

# Image Processing Implementation on FPGA Hardware with Mathematical Morphological Operators Centered in VHDL

J. Carmona-Suárez<sup>1</sup>, I. Santillan<sup>2</sup>, S. Martínez-Díaz<sup>3</sup>, J. L. Gómez-Torres<sup>4</sup>, J.A.Sandoval-Galarza<sup>5</sup>

Instituto Tecnológico de La Paz  
División de Estudios de Posgrado e Investigación  
La Paz, B.C.S., México.

<sup>1</sup>[jdcarmona@itlp.edu.mx](mailto:jdcarmona@itlp.edu.mx), <sup>2</sup>[israel.santillan@itlp.edu.mx](mailto:israel.santillan@itlp.edu.mx), <sup>3</sup>[smdiaz06@hotmail.com](mailto:smdiaz06@hotmail.com), <sup>4</sup>[jgomezt@itlp.edu.mx](mailto:jgomezt@itlp.edu.mx),  
<sup>5</sup>[jsandoval@itlp.edu.mx](mailto:jsandoval@itlp.edu.mx)

---

**Abstract**—An image processing system implemented on a low range FPGA device is presented in this paper. The system is designed in VHDL. Mathematical morphological operators are implemented as a parallel process. An operator unit proposed is presented. The data flow management is detailed since the information in a row is processed at the same time. Also, an application example is presented. The system was applied to remove impulsive noise and to detect the gradient of an image. Furthermore, basic mathematical morphology concepts are illustrated.

**Keywords:** Image Processing; Mathematical Morphology; FPGA; VHDL; Parallel Processing.

---

## I. INTRODUCTION

The main role of the image processing is to obtain information, transform, or enhance a digital image; to recover, stand out, segment, improve, merge, remove or recognize objects, patterns, or characteristics contained in that digital image. Many approaches have been proposed but Mathematical Morphology (MM) is widely used and studied because solves several image processing issues. In this work the basic MM operands are implemented on a FPGA to remove artificially added impulsive noise, to find objects' contour via the gradient in a real image, to find the 2D spatial location of an object in an under controlled conditions image. These operands are the beginning to solve more complex problems.

On the other hand, the FPGA is a customizable system best suitable to implement intricate issues in the signal and image processing. The complexity of some approaches is reduced, and the processing speed can greatly be increased by the possibility of parallel processing. VHDL lets configure or 'programming' the FPGA with the advantage that is a reputable IEEE standard. Also, it allows flexibly to determine specific hardware units to be used in the FPGA. Moreover, the technology is producer independent, so it can be used in a capable FPGA device from any manufacturer.

In this work, the advances in image processing for an autonomous system is presented. The first efforts concentrate in a filter to clean an image from noise, second to get the contour of objects in an image as initiated segmentation, and last to a 2D spatial location of an object. The trend is to have a reduced size and energy consumption, fast, optimal and efficient units, also an easy and low-cost manufacturing. The cameras and the necessity to process the information from digital images, in this case, is extended widely. A camera is now an efficient substitution of many sensors but increases the

complexity of processing. Many autonomous systems currently have a camera as a main sensor for pattern recognition of specific objects to process real time video images.

In this work some previous works are reviewed in Section II. Basic concepts in Mathematical Morphology are checked in Section III. The image processing implementation in a FPGA is described in Section IV. The results are presented in Section V. Finally, in Section VI are the conclusions.

## II. PREVIOUS WORK

The implementation of algorithms for image processing on FPGAs with VHDL, is still seldom, also less involving the Mathematical Morphology approach. Nevertheless, almost all the works concentrates their efforts in the control of data flow, increase the complexity of the processing or accelerate it. In [1] the authors present a VHDL implementation for a median filter that process the pixel information while the data is received for the system. They optimized the processing time by starting the process before received all the information. The implementation in [2] are parallelize low-level operators. The data flow is managed between an external memory and a dedicated processor array. This array has the 'Processing Elements (PE)' organized in a 2-D systolic approach. These PE's can perform the multiplication, addition, subtraction, accumulation, maximum, minimum, and absolute value. A pipeline approach is presented in the continued work [3], [4] and [5]. They used morphological operators mostly to alternate sequential filters that need many iterations of processing. Also, they use process units in parallel manner but limited to few units. The method of management the data flow in [6] is done by a FIFO-to-USB module and a RAM-based shift register. They implement with VHDL a Gauss and mean filters and an edge detector. In this work the VHDL is the main tool for the

---

This work was possible due the support of PRODEP and TecNM/ITLP

implementation. The parallel process of the morphological operators and morphological filters increases the speed. The data flow is managed in a PC, a camera module or an external memory. Currently is presented a parallel morphological process implemented in a FPGA using VHDL solely.

### III. BASIC CONCEPTS OF MATHEMATICAL MORPHOLOGY

Mathematical Morphology (MM) is an image processing branch highly supported by many mathematical concepts as set theory, topology algebra, and others. Morphological opening  $\gamma_{\mu B}$  and morphological closing  $\varphi_{\mu B}$  are the basic morphological filters in the MM theory, with a given structuring element, in this work represented by  $B$  as a  $3 \times 3$  pixels that contains its origin.  $\check{B}$  is the transposed set ( $\check{B} = \{-x: x \in B\}$ ) and  $\mu$  is named an homothetic parameter. The follow equations represent respectively the morphological operation named opening and closing:

$$\gamma_{\mu B}(f)(x) = \delta_{\mu \check{B}}(\varepsilon_{\mu B}(f))(x)$$

$$\varphi_{\mu B}(f)(x) = \varepsilon_{\mu \check{B}}(\delta_{\mu B}(f))(x)$$

where  $\varepsilon_{\mu B}$  is the morphological erosion and  $\delta_{\mu B}$  is the morphological dilation, expressed as  $\varepsilon_{\mu B}(f)(x) = \wedge \{f(y) : y \in \mu \check{B}_x\}$  and  $\delta_{\mu B}(f)(x) = \vee \{f(y) : y \in \mu B_x\}$ .  $\wedge$  is the inf operator and  $\vee$  is the sup operator. These basic morphological operators are illustrated in Figure 1. The increase property is one of the most important in MM, it means that for a pair of functions  $f$  and  $g$ , with  $f \leq g \Rightarrow T(f) \leq T(g)$ ,  $T$  is an increasing transformation that preserves the order. Other important property in MM is the idempotent notion, a transformation  $T$  is idempotent if and only if  $T(T(f)) = T(f)$ . The morphological opening and morphological closing meet the conditions to be increasing and idempotent. In the MM, a morphological filter is one that is increasing and idempotent.

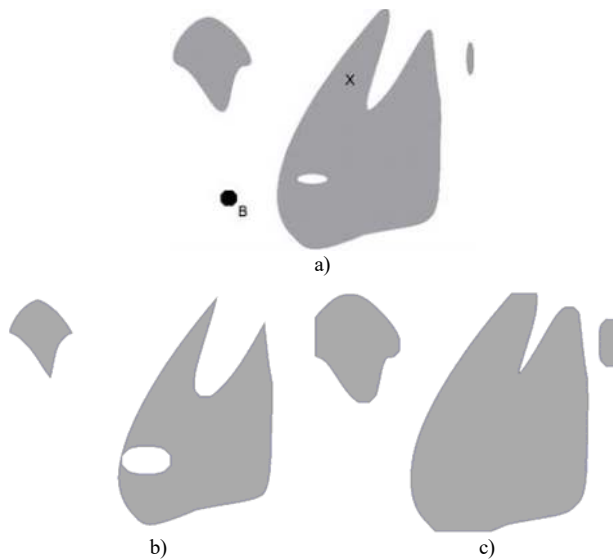


Fig. 1. Basic morphological operands in binary case. a) Original image  $X$ . b) Morphological erosion by structuring element  $B$ . c) Morphological dilation by structuring element  $B$ .

Binary functions case was the first proposal for the morphological opening and morphological closing [7], later they were extended to the numeric functions case [8] and flat graphs [9].

The morphological opening and morphological closing allow to remove components according to a predefined criterion. The increasing size of the structuring element by the homothetic parameter is the criteria in these morphological filters. Geometrically, its behavior is interpreted for the binary case (numeric case) as follows: the area swept by the structuring element within the set  $X$  (or inside of the mask  $f$ ). Morphological opening is illustrated in Figure 2 in a binary case and continuous case.

Salt and pepper noise is an impulsive noise that can be removed by the morphological filters. The sequential

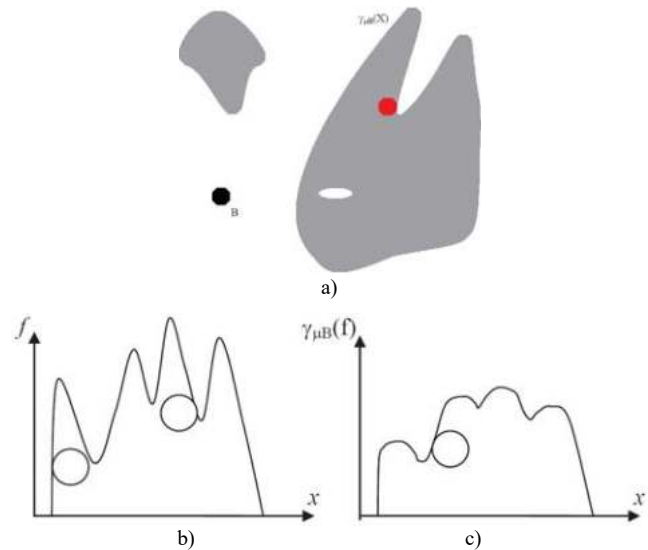


Fig. 2. Morphological opening. a) the opening  $\gamma_{\mu B}(X)$  represents the area swept by the structuring element  $B$  inside the set  $X$ , in Fig 1.a is represented the original image. b) Original image  $f$ . c) The opening  $\gamma_{\mu B}(f)$  represents the area swept by the structuring element  $B$  inside the mask  $f$ .

application of the morphological filters works to remove this type of noise without removing the contours and without an extremely modification of the original image. On the other hand, with the basic morphological operators it is possible to obtain valuable information about the image, i.e. the slope or gradient between different regions represented in the image and it is typically the first step to obtain the edge detection of objects and regions in the image to segmenting it.

### IV. IMPLEMENTATION

The main modules of the system are indicated in the block diagram present in Figure 3. The UART unit ensures the acquisition of the image information, it can be replaced with other unit that acquire the information in other way such as a camera daughter board. The register bank module manages the information to be processed and the data results. The processing module process the data in parallel manner. And the control unit ensures that every module works in synchrony. The sequence of the data management and the processing is referred in the finite state machine diagram represented in the Figure 4.

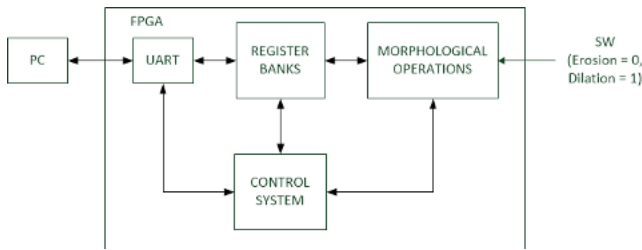


Fig. 3. Block diagram with the main modules.

A communication module named UART is the system that controls the pixel information data flow to be processed. The information is coded as byte type data for gray level images between 0 and 255. Every line to be processed of the image needs 3 storage lines in registers banks, in the manner that when the line 1 is processed the lines 0 and 2 are necessary. After the UART module receives the first byte, this is stored into the last register of the line 2 register bank. When the next byte arrives, the register bank shifts horizontally the data to the next register in the same line 2. When the whole line 2 is filled, the data shifts vertically to the line 1 register bank and at the same time the line 1 shift vertically to line 0 register bank. This sequence is done until the first three lines are received and stored in the line register banks module. This sequence is illustrated in the Figure 5. Now, the implied 9 bytes of the structural element are processed in the process module. The result is stored in an auxiliary register bank. When this bank is filled, the data information is send back through the UART module.

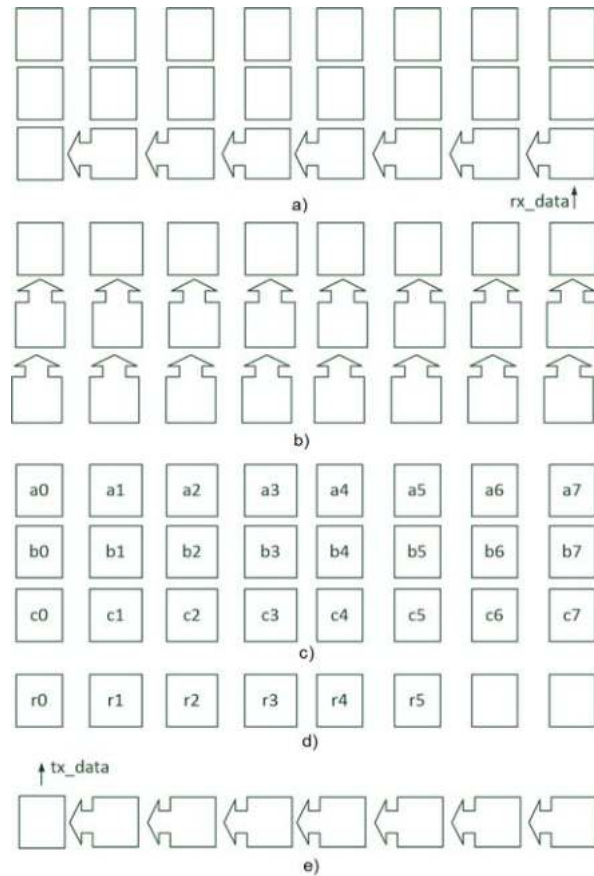


Fig. 5. Data Flow. a) Received data and horizontal shift in line 2, b) double shift vertically, c) register bank ready for parallel processing, d) auxiliary register bank with the results, e) horizontal shift to transmit the results.

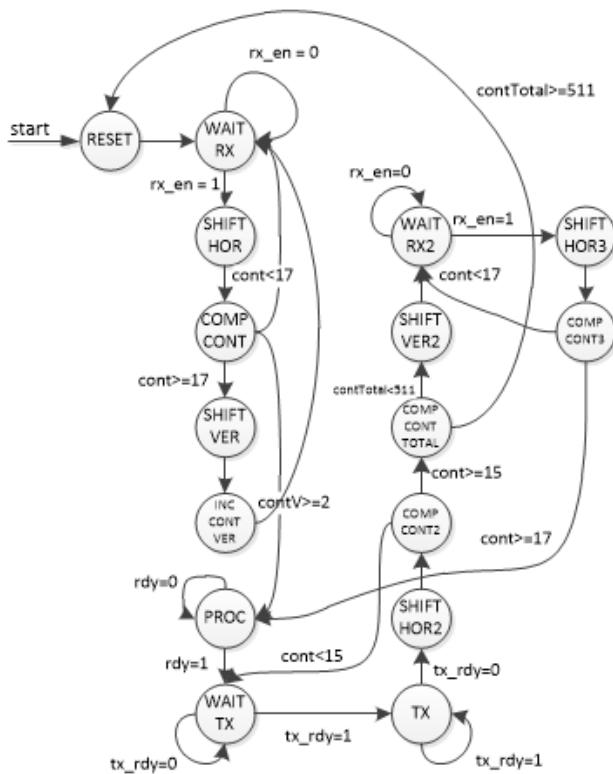


Fig. 4. Finite State Machine diagram demonstrates the sequence of data.

The process module is illustrated in a block diagram showed in Figure 6. According to the structuring element, the central pixel must be operated with its eight neighbor pixels, San Luis Potosí, San Luis Potosí, México, 10-12 de Octubre de 2018

so the eight neighbor pixels are presented in the mux1 waiting to be operated with the central pixel, selected by the mux0. At the first clock rising edge, the central pixel is operated with the first neighbor pixel. At the second clock rising edge, the result from this operation selected by the mux0, is operated with the second neighbor pixel and so on with all the eight neighbor pixels. At the eighth clock rising edge, the morphological erosion or dilation has been done for the current central pixel. The same proceeding must be done for all the pixels in the image. In this implementation the whole line is parallel processed, that means that there are the same process units as pixels in a line. The operation can be selected between the morphological erosion and dilation. The implication that a whole line is processed at once is the reason why the registers are managed by lines.

When the first line is processed its information and the information of line 2 will be used for the processing of the second line, so a dual shift its done vertically to pass line 1 to the line 0 and to pass line 2 to line 1. The information in line 0 register bank is no longer necessary. The system is ready to receive the new line in the line 2 register bank to process the second line of the image. This sequence is repeated until all the lines of the original image are being processed. In the Figure 7.a the Lena image is shown as an original image. The FPGA system performs the following operations: erosion with

homothetic size  $\mu=1$  is shown in Figure 7.b, and dilation with the same homothetic size is shown in Figure 7.c

Previously the image is send for processing, there is an especial preparation for it. A contour is added to the original

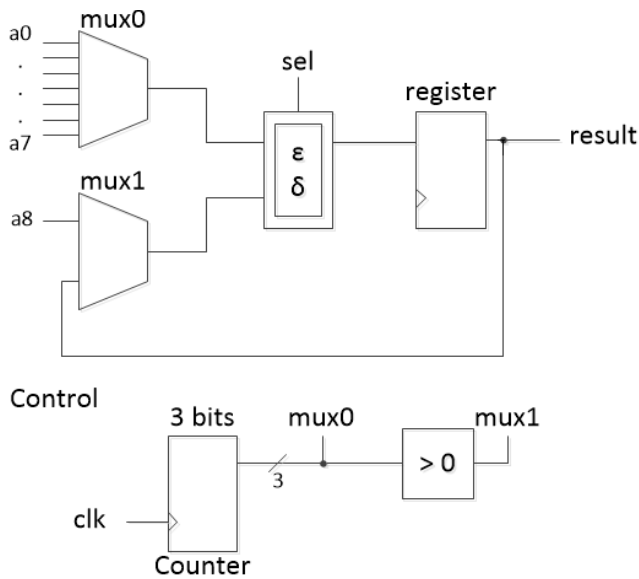


Fig. 6. Block diagram with the main modules.

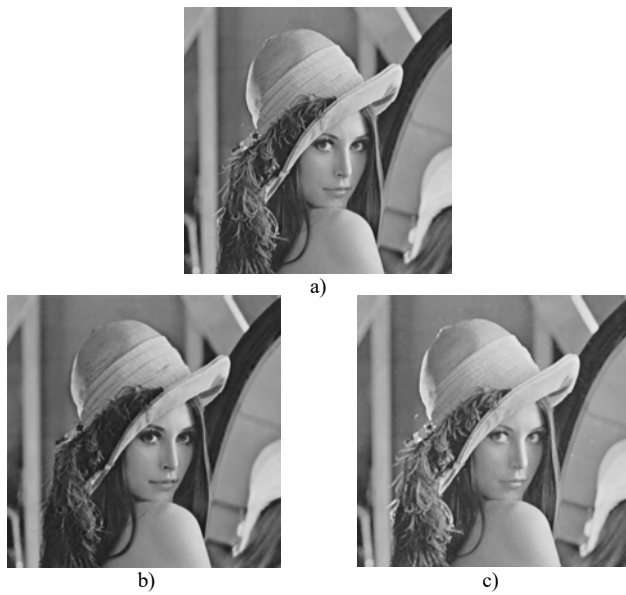


Fig. 7. a) Original image. b) erosion. c) dilation.

image to process all the pixels on it. The values for the pixels added in the contour are taken from the contiguous edge of the image itself. In the contour corners the value is taken from the same image corners, in the Table I the shadowed pixels represent the artificially added contour showing the row and column from them values are taken.

## V. RESULTS

The system proposed was implemented on an evaluation board from Digilent, the Atlys with a Xilinx FPGA Spartan-6 XC6SLX45. The image information is transmitted from a PC via USB by an UART protocol at 2'083,333 bps. The principal operations in the process module are the morphological erosion and the morphological dilation. With these operations

the morphological opening and the morphological closing are obtained. The morphological opening and closing are shown in Figure 8.a y 8.b respectively from the original shown in Figure 7.a. The homothetic parameter for these last operations is  $\mu = 1$ . These basic morphological filters are employed as a noise remover. In this case, an original image was contaminated with salt & pepper impulsive noise with a 5% distribution. The 'imnoise' function from MATLAB software was used. The result from noise removal by the FPGA system is showed in Figure 9.b. The original image is the same Lena in Figure 7.a. The contaminated image is shown in Figure 9.a. The operation to complete the noise removal a morphological opening with homothetic size 1 followed by a morphological closing with the same homothetic size.



Fig. 8. Morphological filters. a) Opening. b) Closing.

TABLE I. CONTOUR ADDED

0,0	0,0	0,1	0,2	0,3	0,3
0,0	0,0	0,1	0,2	0,3	0,3
1,0	1,0	1,1	1,2	1,3	1,3
2,0	2,0	2,1	2,2	2,3	2,3
3,0	3,0	3,1	3,2	3,3	3,3
3,0	3,0	3,1	3,2	3,3	3,3

Finally, in Figure 10 is illustrated the internal gradient obtained by the arithmetic subtraction of the morphological erosion from the original image. In this example the erosion is done with the homothetic parameter  $\mu=1$ .



Fig. 9. Impulse noise removal. a) Salt & pepper contaminated image. b) Image with noise removed.

## VI. CONCLUSIONS

The system implements the morphological filters as expected using a parallel processing. It was capable to eliminate the impulsive noise of salt & pepper. Also, the system presents the same behavior in the generation of the gradient. Because these applications are well known it is not

necessary to demonstrate the results but a comparison with MATLAB's software functions shows the same result. In this implementation the system flexibility allows the simple configuration to process different size of images.

The VHDL development process may increase the time to finish the system, but it allows a development control and the flexibility to modify and update the system without other technological needs. As well, we can have the knowledge of how the entire system works. When other technologies were applied, the system would perform like a black box.

During the development, problems in the sequential system were found involving external systems. The simulations in the development software cannot imitate the behavior of the external systems and the asynchronous control signals, and the analysis needs to be in a wider time interval than the one available by the software used (a web accessible development software). Those problems can be detected but not fixed unless the use of other tools like logic analyzer or specialize software (but not web accessible) like ChipScope or SignalTap, since it is a low level digital system what it is being developed.

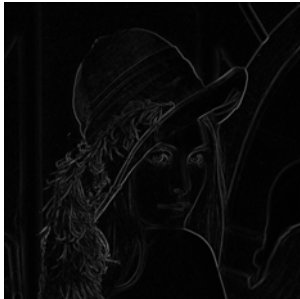


Fig. 10. Gradient.

To parallelize the image processing there is need to implement different amounts of morphological operator modules, if we increase this quantity, the compilation process of the system becomes more complicated and requires several amounts of time to finish, since the compiler tries to synthesize and implement a system on a FPGA limited in resources. This tends to slow the development process and affect the optimization time. The fundamental part of this work is the

data processing with the morphological operators, nonetheless, better systems of communication can be applied to improve the acquisition and management of the data.

The system presented here is an advance for an autonomous system that processes real time video images. It is necessary to add a camera module and the necessary blocks to manage the images information in an external memory. The used resources and the processing time will be analyzed for optimization.

#### ACKNOWLEDGMENT

I. Santillan want to thank L, L & D for the great encouragement and strengthening.

J. Carmona-Suárez is grateful with the ITLP.

#### REFERENCES

- [1] K. J. Palaniswamy, M. E. Rizkalla, A. C. Sinha, M. El-Sharkawy and P. Salama, "VHDL Implementation of 2-D median filter", 42nd Midwest Symposium on Circuits and Systems, vol. 2, pp. 744 – 747, Aug 1999.
- [2] G. Saldaña-González and M. Arias-Estrada, "FPGA Based Acceleration for Image Processing Applications", Image Processing, Yung-Sheng Chen, Ed. InTech, Chapter 26, Dec 2009.
- [3] J. Bartovsky, E. Dokladalova, P. Dokladal, V. Georgiev, "Pipeline Architecture for Compound Morphological Operators", IEEE International Conference on Image Processing, pp. 3765 – 3768, Sept 2010.
- [4] J. Bartovsky, M. Holik, V. Kraus, A. Krutina, R. Salom, V. Georgiev, "Overview of Recent Advances in Hardware Implementation of Mathematical Morphology", 20th Telecommunications Forum, pp. 642 – 645, Nov 2012.
- [5] J. Bartovský, P. Dokládál, E. Dokládálová, V. Georgiev, "Parallel implementation of sequential morphological filters", Journal of Real-Time Image Processing, vol. 9, iss. 2, pp. 315-327, June 2014.
- [6] J.V. Vourvoulakis, Xanthi, J. Lygouras, J.A. Kalomirois, "Acceleration of Image Processing Algorithms Using Minimal Resources of Custom Reconfigurable Hardware", 16th Panhellenic Conference on Informatics, pp. 68 – 73, Oct 2012.
- [7] J. Serra, "Image Analysis and Mathematical Morphology, Vol. II, Theoretical advances", Academic Press, New York, 1988.
- [8] P. Soille, "Morphological Image Analysis: Principles and Applications", Springer-Verlag, Heidelberg Berlin, 2nd edition, 2003.
- [9] G. Matheron, "Eléments pour une Théorie des Milieux", Masson, Paris, 1967.