A Comparative Analysis of Noise Filtering Performance of Quadratic Image Filters

Süleyman Uzun[1,*], Devrim Akgün[2]
[1]*Department of Computer Engineering, Bilecik Seyh Edebali University, Gulumbe Campus, 11210, Bilecik, Turkey*
[2]*Department of Software Engineering, Sakarya University, 54050, Sakarya, Turkey*

[1]***Abstract*— Quadratic image filters belong to a subclass of nonlinear model known as Volterra filters. Because of the nonlinear characteristics of images, nonlinear image filters generally produce better results than linear filters. In the present study, performance of the Quadratic image filters for Gaussian noise is examined by comparing with Gaussian filter and Median filter. For this purpose, the mask weights used were determined by using Differential Evolution algorithm on synthetic training images. Noise added colour test images were filtered using Quadratic image filter using the calculated weights and the results were compared with Gaussian filter and Median image filter.**

***Keywords*— Differential Evolution Algorithm, Gaussian Filtering, Noise Removal, Quadratic Image Filter.**

I. INTRODUCTION

Image filtering is one of the most important image processing applications such as noise filtering, edge detection, image enhancement or feature extraction applications. Image filtering performed in the time domain by linear convolution involves moving a window over the image to calculate the output image by multiplying the selected pixel window with a matrix of coefficients [1]. Because of the non-linear nature of the image content, in some applications, filters with non-linear properties are preferred over linear filters [2]. One of the nonlinear alternatives to linear convolution is quadratic image filter which is a subclass of Volterra filters. The Quadratic Image Filter (QIF), which works similarly to linear convolution but it also uses quadratic multiplications of the pixels in the mask selected as input [3]. A nonlinear system can theoretically be modelled with the Volterra series with infinite elements. Most image processing applications use infinite series to include the quadratic term. In recent years, QIFs attracts researchers due to its success in various image processing applications. QIFs are often used in noise filtering, such as eliminating Gaussian noise or reducing impulse noise due to edge protection properties [4]–[6] and edge detection applications [7]. QIFs has been used to remove classification features in face recognition applications [8]–[10]. QIFs has also been preferred in some medical image processing applications, especially for processing mammogram images for enhancement or noise reduction [11]–[13]. Determining the proper weights of QIFs for the desired application is of crucial importance. One of the alternatives is to use optimization approach [14], [15]. Definition of fitness function directly affects the optimized weights and therefore the success of the QIF. The aim of the presented study is to show the filtering quality of quadratic image filter for Gaussian noise using synthetic training image. Then, the results will be compared with Gaussian and Median filter results visually. In the following section a background information about mathematical description of QIFs were given. In the third section optimization approach for determining the QIF weights was explained. In the fourth section QIF were applied to test image and compared with Gaussian and Median image filters. In the final section, conclusions about the study were given.

Manuscript received December 01, 2019; accepted December 27, 2019.
*Corresponding author: suleyman.uzun@bilecik.edu.tr

II. BACKGROUND

QIF is a subclass of Volterra filters and it is used for modelling nonlinear systems. QIF contains terms of Volterra filters truncated to second order as shown by Eq. (1). Note that constant term is excluded in the present study.

$$y(m,n) = y_1(m,n) + y_2(m,n) \tag{1}$$

Where, m and n show the coordinates of pixel to be filtered, $y_1(m,n)$ represents the linear component, $y_2(m,n)$ represents the quadratic component. Sum of these terms yields $y(m,n)$ which is the filtered pixel. The expressions for $y_1(m,n)$ and $y_2(m,n)$ are given by Eq. (2). M represents the number of rows and N represents the number of columns. These expressions are used for gray images and can be repeated for red, green and blue to perform colour image filtering.

$$\left.\begin{aligned} y_1(m,n) &= \sum_{i=-M}^{M}\sum_{j=-M}^{M} w^1_{i,j} x_{m+i,n+j} \\ y_2(m,n) &= \sum_{i=-M}^{M}\sum_{j=-M}^{M}\sum_{k=-M}^{M}\sum_{l=-M}^{M} w^2_{i,j,k,l} x_{m+i,n+j} x_{k+i,l+j} \end{aligned}\right\} \tag{2}$$

Above equation can be simplified as follows;

$$y(m,n) = W_1 X^{1\,T}_{m,n} + W_2 X^{2\,T}_{m,n} \tag{3}$$

Where the vector variables W_1 and W_2 describes the weights and X^1 describes input pixels and X^2 describes the second

order multiplications. The content of these variable are shown by Eq. (4). p is the number of weights in the quadratic part. Note that repeating terms in $X^2_{m,n}$ are excluded.

$$\left.\begin{aligned} W_1 &= \left[w_0^1\ w_1^1\ w_2^1 \cdots w_{NxN-1}^1 \right] \\ W_2 &= \left[w_0^2\ w_1^2\ w_2^2 \cdots w_p^2 \right] \\ X^1_{m,n} &= \left[x_0\ x_1\ x_2 \cdots x_{NxN-1} \right] \\ X^2_{m,n} &= \left[x_0 x_0\ \ x_0 x_1 \cdots x_{NxN} x_{NxN-1} \right] \end{aligned}\right\} \quad (4)$$

III. DETERMINING THE WEIGHTS USING OPTIMIZATION

Determining optimal filter weights is of importance in obtaining desired filtering quality. In order to determine the weights of QIF, Differential Evolution algorithm were utilized in this study. Basic working principle of Differential Evolution algorithm is given below [16].

Algorithm 1: Basic code for Differential Evolution algorithm

```
1 Generate initial population of candidate solutions
2 while termination condition is not satisfied do
3     for each Indvidual in population do
4         MutantIndividual ← Mutation(Individual)
5         TrialIndividual ←
            Recombination(MutantIndividual)
6         if fitness(TrialIndividual) <=
            fitness(Indvidual) then
7             Indvidual ← TrialIndividual
```

Initial populations which forms the candidate solution in vector form are generated randomly. The size of the vector is determined by the number of weights. In the present case, the total number of weights is 54 where 9 terms for linear part and 45 terms for quadratic part. Mutation operation is used for generating new population from randomly chosen individuals. Obtained individuals are recombined with other individuals yielding trial vector. If trial individual decrease the fitness function then it is selected for the next generation.

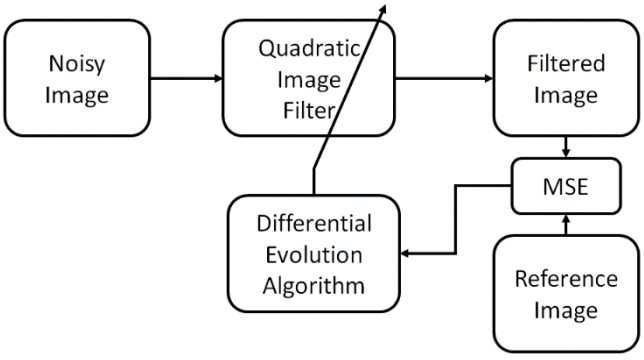

Fig. 1. Training QIF using Differential Evolution Algorithm and MSE as fitness function.

As shown by Figure 1, weights of the QIF is determined by an optimization algorithm which can be other than Differential Evolution. Candidate solutions are modified according to fitness function. Fitness function is determined using Minimum Squared Error (MSE) function where the difference of reference image and filtered image are squared and divided to the number of pixels. This is shown by Eq. (5) where the R and F describes the reference and filtered images which are of the size of $M \times N$.

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (R(i,j) - F(i,j))^2 \quad (5)$$

Figure 2 show the reference image and noise added image. Filtered image is computed by applying noisy image to QIF to obtain the desired image which is reference image.

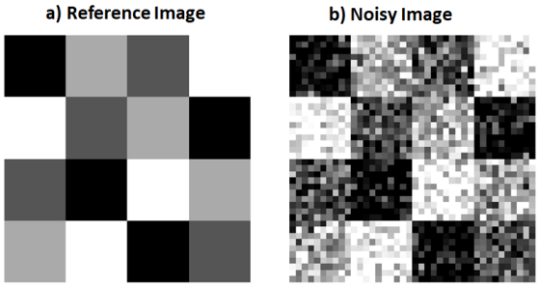

Fig. 2. Synthetic reference image and noisy image for fitness computation.

IV. EXPERIMENTAL RESULTS

Experimental results were obtained using GNU Octave 5.1.0 and optimization of the QIF weights were realized using *de_min*() function in *Optim* package [17]. The training was realized 1000 number of populations and 300 iterations. Also the bounds of the population are set to -0.4 and 0.4 interval. These bounds can be expanded or narrowed experimentally according to problem type and affects the convergence of the algorithm significantly. Gray images were used in training as shown by Figure 2. The same weights were used to filter color images for the experimental results that are shown by Figure 3 and Figure 4. In the both cases, reference images are added Gaussian noise for testing where mean and variance are 0 and 0.8 respectively. The window sizes of both Gaussian filter and Median Filter is 3×3. Table I shows the numerical evaluation results in terms of MSE. According to Figure 3, Gaussian filter produced better filtering quality than QIF. On the other hand, QIF is clearly shown to preserve small details better than Gaussian filter when inspected visually. The same is also valid for synthetic image filtering example as shown by Figure 4. In this case, QIF was compared with Median filter which another type of nonlinear filter. It should be noted that the filtering results highly dependent on the filter coefficients which are determined by the training image. The coefficients determined by Differential evolution algorithm are given by Table II. First nine terms are used for linear part and the remaining terms are used for the quadratic terms.

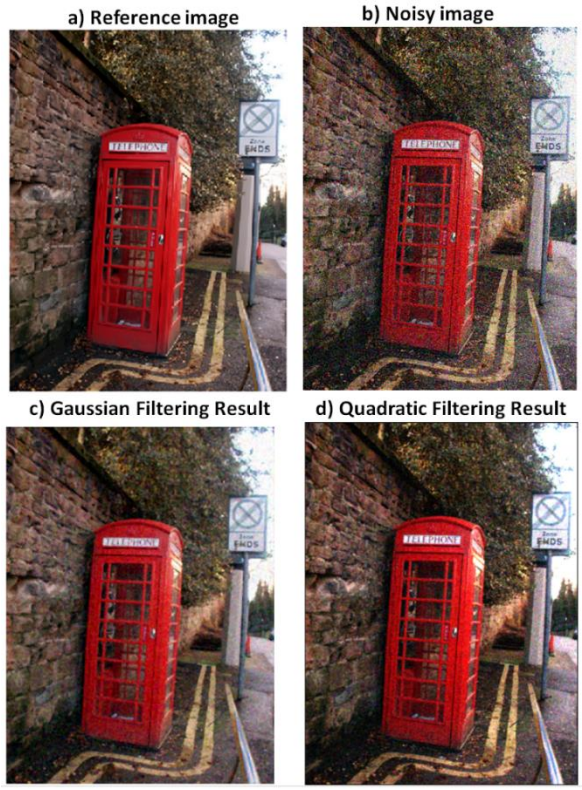

Fig. 3. Filtering results for test image with Gaussian noise.

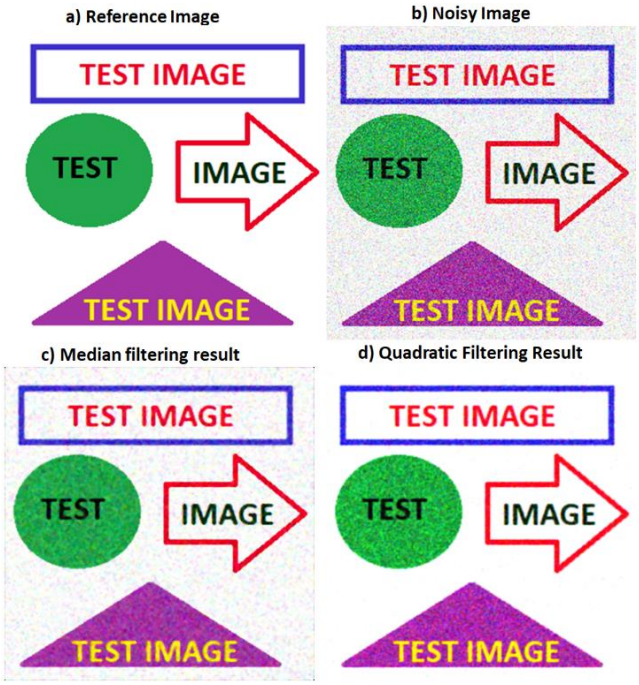

Fig. 4. Filtering results for synthetic image with Gaussian noise.

TABLE I. MSE COMPARISON OF THE QIF, GAUSSIAN AND MEDIAN FILTERS

Figure	Noisy Image	QIF	Gaussian filter	Median Filter
Fig. 3	1661	479.7	365.2	562.1
Fig. 4	811.8	241.2	422.7	247.8

TABLE II. QUADRATIC FILTER WEIGHTS OBTAINED FROM TRAINING USING SYNTHETIC IMAGE

Term	Weight	Term	Weight
x1	0.02938489	x4x3	0.06278219
x2	0.09462171	x5x3	0.05273079
x3	-0.03092593	x6x3	-0.04749037
x4	0.14087978	x7x3	0.07850170
x5	0.25860455	x8x3	-0.01930844
x6	0.10114809	x9x3	-0.05494301
x7	0.00016975	x4x4	-0.01977208
x8	0.08572573	x5x4	0.04287500
x9	-0.02206239	x6x4	-0.05896785
x1x1	-0.00450145	x7x4	-0.05693241
x2x1	-0.05200057	x8x4	0.23985235
x3x1	0.02646522	x9x4	0.00461186
x4x1	0.05254662	x5x5	-0.02658849
x5x1	0.00955084	x6x5	0.02124616
x6x1	-0.01685976	x7x5	0.05042445
x7x1	-0.03299340	x8x5	0.06988566
x8x1	-0.07019090	x9x5	-0.03187242
x9x1	0.06706213	x6x6	0.02430864
x2x2	0.06686967	x7x6	0.00213875
x3x2	-0.07205827	x8x6	-0.05021069
x4x2	-0.13747218	x9x6	0.08822913
x5x2	0.09917625	x7x7	0.01975957
x6x2	0.21809982	x8x7	-0.08416840
x7x2	0.06577243	x9x7	-0.00636219
x8x2	0.01875156	x8x8	0.02548373
x9x2	-0.09418433	x8x9	0.04497577
x3x3	0.06294752	x9x9	0.04151030

V. CONCLUSION

Linear convolution and QIF work in the same way except that QIF uses second order multiplications in addition to linear terms. This gives QIFs better processing ability over linear convolution, if the QIF weights are well determined. In the present study performance of QIFs for image filtering is examined for Gaussian noise. Differential Evolution algorithm was utilized to obtain filter weights according to fitness function. The filter was trained by using a synthetic image as reference image. The input image for the QIF was obtained by adding Gaussian noise to reference image. According to results QIF produced better visual results over Gaussian filter which is a kind of linear filter. QIF also produced better performance over Median filter which is a kind of nonlinear filter. Performance of the results will obvious to vary according to the training images used in the fitness function. In addition, optimization algorithm type and parameters are other factors that affect performance and can be included in future studies.

ACKNOWLEDGMENT

This study was presented orally as abstract paper at the ICONDATA 2019 conference.

REFERENCES

[1] J. C. Russ, *The image processing handbook*. CRC press, 2016.

[2] I. Pitas and A. N. Venetsanopoulos, *Nonlinear digital filters: principles and applications*, vol. 84. Springer Science & Business Media, 2013.

[3] G. F. Ramponi, G. L. Sicuranza, and W. Ukovich, “A

computational method for the design of 2-D nonlinear Volterra filters," *IEEE Trans. Circuits Syst.*, vol. 35, no. 9, pp. 1095–1102, 1988.

[4] J. Zhang and Y. Pang, "Pipelined robust M-estimate adaptive second-order Volterra filter against impulsive noise," *Digit. Signal Process.*, vol. 26, pp. 71–80, Mar. 2014.

[5] L. Thomas, G. Krishnan, R. A. Mol, and A. Roy, "Removal of Impulsive Noise from MRI Images using Quadratic Filter," *Int. J. Eng. Res. Technol.*, vol. 3, no. 4, pp. 2220–2223, 2014.

[6] M. B. Meenavathi and K. Rajesh, "Volterra Filtering Techniques for Removal of Gaussian and Mixed Gaussian-Impulse Noise," *Int. J. Electr. Robot.*, vol. 1, no. 2, pp. 1–7, 2007.

[7] V. S. Hari, V. P. Jagathy Raj, and R. Gopikakumari, "Quadratic filter for the enhancement of edges in retinal images for the efficient detection and localization of diabetic retinopathy," *Pattern Anal. Appl.*, vol. 20, no. 1, pp. 145–165, Feb. 2017.

[8] A. Chakrabarty, H. Jain, and A. Chatterjee, "Volterra kernel based face recognition using artificial bee colonyoptimization," *Eng. Appl. Artif. Intell.*, vol. 26, no. 3, pp. 1107–1114, 2013.

[9] G. Feng, H. Li, J. Dong, and J. Zhang, "Direct Discriminant Analysis Using Volterra Kernels for Face Recognition," 2016, pp. 404–412.

[10] G. Feng, H. Li, J. Dong, and J. Zhang, "Face recognition based on Volterra kernels direct discriminant analysis and effective feature classification," *Inf. Sci. (Ny).*, vol. 441, pp. 187–197, 2018.

[11] V. Bhateja, M. Misra, and S. Urooj, "Non-linear polynomial filters for edge enhancement of mammogram lesions," *Comput. Methods Programs Biomed.*, vol. 129, pp. 125–134, 2016.

[12] A. Pandey, A. Yadav, and V. Bhateja, "Design of new volterra filter for mammogram enhancement," in *Advances in Intelligent Systems and Computing*, vol. 199 AISC, pp. 143–151, 2013.

[13] Y. Zhou, K. Panetta, and S. Agaian, "Mammogram enhancement using alpha weighted quadratic filter," in *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*, 2009, pp. 3681–3684.

[14] S. Uzun and D. Akgün, "An Accelerated Method for Determining the Weights of Quadratic Image Filters," *IEEE Access*, vol. 6, 2018.

[15] S. Uzun and D. Akgün, "Accelerated method for the optimization of quadratic image filter," *J. Electron. Imaging*, vol. 28, no. 03, p. 1, Jun. 2019.

[16] V. P. Plagianakos, D. K. Tasoulis, and M. N. Vrahatis, "A review of major application areas of differential evolution," in *Advances in differential evolution*, Springer, 2008, pp. 197–238.

[17] W. Eaton John, B. David, and W. Rik, "GNU Octave version 5.1. 0 manual: a high-level interactive language for numerical computations, 2019," *URL https//www. gnu. org/software/octave/doc/v5*, vol. 1.