Self-Quotient $\varepsilon$-Filter for Feature Extraction from Noise Corrupted Image

Mitsuharu MATSUMOTO$^{1a}$, Member

SUMMARY This paper describes a nonlinear filter that can extract the image feature from noise corrupted image labeled self-quotient $\varepsilon$-filter (SQEF). SQEF is an improved self-quotient filter (SQF) to extract the image feature from noise corrupted image. Although SQF is a simple approach for feature extraction from the images, it is difficult to extract the feature when the image includes noise. On the other hand, SQEF can extract the image feature not only from clear images but also from noise corrupted images with simple calculation. To achieve this goal, we propose a self-quotient $\varepsilon$-filter (SQEF) by combining the concepts of $\varepsilon$-filter and SQF. It is defined as the ratio of two different $\varepsilon$-filters, and can extract the feature not only from clear images but also from uniform noise corrupted images, Gaussian noise corrupted images and impulse noise corrupted images. The calculation cost of SQEF is still small due to its simple design.

The paper is organized as follows. In Sect. 2, we first describe the algorithm of the conventional $\varepsilon$-filter. In Sect. 3, we describe the algorithm of the SQF to clarify the handling problem. The algorithm and the features of the SQEF are described in Sect. 4. In Sect. 5, we show our experimental results to confirm the effectiveness on feature extraction from uniform noise corrupted images, Gaussian noise corrupted images and impulse noise corrupted images. We also give the results when we used light fluctuated images to evaluate the robustness of fluctuations of light and shade. Calculation cost is also compared to the existing methods. Discussions and conclusion are given in Sect. 6.

1. Introduction

Self-quotient filter (SQF) is a simple nonlinear filter to extract the feature from an image [1], [2]. It is based on quotient image (QI) method [3], [4], and related to the existing works of Spherical Harmonic [5]–[8] and Image Ratio [9]–[12]. It needs only an image, and can extract intrinsic lighting invariant property of an image while removing extrinsic factor corresponding to the lighting. Feature extraction by SQF is simpler than that based on multi-scale smoothing [13]. SQF can extract the outline of the objects independent of shadow region. However, as it assumes that the image does not include noise, it can not extract the shape and texture when the noise damages the image. The noise influence becomes large due to the self-quotient effect of SQF.

To solve the problem, we look to a nonlinear filter labeled $\varepsilon$-filter [14]–[16]. $\varepsilon$-filter is a nonlinear filter, which can reduce the noise while preserving the edge. Although many studies have been reported to reduce the small amplitude noise while preserving the edge [17]–[19], it is considered that $\varepsilon$-filter is a promising approach due to its simple design. It does not need to have the signal and noise models in advance. It is easy to design and the calculation cost is small because it requires only switching and linear operation.

The motivation of our work is extracting the image feature not only from clear images but also from noise corrupted images with simple calculation. To achieve this goal, we propose a self-quotient $\varepsilon$-filter (SQEF) by combining the concepts of $\varepsilon$-filter and SQF. It is defined as the ratio of two different $\varepsilon$-filters, and can extract the feature not only from clear images but also from uniform noise corrupted images, Gaussian noise corrupted images and impulse noise corrupted images. The calculation cost of SQEF is still small due to its simple design.

The window size of $\varepsilon$-filter is $2M + 1$. $F(x)$ is the nonlinear function described as follows:

$$|F(x)| \leq \varepsilon : -\infty \leq x \leq \infty,$$

where $\varepsilon$ is the constant number. This method can reduce small amplitude noise while preserving the signal. For example, we can set the nonlinear function $F(x)$ as follows:
is designed as follows:
\[
\begin{align*}
y(i_1, i_2) &= \Phi_{\epsilon,M}[x(i_1, i_2)] \\
&= y(i_1, i_2) \\
&+ \sum_{j_1=-M}^{M} \sum_{j_2=-M}^{M} a(j_1, j_2) F(x(i_1 + j_1, i_2 + j_2) - x(i_1, i_2)),
\end{align*}
\]

where \(a(j_1, j_2)\) represents the filter coefficient. \(a(j_1, j_2)\) is usually constrained as follows:
\[
\sum_{j_1=-M}^{M} \sum_{j_2=-M}^{M} a(j_1, j_2) = 1.
\]

The feature of two dimensional \(\epsilon\)-filter is similar to that of one dimensional \(\epsilon\)-filter. We can smooth the small amplitude noise near by the processed point while preserving the edge. It requires less calculation when it is compared to bilateral filter because it requires only switching and linear operation. However, due to this procedure, \(\epsilon\)-filter basically contains low-pass characteristics.

3. Self-Quotient Filter

Let us consider \(x(i_1, i_2)\) as the image intensity at the point \(i = (i_1, i_2)\) in the image. In self-quotient filter, we restrict our consideration to objects with a Lambertian model, i.e., the image is described by the product of the albedo (texture) and the cosine angle between a point light source and the surface normal. Under the above situation, \(x(i_1, i_2)\) can be described as follows:
\[
x(i_1, i_2) = \rho(i_1, i_2) l(i_1, i_2)^T s,
\]
where \(\rho(i_1, i_2)\) is the surface reflectance (gray-level) associated with the point \((i_1, i_2)\) in the image. It is constrained as follows:
\[
0 \leq \rho(i_1, i_2) \leq 1.
\]
\(l(i_1, i_2)\) is the surface normal direction associated with point \((i_1, i_2)\) in the image. \(s\) is the light source direction. Its magnitude is the light source intensity. \(T\) represents the transpose. The aim of self-quotient filter is to separate the intrinsic property and the extrinsic factor, and to remove the extrinsic factor. The analytical solutions of \(\rho(i_1, i_2)\) and \(l(i_1, i_2)\) can not be obtained from a single image. To solve the problem, self-quotient filter assumes that a smoothed version of an image has approximately the same illumination as the original one. In self-quotient filter, we first calculate the following equation:
\[
z(i_1, i_2) = \frac{x(i_1, i_2)}{\Psi[x(i_1, i_2)]},
\]
where \(x(i_1, i_2)\) is the original image and \(\Psi\) is the smoothing function. \(z(i_1, i_2)\) is then binarized to extract face features from the images. When the quantization bit of the image is 8 bits, The binarized image \(q(i_1, i_2)\) is described as follows:
4. Self-Quotient $\varepsilon$-Filter

In this section, we describe the self-quotient $\varepsilon$-filter (SQEF). A simple idea to solve the noise influence in self-quotient filter is to use two smoothed filters instead of original image as follows:

$$z(i_1, i_2) = \frac{\Psi_1[x(i_1, i_2)]}{\Psi_2[x(i_1, i_2)]}, \quad (12)$$

$\Psi_1$ and $\Psi_2$ should be different because the output always becomes 1 if $\Psi_1$ and $\Psi_2$ are the same smoothed filter.

However, even if we design SQF by using two different smoothed filters, some problems still remain. For instance, when we consider Gaussian noise corrupted image, not only the noise is smoothed but also the texture and shape are blurred by the smoothed filter. When we consider impulse noise corrupted image, the smoothing effects for impulse noise of two smoothed filter are different. Hence, the influence from the impulse noise in SQF is still large when we handle the impulse noise corrupted image.

Hence, we need to employ alternative filters, which can reduce the noise effectively with preserving the texture and shape information instead of simple smoothed filter. The alternative filters should be simple to keep the simplicity of self-quotient filter.

Based on the above prospects, SQEF is designed as follows:

$$z(i_1, i_2) = \frac{\Phi_{\varepsilon_1}[x(i_1, i_2)]}{\Phi_{\varepsilon_2}[x(i_1, i_2)]}, \quad (13)$$

where $\Phi_{\varepsilon}$ represents $\varepsilon$-filter described as follows:

$$z(i_1, i_2) = \Phi_{\varepsilon}[I(i_1, i_2)] = x(i_1, i_2) + \sum_{j_1=-K}^{K} \sum_{j_2=-K}^{K} a(j_1, j_2) F(x(i_1 + j_1, i_2 + j_2) - x(i_1, i_2)), \quad (14)$$

where $a(j_1, j_2)$ represents the filter coefficient. $a(j_1, j_2)$ is usually constrained as follows:

$$\sum_{j_1=-K}^{K} \sum_{j_2=-K}^{K} a(j_1, j_2) = 1. \quad (15)$$

$F(x)$ is the nonlinear function described as follows:

$$|F(x)| \leq \varepsilon : -\infty \leq x \leq \infty, \quad (16)$$

where $\varepsilon$ is a constant number constrained as follows.

$$0 \leq \varepsilon. \quad (17)$$

It should be noted that calculation cost of $\varepsilon$-filter is small because it requires only switching and linear operation. $z(i_1, i_2)$ is then binarized as follows:

$$Q(i_1, i_2) = \begin{cases} 255 & \text{if } z(i_1, i_2) > Th \\ 0 & \text{if } z(i_1, i_2) \leq Th \end{cases}, \quad (18)$$
where \( Th \) represents a threshold value.

In \( \varepsilon \)-filter, \( \varepsilon \) is an essential parameter to reduce the noise appropriately. If \( \varepsilon \) is set to an excessively large value, the \( \varepsilon \)-filter becomes the same as linear filter. On the other hand, if \( \varepsilon \) is set to 0, it does nothing to reduce the noise anymore, that is, the filter output becomes the input image itself. Hence, SQF is a subset of SQEF. When we take into account the design of SQF, numerator in Eq. (13) should become similar to the original image, while denominator in Eq. (13) should become an smoothed image.

Although SQEF can extract the features from uniform noise corrupted image, Gaussian noise corrupted images and impulse noise corrupted images, it is considered that the reasons are different. When we handle uniform noise corrupted image and Gaussian noise corrupted images, \( \varepsilon \)-filters in SQEF smoothes the uniform noise or Gaussian noise. Due to the above reasons, SQEF could reduce the influence of the noise by self-quotient effect. On the other hand, when we handle impulse noise corrupted image, both \( \varepsilon \)-filters in SQEF ignore the impulse noise due to the feature of \( \varepsilon \)-filter. Hence, the impulse noise remains in both filter outputs. And therefore the effect from the impulse noise is reduced when we divide one filter output of \( \varepsilon \)-filter by the other filter output of \( \varepsilon \)-filter.

Due to the above reasons, when we handle uniform noise corrupted images and Gaussian noise corrupted images, \( \varepsilon_1 \) should be set to the value to the noise fluctuation to reduce the noise while preserving the signal in Eq. (13). A sufficient large \( \varepsilon_2 \) should be used to emphasize the feature of the image.

On the other hand, when we handle impulse noise corrupted images, \( \varepsilon_1 \) should be set to a sufficiently small value (or 0) to keep not only the signal but also noise. \( \varepsilon_2 \) should be set to the value between the noise fluctuation and signal fluctuation to smooth the signal while preserving the noise unlike handling uniform noise and Gaussian noise.

5. Experiment

5.1 Feature Extraction from Uniform Noise Corrupted Image

We first conducted the evaluation experiments on feature extraction from uniform noise corrupted images. To evaluate the filter characteristics of SQEF, we conducted the evaluation experiments using various types of facial images. Some facial images are selected from Yale image database [20] and facial parts are cut from them. The image size is 256 pixels \( \times \) 256 pixels. We added various levels of random noise with uniform distribution to the original image. The maximal intensity of noise changes from 10 to 30 with 10 intervals. When the maximal intensity of noise is \( J \), it means that the noise range is \([-J, J] \). The averages of the noise were set to 0. Throughout the experiments, the filter coefficient \( a_j \) is set to \( 1/(2K + 1)^2 \) to make it uniform weight. To test the robustness of the proposed method concerning the window size, the window size was changed from 3 \( \times \) 3 to 9 \( \times \) 9. We show the results when the window size was set to 5 \( \times \) 5 as examples. \( \varepsilon_1 \) is set to the same value as maximal intensity of noise, that is, 10, 20 and 30, respectively. This procedure aims to reduce the noise while preserving the signal described in the previous section. In fact, the \( \varepsilon_1 \), which minimizes root mean square error (RMSE) between the original image and filter output was 11, 22 and 32, respectively when we handle the image with uniform noise whose maximal intensity is 10, 20 and 30, respectively. To simplify the experiments, \( \varepsilon_2 \) is set to 255, that is, a simple linear filter. Similar results could be obtained throughout all the experiments regardless of the window size. Figures 3, 4, and 5 show the experimental results when we used the image with noise whose intensity was 10, 20 and 30, respectively. Figures 3 (c), 4 (c), and 5 (c) show the filter outputs of SQF when we used the input image with uniform noise whose intensity is 10, 20 and 30, respectively. Figures 3 (d), 4 (d), and 5 (d) show the filter outputs of SQEF when we used the input image with uniform noise whose intensity is 10, 20 and 30, respectively. As shown in Figs. 3, 4 and 5, SQEF can extract the shape and texture information with smoothing the noise while SQF can not extract the feature clearly.

5.2 Feature Extraction from Gaussian Noise Corrupted Image

We second conducted the evaluation experiments on feature extraction from Gaussian noise corrupted images. The same images as the previous experiment were utilized in the experiment. We added three types of Gaussian noise whose standard deviation is 10, 20 and 30. The averages of the noise were set to 0. Throughout the experiments, the filter coefficient \( a_j \) is set to \( 1/(2K + 1)^2 \) to make it uniform weight. To test the robustness of the proposed method concerning the window size, the window size was changed from 3 \( \times \) 3 to
9 × 9. We show the results when the window size was set to 5 × 5 as examples. \( \epsilon_1 \) was set to the same value as standard deviation, that is, 10, 20 and 30, respectively. This is because standard deviation represents the fluctuation of noise. In fact, the \( \epsilon_1 \), which minimizes RMSE between the original image and filter output was 8, 16 and 27, respectively when the standard deviation was 10, 20 and 30, respectively. To simplify the experiments, \( \epsilon_2 \) was set to 255, that is, the filter was set to a simple linear filter. Similar results could be obtained throughout all the experiments regardless of the window size.

Figures 6, 7, and 8 show the experimental results when we used the image with noise whose standard deviation was 10, 20 and 30, respectively. Figures 6 (a), 7 (a) and 8 (a) show the original image for comparison. Figures 6 (b), 7 (b) and 8 (b) show the input image with Gaussian noise whose standard deviation is 10, 20 and 30, respectively. Figures 6 (c), 7 (c), and 8 (c) show the filter outputs of SQF when we used the input image with noise whose intensity is 10, 20 and 30, respectively. Figures 6 (d), 7 (d), and 8 (d) show the filter outputs of SQEF when we used the input image with Gaussian noise whose standard deviation is 10, 20 and 30, respectively.
and 30, respectively. As shown in Figs. 6, 7 and 8, SQEF can extract the shape and texture information with smoothing the noise while SQF can not extract the feature clearly.

5.3 Feature Extraction from Impulse Noise Corrupted Image

We also conducted the evaluation experiments on feature extraction from impulse noise corrupted images. As well as the previous experiments, some facial images are selected from Yale image database [20] and facial parts are cut from them. The image size is 256 pixels $\times$ 256 pixels. We added 10%, 20% and 30% impulse noise to images, respectively. Throughout the experiments, the filter coefficient $a_i$ is set to $1/(2K+1)^2$ to make it uniform weight. To test the robustness of the proposed method concerning the window size, the window size was changed from $3 \times 3$ to $9 \times 9$. We show the results when the window size was set to $5 \times 5$ as examples. $\varepsilon_1$ was set to 0, that is, we use the original images. $\varepsilon_2$ was set to the same value as the average of the input signal, that is, 82, 101, and 121, respectively. This is to smooth the original image except impulse noise. Similar results could be obtained throughout all the experiments regardless of the window size.

Figures 9, 10, and 11 show the experimental results when we used the image with impulse noise whose percentage was 10%, 20% and 30%, respectively. Figures 9(a), 10(a) and 11(a) show the original image for comparison.

Figures 9(b), 10(b) and 11(b) show the input image with impulse noise whose percentage is 10%, 20% and 30%, respectively. Figures 9(c), 10(c), and 11(c) show the filter outputs of SQF when we used the input image with noise whose percentage is 10%, 20% and 30%, respectively. Figures 9(d), 10(d), and 11(d) show the filter outputs of SQEF when we used the input image with noise whose percentage is 10%, 20% and 30%, respectively. As shown in Figs. 9, 10 and 11, SQEF can extract the shape and texture information with reducing the noise, while SQF can not extract the feature clearly.

5.4 Feature Extraction from Light Fluctuated Images

One of the feature in SQF is the robustness of shade fluctuated images. To clarify the feature of the robustness of shade fluctuated images, we show the example when we utilized shade fluctuated image with 30% impulsive noise.
As the light fluctuated images, we selected two images (\texttt{yaleB01P00A-070E-35.pgm} and \texttt{yaleB01P00A+095E+00.pgm}) from Yale image database. \(\epsilon_1\) was set to 0, that is, the original image was used. \(\epsilon_2\) was set to 115 and 94, which is the same values of the averages of input signal.

Figures 12 and 13 show the results when we used \texttt{yaleB01P00A-070E-35.pgm} and \texttt{yaleB01P00A+095E+00.pgm} with 30\% impulse noise corrupted image, respectively. Figures 12(a) and 13(a) show the original images of \texttt{yaleB01P00A-070E-35.pgm} and \texttt{yaleB01P00A+095E+00.pgm}, respectively. Figures 12(b) and 13(b) show the noisy images (\texttt{yaleB01P00A-070E-35.pgm} and \texttt{yaleB01_}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig11.png}
\caption{Fig. 11 Experimental results when we added 30\% impulse noise.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig12.png}
\caption{Fig. 12 Experimental results when we added 30\% impulse noise to \texttt{yaleB01P00A-070E-35.pgm}.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig13.png}
\caption{Fig. 13 Experimental results when we added 30\% impulse noise to \texttt{yaleB01P00A+095E+00.pgm}.}
\end{figure}

P00A+095E+00.pgm) corrupted with 30\% impulse noise. Figures 12(c) and 13(c) show the filter outputs of SQF applying to impulse noise corrupted images (\texttt{yaleB01P00A-070E-35.pgm} and \texttt{yaleB01P00A+095E+00.pgm}), respectively. Figures 12(d) and 13(d) show the filter outputs of SQEF applying to impulse noise corrupted images (\texttt{yaleB01P00A-070E-35.pgm} and \texttt{yaleB01P00A+095E+00.pgm}), respectively. As shown in Figs. 12 and 13, SQEF could extract the feature from noise corrupted images even when the images have light fluctuation, while SQF could not extract the feature from the noise corrupted images.

5.5 Samples of Other Face Images

We also show some examples using other face images to show the efficiency of the proposed method under the various conditions. We selected \texttt{yaleB10P00A-110E+65.pgm} and \texttt{yaleB02P01A+110E-20.pgm} from Yale image database. Figures 14 and 15 show the results when we used \texttt{yaleB10P00A-110E+65.pgm} and \texttt{yaleB02P01A+110E-20.pgm} with 30\% impulse noise corrupted image, respectively. \(\epsilon_1\) was set to 0, that is, the original image was used. \(\epsilon_2\) was set to 93 and 94, which is the same values of the averages of input signal.

Figures 14(a) and 15(a) show the original images of \texttt{yaleB10P00A-110E+65.pgm} and \texttt{yaleB02P01A+110E-20.pgm}, respectively. Figures 14(b) and 15(b) show the noisy images (\texttt{yaleB10P00A-110E+65.pgm} and \texttt{yaleB02P01A+110E-20.pgm}) corrupted with 30\% impulse noise. Figures 14(c) and 15(c) show the filter outputs of SQF applying to impulse noise corrupted images (\texttt{yaleB10P00A-110E+65.pgm} and \texttt{yaleB02P01A+110E-20.pgm}), respectively. Figures 14(d) and 15(d) show the filter outputs of SQEF applying to impulse noise corrupted images
5.6 Robustness for Noise Change

It is considered that the important feature of SQEF is robustness for detecting the features not only from the original image but also from the noise corrupted images. Hence, as an objective evaluation, we compare SQEF to SQF with regard to the robustness for noise change. In the experiments, we first filtered original image and noise corrupted image by using SQF and SQEF with the same parameters. We then calculated RMSE between the filter output of original image and that of the noise corrupted image. Figure 16 shows RMSE between the filter output of the original image and that of uniform noise corrupted image when we employed SQF and SQEF, respectively. The maximal intensity of uniform noise changed from 10 to 30 with 10 intervals. $\varepsilon_1$ and $\varepsilon_2$ were set to 30 and 255, respectively. Figure 17 shows RMSE between the filter output of the original image and that of Gaussian noise corrupted image when we employed SQF and SQEF, respectively. The standard deviation of Gaussian noise changed from 10 to 30 with 10 intervals.
$\varepsilon_1$ and $\varepsilon_2$ were set to 30 and 255, respectively. Figure 18 shows RMSE between the filter output of the original image and that of impulse noise corrupted image when we employed SQF and SQEF, respectively. Noise percentage of impulse noise changed from 10% to 30% with 10% intervals. $\varepsilon_1$ and $\varepsilon_2$ were set to 0 and 121 respectively. As shown in Figs. 16, 17 and 18, the filter outputs of SQEF when we used noise corrupted images are similar to that of SQEF when we used the original image compared to SQF. In other words, we could obtain the robust filter outputs regardless of noise existence when we employed SQEF.

5.7 Calculation Cost

We also conducted the experiments to compare the calculation cost to other methods. Three different size of images (128 pixels $\times$ 128 pixels, 256 pixels $\times$ 256 pixels and 512 pixels $\times$ 512 pixels) were utilized for testing. The window size of SQEF was set to $3 \times 3$. For comparison, we employed some types of SQF. We used a computer with an Intel Core 2 Duo 1.58 GHz CPU. The programs were implemented by MATLAB. The first SQF is the simplest one, which uses an original image and the image smoothed by linear filter. The window size of linear filter is set to $3 \times 3$. The second SQF is the filter, which uses an original image and the image smoothed by Gaussian filter instead of linear filter. The window size of Gaussian filter is set to $3 \times 3$. The third SQF is the alternative approach mentioned in the paper, i.e.,

$$z(i_1, i_2) = \frac{\Psi_1[x(i_1, i_2)]}{\Psi_2[x(i_1, i_2)]}, \tag{19}$$

where $\Psi_1$ and $\Psi_2$ are linear filters. The window sizes of $\Psi_1$ and $\Psi_2$ were set to $3 \times 3$ and $5 \times 5$ to make $\Psi_1$ and $\Psi_2$ different. The forth SQF is the same filter as the third one except using Gaussian filters instead of linear filters. Figure 19 shows the experimental results. Horizontal axis and vertical axis in Fig. 19 represent the image size used in the experiment and calculation time, respectively. As shown in Fig. 19, although the simplest SQF is the fastest of all, the calculation cost of SQEF is smaller than the other SQFs.

6. Discussions and Conclusion

In this paper, we introduced the algorithm of self-quotient $\varepsilon$-filter and showed the applications of a self-quotient $\varepsilon$-filter to uniform noise corrupted images, Gaussian noise corrupted images and impulse noise corrupted images. We reported some experimental results and confirmed the effectiveness of the proposed method. The algorithm is simple and calculation cost is small. It can extract face features not only from the clear images but also from the noise corrupted images with Gaussian noise and impulse noise.

In our method, parameter setting is an important factor to obtain the adequate filtering. As described in the experiments, when we handle uniform noise corrupted image and Gaussian noise corrupted images, it is considered that noise fluctuation is strongly related to the optimal parameter. It may be useful to use signal-noise decorrelation criterion [21] to obtain the adequate parameters. This algorithm assumes signal-noise decorrelation and employs correlation coefficient of the filter output and the difference between the input and the filter output as the evaluation function of the parameter setting of $\varepsilon$-filter. We confirmed that the adequate $\varepsilon$ could be obtained in some examples. However, as the obtained $\varepsilon$ values were not always optimal due to some reasons, the approach is not a panacea for parameter setting. Parameter setting problem needs further research.

For future works, we would like to apply the proposed method to face recognition system. We also would like to apply it to medical images to extract disease site and employ it for non-photorealistic rendering. Theoretical analyses of SQEF should also be studied.

Acknowledgement

This research was supported by the research grant of Support Center for Advanced Telecommunications Technology Research (SCAT), by the research grant of Foundation for the Fusion of Science and Technology, by Special Coordination Funds for Promoting Science and Technology, and by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (B), 20700168, 2008.

References


