parallel implementation on multiple processing modules. Using a FPGA, the number of processing elements can be made as large as the number of image pixels.

Thus, in this paper we propose the development of an architecture for fast morphological processing of color images using FPGAs. The system forms a complete lattice environment for color mathematical morphology on the RGB color space. This color space is derived from a digitalization process of a video signal supplied by a CCD camera. The pixels of image are supplied on a raster line by line. The results can be shown by a RGB monitor. We have implemented the basic operators of mathematical morphology (dilation and erosion) extended to color images in a FPGA unit. Examples of applications are discussed and the results obtained are compared with a software implementation.

This article is organized as a brief review of the basic concepts of morphological operations, section 1; a detailed description of the developed system, section 2; results and application examples are presented in section 3; section 4 presents the conclusions.

## II. PROPOSED HARDWARE

The block diagram of the developed architecture can be seen in Fig. 1. The project files containing the description of the architecture are transferred to the FPGA board by means of the USB interface from a host computer. In this project the DE2 board from Altera has been used to develop the video architecture. The DE2 board contains a Cyclone II (2C35) FPGA, a NTSC/PAL TV decoder circuit and a VGA output circuit. A composite video supplied by a commercial video camera is de-interlaced and converted to 10 bit RGB data (640x480 pixels) through a video decoder stage. The RGB frames are processed through pipeline stages of the color architecture (FPGA) and the results are converted to an analog format again through a DA converter. Then, the processed images can be shown in a VGA monitor. A 27MHz oscillator was used as a clock source.

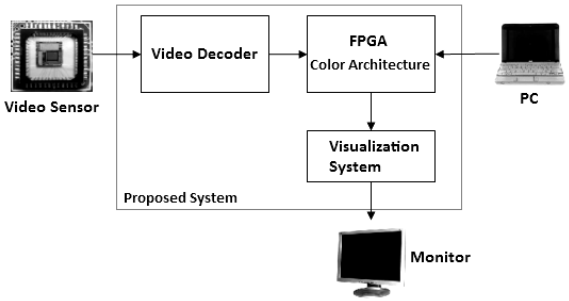


Fig. 1. Proposed Hardware for Color Mathematical Morphology.

The video decoder solution is based on the TV box demonstration example supplied by Altera on the DE2 CD-ROM. In Fig. 2 it is shown a block diagram of the video decoder. The itu_656_decoder block extracts YCrCb (4:4:4) video signals from the 4:2:2 data source sent from the TV Decoder. It also generates a 13.5 MHz pixel clock (YPixel Clock) with blanking signals indicating the valid period of data output. The video signal from the TV Decoder is interlaced, thus, it's necessary to perform de-interlacing on the data source. For this task it's used the Dual Port Line Buffer block and Hsyncx2 block to perform the de-interlacing operation where the pixel clock is changed to 27 MHz from 13.5 MHz and the Hsync is changed to 31.4 kHz from 15.7 kHz. Finally, the YCrCb2RGB block converts the YCrCbx2 data into RGB output. The VGA Timing Generator block generates standard VGA sync signals VGA_HS and VGA_VS. The block diagram of pipeline stages implemented in FPGA can be seen in Fig. 3. The instructions of each stage are loaded into the stages through a state machine named ROM that contains the original program.

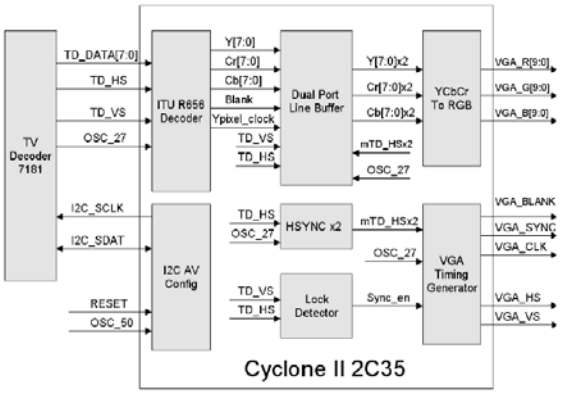


Fig. 2. Video decoder stage.

The state machine ROM uses the bus dat and the bus add to distribute the data (instructions) to each processor that uses an add (address) in the architecture. For example, the P1 has the add=01h, the P2 has the add=02h and so on. The "program.mif" contains a binary chain representing the

instructions (opcodes) of each stage. In Fig. 4 it is shown a simulation of this unit. After the reset process of the architecture, while the end_add pin is low, the state machine loads the instructions referring to a certain problem into the instruction register (IR) of each processor of the pipeline. When the load process ends, the end_add pin will be high and the state machine will indicate the time of video processing and this cycle will be repeated when the state machine reads a reset state again.

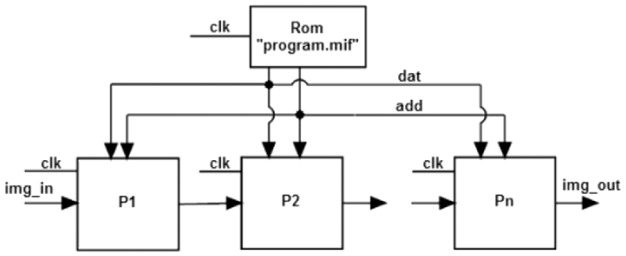


Fig. 3. Pipeline stages from developed archictecture.

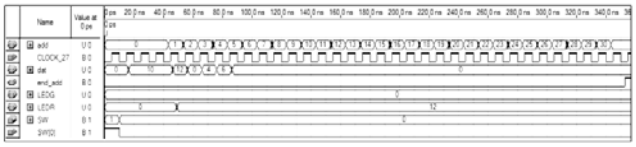

Fig. 4. Simulation from ROM unit.

Each stage stores two adjacent 640-pixel lines followed by a 3-pixel line to constitute a (3x3)-input to a morphological processor implemented in that stage. This same structure is used by a previous stored result that is delayed by each stage, too. This result is stored in img_temp register. Each stage has been built using the customized megafunctions available on the Quartus II software from Altera and the Verilog HDL. In Figure 5, p1 is connected to p1_1, p2 in p1_2, p3 in p1_3, p4 in p2_1, and so on. Fig. 5 shows the block diagram of a stage. The Instr_dec block in the processor decodes the instruction stored in IR register and apply an morphological operation to input pixels p1..p9 and/or s2_2 (previous stored result from n-1 stage). In Instr_dec block, the dilation and erosion operations are implemented according equations 5 and 4, respectively. Each stage is capable to work with a RGB digital image of 10 bit/channel. For color processing is used a combination of R, G and B to form a lattice structure required for morphological processing. This combination is as follows: Pn={R1G1B1,…,R10G10B10}, where n=1..9, thus, Pn can be a 30-bit scalar number, but, to simplify the hardware, in this work was used only the three most significant bits of each channel, forming a 9-bit scalar number. Thus, the morphological operations (equations 4 and 5) can be defined for color images. After processing, the resulting scalar is decomposed again into its RGB component. The *sto_out* result from one processor can be used in conjunction with the input pixels of other processor from the architecture to form more complex operations.

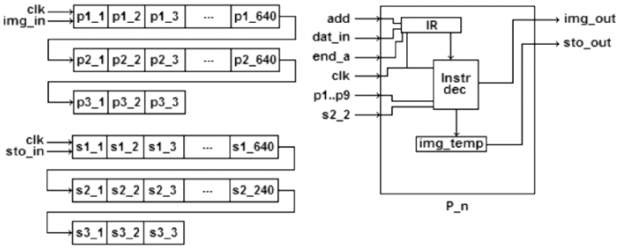


Fig. 5. Hardware implementation of one stage.

The instructions from each stage have been implemented in Verilog HDL. Each instruction is processed in one clock cycle. It's possible to add more instructions to this process, for example, an arithmetic instruction can be added.

The visualization system is based on the 10-bit VGA DAC ADV7123 chip from Analog Devices present on the DE2 board. The block diagram of this stage can be seen in Fig. 6. The processed images from the architecture are sent to this chip with the sync signals to enable the display on a VGA monitor.

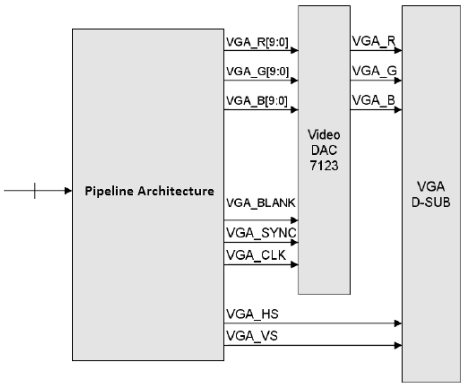


Fig. 6. Visualization system from the architecture.

## III. RESULTS

In this section is presented some results obtained through the developed architecture. The resolution of the images was 640x480 pixels (RGB). In Fig. 7 it is shown the original image to be processed captured by the system. The result of a dilation operation by a 9x9 square structuring element is shown in Fig. 8. A result of erosion by a 9x9 square structuring element can be seen in Fig. 9. Using these ideas, it is possible to construct color morphological filters too. The procedures presented here were implemented in Matlab in a PC computer equipped with an AMD Athlon 64 Processor 3200+ and 1.25 GB of RAM memory. The comparison results are presented in Table 1. In Table 1, it's shown the time to process a frame of the image in each case. In tests were used dilations by squares

structuring elements (SE). The processing times refer to a frame of 640x480 pixels. It's very clear to us that morphological color processing demand too much processing time. Using a dedicated hardware, it's possible to accelerate the color morphological processing for real time applications. Table 2 shows the summary of the used FPGA device.

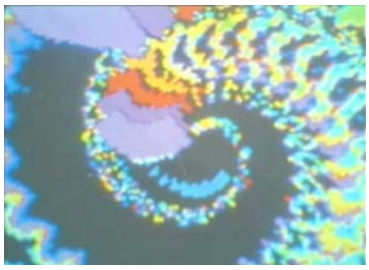

Fig. 7. Original image captured by the system.

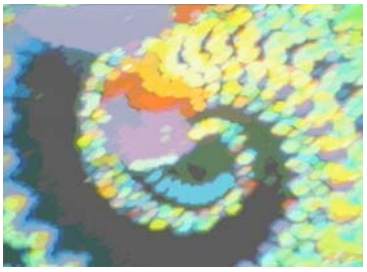

Fig. 8. Obtained result of image from Fig. 7 dilated by a 9x9 square structuring element through the developed hardware.

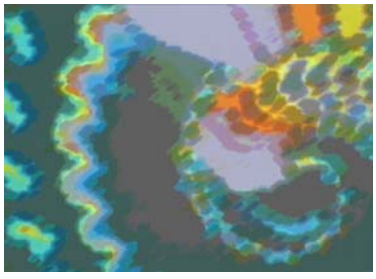

Fig. 9. Erosion result by a 9x9 square structuring element from a version of the original image.

## IV. CONCLUSION

In this paper we have presented the development of a system for fast morphological processing of color images using FPGAs that forms a complete lattice environment for color mathematical morphology on the RGB color space. The system is capable of processing 60 frames/s at VGA resolution. The basic operators of mathematical morphology (dilation and erosion) were implemented. Application examples show the good performance of the system when implemented in a FPGA unit.

Table 1. Comparison results of the developed hardware versus the Matlab implementation.

| Color Morphological Operation | Developed Hardware t(ms) | Matlab t(s) |
|---|---|---|
| SE (3x3) | 16.70 | 5.9 |
| SE (5x5) | 16.76 | 11.4 |
| SE (7x7) | 16.80 | 17.0 |
| SE (9x9) | 16.86 | 23.0 |

Table 2. Compilation report.

| Device (Family: Cyclone II) | Logic Elements | Memory Bits | Embedded Multiplier 9-bit Elements |
|---|---|---|---|
| EP2C35F672C6 | 4,810 (14%) | 233,032 (48%) | 18 (26%) |

## ACKNOWLEDGEMENTS

Hermes Senger thanks FAPESP(process 2009/09847-0) and CNPQ (process number 201812/2009-6) for their support.

## REFERENCES

[1] E. R. Dougherty, An introduction to Morphological Image Processing, SPIE, Bellingham, Washington, 1992.

[2] J. Serra, Image Analysis and Mathematical Morphology, Academic Press Inc, California, 1982.

[3] F. Ortiz, F. Torres, E. De Juan, N. Cuenca, "Colour Mathematical Morphology for Neural Image Analysis," Real Time Imaging 8, 455-465 (2002).

[4] E. C. Pedrino; V. O. Roda, "Real time morphological pipeline architecture using high capacity programmable logic devices," Journal of Electronic Imaging (Springfield), vol. 14, pp. 023002-023009, 2007.

[5] F. Cappabianco et al., "A general image processing architecture for FPGA," In: V SPL, 2009, São Carlos, SP, Brazil. EESC, Universidade de São Paulo (2009).

[6] I. Jivet, "Architectures for morphological decomposition by reconstruction in real time," In: 9th International symposium on mathematical morphology, 49-52 (2009), Groningen, Netherlands (2009).