International Journal of Multidisciplinary Approach and Studies

ISSN NO:: 2348 – 537X

Analysis and Comparison of Filtering Techniques for Image Restoration

Manjunath Savadatti

M. Tech. Scholar, VTU Belgaum, Karnataka, India

ABSTRACT:

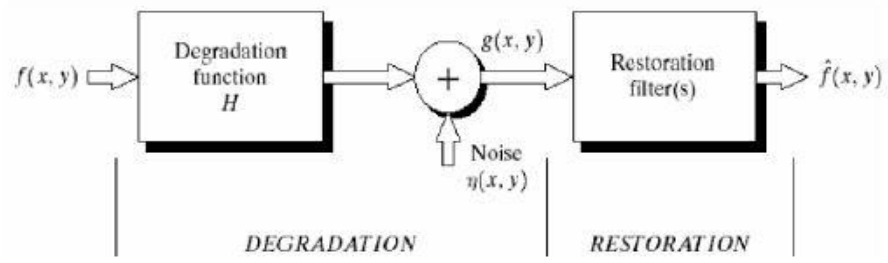
Image restoration is an important issue in high-level image processing. Images are often degraded during the data acquisition process. The degradation may involve blurring, information loss due to sampling, quantization effects, and various sources of noise. The purpose of image restoration is to estimate the original image from the degraded data. It is widely used in various fields of applications, such as medical imaging, astronomical imaging, remote sensing, microscopy imaging, photography deblurring, and forensic science, etc. Often the benefits of improving image quality to the maximum possible extent for outweigh the cost and complexity of the restoration algorithms involved. In this project we are comparing various image restoration techniques like arithmetic mean filter, geometric mean filter, harmonic mean filter, contra harmonic mean filter, midpoint filter, alpha-trimmed mean filter, Median filter, Improved Median filter, Center Weighted filter, Arithmetic Mean on the basis of PSNR (Peak Signal to Noise Ratio).The project also encompasses building a front end GUI in MATLAB.

i. INTRODUCTION

Image Restoration is the process of obtaining the original image from the degraded image given the knowledge of the degrading factors. Digital image restoration is a field of engineering that studies methods used to recover original scene from the degraded images and observations. Techniques used for image restoration are oriented towards modeling the degradations, usually blur and noise and applying various filters to obtain an approximation of the original scene. There are a variety of reasons that could cause degradation of an image and image restoration is one of the key fields in today's Digital Image Processing due to its wide area of applications. Commonly occurring degradations include blurring, motion and noise . Blurring can be caused when object in the image is outside the camera’s depth of field sometime during the exposure, whereas motion blur can be caused when an object moves relative to the camera during an exposure. The purpose of image restoration is to "compensate for" or "undo" defects which degrade an image. Degradation comes in many forms such as motion blur, noise, and camera misfocus. In cases where the image is corrupted by noise, the best we may hope to do is to compensate for the degradation it caused. In this project, we will introduce and implement several of the methods used in the image processing world to restore images.

The field of image restoration (sometimes referred to as image de-blurring or image de-convolution) is concerned with the reconstruction or estimation of the uncorrupted image from a blurred and noisy one. Essentially, it tries to perform an operation on the image that is

the inverse of the imperfections in the image formation system. In the use of image restoration methods, the characteristics of the degrading system and the noise are assumed to be known a priori. In practical situations, however, one may not be able to obtain this information directly from the image formation process. Image restoration algorithms distinguish themselves from image enhancement methods in that they are based on models for the degrading process and for the ideal image. For those cases where a fairly accurate blur model is available, powerful restoration algorithms can be arrived at. Unfortunately, in numerous practical cases of interest, the modeling of the blur is unfeasible, rendering restoration impossible. The limited validity of blur models is often a factor of disappointment, but one should realize that if none of the blur models described in this chapter are applicable, the corrupted image may well be beyond restoration. Therefore, no matter how powerful blur identification and restoration algorithms are the objective when capturing an image undeniably is to avoid the need for restoring the image.
In restoration process, degradation is taken to be a linear spatially invariant operator –



Processing of Image Restoration [1]

$$g(x, y)= h(x, y) * f(x, y) + \eta(x, y) \qquad (1)$$

where, if g(x, y) is noise free, restoration can be done by using the inverse transfer function of h(u, v) as the restoration filter and η(x, y) is the noise

ii. NOISE MODELS

The principal source of noise in digital images arises during image acquisition (digitization) and transmission. The performance of imaging sensors is affected by a variety of factors such as environmental conditions during image acquisition and by the quality of the sensing elements themselves. For instance, in acquiring images with a CCD camera, light levels and sensor temperature are major factors affecting the amount of noise in the resulting image. Images are corrupted during transmission principally due to interference in the channel used for transmission.

a)SALT AND PEPPER NOISE

The PDF of impulse noise is given by

$$p_I(z) = \begin{cases} P_a & for\ z = a \\ P_b & for\ z = b \\ 0 & otherwise \end{cases}$$

If b > a, gray-level b will appear as a light dot in the image. Conversely a level will appear like a dark dot. If either Pa or Pb is zero, the impulse noise is called uni-polar.

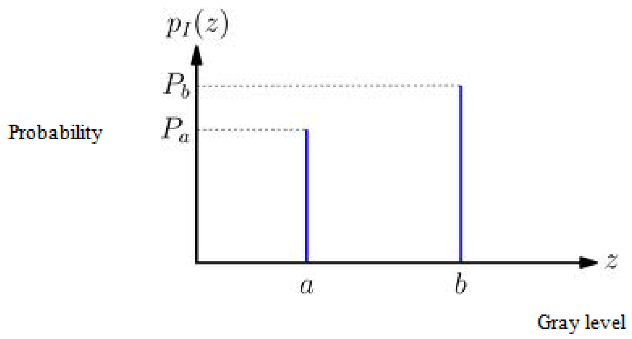


Fig 3.1 PDF of Impulse noise

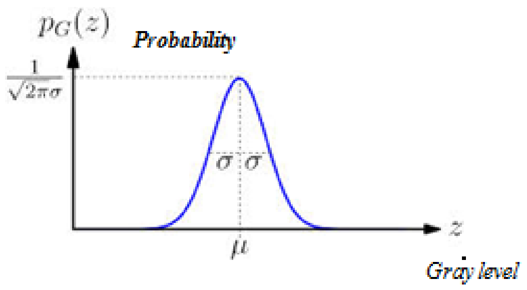
If neither probability is zero and especially if they are approximately equal impulse noise values will resemble salt-and-pepper granules randomly distributed over the image. For this reason bipolar impulse noise is also called *salt-and-pepper noise*. Shot and spike noise terms are also used to refer this type of noise. Noise impulses can be negative or positive. Scaling usually is part of the image digitizing process. Because impulse corruption usually is large compared with the strength of the image signal, impulse noise generally is digitized as extreme (pure white or black) values in an image. Thus the assumption usually is that a and b are “saturated” values in the sense that they are equal to the minimum and maximum allowed values in the digitized image. As a result, negative impulses appear as black (pepper) points in an image. For the same reason, positive impulses appear white (salt) noise. For an 8-bit image this means that a = 0 (black) and b = 255 (white).

b)GAUSSIAN NOISE

The PDF of a Gaussian random variable, z is given by

$$p_G(z) = \frac{1}{2\Pi\sigma} e^{-(z-\mu)^2/2\sigma^2}$$

Where z represents gray level, μ is the mean of average value of z, and σ is its standard deviation. The standard deviation squared, σ^2 is called the variance of z. Because of its mathematical tractability in both the spatial and frequency domains, Gaussian (also called *normal)* noise models are frequently used in practice. In fact, this tractability is so convenient that it often results in Gaussian models being used in situations in which they are marginally applicable at best.



About 70% of all the values fall within the range from one standard deviation (σ) below the mean (μ) to one above, and about 95% fall within two standard deviations.

c) SPECKLE NOISE

Speckle is a granular 'noise' that inherently exists in and degrades the quality of the active radar and synthetic aperture radar (SAR) images. Speckle noise in conventional radar results from random fluctuations in the return signal from an object that is no bigger than a single image processing element. It increases the mean grey level of a local area. One of the methods to eliminate speckle noise includes adaptive and non-adaptive filters on the signal processing (where adaptive filters adapt their weightings across the image to the speckle level, and non-adaptive filters apply the same weightings uniformly across the entire image). Speckle adds multiplicative noise to the image `I`, using the equation

`J = I+n*I`

Where `n` is uniformly distributed random noise with mean 0 and variance `v`. The default for `v` is 0.04.

d) POISSON NOISE

Poisson noise is generated from the data instead of adding artificial noise onto the data. If the image is double precision, then input pixel values are interpreted as means of Poisson distributions scaled up by le12. For example, if an input pixel has the value `5.5e-12`, then the corresponding output pixel will be generated from a Poisson distribution with mean of 5.5 and then scaled back down by `1e12`. If `the image` is single precision, the scale factor used is `1e6`. If `the image` is `uint8` or `uint16`, then input pixel values are used directly without scaling. For example, if a pixel in a `uint8` input has the value 10, then the corresponding output pixel will be generated from a Poisson distribution with mean 10.

iii. FILTERS

Noise elimination is a main concern in computer vision and image processing. A digital filter is used to remove noise from the degraded image. As any noise in the image can be result in serious errors. Noise is an unwanted signal, which is manifested by undesirable information. Thus the image, which gets contaminated by the noise, is the degraded image and using different filters can filter this noise. Thus filter is an important subsystem of any signal processing system. Thus filters are used for image enhancement, as it removes undesirable signal components from the signal of interest. Filters are of different type i.e. linear filters or nonlinear filters. In early times, as the signals handled were analog, filters used are of analog. Gradually digital filters were took over the analog systems because of their flexibility, low cost, programmability, reliability, etc. for these reasons digital filters are designed which works with digital signals. Different filters are discussed in this chapter.

a. MEAN FILTERS

Mean filtering is a simple, intuitive and easy to implement method of *smoothing* images, *i.e.* reducing the amount of intensity variation between one pixel and the next. It is often used to reduce noise in images. The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings. Mean filtering is usually thought of as a convolution filter.

b. ARITHMETIC MEAN FILTER

This is the simplest of the mean filters. Let Sxy represents the set of coordinates in a rectangular sub image window of size m × n centred at point (x, y). The arithmetic mean filtering process computes the average value of the corrupted image g(x, y) in the area defined by Sxy. The value of the restored image f at any point (x, y) is simply the arithmetic mean computed using the pixels in the region defined by Sxy .In other words,

$$f^{\wedge}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

This operation can be implemented using a convolution mask in which all coefficients have value 1/mn. A mean filter simply smoothes local variations in an image. Noise is reduced as a result of blurring.

c. CONTRA-HARMONIC MEAN FILTER:

Contra-Harmonic Mean filter is used to remove Gaussian type noise and preserve edge features and works for only monochrome, 8 bit per pixel and 24 bit per pixel images. The contra-harmonic mean filtering operation yields a restored image based on the expression,

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

Where Q is called the order of the filter. This filter is well suited for reducing or virtually eliminating the effects of salt-and-pepper noise. For positive values of Q, the filter eliminates pepper noise. For negative values of Q it eliminates salt noise. It cannot do both simultaneously. Note that the contra-harmonic filter reduces to the arithmetic mean filter if Q = 0, and to the harmonic mean filter if Q = - 1.

ORDER- STATISTICS FILTERS

Order-statistics filters are spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter. The response of the filter at any point is determined by the ranking result. This type of filtering replaces the value of the center pixel with the value determined by ranking.

d. MEDIAN FILTER

The best known order-statistics filter is the *median filter*, which replaces the value of a pixel by the median of the gray levels in the neighbourhood of that pixel. In other words,

$$f^{\wedge}(x,y) = median_{(s,t)\in S_{xy}}\{g(s,t)\}$$

The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. The original value of the pixel is included in the computation of the median. Median filters are quite popular because, for certain types of random noise they provide excellent noise reduction capabilities, with considerably less blurring than linear smoothing filters of similar size.

e. IMPROVED MEDIAN FILTER

The algorithm for improved median filter is as follows.

Step 1: A two dimensional window (denoted by 3×3 W) of size 3x3 is selected and centered on the processed pixel p(x, y) in the corrupted image.

Step 2: Sort the pixels in the selected window according to the ascending order and find the median pixel value denoted by Pmed), maximum pixel value (Pmax) and minimum pixel value (Pmin) of the sorted vector V0. Now the first and last elements of the vector V0 is the Pmin and Pmax respectively and the middle element of the vector is the Pmed.

Step 3: If the processed pixel is within the range Pmin < P(x, y) < Pmax, Pmin > 0 and Pmax < 255, it is classified as uncorrupted pixel and it is left unchanged. Otherwise p(x, y) is classified as corrupted pixel.

Step 4: If p(x, y) is corrupted pixel, then we have the following two cases:

Case 1: If Pmin < Pmed < Pmax and 0 < Pmed < 255, replace the corrupted pixel p(x, y) with Pmed.

Case 2: If the condition in case 1 is not satisfied then Pmed is a noisy pixel. In this case the pixel value is replaced with value of pixel which is above the center pixel.

Step 5: Step 1 to step 4 are repeated until the processing is completed for the entire image.

f. ADAPTIVE MEDIAN FILTER:

The median filter discussed performs well as long as the spatial density of the impulse noise is not large. Adaptive median filtering can handle impulse noise with probabilities even larger than these. An additional benefit of the adaptive median filter is that it seeks to preserve detail while smoothing non-impulse noise, something that the "standard" median filter does not do. As in all the nonlinear ordered statistics filters in literature, the adaptive median filter also works in a rectangular window area unlike those filters, however, the adaptive median filter, changes (increases) the size of Sxy during filter operation, depending on certain conditions listed in this section. Keep in mind that the output of the filter is a single value used to replace the value of the pixel at (x, y), the particular point on which the window Sxy, is centered at a given time. Consider the following notations [4] and Algorithm,

Zmin = minimum gray level value in Sxy
Zmax = maximum gray level value in Sxy
Zmed = median of gray levels in Sxy
Zxy = gray level at coordinates (x, y)
Smax = maximum allowed size of Sxy

The adaptive median filtering algorithm uses two processing levels, denoted level A and level B:

```
Level A:  A1 = Zmed - Zmin
          A2 = Zmed - Zmax
          If A1 > 0 AND A2 < 0, go to level B
          Else increase the window size
          If window size < Smax, repeat level A
          Else output Zxy
Level B:  B1 = Zxy - Zmin
          B2 = Zxy - Zmax
          If B1 > 0 AND B2 < 0, output Zxy
          Else output Zmed
```

g. CENTERWEIGHTED MEDIAN FILTER:

The median filter is strictly a rank order operator. Thus all temporal locations within the observation window are considered equivalent. That is, given a window of observation samples, any permutation of the samples within the observation window results in an identical median filter output. As stated above, for most signals certain samples within the observation window are more correlated with the desired estimate than are others. Due to the symmetric nature of the observation window, the sample most correlated with the desired estimate is, in general, the centre observation sample.

The centre observation sample can be weighted to reflect its importance, or correlation with the desired estimate. Since median filters select the output in a different fashion than do linear filters, i.e., ranking versus summing, the observation samples must also be weighted differently.

In the median filtering case, weighting is accomplished through repetition. Thus, the output of the CWM filter is given by

$$Y(n) = MED[x_1 \ldots xc_{-1}, x_c \odot w_c, x_{c+1} \ldots\ldots x_N],$$

Where $x_c \odot w_c$ denotes the replication operator and c= (N+1)/2 = N_1+1 is the index of the centre sample.

h. MIDPOINT FILTER:

The midpoint filter simply computes the midpoint between the maximum and minimum values in the area encompassed by the filter:

$$f^{\wedge}(x,y) = \frac{1}{2}\left[max_{(s,t)\in S_{xy}}\{g(s,t)\} + min_{(s,t)\in S_{xy}}\{g(s,t)\}\right]$$

This filter works best for randomly distributed noise like Gaussian noise or uniform noise.

i. ALPHA TRIMMED MEAN FILTER

Alpha-trimmed mean filter is windowed filter of nonlinear class; by its nature is hybrid of the mean and median filters. The basic idea behind filter is for any element of the signal (image) look at its neighborhood, discard the most atypical elements and calculate mean value using the rest of them. Alpha you can see in the name of the filter is indeed parameter responsible for the number of trimmed elements.

Now let us see, how to get alpha-trimmed mean value in practice. The basic idea here is to order elements; discard elements at the beginning and at the end of the got ordered set and then calculate average value using the rest. For instance, let us calculate alpha-trimmed mean for the case, depicted.

Thus, to get an alpha-trimmed mean we are to order elements, eliminate elements at the beginning and at the end of the got sequenced collection and get average of the remaining elements. Suppose we delete the $\alpha/2$ lowest and the $\alpha/2$ highest intensity values of g(s,t) in S_{xy}. Let $g_r(s, t)$ represent the remaining (mn-α) pixels, an alpha trimmed mean filter is thus given by,

$$f^{\wedge}(x,y) = \frac{1}{\mathrm{mn}-\alpha} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$

When α=0, the alpha trimmed mean filter reduces to arithmetic mean filter. If α = (mn-1), tit reduces to median filter.

Alpha-trimmed mean filter algorithm:

1. Place a window over element
2. Pick up elements
3. Order elements
4. Discard elements at the beginning and at the end of the got ordered set
5. Take an average — sum up the remaining elements and divide the sum by their number.

In practice alpha is the number of elements to be discarded, for instance, in our case alpha is two. Since our filter is symmetric one alpha is an even nonnegative number less than size of the filter window. Minimum value for alpha parameter is zero and in this case alpha-trimmed mean filter degenerates into mean filter. Maximum value for alpha is filter window size minus one and in this case filter degenerates into median filter.

PERFORMANCE PARAMETER

Performance parameters are used in evaluating and comparing the different attributes of a system. Different performance parameters in image processing may include calculation of peak signal to noise ratio (PSNR), time required for processing the image according to the noise density and so on. Image restoration or improving the visual quality of a digital image can be objective. Saying that one method provides a better quality image could vary from person to person. For this reason, it is necessary to establish quantitative/empirical measures to compare the effects of image restoration algorithms on image quality. Using the same set of tests images, different image restoration algorithms can be compared systematically to identify whether a particular algorithm produces better results. If we can show that an algorithm or set of algorithms can enhance a degraded known image to more closely resemble the original, then we can more accurately conclude that it is a better algorithm.

PEAK SIGNAL-TO-NOISE RATIO (PSNR)

In our project, the performances of different filters implemented are compared using the performance parameter –PSNR. Peak signal-to-noise ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

For the following implementation, let us assume we are dealing with a standard 2D array of data or matrix. The dimensions of the correct image matrix and the dimensions of the degraded image matrix must be identical. The mathematical representation of the PSNR is as follows:

$$PSNR = 20\log_{10}\left(\frac{MAX_f}{\sqrt{MSE}}\right)$$

Where the MSE (Mean Squared Error) is:

$$MSE = \frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1}\|f(i,j)-g(i,j)\|^2$$

Legend:

f represents the matrix data of our original image.g represents the matrix data of our degraded image in question.m represents the numbers of rows of pixels of the images and i represents the index of that row.

n represents the number of columns of pixels of the image and j represents the index of that column.MAX_f is the maximum signal value that exists in our original “known to be good” image.

DESIGN AND IMPLEMENTATION

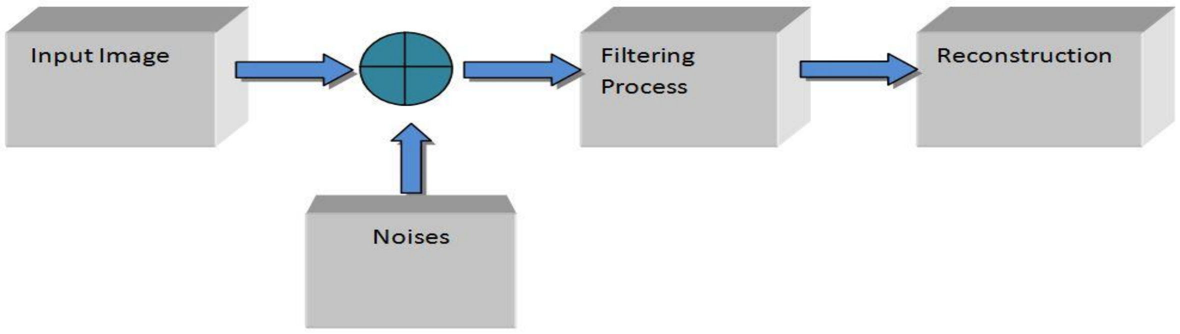
OVERVIEW

The system consists of a computer system, a file for storing images and the application user. The application user selects the image and asks the user to add any of the four noises considered during the implementation. Thus the degraded image is obtained. Now, to restore this degraded image the user can select the corresponding filtering technique. The corresponding performance parameter PSNR and time required for filtering are displayed on the screen.

MODEL

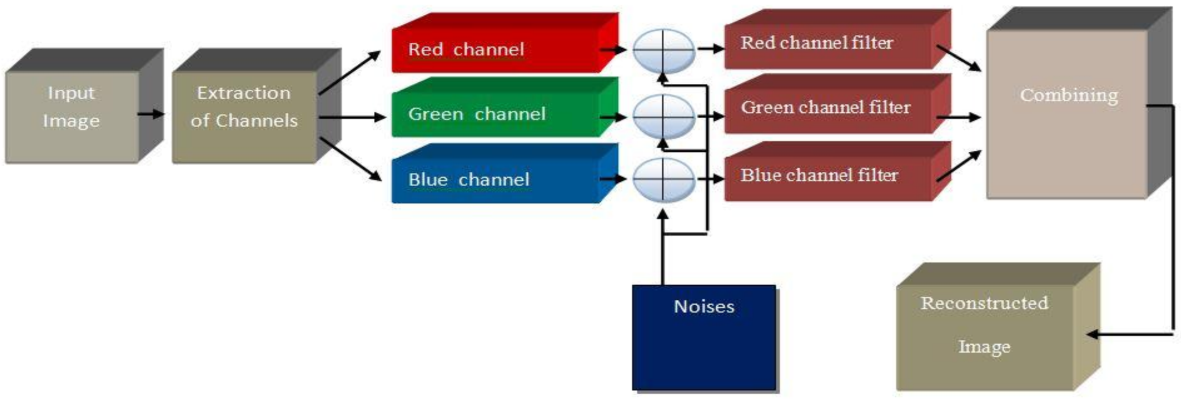
Our project model consists of two sections: 1. Restoration of gray-scale images 2. Restoration of color images.

For the gray-scale images, the model is as shown in figure 6.1, an image is selected (example: hill.jpg) from the system and any one of the four noises considered (Salt and pepper, Gaussian, Speckle, Poisson) is added to the image. The addition process is represented by an adder. Once the degraded image is obtained, we can use different filtering algorithms specified to reconstruct the image.



Project model for gray-scale images

Thus the efficiency of different filters can be compared and the best suited filter for particular type of noise is determined.
In case of the color images, an RGB image is selected from the system which is further divided into three gray scale images of Red, Green and Blue channels. The model is shown is fig 6.2. Now, a particular noise is added to each of these gray scale images and the filtering techniques specified can be used to obtain the restored image.



Project model for color images

The performance of each filtering technique is compared using performance parameter PSNR. The PSNR value for colour images can be calculated as average of PSNR value for each individual channel.

RESULTS

The median filter, improved median filter, midpoint filter, center weighted median filter, contra-harmonic mean filter, alpha trimmed mean filter, arithmetic mean filter and adaptive median filter are implemented in MATLAB Version 7.7.0.471 (R2008b). A variety of images are used to test the algorithms and study the performance of filters for restoration.

Input images for the algorithm are obtained by taking original clean images and intentionally adding noise to them. These degraded images are then reconstructed. The quantitative valuation is performed by calculating peak signal to noise ratio (PSNR).
reference images are

a. FOR SALT AND PEPPER NOISE

The PSNR values and the processing time for different filters at different noise densities of salt and pepper noise are tabulated in the following section.

PSNR VALUES

For gray scale images, Table 7.1 gives the PSNR values for the following filters. For analysis purpose, noise densities ranging from 1 % to 9% are considered and the performances of different filters are noted.

Filter	Noise density(Salt & Pepper Noise)								
	1%	2%	3%	4%	5%	6%	7%	8%	9%
Improved median	37.41	37.02	36.81	36.43	36.11	35.84	35.58	35.58	35.17
Alpha trimmed mean	33.40	33.36	33.27	33.23	33.08	35.95	32.88	32.69	32.49
Median	34.86	34.85	34.80	34.77	34.75	34.53	34.72	34.69	34.65
Arithmetic Mean	32.82	32.57	32.17	32.02	31.75	31.45	31.42	31.21	30.91
Adaptive median	38.74	38.70	38.00	37.82	37.80	36.91	36.72	36.2	35.85
Center weighted median	34.63	34.63	34.6	34.59	34.41	32.32	34.27	34.1	33.9

Table 7.1 PSNR at different noise density for 'House' image

The PSNR values are expressed in decibels. Greater the PSNR value, more efficient is the filter to remove noise. From the Table 7.1, we observe that with the increase in the noise density, the PSNR intensity decreases. Considering the noise level of 5% as a reference, we can see that the Adaptive Median filter is best suited to remove salt and pepper noise.

For color images, the following filters are considered as shown in table 7.2.

Filter	Noise density(Salt & Pepper Noise)								
	1%	2%	3%	4%	5%	6%	7%	8%	9%
Improved median	41.59	41.56	41.51	41.39	41.36	41.31	41.27	41.21	41.19
Centre Weighted Median	25.06	25.06	25.00	25.00	24.97	24.95	24.90	24.86	24.80
Median	52.17	51.61	50.96	50.56	49.94	49.84	49.25	49.06	48.63

Table 7.2 PSNR at different noise density for 'Flower' image

By observing the table, we can say that Median Filter works better for a constant level of salt and pepper noise.

PROCESSING TIME :

The processing time for restoration by different filters differs from one processor to another. Using Intel®Core™2 Duo CPU operating at 2GHz frequency with 32-bit Operating System the following table 7.3 specifies the different processing times for various filters.

As we can see from the Table, the processing times vary randomly since the filters implemented use a MATLAB function 'rand'. This function generates different random numbers at each run.

Filter	Processing Time (Salt & Pepper Noise) in Sec								
	1%	2%	3%	4%	5%	6%	7%	8%	9%
Improved median	1.5814	2.0136	2.2250	2.7792	1.9679	1.8938	2.2067	2.0734	1.5658
Alpha trimmed mean	0.2282	0.2476	0.2693	0.2651	0.3035	0.2493	0.4392	0.9507	0.2781
Median	0.1467	0.1156	0.1440	0.1602	0.0694	0.1072	0.1475	0.0972	0.2043

Arithmetic Mean	0.2236	0.1564	0.1132	0.1240	30.2264	0.1645	0.1702	0.2706	0.1847
Adaptive median	0.4215	0.2715	0.2116	0.2730	0.4125	0.2615	0.4132	0.3879	0.4208
Center weighted median	4.1940	11.328	11.243	9.6875	10.797	9.5510	11.6141	10.741	11.321

Table 7.3 Processing Time at different noise densities for 'House' Image

For color images, the table 7.4 gives the processing time for three filters with different noise densities varying from 1% to 9%.

(color) Filter	Processing Time (Salt & Pepper Noise) in Sec								
	1%	2%	3%	4%	5%	6%	7%	8%	9%
Improved median	7.494	9.175	10.172	11.224	8.737	12.122	12.128	10.376	9.957
Median	0.582	0.247	0.261	0.253	0.228	0.251	0.252	0.252	0.249
Center weighted median	31.192	31.264	31.035	21.361	26.120	31.230	30.944	33.419	31.800

Table 7.4 Processing Time at different noise densities for 'Flower' Image

b. FOR GAUSSIAN NOISE

The PSNR and the processing times of the filters for different noise densities of Gaussian noise are tabulated as follows.

PSNR VALUES

Table 7.7 gives the PSNR values of the gray scale images for the noise densities ranging from 1% to 9% and the performances of filters are compared.

Filter	Noise density(Gaussian Noise)								
	1%	2%	3%	4%	5%	6%	7%	8%	9%
Alpha trimmed mean	30.65	29.71	28.85	28.27	27.58	27.03	26.56	26.18	25.89

Mid point	30.23	29.55	28.86	28.34	27.79	27.35	26.92	26.61	26.34
Arithmetic Mean	30.73	29.82	28.95	28.39	27.69	27.14	26.67	26.29	26.00
Center weighted median	30.37	29.46	28.57	28.87	27.24	26.76	26.34	26.02	25.70
Contra harmonic	28.11	27.53	27.02	26.71	26.33	26.02	25.78	25.59	25.44

Table 7.7 PSNR for gray scale images at different noise densities for 'cat' image

From the PSNR values example, considering the 5% noise density the Midpoint Filter is efficient for removing Gaussian noise.

In case of color images, the results are obtained for center weighted median filter as shown in table 7.8

Filter	Noise density(Gaussian Noise)								
	1%	2%	3%	4%	5%	6%	7%	8%	9%
Center weighted median	18.60	17.35	16.24	15.33	14.58	13.99	13.50	13.16	12.88

Table 7.8 PSNR for color image at different noise densities for 'House' image

PROCESSING TIME

The processing time for the removal of Gaussian noise in gray scale images is encapsulated in Table 7.9 Because of the MATLAB function 'rand' used, the processing times are not in co-relation.

Filter	Processing Time (Gaussian Noise) in Sec								
	1%	2%	3%	4%	5%	6%	7%	8%	9%
Alpha trimmed mean	0.078	0.065	0.055	0.067	0.085	0.056	0.115	0.071	0.089
Midpoint filter	0.119	0.167	0.135	0.111	0.110	0.169	0.182	0.156	0.085

Contra-harmonic mean	0.156	0.359	0.189	0.238	0.255	0.172	0.165	0.152	0.146
Arithmetic mean	0.029	0.026	0.022	0.027	0.028	0.029	0.030	0.045	0.041
Center weighted median	10.355	11.395	10.504	11.606	11.557	11.454	11.193	11.073	10.977

Table 7.9 Processing time at different noise densities for 'cat' Image

For color images the processing time of the center weighted median filter is shown in table 7.10

Filter	Processing Time (Gaussian Noise) in Sec								
	1%	2%	3%	4%	5%	6%	7%	8%	9%
Center weighted median	33.088	25.534	31.727	32.843	28.513	33.323	30.723	33.886	36.164

Table 7.10 Processing time at different noise density for 'House' image

c. POISSON NOISE
The restoration of the gray scale and color images with the Poisson noise is evaluated through PSNR and the processing time is calculated for each filter.

PSNR VALUES
The Poisson noise has no parameter to vary the noise density hence for its default value the PSNR values are tabulated in the table 7.12

Filters	Noise Density(Poisson)
Arithmetic mean	34.15
Improved Median	33.79
Center weighted Median	33.14
Midpoint	32.01
Contra harmonic	31.26

Table 7.12 PSNR values for 'Road' image

From the analysis of the above table, the Arithmetic mean filter is best suited for removal for Poisson noise.

The table 7.13 gives the PSNR values for the color images with Poisson noise.

Filters	Noise Density(Poisson)
Improved Median	40.79
Center weighted Median	24.19

Table 7.13 PSNR values for 'Road' image

From the filters taken into consideration for the removal of Poisson noise in color images, the Median filter is best suited.

PROCESSING TIME

The processing time for removal of Poisson noise is given in Table 7.14 for gray scale images.

Filters	Processing Time(Poisson) in sec
Arithmetic Mean	0.3681
Improved Median	0.9419
Center weighted median	2.1752
Midpoint	0.2819
Contra harmonic	0.2026

Table 7.14 Processing time for gray scale images of 'Road' image

The table 7.15 gives the processing time of the filters for color images. Here three filters have been implemented and compared.

Filters	Processing time (Poisson) in sec
Improved Median	3.4799
Center weighted median	6.0975

Table 7.15 Processing time for color images of 'Road' image

Original image Noisy image (poisson)

d. FOR SPECKLE NOISE

The PSNR and the processing times of the filters for the restoration of the degraded images is tabulated in the following sections.

PSNR VALUES

The speckle noise is multiplicative in nature and the PSNR values for the gray scale images are tabulated in Table 7.18

Filter	Noise density(Speckle Noise)								
	1%	2%	3%	4%	5%	6%	7%	8%	9%
Alpha trimmed mean	32.02	30.92	30.29	30.08	29.99	29.74	29.70	29.62	29.58
Mid point	32.02	31.42	31.35	31.24	31.20	31.17	31.12	31.09	31.03
Arithmetic Mean	32.20	31.18	30.61	30.42	30.39	30.17	30.20	30.19	30.19
Contra harmonic	30.18	28.63	27.71	27.19	26.89	26.59	26.43	26.21	26.06

Table 7.18 PSNR at different noise density for Speckle noise

Here, we observe that for a constant level of noise the Midpoint filter works better compared to the other filters taken into consideration.

PROCESSING TIME

The time required for processing the filter depends on the system hence the values are not in co-relation as seen in table 7.19 for the gray scale images.

Filter	Processing Time (Speckle Noise) in Sec								
	1%	2%	3%	4%	5%	6%	7%	8%	9%
Alpha trimmed mean	0.392	0.393	0.307	0.398	0.376	0.375	0.385	0.382	0.388
Midpoint filter	0.162	0.162	0.160	0.152	0.102	0.160	0.168	0.172	0.174
Contra-harmonic mean	0.204	0.218	0.229	0.333	0.343	0.210	0.295	0.199	0.240
Arithmetic mean	0.029	0.026	0.022	0.027	0.028	0.029	0.030	0.045	0.041

vii. CONCLUSION

In the present work different image denoising filters and the merits and demerits of all those filters is discussed along with different noise models such as Gaussian noise, salt and pepper noise, speckle noise and Poisson noise. Depending on the noise present in an image a particular algorithm is to be selected. To summarize, it is concluded that filters are an important engineering technique which help to produce a good quality image by removing noises from an image which are bound to creep into an image because of various reasons which may be controllable or sometimes uncontrollable. The filtering approach to be adopted for a better result depends upon the kind of noise which exists in an image. For salt and pepper noise Adaptive median filter works better when gray scale images are considered and median filter for color images. For Gaussian noise Arithmetic Mean filter and Midpoint filter are best suited. The Arithmetic mean filter is best suited for removal for Poisson noise for gray scale image and for color image Median filter. According to the calculations we have made for different types of filters for Speckle Noise the Midpoint filter works better compared to the other filters taken into consideration.

viii. FUTURE SCOPE

In the future, various techniques can be considered to incorporate in this scheme to further improve the performance and preserve the edges in both highly and lowly corrupted images. If the features of the denoised signal are fed into a neural network pattern recognizer, then the rate of successful classification should determine the ultimate measure by which to compare various denoising procedures. Besides, the complexity of the algorithms can be measured according to the CPU computing time flops. This can produce a time complexity standard for each algorithm. These two points can be considered as an extension to the present work done.

REFERENCES:

i. Charu Khare and Kapil Kumar Nagwanshi, "*Image Restoration Technique with Non Linear filter",* International Journal of Advanced Science and Technology, Vol.39, February 2012.

ii. Gajanand Gupta, *"Algorithm for Image processing using Improved Median filter and comparison of Mean, Median and Improved Median Filter",* International Journal of Soft Computing and Engineering (IJSCE), vol 1, Issue 5, November 2011.

iii. Mamta Juneja and Rajni Mohana, "*An improved adaptive median filtering method for impulse noise detection",* International Journal of Recent Trends in Engineering, vol 1, May 2009.

iv. R. C. Gonzalez and R. E. Woods, *"Digital Image Processing",* 2nd edition, Addison-Wesely, 2004.

v. Max Mignotte, "*Fusion of Regularization Terms for Image Restoration"*

vi. Tong Sun, Moncef Gabbout and Yrjo Neuvo, "*Analysis of two-dimensional center weighted median filters"*

vii. A. K. Jain, *"Fundamentals of Digital Image Processing",* Engelwood Chiff, NJ, Prentice Hall, 2006.

viii. www.ni.com/white-paper/13306/en/

ix. Vipula Singh, "*Digital Image Processing with MATLAB and Labview",* Elsevier India, First Edition, 2006

x. Rudra Pratap, "*Getting started with MATLAB",2002 Edition.*

xi. Nieminen, A. P. Heinonen and Y. Neuvo, *"A new class of detail preserving filters for Image processing",* IEEE Trans. Pattern Anal. Mach. Intell.,9, 1987,pp. 98-106

xii. I. Pitas and A. N. Venetsanopoulos, "*Order statistics in Digital Image Processing",* Proc. IEEE, vol. 80, no. 12, pp. 1893-1921, Dec. 1992.