Computer Science Journal of Moldova, vol.**12**, no.2(35), 2004

A Pseudo-Statistic Edge Detector using Fuzzy Computation

Horia-Nicolai Teodorescu, Frédéric Mayer

Abstract

We present and analyze a new edge detector algorithm using a neuro-fuzzy architecture. The analysis also includes a comparison of the best known algorithms for edge detection.

1 Introduction

The aim of this paper is to present and analyze a new approach in edge detection, namely a pseudo statistical fuzzy filter and its capabilities. The edge detection is a complex task of the image analysis process. Many edge detectors exist and the results they achieve are not always satisfactory. This is partly due to the variety of criteria that cab be adopted in judging the quality of the edge detection process. Edge detection is an application-driven process, optimality is subjective in this process. We deal with medical radiographic images only, and the subsequent optimality criteria. The optimality criteria in this field are quite specific and their influence is clearly remnant in our analysis.

The edge detection is an important step of the process that permits to understand the image content [EFF-00]. An edge is characterized by the variation of the gray-level intensity between adjacent pixels [BOW-02]. This variation can be more or less distributed over a continuous area: this makes the difference between step or line edges.

There are numerous methods to detect edges reported in the literature. Among these methods are included the classical Laplace and Sobel operators that use the first and second derivative of the image, neural network-based approaches, e.g. Bezdeck [BEZ-94], Cohen

[COH-97], the mathematical treatment of the information with the Canny filter [CAN-86], and the Infinite Symmetric Exponential Filter (ISEF). Fuzzy systems have also been implemented as Competitive Fuzzy Edge Detectors (CFED) [LIA-03]. Some recent researches associated the segmentation and the edge detection processes [BEL-98]. For medical images, the main difficulties encountered when detecting edges are [SAL-96] sensibility to noise, detecting small gradient edges, detecting curve corners as well as line edges, and preserving lines continuity. Fortunately, in case of medical images, we don't have to manage the multiple shadows that are due to different light source exposures, like in the general case.

In edge detection, several categories of image filter are used: differential, statistical and fuzzy. Each of them has some limits and drawbacks - accuracy, computational burden/ complexity, statistical properties, and noise immunity.

The Sobel and Laplace operators generally give good results in edge detection [MAR-00], [SAL-96], but they can miss precision and continuity in the layout of edges. The Sobel operator is not isotropic, i.e. its sensitivity depends on edge orientation. It is very effective to detect the vertical and horizontal edges but can miss some corners, round or diagonal edges. It is also sensitive to noise.

One of the main disadvantages of the Laplacian filter is that it amplifies the noise, polluting the image with some speckle points. If first derivatives are sensitive to noise, the second difference derivatives can really misunderstand some polluted information generating a wrong result. Noise should be reduced before edge detection is performed. Indeed, as the Laplacian operator is based on zero crossing, it produces a one-pixel boundary and situates the edge in the place of the larger luminosity variation. In case the original edge gradient variation is not uniform, the Laplacian operator could produce double edges. Preprocessing the image with a Gaussian filter decreases the sensibility to noise, but also reduces the precision of the detection because it changes the edge gradient variation.

When a Gaussian filter is applied on the image before the edge detection, it reduces the sensibility to the noise, but some information

is also lost: fine details may become insignificant and the location of the edges can be misinterpreted. The quality of edge detection is reduced when using such a filter on a clear image with sharp edges.

Usually, another step is added after the edge detection: Once the derivative has been calculated, this step is used to threshold the data and determine where the results suggest an edge is present. The lower the threshold is, the more lines will be detected, and the results will become increasingly sensitive to noise and may also pick out irrelevant features from the image. Conversely, a high threshold may miss subtle lines or sections of lines.

The Canny operator is efficient in detecting most of the edges but it can miss to connect some Y-junctions. It produces a fine edge and is not very sensitive to noise. According to [LIA-03], the Competitive Fuzzy Edge Detector method is much quicker than the Canny method. The Canny filter, as the Sobel and Laplace operator when they are used with a post-treatment threshold, must be set by an operator to adapt the result of the edge detection to the quality of the source image and to the requirements of the application.

The complexity of these edge detectors is variable: the Canny or CFED apply some other steps to upgrade the result of the edge detection. The algorithm tested here is closer to basic Laplace or Sobel. For this reason, we will compare the obtained results to those of obtained with these last filters, which have similar complexity.

A comparison between the algorithms should be made regarding to the complexity of these algorithms. One can determine two classes of algorithms: the simplest ones to which Laplace and Sobel belong, and the more complex ones to which CFED and Canny belong. The Laplace and Sobel are respectively simple or double convolutions. The neuronal approach cannot be compared in terms of performances, because it works differently: the learning phase is long [COH-97], but the computation phase is very fast.

Fuzzy theory is useful in modeling and reasoning with uncertainty. This seems to be the case in the edge detection problematic. Indeed, the edge detection problem is an example of an uncertain information treatment. The fuzzy logic provides a mathematical framework to the

representation and processing of expert knowledge [TIZ-98]. Actually, the edge detection is linked to an expertise: none of the existing filters is able to detect the edges without settings provided by an expert. The fuzzy logic is already used in the image analysis field [TIZ-97]. Some fuzzy filters have already been studied to filter the noise in an image [VIL-03]. Image content understanding systems were implemented with fuzzy logic [KOP-96]. In [GAL-00], the filter tries to detect edges on noisy images. Edge detection results with fuzzy filter were also presented in [BAU-02] or more recently in [KIM-04]. The last edge filter is especially interesting because it tries to detect the edges without any threshold adjustment. We also note that this method is very fast.

The edge detection on a simple image (a cube on a plan for instance) is easy, but in many cases the image is much more complex and the border between the objects cannot be clearly defined. The medical images have specific characteristics: they have low contrast, low definition and much noise. If we ask several specialists to draw the edges on an image, then we may obtain very different interpretations: their drawings will be very subjective. Therefore, the use of the fuzzy logic could be a good way to solve this kind of problem.

To compare the capacities of this filter, and to determine its best settings, we need to be able measure the correctness of the computed edge maps. In the X section, we explain how we will assess the resulting edge maps: which tools are used and to which reference the comparison is made. The Y section will present the different algorithms we use to search the best settings combination for the filter. In the Z section, the results are compared with the capabilities of other edge filtering methods.

2 Principles of the fuzzy-statistical filter

2.1 Background

Deterministic filters deal with local pointwise information, where information of every pixel enters a formula with some fixed coefficients to generate the gray level of the central pixel.

In the operation of the statistical filters, only averaged information over all pixels in some predefined region around the currently computed pixel is important. A statistical filter can take into account the overall information, on the whole image, or the statistically local information, in some restricted region of the image. The restricted region is generally specified as a rectangular or square "window" centered in the desired (processed) point of the image.

A typical statistical filter takes into account the average and the spreading of the values in some window. Higher order moments are generally skipped. This means that only some 'essential' part of the statistical information is used in the processing.

One of the problems with existing, intensity-based, differential edge detectors is that they are unable to detect edges that have gray levels close to the gray level of the background. The gradient- and difference-based Sobel, Roberts and Prewitt filters perform poorly when the gray intensity changes lightly from the edge to the background. One of the reasons is that these operators are fixed, non-adaptive. However, even adaptive filters are confused by the small difference in gray levels in the window where they operate, because they can not compare the window properties with the surrounding image. To alleviate this drawback, we propose the use of two windows, allowing us to determine the information on the background and to contrast it with the current window information. Specifically, if the current pixel is close to both the background and the average in vicinity, yet it may be an edge pixel. The reason to use a fuzzy system in the edge detection is that this system represents a non-linear mapping of the brightness scale. Moreover, it allows interpolation of any non-linear mapping according to the universal approximation theorem. This, in principle at least, allows us any mapping of the brightness scale, mapping that would permit optimum edge detection, according to some given criteria.

2.2 Description of the Neuro-Fuzzy Edge Detector

The basic operation of the edge detector is presented in the sequel. Firstly, the edge detector computes in every point of the image the

gradients on a small and a large window. Then, a fuzzy classifier evaluates the gradients. The classifier is built with a set of three classes, a rule engine, and a Sugeno inference system with defuzzification. The system provides a correction on the original pixel value. The resulted value is then normalized and a filter is applied to obtain a final binary information about the edge property value. The system scheme is sketched in Fig. 1.

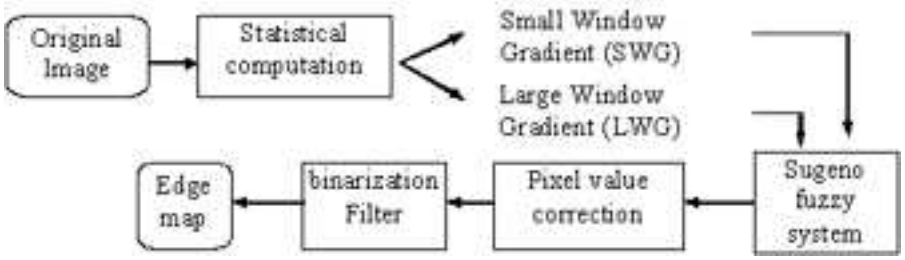


Figure 1. Fuzzy edge detector scheme

The filter is set with seven thresholds for the input membership functions; the parameters for each rule p, q, r and the parameters of the output filter.

Two methods have been implemented to find the best settings of the filter. The first method systematically tries all the values for the seven thresholds, while the other parameters are arbitrary fixed. The second method is based on a genetic algorithm (GA). To simplify the computation, we arbitrary set each p_k and q_k with the null value. This method is described in Section 5.

2.3 Filter analytical equations

Procedure "fuzzy pseudo-statistical filter"

This procedure has been introduced by the first author in the frame of an internal research [TEO-04]. We denote the brightness of the pixel in point (i, j) by $s_{i,j}$. The main idea and notations are as follows.

Instead of using a single window, two windows are used for the characterization of the local image properties. The two windows will subsequently be named "small window" (SW) and "large window" (LW). We compute the average brightness in the two windows,

$$\bar{s}_{i,j}^{SW} = \frac{1}{N_{SW}} \sum_{(i,j)\in SW} s_{i,j}, \quad \bar{s}_{i,j}^{LW} = \frac{1}{N_{LW}} \sum_{(i,j)\in LW} s_{i,j}.$$

We compare $s_{i,j}$ with the averages. Precisely, we compute the distances between the brightness of the current pixel and the average brightness in the two windows,

$$d\left(s_{i,j}, \bar{s}_{i,j}^{SW}\right), \quad d\left(s_{i,j}, \bar{s}_{i,j}^{LW}\right).$$

These distances can be interpreted as the statistical brightness gradient in the corresponding windows. Finally, we evaluate the distances using fuzzy degrees. We assign three fuzzy degrees, "small" (S), "zero" (Z), and "large" (L) to these distances. We assign triangular membership functions to the fuzzy degrees (see Fig. 3 and Fig. 4). Notice that, in case of an optimization process, the rules, the number of membership functions, and their shape are parameters to be optimized. With these preliminaries, we are prepared to describe the edge detection procedure.

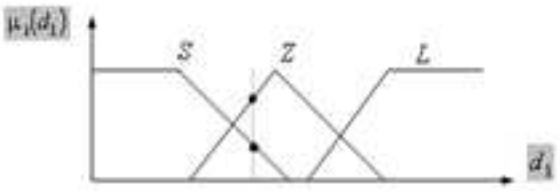


Figure 3. Small window gradient fuzzy evaluation

Filtering procedure

The procedure applies corrections to the brightness of the pixel, depending on the two distances above, i.e., according to the "oddness"

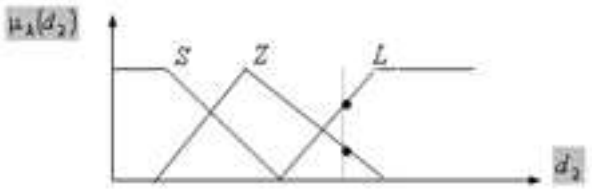


Figure 4. Large window gradient fuzzy evaluation

of the pixel with respect to the pixels in a close and in a larger vicinity. The correction is applied as:

$$s_{i,j} \leftarrow s_{i,j} + \Delta$$

We characterize the increment by linguistic degrees and we assign triangular fuzzy membership functions to these degrees. Then, the correction is determined by rules in the form:

If $d\left(s_{i,j}, \bar{s}_{i,j}^{LW}\right)$ *is Large and* $d\left(s_{i,j}, \bar{s}_{i,j}^{SW}\right)$ *is Large Then brightness correction* Δ *is Large Positive* $(+L)$.

The table determining the correction rules is like Table 1 and Fig. 5:

Table 1. Defuzzifier rules

	$d\left(s_{i,j}, \bar{s}_{i,j}^{LW}\right)$		
$d\left(s_{i,j}, \bar{s}_{i,j}^{LW}\right)$	S	Z	L
S	+ L	+ S	+ Z
Z	+ S	Z	- S
L	Z	- S	- L

To simplify computations, a 0-type Sugeno system is used to implement the rules (see Fig. 6). Finally, the result is defuzzified and the correction is applied.

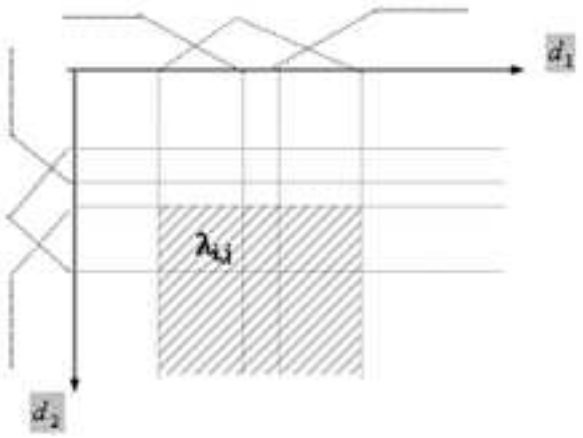


Figure 5. Defuzzifier principle

2.4 Filter statistical properties

The fuzzy filter is a non-linear system. The image mapping is, with respect to the brightness variable, a rational function on every interval defined by the membership functions.

The 0–order Sugeno system equations are:

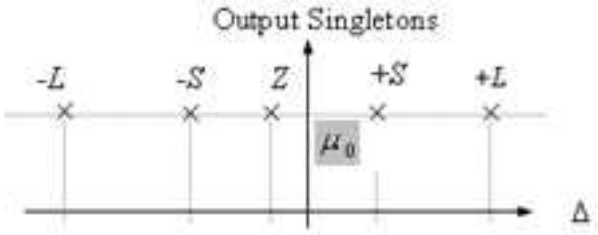


Figure 6. Defuzzifier correction principle

$$\mu_{0ij}(\Delta) = \begin{cases} \min\ (\mu_i(d_1), \mu_j(d_2)) & if\ \Delta = \lambda_{ij} \\ 0 & elsewhere \end{cases}$$

$$\Delta(d_1, d_2) = \frac{\sum_i \sum_j \min\ (\mu_i\,(d_1)\,,\ \mu_j\,(d_2)) \cdot \lambda_{ij}}{\sum_{ij} \min\ (\mu_i\,(d_1)\ ,\ \mu_j\,(d_2))}$$

$$\lambda_{ij} \in \{-L,\ -S,\ Z,\ +S,\ +L\}$$

The binarization filter operates according to:

$$s_{ij}^e = \begin{cases} 1 & if \quad s_{ij}^* > \theta \\ 0 & elsewhere \end{cases}$$

where θ is a threshold. The θ value is one of the parameters to optimize during the training procedure.

The first-order Sugeno system equations are:

$$x_{ij} = f\ (d_1,\ d_2) = p_i \cdot d_1 + q_j \cdot d_2 + r_{ij}$$

and

$$\mu_{0ij}(\Delta) = \begin{cases} \min\ (\mu_i\,(d_1)\,,\ \mu_j\,(d_2)) & if\ \Delta = x_{ij} \\ 0 & elsewhere \end{cases}$$

2.5 Algorithm for filter implementation

The algorithm, in pseudo code, is:

Procedure FUZZY_FILTER_1
1. for i=0 to N
2. for j=0 to M
3. apply procedure CORRECTION
4. $S_{i,j} \leftarrow S_{i,j} + \Delta$

END FUZZY_FILTER_1
Procedure CORRECTION
1.Compute $\bar{S}_{i,j}^{LW}$
2.Compute $\bar{S}_{i,j}^{SW}$

3.Compute $d(S_{i,j}, \bar{S}_{i,j}^{LW})$
4.Compute $d(S_{i,j}, \bar{S}_{i,j}^{SW})$
5.Compute Δ' by the FUZZY_CORRECTION procedure.
END CORRECTION

2.6 Filter algorithm complexity

Algorithmic complexity of the basic fuzzy filter

Statistical filters are reputedly complex, due to the larger number of multiplications involved. Assuming a $N \times M$ image and a $K \times L$ window, linear filter operation requires a number of multiplications and additions of the order of $N \times M \times K \times L$. A typical statistical filter using the same window requires $N \times M \times K \times L$ additions to compute the averages p_{ij} in the window, the same number of substractions to compute conditions $p_{ij} - \overline{p_{ij}}$, an equal number of multiplications to compute $\sum (p_{ij} - \overline{p_{ij}})^2$and $N \times M$ divisions. With suitable choices of the variable, linear filters perform operations with integers only, while the statistical filters require float (real) variables further increasing the computational burden.

Further compounding the algorithmic complexity, the use of fuzzy logic requires $N \times M \times V$ operations to determine the membership functions (V is the number of membership functions of the Sugeno system) moreover $2 \times N \times M \times V$ additions and multiplications and $N \times M$ divisions to compute the output of the Sugeno system. All operations for fuzzy system are real-type variable operations.

Complexity of the training procedure

The training is aimed to choose the best parameters of the filter when operating on a set of Q representative images. Assuming S parameters to be adapted, each taking V possible values, the search space is S^V. The number of filtering operations is thus $Q \cdot S^V$. The same number of evaluations of the filtering results is required.

Sorting these results would require at least $O\left(Q \cdot S^V \log\left(Q \cdot S^V\right)\right)$ operations. This upper limit adds to the complexity of performing $Q \cdot S^V$ values. The second operation, i.e. sorting, dominates the first one for large S and V values.

3 Assessment of the results

3.1 Subjective vs. objective assessment

The assessment of the results of the training can be performed through a subjective process. This means to ask a pool of specialists to classify the images obtained from the filter. But depending on the used method, the produced number of images can be very large. Therefore, this process would be very long and it is not compatible with the idea of an automatic learning process. Such a manual process would be suitable to compare the result of detection between several algorithms, but not to classify images that might be very similar.

Because we need to automatically assess the results of the learning, we define an objective method that aims to measure the distance between an edge map built from the original image and an ideal map. This ideal map is established as the consensus of several drawings manually made by a pool of experts. Then a distance can be computed between them. In the next section, we present the reasons for the choice of a specific distance.

We chose five angiographical images with different characteristics of luminosity, low/high contrast blood vessel share, and vessel network complexity. They represent a statistically representative panel of the properties that are usually encountered in this class of images.

3.2 Reference Edge Map

We asked some bio-medical engineers to draw reference edge maps. They did it manually on a layer over the original image (see Fig. 7). The experts have been chosen to have enough knowledge to interpret radiographic images.

The obtained maps show results that look close one to the other, but comparing them in the details, we noticed some important differences.

For instance, if we focus on the circled zone, the experts proposed different interpretations, as shown in Fig. 8, namely:

- While the edges look similar in most regions, there are important differences (zone A) at the pixel level.

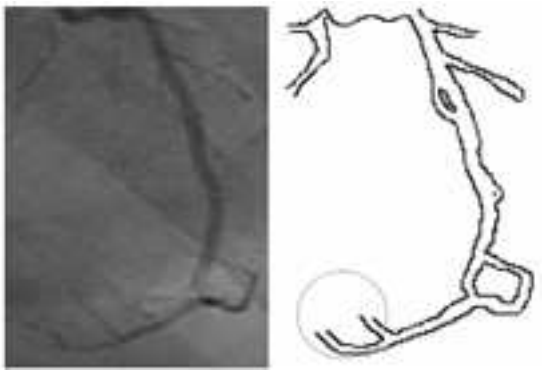


Figure 7. Edge map example

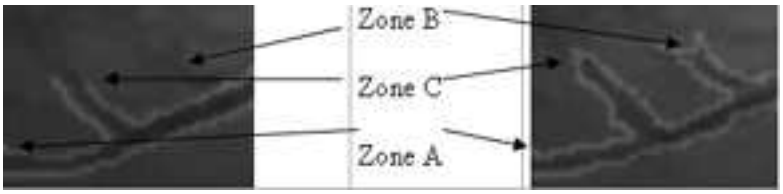


Figure 8. Edge map details

- Some details are clearly understood in different ways (zones B and C).

Concluding, after evaluation of these results, it seems that an ideal reference edge map can not be determined: the "true" edge map depends very much on the experts' interpretation. Nevertheless, it is possible to use the manual evaluations in two different ways:

- Establish a subjective choice on a map that defines the edges. Then, we consider pixels in the area in the neighborhood of an edge line with a smaller error.
- Establish an average edge map from all the drawn maps. This

edge map will then give a probability for a pixel to be on an edge. Then, this probability can be used to compute an error.

3.3 A distance based assessment

Here, we aim to determine the best choice for our application and the rational for optimization. Therefore, we first aim to determine which is the best measure of distance between an ideal edge map and the detected one. We denote by n_1 and n_2 the number of points wrongly classified in the background and edge classes respectively, M and N being the dimensions of the image. We also denote S^{EM} and S^{GT} respectively the detected edge map and the ground truth.

The Euclidean distance is frequently used to measure the noise between an image and a reference image. It is defined as:

$$d^2_{Euc} = \frac{1}{M \cdot N} \cdot \sum_{i=1}^{M} \sum_{j=1}^{N} \left(S^{GT}_{i,j} - S^{EM}_{i,j} \right)^2$$

that is:

$$d^2_{Euc} = \frac{1}{M \cdot N} \cdot (n_1 + n_2). \tag{3.1}$$

As we want to use this distance in a learning process on images with different characteristics, we will study the function between the distance, n_1 and n_2 parameters with different relative values. The Euclidean distance function is plotted in Fig. 9.

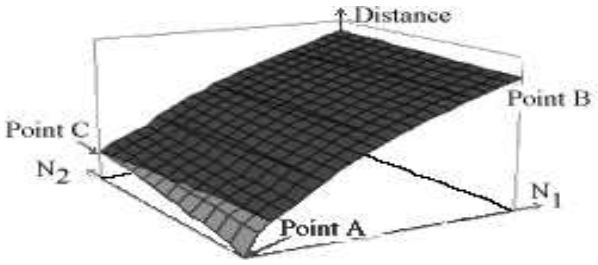


Figure 9. Distance function graph

There are three outstanding points in this graph:

- The point A shows the error when n_1 and n_2 are null; this is the perfect edge detection.
- The point B shows the error when n_1 is null; this is a white image where all points are detected as edges.
- The point C shows the error when n_2 is null; this is a black image where no edge points are detected.

The darkest surface represents the error cases for which the error is smaller than the error in point B and bigger than the error in point C (the black image). We wish to implement an algorithm to learn the nearest solution to that in point A. Fix this, the algorithm has to find at least one solution with an error smaller than the error in point C and should not reach the point C as the best solution. But the light surface is rather small and an experiment on Sobel's filter showed that its best results are above this limit. Even with a good filter, chances are nearly null to reach such a solution. Therefore, the use of this measure is inadequate in this problem.

Weighted error criterion

The parameters are the same as above. Intuitively, looking to the graphs, we can try to "lift" the point C value to be equal to the point B value to increase the light surface. We can obtain this if we weight the $n_1$parameter with an integer Rgreater than one. The relationship between the parameters becomes:

$$d_w^2 = \frac{1}{M \cdot N} \cdot (n_1/R + n_2). \tag{3.2}$$

The R parameter has a major influence on the lightly-colored surface variation, as shown in Fig. 10:

In a learning process, we can not guess the number of edge points at the beginning of the process. However, the results will depend on

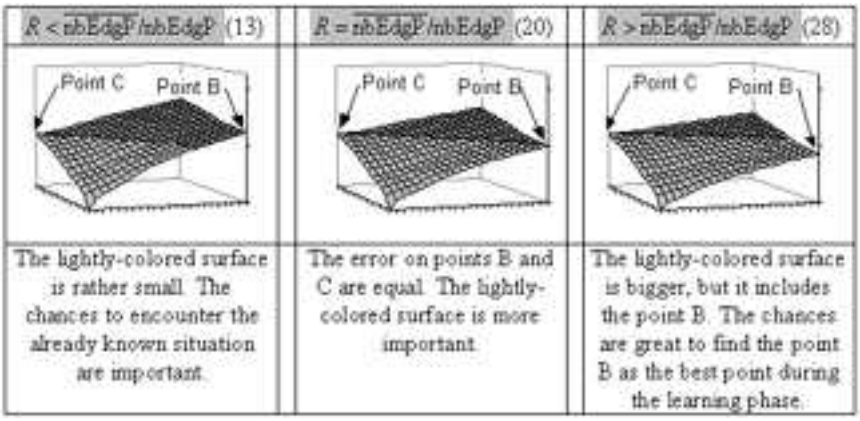


Figure 10. Weighted error graph

the intrinsic characteristics of the image. Therefore, this weighted error might not be the best answer to our problem.

Statistical error proposal

If we consider the edge detection as a classification process, then we can try to count the number or the proportion of points that are wrongly classified. We denote:

- The number of edge points of the image will be defined as $nbEdgP$.
- The number of points that are not part of the edge will be noted as $\overline{nbEdgP}$.
- If an image I has dimensions M and N, then: $\overline{nbEdgP} = M \cdot N - nbEdgP$.

In this statistical approach, the parameters n_1 and n_2 define two classes of errors and we consider the distance as a global error on the image. This error is computed as the average of the error in each class:

$$Dstat = \frac{1}{2} \cdot \left(\frac{n_1}{NbEdgP} + \frac{n_2}{NbEdgP}. \right).$$

If we consider $R = \frac{M \cdot N}{NbEdgP}$,
then

$$Dstat = \frac{1}{2} \cdot \left(\frac{n_1}{(M \cdot N - M \cdot N/R)} + \frac{n_2 \cdot R}{M \cdot N} \right) =$$
$$= \frac{1}{2} \cdot \frac{1}{M \cdot N} \cdot \left(\frac{n_1}{(1 - 1/R)} + n_2 \cdot R. \right). \quad (3.3)$$

The Fig. 11 presents the influence of the variation of the parameter R on the error computation.

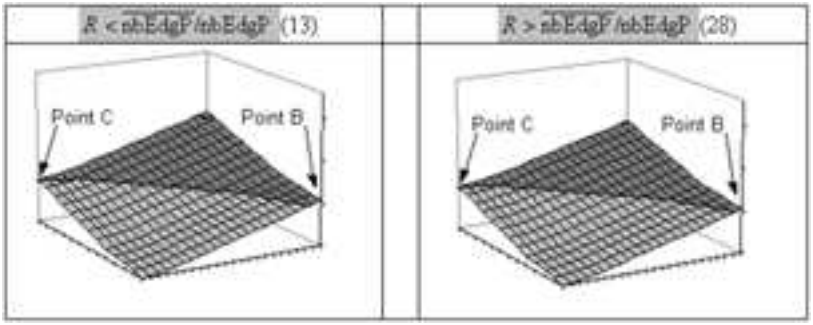


Figure 11. Statistical error graph

As the variation of R has almost no influence on the limit of the surface, this error can be considered as a good indicator for our experimentation. As long as $R > 5$, the variation of the surface can be considered as acceptable. There is no upper limit, but we remark that if R tends to a big number then the first term of the equation becomes insignificant with respect to the second term. R should be set at a higher value in case of very high number of edge points in the images.

We can also notice that this distance measure is dependent neither on the image size, nor on the edge point count number. Consequently,

we also use it to compare the quality of the edge detection between two different images.

Fuzzy distance criterion

We analyzed two distances, both deterministic, and both based on pixel comparison: the value for each pixel on the computed edge map was compared with the value for the pixel with the same coordinates on the reference map. If the detected edge doesn't match exactly the reference edge map, then the detection is considered erroneous. This does not take into account the subjectivity of an ideal edge map. As we proposed in the edge map definition section, it may be profitable to consider an area around the ideal edge where the error is less weighted than at a larger distance.

We define two areas around a pixel point:

- A nearby space, where the error is considered null.
- A small distance area, where the error is proportional to the distance to the closest edge pixel.

We can give this definition: We assume that a pixel $p_{ij} \in I$ can have any value in the range of gray levels, while a pixel $p^*_{hk} \in I^*$ can have only the values corresponding to black and white. The ideal contour image is a B/W image.

To any pixel in the obtained image we assign a pixel-level error computed as follows. First, define a weighting matrix $\mathbf{w}^{(1)} = \left[w^{(1)}_{ij}\right]_{i=1..M,j=1..N}$, named closeness matrix, with values:

$$w^{(1)}_{i,j} = \begin{cases} 0 & if\ \exists p_{h,k} \in C,\ p_{h,k} = contour\ brightness,\ and \\ & |i-h| = 0\ and\ |j-k| = 0 \\ 1/4 & if\ \exists p_{h,k} \in C,\ p_{h,k} = contour\ brightness,\ and \\ & |i-h| = 2\ or\ |j-k| = 2 \\ 1/2 & if\ \exists p_{h,k} \in C,\ p_{h,k} = contour\ brightness,\ and \\ & |i-h| = 3\ or\ |j-k| = 3 \\ 3/4 & if\ \exists p_{h,k} \in C,\ p_{h,k} = contour\ brightness,\ and \\ & |i-h| = 4\ and\ |j-k| = 4 \\ 1 & else. \end{cases}$$

The *closeness matrix* can be graphically plotted as showed in Fig. 12:

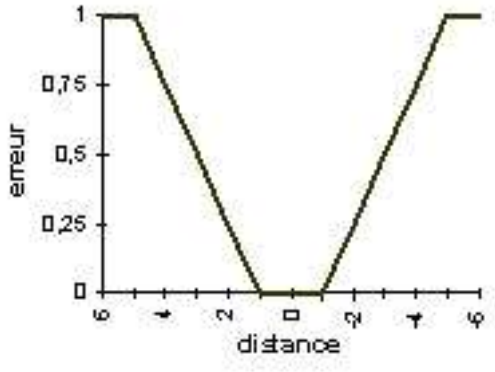


Figure 12. Distance – error function

Build the matrix "false contour":

$$\mathbf{D}_{ij}^{(1)} = \begin{cases} 1 & if\ (p_{ij}^{*} = contour\ \&c_{ij} = non - contour) \\ 0 & else. \end{cases}$$

Build the matrix "missing contour":

$$\mathbf{D}_{ij}^{(2)} = \begin{cases} 1 & if\ (p_{ij}^{*} = non - contour\ \&c_{ij} = contour) \\ 0 & else. \end{cases}$$

Compute n_1^{Fuz}, the weighted number of points that are **not** from the contour but are marked as contour points (*false contour pixels*), as

$$n_1^{Fuz} = \sum_{i=1}^{M} \sum_{j=1}^{N} d_{ij}^{(1)} \cdot w_{ij}^{(1)}.$$

Compute n_2^{Fuz}, the weighted number of points that are from the contour but are **not** marked as contour points (*missing contour pixels*),

$$n_2 = \sum_{i=1}^{M} \sum_{j=1}^{N} d_{ij}^{(2)} \cdot w_{ij}^{(2)}.$$

We have $n_1^{Fuz} \in [0, n_1[$ and $nb_2^{Fuz} \in [0, n_2[$. As n_1^{Fuz} and n_2^{Fuz} have comparable values to n_1 and n_2, we can use them to compute the statistical error with good results.

Discussion of the Hausdorff distance

Another criterion is based on the Hausdorff distance. This concept is based on the idea of measuring the similarity between two objects in an image. It exists two conditions to use it: firstly, the shapes to be detected have to be limited, i.e. it should be possible to surround them by a disc. Secondly, the shapes have to be closed: this is not the case with the images of blood vessels that are anyway open at one end at least. We conclude that we can not use the Hausdorff distance in this problem.

3.4 Discussion of the relationship between quality assessment method and training

An objective assessment of the results of the edge detection should first realize two objectives: first, to compare the results obtained by this filter to the results obtained with other filters, moreover to improve the settings of the filter. If the assessment method is not precise enough, then we will not be able to detect small differences between the images. If it takes in account an error more than another one, then the edge detection will be the best one for the computer but it will not be for the specialist that finally is the reference.

For this reason, we made a second manual feedback in our process of qualification of assessment methods: we presented the results of the best detection to the specialists, as we measured them and classified them. The pool of images was restricted and the differences between the edge maps were big enough to permit a quick classification. The

fuzzy operator gave us some edge maps with fine edges but they were also very often drawn as a double line: it was a double detection of the edge. As, according to Canny [CAN-86], this is one of drawbacks to avoid on edge detection, we decide not using this distance such it is described here.

One of the main features of this filter should be to produce an edge map without setting any parameter. One way to obtain this is to reduce the differences that exist between images through a preprocessing like normalization. This processing should also change the characteristics of the images closer to a zone of best detection of the filter. But some preliminary tests show that this does not give the expected results.

The next section presents the phase of training. It aims to adapt the parameters of the filter to a detection phase, without any manual settings as required till now by any other edge filter.

4 Training and fine-tuning of the filter

In this section, we present the issues related to the training phase of the filter. First, we estimate the space of possible solutions. As there are numerous solutions, we propose to limit the set of possible parameters and experiment with an exhaustive search on the restricted space. Repeating the learning stage on different images provides several sets of possible solutions. Analyzing the results, we can determine a correlation between some of these parameters. So, it is possible to reduce the space of searched values, fixing some extra rules between the parameters.

As a next step, we look for solutions in a non-restricted space of solutions. As this is impossible to realize with an exhaustive search, we use a genetic algorithm. The space of the possible parameters has to be considered with some new limits. The solutions we get from the training stage are quite different, so we tried to identify a correlation between the parameters to reduce the space of possible parameters.

4.1 The Filter Parameters and Constraints

4.1.1 Edge Detector Parameters

The parameters of fuzzy edge filter presented in the Section 2 are:

- Seven thresholds for the three input triangular membership functions;
- p, q and r parameters are for each rule, that is each possible combination of input function for both small and large window gradients on the defuzzification stage;
- Two thresholds to set the final pass band filter.

Usually, the values at the input of a fuzzy filter are normalized. The possible range of values is $[0, 127]$; zero if all the points have the same luminosity. 127 if the media is equal to 127 and the luminosity of the points shared into two groups with null and 255 values. The sharing of the Small Window Gradient and the Large Window Gradient over the possible range of values has been evaluated over a set of medical images. An average pattern of histogram of the gradient values is shown in Fig. 13. They have been normalized on the interval $[0, 30]$. We can presume from the histograms that pertinent values for the three input membership functions are in this interval.

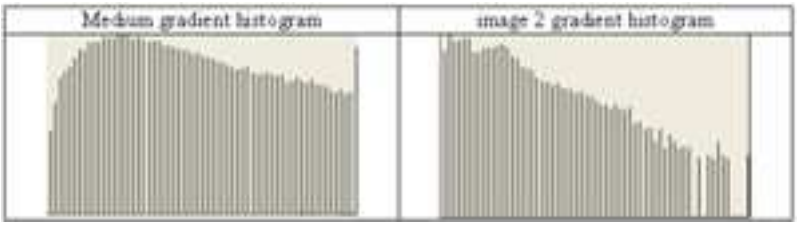


Figure 13. Gradient histogram samples

4.1.2 Edge Detector Constraints

To evaluate the complexity of this learning task, it is important to take into account all the parameters and the linked constraints. The ranges of the parameter values are:

The thresholds of the three input triangular membership functions can vary in the interval $[0, 30]$.

The defuzzification stage has p, q, r parameters for each rule, that is each possible combination of input function for both small and large window gradients. The p, q and r parameters can vary in the interval $[-256, 256]$.

The final pass band filter has two thresholds in the interval $[0, 256]$.

The constraints on the thresholds of the three input triangular membership functions:

The constraints are inter dependent:

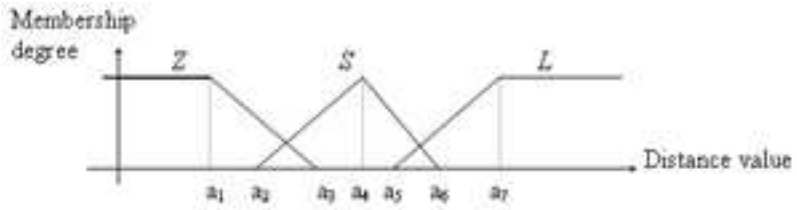


Figure 14. Fuzzy filter thresholds

The following constraints are built-in $a_1 \leq a_3$, $a_2 \leq a_4$, $a_4 \leq a_6$ and $a_5 \leq a_7$.

Moreover, we fixed these ones: $a_1 \leq a_2$, $a_2 \leq a_3$, $a_5 \leq a_6$, $a_4 \leq a_5$ and $a_3 \leq a_4$.

The constraints on the p, q and r parameters:

We choose to initially set the p and q parameters to zero. To restrict the number of solutions, we fixed three classes of r parameters: Zero, Small and Large. Every r_k is affected to one of these classes and is used to calculate the final crisp correction, according to the algorithm described in Section 2.5. The next rule defines the relationship between

the r parameters:

$$r_{Zero} < r_{Small} < r_{Large}.$$

The influence of the band pass filter settings on the global edge filter performance in the final stage:

We can determine several different cases: filtering with a low band, a band pass, or a high band filter. The filters are defined as:
low band filter:

$$edge_{i,j} = \begin{cases} 1 & if\ \Delta < LBT \\ 0 & otherwise \end{cases} \tag{4.1}$$

high band filter:

$$edge_{i,j} = \begin{cases} 1 & if\ \Delta > HBT \\ 0 & otherwise \end{cases} \tag{4.2}$$

pass band filter:

$$edge_{i,j} = \begin{cases} 1 & if \begin{pmatrix} \Delta > LBT\ and\ \Delta < HBT \\ with\ LBT < HBT \end{pmatrix} \\ 0 & otherwise \end{cases} \tag{4.3}$$

where LBT and HBT are respectively the Low and High Band Thresholds.

The low band filter puts in evidence as edges the values that are below a threshold; Conversely, a high band filter keeps the highest values. These two cases provide somewhat symmetrical results. They are very similar, so we can analyze only one of them. If we choose the high band filter case, the first stage of the edge detector provides a crisp correction that represents the edginess of the point.

If the pass band filter is set with two thresholds, then its influence on the previous stage settings is different. If the crisp values from the previous stages of the filter are centered on the average value (127), then the representative values of an edge are the smallest ones for which:

$$LBT - 127 < \Delta\ < HBT - 127.$$

In fact, it is an inverse case of the high band filter where the edginess is represented by a high Δ value. If the filter is not centered on an average value, then a range of values without any signification will represent the significant edginess. For these reasons, we use a high band filter and configure it with only one parameter, namely the high band threshold.

4.1.3 The Dimension of the Training Space

If we have wish to try all the solutions with a 0.5 gradient step for the entry filter thresholds in the interval $[0, 30]$, the number of possible solutions to calculate for the entry filter is around $43 \cdot 10^9$. The number of rules of the Sugeno defuzzifing engine combination is around 9^6 that means $5 \cdot 10^5$ and the number of possible r_k vales is $4 \cdot 10^9$. The number of possible thresholds being around 100, this finally means that we have to perform more than $800 \cdot 10^{25}$ tests, that is 2^{96} tests. These evaluations are complete edge detection over sample images and they have to be repeated for each image of the training set. This requires a high time processing. This task is impossible to solve with a personal computer in a reasonable period of time.

As a comparison, the size of the search space of possible moves in the chess problem is around 2^{400} [WIN-92]. It means that the problem of the search of the best parameters cannot be solved searching by enumeration all the search space. We will consider two different adaptation strategies: a genetic algorithm based approach and a gradient based approach using a limited number of parameters.

4.2 The gradient-based approach using a limited number of parameters

In this approach, to restrict the search space, we fix the Sugeno engine rules, the $r_k$values and the output filter thresholds, as presented in Fig. 15.

Then, we successively estimate what are the most efficient combinations of values for the entry filter thresholds respecting the conditions

Δ'		$d(S_{i,j}, \bar{S}_{i,j}^{LW})$		
		Z	S	L
$d(S_{i,j}, \bar{S}_{i,j}^{SW})$	Z	-L	+Z	-S
	S	+Z	+S	+L
	L	-S	+L	+L

r_k		
Z	S	L
25	64	140

Figure 15. Gradient-based bests settings

we have defined in Section 3.1. To reduce the dimension of the searching values space, we define the smallest interval of values giving some representative results. We first consider the possible range of gradient values and define the $[0, 30]$ interval: the first tests show that the best results are always obtained, however, with values in the $[0, 15]$ interval.

Regarding the gradient, its initial value was fixed at 0.5 and the further investigation shows that a smaller value does not increase very much the precision of the results, but it does require much more computing time.

A training phase validates the theoretical considerations about distance. For each image in a set of five, we obtained a comparison of the best distances using the weighted and the statistical distances combined with a numerical or fuzzy assessment of the points. The results are presented in Fig. 16 and 17.

In a classical learning process on a classification problem, it is usually recommended to make the learning on a subset of the sample and to measure the accuracy of the obtained results on the other subsets. In our case, we consider that each image is a subset of points that has to be classified in two categories. We can perform a learning phase on one image and then measure the correctness of the classification on the other images. In this way, we can determine which parameter

	Thresholds							distance
image	a_1	a_2	a_3	a_4	a_5	a_6	a_7	*Weighted*
1	0	3.5	4	4	3.5	8	7	0.0187
2	0	6	6.5	6.5	6	7	11	0.0108
3	0	4	4.5	4.5	4	6	6	0.0213
4	0	3.5	4	4	3.5	8	7	0.0308
5	0	5	5.5	5.5	5	6	9	0.0156

Figure 16. Best weighted distances with images

	Thresholds							distance
image	a_1	a_2	a_3	a_4	a_5	a_6	a_7	*Statistical*
1	0	4.5	5	5	4.5	9	9	0.1930
2	0	8.5	9	9	9	11	9.5	0.1131
3	0	5	5.5	5.5	6	6.5	6.5	0.2203
4	0	4	4.5	4.5	5	6	7	0.3171
5	0	7	7.5	7.5	9	9.5	9.5	0.1592

Figure 17. Best statistical distances with images

combination is the best for our problem.

4.3 The fine tuning

The results of this first phase allows us to reduce the space of the search values:

- They show some interesting similarity: the pairs of parameters a_2, a_5 and a_3, a_4 are equal.
- The parameter a_1 is always null
- The maximum value is 11.

Therefore, we are able to extend the investigations to the fuzzy measure of the distance and to test some other combination of rules.

Observing the values of the entry filter threshold and the values of the Small Window gradients sharing over the set of images, it is possible to foresee a link between their values. If the average of the Small Window gradients over an image is higher, then the thresholds of the entry filter will have larger values. We try to look for a linear relation between the media of the gradients over a very large window and the values of the thresholds. This adds two new parameters, as described in the formula 4.4.

$$a_{n_{cor}} = \alpha \cdot var_{VLW}(G_{SW}) \cdot a_n + \beta \cdot \overline{G_{SW}} \tag{4.4}$$

where α and β are the two parameters of the linear correlation.

4.4 The genetic algorithm approach

In this section, we briefly present the genetic algorithm and then we show how we use it.

4.4.1 Genetic Algorithm (GA)

GA represent a statistical approach to the resolution of a problem. They are considered as a suitable method in case of search in a huge space of solutions. The power of these algorithms has been demonstrated by solving many problems.

In a genetic approach, the space of solutions is investigated randomly. The description of each of the solution of the problem is encoded in a genotype, a chromosome-like data structure. The condition to apply such algorithms is to have a function that returns an objective assessment of each of the solutions. This *fitness* function measures the objectively assessable performance of each element of the population. It is used to rank the elements of the population and to compare them. The search provides one or several suitable results.

A basic genetic algorithm uses this scheme:

Generates a random population

Evaluates the fitness of each element of the population

Repeat until the end criteria is reached

Selection of two elements from the population according to their performance.

Recombination cross over these selected elements to form new offspring (children).

Mutation With a mutation probability mutate new offspring at each position in chromosome.

Evaluate the fitness of each new possible element of the population

Insertion of the new elements in the population

End Loop If the end condition is satisfied, **stop**, and return the best elements in current population

Different strategies can be used in the selection, recombination, mutation and insertion steps.

The selection permits to choose the parents that generate the next generation of solutions. Usually, the selection is partially based on the quality of the parents. It is assessed by comparison between the fitness and the medium fitness over all the population.

The recombination phase simulates the natural reproduction phenomena. The information from the chromosome of the two parents is combined to generate a new offspring. The phase can use crossover mechanisms in a single or multiple points. It can also be based on the real values of the variables.

The mutation phase aims to introduce small changes with low probability in the properties of the elements. The mutation rate should inverse proportional to the number of dimensions of the problem.

Many insertion strategies are available. Firstly, a decision must be done on the size of the population: should it be fix or growing. If it is fixed, the replaced elements can be some the parents randomly chosen or the weakest elements of the population. The number of created offspring can be greater than the needs and then only the best children are inserted.

This basic algorithm can be ameliorated to respond to some existing drawbacks. By instance, in case of a space of solutions that have much local minimums, it can be difficult to find the best minimum point. It is possible to find a good solution but that is somehow hiding the best one.

To avoid this, it is possible to generate several populations. In each population, the basic algorithm will apply but after the insertion of the new elements, a new phase of migration between the sub-populations will be added.

The sub-populations can be organized over a topology as a circle or a bi-dimensional map. The migration can be based on a general diffusion strategy or a limited to the neighbors. It can be systematic or randomly based. This organization generally gives better results than a single population even if it has with more elements.

4.4.2 Application to the fuzzy filter training

In the Section 4.1.3, we demonstrated that the space of solutions is smaller than 2^{96}. It means that a 96 bits chromosome can contain a representation of all the solutions of this problem. In the Section 4.1.3, we made this evaluation taking into consideration a restricted space of values for the a_n parameters. Using genetic algorithms, it can be possible to extend the precision of the evaluation of the values without much increasing the computing time.

Some decisions about the selection, recombination, mutation and insertion phases are dependent on the implementation we use. Some of the implementations have limited possibilities. Nevertheless, we require using some sub-populations.

4.5 Discussion of training results

Genetic algorithm training does find better results than a gradient-based exploration of the space of solutions with fixed parameters. In the gradient-based approach, the assumptions we made over the rules of the defuzzifying engine or the output filter were not the best existing ones.

If the genetic algorithm approach gave us some good results, we can not guaranty that we found the best combination applied to our problem. During this process, we get about fifty possible good solutions. But they are very different each others. A fine analysis of the results has been done to find some correlation between the parameters.

The Genetic Algorithm approach puts forward that all the parameters don't have the same influence during the learning over the distance measure evolution:

During a training phase, i.e. learning over a population, the output filter value does not vary much. The values chosen at the beginning of this phase for the classes Zero and Small of defuzzying filter are also very stable. But this does not mean the values over all the training phases are coherent.

The α and β parameter settings are volatile. If a change occurs on their values, then they are also big changes on the entry filter values and sometimes on the rule engine.

The rule learning is dependent on the settings of the output filter. Some small changes can happen till the end of the learning phase. One rule remains very stable: if the gradient is owned by the Zero Class on the Large and on the Small windows, then the deffuzying is largely negative (-Large). For the other rules, it is difficult to establish a clear table of decision.

Nevertheless, if some statistics are computed from the best results, we can establish the results in Fig. 18:

	Thresholds						
	a_1	a_2	a_3	a_4	a_5	a_6	a_7
value	3.483	7.142	5.585	9.323	15.07	11.47	15.43
standard deviation	0.132	0.161	0.206	0.431	0.26	0.303	0.344

5 A synthesis of the comparative analysis of the results and applications in medicine

5.1 Methodology

To compare the results obtained with the pseudo statistical filter and those obtained with the classical filters, we use the same methodology that we used to find the best settings of the pseudo statistical filter.

	r_k					
	Z	S	L	α	β	output filter low threshold
value	3.93	21.9	86.3	6.333	0.827	62.02
standard deviation	0.113	0.818	2.36	0.051	0.14	1.694

Figure 18. Statistical GA results

If the edge detection is considered as a classification problem [KON-03], then the Receiver Operating Characteristics (ROC) graphs is a suitable method to classify the results. We denote:

	True class	
	P	**n**
Hypothesized Class	True Positives (TP)	False Positives (FP)
	False Negatives (FN)	True Negatives (TN)
ColumnTotal	Po	Ne

Figure 19. Two classes classification basic notations

The false positive and true positive rates are defined as:

$$fp = \frac{FP}{Ne}$$

and

$$tp = \frac{TP}{Po}.$$

The previously defined $n1$ and $n2$ are respectively corresponding to FP and FN.

The graphs show a graphical representation of the results of edge detection over a set images and allow classification. Each plot is representative for the quality of the result of an edge map, having as coordinates the proportion of points not detected as edges (fp rate) and the proportion of points truly detected (tp rate).

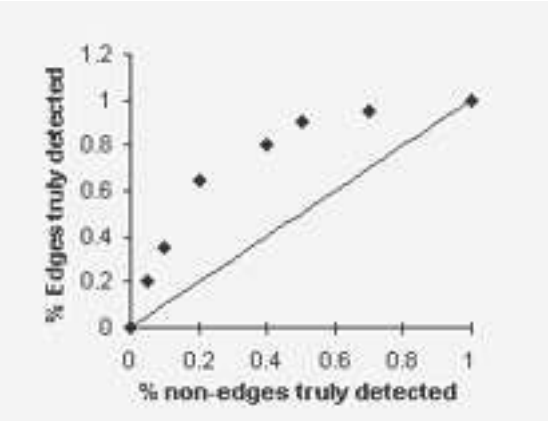


Figure 20. ROC curves evaluation

Any point over the line $x = y$ is considered as better than the result of a hazardous detection. Then, the assessment of the quality of detection of an algorithm is not evaluated by the addition or the media of the distances of each edge detection for all the images of a set, but by the size of surface under the curve representing the assessment. Ideally, using the ROC graphs, the ideal edge maps are known. Therefore, the proportion of points can be exactly evaluated. In our case, we decided to approximate the number of edge points as a ratio of the number of points of the image. What are the consequences of an error on this approximation? Using the same value of ratio for all the images, the deformation of the computation of the assessment is the same for all the algorithms computing one image.

If we can consider that all the points situated on a parallel line to the line $x = y$ represent a same level of error, then the distance can be evaluated as the distance from point (0,1) to the projection of the point on the axis $x = 0$. The value of this distance is:

$$D = 1 - \left(\frac{TP}{Po} - \frac{FP}{Ne}\right) = 1 - \frac{n_2}{Po} - \frac{n_1}{Ne}.$$

Let $Ne = M \cdot N \cdot \left(1 - {}^{1}/_{R}\right)$
and $Po = {}^{M \cdot N}/_{R}$,
then

$$D = \frac{(n_2 \cdot (R-1) + n_1)}{M \cdot N \cdot (1 - 1/R)}. \tag{5.1}$$

that is the same that formula 3.3.

What is the influence of the error on the arbitrary choice on R value? In the set of images we studied, R value was varying from 13 to 25. Choosing an arbitrary value of 19, which kind of error will it generate to the ROC graph?

$$fp = \frac{n_1}{M \cdot N \cdot (1 - 1/R)} \quad \text{and} \quad tp = 1 - \frac{R \cdot n_2}{M \cdot N}. \tag{5.2}$$

The choice of this value lead to an error of plus or minus 2 percent on *fp rate* and plus or minus 30 percent on *tp rate*. This variation on *tp rate* is homogeneous for all the tests made over an image. Therefore, we can use this evaluation to compare the results between the algorithms. We can note that error can lead to get a *tp rate* smaller than 0.

The ROC graphs also show why the Euclidean distance is not representative of this distance: The points that are at the same Euclidean distance from the ideal result are situated on a line. All the points on such a line definitively do not have the same error level. $d^2_{Euc} = \frac{1}{M \cdot N} \cdot (n_1 + n_2)$.

The evaluation of the quality of detection of algorithms using ROC graphs is made measuring the size of the surface under the curve [FAW-03].

When filters as Sobel or Laplace build edge maps, they produce an evaluation of the edginess of each point of the image. To compare these edginess values to the ideal map, this first evaluation must be filtered. The values of the threshold of the filter for a good detection are different from an image to another one. Using ROC graphs [FAW-03], it is possible to plot the quality of detection for each image depending on the threshold variation. In the Fig. 20, we can compare the results of the Sobel filter on a set of five images. As advised by Holte in [HOL-02]. In this case it is necessary to determine a threshold averaged ROC

graph. Then, the *ft* and *tp* values are calculated by Formulas 5.3 and 5.4. We denote *Ne* as the number of elements of the tested set and *i* as index of all these elements.

$$\mathrm{tp} = \frac{\sum\limits_{i} tp_i}{Ne}, \tag{5.3}$$

$$\mathrm{fp} = \frac{\sum\limits_{i} fp_i}{Ne}. \tag{5.4}$$

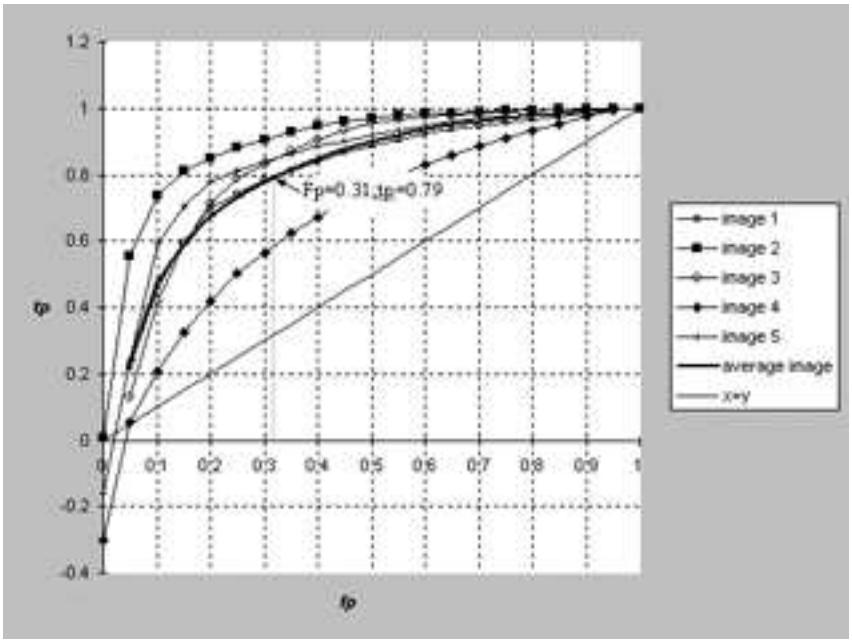


Figure 21. ROC graph of the Sobel edge detection on the set of 5 medical images

In Fig. 21, the emphasized point of coordinates (0.31, 0.79) has the best interpretation of the edge map for all the images. It is the point

of the curve with the smallest error and the tangent of the curve in this point is the closest to the value 1 (parallel to the line $x = y$). The error in this point is equal to the media of the pseudo statistical errors as proposed in formula 3.3.

If these graphs allow us an appreciation of the quality of the classification, we can use them to compare algorithms. But, in a recent paper [HOL-04], [HOL-02], Holte emphasized that the ROC graph misses some important data: It is not possible to evaluate the confidence on this graph. The use of the Cost Curves is proposed to solve this issue.

5.2 Comparative analysis with differential filters

The gradient results of the Sobel filter from the Fig. 21 or 22 are to be compared with the results from the Fig. 16, remembered on the second line and the results from genetic algorithm on third line:

Image	1	2	3	4	5
Distance with Sobel	0.249	0.169	0.231	0.362	0.209
distance with Pseudo statistical tuned on a gradient based process	0.1930	0.1131	0.2203	0.3171	0.1592
distance with Pseudo statistical tuned on a genetic algorithm based process	0.1853	0.1075	0.186	0.2778	0.1456

Figure 22. Best edge detection

The measured difference is significant between the best Sobel detection and the pseudo statistical filter. Nevertheless, we cannot say that we found the best result with the genetic search algorithm. The search produced many different answers that are not always coherent. But these results are very encouraging on the possible performance of this fuzzy edge detector.

Some other research results on a comparable detector named Competitive Fuzzy Edge Detector (CFED) were published in [LIA-03]. The general structure of the edge filter is similar that of our filter. The main advantage of CFED is that it produces edges, which are very fine and precise, while we demonstrated that this cannot be the case with our filter. Another major difference is the statistics at the entry of the filter are not computed on a window size base, but on the gradient variation along the four different axis crossing in a point. If we compare the resulted edge maps, we obtain the Fig. 23. Yet, we can observe that some details are not present on the CFED edge map

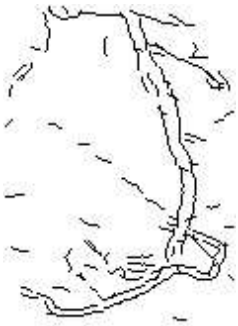
Pseudo statistical edge detection	CFED
	

Figure 23. Fuzzy edge detector comparison

We got another surprise when measuring the distances between the edge map and the ground truth: with this edge detector producing thinner edges, the measured error is much larger. This is mainly due to the fact that the edge being so fine does not exactly correspond to the ground truth and is evaluated as erroneous. The counterpart that does not solve this filter is the presence of two parameters that have to be set to generate the edge map.

5.3 Immunity to Gaussian noise

The measured distances with this new filter are twelve percent smaller than with the Sobel's one. The Fig. 24 presents the immunity to Gaussian noise.

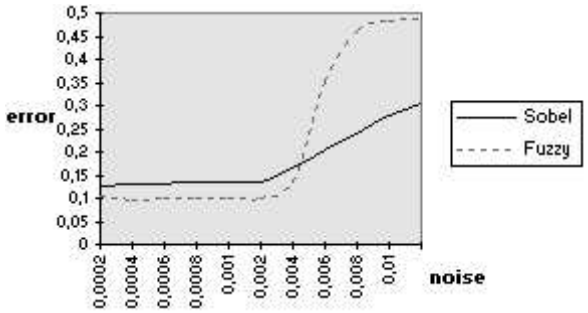


Figure 24. Noise immunity

For small Gaussian noise, our approach assesses better. The quality degrades for higher values of noise, with the Sobel filter.

5.4 Envisaged applications

The fields of application of an efficient edge filter are numerous. In the medical field, the part of automatic imagery is increasing very fast. The aim is not to replace the specialist but to provide him some powerful tools to assist him in the diagnostic.

We applied this edge detector on some angiographical images. Some next studies could be realized to study the efficiency of this detector on other classes of medical images as X-ray or echographies.

If it is checked that this filter is efficient on other medical image classes, some other kind of images should be studied. By instance, the SIDBA or Sowerby sets are usually used to evaluate the performance of edge detectors. The preliminary but informal tests were made and demonstrated some good results.

6 Conclusion and future research

The reported edge detection uses only the first order (average) statistical information, because only the averages are considered in the distances expressions. Further works needs to be done to include the differences between the spreading in the two windows. Indeed, one may expect the spreading τ due to an edge in a small window τ_{SW} is larger than the spreading τ_{LW} an edge produces on a large window of relatively homogeneous background. However, including $d(\tau_{SW}, \tau_{LW})$ as an input to the fuzzy system increases the dimensionality of the search space and makes more difficult the training of the system.

We presented a new edge detector algorithm based on fuzzy logic. We also presented an original method to measure a distance between the ground truth and the detected edge map. This permits us to objectively assess the results of the filter and to calibrate its settings to optimize the results.

The edges as drawn on a resulting edge map are good: they do not miss much edge points and when they do so, it is often due to a problem in the image. In this case, only the expertise of a specialist, or a higher-level process can detect the correction to the map. As the drawn edges are not fine, a last stage should be added to refine the results. This operation have to be realized with much care to situate the edge as near of its real position as possible.

The problem of edge detection is strongly linked to the segmentation problem. An improved algorithm in one of the two fields can provide new perspectives.

References

[1] [BAU-02] M. Baudry, N. Vincent, Multicriteria decision making RFAI team publication, First annual meeting on health science and technology. Tours (France), May 2002, http://www.rfai.li.univ-tours.fr/RFAI/rapports/bau02a.pdf

[2] [BEL-98] F. Bellet, Une approche incrémentale a base de processus coopératifs et adaptatifs pour la segmentation des images en niveaux de gris, rapport de thèse / Juin 1998, INPG Grenoble

[3] [BEZ-94] J.C. Bezdek, D. Kerr, Training Edge Detecting Neural networks with Model-Based Examples, Proceedings 3rd International Conference on Fuzzy Systems, FUZZ-IEEE'94, Orlando, Florida, USA., June 26 – 29, 1994, pp 894-901

[4] [BOW-02] S. T. Bow, Pattern Recognition and Image Preprocessing, second Edition, 698 pp., ISBN 0-8247-0659-5, Marcel Decker Inc., New-York – Basel, 2002

[5] [CAN-86] J. F. Canny.: A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, pp 679-698

[6] [COH-97] A. H. Cohen, C. McKinnon, J. You, Neural-Fuzzy Feature Detectors, DICTA-97, Auckland, N.Z., Dec 10-12 1997, pp 479-484.

[7] [EFF-00] N. Efford, Digital Image Processing, Addison-Wesley, Reading, MA, 2000, pp.164–173.

[8] [FAW-03] T. Fawcet, ROC Graphs: Notes and Practical Considerations for Researchers, document acceded on the web on Sept. 2004 at http://www.hpl.hp.com/personal/Tom_Fawcett/papers

[9] [GAL-00] G. Gallo, S. Spinello, Thresholding and Fast Iso-contour extraction with Fuzzy Arithmetic, Pattern Recognition Letters, Volume 21, 2000, pp 31-44

[10] [HOL-02] C. Drummond, R. C. Holte, 'Explicitly representing expected cost: An alternative to ROC representation', in Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, pp. 198–207.

[11] [HOL-04] C. Drummond, R. C. Holte, What ROC Curves Can't Do (and Cost Curves Can), ECAI 2004, First Workshop on ROC Analysis in AI, ECAI, 2004

[12] [KIM-04] D.-S. Kim, W.-H. Lee, I.-S. Kweon, Automatic edge detection using $3 \cdot 3$ ideal binary pixel patterns and fuzzy-based edge thresholding, Pattern Recognition Letters 25 (2004), pp. 101–106

[13] [KON-03] S. Konishi, A. L. Yuille, J. M. Coughlan, S. C. Zhu, Statistical Edge Detection: Learning and Evaluating Edge Cues, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 25, no 1, Jan. 2003, pp. 57-74

[14] [KOP-96] H. Kopp-Borotschnig, A. Pinz, A New Concept for Active Fusion in Image Understanding applying Fuzzy Set Theory, proceedings of FUZZ-IEEE '96, New Orleans.

[15] [LIA-03] L.R. Liang, Carl G. Looney, Competitive Fuzzy Edge Detector, International Journal of Applied Soft Computing. 3(2): 2003, pp. 123-137

[16] [MAR-00] A. Martin and S. Tosunoglu, Image Processing Techniques for Machine Vision, FIU, Florida Conference on the Recent Advances in Robotics - 2000

[17] [SAL-96] M. Salotti, F. Bellet, C. Garbay, Evaluation of Edge Detectors: Critics and Proposal. Workshop on Performance Characteristics of Vision Algorithms, Cambridge – 1996

[18] [TEO-98] H.N. Teodorescu, A. Kandel, and L.C. Jain (Eds.): Fuzzy and Neuro-fuzzy Systems in Medicine. CRC Press, Florida, USA, 394 pp.+ xxviii, (ISBN0-8493-9806-1), 1998

[19] [TEO-00] H.N. Teodorescu, D. Mlynek, A. Kandel, H.J. Zimmermann (Eds.): Intelligent Systems and Interfaces. 480pp., ISBN: 079237763X, Kluwer Academic Press, Boston. 2000

[20] [TEO-04] H.N. Teodorescu , unpublished paper, Internal Research Report, Techniques & Technologies Ltd., Iasi 2004.

[21] [TIZ-97] H. R. Tizhoosh, Fuzzy Image Processing. Subtitle: Introduction in Theory and Practice. Publisher: Springer-Verlag. October 1997 ISBN: 3-540-63137-2 Language: German (The English translation will be published within next months)

[22] [TIZ-98] H. R. Tizhoosh, Fuzzy Image Processing: Potentials and State of the Art, 5th Int. Conference on Soft Computing, Iizuka, Japan, October 16-20, 1998, vol. 1, pp. 321-324

[23] [VIL-03] D. Van De Ville, M. Nachtegael, D. Van der Weken, E. E. Kerre,W. Philips, I. Lemahieu, Noise Reduction by Fuzzy Image

Filtering,. IEEE Transactions on Fuzzy Systems, Vol. 11, No. 4, August 2003, pp429-436.

[24] [WIN-92] P. H. Winston, Artificial Intelligence, Third Edition, Addison-Weysley, London, ISBN 02015337774, 1992, pp 102

Horia-Nicolai Teodorescu, Frédéric Mayer Received October 27, 2004

Horia-Nicolai Teodorescu,
Technical University of Iaşi and
Institute for Theoretical Informatics
of the Romanian Academy
E–mail: *hteodor@etc.tuiasi.ro*1−

Frédéric Mayer,
Conservatoire National des arts et métiers, Grenoble and
TIMC - SIC team - Grenoble and
Institute for Theoretical Informatics
of the Romanian Academy
E–mail: *fmayer@etc.tuiasi.ro*