Further Properties and a Fast Realization of the Iterative Truncated Arithmetic Mean Filter

Zhenwei Miao, *Student Member, IEEE*, and Xudong Jiang, *Senior Member, IEEE*

***Abstract*—The iterative truncated arithmetic mean (ITM) filter has been recently proposed. It possesses merits of both the mean and median filters. In this brief, the Cramer–Rao lower bound is employed to further analyze the ITM filter. It shows that this filter outperforms the median filter in attenuating not only the short-tailed Gaussian noise but also the long-tailed Laplacian noise. A fast realization of the ITM filter is proposed. Its computational complexity is studied. Experimental results demonstrate that the proposed algorithm is faster than the standard median filter.**

***Index Terms*—Computational complexity, iterative truncated arithmetic mean (ITM) filter, median filter, noise suppression, nonlinear filter.**

I. Introduction

COMPARED with the mean or weighted mean filters [1], the median filter [2] has better performance in impulsive noise suppression and edge preservation. However, it destructs image details and cannot effectively suppress Gaussian noise. To tackle the problem of the detail destruction, multistage median filters [3], truncation filters [4], and various noise adaptive median filters were proposed. Effort was also devoted to attenuate both long- and short-tailed noise with edge preservation. Most of them, such as the mean–median filter and the α-trimmed mean $(\alpha\mathrm{T})$ filter [5], make a compromise between the mean and median filters by using both the arithmetic computing and the time-consuming data sorting operations [6]. However, the compromise may not be well adjusted to different kinds of images and noise.

The iterative truncated arithmetic mean (ITM) filter [7] has been recently proposed. It iteratively truncates the extreme values of samples in the filter window to a dynamic threshold. This threshold guarantees that the filter output converges to the median of the input samples. A proper stop criterion enables the ITM filter owning merits of both the arithmetic mean and the order-statistical median operations. Both edge preservation and noise attenuation can be achieved within just a few iterations. The ITM filter outperforms the median filter in attenuating the single type of noise, such as Gaussian and Laplacian noise, and the mixed type of noise, such as the mixed Gaussian and impulsive noise. It also offers a way to estimate the median by a simple arithmetic computing algorithm.

Manuscript received April 25, 2012; revised July 14, 2012; accepted August 26, 2012. Date of publication October 26, 2012; date of current version January 4, 2013. This brief was recommended by Associate Editor Y. Yu.

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: mi0001ei@ntu.edu.sg; exdjiang@ntu.edu.sg).

Color versions of one or more of the figures in this brief are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2012.2218473

Using the mean absolute error as a criterion, it is shown in [7] that the ITM filter outperforms the median filter in attenuating Laplacian noise. This opens to doubt because the median is the optimum location estimator of Laplacian noise in the sense of maximum-likelihood estimation (MLE). Therefore, in this brief, we use the Cramer–Rao lower bound (CRLB) [8] and the mean square error (MSE) to put it beyond doubt that the ITM filter outperforms the median filter in attenuating Laplacian noise. As the ITM filter employs an iterative algorithm in the filtering process, people tend to think its computational complexity must be much higher than the median filter. This brief demonstrates that it is not the case. We show that, compared with the standard median filter, the ITM filter is faster for small filter size but slower for large filter size. Therefore, we propose a fast ITM (FITM) filter, which is faster than the standard median filter for all filter sizes.

II. Noise Suppression Properties of the ITM Filter

Given a set of n samples $\mathbf{x_0} = \{x_i\}$ in the filter window, the ITM algorithm [7], starting from $\mathbf{x} = \mathbf{x}_0$, iteratively truncates the extreme values of samples in $\mathbf{x}$ to a dynamic threshold τ.

Algorithm 1: Truncation Procedure of the ITM Filter
Input: $\mathbf{x_0} \Rightarrow \mathbf{x}$; **Output**: Truncated $\mathbf{x}$;
1 **do**
2 Compute the sample mean: $\mu = mean(\mathbf{x})$;
3 Compute the dynamic threshold: $\tau = mean(|\mathbf{x} - \mu|)$;
4 $b_l = \mu - \tau$, $b_u = \mu + \tau$, and truncate $\mathbf{x}$ by

$$x_i = \begin{cases} b_u, & \text{if } x_i > b_u \\ b_l, & \text{if } x_i < b_l; \end{cases}$$

5 **while** *the stopping criterion S is violated*;

The stop criterion S proposed in [7] is composed of four termination rules. In general, it terminates the iteration if the truncated mean is close to the median or has little change. It utilizes some relationships of the numbers of samples in different conditions to terminate the iteration.

The type-I output of the ITM filter [7] is

$$\textbf{ITM1}: \quad y_{t1} = mean(\mathbf{x})$$

where $\mathbf{x}$ is the truncated data set output from Algorithm 1. As a necessary preliminary of the study, two properties of the ITM filter are presented as follows.

Property 1: The distribution of the ITM output is symmetric, if the samples of the input data set $\mathbf{x_0} = \{x_1, x_2, \ldots, x_n\}$

1549-7747/$31.00 © 2012 IEEE

TABLE I
CRLB AND MSE OF SAMPLE MEAN μ AND MEDIAN ϕ

	CRLB	MSE (μ)	MSE (ϕ)
Gaussian noise	σ^2/n	σ^2/n	$\frac{\pi}{2(n+2)}\sigma^2$
Laplacian noise	$\sigma^2/(2n)$	σ^2/n	$\sigma^2 \sum_{r=0}^{m} \frac{K(m,r)}{(m+r+1)^2}$

are independent identically distributed (i.i.d.) samples of the random variable X with a symmetric distribution around the symmetry center a.

Proof: If x_i is symmetric around a, $2a - x_i$ has the same distribution as x_i. Thus, the ITM output $y_{t1}(x_1, x_2, \ldots, x_n)$ has the same distribution as $y_{t1}(2a - x_1, 2a - x_2, \ldots, 2a - x_n)$, which is equal to $2a - y_{t1}(x_1, x_2, \ldots, x_n)$ because the ITM filter is invariant to scale and shift [7]. It follows that the distribution of y_{t1} is symmetric around a. ■

Property 2: The output of the ITM filter is an unbiased estimate of the population mean of X, if the samples in $\mathbf{x_0} = \{x_1, x_2, \ldots, x_n\}$ are i.i.d. samples of the random variable X with a symmetric distribution around the symmetry center a.

Proof: As x_i is symmetrically distributed around a, $E\{X\} = a$. Similarly, $E\{y_{t1}\} = a$ according to Property 1. Therefore, $E\{y_{t1}\} = E\{X\}$. This completes the proof of Property 2. ■

Here, we analyze why the ITM1 filter (a) cannot be better than the mean filter for Gaussian noise and (b) can be superior to the median filter for Laplacian noise even though the median is the optimum location estimator of Laplacian noise in the sense of MLE. As Property 2 shows that the output of the ITM filter is unbiased, the analysis is based on the CRLB [8], which provides a lower bound of the MSE of unbiased estimators.

A. Gaussian Noise

The probability density function (pdf) of Gaussian noise X with the population mean μ_o and the standard deviation σ is

$$f_G(X = x|\mu_o, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu_o)^2}{\sigma^2}\right). \quad (1)$$

As shown in Table I, the CRLB for the estimation of μ_o is σ^2/n. The MSE of the sample mean μ is σ^2/n [9], which is equal to the CRLB. Thus, μ is the minimum MSE estimator, which means that no estimator can outperform μ in estimating signals in Gaussian noise in terms of MSE. For large n, the MSE of the sample median ϕ approximates $\text{MSE}(\phi) = \pi/(2(n+2))\sigma^2$ [9]. It is about $\pi/2$ times of the CRLB.

It is difficult to get a mathematical expression of the MSE of the ITM1 output. Here, we use the Monte Carlo simulations [9] to evaluate the ITM1 filter. 10^6 independent input data sets are used in the simulations. As the αT filter approaches the mean if $\alpha \to 0$ and approaches the median if $\alpha \to 0.5$, $\alpha = 0.25$ is chosen in this brief. Fig. 1(a) shows the normalized MSE of the ITM1 output against the number of iterations. The filter size is $n = 49$. When the number of iterations is zero, the ITM1 output is equal to the mean. Its MSE is approximately equal to the CRLB. By increasing the number of iterations, the ITM1 output approaches the median, and its MSE increases and approaches that of the median. Fig. 1(b) shows the normalized MSE against the filter size. The ITM1 filter employs the default stop criterion in [7]. We see that both the ITM1 and αT filters significantly outperform the median filter.

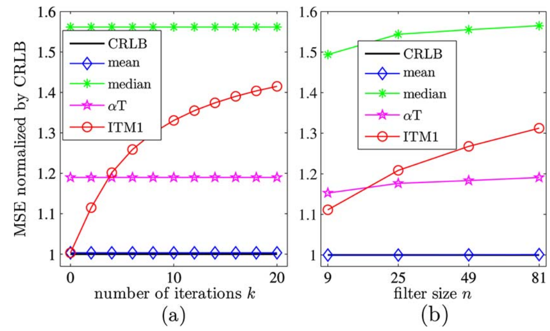

Fig. 1. Normalized MSE against (a) the number of iterations k and (b) the filter size n for Gaussian noise.

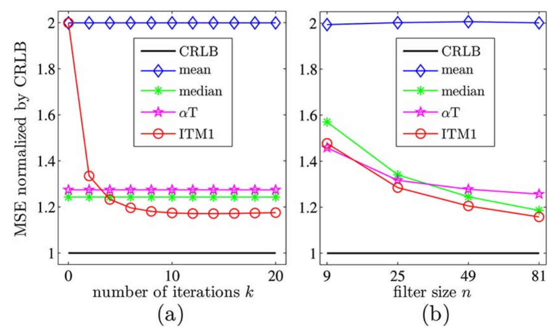

Fig. 2. Normalized MSE against (a) the number of iterations k and (b) the filter size n for Laplacian noise.

B. Laplacian Noise

The pdf of Laplacian noise X is

$$f_L(X = x|\mu_o, b) = \frac{1}{2b} \exp\left(-\frac{|x - \mu_o|}{b}\right) \quad (2)$$

where μ_o is the population mean of X and b is a scale parameter. Its variance is $\sigma^2 = 2b^2$. Table I shows its CRLB is $\sigma^2/(2n)$. The MSE of the sample mean μ is two times of the CRLB [10]. Therefore, μ is ineffective in estimating signals in Laplacian noise. The MSE of the sample median ϕ is

$$\text{var}(\phi) = 2b^2 \sum_{r=0}^{m} \frac{K(m,r)}{(m+r+1)^2} \quad (3)$$

where $K(m,r) = (-1)^r(2m+1)!/(m!r!(m-r)!(m+r+1)2^{m+r})$ and $m = (n-1)/2$ [11]. When the filter size n is small, the MSE of ϕ is far away from the CRLB. For example, when $n = 9$, $\text{var}(\phi) \approx 0.175b^2$. It is about 1.58 times of the CRLB. From this, we can conclude that ϕ is not the minimum MSE estimator for small filter size, although it is the MLE [12]. Therefore, it is not a surprise that the ITM1 filter can outperform the median filter even for the long-tailed Laplacian noise.

The performance of the ITM1 filter is analyzed based on the Monte Carlo simulations with 10^6 independent input data sets. Fig. 2(a) shows the MSE of the ITM1 output against the number of iterations. The filter size is $n = 49$. When the number of iterations is zero, the ITM1 output is equal to the mean. Its MSE is two times of the CRLB. After a few iterations, the MSE of the ITM1 output becomes smaller than that of the median. The MSE of the ITM1 output approaches that of the median when the number of iterations is large enough. The normalized MSE against the filter size is shown in Fig. 2(b). The αT filter is superior to the median filter when $n \leq 25$ but inferior when $n > 25$. The ITM1 filter uses the stop criterion in [7]. This stop criterion is a general criterion that is applied in all experiments in [7]. Although this stop criterion is not optimized for Laplacian noise, Fig. 2(b) shows that the ITM1 filter still outperforms the median filter.

III. Proposed FITM Filter

From the ITM algorithm (Algorithm 1), we see that all samples are visited in all iterations. This becomes a heavy burden when the filter size and the number of iterations are large. In order to reduce the computational burden, we propose an FITM filter by only visiting the untruncated samples in each iteration. This is enabled by the following proposition.

Proposition 1: Samples, once being truncated in an iteration of the ITM algorithm, must be truncated in all subsequent iterations.

Proof: Let $\mathbf{x}_h = \{x_i | x_i > \mu\}$, n_h be the number of the samples in $\mathbf{x}_h$, and $\delta_h = sum(\mathbf{x}_h - \mu)/n_h$ in the kth iteration. Let x_{i+}, μ_+, τ_+, n_{h+}, and δ_{h+} be the corresponding notations in the $(k+1)$th iteration.

Assume a sample x_{iu} is truncated to the upper bound $\mu + \tau$ in the kth iteration. Obviously, $x_{iu+} = \mu + \tau$, which has the maximum value in $\mathbf{x}$ in the $(k+1)$th iteration. As τ monotonically decreases [7], $\tau_+ < \tau$, we have

$$x_{iu+} > \mu_+ + \tau_+, \qquad \text{if } \mu_+ \leq \mu. \tag{4}$$

In the case of $\mu_+ > \mu$, from $\tau = 2n_h\delta_h/n$ [7], we have

$$\tau = \frac{2}{n}\sum_{x_i>\mu}(x_i - \mu) \geq \frac{2}{n}\sum_{x_i>\mu_+}(x_i - \mu_+ + \mu_+ - \mu). \tag{5}$$

As at least one sample x_{iu} is truncated to the upper bound in the kth iteration, we have

$$\frac{2}{n}\sum_{x_i>\mu_+}(x_i - \mu_+) > \frac{2}{n}\sum_{x_{i+}>\mu_+}(x_{i+} - \mu_+) = \tau_+. \tag{6}$$

Substituting (6) into (5) yields

$$\tau > \tau_+ + \frac{2n_{h+}}{n}(\mu_+ - \mu). \tag{7}$$

As $\tau_+ = 2n_{h+}\delta_{h+}/n$ [7], (7) becomes

$$\tau > \tau_+ + \frac{\tau_+}{\delta_{h+}}(\mu_+ - \mu). \tag{8}$$

As $\delta_{h+} \leq x_{iu+} - \mu_+$, we have $\delta_{h+} \leq \tau_+$, if

$$x_{iu+} - \mu_+ \leq \tau_+. \tag{9}$$

Therefore, under the condition of equation (9), equation (8) becomes

$$\tau > \tau_+ + \mu_+ - \mu. \tag{10}$$

Since $x_{iu+} = \mu + \tau$, equation (10) becomes

$$x_{iu+} > \mu_+ + \tau_+. \tag{11}$$

The conclusion (11) conflicts with (9). Hence, the condition (9) is not true, which means

$$x_{iu+} > \mu_+ + \tau_+, \qquad \text{if } \mu_+ > \mu. \tag{12}$$

From (4) and (12), we have

$$x_{iu+} > \mu_+ + \tau_+. \tag{13}$$

In the same way, we can prove that if a sample x_{il} is truncated to the lower bound $\mu - \tau$ in the kth iteration

$$x_{il+} < \mu_+ - \tau_+. \tag{14}$$

Inequalities (13) and (14) prove Proposition 1. ■

Proposition 1 shows that all truncated samples must be truncated in the subsequent iterations. In other words, all truncated samples have the same values of either the lower or upper bound in all subsequent iterations. Therefore, we do not need to access such samples one by one. There is also no need to remember the positions of the truncated pixels. We only need to count the number of such samples and replace them by the constant $\mu - \tau$ or $\mu + \tau$ in all subsequent iterations. This leads to the FITM algorithm, which speeds up the truncation procedure by only visiting the untruncated samples. Let $n_{\tau l}$ and $n_{\tau u}$ be the numbers of the samples smaller than the lower bound and larger than the upper bound, respectively. The proposed FITM algorithm is shown as follows.

Algorithm 2: Truncation Procedure of the FITM Filter

Input: $\mathbf{x_0} \Rightarrow \mathbf{x}$, $n_{\tau l} = 0$, $n_{\tau u} = 0$; **Output**: $\mathbf{x}$, b_l, b_u, $n_{\tau l}$, and $n_{\tau u}$;

1 **do**

2 $\mu = (sum(\mathbf{x}) + n_{\tau l}b_l + n_{\tau u}b_u)/n$;

3 $\tau = (sum(|\mathbf{x} - \mu|) + n_{\tau l}(\mu - b_l) + n_{\tau u}(b_u - \mu))/n$;

4 $b_l = \mu - \tau$, $b_u = \mu + \tau$, $\mathbf{x} = \{x_i | b_l \leq x_i \leq b_u\}$, and update $n_{\tau l}$ and $n_{\tau u}$;

5 **while** *the stopping criterion S is violated*;

Comparing Algorithm 2 with Algorithm 1, we can find that both μ and τ computed in these two algorithms are the same. Therefore, the ITM and FITM filters have the same outputs. The difference is that in steps 2–4, Algorithm 2 only visits the untruncated samples, whereas Algorithm 1 visits all the samples in each iteration. This modification, enabled by Proposition 1, speeds up the ITM algorithm.

IV. Computational Complexity

The computational complexity of the ITM and FITM filters can be measured by the times that all the samples are visited in the iterations. The visiting times are determined by two factors: a) the number of iterations N_s and b) the probability p_k of a sample being visited in the kth iteration. The number of iterations N_s for both the ITM and FITM filters are the same because they utilize the same default stop criterion in [7]. The probability p_k is different for these two filters. For the ITM filter, $p_k = 1$ because all the samples are visited in each iteration. For the FITM filter, only the untruncated samples are visited. Therefore, p_k monotonically decreases against the number of iterations.

We use the Monte Carlo simulations [9] to analyze the number of iterations N_s. Three types of noise, namely, Gaussian, Laplacian, and the uniform distributed noise, are employed. 10^6 independent input data sets are used in each experiment. The experimental results in Fig. 3 illustrate that the numbers of iterations, which are determined by the default stop criterion, of different noise types are approximately the same. Fig. 3 shows

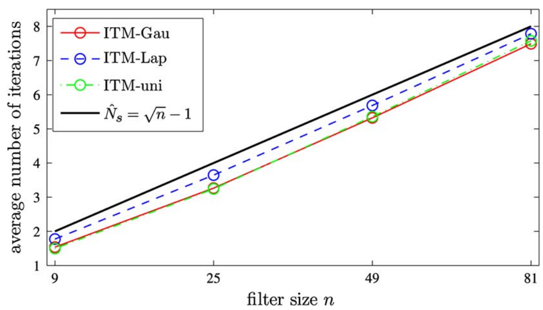

Fig. 3. Average number of iterations determined by the stop criterion in [7] against the filter size n.

that N_s is approximately a linear function of $\sqrt{n}$. Therefore, we use

$$\hat{N}_s = \sqrt{n} - 1 \tag{15}$$

as an upper bound of N_s, which is plotted in Fig. 3.

For the ITM filter, as the probability of a sample being visited in the kth iteration is $p_k = 1$, the total visiting times of all the samples is $\sum_{k=1}^{\hat{N}_s} np_k = n(\sqrt{n} - 1)$. Its computational complexity is $\mathcal{O}(n(\sqrt{n} - 1)) = \mathcal{O}(n\sqrt{n})$.

The FITM filter only visits the untruncated samples in each iteration. As the dynamic threshold τ_k monotonically decreases [7], the probability of a sample within the range $(\mu_{k-1} - \tau_{k-1}, \mu_{k-1} + \tau_{k-1})$ decreases. Therefore, the probability of a sample being visited p_k decreases. In order to simplify the analysis of the probability p_k, we employ the uniform distributed noise as an example. The pdf of a uniform distributed random variable X is

$$f_u(X = x) = \begin{cases} 1, & \text{if } -0.5 \leq x \leq 0.5 \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

From (16), we find the following lemma of τ_k.

Lemma 1: When the filter size n is sufficiently large, the dynamic threshold τ_k of X drawn from the uniform distribution (16) has a recurrence relation, i.e.,

$$\tau_k = \tau_{k-1}(1 - \tau_{k-1}), \qquad k > 1 \tag{17}$$

with $\tau_1 = 0.25$.

Proof: When the filter size n is sufficiently large, the sample mean is equal to the expectation of X, as $\mu = E[X] = 0$. The dynamic threshold of the first iteration is

$$\tau_1 = \frac{1}{n}\sum_{i=1}^{n} |x_i - \mu| = E[|X|] = 0.25. \tag{18}$$

After the $(k-1)$th iteration, X can be either untruncated or truncated. Only the samples within the range $(\mu_{k-1} - \tau_{k-1}, \mu_{k-1} + \tau_{k-1})$ are untruncated. Thus, the probability of a sample untruncated is $2\tau_{k-1}$, and the probability of truncation is $1 - 2\tau_{k-1}$. As the output of the FITM filter is unbiased, $\mu_{k-1} = 0$. The deviation of an untruncated sample from the mean is $|x|$, and that of a truncated sample is τ_{k-1}. Therefore, the dynamic threshold of the kth iteration τ_k is

$$\begin{aligned} \tau_k &= E[|X_{k-1}|] = (1 - 2\tau_{k-1})\tau_{k-1} + \int_{-\tau_{k-1}}^{\tau_{k-1}} |x| dx \\ &= \tau_{k-1}(1 - \tau_{k-1}) \end{aligned} \tag{19}$$

where X_{k-1} is the random variable X after the $(k-1)$th iteration. Equations (18) and (19) complete the proof of (17). ∎

Since τ_k has the property in Lemma 1, the summation of τ_k is constrained by the following lemma.

Lemma 2: The summation of the dynamic threshold τ_k specified by (17) is bounded by two logarithm curves, i.e.,

$$0.5\left(\ln\left(0.5m + 1\right)\right) < \sum_{k=1}^{m} \tau_k < \ln(m + 1). \tag{20}$$

Proof: For $k = 1$, we have $\tau_k = 0.25 < 1/(k + 1)$. For $k > 1$, we will prove $\tau_k < 1/(k + 1)$ is true under the assumption of $\tau_{k-1} < 1/(k - 1 + 1)$. Let $t = 1/\tau_{k-1}$, and (17) yields $\tau_k = (t - 1)/t^2$. Since $t > k$, we have

$$\tau_k = \frac{t - 1}{t^2} < \frac{t - 1}{t^2 - 1} = \frac{1}{t + 1} < \frac{1}{k + 1}. \tag{21}$$

This proves that

$$\tau_k < 1/(k + 1), \qquad k \geq 1. \tag{22}$$

Therefore

$$\sum_{k=1}^{m} \tau_i < \sum_{k=1}^{m} \frac{1}{k + 1} < \int_{1}^{m+1} \frac{1}{x} dx = \ln(m + 1). \tag{23}$$

In the analogous way, it can be proved that

$$\tau_k \geq 1/\left(2(k + 1)\right), \qquad k \geq 1. \tag{24}$$

Therefore

$$\sum_{k=1}^{m} \tau_k \geq \sum_{k=1}^{m} \frac{1}{2(k+1)} > \int_{2}^{m+2} \frac{1}{2x} dx = 0.5\ln(0.5m+1). \tag{25}$$

Inequalities (23) and (25) complete the proof of (20). ∎

As only the untruncated samples are visited by the FITM algorithm, the probability of a sample being visited at the kth iteration is $p_k = 2\tau_{k-1}$, where we define $\tau_0 = 0.5$. Therefore, its computational complexity is $\mathcal{O}(\sum_{k=1}^{\hat{N}_s} np_k) = \mathcal{O}(2n\sum_{k=1}^{\hat{N}_s} \tau_{k-1})$. From (20), we can get that $\mathcal{O}(\sum_{k=1}^{\hat{N}_s} \tau_{k-1}) = \mathcal{O}(\ln(\hat{N}_s))$. Thus, the computational complexity of the FITM filter is $\mathcal{O}(2n\ln(\hat{N}_s)) = \mathcal{O}(2n\ln(\sqrt{n} - 1)) = \mathcal{O}(n\ln(n))$. It is smaller than that of the ITM filter and has the same order as the quick-sort algorithm.

The Monte Carlo simulations are also carried out to analyze the visiting times for the FITM filter when the filter size is not large enough. Experimental results in Fig. 4 illustrate that the average visiting times of a sample in the FITM filter is approximately a linear function of $\ln n$. Fig. 4 shows that

$$\hat{N}_{\text{FITM}} = 0.7\ln n \tag{26}$$

is a close upper bound for the FITM filter for $9 \leq n \leq 81$. Therefore, the total visiting times of all the samples for the FITM filter is about $0.7n\ln n$. It is smaller than that of the quick-sort algorithm, which is approximately equal to $2n\ln n$ [13]. The average visiting times for both the ITM and FITM filters are compared in Fig. 5. It is seen that the visiting times for the FITM filter are smaller than that for the ITM filter.

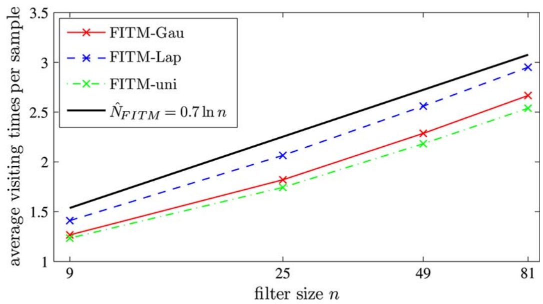

Fig. 4. Average visiting times of a sample against the filter size n. The x-axis is in log scale of n.

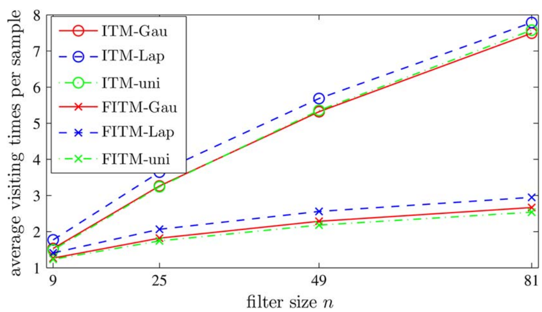

Fig. 5. Average visiting times of a sample against the filter size n. The x-axis is in linear scale of n.

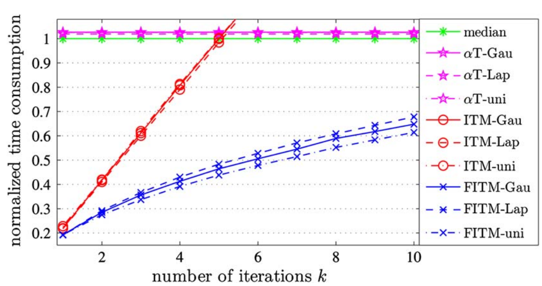

Fig. 6. Normalized time consumption against the number of iterations k. The time consumption is normalized by that of the median filter.

We further evaluate the computational complexity of the ITM and FITM filters in two experiments. These experiments are performed under the Windows 7 system with the Intel Core i5 CPU 3.2 GHz. All of the filters are implemented by C programming language. As data sorting is the basic building block that many rank order statistic filters, such as the popular αT filter, are built around, we implement both the median and αT filters using the quick-sort algorithm. The first experiment tests the running time of the filters against the number of iterations, and the second experiment tests that with the default stop criterion given in [7] against the filter size. The time consumption is normalized by that of the median filter. The normalized time consumption against the number of iterations is shown in Fig. 6. The filter size is $n = 49$. The time consumption for the ITM filter is a linear function of the number of iterations because all the samples are visited in each iteration. As the FITM filter only visits the untruncated samples, its time consumption slowly increases compared with that of the ITM filter. The ITM filter is faster than the median filter when the number of iterations $k \leq 5$ but slower for $k > 5$. The FITM filter is faster than the median filter for all the numbers of iterations in Fig. 6. The experimental results using the default stop criterion are shown in Fig. 7. As the αT filter requires both arithmetic computing and data sorting operations, its time consumption is larger than that of the median filter. Compared with the median filter, the ITM filter is faster for the filter size $n \leq 49$ but slower for $n > 49$. The proposed FITM filter is faster than both the ITM and median filters for all filter sizes.

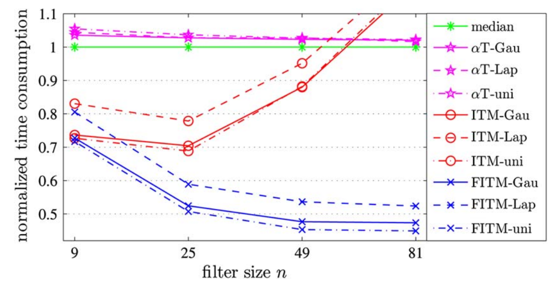

Fig. 7. Normalized time consumption against the filter size n. The time consumption is normalized by that of the median filter.

V. Conclusion

In this brief, some further properties of the ITM filter are analyzed. It shows that the ITM filter outperforms the median filter in dealing with both the short-tailed Gaussian noise and the long-tailed Laplacian noise. The computational complexity of the ITM filter is studied. It is $\mathcal{O}(n\sqrt{n})$. Experimental results show that the ITM filter is faster than the median filter when the filter size $n \leq 49$ but slower when $n > 49$. A fast implementation of the ITM filter is proposed. The computational complexity of the FITM filter is analyzed. The analysis reveals that the computational complexity of the FITM filter is $\mathcal{O}(n \ln n)$. Although it is of the same order as the median filter, experimental results demonstrate that the FITM filter is faster than the standard median filter implemented by the quick-sort algorithm.

References

[1] A. Buades, B. Coll, and J. M. Morel, "Nonlocal image and movie denoising," *Int. J. Comput. Vis.*, vol. 76, no. 2, pp. 123–139, Feb. 2008.
[2] I. Pitas and A. N. Venetsanopoulos, "Order statistics in digital image processing," *Proc. IEEE*, vol. 80, no. 12, pp. 1893–1921, Dec. 1992.
[3] D. Petrescu, I. Tabus, and M. Gabbouj, "L-M-S filters for image restoration applications," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1299–1305, Sep. 1999.
[4] X. D. Jiang, "Image detail-preserving filter for impulsive noise attenuation," *Proc. Inst. Elect. Eng.—Vis., Image Signal Process.*, vol. 150, no. 3, pp. 179–185, Jun. 2003.
[5] J. B. Bednar and T. L. Watt, "Alpha-trimmed means and their relationship to median filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 1, pp. 145–153, Feb. 1984.
[6] N. Zaric, N. Lekic, and S. Stankovic, "An implementation of the L-estimate distributions for analysis of signals in heavy-tailed noise," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 7, pp. 427–431, Jul. 2011.
[7] X. D. Jiang, "Iterative truncated arithmetic mean filter and its properties," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1537–1547, Apr. 2012.
[8] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[9] P. M. Narendra, "A separable median filter for image noise smoothing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-3, no. 1, pp. 20–29, Jan. 1981.
[10] T. H. Li and K. S. Song, "Estimation of the parameters of sinusoidal signals in non-Gaussian noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 62–72, Jan. 2009.
[11] O. J. Karst and H. Polowy, "Sampling properties of the median of a Laplace distribution," *Amer. Math. Monthly*, vol. 70, no. 6, pp. 628–636, Jun./Jul. 1963.
[12] G. R. Arce, *Nonlinear Signal Processing: A Statistical Approach*. New York: Wiley, 2004.
[13] E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms*. Potomac, MD: Computer Science Press, 1978.