A JavaScript tool to present Mathematical Morphology to beginner

CÉSAR C. NUÑEZ[1] and AURA CONCI[1]
[1] Universidade Federal Fluminense (UFF), Brazil
{cnunez,aconci}@ic.uff.br

1. Introduction

This work presents an incredible light (71KB) Object Oriented JavaScript program for Mathematical Morphology, running in Internet, available at: http://www.ic.uff.br/~aconci/Morfologia. It employs DOM (Document Object Model) resources supported by the following browsers (or compatibles): Internet Explorer v 5.0+ ; Netscape v 6.0+; Opera v 7.0+; Firefox v 1.0+;or Mozilla v 1.7+. It is composed of only 4 Hyper Text Markup Language (.html) files, 1 file of Cascading Style Sheets (.css), 3 external JavaScript files (.js) and 14 (.gif) figures. The code is completely open and clear to be included in html pages or adapted to other applications. Figure 1 shows its appearance.

The implementation can be used for instruction in several different ways. In the simplest one, on the internet, it allows experiences using the implemented operations (as expansion, contraction, dilation, erosion, opening, closing, intersection, union, subtraction, complement, reflection, etc). It permits direct experiences using binary images and structuring elements that may be drawn on the screen using painting tools of various types. Composed operators such as the top-hat, hit-or-miss, morphological gradient, and any other built from previously defined operators may be defined using the tool. Combination may be in cascade or in a parallel structure. Once tested in the tool, the corresponding code of the composed operators can be used in another program. Operations may be composed on the screen of the tool by the user. They can be combined in cascade or in parallel. Students interested on special topics on image analysis, for instance, after trying the functions in a combined way can use the code in their specific programs. New morphological function can be included as a specific function in a new code version adding new buttons or options to the menu. Moreover, additional structuring elements can be added easily.

Students of JavaScript language can learn the basic structure of the program and improve it including new functions since the code is easy and comprehendible. For beginner in mathematical courses it can be used to explain the common set theory operations (union, complement, intersection, subtraction, etc). In addition, it works as laboratory experiments for use in classrooms.

2. The Tool

The tool is an open code and it has been created to be used on the main browsers/compatible in the market, as long as they support CSS and DOM. There is no need for any additional plug-in or any other components. Its codification uses object orientation paradigms and it was developed in JavaScript. It makes possible the construction of simple images in a matrix of 30x30 points, with color depth of 1 bit (black and white), and it applies up to 11 transformation operations combined in any quantity and order. However, all these limits can be increased in the implementation according to the user needs. The number of options has been kept small in order to maintain simplicity of use and also only one interface frame.

2.1 Tool's Objects and Methods

The tool has two main classes of objects: matrixDisplay and imageObject. The first defines a type of object that simulates the pixels on monitor screen. It creates a square block matrix (with 9x9 pixels), which works as the elements of an image in the user display. This matrix's dimension can be dynamically defined, on the creation of the object's instance. Each block can assume two states: black and white, it allows the representation of images as bitmap.

The data about each block's conditions are kept in a bi-dimensional array and can be altered according to the image being viewed. This image is linked to an object and at any time, the active image can be altered (also stored as an array). This class has only one method, clearDisplay, which is used to clean the display, changing all blocks into white. On the program interface, two image display areas have been created. One for the image definition, with 30x30 blocks, and other for the Structuring Element definition, with 5x5 blocks.

The imageObject has, basically, two attributes: Its own identifier, and the bi-dimensional coordinate array, which represents the active image's points. This imageObject presents five methods: addPixel; delPixel; showArray; showImage and clearAll. The method addPixel adds a new pair of coordinate to the array, on a specific point in the display. Method delPixel removes a given pair of coordinates from the array. Method showArray sets up a string that serves to represent, textually, the coordinate's array. Method showImage shows the linked display's image. Method clearAll deletes array's coordinates. In addition to these two classes, the tool has various functions which perform tasks related to interaction with the user and apply the operations, transforming the original image.

2.2 User interface

The user can interact with the tool through four distinct areas: The brush area, the main design area, the structuring element, and operations definition area (Figure 1). Brush areas, located on the top-left are used for

definition of the brushes to be used for image construction. The possible types are arranged vertically: 1, 2x2, 3x3, 4x4 and 5x5 square block. There is also a tool to fill quickly an entire bounded area. When clicking on a white block, the corresponding area will be painted black. If the clicked block has already been painted, it acts as an eraser, removing the black points from the area. The brushes only work on the design area. Design Area is the area that displays the original image and its operations. It is presented as a matrix with 30x30 blocks, where the user image is created. This image will suffer the selected transformations, and the resultant image will be shown on the same area. The work image is linked to the letter "A" for identification on the transformation operations. The label indicates what image is being viewed. There is a box to display the image's coordinate points. The "x" button hides this box if selected by the user.

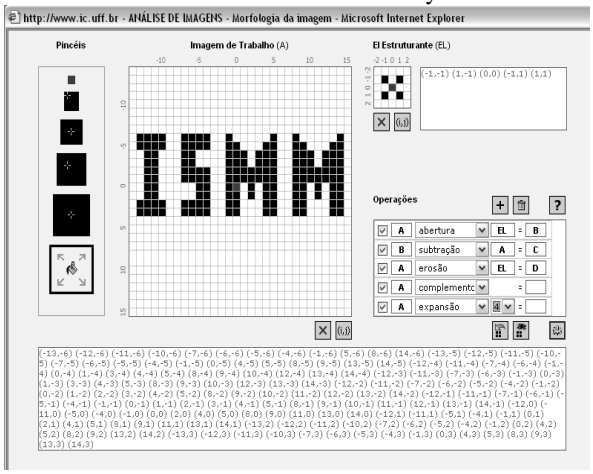

Figure 1: Tool Interface.

The Structuring Elements area, similar to the design area, simulates 5x5 pixels bitmap for construction of structuring elements. Here the brush size is always 1 point, independently from whatever is selected. On the right hand side there is a box showing the coordinate of the structuring elements points. There are buttons for cleaning the design and for hiding or showing the coordinate box. The operator definition area is where the user defines the transformations to be applied to the image. The controls in this box are concentrated in five operations: Add; delete; make; quick access; done and open help file. They promote the addition of a new operation to the stack; remove the non-selected operations from the stack, make an image active in the design matrix, allow a quick access to the on work image without making it the active image, perform the operations stored in the stack and open the help file.

2.3 Shortcut keys

The interface has shortcut keys to the interface visual controls. TAB changes the brush size, sequentially; 1,2,3,4,5,6 turns possible to select directly the brush size; "1" being the smallest, and "6" the bucket. DELETE cleans the main matrix active image. If the active image is the resultant image, the work image becomes the active image. F2 key hides or exhibits the box with the active image coordinate on the main matrix. DEL (in numeric keyboard part) clears the structuring element's image. F4 key hides or exhibits the box with the structuring element's coordinate. Key + (in the numeric keyboard) adds an operation to the operation stack. Key - removes the non-selected operations from the operation stack. F1 key opens a pop-up window with the help file. ESCAPE makes the work image the active image of the main matrix. SPACE BAR key exhibits the work image, without making it the active image. When releasing the space bar, the active image returns as the resultant image. The ENTER key performs the operations selected from the stack;

2.4 Configuring Transformations

To configure a sequence of transformations, the user must click the "+" button of the interface. The first selected operation must always have letter "A". The user then selects the operation from the list-box. The dilation, erosion, opening, and closing operations will use the structuring element (EL). Others operations ask for definition of the neighborhood shape or for the name of another image that has already been calculated (each image may be represented by one letter only). In the last field to the right of each operation one letter has to be chosen for the resultant image. This letter can be used in other operations, if one wants to apply transformations on cascade. In case of many cascade operations, the resultant image will be the answer to the last operation, or their sequence, if they have been configured on cascade form. To cascade the operations, the letter of the last field has to be the same as the letter of the first field of the next operation.

3. Conclusions

The tool presented in this article only requires a browser to be executed. It has an open code, and it can be used also in JavaScript classes. This tool can be downloaded from http://www.ic.uff.br/~aconci/Morfologia where the source-code is also available. The interface is completely intuitive (try it!). The help and hints of each element are written in Portuguese.

References

[1] Gerald J. F. Banon and Junior Barrera, Bases da morfologia matemática para a análise de imagens binárias, 2nd ed., INPE, São José dos Campos, 1998. http://bibdigital.sid.inpe.br/rep/dpi.inpe.br/banon/1998/06.30.17.56.

[2] Serra, J. Image Analysis and Matematical Morfology, Academic Press, London, 1982 Website: http://cmm.ensmp.fr/~serra/cours/