UNIVERSIDAD DE OVIEDO

MASTER IN SOFT COMPUTING AND INTELLIGENT DATA ANALYSIS

PROYECTO FIN DE MASTER
MASTER PROJECT

Image analysis using fuzzy mathematical morphology

Carlos García Mangas

July 2012

TUTOR / ADVISOR:
Jose Otero Rodriguez

Keywords: Fuzzy sets, mathematical morphology, genetic algorithm, computer vision, image restoration.

Abstract

Mathematical morphology is a technique that allows analyzing and transforming images. In this work I'm going to experiment with the possibility of using fuzzy logic and sets, instead of the classic ones, to improve the results of the mathematical morphology operations and its robustness against noise. I also have used a genetic algorithm to optimize the result between different sizes and shapes of structural elements and different arrays of operations, thus obtaining the best possible outcome for each case using the fuzzy mathematical morphology, and then comparing it to the results of the classic morphology.

Contents

List of Figures

1 Introduction

Mathematical morphology is a series of methods and processes used to analyze images, based in classical mathematical theories. Basically the goal of all those processes is to remove useless information of the image, like noise or errors in the image, while preserving its shape and all the important aspects of it. But, in the presence of too much noise and imperfections, this operations do not perform as well as we would like to. This is the main motivation for this work. Using certain fuzzy operators, particularly the Fuzzy Inclusion Indicator approach in [1], I'm going to try and solve these cases where to much noise could be a hindrance to our work. Of course this methods have already been tested in [1] but, not only I'm going to use a Fuzzy Mathematical Morphology (FMM), but also a Genetic Algorihtm (GA), based on the work of [2] which uses a GA to filter images using the classical mathematical morphology. These algorithms, based on the theory of the evolution, have shown good results before, in other aspects of image processing as I will show in the next sections.

So, my main approach to the problem will be a Genetic Algorithm to optimize different sizes and shapes of structural elements, and different strings of operations based on a FMM, this way I will determine a sequence of operations and compare how well different types of structural elements will do when filtering an image and providing a program capable of adapt to the best situation given any image.

1.1 Objectives

The main objectives of this work are:

- Investigate the state of the FMM to analyze images.
- Implement an original algorithm based on the work of [2] and the Generalized Fuzzy Mathematical Morphology of [1].
- Run several experiments to solve noise problems and compare the results with the classical morphology.

1.2 Structure of the work

This work is structured in different sections, the section "Theoretical Concepts" will explain each concept that could be important to understand the main contents of this work, in "State of the Art of Fuzzy Mathematical Morphologies and Genetics Algorithms for image processing" I will show how is the state of the Fuzzy Mathematical Morphology and image analysis in general, in "Metodology, experiments and results" you will find the main experiment and the results, as well as the process and program to achieve these results, the next section "Future work" will cover a few concepts that were researched, but did not make it into the final experiment, finally in "Conclusions" I will give the conclusions I have gathered from this work.

2 Theoretical Concepts

In this section I'm going to present a few key concepts to facilitate the understanding of this work and the subjects related to it.

2.1 Computer vision

Computer vision is a field of research that tries to imitate one of the most efficient ways of acquiring information that the human brain has, the vision, which basically is processing information from the light reflected on every object. The methods used in this field process, analyze and understand images, to determine certain features of them, like how far they are, their shape, color, orientation and others. In the current state of the field we can achieve things like face recognition, reconstruction of 3D and 2D images, tracking of the movement of objects in different images and more.

For more about how this subject started you can check [22], a classical reference on computer vision.

2.2 Mathematical morphology

As said in the previous section the mathematical morphology is a powerfull series of tools, the basic ones are Set theory and Boolean operators, to enhance the information given by an image, based on mathematical theories. The processes that could be done with these techniques involve pre- and post-processing methods, as well as discovering borders and boundaries of the image, for more information about the subject the book of J. Serra, [10], is a good source.

Some of the applications of mathematical morphology:

- Edge sharpening.
- Contrast enhancement.
- Gradient operators.

The main operations of the morphology are erosion and dilation, which consists basically in shrinking and expanding certain areas of the binary image, gray-scale images are treated in a different way. Usually the mathematical morphology is applicated to binary and gray scale images only.

2.2.1 Structural Element

The structural element is our tool to change the image, is a very small image, that usually has an odd numbered size because there is a center in it. For every problem the structural element may need an specific shape and size, that's where a genetic algorithm will come in handy, changing the size and shape, trying the best combination possible for the image. The way the structural element change the image is by applying it in different ways defining the two basic operations are erosion and dilation.

2.2.2 Erosion

Erosion is one of the main operations of the mathematical morphology. Basically consists in eroding certain zones of the image, by putting the center of the structural element in each black point and transforming every white point into a black one where the ones of the structuring element overlap them.

A more mathematical definition would be:

$$A \ominus B = \{z | (B)_z \cap A^c \neq 0\} = \bigcup_{z \in A} B_z$$

where A is the image, B is the structural element, and z is a point belonging to A. In Figure 1 the white pixels are 1s and the black ones are 0s, see how applying the structural element "eroding" eliminates the object.

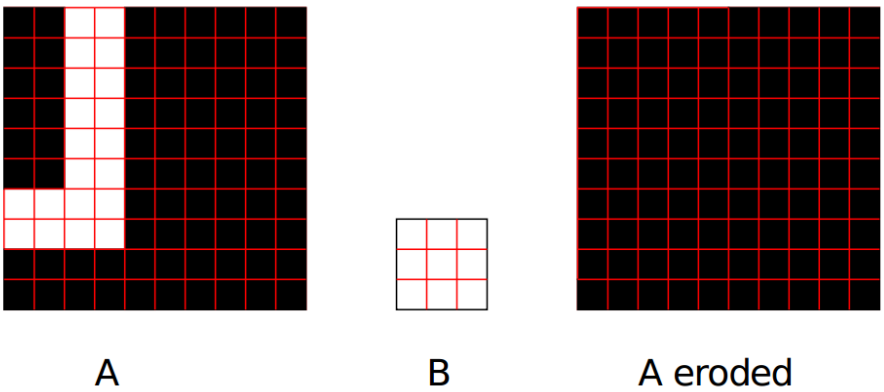

Figure 1: Example of an erosion, the white pixels are 1s and the black ones are 0s, see how applying the structural element "eroding" eliminates the object.

2.2.3 Dilation

Dilation is the other main operation of the morphology, as well as the erosion it's applied by the structuring element. This operation is the total oposite of the erosion, it basically puts the center of the structural element in each white point and transforms every black point into a white one where the ones of the structuring element overlap them.

The mathematical definition:

$$A \oplus B = \{z | (B)_z \cap A \neq 0\} = \bigcup_{z \in A} B_z$$

where A is the image, B is the structural element, and z is a point belonging to A. In Figure 2 the white pixels are 1s and the black ones are 0s, see how applying the structural element "dilating" expands the object.

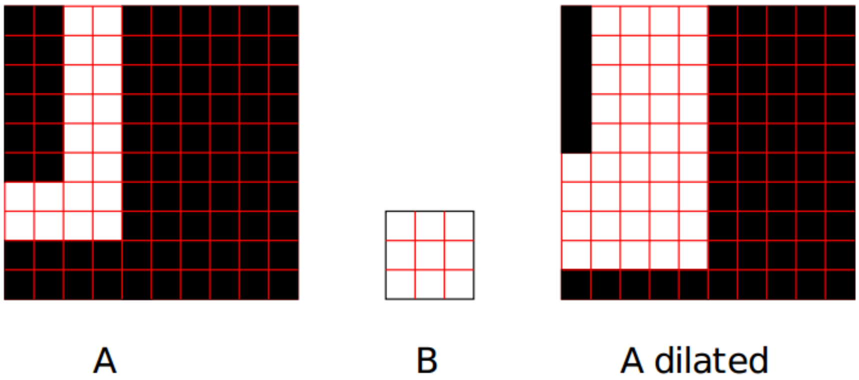

Figure 2: Example of a dilation, the white pixels are 1s and the black ones are 0s, see how applying the structural element "dilating" expands the object.

2.2.4 Opening

The opening operation is erosion followed by dilation, it removes all areas smaller than the structural element, smooths boundaries, erase thin portions of the image with almost no change in the object area, it's defined as:

$$A \circ B = (A \ominus B) \oplus B$$

where A is the image and B is the structural element. In Figure 3 eroding erases the lines and most of the noise, dilating then enhances the words in the image.

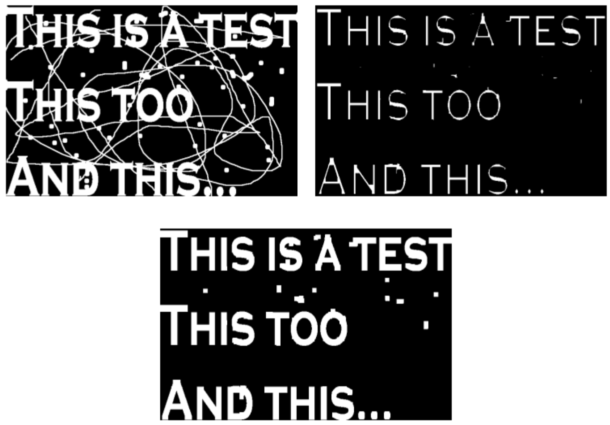

Figure 3: Opening example, clockwise: original, eroded, dilated, see how eroding erases the lines and most of the noise, dilating then enhances the words in the image.

2.2.5 Closing

Closing is dilation followed by erosion, it fills small areas of the image, connects close objects, and just as the opening operation it smooths boundaries and makes almost no changes in the object area, it's defined as:

$$A \bullet B = (A \oplus B) \ominus B$$

where A is the image, B is the structural element. In Figure 4 the dilation erases the internal lines and the erosion enhances the words.

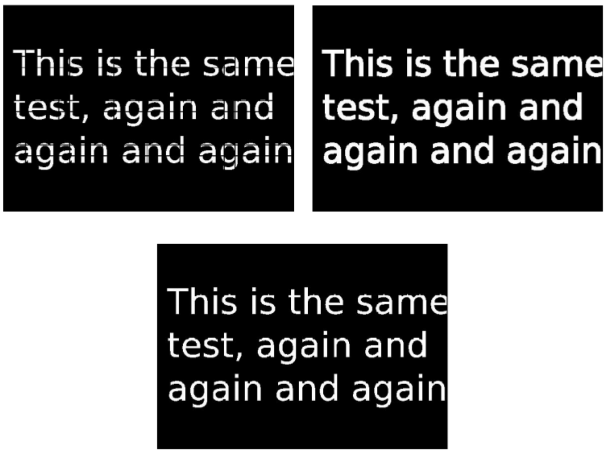

Figure 4: Closing example, clockwise: original, dilated, eroded, here the dilation erases the internal lines and the erosion enhances the words.

2.2.6 Gradient

Applying erosion and dilation operations to the image can be used to find the boundaries of objects, to discover difference between the original image and the transformed ones we use the gradient:

- External gradient: arithmetic difference between the dilated image and the original image: $A\Omega^{+}B = (A \oplus B) - A$
- Internal gradient: arithmetic difference between the original image and the eroded image: $A\Omega^{-}B = A - (A \ominus B)$
- Morphological gradient: arithmetic difference between the dilation and the erosion: $A\Omega B = (A \oplus B) - (A \ominus B)$

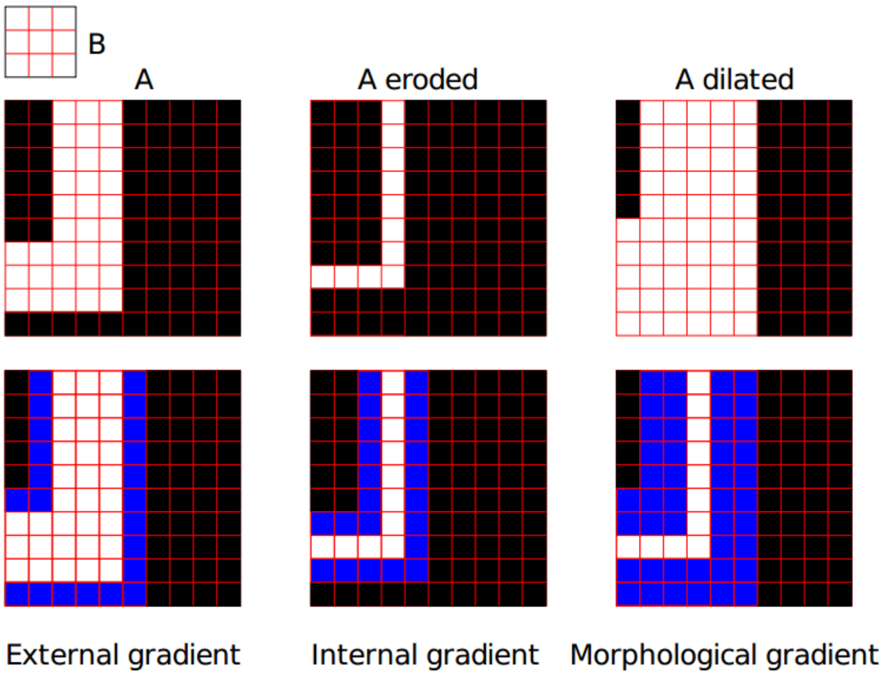

Figure 5: Gradient examples.

2.2.7 Hit or miss transformation

This transformation is a basic tool for pattern recognition, it allows the selection of sub-images with certain properties, and with them you can obtain thinning, skeleton and pruning. It is defined as:

$$A \otimes B = (A \ominus B_1) \bigcap (A^c \ominus B_2)$$

where A is the image, B_1 and B_2 are structural elements and A^c is the complement of A. For a point x to be in the result, the structural element B_1 matches A at x and B_2 matches A at x.

2.3 Fuzzy logic

This concept was introduced by Lofti A. Zadeh in [3] and refers to a logic made by predicates that are not completely true or false, that belongs or not entirely to a class, but instead each concept is defined by a membership grade to a certain class, usually a grade that range between 0 and 1, and defines how true it is.

The main concept behind this logic is that if humans can gain knowledge with imprecise or incomplete information, computers should be able to do the same.

This concept has given birth to many theories and aid in many fields, from machine learning or control theory to computational intelligence and many more.

2.3.1 Fuzzy sets

These sets are characterized by all their elements having a membership degree, which is defined by the memebership function represented as:

$$A = \{(x, \mu_A(x)) | x \in U\}, \mu_A : U \to [0, 1]$$

where A is a fuzzy set, x is a point in the set and $\mu_A(x)$ is the membership function, as shown in [19] by Zadeh. In this work, we are going to use a fuzzy set as the structuring element, hoping to introduce resistance to imprecision or imperfections in our algorithm.

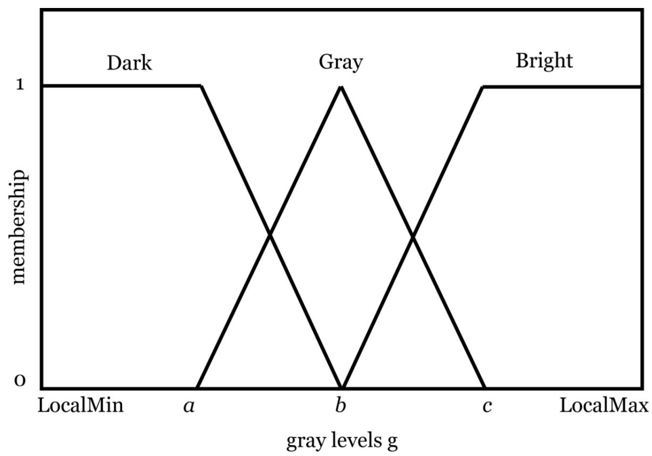

Figure 6: Example of a typical fuzzy set defining gray levels.

2.4 Thresholding

Thresholding is an operation that separates, or classifies, all the pixels of an image in two classes, background and foreground, thus transforming an RGB or gray-scale image into a binary image. Basically the method consits in calculating a threshold, by any mean, like for example calculating the histogram of an image and choosing a mid point, which will be considered the minimum of the threshold, or just calculating the local minimun of a bimodal histogram and then separate every pixel between the ones greater than the threshold and the lower ones.

Here I will list some of these methods:

- **Statistical Decision Theory based Threshold Computation:** Statistical Decision Rule, Gaussian Distributions and Model fitting method.

- **Thresholding minimizing the intra-class variance, Otsu's method:** Tries to minimize the spread of the left and right sides of the histogram halved by the threshold, the goal is to find a threshold that minimizes the sum of the spreads for background and foreground pixels, this spread is measured using the intra-class (within-class) variance.
- **Thresholding minimizing Kullback divergence:** For each possible thresholding value, the distance, which is the Kullback's divergence, to the image histogram is measured.
- **Fuzzy Thresholding Algorithm (Huang and Wang method):** The image is an array of fuzzy singletons, each one represents a pixel has a membership belonging to background or foreground.
- **Hierarchical thresholding:** The image is processed with different resolutions.

Thresholding is usually used to discover forms and counting or just as a way to obtain a binary image.

Figure 7: Example of a thresholded image.

2.4.1 Top-hat transformation

The top-hat transformation is used when the thresholding of an image is difficult, usually because the gray level of the image changes slowly. This is a morphological operation thus is applied with a structural element, and is defined as follows:

$$T = A - (A \circ B)$$

where A is the image, B is the structural element.vision

2.5 Genetic algorithm

Genetic Algorithms, presented in [4], are part of the Evolutionary Computation, a branch of computational intelligence which is based on Darwin's theory of Evolution. Good references to this subject are [23] or [24].

The main component of the GA is the chromosome, which tries to imitate a string of DNA composed by genes, the location and encoding of this genes makes certain traits to appear. In this case the chromosome is a string of characters, usually 0s ans 1s, which are part of a population of chromosomes and defines a solution of the algorithm.

These are the main steps of a basic GA:

1. **Initialization:** initialize the population creating the chromosomes, usually this process is totally random. After the creation of the population, the algorihthm calculates a fitness function for each chromosome, the fitness of each chromosome represents how good a solution is.

2. **Selection:** select the best individuals of the population to make the new ones with them as parents.

3. **Reproduction:** in this step two parents of the previously selected are going to create two new individuals from them, usually comprehends two operations: crossover and mutation. Crossover consists on switching part of one parent with the other to create a new one. Mutation runs over every character of the new individual, and if triggered then changes it. Both operations have a probability of happening.

This process is repeated a fixed number of iterations or whith any other stopping criteria, each time the algorithm iterates the new chromsomes created in step 3 replace the worst ones of the population. The result is the chromosome with the best fitness.

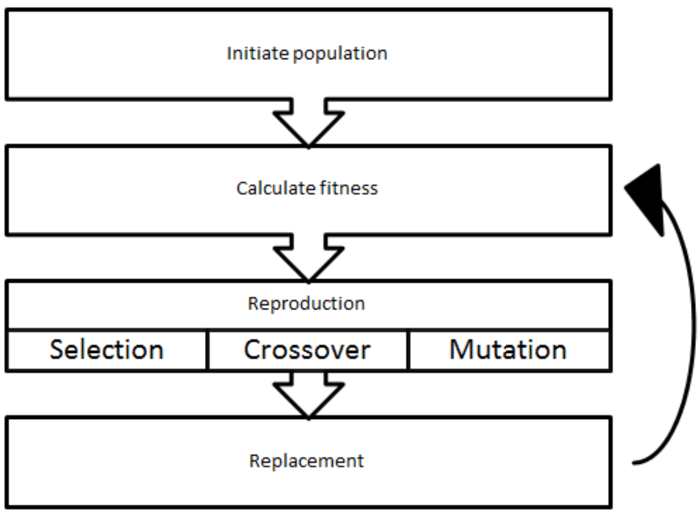

Figure 8: Sequence of a genetic algorithm, adapted from [4]

3 State of the Art of Fuzzy Mathematical Morphology and Genetic Algorithms for image processing

In this section I'm going to talk about the current state of the Image Analysis using Fuzzy Mathematical Morphology and Genetic Algorithms, which are the main themes of this work. I'm going to show papers and books related to this subjects to illustrate their state.

3.1 Fuzzy Mathematical Morphology

Fuzzy mathematical morphology is supposed to show better results with images corrupted with noise than the classical morphology so I'm going now to present [1], and many other references, which is one of the papers that inspired my work, it covered the problem of the creation of a fuzzy mathematical morphology that showed good results on the presence of noise.

Binary morphology has some problems, mainly in the presence of too much noise as demonstrated in [5], therefore many researchers have tried to use soft computing solutions to solve it, with success only in certain conditions as we can see in [6]-[8]. In [1] a new generalized fuzzy mathematical morphology (GFMM) based on a fuzzy inclusion indicator (FII) is presented, and provides good results with 2D and 3D grayscale images.

This fuzzy inclusion indicator presented in [1] let us known when a fuzzy set is included on another, it's based on an binary inclusion indicator and translated into fuzzy mathematics, having all the propierties of the original indicator, and this will be the base of all the GFMM. The definition of the inclusion indicator is:

$$\mu_{I(A,B)}(u) = inf_{x:\mu_A(x)=u}\mu_B(x)\forall u\epsilon[0,1]$$

where A and B are fuzzy sets, μ represents a membership function, x is a point that belongs to A and gives as a result u and $I(A,B)$ is the result of applying the FII to A and B.

The main operation of the GFMM is erosion, dilation is defined as the contrary of it. Of course opening and closing are defined by erosion and dilation, so the GFMM is analog with a binary mathematic morphology and this one is a particular case of the GFMM, if we use a bi-valued membership function. To obtain the robustness we are looking for, this GFMM uses a fuzzy structural element as represented in Figure 9, when its core set it's bigger than a pixel then it gets good results, even with noise, when applying previously mentioned operations.

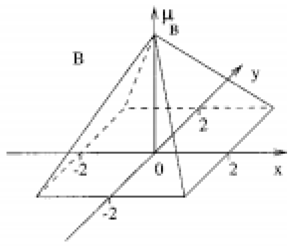

Figure 9: Fuzzy structural element, picture taken from [1].

For instance the morphological skeleton and morphological shape are good methods to get the shape of an image, so these will be defined in the GFMM as follows:

$$X = \bigcup_{k=0}^{N} S(k) \oplus kB$$

$$X = \bigcup_{k=0}^{N} L(k) \oplus kB$$

where X is the resulting object, B is the structural element, $S(k)$ are the skeletal subsets defined on [1], $L(k)$ are the spines to find the shape and N is the largest integer such that $X = NB \neq 0$. Finally some experiments were done in [1] to show that it's true that GFMM performs better analyzing images with noise than binary and grayscale techniques, both with 2D and 3D images.

Mathematical morphology was created for binary images, and later on, extended to gray-level images, transforming these ones into binary images through thresholding, Serra was the first to try this in [10]. In [9] is proposed that to transform this gray-level images into fuzzy sets you can consider the gray-level information of each pixel as a membership degree to the data set, to do this she scales the information of every pixel into the range [0,1], using any kind of normalization, a N-function, a single-sigmoid or any other, this process can be seen in Figure 10.

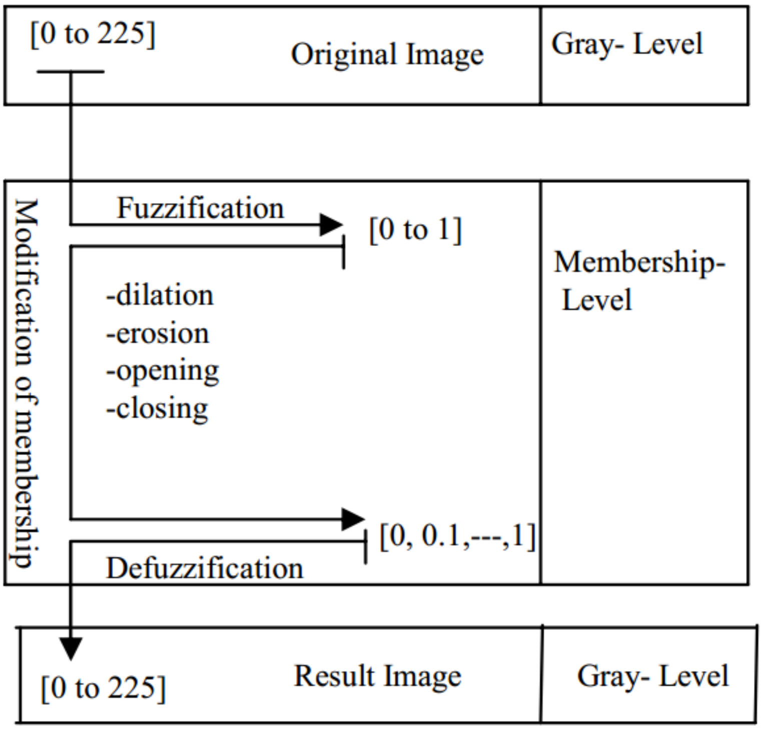

Figure 10: Fuzzy image processing, taken from [13]

With the gray-scale image fuzzified it's time to apply the morphology operations. The fuzzy erosion in [9] is defined by two different methods, Minimum or Average functions:

$$E_{min} = (I' \ominus SE)_{min} = min\{1 - |I' - SE|\}$$

$$E_{ave} = (I' \ominus SE)_{ave} = 1 - \sum |I' - SE|/sizese$$

where I' is the image, SE the structural element and $sizese$ is the number of active pixels in the structuring element. The Minimum erosion corresponds to the classical erosion, and the average one provides smoother results, which to apply depends on the image, with high degree of connectivity between pixels a Minimum erosion would be enough, for example. The dilation is then defined as the opposite of the erosion:

$$I' \oplus SE = E^c(I'^c, SE) = 1 - E(1 - I', SE)$$

where I' is the image, SE the structural element, I'^c is the fuzzy complement of

I', and E^c is the complement of the erosion. Apart from this [9] covers a fuzzy method to detect edges, the Normalized Fuzzy Sigma NFS distance transform based on Euclidian distance:

$$NFS(I_{ij}, E_{ij}) = \begin{cases} NFE(I_{ij}, E_{ij})/2 & for 0 \leq d_{ij} \leq 0.5 \\ 1 - NFE(I_{ij}, E_{ij})/2 & for 0.5 \leq d_{ij} \leq 1 \end{cases}$$

where I is the image, E is the erosion, NFE is the Normalized Fuzzy Entropy function and d_{ij} is the distance between the image and the erosion in the point (i, j). This will detect the fuzzy boundaries of the image, and the fuzzy skeleton by thresholding and thining the fuzzy boundary set. Some results can be seen in Figure 11.

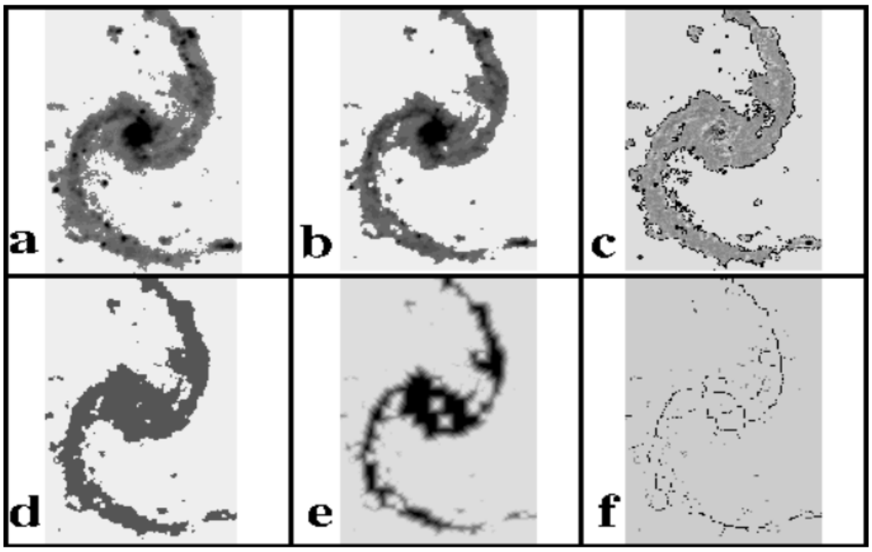

Figure 11: Finding the fuzzy skeleton, a) image b) Minimum erosion c) NFS d) boundary set e) radius image f) result. Image taken from [9]

An interesting part of [13] was how the Top-Hat transform was translated to fuzzy morphology, in [14] the author gives a universal framework to create a fuzzy mathematical morphology and his approach on the fuzzy hit-or-miss transform.

He defines erosion and dilation as an adjunction, that means that for every dilation (δ) there is an erosion (ε) that follows:

$$\delta : \mathcal{L} \rightarrow \mathcal{F} \Longleftrightarrow \varepsilon : \mathcal{F} \rightarrow \mathcal{L}$$

where $\mathcal{L}$ and $\mathcal{F}$ are two lattices. Now, if there is a conjunctor in which $c(0,1) = c(1,0) = 0$ and $c(1,1) = 1$, and there is an implicator in which $i(0,0) = i(1,1) = 1$ and $i(1,0) = 0$ then they are an adjoint conjunctor-implicator pair and then the erosion and dilation are universally define by:

$$\delta(X)(x) = \bigvee_{y \in E} c(A_x(y), X(y))$$

$$\varepsilon(X)(x) = \bigwedge_{y \in E} c(A_y(x), X(y))$$

in which E is an universal set, and A and X are fuzzy subsets of E. This way the author defines a universal framework for any fuzzy mathematical morphology.

Another interesting aspect of that paper is the hit-or-miss transform, which is mathematically defined by:

$$\tilde{\pi}_{A,B}(X) = \varepsilon_A(X) \triangle \varepsilon_B(X^c)$$

where $\triangle$ is a t-norm of the membership function of the fuzzy subsets A and B.

A lot of the papers related to image processing have as an objective image segmentation, for example the ones that I'm showing here, which is a basic operation in many researches this and the fact that a fuzzy structural element is one of the pillars of this work made [11] a really interesting paper, basically in [11] the authors describe an approach for fuzzy structural elements in image segmentation.

Image segmentation separates an image into uniform regions or subsets, in which all pixels share a commom feature, and each region is significantly different from the other. There are three major approachs to segment an image, region growing, region merging and pixel classification, and the last one is the one that it's used in [11], as it's the one that allows fuzzy techniques and facilitates uncertainty management.

The algorithm proposed in [11] is divided into three steps:

- A fuzzy set is associated to every region which is going to be segmented.
- Analyze each pixel and discover the membership degree to every previously mentioned fuzzy set, to discover in which one its going to be.
- Transform the fuzzy segmentation into a crisp one.

The problem with this algorithm it's that it has a very restrictive of rule:

$$\mu_{R_k}(p_{ij}) > \mu_{R_l}(p_{ij}) \forall p_{ij} \in R_k; l \neq k$$

where R is a region, and p is a pixel. For this reason a fuzzy structural element is proposed, such that satisfies that for every region a high enough number of pixels satisfy the previously defined rule, and that the regions to be segmented must be big enough. To do this the authors suggest to decide the best shape, size and structure for the structural element for every different image. In relation to the shape, it should be elliptical when the objects of the image are almost linear with only a few corners, rectangular when there are a lot of corners, and a circle when neither of the previous conditions are met. The size would be usually small, to not overlap adjacent regions. The structure depends on the size and the shape as show in Figure 12, if the size is to small it makes no sense to divide the structural element into sub-regions, and when the shape is not a circle the orientation of the axis must be considered like in Figure 13.

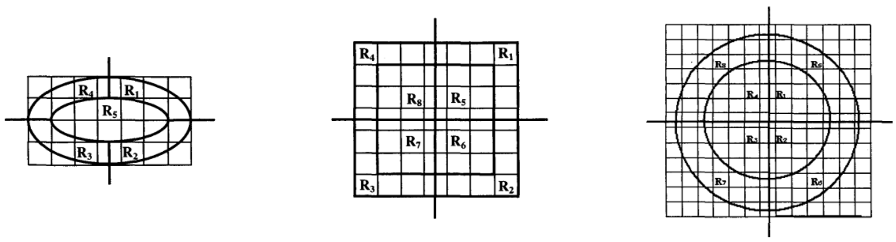

Figure 12: Different sizes and shapes for structural elements, and sub-regions. Taken from [11]

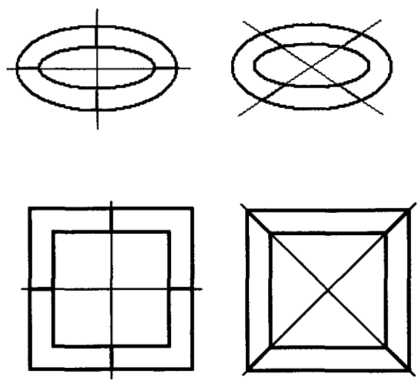

Figure 13: Axis orientation on rectangular and elliptical structural elements. Image taken from [11]

Once the main aspects of the structural element have been decided, an aggregation function that will assign a new membership degree to each pixel is necessary, the greater the initial membership degree of the pixel was, the greater the new one is going to be, and the larger the number of pixels with great membership degrees the larger the final membership degree of the region will be.

This structural elements have proved to give very accurate results for segmentation in [11], always if the tuning of the parameters is right.

Following on the segmentation subject, in [12] the authors introduce a different approach in fuzzy morphology to solve it.

The definition of the process, the erosion, the dilation and the different operators of a morphology it's what differentiates a fuzzy morphology from another one. In [12] the authors define this parameters as follows, with the objective of obtaining segmentation on medical images to discover blood vessels. The fuzzification is done in that paper with the sigmoid function for the image and a specific formula for the structural element. Once the image and the structural element are fuzzified, fuzzy erosion is then defined by a binary inclusion operator, very similar to what it's shown in [1]:

$$\varepsilon^F(f, B)(x) := inf_{y \in f}\{I(B(y)), f(y)\}$$

where f is the image, B is the structural element and $I(B, f)$ is the binary inclusion operator. The dilation is defined by the binary conjunction:

$$\delta^F(f, B)(x) := sup_{y \in f}\{C(B(y)), f(y)\}$$

where f is the image, B is the structural element and $C(B, f)$ is the binary conjuction operator. It's interesting to note that in [13] it's proposed these almost the same operations, but using a fuzzy inclusion and fuzzy conjunction operators, instead of the binary ones. Anyway it doesn't seems to have much impact on the results, as the fuzzy inclusion and conjunction operators give the same output as the boolean ones.

Opening, closing, gradient and the fuzzy top-hat operations are then defined with these two operations, and the method followed in [12] to segmentate the image is:

1. Image fuzzification.
2. Image fuzzy dilation.
3. Calculation of fuzzy Top-Hat transform.
4. Image Deffuzification.

5. Visualization.

Results for this algorihtm can be seen in Figure 14.

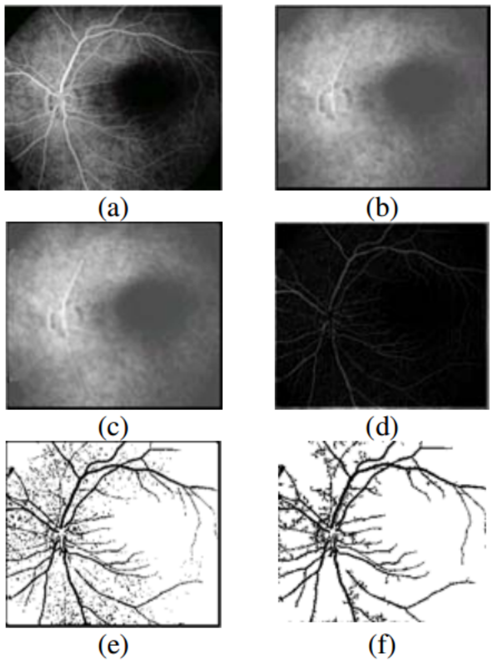

Figure 14: Example taken from [12]. a) Original image b) Fuzzy erosion c) Fuzzy dilation d) Fuzzy Top-Hat e) Deffuzification f) Result

3.2 Genetic Algorithms on image processing

Another paper that was key at the time of doing this work and one of the first to use GA as a image proccessing tool is [2], which demonstrates that a good implementation of a genetic algorithm shows good results for image filtering.

Sometimes the techniques used for image analysis give bad performance, this happens as well with morphology based operations, even if they are good in theory and have solved many problems, like the presence of too much noise on an image, the methods designed tend to be unefficient.

In [2] the authors suggest the use of Genetic Algorithms (GA), which, as stated before, are probabilistic methods, based on natural selection, to search for the optimal morphological operator. The paper introduces both mathematical morphology and GA, but I'm going to leave out those sections, so I can explain their work better.

The first thing you can notice when trying to solve a filtering problem is how big the search space is, you need to search for the optimal morphological operator and structural element, so the possibilities are approximately $(5^{25})^{31}$, which gives us an idea of how important the GA are in these situations. So, to be able to use GA in my work, first I need to convert the structural elements and the morphological operators in chromosomes. To do this the structural elements will be mapped as 0s and 1s in their appropriate position of the chromosome if they're binary, if not, just transform their digits into binary code and the mapped each digit in its proper position. In case of having positions of the structural element that are not used, they can be coded as 0s and then add one to every digit, for example. As for the morphological operations, I just need to codify the operators ("erosion", "dilation", "do-nothing"), and put them in sequence. Once I have both the structural element and the morphological operations I put them together and form the chromosome.

The next thing to do is define the properties and operators of the GA. In [2] the selection operator is totally random, the crossover operator is a simple one with a probability of 0.75, it takes a random point of the parents and separates each of them in that point, then join one part of a parent with other part of the other parent. The mutation operator presented in [2] has a probability of 0.025 of flipping a binary digit of each gene. And finally the stopping criterion is a fixed number of iterations.

To know if the GA has been successful the authors recommended the use of images from which we know the optimum filter. Finally they recommend "off-line" and "on-line" performance meassurement, during optimization, as well as mean absolute error (MAE) and mean squared error (MSE), to measure the overall performance of the method.

Later I will show that the coding of the chromosomes of the genetic algorithm will be based in the work of [2].

An interest way to solve segmentation can be seen in [16], the authors use a Genetic Algorithm to solve the problem and make possible object counting, Note that the method showed in [16] does not use any kind of mathematical morphology, but instead is centered in the use of the GA to improve efficiency.

To find the different objects on an image the algorithm must look for point features, the authors of [16] suggest the use of a Hough Transform to do this, which provide the robustness necessary and reduces the overall complexity of the GA. The paper's approach for the main body of the GA is an steady-state model, which differentiates from the rest for having very small generations gaps, with few offspring chromosomes and a much quicker availability of them. The selection process is based on the rank of every chromosome of the population, the population is ordered and the higher the chromosome is the more possibilities of reproduction.

When trying to find an object in an image there a certain aspects that can make harder this process, occlusions, noise, damage and distortion, so it's important to choose the more robust fitness function possible so the algorithm gets the exact number of objects, no more and no less. The function that the authors found to give the best results is:

$$f_4 = \{\sum_{i=1}^{p'} c_i''\} - \{\sum_{i=1}^{p'} c_i' - \sum_{i=1}^{p'} c_i''\}$$

where c_i'' is the number of pairwise compatibilities between features in the image, which defines the number of objects, and p' is the number of pseudo-objects, real and noise objects, this function did well in the tests of that paper showing the potential of GAs as an image processing tool.

3.3 Fuzzy Mathematical Morphology and Genetic Algorithms on colour images

I've shown in this work how the fuzzy mathematic morphology and genetic algorithms can be applied to binary and gray-scale images, but what about colour images? there a few interesting works on the subject that I'm going to show.

This topic is covered in [15] where the morphological operations follow the same definition as in [14], thus I'm going to focus into the new aspects of [15] and it's approach to colour image processing.

The main problem of the colour images is that they work in a multidimensional space, and there is no natural order of the elements, so there are three different approaches to represent colour images that could be used in a fuzzy morphology, the HSV (hue, saturation, value), the HLS (family, purity, intensity) and YCrCb (luma, red-cyan, yellow-blue). Of those three the desirable approach is the last one, YCrCb, because it's the only one whose paremeters are all linear and correspond to RGB, and also because Cr and Cb have the same weight, so the only parameter with priority is the luma, which is the parameter that let the human eye to distinguish different objects.

Now, if X is a colour image, N the number of labels in the range [0,1], $T(x)$ is the colour integer code of each pixel and $Y(x)$ is the luma, then with this transformation:

$$(\chi(X))(x) = \frac{N^2[(N^2-1)Y(x)] + T(x) - 1}{N^4 - 1}$$

we can obtain $Y(x)$ when we have $\chi(X)$, and $\chi^{-1}(\chi(X))$ gives you an approximation of the original colour of the image, so for any gray-scale image

$\chi(\chi^{-1}(Y)) = Y$. And now it's possible to define the erosion and dilation, with B as an structural element:

$$\delta_B(X)(x) = \chi^{-1}\left[\bigvee_y c(B(x-y),(\chi(X))(y))\right]$$

$$\varepsilon_B(X)(x) = \chi^{-1}\left[\bigwedge_y c(B(y-x),(\chi(X))(y))\right]$$

The authors of [17] propose another example of colour images treated with a fuzzy morphology, in this case transforming the image into a fuzzy set following the approach of pixel classification and getting the different colours that appear in the image.

The first step, of the method the authors propose, is to calculate the modes of the probability distribution function (PDF) of the image to get the fuzzy set from it. To do this they use a co-ocurrence matrix (CCM), which considers the spatial and color interactions between pixels, calculate the CCM for each observation of each color feature of every point of the image and then normalize the results into the range [0,1], so it can be considered a membership function, this normalization is the fuzzification of the results, which is done by analysing the variation of the PDF in the domain of observations of every point.

Now that the image is transformed into a fuzzy set, the authors define erosion and dilation as follows:

$$E_v[\mu_x(X(c_1,c_2))] = inf(max[\mu_x(Y), 1 - v(Y - X(c_1,c_2))]$$

$$D_v[\mu_x(X(c_1,c_2))] = sup(min[\mu_x(Y), v(Y - X(c_1,c_2))]$$

in these formulae Y is the structural element, X is the image and v is an structuring function which is defined for erosion and dilation in a different way:

$$v_E(X(c_1,c_2)) = [1 - \mu_x(X(c_1,c_2)).(1 - g_{\mu x}(X(c_1,c_2))]$$

$$v_D(X(c_1,c_2)) = [\mu_x(X(c_1,c_2)).(1 - g_{\mu x}(X(c_1,c_2))]$$

where X is the image, and $g_{\mu x}$ is the gradient of μ_x . In Figure 15 you can see the result of the method, from a picture with five different colours.

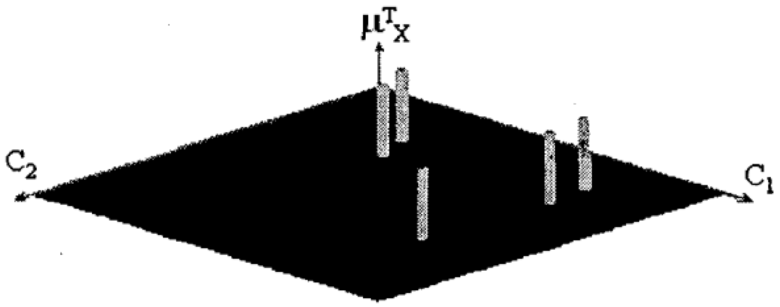

Figure 15: Example of fuzzy set after applying the method of [17], taken from that paper. Each column is a segmented colour.

In [18] the authors use a genetic algorithm to find image similarity in colour images.

The main motivation behind [18] is the retrieval of images from databases, the usual approach is to assign keywords to pictures and images and search for similar keywords to find similar images, but if segmenting the colours of an image you can search for similar images based on the colours and textures. First the image is segmented based in the colours and the texture features with a segmentation algorithm, to do this the authors normalize the colours and features mapping them into the colour space S-CIELAB, which is a metric to measure distances between colours, from them the segmentation process is done through clustering and Bayesian networks, the results can be seen in Figure 16. The colour information will be stored as a probability function, to discover dissimilarities though the Kolmogorov-Smirnov distance, and the texture features would be a distribution of the magnitude of its complex wavelet coefficients.

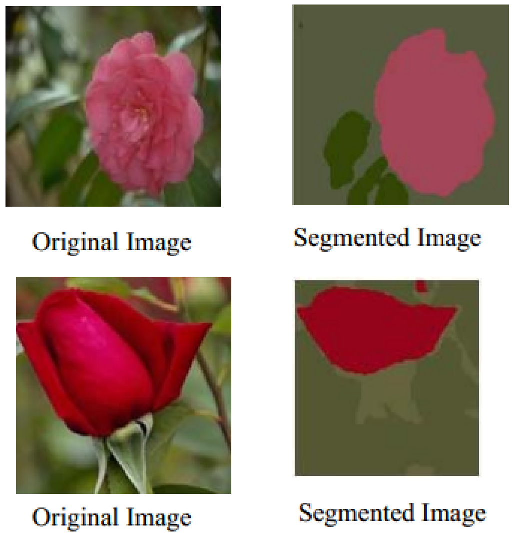

Figure 16: Example of the colour segmentation process, taken from [18]

The genetic algorithm in [18] is used to measure the degree of similarity between two sets of features, and it's a really basic one. The initial population is randomly created, the chromosome is a binarization of the feature set, the fitness function is defined as:

$$F = \frac{1}{a + H_{LK}(P, Q)}$$

where H is the Hausdorff distance between the two sets P an Q, and a is a positive constant. The selection is done following this function:

$$p_s = \frac{F}{\sum_{j=1}^{l} F_j}$$

where F is the fitness of the chromosomes and l is the number of chromosomes in the population. The crossover and mutation operations are the basic ones, but the paper does not mention their probabilities of happening, the stopping criteria is not mentioned either.

In Figure 17 is shown the main process of segmentation described on [18]

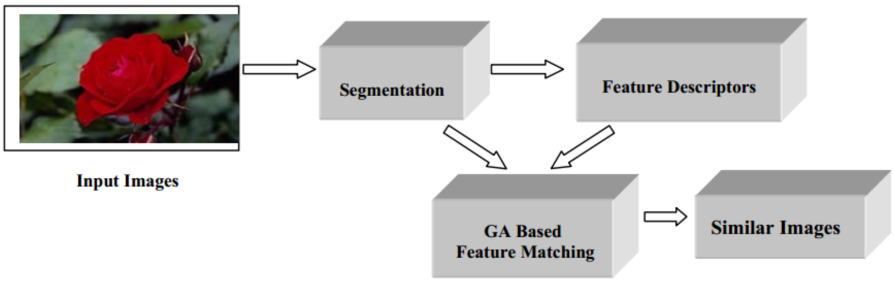

Figure 17: Process of segmentations from [18]

4 Metodology, experiments and results

In this section I'm going to present my experiment and all it's components, first I'm going to explain how it works, and all the theory behind it, and after that I'm going to show the results of this work in different images.

My approach consists in a genetic algorithm that uses a fuzzy mathematical morphology, based on a fuzzy inclusion indicator, to find the best shape and sequence of operations needed to filter and eliminate the noise of an image. The GA has been adapted to work also with the classical morphology to compare the results between these two different approaches, and see which one does better, after this I'm going to test the results on different images than the ones used for the previous steps.

4.1 A fuzzy inclusion indicator and the fuzzy mathematical morphology operators

First of all I'm going to explain one of the main concepts of my approach, the fuzzy inclusion indicator, and how it affects the different operations.

To better understand what is the fuzzy inclusion indicator I'm going to show how the binary inclusion indicator works. An inclusion indicator calculates how true is that a fuzzy set is included into another, so if we have a fuzzy set A and a fuzzy set B and the inclusion indicator is defined as $I(A, B)$, this will calculate how true is that "A is a subset of B", now if we look at [20] there are a number of properties that the inclusion indicator must follow:

1. $I(A, B) = 0 \leftrightarrow \{x : \mu_A(x) = 1\} \bigcap \{x : \mu_B(x) = 0\} \neq 0$
2. $B \subset C \Rightarrow I(A, B) \leq I(A, C)$
3. $B \subset C \Rightarrow I(C, A) \leq I(B, A)$
4. $I(A, B) = I(T(A; v), T(B; v)) \forall v, I(A, B) = I(-A, -B)$
5. $I(A, B) = I(B^c, A^c)$
6. $I(\bigcup_i B_i, A) = \inf_i I(B_i, A)$
7. $I(A, \bigcap_i B_i) = \inf_i I(A, B_i)$
8. $I(A, \bigcup_i B_i) \geq \inf_i I(A, B_i)$

where A, B and C are fuzzy sets, A^c and B^c are the complements of A and B, and $T(A, v)$ is the translation of A by the crisp vector v. The authors of [21] found out some similarities of this inclusion indicator with the fuzzy mathematical morphology so the authors of [1] used it to create the fuzzy inclusion indicator which is defined by this axioms or properties:

1. $I(A, B) = O \leftrightarrow \forall \alpha \in [0, 1], \{x : \mu_A(x) = \alpha\} \bigcap \{x : \mu_B(x) = 0\} \neq 0$

2. $B \subset C \Rightarrow I(A,B) \subset I(A,C)$

3. $If\ A\ is\ convex\ and\ B \subset C \Rightarrow I(C,A) \subset I(B,A)$

4. $I(A,B) = I(T(A;v), T(B;v))\forall v$

5. $In\ the\ general\ case\ I(\bigcup_i B_i, A) \supset \bigcap_i I(B_i, A).\ If\ A\ is\ convex\ I(\bigcup_i B_i, A) = \bigcap_i I(B_i, A)$

6. $I(A, \bigcap_i B_i) = \bigcap_i I(A, B_i)$the

7. $I(A, \bigcup_i B_i) \supset \bigcup_i I(A, B_i)$

where A, B and C are fuzzy sets, A^c and B^c are the complements of A and B, $T(A,v)$ is the translation of A by the crisp vector v and O is a fuzzy set with membership function $\mu_O(x) = 0$. As the fifth property, the duality principle, is lost in the translation, it will be solved by defining the dilation as the dual operation of the erosion.

As shown in the previous section of this work, while reviewing the research made on [1], the authors propose a fuzzy inclusion indicator following the previous axioms that will be used in this work as well and it's defined as:

$$\mu_{I(A,B)}(u) = inf_{x:\mu_A(x)=u}\mu_B(x)\forall u\epsilon[0,1]$$

where A and B are fuzzy sets, μ represents a membership function, x is a point that belongs to A and gives as a result u and $I(A,B)$ is the result of applying the fuzzy inclusion indicator to A and B. In Figure 18 it's shown graphically how the fuzzy inclusion indicator works with two fuzzy sets, and a crisp and fuzzy set.

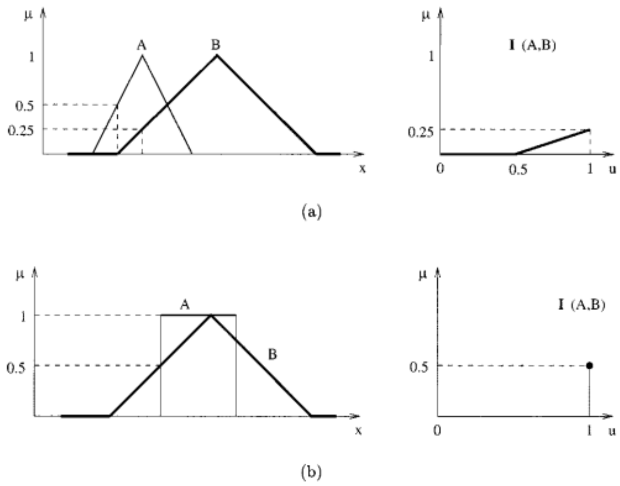

Figure 18: Example of the fuzzy inclusion indicator taken from [1]. a) Two fuzzy sets b) Crisp and fuzzy set

The authors of [1] then define erosion and dilation based on the previously defined fuzzy inclusion indicator, the membership function of the erosion $\varepsilon(A, B)$ is:

$$\mu_{\varepsilon(A,B)}(x) = \mu_{I(T(B,x),A)} D(I(T(B,x), A))$$

where A and B are fuzzy sets, x is a point of the image, $T(B, x)$ is the translation of B to the point x, $I(B, A)$ is the fuzzy inclusion indicator and $D(I(B, A))$ is the defuzzification of the operation. The dilation is defined in [1] considering the duality principle mentioned before:

$$\delta(A, B) = \varepsilon(A^c, -B^c)$$

where A and B are fuzzy sets, and A^c and B^c are the complements of A and B.

Now that I've defined the morphological operations it's time to define the framework, I'm going to use a fuzzy structural element with a binary image, this can be done because the fuzzy mathematical morphology can be compatible with the binary mathematical morphology as shown in [1], if we have a crisp set A with a function equivalent to a membership function with two values $\mu_A(x) = f_A(x)$ and a fuzzy set B with a membership value $\mu_B(x) = 1 \forall x$ then the fuzzy inclusion indicator membership function is defined as:

$$\mu_{I(B,A)}(u) = \begin{cases} 0, & \exists x \in B : \mu_A(x) = 0 \\ 1, & otherwise \end{cases}$$

then the erosion is defined as:

$$\mu_{\varepsilon(A,B)}(x) = \begin{cases} 0, & \exists y \in T(B;x) : \mu_A(y) = 0 \\ 1, & otherwise \end{cases}$$

After calculating the membership function of the inclusion indicator we need to define a defuzzification process. For the erosion I will take the minimum as the defuzzification process, and the maximum for the dilation.

I'm going to use a binary image with a fuzzy structural element, thus the only thing left to be defined is the structural element. The structural element will have a pyramidal membership function, with different sizes from 4x4 pixels to 7x7 pixels, and different support areas, the support area is the top of the pyramid and the only point or points in which $\mu(x) = 1$ for the structural element's membership function.

4.2 Coding the chromosomes of the genetic algorithm

Genetic algorithms work with direct solutions, they select the best ones, reproduce them and change them to find new solutions, so there is the obvious need to code the information I want as a solution into a chromosome, which in this case is a string of binary digits.

There are two things that need to be coded, the first one is the sequence of operations that are going to be done to the image, there is going to be two operations (erosion and dilation) and the possibility of doing nothing, and for each operation there is a binary code of two digits, I need at least two digits to code three different elements. The representation of the different operations is In Table 1 . There is no need to indicate which operations are first, the position in the chromosome serves as the position in which the operations are done. In Figure 19 there is an example of the translation of operations into the chromosome.

Operation	Binary code
Do nothing	00
Do nothing	01
Erosion	10
Dilation	11

Table 1: How the different operations of the solution are coded.

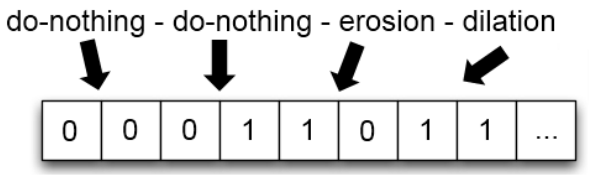

Figure 19: Sequence of operations translated into a chromosome.

The last thing to be coded is the size and shape of the structural element, it is going to be coded into three binary digits, allowing until eight different combinations of sizes and shapes, the codification for the structural element is in Table 2. In Figure 20 you can see the translation of a 5x5 fuzzy structural element with a 1x1 support area into a two dimensional array. In Figure 21 there is an example of a complete chromosome.

Size	Support area	Binary code
4x4	2x2	000
5x5	1x1	001
5x5	3x3	010
6x6	2x2	011
6x6	4x4	100
7x7	1x1	101
7x7	3x3	110
7x7	5x5	111

Table 2: How the different sizes and shapes of the structural element are coded.

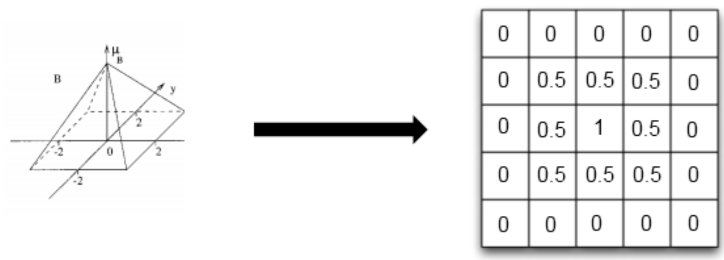

Figure 20: A 5x5 fuzzy structural element is translated to a two dimensional array. The fuzzy structural element has been taken from [1].

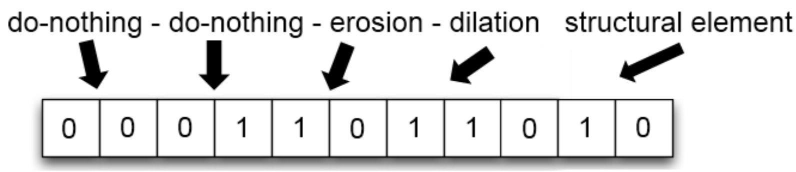

Figure 21: Example of a complete chromosome.

4.3 Structure and properties of the genetic algorithm

The structure and the properties of the genetic algorithm will define how it works, so in this section I'm going to show the different aspects of my implementation.

4.3.1 Population and initialization

How big is the number of chromosomes in the population will affect the running time to initialize the entire population or the selection of better individuals, and apart from this if we have a many individuals and we select only a few, we are just consuming memory saving a lot of individuals that have low fitness and are not going to be selected, in fact they are going to be replaced. As the number of individuals that are going to be selected as the best ones is going to be small to save running time in the experiment, genetic algorithms could get too complex thus wasting a lot of efficiency, then the population will count with ten individuals.

A random binary number generator will initialize each chromosome of the population at the beginning of the algorithm.

4.3.2 Fitness function

After creating the initial population or introducing the new individuals from the reproduction, the genetic algorithms needs to calculate their fitness to know which solutions are better than the others.

The objective of this procedure is, as said previously, to filter the image and get rid of the noise, so if we have the original image, the one without noise, and the corrupted image, we can indeed calculate the difference between the filtered one and the corrupted one, so first each chromosome will apply its sequence with its particular structural element to the image to get the filtered image by that particular chromosome. With the filtered image, the corrupted and the original then it's possible to define a fitness function as follows:

$$F = d(C, O) - d(I - O)$$

where C is the corrupted image, O is the original and I is the filtered one and $d(C,O)$ is the distance between the corrupted image and the original the same for $d(I,O)$. Note that if the result of this equation is negative then the filtered image is worse than the corrupted one and the solution should be replaced.

4.3.3 Selection

For reproduction the algorithm selects the four best individuals of the population, those are the ones with greater fitness, from this group of four the algorithm will select two different individuals each time that will be use as parents for two new individuals, the resulting individuals which will later replace the worst four individuals of the whole population.

4.3.4 Crossover

Once two parents are selected to create the new individuals there is a probability of 0.75 to produce this new individuals with crossover, if not then the two individuals will be indentical to their parents. The crossover works as follows, with the two parents selected the algorithm choose one position, from this position the parents are separated in two and then the part to the right from one parent is exchanged with the part from the other parent. An example of crossover can be seen in Figure 22.

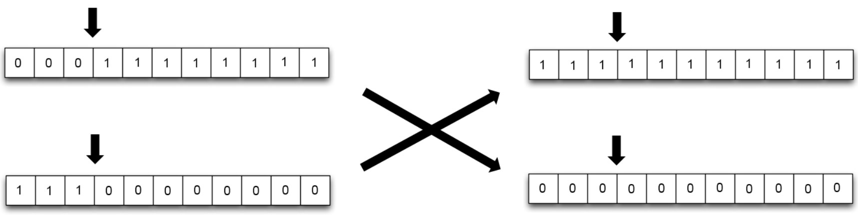

Figure 22: Example of crossover between two chromosomes.

4.3.5 Mutation

After the previous step every digit of the new individual have a probability of 0.10 to be flipped, this means that the 0s will be transformed into 1s and the 1s into 0s. The high probability of mutation enhances variability in the offspring, as I'm going to use a relatively small number of iterations for a genetic algorithm this will aid in the search of totally new solutions.

4.3.6 Stopping criterion

The algorithm will stop when it reach a certain number of iterations, in this case 100 and 250 iterations, for two different tests, this way I will have a better way to see how the genetic algorithm behaves.

4.4 General procedure of the experiment

The experiment will be separated into four steps:

1. Add noise to the selected images and binarize them through thresholding.
2. Run the genetic algorithm for each image with fuzzy mathematical morphology.
3. Run the genetic algorithm for each image with classical mathematical morphology.
4. Use the results of the fuzzy mathematical run to test with images different than the ones used on train.

4.5 Results

I've tested the algorithm with various images, the most significant results were found with three particular images, the first one can be seen in Figure 23, it has simple objects and forms, the second one it's in Figure 24 it has more complicated objects and numbers and in Figure 25 there is the last image which have more complicated forms. I've applied random noise to all of them, 5% speckle noise, the results can be seen in their respective figures.

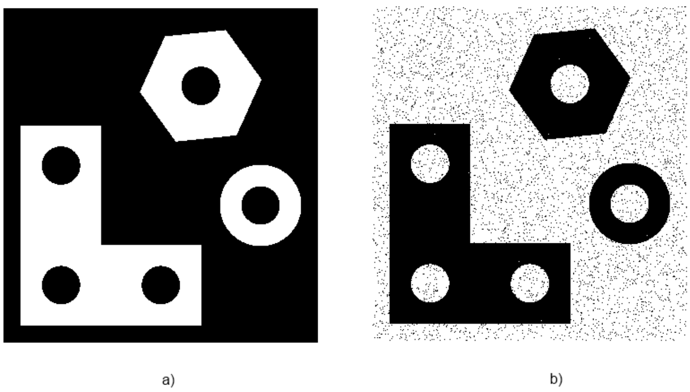

Figure 23: First image, "Parts.tif". a) Original image b) Corrupted one

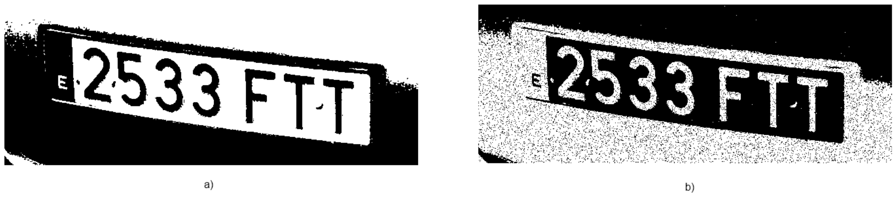

Figure 24: Second image, “Matricula.tif”. a) Original image b) Corrupted one

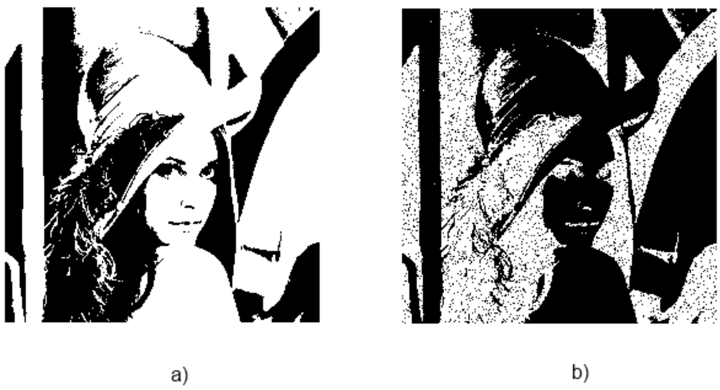

Figure 25: Last image, “lena.tif”. a) Original image b) Corrupted one

In Table 3 and Figure 26 there are the results for the image “Parts.tif” with both fuzzy and classical morphology, in Table 4 and Figure 27 there are the results for the image “Matricula.tif” and finally in Table 5 and Figure 28 there are the results for “lena.tif.”

In the tables “Morphology” indicates the type of morphology of that particular case, “Iterations” refers to the number of iterations used for test, “Fitness” is the result of the fitness equation defined previously, “Sequence” is the sequence of operations of the solution, “dn” is “do nothing”, “d” is dilation and “e” is erosion, “SE” is the size (and shape) of the selected structural element and “Improvement” is the percentage of improvement of the filtered image over the corrupted one defined by this function:

$$Average = \frac{(C - I)}{(C - O)} * 100$$

where C is the corrupted image, O is the original and I is the filtered one. For example, if C has 10 pixels of noise and I only 5 then the result would be 50%.

Morphology	Iterations	Fitness	Sequence	SE	Improvement
Fuzzy	100	4452	dn-d-d-dn	7x7 (5x5)	96.99%
Classical	100	4536	e-d-dn-dn	6x6	98.82%
Fuzzy	250	4575	dn-e-d-dn	6x6 (2x2)	99.67%
Classical	250	4536	dn-e-d-dn	7x7	98.82%

Table 3: Results of the experiment for the image “Parts.tif”

Morphology	Iterations	Fitness	Sequence	SE	Improvement
Fuzzy	100	5655	dn-dn-e-d	5x5 (3x3)	76.14%
Classical	100	5459	dn-dn-d-e	6x6	73.5%
Fuzzy	250	5947	e-d-d-e	5x5 (1x1)	80.07%
Classical	250	5783	dn-dn-e-d	7x7	77.86%

Table 4: Results of the experiment for the image “Matricula.tif”

Morphology	Iterations	Fitness	Sequence	SE	Improvement
Fuzzy	100	1079	dn-e-e-d	6x6 (2x2)	39.7%
Classical	100	-267	dn-dn-d-e	4x4	-9.82%
Fuzzy	250	1263	d-dn-e-d	4x4 (2x2)	46.47%
Classical	250	-267	d-e-dn-dn	4x4	-9.82%

Table 5: Results of the experiment for the image “lena.tif”

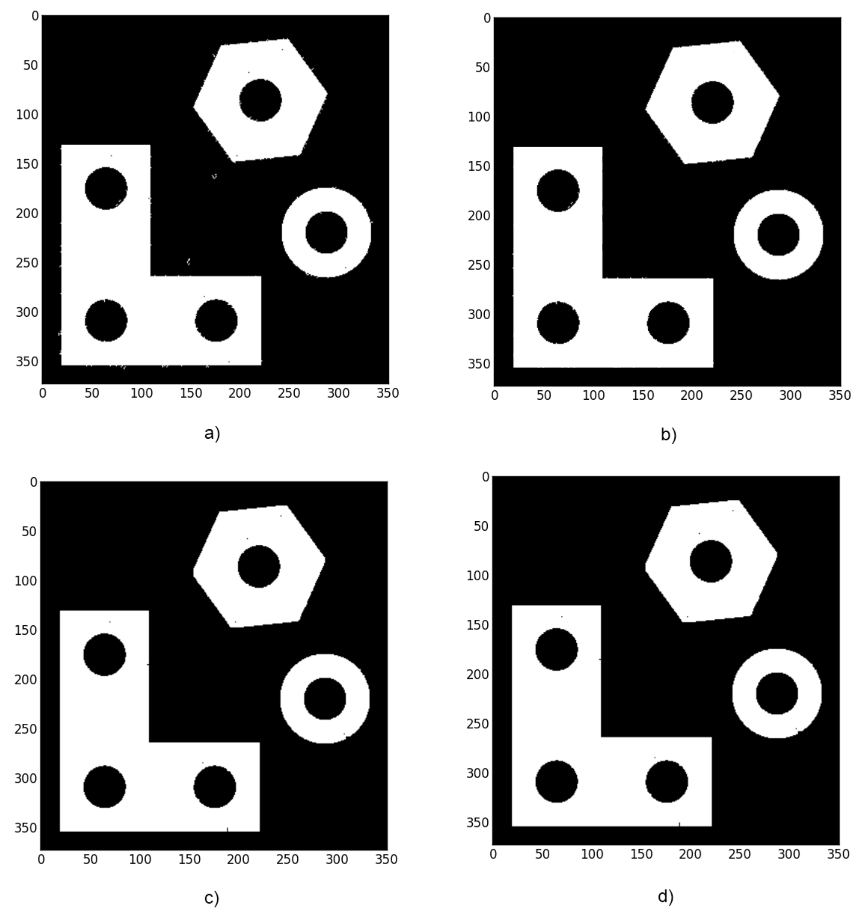

Figure 26: Results for the image "Parts.tif".a) Fuzzy morphology with 100 iterations b) Fuzzy morphology with 250 iterations c) Classical morphology with 100 iterations d) Classical morphology with 250 iterations.

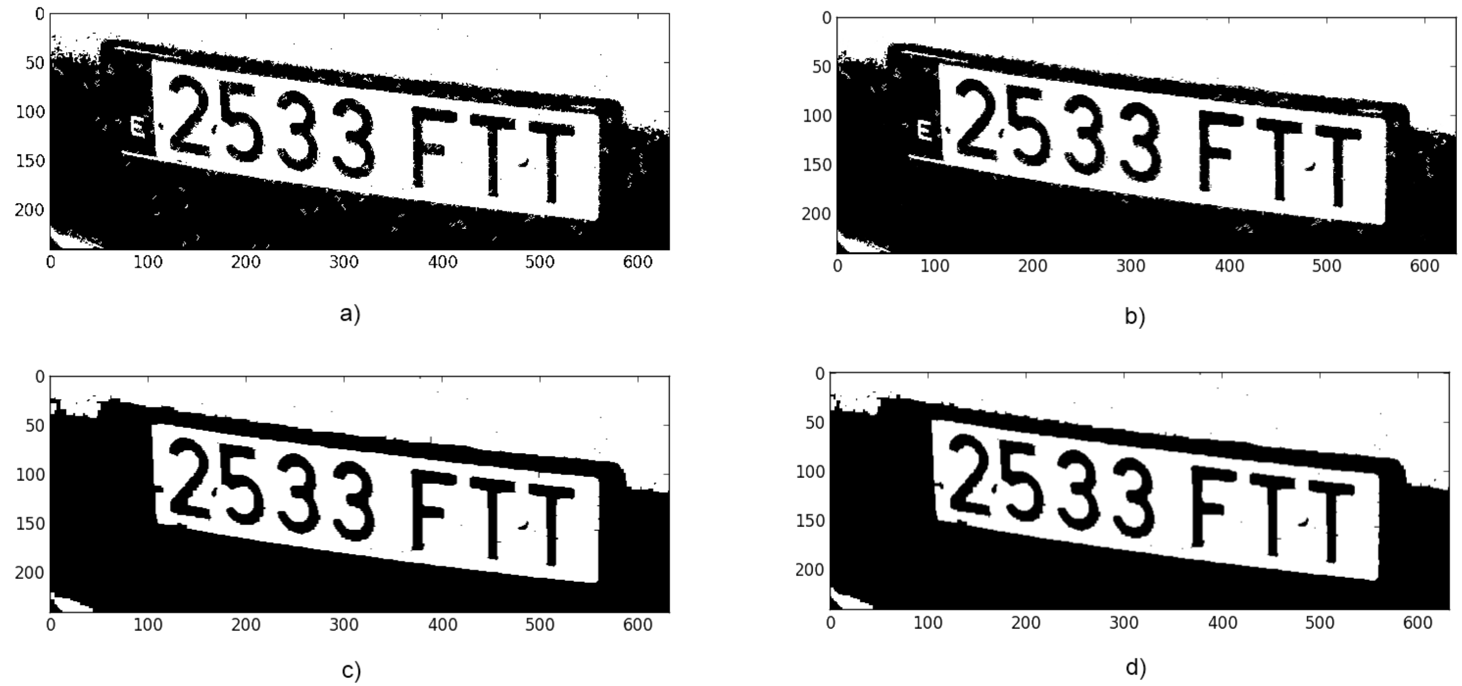

Figure 27: Results for the image “Matricula.tif”.a) Fuzzy morphology with 100 iterations b) Fuzzy morphology with 250 iterations c) Classical morphology with 100 iterations d) Classical morphology with 250 iterations.

Figure 28: Results for the image “lena.tif”.a) Fuzzy morphology with 100 iterations b) Fuzzy morphology with 250 iterations c) Classical morphology with 100 iterations d) Classical morphology with 250 iterations.

After getting these results they were tested with three different images, “Girl2.tif”, “House.tif”, “Lisa.tif”, because of their similarity to “lena.tif”, “Girl2.tif” and “Lisa.tif” were tested with the results obtained from that image with 250 iterations and fuzzy morphology, “House.tif” was tested with the results of “Matricula.tif” with 250 iterations and fuzzy morphology. The results of this second part of the experiment can be seen in Table 6 and Figures 29, 30 and 31.

Image	Fitness	Sequence	SE	Improvement
“Girl2.tif”	854	d-dn-e-d	4x4 (2x2)	44.5%
“House.tif”	417	e-d-d-e	5x5 (1x1)	36.74%
“Lisa.tif”	773	d-dn-e-d	4x4 (2x2)	44.88%

Table 6: Results of the test images.

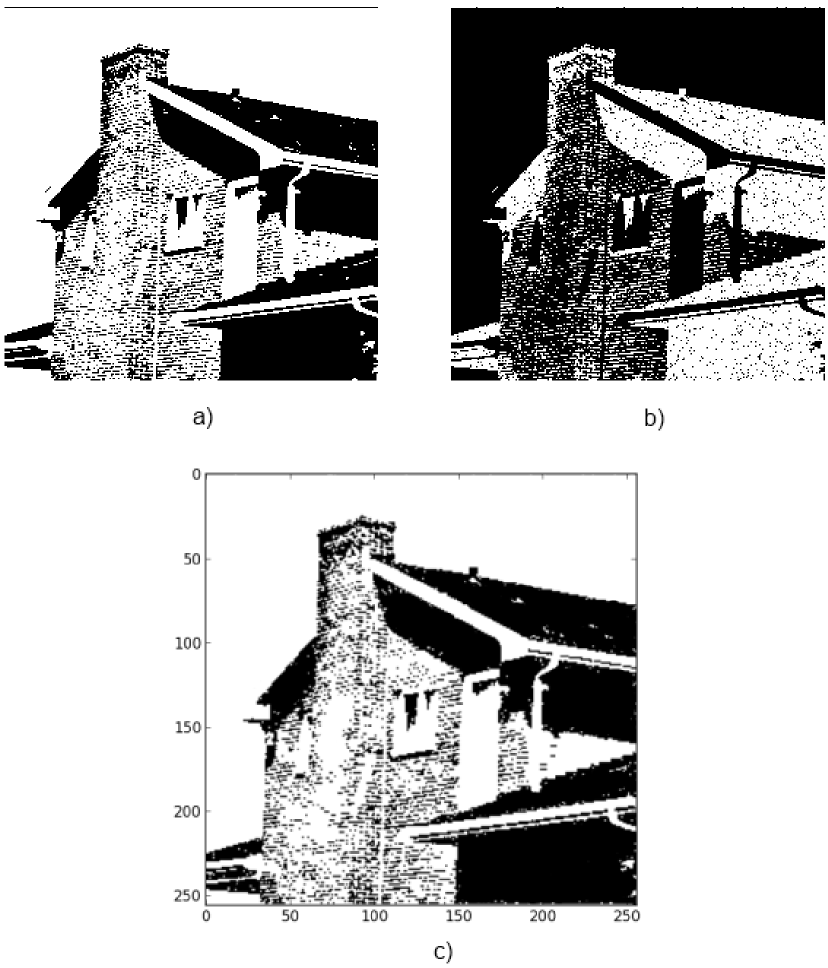

Figure 30: Results for the image “House.tif”.a) Original image b) Corrupted image c) Fuzzy morphology with 250 iterations.

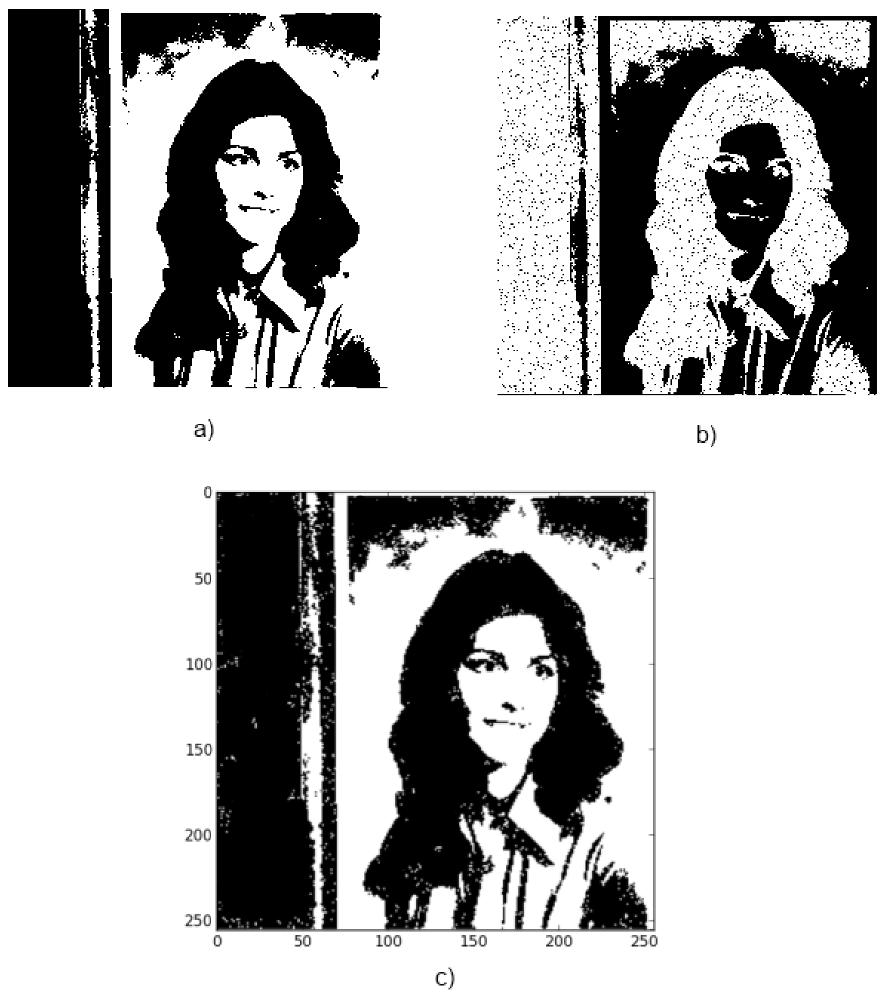

Figure 29: Results for the image “Girl2.tif”.a) Original image b) Corrupted image c) Fuzzy morphology with 250 iterations.

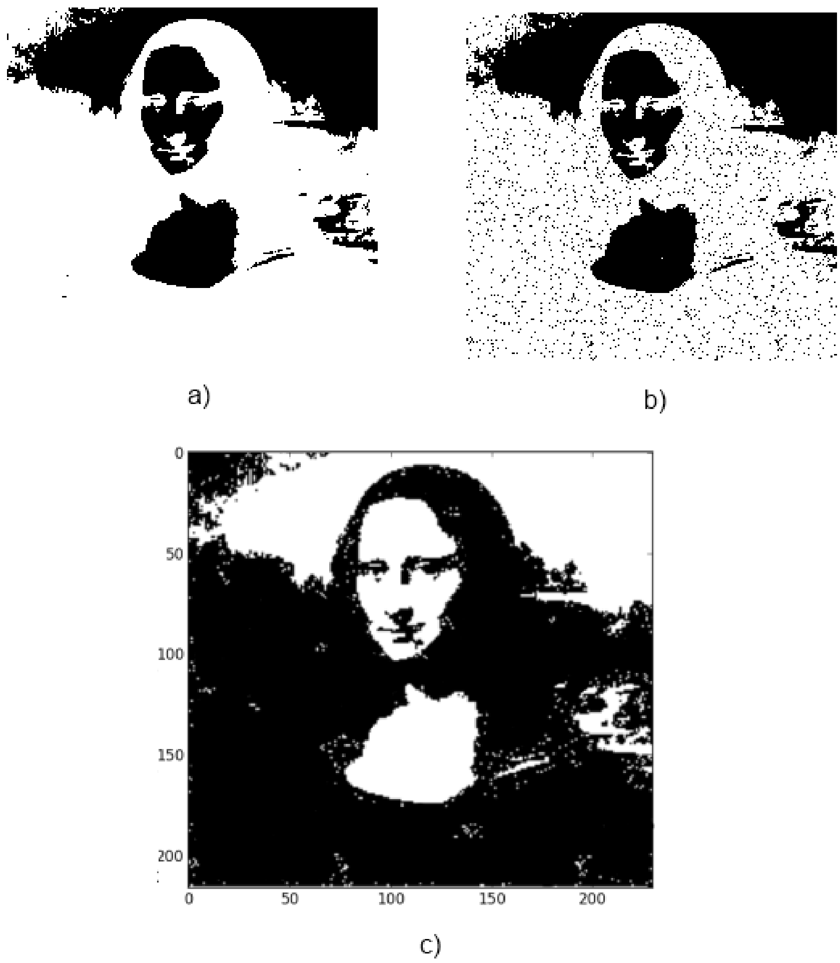

Figure 31: Results for the image “Lisa.tif”.a) Original image b) Corrupted image c) Fuzzy morphology with 250 iterations.

5 Future work

In this section I'm going to present two alternatives that didn't get to the final experiment, the first one are different shapes of structural elements an how to code them, the second one is translating the image directly to a fuzzy set.

5.1 Structural element shapes

In my experiment I only use square shaped sctructural elements, changing size and only the shape of the support area, but it could be really a good alternative to change the shape of the structural element as hinted in [11], for example they demonstrate that rectangular structural elements like the one I use are good for when the image has objects with several corners, elliptical structural elements when the objects have few corners and circular when neither of those conditions are met, or just as an standar structural element.

To code a different shape of structural element the 2D array representing it will have a null value that will indicate that that place is empty and it's not part of the structural element. In figure 29 I show an example, taking as null value "-1".

-1	-1	0	-1	-1
-1	0	0.5	0	-1
0	0.5	1	0.5	0
-1	0	0.5	0	-1
-1	-1	0	-1	-1

Figure 32: Example of circular structural element, with "-1" as null value.

Apart from this the only thing left to do would be introduce a new parameter into the chromosome, to do this I would insert two new digits so I could code three different possibilities as shown in Table 6.

Shape	Binary code
Rectangular	00
Rectangular	01
Elliptical	10
Circular	11

Table 7: Coding different shapes for the structural element.

5.2 Images into fuzzy sets

Another approach to the Fuzzy mathematical morphology is using gray-level images transfromed into fuzzy sets instead of binary images, this can be done fuzzifying the gray-level image, as a gray level pixel is already a grade of gray it's really easy to normalize every pixel into the range [0,1], thus acting like a membership degree.

The process then is very simple:

1. Load the image into the program.
2. If the image is in RGB transform it to gray-levels.
3. Normalize the pixels of the image into the range [0,1]
4. Certain libraries use fuzzy sets as 1D arrays, in that case transform the 2D array into a 1D array and then convert it into a fuzzy set.

6 Conclusions

Fuzzy mathematical morphology has proven to be a usefool tool to give more robustness to the classical operations of morphology, in the experiment is obvious that with basic images fuzzy and classical morphologies perform almost as well and maybe classical morphology is more efficient in terms of time and resources, anyway when dealing with noise in more complicated images like “Matricula.tif” or “lena.tif” fuzzy morphology achieves to eliminate noise without changing too much the image, while classical morphology changes the image too much and could end up being worse than the original image corrupted with noise, as you can see in the results of “lena.tif”.

As for the addition of genetic algorithms to the equation, they can consume a lot of resources, but they’ve demonstrated to be very useful as it’s obvious that different structural elements and different sequences of operations are selected for each image, I want to remark that, in the results, opening and closing appear several times, which was expected knowing the results of applying both operations.

Another interesting result is that for the image “lena.tif” the structural element has always an even size, as shown before these structural elements have support areas usually bigger than the odd sizes, so it could be possible than with objects with a lot of curves and few hard corners, even-sized structural elements are better.

The test images also shown overall good results, they were not perfect and in some cases part of the original image were even erased like in “House.tif” but it seems that knowing how to filter one image can help us filtering simillar images, or at least give close enough results.

7 References

[1] "A Generalized Fuzzy Mathematical Morphology and its Application in Robust 2-D and 3-D Object Representation". Vassilios Chatzis, and Ioannis Pitas. 2000

[2] "The use of genetic algorithms in morphological filter design". Neal R. Harvey, Stephen Marshall. 1994

[3] "Fuzzy sets and Systems". L.A. Zadeh. Elseveir 1978

[4] "Adaptation in Natural and Artificial Systems, University of Michigan Press". Holland, John H . Ann Arbor 1975

[5] "Mathematical Morphology and its Applications to Image Processing". J. Serra and P. Soille. Norwell, MA: Kluwer, 1994

[6] "Soft morphological filters". L. Koskinen, J. Astola, and Y. Neuvo in Proc. SPIE Symp. Image Algebra Morphological Image Processing, vol. 1568, 1991.

[7] "Statistical properties of soft morphological filters". L. Koskinen and J. Astola in Proc. SPIE Symp. Nonlinear Image Processing, vol. 1658, 1992.

[8] "Convexity indicators based on fuzzy morphology". A. T. Popov. Pattern Recognit. Lett., vol. 18, pp. 259–267, 1997.

[9] "Fuzzy mathematical morphology: concepts and applications". Maria Concetta Maccarone. IFCAI/CNR,Palermo,Italy, 1996

[10] "Image Analysis and Mathematical Morphology".J. Serra. Academic Press, London, 1982.

[11] "Using a Fuzzy Morphological Structural Element for Image Segmentation". P. Sobrevilla, E. Montseny, J. Kelle. 2000

[12] "Segmentation of Medical Images using Fuzzy Mathematical Morphology" Bouchet, A., Pastore, J. and Ballarin, V. Measurement and Signal Processing Laboratory, School of Engineering, 2007

[13] "Fuzzy Mathematical Morphology approach in Image Processing". Yee Yee Htun, Dr. Khaing Khaing Aye. World Academy of Science, Engineering and Technology 42 2008

[14] "General approach for fuzzy mathematical morphology". Antony T. Popov. Proceedings of the 8th International Symposium on Mathematical Morphology, Rio de Janeiro, Brazil, Oct. 10–13, 2007

[15] "Fuzzy mathematical morphology and its applications to colour image processing". Antony T. Popov.

[16] "THE APPLICATION OF GENETIC ALGORITHMS TO FEATURE RECOGNITION AND OBJECT COUNTING". M S White, E R Davies and S J Flockton. 1994.

[17] "COLOR IMAGE SEGMENTATION BASED ON FUZZY MATHEMATICAL MORPHOLOGY". Aymeric Gillet, Ludovic Macaire, Claudine Botte-Lecocq and Jack-Ge'rard Postaire. 2000.

[18] "Image Similarity: A Genetic Algorithm Based Approach". R. C. Joshi, and Shashikala Tapasw. 2007.

[19] "Fuzzy sets". L. Zadeh. IEEE Inform. Contr., vol. 8, no. 3, pp. 338–353, 1965.

[20] "Convexity indicators based on fuzzy morphology". A. T. Popov. Pattern Recognit. Lett., vol. 18, pp. 259–267, 1997.

[21] "Fuzzy mathematical morphologies: A compar-ative study". I. Bloch and H. Maitre. Pattern Recognit., vol. 28, pp. 1341–1387, Sept. 1995

[22] "Vision". David Marr. 1982

[23] "Genetic Algorithms in Search Optimization and Machine Learning". Goldberg, David E. Addison Wesley, 1989

[24] "Genetic Algorithms + Data Structures = Evolution Programs". Michalewicz, Zbigniew.Springer-Verlag, 1999