ANNALS OF FACULTY ENGINEERING HUNEDOARA
– INTERNATIONAL JOURNAL OF ENGINEERING
Tome X (Year 2012) – FASCICULE 1 (ISSN 1584 – 2665)

[1.] Nagamani MODALAVALASA, [2.] K. Satya PRASAD, [3.] S. Swapna RANI,
[4.] G. Sasi Bhushana RAO, [5.] Rajkumar GOSWAMI

A QUANTITIVE EVALUATION OF VARIOUS SPATIAL FILTERS FOR UNDERWATER SONAR IMAGES DENOISING APPLICATION

[1.] DEPT.OF ECE, SBTET, ANDHRA PRADESH, INDIA
[2.] DEPT. OF ECE, JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA, KAKINADA, INDIA
[3-5.] DEPT. OF ECE, ANDHRA UNIVERSITY, VISAKHAPATNAM, INDIA

ABSTRACT: *Image denoising is a key issue in all image processing researches. The great challenge of image denoising is how to preserve the edges and all fine details of an image when reducing the noise. In this paper, a comparative study of image denoising techniques for underwater SONAR (Sound Navigation and Ranging) images relying on spatial filters is presented. In particular four types of spatial filters (Average, Gaussian, Laplacian of Gaussian and Median filters) are applied to judge the efficiency. On each image, different window size configurations starting from 3x3 to 29x29 are applied and the performances of image filtering techniques are analyzed by the estimation of parametric values such as Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR) and execution time for filtering the images. It is observed that by increasing the window size the execution time will increase and PSNR values will decrease. With this analysis and from the results it is found that the optimum filter is Gaussian and the optimum window size is 3×3 for the underwater SONAR images which gives the best execution time, MSE and the PSNR value (37.87dB).*
KEYWORDS: *Denoising, SONAR, Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR)*

INTRODUCTION

The term spatial domain refers to the image plane itself, and the methods in this category are based on direct manipulation of pixels in an image. In this paper the main focus is on two important categories in spatial filtering i.e. linear and nonlinear spatial filtering. The spatial filtering is also called as neighbourhood processing. An image can be modified by applying a particular function to each pixel value. The neighborhood processing may be considered as an extension of this, where a function is applied to a neighborhood of each pixel. The idea is to move a "mask (kernel)": a rectangle (usually with sides of odd length) or other shape over the given image. Depending on the computations performed on the pixels of neighborhoods the operations is called as linear or nonlinear spatial filtering and are clearly described in this paper. Though various filters are already available in the open literature, here this paper focuses on finding out an optimum filtering technique for underwater SONAR images. For this purpose, several spatial filtering techniques such as Average, Gaussian, Laplacian of Gaussian and Median filters are analyzed by the estimation of parametric values such as Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR) and execution time for filtering the images.

SPATIAL FILTERS

Linear spatial filtering: *If the function by which the new gray value is calculated is a linear function of all the grey values in the mask, then the filter is called linear filter. Example of a linear filter is average filter. A linear filter can be implemented by convolving the mask with the input image. For a 3x5 mask, the convolution is shown in Eq. 1:*

$$\sum_{s=-1}^{1}\sum_{t=-2}^{2} m(s,t)p(i+s,j+t) \quad (1)$$

where m(s,t) is the mask coefficients; p(i,j) is the image coefficients.

The spatial thus requires three steps:

1. *position the mask over the current pixel,*
2. *form all the products of the filter elements with the corresponding elements of neighbourhood,*
3. *Sum up all the products, now this is the output pixel with which the current pixel is to be replaced.*

This must be repeated for every pixel in the image.

Spatial filtering is nothing but spatial convolution. The method for performing a convolution is the same as filtering, except that the filter must be rotated by 180^0 before multiplying and adding. Using *the m(s,t) and p(i,j) notation as before, the output of a convolution with a mask for a 3x5 mask for a single pixel is*

$$\sum_{s=-1}^{1}\sum_{t=-2}^{2} m(-s,-t)p(i+s,j+t) \quad (2)$$

where, m(s,t) is the mask coefficients; p(i,j) is the image pixel values

The same result can be achieved by rotating the image pixels by 180° (Eq. 3), this does not affect the result obtained from Eq. (2).

$$\sum_{s=-1}^{1}\sum_{t=-2}^{2} m(s,t)p(i-s,j-t) \quad (3)$$

Non linear filters: *The non linear filters are the filters whose response is based on the ordering the pixels contained in an image neighborhood and then replacing the value of the center pixel in the neighborhood with the value determined by the ranking result. Simple examples are the maximum filter, which has as its output the maximum value under the mask, the corresponding minimum filter, which has as its output the minimum value under the mask, and the median filter, which has the output the median value under the mask.*

AVERAGE FILTER

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. For example, linear filtering for a 3x3 mask is, taking the average of all nine values within the mask. This value becomes the grey value of the corresponding pixel in the new image. This operation is described in the Figure 1.

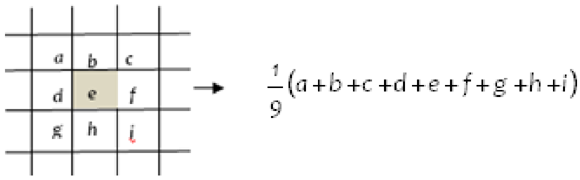

Figure 1. Operation of averaging filter

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. Like other convolutions it is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean. Often a 3×3 square kernel is used, although larger kernels (e.g. 5×5 squares) can be used for more severe smoothing. (Note that a small kernel can be applied more than once in order to produce a similar but not identical effect as a single pass with a large kernel.) Computing the straightforward convolution of an image with this kernel carries out the mean filtering process.

GAUSSIAN FILTERING

The Gaussian smoothing operator is a 2-D convolution operator that is used to `blur' images and remove noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian (`bell-shaped') hump. This kernel has some special properties which are detailed below. The 1-D Gaussian filter has an impulse response given by

$$G(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}} \quad (4)$$

For the 2-D Gaussian filter, it is the product of two such 1-D Gaussians. And is given by

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5)$$

where, x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

The idea of Gaussian smoothing is to use this 2-D distribution as a `point-spread' function, and this is achieved by convolution. Since the image is stored as a collection of discrete pixels we need to produce a discrete approximation to the Gaussian function before we can perform the convolution. In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel, but in practice it is effectively zero more than about three standard deviations from the mean, and so we can truncate the kernel at this point. Once a suitable kernel has been calculated, then the Gaussian smoothing can be performed using standard convolution methods. The convolution *can in fact be performed fairly quickly since the equation for the 2-D isotropic Gaussian shown in Eq. 5 is separable into x and y components.*

$$G_{2D}(x,y)=\left(\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}\right)\times\left(\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{y^2}{2\sigma^2}}\right)=G_{1D}(x)\times G_{1D}(y) \quad (6)$$

Thus the 2-D convolution can be performed by first convolving with a 1-D Gaussian in the x direction, and then convolving with another 1-D Gaussian in the y direction. The Gaussian filter not only has utility in engineering applications. It is also attracting attention from computational biologists because it has been attributed with some amount of biological plausibility, e.g. some cells in the visual pathways of the brain often have an approximately Gaussian response.

LAPLACIAN OF GAUSSIAN

The laplacian is a 2-D isotropic measure of the 2nd order spatial derivative of an image. The laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian smoothing filter in order to reduce its sensitivity to noise, and hence two variants will be described together here. The operator normally takes a single gray level image as input and produces another gray level image as output. The laplacian of an image f(x, y), denoted $\nabla^2 f(x,y)$, is defined as

$$\nabla^2 f(x,y)=\frac{\partial^2 f(x,y)}{\partial x^2}+\frac{\partial^2 f(x,y)}{\partial y^2} \quad (7)$$

Commonly used digital approximations of the second order derivatives are

$$\frac{\partial^2 f}{\partial x^2}=f(x+1,y)+f(x-1,y)-2f(x,y) \quad (8)$$

and

$$\frac{\partial^2 f}{\partial y^2}=f(x,y+1)+f(x,y-1)-2f(x,y) \quad (9)$$

So that

$$\nabla^2 f=[f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)]-4f(x,y) \quad (10)$$

This expression can be implemented at all points (x, y) in an image by convolving the image with the following mask:

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

An alternate definition of the digital second derivatives takes into account diagonal elements and can be implemented using the mask:

$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Both derivatives sometimes are defined with the signs opposite to those shown here, resulting in masks that are the negatives of the preceding two masks. Enhancement using the Laplacian is based on the equation

$$g(x,y)=f(x,y)+c[\nabla^2 f(x,y)] \quad (11)$$

where, f(x, y) is the input image, g(x, y) is the enhanced image, and c is 1 if the center coefficient of the mask is positive or -1 if it is negative. Because the Laplacian is a derivative operator, it sharpens the image but drives constant areas to zero. Adding the original image back restores the gray level image.

MEDIAN FILTER

The median filter is normally used to reduce noise in an image, somewhat like the mean filter. However, it often does a better job than the mean filter of preserving useful detail in the image. Like the mean filter, the median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. 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. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.)

RESULTS AND ANALYSIS

The sample real time sonar images are considered for filtering analysis. For each filter by varying the window size, its PSNR, execution time for computational complexity estimation and MSE are computed. The mathematics relevant to PSNR and MSE calculations are given below.

The peak signal-to-noise ratio (PSNR) is an important parameter to measure the quality of the reconstructed images. The PSNR is the ratio between the image's maximum intensity and the mean square error of that image. The PSNR for the resultant image of size m×n is calculated by

$$PSNR = 20\log_{10}{\frac{MAX_I}{\sqrt{MSE}}} \quad (12)$$

where, MSE is the :

$$\text{Mean Square Error} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2 \quad (13)$$

I (i, j) and K (i, j) represent the input image and its corresponding resultant image accordingly. Figure 2 represents the original sonar image as input image. Figure 3 to Figure 6 represents the corresponding output images of Average, Gaussian, LoG and Median filters respectively.

Figure 7 to Figure 10 represents the PSNR values of all the filters by varying the window size from 3×3 to 29×29 and are compared for analysis which performs better for underwater SONAR images. Here it is observed that, as the window size increases, the PSNR values are decreases. From the Table 1, it is observed that the maximum PSNR and minimum MSE is for Gaussian filter. So the Gaussian filter is best suits for underwater SONAR images.

Table 1: Comparison of PSNR, Execution time and MSE of 3x3 window size

No	Name of the filter	Maximum PSNR value (dB)	Execution time (sec)	MSE
1	Average filter	23.16	0.28696	17.7165
2	Gaussian filter	37.87	0.119315	3.2581
3	Laplacian of Gaussian filter	23.18	0.586001	17.6892
4	Median filter	28.94	0.1313	9.1157

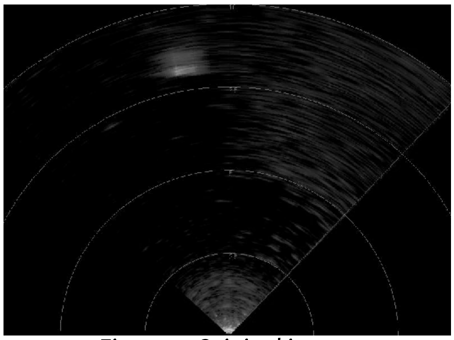

Figure 2. Original image

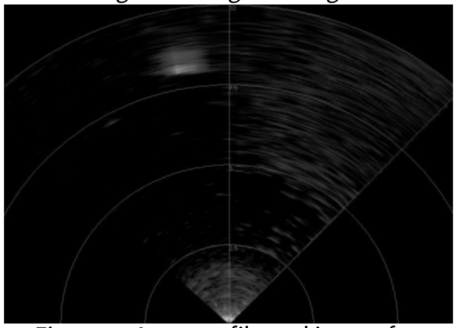

Figure 3. Average filtered image for window size 3x3

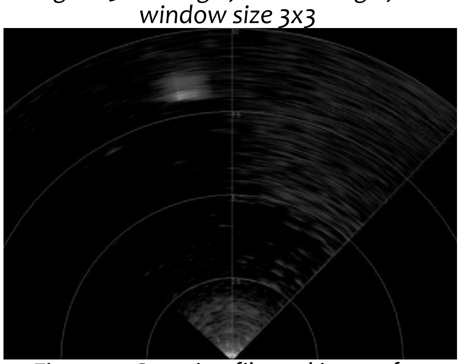

Figure 4. Gaussian filtered image for window size 3x3

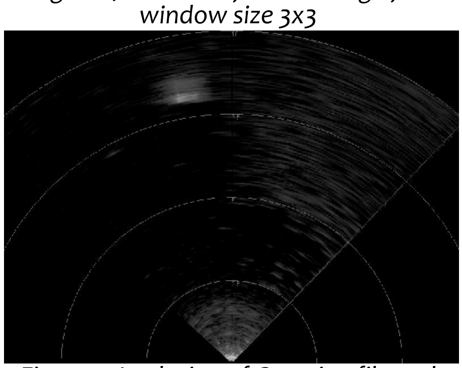

Figure 5. Laplacian of Gaussian filtered image for window size 3x3

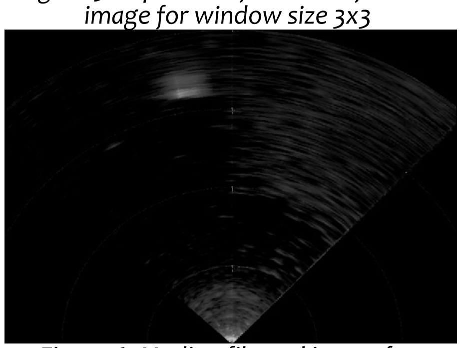

Figure 6. Median filtered image for window size 3x3

CONCLUSIONS

In this paper, the performance analysis of various spatial filters is carried out to find out the optimum filter that suits the underwater SONAR images denoising application. The analysis is carried out on the real time Sector Scan SONAR images. On each image, different window size configurations starting from 3 x 3 to 29x29 are applied and the MSE, execution time and the PSNR values are computed for various denoising filters. As the window size increases the execution time for filtering the image is increasing and PSNR value is decreasing. From the results it is found that the optimum window size configuration and the optimum filter that gives the best execution time, MSE and the PSNR value (37.87dB) are 3×3 window configuration and Gaussian filter respectively.

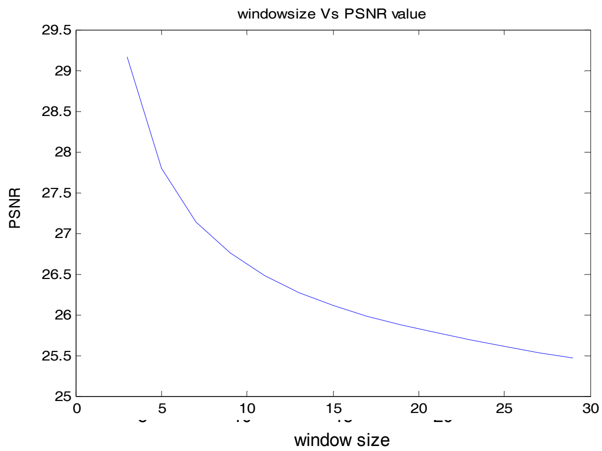

Figure 7. Graph between window size and PSNR value of average filter. The max PSNR is 29.6dB at window size 3

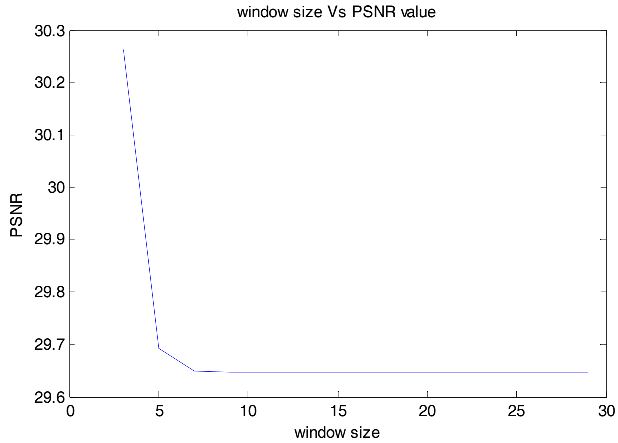

Figure 8. Graph between window size and PSNR value of Gaussian filter. The max PSNR is 30.26 dB at window size 3

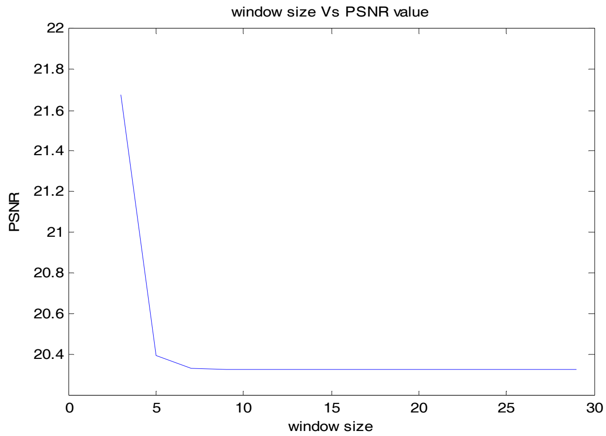

Figure 9. Graph between window size and PSNR value of LoG filter. The max PSNR is 21.6 dB at window size 3

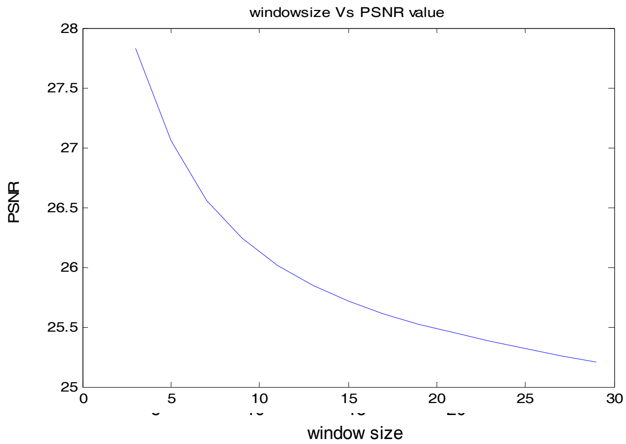

Figure 10. Graph between window size and PSNR value of Median filter. The max PSNR is 27.82 dB at window size 3

ACKNOWLEDGMENTS

The above work has been undertaken towards the research project of DRDO (NSTL), Visakhapatnam, Andhra Pradesh, INDIA, No: NSTL/NA/9162/AUV/Consultancy/4 Dated 21/11/2007. Authors are thankful to the management of their respective organizations. As well as grateful to my research guide, Dr. G Sasi Bhushana Rao, Head of the Department, Electronics and Communication Engineering, Andhra University, India, towards the Research Degree.

REFERENCES

[1.] Gonzalez, R. C., Woods, R. E. and Eddins, S. L., "Digital Image Processing using MATLAB", Pearson Education, 3rd edition, 2005
[2.] J. C. Russ. "The Image Processing Handbook". CRC Press, Boca Raton, Fla., 1992.
[3.] D. Anoraganingrum, "Cell segmentation with median filter and mathematical morphology operation," in Proceedings of IEEE International Conference on Image Analysis and Processing, Venice, Italy, September 27-29 1999.
[4.] Aurich, V., Weule, J.: "Non-linear gaussian filters performing edge preserving diffusion". In: Mustererkennung 1995, 17. DAGM-Symposium, Springer-Verlag (1995) 538–545
[5.] I.Pitas and A.N. Venetsanopoulos, "Nonlinear Digital Filters: Principles and Applications". Boston, MA: Kluwer Academic, 1990.
[6.] D. Anoraganingrum, "Cell segmentation with median filter and mathematical morphology operation," in Proceedings of IEEE International Conference on Image Analysis and Processing, Venice, Italy, September 27-29 1999.
[7.] Bellon O.R.P., Direne A.I. and Silva L., "Edge detection to guide range image segmentation by clustering techniques", International Conference on Image Processing, ICIP 99. Proceedings. Vol.2, 24-28, pp:725 – 729, Oct. 1999
[8.] Xu L., Landabaso J. L. and Lei, B., "Segmentation and tracking of multiple moving objects for intelligent video analysis", BT Technology Journal.
[9.] Fox W. L. J., Hsieh J. and Polwarth C., "Segmentation of images from an acoustic lens sonar", MTTS/IEEE TECHNO-OCEAN '04, Vol.4, pp: 2029– 2035, 2004.
[10.] Satya Swaroop Pradhan, Dipti Patra, Pradipta Kumar Nanda, "Adaptive Thresholding Based Image Segmentation with Uneven Lighting Condition", Proc. on IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, INDIA, 8-10 December, 2008.