

ADDITIVE NOISE REMOVAL FOR COLOR IMAGES USING FUZZY FILTERS

G.Sudhavani¹, G.Madhuri², P.Venkateswara Rao³, Dr.K.Satya Prasad⁴

¹ Assoc.Prof, Dept. of ECE, R.V.R& J.C College of Engineering, Guntur, A.P, India
gsudhavani@gmail.com

² Student (M.Tech), Dept. of ECE, R.V.R& J.C College of Engineering, Guntur, A.P,
India
madhuri471@gmail.com

³ Assoc.Prof, Dept. of CSE, R.V.R& J.C College of Engineering, Guntur, A.P, India
Paladugu.vrao@gmail.com

⁴ Professor, Dept. of ECE, JNT University, Kakinada, A.P, India
prasad_kodati@yahoo.co.in

ABSTRACT

Processing of images plays a vital role in any field such as medical and satellite applications. While capturing or transmission of images, effect of noise plays a predominant role. A filter is presented with fuzzy rules for additive noise removal from color images. The filter consists of two parts. The first part computes the distances among the color components of the center pixel and its neighboring pixels. The second part of the filter calculates the color components differences to retain the fine details of the image. One distance measure as the Minkowski's distance and other as the Absolute distance is selected and compared their performances using Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE). In this paper the performance of the fuzzy noise filter with two distance measures is compared by changing the noise ratio. The performance of these filters is compared with traditional filters.

KEY WORDS

Absolute distance, Additive noise, fuzzy filter, fuzzy rule-based systems, Minkowski distance.

1. INTRODUCTION

To carry and transfer information images are one of the most important tools. The research field of image processing not only comprehends technologies for capture and transfer images, but also techniques to anatomize these images. Image processing is a growing research field. In human life images have always played an important and essential role, not only used for communication, but also for other applications like scientific, commercial, military and industrial. In this paper we will focus on a technique that removes noise from color images using fuzzy logic.

Especially there are three types of noise exist: Gaussian noise, Salt & pepper noise, Speckle noise. Gaussian noise is statistical noise that has a probability density function of the normal distribution (also known as Gaussian distribution). It is most commonly known as additive white Gaussian noise (AWGN). Salt & pepper noise itself as randomly occurring white and black pixels. A noise reduction method which uses median filter is suitable for reduction of salt & pepper noise. This type of noise can be described as dark pixels on bright background and bright pixels on dark background. Images corrupt with Salt and pepper noise when quick transients, such as faulty

switching, take place. Speckle noise is also known as multiplicative noise, i.e. the gray levels of speckle noisy pixels are in direct proportion to the local gray levels in any area of an image.

Distance computation plays an important role in removal of noise from color images. The distance among two couples is calculated according to the Minkowski's and absolute distances. To calculate scaling factors of the filter Takagi-sugeno fuzzy model is used. In this method three 2-D distances (distance between red-green, red-blue, and green-blue) along with three fuzzy rules are used to compute the scaling factors. The concept behind evolution of fuzzy rules is to assign large scaling factors to the neighbors that have similar colors as the centre pixel.

Some of the filters designed earlier for noise reduction are, a new fuzzy filter for image enhancement [1], noise adaptive soft switching median filter [2], noise reduction by fuzzy image filtering [3] and so on. Many of these filters are designed for the reduction of fat-tailed noise. These filters are performing better than the basic existing filters such as mean and median filters. The current fuzzy techniques developed for noise reduction are not capable of producing potent results for removal of additive noise, which is illustrated in [4]. Other flaw of the current methods is that most of filters are especially developed for grayscale images. It is possible to extend these filters to color images by applying them individually on each color component. Processing of three components separately may introduce many artifacts, especially on edge or texture elements [5] while applying grayscale image filtering techniques on color components separately. Therefore, this paper presents a new and simple fuzzy technique for filtering color images corrupted with narrow-tailed and medium narrow-tailed noise (e.g., Gaussian noise) without introducing these artifacts.

A digital color image (denoted as X) can be represented by using a certain color space (e.g., RGB, HSV, $L^*a^*b^*$). The most commonly used color space is called "RGB," which has one overwhelmingly important characteristic i.e., every scanner and digital camera can produce images with RGB-encoded color. And also every image-handling device and color-aware application can handle images with RGB-encoded color. It is the official default color space for the World Wide Web. Colors in the RGB model are represented by a 3-D vector, with the first vector element being the red, the second being the green and the third being the blue, respectively. The red, green and blue components are called the three primary components, each component is quantized to the range $[0, 2^m - 1]$. In practice, a digital color image X can be represented by a 2-D array of vectors where an address (k, l) defines a position in X , called a pixel or picture element. If $X(k, l, 1)$ defines the red component, $X(k, l, 2)$ defines the green component and $X(k, l, 3)$ defines the blue component of a pixel at position (k, l) in an (noise-free) image X , then we can denote the noisy color image NI at position (k, l) as follows:

$$[NI(k, l, 1) \quad NI(k, l, 2) \quad NI(k, l, 3)] = [(X(k, l, 1) + \partial_1) \quad (X(k, l, 2) + \partial_2) \quad (X(k, l, 3) + \partial_3)] \quad (1)$$

With ∂_1, ∂_2 and ∂_3 three separate randomly Gaussian distributed values with means $(\mu_1, \mu_2$ and $\mu_3)$ and standard deviations $(\sigma_1, \sigma_2$ and $\sigma_3)$, respectively. NI is the color image that is corrupted with additive white Gaussian noise.

2. FIRST PART OF THE FILTER WITH FUZZY RULES

While capturing or transmission of images the effect of noise may degrade the quality of the image so it is desirable to perform some sort of noise reduction on an image. The general idea for noise reduction is to average a pixel using other pixel values from its neighborhood, but simultaneously take into account the important image structures such as edges and color component distances, those properties of the image should not be destroyed by the filter. The goal of the first part of the filter is to discriminate between local variations due to noise and due to image structures. This is accomplished by using the color component distances instead of component differences as done in most current filters. For example, to filter a certain red component at position (k,l) , we use the distances between red–green and red–blue of a certain neighborhood centered at (k,l) instead of just a neighborhood in the red component array. The difference between this new proposed filter and other vector based approaches as [6]–[7] is that we do not calculate the 3-D distances between pixels (distances between two pixels where a pixel is considered as a vector), but we use three 2-D distances (distances between red–green, red–blue, and green–blue) together with three fuzzy rules to calculate the weights used for the Takagi–Sugeno fuzzy model [8] explained in the next section.

2.1 Distance determination:

The red, green, and blue component at a certain pixel position of a noisy input image NI is denoted as $NI(k,l,1)$, $NI(k,l,2)$ and $NI(k,l,3)$, respectively. So, for each pixel position, we have three components that define the color. For each pixel position (k,l) we define the following pairs: the pair red and green denoted as $RG(k,l) = (NI(k,l,1), NI(k,l,2))$ the pair red and blue denoted as $RB(k,l) = (NI(k,l,1), NI(k,l,3))$ and the pair green and blue denoted as $GB(k,l) = (NI(k,l,2), NI(k,l,3))$. To filter the current image pixel, we use a window of size $(2I+1) \times (2I+1)$ (we considered I as 1 so we have a window size of 3×3) centered at position (k,l) . Next we assigned certain scaling factors to each of the pixels in the window. The scaling factors $w(k+i,l+j,1)$, $w(k+i,l+j,2)$ and $w(k+i,l+j,3)$ are for the red, green and blue components at position $(k+i,l+j)$ respectively. These scaling factors are assigned according to the following fuzzy rules, where a Takagi-sugeno fuzzy model [8] was used.

Fuzzy rule 1: The first fuzzy rule defines the weight $w(k+i,l+j,1)$ for the red component of the neighbor $NI(k+i,l+j,1)$, i.e.

IF the distances are small between the pairs $RG(k,l)$, $RG(k+i,l+j)$ and $RB(k,l)$, $RB(k+i,l+j)$ **THEN** the scaling factor $w(k+i,l+j,1)$ is large.

Fuzzy rule 2: The second fuzzy rule defines the weight $w(k+i,l+j,2)$ for the green component of the neighbor $NI(k+i,l+j,2)$, i.e.

IF the distances are small between the pairs $RG(k,l)$, $RG(k+i,l+j)$ and $GB(k,l)$, $GB(k+i,l+j)$ **THEN** the scaling factor $w(k+i,l+j,2)$ is large.

Fuzzy rule 3: The third fuzzy rule defines the weight $w(k+i,l+j,3)$ for the blue component of the neighbor $NI(k+i,l+j,3)$, i.e.

IF the distances are small between the pairs $RB(k,l)$, $RB(k+i,l+j)$ and $GB(k,l)$, $GB(k+i,l+j)$ **THEN** the scaling factor $w(k+i,l+j,3)$ is large.

The concept behind these simple fuzzy rules is to assign large scaling factors to the neighbors that have similar colors as the center pixel. In color images it is optimum to consider pixels as colors instead of taking them as three separate color components. When only the separate color components are considered, more artifacts are introduced, especially on the contour of objects. The distance between two couples is calculated according to the Minkowski's distances. This is illustrated in equation (1) for red-green pair

$$D(RG(k,l), RG(k+i,l+j)) = \left[(NI(k+i,l+j,1) - NI(k,l,1))^2 + (NI(k+i,l+j,2) - NI(k,l,2))^2 \right]^{1/2} \quad (2)$$

Another distance measure which is the Absolute distance is also illustrated in the following equation (3) for red-green pair.

$$D(RG(k,l), RG(k+i,l+j)) = |NI(k+i,l+j,1) - NI(k,l,1)| + |NI(k+i,l+j,2) - NI(k,l,2)| \quad (3)$$

To compute the value that expresses the degree to which the distance of two couples is small, we will make use of the fuzzy set *small*. Fuzzy sets are commonly represented by membership functions. From such functions we can derive the corresponding membership degrees. If the distance between two couples has a membership degree one (zero) in the fuzzy set *small*, it means that this distance is considered as (not) small for sure. Membership degrees between zero and one indicate that we do not know for sure if such distance is small or not, so there is some kind of uncertainty. For more background information about fuzzy logic, we refer to [9], and [10]. A membership function *small*, denoted as μ_{small} is shown in Fig. 1. An alternative notation for the membership function *small* is given in equation (4)

$$\mu_{small}(y) = \begin{cases} \left(\frac{t-y}{t} \right)^2, & \text{if } y \leq t \\ 0, & \text{if } y > t \end{cases} \quad (4)$$

With $t \in]0, \sqrt{2}]$ for the normalized input image NI

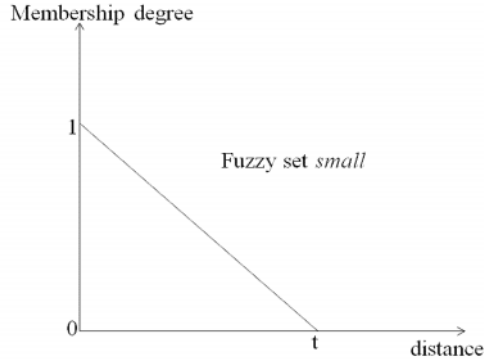


Fig1: Membership function small

We have define three such fuzzy sets, one for each pair (one for the pair red–green, another for the pair red–blue, and the last one for the pair green–blue). All these fuzzy sets depend on the parameter as seen in (4) and Fig. 1. These parameters are denoted as P_{RG} , P_{RB} and P_{GB} , are determined adaptively as follows:

$$\begin{aligned}
 P_{RG}(k,l) &= \max_{k,l \in \Omega} \left(\gamma_{RG}(k,l,i,j) \right) \\
 P_{RB}(k,l) &= \max_{k,l \in \Omega} \left(\gamma_{RB}(k,l,i,j) \right) \\
 P_{GB}(k,l) &= \max_{k,l \in \Omega} \left(\gamma_{GB}(k,l,i,j) \right)
 \end{aligned} \tag{5}$$

Where Ω defines the $(2I+1) \times (2I+1)$ neighborhood around the central pixel, i.e. $i, j \in \{-I, -I+1, \dots, 0, \dots, I-1, I\}$, and where $\gamma_{RG}(k,l,i,j)$, $\gamma_{RB}(k,l,i,j)$ and $\gamma_{GB}(k,l,i,j)$ were used as convenient notation for the following distances:

$$\begin{aligned}
 \gamma_{RG}(k,l,i,j) &= D(RG(k,l), RG(k+i,l+j)) \\
 \gamma_{RB}(k,l,i,j) &= D(RB(k,l), RB(k+i,l+j)) \\
 \gamma_{GB}(k,l,i,j) &= D(GB(k,l), GB(k+i,l+j))
 \end{aligned} \tag{6}$$

So, the parameters (for the spatial position (k,l)) $P_{RG}(k,l)$, $P_{RB}(k,l)$, and $P_{GB}(k,l)$ are equal to the maximal distance between the red–green, red–blue, and green–blue pairs in a $(2I+1) \times (2I+1)$ neighborhood around the centre pixel (k,l) . The γ 's of expression (6) are used to calculate the scaling factors as introduced by fuzzy rules. In these rules, we can observe an intersection of two fuzzy sets, which is generally specified by mapping T . We have used the algebraic product T -norms. This means for instance that the fuzzification of the antecedent of fuzzy rule 1 is $\mu_{small1}(\gamma_{RG}(k,l,i,j)) \sqcup \mu_{small2}(\gamma_{RB}(k,l,i,j))$, where μ_{small1} and μ_{small2} are

equal to membership function $small$, shown in expression (2), with parameters $P_{RG}(k,l)$ and $P_{RB}(k,l)$, respectively. The obtained value $\left(\mu_{small1}(\gamma_{RG}(k,l,i,j)) \sqcup \mu_{small2}(\gamma_{RB}(k,l,i,j))\right)$ is called the activation degree of the fuzzy rule and is used to obtain corresponding scaling factor, i.e. $w(k+i,l+j,1) = \mu_{small1}(\gamma_{RG}(k,l,i,j)) \sqcup \mu_{small2}(\gamma_{RB}(k,l,i,j))$.

2.2 Weights modification

The scaling factors $w(k,l,1)$, $w(k,l,2)$ and $w(k,l,3)$ are computed as follows:

$$\begin{aligned} w(k+i,l+j,1) &= \mu_{small1}(\gamma_{RG}(k,l,i,j)) \sqcup \mu_{small2}(\gamma_{RB}(k,l,i,j)) \\ w(k+i,l+j,2) &= \mu_{small1}(\gamma_{RG}(k,l,i,j)) \sqcup \mu_{small3}(\gamma_{GB}(k,l,i,j)) \\ w(k+i,l+j,3) &= \mu_{small2}(\gamma_{RB}(k,l,i,j)) \sqcup \mu_{small3}(\gamma_{GB}(k,l,i,j)) \end{aligned} \quad (7)$$

Where μ_{small1} , μ_{small2} and μ_{small3} are equal to membership function $small$, shown in expression (2), with parameters $P_{RG}(k,l)$, $P_{RB}(k,l)$ and $P_{GB}(k,l)$ respectively. The output of the first part of the fuzzy filter is illustrated in (8) for the red component,

$$A(k,l,1) = \frac{\sum_{i=-I}^{+I} \sum_{j=-I}^{+I} w(k+i,l+j,1) \sqcup NI(k+i,l+j,1)}{\sum_{i=-I}^{+I} \sum_{j=-I}^{+I} w(k+i,l+j,1)} \quad (8)$$

Where the output image is denoted as A . The filtering method for the green and blue component is similar to the one above.

3. SECOND PART OF THE FILTER TO RETAIN FINE DETAILS

The Second part of the fuzzy filter is a complementary filter to the first one. The objective of this filter is to improve the first method by reducing the noise in the color components differences without deteriorating the fine details of the image. The second part of the filter is realized by taking the local differences in the red, green, and blue environment separately. The local differences are used to calculate the local estimation of the center pixel by taking the average of those differences.

Analogous to the first part of the filter, we use a window of size $(2J+1) \times (2J+1)$ (we considered J as 1 so we have a window size of 3×3) where J is not necessarily equal to I , centered at (k,l) to filter the current image pixel at position (k,l) . Next, we calculate the local differences (also called gradients or derivatives) for each element of the window denoted as BD_R , BD_G and BD_B for the red, green, and blue environment, respectively. If the output image of the previous filter is denoted as A , then the differences are calculated as follows:

$$\begin{aligned}
 BD_R(i, j) &= A(k+i, l+j, 1) - A(k, l, 1) \\
 BD_G(i, j) &= A(k+i, l+j, 2) - A(k, l, 2) \\
 BD_B(i, j) &= A(k+i, l+j, 3) - A(k, l, 3)
 \end{aligned} \tag{9}$$

For all $i, j \in \{-J, \dots, 0, \dots, +J\}$. To calculate the following correction terms these differences are combined as in equation (10)

$$e(i, j) = \frac{1}{3} (BD_R(i, j) + BD_G(i, j) + BD_B(i, j)) \tag{10}$$

i.e. taking the average of the difference for the red, green, and blue component at the same position.

3.1. Output of the second part of the filter

Finally, the output of the second part of the filter, denoted as B , is determined as follows:

$$\begin{aligned}
 B(k, l, 1) &= \frac{\sum_{i=-J}^{+J} \sum_{j=-J}^{+J} (A(k+i, l+j, 1) + e(i, j))}{(2J+1)^2} \\
 B(k, l, 2) &= \frac{\sum_{i=-J}^{+J} \sum_{j=-J}^{+J} (A(k+i, l+j, 2) + e(i, j))}{(2J+1)^2} \\
 B(k, l, 3) &= \frac{\sum_{i=-J}^{+J} \sum_{j=-J}^{+J} (A(k+i, l+j, 3) + e(i, j))}{(2J+1)^2}
 \end{aligned} \tag{11}$$

With $B(k, l, 1)$, $B(k, l, 2)$ and $B(k, l, 3)$ the red, green, and blue component of the output image and where $e(i, j)$ is the correction term for the neighboring pixel $A(k+i, l+j, 1)$, $A(k+i, l+j, 2)$ and $A(k+i, l+j, 3)$.

4. SIMULATION RESULTS:

The effectiveness of the designed fuzzy filter is known by calculating peak signal-to-noise ratio ($PSNR$) in decibels (dB)

$$PSNR(FI, NI) = 10 \log_{10} \frac{S^2}{MSE(FI, NI)} \tag{12}$$

$$MSE(FI, NI) = \frac{\sum_{c=1}^3 \sum_{k=1}^N \sum_{l=1}^M [NI(k, l, c) - FI(k, l, c)]^2}{3NM} \quad (13)$$

The mathematical results for Minkowski's distance and for absolute distance in terms of *PSNR* values are shown in table I and table II. Table I shows the numerical results for colored Lena image of size (512x512) for different noise levels. Table II shows the numerical results for colored Baboon eye image of size (512x512) for different noise levels. Figure-2 to figure-4 shows the visual observations for the results tabulated in tables I and II.



Fig2: Original image

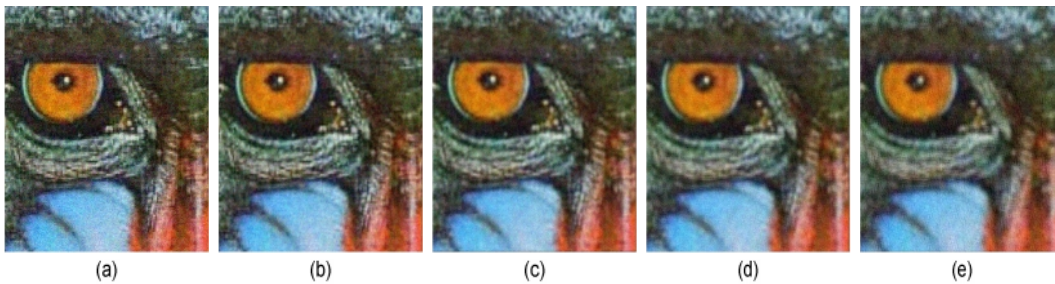


Fig3: Illustration of Minkowski's distance results for Baboon image with $\sigma = 30$ (a) Noise image (b) corresponding output image for (3x3, 3x3) window (c) corresponding output image for (3x3, 5x5) window (d) corresponding output image for (3x3, 7x7) window (e) corresponding output image for (3x3, 9x9) window.

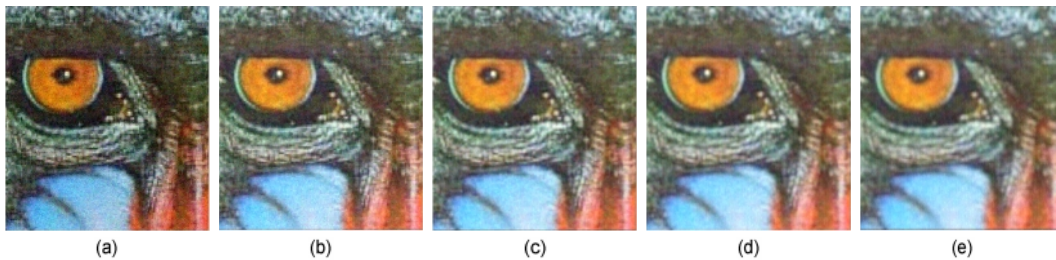


Fig4: Illustration of Absolute distance results for Baboon image with $\sigma = 30$ (a) Noise image (b) corresponding output image for (3x3, 3x3) window (c) corresponding output image for (3x3, 5x5) window (d) corresponding output image for (3x3, 7x7) window (e) corresponding output image for (3x3, 9x9) window.

Table-I Performance comparison with other filtering methods for the (512x512) colored Lena image

| | <i>PSNR values in dB</i> | | | | |
|--|--------------------------|---------|---------|---------|---------|
| | Lena image | | | | |
| σ | 5 | 10 | 20 | 30 | 40 |
| Noisy | 37.6898 | 32.6498 | 29.9416 | 29.1524 | 28.7689 |
| Mean | 33.8422 | 31.5334 | 29.4863 | 28.7357 | 28.3610 |
| Median | 35.1713 | 31.9470 | 29.8008 | 29.0938 | 28.7528 |
| Minkowski's distance | 32.7361 | 30.6698 | 29.0070 | 28.4260 | 28.1320 |
| Absolute distance (proposed method) | 38.5100 | 37.4666 | 35.1361 | 33.9245 | 33.2771 |

Table-II Performance comparison with other filtering methods for the (512x512) colored Baboon eye image

| | <i>PSNR values in dB</i> | | | | |
|--|--------------------------|---------|---------|---------|---------|
| | Baboon image | | | | |
| σ | 5 | 10 | 20 | 30 | 40 |
| Noisy | 37.1812 | 32.0382 | 29.4092 | 28.5992 | 28.2410 |
| Mean | 36.1862 | 32.0382 | 29.4092 | 28.5992 | 28.2410 |
| Median | 36.8722 | 32.3847 | 29.8070 | 29.0101 | 28.6407 |
| Minkowski's distance | 34.0968 | 30.8973 | 28.8649 | 28.2453 | 27.9835 |
| Absolute distance (proposed method) | 44.7091 | 41.0410 | 36.4986 | 34.7709 | 33.8436 |

5. CONCLUSION:

This paper proposed a fuzzy noise filter for removal of additive noise for any type of digital color images. At higher levels of noise the fuzzy noise filter is showing better results. Numerical measures and visual observations have shown convincing results. By observation from tables I & II it is clear that the fuzzy noise filter outperformed when compared with other methods.

REFERENCES:

- [1] Farzam Farbiz, Mohammad Bager Menhaj, Seyed A. Motamedi, and Martin T. Hagan "A New Fuzzy Logic Filter for Image Enhancement" IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 30, NO. 1, FEBRUARY 2000.
- [2] How-Lung Eng, Student Member, IEEE, and Kai-Kuang Ma "Noise Adaptive Soft-Switching Median Filter" IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 10, NO. 2, FEBRUARY 2001
- [3] Dimitri Van De Ville, Mike Nachtgeael, Dietrich Van der Weken, Etienne E. Kerre, Wilfried Philips and Ignace Lemahieu "Noise Reduction by Fuzzy Image Filtering" IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 11, NO. 4, AUGUST 2003.
- [4] S. Schulte, B. Huysmans, A. Pižurica, E. E. Kerre, and W. Philips, "A new fuzzy-based wavelet shrinkage image denoising technique," Lecture Notes Comput. Sci., vol. 4179, pp. 12–23, 20
- [5] S. Schulte, Valerie De Witte, and Etienne E. Kerre "A Fuzzy Noise Reduction Method for Color Images", IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 16, NO.5, MAY 2007.

- [6] C. Vertan and V. Buzuloiu, "Fuzzy nonlinear filtering of color images," in *Fuzzy Techniques in Image Processing*, E. E. Kerre and M. Nachtgaeel, Eds., 1st ed. Heidelberg, Germany: Physica Verlag, 2000, vol. 52, pp. 248–264.
- [7] S. M. Guo, C. S. Lee, and C. Y. Hsu, "An intelligent image agent based on soft-computing techniques for color image processing," *Expert Syst Appl.*, vol. 28, pp. 483–494, Apr. 2005
- [8] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.
- [9] *MATHEMATICS OF FUZZY SETS*, Logic, Topology, and Measure theory, edited by Ulrich Hohle, Stephen E. Rodabaugh.
- [10] ———, "Fuzzy logic and its application to approximate reasoning," *Inf. Process.*, vol. 74, pp. 591–594, 1973.