A Statistical Novel Approach for Image Correction & Noise Reduction Using Fuzzy Logic

J.Sudhakar[1], J. Hari[2]
[1] Department of Electronics and Communication Engineering
Vignan's Institute of Engineering for Women, Vishakapatnam

[2] Department of Computer Science Engineering
Vignan's Institute of Engineering for Women, Vishakapatnam

Abstract— **This paper deals with fuzzy techniques for image filtering. Fuzzy techniques have already been applied in several domains of image processing for example filtering, interpolation, and morphology and have numerous practical applications such as in industrial and medical image processing. In this project, we will focus on 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 to 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

IJERT

I. INTRODUCTION

The application of fuzzy techniques in image processing is a promising research field. Among the different topics, this project will focus on the construction and application of fuzzy filters for image processing. It is well known that fuzzy filters have a more robust performance than classical filters. For example, most classical filters that remove noise simultaneously blur the edges, while fuzzy filters have the ability to combine edge-preservation and smoothing. Compared to other non-linear techniques, fuzzy filters are able to represent knowledge in a comprehensible way. Several researchers have already presented interesting fuzzy filters with promising results. The aim of this project is to give an extended survey of the current state-of-the-art and to indicate future key points for research.

Fuzzy techniques have already been applied in several domains of image processing and have numerous practical applications such as 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., well-known FIRE-filter from, the weighted fuzzy mean filter from, and the iterative fuzzy control based filter .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.

1.1 Noise in images

Real images are often degraded by some random errors ---this degradation is usually called noise. Noise can occur during image capturing, transmission, or processing, and may be dependent on, or independent of image content. Noise is usually described by its probabilistic characteristics. Idealized noise, called White noise, is frequently applied as worst approximation of degradation, its advantage being that it uses simplifies calculations. A special case of white noise is Gaussian noise. Gaussian noise is a very good approximation to noise that occurs in many practical cases. Therefore, this project 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 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. While the membership functions for and are fixed, the membership function for is 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.

1.2 Image Smoothing

Image smoothing is the set of local pre-processing methods whose predominant use is the suppression of image noise---it uses redundancy in the image data. Calculation of the new value is based on the averaging of brightness values in some neighbouring. Smoothing poses the problem of blurring sharp edges in the image, and so we shall concentrate on smoothing methods which are edge preserving. They are based on the general idea that the average is computed only from those points in the neighboring which have similar properties to the point being processed.

Local image smoothing can effectively eliminate impulse noise or degradations appearing as thin strips, but does not work if degradations are large blobs or thick strips. Smoothing approaches based on direct averaging blur image edges. More sophisticated approaches reduce blurring by averaging in homogeneous local neighborhoods.

II. FRAMEWORK

The system "Noise Reduction by Fuzzy Image Filtering" presents a new technique for filtering narrow-tailed and medium narrow-tailed noise by a fuzzy filter. Two important features are: 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 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. While the membership functions for and are fixed, the membership function for is adapted after each iteration. 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 such as edges. 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.

The further construction of the filter is then based on the observation that a small fuzzy derivative most likely is caused by noise, while a large fuzzy derivative most likely is caused by an edge in the image. Consequently, for each direction we will apply two fuzzy rules that take this observation into account (and thus distinguish between local variations due to noise and due to image structure), and that determine the contribution of the neighboring pixel values. The result of these rules is defuzzified and a "correction term" is obtained for the processed pixel value. This paper "Noise Reduction by Fuzzy Image Filtering" is based on the following modules: Mean Filter, Median Filter and Fuzzy Filter

Mean Filter is such that the image is to be filtered using the averaging of a pixel using other pixel values from its neighborhood. Then we calculate mean value which gives the desired output.

Median Filter is such that the image is to be filtered using the averaging of a pixel using other pixel values from its neighborhood. Calculating median filter is based on sorting the pixels values from is neighbourhood, which gives the desired output.

Fuzzy Filter, 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. First it performs fuzzy derivate and then calculates the membership functions to perform fuzzy smoothing.

IJERT

III FUZZY LOGIC FOR DIGITAL IMAGES

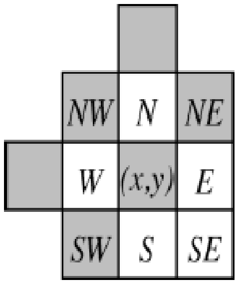
In this Paper Noise Reduction by Fuzzy Image Filtering, we focus on fuzzy techniques for image filtering.

Fuzzy Filter. 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 such as edges. 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.

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.

NW	N	NE
W	(x,y)	E
SW	S	SE

The fuzzy derivatives are calculated from the co-ordinates mentioned in the gray shading.



The pixel involved to calculate the fuzzy derivative in each direction as mentioned in the below table.

Direction	Position	Set w.r.t (x,y)
NW	(x-1,y-1)	{(-1,1),(0,0),1,-0)}
W	(x-1,y)	{(0,1),(0,0),(0,-1)}
SW	(x-1,y+1)	{(1,1)(0,0),(-1,-1)}
S	(x,y+1)	{(1,0),(0,0),(-1,0)}
SE	(x+1,y+1)	{(1,-1),(0,0),(-1,1)}
E	(x+1,y)	{(0,-1),(0,0),(0,1)}
NE	(x+1,y-1)	{(-1,-1),(0,0),(1,1)}
N	(x,y-1)	{(-1,0),(0,0),(1,0)}

In our approach, we start by looking for the edges. We try to provide a robust estimate by applying fuzzy rules. The simple derivative at the central pixel position (x, y) in the direction D, where D belongs to dir = {NE, W,SW,S, SE, E, NE, N} is defined as the difference between the pixel at (x,y) and its neighbor in the direction D. This derivative value is denoted by $\nabla_D(x,y)$.

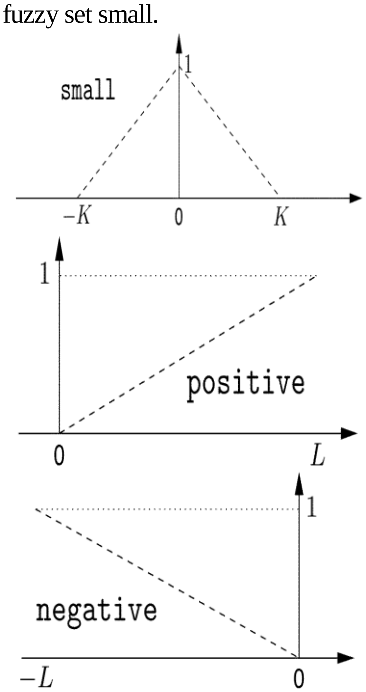
∇_N (x, y) = I(x,y-1) – I(x, y)

∇_{NW} (x, y) = I(x-1,y-1) – I(x, y)

IJERT

Next, the principle of the fuzzy derivative is based on the following observation. Consider the edge passing through the neighborhood of pixel (x, y) in the direction SW – NE direction. The derivative value ∇_{NW} will be large, but also derivative values of neighbor pixels perpendicular to the edge's direction can expected to be large. The idea is to cancel out the effect of one derivative value which turns out to be high due to noise. Therefore, if two out of three derivative values are small, it is safe to assume that no edge is present in the considered direction. This observation will be taken into account when we formulate the fuzzy rule to calculate the fuzzy derivative values.

In the table, an overview of the pixels we use to calculate the fuzzy derivative for each direction. Each direction (column 1) corresponds to a fixed position (column 2); the sets in column 3 specify which pixels are considered with respect to the central pixel . To compute the value that expresses the degree to which the fuzzy derivative in a certain direction is small, we will make use of the fuzzy set small.



Membership functions (a)small(b)positive(c)negative

The membership function for the property is the following [see Fig.(a)]:

$$m_K(u) = \begin{cases} 1 - \frac{u}{K} & ,0 \le u| \le K \\ 0 & ,|u > K \end{cases}$$

where K is an adaptive parameter .

ISSN: 2278-0181
Vol. 1 Issue 8, October - 2012

For example, the value of the fuzzy derivative ∇^F_{NW}(x,y) for the pixel (x,y) in the NW-direction is calculated by applying the following rule:

If
($\nabla_{NW}(x,y)$ *is small and* $\nabla_{NW}(x-1,y+1)$ *is small*) *or*
($\nabla_{NW}(x,y)$ *is small and* $\nabla_{NW}(x+1,y-1)$ *is small*) *or*
($\nabla_{NW}(x-1,y+1)$ *is small and* $\nabla_{NW}(x+1,y-1)$ *is small*
Then $\nabla^F_{NW}(x,y)$ *is small*

Eight such rules are applied, each computing the degree of membership of the fuzzy derivatives $\nabla^F_D(x,y), D \in dir$, to the set 'small'.These rules are implemented using the minimum to represent the AND-operator, and the maximum for the OR-operator. A defuzzification is not needed since the second stage, i.e., the fuzzy smoothing, directly uses the membership degrees to 'small'.

The robustness we try to achieve by the fuzzy derivative is by combining multiple simple derivatives around the pixel and by making the parameter K adaptive.

We start by dividing the image in small NxN non-overlapping blocks. For each block B, we compute a rough measure for the homogeneity of this block by considering the maximum and minimum pixel value.

$$\mu = 1 - \frac{max_{(x,y)\epsilon B} I(x,y) - min_{(x,y)\epsilon B} I(x,y)}{L}$$

This measure is commonly used in the context of fuzzy image processing.

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. For example, let us consider the direction NW. Using the values $\nabla^F_{NW}(x,y)$ and $\nabla_{NW}(x,y)$, we fire the following two rules, and compute their truthiness λ^+_{NW} and λ^-_{NW} :

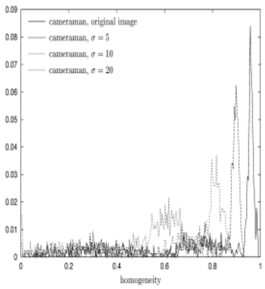
λ^+_{NW} : if
$\nabla^F_{NW}(x,y)$ *is small and* $\nabla_{NW}(x,y)$ *is positive*
then c is positive

λ^-_{NW} : if
$\nabla^F_{NW}(x,y)$ *is small and* $\nabla_{NW}(x,y)$ *is negative*
then c is negative

The final step in the computation of the fuzzy filter is the defuzzification. We are interested in obtaining a correction term Δ , which can be added to the pixel value of location (x,y) . Therefore, the truthness of the rules $\lambda^+_D(x,y)$ and $\lambda^-_D(x,y), D \in dir$ (so for all directions) are aggregated by computing and rescaling the mean truthness as follows:

$$\Delta = \frac{L}{8} \sum_{D \in dir} (\lambda^+_D - \lambda^-_D)$$

Where dir contains the directions and L represents the number of gray levels. So, each directions contributes to the correction term Δ. The proposed filter is applied to grayscale test images (8-bit), after adding Gaussian noise of different levels. Such a procedure allows us to compare and evaluate the filtered image against the original one.



The original test image of Cameraman
Histogram of the homogeneity

This figure shows the normalized histogram of the homogeneity of "cameraman," for the original image, but also for the image corrupted with different noise levels, i.e., $\sigma = 5$, $\sigma = 10$, and $\sigma = 20$.

Finally, we like to show a practical application of the fuzzy filter. In particular, this image restoration scheme could be used to enhance satellite images. Of course, since the original image is already corrupted by noise, it is not possible to obtain a numerical measure which indicates how "good" the image is. Depending on the application (e.g., visual inspection, segmentation), one could prefer lighter or heavier filtering.
Representation of images

IJERT

ISSN: 2278-0181
Vol. 1 Issue 8, October - 2012

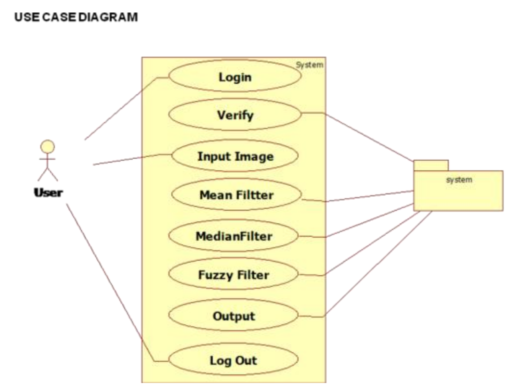
Every black-and-white picture in the real world is a continuous function of brightness values. For every point of a picture, we may determine the corresponding gray-level value, which shows the brightness of this particular point. When storing an image in a computer, we specify the brightness for a finite number of selected points, thus approximating the real picture; the points that form a digitized image are called pixels. Scanners and digital cameras automatically convert a continuous picture into a set of points. The simplest technique to select points is to use a regular square grid: a scanner overlays a uniform grid on a given image and determines the brightness at the center of each square. The gray-level values are usually represented by integer numbers, ranging from 0 for black to 255 for white. The PGM format is a lowest common denominator grayscale file format. It is designed to be extremely easy to learn and write programs for. There are no data, delimiters, or padding before, after, or between images. A PGM image represents a grayscale graphic image. There are many psueudo-PGM formats in use where everything is as specified herein except for the meaning of individual pixel values. For most purposes, a PGM image can just be thought of an array of arbitrary integers, and all the programs in the world that think they're processing a grayscale image can easily be tricked into processing something else.A PGM file consists of a sequence of one or more PGM images. There are no data, delimiters, or padding before, after, or between images.
Each PGM image consists of the following:

* A 'magic number' for identifying the file type. A pgm image's magic number is the two characters 'P5'.
* Whitespace (blanks, TABs).
* A width, formatted as ASCII characters in decimal.
* A height, again in ASCII decimal.
* The maximum gray value (Maxval), in ASCII decimal. Must be less than 65536,

IV. PROBLEM DEFINITION

Use cases are used during requirement elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from an external point of view. A use case describes a function provided by the system that yields a visible result for an actor. An actor describes any entity that interacts with the system. The identification of actors and use cases results in the definition of the boundary of the system, which is, in differentiating the tasks accomplished by the system and the tasks accomplished by its environment. The actors are outside the boundary of the system, where as the use cases are inside the boundary of the system.

Actors are external entities that interact with the system. Use cases describe the behavior of the system as seen from an actor’s point of view. Actors initiate a use case to access the system functionality. The use case then initiates other use cases and gathers more information from the actors. When actors and use cases exchange information, they are said to be communicate.

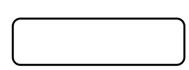


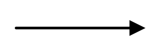
When the user login to the project the system first verifies the username and password, if its true then the user can enter into the project. The user gives the input image to the system as it can perform either of the filter as provided as Mean filter, Median filter, Fuzzy filter. Then the system displays the output as required by the user and finally the user can logout from the project.

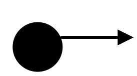
IJERT

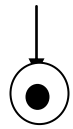
ACTIVITY DIAGRAM

An activity diagram describes the behavior of the system in terms of activities. Activities are modeling elements that represent the execution of set of operations. The completion of these operations triggers a transition to another activity. Activity diagrams similar to flowchart diagrams in that they can be used to represent control flow (i.e., the order in which operations occurs) and data flow (i.e., the objects that are exchanged among operations). Activities are represented by rounded rectangles and arrows represent transition between activities. Thick bars represent the synchronization of the control flow.


Activities

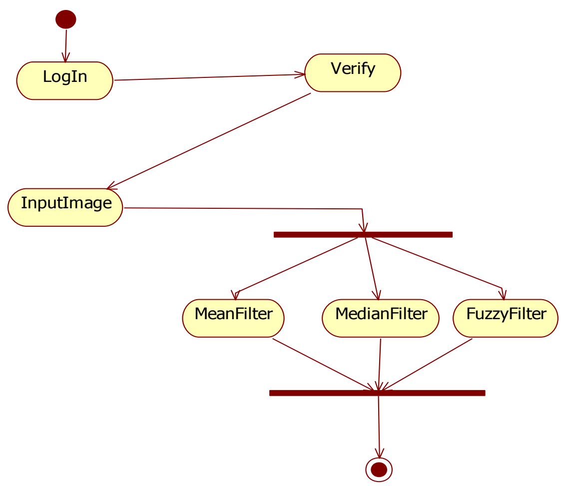

Transactions


Start



End

ACTIVITY DIAGRAM

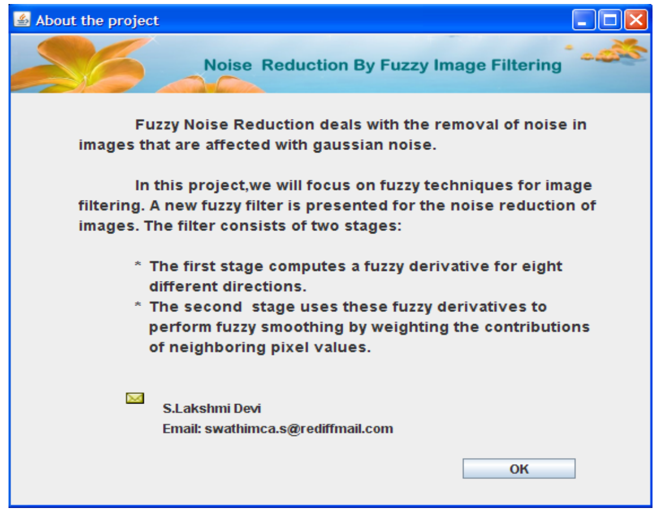


First the user gets login to the project and the system verifies the username and password. If its true then the user can give input image and can apply any one of the filter as required by the user.

V. RESULTS AND DISCUSSIONS

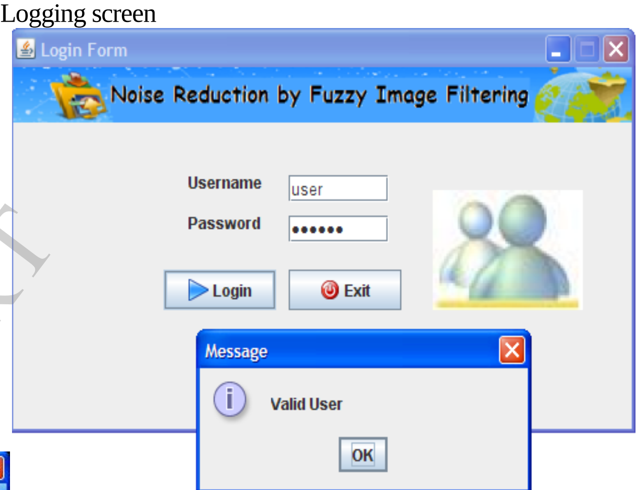
The starting screen of the project which guides the user to proceed to the next step.The Login is available to login to the project and About is given to know about the project.

About the project



A brief note is given to know about the project.

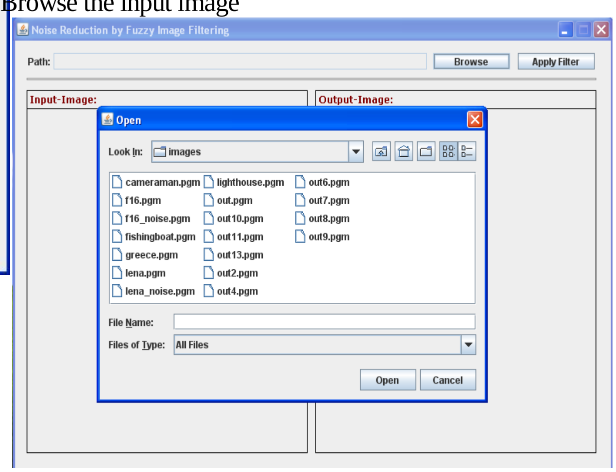
Logging screen



IJERT

The Login screen is given such that where the user can get through the project by specifying the username and password. If it is true then it displays a message as valid user and the user can get services.

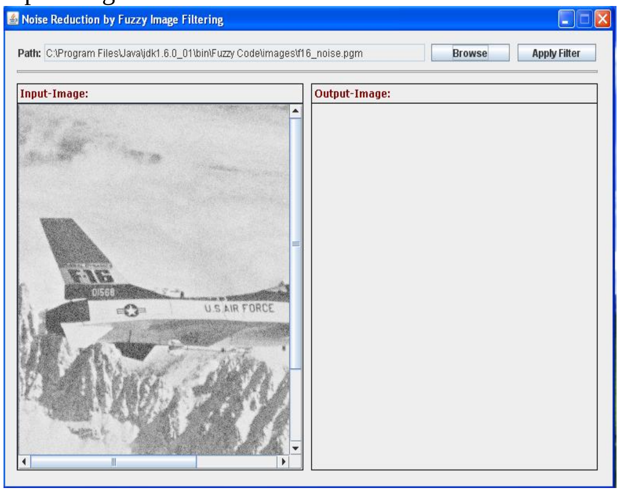
Browse the input image



ISSN: 2278-0181
Vol. 1 Issue 8, October - 2012

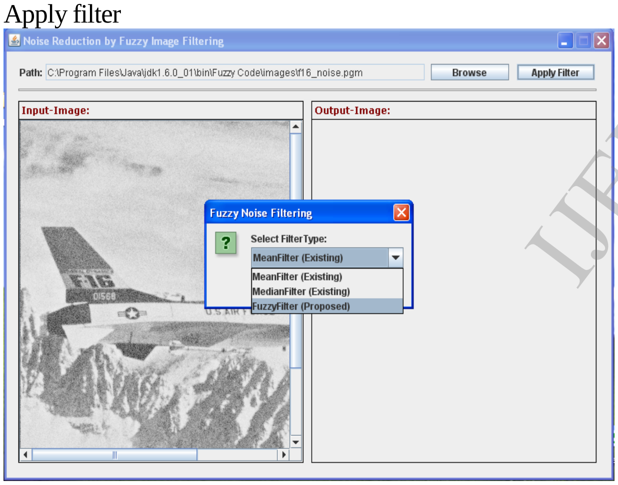
Here, on clicking browse button the user can give input image from the training set data. The input image will be displayed in the space provided.

Input Image



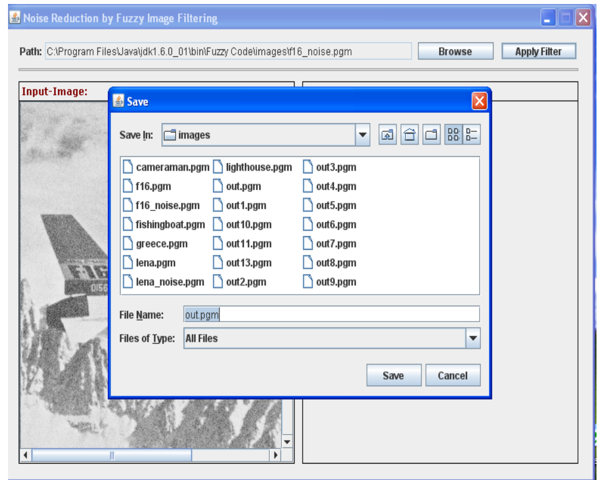
The input image selected from the training set data is displayed on the screen provided.

Apply filter



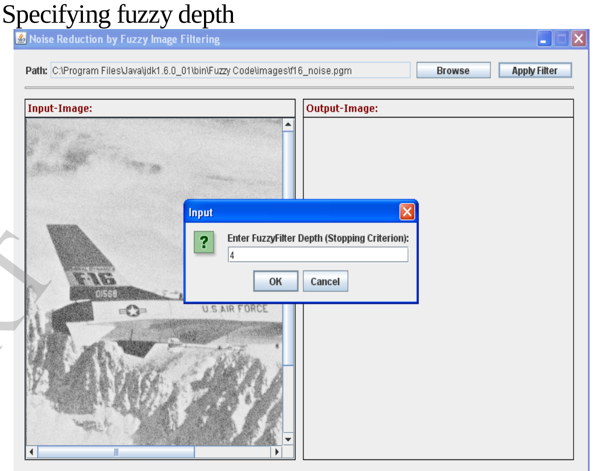
Click on Apply Filter button to select filter type. Here, the proposed FuzzyFilter is selected.

Saving the file



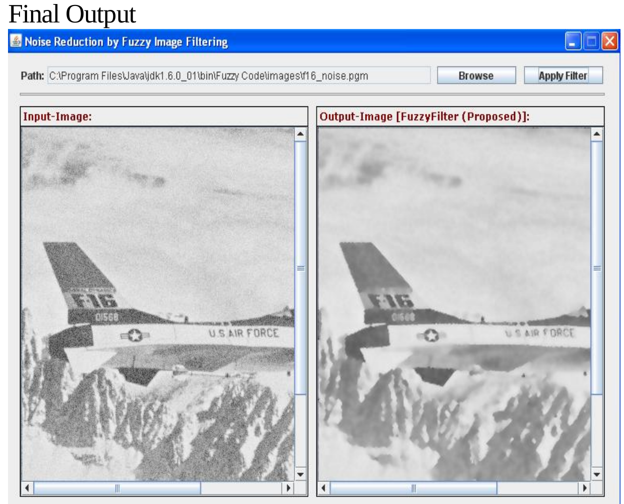
Now the input image is saved to get the desired output.

Specifying fuzzy depth



Here the depth value will be given to view the output image as required by the user. It can also specifies the stopping criterion.

Final Output



IJERT

The output for the proposed Fuzzy Filter is displayed in the space provided for the output.

VII. CONCLUSION

The thesis is proposed for a new fuzzy filter for Gaussian noise reduction. 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 its 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. A numerical measure, such as MSE, and visual observation show convincing results. Finally, the fuzzy filter scheme is sufficiently simple to enable fast hardware implementations. The proposed filter is applied to grayscale test images (8-bit, L=255), after adding Gaussian noise of different levels. Such a procedure allows us to compare and evaluate the filtered image against the original one. For low noise levels, one iteration is sufficient to efficiently remove the noise. Also, a low amplification factor gives the best results. Images with low noise levels and containing fine textures should be treated carefully.

The proposed filter gives a more natural image without the "patchy look".There are still many interesting research issues related to the extensions of fuzzy noise reduction. Genetic-neuro-fuzzy technique is very effective in speckle noise reduction as well as detail preserving even in the presence of highly noise corrupted data, and it works significantly better than other well-known conventional methods in the literature.

New alternative noise reduction method for color images that were corrupted with additive Gaussian noise

IJERT

REFERENCES

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

[2] D. Van De Ville, W. Philips, and I. Lemahieu, Fuzzy Techniques in Image Processing. New York: Springer-Verlag, 2000, vol. 52, Studies in Fuzziness and Soft Computing.

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

[4] Herbert Schildt, "The Complete Reference Java2", TATA McGraw-HILL Publication, Fourth Edition, 2001.

[5] 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.

[6] C.-S. Lee and Y.-H. Kuo, Fuzzy Techniques in Image Processing. New York: Springer-Verlag, 2000, vol. 52, Studies in Fuzziness and Soft Computing, ch. Adaptive fuzzy filter and its application to image enhancement, pp. 172–193.