Signal Processing 99 (2014) 147–158

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

# Additive and exclusive noise suppression by iterative trimmed and truncated mean algorithm

Zhenwei Miao *, Xudong Jiang

*School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Link, Singapore 639798, Singapore*

ARTICLE INFO

*Article history:*
Received 13 July 2013
Received in revised form
29 November 2013
Accepted 2 December 2013
Available online 10 December 2013

*Keywords:*
Median filter
Median approximation
Nonlinear filter
Noise suppression
Additive noise
Exclusive noise

ABSTRACT

An iterative trimmed and truncated arithmetic mean (ITTM) algorithm is proposed, and the ITTM filters are developed. Here, trimming a sample means removing it and truncating a sample is to replace its value by a threshold. Simultaneously trimming and truncating enable the proposed filters to attenuate the mixed additive and exclusive noise in an effective way. The proposed trimming and truncating rules ensure that the output of the ITTM filter converges to the median. It offers an efficient method to estimate the median without time-consuming data sorting. Theoretical analysis shows that the ITTM filter of size $n$ has a linear computational complexity $\mathcal{O}(n)$. Compared to the median filter and the iterative truncated arithmetic mean (ITM) filter, the proposed ITTM filter suppresses noise more effectively in some cases and has lower computational complexity. Experiments on synthetic data and real images verify the filter's properties.



## 1. Introduction

Noise suppression has drawn great attention and is used in a broad range of applications, such as imaging, communications, geology, hydrology and economics [1]. A noise corrupted signal can be modeled as

$$x_i = \begin{cases} s_i + v_i & \text{with probability } p \\ e_i & \text{with probability } (1-p), \end{cases} \tag{1}$$

where $s_i$, $v_i$ and $e_i$ denote the noise free signal, the additive and exclusive noise, respectively. The occurrence probability of the two types of noise is controlled by $p$, $p \in [0,1]$. The additive noise $v_i$ is in general symmetrically distributed with zero mean. It could be short- or long-tailed noise, such as Gaussian or Laplacian noise. The exclusive noise $e_i$ could be impulsive noise with uniform distribution, or pepper & salt noise. Great effort has been devoted to developing the noise suppression filters based on the noise model.

Many filters were designed to attenuate the additive noise that corresponds to $p=1$. The most frequently occurring noise in practice is the additive Gaussian noise, and the optimal filter in suppressing it is the mean filter. Its simplicity in realization and the availability of rigorous mathematical tool lead to the rich class of the linear finite impulse response (FIR) filters. The linear FIR filters are effective in attenuating the additive Gaussian noise but not the long-tailed noise. This results in the development of the nonlinear filters. The median filter [1], which is the most widely used one among the nonlinear filters, provides a powerful tool for signal processing. It has good properties in long-tailed noise suppression and structure preservation. However, it destructs fine signal details and cannot effectively suppress the additive Gaussian and other short-tailed noise. This leads to the various extensions of the median filter, including the weighted median filters [1], the weighted rank order Laplacian of Gaussian filter [2,3], the steerable weighted median filter [4], the fuzzy rank filters [5], the truncation filters [6] and various adaptive noise switching median filters [7–10]. The merits

* Corresponding author. Tel.: +65 9755 0569.
*E-mail addresses:* mi0001ei@e.ntu.edu.sg (Z. Miao), exdjiang@ntu.edu.sg (X. Jiang).

0165-1684/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.sigpro.2013.12.002

of the mean and median filters lead to another branch of filters which make compromises between these two filters, such as the mean–median (MEM) filter [11] and the median affine (MA) filter [12]. The output of these filters varies smoothly between the mean and median by adjusting some free parameters. Selecting the optimal parameters to make them adaptive to signal types is not an easy task though some efforts were made [12].

Both the additive and exclusive noise exist if $p$ in (1) is smaller than 1. As the exclusive noise completely replaces the original samples, the most effective way is to trim (remove) such samples and use the exclusive-noise-free ones in a local region to restore the signal. The median filter is optimal in suppressing the exclusive noise by trimming all the samples except the middle one. The filters to suppress the mixed additive and exclusive noise include the $\alpha$-trimmed mean ($\alpha$T) filter [13], the modified trimmed mean (MTM) filter [14] and the switching bilateral filter (SBF) [15]. The $\alpha$T filter discards some samples strictly relying on their rank. This may not be effective as it does not consider the dispersion of the data [12]. The MTM filter is sensitive to the small variation of samples located close to the threshold [12]. The SBF separates the impulsive noise from the Gaussian noise [15] and suppresses these two types of noise respectively. Its performance drops in dealing with the long-tailed noise. In addition, all these filters require both data sorting and arithmetic computing. Compared to the arithmetic computing, the data sorting is much more time-consuming [16].

Nonlinear filters without data sorting were desirable and proposed in [17–19]. The iterative truncated arithmetic mean (ITM) filter [17] iteratively truncates the extreme samples to a dynamic threshold that ensures the filter's output converges to the median. The stopping criterion of the ITM filter makes it own merits of both the arithmetic mean and order statistic median operations in attenuating the short- and long-tailed noise. By truncating the extreme samples, the ITM1 filter outperforms both the mean and median filters in suppressing Laplacian noise and the Gaussian–Laplacian mixed noise [17]. By discarding all the truncated samples and using the mean of the remaining ones as the output, the ITM2 filter surpasses other filters in attenuating the impulsive-contaminated Gaussian noise [17]. A realization of the ITM filter [18] is verified to perform faster than the standard median filter. The ITM filter is extended to the weighted ITM filter to realize the band- and high-pass filters [19]. However, keeping all the truncated samples (ITM1) or trimming all the truncated samples (ITM2) may not be optimal if both the additive and exclusive noise exist. Moreover, further analysis in this paper reveals that the truncation threshold is largely affected by the extreme samples even if they are truncated. This reduces the convergence of the truncation threshold, and therefore leads to a high computational complexity.

In this paper, we propose a trimmed and truncated arithmetic mean (ITTM) algorithm to alleviate the above problems. The proposed algorithm iteratively trims and truncates the extreme samples simultaneously. Without sorting, the extreme samples are symmetrically trimmed from the input data set, and the remaining ones are truncated to a dynamic threshold. Three types of filter outputs are developed on the basis of the ITTM algorithm. The proposed trimming and truncating rules guarantee the filters' outputs approaching the median by increasing the number of iterations. With the stopping criterion given in [17] to terminate the iteration, the proposed ITTM filter is not only faster than the ITM filter, but also more effective in attenuating some types of noise.

## 2. The proposed ITTM filter

We propose the iterative trimmed and truncated arithmetic mean (ITTM) filters based on analysis of the ITM filter.

### 2.1. Iterative truncated arithmetic mean filter

As distinct from the mean filter that averages all samples and the median filter that chooses one sample as the output, the iterative truncated arithmetic mean (ITM) filter [17] iteratively truncates the extreme samples and uses the truncated mean as the filter output. Starting from $\mathbf{x}=\mathbf{x}_0$, it truncates samples in $\mathbf{x}$ to a dynamic threshold as shown by Algorithm 1.

**Algorithm 1.** Truncation procedure of the ITM algorithm.

**Input**: $\mathbf{x}_0 \Rightarrow \mathbf{x}$; **Output**: Truncated $\mathbf{x}$;
**do**
1) Compute the sample mean : $\mu=\text{mean}(\mathbf{x})$;
2) Compute the dynamic threshold : $\tau=\text{mean}(|\mathbf{x}-\mu|)$;
3) $b_l=\mu-\tau,\ b_u=\mu+\tau,$ and truncate $\mathbf{x}$ by :

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

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

The ITM filter has two types of outputs [17]. The type I output ITM1 is

$$y_{t1}=\text{mean}(\mathbf{x}). \tag{2}$$

Let $\mathbf{x}_r=\{x_i|b_l<x_i<b_u\}$ and $n_r$ be the number of samples in $\mathbf{x}_r$. The type II output ITM2 is

$$y_{t2}=\begin{cases} \text{mean}(\mathbf{x}_r) & \text{if } n_r>\xi \\ \text{mean}(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{3}$$

The parameter $\xi$ is used to avoid an unreliable mean in case that too few samples remain in $\mathbf{x}_r$. It is set to $\xi=n/4$ in [17].

### 2.2. The proposed ITTM filters

Keeping all the truncated samples makes the ITM1 filter less effective in suppressing the exclusive noise. Trimming all the truncated samples causes the ITM2 filter not optimal in dealing with the additive noise. Neither the ITM1 nor ITM2 filters can effectively deal with the case that both the additive and exclusive noise exist. In addition, a large number of iterations may be in demand for the ITM algorithm to converge as its truncation threshold, which is the mean absolute deviation (MAD) of the truncated $\mathbf{x}$, i.e. $\tau \triangleq \text{mean}(|\mathbf{x}-\mu|)$, is largely affected by the extreme samples even though they are truncated.

Trimming the truncated samples in the subsequent iterations is helpful to increase the convergence of the truncation threshold. Unfortunately, trimming all truncated samples violates the rule of the ITM algorithm that the output converges to the median. As the median is within the truncation bounds [17], trimming the same number of the truncated samples from both sides of the mean does not change the median. The following Theorem 1 guarantees that trimming the extreme samples leads to a higher convergence of the truncation threshold.

**Theorem 1.** *For any finite data set, simultaneously trimming the minimum and maximum samples decreases the MAD if the mean deviates from the median. Let* $\mathbf{x}=\{x_a,\mathbf{x}_r,x_b\}$ *be a set of* $n$ *samples, where* $\mathbf{x}_r=\{x_1,x_2,\dots,x_{n-2}\}$, $x_a \le \{x_i\}_{i=1}^{n-2} \le x_b$, *and* $\tau$ *and* $\tau_r$ *be the MAD of* $\mathbf{x}$ *and* $\mathbf{x}_r$ *respectively. We have*

$$\tau_r < \tau \quad \text{if } \mu \neq \phi, \tag{4}$$

*where* $\mu$ *and* $\phi$ *are the mean and median of* $\mathbf{x}$.

**Proof.** Let $\mu_r=\text{mean}(\mathbf{x}_r)$, and $n_h$ and $n_l$ denote the number of samples in $\{x_i | x_i \in \mathbf{x}_r, x_i > \mu_r\}$ and $\{x_i | x_i \in \mathbf{x}_r, x_i \le \mu_r\}$, respectively. The MAD $\tau_r$ of $\mathbf{x}_r$ satisfies

$$\begin{aligned}(n-2)\tau_r &= \sum_{i=1}^{n-2} |x_i-\mu_r| \\ &= \sum_{x_i>\mu_r}(x_i-\mu_r)+\sum_{x_i\le\mu_r}(\mu_r-x_i) \\ &= \sum_{x_i>\mu_r}(x_i-\mu)+\sum_{x_i\le\mu_r}(\mu-x_i)+(n_h-n_l)(\mu-\mu_r) \\ &\le \sum_{i=1}^{n-2}|x_i-\mu|+(n_h-n_l)(\mu-\mu_r).\end{aligned} \tag{5}$$

Based on (5), the MAD $\tau$ of $\mathbf{x}$ can be expressed as

$$\begin{aligned}\tau &= \frac{1}{n}\left[\sum_{i=1}^{n-2}|x_i-\mu|+(x_b-x_a)\right] \\ &\ge \frac{1}{n}\left[(n-2)\tau_r-(n_h-n_l)(\mu-\mu_r)+(x_b-x_a)\right].\end{aligned} \tag{6}$$

As $n_h+n_l=n-2$ and

$$n(\mu-\mu_r)=\sum_{i=1}^{n-2}x_i+x_a+x_b-\frac{n}{n-2}\sum_{i=1}^{n-2}x_i=x_a+x_b-2\mu_r, \tag{7}$$

the second and third terms of (6) can be reformulated as

$$\begin{aligned}&(x_b-x_a)-(n_h-n_l)(\mu-\mu_r) \\ &\quad=(x_b-x_a)-(n_h-n_l)(x_a+x_b-2\mu_r)/n \\ &\quad=2(n_l+1)(x_b-\mu_r)/n+2(n_h+1)(\mu_r-x_a)/n.\end{aligned} \tag{8}$$

Since $\tau_r=(2/(n-2))\sum_{x_i>\mu_r, x_i\in\mathbf{x}_r}(x_i-\mu_r)$ [17], we have

$$x_b-\mu_r \ge \frac{1}{n_h}\sum_{\substack{x_i>\mu_r\\ x_i\in\mathbf{x}_r}}(x_i-\mu_r)=\frac{n-2}{2n_h}\tau_r. \tag{9}$$

Similarly,

$$\mu_r-x_a \ge \frac{1}{n_l}\sum_{\substack{x_i\le\mu_r\\ x_i\in\mathbf{x}_r}}(\mu_r-x_i)=\frac{n-2}{2n_l}\tau_r. \tag{10}$$

Substituting (9) and (10) into (8) yields

$$\begin{aligned}&(x_b-x_a)-(n_h-n_l)(\mu-\mu_r) \\ &\ge \frac{2(n_l+1)}{n}\frac{(n-2)}{2n_h}\tau_r+\frac{2(n_h+1)}{n}\frac{(n-2)}{2n_l}\tau_r \\ &=\tau_r\frac{n-2}{n}\left(\frac{(n_h-n_l)^2+2n_hn_l+(n_h+n_l)}{n_hn_l}\right) \\ &\ge \tau_r\frac{n-2}{n}\left(2+\frac{n_h+n_l}{n_hn_l}\right)\ge\tau_r\frac{n-2}{n}\left(2+\frac{4}{n_h+n_l}\right) \\ &=2\tau_r.\end{aligned} \tag{11}$$

Therefore, substituting (11) into (6) yields

$$\tau \ge \tau_r. \tag{12}$$

Note that $\tau=\tau_r$ occurs if and only if all the inequalities of (9), (10) and (11) are respectively equal, i.e.

$$x_b-\mu_r=\frac{n-2}{2n_h}\tau_r, \tag{13}$$

$$\mu_r-x_a=\frac{n-2}{2n_l}\tau_r, \tag{14}$$

and

$$(x_b-x_a)-(n_h-n_l)(\mu-\mu_r)=2\tau_r, \tag{15}$$

which require (a) all samples are equal to either $x_b$ or $x_a$ and (b) $n_h = n_l$. This specific case does not occur if $\mu \neq \phi$. This completes the proof of (4). □

Theorem 1 guarantees that trimming a pair of extreme samples (one minimum and one maximum samples) from any finite data set reduces the MAD of this data set. It is further reduced by increasing the number of trimmed sample pairs. This inspires the proposed iterative trimmed and truncated arithmetic mean (ITTM) algorithm shown by Algorithm 2.

**Algorithm 2.** Procedure of the ITTM algorithm.

**Input**: $\mathbf{x}_0 \Rightarrow \mathbf{x}$; **Output**: Trimmed and truncated $\mathbf{x}$;
**do**

1) Compute the sample mean : $\mu=\text{mean}(\mathbf{x})$;
2) Compute the dynamic threshold : $\tau=\text{mean}(|\mathbf{x}-\mu|)$;
3) $b_l=\mu-\tau,\ b_u=\mu+\tau,$ and trim all sample pairs $(x_i,x_j)$ from $\mathbf{x}$ that satisfy $x_i \ge b_u,\ x_j \le b_l$;
4) Truncate the rest samples by :

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

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

Let $n_{\tau u}(k)$ and $n_{\tau l}(k)$ be the number of samples respectively satisfying $x_i \ge b_u$ and $x_j \le b_l$ in the $k$th iteration. Obviously, the number of samples trimmed from $\mathbf{x}$ in the $k$th iteration in Step 3 of Algorithm 2 is $2\min\{n_{\tau l}(k), n_{\tau u}(k)\}$.

Three types of the ITTM filter outputs are proposed. The type I and II outputs are analogous to those of the ITM filter. Assume the total number of the trimmed samples on each side of the mean be $n_t$. By padding the trimmed samples with the lower or upper bound, the padded $\mathbf{x}$ is $\mathbf{x}_p=\{n_t\diamond b_l,\mathbf{x},n_t\diamond b_u\}$, where $\diamond$ is the replication operator defined as

$$n_t\diamond x=\underbrace{x,x,\dots,x}_{n_t\ \text{times}}. \tag{16}$$

The type I output ITTM1 is

$$y_{tt1} = \text{mean}(\mathbf{x}_p). \tag{17}$$

The type II output ITTM2 is

$$y_{tt2} = \begin{cases} \text{mean}(\mathbf{x}_r) & \text{if } n_r > \xi \\ y_{tt1} & \text{otherwise,} \end{cases} \tag{18}$$

where $\mathbf{x}_r = \{x_i | b_l < x_i < b_u\}$, and $n_r$ is the number of samples in $\mathbf{x}_r$. Resembling the ITM filter, the ITTM1 filter keeps all the truncated samples, and the ITTM2 filter trims all the truncated samples. As the truncation threshold of the ITTM algorithm decreases faster than that of the ITM algorithm, with a same stopping criterion, the ITTM1 and ITTM2 filters will be faster than the ITM1 and ITM2 filters.

The type III output ITTM3 is designed to alleviate the problem that both the additive and exclusive noise exist. Similar to that of the ITM filter, keeping all the truncated samples (ITTM1) or trimming all the truncated samples (ITTM2) may not be optimal in dealing the mixed additive and exclusive noise. The proposed ITTM3 filter uses the mean of the trimmed and truncated data set as the filter output, i.e.

$$y_{tt3} = \begin{cases} \text{mean}(\mathbf{x}) & \text{if } n-2n_t > \xi \\ y_{tt1} & \text{otherwise,} \end{cases} \tag{19}$$

where $n-2n_t$ is the number of samples in $\mathbf{x}$. $\xi = n/4$ is used in this paper to avoid an unreliable mean caused by trimming too many samples.

The ITTM filter's output moves from the mean towards the median by increasing the number of iterations. Since neither mean nor median is the optimal solution for many real signals/images, proper stopping criterion is applied to suppress noise and preserve edges within a few iterations. In the ITTM filter, we use the stopping criterion $S$ proposed in [17] to automatically terminate the ITTM algorithm.

## 3. Properties of the ITTM filters

**Property 1** (*Faster convergence*). *The truncation threshold $\tau$ of the ITTM algorithm has a faster convergence than that of the ITM algorithm. It decreases monotonically to zero if the mean $\mu$ deviates from the median $\phi$, i.e.*

$$\tau(k+1) < \tau(k) \quad \textit{if } \mu \neq \phi, \tag{20}$$

*and*

$$\lim_{k\to\infty} \tau = 0, \quad \textit{if } \mu \neq \phi. \tag{21}$$

**Proof.** The faster convergence of ITTM than ITM can be derived from Theorem 1. The threshold $\tau$ of the ITTM algorithm converges to zero as it is a non-negative value and smaller than that of the ITM algorithm, which is proven to converge to zero in [17]. □

Fig. 1 shows the average of the truncation thresholds over $10^6$ Laplacian distributed input data sets against the number of iterations. They are normalized by the average MAD of the input data sets. The filter size is $n = 7 \times 7$. This figure demonstrates that the convergence of the ITTM algorithm is visibly faster than that of the ITM algorithm.

**Property 2** (*Converge to the median*). *The ITTM1, ITTM2 and ITTM3 outputs $y_{tt1}$, $y_{tt2}$ and $y_{tt3}$ converge to the median $\phi$ of the samples in the filter window, i.e.*

$$\lim_{k\to\infty} y_{tt1} = \lim_{k\to\infty} y_{tt2} = \lim_{k\to\infty} y_{tt3} = \phi. \tag{22}$$

**Proof.** Using the MAD of the data from the mean as the truncation threshold, the truncating procedure does not change $\phi$ [17]. Trimming the same number of the extreme samples from both sides of the mean also does not change $\phi$. Therefore, $\phi$ is not changed in the iterative trimming and truncating procedures. As Property 1 proves that the truncation threshold converges to zero, the outputs $y_{tt1}$, $y_{tt2}$ and $y_{tt3}$ converge to $\phi$. □

**Property 3** (*Scale and shift invariance*). *The ITTM filters are invariant to scale and shift, i.e. if $\mathbf{z} = \{ax_i + c\}$, $\forall x_i, x_i \in \mathbf{x}$, we have*

$$y_{tt}(\mathbf{z}) = a y_{tt}(\mathbf{x}) + c, \tag{23}$$

*where $a$ and $c$ are two constants, and $y_{tt}$ is the notation shared by all the three types of the ITTM filter outputs. The proof is trivial and hence omitted.*

**Property 4** (*Symmetric distribution*). *The distribution of the ITTM filter output is symmetric, if the samples of the input data set $\mathbf{x}_0 = \{x_1, x_2, \ldots, x_n\}$ are drawn from the random variable $X$ with a symmetric distribution.*

**Proof.** If $x_i$ is symmetrically distributed around $c$, $2c - x_i$ has the same distribution as $x_i$. According to Property 3, $y_{tt}(2c-x_1, 2c-x_2, \ldots, 2c-x_n) = 2c - y_{tt}(x_1, x_2, \ldots, x_n)$. Thus, the distribution of $y_{tt}$ is symmetric around $c$. □

**Property 5** (*Unbiased estimate*). *The ITTM filter output 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 drawn from the random variable $X$ with a symmetric distribution.*

**Proof.** According to Property 4, $y_{tt}$ is symmetrically distributed around $c$. Therefore, $E\{y_{tt}\} = c = E\{X\}$. This completes the proof of Property 5. □

**Property 6** (*Edge preservation*). *The ITTM2 filter of size in odd number preserves image step edges with any number of iterations. The proof is similar to that of the ITM2 filter given in [17], and hence is omitted.*

Fig. 2 shows a step edge profile. The filter size of $n = 11 \times 11$ is applied. After the first iteration, the output of the ITTM2 filter is the same as the median filter. After 3 iterations, the ITTM1 and ITTM3 filters produce much lighter blur effect than the mean filter.

**Property 7** (*Impulsive noise suppression*). *The ITTM2 filter with any iterations removes impulse $\mathbb{D}_1$ from the homogeneous area $\mathbb{D}_2$:*

$$x_i = \begin{cases} c_1 & \text{for } x_i \in \mathbb{D}_1, n_1 < n/2 \\ c_2 & \text{for } x_i \in \mathbb{D}_2, n_2 > n/2, \end{cases} \quad \mathbb{D}_1 \cup \mathbb{D}_2 = \mathbf{x}_0, \tag{24}$$

*where $c_1 \neq c_2$, $n_1$ and $n_2$ are the numbers of samples in sets $\mathbb{D}_1$ and $\mathbb{D}_2$, respectively. The proof is analogous to that in [17], and hence is omitted.*

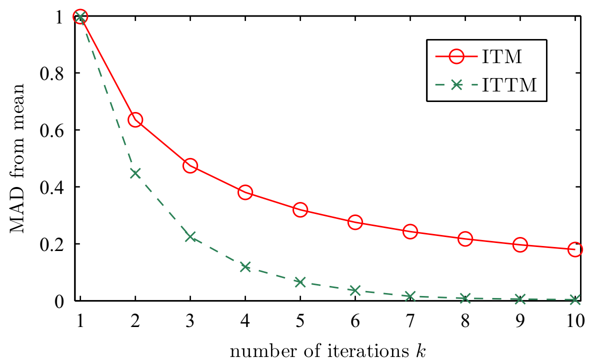


**Fig. 1.** Average MAD of the samples from the mean against the number of iterations. The input data sets are Laplacian noise. The filter size is $n = 7 \times 7$.

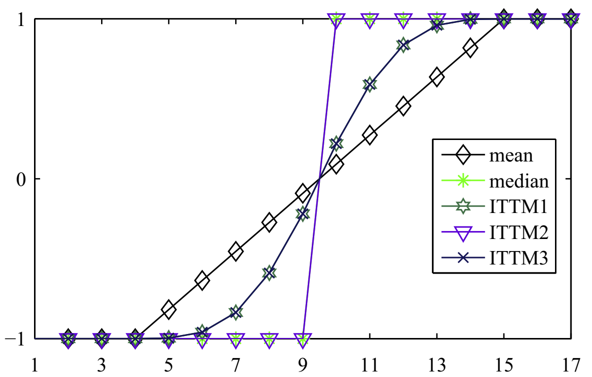


**Fig. 2.** Profile outputs of the mean, median, ITTM1, ITTM2 and ITTM3 filters of size $n = 11 \times 11$ after 3 iterations for a step edge.

## 4. Computational complexity

The computational complexity depends on how the algorithm is realized. A realization of the ITTM filter is proposed based on the following proposition.

**Proposition 1.** *Samples, once being truncated in an iteration of the ITTM algorithm, must be either trimmed or truncated in all subsequent iterations.*

**Proof.** If the number of samples smaller than the lower bound equals to that larger than the upper bound, all these samples are trimmed in this iteration according to the definition of the ITTM algorithm. For the case that these two numbers are not equal, all pairs of extreme samples are trimmed from the input data set, and the remaining samples that are out of one of the two bounds are truncated to that bound. In the following, we prove that the truncated samples will be either truncated or trimmed in the subsequent iteration.

Let $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ be the input data in the $k$th iteration, $\mathbf{x}_h = \{x_i | x_i > \mu\}$, $n_h$ be the number of the samples in $\mathbf{x}_h$, and $\delta_h = \mathrm{sum}(\mathbf{x}_h - \mu)/n_h$. Let $\mathbf{x}_+$, $x_{i+}$, $\mu_+$, $\tau_+$, $n_{h+}$ and $\delta_{h+}$ be the corresponding notations in the $(k+1)$th iteration.

Assume that $m$ samples are trimmed from each side of the input data, and a sample $x_{iu}$ is truncated to the upper bound $\mu + \tau$ in the $k$th iteration. Obviously, $x_{iu} = \mu + \tau$, which has the maximum value of the input data set $\mathbf{x}_+$ of the $(k+1)$th iteration. As Property 1 proves that the truncation threshold decreases monotonically, $\tau_+ < \tau$. Therefore, we have

$$x_{iu+} > \mu_+ + \tau_+ \quad \text{if } \mu_+ \le \mu. \tag{25}$$

In case of $\mu_+ > \mu$, from $\tau = 2n_h\delta_h/n$ [17], we have

$$\tau = \frac{2}{n}\sum_{\substack{x_i \in \mathbf{x} \\ x_i > \mu}} (x_i - \mu) \ge \frac{2}{n}\sum_{\substack{x_i \in \mathbf{x} \\ x_i > \mu_+}} (x_i - \mu_+ + \mu_+ - \mu)$$
$$= \frac{2}{n}\sum_{\substack{x_i \in \mathbf{x} \\ x_i > \mu_+}} (x_i - \mu_+) + \frac{2(n_{h+} + m)}{n}(\mu_+ - \mu). \tag{26}$$

As there are $m$ samples trimmed from $\mathbf{x}_h$ and at least one sample $x_{iu}$ truncated to the upper bound in the $k$th iteration, we have

$$\frac{2}{n}\sum_{\substack{x_i \in \mathbf{x} \\ x_i > \mu_+}} (x_i - \mu_+) > \frac{2}{n}\sum_{\substack{x_{i+} \in \mathbf{x}_+ \\ x_{i+} > \mu_+}} (x_{i+} - \mu_+) + \frac{2m}{n}(x_{iu} - \mu_+)$$
$$= \frac{n-2m}{n}\tau_+ + \frac{2m}{n}\left(\tau - (\mu_+ - \mu)\right). \tag{27}$$

Substituting (27) into (26) yields

$$\tau > \frac{n-2m}{n}\tau_+ + \frac{2m}{n}\tau + \frac{2n_{h+}}{n}(\mu_+ - \mu). \tag{28}$$

With some manipulation, (28) becomes

$$\tau > \tau_+ + \frac{2n_{h+}}{n-2m}(\mu_+ - \mu). \tag{29}$$

As $\tau_+ = 2n_{h+}\delta_{h+}/(n-2m)$, (29) becomes

$$\tau > \tau_+ + \frac{\tau_+}{\delta_{h+}}(\mu_+ - \mu). \tag{30}$$

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

$$x_{iu+} - \mu_+ \le \tau_+. \tag{31}$$

Therefore, under the condition (31), (30) becomes

$$\tau > \tau_+ + \mu_+ - \mu. \tag{32}$$

Since $x_{iu+} = \mu + \tau$, (32) becomes

$$x_{iu+} > \mu_+ + \tau_+. \tag{33}$$

The conclusion (33) conflicts with (31). Hence, the condition (31) is not true, which means

$$x_{iu+} > \mu_+ + \tau_+ \quad \text{if } \mu_+ > \mu. \tag{34}$$

From (25) and (34), we have

$$x_{iu+} > \mu_+ + \tau_+. \tag{35}$$

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

$$x_{il+} < \mu_+ - \tau_+. \tag{36}$$

Inequalities (35) and (36) guarantee that the truncated samples will be either trimmed or truncated in the following iteration. This completes the proof of Proposition 1. □

As all truncated samples must be either trimmed or truncated in the subsequent iterations, we do not need access such samples for computing the mean, threshold and checking whether they should be trimmed, truncated or not. We only need count the number of such samples in all subsequent iterations. Let $n_{\tau u}$ and $n_{\tau l}$ be the total numbers of the samples of $\{x_i\}$ and $\{x_j\}$ in $\mathbf{x}_0$ satisfying $x_i \ge b_u$ and $x_j \le b_l$, respectively. The proposed implementation of the ITTM algorithm is shown by Algorithm 3.

**Algorithm 3.** An implementation of the ITTM algorithm.

**Input**: $\mathbf{x}_0 \Rightarrow \mathbf{x}_r$, $n_t{=}0$ $n_\tau = 0$; **Output**: $\mathbf{x}_r$, $b_l$, $b_u$, $b$, $n_{\tau l}$, $n_{\tau u}$, $n_\tau$ and $n_t$;
**do**
1) $\mu = (\text{sum}(\mathbf{x}_r) + n_\tau b)/(n - 2n_t)$;
2) $\tau = (\text{sum}(|\mathbf{x}_r - \mu|) + n_\tau |b - \mu|)/(n - 2n_t)$;
3) $b_l = \mu - \tau$, $b_u = \mu + \tau$, $\mathbf{x}_r = \{x_i | b_l < x_i < b_u\}$ and update $n_{\tau l}$ and $n_{\tau u}$;
4) Compute $n_t = \min\{n_{\tau l}, n_{\tau u}\}$, $n_\tau = |n_{\tau l} - n_{\tau u}|$, and $b = b_u$ if $n_{\tau u} > n_{\tau l}$, else $b = b_l$;
**while** *the stopping criterion S is violated*;

The three types of the ITTM filter outputs are reformulated as

$$y_{tt1} = \frac{1}{n}(\text{sum}(\mathbf{x}_r) + n_{\tau l} b_l + n_{\tau u} b_u), \tag{37}$$

$$y_{tt2} = \begin{cases} \text{mean}(\mathbf{x}_r) & \text{if } n_r > \xi \\ y_{tt1} & \text{otherwise}, \end{cases} \tag{38}$$

and

$$y_{tt3} = \begin{cases} \dfrac{1}{n - 2n_t}(\text{sum}(\mathbf{x}_r) + n_\tau b) & \text{if } n - 2n_t > \xi \\ y_{tt1} & \text{otherwise}. \end{cases} \tag{39}$$

The results of (37), (38) and (39) are the same as those of (17), (18) and (19), respectively.

The computational complexity of the ITTM filter can be measured by the number of the times that a sample in $\mathbf{x}_0$ is visited in the ITTM algorithm. It is determined by (a) the number of iterations $N_s$, and (b) the probability $p_k$ of a sample being visited in the $k$th iteration.

We use the Monte Carlo simulations [20] to analyze the number of iterations $N_s$. The stopping criterion is set the same as that in [17]. Three types of noise, Gaussian, Laplacian and the uniform distributed noise, are simulated. $10^6$ independent input data sets are used in each experiment. The experimental results in Fig. 3 illustrate that the numbers of iterations of different noise types have the same tendency. They are approximately linear functions of ln $n$. Therefore, we use

$$\hat{N}_s = 0.8 \ln n \tag{40}$$

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

As Algorithm 3 only accesses the un-trimmed and un-truncated samples, the visited samples in the $k$th iteration are the ones within the range $(\mu_{k-1} - \tau_{k-1}, \mu_{k-1} + \tau_{k-1})$. As the truncation threshold decreases monotonically, the probability of a sample within this range decreases. Therefore, the probability of a sample being visited $p_k$ decreases. In order to simplify the analysis of $p_k$, we employ the uniform distributed noise as an example. The probability density function (pdf) of a uniform distributed random

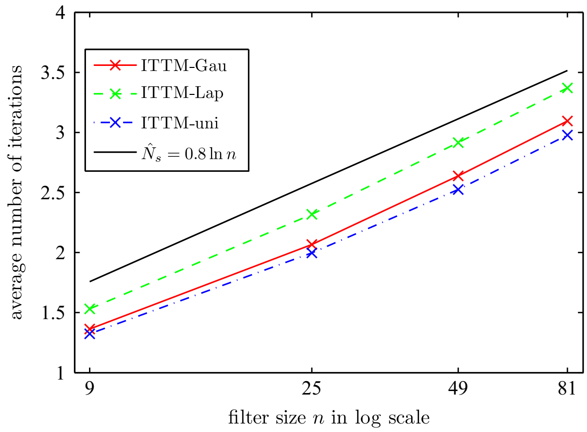


**Fig. 3.** Average number of iterations determined by the default stopping criterion against the filter size $n$.

variable $X$ is given by

$$f_u(X = x) = \begin{cases} 1 & \text{if } -0.5 \le x \le 0.5, \\ 0 & \text{otherwise}. \end{cases} \tag{41}$$

From (41), we find the following lemma of $\tau_k$.

**Lemma 1.** *For sufficiently large filter size $n$, the truncation threshold $\tau_k$ of the trimmed and truncated X drawn from the uniform distribution (41) follows*

$$\tau_k = \frac{1}{2^{k+1}}, \quad k \ge 1. \tag{42}$$

**Proof.** When the filter size $n$ is sufficiently large, the sample mean equals the expectation of $X$, i.e. $\mu = E[X] = 0$. The truncation threshold of the first iteration is

$$\tau_1 = \frac{1}{n}\sum_{i=1}^{n} |x_i - \mu| = E[|X|] = \frac{1}{2^2}. \tag{43}$$

As the output of the ITTM filter is unbiased, $E[\mu_{k-1}] = 0$. Thus, almost all samples outsides the range $(-\tau_{k-1}, \tau_{k-1})$ are trimmed. Therefore, the truncation threshold of the $k$th iteration $\tau_k$ is

$$\tau_k = \frac{1}{2\tau_{k-1}} \int_{-\tau_{k-1}}^{\tau_{k-1}} |x| \, dx = \frac{\tau_{k-1}}{2}. \tag{44}$$

Eqs. (43) and (44) lead to (42). This completes the proof of Lemma 1. □

As only the un-truncated samples are visited by the ITTM algorithm, the probability of a sample being visited at the $k$th iteration is $p_k = 2\tau_{k-1}$, where we define $\tau_0 = 0.5$. Therefore, the computational complexity of the ITTM algorithm is $\mathcal{O}(\sum_{k=1}^{\hat{N}_s} np_k) = \mathcal{O}(2n\sum_{k=1}^{\hat{N}_s} \tau_{k-1})$. From Lemma 1 we get that $1 \le 2\sum_{k=1}^{\hat{N}_s} \tau_{k-1} < 2$. Thus, the computational complexity of the ITTM algorithm is $\mathcal{O}(n)$. It is lower than the standard median filter and the fast ITM (FITM) filter given in [18], both of which are $\mathcal{O}(n \log n)$ [18].

The Monte Carlo simulations are also carried out to analyze the visiting times for the filter size $n$ that is not sufficiently large for the above statistics. $10^6$ independent input data sets are used in each experiment. The experimental results in Fig. 4 show that the average visiting times of a sample in the ITTM filter window is constrained by the upper bound $\hat{N}_{ave} = 2$. Therefore, for $n$ samples, the average visiting times is smaller than $2n$.

We further evaluate the computational complexity of the ITTM filter in two experiments. These experiments are performed under the Window 7 system with the Intel Core i5 CPU 3.2 GHz. All filters are implemented by C programming language. As data sorting is the basic building block of many rank order statistic filters, such as the popular $\alpha$T filter, we implement both the median and $\alpha$T filters using the quick-sort algorithm. The ITM filter is implemented based on the fast algorithm FITM given in [18]. The MA, OM (optimal Myriad) and MLC (Mean-LogCauchy) filters are implemented according to the algorithms given in [12,21,22]. The first experiment tests the running time of the filters against the number of iterations, and the second experiment tests that with the stopping criterion against the filter size. $10^6$ independent Laplacian distributed input data sets are used in each experiment. The time consumption is normalized by that of the median filter. The normalized time consumption with the filter size $n=7\times7$ against the number of iterations is shown in Fig. 5. The running time of all the iterative-algorithm based filters, the MA, OM, MLC, FITM and ITTM filters, increases with the increasing number of iterations. As the truncation threshold of the ITTM filter decreases faster than that of the ITM filter, its time consumption increases slower compared to that of the FITM filter. Both the FITM and ITTM filters are faster than the median filter for all the numbers of iterations in Fig. 5. The ITTM filter is the fastest one. The computational complexity of the MA, OM and MLC filters is much higher than the median filter. The experimental results using the stopping criterion are plotted in Fig. 6. The stopping criterion for both the FITM and ITTM filters is set the same as that in [17]. The fixed number of iterations 10 is applied for the MA, OM and MLC filters. As the MLC filter is a linear combination of the mean and OM filters, its computational complexity is a little bit higher than that of the OM filter. Similarly, the time consumption of the $\alpha$T filter is slightly higher than that of the median filter because it averages part of the data after data sorting. The MA, OM and MLC filters are

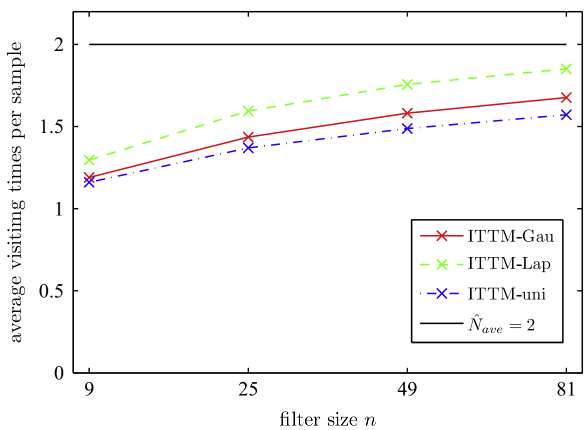


**Fig. 4.** Average visiting times of a sample against the filter size $n$.

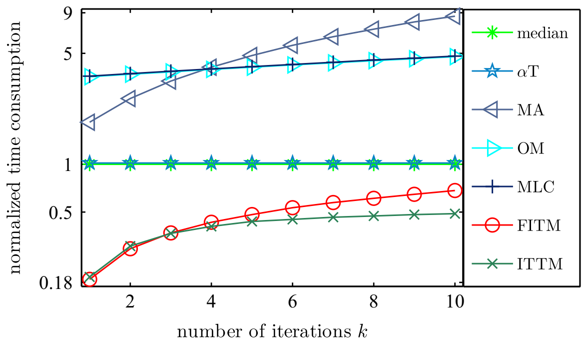


**Fig. 5.** Normalized time consumption against the number of iterations $k$. The time consumption is normalized by that of the median filter. The $y$-axis is in log scale.

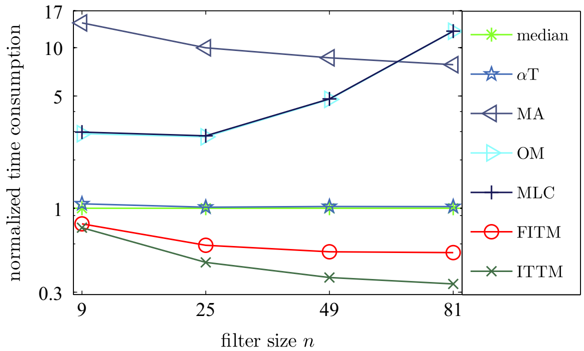


**Fig. 6.** Normalized time consumption against the filter size $n$. The time consumption is normalized by that of the median filter. The $y$-axis is in log scale.

much slower than the median filter. Both the FITM and ITTM filters are faster than the median filter. The ITTM filter is the fastest one for all filter sizes.

## 5. Experiments

Although better performance can be achieved by tuning the parameters, no parameter of the proposed filter is optimized for specific data set or noise distribution. The same parameters of the ITTM filter are employed throughout all experiments. To compare the proposed ITTM filter with the ITM filter, the parameter setting of the stopping criterion given in [17] is applied to these two filters. All additive noise applied in this work has i.i.d. and zero mean. The standard deviation of Gaussian noise is denoted by $\sigma_n$. Six sets of experiments are reported in this section.

The filters' noise attenuation capability in a constant signal is tested in the first two sets of experiments, and the filters' overall performance in image structure preservation and noise attenuation is tested in the next two sets. The ITTM filters are compared with the mean, median, $\alpha$T [13,23], MEM [11,24] and ITM [17] filters. The mean absolute error (MAE) over $10^7$ independent outputs is used as the performance indicator for synthetic data. The mean square error (MSE) is used for real images. As none of the $\alpha$-adapted $\alpha$T filters [25,26] outperform an $\alpha$-fixed $\alpha$T filter averagely over the experiments, the $\alpha$-fixed $\alpha$T filter is compared in this section. $\alpha=0.25$ is chosen as the $\alpha$T filter approaches the mean if $\alpha \to 0$ and approaches the median if $\alpha \to 0.5$.

Furthermore, the proposed filters are compared with four iterative-algorithm based filters, the MA [12], OM [27], MLC [11] and ITM [17] filters in the last two sets of experiments. The $\alpha$-stable noise is tested as the MA, OM and MLC filters were proposed specifically to tackle the problem of this noise model. The MLC filter is a weighted sum of the mean and LogCauchy filters to tackle the $\varepsilon$ contaminated Gaussian noise [28]. The weight $\lambda$ was suggested to be equal to the prior probability of the Gaussian noise [11,24]. $\lambda=0.5$ is chosen for both the MLC and MEM filters in this work.

### 5.1. Single type of noise in constant signal

Filters' performance in suppressing single type of noise on a constant signal is tested. Experimental results in suppressing Gaussian and Laplacian noise are shown in Fig. 7(a) and (b), respectively. The filters' MAEs are normalized by that of the median filter.

As the mean filter is optimal for Gaussian noise, the ITM, ITTM, MEM and $\alpha$T filters perform between the mean and median filters by making a compromise between them. The median filter is not the minimum MSE estimator [18] though it is the ML estimator for the Laplacian distribution. Therefore, it is not a surprise that the ITM1 filter and the proposed ITTM1 and ITTM3 filters outperform the median filter even for the long-tailed Laplacian noise. The $\alpha$T filter is better than the median filter for $n=3\times 3$ and surprisingly much worse than the median filter for larger filter size. It is seen that the numbers of iterations of the ITTM filters are smaller than those of the ITM filters for both Gaussian and Laplacian noise. Note that the mean filter is not shown in Fig. 7(b), and the mean and MEM filters are not shown in Fig. 8(b) as they are much worse than the other filters.

### 5.2. Mixed types of noise in constant signal

The mixed noise is generated from the noise model (1). It contains two types of noise, the additive noise with probability $p$ and the exclusive noise with probability $1-p$. The additive noise is set to have the $\varepsilon$-contaminated normal distribution as $\mathcal{P}_\varepsilon = \{(1-\varepsilon)\Phi+\varepsilon H\}$ [28], where $\Phi$ and $H$ are Gaussian and a longer-tailed distributions, respectively, $\varepsilon \in [0,1]$. Similar to the setting in [17], we choose the Laplacian distribution as $H$ with the standard deviation $1.3\sigma_n$ and $\varepsilon=0.5$. The exclusive noise is generated from pdf $I(x)=0.5\delta(x-6\sigma_n)+0.5\delta(x+6\sigma_n)$.

We first set $p=1$ so that only the additive noise exists. Results in Fig. 8(a) show that all the ITM and ITTM filters have better performances than both the mean and median filters. Then, we decrease the probability of the additive noise to $p=0.9$. The probability of the exclusive noise increases to $1-p=0.1$. The experimental results are shown in Fig. 8(b). Here it is evidenced that the proposed ITTM3 filter that trims and truncates the samples performs the best, outperforming filters ITM1 and ITTM1 that only truncate the extreme samples and outperforming filters ITM2, ITTM2 and $\alpha$T filters that only trim extreme samples.

### 5.3. Noise step edge

A horizontal or vertical step edge with the grey level 1 on one edge side and $-1$ on the other side is tested. Such an edge is contaminated by Gaussian noise of different levels. Outputs of a filter are used for computing MAE if and only if the filter mask covers both sides of the edge. Experimental results are shown in Fig. 9. Filters' MAE is normalized by the MAD of the noise.

The normalized MAEs of the median filter over different noise levels are almost a constant. This confirms the excellent ability of the median filter in edge preservation. The mean, MEM, $\alpha$T, ITM1, ITTM1 and ITTM3 filters all blur the edge. Therefore, their MAEs are much higher than the median filter for low noise levels. Since they attenuate

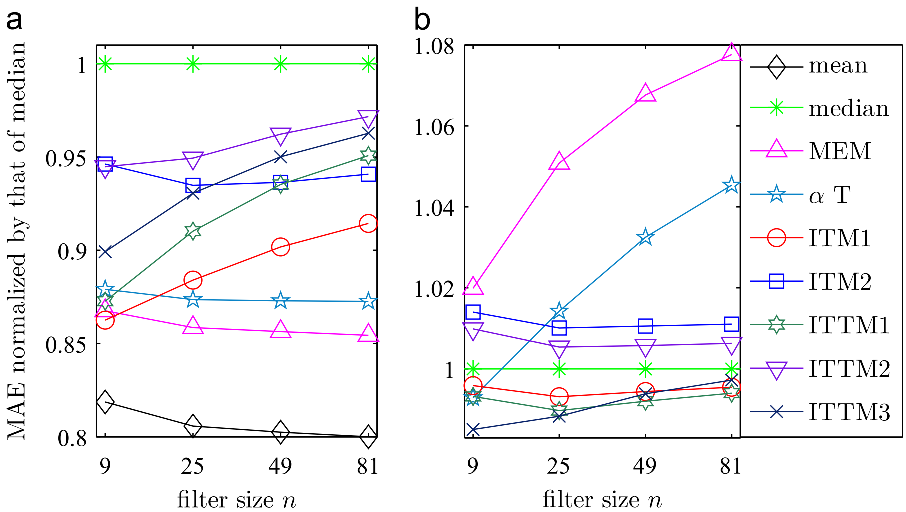


**Fig. 7.** Normalized MAE against the filter size $n$ for (a) Gaussian and (b) Laplacian noise. The average numbers of ITM iterations are 1.5, 3.3, 5.3, 7.5 in (a) and 1.8, 3.6, 5.7, 7.7 in (b), respectively for the filter size of 9–81. The average numbers of ITTM iterations are 1.4, 2.1, 2.6, 3.1 in (a) and 1.5, 2.3, 2.9, 3.4 in (b).

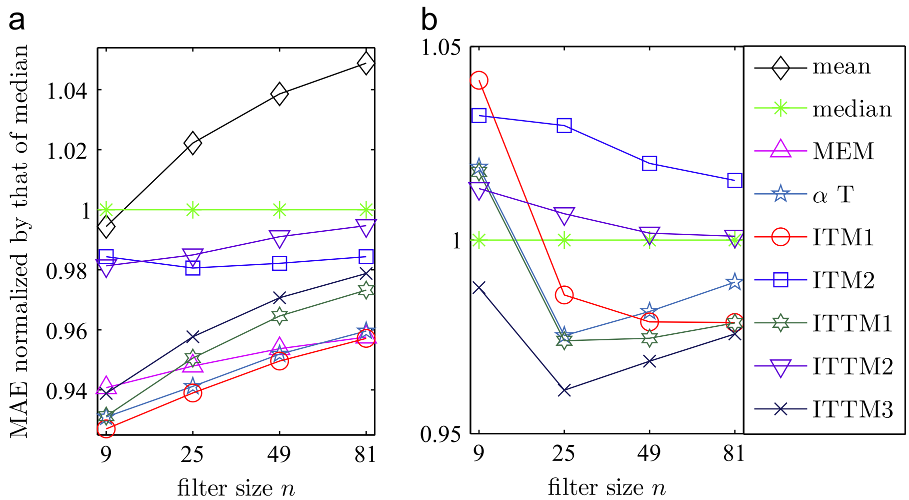


**Fig. 8.** Normalized MAE against the filter size $n$ for (a) Laplacian and (b) Laplacian and impulsive $\varepsilon$-contaminated Gaussian noise. The average numbers of ITM iterations are 1.6, 3.4, 5.4, 7.5 in (a) and 1.6, 3.4, 5.5, 7.6 in (b), respectively for the filter size of 9–81. The average numbers of ITTM iterations are 1.4, 2.2, 2.8, 3.2 in (a) and 1.4, 2.2, 2.8, 3.2 in (b).

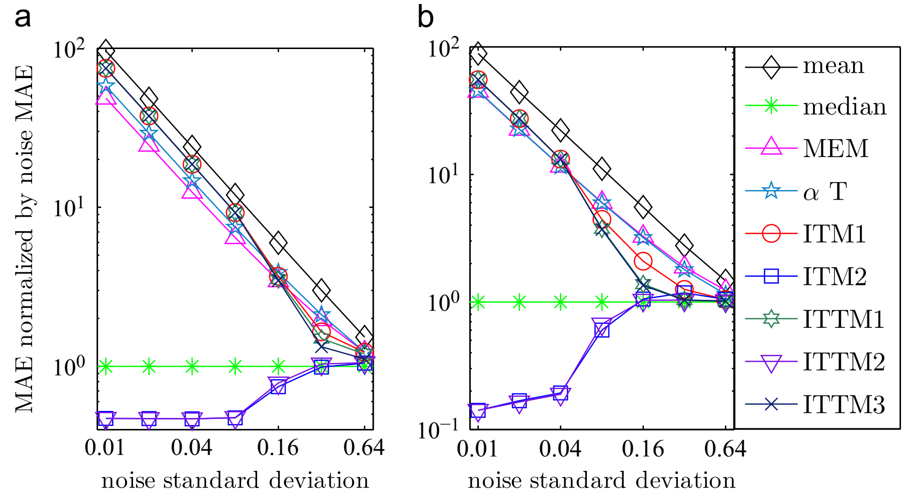


**Fig. 9.** Normalized MAE of filters of size $3\times3$ in (a) and $11\times11$ in (b) against the noise level $\sigma_n$. The average numbers of ITM iterations are 1, 1, 1, 1, 2.7, 3.7, 2 in (a) and 1.8, 3.8, 4.7, 11.6, 13.5, 14.7, 12.2 in (b). The average numbers of ITTM iterations are 1, 1, 1, 1, 2.3, 2.8, 1.6 in (a) and 1.8, 2.6, 3.0, 6.7, 6.6, 5.2, 4.1 in (b).

Gaussian noise better than the median filter, they perform about the same as the median filter when $\sigma_n$ reaches 0.64. The performance of the ITTM2 filter is similar to that of the ITM2 filter. Both of them significantly outperform the other for the low and medium noise levels. Their performances approach those of the median at the three highest noise levels in which the pixel gray levels of the two edge sides are overlapped.

### 5.4. *Real images*

Three real natural images shown in Fig. 10 are tested. These images, named Crowd, Bank and Girl, represent different image types and complexity levels. The image is composed of $512\times512$ pixels, of which the gray levels range from 0 to 255. The filter size is $n=5\times5$.

Additive Gaussian noise of different levels $\sigma_n$ is used to contaminate the images. For all three images, the MSE of the median filter increases from the minimum among the eight filters to the maximum along with the increasing of $\sigma_n$. Therefore, five different noise levels are selected for each image so that the median filter performs best at level $\sigma_n(1)$ and worst at level $\sigma_n(5)$. The other three noise levels are determined by $\sigma_n(5)/\sigma_n(4)=\sigma_n(4)/\sigma_n(3)=\sigma_n(3)/\sigma_n(2)=\sigma_n(2)/\sigma_n(1)$. All MSEs are normalized by that of the median filter. Average MSEs over 10 runs for image Crowd, Bank and Girl are plotted in Fig. 11(a), (b) and (c), respectively. The results of Fig. 11 coincide with the theory that the median filter preserves image structures best while attenuates Gaussian noise worst. The proposed ITTM3 filter performs best except for the lowest noise level where the median filter is the best. As the gray value abrupt change of image structure that fall in the filter window can be considered as impulsive noise or exclusive noise, the noise of the real images should be modeled by (1) with $1-p>0$ even if only additive Gaussian noise

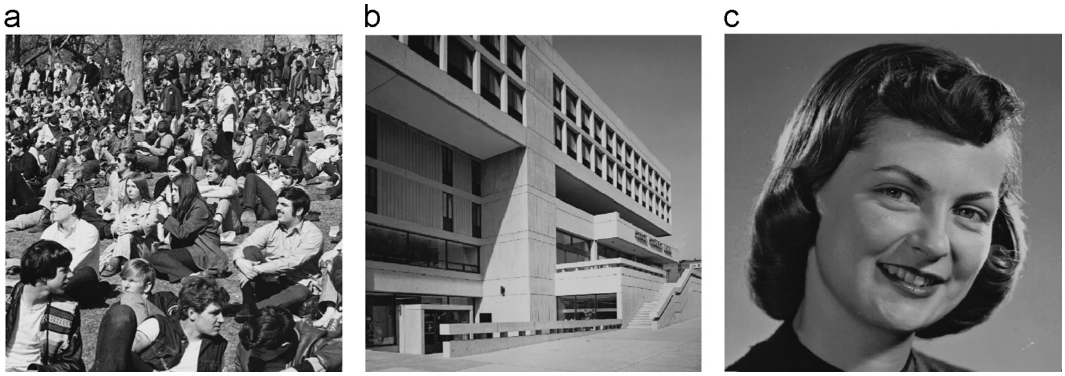


**Fig. 10.** Three real natural images of size 512 × 512 pixels, Crowd, Bank and Girl applied in testing.

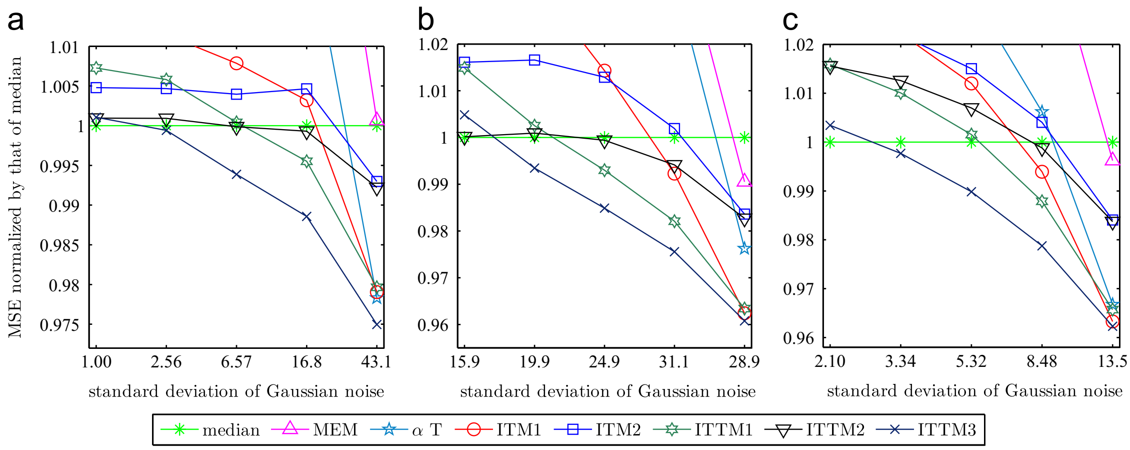


**Fig. 11.** Average MSEs over 10 runs for the 3 real images at 5 different noise levels of (a) Crowd, (b) Bank and (c) Girl. The average number of ITM iterations is closely around 3.4. The average number of ITTM iterations is closely around 2.1.

exists. This result further confirms that the proposed ITTM3 filter performs the best in attenuating the noise mixed by additive and exclusive noise. The average number of iterations for the ITTM filters is 2.1. It is smaller than that of the ITM filters which is 3.4.

### 5.5. Constant signal corrupted by $\alpha$-stable noise

The ITTM filters are compared with some other iterative-algorithm based filters on a constant signal corrupted by the $\alpha$-stable noise. The heaviness of the noise tails (degree of impulsiveness) is controlled by adjusting the parameter $\alpha$ of the $\alpha$-stable noise ($0 < \alpha \le 2$). The noise impulsiveness increases as $\alpha$ decreases [11]. The “linearity parameter” $k$ of the OM filter [27] is computed by $k = \sqrt{\alpha/(2-\alpha)}\gamma^{1/\alpha}$ where $\gamma$ is the dispersion of the $\alpha$-stable noise. The OM and MLC filters are implemented based on the algorithm given in [21,22]. For the MA, OM and MLC filters, the fixed number of iterations 20 is applied as more iterations do not lead to a visible performance gain [17]. All filters applied are in the size of 25.

Five different $\alpha$ values are set for the $\alpha$-stable noise with $\gamma = 10$. For each $\alpha$, $10^7$ independent input data sets are generated to get the MSEs. Table 1 records the MSEs of various filters where the results of the mean filter help show the noise impulsiveness. For each $\alpha$ value, the smallest MSE among all filters is underlined and in bold font, and the second smallest is in bold font. The OM filter is derived based on the maximum likelihood estimation from the samples of the $\alpha$-stable distribution ($\alpha = 1$). With the help of the noise distribution information $\alpha$ and $k$, the OM filter performs best for $\alpha$ value around 1. Among the filters which do not require the prior knowledge of the noise distribution, including the ITM, ITTM and MA filters, the ITTM3 filter performs best for $\alpha$ value around 1. The

**Table 1**
MSEs for constant image contaminated by $\alpha$-stable noise.

| | $\alpha$ | | | | |
|---|---|---|---|---|---|
| Filters | 0.5 | 0.8 | 1.2 | 1.5 | 1.8 |
| Mean | 486.41 | 217.44 | 73.050 | 32.235 | 14.175 |
| MA | 215.84 | 62.108 | 18.550 | 11.315 | **9.2205** |
| OM | 28.796 | **8.2874** | **10.617** | 12.479 | 11.642 |
| MLC | 164.71 | 62.017 | 24.940 | 15.216 | 10.555 |
| ITM1 | 26.924 | 13.343 | 11.409 | **10.631** | **10.074** |
| ITM2 | **15.050** | 12.482 | 12.320 | 12.186 | 11.559 |
| ITTM1 | 18.827 | 11.827 | 11.128 | 10.779 | 10.516 |
| ITTM2 | **13.503** | 11.547 | 11.759 | 11.925 | 11.679 |
| ITTM3 | 17.536 | **11.252** | **10.796** | **10.678** | 10.698 |

MA filter achieves the best performance for $\alpha$ value approaching 2, where the $\alpha$-stable noise degenerates to Gaussian noise. For high degree of impulsiveness ($\alpha=0.5$) the ITTM2 and ITM2 filters perform best. The average numbers of ITTM iterations are 4.2, 3.3, 3.1, 2.9, 2.8, respectively for the $\alpha$ values from 0.5 to 1.8. The corresponding average numbers of ITM iterations are 6.6, 5.1, 4.7, 4.5, 4.3. It is seen that the numbers of iterations of the ITTM filters are smaller than those of the ITM filters for all $\alpha$ values.

### 5.6. *Real image corrupted by Gaussian and $\alpha$-stable noise*

The original image “Lena” shown in Fig. 12 is corrupted by $\varepsilon$-contaminated [28] ($\varepsilon=0.5$) Gaussian ($\sigma_n^2=100$) and $\alpha$-stable ($\gamma=10$) noise. The image size is $512\times512$ pixels. The settings of filters are the same as those in Section 5.5. Fig. 13 shows the average MSEs of filters over 10 different noised versions of the image “Lena”. All MSEs are normalized by that of the median filter. It demonstrates that the

**Fig. 12.** Real image “Lena” of size $512\times512$ pixels tested for the mixed $\alpha$-stable noise.

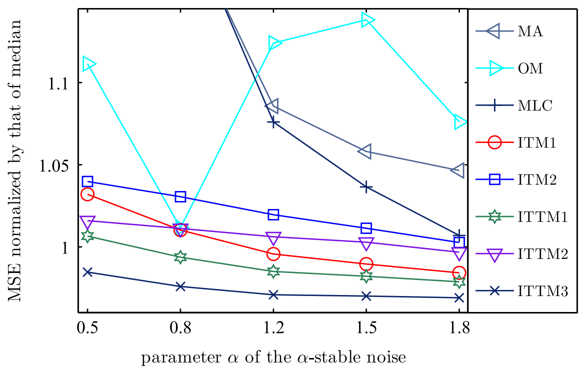


**Fig. 13.** Normalized MSEs for real image “Lena” corrupted by $\varepsilon$-contaminated ($\varepsilon=0.5$) Gaussian ($\sigma_n^2=100$) and $\alpha$-stable ($\gamma=10$) noise.

ITTM3 filter achieves the best performance for all the 5 values of $\alpha$. The performance of the ITTM1 filter is the second best in dealing with this real image. The average numbers of ITTM iterations are 3.1, 2.7, 2.5, 2.4 and 2.3, respectively for the $\alpha$ values from 0.5 to 1.8. They are smaller than the corresponding average numbers of ITM iterations, which are 4.5, 4.0, 3.8, 3.7 and 3.6.

## 6. Conclusion

The proposed iterative trimmed and truncated arithmetic mean (ITTM) filters circumvent the data sorting process and guarantee the outputs approaching the median with the increasing number of iterations. It is shown in the experiments that the proposed ITTM filters with the rule proposed in [17] that automatically stops the iterations own some merits of both the mean and median filters, and outperform these two fundamental filters in many de-noising applications. By simultaneously trimming and truncating the extreme samples, the ITTM algorithm has a higher convergence rate than the ITM algorithm. The resulting ITTM filters are hence faster than the ITM filters. The computational complexity of the ITTM filters is $\mathcal{O}(n)$. It is smaller than that of the median and ITM filters, both of which are $\mathcal{O}(n \log n)$ [18]. Although the ITTM filters use an iterative algorithm, only a few iterations are needed in all the experiments of this paper to achieve a good de-noising performance. The number of iterations of the ITTM filter is smaller than that of the ITM filter in all the experiments of this work.

Three types of the ITTM filter outputs, ITTM1, ITTM2 and ITTM3, are proposed. By averaging the truncated input samples, the ITTM1 filter has the best performance in attenuating the short- and long-tailed additive noise among these three filters. By trimming all the truncated samples, the ITTM2 filter has the best performance in suppressing exclusive noise. Simultaneously trimming and truncating the extreme samples lead the ITTM3 filter the best one in attenuating the noise mixed by both the additive and exclusive noise. As the gray value abrupt change of image structure in a filter window can be considered as exclusive or impulsive noise, the ITTM3 filter achieves the best performance in dealing with the real images of this paper. The superiority and flexibility of the proposed filters are demonstrated by the comprehensive simulation results with the same parameter setting throughout all experiments.

## References


[1] G.R. Arce, Nonlinear Signal Processing: A Statistical Approach, Wiley, New York, 2004.
[2] Z.W. Miao, X.D. Jiang, A novel rank order LoG filter for interest point detection, in: Proceedings of International Conference on Acoustics, Speech and Signal Processing, 2012, pp. 937–940.
[3] Z.W. Miao, X.D. Jiang, Interest point detection using rank order LoG filter, Pattern Recognit. 46 (11) (2013) 2890–2901.
[4] D. Charalampidis, Steerable weighted median filters, IEEE Trans. Image Process. 19 (4) (2010) 882–894.
[5] K.E. Barner, Y. Nie, W. An, Fuzzy ordering theory and its use in filter generalization, EURASIP J. Adv. Signal Process. (4) (2001).
[6] X.D. Jiang, Image detail-preserving filter for impulsive noise attenuation, IEE Proc.: Vis. Image Signal Process. Vis. 150 (3) (2003) 179–185.
[7] S. Akkoul, R. Ledee, R. Leconge, R. Harba, A new adaptive switching median filter, IEEE Signal Process. Lett. 17 (6) (2010) 587–590.

[8] W. Wang, P. Lu, An efficient switching median filter based on local outlier factor, IEEE Signal Process. Lett. 18 (10) (2011) 551–554.
[9] G.H. Wang, D.H. Li, W.M. Pan, Z.X. Zang, Modified switching median filter for impulse noise removal, Signal Process. 90 (12) (2010) 3213–3218.
[10] C.C. Kang, W.J. Wang, Fuzzy reasoning-based directional median filter design, Signal Process. 89 (3) (2009) 344–351.
[11] A.B. Hamza, H. Krim, Image denoising: a nonlinear robust statistical approach, IEEE Trans. Signal Process. 49 (12) (2001) 3045–3054.
[12] A. Flaig, G.R. Arce, K.E. Barner, Affine order-statistic filters: "medianization" of linear FIR filters, IEEE Trans. Signal Process. 46 (8) (1998) 2101–2112.
[13] J.B. Bednar, T.L. Watt, Alpha-trimmed means and their relationship to median filters, IEEE Trans. Acoust. Speech Signal Process. 32 (1) (1984) 145–153.
[14] Y. Lee, S. Kassam, Generalized median filtering and related nonlinear filtering techniques, IEEE Trans. Acoust. Speech Signal Process. 33 (3) (1985) 672–683.
[15] C.H. Lin, J.S. Tsai, C.T. Chiu, Switching bilateral filter with a texture/noise detector for universal noise removal, IEEE Trans. Image Process. 19 (9) (2010) 2307–2320.
[16] N. Zaric, N. Lekic, S. Stankovic, An implementation of the l-estimate distributions for analysis of signals in heavy-tailed noise, IEEE Trans. Circuits Syst. II 58 (7) (2011) 427–431.
[17] X.D. Jiang, Iterative truncated arithmetic mean filter and its properties, IEEE Trans. Image Process. 21 (4) (2012) 1537–1547.
[18] Z.W. Miao, X.D. Jiang, Further properties and a fast realization of the iterative truncated arithmetic mean filter, IEEE Trans. Circuits Syst. II: Express Briefs 59 (11) (2012) 810–814.
[19] Z.W. Miao, X.D. Jiang, Weighted iterative truncated mean filter, IEEE Trans. Signal Process. 61 (16) (2013) 4149–4160.
[20] P.M. Narendra, A separable median filter for image noise smoothing, IEEE Trans. Pattern Anal. Mach. Intell. 3 (1) (1981) 20–29.
[21] S. Kalluri, G.R. Arce, Fast algorithms for weighted myriad computation by fixed-point search, IEEE Trans. Signal Process. 48 (1) (2000) 159–171.
[22] R.C. Nunez, J.G. Gonzalez, G.R. Arce, J.P. Nolan, Fast and accurate computation of the myriad filter via branch-and-bound search, IEEE Trans. Signal Process. 56 (7) (2008) 3340–3346.
[23] S.R. Peterson, Y.H. Lee, S.A. Kassam, Some statistical properties of alpha-trimmed mean and standard type M filters, IEEE Trans. Acoust. Speech Signal Process. 36 (5) (1988) 707–713.
[24] T. Aysal, K. Barner, Generalized mean–median filtering for robust frequency-selective applications, IEEE Trans. Signal Process. 55 (3) (2007) 937–948.
[25] R. Oten, R.J.P. de Figueiredo, Adaptive alpha-trimmed mean filters under deviations from assumed noise model, IEEE Trans. Image Process. 13 (5) (2004) 627–639.
[26] G. Grimmett, D. Stirzaker, Probability and Random Processes, Oxford Univ. Press, Oxford, U.K., 2001.
[27] J.G. Gonzalez, G.R. Arce, Optimality of the myriad filter in practical impulsive-noise environments, IEEE Trans. Signal Process. 49 (2) (2001) 438–441.
[28] P.J. Huber, Robust Statistics, Wiley, New York, 1981.