Advances in Intelligent Systems and Technologies
Proceedings ECIT2004 - Third European Conference on Intelligent Systems and Technologies
Iasi, Romania, July 21-23, 2004

Pseudo Statistic Edge Detector using Neuro Fuzzy Computation: Error Distance Evaluation

Frédéric Mayer[1], Horia-Nicolai Teodorescu[2]

1-Conservatoire National des arts et métiers - Paris and TIMC - SIC team - Grenoble France and Institute for Theoretical Informatics of the Romanian Academy - fmayer@etc.tuiasi.ro
2-Technical University of Iasi and Institute for Theoretical Informatics of the Romanian Academy - hteodor@etc.tuiasi.ro

Abstract: During our investigations to improve a new type of edge detector based on fuzzy logic, we had to confront with several questions: how to objectively estimate the correctness of edge detection? Which distance should be used to measure it? Does it exist a true reference? Several algorithms are investigated here to measure the distance between images, starting from the simplest one - the Euclidean distance. We finally defined a statistical measure used in combination with a fuzzy approximation of the distance between an image and a model.

Keywords: edge detection, low level processing, image processing, fuzzy logic

1. Introduction

The image pre-processing is a wide area of research with multiple applications. Edge detection is a step of the process that permits a computer to understand the image content. Moreover, edge detection evidences elements that help the humans to make decisions. In computer vision applications, the image pre-processing stage aims to emphasize elements in the picture that are essential in specific fields of applications. The stages of this information process are numerous and one of them is classification [1]. The latest trend to solve this task combines two technical areas: edge detection and segmentation [2]. The edge detection treatment is usually shared in two phases: a low level and a high level. In a classical approach, with gray level images, the low level edge detection is based on a statistical computation of the luminosity, extracting gradients and leveling the higher variations. The well-known Laplace and Sobel methods are both based on this idea [3]. Another known method, the Canny edge detector tries to counter a major weaknesses of Sobel and Laplace detector, namely their noise sensibility [4]. The Canny detector is more sophisticated and acts in several steps: it filters noise before processing, then post-processes the result to thin the drawing of the edge map. The fuzzy pseudo statistical edge detector is also based on such a scheme and its principle will be presented in section 2.

In this paper, we present the methodology we followed to evaluate this new detector versus the other ones and, also, as this detector has many possible parameters conjunctions, we had to learn the best ones, comparing the outputs to an ideal result. In the third section, we present how we determined what is the best edge map for a given image. In the fourth one, we justify why the Euclidean distance measure cannot be used in this case. In the next sections, several comparison tools are evaluated and we study the conditions in which they can be applied to our problem. We will present some results based on an angiographical image.

2. System Description

This system is described in more details in [5]. This edge detector to provide some statistics properties computes every point of the image. These ones are evaluated by a fuzzy system: based on a set of three rules and a fuzzy inference engine type SUGENO, the fuzzy system provides a correction on the original pixel value. This value is then normalized and a filter is applied to give a final binary information about the edge property value of every pixel. This system scheme is sketched in figure 1:

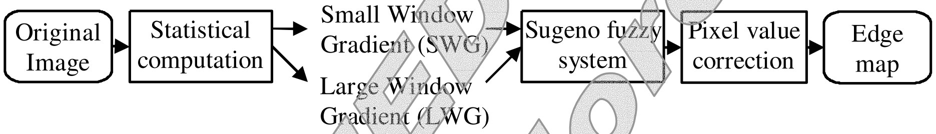
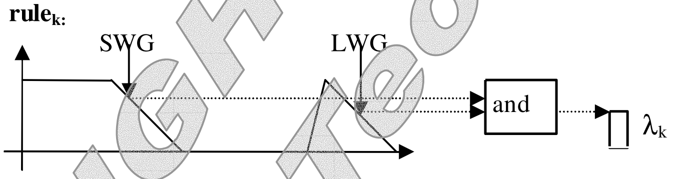


Fig. 1. Fuzzy Pseudo Statistical Edge detector scheme

The SUGENO fuzzy system is based on k inference rules. For each rule, we have:



and $\Delta=\frac{\sum_{i=1}^{k} x_k \cdot \lambda_k}{\sum_{i=1}^{k} \lambda_k}$

Fig. 2. Fuzzy Computation scheme

where $x_k=F(SWG,LWG)=p_k \cdot SWG+q_k \cdot LWG+r_k$

The edge detector depends on many settings: input membership functions, *p, q, r* parameters for each rule, and the thresholds of the output filter. To simplify the computation, we arbitrary set each p_k and q_k with the null value.

To find the best settings, a system has been implemented that computes the different possibilities and compares them. In a first step, the possibilities are calculated in two different ways: the system tries successively some sets of arbitrary chosen parameter conjunctions, or it looks for the best choice using a genetic algorithm. On the second step, the system compares the results confronting an edge map to a reference one and makes an evaluation of the differences using a comparison tool. To do so, we need the reference edge map and the evaluation tool that are presented in the next sections.

3. Reference Edge Map

We worked with several medical images extracted from a angiographical examination. We asked some bio-medical engineers to draw reference edge maps. They did it manually on a layer over the original image (see figure 3). Even if they are not experts, we have considered they have enough knowledge to make a good interpretation from a radiographic image.

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

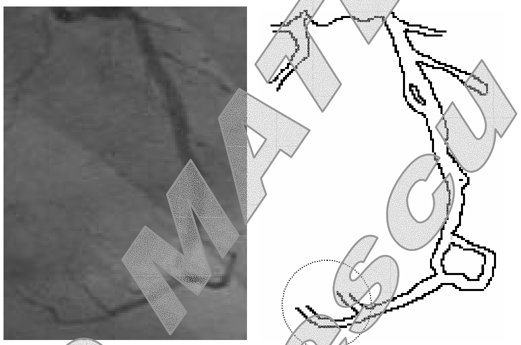


Fig. 3. Edge map example

For instance, if we focus on the circled zone, we get different versions, as in figure 4:

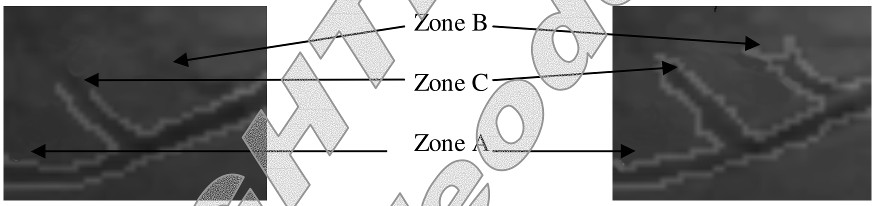


Fig. 4. Edge map details

- If the edge seems to be the same, at the pixel scale, there is a important difference (zone A).
- Some details are clearly understood in different ways (zones B and C).

Concluding, after evaluation of these different results, it seems that it doesn't exist an ideal reference edge map: 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 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 pseudo probability for a pixel to be on an edge. Then, this probability can be used to compute an error.

4. The comparison tool

The comparison tool will be used to evaluate the best settings of the edge detector parameters. We can see this process from different perspectives: Firstly, as a simple measure of the distance between two images with the Euclidean distance. It is frequently used to measure the noise between an image and a reference. Secondly, if the edge detection is seen as a classification process then we can try to count the number or the proportion of points that are wrongly classified. We will subsequently determine, which is the best choice for our application and the rational for optimization.

Notations

- 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 $\overline{nbEdgP}$.
- We consider that an image *I* has dimensions *M* and *N*.
- Then: $\overline{nbEdgP}=M\cdot N-nbEdgP$.

Euclidean distance definition

If *I* and *I^r* are two gray level images and $p_{i,j}$ is the gray level at the point (*i,j*), then:

$$d^2(I,I^r)=\sum_{i=1}^{M}\sum_{j=1}^{N}\left(p_{i,j}-p^r_{i,j}\right)^2$$

Then, the average distance between the points of the two images is defined as:

$$d^2(I,I_r)=\frac{1}{N\cdot M}\cdot\sum_{i=1}^{M}\sum_{j=1}^{N}\left(p_{i,j}-p^r_{i,j}\right)^2$$

If the image is coded on P gray levels, then we define the averaged Euclidean distance:

$$d^2_P(I,I_r)=\frac{1}{N\cdot M\cdot P}\cdot\sum_{i=1}^{M}\sum_{j=1}^{N}\left(p_{i,j}-p^r_{i,j}\right)^2$$

In our case, the blood vessel edges are expressed with a binary information:

arbitrarily, we chose that the white points show the edges. So, we have *P=1* and then:

$$d_p^2(I,I^r)=d^2(I,I^r)$$

Comparing the two images, we determine:

- the points wrongly marked as edge: *nb1*;
- the points marked as background when they are edge: *nb2*.

Then, we obtain the relation:

$$d^2=\frac{1}{M\cdot N}\cdot(nb1+nb2)$$

If we compare the reference edge image with a black one, then $nb1$ is null because the black image does not contain any point detected as an edge. Then, the computed distance is directly linked to the number of edge points in the edge reference map:

$$d^2=\frac{1}{M\cdot N}\cdot nb2$$

The comparison tool should is used to learn the best parameters of the edge detector. We investigated theoretically on the possible values to check if this distance would be able to provide improved results. We have:

$$nb1\in[0,\mathrm{NbEdgP}] \text{ and } nb2\in[0,\overline{\mathrm{NbEdgP}}]$$
$$\text{so } nb1+nb2\in[0,M\cdot N] \text{ and } d^2\in[0,1]$$

The figure 5 represents the plot of the error values depending on the $nb1$ and $nb2$ parameters:

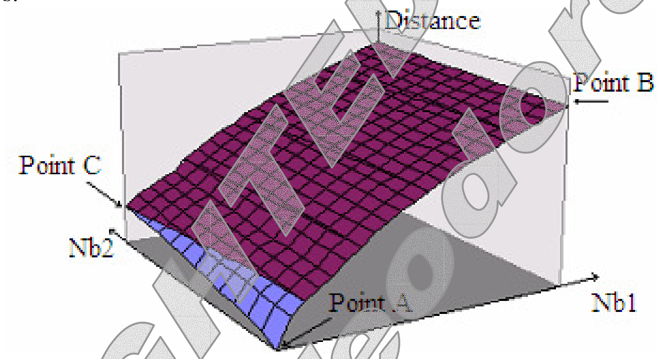


Fig. 5. Error function graph

There are three outstanding points in this graph:

- The point A shows the error when $nb1$ and $nb2$ are null; this is the perfect edge detection.

- The point B shows the error when $nb1$ is null; this is a white image where all points are detected as edges.
- The point C shows the error when $nb2$ is null; this is a black image where no edge points are detected.

The darkest surface represents all 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 want 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 not to reach the point C as 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 proposal

The parameters are the same as above. Intuitively, looking to the graphics, 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 $nb1$ parameter with an integer R greater than one. Then relationship between the parameters becomes:

$$d_w^2 = \frac{1}{M \cdot N} \cdot (nb1/R + nb2)$$

The R parameter has an major influence one the lightly-colored surface variation, as showed in figure 6:

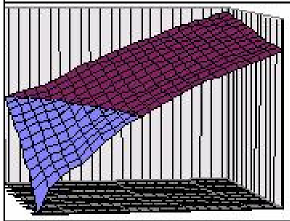
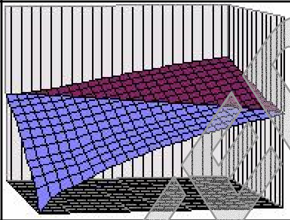
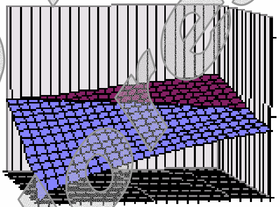
R < $\overline{\text{nbEdgP}}$/nbEdgP (13)	R = $\overline{\text{nbEdgP}}$/nbEdgP (20)	R > $\overline{\text{nbEdgP}}$/nbEdgP (28)
		
The lightly-colored surface is rather small. The chances to encounter the already known situation are important.	The error on points B and C are equal. The lightly-colored surface is important.	The lightly-colored surface is bigger but it includes the point B. The chances are great to find the point B as the best point during the learning.

Fig. 6. Weighted error graph

In a learning process, we can not guess the number of edge points at the beginning of the process. We can only make estimations. But the results will depends on this intrinsic characteristics of the image. Therefore, this weighted error might not be the best answer to our problem.

Statistical error proposal

The same parameters can be used in a more statistical view. The parameters $nb1$ and $nb2$ define two classes of errors. We can consider the distance as a global error for the image: this error is computed as the average of the error in each class:

$$Dstat = \frac{1}{2} \cdot \left(\frac{nb1}{\overline{NbEdgP}} + \frac{nb2}{NbEdgP} \right)$$

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

then

$$Dstat = \frac{1}{2} \cdot \left(\frac{nb_1}{(M \cdot N - \frac{M \cdot N}{R})} + \frac{nb_2 \cdot R}{M \cdot N} \right) = \frac{1}{2} \cdot \frac{1}{M \cdot N} \cdot \left(\frac{nb_1}{(1 - \frac{1}{R})} + nb_2 \cdot R \right)$$

The figure 7 presents the influence of the variation of the parameter *R* on the error computation.

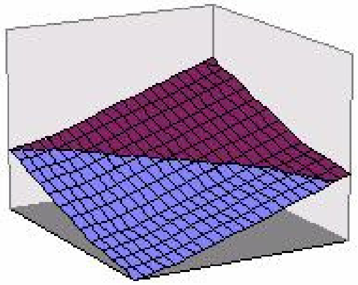
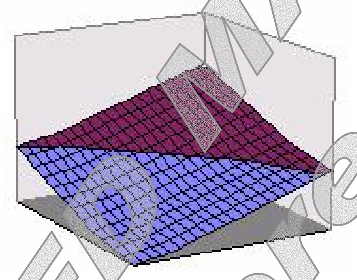
R < $\overline{\text{nbEdgP}}$/nbEdgP (13)		R > $\overline{\text{nbEdgP}}$/nbEdgP (28)
		

Fig. 7. Statistical error graph

As the variation of *R* has nearly 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 in front of the second term. Perhaps, R should be set at a higher value in the case of very high number of edge points in the images.

We can also note that this distance measure is dependent neither on the image size, nor on the edge point count number. Therefore, we also use it to compare the quality of the edge detection between two different images.-

Fuzzy distance proposal

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 co-ordinates on the reference map. If the detected edge doesn't match exactly the reference edge map then the detection is considered as erroneous. As we proposed in the edge map definition section, it can 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 as null.
- A small distance area where the error is proportional to the distance to the closest edge pixel.

The distance function graph is showed in figure 8:

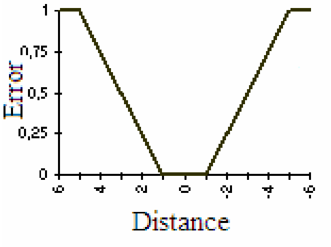


Fig. 8. Distance – error function

This calculation is to be used for the edge points but also for the non-edges points. Then, we can obtain nb_1^{Fuz} and nb_2^{Fuz} where $nb_1^{Fuz} \in [0, nb1[$ and $nb_2^{Fuz} \in [0, nb2[$

As nb_1^{Fuz} and nb_2^{Fuz} have some comparable values to nb_1 and nb_2, we can use them to compute the statistical error with comparable results.

5. Applications

The distance evaluation is to be used in:

- the learning process of the edge filter parameters as a feedback function;
- the comparison between different algorithms results.

The Figure 9 presents edge maps images obtained with the filter from the image in figure 1. The lower row contains the distance evaluation with the comparison tool.

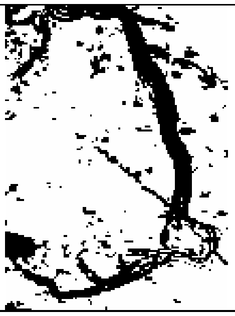
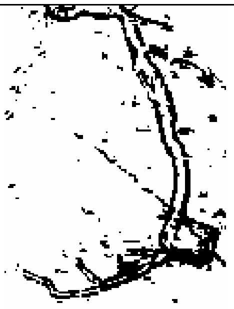
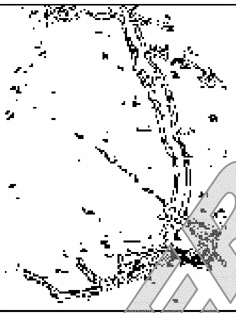
Edge maps			
Distance evaluation	0.077654	0.028419	0.02113

Fig. 9. Distance evaluation examples

The statistical error is able to provide a quite reliable estimation. However, we notice that the edge map measured as the best one is faulty: some edges appear as a double line. If we consider our detector as the low level of a more complex architecture, then this error should be managed at a higher level.

6. Perspectives

The corrected distance takes in account the lack of precision of the estimation of the reference edge map. But this lack of precision is equally evaluated on all the surface of the image. It could be possible to use the specialists' edge drawings to establish a likely edge map. The probability information contained in this map could be used in place of the described fuzzy distance function. We would then consider the different points of view of the specialists and the lack of imprecision of the eye perception about on edge at one point.

7. Conclusions

By this paper we proposed and analyzed a tool to evaluate the distance between two edge maps. We demonstrated why the Euclidean error is not appropriate in this problem. The statistical error associated to the fuzzy distance provides a realistic measure to compare two computed edges and sort the best one. This statistical error has the advantage to be independent of the intrinsic characteristics of the image and the edge map, at least if the proportion of edge points compared to non-edge points is greater than five.

With this distance measure, we were able to learn the best parameters of our fuzzy edge detector and secondly to compare the results obtained with other edge detection algorithms.

References

[1]. Sing-Tze Bow, Pattern Recognition and Image Pre-processing, second Edition, 698pp., ISBN 0-8247-0659-5, Marcel Decker Inc., New York, 2002

[2]. Fabrice Belet, Une approche incrémentale à base de processus coopératifs et adaptatifs pour la segmentation des images en niveaux de gris, rapport de thèse / Juin 1998, INPG Grenoble

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

[4]. Alberto Martin and Sabri Tosunoglu, Image Processing Techniques for Machine Vision, Proc. of Florida Conference on the Recent Advances in Robotics, from http://www.eng.fau.edu/conf/fcrar2000/ , FIU 2000

[5]. Horia N. Teodorescu, Frédéric Mayer, On a Pseudo Statistic Edge Detector using Neuro Fuzzy Computation. Proc. EEIA symp. - Pitesti (2004)