國立交通大學

電控工程研究所

碩士論文

近紅外線影像之非均勻校正與壞點修復

Non-Uniformity and Bad Pixel Correction for NIR image

研究生：彭彥凱

指導教授：張志永

中華民國一百零一年七月

近紅外線影像之非均勻校正與壞點修復

Non-Uniformity and Bad Pixel Correction for NIR image

學　　生：彭彥凱　　Student : Yen-Kai Peng

指導教授：張志永　　Advisor : Jyh-Yeong Chang

國立交通大學

電機工程學系

碩士論文

A Thesis

Submitted to Department of Electrical Engineering

College of Electrical and Computer Engineering

National Chiao-Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master in

Electrical and Control Engineering

July 2012

Hsinchu, Taiwan, Republic of China

中華民國一百零一年七月

近紅外線影像之非均勻校正與壞點修復

學生:彭彥凱　　　　　　指導教授: 張志永博士

國立交通大學電機與控制工程研究所

摘要

本論文使用非均勻校正(NUC)與壞點修正演算法修正近紅外線影像。在非均勻現象上，我們採用了 2 點式校正與最小平均平方法，此 2 種方法分別為非均勻校正中主要使用的 2 大類:基於參考平面與基於環境 2 種方式，兩點校正是一個高度精確的方法，不幸的是，他需要精密的儀器來測量參考圖像；最小平均平方法不需使用參考圖像，但演算法之速度難以使用在即時影像中，我們將測試並分析 2 種演算法。

為了提高壞點校正效能，我們使用改進基於同儕濾波器(peer group filter)之壞點修正法。在紅外線焦平面探測器所偵測影像中，經常出現團狀之壞點，我們必須執行一種特別的影像校正法。我們採用可自動調整濾波器遮罩大小來修正團狀之壞點。我們預設使用之濾波器遮罩大小為 3×3 之遮罩，可以盡可能的保持圖像清晰度；在遇到無法修正之壞點時，濾波器遮罩之大小將會自動增加，以提高修正能力。通過這項計劃，它是更準確地找出壞像素，將由同儕濾波器的中值取代壞點。

Non-Uniformity and Bad Pixel Correction for NIR image

STUDENT: Yen-Kai Peng ADVISOR: Dr. Jyh-Yeong Chang

Institute of Electrical and Control Engineering

National Chiao-Tung University

ABSTRACT

In this thesis, we use non-uniformity correction (NUC) and bad pixel correction to correct infrared image of a NIR sensor. We employ Two-point calibration and Least Mean Squares method, there are mainly used two categories of the calibration for non-uniformity correction, reference-based and scene-based correction algorithm. Two-point correction is a highly accurate method, unfortunately, it needs sophisticated instruments to measure the reference image; LMS method only need the readout infrared data captured by the imaging system and compensate the non-uniform response of pixels during its normal operation, But the speed of the algorithm is difficult to use in real-time video. We will test and analyze the two kinds of algorithms.

For bad pixel correction, in order to improve the performance of bad pixel correction, we have improved bad pixel correction which is based on the peer filter. Because bad pixels in the infrared images which are detected by infrared focal plane array sensor are frequently in blobs than kinds of images, we must implement a specific method of image correction. We employ adjustable window size that can increase the window automatically where bad pixels are in blobs. We use 3×3 window as default working window for sharpness maintenance, if the small window does not correct a bad pixel, the window size will increase automatically to enhance the correction capability. By this scheme, it is more accurate to locate bad pixel, and bad pixels will be replaced by the median of the peer group.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Jyh-Yeong Chang for valuable suggestions, guidance, support and inspiration he provided. Without his advice, it is impossible to complete this research. Thanks are also given to all of my lab members for their suggestion and discussion.

Finally, I would like to express my deepest gratitude to my family for their concern, supports and encouragements.

Contents

List of Figures

List of Tables

Chapter 1　Introduction

1.1 Motivation

With the rapid technological development, the visible light technology has become popular. Recently infrared sensing has also been widely applied in various domains of low/no light environments. One spectrum of infrared is near infrared (NIR), whose bandwidth is close to visible red light band, with a higher reflective image sensing capability under low/no light environments. In addition, NIR band also widely used in the military application providing high resolution images the sensing device of NIR is. It can be used as the enemy recognition systems and surveillance systems, in low/no light or fog or smoke in the environments.

It is well known that near infrared focal plane array (NIR FPA) has non-uniformity and bad pixels in the produced sensor cells. Hence, the infrared image must do non-uniformity correction (NUC) and bad pixel correction. Bad pixel is the pixel that does not respond (non responsive) i.e., dark situation (commonly) or always responsive i.e. In the NIR sensor bad pixel saturation is most often observed.

In the low light military applications, infrared image processing must be fast and efficient. Because military NIR sensor has to be lightweight and easy to used, there in NUC, the most popular reference-based correction method, so-called “two-point” correction method in which two uniform sources of known intensity are sequentially imaged [1], [2] is widely used.

Bad pixel replacement of infrared focal plane arrays is also known as Dead Pixel Correction. Bad pixels are non-responsive, permanently dark or saturating. Bad pixels in infrared digital images might still remain uncorrected after non-uniformity correction, so we need to correct bad pixels afterward. Because dead pixels in the infrared images are frequently in blobs than kinds of images, we must implement a specific method of image correction.

In this thesis, we use two-point correction to correct infrared images, non-uniformity and correct bad pixel. In the noise reduction scheme developed for better bad pixel detection, we propose a new impulse noise filter based on peer group concept. We employ adjustable window size that can increase the window automatically where bad pixels are in blobs. We use 3×3 window as default working window for sharpness maintenance, if the small window does not correct a bad pixel, the window size will increase automatically to enhance the correction capability. By this scheme, it is more accurate to locate bad pixel, and bad pixels will be replaced by the median of the peer group.

1.2 Non-uniformity Correction

With the development of infrared imaging technology, near infrared focal plane arrays (NIR FPA) imaging system has become the focus the next generation infrared imaging system. Compared with other thermal imaging systems, NIR FPA has simple structure, high reliability, high detection sensitivity and high frame rate, The NIR FPA is widely applied to various fields of military, medical, civil, and forest fire prevention.

Unfortunately, due to the limitations of semiconductor materials and process conditions, the output response of the detector is not the same, which resulted in the NIR FPA response non-uniformity. In general, the non-uniformity is called fixed pattern noise (FPN) will be striped or grid-like noise model. Therefore, how to effectively track and remove the device non-uniformity, non-uniformity correction (NUC) is the key to improve the NIR FPA imaging quality.

There are several calibration methods for the NUC of an NIR FPA. In general, there are two categories of the calibration methods, reference-based and scene-based correction algorithm. Reference-Based (or calibration-based) NUC techniques are based on the use of uniform infrared sources. The most used one is the Two-Point Calibration method [3], which employs at least two blackbody sources at different luminance to calculate the gain and the offset of each detector on the NIR FPA. Unfortunately, when the system is in use of increased working hours, its performance would be decreased for the working environment may change, Correction parameters which were measured before cannot meet the correct situation. Such kinds of Reference-Based NUC methods require to halt the operation of the system, and re-do the procedure and re-set the correction parameters to operate again.

For these reasons, Scene-Based NUC techniques are actually becoming more popular, since they only need the readout infrared data captured by the imaging system and compensate the non-uniform response of pixels during its normal operation. The constant statistics constraint method is the most referred scene-based technique However, its algorithm structure is complex, hardware implementation is difficult, thus reduces its engineering applications.

In this thesis, we propose to utilize two-point correction and adaptive scene-based NUC method [4] to correct infrared image that has non-uniformity. We also present varying-size impulse noise filter to correct bad pixels, and NIR sensor flowchart is illustrated in Fig. 1.1 below.

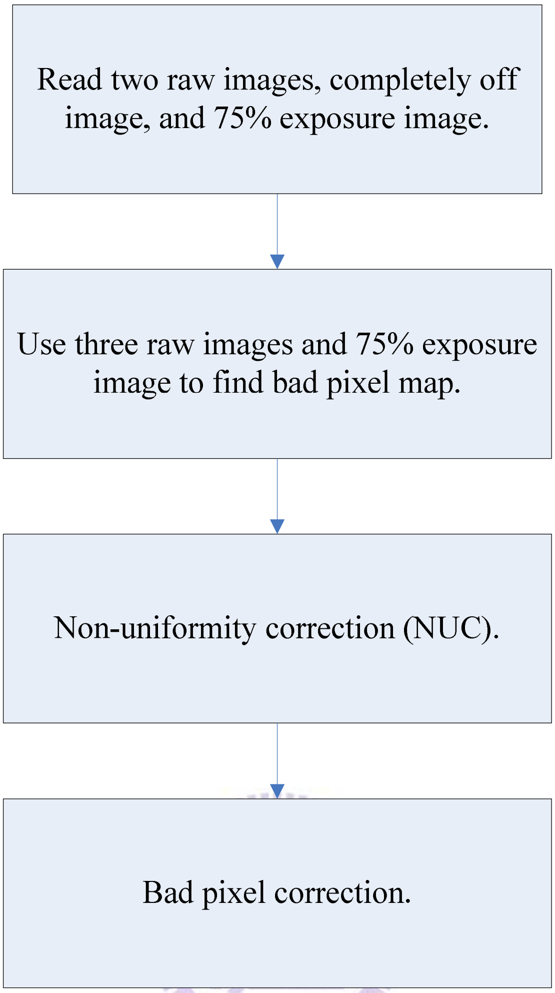

Read two raw images, completely off image, and 75% exposure image.
Use three raw images and 75% exposure image to find bad pixel map.
Non-uniformity correction (NUC).
Bad pixel correction.

Fig 1.1 The flowchart of our NIR sensor.

1.3 Bad Pixel Correction

In many practical situations, the sensing devices and the transmission process tend to degrade the quality of the digital images by introducing noise, images are corrupted by the so-called impulsive noise of short duration and high energy. The presence of noise in an image may be a drawback in any subsequent processing to be done over the noisy image such as edge detection, image segmentation or pattern recognition. As a consequence, filtering the image to reduce the noise without degrading its quality, preserving edges, corners and other details is a major step in imaging systems such as image content retrieval, medical image processing, industrial visual inspection [5]. This type of noise occurs mostly in the over-the-air transmission such as in standard broadcasting and satellite transmission. Common sources of impulse noise include lightening, industrial machines, car starters, faulty or dusty insulation of high-voltage powerlines and various unprotected electric switches [6–8].

In order to recovery the original image pixel values, the vector median filter (VMF) [9], which is probably the most well-known vector filter, uses the L_1 (City-Block) or L_2 (Euclidean) norm to define the above distance function. The filtering method sorts pixels vectors in the working window by space vector distance sum. On this basis, the Basic Vector Directional Filtering (BVDF) [10] sorts color vector by vector angle sum. Distance Directional Filtering (DDF) [11] sorts color vector by product of vector distance and vector angle. The above methods are too much smoothing, which results in an extensive blurring of the output image. This undesired property is caused by the unnecessary filtering of the noise-free samples that should be passed to a filter output without any change. To remove this drawback, a switching mechanism has been introduced into the structure of the robust smoothing filters, [12,13]. Such a switching filter detects if the pixel under consideration is affected by the noise process and if it is found to be noisy, then it is being replaced by the output of some robust filter, otherwise it is left unchanged. For example, Adaptive center-Weighted vector directional filter (ACWVDF) [14], and robust switching vector median filtering (RSVMF) [15]. When the noise ratio is low, the class switch-type methods have achieved good results.

In this thesis, we use improved peer group filter to correct infrared image that has non-uniformity and correct bad pixel.

1.4 Thesis Outline

The thesis is organized as follows. The basic concepts and technique concerning the NUC introduced in Chapter 2. The basic concepts and technique concerning the bad pixel correction are described in Chapter 3. In Chapter 4, the results of our NIR methods which are introduced in Chapter 2 and Chapter 3 are shown and compared. At last, we conclude this thesis with a discussion in Chapter 5.

Chapter 2 Non-uniformity Correction

There are two calibration methods for the NUC of an NIR FPA, reference-based and scene-based correction algorithm. We will introduce one of reference-based correction algorithm, two-point correction, and one of reference-based correction algorithm, least mean squares algorithm.

2.1 Two-Point Correction

One of the most developed methods is the two-point calibration method [3], which is earlier for the NUC in infrared imaging systems. In the method, it assumes that the response of the detector is linear in range of illumination, and the response of the detector is more stable and less affected by random noise. Non-uniformity can be said for the results of the multiplicative noise and additive noise. At this point, the response model of the detection unit *ij*-th can be expressed as:

$$\begin{cases} P_1(i,j) = \alpha_{ij} T_1(i,j) + \beta_{ij} \\ P_2(i,j) = \alpha_{ij} T_2(i,j) + \beta_{ij} \end{cases} \tag{2.1}$$

where $P_1(i,j)$ and $P_2(i,j)$ represent the *ij*-th pixel value in completely off image and 75% exposure image, respectively. The α_{ij} and β_{ij} represent gain and offset of the *ij*-th pixel. The T_1 and T_2 represent illumination level. In order to calculate the values of the gain and offset , we chose two illumination level. By using Eq. (2.1), we can assume $T_2(i,j) = kT_1(i,j)$, then:

$$\alpha_{ij} = \frac{P_2(i,j) - P_1(i,j)}{(\mathrm{k}-1)T_1(i,j)} \tag{2.2}$$

$$\beta_{ij} = P_1^i(i,j) - \frac{P_2(i,j) - P_1(i,j)}{(\mathrm{k}-1)} \tag{2.3}$$

We hope that all the output values of the detector are the same after correction, so we calculate the averages of the completely off image and 75% exposure image, the results are shown in Figs. 2.1(a)−(b) as $\overline{P_1}$ and $\overline{P_2}$, respectively. In other word, we correct all the output as:

$$\begin{cases} \overline{P_1} = \alpha_{ij} A_{ij} T_1(i,j) + \beta_{ij} B_{ij} \\ \overline{P_2} = \alpha_{ij} A_{ij} T_2(i,j) + \beta_{ij} B_{ij} \end{cases} \tag{2.4}$$

and we combine the Eq. (2.4) with Eq. (2.2) and Eq. (2.3):

$$A_{ij} = \frac{\overline{P_2} - \overline{P_1}}{\alpha_{ij}(T_2 - T_1)} = \frac{\overline{P_2} - \overline{P_1}}{P_2(i,j) - P_1(i,j)} \tag{2.5}$$

$$B_{ij} = \frac{1}{\beta_{ij}}\left(\overline{P_1} - \alpha_{ij} A_{ij} T_1(i,j)\right) \tag{2.6}$$

Finally, we would like to correct infrared image which is under unknown illumination level T_x, then :

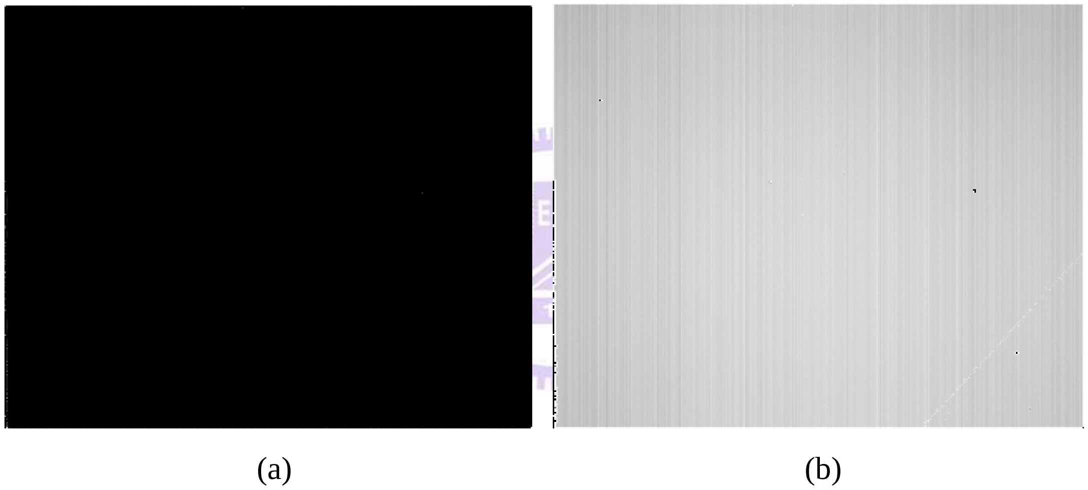
$$P_x(i,j) = \alpha_{ij} T_x(i,j) + \beta_{ij} \tag{2.7}$$

Consequently, corrected pixel output signal should be:

$$P_C(i,j) = \alpha_{ij} A_{ij} T_x(i,j) + \beta_{ij} B_{ij} = \alpha_{ij} A_{ij} T_x(i,j) + \overline{P}_1 - \alpha_{ij} A_{ij} T_1(i,j)$$

$$= \overline{P}_1 + \frac{\left(P_x(i,j) - P_1(i,j)\right)(\overline{P}_2 - \overline{P}_1)}{P_2(i,j) - P_1(i,j)} \tag{2.8}$$

Therefore, we can correct directly without any other illumination level that always needs reference image from the laboratory.


(a)

(b)

Fig. 2.1 (a) The completely off image; (b) The 75% exposure image.

2.2 Least Mean Squares

A commonly used bias-gain linear model for an FPA sensor is given by:

$$Y_{ij}(n) = a_{ij}(n) \times X_{ij}(n) + b_{ij}(n) \tag{2.9}$$

where $a_{ij}(n)$ and $b_{ij}(n)$ are the gain and the offset of the *ij*-th detector at frame *n*, $X_{ij}(n)$ is the real incident infrared radiation collected by the respective detector, and $Y_{ij}(n)$ is the measured output signal. The main idea of the NUC scene-based methods relies on estimating the gain and the offset parameters of each detector on the NIR FPA with only the readout data $Y_{ij}(n)$. The algorithm has the ability of adapting sensor's parameters over time under a frame by frame basis. To understand how the neural network based approach proposed, Eq. (2.9) must be reordered as following:

$$X_{ij}(n) = g_{ij}(n) \times Y_{ij}(n) + o_{ij}(n) \qquad (2.10)$$

where the new parameters $g_{ij}(n)$ and $o_{ij}(n)$ are related to the real gain and offset parameters of the detectors, as expressed in the following expression:

$$g_{ij}(n) = \frac{1}{a_{ij}(n)} \qquad o_{ij}(n) = -\frac{b_{ij}(n)}{a_{ij}(n)} \qquad (2.11)$$

In order to minimize some error functions that allow good estimations for the real infrared data $X_{ij}(n)$, the parameters $g_{ij}(n)$ and $o_{ij}(n)$ must be recursively updated.

Then, using linear regression to perform the parameter estimation, the error function $E_{ij}(n)$ for each neuron is usually defined as the difference between a desired target value $T_{ij}(n)$ and the estimated infrared data $\widehat{X}_{ij}(n)$. The function is expressed as:

$$E_{ij}(n) = T_{ij}(n) - \widehat{X}_{ij}(n) \tag{2.12}$$

The unknown parameters are estimated by using the neural network method, and the desired target value can be calculated as the local spatial average (mean filter) of the output data $\widehat{X}_{ij}(n)$.

Thus, to minimize the error $E_{ij}(n)$ in the mean squares error sense, a functional $J_{ij}(\mathrm{n})$ is defined as following:

$$J_{ij}(\mathrm{n}) = \sum_{\mathrm{n}} \left(E_{ij}(n)\right)^2 = \sum_{\mathrm{n}} \left(T_{ij}(n) - \widehat{X}_{ij}(n)\right)^2 \tag{2.13}$$

Then, we can get gradients relative to each parameter in Eq. (2.14).

$$\frac{\partial J_{ij}}{\partial \hat{g}_{ij}} = -2E_{ij} \times Y_{ij}$$

$$\frac{\partial J_{ij}}{\partial \hat{o}_{ij}} = -2E_{ij} \tag{2.14}$$

The steepest descent algorithm is a good way to solve this Least Mean Squares (LMS) optimization problem. In this gradient-based search algorithm, the parameters to be estimated are recursively updated with a portion of each respective error gradient. The parameter learning procedure is finally described as following:

$$\hat{g}_{ij}(n+1) = \hat{g}_{ij}(n) + \eta \times E_{ij}(n) \times Y_{ij}(n)$$

$$\hat{o}_{ij}\left(n+1\right)=\hat{o}_{ij}\left(n\right)+\eta\times E_{ij}\left(n\right) \tag{2.15}$$

where η is a fixed parameter known as the learning rate.

Basic LMS method for NUC of sensor array pixels is described above. Then we use three improvements on LMS method [4], which include regularization, momentum term and adaptive learning rate as explained below.

2.2.1. Regularization

To test the algorithm, we note that the gain is usually much larger than the offset, and gain value is usually around 1. When the gain value is much larger than the offset value, it is difficult to select the appropriate learning rate. In the LMS algorithm, when the learning rate is very small, the numerical convergence is slow; when the learning rate is large, the iteration would not converge. This makes the algorithm encounter an obstacle in practical applications. According to above analysis, the gain value was normalized to improve the calibration of artificial neural network algorithm. When the gain and offset are adjusted, you can get the same adjustment in the same order of magnitude. the issues of the difficulty in selecting the learning rate are resolved, and that also eliminates the obstacles of the LMS algorithm in practical applications. Thus, the purpose of regularization is to eliminate the difference in the magnitude of the gain and offset adjustment. Parameter adjustment is as following:

$$r\left(n\right)=\lambda\cdot\left(1-\frac{1}{NM}\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\hat{g}_{ij}\left(n\right)\right)\right) \tag{2.16}$$

$$
\begin{aligned}
\tilde{g}_{ij}(n) &= \hat{g}_{ij}(n) + r(n) \\
&= \hat{g}_{ij}(n) + \lambda \cdot \left(1 - \frac{1}{NM} \left(\sum_{i=1}^{N} \sum_{j=1}^{M} \hat{g}_{ij}(n) \right) \right)
\end{aligned}
\tag{2.17}
$$

where λ is the regularization constant, $N \times M$ is the number of pixels on the NIR FPA, and $\tilde{g}_{ij}(n)$ is the normalized gain.

2.2.2. Momentum

Another possible enhancement to the steepest descent algorithm is the well-known momentum. The gradient descent can be very slow of if the learning constant eta is small and can oscillate widely if eta is too large. This problem essentially results from error-surface valleys with steep sides but a shallow slope along the valley floor. Other efficient and commonly used method that allows larger learning constant without divergent oscillations occurring is the addition of a momentum term to the normal gradient-descent method. The idea is to give each weight some inertia or momentum so that it tends to change in the direction of the average downhill force that it feels. This scheme is implemented by giving a contribution from the previous time step to each weight change:

$$
\begin{aligned}
\hat{g}_{ij}(n+1) &= \hat{g}_{ij}(n) - \eta \times E_{ij}(n) \times Y_{ij}(n) + \alpha \cdot \left(\hat{g}_{ij}(n) - \hat{g}_{ij}(n-1) \right) \\
&= \hat{g}_{ij}(n) - \eta \times E_{ij}(n) \times Y_{ij}(n) + \alpha \cdot \Delta \hat{g}_{ij}(n)
\end{aligned}
\tag{2.18}
$$

$$\begin{aligned}\hat{o}_{ij}\left(n+1\right) &= \hat{o}_{ij}\left(n\right)-\eta\times E_{ij}\left(n\right)+\alpha\cdot\left(\hat{o}_{ij}\left(n\right)-\hat{o}_{ij}\left(n-1\right)\right)\\ &= \hat{o}_{ij}\left(n\right)-\eta\times E_{ij}\left(n\right)+\alpha\cdot\Delta\hat{o}_{ij}\left(n\right)\end{aligned}\tag{2.19}$$

where $\alpha\in[0,1]$ is a momentum parameter and a value of 0.9 is often used. For example, Figure 2.1 shows the offset of the correction curve. Note that the trajectory without momentum (the left curve) has larger oscillations than the one with momentum (the right curves). We further observe from the right curves in Fig. 2.1 that the momentum can enhance process toward the target point if the weight update is in the right direction (point A to A' in Fig. 2.1). On the other hand, it can redirect movement in a better direction toward the target point in the case of overshooting (point B to B' in Fig. 2.1). This observation indicates that the momentum term typically helps to speed up the convergence and to achieve an efficient and more reliable learning profile.

The use of momentum could improve the performance of the adaptive algorithm, improving its stability and probably reducing the production of ghosting artifacts, and it can leave some beneficial fluctuation in trajectory.

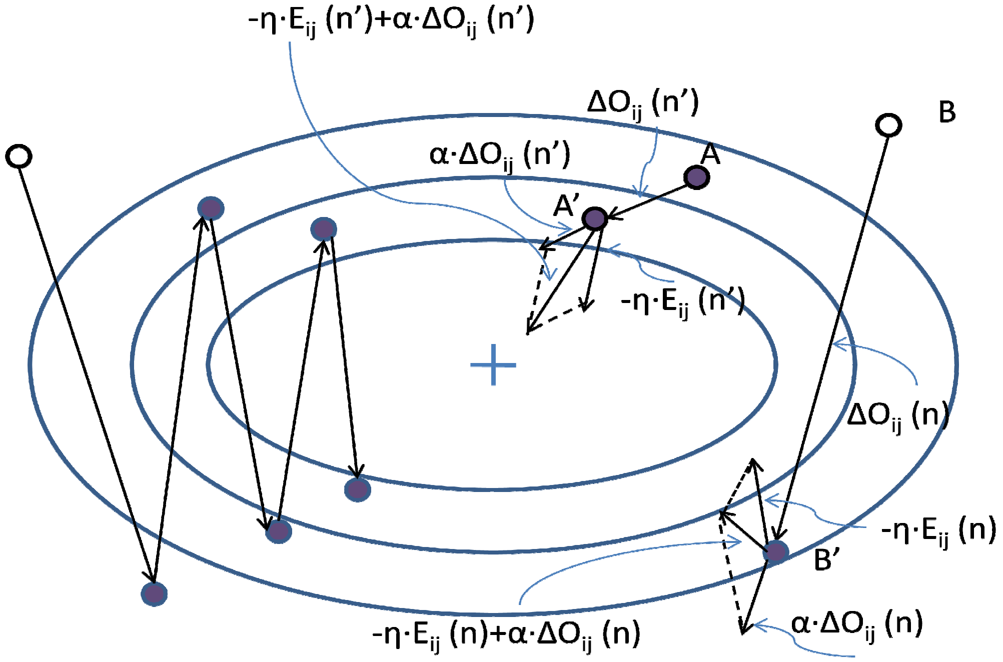

$-\eta\cdot E_{ij}\ (n')+\alpha\cdot\Delta O_{ij}\ (n')$
$\Delta O_{ij}\ (n')$
B
$\alpha\cdot\Delta O_{ij}\ (n')$
A
A'
$-\eta\cdot E_{ij}\ (n')$
$\Delta O_{ij}\ (n)$
$-\eta\cdot E_{ij}\ (n)$
B'
$-\eta\cdot E_{ij}\ (n)+\alpha\cdot\Delta O_{ij}\ (n)$
$\alpha\cdot\Delta O_{ij}\ (n)$

Fig. 2.2. The offset of the correction curve.

2.2.3. Adaptive Learning Rate

In the neural network algorithm, desired target value $T_{ij}(n)$ is used in the calculation of the error $E_{ij}(n)$ as follows:

$$T_{ij}(n) = \frac{\left(\sum_{k=-1}^{1}\sum_{l=-1}^{1} X_{i+k,j+l}(n)\right) - X_{ij}(n)}{8} \tag{2.20}$$

Ideal $T_{ij}(n)$ is the output value of the 2-points correction algorithm. In fact, neighborhood averaging method is the image smoothing filter. After doing neighborhood average, the variance becomes small, and the noise intensity is weakened. However, in particular, the boundary of the image may become blurred.

Thus, based on the knowledge that the local spatial average is not always a good estimation for the desired target response of an adaptive NUC method, the proposed adaptive learning rate $\eta_{ij}(n)$ showed in Eq. (2.21) is designed to be dependent, and inversely proportional to the local spatial variance of the input image $\sigma^2_{\mathrm{Y_{ij}}}(\mathrm{n})$.

$$\eta_{ij}(n) = K \cdot \frac{1}{1+\sigma^2_{Y_{ij}}(n)} \tag{2.21}$$

Therefore, if input image of the working window is smooth enough, the desired averaged target value at the output is more confident, and the learning rate gets larger. On the other hand, if a given sliding window size of the input image is not smooth enough, the local variance is too high, like in a object border, and the learning rate gets a smaller. To add this adaptive learning rate to the adaptive NUC algorithm, η in equation (2.15) must be replaced by its counterpart $\eta_{ij}(n)$ in Eq. (2.20), where K is a constant that limits the maximum learning rate. The local variance $\sigma^2_{\mathrm{Y_{ij}}}(\mathrm{n})$ can be calculated with any desired window size, and a 3 × 3 window size would be assumed along this paper.

Chapter 3 The Improvement of Bad Pixel Correction

After the Non-uniformity Correction, the infrared images often have bad pixels because the manufacturing process is not perfect, Most of the bad pixels are impulse noise. A few of dead pixels will be gathered in blobs, and that results in the difficulty and hence the failure of bad pixel correction. We proposed a new method can correct bad pixels observed in NIR sensor array.

In order to remove the noise of the NIR sensor, we use some color image filter for impulse noise. Although the NIR images are grayscale, we will improve filter from grayscale filter to color filter. When the filter can remove the noise in color images, grayscale image should also have good results. In Section 3.1, we introduce some basic and state-of-the-art impulse removing filter, and Section 3.2 will introduce our method.

3.1 Bad Pixel Correction

From Sec. 3.1.1 to Sec. 3.1.3, we will describe the basic impulse noise filter, and then from Sec. 3.1.4 to Sec. 3.1.8, we will introduce switching filters.

3.1.1 Vector Median Filters

Let $y(x): Z^l \to Z^m$ represent a multichannel image, where l is an image dimension and m characterizes a number of color channels. In the case of standard color images, parameters l and m are equal to 2 and 3, respectively. Let $W = \{x_i \in Z^l;\ i = 1, 2, \ldots, N\}$ represents a filter window of a finite size N, where $x_1, x_2, \ldots, x_N$ is a set of noisy samples. The central sample $x_{(N+1)/2}$ determines the position of the filter window. Let us consider that each input vector x_i is associated with the distance measurement:

$$L_i = \sum_{j=1}^{N} \left\| x_i \text{ - } x_j \right\|_\gamma \qquad \text{for } i = 1, 2, \ldots, N \tag{3.1}$$

where γ represents the selected norm, e.g. absolute ($\gamma = 1$), Euclidean ($\gamma = 2$), etc. The quantification of the distance between two m-channel samples $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ and $x_j = (x_{j1}, x_{j2}, \ldots, x_{jm})$ given by the expression $\left\| x_i - x_j \right\|_\gamma$ follows from the generalized Minkowski metric (Plataniotis and Venetsanopoulos, 2000) defined by

$$\left\| x_i - x_j \right\|_\gamma = \left(\sum_{k=1}^{m} \left| x_i - x_j \right|^\gamma \right)^{1/\gamma} \tag{3.2}$$

where γ characterizes the used norm, m is the dimension of vectors and x_{ik} is the k-th element of x_i.

If distance measures $L_1, L_2, \ldots, L_N$ serve as ordering criterions, i.e.

$$L_1 \le L_2 \le \ldots \le L_{(r)} \le \ldots \le L_{(N)} \tag{3.3}$$

it means that the same ordering is implied to the input set $x_1, x_2, \ldots, x_N$ which results in ordered input sequence

$$x^{(1)} \le x^{(2)} \le \ldots \le x^{(r)} \le \ldots \le x^{(N)} \tag{3.4}$$

The sample $x^{(1)} \in W$ associated with the minimum vector distance $L_{(1)}$ constitutes an output of the well-known vector median filter (VMF) introduced by Astola et al. [9]. Equivalently, the VMF output of the set $x_1, x_2, \ldots, x_N$ is defined as the sample $x_{VMF} \in \{x_1, x_2, \ldots, x_N\}$ that satisfies the following expression:

$$\sum_{i=1}^{N} \left\| x_{VMF} - x_j \right\|_\gamma \le \sum_{i=1}^{N} x \left\|_i - x_j \right\|_\gamma \quad \text{for } j = 1, 2, \ldots, N \tag{3.5}$$

where γ characterizes the used norm.

3.1.2 Vector Directional Filters

Vector directional filters (VDFs) [8] operate on the direction of image vectors and the VDF output is determined according to these directions in the vector space. By above operation, image vectors with atypical directions in the vector space are eliminated and VDFs lead to optimal estimates in the sense vectors' directions, so that VDFs preserve the color chromaticity well.

Let each input sample x_i, for $i = 1, 2, \ldots, N$, be associated with a sum of vector angles

$$\alpha_i = \sum_{j=1}^{N} A\left(x_i, x_j\right) \qquad \text{for } i = 1, 2, \ldots, N \tag{3.6}$$

where

$$A\left(x_i, x_j\right) = \cos^{-1} \cos^{-1}\left(\frac{x_i \cdot x_j^{\ T}}{\left|x_i\right| \cdot \left|x_j\right|}\right) \tag{3.7}$$

represents the angle between two m -dimensional vectors $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ and $x_j = (x_{j1}, x_{j2}, \ldots, x_{jm})$.

If $\alpha_1, \alpha_2, \ldots \alpha_N$, i.e. the sums of vector angles, serve as ordering criterions, i.e.

$$\alpha_{(1)} \le \alpha_{(2)} \le \cdots \le \alpha_{(N)} \tag{3.8}$$

and the same ordering is applied to the input set $x_1, x_2, \ldots, x_N$. This operation results in Eq. (3.4). The sample $x^{(1)}$, i.e. the sample that minimizes the sum of angles with other vectors, represents the output of basic vector directional filter (BVDF) [10]. Since the BVDF passes to the filter output the sample associated with minimal angle distance $\alpha_{(1)}$, it preserves the color chromaticity better than the VMF.

3.1.3 Directional Distance Filters

Directional Distance Filters (DDF) combine advantage of VMF and BVDF [11], the Algorithm is as following: If the minimization formula is expressed through a minimization of products

$$\Omega_i = L_i \times \alpha_i \qquad \text{for } i = 1, 2, \ldots, N \tag{3.9}$$

$$\Omega_i = \left(\sum_{j=1}^{N} \left\| x_i - x_j \right\|_{\gamma} \right) \left(\sum_{j=1}^{N} A\left(x_i, x_j \right) \right) \quad \text{for } i = 1, 2, \ldots, N \tag{3.10}$$

and the filter output is given by the sample $x^{(1)}$ associated with $\Omega_{(1)}$, i.e. the minimum value from products $\Omega_1, \Omega_2, \ldots, \Omega_N$, and such their ordered set is simply written as

$$\Omega_{(1)} \le \Omega_{(2)} \le \cdots \le \Omega_{(N)} \tag{3.11}$$

then the sample $x^{(1)}$ determines the output of DDF.

Although the minimization of products $L_i \times \alpha_i$, for $i = 1, 2, \ldots, N$, does not necessarily imply a minimum for either L_i and α_i, it results in very small values for both of them [10]. For that reason, the product minimization will select as the filter output the vector-valued sample that results in a very small sum of vector distances Eq. (3.1) and a very small sum of vector angles Eq. (3.6), simultaneously.

3.1.4 Peer Group Filter

These are adaptive switching filters based on the peer group concept [16]. Essentially, the peer group of pixels in a given window represents the set of neighbor pixels that are sufficiently similar to each pixel according to a particular measurement. In the Peer Group Filter (PGF) [16] the pixels in the window are sorted in ascending order according to their distances to the center pixel. The center pixel $x^{(1)}$ of the peer group is then determined as the filtering window W pixels that rank the lowest in this sorted sequence. In order to remove the effect of the impulsive noise, if the distance between $x^{(1)}$ and the central pixel of W x_i is not exceeding threshold d, the central pixel is free of noise. Otherwise, the center pixel of W is considered noisy (see fig. 3.1). In this case, the center pixel is replaced with the VMF output $x^{(1)}$; otherwise it remains unchanged. The range of the threshold d is set to [40,60]. Equation is expressed as follows:

$$\mathrm{PGF_{out}} = \begin{cases} x^{(1)}, & \text{if } \left\| x^{(1)} - x_i \right\|_2 > d \\ x_i, & \text{if } \left\| x^{(1)} - x_i \right\|_2 \le d \end{cases} \tag{3.12}$$

The following proposed method is the improvement of PGF, known as Fast Peer Group Filter (FPGF) [17]. The FPGF is a fast modification of the PGF in which the center pixel is considered to be noise-free as soon as m pixels in the window are determined to be sufficiently similar. The peer group $P(x_i, m, d)$ will denote the set of pixels of the filtering window W which satisfy the following condition:

$x_i - x_{j2} \le d$, $x_j \in W$. In other words, if the central pixel of W x_i has m neighbors, and distance between x_i and neighbors is not exceeding d, the central pixel x_i is determined to belong to the peer group $P(x_i, m, d)$. If m is low, and the amount of noise in the image is not very much, the number of distance computations which is originally necessary in PGF can be dramatic reduced. Equation is expressed as follows:

$$\mathrm{FPGF}_{\mathrm{out}} = \begin{cases} x_i, & \text{if } x_i \in P\left(x_i, m, d\right) \\ x^{(1)}, & \text{otherwise} \end{cases} \tag{3.13}$$

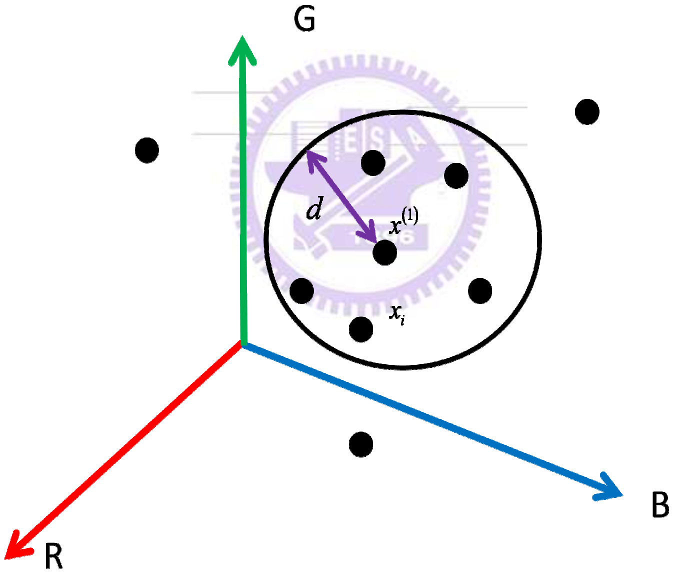


Fig. 3.1. The concept of the peer group centered at $x^{(1)}$ (m = 5).

3.1.5 Fuzzy Modified Peer Group Filter

Fuzzy Modified Peer Group Filter (FMPGF) describe in [18], the peer group concept is adapted to the use of a novel fuzzy metric. The use of the fuzzy metric is considered because it has been proved to be an appropriate alternative to some classical measures [19] and in order to further introduce it in the area. This fuzzy metric may be also useful in other approaches based on fuzzy techniques, which are interesting in image processing. Due to the non-stationarity of images and the difficulty in distinguishing between noise and edges, fuzzy modeling is considered quite appropriate in image filtering [20]. The fuzzy similarity between two pixels x_i and x_j can be computed as:

$$M_p\left(x_i, x_j\right) = \prod_{i=1}^{p} \frac{\min\min\left\{x_i, y_i\right\} + K}{\max\max\left\{x_i, y_i\right\} + K} \tag{3.14}$$

They denote by $P(x_i, d)$ the set

$$\left\{x_j \in P\left(x_i, d\right), M_p\left(x_i, x_j\right) \geq d\right\} \tag{3.15}$$

In order to increase computing speed, FMPGF use non-overlap window, FMPGF algorithm is as follows:

(1) The image under processing is divided into $\frac{N_1 \times N_2}{n^2}$ disjoint $n \times n$ windows W. Let x_i denote the central pixel of one of these windows. Then, the following rule is computed for each central pixel x_i:

```
IF $\exists P(x_i, m, d) \subset W$; THEN
    $\forall x_j \in P(x_i, m, d)$; $x_j$ is declared as non-corrupted and
  $\forall x_k \in W, x_k \notin P(X_i, m, d)$; $x_k$ is declared as *non-diagnosed*.
```

```
ELSE
 $x_i$ is declared as *provisionally corrupted* and
 $x_j \in W, j \neq i$ ; $x_j$ is declared as *non-diagnosed*.
```

(2) Each *non-diagnosed* pixel x_i is now considered centered in a $n \times n$ window W and the following rule is computed:

```
If $\exists P\left(x_i, m, d\right) \subset W$, THEN
$\forall x_j \in P(x_i, m, d)$, $x_j$ is declared as *non-corrupted*.
ELSE
 $x_i$ is declared as corrupted.
```

(3) In this step the switching filtering to compute the outputs y from the inputs x is carried out in a $n \times n$ window W as follows:

$$y_i = \begin{cases} x_i, & \text{if } x_i \text{ is non-corrupted.} \\ AMF_{out}, \text{over the non-corrupted pixels in } W & \text{if } x_i \text{ is corrupted.} \end{cases}$$

3.1.6 Fast Similarity-Based Impulsive Noise Removal Vector Filter

According to the family of filters introduced by Smolka et al. in [21–24], the *fast similarity-based impulsive noise removal vector filter* (FSVF) is defined as follows: Let us assume a filtering window W containing $n+1$ image pixels $\left\{F_0, F_1, \ldots, F_n\right\}$, where n is the number of neighbors of the central pixel F_0. It is considered a similarity function $\mu:[0;\infty)\rightarrow \mathrm{R}$ which is non-ascending and convex in $[0;\infty)$ and satisfies $\mu(0)=1$, and $\lim_{x\rightarrow\infty}\mu(x)=0$. The similarity between two pixels of the same color should be 1, and the similarity between pixels with very different colors should be very close to 0. The function defined as $\mu\left(\left\|F_i-F_j\right\|\right)$ where $\|\bullet\|$ denotes the specific vector norm (typically the L_1 or L_2 vector norms) can easily satisfy the above conditions when it is a decreasing function and $\mu(0)=1$. The cumulated sum M_k of similarities between a given pixel $F_k(k=0,\ldots,n)$ and all other pixels belonging to the window W is defined as

$$M_0=\sum_{j=1}^{n}\mu\left(F_0,F_j\right),\quad M_k=\sum_{\substack{j=1\\ j\neq k}}^{n}\mu\left(F_k,F_j\right) \tag{3.16}$$

which means that for those F_k which are neighbors of F_0, the similarity between F_k and F_0 is not taken into account (see fig. 3.2), which privileges the central pixel. Hence, the reference pixel F_0 is replaced by one of its neighbors if $M_0<M_k,\quad k=1,\ldots,n$, only when it is really noisy, preserving the original undistorted image structures. If this is the case then, F_0 is replaced by that F_{k^*} for which $k^*=\arg\arg\max_k M_k$. Equation is expressed as follows:

$$\begin{cases} F_{k^*},\ M_0<M_{k^*} \\ F_0,\ M_0\geq M_k \end{cases} \tag{3.17}$$

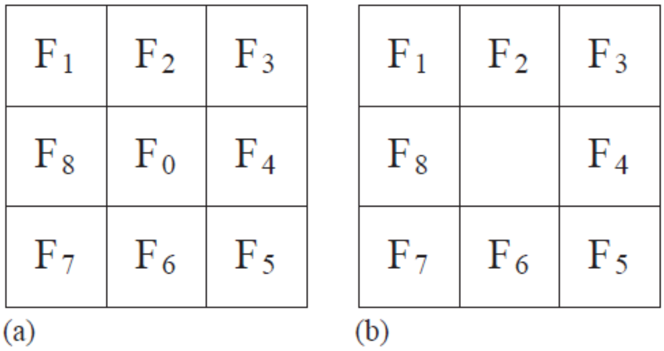


Fig. 3.2. (a) First the cumulative similarity value M_0 between the central pixel F_0 and its neighbors is calculated. (b) Then pixel F_0 is rejected from the filter window and the cumulative similarity values M_k, $k = 1, \ldots, n$ of the pixels $F_1, \ldots, F_n$ are determined [22].

Several convex functions fulfilling the above conditions have been proposed in [21–24]. The best results were achieved [22] for the simplest similarity function

$$\mu_7\left(F_i, F_j\right) = \begin{cases} 1 - \dfrac{\rho\left(F_i, F_j\right)}{h}, & \text{for } \rho\left(F_i, F_l\right) < h \\ 0, & \text{otherwise} \end{cases} \tag{3.17}$$

Where $h \in (0, \infty)$, and ρ denotes the particular distance function, typically the L_1 or L_2 distances. This function allows to construct a fast noise reduction algorithm [21–24].

3.1.7 Fast Impulsive Noise Filter Using Fuzzy Metrics

The filter of the paper [19] is similar to FSVF, but similarity function just like FMPGF. The proposed fuzzy metric has been combined with the FSVF technique [21–24] to define a computationally efficient filter. This filter is faster than FSVF since the filtering process is simpler and the fuzzy metric used is faster than the classical metrics used in FSVF [21–24]. The fuzzy similarity between two pixels x_i and x_j can be computed as:

$$M_p^{\alpha}(x,y)=\prod_{i=1}^{p}\left(\frac{\min\min\left\{x_i,y_i\right\}+K}{\max\max\left\{x_i,y_i\right\}+K}\right)^{\alpha} \tag{3.18}$$

where $x=\left(x_1,..,x_p\right)$, $y=\left(y_1,..,y_p\right)$, the particular case of the proposed fuzzy metric Map suitable for 3-channel image processing tasks will be M_3^{α}, and then $M_3^{\alpha}(F_i,F_j)$ will denote the fuzzy distance between the pixels F_i and F_j in the image.

In the proposed filtering, the cumulated sum M_k of similarities between a given pixel $F_k(k=0,\ldots,n)$ and all other pixels belonging to the window W is defined as

$$M_0=\sum_{j=1}^{n}M_3^{\alpha}(F_0,F_j),\quad M_k=\sum_{\substack{j=1\\ j\neq k}}^{n}M_3^{\alpha}(F_k,F_j) \tag{3.19}$$

the reference pixel F_0 is replaced by F_{k^*} if $M_0<M_k$, the equation is the same as Eq. (3.17).

3.1.8 Fuzzy Peer Group Averaging Filter

In the paper [25], they introduced the concept of fuzzy peer group for a color image pixel which extends the concept of peer group in the fuzzy setting. This novel concept aims to represent the set of all pixel neighbors to a given pixel which are similar to it. Since the similarity between color pixels is an imprecise concept, they have represented it using fuzzy similarities. For this, they have introduced a method based on fuzzy logic that builds the fuzzy peer group of a color image pixel by first determining the members of the fuzzy peer group and then assigning their corresponding membership degrees. The proposed method is able to accurately determine the fuzzy peer group of any color image pixel overcoming shortcomings of previous peer group approaches.

The fuzzy peer group averaging filter (FPGA) can correct impulse noise and Gaussian noise, but we just use it for impulse noise here. The filter method is as follows:

1. Determine the fuzzy peer group
2. Determine the best number of members for a fuzzy peer group.
3. Using above results to detect and replace impulses.

where $\mathrm{C}_{\mathrm{FR}\,2}$ will be discussed later.

In order to establish our concept of fuzzy peer group we will define two fuzzy sets on the ordered set of pixels . Firstly, we consider the proposition " is similar to ." In the approach presented in this paper, they propose to use a fuzzy similarity function, ρ, as the function above which, following the above terminology, is given by

$$\rho\left(F_i, F_j\right) = e^{-\frac{F_i - F_j}{F_\sigma}}, \qquad i, j = 0,1,2,\ldots,n^2-1 \tag{3.20}$$

where $\|\cdot\|$ denotes the Euclidean norm and $F_\sigma > 0$ is a parameter which will be discussed later. Notice that now ρ takes values in [0,1] and that $\rho\left(F_0, F_{(i)}\right) = 1$ if and only if $F_0 = F_{(i)}$. Again, as discussed above, the color vectors $F_i \in W$ are sorted in a descending order with respect to its similarity to F_0, which results in an ordered set W' defined $W^{'} = \left\{F_{(0)}, F_{(1)}, \dots, F_{\left(\mathrm{n}^2-1\right)}\right\}$ as follows: such that $\rho\left(F_0, F_{(0)}\right) \geq \rho\left(F_0, F_{(1)}\right) \geq \dots \geq \rho\left(F_0, F_{(\mathrm{n}^2-1)}\right)$, where $F_{(0)} = F_0$.

Secondly, they define the accumulated similarity for $F_{(i)}$, denoted $\mathrm{A}^{F_0} F_{(i)}$, as

$$A^{F_0}\left(F_{(i)}\right) = \sum_{k=0}^{i} \rho\left(F_i, F_{(k)}\right),\ \ i \in \left\{0, 1, \dots, n^2 - 1\right\} \tag{3.21}$$

For computing its certainty, author used a fuzzy membership function, L^{F_0} ,on $\left\{F_{(0)}, F_{(1)}, \dots, F_{\left(\mathrm{n}^2-1\right)}\right\}$. In this work author prefer to define L^{F_0} as a function of A^{F_0} by means of a custom membership function defined on that fulfills the following requirements:

1. $\mu\left(A^{F_0}\left(F_{(i)}\right)\right) = 0$ for the minimum possible value of A^{F_0}, that is, $\mu(1) = 0$.

2. $\mu\left(A^{F_0}\left(F_{(i)}\right)\right) = 1$ for the maximum possible value of A^{F_0}, that is, $\mu\left(n^2\right) = 1$.

Author prefer the membership function to be more sensitive in the low value range than in the high value range, and, therefore, author devise μ so that the derivative of μ should be a strictly decreasing function. The function is given by

$$\mu(x) = \left(-\frac{1}{\left(n^2-1\right)^2}\right) \times (x-1)\left(x - 2n^2 + 1\right) \tag{3.22}$$

$$L^{F_0}\left(F_{(i)}\right)=\mu\left(A^{F_0}\left(F_{(i)}\right)\right)=\left(-\frac{1}{\left(n^2-1\right)^2}\right)\times\left(A^{F_0}\left(F_{(i)}\right)-1\right)\left(A^{F_0}\left(F_{(i)}\right)-2n^2+1\right)$$

$$i=0,1,2,\ldots,n^2-1 \tag{3.23}$$

According to the above, they determine the best number of members for a fuzzy peer group. The best number of members of a peer group for a given pixel will be determined by choosing so that all similar pixels are included in the set and the rest of the pixels are not. In other words, if a pixel has similar neighbors, the best number of members for its peer group is $\widehat{m}$, and vice versa. Author proposal is based on determining $\widehat{m}$ as the value for which $P_i^{F_0}$ is the largest set that contains only similar pixels. The best number of members $\widehat{m}$ of $P_i^{F_0}$ will be the value of $m\in N_w=\{1,2,\ldots,n^2-1\}$ maximizing the certainty of the following fuzzy rule.

Fuzzy Rule 1: Determining the certainty of to be the best number of m members for $P_m^{F_0}$

IF “$F_{(m)}$ is similar to F_0” and “$A^{F_0}\left(F_{(m)}\right)$ is large”

THEN “the certainty of to be the best number of members is high”.

The mathematical equation is as follows:

$$\widehat{m}=argmax_{m\in N_w}C_{FR1}\left(m\right) \tag{3.24}$$

$$C_{FR1}\left(m\right)=C^{F_0}\left(F_{(m)}\right)L^{F_0}\left(F_{(m)}\right) \tag{3.25}$$

$$C^{F_0}\left(F_{(i)}\right)=\rho\left(F_0,F_{(i)}\right) \tag{3.26}$$

According to the above, we can formulate this condition in terms of fuzzy peer groups as follows: a pixel F_0 is free of impulse noise if for the *fuzzy peer group* $FP_m^{F_0}$ it is satisfied that “ $A^{F_0}\left(F_{(\hat{m})}\right)$ is large” and “ $F_{(\hat{m})}$ is similar to F_0 ”. The following Fuzzy Rule 2 represents this condition:

Fuzzy Rule 2: Determining the certainty of the pixel to be free of impulse noise

IF “ $A^{F_0}\left(F_{(\hat{m})}\right)$ is large” and “ $F_{(\hat{m})}$ is similar to F_0 ”

THEN “ F_0 is free of impulse noise” .

The mathematical equation is as follows:

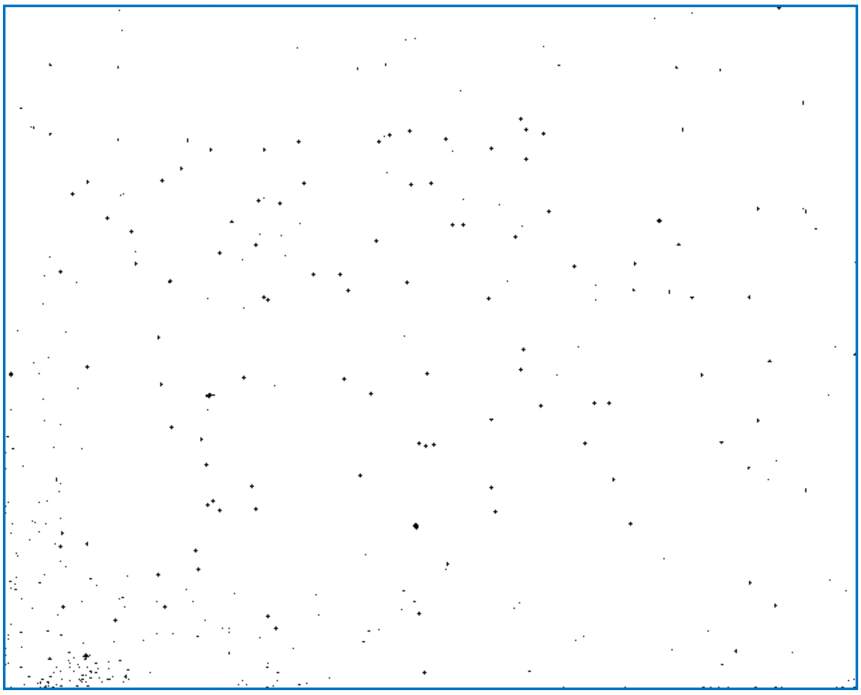
$$\mathrm{C_{FR\,2}}\left(\mathrm{F_0}\right)=\mathrm{C}^{\mathrm{F_0}}\left(\mathrm{F}_{(\hat{m})}\right)\mathrm{L}^{\mathrm{F_0}}\left(\mathrm{F}_{(\hat{m})}\right) \tag{3.27}$$

They use to detect and replace impulses according to threshold-based rule are shown in Eq. (3.28) [28],

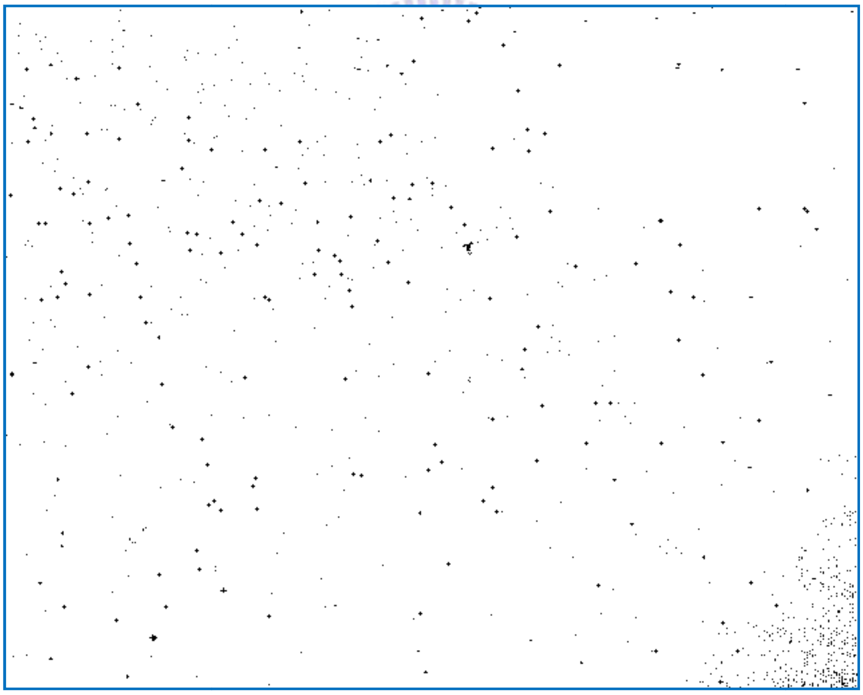
$$\begin{cases} \text{ if } \mathrm{C_{FR\,2}}\left(\mathrm{F_0}\right)\geq \mathrm{F_t}, \text{then } \mathrm{F_0} \text{ is free of impulse noise} \\ \text{else } \mathrm{F_0} \text{ is an impulse and it is replaced with } \mathrm{VMF_{out}} \end{cases} \tag{3.28}$$

3.2 Proposed Filtering

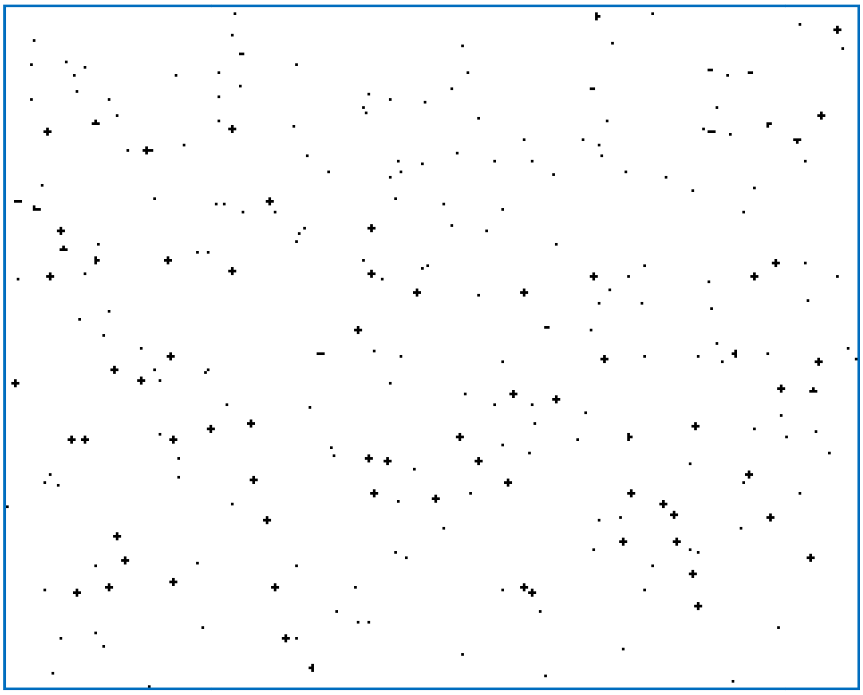
When the noise is impulse noise and noise of the low density, above the noise filter usually can achieve good results. In fact, the noises which are detected by NIR sensor are frequently in blobs. Fig. 3.3 shows the noise maps of the sensor, noise types as shown in fig. 3.4. From type 1 to type 16 is a basic type, others are basic type combinations.



(a)



(b)



(c)

Fig. 3.3. The noise maps of the sensor. (a) Sample 1. (b) Sample 2. (c) Zoomed "sample 2".

When concentration of the impulse noise is not high, peer group filter with 3 × 3 working window is enough; however, when bad pixels are blobs, the number of bad pixels in working window is too much, bad pixels may be misjudgment as free of noise. In order to detect dead pixels in blobs, we use the larger size of the working window. Because the number of dead pixels becomes a minority in a large working window, bad pixels can be detected. But the boundary in the image is blurred. The large size of the working window can detect a group of bad pixels, but the details will be blurred; small size of the working window can keep the details, but cannot detect a group of bad pixels. We propose a method: we hope that the noises are not in blobs, we use a 3 × 3 working window; the noises are in blobs, we use larger working window size. The block diagram of the method is as follows:

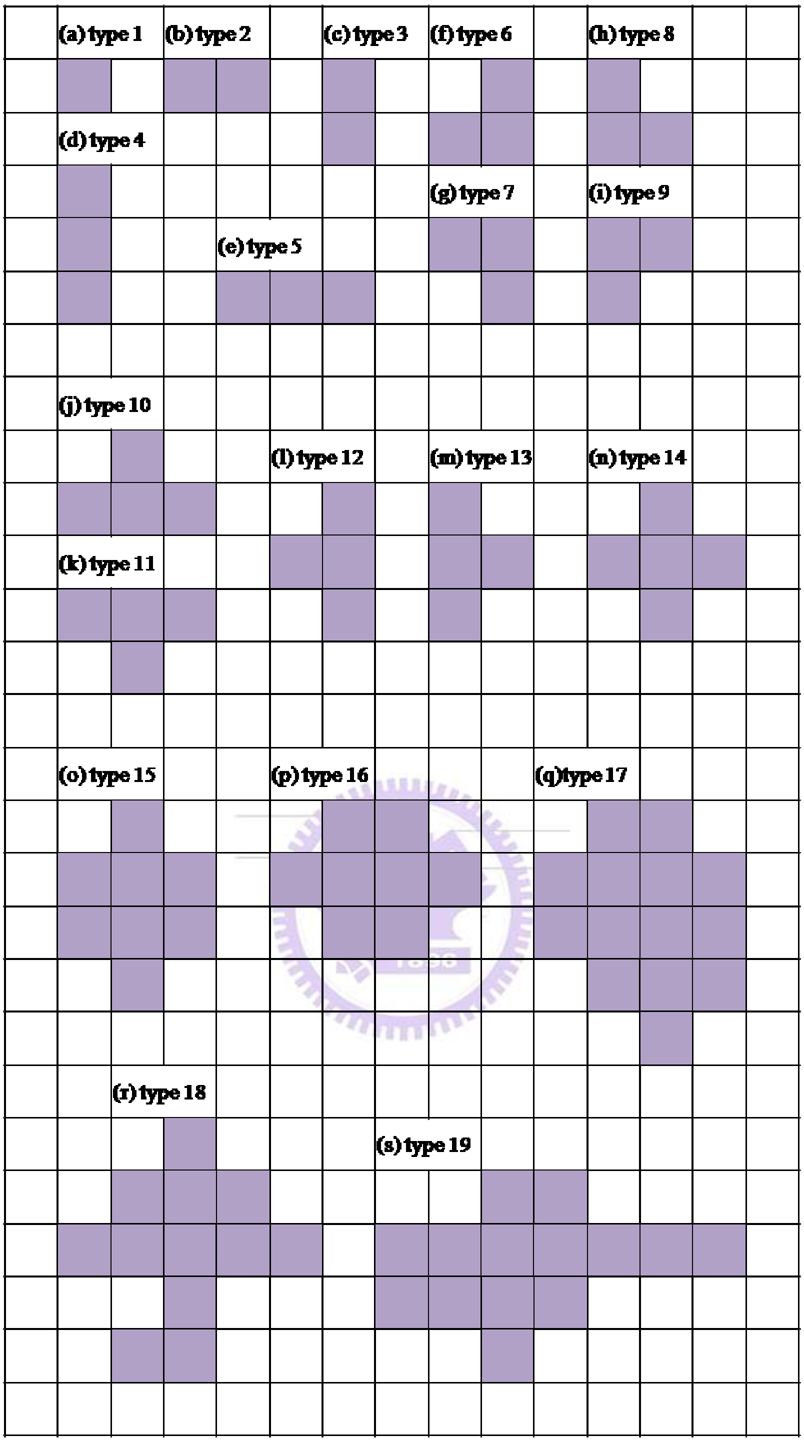


Fig. 3.4. Noise types in sensor. (a) – (p) are basic type; (q) – (s) are basic type combinations.

We use PGF with 5×5 working window to detect bad pixel, because 3×3 working window cannot detect bad pixels in blobs. Peer group filter algorithm as section 3.1.4. We calculate the vector median of all pixels within the working window W (section

3.1.1). Vector is composed of RGB three dimensions, if the distance between $x^{(1)}$ and the central pixel of W x_i is not exceeding threshold d, the central pixel is free of noise. Otherwise, the center pixel of W is considered noisy.

When the pixel is detected as a bad pixel, we will use the vector median of the 3 × 3 working window to correct (see Fig. 3.5, Fig. 3.6). If the distance between $x^{(1)}$ and the central pixel of W x_i is not exceeding threshold d, the central pixel is free of noise. Otherwise, the center pixel of W is considered noisy. After Correction, we use peer group with 5 × 5 working window to detect central pixel of W x_i again. If x_i is still bad pixel, we increase the size of the working window. After Correction, We will detect x_i whether x_i is bad pixel or not again, and so on, until the central pixel x_i is corrected.

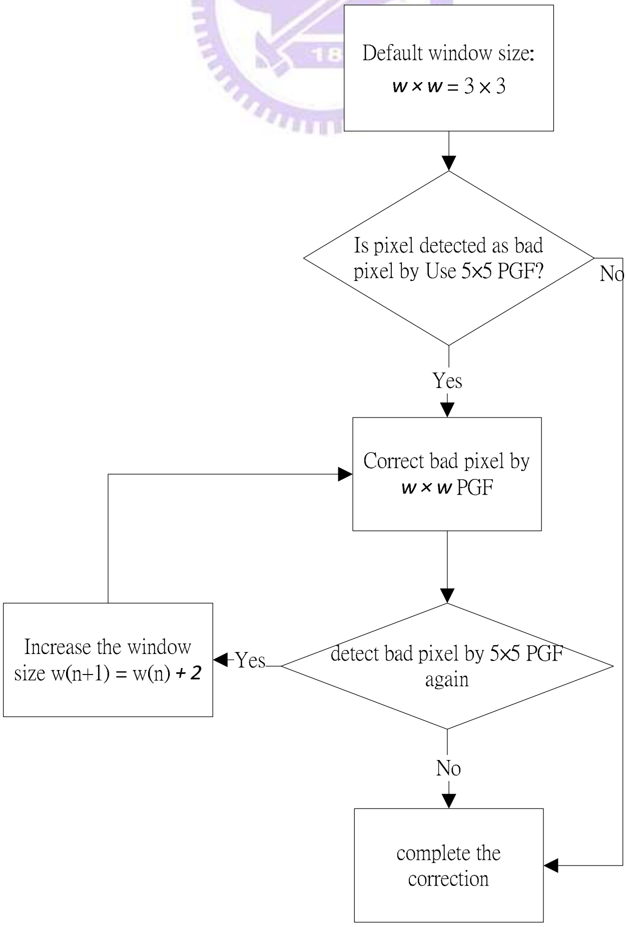

Default window size:
$w \times w = 3 \times 3$
Is pixel detected as bad pixel by Use 5×5 PGF?
No
Yes
Correct bad pixel by $w \times w$ PGF
Increase the window size w(n+1) = w(n) + 2
Yes
detect bad pixel by 5×5 PGF again
No
complete the correction

Fig. 3.5. Block diagram of the method.

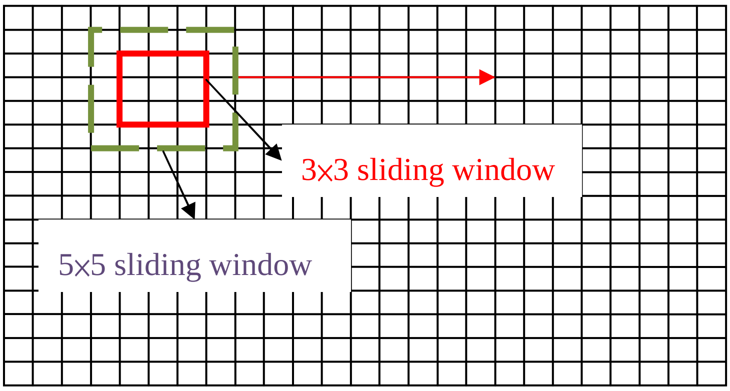


Fig. 3.6. The working window *W* size of n x n, (n=3,5,7,…)

The purpose of this method is as much as possible to keep the details of image, and we use a 5 × 5 PGF at the right time, let bad pixels in blobs be corrected.

Chapter 4 Experimental Results

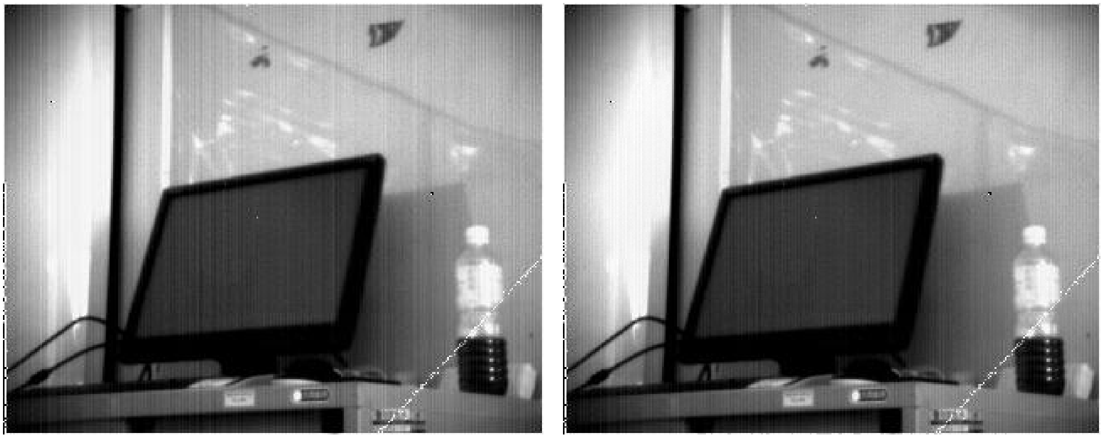
The experimental results are divided into two parts, we show the experimental results of NUC in Section 4.1, and we show experimental results of bad pixel correction in Section 4.2.

4.1 Results of NUC

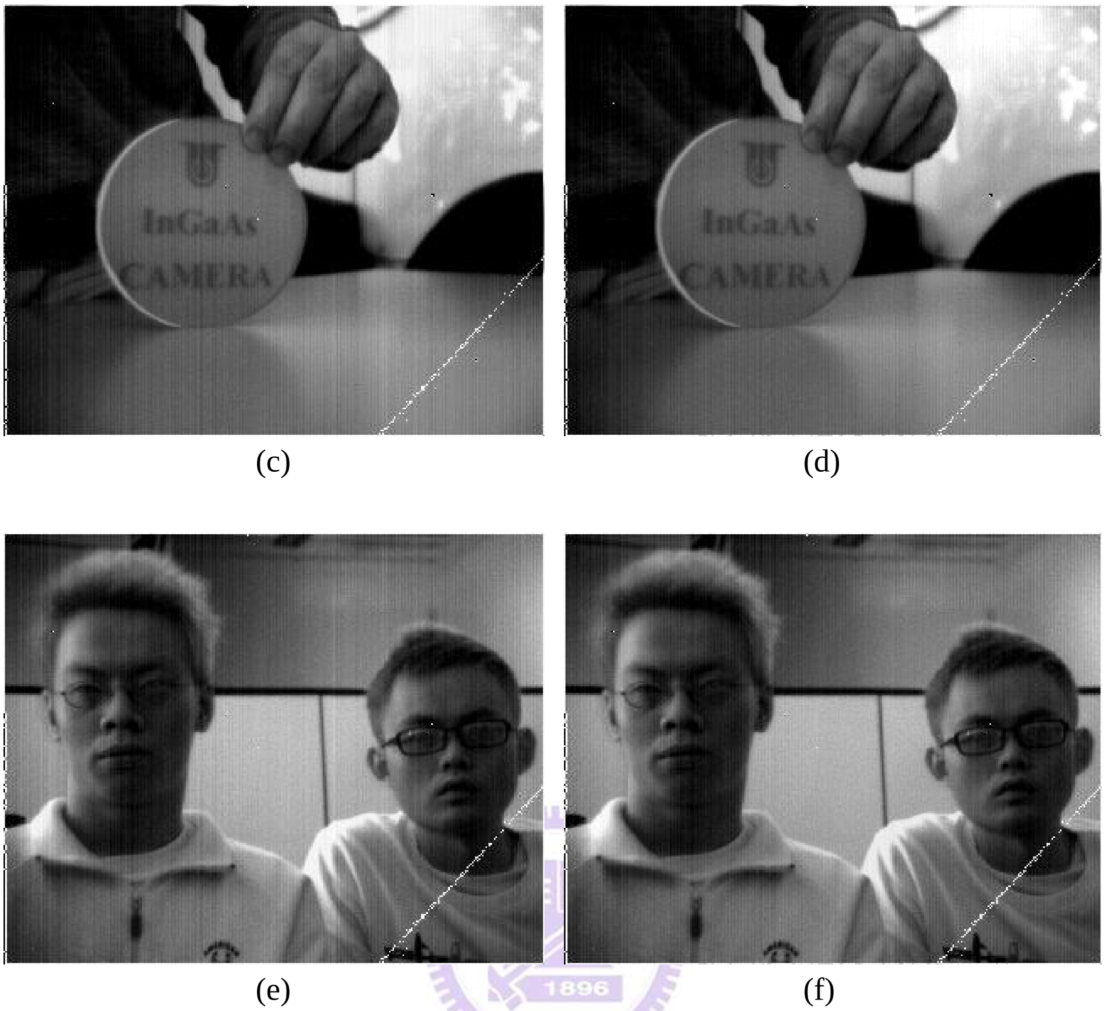
We show the result of two of two-point correction in Section 4.1.1, and we show the result of LMS method in Section 4.1.2.

4.1.1 Result of Two-point Correction

Applying two-point correction to “Monitor,” “Words,” and “Two-persons” images of Figs. 4.1 (a), (c), and (e), the corrected images are shown in Figs. 4.1(b), (d), and (f), respectively.



(a) (b)

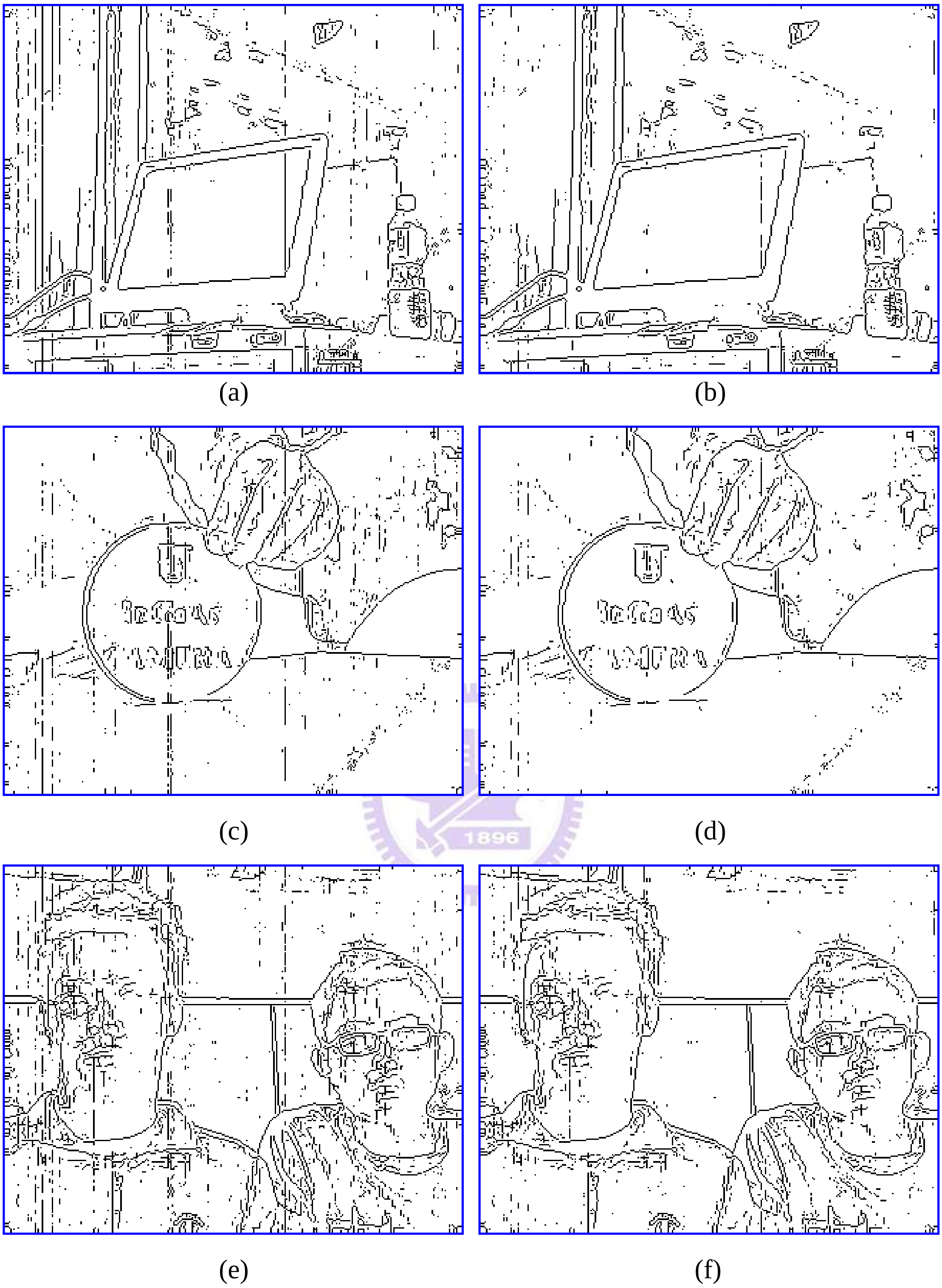


(c) (d)

(e) (f)

Fig. 4.1 (a) The raw image of “Monitor”, (b) The corrected image of “Monitor”; (c) The raw image of “Words”, (d) The corrected image of “Words”; (e) The raw image of “Two-persons”, (f) The corrected image of “Two-persons”.

To further validate the effectiveness of NUC, we exploit Sobel edge detection with a sensitive threshold to test raw images and NUC corrected images above, leading to Figs. 4.2 (a) – (f), respectively.



(a) (b)

(c) (d)

(e) (f)

Fig. 4.2. Sobel edge map images of (a) The raw image of “Monitor”, (b) The corrected image of “Monitor”; (c) The raw image of “Words”, (d) The corrected image of “Words”; (e) The raw image of “Two-persons”, (f) The corrected image of “Two-persons”.

From these figures in Fig. 4.2, it is easy to see the powerfulness of NUC in reducing the excessive or un-necessary edges due to the non-uniformity of the image sensors. Excessive lines in left-up corner of Fig. 4.2 (b) is removed in Fig. 4.2 (a), NUC corrected counterpart. From Fig. 4.2 (c), it is evident that removed of excessive lines and edges, in tables, wafer, and hand right, of Fig. 4.2 (d). Unnecessary lines and edges of left person in Fig. 4.2 (f) has been removed in NUC corrected image Fig. 4.2 (e).

4.1.2 Result of Least Mean Squares

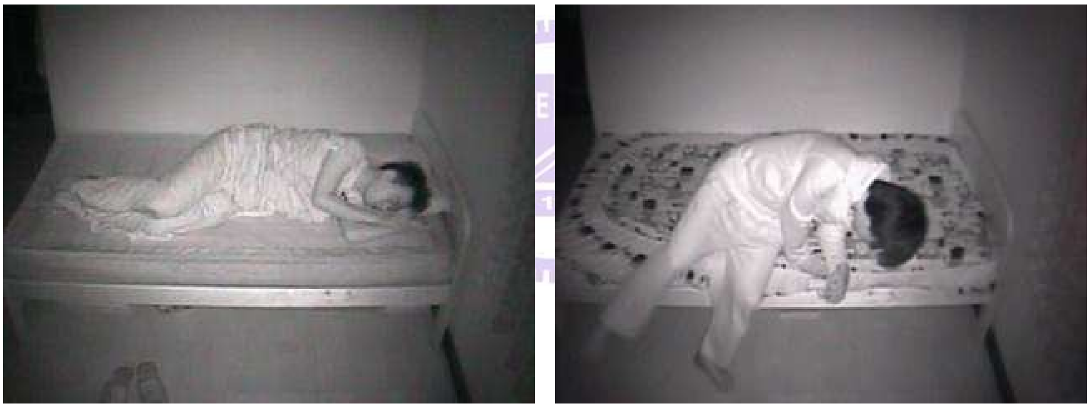
In this section, the proposed enhancements to the LMS method are tested with infrared data corrupted with simulated non-uniformity. The infrared sequences with artificial non-uniformity were generated from a clean 1200 frame infrared video sequence. Then, several 1200 frame corrupted video sequences were obtained using a synthetic gain with an unitary- mean gaussian distribution with 3% of variance, and a synthetic offset with a zero-mean gaussian distribution with 5% of variance. As an example, fig. 4.3 a) shows a sleeping person image and fig. 4.3 b) shows a walking to bed person. The parameters used to initialize the weight and bias estimation process are random.

To study the performance of the proposed enhanced methods, we employ the Pseudo Signal to Noise Ratio (PSNR) performance metric, which is based on the Root Mean Square Error (RMSE). The PSNR is expressed in dB and it is defined as follows:

$$RMSE = \sqrt{\frac{1}{NM}\sum_{ij}(I_{ij} - \hat{I}_{ij})^2} \tag{4.1}$$

$$PSNR = 20 \times \log_{10}(\frac{2^b}{RMSE}) \tag{4.2}$$

where I_{ij} is the *ij* pixel value of the true frame, and $\hat{I}_{ij}$ is the *ij* pixel value of the corrected frame. The frame size is $N \times M$ pixels, and b represents the number of bits per pixel in the image, which in this case is equal to 8 for all the simulations. The PSNR values shown in this section were computed averaging the results obtained. Larger value for the PSNR indicates better performances.



(a) (b)

Fig. 4.3. IR image of (a) Sleeping person. (b) Walking to the bed person.

First, we use three modified method in [3], regularization (REG), momentum (MOM) and adaptive learning rate (ALR), and we try several combination of the modified method in [3]. The results are shown in Fig. 4.4.

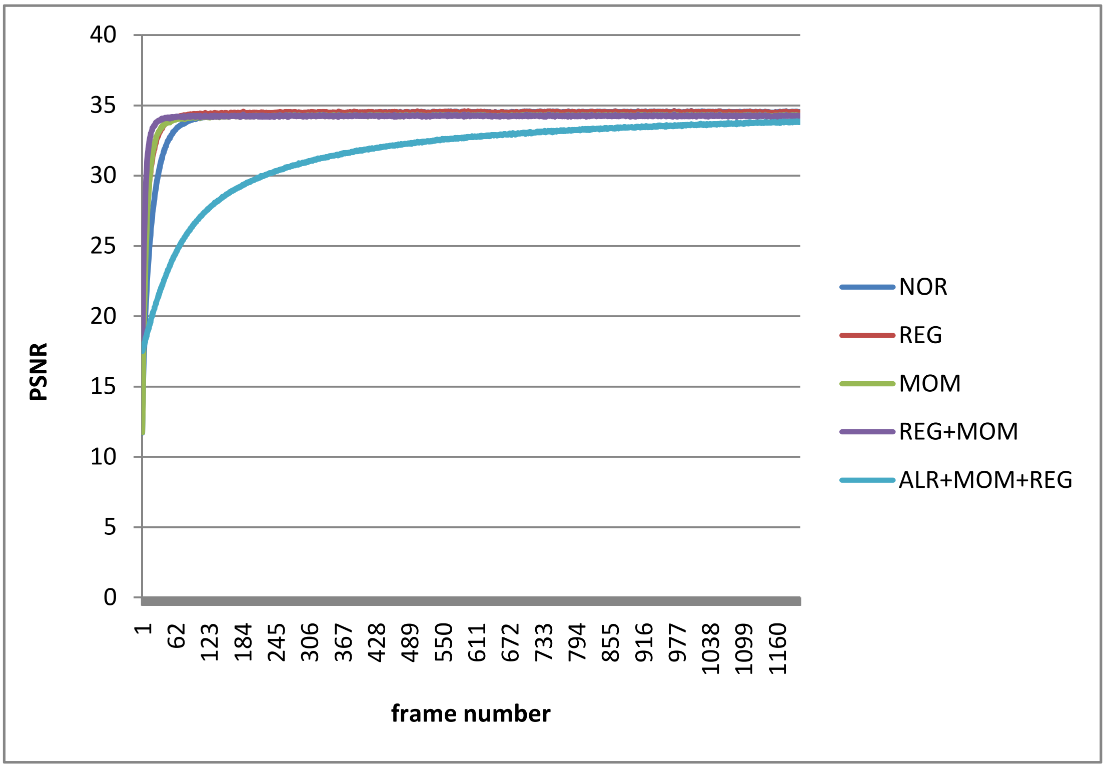


Fig 4.4. PSNR for the LMS algorithm and its proposed modifications versus frame number. ’NOR’ indicates LMS method, ’MOM’ indicates LMS method plus momentum, ’REG’ indicates LMS method plus regularization, ’MOM+REG’ indicates LMS methods plus momentum and regularization, and ’ALR+MOM+REG’ indicates LMS methods plus momentum, regularization and adaptive learning rate.

Figure 4.3 shows that ALR in the initial learning rate was significantly slower, because ALR formula in Eq. 2.20 will reduce the learning rate of the initial settings.

$$\eta_{ij}(n) = K \cdot \frac{1}{1+\sigma^2_{Y_{ij}}(n)} \tag{2.20}$$

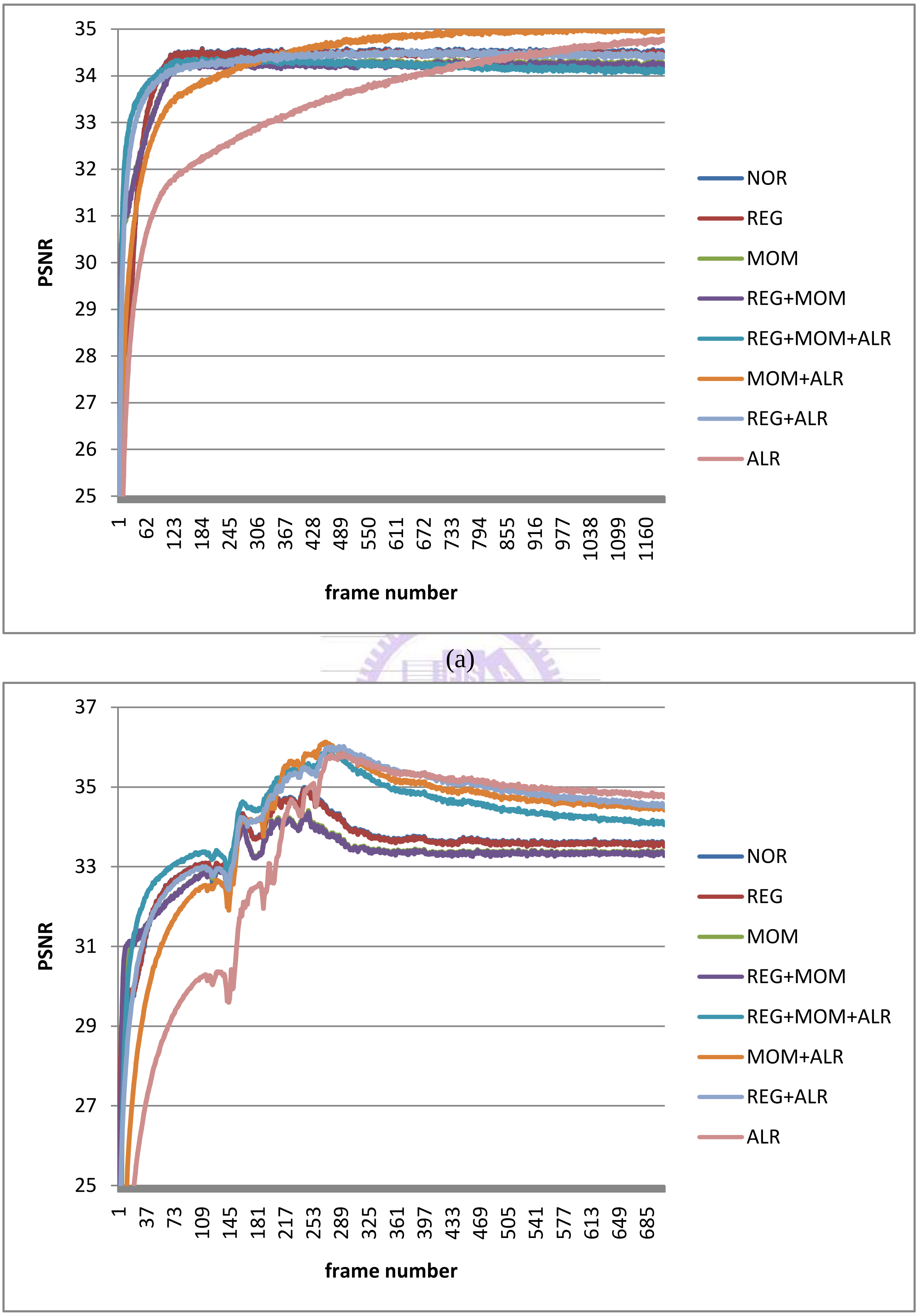
where K is a constant that limits the maximum learning rate. If local variance $\sigma^2_{Y_{ij}}(n)$ is too large, the K value must be larger. For example, $\sigma^2_{Y_{ij}}(n)=4$, and the

$\eta_{ij}\left(n\right) = K\frac{1}{4+1} = 0.2K$ .If each learning rate η of modifications is the same, the K must be $5\times\eta$. Therefore, we adjust the value of K, let average of the learning rate be our settings. In addition, RMSE is larger at the beginning, so we should have a larger learning rate; and the learning rate which is gradually smaller performs minor adjustments.

The paper described three modified method, there are eight kinds of combinations as follows: NOR, REG, MOM, ALR, REG+MOM, REG+ALR, MOM+ALR, and REG+MOM+ALR. They only try several combinations in [3], we will find the best method of all combinations, and we will test two video: “sleeping person” and “walking to bed person”. According to the above experimental results, η = 5 × 10^{-5} when the frame number is 1–20, η will linear down to 10^{-5} when the frame number is 21–120. The results are shown in Fig. 4.5.

In the beginning, learning rate of "REG + MOM + ALR" is better in video “sleeping person”; when PSNR approaches the stability, PSNR of the “MOM+ALR” and “ALR” are larger (see fig. 4.5.(a)). In video “walking to bed person”, The person is moving in video “walking to bed person” when frame number is 120 – 300. When person is walking to bed, "REG + MOM + ALR", "REG + ALR", and "MOM + ALR" have a better effect; when PSNR approaches the stability, PSNR of “ALR” is the largest. This shows that the "MOM + ALR" is the best in the dynamic video; the "ALR" is the best in static video, and "MOM + ALR" also has a good effect in static video. Based on the above, we should use the "MOM + ALR"

The corrected images are shown in Fig.4.6 and Fig. 4.7. The accuracy of the LMS method is low. Although gain and offset can correct in the right trend (see fig. 4.6), gain and offset cannot reach precise effect such as two-point correction (see fig. 4.7).

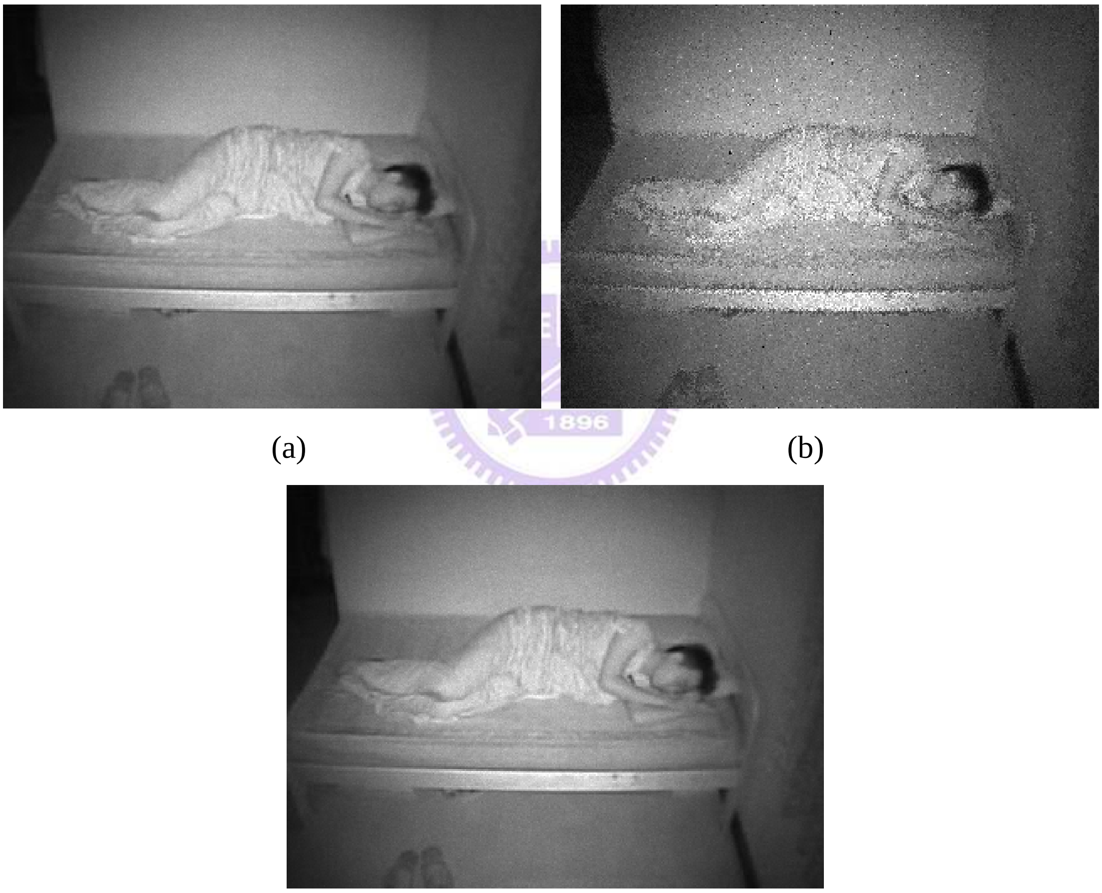


(a)

(b)

Fig. 4.5. PSNR for the LMS algorithm and its proposed modifications versus frame number. (a) Sleeping person. (b) Walking to bed person. 'NOR' indicates LMS

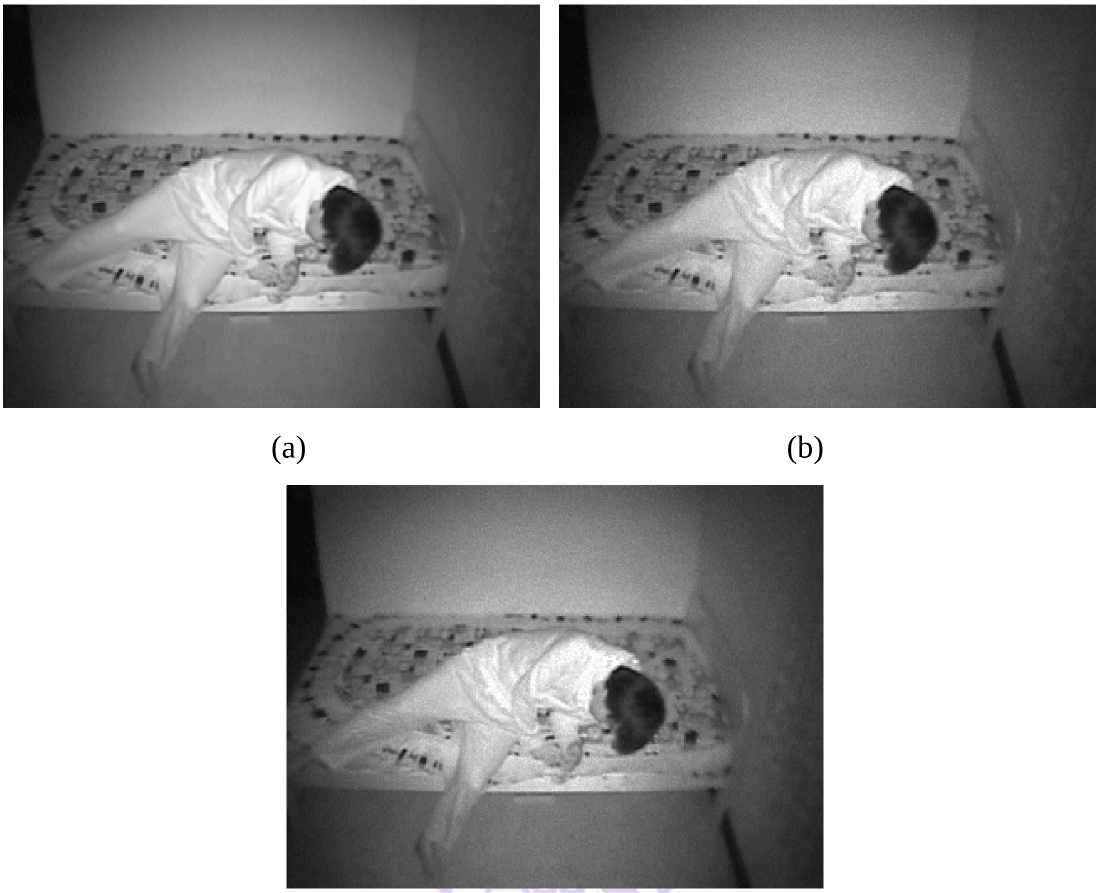
method, ‘MOM’ indicates LMS method plus momentum, ‘REG’ indicates LMS method plus regularization, ‘ALR’ indicates LMS method plus adaptive learning rate, ‘MOM+REG’ indicates LMS methods plus momentum and regularization, ‘REG+ALR’ indicates LMS methods plus regularization and adaptive learning rate, ‘MOM+ALR’ indicates LMS methods plus momentum and adaptive learning rate, and ‘ALR+MOM+REG’ indicates LMS methods plus momentum, regularization and adaptive learning rate.



(a) (b)

(c)

Fig. 4.6. (a) Original image of frame 3. (b) The corrected frame of the frame 3. (c) The corrected frame of the frame 300.



(a)

(b)

(c)

Fig. 4.7. Frame 240 of the “walking to bed person” (a) Original image of frame 3. (b) Corrupted image. (c) Corrected image with the proposed NUC method using adaptive learning rate plus momentum and regularization.

4.2 Results of Bad Pixel Correction

For the evaluation of noise suppression algorithms, we will use two noise models to simulate different models of distortions.

(I) Impulsive noise:

Let $x_i = \{x_{i1}, x_{i2}, x_{i3}\}$ denote the original pixel and let x_i' denote the pixel corrupted by the noise process. Then the image pixels are distorted according to the following scheme:

$$x_i' = \begin{cases} \{v_{i1}, x_{i2}, x_{i3}\} & \text{with probability } p_1 p \\ \{x_{i1}, v_{i2}, x_{i3}\} & \text{with probability } p_2 p \\ \{x_{i1}, x_{i2}, v_{i3}\} & \text{with probability } p_3 p \\ \{v_{i1}, v_{i2}, v_{i3}\} & \text{with probability } p_4 p \end{cases} \qquad (4.3)$$

where v_{i1}, v_{i2}, v_{i3} are independent and equal to 0 or 255. p is the sample corruption probability and p_1, p_2, p_3 are corruption probabilities of each color channel, so that $\sum_{k=1}^{4} P_k = 1$. In this work, noise model will generally denote the case with $P_k = 0.25$, $k = 1, ..., 4$.

(II) Realistic noise:

In this noise model, we use fig. 3.3. (a) as sample to calculate the proportion of each noise type which is detected by sensor, and we add realistic noise with the ratio of noise types. We can accordance with the proportional increase or decrease in noise until the sample corruption probability p is our settings. The image pixels are distorted like Eq. (4.3), and each neighboring noise have the same distortion.

In this experiment, we add the impulse noise with 5%, 10%, 15%, and 20%; and we add the realistic noise with 1%, 4%, 7%, 10%. Because noise density is less than 1% in Figure 3.3, we add noise proportion of the realistic noise is smaller than the noise proportion of the realistic noise.

For the measurement of the restoration quality, we employ the Pseudo Signal to Noise Ratio (PSNR) performance metric, which is based on the Root Mean Square Error (RMSE). The RMSE and PSNR are defined as:

$$RMSE = \sqrt{\frac{1}{N \times M \times Q} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{q=1}^{Q} \left(x^{q}\left(i, j\right) - o^{q}(i, j) \right)^{2}} \tag{4.4}$$

$$PSNR = 20 \times \log \left(\frac{255}{RMSE} \right) \tag{4.5}$$

where M , N are the image dimensions, Q is the number of channels of the image ($Q = 3$ for color image), and $o^{q}(i, j)$ and $x^{q}\left(i, j\right)$ denote the q -th component of the original image vector and the filtered image, at pixel position $\left(i, j\right)$, respectively.

For the evaluation of the detail preservation capabilities of the proposed filtering design the mean absolute error (MAE) has been used

$$MAE = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{q=1}^{Q} \left| x^{q}\left(i, j\right) - o^{q}(i, j) \right|}{N \times M \times Q} \tag{4.6}$$

Since RGB is not a perceptually uniform space in the sense that differences between colors in this space do not correspond to color differences perceived by humans, the restoration errors are often analyzed using the perceptually uniform color spaces. In this paper, we will use the CIE LUV color space and the normalized color difference (NCD) defined as:

$$NCD_{Lab} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \Delta E_{lab}}{\sum_{i=1}^{N} \sum_{j=1}^{M} \Delta E_{Lab}^{*}} \tag{4.7}$$

$$\Delta E_{lab} = \left| (\Delta L^{*})^{2} + (\Delta a^{*})^{2} + (\Delta b^{*})^{2} \right|^{1/2} \tag{4.8}$$

$$\Delta E_{Lab}^{*} = \left| (L^{*})^{2} + (a^{*})^{2} + (b^{*})^{2} \right|^{1/2} \tag{4.9}$$

where ΔE_{lab} denotes the perceptual color error and ΔE_{Lab}^{*} is the norm or magnitude of the original image color vector in the $L^{*}a^{*}b^{*}$ color space.

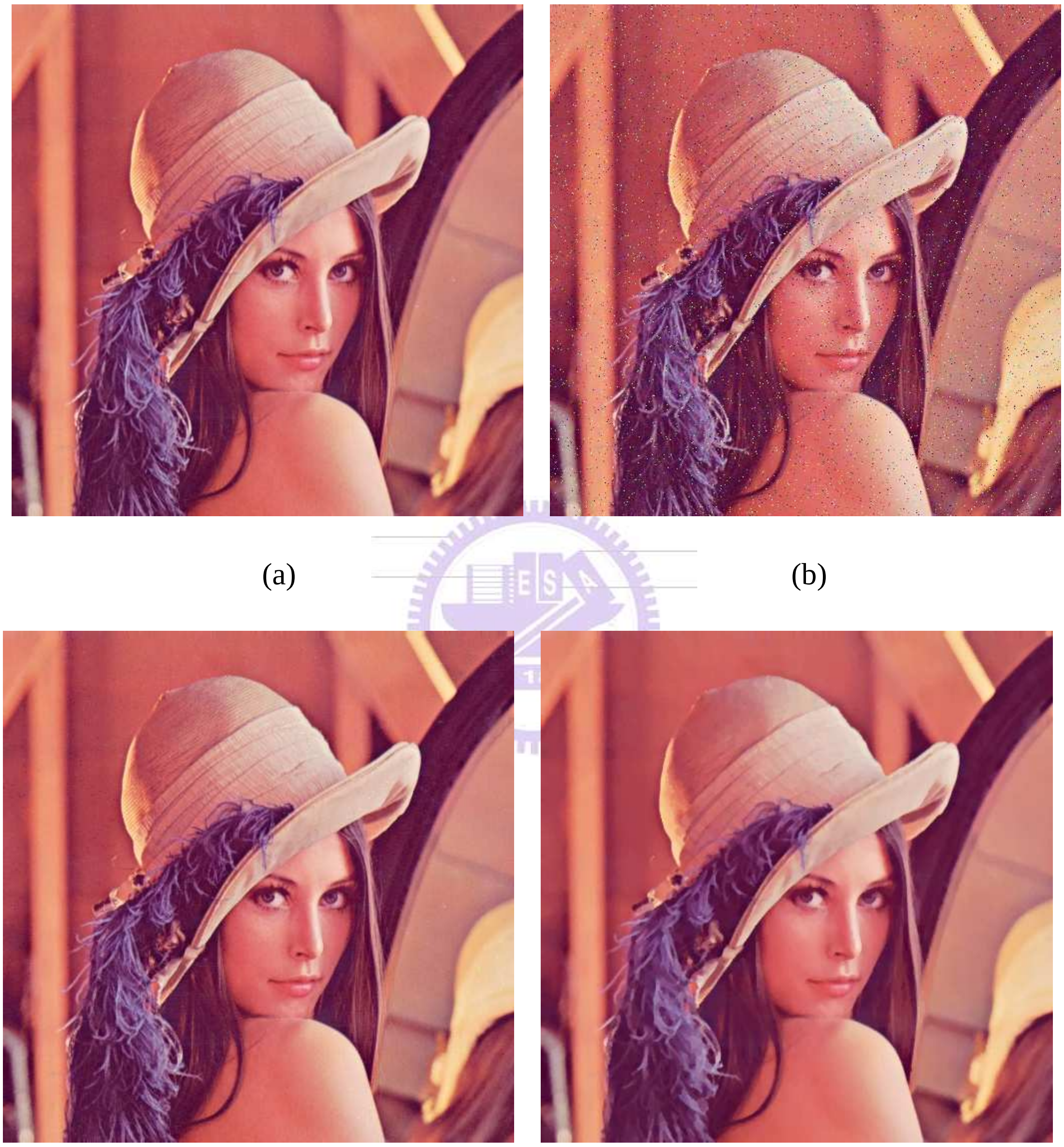
Figure 4.8 shows the results of each filter in 5% impulse noise density, and table 4.1 shows the performances with each sample corruption probability of impulse noise; Figure 4.7 shows the results of each filter in 1% realistic noise density, and table 4.2 shows the performances with each sample corruption probability of realistic noise. In order to quickly see the advantages and disadvantages of each method, I will normalize parameters which are RMSE, MAE, and NCD. The parameters are normalized numbers which are between 0 and 1, and the larger value for the parameters indicates better performances. Equation is expressed as follows:

$$RMSE_{normalize} = \frac{\max(RMSE) - RMSE(i)}{\max(RMSE) - \min(RMSE)} \tag{4.10}$$

$$MAE_{normalize} = \frac{\max(MAE) - MAE(i)}{\max(MAE) - \min(MAE)} \tag{4.11}$$

$$NCD_{normalize} = \frac{\max(NCD) - NCD(i)}{\max(NCD) - \min(NCD)} \tag{4.12}$$

According to Table 4.1, when the noise is impulse noise, the performance of switch filters is better than the non-switch filter (VMF, BVDF, DDF). Although our propose method is not the best method, the performance of our proposed method is still above average; when the noise is realistic noise, no matter what the sample corruption probability p is, the performance of our proposed method is the best. The fig. 4.9. show that the bad pixels which are blobs are corrected by our proposed method, and others filter cannot correct the bad pixels in blobs. In the zoomed images (see fig. 4.6 (c) – (k)), we can see that some bad pixels are similar to neighbor pixels, and the bad pixels cannot be detected by our proposed method. The reason is that we use the same threshold d value in the different working window. If we can automatically select the threshold, the problem may be solved. The image is blurred in the detail part, because we detect dead pixels with 5 × 5 working window. We hope that if we can detect dead pixels with 3 × 3 working window, and we use the larger working window when the bad pixels are blobs.



(a) (b)

(c) (d)

(e)

(f)

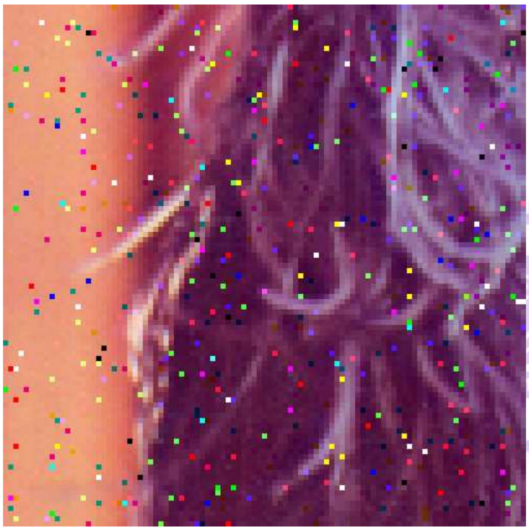
(g)

(h)

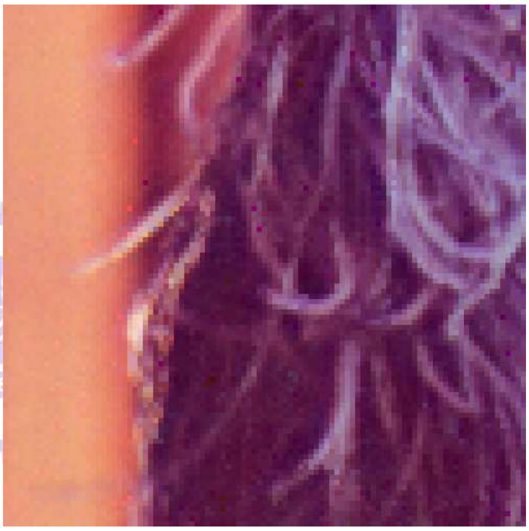
(i)

(j)

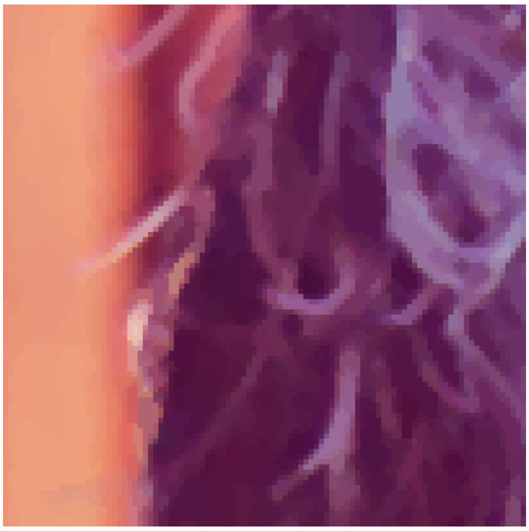
(k)



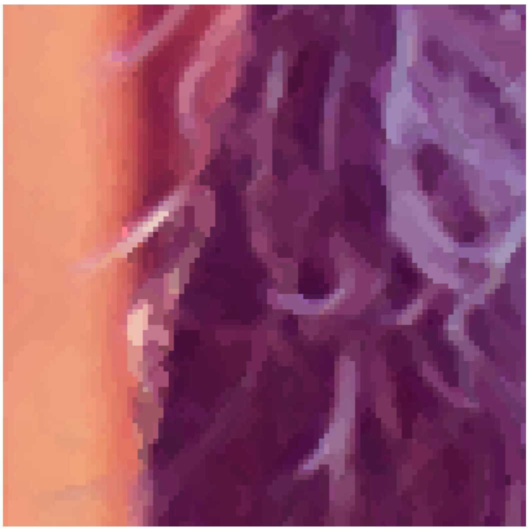
(l)



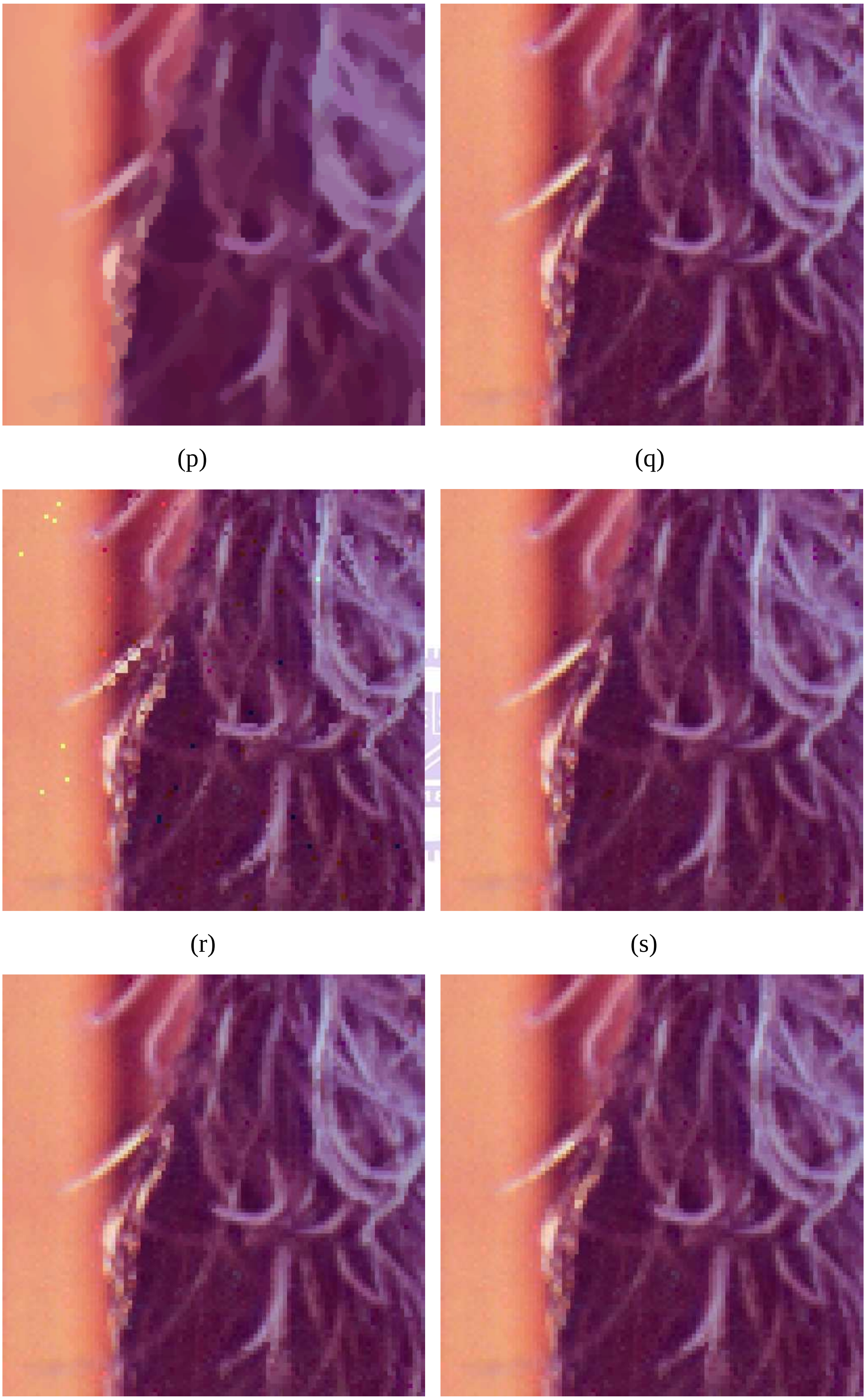
(m)



(n)



(o)



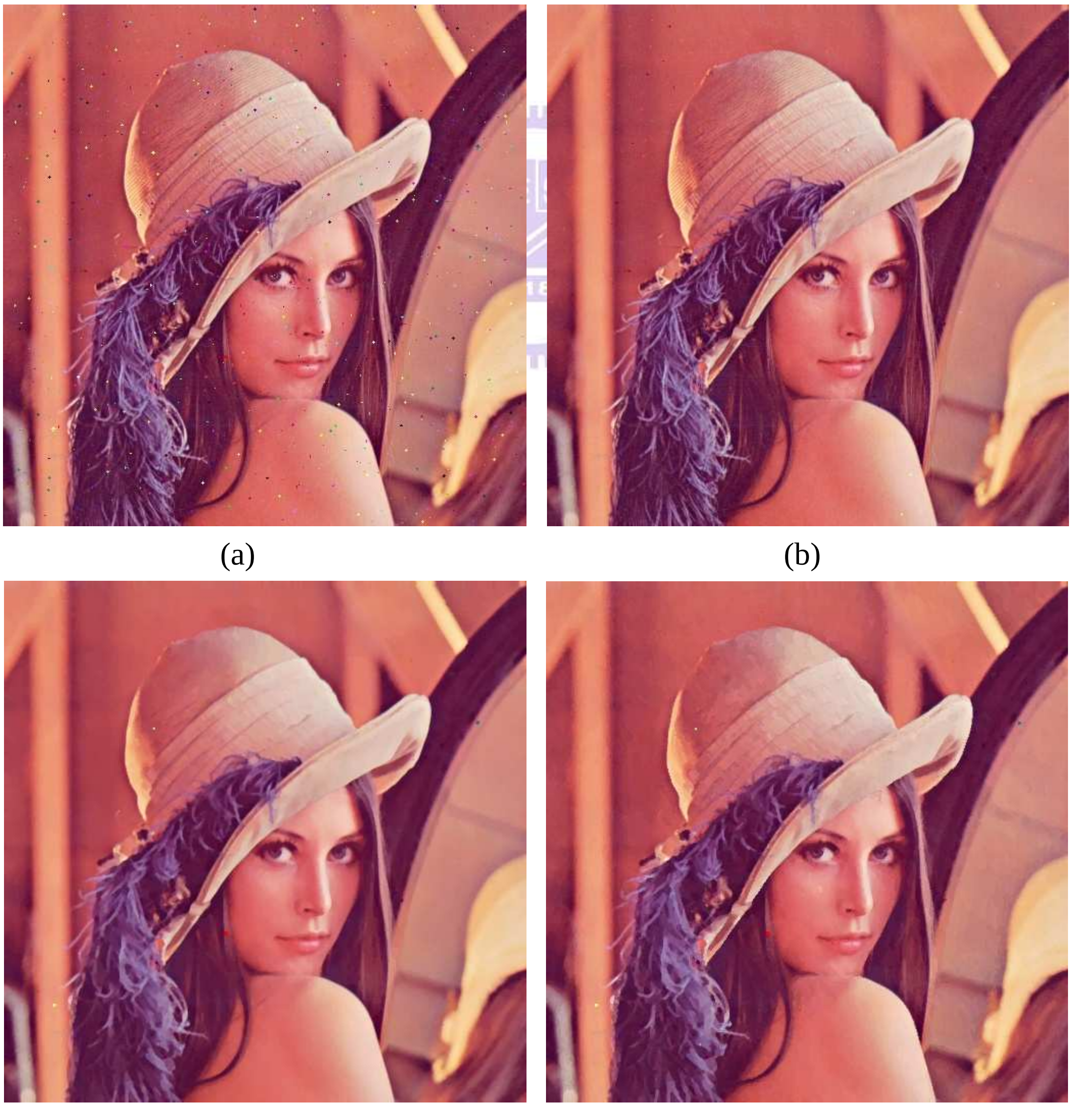
(p) (q)

(r) (s)

(t) (u)

Fig. 4.8. Bad pixel correction results of Lena image filtered by different impulse noise

filters. (a) Original image. (b) Corrupted image with 5% impulse noise. (c)－(k) are filtering results. Image filtering results filtered by (c) our proposed filter. (d) Vector median filter (VMF). (e) Basic vector directional filter (BVDF). (f) Directional distance Filter (DDF). (g) Fast Peer Group Filter (FPGF). (h) Fuzzy Modified Peer Group Filter (FMPGF). (i) Fast similarity-based impulsive noise removal vector filter (FSVF). (j) Fuzzy metric FSVF (FMFSVF). (k) Fuzzy Peer Group Averaging Filter (FPGA). (l) is zoomed parts of (b). (m)－(u) are zoomed “Lena” filtering results. Zoomed results filtered by (l) our proposed filter. (m) VMF. (n) BVDF. (o) DDF. (p) FPGF. (q) FMPGF. (r) FSVF. (s) FMFSVF. (t) FPGA.



(a) (b)

(c) (d)

(e)

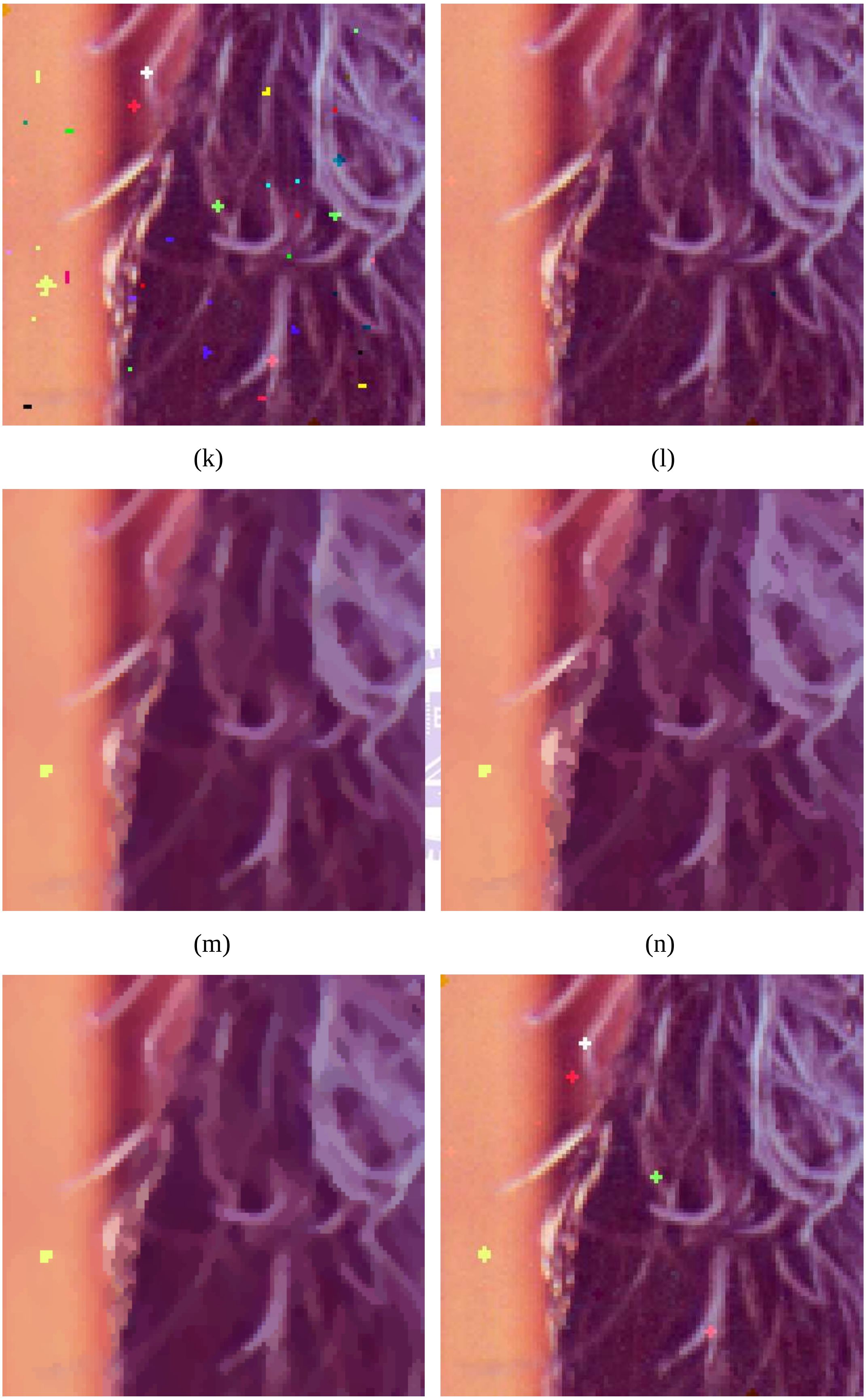
(f)

(g)

(h)

(i)

(j)



(k)

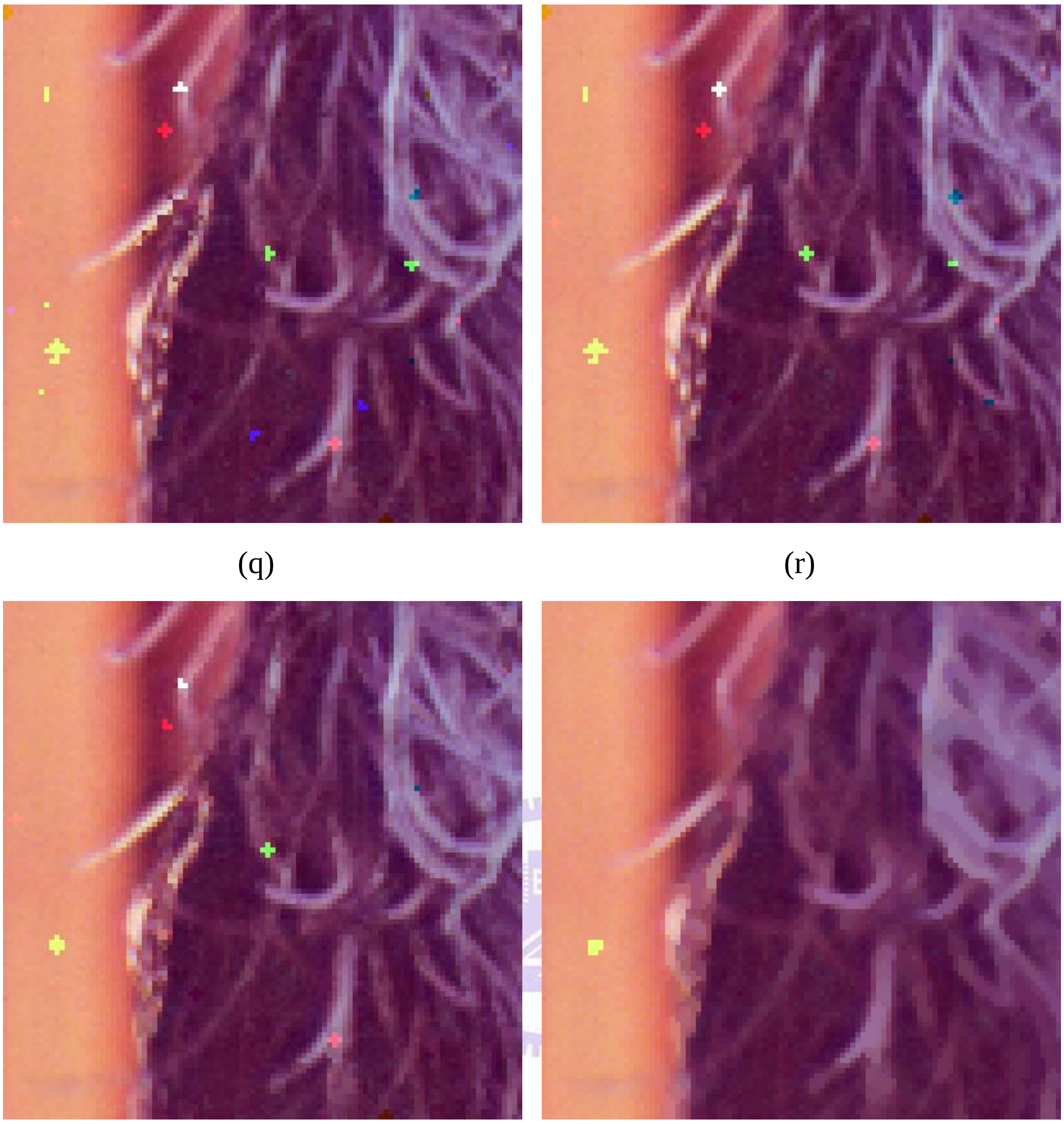
(l)

(m)

(n)

(o)

(p)



(q) (r)

(s) (t)

Fig. 4.9. Bad pixel correction results of Lena image filtered by different realistic noise filters. (a) Corrupted image with 1% realistic noise. (b) – (j) are filtering results. Image filtering results filtered by (b) our proposed filter. (c) Vector median filter (VMF). (d) Basic vector directional filter (BVDF). (e) Directional distance Filter (DDF). (f) Fast Peer Group Filter (FPGF). (g) Fuzzy Modified Peer Group Filter (FMPGF). (h) Fast similarity-based impulsive noise removal vector filter (FSVF). (i) Fuzzy metric FSVF (FMFSVF). (j) Fuzzy Peer Group Averaging Filter (FPGA). (k) is zoomed parts of (a). (l) – (t) are zoomed “Lena” filtering results. Zoomed results filtered by (l) our proposed filter. (m) VMF. (n) BVDF. (o) DDF. (p) FPGF. (q) FMPGF. (r) FSVF. (s)

FMFSVF. (t) FPGA.

Table 4.1

The evaluation results of Lena image filtered by the following filter:

(a) Corrupted image with 5% impulse noise.

Filter	RMSE	MAE	NCD	RMSE-normalized	MAE-normalized	NCD-normalized	Sum
Our method	3.5957	0.4377	0.0048	0.8865	0.9701	0.9574	2.8140
FIVF	2.5723	0.3069	0.0033	0.9703	0.9939	0.9935	2.9576$_3$
FMPGF	6.0304	0.6896	0.0080	0.6872	0.9242	0.8834	2.4948
FPGA	3.1922	0.4384	0.0039	0.9195	0.9699	0.9782	2.8677
FSVF	2.3992	0.2756	0.0034	0.9845	0.9996	0.9908	2.9749$_2$
PGF	2.2093	0.2735	0.0030	1.0000	1.0000	1.0000	3.0000$_1$
BVDF	14.4265	5.7611	0.0456	0.0000	0.0000	0.0000	0.0000
DDF	14.0460	5.2382	0.0446	0.0311	0.0953	0.0249	0.1513
VMF	6.6024	3.8029	0.0356	0.6404	0.3568	0.2344	1.2317

(b) Corrupted image with 10% impulse noise.

Filter	RMSE	MAE	NCD	RMSE-normalized	MAE-normalized	NCD-normalized	Sum
Our method	4.2708	0.7164	0.0076	0.9209	0.9719	0.9638	2.8566
FIVF	3.3270	0.5535	0.0060	0.9820	0.9966	0.9941	2.9727$_2$
FMPGF	7.2660	1.1532	0.0122	0.7270	0.9054	0.8748	2.5072
FPGA	3.8339	0.7040	0.0066	0.9492	0.9738	0.9824	2.9053
FSVF	3.7577	0.5679	0.0077	0.9541	0.9945	0.9613	2.9098$_3$
PGF	3.0491	0.5314	0.0057	1.0000	1.0000	1.0000	3.0000$_1$
BVDF	18.4944	7.1068	0.0578	0.0000	0.0000	0.0000	0.0000
DDF	18.1712	6.5833	0.0567	0.0209	0.0796	0.0208	0.1214
VMF	6.8445	3.9487	0.0367	0.7543	0.4803	0.4039	1.6384

(c) Corrupted image with 15% impulse noise.

Filter	RMSE	MAE	NCD	RMSE-normalized	MAE-normalized	NCD-normalized	Sum
Our method	4.8266	0.9825	0.0103	0.8183	0.9528	0.9397	2.7109
FIVF	4.0217	0.8102	0.0085	0.9601	0.9929	0.9937	2.9467$_2$
FMPGF	8.6644	1.6116	0.0177	0.1424	0.8066	0.7164	1.6654
FPGA	4.6494	0.9683	0.0099	0.8495	0.9561	0.9515	2.7572$_3$
FSVF	5.1482	0.9003	0.0133	0.7617	0.9719	0.8492	2.5829
PGF	3.7952	0.7795	0.0083	1.0000	1.0000	1.0000	3.0000$_1$
BVDF	9.4727	5.0826	0.0414	0.0000	0.0000	0.0000	0.0000
DDF	8.6538	4.5245	0.0403	0.1442	0.1297	0.0335	0.3074
VMF	7.0543	4.0807	0.0378	0.4260	0.2328	0.1091	0.7679

(d) Corrupted image with 20% impulse noise.

Filter	RMSE	MAE	NCD	RMSE-normalized	MAE-normalized	NCD-normalized	Sum
Our method	5.2512	1.2236	0.0127	0.9350	0.9659	0.9608	2.8618
FIVF	4.3763	1.0194	0.0110	1.0000	1.0000	1.0000	3.0000_1
FMPGF	9.8929	2.0342	0.0229	0.5902	0.8306	0.7281	2.1489
FPGA	5.1914	1.2713	0.0125	0.9394	0.9580	0.9659	2.8633_3
FSVF	7.4217	1.3642	0.0225	0.7738	0.9425	0.7370	2.4533
PGF	4.4594	1.0709	0.0113	0.9938	0.9914	0.9931	2.9783_2
BVDF	17.8379	7.0108	0.0549	0.0000	0.0000	0.0000	0.0000
DDF	17.4112	6.4963	0.0538	0.0317	0.0859	0.0243	0.1419
VMF	7.3442	4.2557	0.0390	0.7795	0.4598	0.3622	1.6016

Table 4.2

The evaluation results of Lena image filtered by the following filter:

(a) Corrupted image with 1% realistic noise.

Filter	RMSE	MAE	NCD	RMSE-normalized	MAE-normalized	NCD-normalized	Sum
Our method	2.4088	0.1387	0.0017	1.0000	1.0000	1.0000	3.0000$_1$
FIVF	4.0657	0.2834	0.0031	0.6786	0.9667	0.9595	2.6048$_2$
FMPGF	7.2351	0.4181	0.0053	0.0638	0.9356	0.8969	1.8964
FPGA	5.5436	1.6414	0.0137	0.3919	0.6540	0.6612	1.7070
FSVF	6.5536	0.3204	0.0048	0.1960	0.9581	0.9131	2.0672
PGF	6.3073	0.3044	0.0041	0.2437	0.9618	0.9312	2.1368$_3$
BVDF	7.5638	4.4813	0.0372	0.0000	0.0000	0.0000	0.0000
DDF	6.7914	3.9355	0.0360	0.1498	0.1257	0.0342	0.3098
VMF	6.4934	3.6915	0.0350	0.2076	0.1819	0.0638	0.4534

(b) Corrupted image with 4% realistic noise.

Filter	RMSE	MAE	NCD	RMSE-normalized	MAE-normalized	NCD-normalized	Sum
Our method	3.6473	0.4080	0.0046	1.0000	1.0000	1.0000	3.0000$_1$
FIVF	7.0845	0.8450	0.0087	0.6830	0.8974	0.8797	2.4601$_2$
FMPGF	13.1474	1.4814	0.0191	0.1238	0.7481	0.5775	1.4494
FPGA	7.0037	1.9142	0.0164	0.6904	0.6465	0.6554	1.9923$_3$
FSVF	14.4900	1.4873	0.0214	0.0000	0.7467	0.5104	1.2570
PGF	13.7416	1.3512	0.0175	0.0690	0.7786	0.6243	1.4720
BVDF	8.5840	4.6686	0.0390	0.5447	0.0000	0.0000	0.5447
DDF	7.8682	4.1339	0.0378	0.6107	0.1255	0.0325	0.7687
VMF	7.7489	3.8851	0.0368	0.6217	0.1839	0.0639	0.8695

(c) Corrupted image with 7% realistic noise.

Filter	RMSE	MAE	NCD	RMSE-normalized	MAE-normalized	NCD-normalized	Sum
Our method	4.1995	0.6394	0.0067	1.0000	1.0000	1.0000	3.0000$_1$
FIVF	8.8677	1.2205	0.0131	0.6354	0.8655	0.8453	2.3462$_3$
FMPGF	16.6498	2.3675	0.0305	0.0275	0.5999	0.4214	1.0489
FPGA	7.2998	1.3124	0.0117	0.7578	0.8442	0.8782	2.4802$_2$
FSVF	17.0025	2.1611	0.0326	0.0000	0.6477	0.3705	1.0182
PGF	14.9220	4.9589	0.0478	0.1625	0.0000	0.0000	0.1625
BVDF	9.5716	4.8688	0.0409	0.5804	0.0209	0.1686	0.7698
DDF	8.8976	4.3302	0.0398	0.6330	0.1455	0.1949	0.9735
VMF	8.5891	4.0665	0.0387	0.6571	0.2066	0.2224	1.0861

(d) Corrupted image with 10% realistic noise.

Filter	RMSE	MAE	NCD	RMSE-normalized	MAE-normalized	NCD-normalized	Sum
Our method	4.7496	0.8634	0.0086	1.0000	1.0000	1.0000	3.0000$_1$
FIVF	10.2902	1.5584	0.0172	0.7343	0.8559	0.8452	2.4355$_3$
FMPGF	19.3414	3.2260	0.0403	0.3003	0.5101	0.4321	1.2425
FPGA	8.4397	1.5953	0.0147	0.8231	0.8482	0.8907	2.5620$_2$
FSVF	25.6040	4.5280	0.0644	0.0000	0.2402	0.0000	0.2402
PGF	17.9512	5.6863	0.0549	0.3670	0.0000	0.1715	0.5384
BVDF	10.4649	5.0824	0.0429	0.7259	0.1252	0.3863	1.2375
DDF	9.7540	4.5290	0.0417	0.7600	0.2400	0.4066	1.4066
VMF	9.5453	4.2751	0.0407	0.7700	0.2926	0.4256	1.4882

Chapter 5 Conclusion

In this thesis, we employ two-point correction method and LMS method to correct non-uniformity among pixels of NIR sensor. Two-point correction is a highly accurate method, unfortunately, he needs sophisticated instruments to measure the reference image, and correction parameters which were measured before cannot meet the correct situation when the system is in use of increased working hours. On the other hand, LMS method only need the readout infrared data captured by the imaging system and compensate the non-uniform response of pixels during its normal operation. We changed the parameters set so that LMS can be adapted to all the circumstances. Furthermore, we use peer group filter which can adjust the size of the working window automatically. We use 3×3 window as default working window for sharpness maintenance, if the small window does not correct a bad pixel, the window size will increase automatically to enhance the correction capability. Although the detail of the image may be blurred, most of the bad pixels can be corrected.

In the future, more advanced NUC method will be investigated to improve NIR sensor performance. We can the above two methods in the application, two-point correction do not need to re-measurement gain and offset, we use the LMS method to make it without sophisticated measurement. Furthermore, in order to apply in real time, If we can find out the location of the dead pixels in advance, we can save a lot of time in the detection, We only need regular dead pixel detection to find out dead pixels generated due to mechanical aging, we can achieve the purpose of quickly and effectively.

References

[1] A. F. Milton, F. R. Barone, and M. R. Kruer, “Influence of nonuniformity on infrared focal plane array performance,” *Opt. Eng.*, vol. 24, pp. 855–862, 1985.

[2] P. Fillon, A. Combette P. Tribolet, “Cooled IR detectors calibration analysis and optimization,” *Proc. of SPIE*, vol. 5784, pp. 343–354, 2005.

[3] Yung-Fa Wu**,** “Edge Detection Based SWIR Image Bad Pixel Detection and Correction,’’ *Mater Thesis*, Inst. of Electrical and Control Eng., National Chiao Tung University, Taiwan, July 2011.

[4] S. N. Torres, E. B. Vera, and S. K. S. R. A. Reeves, “Adaptive scene-based non-uniformity correction method for infrared focal plane arrays,” in Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XIV, G. C. Holst, ed., *Proc. of SPIE*, vol. 5076, pp.130–139, 2003.

[5] Plataniotis KN, Venetsanopoulos AN. In *Color image processing and applications. Berlin*: Springer; 2000.

[6] C Boncelet. Image noise models. In: A Bovik, editor. *Handbook of image and video processing*. New York: Academic Press, 2000.

[7] Lopez DB, Mendoza FH, Ramirez JM. Noise in color digital images. In: *Proceedings of the midwest symposium on circuits and systems*, pp. 403 – 6, Aug. 1999.

[8] J. Zheng, K. P. Valavanis, and J. M. Gauch, “Noise removal from color images,” *.J. intell. Robot. Syst., vol.* 7, no. 3, pp. 257-285, June 1993.

[9] J. Astola, P. Haavisto, and Y. Neuvo, “Vector median filters,” *Proc. of IEEE*, vol. 78, no. 4, pp. 678-689, Apr. 1990.

[10] P. E. Trahanias and A. N.Venetsanopoulos, “Vector directional filters—a new class of multichannel image processing filters,” *IEEE Trans. Image Processing*,

vol. 2, no. 4, pp. 528–534, 1993.

[11] D. Karakos and P. Trahanias, “Generalized multichannel image-filtering structure,’’ *IEEE Trans. Image Processing*, vol. 6, no. 7, pp. 1038–1045, July 1997.

[12] R. Lukac, V. Fischer, G. Motyl, and M. Drutarovsky, “Adaptive video filtering framework,” *Int. J. Imag. Syst. Technol.*, vol. 14, no. 6, pp. 223–237, Dec. 2004.

[13] R. Lukac, “Adaptive vector median filtering,” *Pattern Recognit. Lett.*, vol. 24, pp. 1889–1899, Aug. 2003.

[14] R. Lukac, “Adaptive color image filtering based on center-weighted vector directional filters,” *Multidimen. Syst. Signal Process.*, vol. 15, no. 2, pp. 169 – 196, Apr. 2004.

[15] M.E. Celebi, Y, A. Aslandorgan, ”Robust switching vector median filter for impulsive noise removal,” J. Electronic Imaging, vol.17, no.4, pp.1-9, 2008.

[16] Y. Deng, C. Kenney, M. S. Moore, and B. S. Manjunath, “Peer group filtering and perceptual color image quantization,” in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, 1999, pp. 21–24.

[17] B. Smolka and A. Chydzinski, “Fast detection and impulsive noise removal in color images,” *Real-Time Imag.*, vol. 11, no. 5–6, pp. 389–402, Oct.–Dec. 2005.

[18] J. G. Camarena, V. Gregori, S. Morillas, and A. Sapena, “Fast detection and removal of impulsive noise using peer groups and fuzzy metrics,” *J. Vis. Commun. Image Represent.*, vol. 19, no. 1, pp. 20–29, 2008.

[19] S. Morillas, V. Gregori, G. Peris-Fajarns, and P. Latorre, “A fast impulsive noise color image filter using fuzzy metrics,” *Real-Time Imag.*,vol. 11, no. 5 – 6, pp. 417 – 428, 2005.

[20] R. Lukac and K. N. Plataniotis, “cDNA microarray image segmentation using root signals,” *Int. J. Imag. Syst. Technol.*, vol. 16, no. 2, pp. 51 – 64,Apr. 2006.

[21] B. Smolka, K. N. Plataniotis, R. Lukac and A. N. Venetsanopoulos, ‘Similarity based impulsive noise removal in color images’. *Proc. IEEE Int. Conf. Image Process.*, Barcelona, Spain, 2003, pp. 105–108

[22] B. Smolka, R. Lukac, A. Chydzinski, K. N. Plataniotis, and W. Wojciechowski, “Fast adaptive similarity based impulsive noise reduction filter,” *Real-Time Imag.*, vol. 9, no. 4, pp. 261–276, Aug. 2003.

[23] B. Smolka, A. Chydzinski, K. N. Plataniotis, and A. N. Venetsanopoulos, “New filtering technique for the impulsive noise reduction in color images,” *Mathematical Problems in Engineering*, vol. 2004, no. 1, pp. 79-91.

[24] B. Smolka, A. Chydzinski, K. Wojciechowski, K. N. Plataniotis, and A. N. Venetsanopoulos, “Self-adaptive algorithm for impulsive noise reduction in color images,” *Pattern Recognit.*, vol. 35, no. 8, pp. 1771–1784, 2002.

[25] S. Morillas, V. Gregori, and A. Hervas, “Fuzzy peer groups for reduction mixed Gaussian-impulsive noise from color images,” *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1452–1466, Jul. 2009.