We have made a generalization of the proposed method and given guidelines for its extension and application to other matching strategies where the similarity constraint is used. It has also been generalized to other types of environments, such as outdoor scenes or aerial images, where edge segments are probably inappropriate features and therefore different features and attributes would be more suitable.

Acknowledgment

The authors would like to thank the reviewers and Associate Editor for constructive recommendations.

References

[1] S. H. Lee and J. J. Leou, "A dynamic programming approach to line segment matching in stereo vision," *Pattern Recognit.*, vol. 27, pp. 961–986, 1994.

[2] D. Marr and T. Poggio, "A computational theory of human stereovision," *Proc. R. Soc. London B*, vol. 207, pp. 301–328, 1979.

[3] G. Medioni and R. Nevatia, "Segment based stereo matching," *Comput. Vis., Graph., Image Process.*, vol. 31, pp. 2–18, 1985.

[4] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby, "PMF: A stereo correspondence algorithm using a disparity gradient limit," *Perception*, vol. 14, pp. 449–470, 1981.

[5] J. M. Cruz, G. Pajares, and J. Aranda, "A neural network approach to the stereovision correspondence problem by unsupervised learning," *Neural Networks*, vol. 8, no. 5, pp. 805–813, 1995.

[6] G. Pajares, J. M. Cruz, and J. Aranda, "Relaxation by Hopfield network in stereo image matching," *Pattern Recognit.*, vol. 31, no. 5, pp. 561–574, 1998.

[7] G. Pajares, J. M. Cruz, and J. A. López-Orozco, "Improving stereovision matching trough supervised learning," *Pattern Anal. Applicat.*, vol. 1, pp. 105–120, 1998.

[8] ——, "Stereo matching using Hebbian learning," *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, pp. 553–559, Aug. 1999.

[9] G. Q. Wei, W. Brauer, and G. Hirzinger, "Intensity- and gradient-based stereo matching using hierarchical Gaussian basis functions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 1143–1160, Nov. 1998.

[10] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ: Prentice-Hall, 1998.

[11] A. Huertas and G. Medioni, "Detection of intensity changes with subpixel accuracy using Laplacian–Gaussian masks," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 5, pp. 651–664, 1986.

[12] J. G. Leu and H. L. Yau, "Detecting the dislocations in metal crystals from microscopic images," *Pattern Recognit.*, vol. 24, no. 1, pp. 41–56, 1991.

[13] M. S. Lew, T. S. Huang, and K. Wong, "Learning and feature selection in stereo matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 869–881, Sept. 1994.

[14] E. P. Krotkov, *Active Computer Vision by Cooperative Focus and Stereo*. Berlin, Germany: Springer-Verlag, 1989.

[15] S. Tanaka and A. C. Kak, "A rule-based approach to binocular stereopsis," in *Analysis and Interpretation of Range Images*, R. C. Jain and A. K. Jain, Eds. Berlin, Germany: Springer-Verlag, 1990.

[16] R. Nevatia and K. R. Babu, "Linear feature extraction and description," *Comput. Vis., Graph., Image Process.*, vol. 13, pp. 257–269, 1980.

[17] R. L. Coultrip and R. H. Granger, "Sparse random networks with LTP learning rules approximate Bayes classifiers via Parzen's method," *Neural Networks*, vol. 7, no. 3, pp. 463–476, 1994.

[18] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[19] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, pp. 1065–1076, 1962.

[20] A. K. Jain and M. D. Ramaswami, "Classifier design with Parzen windows," in *Pattern Recognition and Artificial Intelligence Toward an Integration*, E. S. Gelsema and L. N. Kanal, Eds. Amsterdam, The Netherlands: Elsevier, 1988, pp. 211–218.

[21] K. Fukunaga and D. M. Hummels, "Bayes error estimation using Parzen and K–NN procedures," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 5, pp. 634–643, 1987.

[22] L. F. Escudero, *Reconocimiento de Patrones*. Madrid, Spain: Paraninfo, 1977.

[23] R. P. W. Duin, "On the choice of smoothing parameters for Parzen estimators of probability density functions," *IEEE Trans. Comput.*, vol. C-25, pp. 1175–1179, 1976.

[24] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 226–239, Mar. 1998.

[25] Z. Zhang and R. S. Blum, "A categorization of multiscale-decomposition-based image fusion schemes with a performance study for a digital camera application," *Proc. IEEE*, vol. 87, pp. 1315–1325, Aug. 1999.

[26] Y. S. Kim, J. J. Lee, and Y. H. Ha, "Stereo matching algorithm based on modified wavelet decomposition process," *Pattern Recognit.*, vol. 30, no. 6, pp. 929–952, 1997.

[27] C. Baillard and O. Dissard, "A stereo matching algorithm for urban digital elevation models," *Photogramm. Eng. Remote Sens.*, vol. 66, no. 9, pp. 1119–1128, 2000.

Histogram-Based Fuzzy Filter for Image Restoration

Jung-Hua Wang, Wen-Jeng Liu, and Lian-Da Lin

***Abstract*—In this paper, we present a novel approach to the restoration of noise-corrupted image, which is particularly effective at removing highly impulsive noise while preserving image details. This is accomplished through a fuzzy smoothing filter constructed from a set of fuzzy membership functions for which the initial parameters are derived in accordance with input histogram. A principle of *conservation in histogram potential* is incorporated with input statistics to adjust the initial parameters so as to minimize the discrepancy between a reference intensity and the output of defuzzification process. Similar to median filters (MF), the proposed filter has the benefits that it is simple and it assumes no *a priori* knowledge of specific input image, yet it shows superior performance over conventional filters (including MF) for the full range of impulsive noise probability. Unlike in many neuro-fuzzy or fuzzy-neuro filters where random strategy is employed to choose initial membership functions for subsequent lengthy training, the proposed filter can achieve satisfactory performance without any training.**

***Index Terms*—Fuzzy filter, histogram, image restoration, impulsive noise, median filter.**

I. Introduction

Median filtering (MF) is a nonlinear technique that is well known for its effectiveness in removing impulsive noise The 1-D MF is realized by passing a window over the input data, and taking the median value of the data inside the window as the output associated with the center of the window. In image processing applications two-dimensional median filters have been used with some success, and various methods [1]–[3] have been proposed to extend the median operation to two dimensions. The simplest way [3] is to pass a 2-D window, such as a square mask, over the 2-D input image. As with the 1-D MF, the pixels inside the window are ranked according to their gray intensity values and the median value taken as the output. By associating the nonlinear operation of the median filter with a linear cost function, Qiu [4] has shown that me-

Manuscript received April 5, 1998; revised October 29, 1999 and November 7, 2001 This work was supported by the National Science Council of Taiwan, R.O.C., under Grants NSC 86-2611-E-019-024 and NSC 89-2611-E-019-054. This paper was recommended by Associate Editors M. Kam and K. Pattipati.

J.-H. Wang and L.-D. Lin are with the Department of Electrical Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C. (e-mail: jhwang@celab1.ee.ntou.edu.tw).

W.-J. Liu is with the Department of Electronics Engineering, Nan Kai College, Nantou, Taiwan, R.O.C..

Publisher Item Identifier S 1083-4419(02)00705-7.

dian filtering is an optimization process that simultaneously measures the neighboring smoothness and the output error, thereby explaining why median-related filters have the essential properties of smoothing without extensive blurring of the signal.

Although noise suppression is obtainable by using 2-D MF, too much signal distortion is introduced and image details such as sharp corners and thin lines are lost. To overcome these problems, several variations of median filters have been developed, e.g., the multistage median filter [2], max/median filter [5], improved recursive median filtering scheme [6], minimum-maximum exclusive mean (MMEM) filter [28], and detection-estimation based filter (DEBF) [29]. Still, performance advantages by these conventional approaches can only be achieved when the occurrence probability of noise γ is small. Due to lack of adaptability, these variations of median filters cannot perform well when $\gamma \geq 0.2$.

Recently, adaptive systems based on neural networks [7], [19] or fuzzy theory [8], [9] with data-driven adjustable parameters have emerged as attractive alternatives. In this category, methods such as the optimal detail-restoring stack filters [10], the hybrid FIR weighted order statistics (WOS) filter [11], and adaptive average iterative filters [12] have been developed. Neural networks exploit their frameworks with abundant theorems and efficient numerical training algorithms. They embed several input-output mappings on a black-box web of connection weights. During the training process, a neural network is shown a sequence of randomly selected input samples, and values of connection weights are modified in accordance with the training algorithm. On the other hand, fuzzy systems can directly encode structured knowledge. Fuzzy systems may invariably store banks of common-sense rules linguistically articulated by an expert, or a fuzzy system may adaptively infer and modify its fuzzy rules from representative symbols (e.g., DARK, BRIGHT) as well as numerical samples. In the latter case, fuzzy systems and neural networks naturally combine. The combination produces an adaptive system in that the neural networks embed in an overall fuzzy architecture, generating and refining fuzzy rules from training data. Currently, hybrid neuro-fuzzy networks do not represent general means in restoring images corrupted by impulsive noise. As a first attempt, Yu and Lee used the adaptive fuzzy median filter (AFMF) [13] with the backpropagation algorithm [14] to tune a set of *randomly* given initial membership functions. Their simulation results have shown that AFMF successfully achieves better filtering than traditional MF when noise rate $\gamma > 0.3$.

Both in AFMF and our previous works [15], it has been shown that a random strategy of choosing initial membership functions often give rise to lengthy training time. However, to estimate a set of *well-conditioned* initial memberships is by no means a trivial task. Even worse, AFMF needs to refer to the source (uncorrupted) image while tuning the membership functions. The need of referring to the source image requires lengthy retraining time whenever a new input image is presented. Therefore, in order to make the fuzzy-neuro [16] or neuro-fuzzy [17], [18], [27] approach feasible in implementations, this “retraining problem” must be removed or at least alleviated. In this paper, we propose a histogram-based approach, which is capable of estimating a set of *well-conditioned* membership functions for constructing the histogram-based fuzzy filter (HFF). These parameters are termed *well-conditioned* because one can use HFF to achieve satisfactory image restoration without any training.

The rest of the paper is organized as follows. In Section II, we introduce the fuzzy inference rules related to the task of median filtering and the system architecture of HFF. In Section III, we first show how a set of initial parameters of fuzzy membership functions is derived from an input histogram. Then, we present a principle of conservation in histogram potential, which is useful for adjusting the initial parameter set to a well-conditioned one. In Section IV, a number of experiments are conducted to characterize HFF as well as to compare it with other fuzzy filters. In particular, we verify that HFF can eliminate the retraining problem encountered in other neuro-fuzzy networks. We conclude in Section V with a summary of results and a brief discussion.

II. System Architecture of HFF

In this section, we first discuss the preprocessing of the 2-D input image. For the problem of interest here, we assume an input image sized 256 × 256 with pixel gray level between 0 and 255. Since a *noise-corrupted* image inherently contains a high level of ambiguity, we can consider it as an array of fuzzy variables. For convenience of explanation, HFF is designed to create three fuzzy membership functions for three fuzzy sets, namely, Dk(Dark), Md(Medium), and Br(Bright). Therefore, each input pixel intensity $p(k,\, \ell)$ is viewed as a fuzzy variable with membership degree of three fuzzy sets Dk, Md, and Br, respectively. Note that, in practice, the number of membership functions certainly is not necessarily limited to just three. Since fuzzy systems can directly encode structured knowledge in a numerical framework, the intensity of each input pixel at $(k,\, \ell)$ is *normalized* to $0 \leq p(k,\, \ell) \leq 1$ for easy processing. A window of size 3 × 3 is used to scan across the entire image, where the filter output associated with the center of the window is denoted as Y. Thus, the elements of the window $W(k,\, \ell)$ centered at $(k,\, \ell)$ are as follows: $x_1 = p(k-1, \ell-1)$, $x_2 = p(k-1,\, \ell)$, $x_3 = p(k-1,\, \ell+1)$, $x_4 = p(k,\, \ell-1)$, $x_5 = p(k,\, \ell)$, $x_6 = p(k,\, \ell{+}1)$, $x_7 = p(k{+}1,\, \ell{-}1)$, $x_8 = p(k{+}1,\, \ell)$, $x_9 = p(k{+}1,\, \ell{+}1)$. Each element is considered a fuzzy variable, and the membership functions identify grade of brightness for each input pixel. Equation (1) gives the bell-shaped membership functions used in HFF:

$$m_j(x_i) = \frac{1}{1 + \left(\dfrac{x_i - c_j}{a_j}\right)^{2b_j}}, \qquad i = 1,\, 2, \ldots 9; \qquad j = 1,\, 2,\, 3 \tag{1}$$

where a_j, b_j, c_j are adjustable parameters. The fuzzy inference rules used in HFF basically follows the Takagi–Sugeno approach [23], with a slight difference in the output decision step:

$$\begin{cases} \text{if } \{\hat{p}(k,\, \ell) \text{ is closest to } Dk\}, & \text{then } \{Y \text{ is } Dk\} \text{ else} \\ \text{if } \{\hat{p}(k,\, \ell) \text{ is closest to } Md\}, & \text{then } \{Y \text{ is } Md\} \text{ else} \\ & \{Y \text{ is } Br\} \end{cases} \tag{2}$$

where $\hat{p}(k,\, \ell)$ is defined as the reference intensity for pixel at $(k,\, \ell)$:

$$\hat{p} = \frac{\displaystyle\sum_{i=1}^{9} x_i' I_i}{\displaystyle\sum_{i=1}^{9} I_i},$$

$$\text{where} \begin{cases} x_i' = 0 \text{ and } I_i = 0, & \text{if } x_i \in N_{imp} \\ x_i' = x_i \text{ and } I_i = 1, & \text{otherwise.} \end{cases} \tag{3}$$

The set N_{imp} contains pixels that are most likely corrupted by impulses. To determine N_{imp}, we first consider a 3 × 3 window $W[(k,\, \ell)]$ that scans the image from left to right and from top to bottom; in each scan nine pixels are ranked according to gray intensity. We then define $p(k,\, \ell) \in N_{imp}$, If $p(k,\, \ell) = \mathrm{Min}\{W[(k,\, \ell)]\}$ or $\mathrm{Max}\{W[(k,\, \ell)]\}$ or $\{p(k,\, \ell) \leq T\}$ or $\{p(k,\, \ell) \geq (1.0 - T)\}$, where T is a threshold. In our previous work [26], it has been shown that this pre-classification can greatly enhance the performance of conventional median filter. Because $\hat{p}$ is a data-driven product (i.e., the *mean* of the input pixels $\notin N_{imp}$), the use of $\hat{p}$ in combination with the fuzzy

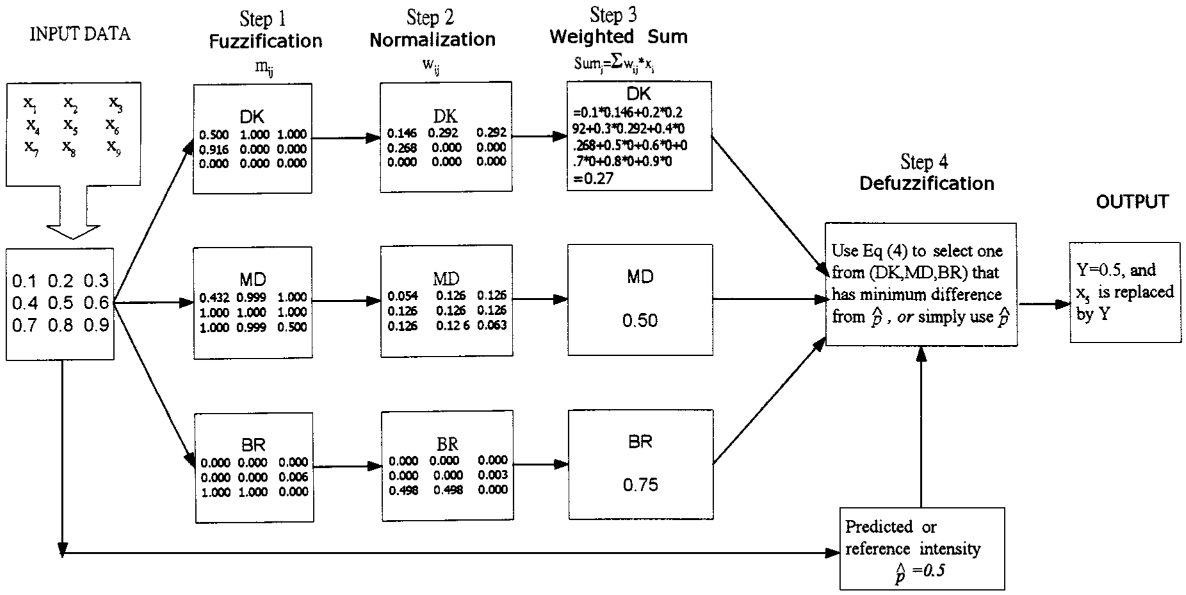


Fig. 1. Block diagram of HFF.

membership functions, which by themselves are also derived from input histograms, have made HFF a totally data-driven filter.

To explain how $\hat{p}(k, \ell)$ is used in HFF to determine the final output, in Fig. 1 we use the input subimage $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ as an illustrative example. Assume we are given a set of well-conditioned parameters for (1) as $(a_1, c_1) = (0.156, 0.256)$, $(a_2, c_2) = (0.318, 0.501), (a3, c3) = (0.118, 0.740)$, and $b_1 = b_2 = b_3 = 15$. Given the nine input pixels x_i of a subimage, HFF performs the fuzzification, i.e., the membership degree $m_{ij} = m_j(x_i)$ is calculated. Note that the slope (specified by b_j) is made much larger than a_j and c_j so that impulsive noise is apt to be filtered out by the membership functions. Step 2 normalizes the membership degree of each input pixel by using the following formula:

$$w_{ij} = \frac{m_j(x_i)}{\sum_{i=1}^{9} m_j(x_i)}, \qquad i = 1, 2 \cdots 9, \qquad j = 1, 2, 3.$$

Then, in Step 3 the weighted sum of input $Sumj = \sum_{i=1}^{9} w_{ij} x_i$ are calculated, which is followed by Step 4 where the final output of HFF is determined using the following rule:

$$\begin{aligned} &\text{If } x_5 \in N_{imp} \\ &\qquad \text{output } Y = Sumj \text{ that minimizes } E_j = \left|\hat{p} - Sum_j\right|, \\ &\qquad j = 1, 2, 3. \\ &\text{else} \\ &\qquad \text{output } Y = \hat{p}(k, \ell) \\ &\text{end.} \end{aligned} \tag{4}$$

In other words, (4) is incorporated with (3) to implement the output rules prescribed in (2).

According to the definition of $\hat{p}(k, \ell)$, HFF in essence takes the strategy of detection-estimation in performing filtering, just as MMEM [28] and DEBF [29] using statistical detection techniques and the median filter to remove impulses. However, one should note that the final filter output of HFF is not determined by conventional median operation, nor is it merely the result of the logical union of membership functions as is commonly adopted in other defuzzification methods such as Max defuzzification [9]. To see these features more clearly, we use the illustrative example in Fig. 1. If $x_5 \in N_{imp}$, the three intermediate weighted results Sum_j are compared with the reference intensity $\hat{p}(k, \ell)$, from which the one closest to $\hat{p}(k, \ell)$ is chosen to replace the pixel intensity at (k, ℓ). If the pixel at $(k, \ell) \notin N_{imp}$, then $\hat{p}(k, \ell)$ is used as the final output in order to better preserve the details. Apparently, with this scheme, performance of HFF hinges on the weights w_{ij} calculated in Step 2, which in turn are determined by the fuzzy membership functions in Step 1. Another factor that greatly affects the per-

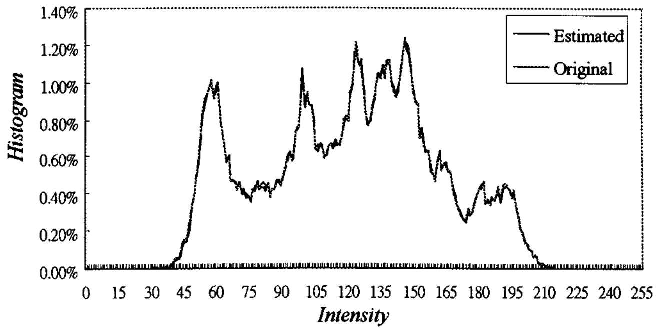


Fig. 2. Comparison of original and estimated histograms of Lena, $NMSE_i = 1.96E\text{-}008$.

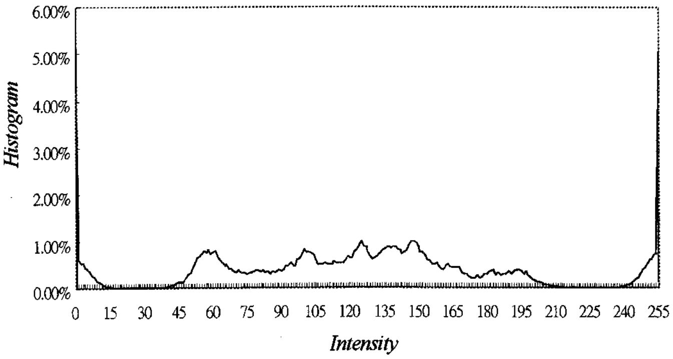


Fig. 3. Histogram of a corrupted Lena image with $\gamma = 0.2$.

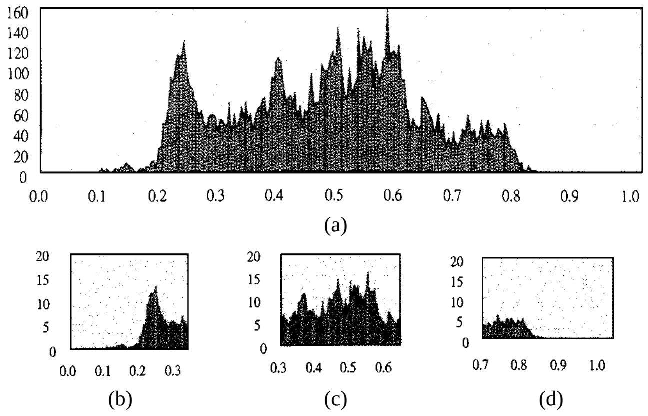


Fig. 4. (a) Estimated histogram, (b) DK segment (c) MD segment, (d) BR segment.

TABLE I
ALGORITHM OF ADJUSTING PARAMETER FOR HMF

```
Step 1. Given the initial parameters $C_\Omega$ and $Mass_\Omega$, $\Omega \in \{DK, MD, BR\}$.

Step 2. To limit the membership degree of input intensity $\{x_i \le T\}$ to be less than 0.5,
    define conservation in histogram potential for the left margin $\{x_i \le T\}$ as
    $(C_{DK} - T) \times Mass_{DK} = (C^*_{DK} - T) \times Mass^*_{DK} = \Lambda_L$,
    where $C^*_{DK}$ and $Mass^*_{DK}$ denote the updated values for $C_{DK}$ and $Mass_{DK}$.
      if $Mass_{DK} \ge (C_{DK} - T)$
          $Mass^*_{DK} = (C^*_{DK} - T) = \sqrt{\Lambda_L}$ ; $C^*_{DK} = \sqrt{\Lambda_L} + T$
      else
          $Mass^*_{DK} = Mass_{DK}$; $C^*_{DK} = C_{DK}$
      end

    Likewise, to limit the membership degree of input intensity $\{x_i \ge (1.0 - T)\}$ to be
    less than 0.5, conservation in histogram potential for the right margin
    $\{x_i \ge (1.0 - T)\}$ is defined as
    $Mass_{BR} \times \{(1.0 - T) - C_{BR}\} = \{(1.0 - T) - C^*_{BR}\} \times Mass^*_{BR} = \Lambda_R$

      if $Mass_{BR} \ge (1.0 - T) - C_{BR}$
          $Mass^*_{BR} = \{(1.0 - T) - C^*_{BR}\} = \sqrt{\Lambda_R}$ ; $C^*_{BR} = (1.0 - T) - \sqrt{\Lambda_R}$
      else
          $Mass^*_{BR} = Mass_{BR}$ ; $C^*_{BR} = C_{BR}$
      end
Step 3. For the membership functions Md, use the following rules.
    Let $T^* = T$ or $(1.0 - T)$,
      if $Mass_{MD} \ge |C_{MD} - T^*|$
          $Mass^*_{MD} = |C_{MD} - T^*|$; $C^*_{MD} = C_{MD}$
      else
          $Mass^*_{MD} = Mass_{MD}$; $C^*_{MD} = C_{MD}$
      end
Step 4. $a_1 = Mass^*_{DK}$, $c_1 = C^*_{DK}$ ; $a_2 = Mass^*_{MD}$ , $c_2 = C^*_{MD}$ ; $a_3 = Mass^*_{BR}$, $c_3 = C^*_{BR}$
```

formance is the way in which $\hat{p}(k,\ \ell)$ is produced. We note that, under extreme circumstances (e.g., when the occurrence probability of noise $\gamma = 1.0$) wherein all of the nine pixels within the $W[(k,\ \ell)]$ would be detected as $\in N_{imp}$. In that case, both nominator and denominator in (3) will be all zeros. To avoid this awkwardness, one can simply use $\hat{p} = (\sum_{i=1}^{9} x_i)/9$ when all of the nine pixels within the $W[(k,\ \ell)]$ are detected as $\in N_{imp}$.

TABLE II
VALUES OF PARAMETERS AFTER ADJUSTMENT

	Initial values			Adjusted		
	$a = Mass_\Omega$	b	$c = C_\Omega$	$a = Mass^*_\Omega$	b	$c = C^*_\Omega$
Dk	0.216	15.000	0.251	0.180	15.000	0.280
Md	0.667	15.000	0.504	0.397	15.000	0.504
Br	0.117	15.000	0.734	0.139	15.000	0.761

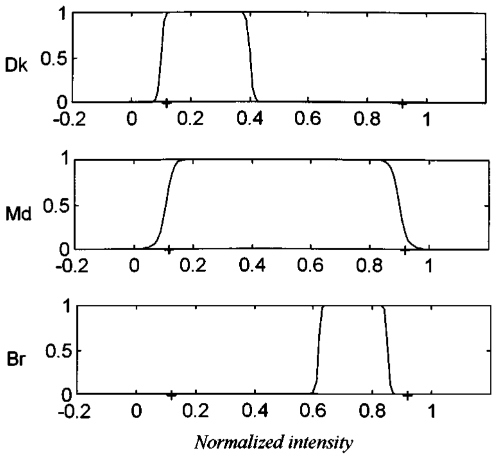


Fig. 5. Membership function of DK, MD, BR. $T = 0.1$ ("+" on the left).

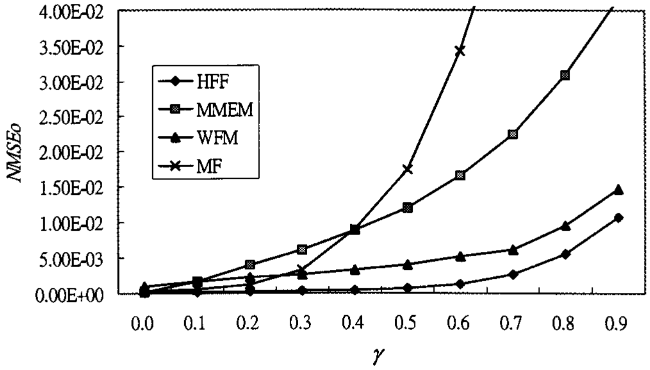


Fig. 6. Comparisons in terms of $NMSE_o$ using Lena image.

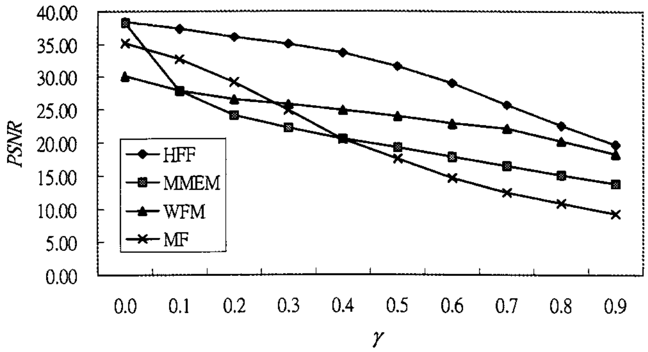
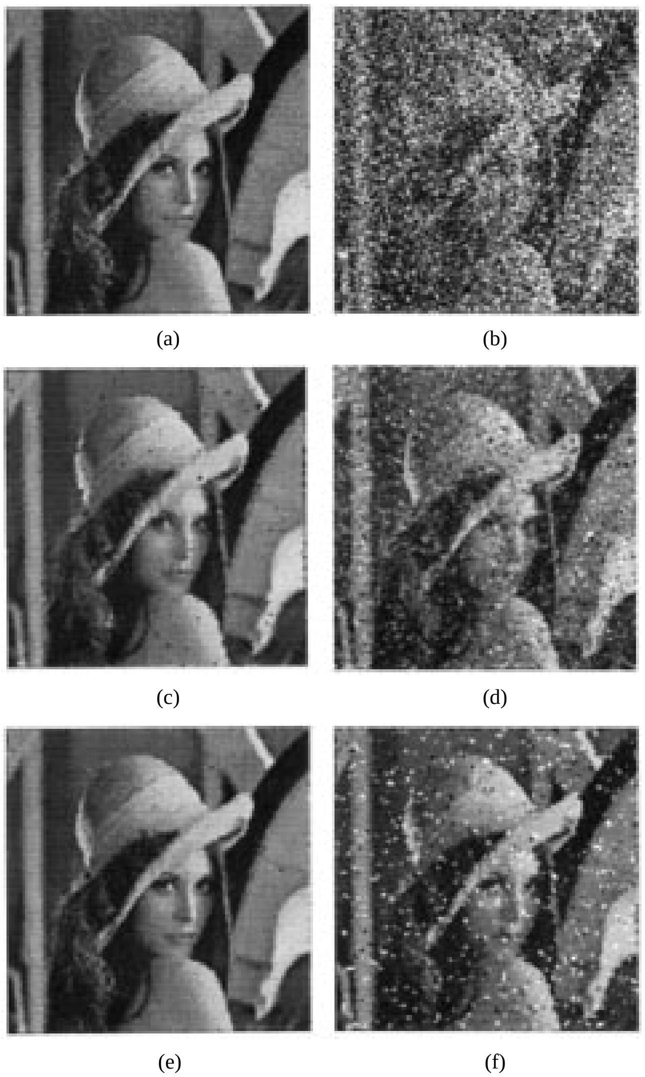


Fig. 7. Comparisons in terms of $PSNR$ using Lena image.

III. HISTOGRAM-BASED MEMBERSHIP FUNCTIONS

A. Estimating the Histogram of the Source Image

The proposed approach for obtaining histogram-based membership functions (abbreviated as HMF) starts with using the histogram statis-



(a) (b)

(c) (d)

(e) (f)

Fig. 8. (a) Uncorrupted Lena, (b) corrupted with $\gamma = 0.5$ (c) restored by WFM, (d) restored by MMEM, (e) restored by HFF, (f) restored by MF.

TABLE III
COMPARISON OF HFF AND MF USING WEATHER_1

	NMSE$_i$	*NMSE$_o$*		*PSNR*	
γ	Input	MF	HFF	MF	HFF
0.0	0.00E-00	1.41E-03	5.88E-04	28.49	32.30
0.1	2.84E-02	1.76E-03	7.71E-04	27.55	31.13
0.2	5.64E-02	2.77E-03	1.11E-03	25.57	29.53
0.3	8.20E-02	5.38E-03	1.60E-03	22.69	27.97
0.5	1.32E-01	2.34E-02	3.18E-03	16.30	24.98
0.7	1.76E-01	6.08E-02	7.66E-03	11.67	21.16
0.9	2.17E-01	1.43E-01	2.22E-02	8.45	16.53

TABLE IV
$IPSNR$ OF HFF AND MF USING WEATHER_1

γ	*PSNR* (Corrupted Input)	*IPSNR* by MF	*IPSNR* by HFF
0.1	15.47	12.08	15.66
0.2	12.49	13.08	17.04
0.3	10.86	11.83	17.11
0.5	8.79	7.51	16.19
0.7	7.54	4.13	13.62
0.9	6.64	1.82	9.89

tics of the corrupted image to estimate the histogram of the original (uncorrupted) image. Note that this is different from the approach employed by the weighted fuzzy mean filter (WFM) [22] in which the LR-type membership functions [9] are derived by referring to the histogram of the source image. However, because both HFF and WFM utilize histograms in deriving membership functions, it would be interesting to contrast their performance (see Section IV). To proceed, we first define the intensity histogram $H[g]$ for the noisy input image with $G = 256$ gray levels. That is, $H[g]$ denotes the number of pixels that has gray level g, $g = 0 \cdots 255$. Correspondingly, H_{imp} is defined as the intensity histogram of for pixels $\in N_{imp}$. Furthermore, we let $H^e[g]$ denote the *estimated* histogram of the original (uncorrupted) image. In this work, $H^e[g]$ is estimated by using the following formula:

$$H^e[g] = \frac{H[g] - H_{imp}[g]}{\sum_{g=0}^{255} (H[g] - H_{imp}[g])}. \tag{5}$$

To see the effectiveness of (5), we used a 256 × 256 noisy Lena image with $\gamma = 0.2$. Fig. 2 depicts $H[g]$ and $H^e[g]$ in percentage.

Note that throughout this paper the source image is corrupted by additive noise having long-tailed [21] input distributions (i.e., impulses). In simulations, the impulses n_i have a mean $M = bm(|b| \gg 1)$, where m = mean intensity of pixels inside the running window W. Given the occurrence probability of impulsive noise γ, the impulses take on positive and negative values with an equal probability. Thus, an arbitrary corrupted input pixel has intensity of $x_i = \mathrm{Max}(0, \mathrm{Min}(1, s_i + n_i))$, where s_i is the signal and

$$n_i = \begin{cases} |b|m & with\ probability \quad \gamma/2 \\ -|b|m & with\ probability \quad \gamma/2. \end{cases} \tag{6}$$

Our simulations will be based on the local mean image statistics as prescribed in (6). This will make only a negligible difference in the case of true impulsive noise [21]. Fig. 3 shows histogram $H[g]$ of a corrupted Lena image ($\gamma = 0.2$). By comparing Figs. 2 and 3, one can clearly see that many pixels have been severely corrupted by the long-tailed impulse noise.

B. Configuring Membership Functions Using Histogram Statistics

Given the estimated histogram $H^e[g]$, we now explain how to derive the parameters (a_1, c_1), (a_2, c_2) and (a_3, c_3) for membership functions Dk, Md and Br, respectively. The first step is to obtain a set of initial parameters. We will use the estimated histogram shown in Fig. 4(a) to illustrate how the *initial* parameters of HMF are calculated. The histogram is first divided into three segments equal in length, namely DK, MD, and BR as shown in Fig. 4(b)–(d). We define three statistics, pdf (potential density function), $Mass$, and C, which are useful for describing the intensity features of the divided histogram segment, namely

$$pdf_{DK}(g) = \frac{H^e[g]}{\sum_{k \in DK} H^e[k]}, \qquad pdf_{MD}(g) = \frac{H^e[g]}{\sum_{k \in MD} H^e[k]}$$

$$pdf_{BR}(g) = \frac{H^e[g]}{\sum_{k \in BR} H^e[k]} \qquad Mass_{DK} = \frac{\sum_{k \in DK} H^e[k]}{\sum_{k \in TOTAL} H^e[k]}$$

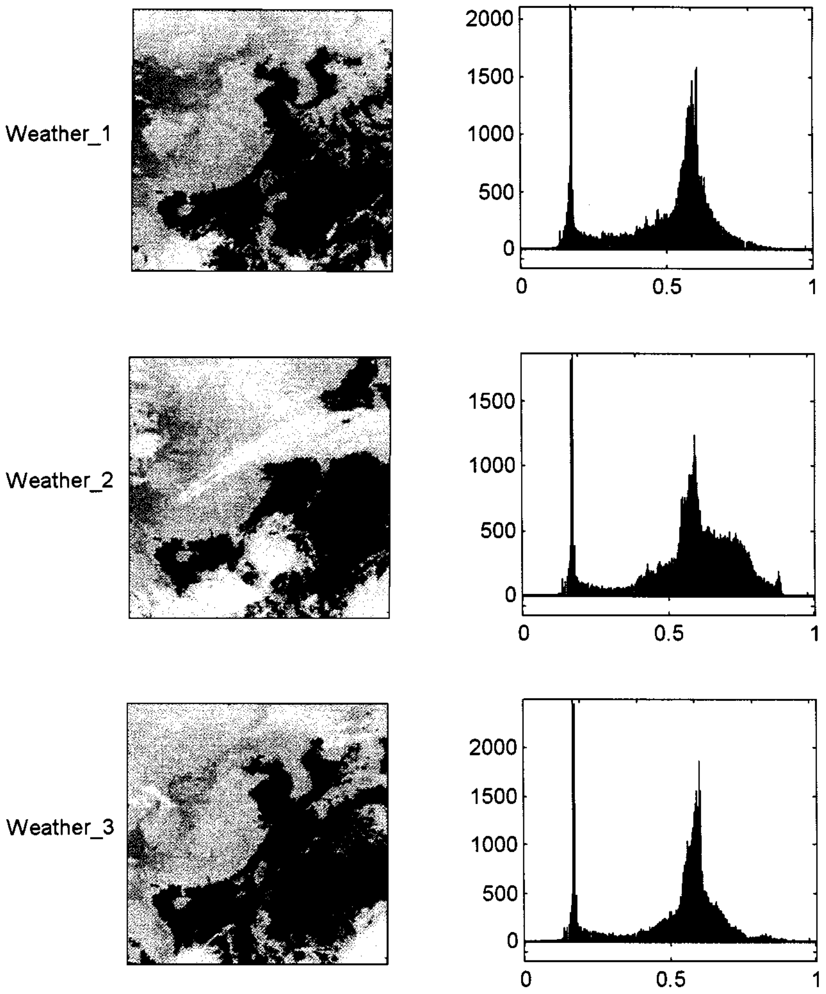


Fig. 9. Uncorrupted pictures (and their histograms) of Weather-1, Weather_2, and Weather_3.

$$Mass_{MD} = \frac{\sum\limits_{k \in MD} H^e[k]}{\sum\limits_{k \in TOTAL} H^e[k]}, \qquad Mass_{BR} = \frac{\sum\limits_{k \in BR} H^e[k]}{\sum\limits_{k \in TOTAL} H^e[k]}.$$

Denote C_{DK}, C_{MD}, and C_{BR} as the *Centroid of* DK, MD and BR, respectively. For illustration purpose, we also denote $k \in DK$ as an arbitrary gray level in the DK segment. Assume an input image with G gray levels, then C_{DK} is calculated as $\sum_{k \in DK} (k/G-1) \times pdf_{DK}(k)$, where $k/(G-1)$ in essence represents a weighting factor for the gray level k. The values of $Mass$ and C are used as initial values for the parameter a_i and c_i, respectively. Clearly, $0 < a_i, c_i < 1$. In particular, $Mass$ corresponds to one half the *support* length for a fuzzy set [9]. It would be expected that the *support* length of Md be longer than that of Br and Dk, because normally more pixels are located at MD for most images encountered.

The algorithm for adjusting parameters for HMF is given in Table I, where we can see, after initialization, a *principle of conservation in histogram potential* is applied to perform one-time update on the initial parameters. The rationale of adjusting initial parameters is twofold. Firstly, limiting the membership degree to be less than ≤0.5 for input intensity $x_i \leq T$ or $x_i \geq (1.0-T)$ can create a desirable effect of preventing heavily corrupted pixels $p(k, \ell) \in N_{imp}$ from interfering the fuzzification process (i.e., Step 1 in Fig. 1). Secondly, for a reasonable membership function such as the one in (1), if the membership degree of input $x_i \geq a_i + c_i$ or $x_i \leq c_i - a_i$ is limited to be less than 0.5, then $T \leq (c_i - a_i) \leq (a_i + c_i) \leq 2a_i \leq (1.0 - T)$, which implies that any variation in c_i calls for a counteractive change in a_i, and vice versa. Therefore, one can be ensured that initial parameters will be adjusted, under the conservation principle, to proper values such that impulsive noise can be eliminated effectively. This method of configuring membership functions is certainly different from that employed in WFM [22] where no specific histogram statistics were derived for configuring the L–R type membership functions. For example, in [22] a pixel with maximum value of source (uncorrupted image) histogram in the DK segment is chosen as the peak of the corresponding L–R function. Using a Lena image (corrupted with $\gamma = 0.2$) as the test input, Table II lists the initial and adjusted values of parameters (a_i, b_i, c_i) of the three membership functions. Fig. 5 depicts the membership functions that correspond to the adjusted parameters in Table II. Note that the support length of MD indeed is longer than those of DK and BR.

IV. Simulation Results

In this section, three experiments are conducted to characterize the performance of HFF in terms of noise suppression and detail-preserving capability. Using the Lena image as input, Experiment 1 compares performances of HFF with MF, MMEM [28], and WFM [22]. Experiment 2 uses three satellite weather pictures taken at different times to empirically validate the feasibility of the presented histogram-based approach. In addition, an interesting property of HFF, namely the similarity property of HFF is introduced. Finally,

TABLE V
COMPARISON OF HFF AND MF USING WEATHER_2

	NMSE$_i$	*NMSE$_o$*		*PSNR*	
γ	Input	MF	HFF	MF	HFF
0.0	0.00E-00	1.06E-03	4.42E-04	29.75	33.55
0.1	2.84E-02	1.40E-03	5.46E-04	28.54	32.63
0.2	5.64E-02	2.42E-03	9.22E-04	26.16	30.35
0.3	8.20E-02	4.53E-03	1.46E-03	23.44	28.76
0.5	1.32E-01	2.06E-02	2.93E-03	16.86	25.33
0.7	1.76E-01	6.02E-02	6.76E-03	12.20	21.70
0.9	2.17E-01	1.21E-01	1.86E-02	9.17	17.30

TABLE VI
$IPSNR$ OF HFF AND MF USING WEATHER_2

γ	*PSNR* (Corrupted Input)	*IPSNR* by MF	*IPSNR* by HFF
0.1	15.47	13.07	17.16
0.2	12.49	13.67	17.86
0.3	10.86	12.58	17.90
0.5	8.79	8.07	16.54
0.7	7.54	4.66	14.16
0.9	6.64	2.53	10.66

TABLE VII
COMPARISON OF HFF AND MF USING WEATHER_3

	NMSE$_i$	*NMSE$_o$*		*PSNR*	
γ	Input	MF	HFF	MF	HFF
0.0	0.00E-00	1.15E-03	5.12E-04	29.39	32.91
0.1	2.84E-02	1.49E-03	6.06E-04	28.27	32.18
0.2	5.64E-02	2.36E-03	9.98E-04	26.27	30.01
0.3	8.20E-02	5.23E-03	1.42E-03	22.81	28.48
0.5	1.32E-01	2.33E-02	3.06E-03	16.33	25.14
0.7	1.76E-01	6.79E-02	7.26E-03	11.68	21.39
0.9	2.17E-01	1.38E-01	2.05E-02	8.60	16.88

TABLE VIII
$IPSNR$ OF HFF AND MF USING WEATHER_3

γ	*PSNR* (Corrupted Input)	*IPSNR* by MF	*IPSNR* by HFF
0.1	15.47	12.80	16.71
0.2	12.49	13.78	17.52
0.3	10.86	11.95	17.62
0.5	8.79	7.54	16.35
0.7	7.54	4.14	13.85
0.9	6.64	1.96	10.24

Experiment 3 examines the effect of changing the value of threshold T on the performance of HFF.

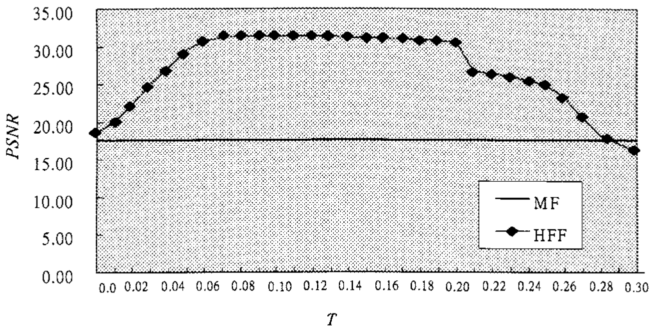
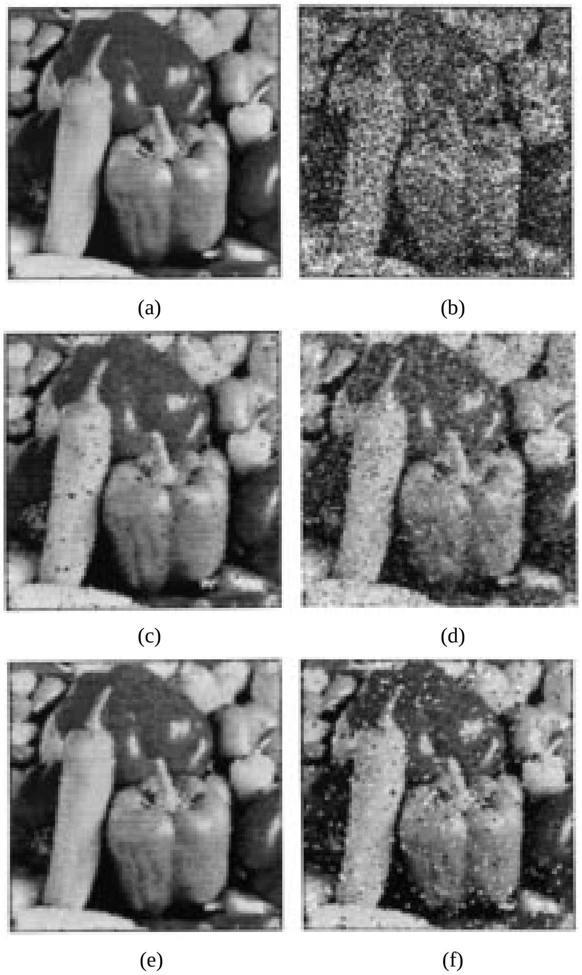


Fig. 10. Effect on performance of HFF when T varies.


(a) (b)

(c) (d)

(e) (f)

Fig. 11. (a) Uncorrupted image, (b) corrupted image with $\gamma = 0.5$ (c) restored by WFM, (d) restored by MMEM, (e) restored by HFF, (f) restored by MF.

As a measure of objective improvement for the restoration techniques discussed here, we refer to both the input normalized mean square error ($NMSE_i$), given by

$$NMSE_i = \frac{1}{t_p} \sum_{p=1}^{t_p} \left(x_p - x'_p\right)^2 \tag{7}$$

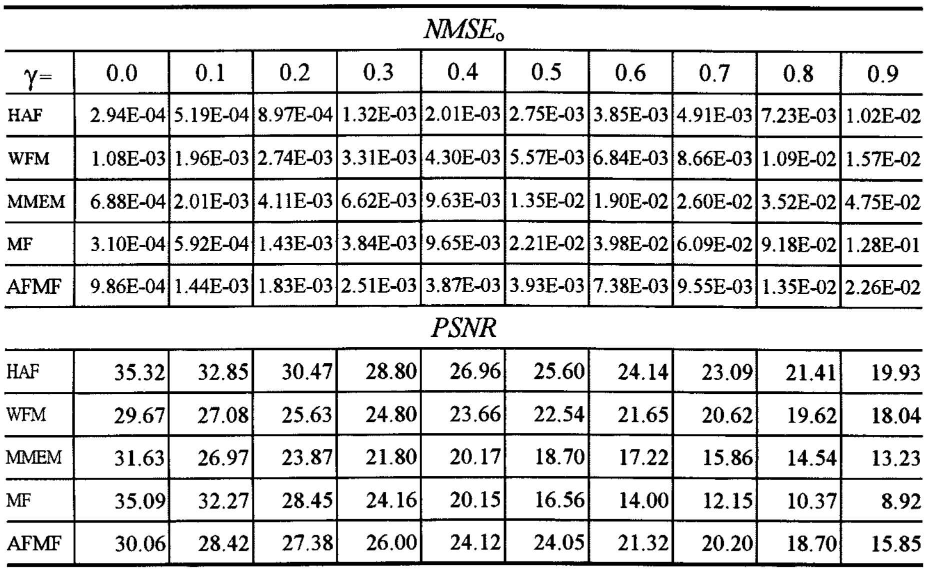
and the output normalized mean square error ($NMSE_o$) given by

$$NMSE_o = \frac{1}{t_p} \sum_{p=1}^{t_p} \left(x'_p - Y_p\right)^2 \tag{8}$$

where t_p is the total number of pixels in the image, x'_p denotes intensity of the pth pixel in the original (uncorrupted) image, x_p is the corresponding intensity in the corrupted image, and Y_p is the corresponding intensity in the filter output. Note that $0 \leq x_p,\, Y_p,\, x'_p \leq 1$. The peak signal-to-noise ratio ($PSNR$) is given by

$$PSNR = 10 \log_{10} \left(\frac{1}{NMSE} \right), \quad \text{where } NMSE = NMSE_o \text{ or } NMSE_i. \tag{9}$$

TABLE IX
RESULTS OF $NMSE$ AND $PSNR$ USING PEPPER IMAGE



					$NMSE_o$					
γ=	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
HAF	2.94E-04	5.19E-04	8.97E-04	1.32E-03	2.01E-03	2.75E-03	3.85E-03	4.91E-03	7.23E-03	1.02E-02
WFM	1.08E-03	1.96E-03	2.74E-03	3.31E-03	4.30E-03	5.57E-03	6.84E-03	8.66E-03	1.09E-02	1.57E-02
MMEM	6.88E-04	2.01E-03	4.11E-03	6.62E-03	9.63E-03	1.35E-02	1.90E-02	2.60E-02	3.52E-02	4.75E-02
MF	3.10E-04	5.92E-04	1.43E-03	3.84E-03	9.65E-03	2.21E-02	3.98E-02	6.09E-02	9.18E-02	1.28E-01
AFMF	9.86E-04	1.44E-03	1.83E-03	2.51E-03	3.87E-03	3.93E-03	7.38E-03	9.55E-03	1.35E-02	2.26E-02
					PSNR					
HAF	35.32	32.85	30.47	28.80	26.96	25.60	24.14	23.09	21.41	19.93
WFM	29.67	27.08	25.63	24.80	23.66	22.54	21.65	20.62	19.62	18.04
MMEM	31.63	26.97	23.87	21.80	20.17	18.70	17.22	15.86	14.54	13.23
MF	35.09	32.27	28.45	24.16	20.15	16.56	14.00	12.15	10.37	8.92
AFMF	30.06	28.42	27.38	26.00	24.12	24.05	21.32	20.20	18.70	15.85

Finally, the improvement in $PSNR, IPSNR$ can be expressed by

$$IPSNR = 10\log_{10}\left(\frac{NMSE_i}{NMSE_o}\right). \tag{10}$$

A. Experiment 1

Using the Lena image and $T = 0.1$, comparisons in terms of $NMSE_o$ and $PSNR$ between HFF, WFM, MF, and MMEM are shown in Figs. 6 and 7, respectively. Since simulation results of DEBF are similar to MMEM, they are omitted here. Clearly, HFF outperforms other filters for γ in the range [0.0, 0.9]. Note that MF performs satisfactorily only when $\gamma \leq 0.2$. Pictorially, Fig. 8(a) shows the original (uncorrupted) Lena, and Fig. 8(b) the image corrupted by impulsive noise with $\gamma = 0.5$. The performance difference in noise-removal by HFF and MF can be clearly seen by comparing Fig. 8(e) and (f), which depict the restored images by HFF and MF respectively. Other than its impressive noise suppression performance, HFF also outperforms other filters in preserving image details such as edges and corners. One can easily see this by comparing Fig. 8(e) with Fig. 8(a), particularly in hairs and the hat.

B. Experiment 2

This experiment uses a 256 × 256 satellite weather picture corrupted by impulsive noise as input data to HFF. The uncorrupted picture is denoted as Weather_1. The resulting $NMSE_i, NMSE_o$, and $PSNR$ (averaged over 10 different runs) are given in Table III for both HFF and MF. As in the case of Lena seen in Table III, HFF outperforms MF for the full range of impulsive noise probability. Corresponding results of $IPSNR$ are shown in Table IV.

Next, we will use the histogram-based membership functions derived from Weather_1 (noted as HMF_1) to test the noise suppression performance on other images having similar histogram statistics. If results are as good as that of Wearher_1, then we can justify the validity of the proposed method. To do this, we may use two other weather pictures taken at different times, and denote them as Weather_2 and Weather_3 respectively. These pictures and their histograms are shown in Fig. 9. Results of applying HMF_1 to Weather_2 are shown in Table V ($NMSE$ and $PSNR$) and Table VI ($IPSNR$). Similarly, results of applying HMF_1 to Weather_3 are shown in Tables VII and VIII.

By comparing Tables III, V, and VII, we see that all three cases show similar quality, despite that HFF performs slightly better on Weather_2 than on the other two pictures for $\gamma = 0.0$ to 0.9. More interestingly, the results of Weather_3 are more similar to those of Weather_1 than to Weather_2. This can be explained by examining Fig. 9 where Weather_3 clearly is more similar, in histogram distribution, to Weather_1 than to Weather_2. Thus, we have shown that membership functions derived from one input picture can be used to restore another image that has similar histogram statistics. Hereafter, this interesting feature of HFF will be referred to as the *similarity* property. More significantly, results of Experiment 2 in effect have empirically justified the feasibility of HFF filter, which is rooted in the image histogram statistics. From $IPSNR$ results shown in Tables IV, VI, and VIII, we have one more important observation to make. Namely, for MF the optimum $IPSNR$ performance always occurs at $\gamma = 0.2$, whereas much more steady performance in $IPSNR$ can be obtained from using HFF as γ varies.

C. Experiment 3

Before we proceed, we first note that $T = 0.1$ in all previous experiments. The next experiment explores the effect of changing threshold T on the performance of HFF. Shown in Fig. 10 is the results of $PSNR$ when the input Lena image is corrupted with $\gamma = 0.5$. For comparison, the $PSNR$ value of MF when $\gamma = 0.5$ is also given in Fig. 10. As can be seen, HFF has the best performance when $0.05 \leq T \leq 0.2$. Some important conclusions can be drawn from the previous three experiments.

1) With $0.05 \leq T \leq 0.2$, HFF always performs better than MF, MMEM, and WFM.
2) Conventional median filter is not suitable for cases where impulsive noise is heavy, especially when noise rate $\gamma \geq 0.2$.
3) Comparatively, HFF shows the best stability in $IPSNR$ criterion for the full range of γ. Note that the above conclusions are applicable not only to experiments on the "Lena" and "Weather" images, but also to experiments on the "Peppers" image, and results for which (with $T = 0.05$) are shown in Table IX and Fig. 11. In particular, simulation results of AFMF [13] are also shown in Table IX.

4) In [22], it has been shown (in fact, also verified in Experiment 1) that when $\gamma \geq 0.3$, WFM surpasses conventional filters such as stack filters [10], WOS [11], generalized mean filters [24], RCRS [25], and CWM [30]. In contrast, we have seen that HFF is superior to WFM, MF, AFMF and MMEM in impulse noise-removal for the full range of γ.

V. Concluding Remarks

A novel histogram-based fuzzy filter (HFF), which is capable of suppressing impulsive noise while preserving image details, has been presented. HFF performs noise suppression by exploiting the input image statistics. The construction of HFF begins with using the corrupted image to estimate the histogram of the original (uncorrupted) image. Then, initial parameters of a set of fuzzy membership functions are derived from the estimated histogram, which is followed by applying a conservation principle to adjust the initial parameters to obtain a set of well-conditioned membership functions. We have shown that HFF can achieve satisfactory image restoration without any training, thereby avoiding the *retraining* problem encountered in other neuro-fuzzy or fuzzy-neuro filters.

Simulation results have shown that HFF outperforms MF, MMEM [28], AFMF [13] and WFM [22] filters for the full range of noise probability γ. HFF shows good capability of removing impulsive noise, while preserving edge sharpness and thin lines. We have conducted three experiments to characterize the effectiveness of HFF, and results have verified that there exists a correlation between input histogram statistics and fuzzy membership functions. In this paper, histogram statistics of input image has been proven useful in i) deriving histogram-based membership functions for HFF to achieve effective image restoration, and ii) exploiting the *similarity* property and other interesting characteristics of HFF. In particular, we have shown that images with similar statistics (pictures not necessarily look seemingly) can be successfully restored by the membership functions inferred from an arbitrary image chosen from these similar images. The *similarity* property is believed to be useful in applying HFF to video transmission where successive image frames usually have similar histograms.

In determining the final output Y for an arbitrary pixel at (k, ℓ), there are two possibilities. First, if $p(k, \ell) \in N_{imp}$, the linear weights (i.e., Step 2 in Fig. 1) associated with the fuzzy membership degree produce candidates to be compared with a reference intensity $\hat{p}$, and the one closest to $\hat{p}$ forms the final output Y to replace $p(k, \ell)$. Second, if $p(k, \ell) \notin N_{imp}$, then $p(k, \ell)$ is replaced by $\hat{p}$. In practice, one can also choose to let $p(k, \ell)$ remain intact if $p(k, \ell) \notin N_{imp}$; doing so might further improve the detail-preserving performance at the expense of noise-removal capability. Either way, since the filtering process is highly *nonlinear*, study on the functional optimization [4], [31] property of HFF certainly forms one future research direction. We also consider applying HFF to eliminate Gaussian noise, for which accurately estimating the original input histogram is more difficult than in the case of impulsive noise.

References

[1] M. McLoughlin and G. Arce, "Deterministic properties of the recursive separable median filter," *IEEE Trans Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 98–106, 1987.

[2] G. Arce and R. Foster, "Detail-preserving ranked-order based filter for image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 83–98, 1989.

[3] T. Y. Young and K. S. Fu, *Handbook of Pattern Recognition and Image Processing*. New York: Academic, 1997.

[4] G. Qiu, "Functional optimization properties of median filter," *IEEE Signal Processing Lett.*, vol. 1, pp. 64–65, 1994.

[5] G. Arce and M. McLoughlin, "Theoretical analysis of max/median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 60–69, Jan. 1987.

[6] G. Qiu, "An improved recursive median filtering scheme for image processing," *IEEE Trans. Image Processing*, vol. 5, pp. 646–647, 1996.

[7] B. Kosko, *Neural Networks and Fuzzy System*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[8] J. Bezdek, "Editorial: Fuzzy models—What are they, and why?," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 1–5, 1993.

[9] T. Ross, *Fuzzy Logic With Engineering Applications*. New York: McGraw-Hill, 1995.

[10] B. H. Zeng, Zhou, and Y. Neuvo, "Synthesis of optimal detail-restoring stack filters for image processing," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, vol. 4, Canada, 1991, pp. 2533–2536.

[11] L. Yin and Y. Neuvo, "Adaptive FIR-WOS hybrid filtering," in *Proc. Int.. Symp. Circuits and Syst.*, vol. 6, San Diego, CA, 1992, pp. 2637–2640.

[12] H. Tang, "Smoothing of noisy images by the adaptive average differential iterative and the improved median filters," in *Proc. IEEE Region 10 Conf. Comp., Commun., Control and Power Eng.*, vol. 2, Beijing, China, 1993, pp. 1054–1057.

[13] P. Yu and C. S. Lee, "Adaptive fuzzy median filter," in *Proc. Int. Symp. Artificial Neural Networks*, Hsinchu, Taiwan, R.O.C., 1993, pp. F25–F34.

[14] E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press, 1986, vol. 1, pp. 318–362.

[15] J. H. Wang and M. D. Yu, "Image smoothing by adaptive fuzzy optimal filter," in *Proc. IEEE Int. Conf. Syst., Man, and Cybern.*, Vancouver, BC, Canada, 1995, pp. 845–848.

[16] T. Furuhashi, "Fusion of fuzzy/neuro/evolutionary computing for knowledge acquisition," *Proc. IEEE*, vol. 89, pp. 1266–1274, 2001.

[17] M. F. Azeem, M. Hanmandl, and N. Ahmad, "Generalization of adaptive neuro-fuzzy inference systems," *IEEE Trans. Neural Networks*, vol. 11, pp. 1332–1346, 2000.

[18] M. B. Menhaj and S. Amani, "A linguistic method of a neuro-fuzzy adaptive filter for online maneuvering target tracking," in *Proc. IJCNN*, vol. 1, 2001, pp. 605–610.

[19] S. Haykin, *Neural Networks*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[20] C. Thompson and L. Shure, *Image Processing TOOLBOX for USE With MATLAB*.

[21] I. Pitas and A. N. Venetsanopoulos, "Nonlinear mean filters in image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 573–584, 1986.

[22] C. S. Lee, Y. H. Kuo, and P. T. Yu, "Weighted fuzzy mean filters for image processing," *Fuzzy Sets Syst.*, vol. 89, pp. 157–180, 1997.

[23] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 116–132, 1985.

[24] A. Kundu, S. K. Mitra, and P. P. Vaidynnathan, "Applications of two-dimensional generalized mean filtering for removal of impulse noises from images," *IEEE Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 600–609, 1984.

[25] R. C. Hardie and K. E. Barner, "Rank conditioned rank selection filters for signal distortion," *IEEE Trans. Image Processing*, vol. 3, pp. 192–206, 1994.

[26] J. H. Wang and L.-D. Lin, "An improved median filter using minmax algorithm for image processing," *Electron. Lett.*, vol. 33, pp. 1362–1363, 1997.

[27] J. S. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[28] W. Y. Han and J. C. Lin, "Minimum–maxmum exclusive mean (MMEM) filter to remove impulse noise from highly corrupted image," *Electron. Lett.*, vol. 33, pp. 124–125, 1997.

[29] K. C. Lee, H. J. Song, and K. H. Sohn, "Detection-estimation based approach for impulsive noise removal," *Electron. Lett.*, vol. 34, pp. 449–450, 1998.

[30] S. J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 984–993, 1991.

[31] J. L. Paredes and G. R. Arce, "Optimization of stack filters based on mirrored threshold decomposition," *IEEE Trans. Signal Processing*, vol. 49, pp. 1179–1188, June 2001.