A New Iterative Procedure

for
Removing Impulse Noise

by

HU, Chen

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Mathematics

Abstract

Abstract of thesis entitled:

A New Iterative Procedure

for
Removing Impulse Noise

Submitted by HU, Chen

for the degree of Master of Philosophy in Mathematics
at The Chinese University of Hong Kong in June 2004

This thesis proposes a two-phase iterative method for the random-valued impulse noise removal. In the first phase, we use the adaptive center-weighted median filter to identify pixels which are likely to be corrupted by noise (noise candidates). In the second phase, these noise candidates are restored using a specialized regularization method which allows edges and noise-free pixels to be preserved. These two phases are applied in an alternate way. Simulation results indicate that the proposed method are significantly better than those using just nonlinear filters or edge-preserving regularization only. The second phase is equivalent to solving a one-dimensional nonlinear equation for each noise candidate. We describe a simple secant-like method to solve these equations. It converges faster than Newton's method, requiring the same number of function and derivative evaluations per iteration.

摘要

香港中文大學碩士論文摘要

論文題目：

一種新的去除隨機脈衝噪聲的迭代過程

胡晨

二零零四年六月

本論文提出了一種二階段迭代法用來去除隨機脈衝噪聲。在第一階段，用適應中間加權中值濾波法來確認那些像被噪聲干擾的位置（稱為疑似噪聲）。在第二階段，這些疑似噪聲被特殊正則化方法還原。這種特殊正則化方法可以保護邊界和非噪聲部分不被破壞。這兩個階段被不斷的交替使用。實驗結果證明這種方法比只用非線性濾波法或者正則化方法得出結果有明顯的提高。在第二階段，等價於在每一個疑似噪聲像素解一個一維非線性方程。我們提出一種簡單的似割線法來解這些方程。這種方法比牛頓法收斂的速度要快，同時在每一步和牛頓法要計算的函數值和導數值是一樣的。

ACKNOWLEDGMENTS

I am greatly indebted to my supervisors, Prof. Raymond H. Chan for his continual guidance and constant encouragement and help throughout the period of my graduate studies. Moreover, I would like to thank Prof. S. H. Lui and Dr. M. Nikolova for their help in this project. Thanks also go to Profs. Philippe G. Ciarlet, J. Zou, and K. M. Yeung, for their excellent teaching in my graduate courses. Also I am grateful to my colleagues Mr. Z. J. Bai, Y. Chen, C. Y. Wong, K. T. Ling, Y. H. Tam, C. W. Ho, K. C. Ma, and Y. S. Wong for their many helpful discussions and keeping an excellent learning environment in our office. In addition, I would like to thank my friends, Mr. G. Y. Li, C. G. Liu, R. Zang, and Y. B. Zhao for their sincere care about my graduate life.

Finally, I thank my parents for their encouraging support, which enables me to devote my energy to my thesis.

To

My Parents

Contents

Chapter 1

Introduction

In image processing, images are often corrupted by noise during image acquisition or digitization and transmission principally due to interference in the channel used for transmission. There are many noise models in image processing applications. Particularly, impulse noise and Gaussian noise are among the most common found in image processing applications [1]. We will introduce these two noise models in the later section. In this thesis, we focus on removing random-valued impulse noise which is a kind of impulse noise from noisy image.

1.1 Noise Model

1.1.1 Impulse Noise

Impulse noise is frequently caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or transmission in a noisy channel.

We first give the definitions of the impulse noise. Let x_{ij} be the gray level of a true image $\mathbf{x}$ at pixel location (i, j) and $[n_{\min}, n_{\max}]$ be the dynamic range of $\mathbf{x}$.

If we let y_{ij} be the gray level of the impulse noisy image $\mathbf{y}$ at pixel (i, j), then

$$y_{ij} = \begin{cases} r_{ij}, & \text{with probability } p, \\ x_{ij}, & \text{with probability } 1 - p, \end{cases} \tag{1.1}$$

where $r_{ij} \in [n_{\min}, n_{\max}]$ are random numbers and p is the noise ratio. Our goal is to obtain a restored image $\hat{\mathbf{x}}$ from the noisy image $\mathbf{y}$. There are two kinds of impulse noise in image processing applications. One is the fixed-valued (salt-and-pepper) impulse noise, noisy pixels y_{ij} can take either $n_{\min}$ or $n_{\max}$, see [2]. Another is the random-valued impulse noise where r_{ij} can be any numbers r_{ij} between $n_{\min}$ and $n_{\max}$, see [3]. Cleaning such noise is far more difficult than cleaning fixed-valued impulse noise since for the latter, the differences in gray levels between a noisy pixel and its noise-free neighbors are significant most of the times.

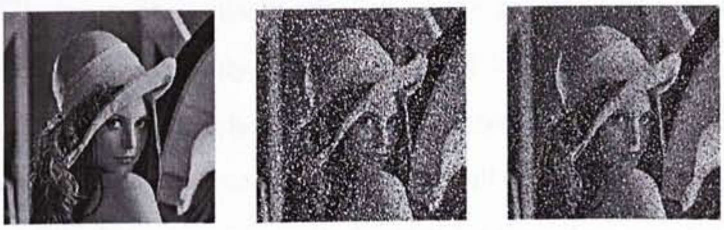


Figure 1.1: (**Left-to-Right:**) The noise-free (original) Lena image, the noisy image corrupted by fixed-valued impulse noise, the noisy image corrupted by random-valued impulse noise.

1.2 Removing Impulse Noise

So far, many techniques have been proposed to remove impulse noise from the corrupted images [4]–[14], [3]. Most of them use the nonlinear filters to remove impulse noise [4]–[13], [3]. In recent years, detail-preserving variational method (DPVM) has been proposed to restore impulse noise [14].

1.2.1 Nonlinear Filter

There are many classical nonlinear filters, such as median filter, mean filter, rank-ordered mean filter, max and min filter etc. [1, 8]. The best-known filter is the median filter, which, as its name implies, replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel:

$$\hat{x}_{ij} = \text{median}\{y_{i-u,j-v} : -h \leq u, v \leq h\}$$

where $(2h+1)^2$ is the window size and $\hat{x}_{ij}$ is the gray level of the restored image $\hat{\mathbf{x}}$ at pixel (i, j).

Median filters are particularly effective in the presence of impulse noise [1]. However, since filters typically are implemented invariantly across the images, they also tend to modify pixels that are not affected by noise. In addition, when the noise ratio is high, they are prone to edge jitter, and that the details and edges of the original image are usually blurred by the filters, see [15] and Figure 1.2.

To improve performance, various decision-based filters have been proposed where possible noise pixels are first identified and then replaced by using the median filter. Examples of decision-based filters are the switch median filter [11], the adaptive median filter [2], the adaptive center-weighted median filter (ACWMF) [12], and the median filter based on homogeneity information [13]. These filters are good in locating the noise even in high noise ratio. However, the main drawback is that the replacement of the noisy pixels by the median filter entails blurring of details and edges, especially when the noise ratio is high.

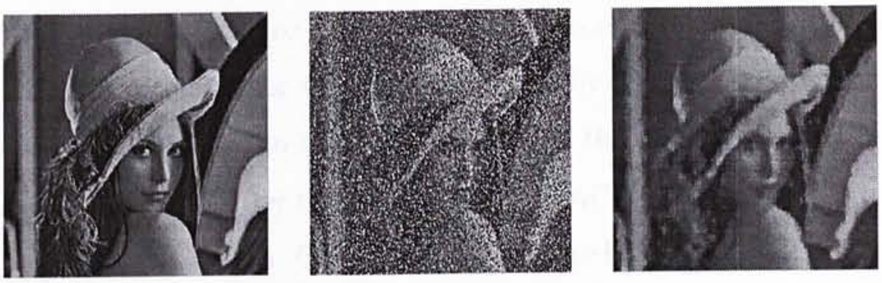


Figure 1.2: **(Left-to-Right:)** The original Lena image, The noisy image corrupted by 50% random-valued impulse noise, The restored image by using median filter with 7×7 window size.

1.2.2 Variational Method

For images corrupted by Gaussian noise, regularized least-squares methods, based on edge-preserving regularization functionals have been used successfully to preserve the edges and the details in the images [16]–[20]. But these methods fail in the presence of impulse noise because the noise is heavy tailed, and the restoration will alter considerable amount of pixels in the image, including those pixels which are not corrupted by the impulse noise.

Recently, a detail-preserving variational method (DPVM) has been proposed to restore impulse noise [14]. It uses a non-smooth data-fitting term (e.g. l_1) along with edge-preserving regularization. Because the l_1 data-fitting term leaves unchanged the pixels which are similar to their neighbors [14, 21], DPVM can remove impulse noise accurately. This variational method does not smear edges. However when removing noise patches involving several adjacent pixels, the distortion of some uncorrupted image pixels at the edges cannot be avoided.

To overcome the drawbacks, in this thesis, we propose a two-phase iterative method for removing random-valued impulse noise. First, noisy pixels are detected using ACWMF; then these pixels are selectively restored by DPVM.

These two phases are applied alternatively. Since in each iteration the edges and the details are preserved for the noise candidates by the regularization method, and no changes are made to the signal candidates, the performance of this combined method is much better than just using either ACWMF or DPVM, especially when the noise ratio is high. Our method can restore large patches of noisy pixels because it introduces pertinent prior information via the regularization term. It is most efficient to deal with high noise ratio, e.g. ratio as high as 50%.

Like other medium-type filters, ACWMF can be done very fast. The aim of DPVM is to minimize the objective functional consisting of a data-fitting term and an edge-preserving regularization term. It is equivalent to solving a system of nonlinear equations for those noise candidates. Usually, Newton's method is preferred to solve these nonlinear equations and a method to locate the initial guess is proposed [22]. However, the complexity of the algorithm is not good as expected. To improve timing, we propose a simple algorithm which we shall call secant-like method to solve these nonlinear equations. It converges in fewer number of iterations than Newton's method with both methods requiring the same number of function and derivative evaluations per iteration.

1.3 Organization of the Dissertation

The focus of this dissertation is on removing random-valued impulse noise, and on using the secant-like method to solve nonlinear equations.

- In Chapter 2, we review the ACWMF and DPVM method for restoring the random-valued impulse noise.
- In Chapter 3, we show our two-phase iterative method for cleaning random-valued impulse noise.

- In Chapter 4, we describe Newton's method, the secant method, and the secant-like method for solving the nonlinear equations in the second phase of our algorithm.

- In Chapter 5, we demonstrate the effectiveness of our two-phase iterative method using various images, and the secant-like method converges faster than Newton's method. The gain is greater for some parameters.

- In Chapter 6, we give the conclusions.

Chapter 2

Review of ACWMF and DPVM

2.1 Review of ACWMF

ACWMF is a good method for removing random-valued impulse noise when the noise ratio is not high, see [12]. Here we give a review of the filter.

Let the window size be $(2h+1)^2$ and $L = 2h(h+1)$. Then

$$x_{ij}^{2k} = \text{median}\{y_{i-u,j-v}, (2k)\Diamond y_{ij} : -h \leq u, v \leq h\},$$

where $2k$ is the weight given to pixel (i, j), and $\Diamond$ represents the repetiton operation. Clearly, x_{ij}^{0} is the output of the standard median filter, whereas x_{ij}^{2k} is the output of the identity filter when $k \geq L$.

For current pixel y_{ij} under consideration, we first define differences

$$d_k = |x_{ij}^{2k} - y_{ij}| \tag{2.1}$$

where $k = 0, 1, \cdots, L-1$. It is readily seen that $d_k \leq d_{k-1}$ for $k \geq 1$, see [23]. These differences provide information about the likelihood of corruption for the current pixel. For example, if d_{L-1} is large, then the current pixel is not only the smallest or the largest one among the observation samples within the window, but also very likely contaminated by impulse noise. On the other hand, in the case where d_0 is small, the current pixel may be considered as signal and be left unchanged in the filtering. Together, the differences d_0 through d_{L-1} reveal even more information about the presence of a corrupted pixel.

To determine whether the current pixel (i, j) is corrupted, a set of thresholds T_k are employed, where $T_{k-1} > T_k$ for $k = 1, 2, \cdots, L-1$. If any one of the inequalities $d_k > T_k$, $k = 0, 1, \cdots, L-1$, is true, then y_{ij} is regarded as a noise candidate and replaced by the median i.e., x_{ij}^0. Otherwise, y_{ij} is regarded as a signal candidate and will not be changed.

If 3×3 windows are used (i.e., $h = 1$ and $L = 4$), four thresholds T_k, $k = 0, 1, \cdots, 3$, are needed. The median of the absolute deviations from the median (MAD), which is defined as

$$\text{MAD} = \text{median}\{y_{i-u,j-v} - x_{ij}^0 : -h \leq u, v \leq h\} \tag{2.2}$$

is a robust estimate of dispersion [9, 24], and its scaled forms are used as the thresholds. Specifically, one sets

$$T_k = s \cdot \text{MAD} + \delta_k, \quad 0 \leq k \leq 3, \tag{2.3}$$

with

$$[\delta_0, \delta_1, \delta_2, \delta_3] = [40, 25, 10, 5]. \tag{2.4}$$

This choice yields satisfactory results in filtering random-valued impulse noise when the noise ratio is not high [12]. And here, parameter s varies for different images degraded with different noise ratios, and it is observed empirically that good results could be obtained using $0 \leq s \leq 0.6$ in suppressing impulse noise for various images.

This median-based impulse detector is shown to be robust for a wide variety of images, which therefore simplifies the selection of the thresholds to the adjustment of a single parameter [12].

2.2 Review of DPVM

Recently, a detail-preserving variational method (DPVM) has been proposed to restore impulse noise [14]. This variational method does not smear edges.

By DPVM, the noisy images are restored by minimizing a convex objective functional $F_{\mathbf{y}}$:

$$F_{\mathbf{y}}(\mathbf{x}) = \sum_{(i,j)\in\mathcal{A}} |x_{ij} - y_{ij}| + \frac{\beta}{2} \sum_{(i,j)\in\mathcal{A}} \sum_{(m,n)\in\mathcal{V}_{ij}} \varphi_\alpha(x_{ij} - x_{mn}), \tag{2.5}$$

where $\mathcal{A}$ is the set of all indices (i,j), and $\mathcal{V}_{ij}$ is the set of the four closest neighbors of (i,j), not including (i,j). β is a regularization parameter, and φ_α is an edge-preserving potential function. Possible choices for φ_α are:

$$\begin{aligned}
&\varphi_\alpha(t) = |t|^\alpha, \quad 1 < \alpha \le 2,\\
&\varphi_\alpha(t) = 1 + \frac{|t|}{\alpha} - \log(1 + \frac{|t|}{\alpha}), \quad \alpha > 0,\\
&\varphi_\alpha(t) = \log\left(\cosh(\frac{t}{\alpha})\right), \quad \alpha > 0,\\
&\varphi_\alpha(t) = \sqrt{\alpha + t^2}, \quad \alpha > 0,
\end{aligned}$$

see [16, 17, 19, 25, 26]. It was shown in [14] and [21] that under mild assumptions and a pertinent choice of β, the minimizer $\hat{\mathbf{x}}$ of $F_{\mathbf{y}}$ ensures $\hat{x}_{ij} = y_{ij}$ for most of the uncorrupted pixels y_{ij}. Furthermore, all pixels $\hat{x}_{ij}$ such that $\hat{x}_{ij} \neq y_{ij}$ are restored so that edges and local features are well preserved.

2.2.1 Minimization Scheme

The minimization algorithm is a Jacobi-type relaxation algorithm and works on the residual $\mathbf{z} = \mathbf{x} - \mathbf{y}$. It is stated as follows.

1. Initialize $z_{ij}^{(0)} = 0$ for each $(i, j) \in \mathcal{A}$.

2. At each iteration k, do the following for each $(i, j) \in \mathcal{A}$:

   a. Calculate

   $$\xi_{ij}^{(k)} = \beta \sum_{(m,n)\in\mathcal{V}_{ij}} \varphi'_\alpha(y_{ij} - z_{mn} - y_{mn}),$$

   where φ'_α is the derivative of φ_α, and z_{mn}, for $(m, n) \in \mathcal{V}_{ij}$, are the latest updates.

   b. If $|\xi_{ij}^{(k)}| \leq 1$, set $z_{ij}^{(k)} = 0$. Otherwise, find $z_{ij}^{(k)}$ by solving the nonlinear equation

   $$\beta \sum_{(m,n)\in\mathcal{V}_{ij}} \varphi'_\alpha(z_{ij}^{(k)} + y_{ij} - z_{mn} - y_{mn}) = \text{sgn}(\xi_{ij}^{(k)}). \qquad (2.6)$$

3. Stop the iteration when

   $$\max_{i,j}\{|z_{ij}^{(k+1)} - z_{ij}^{(k)}|\} \leq \tau_A$$

   and

   $$\frac{F_{\mathbf{y}}(\mathbf{y} + \mathbf{z}^{(k)}) - F_{\mathbf{y}}(\mathbf{y} + \mathbf{z}^{(k+1)})}{F_{\mathbf{y}}(\mathbf{y} + \mathbf{z}^{(k)})} \leq \tau_A,$$

   where τ_A is some given tolerance.

It was shown in [14] that the solution $z_{ij}^{(k)}$ of (2.6) satisfies

$$\text{sgn}(z_{ij}^{(k)}) = -\text{sgn}(\xi_{ij}^{(k)}),$$

and that $\mathbf{z}^{(k)}$ converges to $\hat{\mathbf{z}} = \hat{\mathbf{x}} - \mathbf{y}$ where $\hat{\mathbf{x}}$ is the minimizer for (2.5). The convergence of the minimization scheme has been shown in [14].

This minimization scheme is equivalent to solving the one-dimensional nonlinear equation (2.6) for all pixels (i, j). Typically, Newton's method is preferred to solve the (2.6). Since the convergence domain of Newton's method can be very narrow, care must be exercised in choosing the initial guess, see [22].

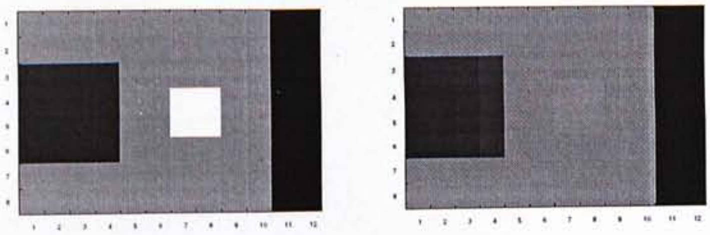


Figure 2.1: **(Left)** The noise image. **(Right)** Restored image by DPVM.

This detail-preserving method can preserve edges when denoising, but it has problem in suppressing noise patches, i.e., when many noise pixels are connecting with each other. Another drawback is that some signal pixels at the edges will be distorted, see Figure 2.1.

Chapter 3

Two-Phase Iterative Method

3.1 Introduction

When the noise ratio is high, ACWMF may falsely detect some noise-free pixels as noisy pixels. If these erroneous noise candidates form patches, and are located near to edges, DPVM will distort them. To alleviate these drawbacks of each one of them, we apply our method iteratively with different thresholds. More precisely, at the early iterations, we take large thresholds in ACWMF so that it will only select pixels that are most likely to be noisy. Then we restore them by DPVM. In the subsequent iterations, we decrease the thresholds to include more noise candidates. Since the edges and the details are preserved by the regularization successfully in each iteration, the restored image will not be distorted by the method.

3.2 Two-Phase Scheme

3.2.1 Detection Phase

Like other decision-based filters, the first phase of our method is to identify noise candidates. Since ACWMF may falsely detect some noise-free pixels as noisy pixels, we will modify the thresholds of ACWMF such that we can identify noise candidates more accurately.

In this phase, we use 3 × 3 window size, and do not fix the thresholds in each iteration. The form of the thresholds is the following:

$$T_k^{(r)} = s \cdot \mathrm{MAD}^{(r)} + \delta_k + 20(r_{\max} - r), \tag{3.1}$$

for $0 \le k \le 3$, $0 \le r \le r_{\max}$, and $0 \le s \le 0.6$, cf. (2.2)–(2.4).

Then we apply ACWMF with the thresholds $T_k^{(r)}$, $0 \le k \le 3$, to the image $\mathbf{y}^r$. If any one of the inequalities $d_k > T_k^{(r)}$, $0 \le k \le 3$, is true, then y_{ij} is regarded as a noise candidate. Otherwise, y_{ij} is regarded as a signal candidate. The difference d_k are computed by (2.1). Then we get the noise candidate set $\mathcal{M}^{(r)}$. Finally, let $\mathcal{N}^{(r)} = \bigcup_{l=0}^{r} \mathcal{M}^{(l)}$. This $\mathcal{N}^{(r)}$ is the final noise candidate set at rth iteration and will be restored at the second phase.

3.2.2 Restoration Phase

When we get the noise candidate set, we do not replace them by median filter. We restore them by DPVM, since the edges can be preserved by DPVM.

For using the DPVM, the minimization function $F_{\mathbf{y}}$ is not the same as the (2.5). Since we just restore the noise candidates $\mathcal{N}^{(r)}$, the signal candidates will not be changed. Then the new minimization functional $F_{\mathbf{y}}$ is:

$$\begin{aligned} F_{\mathbf{y}}(\mathbf{x}) = & \sum_{(i,j)\in\mathcal{N}^{(r)}} |x_{ij}-y_{ij}| \\ & +\frac{\beta}{2}\left(\sum_{(i,j)\in\mathcal{N}^{(r)}} \sum_{(m,n)\in\mathcal{V}_{ij}} \varphi_\alpha(x_{ij}-x_{mn}) \right. \\ & \left. + \sum_{(m,n)\in\mathcal{V}_{\mathcal{N}^{(r)}}} \sum_{(i,j)\in\mathcal{V}_{mn}\cap\mathcal{N}^{(r)}} \varphi_\alpha(y_{mn}-x_{ij}) \right) \end{aligned} \tag{3.2}$$

where $\mathcal{N}^{(r)}$ is the set of noise candidates at the rth iteration, and $\mathcal{V}_{\mathcal{N}^{(r)}} = \left(\bigcup_{(i,j)\in\mathcal{N}^{(r)}} \mathcal{V}_{ij}\right) \setminus \mathcal{N}^{(r)}$.

The minimizer $\hat{\mathbf{x}}$ of (3.2) is obtained by using minimization scheme but restricted onto the set of noise candidates $\mathcal{N}^{(r)}$. Hence, this phase is equivalent to solving the one-dimensional nonlinear equation (2.6) for each noise candidate. In the next chapter, we will give the Newton's method and secant-like method to solve this nonlinear equations.

3.2.3 Summary of the Algorithm

So far, we know the two-phase scheme clearly. To restore the noisy images, we apply these two-phase alternatively. In the following we give our algorithm together.

Algorithm:

1. Set $r = 0$. Initialize $\mathbf{y}^{(r)}$ to be the observed image.

2. Apply ACWMF with the thresholds $T_k^{(r)}$, $0 \le k \le 3$, which is formed by (3.1), to the image $\mathbf{y}^{(r)}$ to get the noise candidate set $\mathcal{M}^{(r)}$.

3. Let $\mathcal{N}^{(r)} = \bigcup_{l=0}^{r} \mathcal{M}^{(l)}$.

4. For all $(i,j) \notin \mathcal{N}^{(r)}$, take $\hat{x}_{ij} = y_{ij}^{(r)}$.

   Restore all pixels in $\mathcal{N}^{(r)}$ by minimizing the functional (3.2) over $\mathcal{N}^{(r)}$.

5. Set $\mathbf{y}^{(r+1)} = \hat{\mathbf{x}}$.

6. If $r < r_{\max}$, set $r = r + 1$ and go back to step 2.

In practice, four iterations are enough, i.e., $r_{\max} = 3$ and the output is $\mathbf{y}^{(4)}$.

This two-phase iterative method has successfully suppressed the noise while preserving most of the details and the edges in both cases, even when the noise ratio is high.

Chapter 4

Nonlinear Equation Solver

4.1 Introduction

According to the previous chapters, the second phase of our method is equivalent to solve the nonlinear equation (2.6) with some φ_α.

There exist many methods for solving this nonlinear equation. Newton's method and the secant method are very popular and can be guaranteed to converge. Usually we use $\varphi_\alpha(t) = |t|^\alpha$ as our edge-preserving function, because it will tend to $|t|$ when α tends to 1. But, if we use $\varphi_\alpha(t) = |t|^\alpha$ as our edge-preserving function, the complexity of the algorithm is not good.

To improve the timing, we describe another simple method which is called the secant-like method to solve (2.6). This method converges in fewer number of iterations than Newton's method with both methods requiring the same number of function and derivative evaluations per iteration.

4.2 Newton's Method

4.2.1 Newton's Method

Suppose we have a function f whose zeros are to be determined numerically. Let $\bar{x}$ be a zero of f and let x be an approximation to $\bar{x}$. Then by Taylor's theorem, if f'' exists and is continuous, we have

$$\begin{aligned} f(x+h) = & \ f(x) + hf'(x) + \mathcal{O}(h^2) \\ = & \ f(\bar{x}) \\ = & \ 0, \end{aligned}$$

where $h = \bar{x} - x$. When h is small enough, it is reasonable to ignore the $\mathcal{O}(h^2)$ term. Then we can get

$$\begin{aligned} & f(x) + hf'(x) = 0 \\ \Rightarrow \ & h = -\frac{f(x)}{f'(x)}. \end{aligned}$$

If x is an approximation to $\bar{x}$, then $x - \dfrac{f(x)}{f'(x)}$ should be a better approximation to $\bar{x}$.

We let x_0 be the initial guess. Newton's method begins with this initial guess x_0 and then defines inductively

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad (n \geq 0).$$

4.2.2 Order of Convergence

Now we shall analyze the convergence of the Newton's method. We assume that f'' is continuous and $\bar{x}$ is a simple zero of f, so that $f(\bar{x}) = 0$ and $f'(\bar{x}) \neq 0$. And the errors are defined as

$$e_n = x_n - \bar{x}.$$

From the definition of the Newton iteration, we have

$$\begin{aligned} e_{n+1} = & \; x_{n+1} - \bar{x} \\ = & \; x_n - \frac{f(x_n)}{f'(x_n)} - \bar{x} \\ = & \; e_n - \frac{f(x_n)}{f'(x_n)} \\ = & \; \frac{e_n f'(x_n) - f(x_n)}{f'(x_n)}. \end{aligned} \tag{4.1}$$

By Taylor's theorem, we have

$$\begin{aligned} f(x_n - e_n) = & \; f(x_n) - e_n f'(x_n) + \frac{1}{2} e_n^2 f''(\xi_n) \\ = & \; f(\bar{x}) \\ = & \; 0, \end{aligned}$$

where ξ_n is a number between x_n and $\bar{x}$. So we get

$$e_n f'(x_n) - f(x_n) = \frac{1}{2} f''(\xi_n) e_n^2.$$

Putting this in (4.1) leads to

$$\begin{aligned} e_{n+1} = & \; \frac{1}{2} \frac{f''(\xi_n)}{f'(x_n)} e_n^2 \\ \approx & \; \frac{1}{2} \frac{f''(\bar{x})}{f'(\bar{x})} e_n^2 \\ = & \; C e_n^2, \end{aligned}$$

where C is a constant.

This tells us that Newton's method is the second order convergence (also called quadratic convergence). From [22] we know how to find the initial guess such that Newton's method is guaranteed to converge. However, the complexity of our algorithm with $\varphi_\alpha = |t|^\alpha$ is not good.

4.3 Secant Method

4.3.1 Secant Method

One of the drawbacks of Newton's method is that it involves the derivative of the function whose zero is sought. To overcome this disadvantage, the secant method has been proposed.

If we use

$$\frac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}}$$

to approximate the $f'(x_n)$, then Newton's iteration is changed as following

$$x_{n+1}=x_n-f(x_n)\left(\frac{x_n-x_{n-1}}{f(x_n)-f(x_{n-1})}\right) \qquad (n\geq 1).$$

This algorithm is called the secant method.

Since the calculation of x_{n+1} requires x_n and x_{n-1}, two initial points must be prescribed at the beginning. However, each new x_{n+1} requires only one new evaluation of f.

4.3.2 Order of Convergence

Similar to Newton's method, we have $e_n=x_n-\bar{x}$. From the definition of the secant method,

$$\begin{aligned} e_{n+1}= {} & x_{n+1}-\bar{x} \\ = {} & \frac{f(x_n)x_{n-1}-f(x_{n-1})x_n}{f(x_n)-f(x_{n-1})}-\bar{x} \\ = {} & \frac{f(x_n)e_{n-1}-f(x_{n-1})e_n}{f(x_n)-f(x_{n-1})} \\ = {} & \frac{1}{f(x_n)-f(x_{n-1})}\left(\frac{f(x_n)}{e_n}-\frac{f(x_{n-1})}{e_{n-1}}\right)e_ne_{n-1}. \end{aligned} \tag{4.2}$$

By Taylor's theorem, we have

$$\begin{aligned} f(x_n) = & \; f(\bar{x}+e_n) \\ = & \; f(\bar{x}) + e_n f'(\bar{x}) + \frac{1}{2}e_n^2 f''(\bar{x}) + \mathcal{O}(e_n^3). \end{aligned}$$

Since $f(\bar{x}) = 0$, this gives us

$$\frac{f(x_n)}{e_n} = f'(\bar{x}) + \frac{1}{2}e_n f''(\bar{x}) + \mathcal{O}(e_n^2),$$

and

$$\frac{f(x_{n-1})}{e_{n-1}} = f'(\bar{x}) + \frac{1}{2}e_{n-1} f''(\bar{x}) + \mathcal{O}(e_{n-1}^2).$$

Hence, we get

$$\frac{f(x_n)}{e_n} - \frac{f(x_{n-1})}{e_{n-1}} = \frac{1}{2}(e_n - e_{n-1}) f''(\bar{x}) + \mathcal{O}(e_{n-1}^2).$$

Moreover, we have

$$e_n - e_{n-1} = x_n - x_{n-1},$$

and

$$\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \approx \frac{1}{f'(\bar{x})}.$$

Put these results in (4.2), we arrive at

$$\begin{aligned} e_{n+1} \approx & \; \frac{1}{2}\frac{f''(\bar{x})}{f'(\bar{x})} e_n e_{n-1} \\ = & \; C e_n e_{n-1}, \end{aligned}$$

where C is a constant. To discover the order of convergence of the secant method, we first suppose that the secant method is αth order convergence, then we have

$$|e_{n+1}| \sim M|e_n|^\alpha$$

where M is a positive constant. Hence,

$$|e_n| \sim M|e_{n-1}|^\alpha$$

and

$$|e_{n-1}| \sim M^{-\frac{1}{\alpha}}|e_n|^{\frac{1}{\alpha}}.$$

From these results, we get

$$M|e_n|^{\alpha} \sim |C||e_n|M^{-\frac{1}{\alpha}}|e_n|^{\frac{1}{\alpha}}.$$

This can be written as

$$M^{1+\frac{1}{\alpha}}|C|^{-1} \sim |e_n|^{1-\alpha+\frac{1}{\alpha}}.$$

Since the left side of this relation is a nonzero constant while $e_n \to 0$, we have

$$1 - \alpha + \frac{1}{\alpha} = 0.$$

Taking the positive root of this equation,

$$\alpha = \frac{1+\sqrt{5}}{2}.$$

So the order of convergence is $\frac{1+\sqrt{5}}{2}$. We call the secant method's rate of convergence is superlinear.

Although each iteration of the secant method requires only one new evaluation of f, the order of convergence of the secant method is not good as Newton's method.

4.4 Secant-like Method

4.4.1 Secant-like Method

As we mentioned before, even for Newton's method, the complexity of our algorithm with $\varphi_\alpha(t) = |t|^\alpha$ is not very good. Fox example, for 30% noise ratio, it will take 30 times more CPU time than ACWMF. To improve the timing and from the idea of secant method, we describe a simple method which is called the secant-like method. This method converges in fewer number of iterations than Newton's method with both methods requiring the same number of function and derivative evaluations per iteration.

For our problem (2.6), we choose $\varphi_\alpha(t) = |t|^\alpha$ with $\alpha > 1$. With some modifications, similar techniques can be applied to other edge-preserving φ_α too.

According to Step 2(b) of the Minimization Scheme, we only need to solve (2.6) if $|\xi_{ij}^{(k)}| > 1$. We first consider the case where $\xi_{ij}^{(k)} > 1$.

When solving (2.6), $z_{mn} + y_{mn} - y_{ij}$, for $(m, n) \in \mathcal{V}_{ij}$, are known values. Let these values be denoted by d_j, for $1 \le j \le 4$, and be arranged in an increasing order:

$$d_j \le d_{j+1}.$$

Then (2.6) can be rewritten as

$$H(z) \equiv -1 + \alpha\beta \sum_{j=1}^{4} \operatorname{sgn}(z - d_j)|z - d_j|^{\alpha-1} = 0. \tag{4.3}$$

Hence (2.6) has a unique solution $z^* > d_1$, see [22]. By evaluating $\{H(d_j)\}_{j=2}^{4}$, we can check that if any one of the d_j, $2 \le j \le 4$, is the root z^*. If not, then z^* lies in one of the following intervals:

$$(d_1, d_2), (d_2, d_3), (d_3, d_4), \text{ or } (d_4, \infty).$$

We first consider the case where z^* is in one of the finite intervals (d_j, d_{j+1}). For simplicity, we give the details only for the case where $z^* \in (d_2, d_3)$. The other cases can be analyzed similarly. Since H is a monotone function in (d_2, d_3), see [22], it has an inverse G with $G(H) = z$. The goal is to compute $G(0)$. Suppose the following data are known:

$$\begin{aligned} G(H_1) &= z_1, \\ G(H_2) &= z_2, \\ G'(H_1) &= p_1, \\ G'(H_2) &= p_2. \end{aligned} \tag{4.4}$$

Note that G is the inverse of H

$$G(H(z)) = z,$$

so we have

$$\begin{aligned} G'(H(z)) = & \ G'(H)\tfrac{dH(z)}{dz} \\ = & \ 1. \end{aligned}$$

In particular,

$$G'(H_i) = \frac{1}{\frac{dH(z_i)}{dz}}, \qquad i = 1, 2.$$

We approximate G by a cubic polynomial P which satisfies the conditions (4.4). Let

$$\begin{aligned} z = & \ P(H) \\ = & \ z_1 + a(H - H_1) + b(H - H_1)(H - H_2) + cH(H - H_1)(H - H_2) \end{aligned}$$

for some constants a, b, c. From the conditions

$$z_i = P(H_i), \ i = 1, 2$$

and

$$p_i = P'(H_i), \ i = 1, 2,$$

we obtain

$$a = \frac{z_2 - z_1}{H_2 - H_1}$$

and

$$b = \frac{H_2(a - p_1) + H_1(a - p_2)}{(H_1 - H_2)^2}.$$

Thus the new approximate zero is given by

$$P(0) = z_1 - aH_1 + bH_1H_2.$$

In summary, the iteration is given by

$$z_{n+1} = z_n - a_nH(z_n) + b_nH(z_n)H(z_{n-1}), \qquad n \geq 1$$

where

$$a_n = \frac{z_n - z_{n-1}}{H(z_n) - H(z_{n-1})}$$

and

$$b_n = \frac{H(z_n)(a_n - \frac{1}{H'(z_{n-1})}) + H(z_{n-1})(a_n - \frac{1}{H'(z_n)})}{(H(z_n) - H(z_{n-1}))^2}.$$

Given z_0, the iterate z_1 is taken as the Newton iterate. Then apply the secant-like method to obtain the solution up to a given tolerance τ_B, that is,

$$|z_n - z_{n-1}| \leq \tau_B.$$

Finally, we turn to the case where $\xi^{(k)} < -1$. The nonlinear equation (4.3) becomes:

$$1 + \alpha\beta \sum_{j=1}^{4} \operatorname{sgn}(z - d_j)|z - d_j|^{\alpha-1} = 0,$$

we can use almost the same method to solve this equation.

4.4.2 Order of Convergence

The secant-like method is equivalent to scheme 3 on p. 233 of [27], where it is stated that the order of convergence is $1 + \sqrt{3}$. Now we stated the details about how to get this number.

Since we use a sequence cubic polynomial $P_n(H)$, $n \geq 2$, to approximate $G(H)$ and

$$P_n(H_i) = G(H_i), \qquad P_n'(H_i) = G'(H_i), \qquad i = n-1, n.$$

The difference between $G(H)$ and $P_n(H)$ is

$$\begin{aligned} R_n(H) = & \; G(H) - P_n(H) \\ = & \; \frac{G^{(4)}(\xi_n)}{4!}(H - H_{n-1})^2(H - H_n)^2, \end{aligned}$$

where ξ_n lies in the interval determined by H_{n-1}, H_n, and H. We know that $z_n = P_{n-1}(0)$ and $z^* = G(0)$, so the errors

$$\begin{aligned} e_{n+1} = & \; z_{n+1} - z^* \\ = & \; P_n(0) - G(0) \\ = & \; -\tfrac{G^{(4)}(\xi_n)}{4!} H_{n-1}^2 H_n^2 \\ = & \; C H_{n-1}^2 H_n^2, \end{aligned} \tag{4.5}$$

where C is a constant.

Similar to the derivative in the previous section, we have

$$H'(\eta) = \frac{H(z_i) - H(z^*)}{z_i - z^*}$$

where η is a number between z_i and z^*. Because $H(z^*) = 0$, we get

$$\begin{aligned} H_i = & \; H(z_i) \\ = & \; H'(\eta)(z_i - z^*) \\ = & \; H'(\eta)e_i. \end{aligned}$$

We put this result in (4.5), then we get

$$|e_{n+1}| \sim |C_1||e_n|^2|e_{n-1}|^2$$

where C_1 is a constant number. Assume that

$$|e_{n+1}| \sim M|e_n|^{\alpha},$$

where M is a positive constant number. Then, for e_{n-1} we have

$$|e_{n-1}| \sim M^{-\frac{1}{\alpha}}|e_n|^{\frac{1}{\alpha}}.$$

Hence we get the relation

$$M|e_n|^{\alpha} \sim |C_1||e_n|^2|M^{-\frac{2}{\alpha}}|e_n|^{\frac{2}{\alpha}}.$$

This can be written as

$$M^{1+\frac{2}{\alpha}}|C_1|^{-1} \sim |e_n|^{2-\alpha+\frac{2}{\alpha}}.$$

Since the left side of this relation is a nonzero constant, we need

$$2 - \alpha + \frac{2}{\alpha} = 0.$$

Taking the positive root of this equation, we get

$$\alpha = 1 + \sqrt{3},$$

which means that the order of convergence of the secant-like method is $1 + \sqrt{3}$.

Hence the method converges faster than Newton's method. In the next chapter, we see the secant-like method always takes fewer number of iterations than Newton's method. Experimentally, the secant-like method is as robust as Newton's method although we do not have any theoretical result in this direction.

Chapter 5

Numerical Experiments

5.1 Removing Noise

In this section, we compare our method with ACWMF [12] and DPVM [14]. The 256-by-256 picture of Lena and Bridge are used as the true image. Then 30% and 50% of the pixels are corrupted by random noise uniformly distributed on its dynamic range $[n_{\min}, n_{\max}]$, see Figures 5.1(a), 5.2(a), 5.3(a), and 5.4(a). Henceforth, we use the potential function $\varphi(t) = |t|^{1.3}$. To measure our results, we use peak signal-to-noise ratio (PSNR) and mean absolute error (MAE) in our simulations. The definitions of PSNR and MAE are given in the following:

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{n} |x_{ij} - y_{ij}|, \\ \mathrm{PSNR} &= 10 \log_{10} \left(\frac{255^2}{\frac{1}{MN} \sum_{ij} (x_{ij} - y_{ij})^2} \right) \end{aligned} \tag{5.1}$$

In the simulations, for each noise level, the parameters s in (2.3) and β in (3.2) are chosen to give the best restoration in terms of PSNR.

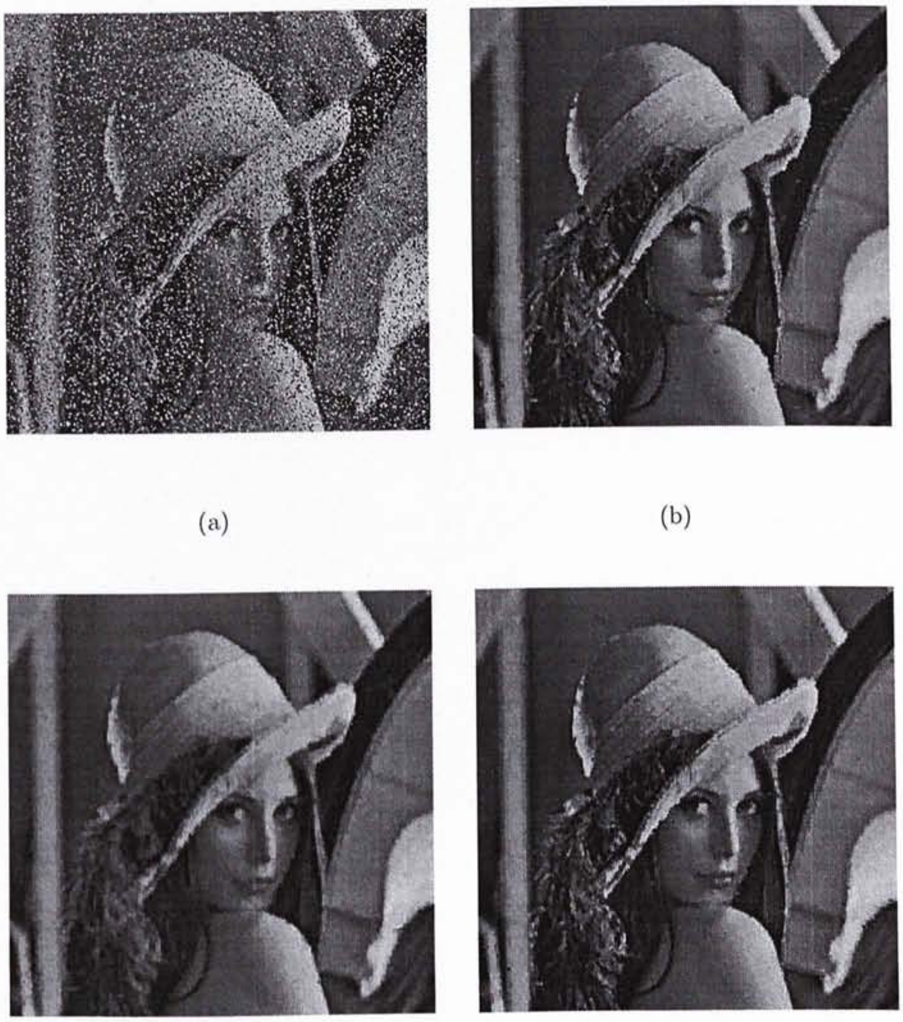
From Figures 5.1–5.4, we see that there are noticeable noise patches in the images restored by either ACWMF or DPVM, especially when the noise ratio

Table 5.1: Errors of Restored Images at 30% Noise

		bird	*bridge*	*camera*	*goldhill*	*lena*
PSNR	Noise Image	15.85	13.98	13.79	15.23	14.48
	ACWMF	32.06	22.21	24.35	26.57	27.18
	DPVM	33.26	22.44	24.72	27.13	27.29
	Our method	33.72	22.76	25.08	27.52	28.33
MAE	Noise Image	18.48	23.10	23.45	19.95	21.63
	ACWMF	1.61	8.43	4.17	4.36	3.32
	DPVM	2.25	11.90	6.06	6.18	4.97
	Our method	1.27	7.95	3.67	3.85	2.80

is 50%. In contrast, our method has successfully suppressed the noise while preserving most of the details and the edges in both cases.

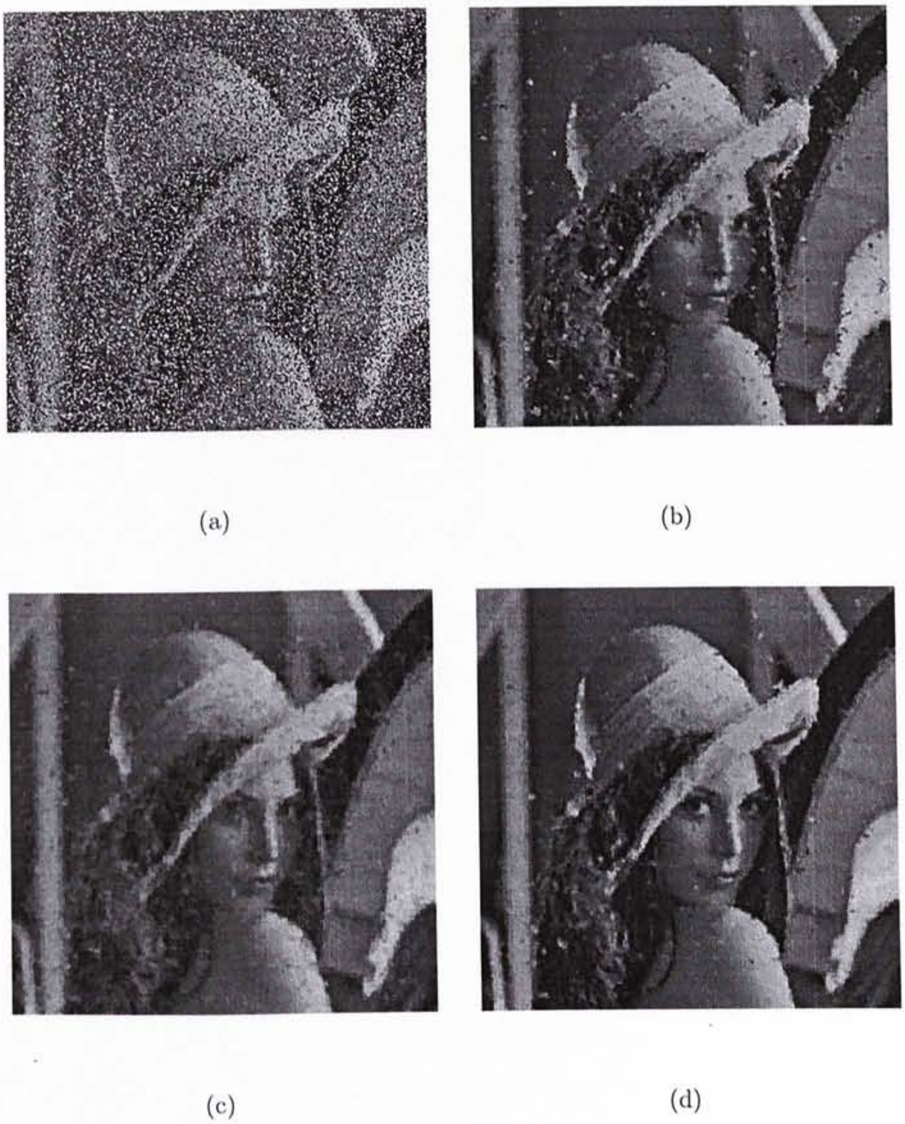
To assess the effectiveness of our method in processing various images, we tried three other 256-by-256 gray scale images. The parameters s and β were chosen to be the same as in the previous simulations. The results in terms of PSNR and MAE, are summarized in Tables 5.1–5.2. From the tables, we see that our method are significantly better than the other two methods. Pictures of the noisy images and the restored images can be found at `www.math.cuhk.edu.hk/~rchan/paper/chn/`. Overall, our restored images are significantly better than those restored by the other two methods.



(a) (b)

(c) (d)

Figure 5.1: (a) Image with 30% noise. Restored images by (b) ACWMF with $s = 0.6$, (c) DPVM with $\beta = 0.19$, and (d) our method with $\beta = 2, s = 0.6$ and 4 iterations.



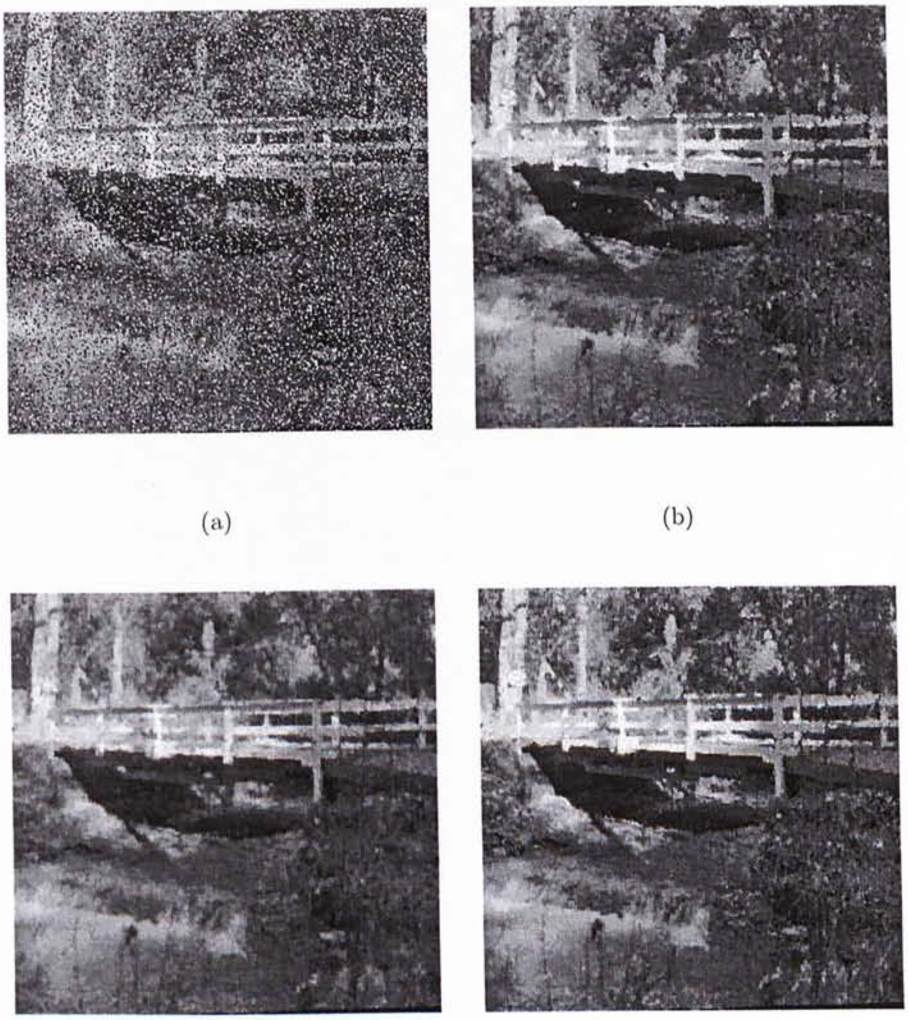
(a)

(b)

(c)

(d)

Figure 5.2: (a) Image with 50% noise. Restored images by (b) ACWMF with $s = 0.3$, (c) DPVM with $\beta = 0.19$, and (d) our method with $\beta = 2.3, s = 0.1$ and 4 iterations.



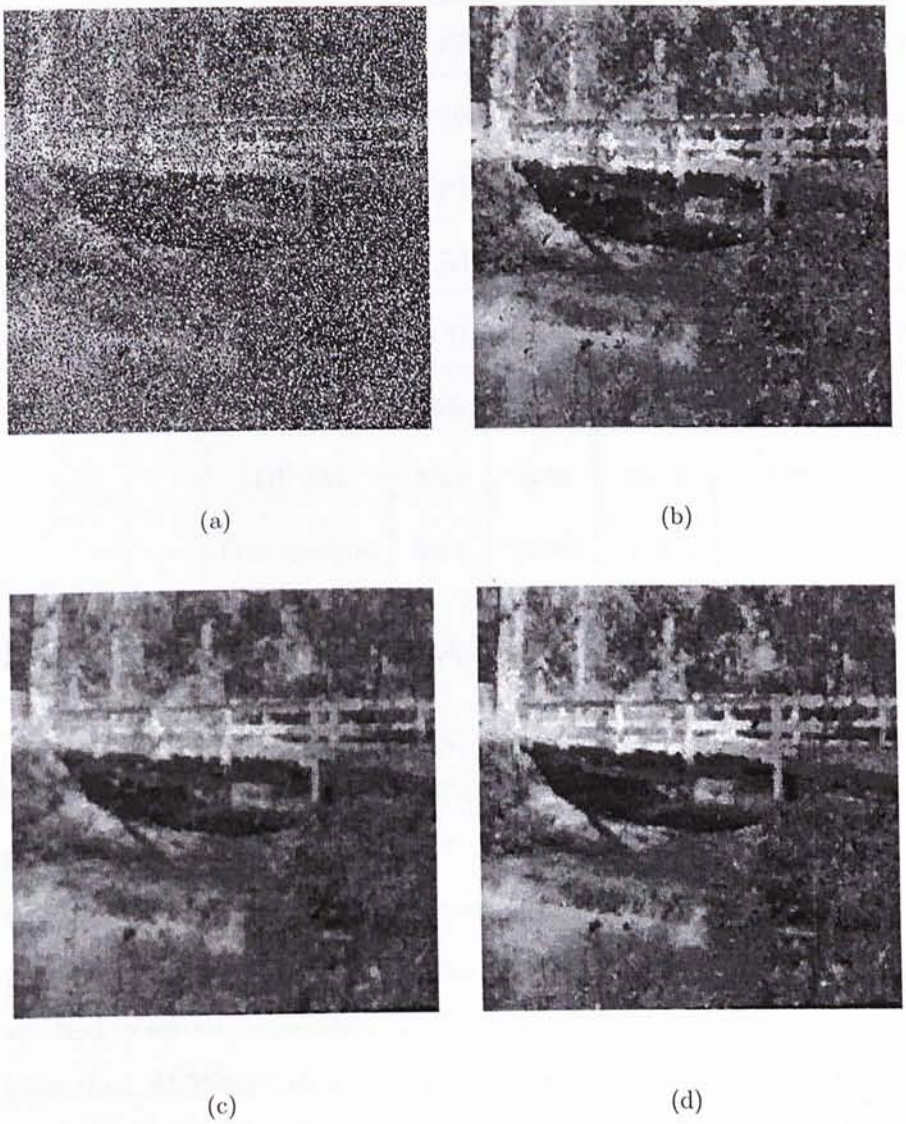
(a)

(b)

(c)

(d)

Figure 5.3: (a) Image with 30% noise. Restored images by (b) ACWMF with $s = 0.6$, (c) DPVM with $\beta = 0.19$, and (d) our method with $\beta = 2, s = 0.6$ and 4 iterations.


(a)

(b)

(c)

(d)

Figure 5.4: (a) Image with 50% noise. Restored images by (b) ACWMF with $s = 0.3$, (c) DPVM with $\beta = 0.19$, and (d) our method with $\beta = 2.3, s = 0.1$ and 4 iterations.

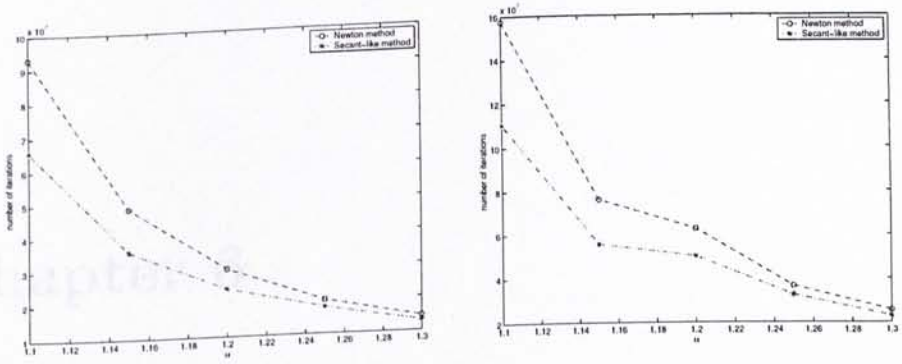
Table 5.2: Errors of Restored Images at 50% Noise

		bird	*bridge*	*camera*	*goldhill*	*lena*
PSNR	Noise Image	13.62	11.82	11.59	12.99	12.28
	ACWMF	25.36	19.53	20.43	22.74	22.40
	DPVM	26.81	20.21	21.17	23.63	23.08
	Our method	29.93	20.77	22.53	25.04	25.48
MAE	Noise Image	30.91	38.18	39.00	33.22	36.04
	ACWMF	4.96	14.87	9.88	9.05	8.26
	DPVM	6.22	16.92	12.12	10.88	10.20
	Our method	2.84	12.84	6.86	6.85	5.41

5.2 Complexity of Algorithm

Now, we consider the complexity of our algorithm. Since $r_{\max} = 3$, the algorithm requires four applications of ACWMF and four applications of DPVM restricted to the set of the noisy pixels $\mathcal{N}^{(r)}$. Like other medium-type filters, ACWMF can be done very fast. The application of DPVM is the most time-consuming part as it requires the minimization of the functional in (3.2). If we use Newton's method to solve this problem, for 30% noise, our method takes 30 times more CPU time than ACWMF. Hence, we describe a secant-like method. We compare the number of iterations of secant-like method with Newton's method with different magnitudes of α.

In the simulations, for each noise pixel, we use ACWMF in the first phase, and detail-preserving regularization in the second phase. Also, we use the secant-like method to solve (2.6) with the potential function $\varphi(t) = |t|^{1.2}$. We choose $\beta = 2$



(a)

(b)

Figure 5.5: Compare the number of iterations of secant-like method with Newton's meth od (a) restore 30% noisy image (b) restore 50% noisy image.

for all settings. The tolerances is chosen to be

$$\tau_A = (n_{\max} - n_{\min}) \times 10^{-4}$$

and

$$\tau_B = 5 \times 10^{-4}.$$

In Figure 5.5, we give, for different values of α, the total num ber of iterations. From the figures, we see that the secant-like method converges faster than Newton's method. The gain is greater as α approaches 1.

Chapter 6

Concluding Remarks

In this thesis, we propose a two-phase iterative method for removing random-valued impulse noise. To improving the CPU time, we describe a simple secant-like method to solve the nonlinear equations.

Simulation results indicate that the proposed method is significantly better than those using just nonlinear filters or regularization only. Our method can restore large patches of noisy pixels because it introduces pertinent prior information via the regularization term. It is most efficient to deal with high noise ratio, e.g. ratio as high as 50%.

In the second phase, we need to solve some nonlinear equations. We illustrate that the secant-like method converges faster than Newton's method, yet requiring the same number of function and derivative evaluations per iteration. The gain is greater as α approaches 1.

Bibliography

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed., Prentice Hall, 2001.

[2] H. Hwang and R. A. Haddad, *Adaptive median filters: new algorithms and results*, IEEE Transactions on Image Processing, vol. 4, 1995, pp. 499–502.

[3] T. Chen and H. R. Wu, *Space variant median filters for the restoration of impulse noise corrupted images*, IEEE Transactions on Circuits and Systems II, vol. 48, 2001, pp. 784–789.

[4] Y. H. Lee and S. A. Kassam, *Generalized median filtering and related nonlinear filtering techniques*, IEEE Transactons on Acoustics, Speech, and Signal Processing, vol. 33, 1985, pp. 965–994.

[5] I. Pitas and A. Venetsanopoulos, *Nonlinear mean filters in image processing*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 34, 1986, pp. 600–609.

[6] G. R. Arce and R. E. Foster, *Detail-preserving ranked-order based filters for image processing*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, 1989, pp. 83–98.

[7] R. C. Hardie and K. E. Barner, *Rank conditioned rank selection filters for signal restoration*, IEEE Transactions on Image Processing, vol. 3, 1994, pp. 192–206.

[8] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, *A new efficient approach for the removal of impulse noise from highly corrupted images*, IEEE Transactions on Image Processing, vol. 5, 1996, pp. 1012–1025.

[9] J. Astola and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering*, Boca Raton, CRC, 1997.

[10] W.-Y. Han and J.-C. Lin, *Minimum-maximum exclusive mean (MMEM) filter to remove impulse noise from highly corrupted images*, Electronics Letters, vol. 33, 1997, pp. 124–125.

[11] Z. Wang and D. Zhang, *Progressive switching median filter for the removal of impulse noise from highly corrupted images*, IEEE Transactions on Circuits and Systems II, vol. 46, 1999, pp. 78–80.

[12] T. Chen and H. R. Wu, *Adaptive impulse detection using center-weighted median filters*, IEEE Signal Processing Letters, vol. 8, 2001, pp. 1–3.

[13] G. Pok, J.-C. Liu, and A. S. Nair, *Selective removal of impulse noise based on homogeneity level information*, IEEE Transactions on Image Processing, vol. 12, 2003, pp. 85–92.

[14] M. Nikolova, *A variational approach to remove outliers and impulse noise*, Journal of Mathematical Imaging and Vision, vol. 20, 2004, pp. 99–120.

[15] T. A. Nodes and N. C. Gallagher, Jr., *The output distribution of median type filters*, IEEE Transactions on Communications, vol. COM-32, 1984, pp. 532–541.

[16] C. Bouman and K. Sauer, *A generalized Gaussian image model for edge-preserving MAP estimation*, IEEE Transactions on Image Processing, vol. 2, 1993, pp. 296–310.

[17] C. Bouman and K. Sauer, *On discontinuity-adaptive smoothness priors in computer vision*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, 1995, pp. 576–586.

[18] T. F. Chan, H. M. Zhou, and R. H. Chan, *A continuation method for total variation denoising problems*, Proceedings of SPIE Symposium on Advanced Signal Processing: Algorithms, Architectures, and Implementations, ed. F. T. Luk, vol. 2563, 1995, pp. 314–325.

[19] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, *Deterministic edge-preserving regularization in computed imaging*, IEEE Transactions on Image Processing, vol. 6, 1997, pp. 298–311.

[20] C. R. Vogel and M. E. Oman, *Fast, robust total variation-based reconstruction of noisy, blurred images*, IEEE Transactions on Image Processing, vol. 7 1998, pp. 813–824.

[21] M. Nikolova, *Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers*, SIAM Journal on Numerical Analysis, vol. 40, 2002, pp. 965–994.

[22] R. H. Chan, C.-W. Ho, and M. Nikolova, *Convergence of Newton's method for a minimization problem in impulse noise removal*, Journal of Computational Mathematics, vol. 22, 2004, pp. 168–177.

[23] T. Chen, K.-K. Ma, and L.-H. Chen, *Tri-state median filter for image denoising*, IEEE Transaction on Image Processing, vol. 8, 1999, pp. 1834–1838.

[24] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*, New York: Wiley, 1986.

[25] P. J. Greeen, *Bayesian reconstructions from emission tomography data using a modified EM algorithm*, IEEE Transactions on Medical Imaging, vol. MI-9, 1990, pp. 84–93.

[26] M. Black and A. Rangarajan, *On the unification of line processes, outlier rejection, and robust statistics with applications to early vision*, International Journal of Computer Vision, vol. 19, 1996, pp. 57–91.

[27] J. F. Traub, *Iterative methods for the solution of equations*, Englewood Cliffs, N. J.: Prentice Hall, 1964.