Journal of Theoretical and Applied Information Technology, Islamabad PAKISTAN

31st May 2010. Vol.15. No.2.

www.jatit.org

NOISE REDUCTION BY USING FUZZY IMAGE FILTERING

[1]MAHESH T R, [1]PRABHANJAN S, [2]M VINAYABABU

[1]Asst Prof., Department ISE, SBMJCE, Bangalore, India

[2]Assoc. Prof., Department of MCA, IARE, Hyderabad, India

ABSTRACT

The existing system available for fuzzy filters for noise reduction deals with fat-tailed noise like impulse noise and median filter. Only impulse noise reduction uses fuzzy filters. Gaussian noise is not specially concentrated; it does not distinguish local variation due to noise and due to image structure. The proposed system presents a new technique for filtering narrow-tailed and medium narrow-tailed noise by a fuzzy filter. The system first estimates a “fuzzy derivative” in order to be less sensitive to local variations due to image structures such as edges. Second, the membership functions are adapted accordingly to the noise level to perform “fuzzy smoothing.” A new fuzzy filter is presented for the noise reduction of images corrupted with additive noise. The filter consists of two stages. The first stage computes a fuzzy derivative for eight different directions. The second stage uses these fuzzy derivatives to perform fuzzy smoothing by weighting the contributions of neighboring pixel values. Both stages are based on fuzzy rules which make use of membership functions. The filter can be applied iteratively and effectively reduce heavy noise. In particular, the shape of the membership functions is adapted according to the remaining noise level after each iteration, making use of the distribution of the homogeneity in the image. A statistical model for the noise distribution can be incorporated to relate the homogeneity to the adaptation scheme of the membership functions. Experimental results are obtained to show the feasibility of the proposed approach. These results are also compared to other filters by numerical measures and visual inspection.

Keywords: *AFuzzy sets, Fuzzy filters, Fuzzy smoothing, Fuzzy derivative*

1. INTRODUCTION

The application of fuzzy techniques in image processing is a promising research field. Fuzzy techniques have already been applied in several domains of image processing (e.g., filtering, interpolation, and morphology), and have numerous practical applications (e.g., in industrial and medical image processing). In this project, we will focus on fuzzy techniques for image filtering. Already several fuzzy filters for noise reduction have been developed, e.g., the well-known FIRE-filter from, the weighted fuzzy mean filter from, and the iterative fuzzy control based filter from. Most fuzzy techniques in image noise reduction mainly deal with fat-tailed noise like impulse noise. These fuzzy filters are able to outperform rank-order filter schemes (such as the median filter). Nevertheless, most fuzzy techniques are not specifically designed for Gaussian (-like) noise or do not produce convincing results when applied to handle this type of noise.

Therefore, this paper presents a new technique for filtering, narrow-tailed and medium narrow-tailed noise by a fuzzy filter. Two important features are presented: first, the filter estimates a “fuzzy derivative” in order to be less sensitive to local variations due to image structures such as edges; second, the membership functions are adapted accordingly to the noise level to perform “fuzzy smoothing.” For each pixel that is processed, the first stage computes a fuzzy derivative. Second, a set of 16 fuzzy rules is fired to determine a correction term. These rules make use of the fuzzy derivative as input. Fuzzy sets are employed to represent the properties and while the membership functions are fixed, the membership functions are adapted after each iteration. The adaptation scheme is extensive and can be combined with a statistical model for the noise. The result of this method can be compared with those obtained by other filters.

2. FUZZY RULES

Human beings make decisions based on rules. Although, we may not be aware of it, all the decisions we make are all based on computer like if-then statements. If the weather is fine, then we may decide to go out. If the forecast says the weather will be bad today, but fine tomorrow, then we make a decision not to go today, and postpone it till tomorrow. Rules associate ideas and relate to one another.

Fuzzy machines, which always tend to mimic the behavior of man, work the same way. However, the decision and the means of choosing that decision are replaced by fuzzy sets and the rules are replaced by fuzzy rules. Fuzzy rules also operate using a series of if-then statements. For instance, if X then A, if y then b, where A and B are all sets of X and Y. Fuzzy rules define fuzzy *patches*, which is the key idea in fuzzy logic.

A machine is made smarter using a concept designed by Bart Kosko called the Fuzzy Approximation Theorem (FAT). The FAT theorem generally states a finite number of patches can cover a curve as seen in the Figure 1.1. If the patches are large, then the rules are sloppy. If the patches are small then the rules are fine.

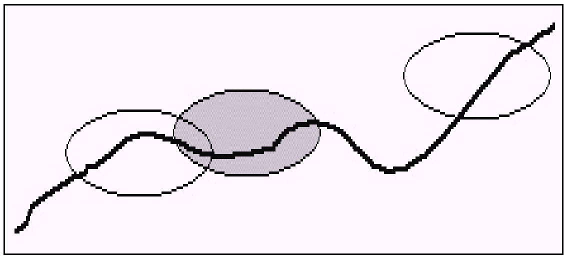


Figure 1 Fuzzy Patches covering a curve

2.1 Fuzzy Patches

In a fuzzy system this simply means that all our rules can be seen as patches and the input and output of the machine can be associated together using these patches. Graphically, if the rule patches shrink, the fuzzy subset triangles get narrower, since novice can build control systems that beat the best math models of control theory. Naturally, it is math-free system.

2.2 Fuzzy Control

Fuzzy control, which directly uses fuzzy rules, is the most important application in fuzzy theory. Using a procedure originated by Ebrahim Mamdani in the late 70s, three steps are taken to create a fuzzy controlled machine:

A) Fuzzification (Using membership functions to graphically describe a situation).

B) Rule evaluation (Application of fuzzy rules).

C) Defuzzification (Obtaining the crisp or actual results).

As a simple example on how fuzzy controls are constructed, consider the following classic situation: the inverted pendulum. Here, the problem is to balance a pole on a mobile platform that can move in only two directions, to the left or to the right. The angle between the platform and the pendulum and the angular velocity of this angle are chosen as the inputs of the system. The speed of the platform hence is chosen as the corresponding output.

Step 1:

First of all, the different levels of output (high speed, low speed etc.) of the platform are defined by specifying the membership functions for the fuzzy sets. The graph is shown below.

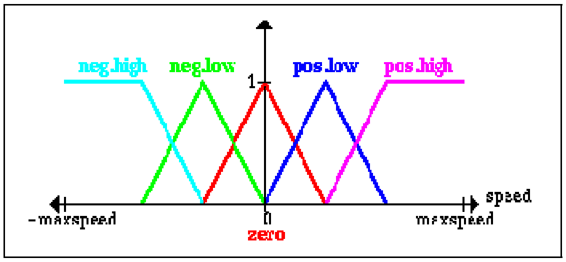


Figure 2 Graph of Fuzzy Set

Similarly, the different angles between the platform and the pendulum and

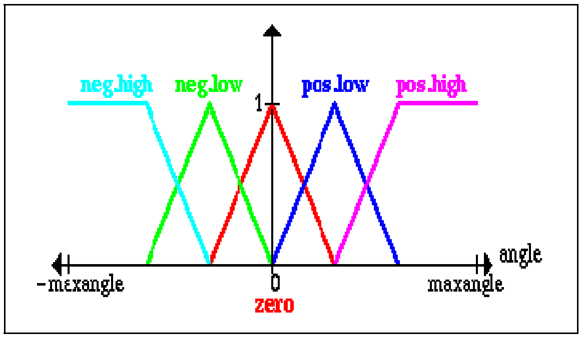


Figure 3 Angles between platform and pendulum

The angular velocities of specific angles are also defined

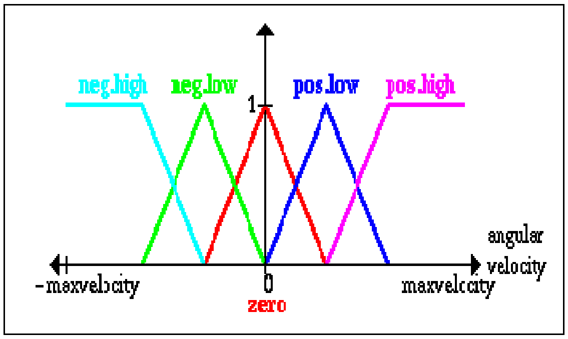


Figure 4 Angular Velocities of Specific angles

Step 2:

The next step is to define the fuzzy rules. The fuzzy rules are merely a series of if-then statements as mentioned above. These statements are usually derived by an expert to achieve optimum results. Some examples of these rules are:

i) If angle is zero and angular velocity is zero then speed is also zero.

ii) If angle is zero and angular velocity is low then the speed shall be low.

An application of these rules is shown using specific values for angle and angular velocities. The values used for this example are 0.75 and 0.25 for zero and positive-low angles and 0.4 and 0.6 for zero and negative-low angular velocities.

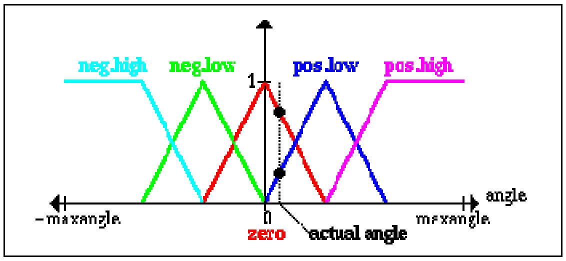


Figure 5 Applications of Rules – Graph1

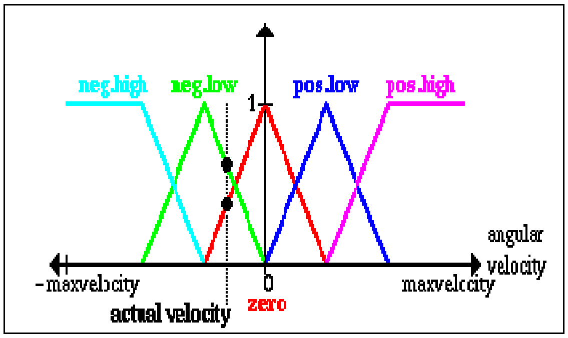


Figure 6 Applications of Rules – Graph2

Consider the rule "if angle is zero and angular velocity is zero, the speed is zero". The actual value belongs to the fuzzy set zero to a degree of 0.75 for "angle" and 0.4 for "angular velocity". Since this is an AND operation, the minimum criterion is used , and the fuzzy set zero of the variable "speed" is cut at 0.4 and the patches are shaded up to that area. This is illustrated in below graph.

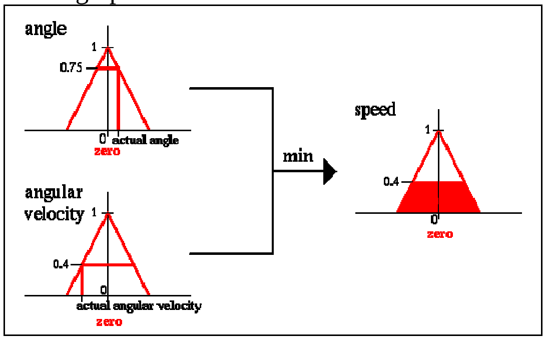
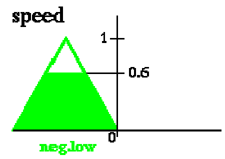
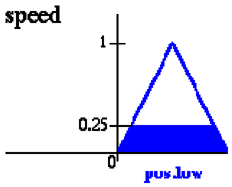


Figure 7 AND operation on Angle and Angular velocity

Similarly, the minimum criterion is used for the other three rules. The following figures show the result patches yielded by the rule "if angle is zero and angular velocity is negative low, the speed is negative low", "if angle is positive low and angular velocity is zero, then speed is positive low" and "if angle is positive low and angular velocity is negative low, the speed is zero".





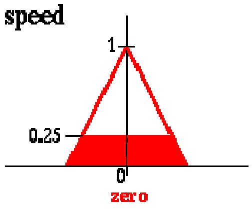


Figure 8 Three Rules Result

The four results overlaps and is reduced to the following resulting graph shown in Fig 1.9.

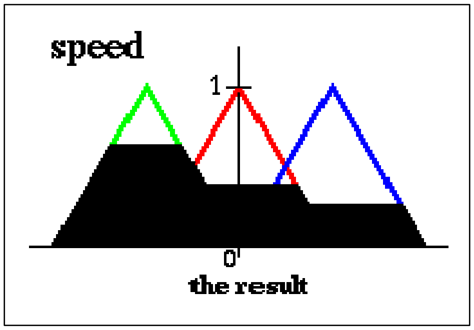


Figure 9 Resulting graph

Step 3:
The result of the fuzzy controller as of know is a fuzzy set (of speed). In order to choose an appropriate representative value as the final output (crisp values), defuzzification must be done. There are numerous defuzzification methods, but the most common one used is the center
of gravity of the set as shown in Fig 1.10.

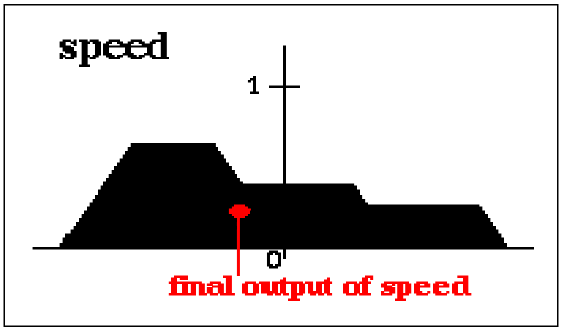


Figure 10 Defuzzification Method

3. PROPOSED SYSYTEM

The proposed system presents a new technique for filtering narrow-tailed and medium narrow-tailed noise by a fuzzy filter. The system

- First estimates a “fuzzy derivative” in order to be less sensitive to local variations due to image structures such as edges
- Second, the membership functions are adapted accordingly to the noise level to perform “fuzzy smoothing.”

For each pixel that is processed, the first stage computes a fuzzy derivative. Second, a set of 16 fuzzy rules is fired to determine a correction term. These rules make use of the fuzzy derivative as input.
Fuzzy sets are employed to represent the properties,, and While the membership functions for and are fixed, the membership function for is adapted after each iteration.

3.1 Noise Model

As previously mentioned, the scope of this project is limited to the study of impulse noise removal. Several distinct models exist for impulse noise.

- "Pure" impulsive noise, i.e., impulsive noise in which a corrupted pixel takes on a gray scale value of either 0 or 255, is relatively simple to remove, since such maximal and minimal gray scale values occur relatively infrequently in actual images.
- A more realistic and challenging, from a noise removal standpoint, is "bit error" impulsive noise. Bit error impulsive noise can be quantitatively described as follows:

Given an uncorrupted two-dimensional signal, s, such that each signal value is quantized to B bits, each s(i , j) can be written as:

$$\mathbf{s(i,j) = k_1(i,j)\,2^{B-1} + k_2(i,j)\,2^{B-2} + \ldots + k_{B-1}(i,j)\,2 + k_B(i,j)}$$

where km (i , j)   [0,1] for all 1 < m < B and all i , j.

If we now assume that each coefficient of each s(i , j) is corrupted with probability p independently of all other coefficients in s(i , j) and of all other signal values, then the corrupted signal values, x(i , j), are of the form:

$$x(i,j) = k_1^*(i,j)\,2^{B-1} + k_2^*(i,j)\,2^{B-2} + \ldots + k_{B-1}^*(i,j)\,2 + k_B^*(i,j)$$

where km*(i , j) = { km(i , j), with probability1-p; 1-km(i , j), with probability p }

4. IMPLEMENTATION

Portable Gray Map (PGM) is a standard bitmap based format consisting of a 4 lines header, and data stored in the unsigned char type, providing a maximum of 256 gray scale levels or 8-bit data per pixel. The general structure of a PGM image file
The header of a PGM image file consists of:

- First line containing the signature of the image file and identifies the file as PGM
- Second line is the comment line
- Third line provides information about number and rows and columns of data stored in the file, and
- Fourth line specifies maximum gray level contained in the image

4.1 Mean Filter Module

Mean filtering is a simple, intuitive and easy to implement method of smoothing images, *i.e.* reducing the amount of intensity variation between one pixel and the next. It is often used to reduce noise in images.
The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings. Mean filtering is usually thought of as a convolution filter. Like other convolutions it is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean.

4.2 Median Filter Module

The median filter and its variants are among the most commonly used filters for impulse noise removal. The median filters, when applied uniformly across the image, tend to modify both noisy as well as noise free pixels, resulting in blurred and distorted features. Recently, some modified forms of the median filter have been proposed to overcome these limitations. In these variants of the median filter, the pixel value is modified only when it is found corrupted with noise.
The median filter is used to remove noise from an image by replacing pixels with the middle pixel value selected from a certain window size. The median filter is very effective at removing noise while not destroying sharp edges in an image.

The general idea behind the filter is to average a pixel using other pixel values from its neighborhood, but simultaneously to take care of important image structures. The main concern of the proposed filter is to distinguish between local variations due to noise and due to image structure. In order to accomplish this, for each pixel we derive a value that expresses the degree in which the derivative in a certain direction is small. Such a value is derived for each direction corresponding to the neighboring pixels of the processed pixel by a fuzzy rule.

4.2.1 Fuzzy Derivative Estimation

Estimating derivatives and filtering can be seen as a chicken-and-egg problem; for filtering we want a good indication of the edges, while to find these edges we need filtering.

```
// calc fuzzy derivative by applying fuzzy rule for 'small' fuzzy set

          for(int t=0;t<8;t++)
fuzzyderiv[t]=SmallFuzzySet.ApplyRule(fofxsmall[0][t],
          fofxsmall[1][t],fofxsmall[2][t]);
```

4.2.2 Fuzzy Smoothing

To compute the correction term for the processed pixel value, we use a pair of fuzzy rules for each direction. The idea behind the rules is the following: if no edge is assumed to be present in a certain direction, the (crisp) derivative value in that direction can and will be used to compute the correction term. The first part (edge assumption) can be realized by using the fuzzy derivative value, for the second part (filtering) we will have to distinguish between positive and negative values.

```
//fuzzy smoothing
//calc membership values for 'positive' and 'negative' fuzzy sets
          for(int t=0;t<8;t++)
```

```
fofxpositive[t]=PositiveFuzzySet.fofx(simpderiv
[0][t],L);

fofxnegative[t]=NegativeFuzzySet.fofx(simpder
iv[0][t],L);
```

4.2.3 Defuzzification

```
//calc correction term (defuzzification)

        delta=0.0;
        for(int t=0;t<8;t++)
        delta=delta+(positivetruthness[t]-
    negativetruthness[t]);
        delta=((double)L/8.0)*delta;
        int outval=inval+(int)delta;
        imgout.setPixel(r,c,outval);
```

5. CONCLUSION

The fuzzy filter for additive noise reduction has its main feature, is that it distinguishes between local variations due to noise and due to image structures, using a fuzzy derivative estimation. Fuzzy rules are fired to consider every direction around the processed pixel. Additionally, the shape of the membership functions is adapted according to the remaining amount of noise after each iteration.

Experimental results show the feasibility of the new filter and a simple stop criterion. Although it's relative simplicity and the straightforward implementation of the fuzzy operators, the fuzzy filter is able to compete with state-of-the-art filter techniques for noise reduction. Finally, the fuzzy filter scheme is sufficiently simple to enable fast hardware implementations. By exploiting fuzzy reasoning at two different stages, the filter is able to perform a very strong noise cancellation without degrading the image structure. Thus the fuzzy filter is being done with fuzzy derivatives and the fuzzy smoothness is carried for the filtering.

REFRENCES:

[1] Plataniotis K., Androutsos D., and Venetsanopoulos A., "Adaptive Fuzzy Systems for Multichannel Signal Processing",Proceedings of the IEEE, Vol. 87, No. 9, September, 1999.

[2] Kam H., Hanmandlu M., and Tan W., "An Adaptive Fuzzy System for Smoothing Noisy Images", IEEE, 2003

[3] Ville D., Nachtegael M., Weken D., Kerre E., Philips W., and Lemahieu I., "Noise Reduction by Fuzzy Image Filtering", IEEE Transactions on Fuzzy system, Vol. II, No.4, August, 2003.

[4] E. Kerre and M. Nachtegael, Eds., Fuzzy Techniques in Image Processing. New York Springer-Verlag, 2000, vol. 52, Studies in Fuzziness and Soft Computing.

[5] "A fuzzy filter for images corrupted by impulse noise," IEEE Signal Processing Lett., vol. 3, pp. 168–170, June 1996.

[6] F. Russo and G. Ramponi, "A fuzzy operator for the enhancement of blurred and noisy images," IEEE Trans. Image Processing, vol. 4, pp. 1169–1174, Aug. 1995.

[7] S. Bothorel, B. Bouchon, and S. Muller, "A fuzzy logic-based approach for semiological analysis of microcalcification in mammographic images," Int. J. Intell. Syst., vol. 12, pp. 819– 843, 1997.