Summer 8-6-2018

Improved Iterative Truncated Arithmetic Mean Filter

Prathyusha Surampudi Venkata
University of New Orleans, psurampu@uno.edu

Improved Iterative Truncated Arithmetic Mean Filter

A Thesis

Submitted to Graduate Faculty of the
University Of New Orleans
in partial fulfillment of the
requirements for the degree of

Master of Science
in
Engineering
Electrical

by

Prathyusha Surampudi Venkata

B.Tech. Vardhaman College of Engineering, 2015

August, 2018

Acknowledgment

First and foremost, I offer my sincerest gratitude to my advisor, Dr. Dimitrios Charalampidis, who has supported me throughout my thesis with his patience and knowledge. His timely response through emails and meeting never let me feel any stress. His encouragement and support have been a significant contribution to the completion of my thesis.

I would like to thank Dr. Ittiphong Leevongwat and professor Kim D. Jovanovich, for accepting the request to serve as members of my thesis committee. Their sincere advice helped to shape the thesis in the best way possible.

Finally, I would like to thank my family, for their constant support and unconditional love throughout my life. None of this would have been possible without them. I also would like to thank my friends for being there for me always and lifting my spirits when I needed it the most.

Dedicated to my Family.

Contents

List of Figures

List of Tables

List of Algorithms

Abstract

This thesis discusses image processing and filtering techniques with emphasis on Mean filter, Median filter, and different versions of the Iterative Truncated Arithmetic Mean (ITM) filter. Specifically, we review in detail the ITM algorithms (ITM_1 and ITM_2) proposed by Xudong Jiang [18].

Although filtering is capable of reducing noise in an image, it usually also results in smoothening or some other form of distortion of image edges and file details. Therefore, maintaining a proper trade off between noise reduction and edge/detail distortion is key.

In this thesis, an improvement over Xudong Jiang's ITM filters, namely ITM_3, has been proposed and tested for different types of noise and for different images. Each of the two original ITM filters performs better than the other under different conditions. Experimental results demonstrate that the proposed filter, ITM_3, provides a better trade off than ITM_1 and ITM_2 in terms of attenuating different types of noise and preserving fine image details and edges.

Keywords: Image Processing, Filter, Linear and Non-Linear filters, ITM filter

Chapter 1

Introduction

1.1 Motivation

In this technologically growing world, images are very often used for better communication. In many applications, they provide the means for understanding complex data. For example, in the medical field [7] X-rays, and MRI scanned images of the internal organs, such as the brain or heart, are used to study and analyze the condition of the diseased person. Another example is satellite imaging [20] which is used to understand the disturbances in the atmosphere and to predict imminent changes in the weather.

Medical images, satellite images, or other types of images may not be clear all the time. In general, images are captured using a camera or another sensor. Considering the imperfections of the various sensors, the acquired images may be corrupted with different types of noise, such as dots or speckles, and they may be blurred. Some reasons for this unwanted disturbance (noise) may include malfunctioning of the camera or the electrical interference of circuit elements while scanning or capturing the image of interest.

The noise found in images may vary from just a single speckle to errors affecting all pixels. Sometimes images are corrupted by a vast amount of noise and employing image processing techniques [13] is necessary in order to reveal as much of the original image information as possible. One of the main image processing operations for retrieving information from corrupted images is image filtering.

Image processing (or digital image processing) is a field in which certain operations performed on an image enable the extraction of useful information while clearing noise or other types of distortion which may have corrupted the image. There are several common image processing algorithms for filtering of noise, image enhancement, image segmentation [14], and feature extraction. Depending on the type of noise and on the type of information which needs to be extracted from images,

the techniques used may vary. More specifically, removing different types of noise may require the employment of different filters.

This thesis introduces an improved Iterative Truncated Mean (ITM) filtering technique for attenuating noise while preserving edges and fine image details.

1.2 Digital Images and Image Processing Software

In order to perform digital image processing techniques, it is necessary to first import scanned or captured images into the computer by converting them into digital images. Digital images are easy to access which facilitates the usage of processing techniques. A digital image is essentially a two-dimensional signal and can be viewed as a two-dimensional array.

Mathematically, an image can be defined as a function, $h(x, y)$, where x and y are the horizontal and vertical coordinates, respectively. A specific point, $h(x_o, y_o)$, located at coordinates (x_o, y_o) provides the pixel value at that particular location in the image. In other words, images can be represented in a matrix form with rows and columns [4]. Thus, most of the matrix operations can also be applied on images. In an 8-bit grayscale image, pixel values range from 0 to 255. Usually, filtering is performed by processing each pixel in the image or by processing a window (block) of pixels around each pixel in the image.

With the technology growing, many software packages and programming languages which can be used for performing image processing are now available. Some examples include MATLAB [8], C/C++, Java, and Python. For most of the programming languages or packages, there are special libraries or built-in functions for performing filtering or image enhancement.

1.2.1 Digital Image Techniques

Digital image techniques are certain operations performed on images. Depending on the objective associated with a particular application, techniques such as image enhancement, image restoration, and filtering can be used.

Image enhancement [19] is the process of enhancing the image for a better visual appearance. This can be achieved by altering the image pixels individually or in a group, in order to improve image brightness, contrast, and other characteristics of the visual quality of the image. By applying

different kinds of filters, one can remove blurriness or fuzziness, or can sharpen the edges and preserve image details.

Sometimes images may be too noisy. In some cases, image restoration, namely the process of restoring the original image from an extremely noisy image, can be used. In this case, the various layers of noise need to be retraced to be able to get back to the original image [13].

Noise can also be removed using various filtering techniques. The following section briefly discusses image filtering.

1.3 Filtering

As mentioned earlier, filtering can be used for enhancing an image, either by improving the appearance of image details or by removing noise. Filters can be used for edge detection, smoothing, and image de-noising. Regarding the latter, if a noisy image is provided as the input to a filter, the filter attempts to remove the noise and to produce a de-noised image as an output. In order to determine what kind of filter to use, one has to understand the operation of filters.

Most filters use masking, a type of matrix operation, to eliminate noise from images. As noted earlier, images can be represented as matrices with rows and columns. To perform masking, we first need to select a matrix which is much smaller than the image array. In image processing, this matrix is also called a filter window or mask.

As the term "masking" suggests, filtering involves sliding of this filter window or mask over all image pixels. When the mask is placed at a specific location in the image, (x, y), only those pixel values around (x, y) overlapping with the mask are considered. Usually, (x, y) corresponds to the central pixel in the filter window or mask. Depending on the filter used, different operations are performed on the selected pixel values. For example, the operations may include finding the mean or median of the pixel values within the window. After the corresponding filtering operation is completed for a particular location (x, y), the resultant pixel value obtained is placed at the same location, namely (x, y), in the output image. This implies that filters change the pixel values by manipulating the neighboring pixel values in succession. The sliding of the selected mask is continued until the center of the mask travels over all the pixels of the considered image [5].

The masking operation is different for different types of filters. For example, in mean (or

moving average) filtering, the overall masked value is obtained by finding the mean of all pixel values within the filter window. Similarly, in a median filter, the pixel value in the output image is computed as the median of all pixel values in the original image located within the filter window. By replacing pixels in this manner using an appropriate filter, it is possible to eliminate or reduce noise. De-noising also depends on the size of the filter window. Larger window sizes are more likely to remove noise more effectively. However, they are also more likely to eliminate image edges and other details.

Filters can be classified into linear and non-linear filters. More details about linear and non-linear filters are provided in the next two subsections.

1.3.1 Linear Filters

In linear filtering, each pixel in the output image is obtained as a linear combination of all pixel values located within a filter window. Most of the linear filters used in image processing are Finite Impulse Response (FIR) filters.

The mean filter is one of the most often-used FIR filters. As discussed earlier, mean filters obtain the output pixels by using the arithmetic mean value of neighborhood pixels in the original image. Mean filters have been shown to be very effective for attenuating Gaussian noise to a large extent [16].

Due to the use of linear operations in filter windows, linear filters cannot deal effectively with sharp edges and fine details in an image. Such filters tend to blur these image details. For instance, mean filters are easy to implement and are successful in suppressing Gaussian noise. However, the overall output of the filtering operation may not be as acceptable as the one using filters which do not suppress Gaussian noise as effectively, but which at the same time result in less blurring. Additionally, mean filters are not the best in attenuating other types of noise such as impulsive noise or mixed types of noise.

Few of the characteristics of mean filters are demonstrated in the experimental section. To improve filtering of noise while retaining image details, non-linear filters have been developed.

1.3.2 Non-Linear Filters

As the name implies, non-linear filters do not use linear combinations of pixel values to obtain the output image. A common example of non-linear filtering operation is the median operation. Therefore, the corresponding widely-used non-linear filter is called the median filter. A median filter [3] replaces pixel values in the output image with the median of the neighborhood pixels in a filter window [5]. The median filter tends to attenuate long-tailed noise and impulsive noise more effectively when compared to a mean filter. Owing to the specific non-linear operation, the median filter is capable of preserving sharp image edges, although it is not usually very effective in attenuating additive noise as some linear filters. Some details regarding the characteristics of the median filter in terms of attenuating different types of noise are discussed in later chapters.

One important disadvantage of median filters is that they tend to eliminate very fine image details such as thin lines. This disadvantage, and the fact that other filters (such as the mean filter) may be more effective in suppressing certain types of noise (such as Gaussian noise), led to the development of modified filters which inhere the merits of both linear and non-linear filters. Some information about modified filters is presented in the next section.

1.4 Modified Filters

As mentioned earlier, the mean filter attenuates Gaussian noise effectively, yet it often blurs edges and other fine image details. At the same time, the median filter is capable of attenuating long-tailed and impulsive noise best, but tends to destroy very fine image details. It can then be concluded that the mean filter and the median filter can complement each other when used together appropriately. In other words, it is desirable to use a filter which can own the merits of both mean and median filters, while eliminating their disadvantages.

Such filters include the α-Trimmed mean (αT) filter [10]. In particular, the αT filter truncates the distribution of pixels located within the filter window before taking their mean. This is achieved by cutting off a portion (of size α) of extreme (largest and smallest) pixel values. Essentially, the αT filter ensures that the maximum and minimum values, which often correspond to noise, do not influence the output. However, the αT filter has a similar disadvantage as the mean filter, although

not to the same extent, in the sense that it is not edge-preserving. The filter applies a mean to the untrimmed data, which implies that edges are still somewhat blurred. This problem is reduced as the number of trimmed values (α) increases.

To overcome this issue, the Modified Trimmed Mean (MTM) filter was proposed [1]. The MTM filter can deal with edge blurring resulting from the mean operation. However, it cannot preserve the fine details of an image. More details about the MTM filter can be found in [1].

The αT filter and MTM filter sort the data in the filter window and then find the averaging samples nearer to the computed median. There are other filters which use linear combination techniques such as, the Mean-Median (MEM) filter [2] which combines the mean and median linearly, and the Median Affine (MA) filter [1] which uses the average of all the samples weighted according to their distance from the median.

It should also be mentioned that the mean operation is a simple arithmetic operation. On the other hand, a median filter operation involves complex data sorting. Therefore, it is preferable to use a filter which can approach the median operation without employing complicated data-sorting algorithms. This was one of the motivations which led to the development of filters which use data truncation.

A new filter using data truncation was proposed in [17] to preserve the image details while suppressing the long-tailed noise in an image. This filter was developed into Iterative Truncated Arithmetic Mean (ITM) filter [18], which is discussed further in the next section.

1.5 Iterative Truncated Arithmetic Mean Filter (ITM)

The ITM filter [18] performs a simple arithmetic mean of truncated data iteratively. The ITM filter can produce an output without sorting the samples in the filter window, and yet can approach the median operation. This filter shares the merits of both mean and median filters.

The ITM filter uses dynamic truncation thresholds and stopping criteria for edge preservation and noise suppression. Instead of removing the extreme values in the window in one step, the ITM filter truncates the values to an automatically computed threshold, τ, in an iterative manner. By using proper stopping criteria, the filter can get closer to either the mean or median value depending on the type of noise. The ITM filter can effectively attenuate Gaussian noise, as well as short and

long-tailed noise, more efficiently than a median filter. It can also preserve the edges and the image details effectively.

The ITM filter is discussed in greater detail in chapter 2. This thesis deals with an improved ITM filtering technique. The proposed filter is discussed in detail and is compared to other filters in chapter chapter 3.

1.6 Summary

In digital image processing, various filtering techniques can be used to remove unwanted noise from images. Different types of filtering techniques use different types of masking. The most commonly used filters are the mean filter and the median filter.

The mean and median filters have certain advantages and disadvantages. Therefore, it was soon recognized that it is important to develop filters which take advantage of the positive characteristics of both filters. The Iterative Truncated Arithmetic Mean Filter (ITM) is a simple filter which obtains the arithmetic mean of truncated data iteratively without sorting the samples in the filter window. It has been demonstrated that ITM filters are capable of preserving edges and other image details efficiently compared to other filters. The filter proposed in this thesis is an improved version of the ITM filter.

1.7 Organization of Thesis

This thesis consists of four chapters which are organized as follows:

- chapter 1 (the current chapter) briefly introduces the reader to basic image processing concepts, including the process of filtering and various available filters.
- chapter 2 introduces the ITM filter and discusses filter properties and their characteristics compared with other filters.
- chapter 3 explains the proposed improvements and provides details regarding the improved filter.
- chapter 4 provides some concluding remarks and discusses future work.

Chapter 2

Iterative Truncated Arithmetic Mean (ITM) Filter and Properties

2.1 Introduction

Many of the modified filters used for image de-noising use some form of sorting of the pixel values. Sorting could be a time consuming process, especially when large filter windows are used. In order to avoid data sorting and complex computational operations, some filters use a simple truncation of data instead. Data truncation implies that only a subset of the pixel values located within the filter window are used for the filtering output. More specifically, extreme values (namely values which are significantly different than the mean or median of the data) are either ignored or modified so that they are closer to the mean or median of the data. Filters which use data truncation are usually less computationally complex and rely on simple arithmetic computations. One of the filters which was recently proposed in the literature and uses data truncation is the Iterative Truncated Arithmetic Mean (ITM) filter [18]. The ITM filter algorithm, as well as its properties and advantages are discussed in this chapter.

2.2 ITM Filter Algorithm

The ITM filter uses a simple iterative truncation technique to eliminate or, more accurately, modify those samples (i.e., pixel values) which are significantly different than the mean of all values within the filter window. Pixel values are considered to be significantly different than the mean, if their difference from the mean exceeds a specific threshold.

The ITM algorithm uses a dynamic threshold value for truncating the samples. The method by which the threshold is determined is described in detail in subsection 2.2.1. At this point, it

should be mentioned that the ITM algorithm follows an iterative approach. The values of those samples exceeding the threshold in each iteration are not completely discarded, but are set equal to the threshold, which is helpful in preserving image details.

Instead of using a fixed number of iterations, the ITM filter uses a stopping criterion (as explained in subsection 2.2.2). Owing to the stopping criterion, it is possible for the ITM filter to own the merits of both mean and median filters [18]. Sometimes it is possible to obtain good filtering results after just one or two iterations, while in other cases several iterations may be required.

2.2.1 Dynamic Threshold

General Discussion about the Dynamic Threshold

Noise may include many extreme values. It may be difficult to decide the range of extremity of the pixel values corresponding to noise. Considering that filter windows are relatively small, the amount of noise, the number of extreme values, and the level of extremity may be different for different filter windows. Therefore, the threshold value may also need to be different for each filter window. In other words, the threshold value should be a dynamic value, so that it adjusts itself to the extreme values in each filter window.

As mentioned earlier, instead of simply eliminating pixel values which are outside the range defined by the dynamic threshold, these extreme pixel values are replaced by the threshold. This approach has been found to be more effective in filtering noise without significantly affecting image details. Image edges can be better preserved if the filter output approaches the median. Using the dynamic threshold to truncate the samples to the threshold value helps the filter to better achieve this objective when compared with other techniques.

Three different types of threshold have been used by the ITM filter for dynamically truncating extreme pixel values. These thresholds have been shown to work out efficiently for various types of noise [18]. The following discussion outlines the way based on which the dynamic threshold can be chosen.

A dynamic threshold should make sure that the median of the sampled data is not changing after every truncation operation. In order to achieve this goal, a dynamic threshold should be

somewhat large in each iteration to ensure that the median is always within the dynamic range of the truncated pixel values [11]. By following this condition, the lower limit of the dynamic threshold should be estimated. Using this lower bound as the dynamic threshold ensures that it is possible for the ITM filter output to approach the unchanged median of the original data in most cases.

As mentioned earlier, the ITM filter can own the merits of both mean and median filters. The ITM algorithm is tailored in such a way so that the filter output tends to approach the mean or the median. Approaching the mean is advantageous for Gaussian noise, while approaching the median is preferrable for impulsive noise. However, for some types of noise, the filter may not be able to approach either the mean or the median. The dynamic threshold should be able to recognize this fact, so that the ITM filter does not waste time in trying to approach the mean or the median. In order to make sure of this, a dynamic threshold should not be considerably large. By following this condition, one should be able to estimate an upper limit for the dynamic threshold.

Mathematical Representation of the Dynamic Threshold

In order to represent the dynamic threshold mathematically, let μ be the mean of the data within a filter window, n be the number of samples (pixel values), and τ be the dynamic threshold. Then,

$$\mu = \sum_{i=1}^{n} x_i \tag{2.1}$$

where x_i is the i-th data sample. Three different versions of the dynamic threshold are presented next:

Dynamic Threshold 1 (τ_1)

$$\tau_1 = \frac{1}{2}(\delta_{high} + \delta_{low}) \tag{2.2}$$

where,

$$\delta_{high} \triangleq \mu_{high} - \mu \tag{2.3}$$

and

$$\delta_{low} \triangleq \mu - \mu_{low} \tag{2.4}$$

In the above equations, μ_{high} is the mean of all samples whose values are higher than mean, μ. More specifically, if there are some n samples of data in $\mathbf{x}$, then the samples with values higher than the mean, mu, are given by

$$x_{high} \triangleq \{x_i | x_i \in \mathbf{x},\, x_i > \mu\} \tag{2.5}$$

Similarly, μ_{low} is the mean of the sampled data whose values are lower than mean μ, namely

$$x_{low} \triangleq \{x_i | x_i \in \mathbf{x},\, x_i \leq \mu\} \tag{2.6}$$

Dynamic Threshold 2 (τ_2)

The second dynamic threshold is the standard deviation σ of the sampled data, i.e.

$$\tau_2 = \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} \tag{2.7}$$

where, as mentioned earlier, n is number of elements, μ is the mean of samples and x_i represents the i-th data sample.

Dynamic Threshold 3 (τ_3)

The third dynamic threshold is employed to make sure that the samples do not deviate significantly from the mean. It is defined as the mean absolute deviation of the samples from the mean. More specifically, the dynamic threshold τ_3 is given by,

$$\tau_3 = \frac{1}{n}\sum_{i=1}^{n}(|x_i - \mu|) \tag{2.8}$$

Properties of the Dynamic Threshold

All three dynamic thresholds have been shown to work well for different types of noise, but τ_3 seems to be more effective compared to τ_2 and τ_1. For any finite data τ_3 is smaller than both τ_1 and τ_2. Therefore, τ_3 makes sure that the ITM filter does not stay idle during iterations. The advantages

of mean absolute deviation τ_3 are as follows:

- The dynamic threshold τ_3 is greater than the distance between the mean and the median. This guarantees that the median of any finite data does not change during the truncation of the data.

- A dynamic threshold should ensure that the ITM filter will always have at least one sample to perform the operation, or else the ITM filter may stay idle. As τ_3 is the mean absolute deviation, for any given finite samples, there will be at least one sample whose distance is greater than the mean absolute value of the samples. As there will be at least a single value, τ_3, will ensure that the ITM filter will not waste the time searching for the values in a filter window.

- As τ_3 is smaller than the other dynamic thresholds, this results in faster truncation of the data, thus resulting in a faster algorithm.

- As τ_3 is the mean absolute deviation, it becomes zero automatically if the sampled mean deviates significantly from the median. This helps the ITM algorithm to approach the result in fewer iterations as the truncation value is decreased.

A dynamic threshold is critical for the filter to effectively remove noise without consuming much time. However, sometimes it results in an infinitely extending truncation loop, and convergence to the result becomes almost impossible. To avoid this issue, the ITM filter uses a stopping criterion which limits the iterations and ensures faster convergence of the algorithm. This stopping criterion is discussed next.

2.2.2 Stopping Criterion

To make sure that the ITM filter owns the merits of mean and median depending on the type of noise, it is necessary to use an appropriate stopping criterion. The stopping criterion should not result in a fixed number of iterations, but the number of iterations should be selected appropriately based on the type and level of noise. For example, stopping the iterations early would leave most samples unchanged. In turn, this would result in an output which most likely approaches the

mean. On the other hand, a larger number of iterations would most likely result in an output which approaches the median of all samples in the filter window.

Of course, knowing when to end the iterations is indeed not an easy task. The ITM algorithm proposes a stopping criterion which results in a variable number of iterations, instead of a fixed number of iterations, for each filter window. More specifically, the stopping criterion used is a combination of four individual criteria. The ITM algorithm stops the iterations if the combined criterion is satisfied.

More specifically, the combined stopping criterion, $\mathcal{S}$, is defined as follows:

$$\mathcal{S} = \mathcal{S}_1(\varepsilon_1) \vee \mathcal{S}_2(\varepsilon_2) \vee \mathcal{S}_3(\varepsilon_3) \vee [\mathcal{S}_3(\varepsilon_3) \wedge \mathcal{S}_4(\varepsilon_4)] \tag{2.9}$$

The individual criteria are described next.

Criterion 1 ($\mathcal{S}_1$)

Criterion $\mathcal{S}_1$ is helpful in ensuring that the ITM filter output can approach the median. The condition is:

$$\mathcal{S}_1(\varepsilon_1) : |n_{high} - n_{low}| \leq \varepsilon_1 \tag{2.10}$$

where, n_{high} is the number of elements in the sampled data greater than mean μ, i.e., the elements in Equation 2.5 and n_{low} is the number of elements in the sampled data lesser than and equal to mean μ, i.e., the elements in Equation 2.6, and ε_1=1.

Criterion 2 ($\mathcal{S}_2$)

The stopping criterion $\mathcal{S}_1$ may be sufficient for the ITM filter to approach the median. However, in some cases, it is possible that the algorithm may get stuck in an infinite loop without terminating. To avoid this issue, criterion $\mathcal{S}_2$ is used. The criterion is defined as:

$$\mathcal{S}_2(\varepsilon_2) : I \geq \varepsilon_2 \tag{2.11}$$

where, I is the number of iterations and $\varepsilon_2 = 2\sqrt{n}$

Criterion 3 ($\mathcal{S}_3$)

This criterion is used for the effective filtering the step edges are present within the filter window. The third criterion, $\mathcal{S}_3$ is defined as follows:

$$\mathcal{S}_3(\varepsilon_3) : \Delta n_\tau \triangleq |n_{\tau h} - n_{\tau l}| \geq \varepsilon_3 \tag{2.12}$$

where, $n_{\tau h}$ is the number of elements which are larger than $\mu + \tau$ and $n_{\tau l}$ is the number of elements which are smaller than $\mu - \tau$. Moreover, $\varepsilon_3 = (n - \sqrt{n})/2$.

Criterion 3 solves the problem when edges are found within the filter window, and especially when the filter window is located close to the center on the edge. If ε_3 was small, this could cause immature stopping of the algorithm. However, ε_3 is relatively large. For example, for a 7×7 window, $n = 49$, and $\varepsilon_3 = 21$. Criterion 3 is also used in combination with criterion 4 which is discussed next.

Criterion 4 ($\mathcal{S}_4$)

This criterion is used as an auxiliary constraint to criterion 3 (as shown in Equation 2.9), so that a smaller ε value can be used without having the iterations stopping immaturely.

This criterion is the combination of $\mathcal{S}_2$ and $\mathcal{S}_3$. It is given as,

$$\mathcal{S}_4 : \; \Delta n_\tau(I) = \Delta n_\tau(I - 1) \tag{2.13}$$

where, n_τ is the number of elements in the truncated data, and I is the number of iterations.

It can be observed that the last term in Equation 2.9 uses $\mathcal{S}_3$ and $\mathcal{S}_4$ with $\varepsilon_4 = \sqrt{n}$ as the condition to eliminate immature stopping of the algorithm. For example, for a 7×7 window, $n = 49$ and $\varepsilon_4 = 7$, which is lower than the ε_3 value used in criterion 3 (which was 21). If $\varepsilon_4 = 7$ was used on its own, the algorithm would terminate prematurely.

Properties of the Stopping Criteria

The stopping criteria make sure that the ITM filter approaches either the mean or the median (or a combination) depending on the type of noise. This is achieved in the following way:

- For an even number of samples, if stopping criterion $\mathcal{S}_1$ is satisfied, then the mean of the truncated data is the median of the samples.
- If $\mathcal{S}_1$ is satisfied for an odd number of samples, then the median is the nearest sample on either side of the truncated data.
- The horizontal and vertical step edges are handled more efficiently.
- The stopping criteria make sure that the ITM filter never stops the iterative process abruptly.
- Due to the combination of all the necessary criteria, they are efficient in edge preservation.

2.3 Algorithm for ITM Filter

The ITM filter uses the dynamic threshold and the stopping criterion described earlier. The pseudocode for the ITM filter is presented next:

Algorithm 1 ITM_1 and ITM_2 filters

Input: Image or sample data with noise
Result: De-noised (filtered) image

1: **for** Sample data x **do**
2: Compute arithmetic mean μ using Equation 2.1
3: Calculate the dynamic threshold τ using Equation 2.8
4: Compute $d_{high} = \mu + \tau$ and $d_{low} = \mu - \tau$.
5: Truncate the sample input data $x = \{x_i\}$ as follows:

$$x_o = x_i = \begin{cases} d_{high}, & \text{if } x_i > d_{high} \\ d_{low}, & \text{if } x_i < d_{low}. \end{cases} \tag{2.14}$$

6: Compute the stopping criterion $\mathcal{S}$ using Equation 2.9
7: **if** $\mathcal{S}$ is violated **then**
8: Return to *step* 1 and execute the process again,
9: **else** Terminate the iterations
10: **end if**
11: **end for**
12: Compute the output ITM_1 using Equation 2.15 and
13: Compute the output ITM_2 using Equation 2.16.

In the pseudocode above, the ITM_1 filter output is computed as:

$$ITM_1 = mean(x_o). \tag{2.15}$$

while the ITM_2 filter output is completed as:

$$ITM_2 = \begin{cases} mean(x_r), & \text{if } n_r > \xi \\ mean(x_o), & \text{Otherwise} \end{cases} \tag{2.16}$$

where, $\xi = n/4$ and n_r is the number of elements in x_r. Moreover, x_r is given as,

$$x_r = \{x_i \| x_i - \mu| \ < \tau\} \tag{2.17}$$

If there are too few samples of data, the mean may sometimes be unreliable. However, the ITM algorithm is based on the mean value. Thus, to prevent this issue from happening, ξ is used in ITM_2.

This process takes place in every filter window. In order to clarify the ITM filter operation, an example is presented next. Let us consider the following data array.

$$\mathbf{x} = \{4, 25, 89, 5, 92, 65, 78, 102, 56, 3, 156, 8\} \tag{2.18}$$

Then, the mean of these data samples is equal to:

$$\mu = mean(\mathbf{x}) = 56.9167 \tag{2.19}$$

using Equation 2.8 we can compute the following parameters:

$$\tau = \frac{1}{n} \sum_{i=1}^{n} (|x_i - \mu|) = 40.0833 \tag{2.20}$$

$$d_{high} = \mu + \tau = 97 \tag{2.21}$$

$$d_{low} = \mu - \tau = 16.8333 \tag{2.22}$$

Replacing the extreme values in array $\mathbf{x}$ (i.e. truncating the samples to the dynamic threshold), results in the following array:

$$x_o = \{16.8333, 25, 89, 16.8333, 92, 65, 78, 97, 56, 16.8333, 97, 16.8333\} \tag{2.23}$$

If we assume that the stopping criterion is satisfied, the output of the ITM_1 filter is:

$$ITM_1 - output = mean(x_o) = 55.5278. \tag{2.24}$$

Because $\varepsilon = n/4 = 3$ and $n_r > 3$ then x_r includes all the values which are not extreme and replaced by the threshold value, namely $x_r = \{25, 89, 92, 65, 78, 56\}$. Therefore, the output of the ITM_2 filter is:

$$ITM_2 - output = mean(x_r) = 67.5000. \tag{2.25}$$

$$\phi = median(x) = 60.5. \tag{2.26}$$

The ITM_1 filter will consider the mean of the truncated data, and ITM_2 will consider the mean of the data which are not truncated (original data). There may be different range of values in a filter window, but the output should contain a value which is nearer to most of the values, otherwise the important details may not be considered. For any given data, a filter would be better, if the output value is smaller and nearer to the sample data.

In the above example, we can see that the output value of ITM_1 is smaller than the ITM_2 output and the median. The output value of ITM_1 is nearer to most of the data compared to the ITM_2 output. The above mentioned example, does not have extreme values. If the above sampled data has extreme values, the output of the ITM_2 filter will be nearer to most of the values in the filter window compared to ITM_1. As, the ITM_2 filter does not consider the truncated data, for the data with many extreme values, ITM_2 filter works better (comparison is shown in the next section). So, for the given sample data, we can say that the output of the truncated data outperformed the median filter. So, for a few cases, it is safe to say that the ITM filters can approach (sometimes outperform) the median filter. Here, we can observe that the ITM filters, without using any complex data sorting algorithms outperformed the median filter.

The ITM filter results for different types of noise and comparisons with the mean and median filters are presented next.

2.4 Comparisons between Different Filters for Different Types of Noise

The performance of any filter can be evaluated by its ability to attenuate noise in an image for different noise scenarios. For example, all pixels in the image or a percentage of pixels in the image (e.g., 75% or 25%) may have been corrupted by noise. Moreover, some pixels may have been corrupted by one type of noise, while other pixels may have been corrupted by a different type of noise. Also, noise may have been present in the image at different levels. In order to properly evaluate a filter, several of these scenarios have to be studied. In addition to studying the filter's capability to remove noise, its capability to preserve edges and other details have to be evaluated.

To test the properties stated above, the performance of the two ITM filters (ITM_1 and ITM_2) has been tested for different types of noises. Moreover, the ITM filters have been compared with the mean and median filters. All comparisons have been performed in terms of the Mean Absolute Error (MAE) or the Mean Square Error (MSE).

The ITM filters were described in section 2.3. It should also be mentioned that all types of noise created were identically distributed with zero mean. Similar results to the ones presented next were also presented in [18]. However, one of the objectives of this thesis was to implement again all algorithms and to confirm the results in [18].

2.4.1 Attenuating Gaussian Noise

If the noise which has contaminated an image obeys a normal distribution, it is called Gaussian noise. To test the capability of filters to attenuate Gaussian noise, a constant image of size 100×100 with a constant pixel value of 100 was chosen. This constant image was contaminated by Gaussian noise, and filters with window sizes equal to 9 (3×3), 25 (5×5), 49 (7×7), and 81 (9×9) were tested. The average MAE results for 1000 independent experiments are shown in Figure 2.1. Using the stopping criteria described earlier, the average number of iterations for both the ITM filters were $1.7, 1.5, 4.2, 3.1$ for filter sizes of $9, 25, 49, 81$ respectively. As a reminder, the two ITM filters use the same stopping criteria. As a result, the number of iterations is the same for both of them. A constant image is used to evaluate the ability of filters to suppress noise, without testing their

ability to preserve edges and other details.

As mentioned earlier, it is evident that mean filter attenuated Gaussian noise better than any other filter. However, we can also observe that both ITM_1 and ITM_2 filters outperformed the median filter, while the ITM_1 filter performed better than the ITM_2 filter.

Of course, it is important to keep in mind that the image used in this example is a constant one. In other words, there are no fine details or edges present in the image. Therefore, although the mean filter outperformed all other filters in this example in terms of MAE, this would not necessarily be the case for a more realistic image. The mean filter tends to blur edges and other details. This implies that the MAE associated with the mean filter tends to deteriorate significantly when many edges and other fine details are present in the image, especially as the filter window size increases.

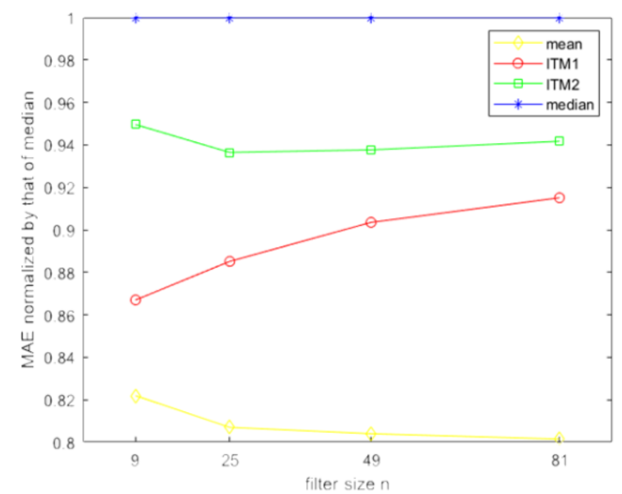


Figure 2.1: Results for Gaussian noise using different filters and window sizes. The MAE values for each window size have been normalized by the corresponding MAE results of the median filter.

2.4.2 Attenuating Laplace Noise

The same constant image contaminated with Laplace noise [15] was also tested. Once again, the image was filtered by mean, median, ITM_1, and ITM_2 filters. The MAE results for 1000 independent experiments is shown in Figure 2.2. The average number of iterations for both the ITM filters were $1.8, 1.7, 4.5, 3.6$ for $9, 25, 49, 81$ filter sizes respectively. The ITM_1 filter outperformed the median filter and any other filter in this example. In general, the MAE performance of the median filter and the two ITM filters is close. On the other hand, the mean filter performed

significantly worse.

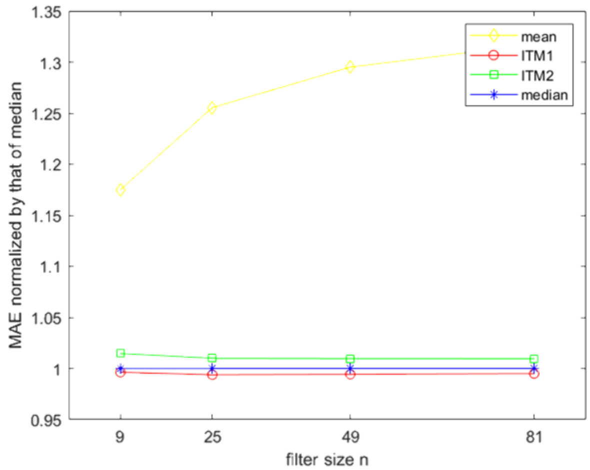


Figure 2.2: Results for Laplace noise using different filters and window sizes. The MAE values for each window size have been normalized by the corresponding MAE results of the median filter.

2.4.3 Mixed Type of Noise

For these experiments, a constant image was contaminated with a weighted sum of two types of noise. The overall noise is specified by $P_\epsilon = \{(1-\epsilon)\Phi + \epsilon H\}$ [9], where $\epsilon \in [0\ 1]$, Φ is Gaussian noise and H is a long-tailed distribution. In what follows, σ_n is the standard deviation of the Gaussian distribution. Two types of long-tailed noise are used in the testing. More details are presented next.

Gaussian and Impulse Noise

In this experiment, Gaussian noise Φ of standard deviation σ_n and impulse noise H which follows the distribution $H(x) = 0.5\delta(x - 3\sigma_n) + 0.5\delta(x + 3\sigma_n)$ were mixed. The value of ϵ was set equal to 0.25. More specifically, 25% of pixels were contaminated with impulsive noise, and 75% of pixels were contaminated with Gaussian noise. The results are shown in Figure 2.3. The average number of iterations for both the ITM filters were $1.8, 1.6, 4.4, 3.4$ for filter sizes $9, 25, 49, 81$, respectively. We can observe that the ITM_2 filter outperforms all other filters. This could be expected, because impulsive noise contains many samples with extreme values. The ITM_2 filter output often uses the average of the x_r data (see Equation 2.16) which completely excludes these extreme values. On the other hand, the ITM_1 filter always uses the thresholded version of these extreme values which are

still present in x_o.

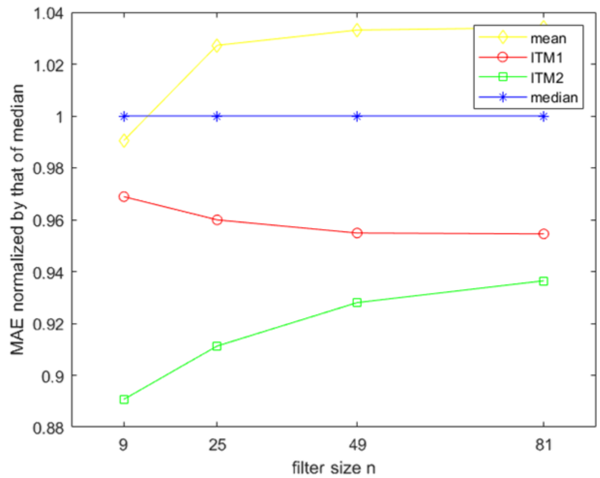


Figure 2.3: Results for mixed noise (Gaussian and impulsive) for different filter and filter sizes. The MAE values were normalized with respect to the corresponding MAE values for the median filter.

Gaussian and Laplace Noise

In this experiment, the same constant image of size 100×100 was contaminated by Gaussian noise Φ with standard deviation σ_n and Laplace noise H with standard deviation $1.3\sigma_n$. In this case, ϵ was set equal to 0.5. The results are shown in Figure 2.4. The average number of iterations for both the ITM filters were $1.7, 1.5, 4.5, 3.4$ for filter sizes $9, 25, 49, 81$, respectively. Here, we can observe that the ITM_1 filter outperforms all other filters. Although the mean filter attenuated Gaussian noise more effectively than any other filter, it cannot attenuate mixed noise as effectively. Similarly, the median filter can attenuate Laplace noise effectively, but not mixed types of noise as in this case.

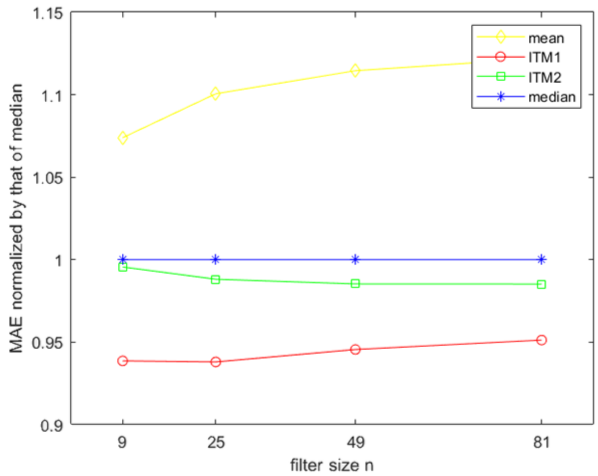


Figure 2.4: Results for mixed noise (Gaussian and Laplace) for different filter and filter sizes. The MAE values were normalized with respect to the corresponding MAE values for the median filter.

2.4.4 $\alpha-$Stable Noise

An $\alpha-$stable [6] noise follows a stable distribution with an α value ranging from 0 to 2. The $\alpha-$stable distribution considers four parameters, namely α, which is a stability parameter, β, which is a skewness parameter, γ, which is a scale parameter, and δ, which is a location parameter. In these experiments, β and δ were considered as zero, while γ was a constant value set equal to 10. Here, α ranged from 0.5 to 1.8. The same constant image was contaminated with noise following these specifications. The $\alpha-$stable distribution has many values which are very large. In these experiments, the $\alpha-$stable noise was tested for two cases: First, without clipping extreme values below 0 and above 255, and, second, after clipping extreme values below 0 and above 255. For 10 runs, and a 5×5 filter window, without clipping, the MSE results are presented in Table 2.1. The average iterations for both the filters were $2.9, 3, 2.8, 2, 1.7$ for α values of $1.5, 0.8, 1.2, 1.5, 1.8$, respectively. It can be observed that the median filter performs better than all filters for small α sizes, while ITM_1 and ITM_2 outperform all filters for higher α values. The mean filter performs significantly poorer than all other filters.

Table 2.1: MSEs for image contaminated with $\alpha-$stable noise without clipping.

α	0.5	0.8	1.2	1.5	1.8
Mean	6.9609e+18	2.4480e+05	615.1054	54.0566	14.3839
Median	11.0359	10.9091	11.9280	11.7697	12.3743
ITM1	38.07	14.6137	11.444	10.3233	11.0503
ITM2	28.9615	12.6198	12.4547	12.9665	11.0503

For 10 runs, a 5×5 filter window with clipping, the MSE results are presented in Table 2.2. The average iterations for both the ITM filters were $4.57, 3.2, 2.24, 6, 1.6$ for α values of $1.5, 0.8, 1.2, 1.5, 1.8$, respectively. The conclusions are the same as the first case for the median and ITM filters. However, the mean filter performs better than in the first case, although it is still outperformed by all other filters.

Table 2.2: MSEs for α - stable noise contaminated image with clipping.

α	0.5	0.8	1.2	1.5	1.8
Mean	210.4157	100.4032	42.7230	23.2344	12.6233
Median	11.6810	10.8239	12.1287	12.6463	12.5901
ITM1	23.1572	13.7089	11.6797	10.9459	10.2116
ITM2	16.3721	12.5727	12.5269	12.1989	11.0503

As the value of α increases the impulsive nature of the noise decreases. As, ITM_2 filter can attenuate the impulsive noise better, for the values lower than 1.2, it outperformed ITM_1 filter. For higher α value of 1.2 to 1.8, the ITM_1 filter outperforms all the filters.

2.4.5 Gaussian and α-Stable Noise for a Real Image

To test both noise removal and fine detail preservation capabilities of the filters, a real image is considered in this experiment. The original image is the commonly used 'Lena' image which is of size 512×512. A mixture of Gaussian and $\alpha-$stable noise with mixture parameter $\epsilon = 0.5$ was used. The standard deviation of Gaussian was fixed at 10, the α value varied from 0.5 to 1.8, and the γ value was fixed to 10. It is evident that all the filters work better when the extreme values of $\alpha-$stable noise (below 0 and above 255) were clipped to 0 and 255. Using 10 runs for the contaminated image, the MSE results are shown in Table 2.3.

Table 2.3: MSEs for the 'Lena' image contaminated with mixed Gaussian and $\alpha-$stable noise where extreme values (below 0 and above 255) were clipped.

α	0.5	0.8	1.2	1.5	1.8
Mean	168.2484	110.8972	78.7190	68.2657	62.8828
Median	55.6640	53.6999	52.2093	51.6081	51.0055
ITM1	57.3673	54.2557	51.9652	50.6081	50.1734
ITM2	57.8661	55.3467	53.1891	52.1087	51.1142

It can be observed that the median and ITM filters perform similarly. The ITM_1 filter performs moderately better than the median filter and ITM_2 as $alpha$ increases. The average iterations for

both the ITM filters were $3, 3.4, 2.5, 2, 2.8$ for α values of $1.5, 0.8, 1.2, 1.5, 1.8$, respectively.

2.4.6 Gaussian Noise and Laplace Noise for Real Image

A second experiment using 'Lena' was performed. This time, the image was contaminated with a mixture of Gaussian ($\sigma_n = 1$) and Laplace noise ($1.3\sigma_n$) using a mixture parameter of $\epsilon = 0.5$. The MAE results for $9, 25, 49, 81$ filter sizes are shown in Figure 2.5. For the same contamination and mixture of noise in Figure 2.4, we observed that the ITM_1 filter performed better compared to the ITM_2 filter and all other filters. However, for a real image, we can observe that the ITM_2 filter outperformed the ITM_1 filter. This is due to the fine image details present in the 'Lena' image. In other words, ITM_2 can better preserve image details than ITM_1. For this exact reason, the median filter provided a lower MAE than all other filters. Of course, this happened in this particular experiment because the noise level ($\sigma_n = 1$) was low. Thus, removing noise was not as crucial as preserving image details.

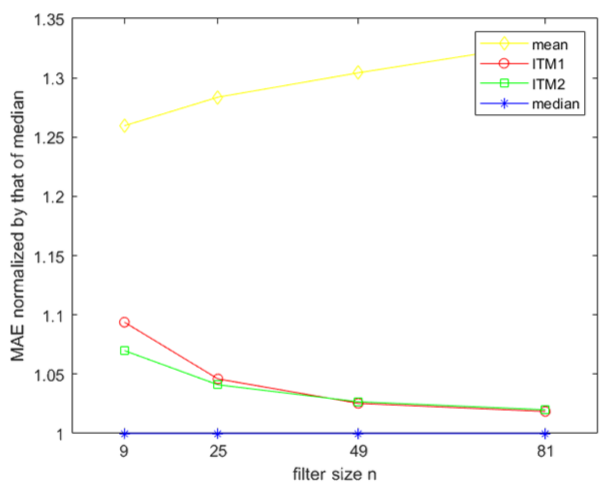


Figure 2.5: Results for the 'Lena' image contaminated by mixed noise (Gaussian and Laplace) for different filters and filter sizes. The MAE are normalized the corresponding MAE values of the median filter.

2.5 Properties

The properties of the ITM filter are presented as,

- Instead of using complex data sorting algorithms, the ITM algorithm approaches the median by using simple arithmetic operations.

- The ITM filter can attenuate different kinds of noise without prior knowledge about the type of noise.
- For a large number of iterations, in any filter window, both the ITM_1 and ITM_2 filters converge to the median.
- Sometimes, for a small number of iterations, the ITM filters can outperform the mean and median filters.
- The output of the ITM_1 filter will be equal to that of the median filter, if there are no samples which are outside of the threshold.
- By using fewer iterations, the ITM_2 filter can preserve image step edges.
- For a noisy step edge, the ITM_2 filter seems to attenuate noise better than the median filter.
- The homogeneous area in any image can be preserved by the ITM filters.
- Due to the truncation algorithm of ITM filters, they remove impulses from an image efficiently.
- Even though ITM filters cannot perfectly preserve image edges, the blurring effect is much lighter compared to the mean filter.

From the above comparison, we can observe that the ITM filters performed better than mean and median filters. Also, ITM_1 filter performed better than ITM_2 filter for short-tailed noises and ITM_2 filter performed better than ITM_1 for long-tailed noises while preserving image details. Here, we can observe that only one of the filter works better for a particular type of noise. For a larger filter window size, the ITM filter requires more time to provide an output. The realization of the filters is discussed in [12]. As there are two filters to compute, it is more time consuming. So, instead of having two filters, we propose a single filter which can own the merits of both the ITM_1 and ITM_2 filters and can outperform the ITM_2 filter in most of the cases. The improved filter operation and properties were proposed in chapter 3.

Chapter 3

Improved Iterative Truncated Arithmetic Mean Filter

3.1 Motivation

From all experiments presented in the previous chapter it is evident that the ITM_1 and ITM_2 filters performed better in terms of MAE or MSE than the mean and median filters for most of the cases. It can also be observed that the ITM_1 filter worked better than ITM_2 for short-tailed noises, while the ITM_2 filter worked better than ITM_1 for long-tailed noises and for real images. The main reason is that ITM_2 can better preserve edges and image details than ITM_1. The performance of the ITM filters provides the motivation to develop an improved ITM filter.

The idea for the improved filter is exactly based on the observation that each of the ITM_1 and ITM_2 filters provided a lower MAE than the other depending on the type of noise. Ideally, one could design a filter with matches the performance of the best out of the two filters depending on the type of noise. In order to achieve this goal, it is important to revisit how the ITM_1 and ITM_2 filters differ from each other.

As a reminder, the computation of the ITM_2 filter output is different than that of the ITM_1 filter in that it sometimes uses the average of the data, x_r, from which the extreme values have been completely eliminated. On the other hand, the ITM_1 filter always uses the average of all data located in the filter window, x_o, in which some values have been replaced by thresholded values. The ITM_2 filter chooses which of the two averages to use (the average of x_r or the average of x_o) based on a condition. In particular, if n_r, namely the number of elements in x_r, is larger than $\xi = n/4$ then the average of x_r is used. Otherwise, instead of using the average of x_r, which would be based only of a few samples, the average of x_o is used.

Therefore, a different filter somewhere between ITM_1 and ITM_2 can be designed by modifying

this part of the filter which determines how the output is computed from x_r and x_o. The proposed filter which is based on this idea is discussed in the following section.

The performance of the proposed filter in comparison to ITM_1 and ITM_2 has evaluated for different types of noise and for different images images. For simplicity, the proposed filter is named ITM_3.

3.2 Outline of the ITM_3 Algorithm

As mentioned earlier, the ITM filter uses a dynamic threshold and a stopping criterion for terminating the iterations. The ITM_3 filter also uses the same dynamic threshold and stopping criterion. The difference is in the computing the output of the ITM_3 filter. The algorithm of the ITM_3 filter is presented below.

3.2.1 Algorithm

The pseudo code for ITM_3 filter is the following:

Algorithm 2 ITM_3 filter

Input: Image or sample data with noise
Result: Noise filtered image
1: **for** Sample data x **do**
2: Compute arithmetic mean μ using Equation 2.1
3: Calculate the dynamic threshold τ using Equation 2.8
4: Compute $d_{high} = \mu + \tau$ and $d_{low} = \mu - \tau$.
5: Truncate the sample input data $x = \{x_i\}$ by,

$$x_o = x_i = \begin{cases} d_{high}, & \text{if } x_i > d_{high} \\ d_{low}, & \text{if } x_i < d_{low}. \end{cases} \tag{3.1}$$

6: Compute the stopping criterion S using Equation 2.9
7: **if** S is violated **then**
8: Return to *step*1 and execute the process again,
9: **else** Terminate the iterations
10: **end if**
11: **end for**
12: Compute the output ITM_3 using Equation 3.2

where,

$$ITM_3 = \begin{cases} mean(x_r), & \text{if } n_r > 2\xi \\ (mean(x_o) + mean(x_r))/2, & \text{if } 2\xi \geq n_r > \xi \\ mean(x_o), & \text{Otherwise .} \end{cases} \tag{3.2}$$

As a reminder, n is the number of elements in x, $\xi = n/4$, n_r is the number of elements in x_r, and x_r is given as follows:

$$x_r = \{x_i || x_i - \mu| \;\; < \tau\}. \tag{3.3}$$

It can be observed that the output of the ITM_3 filter depends on three conditions. The filter chooses one out of three different outputs depending on the overall number of elements, n, in the filter window, and the number of elements, n_r, in the truncated array, x_r.

Intuitively, if the number of elements in the truncated data, x_r, is large, then using the mean of x_r provides a reliable output. At the same time, x_r excludes extreme (and thus potentially noisy) pixel values. If the number of elements in the truncated data, x_r, is very small, then using the mean of x_r does not provide a reliable output. Therefore, the mean of the data in x_o is used instead. Finally, if n_r is equal to a moderate value, then the average of the two means (the mean of x_r and the mean of x_o) is used. In this way, the proposed filter uses a smoother transition between the two extreme conditions to calculate the output.

If the MAE performance of the ITM_3 filter was simply somewhere between that of the ITM_1 and ITM_2 filters, the new filter would not be a very attractive alternative. As was mentioned earlier, each of the two original ITM filters performs better than the other depending on the type of noise. In this sense, the two filters can be considered equivalent. However, experimental results demonstrated that the proposed ITM_3 filter performed consistently better than ITM_2, making ITM_3 a better alternative.

In what follows, experimental results which compare the two original ITM filters and the proposed ITM_3 are presented.

3.3 Experimental Study

In this section, the performance of the ITM_3 filter which was described in subsection 3.2.1 is evaluated. All types of noise used are identically distributed with zero mean. For all types of noise, the images tested include a constant image, the 'Lena' image, and the 'Bank' image. The latter two real images have been used to test the edge preservation capabilities ('Bank' image) and fine detail preservation capabilities ('Lena' image) of the filters. The images of 'Lena' and 'Bank' are shown in Figure 3.1.

(a) Image of 'Lena'.

(b) Image of 'Bank'

Figure 3.1: Real images used for testing the edge preservation capabilities and the fine image detail preservation capabilities of the filters.

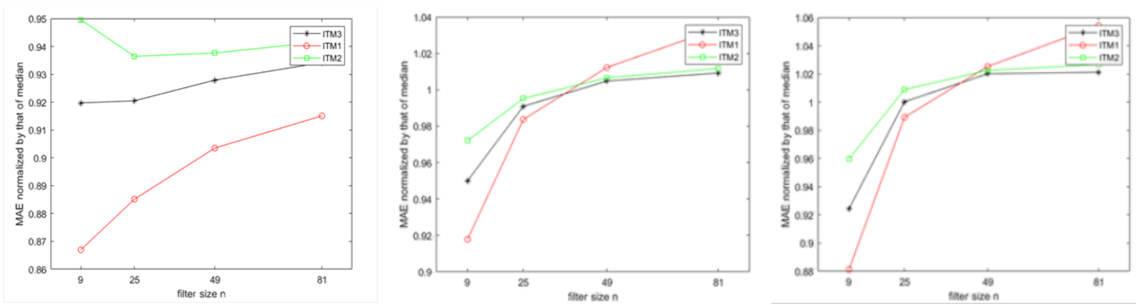
It should also be mentioned that for all results presented in this section, the MAE values associated with different filters were normalized by the corresponding MAE values of the median filter. For example, if the normalized MAE for a particular filter is equal to 1, this implies that this filter and the median filter of the same window size provide the same MAE.

3.3.1 Gaussian Noise

For this experiment, the three images have been contaminated with Gaussian noise. For the constant image the standard deviation was $\sigma_n = 1$, while for the real images two standard deviation values, namely $\sigma_n = 10$ and $\sigma_n = 15$, were used. Filter windows of size $9(3 \times 3), 25(5 \times 5), 49(7 \times$

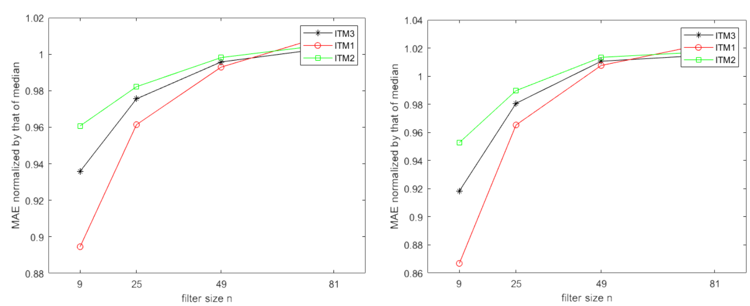
7), 81(9×9) were used. The average MAE results were obtained for 1000 runs for the constant image and for 10 runs for the two real images. The results are presented in Figure 3.2 and Figure 3.3.

As mentioned earlier, it is evident that the proposed ITM_3 filter attenuates Gaussian noise consistently better than the ITM_2 filter. It can also be observed that for the two real images, the proposed ITM_3 filter also outperforms the ITM_1 filter for larger window sizes.



(a) (b) (c)

Figure 3.2: Comparison in terms of MAE between the two original ITM filters and the proposed ITM_3 filter for images contaminated with Gaussian noise: (a) Constant image (standard deviation of noise, $\sigma_n = 1$), (b) 'Lena' image (standard deviation of noise, $\sigma_n = 10$), (c) 'Bank' image (standard deviation of noise, $\sigma_n = 10$).



(a) (b)

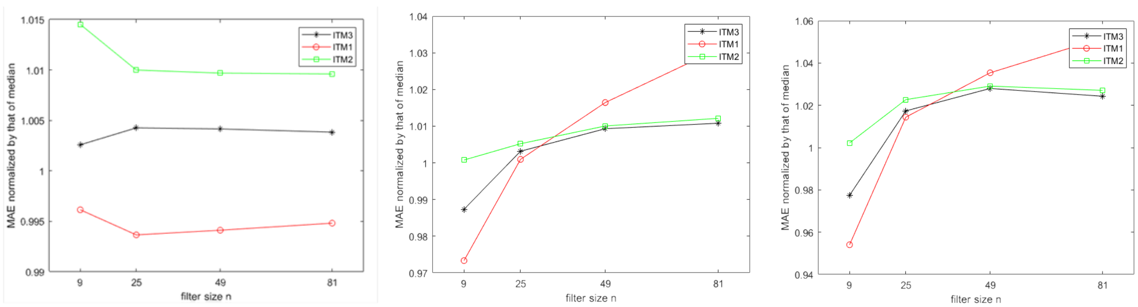
Figure 3.3: Comparison in terms of MAE between the two original ITM filters and the proposed ITM_3 filter for images contaminated with Gaussian noise of standard deviation 15: (a) 'Lena' image, (b) 'Bank' image.

3.3.2 Laplace Noise

In this section, the same experiment was repeated but for images contaminated with Laplace noise. For the constant image the standard deviation was $\sigma_n = 1$, while for the real images two standard deviation values, namely $\sigma_n = 10$ and $\sigma_n = 15$, were used. Again, filter windows of size $9(3 \times 3), 25(5 \times 5), 49(7 \times 7), 81(9 \times 9)$ were used. The average MAE results were obtained for 1000

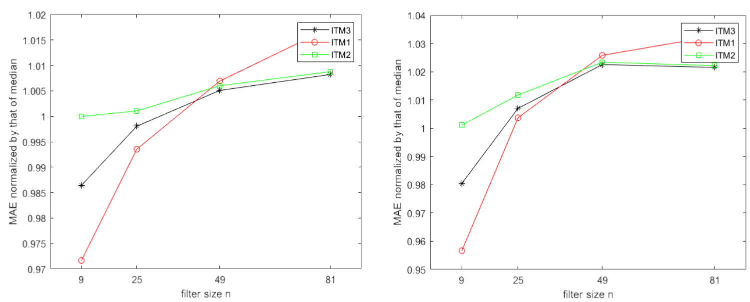
runs for the constant image and for 10 runs for the two real images. The results are presented in Figure 3.4 and Figure 3.5 .

The results are very similar to the ones obtained for Gaussian noise. More specifically, the proposed ITM_3 filter attenuates Laplace noise consistently better than the ITM_2 filter. Moreover, the proposed ITM_3 filter outperforms the ITM_1 filter for larger window sizes.



(a) (b) (c)

Figure 3.4: Comparison in terms of MAE between the two original ITM filters and the proposed ITM_3 filter for images contaminated with Laplace noise: (a) Constant image (standard deviation of noise, $\sigma_n = 1$), (b) 'Lena' image (standard deviation of noise, $\sigma_n = 10$), (c) 'Bank' image (standard deviation of noise, $\sigma_n = 10$).



(a) (b)

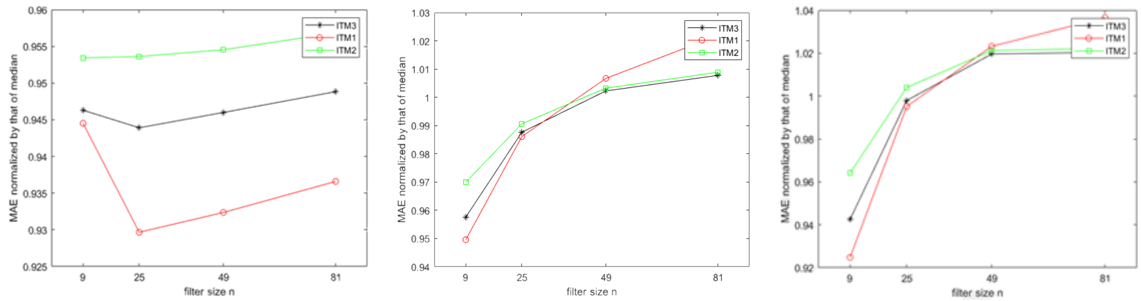
Figure 3.5: Comparison in terms of MAE between the two original ITM filters and the proposed ITM_3 filter for images contaminated with Laplace noise of standard deviation $\sigma_n = 15$: (a) 'Lena' image, (b) 'Bank' image.

3.3.3 Gaussian and Impulsive Noise

Similarly to the experiments of subsection 2.4.3, in this experiment, the images are contaminated with a mix of Gaussian and impulsive noise.

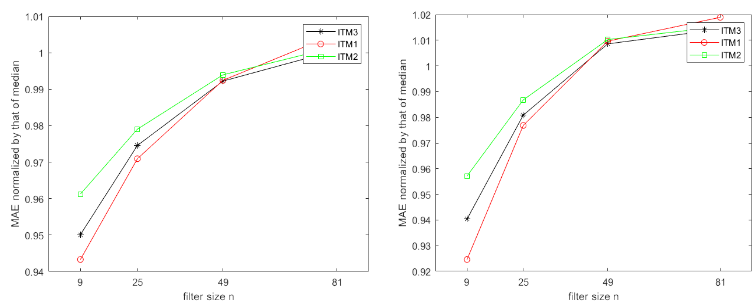
First, results for ϵ=0.15 (i.e., 85% of Gaussian noise and 15% of impulsive noise) are shown in Figure 3.6 and Figure 3.7, for standard deviations of the Gaussian noise equal to $\sigma_n = 10$ and

$\sigma_n = 15$, respectively. We can observe that the proposed ITM_3 filter outperformed the ITM_2 filter for all cases. In the case of the two real images, the proposed filter also outperformed ITM_1 filter for larger window sizes.



(a) (b) (c)

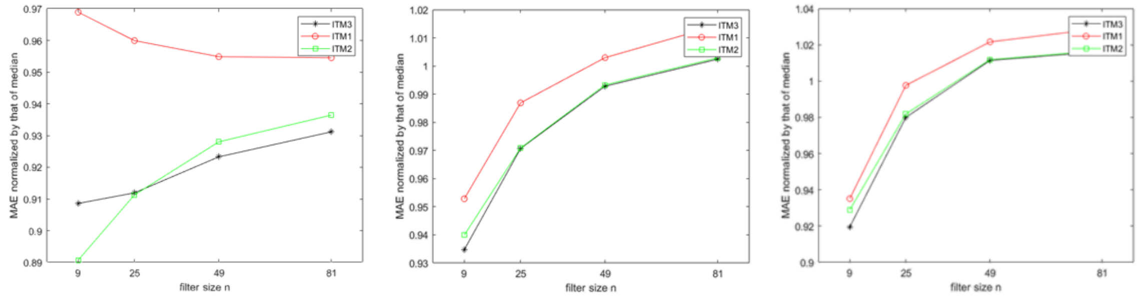
Figure 3.6: Comparison in terms of MAE between the two original ITM filters and the proposed ITM_3 filter for images contaminated with mixed noise (Gaussian and impulsive with $\epsilon = 0.15$): (a) Constant image (standard deviation of the Gaussian noise is $\sigma_n = 1$), (b) 'Lena' image (standard deviation of the Gaussian noise is $\sigma_n = 10$), (c) 'Bank' image (standard deviation of the Gaussian noise is $\sigma_n = 10$).



(a) (b)

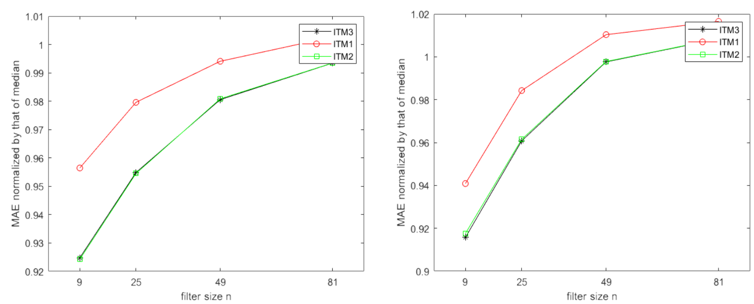
Figure 3.7: Comparison in terms of MAE between the two original ITM filters and the proposed ITM_3 filter for images contaminated with mixed noise (Gaussian and impulsive with $\epsilon = 0.15$). The standard deviation of the Gaussian noise is $\sigma_n = 15$: (a) 'Lena' image, (b) 'Bank' image.

Similar results, but this time for ϵ=0.25, are presented in Figure 3.8 and Figure 3.9. In this case where the impulsiveness of noise has increased, the proposed ITM_3 filter outperformed ITM_1 in all cases, and ITM_2 in almost all cases.



(a) (b) (c)

Figure 3.8: Comparison in terms of MAE between the two original ITM filters and the proposed ITM_3 filter for images contaminated with mixed noise (Gaussian and impulsive with $\epsilon = 0.25$): (a) Constant image (standard deviation of the Gaussian noise is $\sigma_n = 1$), (b) 'Lena' image (standard deviation of the Gaussian noise is $\sigma_n = 10$), (c) 'Bank' image (standard deviation of the Gaussian noise is $\sigma_n = 10$).



(a) (b)

Figure 3.9: Comparison in terms of MAE between the two original ITM filters and the proposed ITM_3 filter for images contaminated with mixed noise (Gaussian and impulsive with $\epsilon = 0.25$). The standard deviation of the Gaussian noise is $\sigma_n = 15$: (a) 'Lena' image, (b) 'Bank' image.

In order to visually assess the performance of the three filters in terms of noise attenuation and detail presernation, the 'Bank' image contaminated with mixed Gaussian and impulsive noise (ϵ=0.25) with Gaussian noise having a standard deviation of 10, as well as the filtered images using ITM_1, ITM_2, and ITM_3 are presented in Figure 3.10.

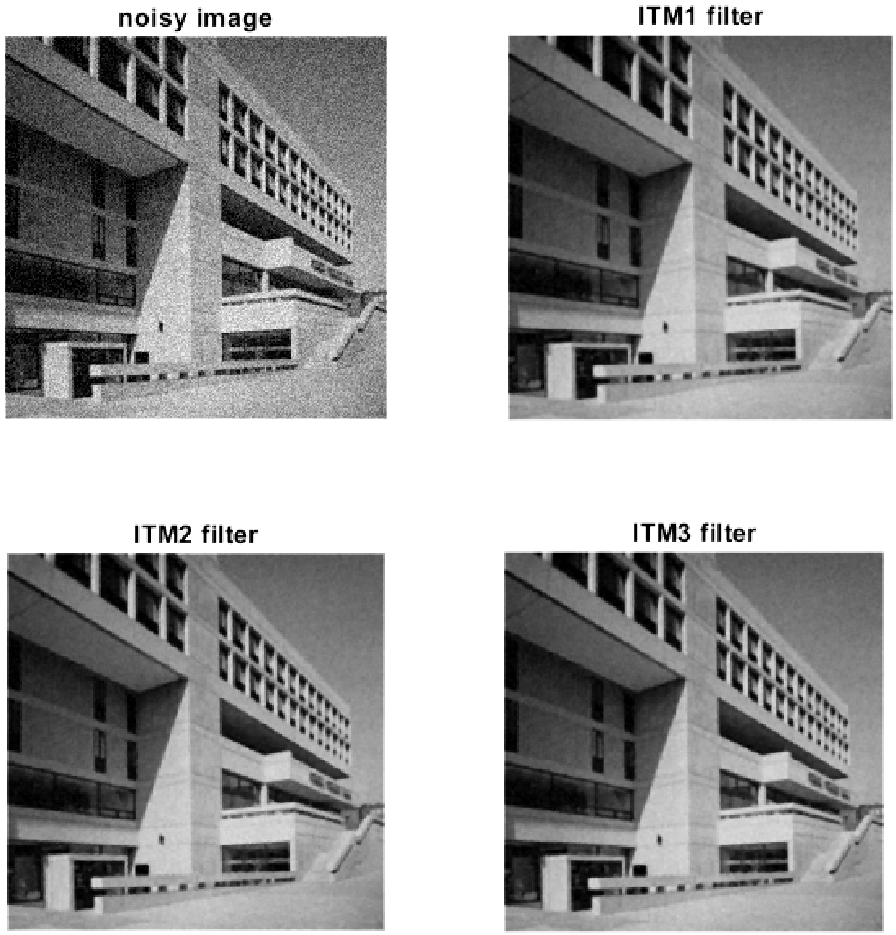
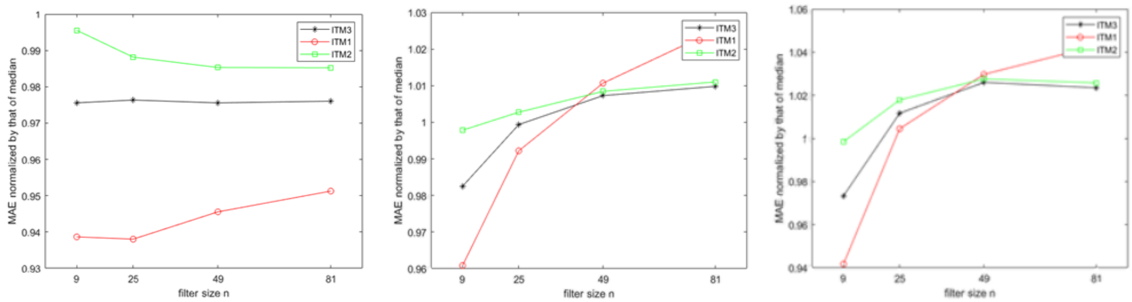


Figure 3.10: Noisy and filtered 'Bank' images of size 512×512 corrupted by mixed noise (Gaussian and impulsive with $\epsilon = 0.25$). The standard deviation of the Gaussian noise is $\sigma_n = 10$.

3.3.4 Gaussian and Laplace Noise

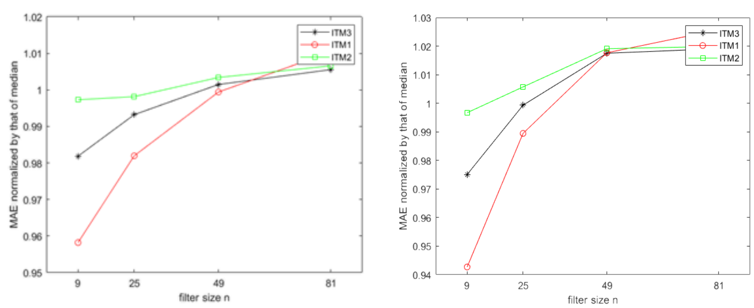
Similarly to the experiments in subsection 2.4.3, for this experiment, the images were contaminated with mixed Gaussian and Laplace noise.

First, mixed noise with ϵ=0.25 (75% Gaussian and 25% Laplace) was used. The results are shown in Figure 3.11 and Figure 3.12 for standard deviation of the Gaussian noise equal to $\sigma_n = 10$ and $\sigma_n = 15$, respectively. The standard deviation of the Laplace noise is $1.3\sigma_n$ in both cases. In this case, it can be observed once again that the ITM_3 filter outperformed the ITM_2 filter for all cases. The ITM_3 filter worked better than the ITM_1 filter for the two real images for larger window sizes.



(a) (b) (c)

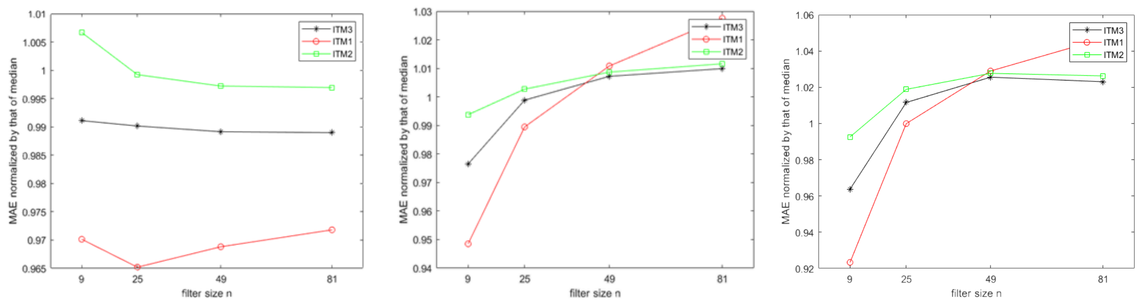
Figure 3.11: Comparison in terms of MAE between the two original ITM filters and the proposed ITM_3 filter for images contaminated with mixed noise (Gaussian and Laplace with $\epsilon = 0.25$). The standard deviation of the Laplace noise is $1.3\sigma_n$, where σ_n is that of the Gaussian noise: (a) Constant image (standard deviation of the Gaussian noise is $\sigma_n = 1$), (b) 'Lena' image (standard deviation of the Gaussian noise is $\sigma_n = 10$), (c) 'Bank' image (standard deviation of the Gaussian noise is $\sigma_n = 10$).



(a) (b)

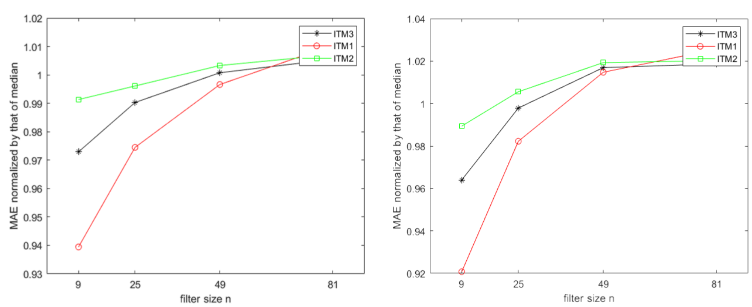
Figure 3.12: Comparison in terms of MAE between the two original ITM filters and the proposed ITM_3 filter for images contaminated with mixed noise (Gaussian and Laplace with $\epsilon = 0.25$). The standard deviation of the Gaussian noise is 15, and of the Laplace noise is $1.3\sigma_n$: (a) 'Lena' image, (b) 'Bank' image.

Similar results for ϵ=0.50 are shown in Figure 3.13 and Figure 3.14. The conclusions are the same as in the $\epsilon = 0.25$ case.



(a) (b) (c)

Figure 3.13: Comparison in terms of MAE between the two original ITM filters and the proposed ITM_3 filter for images contaminated with mixed noise (Gaussian and Laplace with $\epsilon = 0.5$). The standard deviation of the Laplace noise is $1.3\sigma_n$, where σ_n is that of the Gaussian noise: (a) Constant image (standard deviation of the Gaussian noise is $\sigma_n = 1$), (b) 'Lena' image (standard deviation of the Gaussian noise is $\sigma_n = 10$), (c) 'Bank' image (standard deviation of the Gaussian noise is $\sigma_n = 10$).



(a) (b)

Figure 3.14: Comparison in terms of MAE between the two original ITM filters and the proposed ITM_3 filter for images contaminated with mixed noise (Gaussian and Laplace with $\epsilon = 0.5$). The standard deviation of the Gaussian noise is 15, and of the Laplace noise is $1.3\sigma_n$: (a) 'Lena' image, (b) 'Bank' image.

For visually assessing the capability of filters to remove noise while preserving image details, the 'Lena' which was contaminated with mixed Gaussian and Laplace noise (ϵ=0.50), and with the Gaussian noise having a standard deviation of 10 is presented in Figure 3.15.

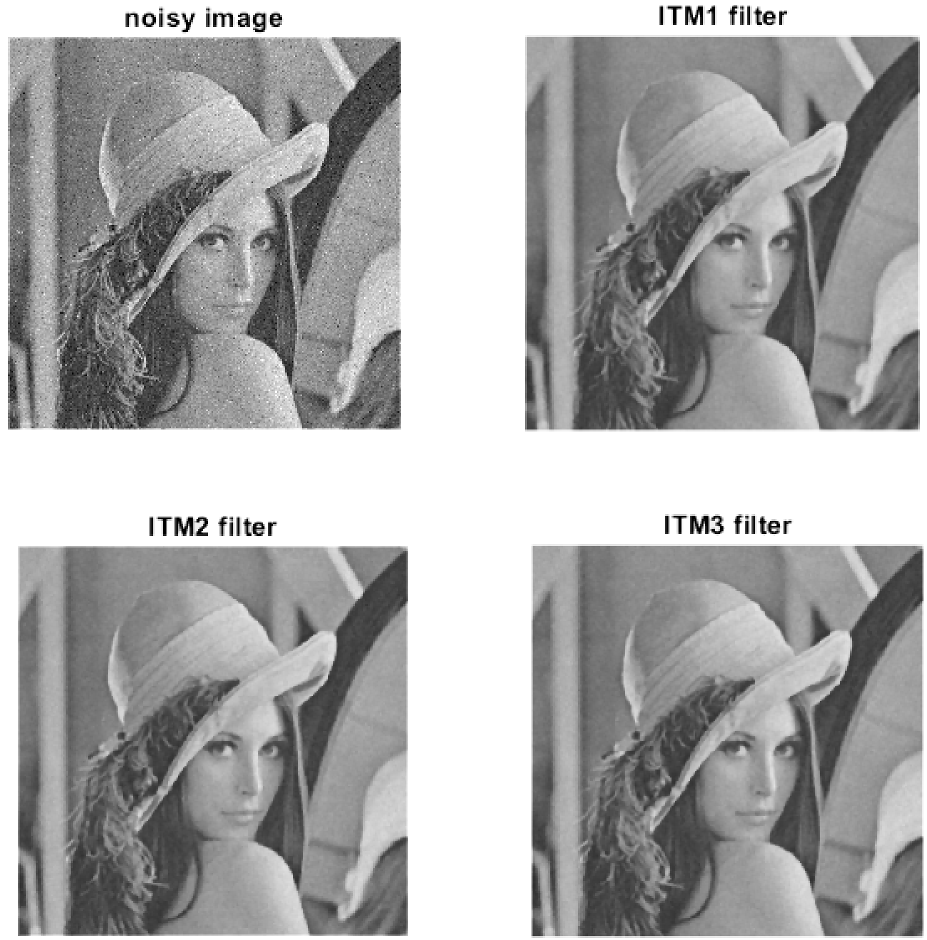


Figure 3.15: Noisy and filtered 'Lena' images of size 512×512 corrupted by mixed noise (Gaussian and Laplace with $\epsilon = 0.5$). The standard deviation of the Gaussian noise is $\sigma_n = 10.$ and of the Laplace noise is $1.3\sigma_n$.

3.4 Summary of Results

The experimental results presented in this section have once again demonstrated that each of the two original ITM filters provides a lower MAE than the other filter, depending on the type of noise which has contaminated the image. In general, the ITM_1 filter provides a lower MAE for Gaussian noise and a mixture of Gaussian and Laplace noise, especially for smaller window sizes. On the other hand, ITM_2 provides a lower MAE for mixed Gaussian and impulsive noise, when the portion of the impulsive noise is somewhat significant (e.g., 25%). The ITM_2 filter also provides a lower MAE when larger filter windows (e.g., 9×9) are used on images with many edges or image details.

In this sense, the two original ITM filters can be considered equivalent, since one is not always better than the other.

Nevertheless, experimental results have also demonstrated that the proposed ITM_3 filter almost always provides a lower MAE than the ITM_2 filter, and can be used to completely replace ITM_2.

It should also be mentioned that in some cases, especially when larger window sizes are used on real images (i.e., images with edges and fine details), the median filter may provide a lower MAE than the ITM filters. This is the case when the normalized MAE values are above 1.

Chapter 4

Conclusion and Future work

4.1 Primary Findings

Previous research [18] had shown that the Iterative Truncated Mean (ITM) filter (and its two versions, ITM_1 and ITM_2) was very promising for removing different types of noise. The purpose of this thesis research was to further improve the ITM algorithm. Advantages and disadvantages of different filters, inluding the mean filter, the median filter, and the ITM filters were also discussed.

Based on the experiments performed for this thesis, it was confirmed that each of the two different versions of the ITM filter works better than the other, in terms of the Mean Absolute Error (MAE), depending on the type of noise. The objective of the proposed modification was to develop an ITM algorithm which provides an improved overall performance than at least one of the two filters.

In this thesis, comparisons were performed between the proposed ITM_3 filter and the existing ITM filters. In summary, the proposed ITM_3 filter showed an improved performance over the ITM_2 filter for almost all cases (different types of noise, different standard deviations, and different images). To be more specific, the proposed ITM_3 filter almost always provided a smaller MAE compared to the ITM_2 filter. Of course, this also implies that the proposed ITM_3 filter provided a smaller MAE than ITM_1 for these cases where ITM_2 provided a smaller MAE than ITM_1.

4.2 Future Work

This thesis was focused on the improvement of the output of the ITM filters. Future work can expand on the following topics:

- An adaptive filter can be developed to estimate the type of noise and to choose between the ITM_1 and ITM_3 filters depending on the type of noise.

- The operation time of the ITM filter can be improved by looking further into the required number of iterations. For a larger window size and image size, the filtering operation is time consuming.
- Advanced adaptive filter algorithms, similar to the one proposed in [11], can be investigated.

Bibliography

[1] A.Flaig, G.R.Arce, and K.E.Barner. Affine Order-Statistic Filters: Medianization of Linear FIR Filters . *IEEE Transactions on Signal Process*, 46(8):pages:2101–2112, August,1998.

[2] T.C. Aysal and K.E. Barner. Generalized Mean-Median Filtering for Robust Frequency-Selective Applications. *IEEE Transactions on Signal Processing*, 55(3):pages:937–948, March,2007.

[3] A. Burian and P. Kuosmanen. Tuning the smoothness of the recursive median filter . *IEEE Transactions on Signal Process*, 50(7):1631–1639, 2002.

[4] Gonzalez Rafael C. *Digital Image Processing*. Pearson Education Indian, 2009.

[5] Dr. Dimitrios Charalampidis. Introduction To Image Processing . *Embedded Multimedia Systems, fs.uno.edu/dchalala/ccli/imageprocessing*.

[6] Nikias. C.L and Shao .M. *Signal processing with Alpha- Stable Distributions and Applications*. John Wiley and Sons Inc, 1995.

[7] Navnish Goel, Dr. A. Yadav, and Dr. B.M. Singh. Medical Image Processing: A Review. *Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity (CIPECH-16)*, (ISSN:978-1-4673-9080):pg no:57–62, 2016.

[8] Rafael C. Gonzalez, Richard E.Woods, and Steven L. Eddins. *Digital Image Processing Using MATLAB*, volume 2. Mc Graw Hill Education, 2010.

[9] Huber and Peter.J. Robust Estimation of a Location Parameter. *Ann.Math.Statist*, 35(1):pages:73–101, 1964.

[10] J.B.Bednar and T.L.Watt. Alpha-Trimmed Means and Their Relationship to Median Filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, VOL.ASSP-32(1):pg no.145–153, February1984.

[11] Houbiao . L, Liangchao Li Song Ye, and Jianya Yang. Adaptive Iterative Truncated Arithmetic Mean Filter in Image Denoising. *IEEE 12th International Conference on Computer and Information Technology*, pages 851–855, 2012.

[12] Zhenwei Miao and Xudong Jiang. Further Properties and a Fast Realization of the Iterative Truncated Arithmetic Mean Filter. *IEEE Trans. on Circuits and Systems*, 59(11):810–814, 2012.

[13] William K. Pratt. *Digital Image Processing*, volume vol:3. John Wiley and Sons.INC., 2000.

[14] G. Scheleyer, C. Cubillos, G. Lefranc, R. O. Comparan, and G. Millan. A new Colour Image Segmentation . *International Conference on Computers Comunications and Control(ICCCC)*, vol:6th(no:978-1-5090-1735):pg:no232–239, 2016.

[15] S.Kortz, T.J. Kozubowski, and K.Podgorski. *The Laplace Distribution and Generalization.* Birkhauser Boston 2001.

[16] S. Suryanarayana, Dr. B. L. Deekshatulu, Dr. K. L. Kishore, and Y. R. Kumar. Estimation and Removal of Gaussian Noise in Digital Images. *International Journal of Electronics and Communication Engineering. Embedded Multimedia Systems*, 5(1):23–33, 2012.

[17] X.D.Jiang. Image detail- preserving filter for impulsive noise attenuation. *Inst. Elect. Eng.-Vis., Image Signal Process*, 150(3):179–185, 2003.

[18] X.D.Jiang. Iterative Truncated Arithmetic Mean Filter and Its Properties. *IEEE Trans. Image Process*, 21(4):pages:1537–1547, April,2012.

[19] Zhengya Xu, Hong Ren Wu, Xinghuo Yu, and Bin Qiu. Colour Image Enhancement by Virtual Histogram Approach . *IEEE Transactions on Consumer Electronics*, vol:56(no:2):pages:704–712, 2010.

[20] Josiane Zerubia. Image Processing For High Resolution Satellite And Aerial Data . *Signal Processing Conference, 11th European journal*, 2002.

Vita

Prathyusha Surampudi Venkata received her Bachelor Degree from Vardhaman College of Engineering, India in 2015. She finished her Masters in Electrical Engineering from the University of New Orleans in August 2018 with a cumulative GPA of 3.87. The author worked with her advisor Dr. Dimitrios Charalampidis for her master's thesis.