ICSV13 - Vienna
The Thirteenth International Congress on Sound and Vibration
Vienna, Austria, July 2-6, 2006

FUZZY BASED IMAGE PROCESSING FOR ROBOTIC VISION

Dorian Cojocaru[*1], Răzvan Tudor Tănasie[2]

Faculty of Automation, Computers and Electronics, University of Craiova
Bdul. Decebal 200440, Craiova, Romania
[1]Department of Mechatronics
[2]Department of Software Engineering
cojocaru@robotics.ucv.ro

Abstract

Robotics applications concerning computer vision may encounter problems that are vague or ambiguous or need simultaneous approaches in solving. This paper presents some fuzzy algorithms as an alternative solution to this kind of problems. It focuses on the general presentation of the designed fuzzy algorithms for grayscale images. Image enhancement, edge detection, edge following, thresholding and image segmentation are the main image operations envisaged. The application is implemented as a software module developed in C++. As it is structured on "specialized" functions (small parts that compute specific tasks), the implementation of needed algorithms makes the adding of further functions (algorithms) easier. The new functions wouldn't need to be fully implemented because they make use of some parts of those algorithms that were already implemented, only the specific part of the new tool having to be created at that time.

INTRODUCTION

Very often one of the first procedures applied to an image in a computer vision task is image enhancement, so this algorithm was chosen be described [5]. According to Gonzalez and Woods (1992), the principal objective of enhancement techniques is to process a given image so that the result is more suitable than the original image for a specific application. Typically we want the enhancement process to be capable of removing noise, smoothing regions where gray levels do not change significantly, and sharpening abrupt gray-level changes. It is, however, hard to incorporate all these requirements into one framework since smoothing a region might destroy a line or an edge, a sharpening might lead to unnecessary noise. A good enhancement process is, therefore, required to be adaptive so that it can process each region differently based on the region properties [13]. By using crisp logic only one of the three filters can be applied at a time. This is of course a drawback. Since fuzzy logic can easily incorporate heuristic knowledge concerning a specific application in the form of rules, it is ideally suited for building an image enhancement system.

Edge detection is a critical part of many computer vision systems. Ideally, edges correspond to object boundaries, and therefore edge detection provides a means of segmenting the image into meaningful regions [9]. Since the definition of what constitutes an edge is rather vague, heuristic, and even subjective, a fuzzy logic approach can be quite efficient.

As a general concept, the job of the cluster analyze is the data partition into a number of groups, or clusters. Applying this partitioning operation on images, the image segmentation - a very important task for image processing - is obtained.

FUZZY IMAGE ENHANCEMENT ALGORITHM

This algorithm is based on Choi's and Krishnapuram's algorithm (1995, 1997) [3] who view image enhancement as replacing the gray level value of every pixel in the image with a new value depending upon the local information [10]. If the local value region is relatively smooth, then the new value of the pixel may be a type of average of the local values. On the other hand, if the local region contains an edge or noise points, a different type of filtering should be used. So we need to create a bank of filters from which to choose the needed one. The choosing, unlike in crisp algorithms, need not be of only one of the filters but according to fuzzy logic each filter is selected in a certain proportion. The filter selection criteria constitute the antecedent clauses of the fuzzy rules, and the corresponding filters values constitute the consequent clauses of the fuzzy rules. The algorithm is based on the if – then – else rule paradigm referred to as FIRE (Fuzzy Inference Ruled by Else-action).

In order to enhance the image, the algorithm uses a spatial window which has a radius that can be defined by the user. This window (matrix) "slides" over the entire image, the central pixel of the matrix being enhanced using the influence of the other pixels of the window. This mask slides over the entire image except twice the radius of the sliding window number of lines or columns from the top, the bottom, the left and the right, because if the central point was on either of those, some of the elements of the matrix or some other values needed would not represent image pixels, but void. This application uses three filters:

- Filter A, edge sharpening filter:
- Filter B, impulse noise removal filter:
- Filter C, smoothing filter:

The FIRE system of rules used for this algorithm is:

If M is mic, then use filter B
If M is mare, then use filter C
Else use filter A

M is a coefficient that is computed for each pixel and its value depends directly on the values of the pixels in the sliding matrix and indirectly, and consequently in a smaller proportion, on every pixel in the region.

The membership functions for the linguistic labels "mare" and "mic" are defined on the domain of M ($0 \leq M \leq 1$). They are fuzzy membership functions that take an exponential form. These functions depend on two parameters: M and the image enhancement coefficient. Its value must be selected carefully considering the type of the image that is enhanced. Different values must be chosen for different types of images. In order to solve the if-then-else rules system, multiplication is selected as composition operator, minimum for aggregation, weighted average for the defuzzyfication function and the filter output for the consequent, so the following output is obtained:

$$I(X_i) = \frac{c_1 \cdot f_b + c_2 \cdot f_c + c_3 \cdot f_a}{c_1 + c_2 + c_3} \tag{1}$$

where: $I(X_i)$ = *the gray-level of pixel* X_i; c_1 = *mic (M)*; c_2 = *mare (M)*; $c_3 = 1 - max(c_1, c_2)$; f_a = *the output from filter A*; f_b = *the output from filter B*; f_c = *the output from filter C*;

After applying the defuzzyfication process to the entire image, the enhanced image is obtained. This image can now be used for other purposes, like, for instance the detection of the forms in the image by using the edge detection algorithm.

FUZZY EDGE DETECTION ALGORITHM

The algorithm used in this application is based on Russo and Ramponi (1994) algorithm. This edge detector is relatively immune to noise and is based on the if-then-else paradigm (FIRE). In order to detect the edges gray-level differences in a 3x3 neighbourhood are used as inputs to the fuzzy rules [14]. Let X denote the central pixel in the window and let:

$$x_i = I(X) - I(X_i), \text{ for } i = 1...8. \tag{2}$$

In this equation $I(X)$ represents the gray level in the central pixel while $I(X_i)$ represents the gray level of the i^{th} pixel in the matrix. This technique of the "sliding window" over the entire image is rather well-known through-out the whole image processing field [15]. The variables x_i are used in the antecedent clauses of the rule base. In the general case, there is a various number of rules in the system. Let c_i be the overall degree of satisfaction of the i^{th} "then" rule of the system. If N_i is the number of input variables in the i^{th} rule, $A_{ij}(\ )$ is the membership function of the linguistic label A_{ij} and the mean operator is used for aggregation (that is the way it is used in this algorithm) then:

$$c_i = \frac{1}{N_i} \cdot \sum_{j=1}^{N_i} A_{ij}(x_j) \tag{3}$$

Letting c_T denote the overall degree of satisfaction of the first M "if" rules (the system has M "if" rules and one "else" rule) and $c_E = c_{M+1}$ denote the degree of satisfaction of the "else" rule we have:

$$c_T = max(c_m) \quad m \in \{1..M\} \tag{4}$$

$$c_E = c_{M+1} = 1 - c_T = 1 - max(c_m) \quad m \in \{1..M\} \tag{5}$$

The output y is obtained by adding the effects induced by the "then" and "else" actions and by performing a suitable defuzzyfication. For this application, a simple edge detector was selected. It uses only one if rule. The FIRE system used is the following:

If (x_2 is zero) and (x_4 is zero) *Then (make y white)* *Else (make y black)*

The "zero" membership function is defined on the domain *[-G+1, G-1]*, where *G* is the number of color levels. The function is a fuzzy membership function that takes an exponential form. It depends on two parameters: x_i and the edge detection coefficient. The edge detection parameter must be selected carefully considering the type of the image on which the edge detection algorithm is applied. Different values must be chosen for different types of images. As a consequence of the fact that the system has only one "if" rule, the values for c_1 and c_2 are:

$$c_1 = \frac{a_1 + a_2}{2} \tag{6}$$

$$c_2 = 1 - c_1 \tag{7}$$

where: a_1=zero(x_2); a_2=zero(x_4)

The value of the i^{th} pixel of the edge image is obtained from the formula:

$$y = \frac{c_1 \cdot 255 + c_2 \cdot 0}{c_1 + c_2} = c_1 \cdot 255 \tag{8}$$

This matrix of 3x3 slides over the entire image except the first and last line and column, because if the central pixel was on either of those, some of the elements of the matrix would not represent image pixels, but void. The final image is a gray level image where the edges are represented by the darker pixels. After applying the edge detection algorithm, a threshold can be applied in order to convert the image to a black and white image where the edges of the objects in the original image are represented by black pixels.

FUZZY IMAGE SEGMENTATION

There are many techniques [7] of fuzzy image segmentation: histogram thresholding, edge based segmentation, region growing, fuzzy clustering algorithms, fuzzy rule-based approach, fuzzy integrals, measures of fuzziness and image information, fuzzy geometry, but among them the most dominant are fuzzy clustering and fuzzy rule based segmentation techniques [6].

Fuzzy C-Means Clustering Algorithm

Fuzzy C-means is an algorithm based on one of the oldest segmentation methods which allows data to have membership of multiple clusters, each to varying degrees [4]. This method, used in pattern recognition, was developed in 1973 by Dunn and improved by Bezdek in 1981 [1][2]. The algorithm is based on minimization of the following function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2 \quad , \ 1 \le m < \infty \tag{9}$$

where: *m is any real number greater than 1;* u_{ij} *is the degree of membership of* x_i *in the cluster j;* x_i *is the i-th of d-dimensional measured data;* c_j *is the d-dimension center of the cluster; ||*|| is any norm expressing the similarity between any measured data and the center*.

This algorithm [1] realizes an iterative optimization of the J_m function, updating membership u_{ij} and the cluster centers c_j using the following formulas:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|} \right)^{\frac{2}{m-1}}} \tag{10}$$

$$c_j = \frac{\sum_{j=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{j=1}^{N} u_{ij}^{m}} \tag{11}$$

The minimization of J_m is achieved only when the u_{ij} function saturates, that is, the stop criterion is given by the equation:

$$max_{ij} \left\{ \ \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| \ \right\} < \varepsilon \tag{12}$$

Where ε is a number between 0 and 1, and k is is the iteration step.

Fuzzy c-means algorithm has the following steps [16]:

1. Consider a set of n data points to be clustered, x_i.
2. Assume that the number of clusters, c, is known, $2 \leq c < n$.
3. Choose an appropriate level of cluster fuzziness, $m \in \Re > 1$.
4. Initialize the (n x c) sized membership matrix U to random values such as $u_{ij} \in [0,1]$ and $\sum_{j=1}^{c} u_{ij} = 1$.
5. Calculate the cluster centers c_j using (11) for j = 1... c.
6. Calculate the distance measures $d_{ij} = \left\| x_i^{(j)} - c_j \right\|$, for all clusters j = 1… c and data points i = 1… n.
7. Update the fuzzy membership matrix U according to d_{ij}.

If $d_{ij} > 0$ then $u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}} \right]^{-1}$

If $d_{ij} = 0$ then the data point x_j coincides with the cluster center c_j, and so full membership can be set $u_{ij} = 1$.

8. Repeat from step 5 until the change in U is less than a given tolerance, ε.

This algorithm's fuzzy behaviour is given by the membership function, which links the data to each cluster [11]. Some conditions must be respected in building the matrix U (conditions from step 4.). Matrix U factors represent the degree of membership between the centers of the clusters and the data. It is important to understand the importance of m. The smaller m is, the crisper the algorithm (m=1 represents a crisp algorithm), while the greater m, the fuzzier the clusters are (smaller values for the membership function, hence fuzzier clusters) [12].

IMPLEMENTATION

In order to test the algorithms, an application was developed. The proposed algorithms were implemented in Microsoft Visual C++ from the package Microsoft Visual Studio .NET.

The application is a visual application concerning image processing using fuzzy techniques. It has a menu from which the user can select the desired function and a toolbar from which images can be opened, saved or the about dialog can be also opened. The functions implemented by this program are mainly those presented in the algorithms in the earlier chapters. This application gives the user the possibility to open and work with .bmp grayscale images. No function is allowed before an image is loaded. Once an image loaded, the user can enhance the image, using his own parameters (in a specified range), draw the histogram of either the opened image or both it and the enhanced image and compare them. The user may choose to find the edges, threshold the original image or the image containing the edges. All of these functions are parameterized, parameters which can be chosen according to wishes of the client.

CONCLUSIONS AND FURTHER DEVELOPMENT

Developing image processing and, implicitly even more, fuzzy image processing application require a lot of heuristic procedures that implies the use of the user experience in the field of application. After experimental tests it was noticed that the obtained results also depend on the type of the image. That is, some algorithms give better results for some images, other algorithms for different images.

It was observed that the image enhancement algorithm has a tendency to give the smoothing filter a bigger weight for bigger sliding matrix radius, thus levelling in a small degree the image.

The edge detection algorithm although it might seem the simplest of the presented algorithms, it has the best results.

A short visual presentation of the results can be seen in Figure 1.

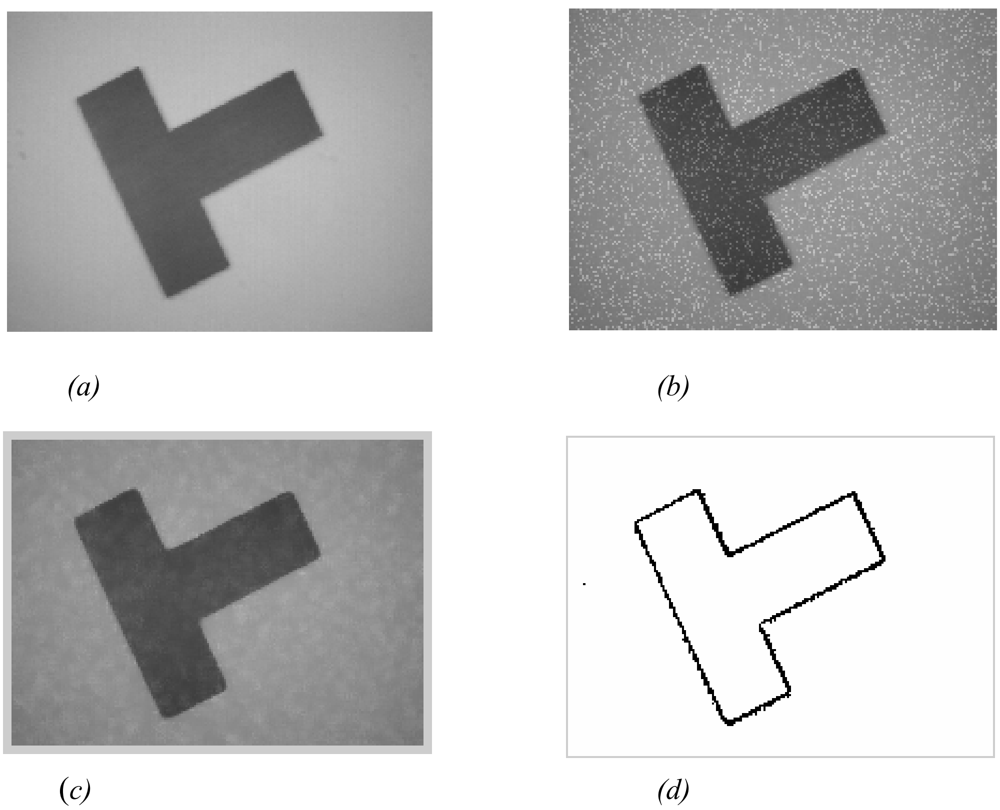

Figure 1 – (a) image without noise; (b) image with noise; (c) image (b) enhanced; (d) image (c) edge detected

As it is structured on “specialized” functions (small parts that compute specific tasks), the implementation of these algorithms makes the adding of further functions (algorithms) easier, the new functions wouldn’t need to be fully implemented because they make use of some parts of those algorithms that were already implemented, only the specific part of the new tool having to be created at that time. Further development of the project would be to add new tools (like fuzzy thresholding, pattern recognition from the general perspective of computer vision) and improving the already implemented ones by adding new rules to the FIRE systems or modifying the weights computing functions. The design of these algorithms was intended as another proof, although not necessary, that the fuzzy approach is an important alternative way to consider when developing an image processing application.

REFERENCES

[1] Bezdek, J. C. *Pattern Recognition with Fuzzy Objective Function Algoritms* (Plenum Press, New York, 1981).

[2] Bezdek, J. C., S. Pal (eds.), *Fuzzy Models for Pattern Recognition* (IEEE Press, 1991).

[3] Choi, Y. S., Krishnapuram, R., *Image Enhancement Based On Fuzzy Logic* (Proceedings of the IEEE International Conference on Image Processing - ICIP '95, 1995).

[4] Cocquerez, J-P., S. Philipp, *Analyse d'images: filtrage et segmentation* (Masson, Paris, 1995).

[5] Cojocaru, D., *Achiziţia, prelucrarea şi recunoaşterea imaginilor* (Editura Universitaria, Craiova, 2002).

[6] Cuevas, E., D. Zaldivar, R. Rojas, *Fuzzy segmentation applied to face segmentation* (Intelligent Bipedal Robot Project, Technical Report B-04-09, 2004).

[7] Hamid R. Tizhoosh, *Fuzzy Image Processing: Introduction in Therory and Practice* (Springer-Verlag, 1997).

[8] Ivănescu, M., *Mircea Ivănescu Şcoala de Robotică din Craiova. Sisteme de conducere fuzzy.* (Editura Universitaria, Craiova, 2003).

[9] Kerre, E. E., Nachtegael, M., *Fuzzy Techniques in Image Processing* (Springer Verlag, 2000).

[10] Krishnapuram, R., *Handbook of Fuzzy Computation – Computer Vision* (IOP Publ., 1998).

[11] Pham D.L and Prince J.L., *An adaptive fuzzy C-means algorithm for image segmentation in the presence of intensity inhomogeneities* (Pattern Recognition Letters vol. 20, no.1, pp.57-68, 1999).

[12] Tolias Y.A and Panas S.M., *On applying spatial constraints in fuzzy image clustering using a fuzzy rule-based system* (IEEE Signal Processing Letters, vol.5, no.10, pp. 245-247, 1998).

[13] Umbaugh, S.E., *Computer Vision, and Image Processing* (Prentice Hall, 1998).

[14] Ville, D. Van der, Nachtegael, M., Weken, D. Van der., Kerre, E. E., Philips and W., Lemahieu, I., *Noise Reduction by Fuzzy Image Filtering* (IEEE Transactions On Fuzzy Systems, Vol. 11, No. 4, 2003).

[15] Wang, L. X., *A Course in Fuzzy Systems and Control* (Prentice Hall PTR, 1997).

[16] Yen J. and Langari R., *Fuzzy logic, intelligence, control and information* (Prentice Hall, New York, 2000).