國立交通大學

電機與控制工程學系

博士論文

基於人類視覺系統之混合雜訊消除技術

Human-Visual-System-Based Mixed-Noise Removal Techniques

研 究 生：盧世茂

指導教授：林進燈

張志永

中華民國九十五年九月

基於人類視覺系統之混合雜訊消除技術

Human-Visual-System-Based Mixed-Noise Removal Techniques

研究生：盧世茂　　Student：Shih-Mao Lu

指導教授：林進燈 博士　　Advisor：Dr. Chin-Teng Lin

張志永 博士　　Dr. Jyh-Yeong Chang

國立交通大學

電機與控制工程學系

博士論文

A Dissertation
Submitted to Department of Electrical and Control Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in
Electrical and Control Engineering
Septemper 2006
Hsinchu, Taiwan, Republic of China

中華民國九十五年九月

基於人類視覺系統之混合雜訊消除技術

研究生：盧世茂　　　　　　　　　　　　　　　　指導教授：林進燈

張志永

國立交通大學電機與控制工程研究所

摘要

雜訊常是嚴重損害影像品質並且破壞重要細節的因素。附加在影像上的雜訊，大體上可以用兩種雜訊模型來模擬表現：突波雜訊及高斯雜訊。本論文提出兩級式的架構，相繼消除混合雜訊影像中非線性的突波雜訊及線性的高斯雜訊。在第一級消除突波雜訊中，基於決策機制的適應性中位數濾波器，應用於消除椒鹽雜訊 (Salt-Pepper noise)，而另一個適應性類神經網路的架構，則用於消除隨機的突波雜訊。然後，我們提出以人類視覺系統為基礎的影像品質增強系統，再補償修改過的圖素。在第二級消除高斯雜訊中，我們提出了線性改良式模糊規則為基礎的濾波器 (MFRB) 消除線性的高斯雜訊，並盡可能的保留影像的邊緣及細節部分。為了考量實際狀況，我們設計了幾個通用的 MFRB 濾波器來處理各種不同摻雜程度的高斯雜訊。利用所估測到影像雜訊值的大小，我們選擇對應到雜訊摻雜程度最接近的 MFRB 濾波器，去除影像中的高斯雜訊。根據實驗結果顯示，所提出的方法不論在客觀的評比數據(PSNR)或主觀的視覺感知上，都優於其他所比較的雜訊消除技術。

Human-Visual-System-Based Mixed-Noise Removal Techniques

Student: Shih-Mao Lu Advisor: Chin-Teng Lin
Jyh-Yeong Chang

Department of Electrical and Control Engineering
National Chiao-Tung University

Abstract

Noise always significantly damages an image and can corrupt most important details. Two noise models can adequately represent most noise added to images: additive impulse noise and Gaussian noise. In this thesis, we propose a two-stage filtering method to sequentially remove the mixed noises of images corrupted with nonlinear impulse and linear Gaussian noises as well. In the first stage, the decision-based recursive adaptive median filter is applied to remove the Salt-Pepper noise, and an adaptive two-level neural network noise reduction procedure is applied to remove the random-valued noise. Then, an HVS-directed neural-network-based image quality enhancement is applied to compensate the modified pixels. In the second stage, we derive a linear modified fuzzy rule-based (MFRB) filter to remove the linear type Gaussian noise while preserving the image edges and details as much as possible. For practical consideration, we design several sets of universal MFRB filters, to be utilized in correspondence to the estimated values of contaminated Gaussian noise variance in the image. The correspondent MFRB filter closest to the estimated Gaussian noise level will be selected to remove the Gaussian noise of the processed image. According to the experiment results, the proposed method is superior, both quantitatively and visually, compared to several other techniques.

誌謝

首先，感謝指導教授林進燈院長及張志永教授多年來的指導。無論是在專業研究上或是生活上的教導，都使我獲益良多。兩位教授敏銳宏觀的思維、創新前瞻技術的研究、努力不懈、實事求是、行事嚴謹的特質，都是非常值得我學習的地方。此外，也非常感謝諸位口試委員寶貴的意見，使得本論文得以更加完備。

在家人方面，感謝我的雙親盧龍修與林麗華，以及弟弟俊宇和妹妹薏如多年來的支持與鼓勵，使我得以無後顧之憂的專心於學業方面。

在學校方面，感謝梁勝富博士、蒲鶴章博士、張俊隆博士、秦群立博士等學長及同學們在學業上與生活上的幫忙與照顧。也感謝實驗室的伙伴們–文昌學長、建文、孝羽、宇文、芳誼、曉佩、得平… 及研究生團契的輔導–家齊老師、珊如老師和弟兄姊妹們–宥達、京荃、朝旺、佩琪、玉芳、安琪…，因為你們的出現及參與，使得我在交通大學的生活更加的多采多姿。

在教會方面，感謝陳正雄牧師、李啟誠牧師及小家人們–慧美姐、毓琴、有廷、其華、敬業、彥欽…的支持與鼓勵，感謝　神一路走來不斷的引導及看顧，使我得以能夠順利完成博士學位。

在此，謹以本論文獻給我的家人與關心我的師長與朋友們。

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

In the real-life, images are often contaminated by mixture of impulse and Gaussian noises of varying noise intensities due to the imperfection of sensors and communication channels when transmitted. The objectives of image noise removal are to remove the mixed noise and to retain the edges or other salient structures in the original image. Noise smoothing and edge enhancement are inherently conflicting processes, since smoothing a region will destroy an edge and sharpening edges might lead to enhance the unnecessary noise. Thus it is a difficult work for a universal algorithm that can remove different kinds and intensities of noise from images and preserve their sharpness and details. In this thesis, we propose a two-stage fuzzy filtering method to sequentially remove the mixed noises of images corrupted with nonlinear impulse and linear Gaussian noises well.

1.2. Impulse and Gaussian Noise Model

The additive impulse noise and Gaussian noise models can adequately represent most noise added to images. Impulse noise is the pixels randomly misfired and replaced by other values in an image. Such noise can be introduced due to transmission errors. With the noise ratio p, only p percent of the pixels in the image are replaced and others keep noise-uncorrupted. For example, when u is an image, $u_{i,j}$ will represent the intensity value of u at the pixel location (i,j) in the image domain. For images corrupted with impulse noise, the noisy image u is related to the original image u^0 by

$$u_{i,j} = \begin{cases} n_{i,j} \text{ with probability p} \\ u^0_{i,j} \text{ with probability } (1-\text{p}) \end{cases} \quad (1.1)$$

where the impulse noise is denoted by n.

In a variety of impulse noise models for images, fixed and random-valued impulse noises are mostly discussed. Fixed-valued impulse noise, known as the “Salt-Pepper” noise, is the corrupted pixels whose values are replaced with values equal to the maximum or minimum (255 or 0) of the allowable range with equal probability (p/2). The random-valued impulse noise is the corrupted pixels whose values are replaced by random values uniformly distributed in the range within [0, 255].

Additive Gaussian noise is characterized by adding to each image pixel a value which is drawn from a zero-mean Gaussian distribution. Such noise is usually introduced during image acquisition. For the case of additive Gaussian noise, the noisy image u is related to the original image u^0 by

$$u_{i,j} = u^0_{i,j} + n_{i,j} \quad (1.2)$$

1.3. Previous Research

Nowadays, image processing techniques have been well developed, but there are still some bottlenecks that have not been solved. Many image processing algorithms cannot work well in a noisy environment; therefore, the noise removal algorithm is adopted as a preprocessing module. A number of approaches have been developed for noise removal and listed as follows.

1.3.1. Existing Impulse Noise Removal Algorithms

The nonlinear filtering technique, standard median (SM) [1]−[2] filter, based on order statistic has been demonstrated generally superior to linear filtering (moving average) on suppressing impulse noise. However, median filter still tends to blur fine details and destroy edges while removing out the impulse noise. To achieve better performance, median filter has been modified in many ways, such as weighted median (WM) [3]−[4], center weight median filters (CWM) [5], adaptive-length median filter [6], the recursive medians [7]−[8] and the alpha-trimmed mean filter [9]. They were expected to increase the signal preservation but relatively decrease the noise suppression ability. Applying these algorithms altogether across the whole image without identification would inevitably remove the uncorrupted detail pixels, destroy the image quality, and cause additional blur.

For that reason, the decision-making schemes [10]−[12] were proposed in which only the identified noisy pixels are processed. Besides, the switching schemes [13]−[17] provide adaptive decision to recover the noisy pixels based on several filters to remain noise-free pixels unchanged. In [18], the progressive switching median filter was proposed for only fixed-valued impulse noise removal. Also, the weighting-average linear combinations of nonlinear median-based filters through learning-rule optimization have been proposed [19]−[21]. Although satisfactory results have been obtained, they tend to remove fine details or retain too much of the noise due to undetection or misdetection of the noise [22]. In addition, since the noisy pixels are replaced without taking into account local features, details and edges are not recovered satisfactorily, especially when the noise level is high. The thresholding filtering [23] which composed of new efficient noise detectors was proposed to prevent the misclassification of noise-free pixels. The edge-directed noise detection and suppression strategy was proposed to preserve the details and edges [24]. Two stage approaches that combine noise identification and edge preserving supplementary have been proposed trying to remove the noise cleanly and keep the detail information well [25]−[26].

1.3.2. Existing Gaussian Noise Removal Algorithms

On the other hand, linear filtering algorithms, such as Wiener [27] and Kalman [28] filters are essentially low-pass filters and well-known for their ability to remove the Gaussian noise, but they cannot remove the impulse noise well and tend to blur the fine features in the image. To remove the mixed noise in an image, a combination approach or hybrid filters [29]−[30] have become a promising approach. Garnett *et al.* [31] introduced the local image static - Rank-Ordered Absolute Differences (ROAD) to quantify how different in intensity the particular pixels are from their most similar neighbors, and proposed the trilateral filter, modified from the bilateral filter [32], for removing mixed Gaussian and impulse noise.

1.3.3. Existing Neural and Fuzzy Noise Removal Algorithms

Since neural networks have the ability to learn from examples and fuzzy systems have the ability of reasoning to deal with uncertainty, they also have a growing number of applications in image noise removal in the past few years [33]−[40]. Zhang *et al.* [33] proposed the fuzzy techniques to detect the impulse noise and to find the ultimate remote window to replace the noise-liked pixel with a linear combination of the pixels. Schulte *et al.* [34] proposed a fuzzy derivative estimation with fuzzy rules in the first stage for noise detection and fuzzy smoothing of neighboring pixels in the second stage. Lee *et al.* [35] proposed a fuzzy image filter based on genetic learning process. Russo [36]−[37] proposed a recursive Neuro-Fuzzy filter with specifically designed original multiple output network structure and learned the parameters based on the genetic algorithm. M. E. Yüksel *et al.* [38] proposed a new impulse noise detector comprises two identical Neuro-Fuzzy subdetectors combined with a decision maker.

On the other hand, there are mainly three kinds of fuzzy approaches used in mixed noise removal of an image. The first kind is the fuzzy weighted average filter [41] and fuzzy weighted median filter [42]. Peng [43]−[44] proposed a multi-level adaptive fuzzy (MLAF) filter, which uses fuzzy sets to adaptively combine simple linear and nonlinear filters to remove varying mixed noise with different levels. The second kind is the fuzzy logic filter, which suggests that individual pixels should not be uniformly fired by each of the fuzzy rules. Choi *et al.* [47] derived three different filters for each of the three objectives using the fuzzy weighted least squares (FWLS) method. And he defined the criteria for selecting each of the filter based on the local context using the fuzzy rules. Taguchi [48] proposed the modified fuzzy filter (MFF) with new local characteristic calculated with fuzzy rules by using multiple difference values between arbitrary pixels in the filter window. Farbiz *et al.* [49] proposed the fuzzy logic filter (FLF), which adopted the general structure of fuzzy if-then-else rules mechanism. The S-type fuzzy function enables the non-uniform firing of the basic fuzzy rules. For the third kind uses fuzzy reasoning which is embedded into the neural network structure through genetic learning algorithm. It is able to adapt the filtering action to different distributions of mixed noise.

1.3.4. Existing Human Visual System Algorithms

Many researches have been made on discovering the characteristics of HVS for years. The characteristics of the HVS have been incorporated into the digital image processing such as watermark encoder design, digital image compression, and image recognition etc [51]−[59]. The perceptual redundancies inherent in a still image are basically due to the inconsistency in sensitivity of the HVS to stimuli of varying levels of contrast and luminance changes in the spatial domain. It was found that the perception of HVS is more sensitive to luminance contrast rather than the uniform brightness [60]. In addition to the magnitude difference between object and the background, different structures of images also cause different visual perceptions for HVS. Many features have been proposed based on the block DCT in frequency domain and Wavelet [61]−[62]. A novel fuzzy decision system inspired by HVS is proposed to classify the image into human perception sensitive and non-sensitive regions.

1.4. Brief Summary

In this chapter, several noise removal algorithms are presented. As mentioned above, it is difficult to remove the mixed noises without blurring the edge and details information. The two tasks, involving suppressing the impulse noise and removing the Gaussian noise in an image, are very different in characteristics because each of them respectively facilitates the nonlinear and linear filtering operations. Besides, the presence of impulse noise can seriously degrade the performance of a restoration technique that is designed mainly to remove Gaussian type noise. Consequently, we propose in this thesis a two-stage filtering technique to remove the nonlinear impulse and linear Gaussian noises sequentially. In the first stage, the decision-based recursive adaptive median filter is applied to remove the Salt-Pepper noise and an adaptive two-level neural network noise reduction procedure is applied to remove the random-valued noise described in Chapter 3. Then, an HVS-directed neural-network-based image quality enhancement described in Chapter 2 is applied to compensate the modified pixels. In the second stage, we derive a linear modified fuzzy rule-based (MFRB) filter to remove the linear type Gaussian noise while best preserving image details, and we combine the two stage filtering and proposed the universal MFRB filter for mixed noise removal described in Chapter 4. Finally, conclusions and perspectives are described in Chapter 5.

2. Human Visual System Based Image Quality Enhancement

In this chapter, we proposed an HVS-directed neural-network-based image quality enhancement. The fuzzy decision rules inspired by human visual system (HVS) are proposed to classify pixels of the image into human perception sensitive class and non-sensitive class. A neural network is proposed to enhance the sensitive regions to perform better visual quality. In addition, the proposed fuzzy decision rules combined with the neural network can balance the trade-off between speed and quality for different applications by just adjusting a threshold parameter. The ideas of using the fuzzy system to implement human visual system and using the neural network for image enhancement are new. The learning results can be directly implemented through the look up table (LUT) to reduce the computational cost and the system complexity for practical applications. It can be combined with any other filtering approach to enhance the visual quality.

2.1. Introduction

The visual acuity of the eye is generally regarded as the most important factor for the ability of the eye for seeing objects. The acuity of the eye is usually measured by acuity tests where the minimum visible separation is measured of black rings with a small interrupted part. These tests are used for decisions that play a role in the properties of the human visual system.

Objects can generally be better distinguished from each other or from their background, if the difference in luminance or color is large. Of these two factors, luminance plays the most important role. In practice, it appears that it is not the absolute difference in luminance that is important, but the relative difference. This relative difference can be expressed by the ratio between two luminance values, which is called contrast ratio, or by the difference between two luminance values divided by the sum of them, which is simply called contrast. Objects that only have a small contrast with respect to their background are difficult to observe. The eye is more sensitive for the observation of objects, if the required amount of contrast is lower. The reciprocal of minimum contrast required for detection is called contrast sensitivity.

In this chapter, Section 2.2 studied the characteristics of spatial visual for improvement in applications of digital image processing. The structure of the image quality enhancement system is described in Section 2.3. Section 2.4 gives conclusions of this chapter.

2.2. Characteristic of Spatial Vision

When we speak of spatial vision, we refer to the visual system's ability to detect and analyze changes in brightness across space.

2.2.1. Characteristics of Sine Wave Grating

Comparing the two gratings in Fig. 2.1, the grating on the bottom shows more alterations between light and dark than the grating on the top. In a given space (the space taken up by the photograph), there are more alterations, or cycles, in the bottom grating. Therefore, this bottom grating has a higher spatial frequency than the top grating. In other words, it has more cycles within unit given space. The top grating shows a relatively lower spatial frequency; it has fewer cycles per space.

It is common to refer to the spatial frequency of a grating in terms of cycles per degree rather then cycles per space. The number of cycles per degree of visual angle is specified rather than the number of cycles per unit of space. As we shall see, the specification of spatial frequency in terms of cycles per degree offers many practical advantages.

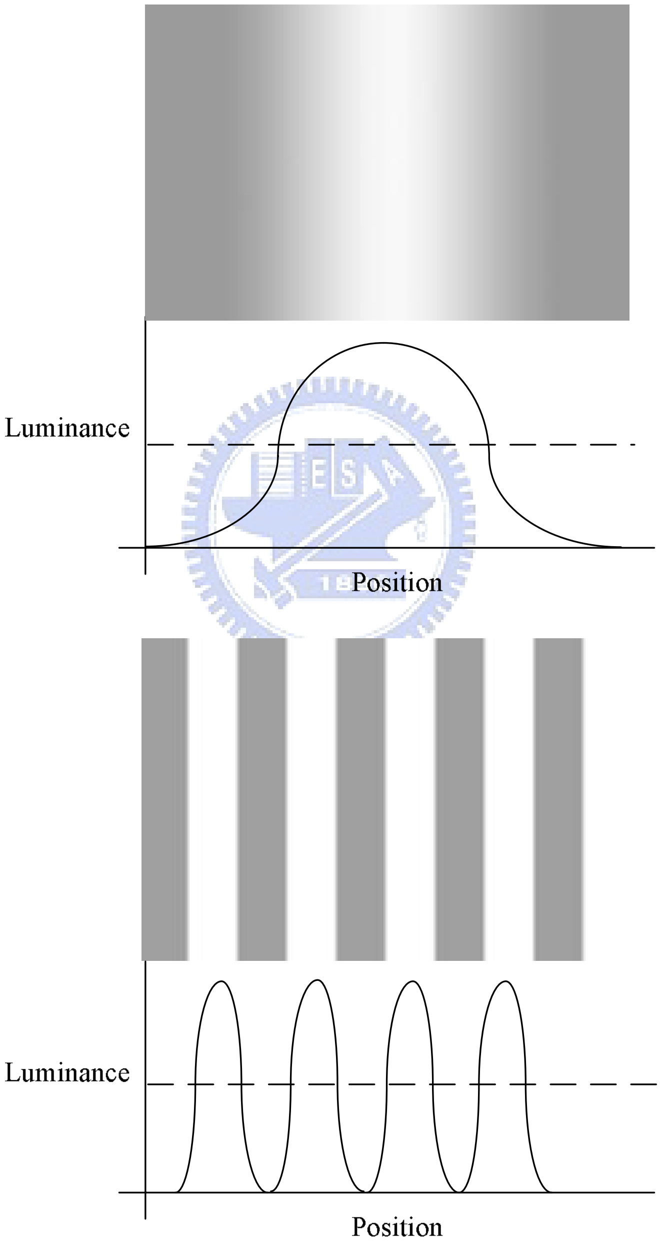


Fig. 2.1. A low frequency spatial grating (top) and a higher frequency grating (bottom). Both gratings have the same contrast.

2.2.2. Characteristics of Contrast

The top of Fig. 2.2 shows a grating of low contrast along with its luminance profile. The bottom of this figure shows a grating of the same spatial frequency but of a higher contrast. The dashed line across the luminance profiles represents the average luminance of the gratings. Note that the average luminance l_{ave} for both gratings is the same; however, the grating on the bottom has a greater difference between its peak and the average luminance value. Consequently, it has a higher contrast.

Contrast is often defined by way of a formula:

$$\text{Contrast}=\frac{\Delta l}{l_{ave}}, \tag{2.1}$$

where

Δl: The difference in luminance between the peak or trough and the average luminance l_{ave}.

l_{ave}: The average luminance of the grating. (the average of the light peaks and dark troughs)

This formula states that contrast is the ratio of the difference between the peak luminance and average luminance to the average luminance. Although this formula is useful in defining and understanding contrast, it is not very practical for the measurement of contrast. In the laboratory, the scientist may find it easiest to measure the peak luminance $l_{\max}$ and the minimum luminance $l_{\min}$ to calculate the contrast. The formula used is

$$\text{Contrast}=\frac{l_{\max}-l_{\min}}{l_{\max}+l_{\min}}, \tag{2.2}$$

where

$l_{\max}=l_{ave}+\Delta l$

$l_{\min}=l_{ave}-\Delta l$

It can be demonstrated that Eq. (2.2) provides the same result as Eq. (2.1). Substituting for $l_{\max}$ and $l_{\min}$ we have

$$\frac{l_{\max}-l_{\min}}{l_{\max}+l_{\min}}=\frac{(l_{ave}+\Delta l)-(l_{ave}-\Delta l)}{(l_{ave}+\Delta l)+(l_{ave}-\Delta l)}=\frac{\Delta l}{l_{ave}}\,. \qquad (2.3)$$

It should be clear from examining Fig. 2.2 and the preceding formulae that contrast ranges between 0% and 100%. Contrast cannot be greater then 100% because of the physical impossibility of making Δl greater than l_{ave}. The trough of the luminance profile is at zero luminance when $\Delta l = l_{ave}$. It is not possible to have less than zero luminance.

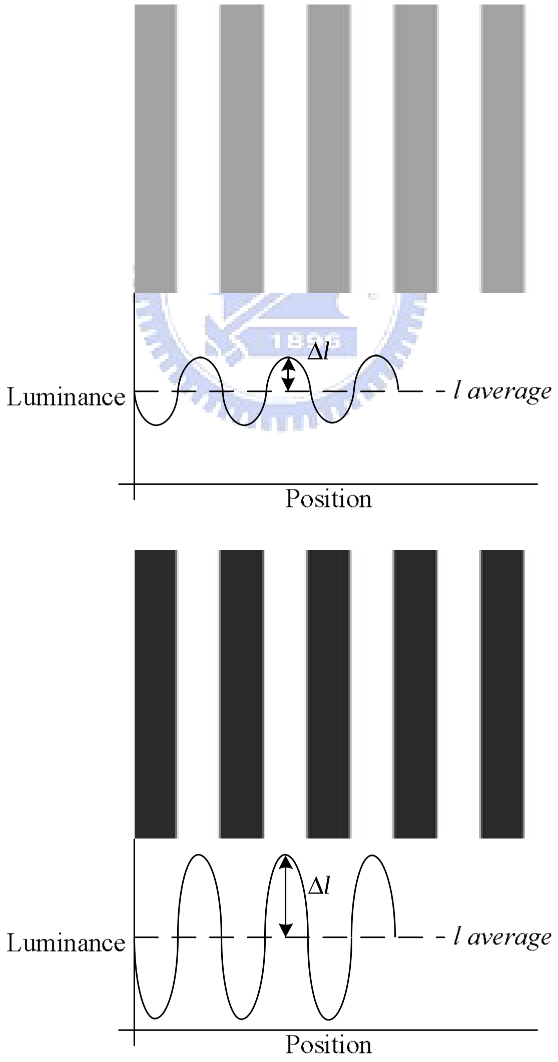


Fig. 2.2. A low contrast spatial grating (top) and a higher contrast grating (bottom). Both grating are of the same spatial frequency.

2.2.3. The Human Visibility Threshold

It was found that the perception of HVS is more sensitive to luminance contrast rather than the uniform brightness. Fig. 2.3 shows the actual visibility thresholds called just-noticeable-distortion (JND) corresponding to different background luminance and it was verified by a subjective experiment [60]. The experiments were conducted in a dark room and a square area was located in the center of a flat field of constant gray level. Through varying the amplitude of the object, the visibility threshold for each gray level was determined when the object was just noticeable. The ability of human eyes to tell the magnitude difference between an object and its background depends on the average value of background luminance. We can find that the visibility threshold is lower when the background luminance is within the interval from 70 to 150, and the visibility threshold will increase if the background luminance becomes darker or brighter away from this interval. In addition, high visibility threshold will occur when the background luminance is in very dark region.

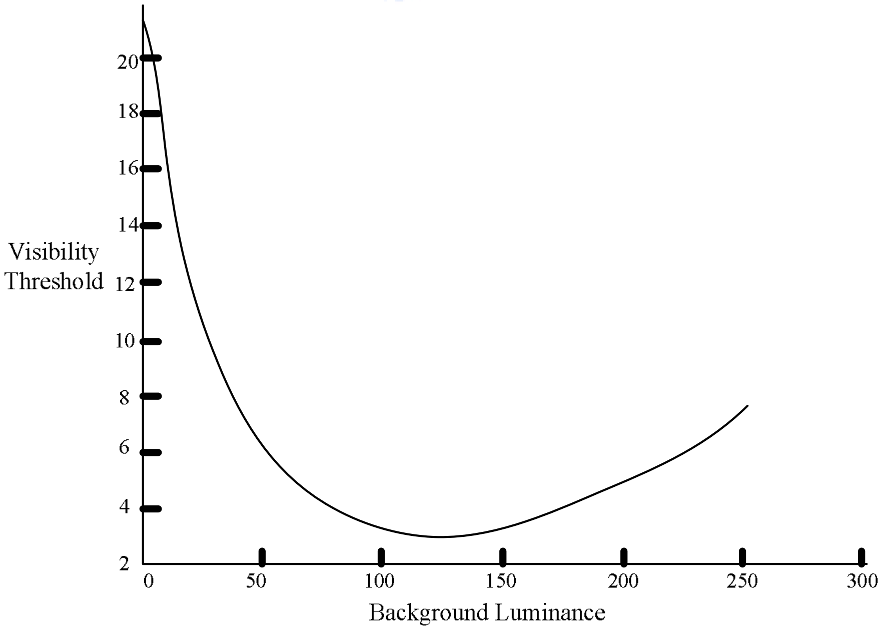


Fig. 2.3. Visibility thresholds corresponding to different background luminance.

2.2.4. The Contrast Sensitivity Function

To determine a human contrast sensitivity function (HCSF), the subject is presented with a sine wave grating of a given spatial frequency. A typical HCSF is shown in Fig 2.4. Note that the coordinates are plotted in log units. The HCSF is a band pass function: it shows distinct peak sensitivity and decreasing sensitivity on either side of this peak. A typical HCSF peaks in the region of 4 cycles/degree. The human visual system is most sensitive to this frequency; it will detect a grating of this frequency at lower contrast levels than it will detect gratings of other frequencies.

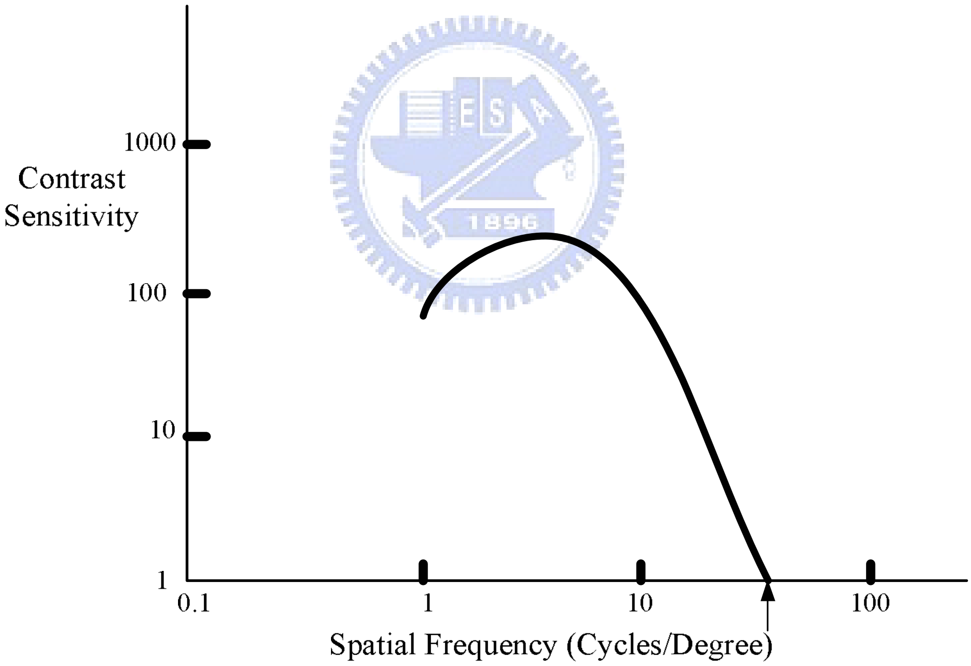


Fig. 2.4. Typical human contrast sensitivity function. The arrow points to the cutoff frequency.

2.3. Structure of the Image Quality Enhancement System

For image analysis, we make use of the properties of human visual system (HVS) to obtain the features of images. Therefore, we could realize which region would be worth quality enhancement, since human eyes would be usually more sensitive to this region. For sensitive regions, we propose an adaptive neural network to enhance the visual quality to match the characteristics of human visual perception.

2.3.1. HVS-Directed Image Analysis

There are three input variables, visibility degree (VD), structural degree (SD), complexity degree (CD), and one output variable (Mo) in the proposed fuzzy decision system.

2.3.1.1. Visibility Degree (VD)

To apply the human visibility threshold as mentioned is Section 2.2.3, the actual data and approximated nonlinear equation of visibility thresholds corresponding to different background luminance was shown in Fig. 2.5.

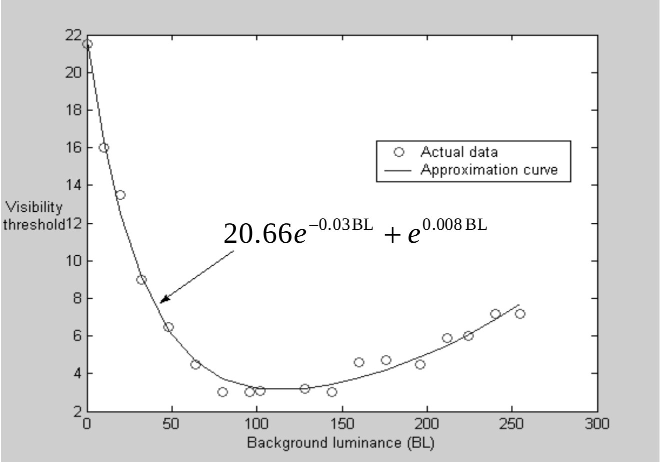
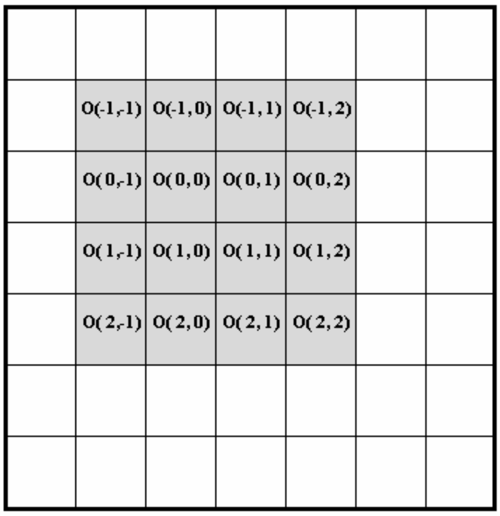


Fig. 2.5. Visibility thresholds corresponding to different background luminance.



	O(-1,-1)	O(-1,0)	O(-1,1)	O(-1,2)		
	O(0,-1)	O(0,0)	O(0,1)	O(0,2)		
	O(1,-1)	O(1,0)	O(1,1)	O(1,2)		
	O(2,-1)	O(2,0)	O(2,1)	O(2,2)		

Fig. 2.6. The 4×4 sliding (overlapping) window blocks applied to the image quality enhancement.

In order to obtain the input variables corresponding to each sliding block shown in Fig. 2.6, two index parameters called background luminance (BL) and difference (D) are defined at first. BL is the average luminance of the sliding block proposed to approximate the actual background luminance and can be calculated by

$$\mathrm{BL} = \frac{1}{23}\sum_{i=-1}^{2}\sum_{j=-1}^{2} O(i,\, j)\times B(i,\, j)\,, \tag{2.4}$$

where

$$B(i,\, j) = \begin{bmatrix} 2 & 2 & 2 & 1 \\ 2 & 0 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \tag{2.5}$$

and the denominator 23 in Eq. (2.4) is the weighted sum of all elements in Eq. (2.5) for normalization.

Feature D is the difference between the maximum pixel value and the minimum pixel value in the sliding block and can be calculated by

$$\mathrm{D} = \max(O(i,\, j)) - \min(O(i,\, j))\,. \tag{2.6}$$

A nonlinear function V(BL) is also designed to approximate the relation between the visibility threshold and background luminance (as Fig. 2.5), and can be represented as

$$\mathrm{V(BL)} = 20.66e^{-0.03\mathrm{BL}} + e^{0.008\,\mathrm{BL}} . \tag{2.7}$$

The first input variable of the fuzzy decision system, VD, is defined as the difference between D and V (BL) and can be represented as

$$\mathrm{VD} = \mathrm{D} - \mathrm{V(BL)} . \tag{2.8}$$

If VD > 0, it means the magnitude difference between the object and its background exceeds the visibility threshold and the object is sensible. Otherwise, this object is not sensible.

The other two input variables, SD and CD, are used to indicate whether the pixels in the sliding block own edge structure.

2.3.1.2. Structural Degree (SD)

SD shows if the sliding block is a high contrast region and the pixels in the block can be obviously separated into two clusters. It is calculated by

$$\mathrm{SD} = \frac{\left|\max(O(i,j)) \text{ - } \mathrm{mean}(O(i,j)) \text{ - } [\mathrm{mean}(O(i,j)) \text{ - } \min(O(i,j))]\right|}{\max(O(i,j)) - \min(O(i,j)))} , \tag{2.9}$$

where

$$\mathrm{mean}(O(i,j)) = \frac{1}{16}\sum_{i=-1}^{2}\sum_{j=-1}^{2} O(i,j) . \tag{2.10}$$

An illustration of Eq. (2.9) is shown in Fig. 2.7. According to Fig. 2.7, Eq. (2.9) can be expressed as $|\sigma_1 - \sigma_2| / (\sigma_1 + \sigma_2)$, where $\boldsymbol{\sigma}_1 = \max(O(i,j)) \text{ - } \mathrm{mean}(O(i,j))$ and $\boldsymbol{\sigma}_2 = \mathrm{mean}(O(i,j)) \text{ - } \min(O(i,j))$. So the SD has been normalized to $[0,1]$ and this rule can also be applied to images with different intensity range. If SD is small

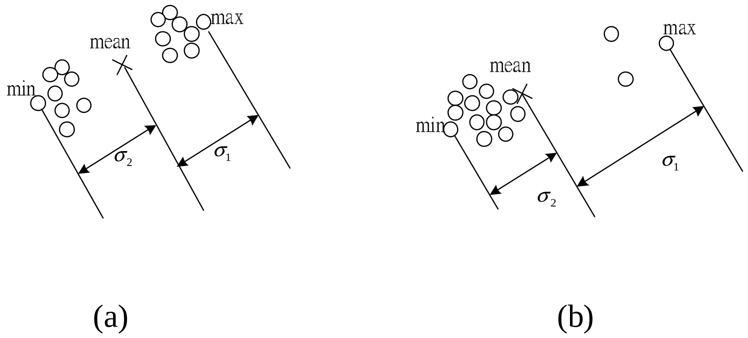


Fig. 2.7. An illustration of the relation between SD parameter and the distribution of pixels in a sliding block.

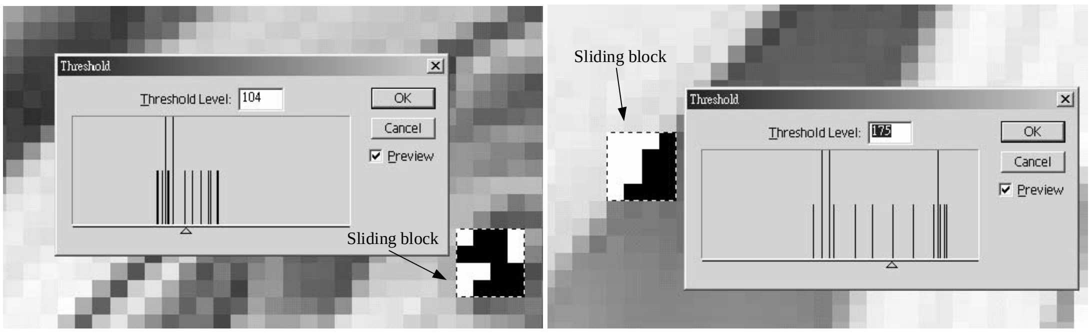
(close to 0), σ_2 and σ_1 are close (see Fig. 2.7(a)), it means the pixels in the block can be separated into two even clusters. The block may contain edge or texture structure. On the contrary, if SD is a large value, $0 << |\sigma_1 - \sigma_2|$ (see Fig. 2.7(b)), it means pixel number of one cluster and that of the other cluster are not even, thus, the block may contain noise.

2.3.1.3.Complexity Degree (CD)

Fig. 2.8(a) and (b) show a texture structure and a delineated edge structure in a sliding block, respectively. In these two plots, pixel numbers of the two clusters are the same. Therefore, the SD values corresponding to these two structures are close. Since the proposed neural network is used to compensate the sensitive regions such as Fig. 2.8(b), CD input variable based on differential process is employed to tell the delineated edge structure from texture structure. It is calculated by

$$\mathrm{CD} = \sum_{i=-1}^{2}\sum_{j=-1}^{2}\left|4O^{'}(i,j) - [O^{'}(i+1,j) + O^{'}(i-1,j) + O^{'}(i,j+1) + O^{'}(i,j-1)]\right|, \quad (2.11)$$

where $O^{'}(i,j)$ is the binarized version of $O(i,j)$. Assuming *mean(O)* is the mean



Threshold Level: 104
☑ Preview

Threshold Level: 175
☑ Preview

(a) (b)

Fig. 2.8. Portions of (a) the sliding block containing texture structure, (b) the sliding block containing edge structure.

gray value of the sliding block, $O^{'}(i, j)$ is defined as :

$$O'(i, j) = \begin{cases} 1 \;, \text{ if } O(i, j) \geq mean(O), \\ 0 \;, \text{ otherwise.} \end{cases} \tag{2.12}$$

In Eq. (2.11), each pixel in the 4×4 sliding block takes the 4-directional local gradient operation and CD is the summation of the 16 local gradient values. If CD is a large value, it means the block may contain texture structure. On the contrary, if CD is a small value, the block may contain delineated edge structure.

2.3.1.4.Fuzzy Decision System

In the proposed HVS-based Fuzzy decision system, the input variable VD has two fuzzy sets, N (negative) and P (positive). The input variable SD has three fuzzy sets S (small), M (medium), and B (Big). The input variable CD has three fuzzy sets, S (small), M (medium), and B (Big). The membership functions corresponding to VD, SD, and CD are shown in Fig. 2.10(a)−(c), respectively. In order to determine the fuzzy membership functions, seven nature images were used to generate the model. The images were separated into smooth, texture and edge regions by the admission of the majority (seven of ten subjects). Then the ranges of VD, CD and SD proposed in

Eqs. (2.8), (2.9) and (2.11) corresponding to these regions were evaluated. Finally, the membership functions of VD, CD and SD could be designed according to the distribution ranges of the parameters in these regions, respectively. Mo is the output variable and the membership functions corresponding to Mo are shown in Fig. 2.10(d). It has two fuzzy sets, NN (neural network) and OP (original pixel).

Seven fuzzy decision rules are used in the proposed fuzzy system and represented as follows:

1. If VD is N then Mo is OP
2. If SD is B then Mo is OP
3. If CD is B then Mo is OP
4. If VD is P and SD is S and CD is S then Mo is NN
5. If VD is P and SD is S and CD is M then Mo is NN
6. If VD is P and SD is M and CD is S then Mo is NN
7. If VD is P and SD is M and CD is M then Mo is OP.

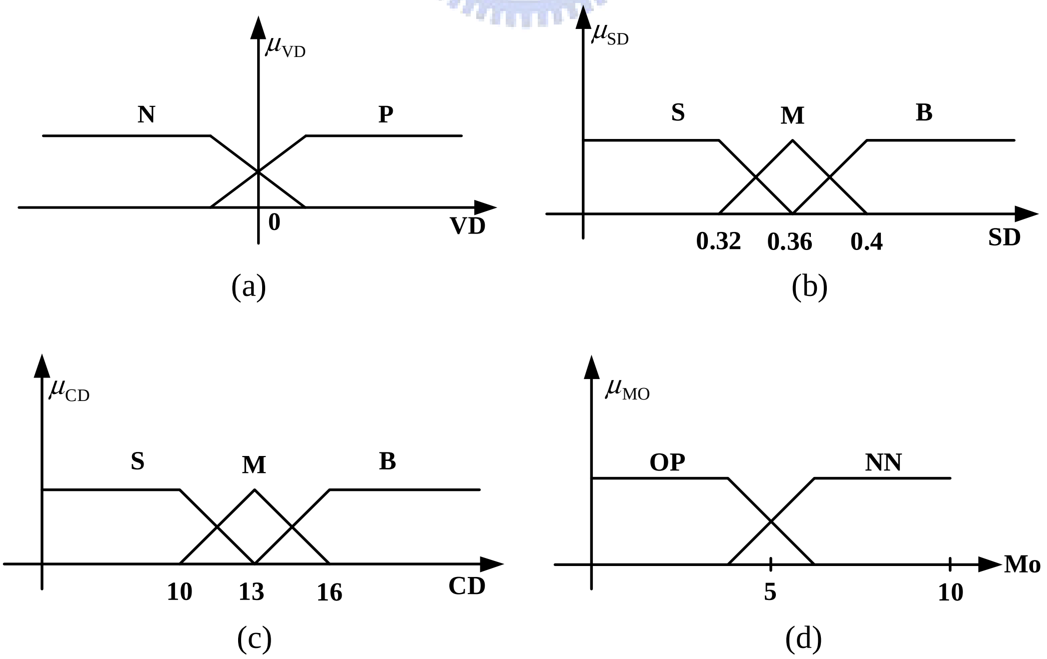


Fig. 2.9. (a)-(d) Membership functions of fuzzy sets on input variables VD, SD, CD, and output variable Mo, respectively.

The numerical value of Mo after defuzzification is compared with a threshold value, Th, where Th is preferably set as the value 5 by experiments. When Mo$\geq$Th, the adaptive neural-network (NN) compensation module with angle evaluation would be chosen; otherwise, the original pixel (OP) value would be used.

2.3.2. Angle Evaluation

As Mo$\geq$Th, the fuzzy system identifies the reference pixel as sensible delineated edge and the trained adaptive neural-network model is chosen for quality enhancement according to its corresponding edge angle. The angle evaluation is performed to determine the dominate orientation of the sliding block. The flow diagram of angle evaluation is shown in Fig. 2.10 to compute the orientation angle of each neighborhood of the original image pixel. When the orientation angle of $O(i, j)$ denoted as $A(i, j)$ is computed, the luminance values of the original pixels nearby $O(i, j)$ are used for the following computations:

$$\begin{aligned} Dx(i, j) = {} & O(i-1, j-1) + 2O(i-1, j) + O(i-1, j+1) \\ & - (O(i+1, j-1) + 2O(i+1, j) + O(i+1, j+1)) \end{aligned}, \qquad (2.13)$$

$$\begin{aligned} Dy(i, j) = {} & O(i-1, j-1) + 2O(i, j-1) + O(i+1, j-1) \\ & - (O(i-1, j+1) + 2O(i, j+1) + O(i+1, j+1)) \end{aligned}, \qquad (2.14)$$

$$A(i, j) = -\frac{180}{\pi}[\tan^{-1}(\frac{Dy(i, j)}{Dx(i, j)})], \qquad (2.15)$$

where $-1 \leq i \leq 2$ and $-1 \leq j \leq 2$.

The obtained angle of each pixel in the sliding window is quantized into eight quantization sectors such as θ = 22.5×k degrees, where k = 0, 1, ... , 7. Assuming θ is the most frequently quantized angle in the window; it is regarded as the dominant orientation of the reference edge pixel. The corresponding weighting coefficient W_θ is derived from the off-line training neural network for compensation filtering.

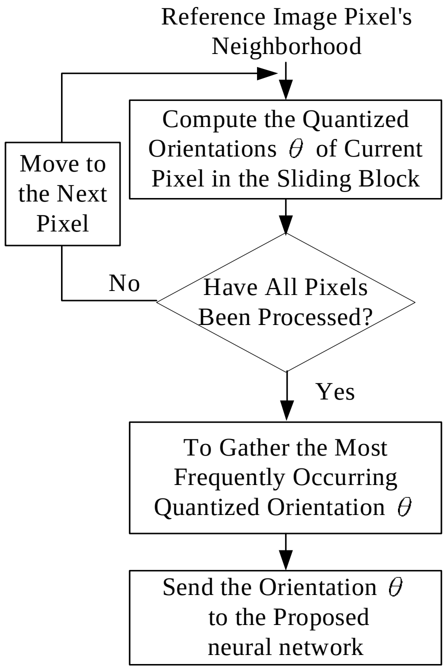

Reference Image Pixel's Neighborhood
Compute the Quantized Orientations θ of Current Pixel in the Sliding Block
Move to the Next Pixel
No
Have All Pixels Been Processed?
Yes
To Gather the Most Frequently Occurring Quantized Orientation θ
Send the Orientation θ to the Proposed neural network

Fig. 2.10. Flow diagram of angle evaluation.

2.3.3. Neural-Network-Based Image Compensation

The function of the proposed neural network is to obtain the weights W_θ defined in Eq. (2.4), where θ represents the quantized dominant orientation of the reference pixel. Thus, the proposed neural network is used to obtain 8 sets of weighting matrices through training. Each weighting matrix W_θ can be represented as

$$W_\theta(i,j)=\begin{bmatrix} w_{-1-1} & w_{-10} & w_{-11} & w_{-12} \\ w_{0-1} & w_{00} & w_{01} & w_{02} \\ w_{1-1} & w_{10} & w_{11} & w_{12} \\ w_{2-1} & w_{20} & w_{21} & w_{22} \end{bmatrix}. \tag{2.16}$$

In order to use supervised learning algorithms to train the proposed neural network, we have to obtain the desired input-output patterns such that the differences of network outputs and the corresponding desired outputs can be used to define the cost function as the goal to minimize. In this section, several clean image portions with dominant orientation are used as training patterns. Assuming a clean image portion is denoted as I, the noise-corrupted version of I has been processed by the proposed noise removal method in the first stage and the filtered result is denoted as I'. According to Fig. 2.11, let $\mathrm{I}'(i,j)$ be the reference pixel, where $O(0,0)=\mathrm{I}'(i,j)$, and it is classified as an edge pixel with dominant orientation θ after angle evaluation. The input of the neural network can be defined as $IP=\theta$ and the network output is the compensated pixel value of $\mathrm{I}'(i,j)$. The pixel value of $\mathrm{I}(i,j)$ obtained from the clean original image is used as the desired output of the neural network for training.

When the input-output patterns are given, the following task is to train a neural network to match the input-output relations. A new neural network as shown in Fig. 2.11 is proposed for image compensation. It is a 4-layer network with two hidden layers. The input layer consists of one node corresponding to θ. The second layer (1st hidden layer) consists of M nodes denoted as $g(z_i)$, where M is 200 in our experiments, and the bipolar sigmoid function is used as the activation function. The weighting vector between the first and the second layer is denoted as $\overline{U}$. The third layer (2nd hidden layer) includes 16 nodes and the bipolar sigmoid function is also used as the activation function. The weighting vector between the second and the third layers is denoted as $\overline{V}$. The output value of each node in the third layer is denoted as $y(s_i)$ and represents an element of the weighting matrix W_θ given in Eq. (2.16), where $y(s_i)=w_{jk}$, $i=4(j+1)+k+2$, $1\leq i\leq 16$, $-1\leq j\leq 2$, and $-1\leq k\leq 2$. The fourth layer is the output layer with one output node and its output value represents the compensated pixel value of $\mathrm{I}'(i,j)$. The vector between the third and the fourth layers is denoted as $\bar{\mathrm{I}}'_\theta$. It represents the vector of the sixteen neighborhood pixels of the reference pixel $\mathrm{I}'(i,j)$ with dominant orientation θ as follows:

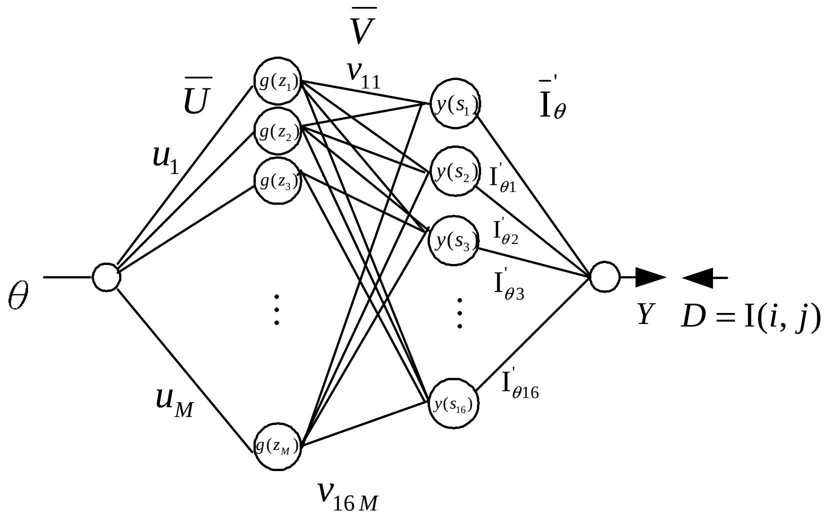


Fig. 2.11. The proposed feedforward neural network for image quality enhancement.

$$\overline{\mathrm{I}}'_\theta = \begin{bmatrix} \mathrm{I}'_{\theta 1} \\ \mathrm{I}'_{\theta 2} \\ \mathrm{I}'_{\theta 3} \\ \mathrm{I}'_{\theta 4} \\ \mathrm{I}'_{\theta 5} \\ \vdots \\ \mathrm{I}'_{\theta 16} \end{bmatrix} = \begin{bmatrix} \mathrm{I}'\ (i-1, j-1) \\ \mathrm{I}'\ (i, j-1) \\ \mathrm{I}'\ (i+1, j-1) \\ \mathrm{I}'\ (i+2, j-1) \\ \mathrm{I}'\ (i-1, j) \\ \vdots \\ \mathrm{I}'\ (i+2, j+2) \end{bmatrix} . \tag{2.17}$$

Then the system estimation output can be calculated by

$$Y = \sum_{x=1}^{16} y(s_x) \cdot \mathrm{I}'_{\theta x}, \tag{2.18}$$

and the corresponding desired output D can be obtained by

$$D = \mathrm{I}(i, j) . \tag{2.19}$$

It should be noted that the weighting vectors need to be updated in the training stage are only $\overline{U}$ and $\overline{V}$. If a reference pixel $\mathrm{I}'(i, j)$ is given, the neighborhood pixel vector $\overline{\mathrm{I}}'_\theta$ of $\mathrm{I}'(i, j)$ can be regarded as an extra input vector for compensation. This unique operating rule is the major difference between the proposed neural network and the common feed-forward neural networks and is specially designed for the image-compensation application.

In the training stage, the updating rules of weights, $v_{ab} \in \overline{V}$, $u_b \in \overline{U}$, can be derived by the back-propagation learning method as

$$v_{ab}(t+1) = v_{ab}(t) + \eta \ (D-Y)[\mathrm{I}'_{\theta a}(1+y(s_a))(1-y(s_a))/2] \times g(Z_b), \tag{2. 20}$$

$$u_b(t+1) = u_b(t) + \eta \left\{ \sum_{i=1}^{16} [(D-Y)(\mathrm{I}'_{\theta i}) \frac{(1+y(s_i))(1-y(s_i))}{2} v_{ib}] \right\} \times \left[(1+g(Z_b))(1-g(Z_b))/2\right] IP, \tag{2.21}$$

where η is the learning constant which determines the rate of learning.

Thirty nature images were used to train the proposed NN for image compensation. The edge regions in these training images are separated into 8 different quantized angles. The variations may be caused by the quantization error (11.25°) and the characteristics of different images and regions. In addition, the vector between the third and the fourth layers of the neural network for image quality enhancement represents the sixteen neighborhood pixels of the reference pixel and it is the filtered results of the first stage (noise removal). This will also cause the variation and nonlinearity in the training. In order to reduce the cost function (MSE) to 1% of the intensity range, i.e. $255 \times 0.01 \cong 2.5$, 200 nodes in the first hidden layer were required to achieve this goal with the learning rate η = 0.2 in our experiments. However, if we release the goal (MSE) to achieve from 2.5 to 5, the hidden nodes in the first hidden layer can be reduced to 80 without affecting the visual quality heavily. When the training process is finished, 8 different input values, θ, can be inputted to the trained network, and the corresponding weighting matrices W_θ can be obtained to build a look up table combined with Eq. (2.4) for image compensation to reduce the computational cost.

Table 2.1 shows the obtained weighting matrices for the 8 orientations (W1~W8). *W*1 and *W*3 look the rotated versions of *W*5 and *W*7, respectively. *W*2, *W*4, *W*6 and *W*8 look the rotated version of each other. It means the proposed neural network can catch the directional characteristics of edges through automatic training.

Table 2.1 The eight weighting matrices obtained from the trained neural network for image quality enhancement.

W1 (0°)	W2 (22.5°)
$\begin{bmatrix} 0.00270896 & 0.00069366 & 0.00056130 & 0.00004264 \\ 0.25111893 & 0.24986401 & 0.24900746 & 0.25007752 \\ 0.00004062 & -0.00081938 & -0.00078190 & 0.00184474 \\ 0.00060843 & 0.00009103 & 0.00104945 & 0.00179080 \end{bmatrix}$	$\begin{bmatrix} 0.03263883 & -0.12737456 & 0.3393475 & 0.20418046 \\ 0.11473297 & 0.24242122 & 0.23237932 & 0.12049203 \\ 0.19398496 & 0.06377091 & -0.03941444 & -0.04396327 \\ -0.05561969 & -0.01565117 & 0.07172672 & -0.01947648 \end{bmatrix}$
W3 (45°)	W4 (67.5°)
$\begin{bmatrix} 0.01068067 & -0.03348986 & 0.17198606 & 0.17003769 \\ -0.0.154362 & 0.16861688 & 0.16765814 & -0.05673975 \\ 0.16927022 & 0.16688901 & -0.05967028 & 0.02869632 \\ 0.17292993 & -0.05580725 & 0.02140568 & -0.00070582 \end{bmatrix}$	$\begin{bmatrix} 0.03404918 & 0.11399747 & 0.19406541 & -0.05537197 \\ -0.12803707 & 0.24180996 & 0.06267749 & -0.01620973 \\ 0.03393999 & 0.23072700 & -0.04103377 & 0.07309391 \\ 0.20509279 & 0.12011293 & -0.04451190 & -0.01901695 \end{bmatrix}$
W5 (90°)	W6 (112.5°)
$\begin{bmatrix} 0.0409030 & 0.2525427052 & 0.00225166 & 0.00410132 \\ 0.00008099 & 0.25283754 & 0.00131791 & 0.00018051 \\ 0.00183930 & 0.24998878 & 0.00158754 & 0.00040739 \\ 0.00171057 & 0.25100192 & -0.00097231 & 0.00257733 \end{bmatrix}$	$\begin{bmatrix} 0.20460927 & 0.11994475 & -0.04483605 & -0.02036048 \\ 0.03346962 & 0.23192659 & -0.04032709 & 0.07141998 \\ -0.12803423 & 0.24167401 & 0.06352055 & -0.01579252 \\ 0.03281431 & 0.11399426 & 0.19355151 & -0.05560378 \end{bmatrix}$
W7 (135°)	W8 (147.5°)
$\begin{bmatrix} 0.25051364 & 0.00072645 & 0.00055420 & 0.00050129 \\ 0.00074467 & 0.25005892 & 0.00000905 & 0.00056688 \\ 0.00043473 & 0.00041066 & 0.25014690 & 0.00108892 \\ 0.00058684 & 0.00054735 & 0.00049473 & 0.25065422 \end{bmatrix}$	$\begin{bmatrix} 0.20489226 & 0.03368954 & -0.12767382 & 0.03285340 \\ 0.12006646 & 0.23204653 & 0.24146707 & 0.11438927 \\ -0.04462188 & -0.04002404 & 0.06342979 & 0.19427589 \\ -0.0200380 & 0.07150226 & -0.01556357 & -0.05482485 \end{bmatrix}$

2.4. Brief Summary

In this chapter, we proposed an HVS-directed neural-network-based image quality enhancement. A fuzzy decision system inspired by the HVS is proposed to classify the input image into human perception sensitive regions and non-sensitive regions. If a pixel is in the perception sensitive region, the proposed neural-network module is applied to this pixel for further compensation. The proposed image quality enhancement system can be combined with any other filtering approach to enhance the visual quality. It can also provide a quite stable performance over a wide variety of images with various noise corrupted probabilities will be discussed in the latter chapter.

3. Image Impulse Noise Removal

In this chapter, a novel noise removal algorithm to deal with impulse noise is proposed. Firstly, the decision-based recursive adaptive median filter is applied to remove the Salt-Pepper noise and an adaptive two-level neural network noise reduction procedure is applied to remove the random-valued noise, and keep the uncorrupted information well. Then the fuzzy decision rules inspired by human visual system (HVS) described in Chapter 2 is applied to compensate the blur of the edge and the destruction caused by median filter. According to the experiment results, the proposed method is superior to conventional methods in perceptual image quality as well as the clarity and smoothness in edge regions.

3.1. Introduction

As mentioned in Sections 1.3, the common median-type methods suffer from the trade-off between cleanness of noise removal and preserving of edge sharpness. They will suffer from the side effect of median filter such as blur and edge destruction especially for images are highly corrupted. These methods exhibit relatively better performance but require more computation and memory cost. It is desired to improve the quality of noise removal and reduce the time consumption at the same time.

In this chapter, the decision-based recursive adaptive median filter to remove the Salt-Pepper noise is described in Section 3.2, and a novel framework for random-valued noise removal is described in Section 3.3. Since the precise noise detection procedure is the key point in noise reduction, we propose a neural network for noise detection such that various widespread densities of noisy pixels can be distinguished from the detail edge pixels well. After suppressing the impulse noise, the image quality enhancement system is applied to compensate the corrupted pixels described in Section 3.4. If a noise-corrupted pixel is in the sensitive region, the proposed neural-network module is applied to this pixel for further compensation. Section 3.5 shows the experimental results with various comparisons to other schemes. Finally, the conclusions are summarized in Section 3.6.

3.2. Salt-Pepper Impulse Noise Removal

3.2.1. System Architecture

In order to remove the noisy images without blurring the edge, we divide the process into two steps as shown in Fig. 3.1. In the first step, the impulse noise is removed cleanly without losing too much detail information and then image quality enhancement is applied to compensate the edge sharpness in the second step.

Fig. 3.2 shows the process in the first step, called the decision-based recursive adaptive median filtering scheme. The proposed system consists of noise range detection, decision-based rules, and the adaptive median filter with two different window sizes excluding from noisy pixels, and the recursive algorithm. The noise range estimation and decision-based rules decide whether the pixel is possibly corrupted by noise or not, and the median filter is applied to the possibly noise-corrupted pixels only. The window size of median filter is chosen depending on

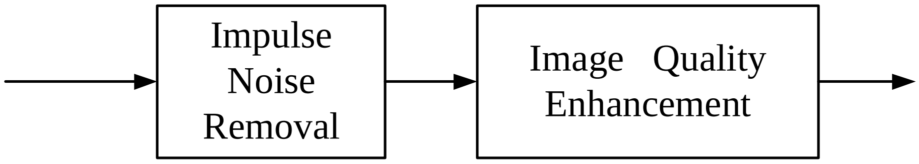


Fig. 3.1. The proposed two-step impulse noise removal algorithm.

how heavily the neighboring pixels are corrupted by noise. Then we can remove the noise entirely without destroying the detail and edges of the image. The recursive algorithm replaces the value of noisy pixel with the processing output of the adaptive median filter before the window shifts to next possible noisy pixel so that we can remove the noise more cleanly when the window shifts to the following positions.

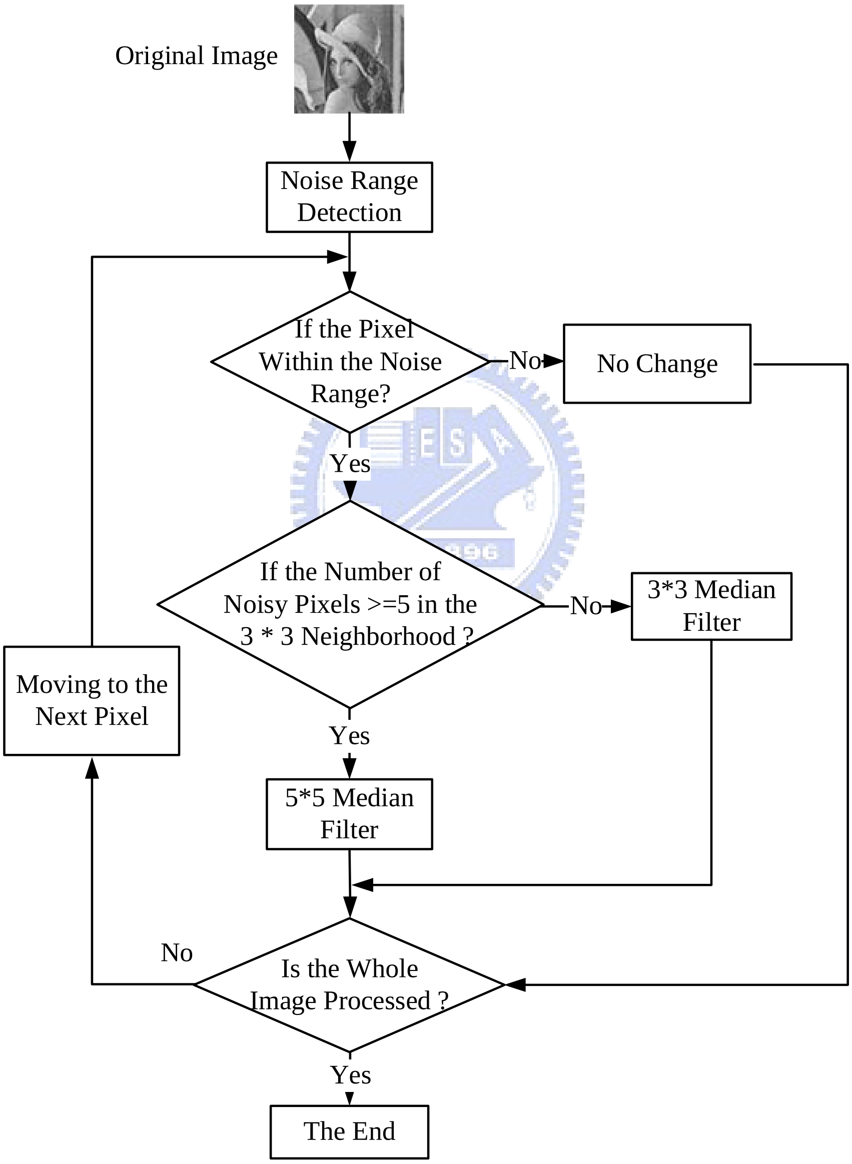

Original Image
Noise Range Detection
If the Pixel Within the Noise Range?
No
No Change
Yes
If the Number of Noisy Pixels >=5 in the 3 * 3 Neighborhood ?
No
3*3 Median Filter
Moving to the Next Pixel
Yes
5*5 Median Filter
No
Is the Whole Image Processed ?
Yes
The End

Fig. 3.2. The decision-based adaptive recursive median filtering scheme.

3.2.2. The Strategy of Salt-Pepper Noise Removal

In the first step, our goal is to totally remove the Salt-Pepper noises and keeping the edge and detailed information as much as possible so that we can well compensate the corrupted pixels by using the relationship of the neighborhood in the second step. The techniques used in the first step are introduced as follows.

3.2.2.1. Noise Range Estimation

The noise model of additive Salt-Pepper impulse noise, as mentioned in Section 1.2, is a popular and practical problem in image processing. Since the noise range is random assigned, estimating the distribution range of noise is the first task. As the histogram statistics of nature images are distributed uniformly, we can observe that the boundary between the values of noise and the normal pixels are abrupt as shown in Fig. 3.3. Therefore, we apply the histogram statistics to find the gap thresholds and estimate the noise ranges near the maximal and minimal gray level adaptively. It is calculated by

$$ThL = \max\left\{\ His(i) - His(i+1)\ \middle|\ 0 \le i \le 30\right\}, \tag{3.1}$$

$$ThH = \max\left\{\ His(i+1) - His(i)\ \middle|\ 225 \le i \le 255\right\}, \tag{3.2}$$

where $His(i)$ is the histogram statistic of noisy image at gray value i , *ThH* and *ThL* are the maximum upper and lower gap thresholds, respectively. If the gray level of the pixel is within the noise range, smaller than *ThL* or greater than *ThH*, we classify it as a suspected noisy pixel and execute the noise removal process in Fig. 3.2. The initial values of *ThH* and *ThL* are 256 and –1 respectively, to deal with the case that only maximum or minimum threshold is existed such as the white noise that only maximum noisy pixels are found.

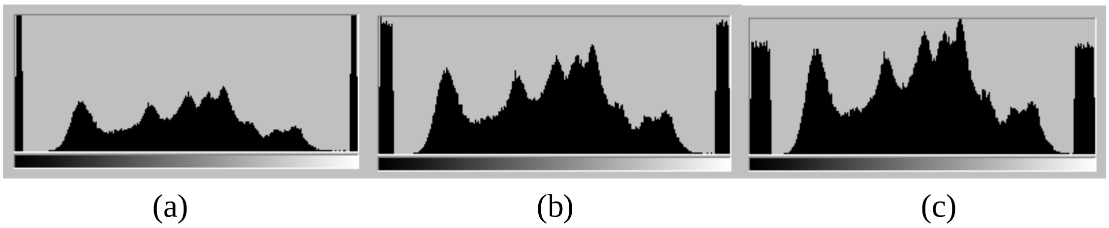


Fig. 3.3. Histogram statistics of Lena with 20% impulse noise within different noise range. (a) range = 5 [(0−4); (251−255)]; (b) range =10 [(0−9); (246−255)]; (c) range = 15 [(0−14); (241−255)].

3.2.2.2. Decision Rules

It is very time consuming and even worse effective to filter all pixels in an image. As a matter of fact, most pixels in the image are uncorrupted and should not be modified. In the detection step, each pixel is classified into the possible noisy pixel or the uncorrupted pixel. The uncorrupted pixels are retained without any modification to avoid blurring caused by unnecessary processing.

In our strategy, the pixels within the detected noise range are regarded as possibly corrupted noisy pixels and processed by the adaptive median filter. In such decision rule, noisy pixels will be completely detected but the original pixels in these intervals will also be misidentified as noisy pixels. Although some pixels are false detected, our algorithm will not destroy the image quality too much. As the false detections of noise are in the smooth regions, gray values of nearby pixels are very close, so the result of median filtering is also very close to the original pixel value. Even if the pixels are false detected in the edge area, our second step − image quality enhancement system will well compensate the jaggy edges.

3.2.2.3. The Noise-Exclusive Adaptive Median Filtering

When the median filter is employed to image processing, it often causes edge blurring with large window sizes (such as 5×5) and suffers in insufficient noise suppression with small window sizes (such as 3×3). Although an image may be degraded with up to 20 or 30 percent impulse noise, the random excited pixels make the distribution of noise uneven. When images are highly corrupted, a large number of noisy pixels may connect into noise blotches, and it will still retain the noise after the processing of median filter with 3×3 window size.

In this chapter, an adaptive median filter is proposed to achieve superior performance of noise suppression and to preserve sufficient image sharpness and detailed information. We first analyze the neighboring 3×3 region of the possible noisy pixel. If there are more than four other possible noisy pixels in this block, it is identified as a highly corrupted region and the 5×5 median filter is applied for processing. Otherwise, the noisy pixel is processed by the 3×3 median filter. The noise-exclusive scheme allows only the clean pixels inside the window to participate in median processing [11].

3.2.2.4. Recursive Method

The recursive algorithm replaces the value of noisy pixel with the processing output before the 3×3 window shifts to the next noisy pixel. It can efficiently reduce the number of noisy pixels and remove the noise more cleanly when the window shifts to the following processing positions. Fig. 3.4 shows the difference between recursive and non-recursive methods. In Fig. 3.4, $x_i(n)$ represent original pixels and $y_i(n)$ represent processed pixels. We can find almost half of pixels in each block have been processed by the recursive method (as shown in Fig. 3.4(b)) such that the number of pixels contaminated by the noise is reduced and the noise can be easily removed by the 3×3 median filter. The drawback of the recursive algorithm is that it

$x_1(n)$	$x_2(n)$	$x_3(n)$
$x_4(n)$	$x(n)$	$x_5(n)$
$x_6(n)$	$x_7(n)$	$x_8(n)$

$y_1(n)$	$y_2(n)$	$y_3(n)$
$y_4(n)$	$x(n)$	$x_5(n)$
$x_6(n)$	$x_7(n)$	$x_8(n)$

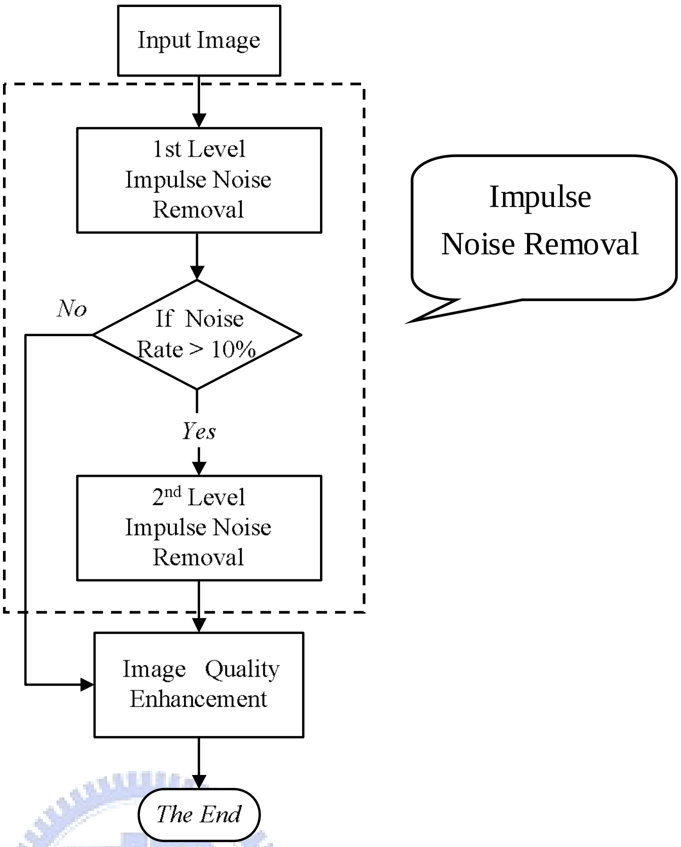
Fig. 3.4. Elements of the 3×3 window centered around $x(n)$. (a) Non-recursive; (b) recursive. $x_i(n)$ represent original pixels and $y_i(n)$ represent processed pixels.

will blur the edge and detailed information at the same time. Therefore, the second step of the proposed method − image quality enhancement is necessary for compensation.

3.3. Random-Valued Impulse Noise Removal

3.3.1. System Architecture

Cleaning random-valued noise is far more difficult than cleaning Salt-Pepper noise, since the differences in gray levels between a noisy pixel and its noise-free neighbors are significant most of the times. In order to remove the random-valued noise cleanly from input images without blurring the edge, we divide the process into two steps − impulse noise removal and image quality enhancement as shown in Fig. 3.5. In the first step, the impulse noise is removed cleanly without losing too much detail information and then the image quality enhancement is applied to compensate the edge sharpness in the second step. The first step, two-level neural network noise removal procedures are shown in Figs. 3.6 and 3.7. Fig. 3.6 shows the procedure diagram of our first-level noise removal process and Fig. 3.7 shows the second-level noise removal procedure. Inside the first level, only the noisy pixels identified by the neural network detection are processed with the 3×3 median filter.



Input Image
1st Level Impulse Noise Removal
If Noise Rate > 10%
No
Yes
2nd Level Impulse Noise Removal
Image Quality Enhancement
The End
Impulse Noise Removal

Fig. 3.5. The proposed scheme for impulse noise removal.

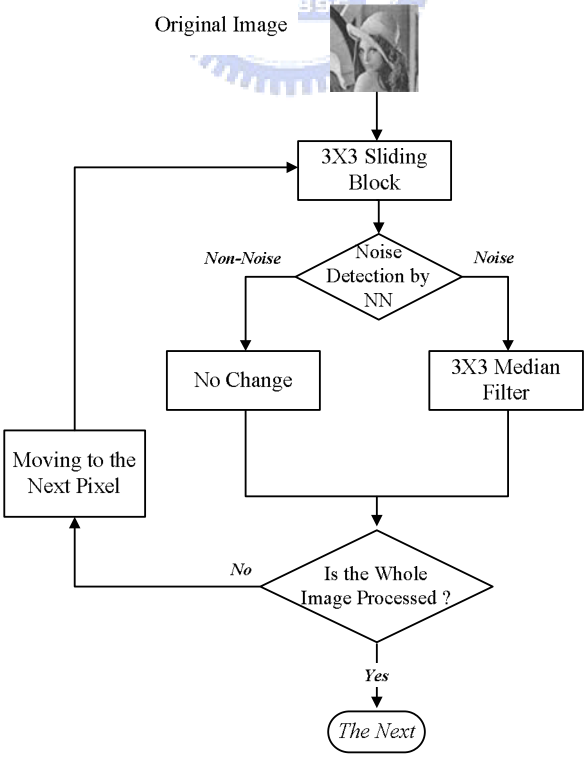


Fig. 3.6. The procedure diagram of the first-level impulse noise removal.

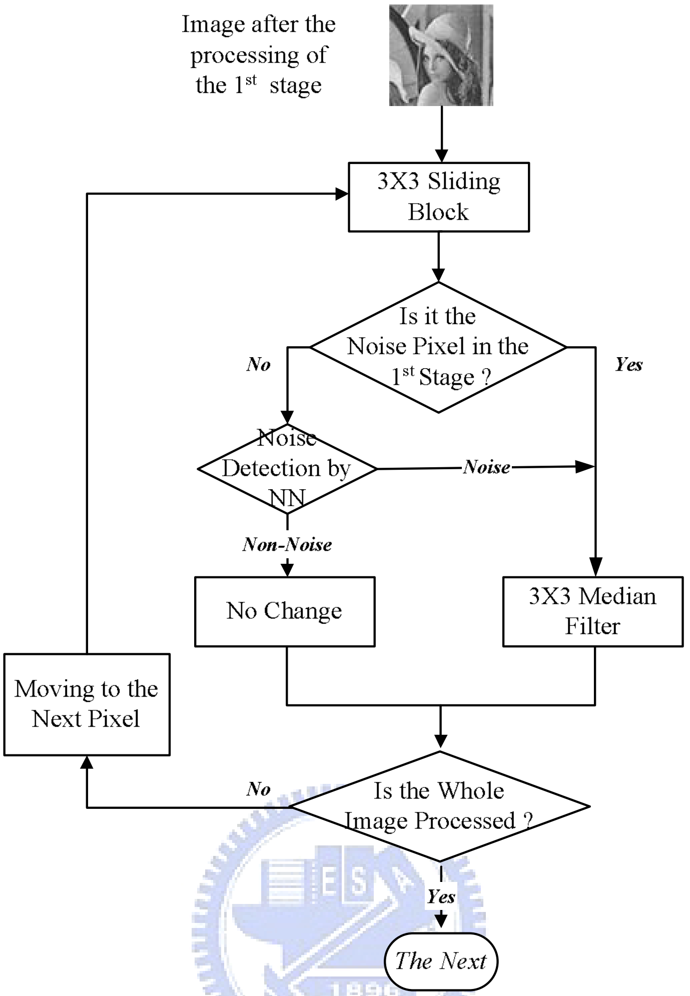


Fig. 3.7. The procedure diagram of the second-level impulse noise removal.

The second-level noise removal procedure as shown in Fig. 3.7 is used to detect and remove the misclassified and the detected but un-removed noise pixels in the first-level noise removal process. The 3×3 window is applied in this stage to obtain the features correspond to the pixel $P(0,0)$ for noise detection.

3.3.2. The Strategy of Random-Valued Noise Removal

Since the residual noise will strongly affect human perception, precise noise detection is the first important step for the noise removal. The noise is much more annoying the human perception in the smooth and edge areas that have the lower just noticeable distortion (JND) values compared with JND in the texture area [56]−[59].

Most algorithms work well on low noise-density images but fail to detect noise pixels in the edge region.

The decision-based algorithms for noise detection can be divided into three types. The first type is to detect whether the pixel is contaminated by noise according to the local features. Florencio *et al.* [10] proposed a decision measure based on second order statistic called normalized deviation to detect the noise by threshold. Zhang *et al.* [17] proposed a detection technique by four convolutions using one-dimensional Laplacian operator. The second type decision measure considers the differences of adjacent pixel values in the rank-ordered median filtering sequence [12]−[13]. The third type approach called switching-schemes [14]−[15] that firstly apply several types of rank-ordered filters and then detect the noise pixels by their relationships with the gray level of the origin pixel.

In this chapter, a neural network with high precision and capability of dealing with images corrupted with various noise densities is proposed for noise detection. It is a 3-layer neural network with one hidden layer as shown in Fig. 3.8. The input layer consists of three nodes corresponding to Gray level Difference (GD), Average Background Difference (ABD) and Accumulation Complexity Difference (ACD) in the 3×3 sliding window. The second layer is the hidden layer that consists of 6

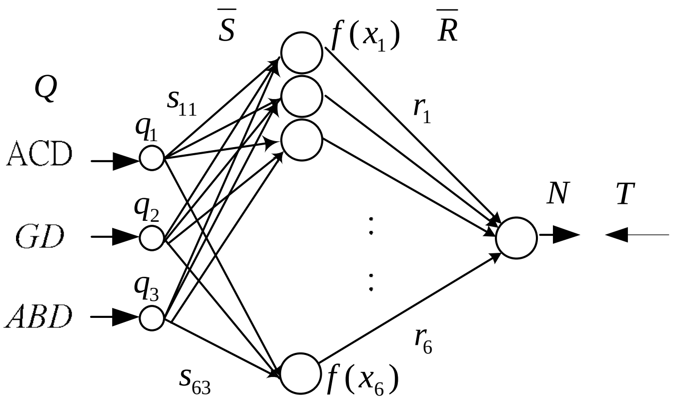


Fig. 3.8. The proposed neural network for noise detection.

nodes and the bipolar sigmoid function is used as the activation function. The weighting vectors between the first and the second layer, and between the second layer and the third layer are denoted as $\overline{S}$ and $\overline{R}$, respectively. The output layer includes one node which represents the identified attribution of the pixel: “noise” or “non-noise”, and the bipolar sigmoid function is also used as the activation function. The three features in the input layer are discussed as follows.

3.3.2.1. Gray level Difference (GD)

The gray level difference represents the accumulated variations between the central pixel for identification and each surrounding local pixel. It is defined by

$$GD = \sum_{i=-1}^{1} \sum_{\substack{j=-1 \\ (i,j)\neq(0,0)}}^{1} \left| P(0,0) - P(i,j) \right|, \tag{3.3}$$

where $P(0,0)$ is the reference pixel and $P(i,j)$ is the surrounding local pixel.

The feature GD is mainly considered to detect the noise in flat area. It is expected that the corrupted pixels should yield much bigger differences as compared to the uncorrupted pixels. However, the pixels in edge and texture areas will also get high GD values so that the obscure region identification between the noise, edge and texture pixels relies on the other two assistant features ABD and ACD.

3.3.2.2. Average Background Difference (ABD)

Averaging the surrounding pixels as the background luminance of the sliding block to compare with the central pixel is another assistant feature to detect the noise. This feature, called average background difference (ABD), representing the overall average variation with the central pixel in the block is defined by

$$ABD = \left| P(0,0) - \frac{\sum\limits_{i=-1}^{1} \sum\limits_{\substack{j=-1 \\ (i,j)\neq(0,0)}}^{1} P(i,j)}{8} \right| . \tag{3.4}$$

The corrupted pixels will yield bigger differences as compared with the clean ones. For the pixels in the texture area, the GD value is large but the ABD feature will be small.

3.3.2.3. Accumulation Complexity Difference (ACD)

Accumulating the difference between each pixel in the 3×3 sliding block and its four neighboring pixels as defined below shows the structure information of the block:

$$ACD = \sum_{i=-1}^{1} \sum_{j=-1}^{1} \left| 4 \times P(i,j) - P(i-1,j) - P(i+1,j) - P(i,j-1) - P(i,j+1) \right| \cdot \tag{3.5}$$

In edge area, the summation is lower than the noise-pixel area though the gray level difference (GD) might be similar. So it provides an assistant feature between the edge and noise pixels.

In order to train the proposed neural network for noise detection, the 512×512 of gray-scale Lena image with 20% of impulse noise generated uniformly within [0,255] is used as a reference pattern for training, and 3000 noisy pixels and 3000 uncorrupted pixels uniformly distributed in the image are adopted as the training data. We also establish a noise table corresponding to these 6000 training data as the desired output for supervised training. The desired output for noise pixels is 1, and the desired output for the clean pixels is -1. The goal is to reduce the mean square error to 0.1. The back-propagation learning method is used to derive the updating rules of weights. In our experiments, 6 nodes in the hidden layer are enough to achieve this goal and the learning rate was 0.1. Experimental results show that our neural network owns the highest detection precision than other compared methods and our detection procedure also gives a better trade-off between the undetection and misdetection rates.

3.4. The Second Step – Image Quality Enhancement System

Since the process of the first step has removed the visible noise as cleanly as possible, the second step focuses the image enhancement on edge region. In order to compensate the edge sharpness, image quality enhancement mentioned in Chapter 2 is applied to the modified pixels. The second step, the schematic block diagram of the proposed image quality enhancement system, is shown in Fig. 3.9. The proposed system consists of a fuzzy decision module, an angle evaluation module, and an adaptive neural-network compensation module. Fuzzy decision module designates each 4×4 sliding block it receives for one of a plurality of predefined classifications. Based on this classification, either the filtered value of the first step or the result of adaptive neural-network compensation module is selected to reconstruct each possible noise corrupted pixel. When the adaptive neural-network compensation is actuated, the angle evaluation module will compute the dominant orientation of the original image located in the sliding block as the input data of the proposed neural network.

When an image has been processed in the first step and enters the second step, it is firstly divided into 4×4 sliding (overlapping) blocks. The block is shown in Fig. 2.6 as an illustration, where $O(0,0)$ is defined as the reference pixel that possibly corrupted by noise at (*m*,*n*) and $O(i, j)\big|_{i\neq 0, j\neq 0}$ are the neighborhoods of $O(0,0)$. The weighted compensation is applied to the visual-sensitive region and can be presented as:

$$\mathrm{F}(m,n)=\sum_{i=-1}^{2}\sum_{j=-1}^{2}O(i,j)\;W_{\theta}(i,j)\,, \tag{3.6}$$

where W_θ is derived from a neural network after off-line training. The neural network is trained according to the edge angle of the reference image pixel to obtain the corresponding weights. After training, weighting matrices W_θ are obtained for testing. In our system, only the filtered pixels in the first step are compensated in the second step.

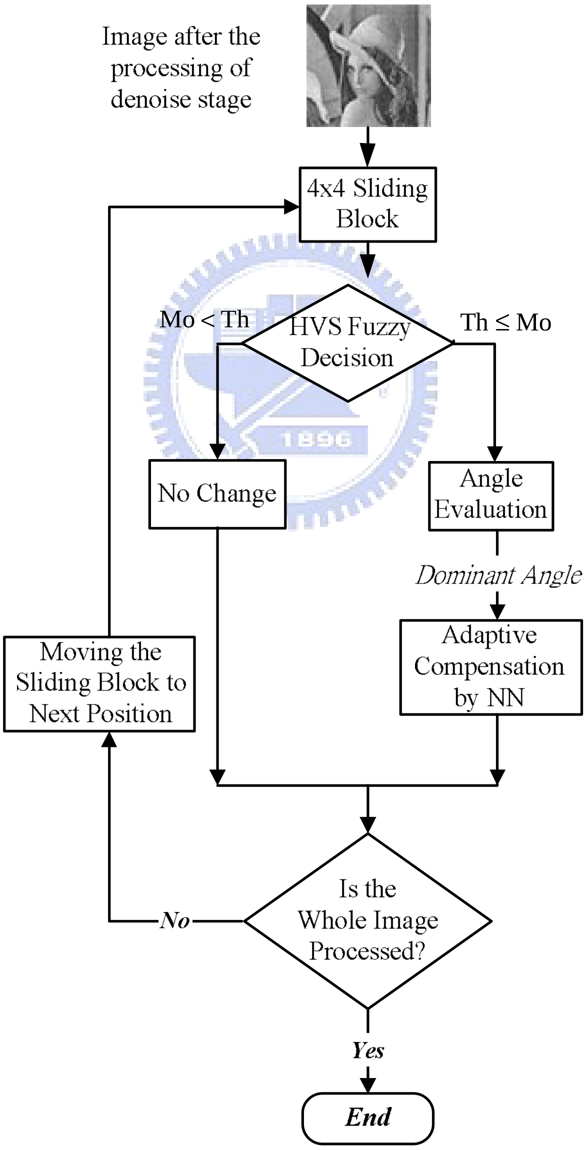


Image after the processing of denoise stage
4x4 Sliding Block
Mo < Th
HVS Fuzzy Decision
Th ≤ Mo
No Change
Angle Evaluation
Dominant Angle
Adaptive Compensation by NN
Moving the Sliding Block to Next Position
No
Is the Whole Image Processed?
Yes
End

Fig. 3.9. Schematic block diagram of the proposed image quality enhancement system.

3.5. Experiment Result

3.5.1. Simulation Results of Salt-Pepper Noise Removal

The performance of the proposed method has been examined on a variety of testing images corrupted with various noise densities. For quantitative performance evaluation, the peak signal-to-noise ratio (PSNR) defined as:

$$PSNR \;=\; 10\log_{10}\left(\frac{255^2 \times N}{\sum_{n=1}^{N}(\hat{z}_n - z_n)^2}\right) \tag{3.7}$$

is used as quantitative performance indication, where z is the gray level of original pixel; $\hat{z}$ is the gray level of the final processed pixel and N is the total number of pixels in an image.

In our experiments, the proposed algorithm is compared with five existing methods including median filter [1], recursive median filter [2], Center Weighted Median (CWM) Filter [3], Tri-State Median Filter [14], and Li's [24] method. Table 3.1 shows the quantitative comparisons of these different methods. It can be found that the proposed method can achieve best PSNR performance compared with the other methods.

The testing results of Lena with 20% impulse noise are shown in Fig. 3.10. According to Fig. 3.10(c) and (d), we can find the recursive median filter removes the noise well but also blurs the edge, so the recursive algorithm cannot balance the noise removal and edge sharpness well. In Fig. 3.10(e) and (f), the Tri-State Median can retain more edge sharpness than the CWM, but both of them cannot remove the noise very well in some highly noise-corrupted area. In Fig. 3.10(g), Li's method might misjudge some noisy pixels as the edge and then increase the size of some noises. The threshold adjustment for different images is another problem that needs to be solved in these methods. In Fig. 3.11, the peppers images with 40% highly corrupted impulse noise are processed. It shows that the proposed method can effectively remove the noise and keep the edge sharpness well even in highly noise-corrupted conditions. Fig. 3.12 is the experimental results of the Boat image with 30% salt-pepper noise obtain from a public data base [63] to demonstrate the performance of the proposed method. The reason why the proposed first-stage processing is so powerful in removing the impulse noise is that we take the advantage of adaptive-length median filter well especially for the images with high-density noise. In a high noise density region, the 5×5 median filter is applied to exhibit a high degree of noise suppression ability. Otherwise, the noisy pixel is processed by the 3×3 median filter to preserve the sharpness. When images are highly corrupted by the noise, a large number of noisy pixels may connect into noise blotches and they cannot be removed well just by the processing of 3×3 median filter. In addition, the recursive and noise-exclusive scheme that allows only uncorrupted pixels inside the window to participate in median processing are also involved in the first stage to improve the performance of noise removal. According to the experimental results, it can be found that the one stage median type filtering wants to preserve the sharpness of the edges but leads to remain more un-removed noise pixels.

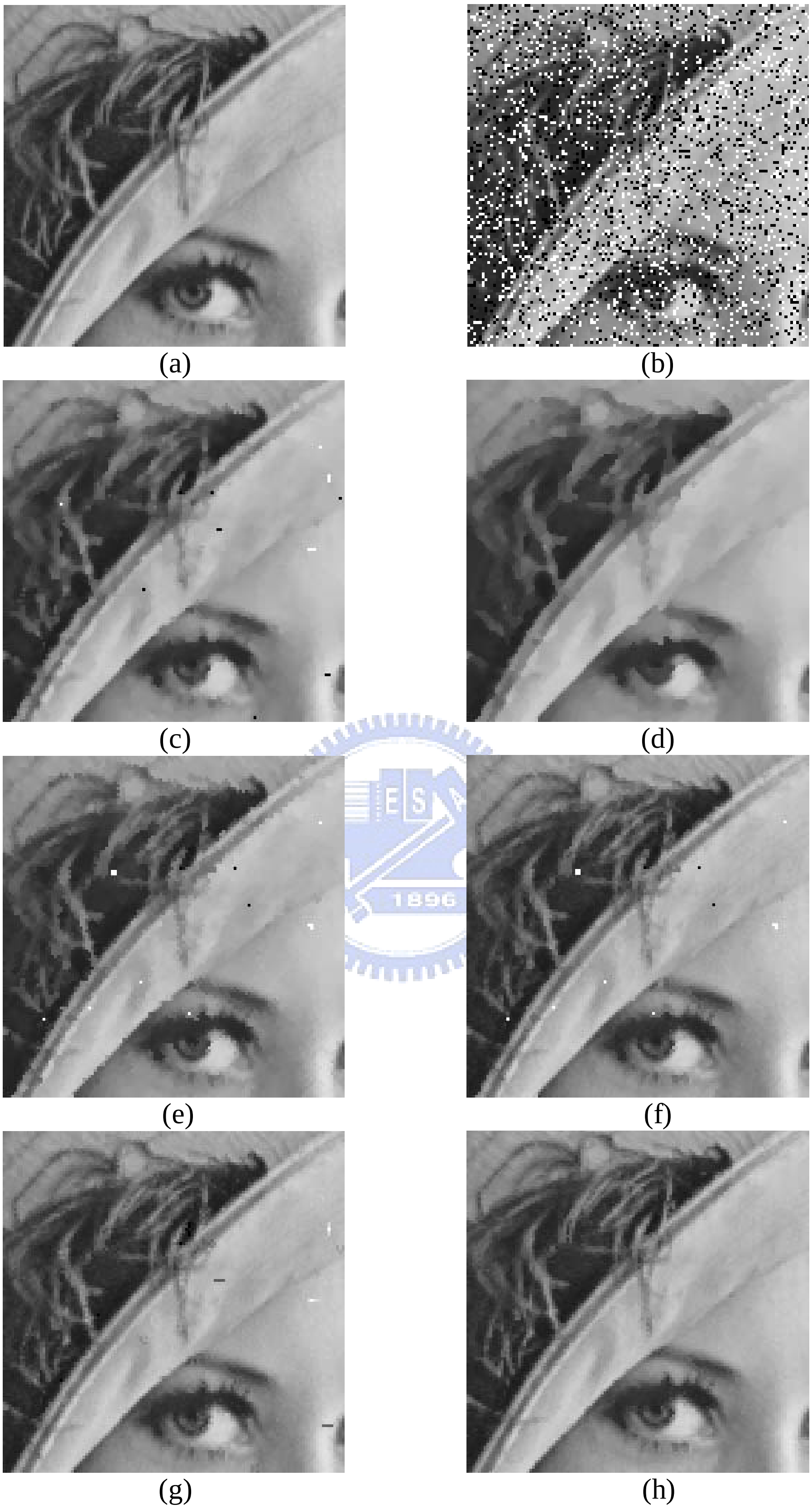
Fig. 3.13 and Table 3.2 show the essential importance of the image quality enhancement stage to compensate the blur and jaggy edge caused by the median filter and recursive algorithm. As the images are more highly corrupted, the large window size median filter (such as 5×5) is used more often to totally remove the noises. The image quality is further improved both in visual perception and quantity of PSNR after quality enhancement in the second step. Noise detection accuracy of the proposed method is shown in Table 3.3. Three detection measures are defined as follows: “Total Correct Classification” [64] means the noisy pixels and uncorrupted pixels are correctly identified. “Un-detection” [22] means noise pixels are not identified that leads to residual noise. “Mis-detection” [22] means clean pixels are misidentified such that unnecessary filtering operation causes image blurring. Noisy pixels will be completely detected but few original pixels within the noise range will also be misidentified as noisy pixels. The median filtering result of the misdetected pixel is close to its original value and will not destroy the image quality too much. The stable performances of experiment results verify our inference well.

Fig. 3.14 shows the experimental result of applying the proposed method to white noise that only maximum noisy pixels are found. The experiment result shows that the proposed method can also deal with the case that only maximum or minimum noisy pixels are existed pretty well.

Fig. 3.15 shows the performance comparisons of different noise removal methods with/without the proposed noise range assumption applying to “Lena” corrupted with 40% impulse noise. Owing to only the detected noise pixels are filtered, they may prevent the blurring caused by filtering the uncorrupted pixels but still can’t remove the noise well. When images are highly corrupted, the noisy pixels may connect into noise blotches and they cannot be removed just by the processing of 3×3 median filter.

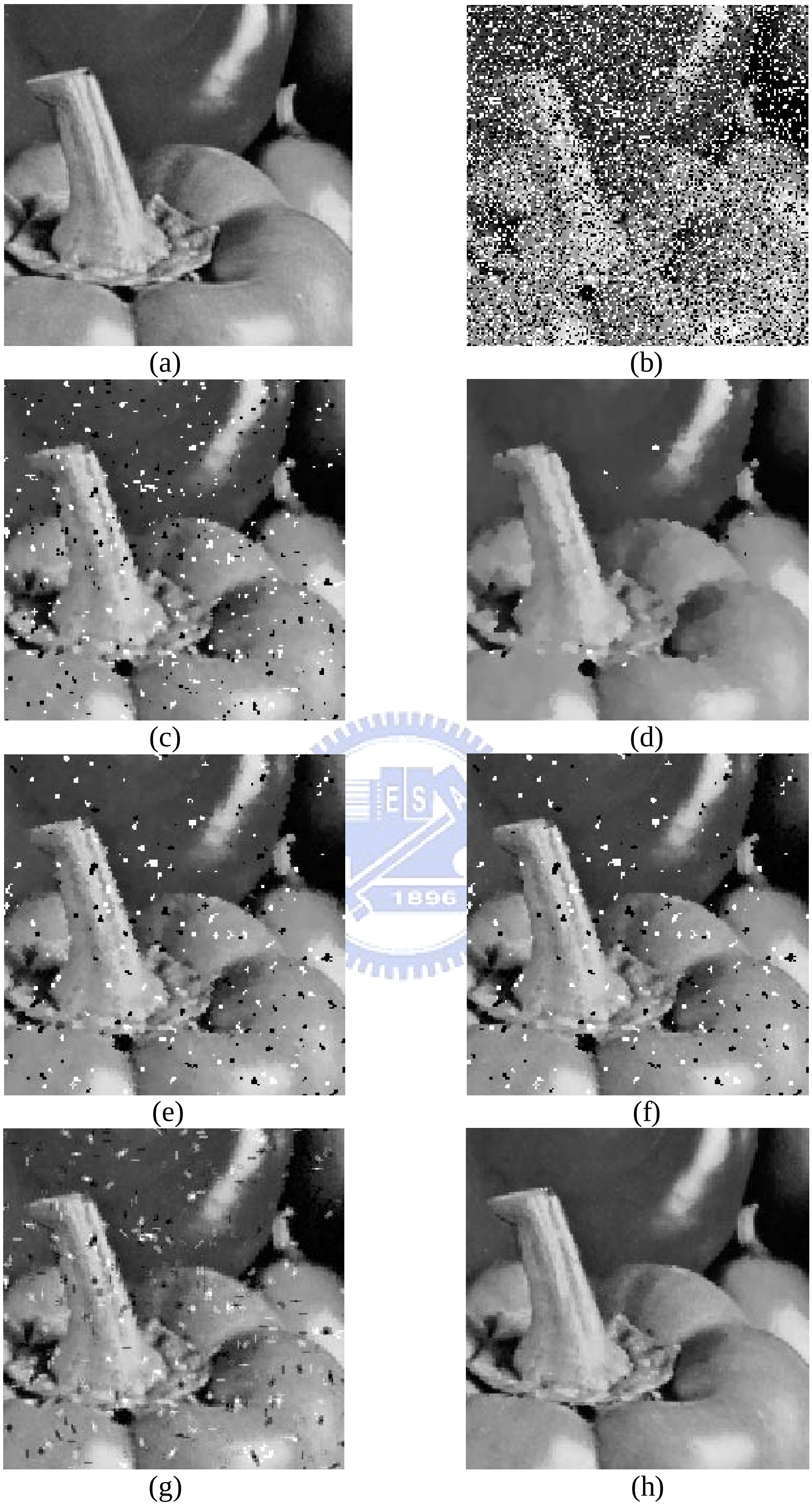
We also used the direct matrix inversion method to obtain the weighting matrices for the second-step processing. According to Fig. 3.16, we can find the second step processing that uses the weighting matrices obtained by matrix-based estimation method can also improve the compensation performance compared with the results of applying 1st step processing only. However, the neural network method still achieves better performance due to the variations of the training images and the selected training portions. The edge regions in the training images are separated into 8 different quantized angles. The variations may be caused by the quantization error (11.25°) and the characteristics of different images and regions. In addition, using nonlinearly filtered (median filters) results of the first stage (noise removal) to estimate the original clean pixels will also cause the variation and nonlinearity in the training.

The hardware and software environment that we implement the algorithms for speed comparisons are described as follows: all the algorithms are implemented in Matlab Language on a 1.8G Hz Pentium Ⅳ-based PC with 256 Mb RAM. Table 3.4 shows the average computation time in second for various algorithms applied to different kinds of noise corrupted images. The time consuming of the proposed algorithm is quite reasonable compared with other methods.



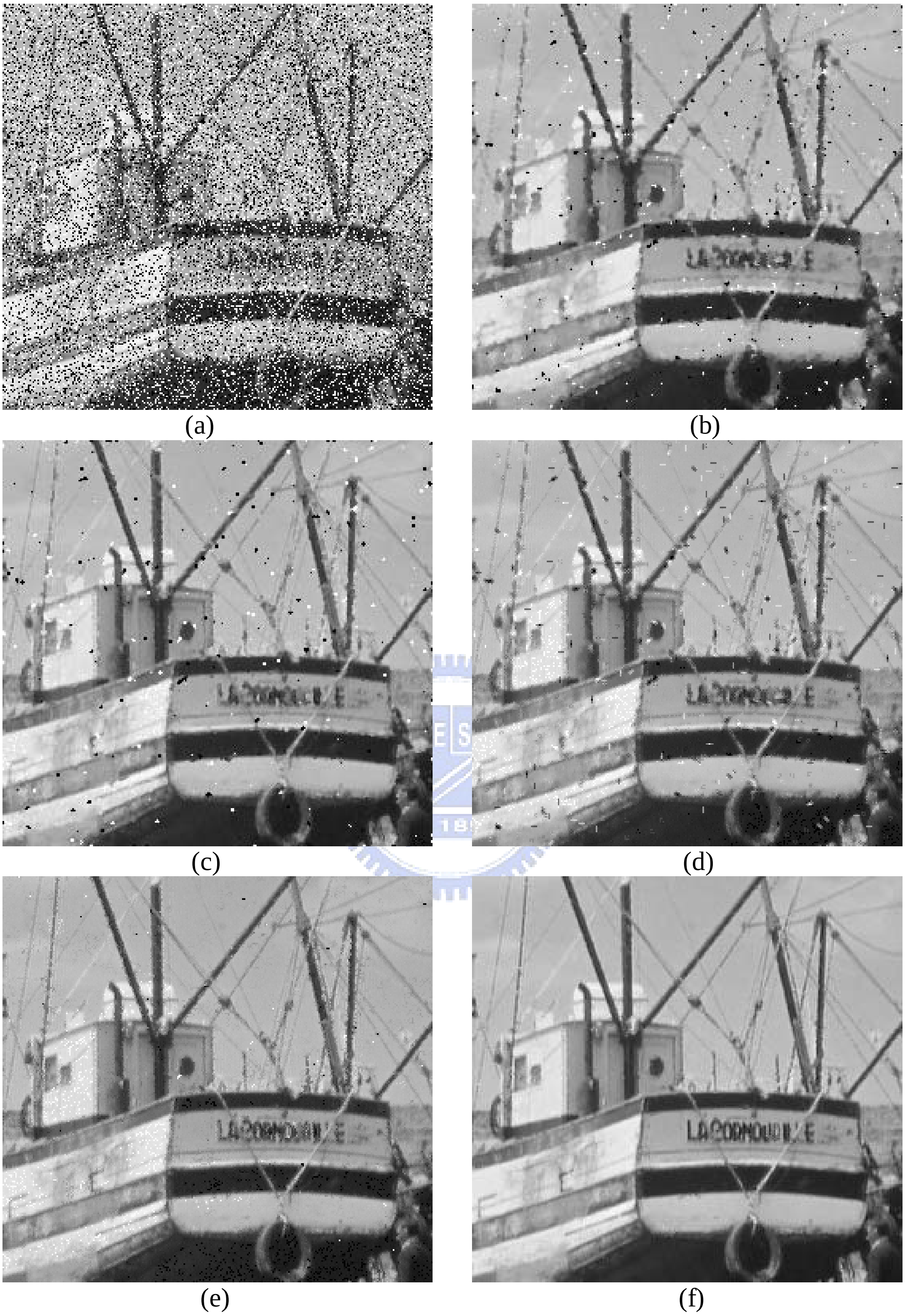
(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 3.10. (a) Original image; (b) Lena with 20% of impulse noise; (c) the 3×3 standard median filter; (d) the 3×3 recursive standard median filter; (e) the recursive CWM filter with weight = 3; (f) the recursive Tri-state median filter with threshold = 25; (g) Li's method with threshold = 32; (h) our proposed method.



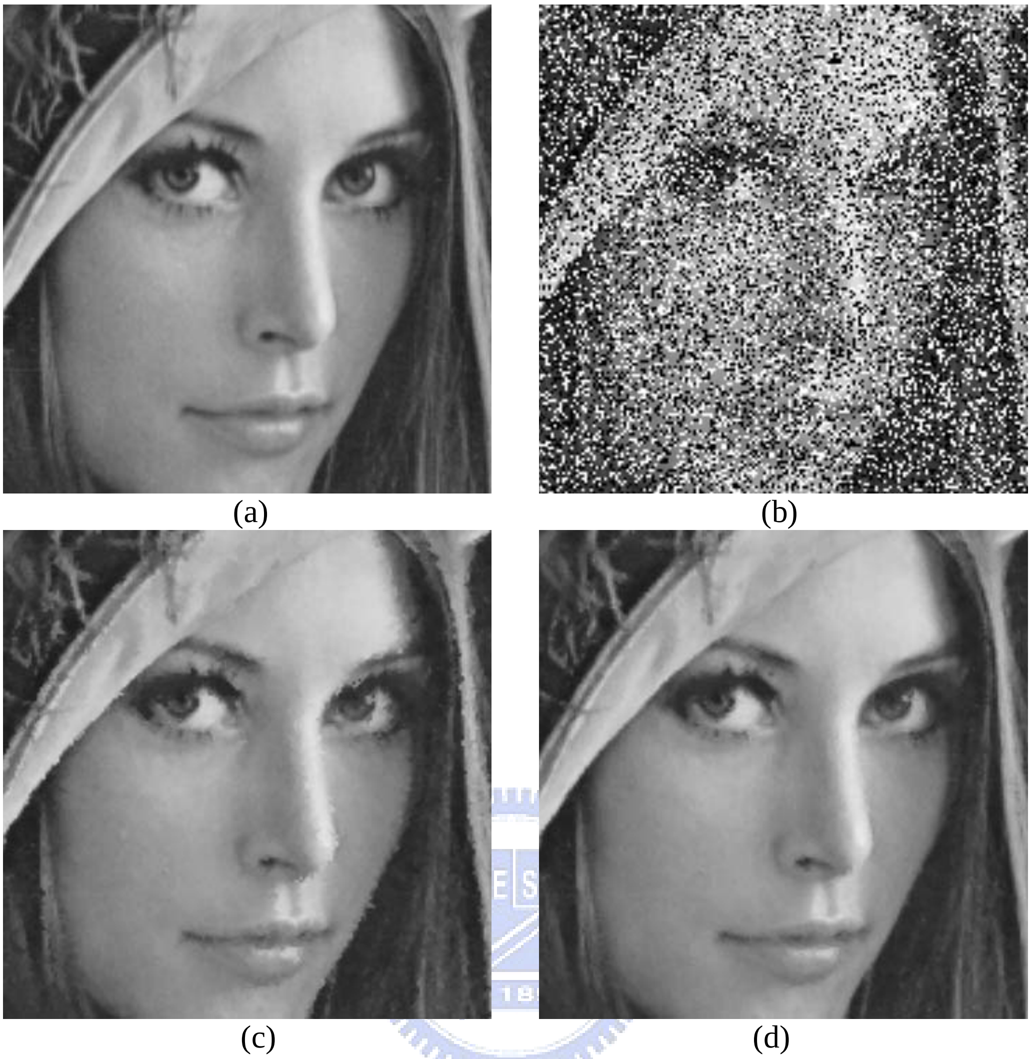
(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 3.11. (a) Original image; (b) Peppers with 40% of impulse noise; (c) the 3×3 standard median filter; (d) the 3×3 recursive standard median filter; (e) the recursive CWM filter with weight = 3; (f) the recursive Tri-state median filter with threshold = 25; (g) Li's method with threshold = 32; (h) our proposed method.



(a) (b) (c) (d) (e) (f)

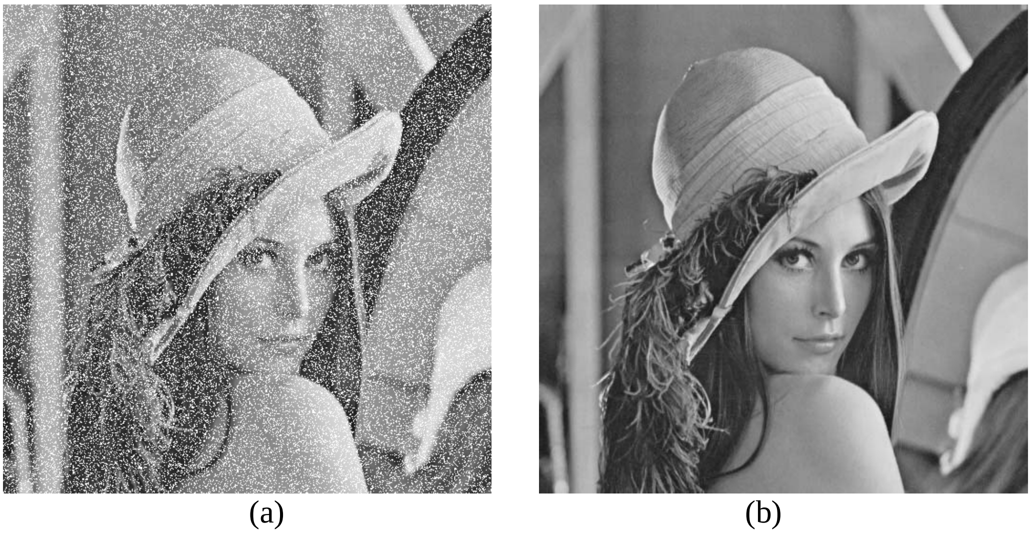
Fig. 3.12. (a) 30% Impulse Salt-Pepper noise [63]; (b) the 3×3 standard median filter; (c) the recursive Tri-state median filter with threshold = 25; (d) Li’s method with threshold = 32; (e) DMMD Denoise Software [65]; (f) our proposed method.



(a) (b)

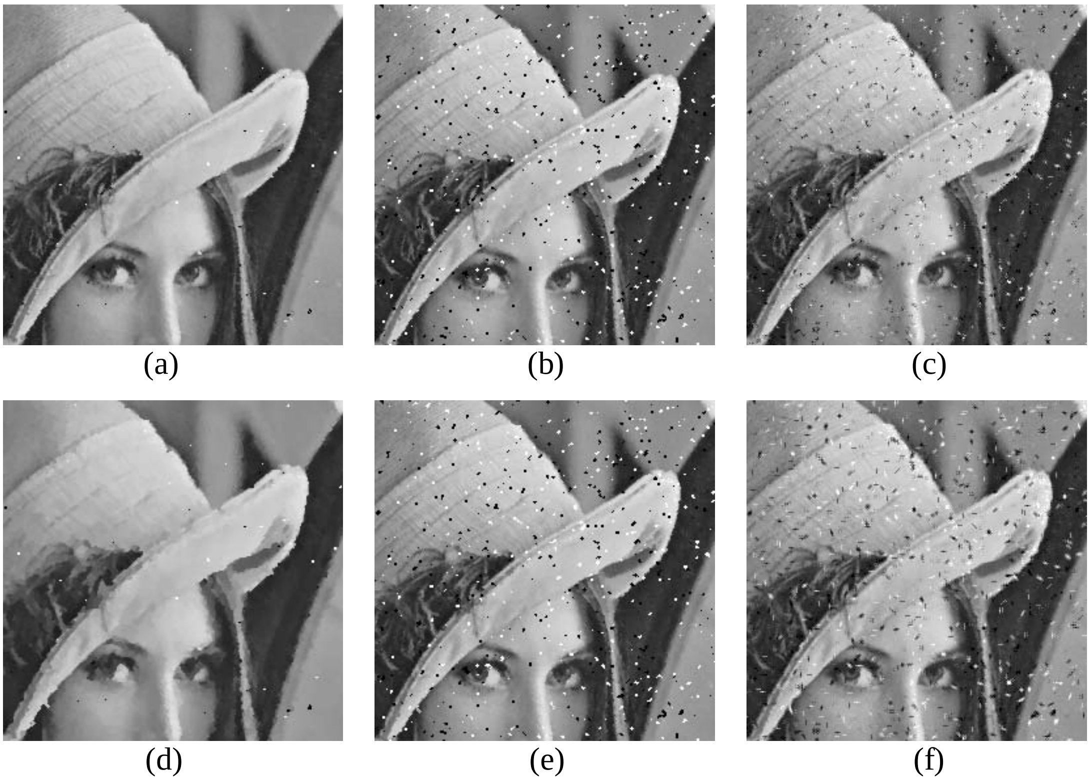
(c) (d)

Fig. 3.13. (a) Original image; (b) Lena with 40% of impulse noise; (c) image after processing of the 1st step (impulse noise removal); (d) image after the processing of 2nd step (image quality enhancement).



(a) (b)

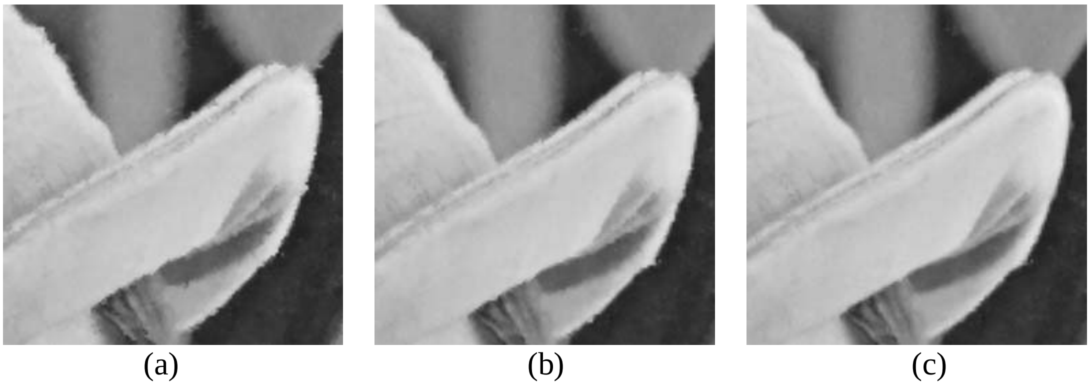
Fig. 3.14. (a) Lena with 20% of white impulse noise; (b) processed by our proposed method.



(a) (b) (c)

(d) (e) (f)

Fig. 3.15. The performance comparisons of different noise removal methods with/without the proposed noise range assumption applying to “Lena” corrupted with 40% impulse noise. (a) and (d) the 3×3 recursive standard median filter with and without the noise range assumption, respectively. (b) and (e) the recursive Tri-state median filter (threshold = 25) with and without the noise range assumption, respectively. (c) and (f) Li’s method (threshold = 32) with and without the noise range assumption, respectively.



(a) (b) (c)

Fig. 3.16. Performance comparison of the matrix-based estimation method and the proposed neural network for the 2nd step compensation with respect to the Lena image with 40% of impulse noise. (a) Image after the processing of the 1st stage; (b) and (c) images after the 2nd step compensation by using the resultant weighting matrices of the matrix-based estimation and the proposed neural network, respectively.

Table 3.1 Quantitative comparisons of different noise removal methods applied to the images with various percentages of impulse noise.

(a)

Filters	Images Corrupted with 10 % Impulse Noise					
	Lena	Peppers	Sailboat	Baboon	Aerial	Boat
Median	33.75	32.61	29.46	23.15	27.28	30.94
R-Median	32.55	32.24	28.40	22.66	25.75	29.72
CWM 3 [5]	34.45	33.36	30.72	25.13	28.49	31.97
Tri-State [14]	37.88	35.59	33.15	25.17	29.38	34.36
Li [24]	34.44	36.19	32.97	24.96	29.31	34.76
After our 1st step processing	43.12	41.70	38.46	31.70	33.04	40.24
The complete processing of our method	43.08	41.52	38.18	31.99	33.07	40.97

(b)

Filters	Images Corrupted with 20 % Impulse Noise					
	Lena	Peppers	Sailboat	Baboon	Aerial	Boat
Median	29.65	28.78	26.94	22.09	25.14	27.88
R-Median	31.02	30.55	26.89	22.11	24.61	28.50
CWM 3 [5]	30.28	29.31	27.40	23.32	25.72	28.27
Tri-State [14]	31.76	30.33	28.70	23.64	26.47	29.49
Li [24]	32.70	30.90	29.52	23.73	27.06	30.91
After our 1st step processing	39.40	37.72	34.70	28.60	30.98	36.45
The complete processing of our method	39.74	38.18	34.61	28.86	31.19	37.31

(c)

Filters	Images Corrupted with 40 % Impulse Noise					
	Lena	Peppers	Sailboat	Baboon	Aerial	Boat
Median	19.13	18.70	18.28	17.31	17.79	18.67
R-Median	26.86	25.79	23.45	20.69	22.01	24.84
CWM 3 [5]	20.18	19.54	19.02	18.30	18.64	19.56
Tri-State [14]	20.26	19.55	19.13	18.43	18.74	19.65
Li [24]	22.02	21.47	20.73	19.12	19.94	21.22
After our 1st step processing	34.55	32.80	30.10	24.96	27.57	31.71
The complete processing of our method	35.66	33.69	30.55	25.20	28.05	32.78

Table 3.2 Compensation ability of our adaptive median filter in the 1^{st} step and the image quality enhancement system in the 2^{nd} step with respect to Lena.

Impulse Noise Ratio	5 %	10 %	20 %	40 %
Percentage of 5×5 Median filter used in the 1^{st} stage	0%	0.05%	0.31%	3%
PSNR after the processing of the 1^{st} step	45.60	42.12	38.40	34.05
PSNR after the processing of the 2^{nd} step	46.26	43.08	39.74	35.66
Total pixels of Mo>Th	8595	16525	33428	65651
Total pixels corrupted by the noise	13218	26007	52813	104868

Table 3.3 Noise detection accuracy of the proposed method with respect to “Lena” image corrupted with various range and density of impulse noise.

Noise Rate	Noise Range	Total Correct Classification	Un-detection	Mis-detection
10%	05	100.0 %	0.0 %	0.0 %
	10	99.99 %	0.0 %	0.0007 %
	20	99.998 %	0.0 %	0.002 %
	30	99.79%	0.0 %	0.21%
20%	05	100.0 %	0.0 %	0.0 %
	10	99.9993 %	0.0 %	0.0007 %
	20	99.9977 %	0.0 %	0.0023 %
	30	99.815 %	0.0 %	0.185 %
40%	05	100.0 %	0.0 %	0.0 %
	10	99.9996 %	0.0 %	0.0004 %
	20	99.9989 %	0.0 %	0.0011 %
	30	99.86 %	0.0 %	0.14 %

Table 3.4 Speed Comparison for Various Algorithms (unit: sec)

Algorithm	**Median**	**R-Median**	**CWM 3**	**TRI**	**Li’s**	**The Proposed**
Ave. Time (sec.)	52	54	53	100	700	120

3.5.2. Simulation Results of Random-Valued Noise Removal

We firstly demonstrate the performance of the proposed neural network for noise detection. Three algorithms [12], [13], [17] were implemented to compare with our neural network and the detection results as shown in Table 3.5. Experimental results show that our neural network owns the highest detection precision than other methods. For Zhang's method [17], though the low Mis-detection can preserve the detail information, a large number of residual noises will damage the image seriously. On the contrary, for Pok's method [12], the high Mis-detection will blur the edge sharpness. Our detection procedure gives a better trade off between the Undetection and Mis-detection. Besides, our two-level impulse noise removal construction will further eliminate the residual noise pixels to get a near noise-free image.

Our proposed method is compared with several existing methods including 3×3 Standard Median Filter [1], Recursive Median Filter [2], Central Weight Median Filter [5], Tri-State Median Filter [14], Optimal Weighted Median Filters [4], Li's Edge-Preserving Filtering [24], Partition-Based Median Type Filters (PBM) [21], Zhang's algorithm using fuzzy approach [33], and Trilateral filter [31]. Various images with Salt-Pepper or random-valued impulse noises are used for testing although the proposed neural network for impulse noise detection is trained by only using the Lena image with 20% of random-valued noise as the training data. The proposed algorithm can work well for both random-valued and Salt-Pepper noises, since the Salt-Pepper noise can be regarded as a subset of random-valued noise with restricted noise distribution interval.

Table 3.6 shows the quantitative comparison of the proposed method and the other methods with respect to various kinds of images with 20% Salt-Pepper impulse noise. Tables 3.7 and 3.8 show the quantitative comparison of the proposed method and the existing methods with respect to images corrupted with random-valued impulse noise. According to these quantitative comparisons, the Partition-Based Median Type Filters (PBM) [21], Trilateral filter [31], and the proposed method can produce better and robust quantitative performance than the other methods. In Table 3.7, we can find that if the noise rate of the image for compensation is higher than 20%, the second stage of the proposed method will obviously improves the metric performance compared with the images processed by only the first stage. Besides, according to Table 3.8, we can find that the metric performance of Zhang's fuzzy approach [33] is not robust and varies depending on the image characteristics.

Fig. 3.17 shows the experimental results of different noise removal techniques applied to the "Lena" with 20% Salt-Pepper noise. Fig. 3.18 shows the processed results of "Boat" with 25% random-valued impulse noise. We can find that the Recursive Median Filter removes more noises than Standard Median Filter but also blurs the edges heavier. On the other hand, the Tri-State Median [14] and the Edge-Preserving Filtering method [18] can retain more edge sharpness but both of them cannot remove the noise very well. The Edge-Preserving Filtering method might even misjudge some noisy pixels as the edge and then increase the size of some noises. Generally, the threshold adjustment for different images is another problem that needs to be solved in these methods. For the Optimal Weighted Median Filters [4], it is obvious that the residual noises are not removed well in highly corrupted noisy images. The PBM filter [21] performs well in most regions except for the noises along edge area. In addition, the PBM filter should be trained separately to determine the filter parameter for images with fixed-valued noise or random-valued noise, respectively. The fuzzy approach [33] can remove the noise well but it destroys some edges and detail information very much especially for the images with high-density noise. In addition, it is much time consuming to find the ultimate remote window for noise cancellation and the coefficients need to be adjusted depending on the image and noise types. The Trilateral filter [31] cannot remove the noises along the edges very well since the linear type weighted average filtering algorithm is not suitable for removal of nonlinear type impulse noise. In addition, this solution would require a deep statistical study for the automatic selection of control parameters and the best way to do is still not clear. Above all, most of the algorithms cannot find the balance between noise removal and edge sharpness very well and they meet the same problem that as the noise rate increases their noise suppression ability decrease due to noise misidentification, edge destroying and blurring. Obviously, the proposed method produces effective and robust results over various noise corrupted images with different noise densities, and achieves a better result both in the noise reduction and detail preservation than other methods. Our HVS based image enhancement algorithm produces a more visually nature-looking image with smooth and sharp edge. Even though it might lose some superiority in the quantitative metric compared with the original image. We have also tested our algorithm and the others with textures and other more complicated images Mandrill and Barbara as shown in Figs. 3.19 and 3.20. According to the experimental results, some methods might get slightly sharper results in texture and complicated regions but they still could not remove the noise cleanly and the destroyed edge will heavily influence the image quality. In visual perception, our proposed algorithm can both appropriately remove the noise and retain the texture and edge regions well. Fig. 3.21 shows that even we apply the conventional algorithm twice to the “Pepper” image with 25% random-valued impulse noise, the noises still remain without obvious improvement. It strongly demonstrates that both our two-level decision-making procedure and the precise neural-network-based noise detection are indispensable.

The hardware and software environment that we implement the algorithms for speed comparisons are described as follows: all the algorithms are implemented in Matlab Language on a 1.8G Hz Pentium IV-based PC with 256 MB RAM. Table 3.9 shows the average computation time for various algorithms applied to different kinds and rates of noise corrupted images. The simple modified median type filters [1], [2], [5], and [4] that applying the same process to all pixels without noise identification, require lower average time consumption but they also produce the worse results for high noise-corrupted images. Since Tri-state median filter [14] incorporates the median filter and the CWM filter, the computation time is twice as the standard median filter. About Li's algorithm [24], owing to the least square estimation, the matrix multiplication, and inverse matrix computation, it takes much time to process. Regarding the PBM algorithm [21], since four types of CWM filters and the linear combination are operated on each pixel, the computation time is more than four times of the simple median filter required. Zhang's method [33] applies noise cancellation only to the pixels that look like impulses. Owing to a large number of 5×5 remote windows is searched within the 25×25 window to choose the ultimate one, and two or three iterations are needed, the algorithm requires most computation cost. About the Trilateral filter, applying two to five iterations is required to provide better results [23]. The proposed method applies noise removal to the identified noisy pixels and the image quality enhancement process is also applied to the noise-corrupted pixels in the visual sensitive region only, so the computation time can be greatly reduced compared with TRI, PBM, Li's and Zhang's methods. According to all the experiment results, it is demonstrated that the proposed method is superior to the existing methods both in perceptual image quality and time consumption.

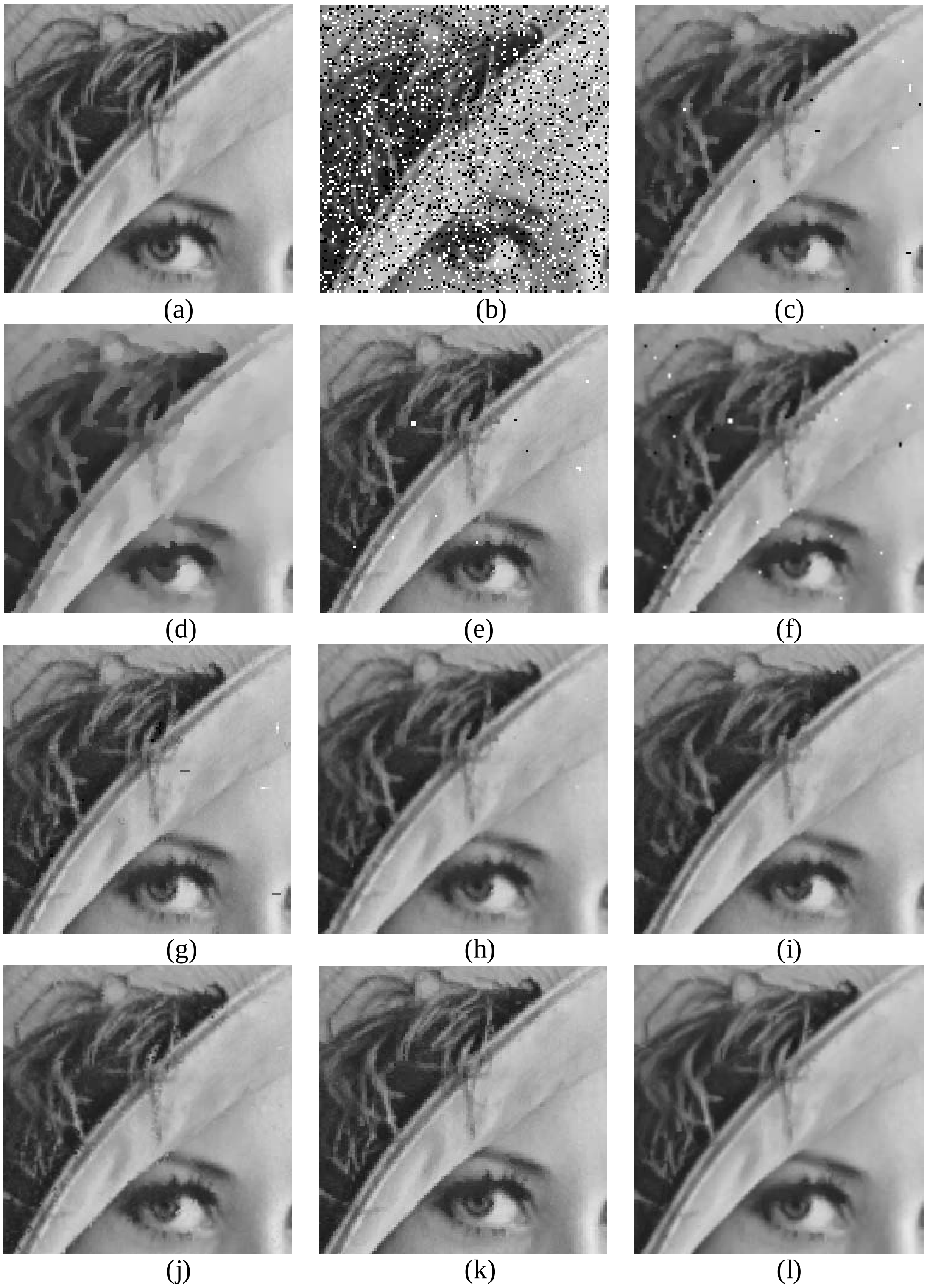
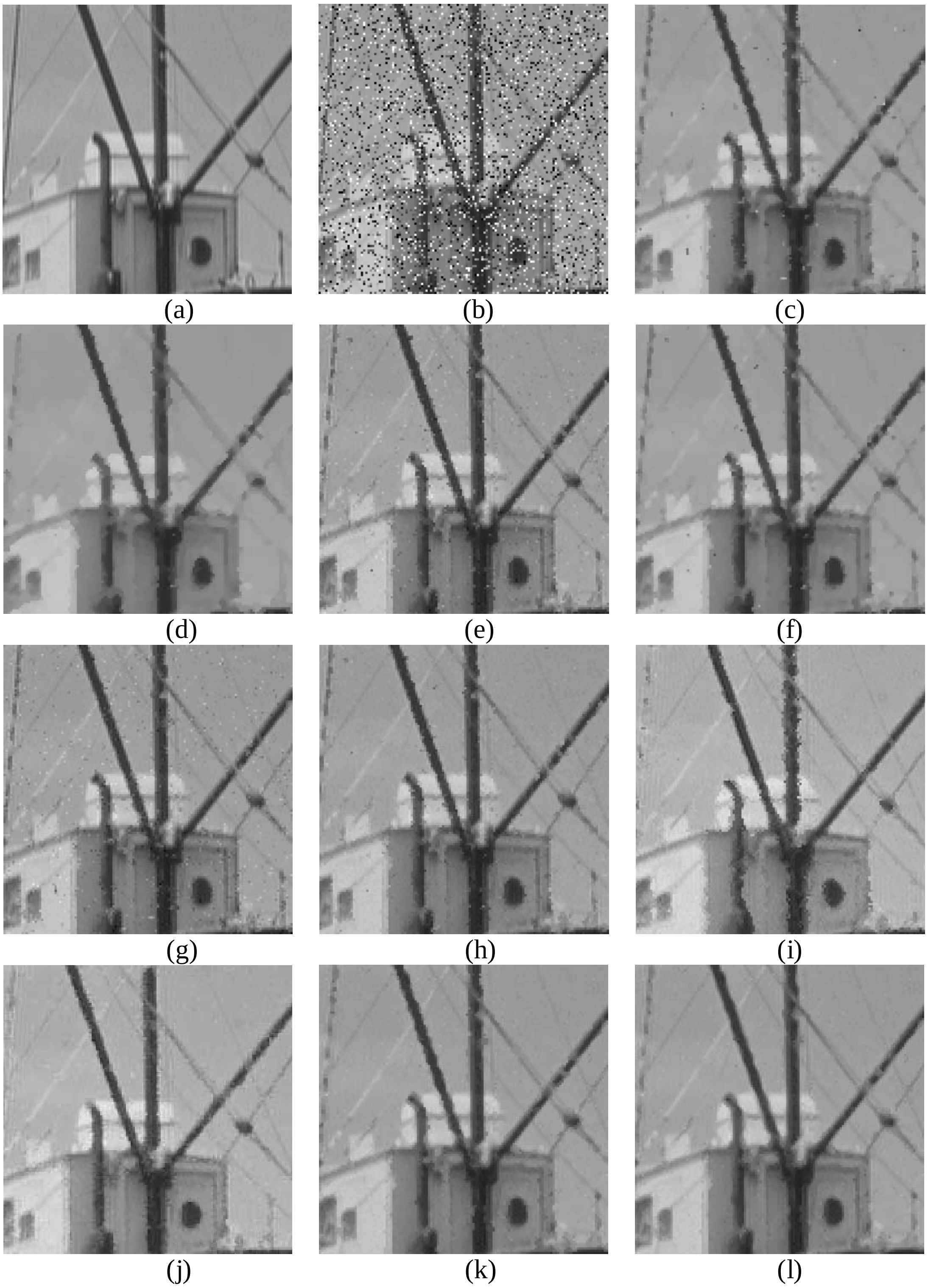


Fig. 3.17. (a) Original image; (b) Lena with 20% of fixed-valued impulse noise; (c) the 3×3 standard median filter; (d) the 3×3 recursive median filter; (e) the recursive tri-state median filter with threshold = 25; (f) optimal weighted median filter; (g) Li's edge preserving method with threshold = 32; (h) PBM filter; (i) Zhang's fuzzy techniques; (j) Trilateral filter; (k) 1st step of the proposed method; (l) The proposed two-step method.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 3.18. (a) Original image; (b) Boat with 25% of random-valued impulse noise; (c) the 3×3 standard median filter; (d) the 3×3 recursive median filter; (e) the recursive tri-state median filter with threshold = 25; (f) optimal weighted median filter; (g) Li’s edge preserving method with threshold = 32; (h) PBM filter; (i) Zhang’s fuzzy techniques; (j) Trilateral filter; (k) 1st step of the proposed method; (l) The proposed two-step method.

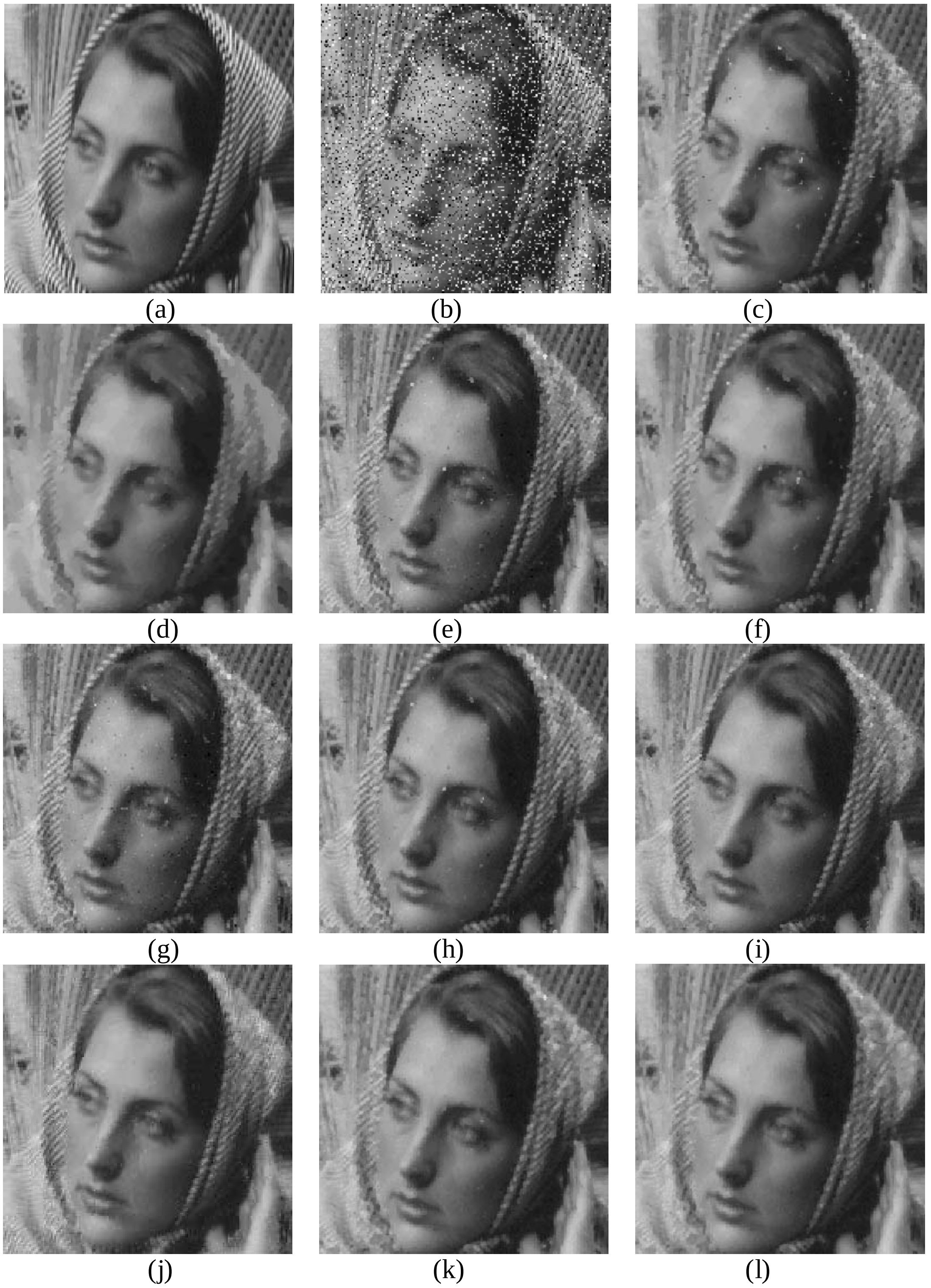
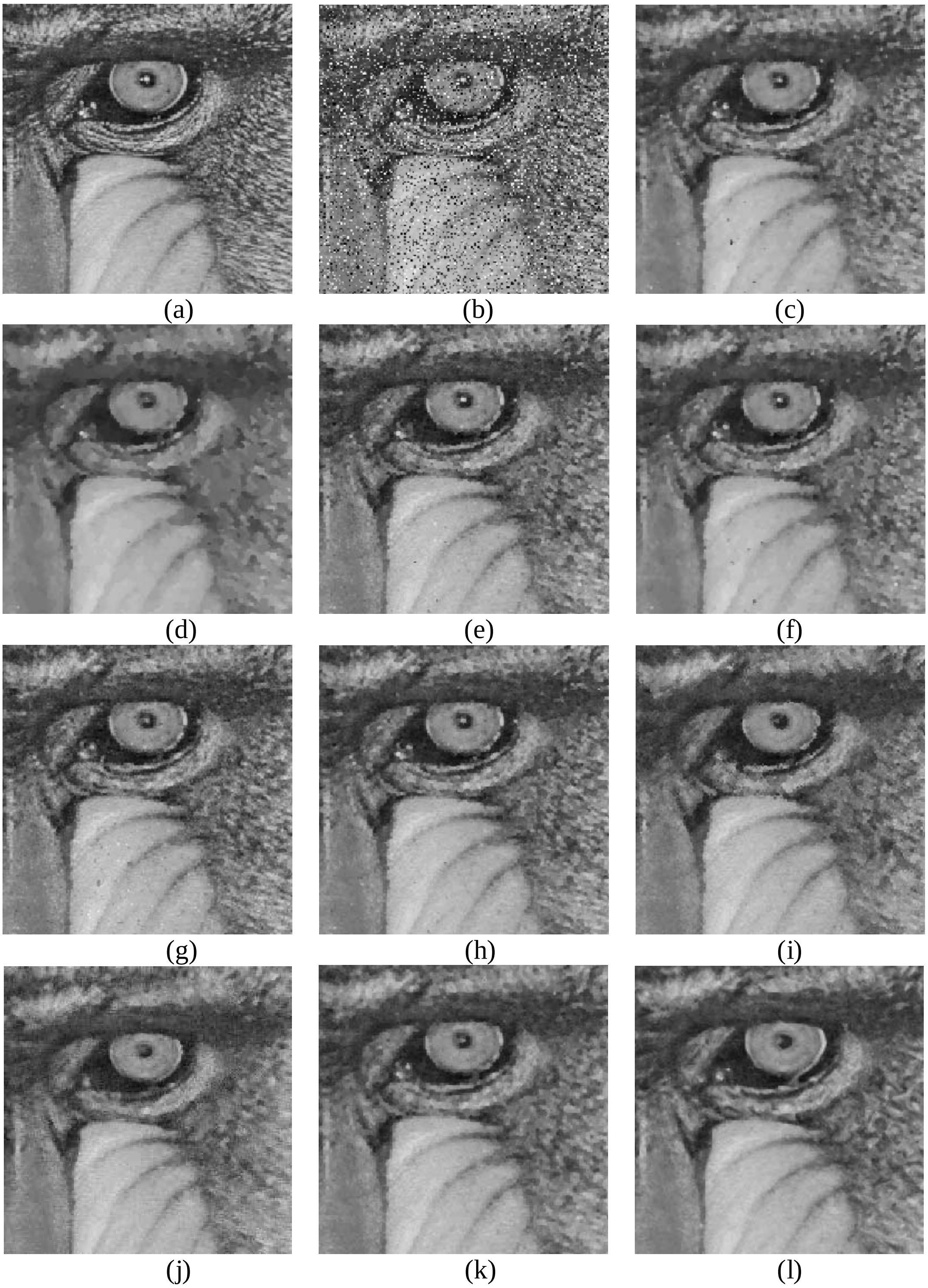


Fig. 3.19. (a) Original image; (b) Barbara with 25% of random-valued impulse noise; (c) the 3×3 standard median filter; (d) the 3×3 recursive median filter; (e) the recursive tri-state median filter with threshold = 25; (f) optimal weighted median filter; (g) Li's edge preserving method with threshold = 32; (h) PBM filter; (i) Zhang's fuzzy techniques; (j) Trilateral filter; (k) 1st step of the proposed method; (l) The proposed two-step method.



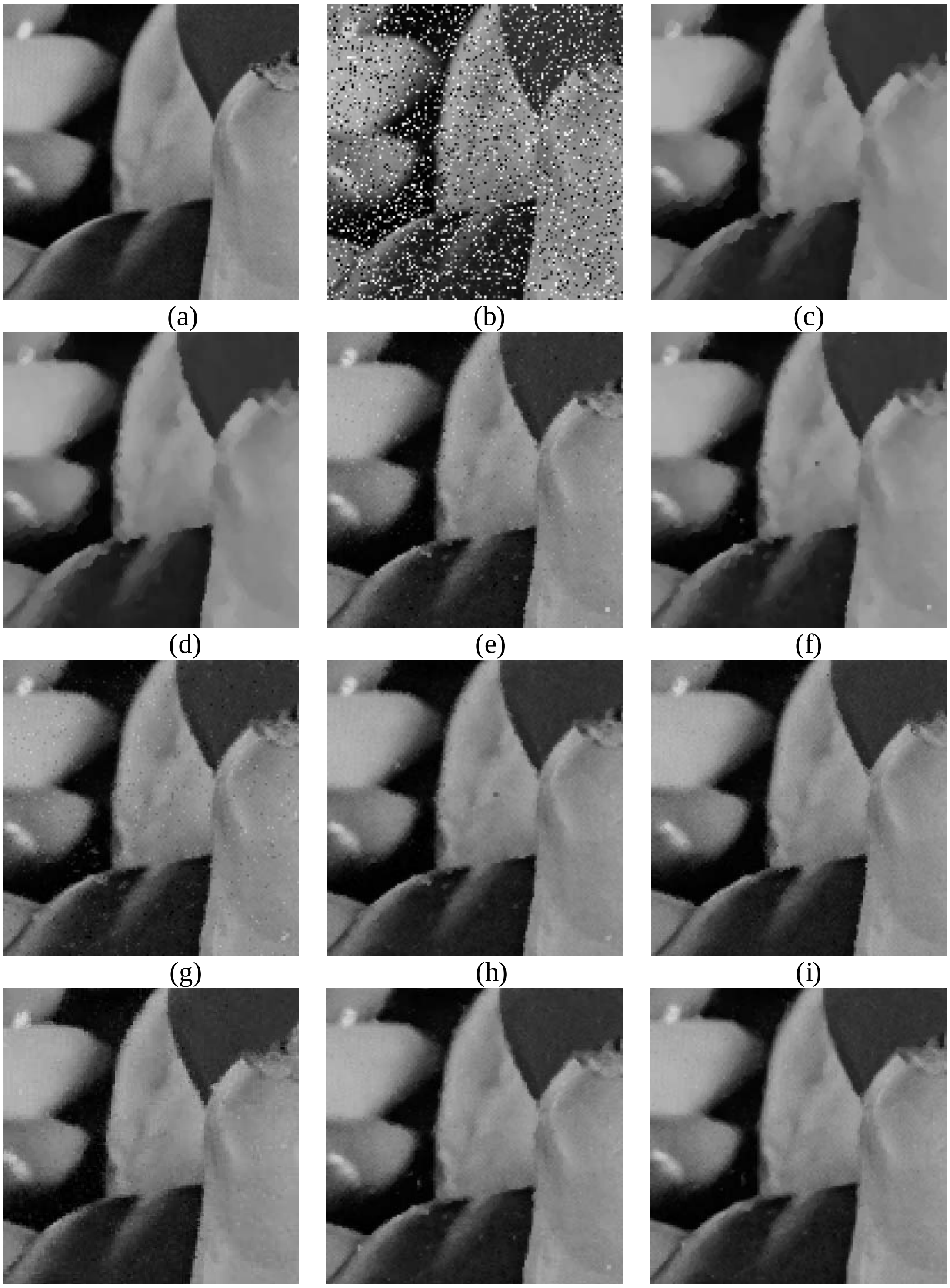
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 3.20. (a) Original image; (b) Baboon with 25% of random-valued impulse noise; (c) the 3×3 standard median filter; (d) the 3×3 recursive median filter; (e) the recursive tri-state median filter with threshold = 25; (f) optimal weighted median filter; (g) Li’s edge preserving method with threshold = 32; (h) PBM filter; (i) Zhang’s fuzzy techniques; (j) Trilateral filter; (k) 1st step of the proposed method; (l) The proposed two-step method.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 3.21. (a) Original image; (b) Peppers with 25% of random-valued impulse noise; (c) twice applying of 3×3 standard median filter; (d) twice applying of 3×3 recursive median filter; (e) twice applying of the recursive tri-state median filter with threshold = 25; (f) optimal weighted median filter; (g) twice applying of Li's edge preserving method with threshold = 32; (h) twice applying of the PBM filter; (i) Zhang's fuzzy techniques; (j) twice applying of the Trilateral filter; (k) result after 1st step processing of the proposed method; (l) results of the proposed two-step method.

Table 3.5 Comparative results of various noise detection algorithms applied to random-valued noise corrupted images with 25% noise density.

Name	Algorithms	Total Correct Classification	Undetection	Mis-detection
Lena	S.K. Mitra [13]	59.40 %	68.04 %	31.45 %
	G. Pok [12]	88.89 %	24.12 %	6.69 %
	S. Zhang [17]	89.52 %	3.00%	12.97 %
	The proposed	92.58 %	14.48 %	5.06 %
Boat	S.K. Mitra [13]	59.50 %	68.60 %	31.13 %
	G. Pok [12]	87.22 %	26.60 %	8.19 %
	S. Zhang [17]	88.81 %	4.72 %	13.33 %
	The proposed	89.40 %	14.64%	9.25 %
Goldhill	S.K. Mitra [13]	58.89 %	67.32 %	32.37 %
	G. Pok [12]	88.49 %	25.48 %	6.85 %
	S. Zhang [17]	89.16 %	3.48%	13.29 %
	The proposed	90.92 %	15.60 %	6.90 %
Peppers	S.K. Mitra [13]	59.07 %	68.64%	31.71 %
	G. Pok [12]	88.43 %	24.00 %	7.42 %
	S. Zhang [17]	88.91 %	4.64 %	13.25 %
	The proposed	91.90 %	14.64 %	5.92 %
Sailboat	S.K. Mitra [13]	58.83 %	68.56 %	32.04 %
	G. Pok [12]	86.96 %	28.52 %	7.88 %
	S. Zhang [17]	87.98 %	6.84 %	13.74 %
	The proposed	87.26 %	15.20 %	11.92 %

Table 3.6 Comparative results in PSNR of different algorithms applied to various kinds of images corrupted with 20% fixed-valued impulse noise.

Filters	Lena	Peppers	Goldhill	Elaine	Cameraman	Boat
Median [1]	29.65	28.78	27.86	29.00	24.60	27.88
R-Median [2]	31.02	30.55	29.24	30.96	24.36	28.50
CWM 3 [5]	30.28	29.31	28.51	29.38	25.22	28.27
Tri-State [14]	31.76	30.33	29.78	30.82	25.26	29.49
OWMF [4]	31.44	30.64	29.52	30.33	25.75	29.36
Li's [24]	32.70	30.90	30.99	31.93	25.97	30.91
PBM [21]	34.62	33.09	32.52	34.31	25.85	31.93
Zhang's [33]	34.35	31.40	30.77	34.67	23.66	29.77
Trilateral [31]	33.53	31.50	32.19	34.02	24.86	31.45
1st Step of The Proposed Method	33.85	32.44	31.45	34.37	25.33	30.61
The Proposed Two-Step Method	33.54	32.30	31.37	34.32	25.10	30.31

Table 3.7 Comparative results in PSNR of different algorithms applied to image "Elaine" corrupted by various rates of random-valued impulse noise.

Filters	5 %	10 %	15 %	20 %	25 %	30 %
Median [1]	32.65	32.29	31.70	30.79	29.67	28.06
R-Median [2]	32.53	32.12	31.55	31.03	30.46	29.58
CWM 3 [5]	34.44	33.76	32.81	31.73	30.44	28.57
Tri-State [14]	40.29	37.29	35.23	33.58	32.27	30.58
OWMF [4]	33.73	33.29	32.56	31.79	30.93	29.45
Li's [24]	32.83	32.45	31.98	31.04	30.81	29.09
PBM [21]	39.37	37.56	35.79	34.39	33.20	31.55
Zhang's [33]	37.44	36.08	34.89	34.01	33.12	32.30
Trilateral [31]	38.53	36.71	35.42	34.17	33.18	32.37
1st Step of The Proposed Method	38.63	36.78	35.43	34.36	33.32	31.89
The Proposed Two- Step Method	37.93	36.19	35.20	34.34	33.48	32.39

Table 3.8 Comparative results in PSNR of different algorithms applied to various kinds of images corrupted with 25% of random-valued impulse noise.

Filters	Lena	Peppers	Goldhill	Elaine	Camera	Boat
Median [1]	29.84	29.34	28.37	29.67	24.07	28.95
R-Median [2]	30.17	29.33	28.71	30.46	24.01	28.87
CWM 3 [5]	30.21	28.88	28.97	30.44	24.89	29.47
Tri-State [14]	31.59	30.58	30.50	32.27	25.13	30.74
OWMF [4]	30.96	29.76	29.60	30.93	25.21	30.09
Li's [24]	30.40	29.19	29.66	30.81	24.91	29.97
PBM [21]	33.07	31.35	31.18	33.20	24.71	31.38
Zhang's [33]	31.54	29.86	29.03	33.12	23.09	27.82
Trilateral [31]	31.46	30.86	30.44	33.18	24.16	29.00
1st Step of The Proposed Method	32.95	30.53	30.69	33.32	24.83	29.76
The Proposed Two- Step Method	33.00	31.32	30.73	33.48	24.72	31.10

Table 3.9 Speed Comparison for Various Algorithms (sec.)

Nose Rate	Median [1]	R-Median [2]	CWM [5]	OPMF [4]	TRI [14]	Li's [24]	PBM [21]	Zhang's [33]	Trilateral [31]	The Proposed
1 0 %	64.9	68.6	69.8	79.1	149.7	731.6	378.7	435.5	45.2	47.7
2 0 %	75.1	76.9	82.8	84.3	149.5	735.3	375.2	671.6	136.43	84.7
4 0 %	72.3	69.9	72.9	94.6	170.8	692.5	369.2	1202.3	215.42	126.3

3.6. Brief Summary

In this chapter, a novel two-step noise removal algorithm to deal with impulse noise is proposed. The decision-based recursive adaptive median filter is applied to remove the Salt-Pepper noise, and an adaptive two-level neural network noise reduction procedure is applied to remove the random-valued noise and keep the uncorrupted information well. Then the fuzzy decision rules inspired by human visual system (HVS) is applied to compensate the blur of the edge and the destruction caused by median filter. According to the experimental results, the proposed technology is much superior to the conventional methods in some aspects such as cleanness of noise removal and the visual quality of the processed images.

4. Modified Fuzzy Rule Based Method for Gaussian Noise Removal

In this chapter, we derive a modified fuzzy rule-based (MFRB) filter to remove the linear type Gaussian noise while preserving the image edges and details as much as possible. For practical consideration, we design several sets of universal MFRB filters in correspondence to the estimated values of contaminated Gaussian noise variance in the image. We combine the Salt-Pepper noise removal algorithm described in Chapter 3, and the universal MFRB filter together to sequentially remove the mixed noise in an image. The generalization ability of the proposed method has been demonstrated by examining it on a variety of noise-corrupted testing images corrupted with various noise densities. According to the experiment results, the proposed method is superior, both quantitatively and visually, compared to several other techniques.

4.1. Introduction

To remove the Gaussian noise artifacts in an image, we wish to design a group of adaptive filters, whose weighted value combinations constitute the fuzzy rules for pixel value restoration. Fuzzy rules have been found useful to reconstruct the pixel value from the pixel itself and its neighboring pixels as well. This is because that the local statistics and structural characteristics of the signals can be taken into consideration by the fuzzy rules. For best-fit and unbiased considerations, fuzzy rule based system through learning can achieve the best combinations of rules, and thus can usually produce a better off one in comparison with the filters constructed from the knowledge or experience of a domain expert. Therefore, the proposed MFRB filters scheme adopts a training method to learn suitable sets of weight parameters for image Gaussian noise removal. Namely, the weights of MFRB filters are obtained by minimizing the mean square output error between the noisy and original image data. As a result, signal restoration aiming for Gaussian noise removal by MFRB filter is possible via pixel restoration and it adapts reasonably well from smooth regions to edge areas.

This chapter is organized as follows. The newly modified fuzzy rule based filter is proposed in Section 4.2. Section 4.3 shows the experimental results with various comparisons to other schemes. Section 4.4 gives conclusions of this chapter.

4.2. Modified Fuzzy Rule-Based Filter

Arakawa [67] developed the Fuzzy Rule Based (FRB) filter that adapted the ambiguity of image signals caused from the following three parameters: gray level variations, signal spatial distribution, and the local structure of the pixels. The first parameter is the difference between the input pixels' gray values, the second is the time (or position) difference between pixel points, and the third is the pixel's local variance in the sliding window. The filter coefficients are determined through learning from the difference between the noisy image and its original image. After training, the dominant roles of the signal distribution texture, spatial and edge structures, and local pixel statistics of the image will be learned by a set of weight parameters constituting the FRB filter. In what follows, a brief review of one-dimensional FRB filter, for simplicity, is described first.

4.2.1. FRB Filter

Suppose that a noisy signal x_n is obtained at time instant n as $x_n = d_n + u_n$, where d_n denotes the original signal and u_n denotes the white Gaussian noise with zero mean and certain variance. The problem is to estimate d_n from the input sequence $\{ x_{n-k} \}$, $-N \le k \le N$. The filter to estimate d_n can be expressed as follows:

$$y_n = \frac{\sum_{k=-N}^{N} \mu_{n;n-k} x_{n-k}}{\sum_{k=-N}^{N} \mu_{n;n-k}}, \tag{4.1}$$

where y_n denotes the estimated value of d_n, the index k in $\mu_{n;n-k}$ denotes the extent that the input sequence x_n should be shifted and calculated in the average to estimate d_n. In order to take ambiguous, uncertain image signal structures, and signals' local statistics into consideration, $\mu_{n;n-k}$ should be trained from typical ensembles of noise corrupted images. In estimating signals, Arakawa [67] proposed the following rules of thumb to accommodate these weights $\mu_{n;n-k}$ to embrace the above mentioned characteristics of the image signal, as summarized below.

(a) $\mu_{n;n-k}$ should be large for the input x_{n-k}, if the difference between the signal values x_n and x_{n-k} is small, since in this case x_{n-k} is considered to be in a similar signal level of x_n. Otherwise, $\mu_{n;n-k}$ should be small.

(b) $\mu_{n;n-k}$ should be large for the input x_{n-k}, if the time difference (reduced to spatial distance for image signal and so does hereafter) k between x_n and x_{n-k} is small, since image signals almost follow first-order

Markov process and are usually more correlated for smaller time distance. Otherwise, $\mu_{n;n-k}$ should be small.

(c) If the local variance observed around the instant *n* is small, these local pixels could be in the flat area and $\mu_{n;n-k}$ could be spread over the temporal axis; on the contrary, $\mu_{n;n-k}$ should be concentrated around the point *n* to avoid quite different member inclusion if the local variance is large.

To simplify the case, suppose the FRB rule is applied to filter a one-dimensional signal. Weight $\mu_{n;n-k}$ represents the weight coefficient of x_{n-k} to estimate x_n, in which $\mu_{n;n-k}$ first only represents characteristic (a) for simplicity. Let $\mu_{n;n-k}$ be denoted as $\mu_j(k)$, and simply as μ_j hereafter, which represents the membership value dealing with the *j*-th interval partition of the signal difference feature. Moreover, the value μ_j can be determined through learning for all possible *j* partitions in the least mean square sense. With respect to training some Gaussian noise images, the mean square error between the filter output y_n and true signal d_n can be minimized by a gradient method. To this end, when the least mean square (LMS) learning algorithm [68] is applied, we can easily show that μ_j can be trained iteratively by

$$\mu_j(T+1) = \mu_j(T) + \alpha \cdot X(T,j) \cdot (d_n - y_n), \tag{4.2}$$

where $\mu_j(T)$ is the value of μ_j at training iteration *T*, α is the learning factor, and $X(T,j)$ is given by

$$X(T,j) = \frac{\sum\limits_{k\in K(j)} x_{n-k} - N(j)\cdot y_n}{\sum \mu_j(T)} . \tag{4.3}$$

In the above, the denominator is the sum of $\mu_j(T)$ for $\{x_{n-k},\ -N \le k \le N\}$; $K(j)$ denotes the set which satisfies the feature value inside the *j*-th interval $\varepsilon_{j-1} \le x_n - x_{n-k} < \varepsilon_j$, and $N(j)$ is the number of the elements in the set above. The learning procedure is repeated until $\mu_j(T)$ converges.

To generalize, $\mu_{n;n-k}$ can be extended to depend on multiple characteristics, (a), (b), and (c) at once. To this end, $\mu_{n;n-k}$ should be subscripted in a multidimensional form so that the subscript can include all the three feature characteristics concerning (a), (b), and (c), which encompass the *j*-th index category for the gray level difference $x_n - x_{n-k}$, the *l*-th index category of temporal distance *k*, and the *m*-th index category for the local variance σ_n of processing window pixels centered around x_n. Namely, μ_{jlm} represents the weighting coefficients at the *j*-th, *l*-th, and *m*-th index category concerning $x_n - x_{n-k}$, *k*, and σ_n, respectively.

In a similar manner, it follows from Eq. (4.2) that the value μ_{jlm} can be obtained by the LMS learning algorithm iteratively as follows:

$$\mu_{jlm}(T+1) = \mu_{jlm}(T) + \alpha \cdot t_{jlm} \cdot (x_{n-k} - y_n)(d_n - y_n) / \sum_{k=-N}^{N} \mu_{jlm} , \tag{4.4}$$

where $\mu_{jlm}(T)$ is the value of μ_{jlm} at learning iteration *T*, α is the learning factor, and t_{jlm} is equal to 1 when x_{n-k} belongs to the specified *j*, *l*, and *m* intervals and equals to zero otherwise. So in the training procedure, signal points are trained to update its involving interval weighting coefficient μ_{jlm} until μ_{jlm} converges.

As illustrated above, it is to be noted that the concept of the fuzzy rule based filter is named in a broad sense. There are no conventional fuzzy rules in the FRB filter because each 3-index weighting coefficients, i.e., the case differentiated by the partition in the 3-D parameter space, will be used to predict a distorted or noise-corrupted value by weighted-sum filtering procedure. Each μ_{jlm} coefficient corresponding to the specified *j*, *l*, and *m* intervals is considered as if it were a rule, and the set of all possible *j*, *l*, and *m* combinations constitutes the whole rule set. Moreover, the weighting coefficients versus each index dimension, i.e., *j*, *l*, or *m*, play the similar role of the membership function defined for that dimension.

The FRB filter is discussed above for the 1-D signal recovery. When the target signal is an image, the rule concerned in (b) will be not only the temporal distance *k* between x_n and x_{n-k} but also the spatial relation between x_n and x_{n-k}. Besides, the partition intervals for the parameters concerned in (a) and (c) can also be better designed to achieve more accurate restoration performance. The improved version of the FRB filter is derived below.

4.2.2. The Proposed Modified FRB Filter

In this section, we will propose the modified fuzzy rule based filters for image Gaussian noise removal. Exploiting Eq. (4.4) above, a set of filter coefficients of MFRB denoising filter can be determined through the training over Gaussian corrupted images. In the denoising phase, any corrupted images can be processed to reduce the noise employing the set of weighting coefficients we have learned. In an image, the pixels' gray levels, statistical variances, and positional distances and orientations are the three most important factors to be considered altogether. For the MFRB design below, we will not only better determine the gray-level difference interval $[\varepsilon_{j-1}, \varepsilon_j)$ and variance interval $[\delta_{m-1}, \delta_m)$ but also propose a new clustering scheme of pixels inside the working window, taking both pixels' positional difference and spatial correlation into consideration.

The details regarding the design of these three parameter spaces will be illustrated below: (1) to better design the error difference interval $[\varepsilon_{j-1}, \varepsilon_j)$ for the first parameter, the gray level difference or variable (a); (2) to choose the cluster the pixel x_{n-k} should belong to, for the second parameter, the spatiality or variable (b); and (3) to better determine variance interval $[\delta_{m-1}, \delta_m)$ for the third parameter, the gray level local variance or variable (c).

Variable (a):

The source image is 8-bit gray-scale level with the size of 512×512. The gray value of each pixel lie in $[0,\ 1, \cdots,\ 2^8 - 2,\ 2^8 - 1]$, hence the magnitude of the maximum difference between x_n and x_{n-k} is 255. For efficiency consideration, the size of the error difference interval could be better partitioned if we can take the statistics of the error signal $|\, x_n - x_{n-k} \,|$ into consideration. Therefore, we borrow the idea from the *A*-law (μ-law) in the DPCM design [69] to determine the sizes of error difference interval to achieve minimal error in the statistical sense. To this end, we first calculate the error signal distribution for images corrupted with different Gaussian noise levels. The probability distribution of several typical images for the rule (a) is shown in Fig. 4.1. We found that gray level difference magnitude $|\, x_n - x_{n-k} \,|$ mostly lies in the interval [0, 100] and occurs most frequently around zero. With these observations in mind, we are easy to determine, in a manner similar to *A*–law, better gray level difference interval through the ensemble of the error curves of

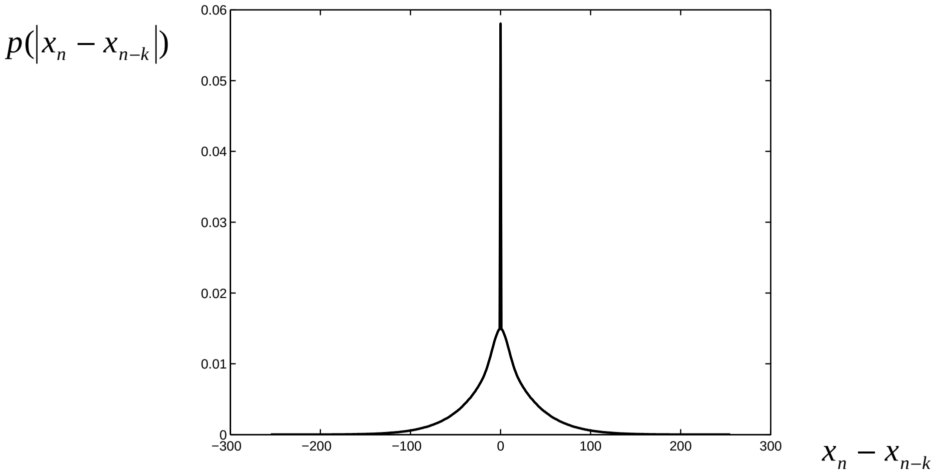


Fig. 4.1. The probability distribution of several typical images for the Variable (a).

the representative noisy images. To do this, we choose in such a way that the size of gray-level difference interval is inversely proportional to the probability of gray level difference. That is, the smaller the gray-level magnitude difference is, the smaller difference interval will be chosen for the gray level difference dimension, and vice versa. In practice, we can approximate to reach the spirit of *A*-law easily by a few trial-and-error iterations. In this setting, for example, we can choose the difference interval set as: [0, 1), [1, 2),..., [31, 32), [32, 34), [34, 36),..., [62, 64), [64, 68), [68, 72),..., [92, 96), [96, 104), [104, 112),..., [120, 128), [128, 138), [138, 158), [158, 198), and [198, 255] for the magnitude difference clusters of pixel values x_n and x_{n-k}. For instance, if the difference interval $|x_n - x_{n-k}|$= 33, then the image pixel x_{n-k} would belong to the cluster $[32, 34)$.

Variable (b):

In our experiment, the size of the processing window of the MFRB filter is chosen to be space 5×5, as shown in. In the Variable (b), it is reasonable to take both the position distance and orientation relation between pixel x_n and pixel

x_{n-k} into consideration. The image structure, edge and/or texture, will be suitably represented if we consider the position distance factor and the orientation factor as well. With this concept in mind, we can divide the window pixels of Fig. 4.2 into the following 13 clusters: $\{x_1, x_{25}\}$, $\{x_2, x_{24}\}$, $\{x_3, x_{23}\}$, $\{x_4, x_{22}\}$, $\{x_5, x_{21}\}$, $\{x_6, x_{20}\}$, $\{x_7, x_{19}\}$, $\{x_8, x_{18}\}$, $\{x_9, x_{17}\}$, $\{x_{10}, x_{16}\}$, $\{x_{11}, x_{15}\}$, $\{x_{12}, x_{24}\}$, and $\{x_{13}\}$. These clusters are generated by considering both the pixel's positional distance and orientation correlation, with respect to the center pixel x_{13}. For example, the elements x_8 and x_{18} are classified into the same cluster because x_{13} is in the center of the line connecting them and they have the same positional distance from x_{13}. According to the spatial relations, the proposed MFRB filter can become adaptive to both edge direction and position distance so that the signal structure, such as texture, edge, and details of the image, can be better restored and compensated. Along a similar spirit of reasoning, the superior performance of the edge-directed prediction [70] has been found to be attributed to the edge-directed property in the least squares optimization prediction.

x_1	x_2	x_3	x_4	x_5
x_6	x_7	x_8	x_9	x_{10}
x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}

Fig. 4.2. Elements and their subscripts in the 5×5 filter window.

Variable (c):

The design of the third parameter, variance, is similar to the way we design Variable (a). The only difference is that we have to analyze the distribution of local variances of the pixels in the chosen 5×5 window of the noise corrupted images first. To observe the example of the local variances of noise corrupted images, the typical local variance σ_n^2 probability distribution is shown in Fig. 4.3. It can be easily seen that most of them lie in [0, 3000]. To faithful realizing *A*-law concept regarding the variance factor, we can divide the variance interval $[\varepsilon_{m-1},\varepsilon_m)$, as the way adopted in Variable (a): the size of variance interval is approximated inversely proportional to the relative frequency of local variance σ_n^2 by a few trials.

In summary, the size of the membership function matrix μ_{jlm} in each noise level is chosen to be a look-up table of size $64\times13\times100$, designed with the spirits of efficient partitioned intervals and adaptive edge-directed spatial relations being proposed above.

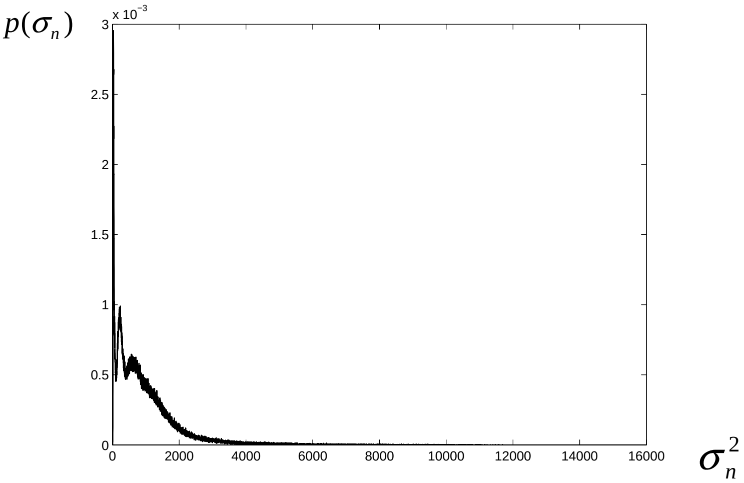


Fig. 4.3. A typical local variance σ_n probability distribution for the Variable (c).

4.2.3. Two-Stage Universal MFRB Filter for Mixed Noise Removal

For practical applications, the percentage of impulse noise and the variance of Gaussian noise are unknown in a noise-corrupted image. It is demonstrated that the proposed impulse noise removal algorithm described in Chapter 3 can effectively remove the impulse noise in spite of the different impulse noise corruption percentage. So the noise remained after the first stage, is mostly the Gaussian noise original in the image and a small fraction impulse noise not removed or imprecisely restored impulsive pixels.

Though the MFRB noise removal filter is most robust to the level of Gaussian noise corrupted, it is impractical and too hardware consuming to train the look-up table of each noise level. Since the filter scheme is quite effective in a certain noise level range, for this reason, a universal MFRB filter can be obtained through training over several typical images having similar levels of Gaussian noise contamination. For cost-effectiveness consideration, we designed three MFRB filters at σ =10, 20, and 30 to constitute a universal MFRB filter. From our experimental experiences, these three MFRB filters can produce well enough filtering result and they do not cost too much in hardware and memory requirements. Accordingly, each MFRB filter of a certain noise level, i.e., σ =10, 20, and 30, can be trained by some ensemble images that are corrupted with the same Gaussian noise level. As usual, a train loop covering all training image set once is called an *epoch*, and let μ_n denote the membership function after learning *n* epochs. When the variation of the membership function μ in the training process is smaller than a predefined threshold T_h, we will stop the training process. In our experiment, if the condition satisfies

$$\frac{\|\mu_{dn}\|}{\|\mu_n\|} \le T_h , \tag{4.5}$$

where $\mu_{dn} = \mu_n - \mu_{n-1}$, and $T_h = 0.01$, we will terminate the training process and obtain the universal MFRB filter μ_{jlm} for this Gaussian noise variance. In this way, we will have a universal MFRB Gaussian noise removal filter in the sense that it is image independent and contains three MFRB de-noise filters with each constituted μ_{jlm}'s being designed for a Gaussian noise level σ =10, 20, or 30, respectively.

Since the particular MFRB filter to be employed from the universal MFRB filter depends on the Gaussian noise level in an image, the noise level estimation is necessary for efficient noise removal. It is well known that the noise variance of a local area can be estimated better by the local variance of a flat area to get rid of excessive offset by the edge and minute details of images. Accordingly, a method will be devised to estimate the Gaussian noise level of an image efficiently. In our implementation, an overlapping 7×7 sliding window is used and each sliding window centered a region of 3×3 pixels for variance estimation at this pixel. As illustrated in, X_i represents the center pixel of the *i*-th region, which is remarked by *i* on the left-upper corner of the processing region. After computing the local variances of the X_i, the smaller variance ones obtained in the sliding window will be the better estimation in order not to over-count the variation caused by local edges. Owing to the small sample size involved in the computation of the local variances, an average of the 25 smallest variances out of the involved 49 pixels' variances will be used to estimate the variance for the block size of 7×7. Then, the final value of the estimated Gaussian variance is the average of those local variances calculated with the overlapping 7×7 sliding window over the whole image.

After the Gaussian noise level estimation routine is activated to find the Gaussian noise level of an image, the nearest corresponding filter in the universal MFRB filter is selected to remove the remained noise of the image processed by the first stage.

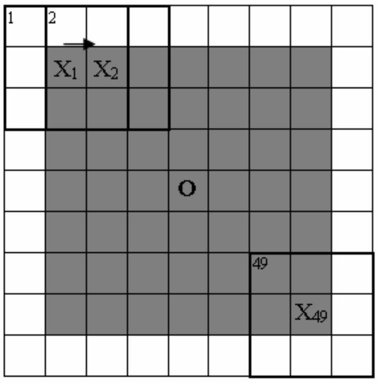


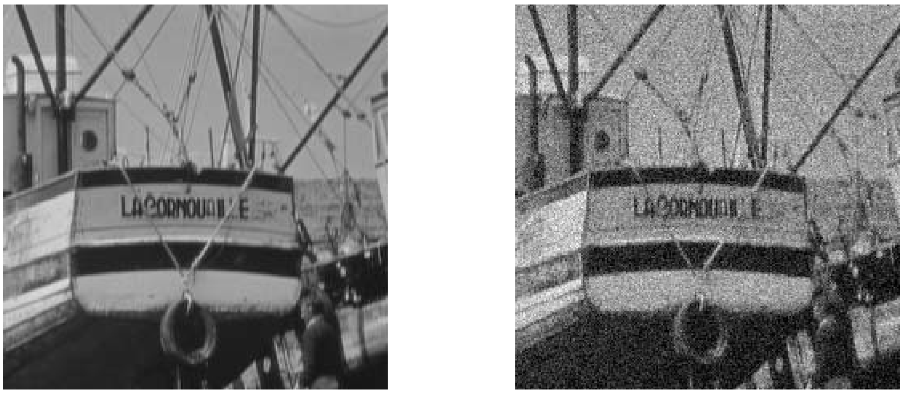
Fig. 4.4. Overlapping 7×7 sliding window for local variance estimation, based on 3×3 sub-area.

4.3. Experimental Results

For performance comparison, the proposed technique is evaluated and compared with other post-processing schemes, which include 3×3 and 5×5 adaptive Wiener filter [27, pp. 356−360] , FWLS by Choi *et al.* [47], MFF by Taguchi [48], FLF by Farbiz *et al.* [49], and PBM filter by Chen *et al.* [21] and trilateral filter by Garnett *et. al* [31]. For the trilateral, we varied its designing parameters and iteration number exhaustively to obtain the best result in each test. The simulations are based on a set of 256 level grayscale images of size 512×512 pixels. For performance evaluation, the peak signal-to-noise ratio (PSNR) is used as a quantitative indication. The proposed algorithm has been extensively tested on various simulated images. Four of them, “Boats”, “Bridge”, “Goldhill”, and “Lena”, are given as examples for brevity. In each performance comparison table, we rank the top three best-performed algorithms by superscript for clarity. We divide our experiments into three parts to demonstrate the capabilities of our proposed algorithm. First, we test the effectiveness of our second stage filter, the MFRB filter, to remove the Gaussian noise in images. Secondly, images contaminated with mixed noise, assume the Gaussian noise level is known, are processed to verify the proposed two-stage MFRB filtering method. At last, images corrupted with real-world mixed noise, assume unknown noise levels, are processed by the proposed two-stage universal MFRB scheme.

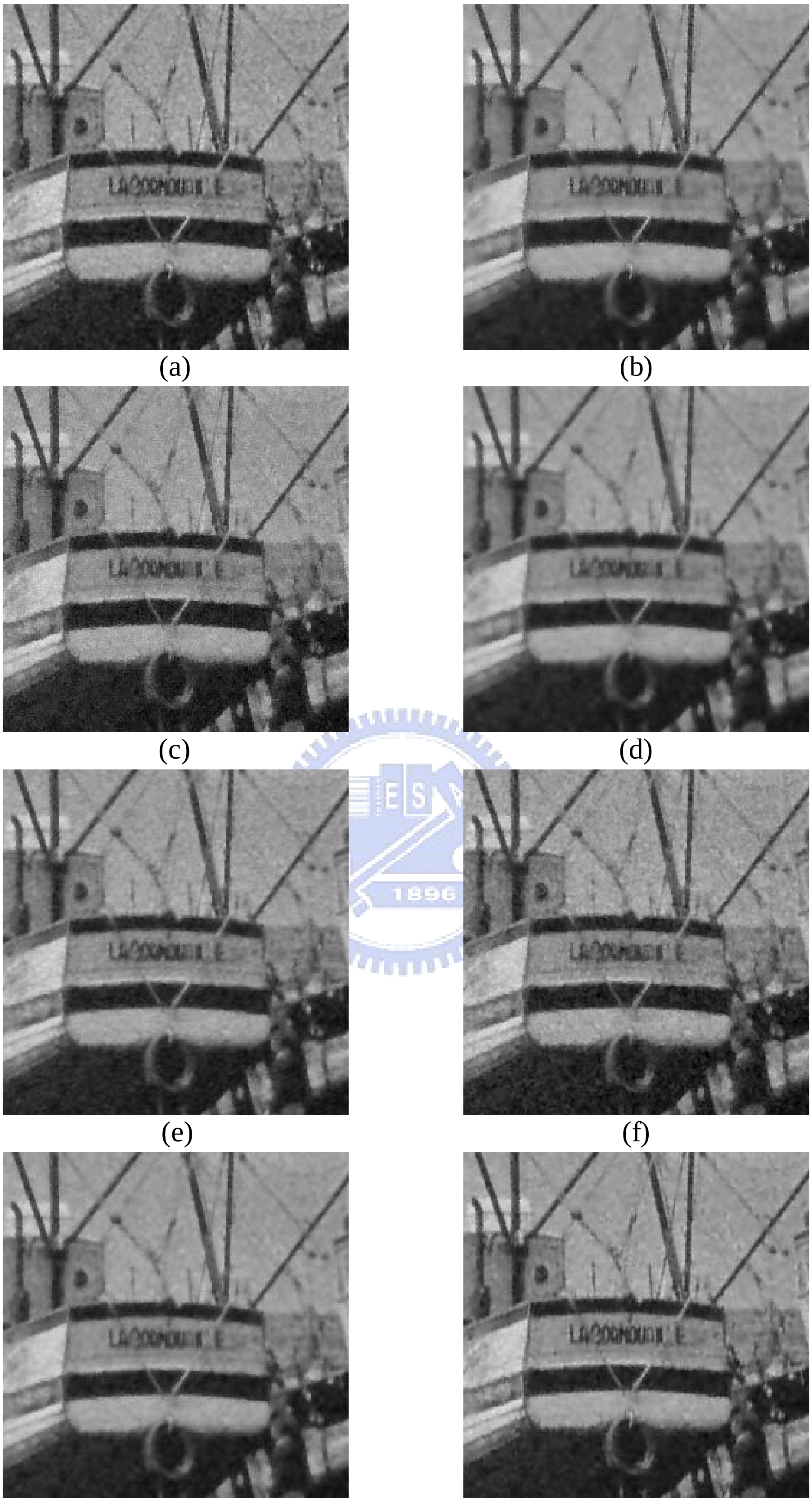
4.3.1. Gaussian Noise (Only) Case

Table 4.1 shows the different filter scheme results for four images corrupted with Gaussian noise of $\sigma = 20$. Only the filtered “Boats” images are shown in Fig. 4.5 for brevity. From Table 4.1, the proposed MFRB algorithm performs the best among the four test images. According to Fig. 4.6(a) and (b), the 5×5 Wiener filter removes Gaussian noise more efficiently than the 3×3 Wiener filter; however, it blurs the edge and induces some extra noise. In Fig. 4.6(c), FWLS [47] does not blur the edge but produces some additional noise in the higher level Gaussian corrupted images. As shown in Fig. 4.6(d) and (e), the MFF [48] and FLF [49] are similar in the PSNR metric but perform differently in the trade-off between the noise removal and edge preservation. Fig. 4.6(d) is good at noise removal while Fig. 4.6(e) is good at edge and details preservation. In Fig. 4.6(f), the PBM [21] does not blur the edge and makes no extra noise, but cannot remove the Gaussian noise completely. In Fig. 4.6(g), the Trilateral Filter [31] can remove the Gaussian noise very well, but it will blur some edge and detail information such as the words on the poop. Our proposed MFRB filter performs excellently both in removing the Gaussian noise and in preserving the edge and details in the image.



(a) (b)

Fig. 4.5. (a) Original “Boats” image; (b) Noisy image of “Boats” with Gaussian noise only of σ =20.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 4.6. Filtered “Boats” images of Fig. 4.5(b) using (a) 3×3 Wiener filter; (b) 5×5 Wiener filter; (c) FWLS [47]; (d) MFF [48]; (e) FLF [49]; (f) PBM [21]; (g) Trilateral Filter [31]; (h) MFRB.

Table 4.1 Comparative Results of PSNR in the Cases of Corruption by Gaussian Noise (σ = 20)

	Boats	Bridge	Goldhill	Lena
Win 3	28.62	26.25^{2}	28.21	28.91
Win 5	28.97^{3}	25.04	28.51^{3}	29.95
FWLS [47]	26.59	24.99	26.46	26.88
MFF [48]	28.95	25.04	28.47	30.12^{3}
FLF [49]	28.87	25.48	28.44	29.63
PBM [21]	27.20	25.46	26.97	27.69
Trilateral Filter [31]	29.56^{2}	25.92^{3}	29.07^{2}	30.41^{2}
MFRB	30.13^{1}	26.40^{1}	29.45^{1}	30.98^{1}

4.3.2. Mixed Noise Case of Known Gaussian Noise Intensities

For general assessment, we experimented the proposed algorithm with various combinations of impulse noise densities (p = 10%, 20%, and 30%) and Gaussian noises (σ = 10, 15, and 20). We individually apply the decision-based recursive adaptive median filter described in Section 3.2 to remove the Salt-Pepper noise and the second step - image quality enhancement system described in Section 3.4, and then apply the MFRB filter with corresponding noise level to remove the Gaussian noise. To be brief, we show the results of the two-stage filter schemes in Tables 4.2, 4.3, and 4.4 in which the image corrupted noise increases gradually. Wiener filters are also combined with the F*K*-NN [72] to remove the impulsive noise in the first stage. This two-stage 3×3 Wiener filter is good at removing low Gaussian variance noisy images as seen from Tables 4.2, and 4.3 while two-stage 5×5 Wiener is good at removing high Gaussian variance ones as shown in Table 4.4. FWLS [47] and FLF [49] decay abruptly in PSNR metric in the cases of high impulse rate noisy images. On the contrary, MFF [48], PBM [21], Trilateral filter [31], and our proposed algorithm are constantly stable in all the noise level combination. Our proposed two-stage MFRB filters can sequentially remove the impulse and Gaussian noises well and perform the best among them. Most of the algorithms are inadequate at processing the mixed noise removal except our two-stage filtering schemes. It verifies our algorithmic approach that the nonlinear filter described in Chapter 3 first suppresses the nonlinear type impulse noise effectively, and then the remained linear type Gaussian noise is efficiently removed by the use of linear MFRB filter.

Table 4.2 Comparative Results of PSNR in the Cases of Corruption by Mixed Gaussian (σ = 10) and Impulse Noise (p = 10%)

	Boats	Bridge	Goldhill	Lena
Win 3+FKNN	32.21^{2}	27.51^{2}	31.28^{2}	33.03^{2}
Win 5+FKNN	30.82^{3}	25.55	29.93^{3}	32.15^{3}
FWLS [47]	29.30	26.23^{3}	29.17	29.91
MFF [48]	28.96	24.75	28.41	30.30
FLF [49]	28.72	25.01	28.23	29.84
PBM [21]	29.71	25.97	29.35	30.5
Trilateral Filter [31]	30.52	25.77	29.81	31.08
First Stage	28.22	27.55	28.15	28.25
Our Two-Stage Method	33.42^{1}	29.12^{1}	32.34^{1}	33.69^{1}

Table 4.3 Comparative Results of PSNR in the Cases of Corruption by Mixed Gaussian (σ = 15) and Impulse Noise (p = 20%)

	Boats	Bridge	Goldhill	Lena
Win 3+FKNN	29.86^{2}	25.99^{2}	29.26^{2}	30.53^{2}
Win 5+FKNN	29.30^{3}	24.68	28.76^{3}	30.47^{3}
FWLS [47]	24.36	22.46	24.22	24.46
MFF [48]	27.57	23.90	27.26	28.44
FLF [49]	25.36	22.97	25.06	25.92
PBM [21]	27.49	24.38	27.21	28.12
Trilateral Filter [31]	29.22	24.84^{3}	28.73	29.27
First Stage	25.07	24.37	24.96	25.09
Our Two-Stage Method	30.73^{1}	26.53^{1}	29.73^{1}	31.44^{1}

Table 4.4 Comparative Results of PSNR in the Cases of Corruption by Mixed Gaussian (σ = 20) and Impulse Noise (p = 30%)

	Boats	Bridge	Goldhill	Lena
Win 3+FKNN	27.83^{3}	24.51^{2}	27.47^{3}	28.32^{3}
Win 5+FKNN	27.92^{2}	23.76^{3}	27.64^{2}	28.93^{2}
FWLS [47]	18.87	17.86	18.87	19.08
MFF [48]	24.99	22.44	25.05	25.66
FLF [49]	21.52	20.23	21.51	21.97
PBM [21]	25.46	22.83	25.38	25.97
Trilateral Filter [31]	26.58	23.32	26.58	27.15
First Stage	22.85	22.14	22.77	22.86
Our Two-Stage Method	28.98^{1}	24.78^{1}	28.33^{1}	29.81^{1}

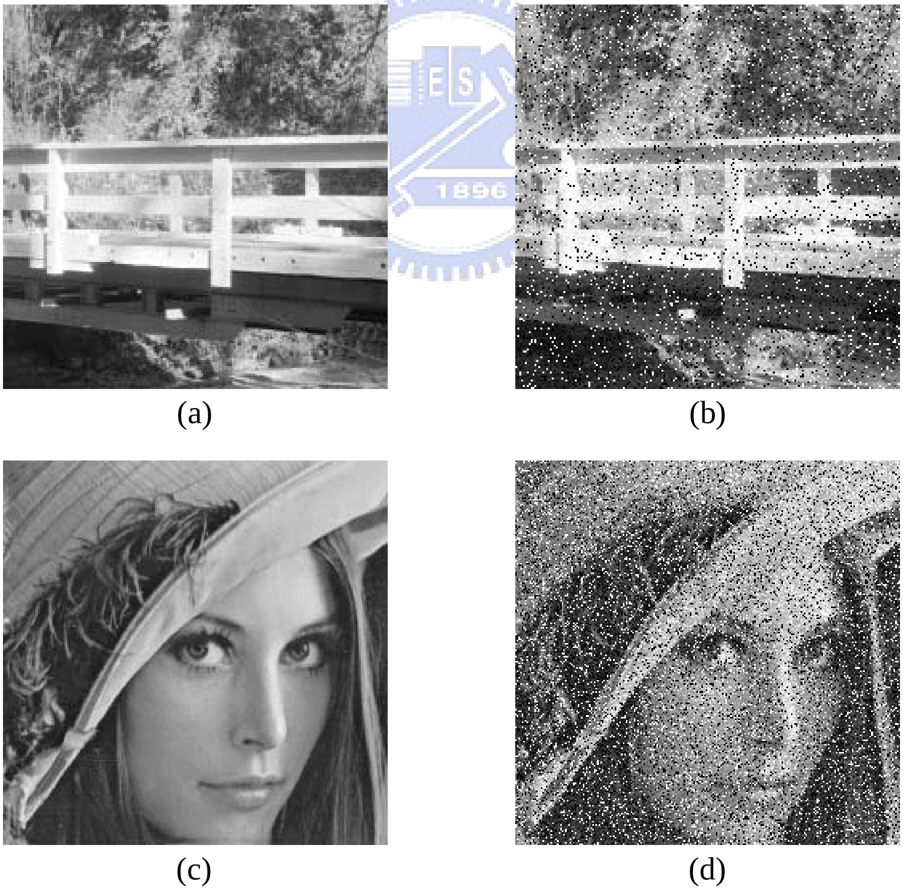

(a) (b) (c) (d)

Fig. 4.7 (a) Original “Bridge” image; (b) Noisy image of “Bridge” with 10% impulses and Gaussian noise of σ =10; (c) Original “Lena” image; (d) Noisy image of “Lena” with 20% impulses and Gaussian noise of σ =15.

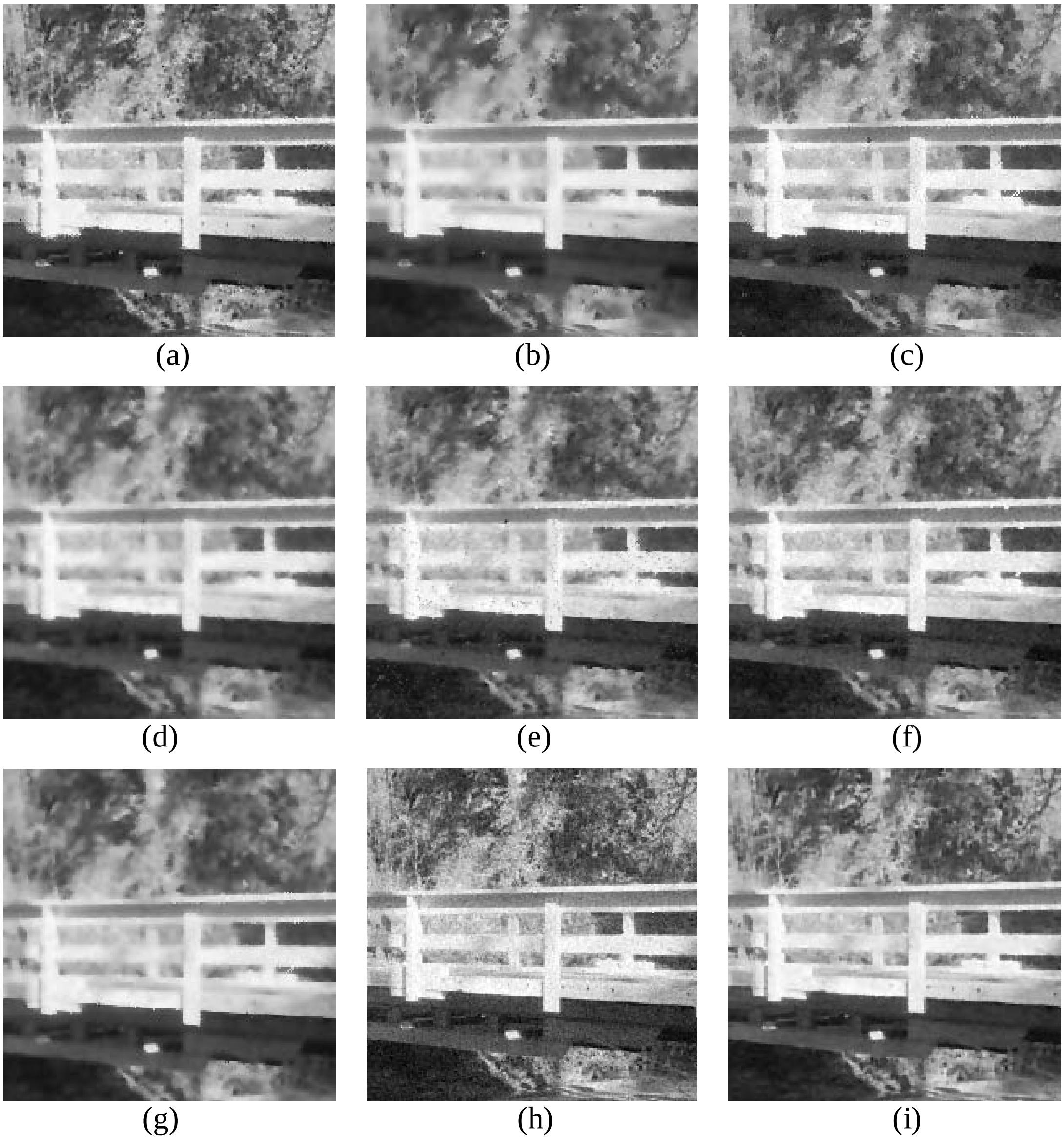


Fig. 4.8. Filtered “Bridge” images of Fig. 4.7(b) using (a) F*K*-NN with 3×3 Wiener filter; (b) F*K*-NN with 5×5 Wiener filter; (c) FWLS [47]; (d) MFF [48]; (e) FLF [49]; (f) PBM [21]; (g) Trilateral Filter [31]; (h) First Stage; and (i) Our Two-Stage Method.

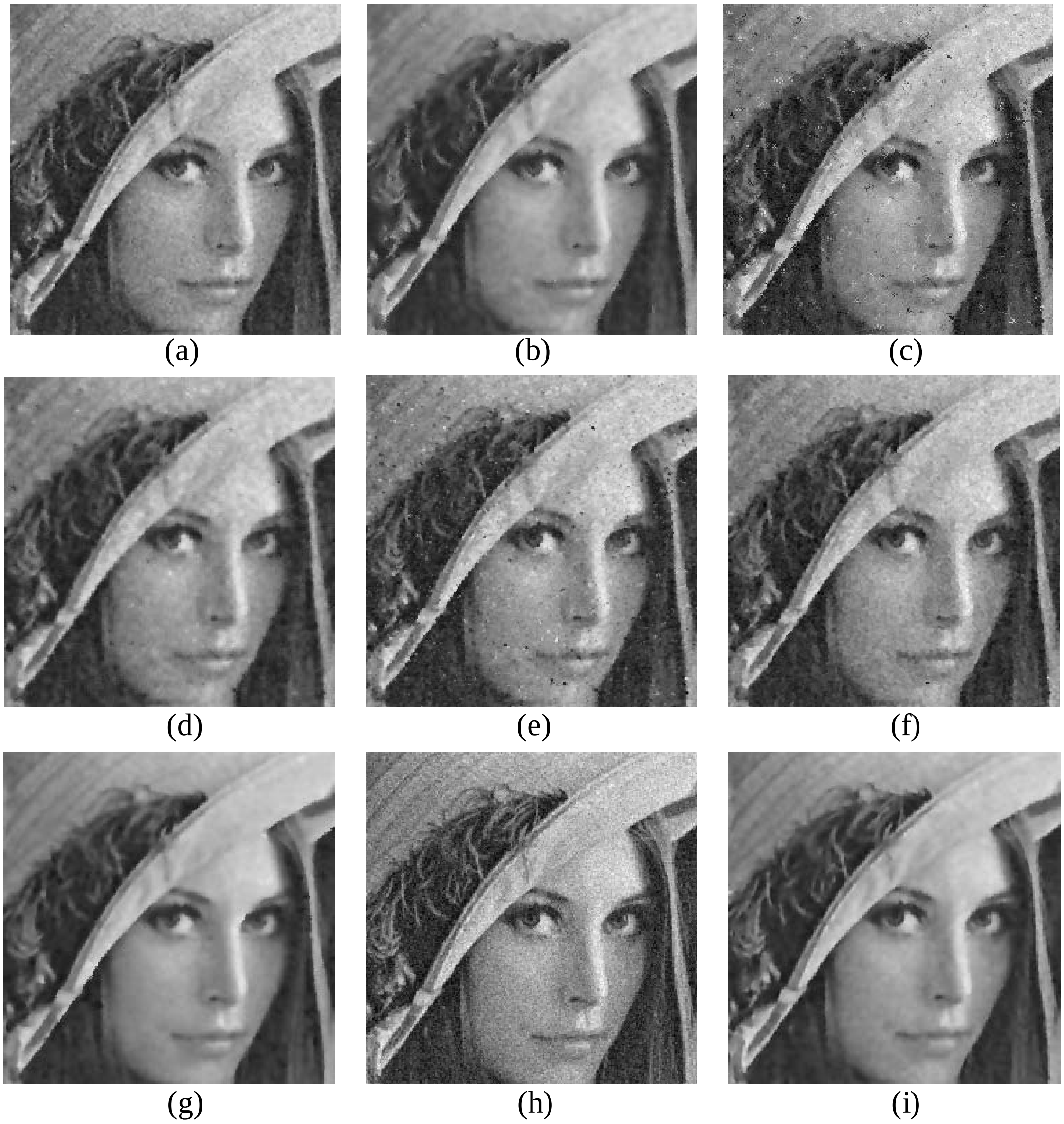
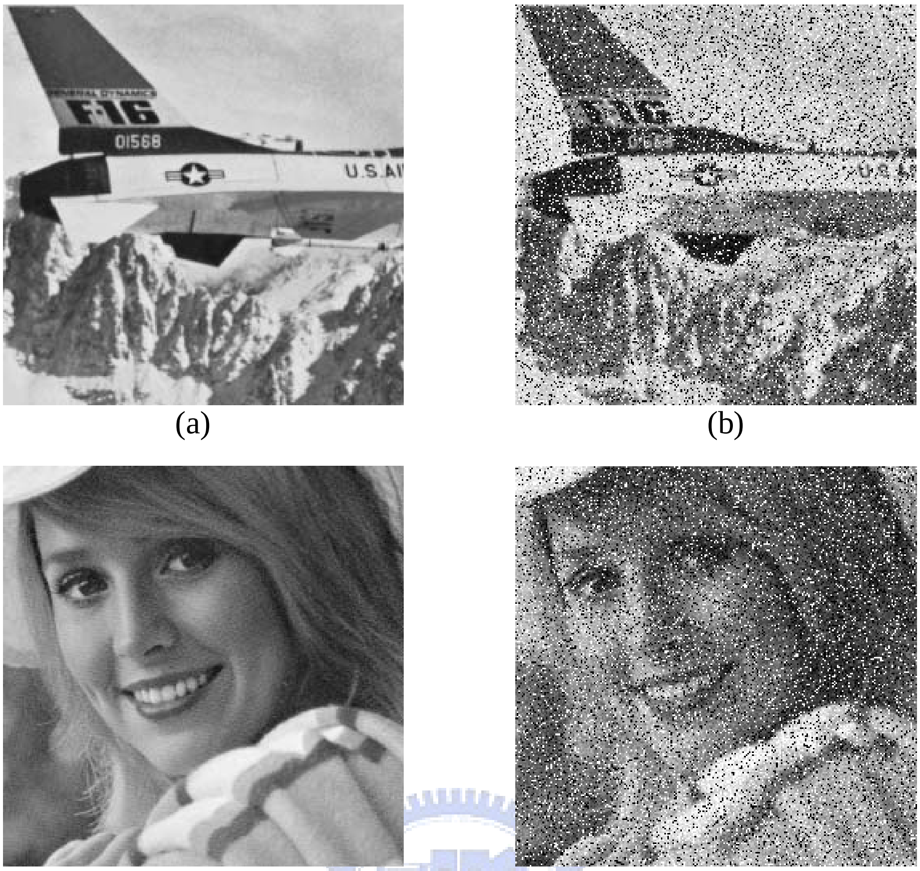


Fig. 4.9. Filtered “Lena” images of Fig. 4.7(d) using (a) F*K*-NN with 3×3 Wiener filter; (b) F*K*-NN with 5×5 Wiener filter; (c) FWLS [47]; (d) MFF [48]; (e) FLF [49]; (f) PBM [21]; (g) Trilateral Filter [31]; (h) First Stage; and (i) Our Two-Stage Method.

4.3.3. Blind Testing Mixed Noise Case

We first test our proposed standard deviation estimation routine by several typical images, each contaminated with various Gaussian noises including σ =10, 20, and 30. Table 4.5 shows the results of the Gaussian noise level estimation routine. From Table 4.5, we can find that most estimated standard deviations fit in with the Gaussian standard deviation values added. Our Gaussian noise estimation routine performs well for images of normal complexity; it overestimates the noise level for complex image like “Baboon,” especially for low noise level case. It can be adequately employed to estimate the standard deviation of residual Gaussian noise after the first-stage impulse noise removal filter.

To demonstrate the effectiveness of the proposed algorithm in the blind testing situation, Table 4.6 summarizes the filtered results by various post-processing schemes, for six images corrupted with unknown impulse noise rates and Gaussian levels as the case of practical application. Our two-stage universal MFRB with Gaussian noise intensity estimation routine still performs the best and obtains pleased resulting images. And Figs. 4.11 and 4.12 show the first two resulting images for brevity. As can be expected, the qualitative effectiveness of visual respect by various methods goes similarly as the way described in Section 4.3.2 above. From these figures and table, we can see that the proposed two-stage universal MFRB filter approach provides a practicable way to remove the mixed noise and produces very good restoration results, in comparison to other methods both in metric measurement and visual quality perception.



(a) (b)

(c) (d)

Fig. 4.10. The corrupted images (a) Original “Airplane” image; (b) Airplane with 20% impulses and Gaussian noise of $\sigma = 13$; (c) Original “Elaine” image; (d) Elaine with 17% impulses and Gaussian noise of $\sigma = 18$.

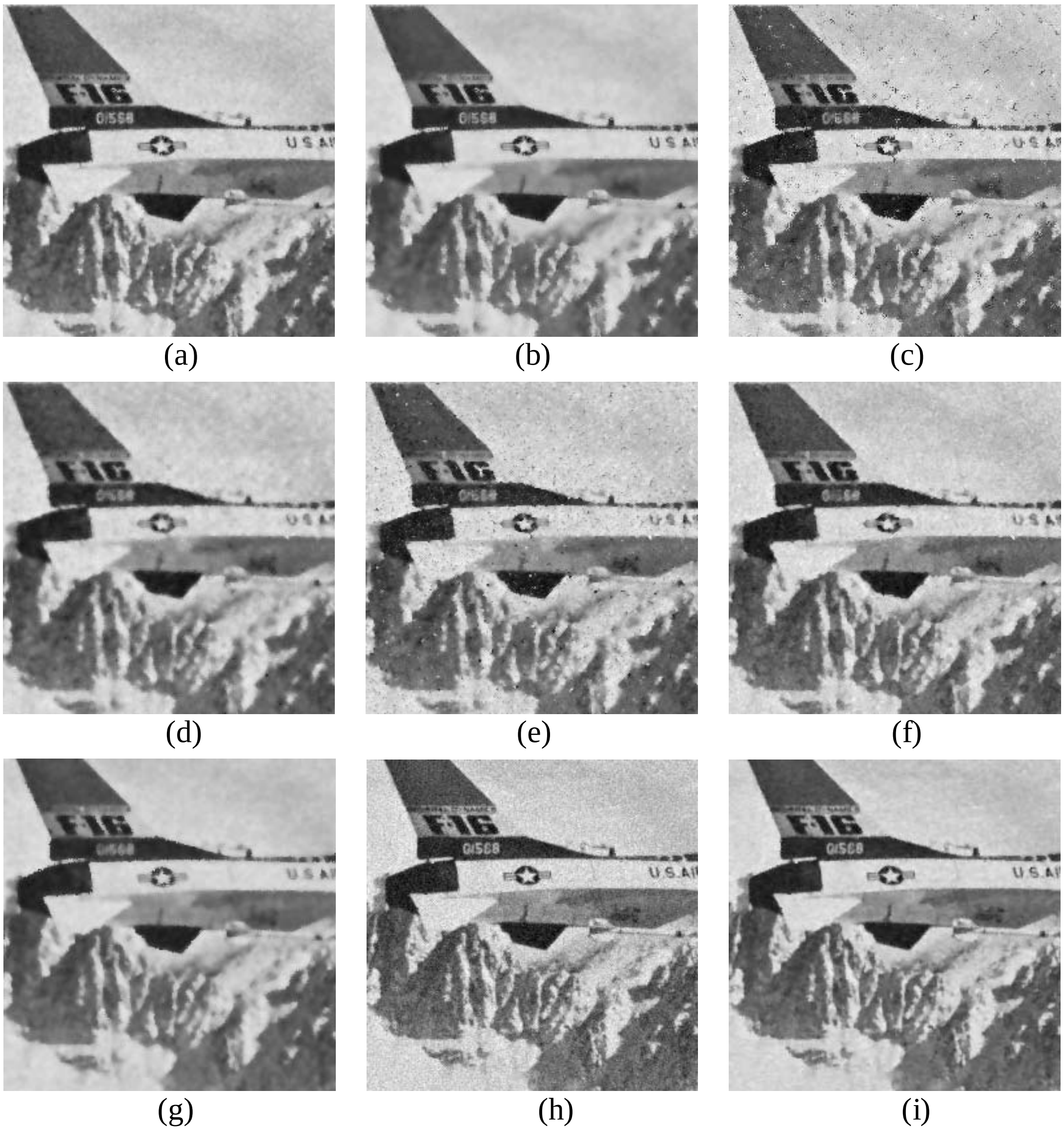
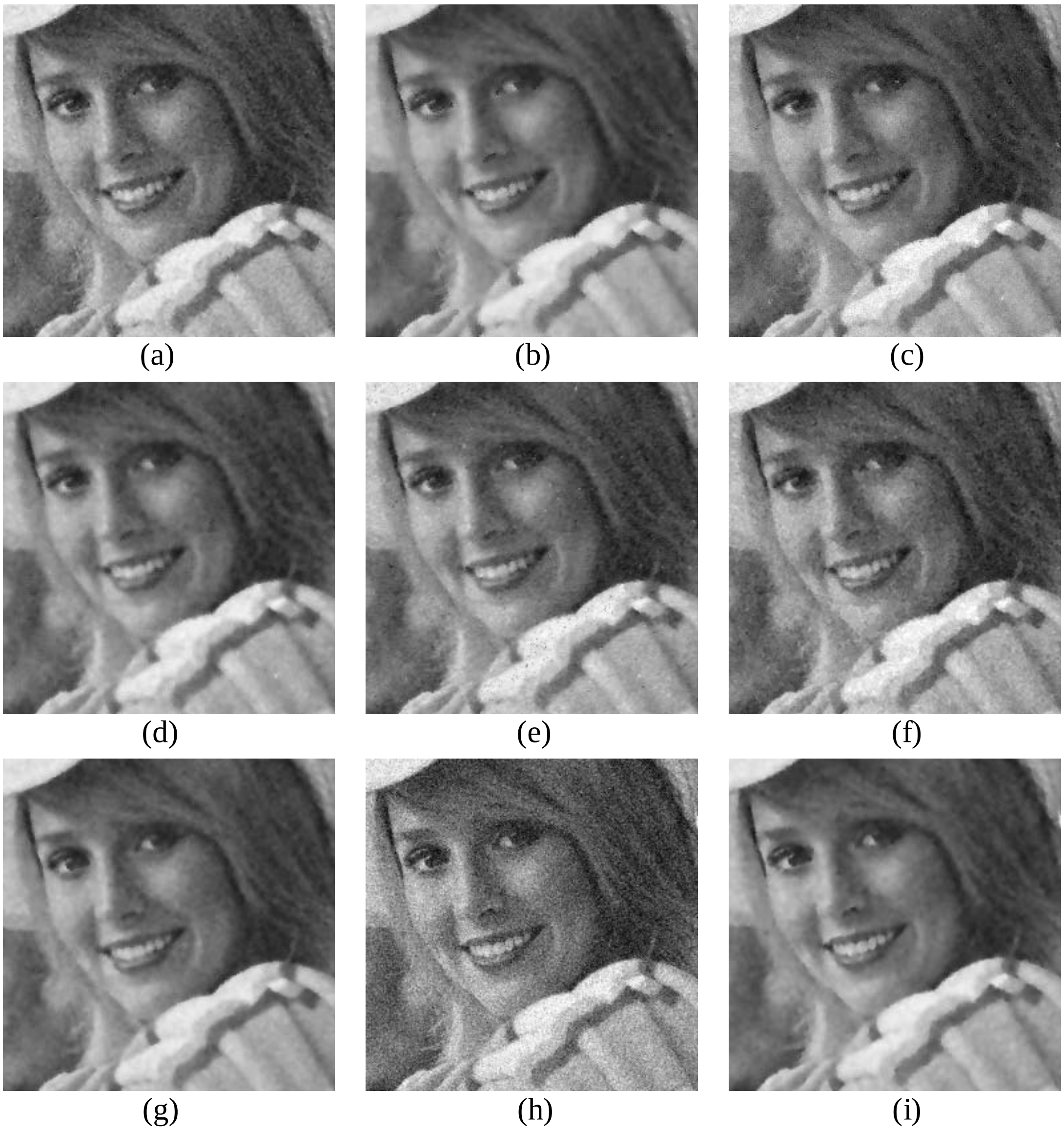


Fig. 4.11. Filtered “Airplane” images of Fig. 4.10(b) using (a) F*K*-NN with 3×3 Wiener filter; (b) F*K*-NN with 5×5 Wiener filter; (c) FWLS [47]; (d) MFF [48]; (e) FLF [49]; (f) PBM [21]; (g) Trilateral Filter [31]; (h) First Stage; and (i) Our Two-Stage Method.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 4.12. Filtered “Elaine” images of Fig. 4.10(d) using (a) F*K*-NN with 3×3 Wiener filter; (b) F*K*-NN with 5×5 Wiener filter; (c) FWLS [47]; (d) MFF [48]; (e) FLF [49]; (f) PBM [21]; (g) Trilateral Filter [31]; (h) First Stage; and (i) Our Two-Stage Method.

Table 4.5 The Estimated Gaussian Noise STD of σ =10, 20, and 30.

Gaussian of Standard Deviation σ =10

	Airplane	Baboon	Elaine	House
Estimated Value	13.53	18.59	12.27	12.14
	Lena	Peppers	Sailboat	Tiffany
Estimated Value	13.02	12.57	12.64	11.21

Gaussian of Standard Deviation σ =20

	Airplane	Baboon	Elaine	House
Estimated Value	21.61	24.49	20.04	20.14
	Lena	Peppers	Sailboat	Tiffany
Estimated Value	20.47	19.82	21.98	17.71

Gaussian of Standard Deviation σ =30

	Airplane	Baboon	Elaine	House
Estimated Value	28.14	31.11	27.86	27.89
	Lena	Peppers	Sailboat	Tiffany
Estimated Value	27.71	26.72	27.81	28.65

Table 4.6 Comparative Results of PSNR in the Cases of Random Mixed Noise

	Airplane	Elaine	Peppers	Lena	Sailboat	House
Win 3+FKNN	30.88^{2}	29.05	25.34	32.18^{2}	26.95^{3}	23.98
Win 5+FKNN	30.05^{3}	29.91^{3}	26.55^{2}	31.21^{3}	26.83	24.63^{2}
FWLS [47]	23.74	26.37	23.49	20.02	24.32	18.67
MFF [48]	27.74	28.87	26.75^{2}	26.89	26.28	23.35
FLF [49]	24.95	25.64	24.55	22.88	25.29	21.05
PBM [21]	27.78	27.02	25.38	26.62	22.40	22.72
Trilateral Filter [31]	29.29	30.27^{2}	26.54	29.28	27.15^{2}	24.32^{3}
First Stage	26.25	23.49	20.07	29.93	21.71	18.76
Our Two-Stage Method	31.23^{1}	30.37^{1}	27.11^{1}	33.23^{1}	27.75^{1}	24.88^{1}

"Airplane" corrupted with Gaussian (σ = 13) and Impulse Noise (*p* = 20%)
"Elaine" corrupted with Gaussian (σ = 18) and Impulse Noise (*p* = 17%)
"Peppers" corrupted with Gaussian (σ = 27) and Impulse Noise (*p* = 15%)
"Lena" corrupted with Gaussian (σ = 8) and Impulse Noise (*p* = 30%)
"Sailboat" corrupted with Gaussian (σ = 22) and Impulse Noise (*p* = 12%)
"House" corrupted with Gaussian (σ = 33) and Impulse Noise (*p* = 25%)

4.4. Brief Summary

In this chapter, a modified fuzzy rule based method is proposed to remove the Gaussian noise. In addition, we design several sets of universal MFRB filters in correspondence to the estimated values of contaminated Gaussian noise variance in the image. We combine the Salt-Pepper noise removal algorithm described in Chapter 3, and the universal MFRB filter together to sequentially remove the mixed noise in an image. The generalization ability of the proposed method has been demonstrated by examining it on a variety of noise-corrupted testing images corrupted with various noise densities. According to the experiment results, the proposed method is superior, both quantitatively and visually, compared to several other techniques.

5. Conclusion and Perspectives

A new two-stage mixed noise removal scheme for images is proposed in this thesis. In the first stage, the decision-based recursive adaptive median filter is applied to remove the Salt-Pepper noise and an adaptive two-level neural network noise reduction procedure is applied to remove the random-valued noise, and keep the uncorrupted information well. Then the fuzzy decision rules inspired by human visual system (HVS) applied to compensate the blur of the edge and the destruction caused by median filter. In the second stage, the MFRB filter is validated to suppress the Gaussian noise and preserve the image details and structures very well. For practical application, we combine the proposed Salt-Pepper noise removal algorithm and universal MFRB filter, together with the Gaussian noise level estimation routine, to sequentially remove the mixed noise in an image. According to the experiment results, the proposed method is superior to the conventional methods in the perceptual image quality and it can provide a quite stable performance over a wide variety of images with various noise densities. The proposed two-stage filtering scheme has demonstrated the effectiveness and robustness, in comparison with other filters in mixed noise removal of images.

References

[1] T. A. Nodes and N. C. Gallagher, “Median Filters: Some Modifications and Their Properties,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP−30, pp. 739−746, Oct. 1982.

[2] G. R. Arce and R. J. Crinon, “Median Filters: Analysis for Two-Dimensional Recursively Filtered Signals,” in *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, pp. 20.11.1−20.11.4, 1984.

[3] D. Brownrigg, “The Weighted Median Filter,” *Commun. Assoc. Computer*, pp. 807−818, Mar. 1984.

[4] R. Yang and L. Yin, “Optimal Weighted Median Filtering Under Structural Constrains,” *IEEE Trans. Signal Processing*, vol. 43, no. 3, pp. 591−604, Mar. 1995.

[5] S. J. Ko and Y. H. Lee, “Center Weighted Median Filters and Their Applications to Image Enhancement,” *IEEE Trans. Circuits Syst.*, vol. 38, no. 9, pp. 984−993, Sep. 1991.

[6] H. M. Lin and A. N. Willson, “Median Filters with Adaptive Length,” *IEEE Trans. Circuits Syst.*, vol. 35, no. 6, pp. 675−690, June 1988.

[7] M. P. McLoughlin and G. R. Arce, “Deterministic Properties of the Recursive Separable Median Filter,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 98−106, 1987.

[8] G. Qiu, “An Improved Recursive Median Filtering Scheme for Image Processing,” *IEEE Trans. Image Processing*, vol. 5, no. 4, pp. 646−648, Apr. 1996.

[9] J. B. Bednar and T. L. Watt, “Alpha-trimmed Means and Their Relationship to Median Filters,” *IEEE Trans. Acoust., Speech, and Signal Processing*, Vol. ASSP−32, pp. 145−153, 1984.

[10] D. A. F. Florencio and R. W. Schafer, “Decision-Based Median Filter Using Local Signal Statistics,” in *Proc. SPIE Symp. Vis. Comm. Image Processing*, vol. 2308, pp. 268−275, 1994.

[11] H. Kong and L. Guan, “A Noise-Exclusive Adaptive Filtering Framework for Removing Impulse Noise in Digital Images,” *IEEE Trans. Circuits Syst. Ⅱ*, vol. 45, no. 3, pp. 422−428, Mar. 1998.

[12] G. Pok and J. C. Liu, “Decision-Based Median Filter Improved by Predictions,” in *Proc. IEEE Int. Conf. on Image Processing*, pp. 410−413, 1999.

[13] E. Abreu and S. K. Mitra, “A Signal-Dependent Rank Ordered Mean (SD-ROM) Filter – A New Approach for Removal of Impulses from Highly Corrupted Images,” in *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing*, vol. 4, pp. 2371−2374, May 1995.

[14] T. Chen, K. K. Ma, and L. H. Chen, “Tri-State Median Filter for Image De-noising,” *IEEE Trans. Image Processing*, vol. 8, no. 12, pp. 1834−1838, Dec. 1999.

[15] T. Chen and H. R. Wu, “Impulse Noise Removal by Multi-State Median Filtering,” in *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing*, vol. 4, pp. 2183−2186, Jun. 2000.

[16] H. L. Eng and K. K. Ma, “Noise Adaptive Soft-Swtching Median Filter,” *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 242−251, Feb. 2001.

[17] S. Zhang and M. A. Karim, “A New Impulse Detector for Switching Median Filters,” *IEEE Signal Processing Lett.*, vol. 9, no. 11, pp. 360−363, Nov. 2002.

[18] Z. Wang and D. Zhang, “Progressive Switching Median Filter for the Removal

of Impulse Noise from Highly Corrupted Images," *IEEE Trans. Circuits Syst. Ⅱ*, vol. 46, no. 1, pp. 78−80, Jan. 1999.

[19] C. T. Chen and L. G. Chen, "A Self-Adjusting Weighted Median Filter for Removing Impulse Noise in Image," in *Proc. IEEE Int. Conf. on Image Processing*, vol. 1, pp.419−422, Sept. 1996.

[20] G. R. Arce and J. L. Paredes, "Recursive Weighted Median Filters Admitting Negative Weights and Their Optimization," *IEEE Trans. Signal Processing*, vol. 48, no. 3, pp. 768−779, Mar. 2000.

[21] T. Chen and H. R. Wu, "Application of Partition-Based Median Type Filters for Suppressing Noise in Images," *IEEE Trans. Image Processing*, vol. 10, no. 6, pp. 829−836, June 2001.

[22] K. Kondo, M. Haseyama and H. Kitajima, "An Accurate Noise Detector for Image Restoration," in *Proc. IEEE Int. Conf. on Image Processing*, vol. 1, I-321 −I-324, Sep. 2002.

[23] I. Aizenberg, C. Butakoff, and D. Paliy, "Impulsive Noise Removal Using Threshold Boolean Filtering Based on the Impulse Detecting Functions," *IEEE Signal Processing Lett.*, vol. 12, no. 1, pp. 63−66, Jan. 2005.

[24] X. Li and M. Orchard, "True Edge-Preserving Filtering for Impulse Noise Removal," in *Proc. 34th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove CA, Oct. 2000.

[25] R. H. Chan, C. W. Ho, and M. Nikolova, "Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization," *IEEE Trans. Image Processing*, vol. 14, no. 10, pp. 1479−1485, 2005.

[26] T. Z. Lin and P. T. Yu, "Thresholding Noise-Free Ordered Mean Filter Based on Dempster-Shafer Theory for Image Restoration," *IEEE Trans. Syst., Man, Cybern. I*, vol. 53, no. 5, pp. 1057−1064, May 2006.

[27] J. S. Lim, *Two-Dimensional Signal and Image Processing*, Englewood Cliff, N. J: Prentice-Hall, 1990.

[28] J. W. Woods and C. H. Radewan, “Kalman Filtering in Two Dimensions,” *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 473−482, July 1977.

[29] L. Yin, and Y. Neuvo, “Adaptive FIR-WOS Filtering.” in *Proc. IEEE Int. Conf. on Symp. Circuits Syst.*, vol. 6, pp. 2637−2640, May 1992.

[30] A. Taguchi, M. Muneyasu, and T. Hinamoto, “Median and Neural Networks Hybrid Filters,” in *Proc. IEEE Int. Conf. on Neural Networks,* vol. 1, pp. 580−583, Dec. 1995.

[31] R. Garnett, T. Huegerich, C. Chui, and W. He, “A Universal Noise Removal Algorithm with an Impulse Detector,” *IEEE Trans. Image Processing*, vol. 14, no. 11, pp. 1747−1754, Nov. 2005.

[32] C. Tomasi and R. Manduchi, “Bilateral Filtering for Gray and Color Images,” in *Proc. IEEE Int. Conf. on Computer Vision*, pp. 839−846, 1998.

[33] D. Zhang and Z. Wang, "Impulse Noise Detection and Removal Using Fuzzy Techniques," *Electronics Lett.*, vol. 33, no. 5, pp. 378−379, Feb. 1997.

[34] S. Schulte, *et al.*, “A Fuzzy Impulse Noise Detection and Reduction Method,” *IEEE Trans. Image Processing*, vol. 15, no. 5, pp. 1153−1162, May 2006.

[35] C. S. Lee, S. M. Guo, and C. Y. Hsu, “Genetic-Based Fuzzy Image Filter and its Application to Image Processing, *IEEE Trans. Syst., Man and Cybern. B,* vol. 35, no. 4, pp. 694−711, 2005.

[36] F. Russo, “Noise Removal from Image Data Using Recursive Neurofuzzy Filter,” *IEEE Trans. Instrum. Meas.*, vol. 49, no. 2, pp. 307–314, Apr. 2000.

[37] F. Russo, “An Image Enhancement Technique Combining Sharpening and Noise Reduction,” *IEEE Trans. Instrum. Meas*, vol. 51, no. 4, pp. 824–828, Aug. 2002.

[38] M. E. Yüksel, and E. Beşdok, “A Simple Neuro-Fuzzy Impulse Detector for

Efficient Blur Reduction of Impulse Noise Removal Operators for Digital Images," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 6, Dec. 2004.

[39] H. S. Wong and L. Guan, "A Neural Learning Approach for Adaptive Image Restoration Using a Fuzzy Model-Based Network Architecture," *IEEE Trans. Neural Networks*, vol. 12, no. 3, May 2001.

[40] D. V. D. Ville, *et al.*, "Noise Reduction by Fuzzy Image Filtering," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 4, pp.429−436, Aug. 2003.

[41] C. Jing, Y. Jinsheng, and D. Runtao, "Fuzzy Weighted Average Filter," in *Proc. IEEE Int. Conf. on Signal Processing*, pp. 525−528, 2000.

[42] A. Taguchi, "A Design Method of Fuzzy Weighted Median Filters," in *Proc. IEEE Int. Conf. on Image Processing*, vol. 1, pp. 423−426, 1996.

[43] S. Peng and L. Lucke, "Fuzzy Filtering for Mixed Noise Removal During Image Processing," in *Proc. IEEE Int. Conf. on Fuzzy Systems*, vol. 1, pp. 89−93, June 1994.

[44] S. Peng and L. Lucke, "Multi-Level Adaptive Fuzzy Filter for Mixed Noise Removal," *IEEE Int. Symp. on Circuits Syst.*, vol. 2, pp. 1524−1527, May 1995.

[45] M. Muneyasu *et al.*, "An Edge-Preserving Fuzzy Filter Based on Differences Between Pixels," in *Proc. IEEE Int. Conf. on Circuits Syst.*, vol. 5, pp. 363−366, June 1999.

[46] M. Salmeri *et al.*, "Noise Estimation in Digital Images Using Fuzzy Processing," in *Proc. IEEE Int. Conf. on Image Processing*, vol. 1, pp. 517−520, Oct. 2001.

[47] Y. Choi and R. Krishnapuram, "A Robust Approach to Image Enhancement Based on Fuzzy Logic," *IEEE Trans. Image Processing*, vol. 6, no. 6, pp. 808−825, June 1997.

[48] A. Taguchi, "Removal of Mixed Noise by Using Fuzzy Rules," *Second Int. Con. Proc. on Knowledge-Based Intell. Electron. Syst.*, vol. 1, pp. 176−179, Apr. 1998.

[49] F. Farbiz, M. B. Menhaj, S. A. Motamedi, and M. T. Hagan, “A New Fuzzy Logic Filter for Image Enhancement,” *IEEE Trans. Syst. Man, and Cybern. B*, vol. 30, no.1, pp. 110−119, Feb. 2000.

[50] H. Qin and S. X. Yang, “Nonlinear Noise Cancellation for Image with Adaptive Neuro-Fuzzy Inference Systems,” *Electronics Lett.*, vol. 41, no. 8, pp. 474−475, Apr. 2005.

[51] J. F. C. Wanderley and M. H. Fisher, “Multiscale Color Invariants Based on the Human Visual System,” *IEEE Trans. Image Processing*, vol. 10, no. 11, Nov. 2001.

[52] S. H. Kim and J. P. Allebach, “Impact of HVS Models on Model-Based Halftoning,” *IEEE Trans. Image Processing*, vol. 11, no. 3, Mar. 2002.

[53] H. Lin and A. N. Venetsanapoulos, “Incorporating Human Visual System (HVS) Models into the Fractal Image Compression,” in *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing*, vol. 4, pp. 1950−1953, 1996.

[54] Y. Chee and K. Park, “Medical Image Compression Using the Characteristics of Human Visual System,” in *Proc. IEEE Int. Conf. on 16th Annual*, vol. 1, pp. 618−619, 1994.

[55] M. Bertran, J. F. Delaigle, and B. Macq, “Some Improvements to HVS Models for Finger-printing in Perceptual Decompressors,” in *Proc. IEEE Int. Conf. on Image Processing*, vol. 2, pp. 1039−1042, 2001.

[56] B. Azeddine, B. B. Kamel, and B. Abdelouahab, “Low-level vision treatments inspired from Human Visual System,” in *Proc. Fifth Int. Symp. Signal Processing and its applications, ISSPA ’99*, Brisbane, Australia, pp. 313−316, Aug. 1999.

[57] J. Johnston, N. Jayant and R. Safranek, “Signal Compression Based on Models of Human Perception,” in *Proc. IEEE*, pp. 1325−1422, Oct. 1993.

[58] M. S. Kankanhalli and K. R. Ramakrishnan, “Content Based Watermarking of Images,” in *Proc. of the 6th ACM International Multimedia Conference*, Bristol, UK, pp. 61−70, Sep. 1998.

[59] E. Izquierdo, “Using Invariant Image Feature for Synchronization in Spread Spectrum Image Watermarking,” *Journal on Applied Signal Processing*, pp. 410−417, 2002.

[60] C. H. Chou and Y. C. Li, “A Perceptually Tuned Subband Image Coder Based on the Measure of Just-Noticeable-Distortion Profile,” *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp.467−476, Dec. 1995.

[61] A. P. Bradley, “A Wavelet Visible Difference Predictor,” *IEEE Trans. Image Processing*, vol. 8, no. 5, May 1999.

[62] J. Malo *et al.*, “Perceptual Feedback in Multigrid Motion Estimation Using an Improved DCT Quantization,” *IEEE Trans. Image Processing*, vol. 10, no. 10, Oct. 2001.

[63] http://www.dmmd.net/research/imgprocessing/saltpepper_denoissing.htm

[64] E. S. Hore, B. Qiu and H. R. Wu, “Adaptive Noise Detection for Image Restoration with a Multiple Window Configuration,” in *Proc. IEEE Int. Conf. on Image Processing*, vol. 1, pp. 329−334, Sep. 2002.

[65] D. D. Muresan and T. W. Parks, “Adaptive Principal Components and Image Denoising,” in *Proc. IEEE Int. Conf. on Image Processing*, vol. 1, pp. 14−17, Sep. 2003.

[66] K. Arakawa, “A Nonlinear Digital Filter Using Fuzzy Clustering,” in *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal Processing*, pp. IV309−IV312, Mar. 1992.

[67] K. Arakawa, “Fuzzy Rule-Based Signal Processing and its Application to Image Restoration,” *IEEE J. Select. Areas Commun.*, vol. 12, no. 9, pp. 1495−1502,

Dec. 1994.

[68] B. Widrow and M. A. Lehr. “30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation.” in *Proc. IEEE*, vol. 78, pp. 1415−1442, Sep. 1990.

[69] H. L. Van Trees, *Detection Estimation, and Modulation Theory*, New York: Wiley, 1968.

[70] X. Li and M. T. Orchard, “Edge-Directed Projection for Lossless Compression of Natural Images,” *IEEE Trans. Image Processing*, vol. 10, no. 6, pp. 813−817, June 2001.

[71] J. S. Lee, “Refined Filtering of Image Noise Using Local Statistics,” *Comput. Graphics Image Proc.*, vol. 15, pp. 380−389, 1981.

[72] J. Y. Chang and J. L. Chen, “Classifier-augmented median filters for image restoration,” *IEEE Trans. Instrum. Meas.*, vol. 53, pp. 1415−1442, Apr. 2004.

List of Publication

著作目錄

姓名: 盧世茂 (Shih-Mao Lu)

已刊登或被接受之期刊論文：(總共 5.6 點)

[1] J. Y. Chang, S. M. Lu, “Image Blocking Artifact Suppression by the Modified Fuzzy Rule Based Filter,” *International Journal of Fuzzy System*, Vol. 6, No. 2, pp. 81−89, June 2004. (2 點)

[2] S. M. Lu, S. F. Liang, and C. T. Lin, “A HVS-Directed Neural-Network-Based Approach for Salt-Pepper Impulse Noise Removal,” *Journal of Information Science and Engineering*, Vol. 22, No. 4, pp. 925−939, July 2006. (1.4 點)

[3] J. Y. Chang, C. W. Cho, and S. M. Lu, “A Fuzzy Integral Based Information Fusion for Drowsiness Detection,” *International Journal of Fuzzy System*, Vol. 7, No. 2, pp. 63−71, June 2005. (1 點)

[4] C. T. Lin, H. C. Pu, K. W. Fan, S. M. Lu and S. F. Liang, “A HVS-Directed Neural-Network-Based Image Resolution Enhancement Scheme for Image Resizing,” *IEEE Trans. Fuzzy Syst.* (Accepted on Jun. 2006) (1.2點)

研討會論文：

[1] J. Y. Chang, S. M. Lu, C. T. Lin, “A Two-Stage Fuzzy Filtering Method to Restore Images Contaminated by Mixed Impulse and Gaussian Noise,” *Proc. 2002 Tenth National Conference on Fuzzy Theory and Its Applications*.

[2] J. Y. Chang, S. M. Lu, “Image Blocking Artifact Suppression by the Modified Fuzzy Rule Based Filter,” *Proc. IEEE Int. Conf. on Syst. Man, and Cybern.*, Vol. 1, pp. 486−491, Oct. 5-8, 2003.

[3] S. M. Lu, H. C. Pu, and C. T. Lin, “A HVS-Directed Neural-Network-Based Approach for Impulse-Noise Removal from Highly Corrupted Images,” *Proc. IEEE Int. Conf. on Syst. Man, and Cybern.* , Vol. 1, pp. 72–77, Oct. 5-8, 2003.

[4] J. Y. Chang, C. W. Cho, and S. M. Lu, “A Fuzzy Integral Based Information Fusion for Drowsiness Detection,” *Proc. Int. Conf. on Neural Information Processing*, Oct 30 - Nov 2, 2005.

[5] J. Y. Chang, and S. M. Lu, “A Two-Stage Fuzzy Filtering Method to Restore Images Contaminated by Mixed Impulse and Gaussian Noises,” *Proc. Int. Conf. on Artificial Intelligence and Soft Computing*, Zakopane, Poland, June 25-29, 2006. (Lecture Note in Artificial Intelligence)

Vita

博士候選人學經歷資料

姓名: 盧世茂

性別: 男

生日: 中華民國 65 年 7 月 9 日

籍貫: 台灣省台南縣

論文題目:　　中文：基於人類視覺系統之混合雜訊消除技術

英文：Human-Visual-System-Based Mixed-Noise Removal Techniques

學歷:

1. 民國 87 年 6 月　國立清華大學動力機械工程系畢業。
2. 民國 89 年 6 月　國立交通大學電機及控制工程研究所畢業。
3. 民國 89 年 9 月　國立交通大學電機及控制工程研究所博士班。

經歷:

1. 民國 93 年至民國 94 年　大華技術學院兼任講師。