

# Proposal of Edge-Preserving, Image Noise Reduction Filter for Using L2-Norm <sup>†</sup>

Yudai Yamaguchi <sup>1</sup>, Ichiro Yoshida <sup>2,\*</sup>  and Yuki Kondo <sup>2</sup>

<sup>1</sup> Mechanical Engineering, Graduate School of Science and Engineering, HOSEI University, Koganei-shi 184-8584, Japan; yudai.yamaguchi.2b@stu.hosei.ac.jp

<sup>2</sup> Department of Mechanical Engineering, Faculty of Science and Engineering, HOSEI University, Koganei-shi 184-8584, Japan; yuki.kondoh.57@hosei.ac.jp

\* Correspondence: yoshida.ichiro@hosei.ac.jp; Tel.: +81-042-387-6033

<sup>†</sup> Presented at the 2nd International Electronic Conference on Applied Sciences, 15–31 October 2021; Available online: <https://asec2021.sciforum.net/>.

**Abstract:** Images taken by digital cameras include noise. The image recognition rate decreases with increasing noise. Reducing noise is essential for improving the accuracy of image recognition. Low-pass filters, such as a Gaussian filter (GF), are often used to reduce noise from images. Low-pass filters can reduce noise. However, low-pass filters always blur the edges. As the edge blur becomes stronger, the accuracy of edge and feature detection of image recognition worsens. Therefore, we propose a noise reduction filter for images that can preserve edges by combining the GF and the L2-norm. The proposed method is expected to improve the image quality and, consequently, the accuracy of image recognition.

**Keywords:** digital image processing; noise reduction; edge preserving; gaussian filter; L2-norm



**Citation:** Yamaguchi, Y.; Yoshida, I.; Kondo, Y. Proposal of Edge-Preserving, Image Noise Reduction Filter for Using L2-Norm. *Eng. Proc.* **2021**, *11*, 27. <https://doi.org/10.3390/ASEC2021-11170>

Academic Editor: Nunzio Cennamo

Published: 15 October 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Images taken by digital cameras include noise [1]. Image quality reduces with increasing noise. In addition, the recognition rate of image recognition decreases with increasing noise. Currently, image recognition is used in security technology through face recognition and in image inspection at production sites. Therefore, the accuracy of image recognition needs to be improved. Reducing noise is essential for improving the accuracy of image recognition. Low-pass filters, such as a Gaussian filter (GF) [2], are often used to reduce noise in images. Low-pass filters can reduce noise; however, low-pass filters always blur the edges [3]. As the edge blur becomes stronger, the accuracy of the edge and the feature detection of image recognition worsens [4]. Therefore, noise reduction filters need to be able to preserve edges.

A non-local mean filter (NLMF) [5] was proposed as a noise reduction filter for images that can preserve edges. The NLMF has high denoising performance for weak noise. However, the NLMF has low denoising performance for strong noise. In addition, the NLMF has a problem in that the amount of calculation is large and the processing time is long.

In order to solve this problem, in this research, we propose a noise reduction filter for images that can preserve edges by combining the GF and the L2-norm [6]. The proposed method is expected to simultaneously achieve high denoising performance and edge-preservation performance compared to NLMF by using the L2-norm. In addition, the proposed method can shorten the processing time compared with NLMF. Therefore, the proposed method is expected to improve the image quality and, consequently, the accuracy of image recognition. The remainder of this paper is organized as follows: we describe the proposed method in Section 2, verification experiments are presented in Section 3, and the paper is summarized in Section 4.

## 2. Proposed Method

The filter proposed in this research is a filter for images that combines the L2-norm and GF (hereinafter referred to as L2GFI). The algorithm for the proposed method is shown in the following bullet points. In addition, the software we used to develop the proposed method is MATLAB.

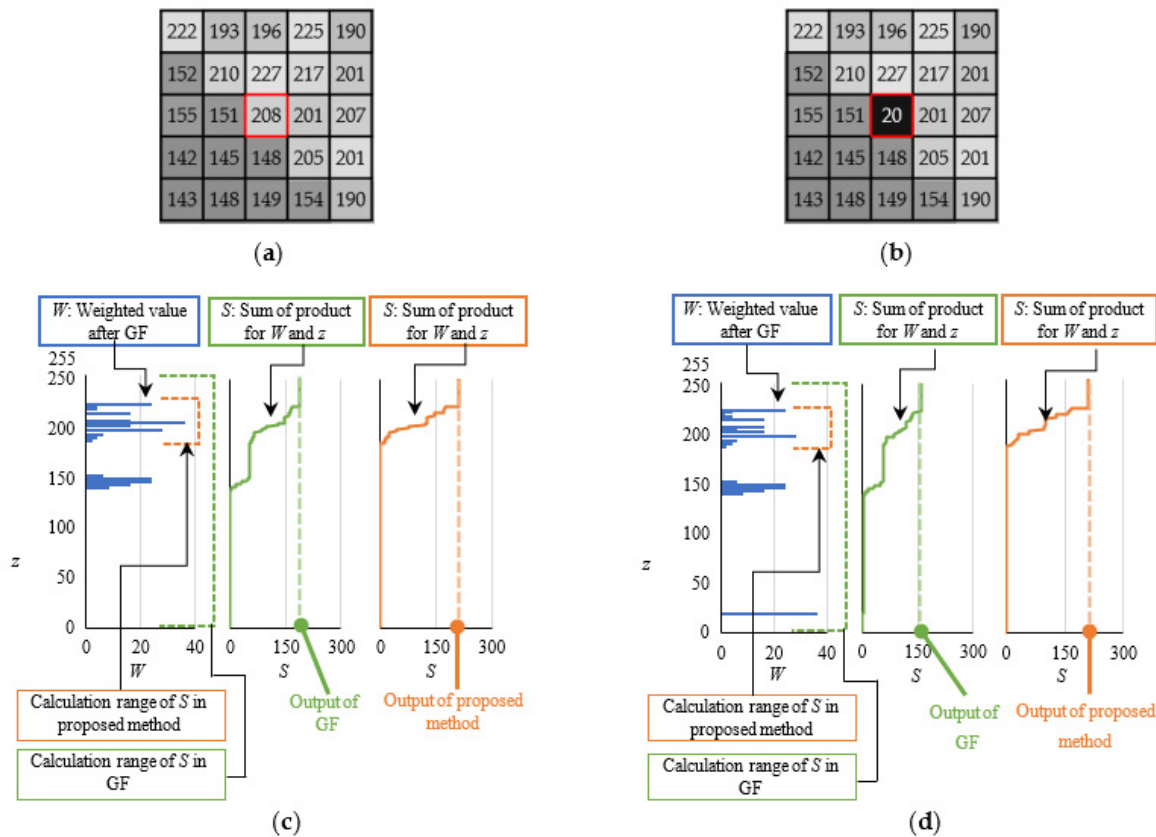
1. The reference points are set in the cells corresponding to each coordinate in the original image;
2. The Gaussian function is applied in the  $xy$ -direction;
3. In the  $z$ -coordinate, the range where the value of the weight does not become zero consecutively is recognized as a mountain of weight;
4. The mountain where the sum of weights is the maximum is selected;
5. The output value is calculated through a convolution of the normalized weights and the pixel value  $z$  in the range of the selected mountain.

The specific processing procedure for the L2GFI is explained. The L2GFI needs to assign an array to pixel value  $z$  in addition to the  $xy$ -coordinate. Therefore, processing the L2GFI requires a three-dimensional array in the case of two-dimensional grayscale images. The L2GFI plots reference points in the corresponding cells of the  $x$ -coordinate,  $y$ -coordinate, and pixel value  $z$  of each pixel in the image. The GF is applied in the  $xy$ -direction for the plotted reference points for each pixel. At that time, the weight array of GF is added to the corresponding cell. After applying the GF in the calculation of the L2GFI algorithm, the distribution of weights in the  $z$  direction is checked for each pixel. The range where the value of the weight does not become zero consecutively is recognized as a mountain of weight. At that time, if the distance between the neighboring mountains is small, the neighboring mountains are recognized as one mountain. The output value is calculated only from the range of the mountain where the sum of weights is the maximum. The output value is calculated as the sum of the product for the normalized weights and pixel value  $z$ . Let the pixel values in the calculation range be  $z_1, z_2, \dots, z_n$ , the weights of the Gaussian function be  $w_1, w_2, \dots, w_n$ , and the output value be  $o$ , and then the output of L2GFI can be shown as Equation (1):

$$o = \sum_{i=1}^n z_i w_i \quad (1)$$

A comparison of the output processing when the target pixel is a normal value and when the target pixel is noise is shown in Figure 1. Figure 1a shows the target pixel and surrounding pixels when the target pixel is a normal value. Figure 1b shows the target pixel and its surrounding pixels when the target pixel is noise. Figure 1c shows the weight distribution in the  $z$ -direction, the output process of the GF, and the output process of the L2GFI after applying GF to Figure 1a. Figure 1d shows the weight distribution in the  $z$ -direction, the output process of the GF, and the output process of the L2GFI after applying the GF to Figure 1b. Comparing the output values of the GF in Figure 1c,d, the output value of Figure 1d shows a larger difference from the normal value than the output value of Figure 1c. The output value of the GF is affected by noise because the GF calculates from all ranges in the  $z$ -direction. Comparing the output values of the L2GFI in Figure 1c,d, it can be seen that the output values of the L2GFI are not different. This is because the output values of the L2GFI are calculated only from the range of the mountain where the sum of weights is maximum. Therefore, L2GFI is expected to achieve high denoising performance. In addition, when L2GFI (with size  $5 \times 5$ ) is applied to an area that includes edges, such as in Figure 1a, the weight distribution has two mountains, as shown in Figure 1c. The sum of the weights tends to be larger for the mountains that include the target pixel due to the property of the Gaussian function. As a result, the L2GFI is able to preserve the edges because the output value is calculated only from the pixel values of the edge that includes the target pixel. Furthermore, as shown in Figure 1b,d, the L2GFI can remove noise without blurring the edges, even when noise is included in the edges. The L2GFI can

also be expected to achieve high edge-preserving performance by processing other pixels in the same way.



**Figure 1.** Comparison of output processing of normal value and noise when using GF and L2GFI; (a) sample image of only normal values; (b) sample image with noise; (c) output processing of GF and L2GFI for (a); (d) output processing of GF and L2GFI for (b).

### 3. Experiments

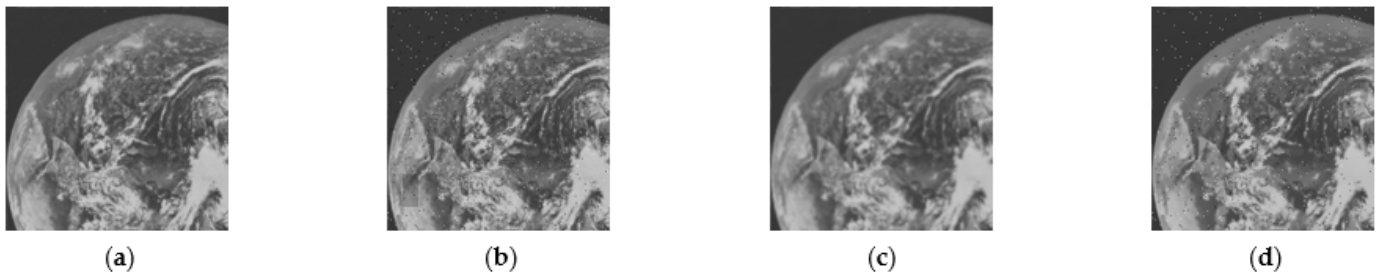
#### 3.1. Noise Reduction Experiment

The denoising and edge-preserving performances of the L2GFI algorithm are verified here. The experiment compares the denoising and edge-preserving performances of the L2GFI and the NLMF on a noise-added sample image using visual inspection and root mean square error (RMSE). As the value of RMSE becomes lower, it can be said that the reproducibility of the original image becomes higher. A higher reproducibility of the original image indicates higher denoising and edge-preservation performances. Three types of standard images are used as sample images for the experiment. The size of the sample image is  $256 \times 256$  pixels. We apply each filter with a filter size of  $5 \times 5$  pixels. In this experiment, to avoid implementation errors, the NLMF is used as the function of OpenCV 4.5.3 [7]. The parameter values of the NLMF use the recommended values of the function.

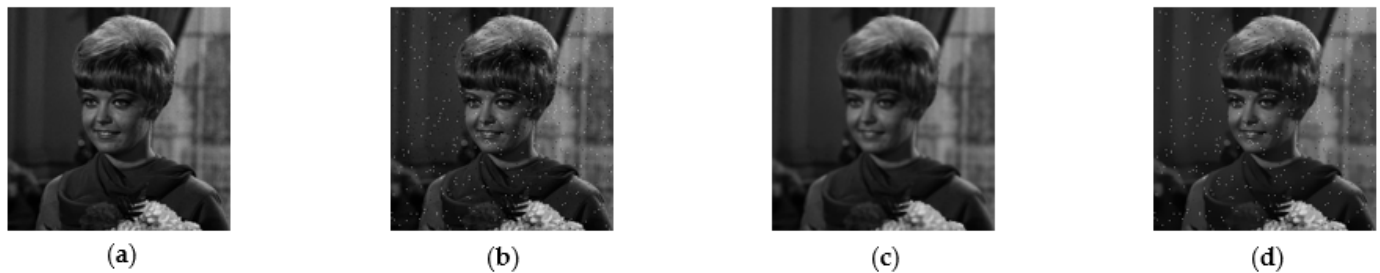
Figures 2–4 show the original image, noise-added image, and result of applying each filter to the noise-added image for each sample image. The output images of each filter in Figures 2–4 are compared through a visual inspection. The output image of the NLMF showed no major changes in any of the samples compared to the noise-added image. It can be confirmed that the NLMF has a high edge-preserving performance, and a low denoising performance for strong noise. The output image for the L2GFI showed that noise is removed in all the samples. Therefore, it can be seen that the L2GFI has higher denoising and edge-preservation performances compared to the NLMF.



**Figure 2.** Filtering results for sample image “Airplane”; (a) input image; (b) image with noise added to (a); (c) filtering result of (b) by L2GFI; (d) filtering result of (b) by NLMF.



**Figure 3.** Filtering results for sample image “Earth”; (a) input image; (b) image with noise added to (a); (c) filtering result of (b) by L2GFI; (d) filtering result of (b) by NLMF.



**Figure 4.** Filtering results for sample image “Girl”; (a) input image; (b) image with noise added to (a); (c) filtering result of (b) by L2GFI; (d) filtering result of (b) by NLMF.

Table 1 lists the RMSE values calculated using each filter (first row) and deviation ratio (%) from the RMSE values calculated using the L2GFI (second row) for each sample image. The first row of Table 1 shows that the RMSE value of NLMF is larger than the RMSE value of L2GFI. The second row of Table 1 shows how large the RMSE value of NLMF is compared to the RMSE value of L2GFI. From Table 1, the RMSE values of the L2GFI are 34 (%) or lower than those of the NLMF for all sample images. Therefore, it can be said that the L2GFI has higher denoising and edge-preservation performances than the NLMF in the evaluation of RMSE values.

**Table 1.** RMSE values calculated using each filter (upper section) and deviation ratio (%) from RMSE values calculated using L2GFI (lower section).

Sample Image	L2GFI	NLMF
Airplane	9.47	14.55
	-	34.88
Earth	5.94	13.56
	-	56.23
Girl	5.20	15.04
	-	65.38

### 3.2. Processing Time

The high-speed processing performance of L2GFI is verified. The experiment compares the processing times of the L2GFI and NLMF. We used the functions provided by the software to measure the processing time. The processing time in the comparison experiment is the average time when the filter is applied 10 times. The sample images and parameter values of each filter are the same as those in the experiment in Section 3.1. As a result of the experiment, the average processing time of the L2GFI is 0.2 s, whereas the average processing time of the NLMF is 17.6 s. The processing time of L2GFI is 88 times faster than that of the processing time of the NLMF. In order to verify the uncertainty of the measurement, we calculated the standard deviation of the data for 10 measurement times. The standard deviation of the calculation time for the L2GFI is 0.007 (s); therefore, the calculation time for the L2GFI is  $0.2 \pm 0.014$  s, with a coverage factor  $k = 2$ . The standard deviation of the calculation time for the NLMF is 0.01 (s); therefore, the calculation time for the NLMF is  $17.6 \pm 0.02$  s ( $k = 2$ ). The uncertainty of these measurements are, therefore, small. Furthermore, in terms of high-speed processing performance, the L2GFI is better than the NLMF.

## 4. Conclusions

In this research, we proposed the L2GFI, a noise reduction filter for images that can preserve edges that combines the GF and L2-norm. The results of this research are as follows.

1. A visual comparison of the experimental results showed that the L2GFI has better denoising and edge-preservation performances than the NLMF.
2. As a result of experimental verification, L2GFI had RMSE values that were 34 (%) or lower than those of the NLMF. Consequently, it can be said that the reproducibility of the original image of the L2GFI is at a higher level than that of the NLMF.
3. The verification results showed that the processing time of the L2GFI was faster than that of the NLMF. Therefore, the L2GFI is better than NLMF in high-speed processing performance.

From these results, it can be said the L2GFI combines high denoising, edge-preservation, and high-speed processing performances compared to the NLMF. We considered that the performance of the L2GFI needs to be further validated for practical applications. Therefore, in the future, we will verify the performance of the L2GFI for more complex sample images and other types of noise.

**Author Contributions:** Conceptualization, I.Y.; methodology, Y.Y., and I.Y.; software, Y.Y., and I.Y.; validation, Y.Y., and I.Y.; formal analysis, Y.Y., and I.Y., Y.K.; investigation, I.Y., and Y.Y.; resources, I.Y.; data curation, Y.Y.; writing—original draft preparation, Y.Y., and I.Y.; writing—review and editing, Y.Y., and I.Y., Y.K.; visualization, Y.Y.; supervision, I.Y.; project administration, I.Y.; funding acquisition, I.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Singh, H.; Kommuri, S.V.R.; Kumar, A.; Bajaj, V. A new technique for guided filter based image denoising using modified cuckoo search optimization. *Expert Syst. Appl.* **2021**, *176*, 114884. [[CrossRef](#)]
2. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 3rd ed.; Prentice-Hall: Hoboken, NJ, USA, 2006; p. 298.
3. Wang, M.; Zheng, S.; Li, X.; Qin, X. A new image denoising method based on Gaussian filter. In Proceedings of the 2014 International Conference on Information Science, Electronics and Electrical Engineering (ISEEE), Sapporo, Japan, 26–28 April 2014.

4. Kondo, Y.; Yoshida, I.; Numada, M.; Koshimizu, H. A study on edge preserving noise reduction filter using fast m-estimation method. *J. Jpn. Soc. Precis. Eng.* **2020**, *86*, 1034–1041. (In Japanese) [[CrossRef](#)]
5. Buades, A.; Coll, B.; Morel, J.M. A non-local algorithm for image denoising. *CVPR* **2005**, *2*, 60–65.
6. Omidikia, N.; Ghaffari, M.; Rajko, R. Sparse non-negative multivariate curve resolution: L0, L1, or L2 norms? *Chem. Intell. Lab. Syst.* **2020**, *199*, 103969. [[CrossRef](#)]
7. OpenCV Denoising. Available online: <https://docs.opencv.org/2.4/modules/photo/doc/denoising.html> (accessed on 3 August 2021).