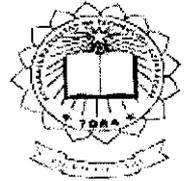


P-3468



A COMPARATIVE STUDY OF IMPULSE NOISE REMOVAL ALGORITHMS

By

S.BHARATHI

Reg. No. 1020106003

of

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution affiliated to Anna University of Technology, Coimbatore)

COIMBATORE - 641049

A MINI PROJECT REPORT

Submitted to the

**FACULTY OF ELECTRONICS AND COMMUNICATION
ENGINEERING**

In partial fulfillment of the requirements

for the award of the degree

of

MASTER OF ENGINEERING

IN

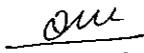
APPLIED ELECTRONICS

MAY 2011

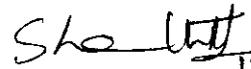


BONAFIDE CERTIFICATE

Certified that this project report titled “A COMPARATIVE STUDY OF IMPULSE NOISE REMOVAL ALGORITHMS” is the bonafide work of **Bharathi S [Reg. No. 1020106003]** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.


17.05.2011
Project Guide

Dr.A.Vasuki


for Head of the Department
17/5/11
Dr. Rajeswari Mariappan

The candidate with university **Register No. 1020106003** is examined by us in the project viva-voce examination held on ...23.05.2011...


Internal Examiner
23/5/11


External Examiner
23/5

ACKNOWLEDGEMENT

At this pleasing moment of having successfully completed the project work, I wish to acknowledge my sincere gratitude and heartfelt thanks to **Kumaraguru College of Technology** for having given me the adequate support and opportunity for completing this project work successfully.

I express my sincere thanks to **Dr.Ms.Rajeswari Mariappan Ph.D.**, the ever active, Head of the Department of Electronics and Communication Engineering, who rendering us all the time by helps throughout this project

In particular, I wish to thank and everlasting gratitude to the project coordinator **Mrs.R.Hemalatha M.E., Asst. Professor**, Department of Electronics and Communication Engineering for her expert counseling and guidance to make this project to a great deal of success. With her careful supervision and ensured me in the attaining perfection of work.

I extend my heartfelt thanks to my internal guide **Dr.A.Vasuki Ph.D., Asst. Professor** for her ideas and suggestion, which have been very helpful for the completion of this project work.

Last, but not the least, I would like to express my gratitude to my family members, friends and to all my staff members of Electronics and Communication Engineering department for their encouragement and support throughout the course of this project.

ABSTRACT

Digital images are frequently corrupted by impulse noise generated in camera sensors, analog-to-digital convertors and communication channels. The intensity of impulse noise has the tendency of being either relatively high or relatively low. Thus, it could severely degrade the image quality and cause great loss of information details. Therefore, it is imperative to remove impulse noise in order to facilitate the subsequent processing such as edge detection, segmentation, analysis, pattern recognition, etc. Many linear and nonlinear filtering techniques have been proposed to remove impulse noise. Filtering an image to attenuate noise while keeping the image details preserved is one of the most important issues. The techniques compared here are Rank Ordered Mean Filter (ROM) and Adaptive Median Filter (AMF).

Two types of impulse noise can be modeled: i) Fixed valued impulse noise (salt & pepper) and ii) Random valued impulse noise. Here we are considering the images corrupted with the fixed valued impulse noise called the 'salt & pepper noise'. The performance of the above mentioned techniques is tested using various impulse noise corrupted images. The simulation results of these techniques are obtained using MATLAB and are compared with respect to their capability to remove the high density impulse noise effectively while preserving the fine details in the image.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	iv
	LIST OF FIGURES	vi
1	INTRODUCTION	1
2	IMPULSE NOISE MODEL	
	2.1 Impulse Noise	3
	2.2 Types of impulse noise	3
	2.3 Algorithm for Rank Ordered Mean Filter	4
	2.4 Noisy and Noise free pixels	6
	2.5 Impulse Noise Detection	6
	2.6 Edge and smooth region detection	7
	2.7 Rank Ordered Mean filter	8
	2.8 Algorithm for Adaptive Median Filter	9
	2.9 Detection of corrupted pixels	11
	2.10 Implementation of the algorithm	11
3	EXPERIMENTAL ANALYSIS	16
4	CONCLUSION	22
	REFERENCES	23

LIST OF FIGURES

FIGURE NO	FIGURE CAPTION	PAGE NO
2.1	Rank Ordered Mean Filter Algorithm	5
2.2	Adaptive Median Filter Algorithm	10
2.3	Window of Uncorrupted pixels	11
2.4	Median in 3x3 window	12
2.5	Median in 5x5 window	13
2.6	Median in 5x5 window with exceeded count	14
2.7	Median for 9 corrupted pixels	15
3.1	Display of input, noisy and filtered images	17
3.5	PSNR performances of algorithms for images with salt and pepper noise	21

LIST OF TABLES

TABLE NO	TABLE CAPTION	PAGE NO
3.1	PSNR performances of algorithms for images with salt and pepper noise	21

CHAPTER 1

INTRODUCTION

Images are produced in order to record or display useful information. Due to imperfections in the electronic or photographic medium, the recorded image often represents a degraded version of the original scene. The degradations may have many causes, but the dominant one is the presence of noise in the image. The restoration and enhancement of the noised images is of fundamental importance in image processing applications. To find the original image the degraded images has to be denoised.

The goal of this project is to remove the impulse noise from the corrupted image. Here we are comparing the impulse noise removal algorithms in denoising the corrupted images and restoring the edge and fine details of the image without loss of any information. These algorithms are efficient in the removal of impulse noise and restoring the fine edge details. The filters adopted to denoise the corrupted image are: 1. Adaptive Median Filter 2. Rank Ordered Mean Filter.

Digital images are frequently corrupted by impulse noise due to errors generated in camera sensors, analog-to-digital conversion and communication channels. The intensity of impulse noise has the tendency of being either relatively high or low. The subsequent image processing including edge detection, image analysis, and object tracking may perform worse if the input image has a larger noise ratio. Filtering a digital image to attenuate noise while keeping the image details preserved is an essential part of image processing. Therefore, how to detect noise pixels and then to replace them with appropriate values is one of the most important issues. Various filtering techniques like Adaptive Median Filter and Rank Ordered Mean Filter have been employed here for removing impulse noise.

The software used here for the implementation of this project to discuss over the results is **MATLAB R2009a**.

- **Chapter 2** discusses about the impulse noise model and the fixed-value impulse noise. It also deals about implementing the Rank Ordered Mean filter algorithm and Adaptive Median Filter algorithm in a 3x3 window of the corrupted image.
- **Chapter 3** gives the experimental analysis of the performance of the two algorithms and exhibits the results obtained.
- **Chapter 4** shows the conclusion of the project.

CHAPTER 2

IMPULSE NOISE MODEL

2.1 IMPULSE NOISE

Noise consisting of random occurrences of energy spikes having random amplitude and spectral content is known as Impulsive noise. Impulse noise should be defined as single or multiple noise events lasting one second or less. A short burst of an acoustic energy consists of either a single impulse or a series of impulses. When the intervals between impulses are less than 500 milliseconds, the noise is considered continuous, which is considered as impulse noise. Impulsive noise appears as black or white spots in an image and hence it is also called as salt and pepper noise. The corruption by impulse noise is a frequently encountered problem in image acquisition and transmission through communication channels. During this process, some image pixels are destroyed and they take high positive (white spots) or low values (black spots).

2.2 TYPES OF IMPULSE NOISE

There are different types of impulse noise namely

1. **Salt and pepper** type of noise
2. **Random valued** impulse noise.

In **salt and pepper** type of noise the noisy pixels takes either salt value (gray level -225) or pepper value (grey level -0) and it appears as black and white spots on the images. For gray level images with 8 bits per pixel, when the images are contaminated by fixed-value impulse noise, a noise pixel takes either a large value 255 or a small value 0. The corrupted pixel n_{ij} is equal to 0 or 255 each with equal probability 0.5p.

In case of **random valued** impulse noise, noise can take any gray level value from zero to 225. In this case also noise is randomly distributed over the entire image and probability of occurrence of any gray level value as noise will be same. In this project, we focus only on the detection and denoising of fixed-value impulse noise, namely "salt and pepper noise".

2.3 ALGORITHM FOR RANK ORDERED MEAN FILTER

Consider the corrupted image. Following are the steps involved in removing the noise in the corrupted image.

Step 1: The impulse noise model is applied to classify the noisy pixels and noise free pixels.

Step 2: To preserve the edges and fine details in the image, the edges and the smooth regions should be detected from the extrema found by min-max algorithm.

Step 3: These details from the extremum are considered as noise free pixels, and are left unchanged.

Step 4: Only noisy pixel is then considered and these pixels are replaced by ROM filter output.

The proposed algorithm takes these issues into account and the filtering processing summary is shown in Figure 1

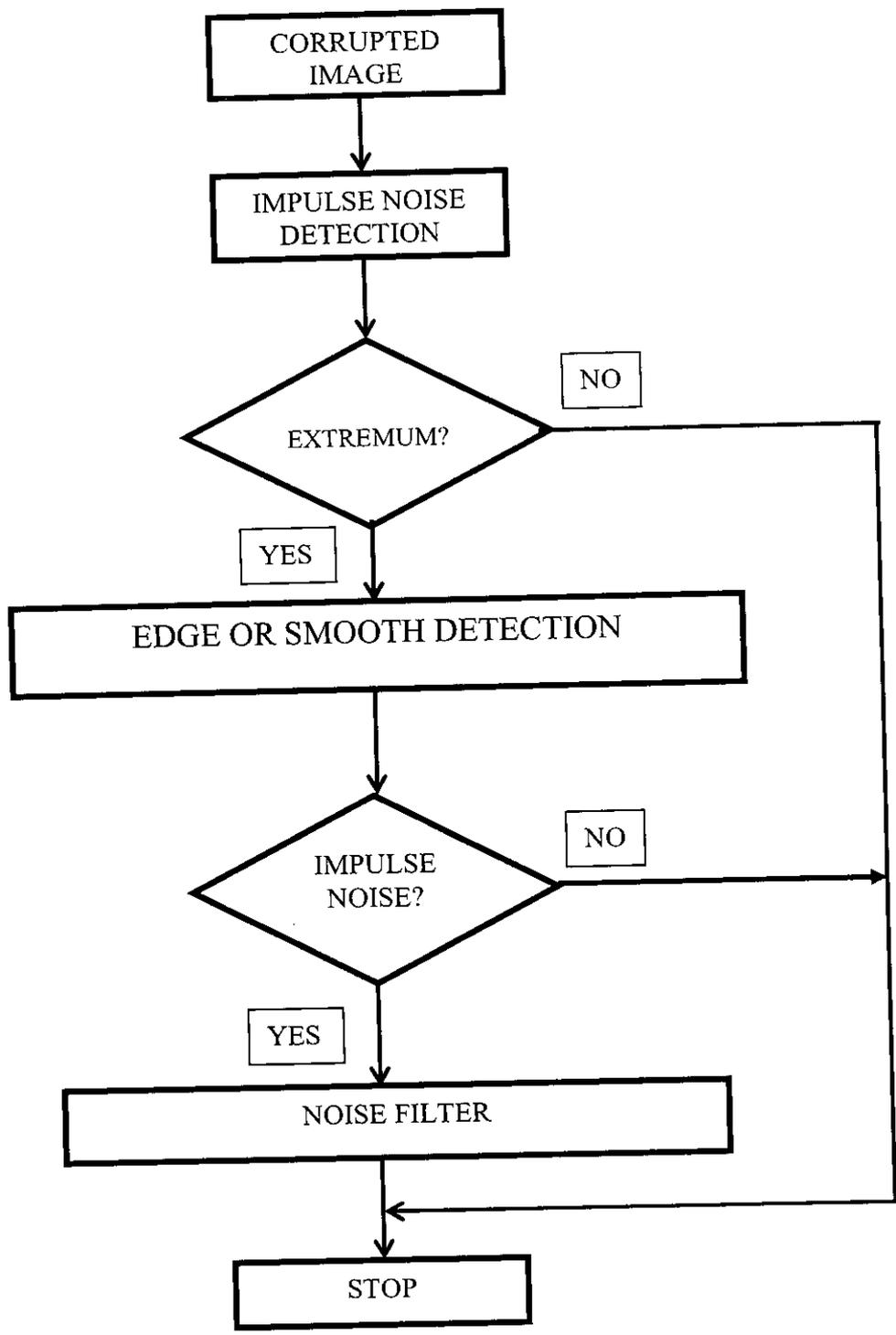


Figure 2.1 RANK ORDERED MEAN FILTER ALGORITHM

2.4 NOISY AND NOISE FREE PIXELS

Consider a corrupted image X , which containing impulse noise with probability p . If p is the total noise density then salt noise and pepper noise will have a noise density of $p/2$. This can be mathematically represented by (1)

$$x_{ij} = \begin{cases} n_{i,j}, & \text{with probability } p \\ s_{i,j}, & \text{with probability } 1 - p \end{cases} \quad (2.1)$$

Where $i=1, 2 \dots s_1$ and $j=1, 2 \dots s_2$ and $0 < p < 1$.

- x_{ij} \longrightarrow the intensity of the pixel located at position (i, j) in the image,
- s_{ij} \longrightarrow the intensity of the pixel (i, j) in the original image
- n_{ij} \longrightarrow the intensity of the pixel (i, j) in the noisy image and n_{ij} is a noise value independent from x_{ij} .

2.5 IMPULSE NOISE DETECTION

In this scheme, a sliding window centered around x_{ij} is employed to detect the extrema for the corrupted pixels. Let we choose M for the sliding window size, Now there are $(M^2 - 1)$ pixels (excluding x_{ij} itself) in the window.

Let $X = \{x_{i-L, j-L}, \dots, x_{i+L, j+L}\}$ represent the input sample in the $M \times M$ sliding window where x_{ij} is excluded and $M = 2L + 1$ with $L = 2$ in this paper. Now, we re-arrange the input samples X in the ascending order: $X_r = \{x^1, x^2, \dots, x^{M^2-1}\}$, where $x^1 \leq x^2 \leq \dots \leq x^{M^2-1}$. Usually, the orders of the noise pixels in the sliding window are either very big or very small, i.e. this kind of impulse noise usually lies either at the beginning or at the end of the sorted sample data.

Then we compare x_{ij} with each element of X_r . If x_{ij} is the extreme pixel alone, then it is considered as noisy pixel, which will be removed. And if x_{ij} is the extreme pixel but not alone, then it is regarded as noise candidate, which will be further processed. Otherwise, it is a noise-free pixel, which will be preserved.

The extreme test can be expressed in the form:

$$x_{ij} \in \begin{cases} N & \text{if } \dots x_{ij} > \max[X_r] \dots \text{or } \dots x_{ij} < \min[X_r]; \\ N' & \text{if } \dots x_{ij} = \max[X_r] \dots \text{or } \dots x_{ij} = \min[X_r]; \\ S & \text{else} \end{cases} \quad (2.2)$$

Where,

- N \longrightarrow the corrupted pixel,
 N' \longrightarrow the noisy candidate,
 S \longrightarrow the noise-free pixel.

2.6 EDGE AND SMOOTH REGION DETECTION

When the noise candidates (N) detected by (2), it means more extreme pixels within the window and some detailed pixels may be regarded as noise pixels. Consequently, we must discriminate the detailed pixels from the noise candidates. In order to identify the edges and fine detailed from the noise candidates, the sub-window is introduced in the algorithm.

In the sliding window can be divided into four one-dimensional sub-windows, i.e., horizontal, vertical, main diagonal and auxiliary diagonal direction. In each sub-window, the sum of absolute value of difference between x_{ij} and the other pixel is denoted as m_1 , m_2 , m_3 , m_4 , respectively.

For example, we choose the 3 x 3 sliding window, the m_1 , m_2 , m_3 , m_4 can be expressed as following

$$\begin{cases} m_1 = |2x_{ij} - x_{i,j-1} - x_{i,j+1}|; & \text{horizontal;} \\ m_2 = |2x_{ij} - x_{i-1,j} - x_{i+1,j}|; & \text{vertical;} \\ m_3 = |2x_{ij} - x_{i-1,j-1} - x_{i+1,j+1}|; & \text{main - diagonal;} \\ m_4 = |2x_{ij} - x_{i+1,j-1} - x_{i-1,j+1}|; & \text{auxillary - diagonal;} \end{cases} \quad (2.3)$$

$$m = \min [m_1, m_2, m_3, m_4] \quad (2.4)$$

Condition (4) denotes that m represents the minimum of m_1, m_2, m_3 and m_4 .

According to the structural characteristics of images, if x_{ij} is the pixel of the smooth region or edge, m should be very little. Therefore, we employ T as a threshold, if m is greater than the threshold, the noise candidate will be considered corrupted pixel, as the following expression:

$$x_{ij} \in \begin{cases} N & \text{if } m > T; \\ S & \text{else;} \end{cases} \quad (2.5)$$

where $T=30$ is the optimal threshold. It is noted that, up to now, the impulse noise classification is just to identify the noise in the edges or in the smooth regions of the image.

2.7 RANK ORDERED MEAN FILTER

After the detection of noisy pixels, the noisy pixels (N) and noise-free pixels are identified. This algorithm adopted a rank-ordered mean (ROM) filter to remove the noisy pixels. According to the first step, X_r is the input samples rearranged in the ascending order. Let ROM represents the median value of X_r and let y_{ij} denotes the output of filter which can be

$$y_{ij} \in \begin{cases} ROM & \text{if } x_{ij} \in N; \\ x_{ij} & \text{if } x_{ij} \in S; \end{cases} \quad (2.6)$$

As in the calculation of ROM, the center pixel is not taken into account, so the filter will be more robust. On the other hand, only a few pixels are performed in order statistics, so the scheme reduces the computation greatly.

2.8 ALGORITHM FOR ADAPTIVE MEDIAN FILTER

Median filter is a non-linear filter, which preserves edges while effectively removing impulse noise. Median operations are performed by row sorting, column sorting and diagonal sorting in images. An image contaminated by impulse noise is processed by the detection of the size of the impulse noise

The following are the steps involved in removing the impulse noise i.e. salt and pepper noise from the corrupted image by preserving the fine details of the image.

Step 1: Consider a window size of 3×3 of the image.

Step 2: The number of corrupted pixels in the window considered is calculated.

Step 3: If the window length is $(2n+1)$ the median filter can remove n impulses.

Step 4: This algorithm detects width of the impulse noise and then apply a median filter having appropriate length.

Step 5: Apply the median filter operation and replace the corrupted pixel with the value which is not equal to either the minimum or the maximum value considered.

The summary of these steps are given in the flow chart below listing the steps in sequence:

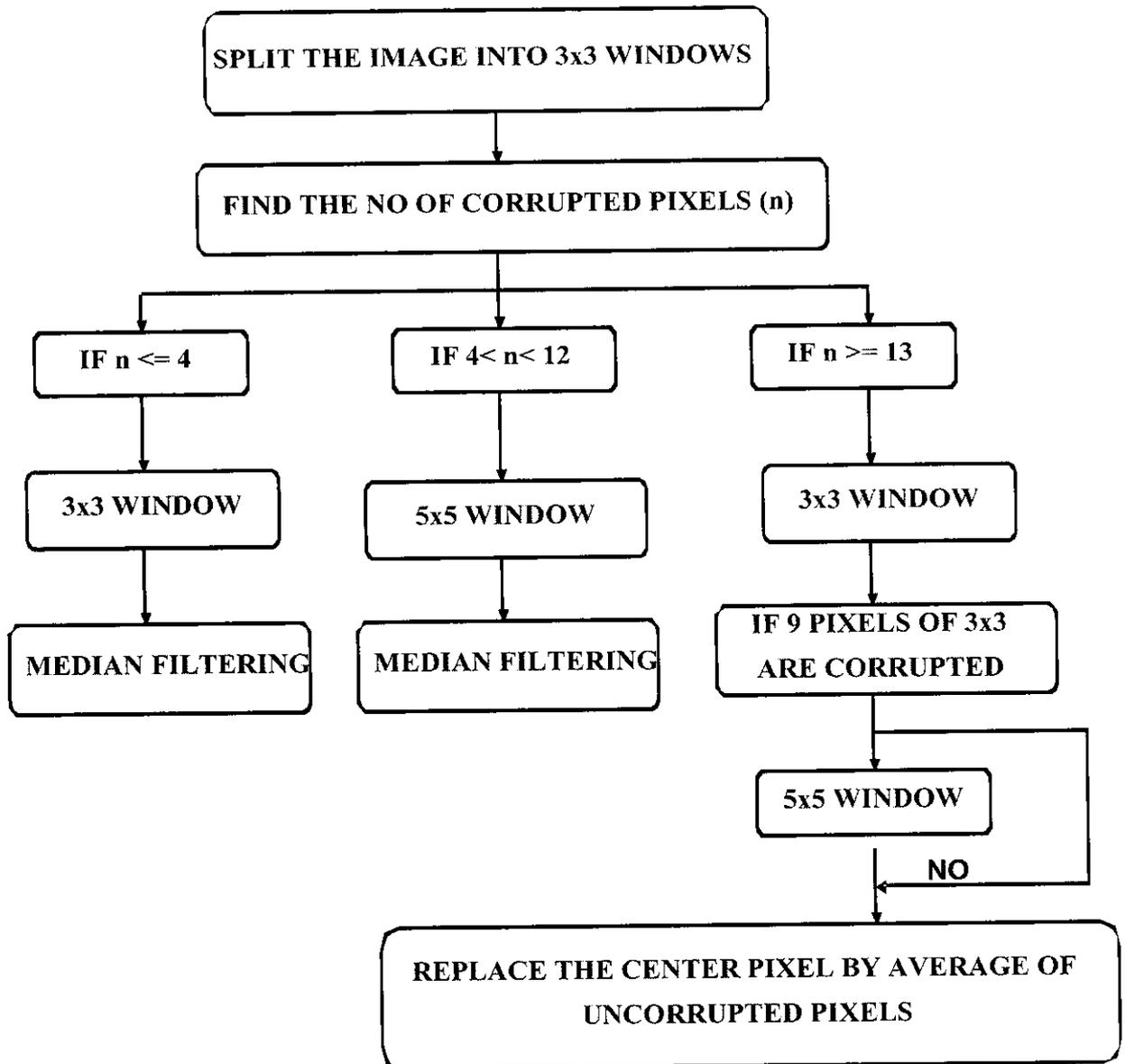


Figure 2.2 ADAPTIVE MEDIAN FILTER ALGORITHM

2.9 DETECTION OF CORRUPTED PIXELS

The first step in implementing the algorithm is to find the number of corrupted pixels in each window. Let the pixel to be processed be $P(i,j)$ and the number of corrupted pixels in the window $W(i,j)$ be 'n'. Let $P_{\max}=255$ & $P_{\min}=0$ be the corrupted pixel values and $P(i,j) \neq 0, 255$ represent uncorrupted pixels. If a pixel value is equal to 255 or 0 then there are several cases in implementing the algorithm.

2.10 IMPLEMENTATION OF THE ALGORITHM

Determine the number of corrupted pixels in the window considered. Based on the number of corrupted pixels, we can analyze the following cases:

CASE (I): $n=0$

Consider window size 3 X 3 with typical values of pixels shown as an array below. Let n be the number of corrupted pixels in the window. If $P(i,j) \neq 0, 255$ and $n=0$, then the pixel values in that window remains unchanged. For the array shown, there are no corrupted pixels in the array, therefore, the pixels are unaltered.

123	214	156
236	167	214
123	234	56



Figure 2.3 UNCORRUPTED PIXELS

CASE (II): $n \leq 4$

If the number of corrupted pixels 'n' in the window $W(i, j)$ is less than or equal to 4, i.e., $n \leq 4$, then 2D window of size 3X3 is selected. Median operation is then performed by column sorting, row sorting and diagonal sorting. The corrupted $P(i, j)$ is replaced by the median value.

Consider the array given below with window size of 3x3:

Corrupted Matrix

255	214	123
0	255	214
123	234	0

Row Sorting

→

123	214	255
0	214	255
0	123	234

Column Sorting

↓

0	123	234
0	214	255
123	214	255

Diagonal Sorting

0	123	123
0	214	255
234	214	255

Figure 2.4 MEDIAN IN 3X3 WINDOW

CASE (IV): $n \geq 13$

The number of corrupted pixels is ≥ 13 (a typical case is shown as an array below). Increasing the window size may lead to blurring; choose 3X3 median filtering. On median filtering with smaller window sizes, the output may happen to be noise pixels whenever the noise is excessive. In this case, find the average of uncorrupted pixels in the window and replace the corrupted value by the average value. The average value is taken instead of median if the number of uncorrupted pixels in the window is even.

Corrupted Matrix	3x3 window	Replaced with median																																											
<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>123</td><td>0</td><td>156</td><td>255</td><td>234</td></tr> <tr><td>255</td><td>255</td><td>123</td><td>255</td><td>0</td></tr> <tr><td>0</td><td>255</td><td>255</td><td>133</td><td>145</td></tr> <tr><td>199</td><td>0</td><td>255</td><td>0</td><td>255</td></tr> <tr><td>255</td><td>167</td><td>0</td><td>198</td><td>178</td></tr> </table>	123	0	156	255	234	255	255	123	255	0	0	255	255	133	145	199	0	255	0	255	255	167	0	198	178	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>255</td><td>123</td><td>255</td></tr> <tr><td>255</td><td>255</td><td>133</td></tr> <tr><td>0</td><td>255</td><td>0</td></tr> </table>	255	123	255	255	255	133	0	255	0	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>255</td><td>123</td><td>255</td></tr> <tr><td>255</td><td>128</td><td>133</td></tr> <tr><td>0</td><td>255</td><td>0</td></tr> </table>	255	123	255	255	128	133	0	255	0
123	0	156	255	234																																									
255	255	123	255	0																																									
0	255	255	133	145																																									
199	0	255	0	255																																									
255	167	0	198	178																																									
255	123	255																																											
255	255	133																																											
0	255	0																																											
255	123	255																																											
255	128	133																																											
0	255	0																																											
		<p>i.e., $(133 + 123) / 2 = 128$</p> <p>(133 & 123 are the uncorrupted pixels)</p>																																											

Figure 2.6 MEDIAN IN 3X3 WINDOW WITH EXCEEDED COUNT

If all the pixels in 3 X 3 windows are corrupted (a typical case is shown as an array below), then perform 5 X 5 median filtering. On median filtering the output may happen to be noise pixels as in the previous case. Find the average of uncorrupted pixels in the window and replace the corrupted value by the average value.

Corrupted Matrix

255	0	255
255	255	255
0	255	0

5x5 window

123	0	156	255	234
255	255	0	255	0
0	255	255	255	145
199	0	255	0	255
255	167	0	198	178

Replace the corrupted pixel

123	0	156	255	234
255	255	0	255	0
0	255	255	255	145
199	0	255	0	255
255	167	0	198	178

Replaced with the median value

123	0	156	255	234
255	255	0	255	0
0	255	175	255	145
199	0	255	0	255
255	167	0	198	178

$$\left\{ \frac{123+156+234+145+199+167+198+178}{8} = 175 \right.$$

175 replaces the corrupted pixel value 255}

Figure 2.7 MEDIAN FOR 9 CORRUPTED PIXELS

CHAPTER 3

EXPERIMENTAL ANALYSIS

In this section, extensive experimental results with commonly used gray-scale test images of size 512 X 512 are presented to assess the performance of the proposed algorithm. These images including "Lena", "Cameraman", "Coins", "Peppers" having distinctly different features in terms of details. These images are corrupted by various levels of salt and pepper noise. In the experimental the sliding window size M is set 3, i.e., 3 x3 sliding window for testing the detection.

The peak signal-to-noise ratio (PSNR) between the restored image and the original image is selected as the performance validated in the experiments. The PSNR is defined as

$$PSNR = 10. \log \frac{255^2}{MSE} \quad (3.1)$$

Here MSE is the mean squared error defined as

$$MSE = \frac{1}{I \cdot J} \sum_{i=1}^I \sum_{j=1}^J (y_{i,j} - s_{i,j})^2 \quad (3.2)$$

Where

- $I * J$ \longrightarrow the image size.
- $y_{i,j}$ \longrightarrow the restored pixels on the location (i, j) in the image.
- $s_{i,j}$ \longrightarrow the original pixels on the location (i, j) in the image.

The corrupted images are considered and are filtered using the two different filters and the results are compared with the obtained PSNR values.

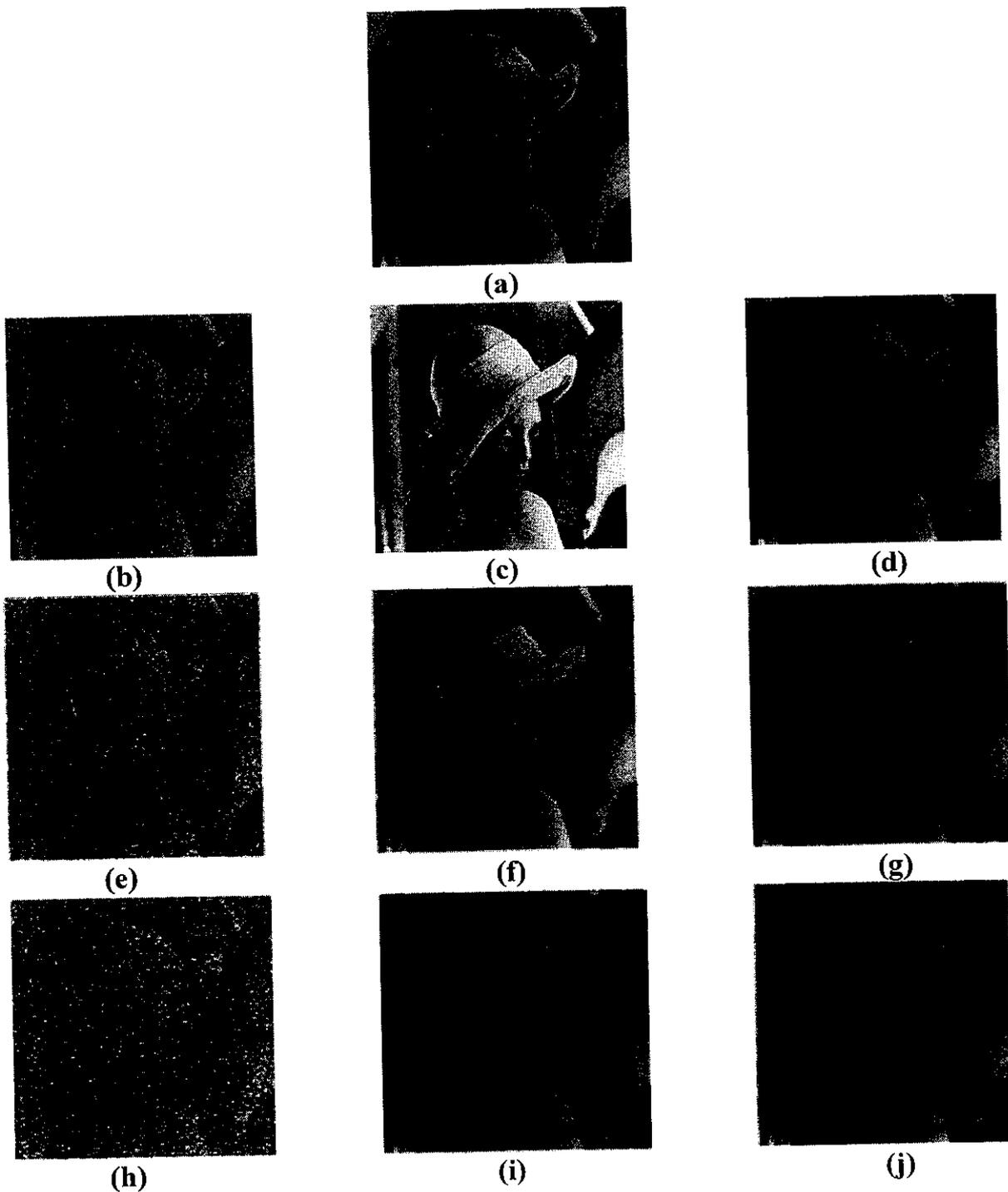


Figure 3.1 (a) The Original image: lena.bmp; (b),(e),(h) are the image corrupted with 10%,20%,30% fixed value impulse noise respectively and (c),(f),(i) are filtered images of (b),(e),(h) by ROM filter and AMF filter respectively.

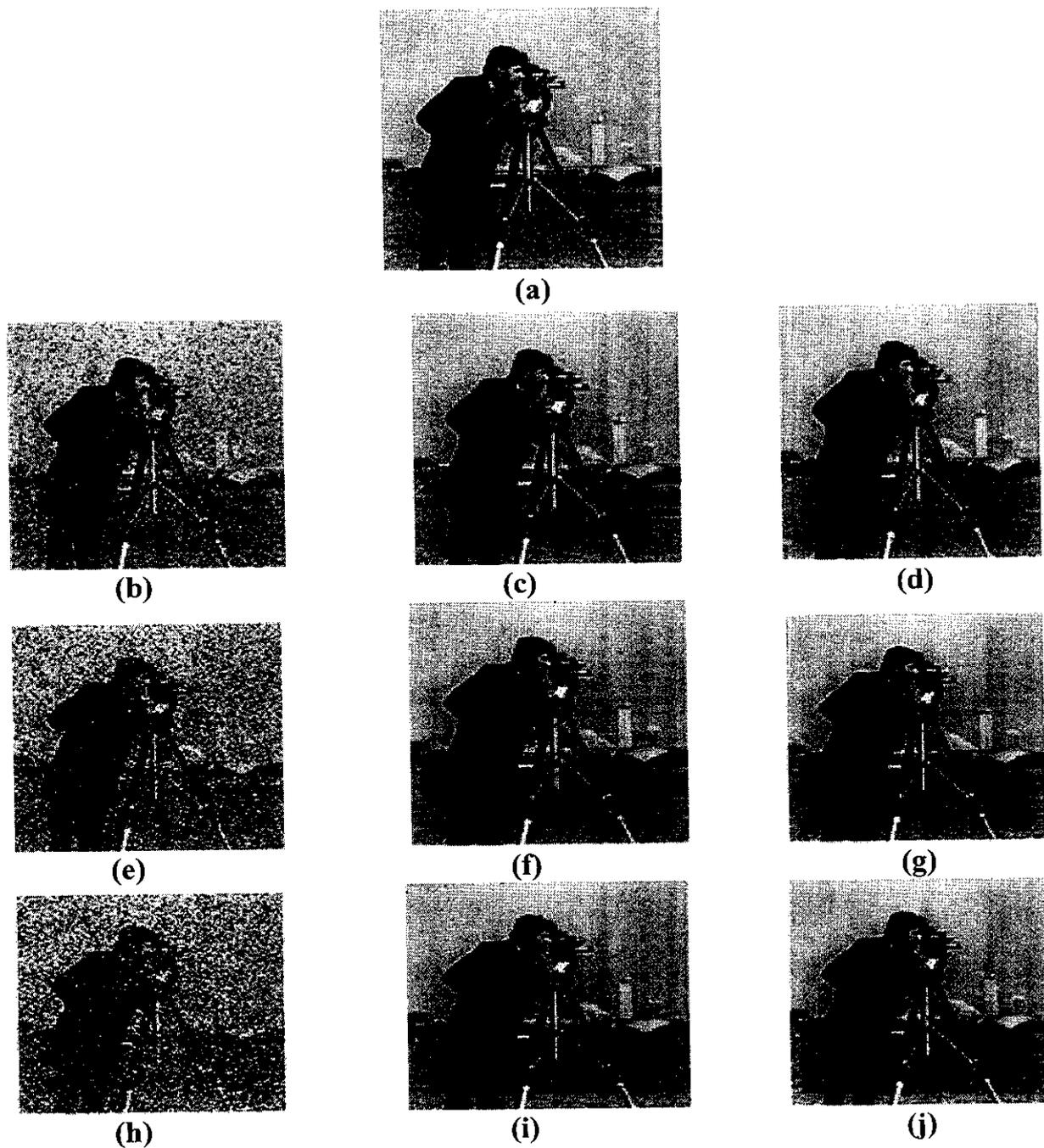


Figure 3.2 (a) The Original image: cameraman.tif; (b),(e),(h) are the image corrupted with 10%,20%,30% fixed value impulse noise respectively and (c),(f),(i) are filtered images of (b),(e),(h) by ROM filter and AMF filter respectively.

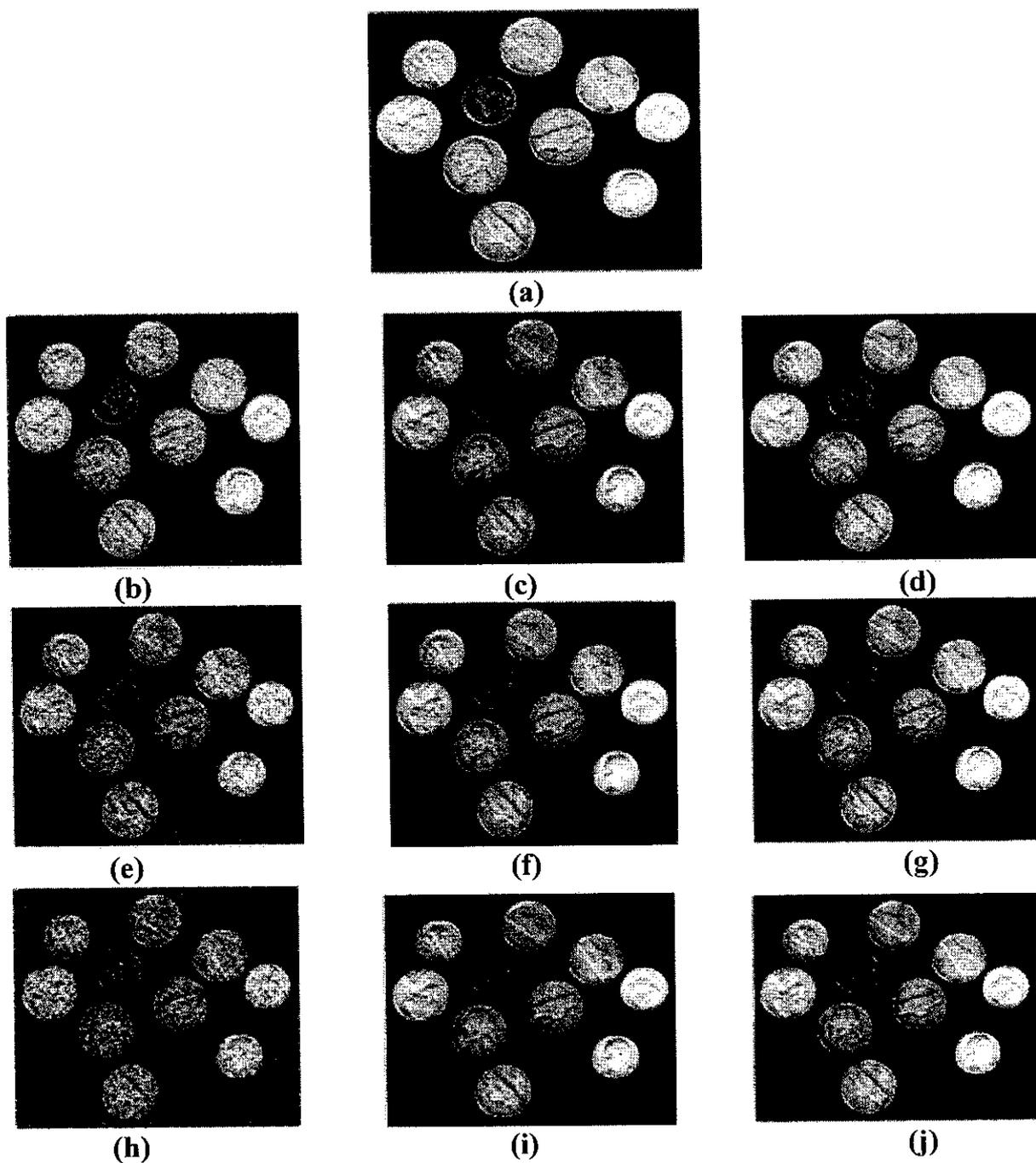


Figure 3.3 (a) The Original image: coins.png; (b),(e),(h) are the image corrupted with 10%,20%,30% fixed value impulse noise respectively and (c),(f),(i) are filtered images of (b),(e),(h) by ROM filter and AMF filter respectively.

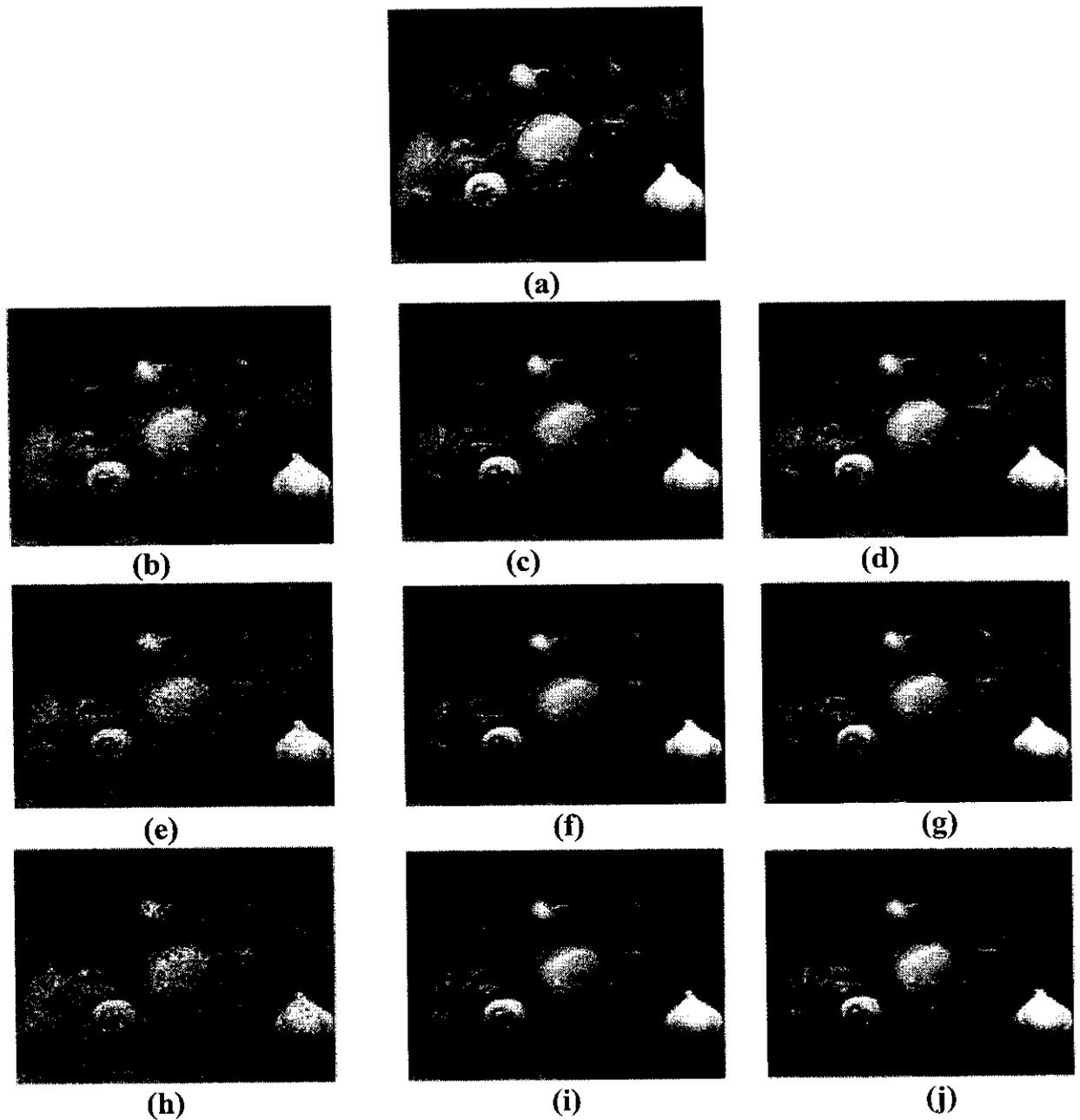


Figure 3.4 (a) The Original image: peppers.png; (b),(e),(h) are the image corrupted with 10%,20%,30% fixed value impulse noise respectively and (c),(f),(i) are filtered images of (b),(e),(h) by ROM filter and AMF filter respectively.

TABLE GIVING PSNR PERFORMANCE OF ALGORITHMS FOR IMAGES CORRUPTED WITH FIXED VALUE IMPULSE NOISE

NAME OF THE IMAGES	PSNR VALUES IN dB FOR VARIOUS NOISE DENSITIES					
	ROM FILTER			ADAPTIVE MEDIAN FILTER		
	10%	20%	30%	10%	20%	30%
LENA	42.86 dB	38.50 dB	34.11 dB	35.29 dB	29.91 dB	24.01 dB
CAMERAMAN	41.06 dB	35.75 dB	31.57 dB	25.85 dB	23.56 dB	20.67 dB
COINS	45.74 dB	39.28 dB	34.02 dB	31.97 dB	27.84 dB	23.08 dB
PEPPERS	45.14 dB	40.16 dB	35.06 dB	34.78 dB	29.71 dB	23.57 dB

Table 3.1 PSNR performances of algorithms for images with salt and pepper noise

The above results of the filtered images from the corrupted images with the help of ROM filter and Adaptive Median Filter infer that the images obtained using ROM filter are having a fine resolution compared to that of those from the Adaptive Median filter. The comparison shown in the table using PSNR values shows that the images are restored well when filtered with ROM filter giving better peak signal to noise ratios but it is less in the case of the Adaptive Median filter.

CHAPTER 4

CONCLUSION

In this project, two algorithms are compared with various kinds of images that are corrupted with different levels of noise densities. The obtained experimental results show that the Rank Ordered Mean filter has good capability of denoising the corrupted image and preserving the edge and fine details with a better resolution when compared with the original image. The PSNR values obtained in decibels for the filtered images display that images are having very low resolution when filtered with the adaptive median filter. Images like 'Lena', 'Coins', 'Cameraman' and 'Peppers' are giving the result as same as the original image when filtered with ROM filter. Hence ROM filter is very effective in removing the fixed value impulse noise from the corrupted images.

REFERENCES

- [1] Dagao Duan, Qian Mo, Yueliang Wan, Zhongming Han. "A Detail Preserving Filter for Impulse Noise Removal." International Conference on Computer Application and System Modeling (ICCASM 2010)

- [2] Abdul Majid, Muhammad Tariq Mahmood. "A Novel Technique for Removal of High Density Impulse Noise from Digital Images." 6th International Conference on Emerging Technologies (ICET 2010)

- [3] Abdul Majid, Muhammad Tariq Mahmood, and Tae-Sun Choi. "A Novel Noise-free Pixels Based Impulse Noise Filtering." Proceedings of 2010 IEEE 17th International Conference on Image Processing September 26-29, 2010, Hong Kong.

- [4] Digital image processing using MATLAB by Rafael C. Gonzalez, Richard E.Woods, Steven L.Eddins.

- [5] W.Luo, "A new efficient impulse detection algorithm for the removal of impulse noise", IEICE Trans. Fundamentals, vol.E88- A,no.1 0,pp.2579-2586,October 2005