Signal, Image and Video Processing (2023) 17:3585–3593
https://doi.org/10.1007/s11760-023-02584-1

ORIGINAL PAPER

Fast iterative reverse filters using fixed-point acceleration

Guang Deng[1] · Fernando Galetto[1]

Received: 1 December 2022 / Revised: 29 March 2023 / Accepted: 2 April 2023 / Published online: 1 May 2023

Abstract
Iterative reverse filters have been recently developed to address the problem of removing effects of a black box image filter. Because numerous iterations are usually required to achieve the desired result, the processing speed is slow. In this paper, we propose to use fixed-point acceleration techniques to tackle this problem. We present an interpretation of existing reverse filters as fixed-point iterations and discuss their relationship with gradient descent. We then present extensive experimental results to demonstrate the performance of fixed-point acceleration techniques named after: Anderson, Chebyshev, Irons, and Wynn. We also compare the performance of these techniques with that of gradient descent acceleration. Key findings of this work include: (1) Anderson acceleration can make a non-convergent reverse filter convergent, (2) the T-method with an acceleration technique is highly efficient and effective, and (3) in terms of processing speed, all reverse filters can benefit from one of the acceleration techniques.

Keywords Reverse image filter · Fixed-point iteration acceleration · Gradient descent acceleration

1 Introduction

In a reverse filtering problem, a filter denoted $g(.)$ is given as a black box of which the user can provide an input $\boldsymbol{x}$ and observe an output $\boldsymbol{b} = g(\boldsymbol{x})$ without knowing exactly how the filter produces certain effects. The problem is to estimate the original input signal $\boldsymbol{x}$ from the black box filter $g(.)$ and the observation $\boldsymbol{b}$.

This problem has been studied by researchers in recent years [1–8]. Methods that have been developed to solve this problem are based on the formulation of either a fixed-point iteration or the minimization of a cost function using gradient descent. In both formulations, the solutions are semi-blind in the sense that although they do not assume any knowledge of the filter, they rely on the input–output relationship of the filter. A notable very recent development is to extend the semi-blind reverse filtering problem to include noise in the filtering model [6, 7] which leads to new results in deblurring and de-filtering.

This paper presents an experimental study of accelerating this class of semi-blind reverse filtering methods. It extends our previous work [8] in which some well-known accelerated gradient descent methods [9] have been used for the acceleration. A major motivation for this work is to explore the application of acceleration techniques for fixed-point iteration to accelerate this class of filters. A further motivation comes from increasing research interests in applying such techniques in accelerating machine learning algorithms, which include techniques named after Anderson [10], Aitken [11], and Chebyshev [12, 13]. We note that numerical techniques are well established in accelerating fixed-point iterations through sequence transformation or extrapolation [14]. We refer to reference [15] for a review of algorithms dealing with fixed-point problems involving both scalar and vector variables. Reference [16] provides a historical account for the main ideas and development of acceleration techniques based on sequence transformation. Filters can benefit from one of the acceleration techniques.

As an extension to [8], the novelty of this paper is the application of fixed-point acceleration techniques in this particular problem, which is, to our best knowledge, not reported in the literature. We also make some modifications to the acceleration techniques such as a modified Chebyshev sequence and adopting a particular type of Anderson acceleration. The main contribution of this work is summarized as follows. In

✉ Fernando Galetto
f.galetto@latrobe.edu.au

Guang Deng
d.deng@latrobe.edu.au

[1] Department of Engineering, La Trobe University, Bundoora, Victoria 3086, Australia

Table 1 Reverse filters in terms of fixed-point iteration. The symbol $||.||_2$ represents matrix 2-norm

Method	Fixed-point $\boldsymbol{x} = f(\boldsymbol{x})$ (Sect. 2.2)
T [1]	$f_T(\boldsymbol{x}) = \boldsymbol{x} + e(\boldsymbol{x})$
R [3]	$f_R(\boldsymbol{x}) = \alpha\boldsymbol{x} + \lambda e(\boldsymbol{x})$
P [4]	$f_P(\boldsymbol{x}) = \boldsymbol{x} + \frac{\|\|e(\boldsymbol{x})\|\|_2}{2\|\|p(\boldsymbol{x})\|\|_2} p(\boldsymbol{x})$
TDA [8]	$f_{TDA}(\boldsymbol{x}) = \boldsymbol{x} + t(\boldsymbol{x})$
F [2]	$f_F(\boldsymbol{x}) = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\boldsymbol{b})}{\mathcal{F}(g(\boldsymbol{x}_k))\mathcal{F}(\boldsymbol{x}_k)}\right)$
p (this work)	$f_p(\boldsymbol{x}) = \boldsymbol{x} + \frac{1}{2} p(\boldsymbol{x})$

section 2.2, we formulate the T-method [1], TDA-method [8], and P-method [4] from a fixed-point iteration point of view. In addition, we derive a new iterative filter called the p-method. We have adopted a version of the Anderson acceleration [17] (Sect. 3.1). We have proposed a new way to define the Chebyshev sequence [12] (Sect. 3.2). We have also tested two vector variable acceleration methods which are related to Aitken's method [18]: Irons' method [19] and Wynn's Epsilon algorithm [20] (Sect. 3.3). In Sect. 4, we present results of reversing the effects of a guided filter and a motion blur filter and compare the performance of acceleration techniques of both fixed-point and gradient descent. We then report results of extensive evaluation of acceleration techniques for reversing effects of 14 black box filters, which are commonly used in image processing. Details of experimental setup and more results can be found in the supplementary material, which also includes further discussion of techniques used in this paper.

2 Filters and fixed-point iterations

2.1 A summary of reverse filters

We follow the terminology used in [4] where a particular reverse filter was called a method. Related previous works are summarized in Table 1 where variables are defined as the following.

- $e(\boldsymbol{x}) = \boldsymbol{b} - g(\boldsymbol{x})$
- $p(\boldsymbol{x}) = g(\boldsymbol{x} + e(\boldsymbol{x})) - g(\boldsymbol{x} - e(\boldsymbol{x}))$
- $t(\boldsymbol{x}) = g(\boldsymbol{x} + e(\boldsymbol{x})) - g(\boldsymbol{x})$

Methods in Table 1 are presented from a fixed-point iteration (detailed in Sect. 2.2) point of view. In Sect. 2.3, we briefly discuss implementations and relationships between fixed-point iteration and gradient descent in iterative reverse filters. To make the discussion self-contained, we present a brief review of each method by pointing out its main idea and advantages/limitations in the supplementary material for easy reference.

2.2 Reverse filters as fixed-point iterations

A fixed-point problem, denoted by: $\boldsymbol{x} = f(\boldsymbol{x})$, $f : \mathbb{R}^N \rightarrow \mathbb{R}^N$, can be solved by the Picard iteration

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k) \tag{1}$$

Starting with an initial guess $\boldsymbol{x}_0$, the iteration is performed until a convergence criterion is met. Formulating a reverse filter method as a fixed-point iteration may have two convergence problems: (a) the iteration may never converge and (b) the convergence is linear which can be extremely slow. This is especially problematic when the evaluation of $f(\boldsymbol{x})$ is computationally expensive [21]. We remark that there are other type of iterations such as Mann iteration [22] and segmenting Mann iteration [23] which are in the same form as the over relaxation method [12] defined as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \omega_k \left(f(\boldsymbol{x}_k) - \boldsymbol{x}_k\right) \tag{2}$$

where ω_k is a real scalar coefficient. In this work, we focus on both iterations as candidates for accelerations. Next, we derive the reverse filters from a fixed-point iteration point of view.

T-method We recall that the T-method can be derived by rewriting the filter model $\boldsymbol{b} = g(\boldsymbol{x})$ as a fixed-point problem in which

$$\boldsymbol{x} = f_T(\boldsymbol{x}) = \boldsymbol{x} + e(\boldsymbol{x}). \tag{3}$$

TDA-method From the T-method, we can write $g(\boldsymbol{x} + e(\boldsymbol{x})) - g(\boldsymbol{x}) = 0$. Adding $\boldsymbol{x}$ to both sides, we have the TDA-method

$$\boldsymbol{x} = f_{TDA}(\boldsymbol{x}) = \boldsymbol{x} + g(\boldsymbol{x} + e(\boldsymbol{x})) - g(\boldsymbol{x}) \tag{4}$$

The p-method We use a small letter "p" to distinguish between the one developed in this section and the one developed in reference [4]. When $\boldsymbol{x}$ is the fixed point, we have $e(\boldsymbol{x}) = 0$. Thus, we rewrite the filter model as

$$\boldsymbol{x} - e(\boldsymbol{x}) = \boldsymbol{x} + e(\boldsymbol{x}) \tag{5}$$

This equation can be further rewritten as

$$\boldsymbol{x} + \frac{1}{2} g(\boldsymbol{x} - e(\boldsymbol{x})) = \boldsymbol{x} + \frac{1}{2} g(\boldsymbol{x} + e(\boldsymbol{x})) \tag{6}$$

where the scaling factor $1/2$ is introduced such that a connection between the p-method and the P-method can be developed. Re-arranging, we have the following fixed-point problem

$$x = f_p(x) = x + \frac{1}{2}(g(x + e(x)) - g(x - e(x))). \tag{7}$$

Comparing the p-method with the P-method, we can see that the former is without the computational expensive factor $\frac{||e_k||_2}{2||p_k||_2}$. Also, since the P-method can be regarded as an approximation of gradient descent, the p-method can be similarly interpreted because they share the same form.

2.3 Relationship with gradient descent

Referring to Table 1, we can implement the reverse filter methods either as a Picard iteration using equation (1) or the segmenting Mann iteration using equation (2). An interesting connection between the segmenting Mann iteration and the gradient descent can be established by relating the two implementations as follows. The implementations of gradient descent (where $c(x)$ is a cost function to be minimized) and segmenting Mann iteration can be written as

- Gradient descent: $x_{k+1} = x_k + \lambda_k(-\nabla c(x_k))$
- Mann iteration: $x_{k+1} = x_k + \omega_k(f(x_k) - x_k)$

Comparing these two points of views, we can clearly see the following relationship:

$$-\nabla c(x) = f(x) - x \tag{8}$$

In addition, such relationship can be established from an optimization point view. The problem of estimating an unknown image from the observation $\boldsymbol{b}$ and a black box filter $g(x)$ can be formulated as solving a minimization problem with the cost function $c(x)$. Assume there is a least a local minimum $\bar{x}$ which satisfies the condition: $\nabla c(\bar{x}) = \mathbf{0}$. The optimization problem becomes one that solves a system of nonlinear equations. One way to solve this problem is to formulate it as a fixed-point iteration writing:

$$x = x - \nabla c(x) \tag{9}$$

In terms of fixed-point iteration defined in (1), we have $f(x) = x - \nabla c(x)$. We then perform the segmenting Mann iteration by substituting $f(x)$ into equation (2); we have the same equation as that of the gradient descent.

Therefore, a reverse filter based on the segmenting Mann iteration is equivalent to the same filter based of the gradient descent. We remark that it is well known that gradient descent can be regarded as a fixed-point iteration when we define $f(x_k) = x_k - \lambda_k \nabla c(x_k)$ where λ_k is a scaling parameter. In this section, we specifically point out the equivalence between the segmenting Mann iteration and gradient descent. This relationship allows us to explore the application of acceleration techniques from both gradient descent and fixed-point literature.

Table 2 Abbreviations of acceleration techniques

Gradient descent		Fixed-point	
ADAM	Adaptive Movement Estimation	AAcc	Anderson
MGD	Momentum	CHB	Chebyshev
NAG	Nesterov	EPSILON	Wynn
SGDR	Stochastic grad descent warn start	IRONS	Irons

3 Acceleration techniques

We review four acceleration techniques: Anderson acceleration [17], Chebyshev periodical successive over-relaxation [12], Irons acceleration [19], and Wynn's ϵ-algorithm [20]. Table 2 shows the abbreviations of acceleration techniques.

3.1 Anderson acceleration

Anderson acceleration (AAcc) [17, 21] uses the following equation to perform the iteration.

$$x_{k+1} = f(x_k) - \sum_{j=0}^{m-1} \theta_j^k (f(x_{k-m+j+1}) - f(x_{k-m+j})) \tag{10}$$

The parameters $\boldsymbol{\theta} = \{\theta_j^k\}_{j=0:m-1}$ are determined by solving a minimization problem:

$$\min_{\boldsymbol{\theta}} ||F(x_k) - \sum_{j=0}^{m-1} \theta_j^k (F(x_{k-m+j+1}) - F(x_{k-m+j}))||_2^2 \tag{11}$$

where $F(x) = f(x) - x$. The implementation does not require derivatives which makes it ideal to accelerate reverse filtering methods. In our experiment, we use a particular version[1] of the Anderson acceleration and set $m = 2$. The two optimized parameters θ_0 and θ_1 can be easily computed.

[1] https://ctk.math.ncsu.edu/TALKS/Anderson.pdf.

3.2 Chebyshev successive over-relaxation

Referring to equation (2), the Chebyshev periodical successive over-relaxation (Chebyshev) [12] uses a periodic sequence with a period of τ. The sequence is defined as:

$$\omega_k = \left[\frac{\lambda_2 + \lambda_1}{2} + \frac{\lambda_2 - \lambda_1}{2}\cos\left(\frac{2k+1}{2\tau}\pi\right)\right]^{-1} \qquad (12)$$

where λ_2 and λ_1 are the lower and upper bound of the sequence and are user-defined parameters. A recent paper also showed that using the Chebyshev sequence as the coefficients for Anderson acceleration leads to good results in machine learning applications [13]. In reference [24], it was shown that using the settings of $\lambda_1 = 0.18$, $\lambda_2 = 0.98$ and $\tau = 8$ leads to good acceleration of the T-method in reversing a mild smoothing effect of a low-pass filter operating on images of low-resolution hand-written numbers.

In our experiments, we observed that a direct application of the above Chebyshev sequence to the reverse filters did not produce satisfactory results. We found that when $\lambda_1 = 0$ and $\lambda_2 = 1$ and the value of the sequence is hard-thresholded to an upper limit α as defined in equation (13), all methods can produce consistently good results over a broad spectrum of filters to be reversed.

$$\omega_k = \min\left(\alpha, 2\left(1 + \cos\left(\frac{2k+1}{2\tau}\pi\right)\right)^{-1}\right) \qquad (13)$$

We now comment on the choice of the period τ. For a function f which is L-Lipschitz, the convergence condition is $\omega_k \le \frac{2-\delta}{1+L} < 2$ for some $\delta > 0$ (Proposition 6, [23]). In light of this convergence condition, we have the following remarks.

- Within one period, the sequence is non-linearly increasing with k. The minimum and maximum can be calculated as:
  - $\min\{\omega_k\} = \omega_0 = \frac{2}{1+\cos(\frac{\pi}{2\tau})}$
  - $\max\{\omega_k\} = \omega_{\tau-1} = \frac{2}{1-\cos(\frac{\pi}{2\tau})}$

  It is because $\cos\left(\frac{\pi}{2\tau}\right)$ is an increasing function of τ and $\cos\left(\frac{\pi}{2\tau}\right) \to 1$ when $\tau >> \pi/2$. We have $1 \le \omega_0 < \omega_k$. Similarly, the maximum value of $\omega_{\tau-1}$ is unbounded when $\tau >> \pi/2$. A bigger τ value leads to a bigger value of $\omega_{\tau-1}$.
- Setting a bigger τ value has three effects.
  - Increasing the number of iterations in which $\omega_k > 1$ before the sequence is reset to ω_0.
  - Increasing the value of ω_k when k is closed to τ.
  - Decreasing ω_0 to near 1.
- At iterations where $\omega_k > 2$, the result is pushed away from a potential local minimum [25]. In fact, the use of the Chebyshev sequence in accelerating the segmenting Mann iteration is closely related to the so-called cosine annealing with warm start.[2] Let N_τ be the number of coefficient satisfying $\omega_k > 2$. A bigger τ value will result in a bigger N_τ. We observed in our experiments that when N_τ is too large; the result is pushed too far away from the optimum, leading to sub-optimal results or unstable iteration. One way to tackle this problem is to properly choose τ such that N_τ is small and to clip the value of ω_k to a predefined value α. In our experiments, we set $\tau = 32$ and $\alpha = 3$ for T-, TDA- and p-Method. We set $\alpha = 1$ for the P-method.

3.3 Irons and Wynn acceleration

Both Irons' method [19] and Wynn's ϵ-method [20] can be regarded as a generalization of Aitken's Δ^2 process [18]. A comprehensive study of how Aitken's Δ^2 process can be generalized to deal with vector variables is presented in reference [15] which provides the following results.

- Irons' method

$$\boldsymbol{x_{k+1}} = f(f(\boldsymbol{x_k})) - \frac{(\Delta f(\boldsymbol{x_k}))^T \Delta^2 \boldsymbol{x_k}}{||\Delta^2 \boldsymbol{x_k}||_2} \Delta f(\boldsymbol{x_k})$$

- Wynn's ϵ-method

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k) + \frac{||\Delta \boldsymbol{x}_k||_2 \Delta f(\boldsymbol{x}_k) - ||\Delta f(\boldsymbol{x}_k)||_2 \Delta \boldsymbol{x}_k}{||\Delta^2 \boldsymbol{x}_k||_2}$$

where $\Delta \boldsymbol{x} = f(\boldsymbol{x}) - \boldsymbol{x}$, $\Delta f(\boldsymbol{x}) = f(f(\boldsymbol{x})) - f(\boldsymbol{x})$, and $\Delta^2 \boldsymbol{x} = \Delta f(\boldsymbol{x}) - \Delta \boldsymbol{x}$. In our implementation, the 2nd term in Irons' acceleration is modified as $\frac{(\Delta \boldsymbol{x_k})^2 \times \Delta^2 \boldsymbol{x_k}}{||\Delta^2 \boldsymbol{x_k}||_2}$. All arithmetic operations are calculated pixel-wise.

3.4 Computational complexity

Computational complexity in one iteration is presented in Table 3, where symbols $\mathcal{C}$, C, and # represent the complexity of reverse filtering method, the complexity the black box filter $g(.)$, and the number calls of $g(.)$. Methods that do not need to calculate the matrix norm have complexity: $\mathcal{C}_1 = \max(O(n), C)$ where n is the number of pixels in the image. On the other hand, methods that need to calculate the matrix norm have complexity: $\mathcal{C}_2 = \max(O(n^2), C)$. Reverse filtering methods with gradient descent acceleration and Anderson/Chebyshev acceleration have the same computational complexity as those without acceleration.

[2] See, for example: https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.CosineAnnealingLR.html.

Table 3 Computational complexity in one iteration. The 3rd row is for no acceleration or gradient descent (GD) accelerations. The last 2 rows are for fixed-point accelerations (ϵ=**EPSILON**)

	T		TDA		P		p	
Acceleration	$\mathcal{C}$	#	$\mathcal{C}$	#	$\mathcal{C}$	#	$\mathcal{C}$	#
None/GD	C_1	1	C_1	2	C_2	3	C_1	3
AAcc/CHB	C_1	1	C_1	2	C_2	3	C_1	3
IRONS/ϵ	C_2	2	C_2	4	C_2	6	C_2	6

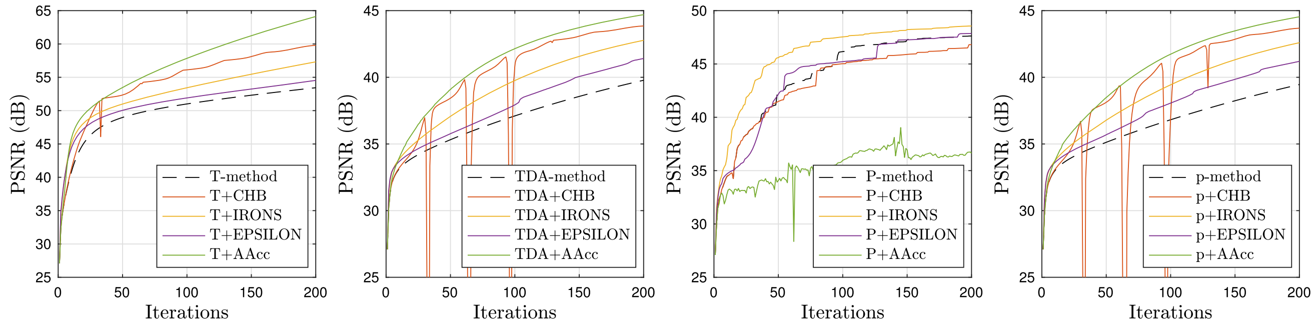
4 Results and discussion

We demonstrate and compare the effectiveness of acceleration techniques to improve the performance of 4 reverse image filters (T-, TDA-, P- and p-method). The F-method and algorithms presented in reference [7] are not considered because they assume that the black box filter $g(.)$ is locally linear. This is assumption is not true for highly nonlinear filters. Results in reversing effects of a guided filter and a motion blur filter are presented in sects. 4.1 and 4.2. Results using a dataset to evaluate the performance of acceleration techniques for 14 black box filters are summarized in sect. 4.3.

4.1 Self-guided filter

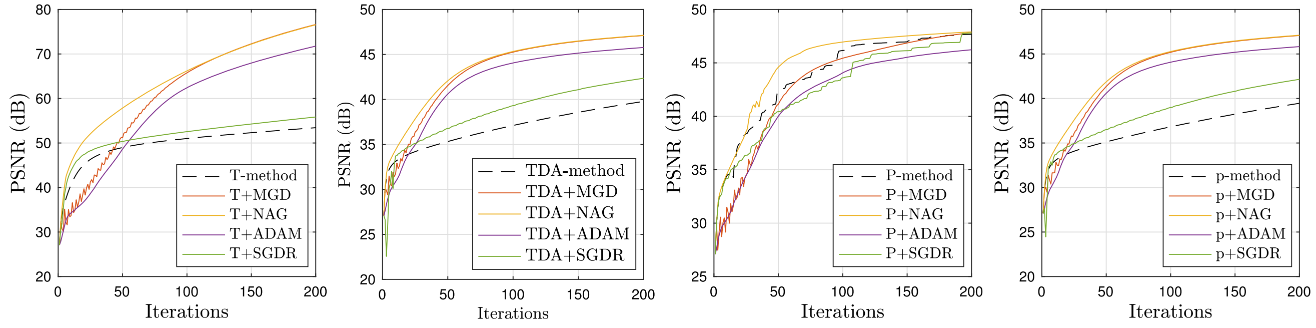
We use a guided filter [26] in a self-guided configuration to slightly blur the *cameraman.tif* image using a patch size of 5×5 and $\epsilon = 0.05$. Figure 1 shows that a higher PSNR value has been achieved for T-, TDA-, and the proposed p-method, when a fixed-point acceleration technique is applied. The effect of periodically resetting of the Chebyshev sequence can also be observed. The P-method is only improved by IRONS acceleration. For gradient descent acceleration techniques, Fig. 2 shows that T-, TDA-, and the proposed p-method are improved with the acceleration. The performance of the P-method has a slight improvement when using NAG acceleration. Other gradient descent acceleration methods do not lead to performance improvement for the P-method.

Comparing Fig. 1 with Fig. 2, we can see that in general, gradient descent acceleration leads to a better performance improvement than fixed-point acceleration. Figure 3 presents the original image and its filtered input as well as the restored versions. We only display the result of the original methods and the best performer for each acceleration type. We can see that the acceleration techniques are able to recover the details and edges smoothed by the self-guided filter.



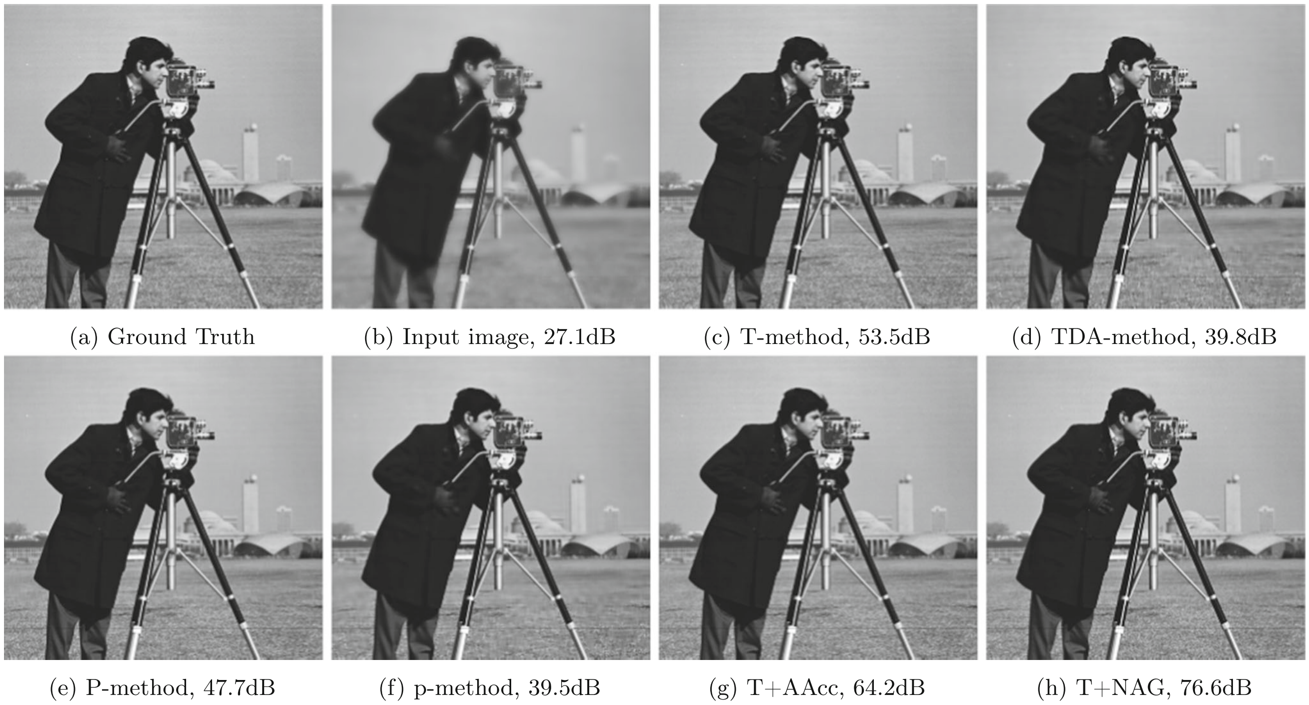
(a) T-Method (b) TDA-Method (c) P-Method (d) p-Method

Fig. 1 Reversing a self-guided filter using different reverse filtering methods and accelerated fixed-point iteration



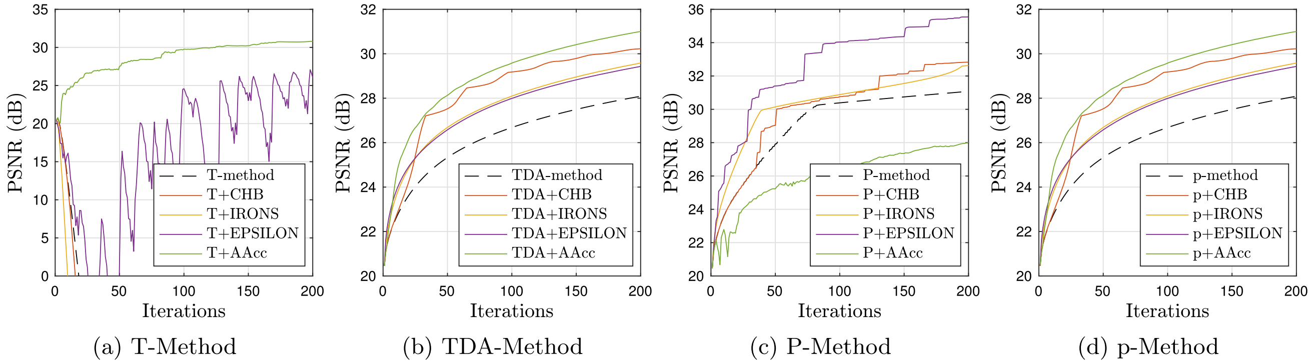
(a) T-Method (b) TDA-Method (c) P-Method (d) p-Method

Fig. 2 Reversing a self-guided filter using different reverse filtering methods and accelerated gradient descent


(a) Ground Truth (b) Input image, 27.1dB (c) T-method, 53.5dB (d) TDA-method, 39.8dB

(e) P-method, 47.7dB (f) p-method, 39.5dB (g) T+AAcc, 64.2dB (h) T+NAG, 76.6dB

Fig. 3 Result of reversing a self-guided filter with different methods after 200 iterations. **a** Ground truth. **b** Blurred image. **c–f** Results using T-, TDA-, P-, and p-methods without acceleration. **g** T+AAcc. **h** T+NAG



(a) T-Method (b) TDA-Method (c) P-Method (d) p-Method

Fig. 4 Reversing a motion blur filter with different reverse filtering methods and accelerated fixed-point iteration

4.2 Motion blur filter

An image is blurred by a filter of parameters of $l = 20$ and $\theta = 45°$. Experimental results are presented in Figs. 4 and 5. The original T-method is not able to reverse the effect of this filter since the PSNR decreases after each iteration. Anderson acceleration (AAcc) helps to achieve a steady improvement in PSNR. The EPSILON acceleration also achieves improvement but has significant fluctuations in PSNR. All other methods fail to make the T-method convergent. This result highlights the importance of studying fixed-point acceleration techniques in reverse filtering. Performances of methods of TDA, P and p are improved by all acceleration techniques. The PSNR values achieved by applying gradient descent acceleration techniques are higher than those achieved by applying fixed-point acceleration.

Figure 6 shows results without applying acceleration and two best results with accelerations which are P+EPSILON and P+NAG. The T-method leads to a totally distorted image after 200 iterations, while TDA-, P- and p-method are able to recover most of the missing details. However, some artifacts or distortion appeared on the girl's face. Comparing Fig. 6e with the best results shown in Figs. 6g and h, we can see that with the same number of iterations, acceleration techniques lead to significant improvement in image quality (Table 4).

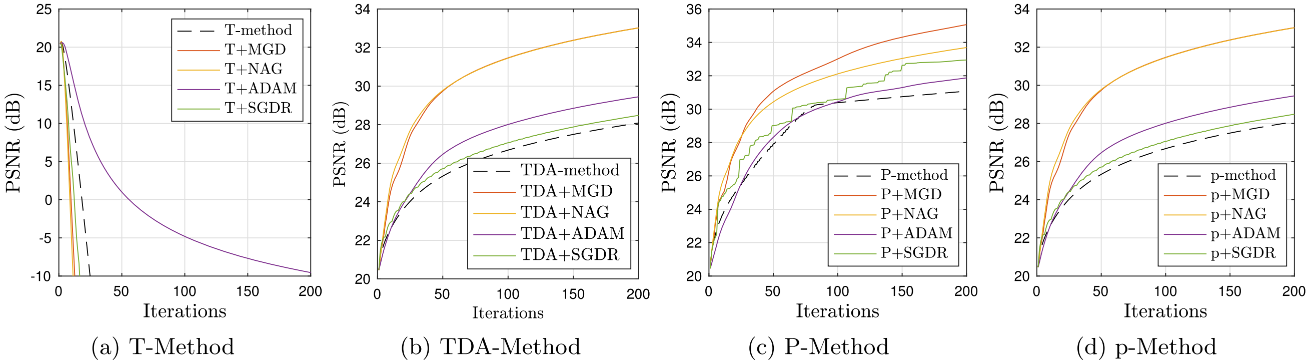


Fig. 5 Reversing a motion blur filter with different reverse filtering methods and accelerated gradient descent

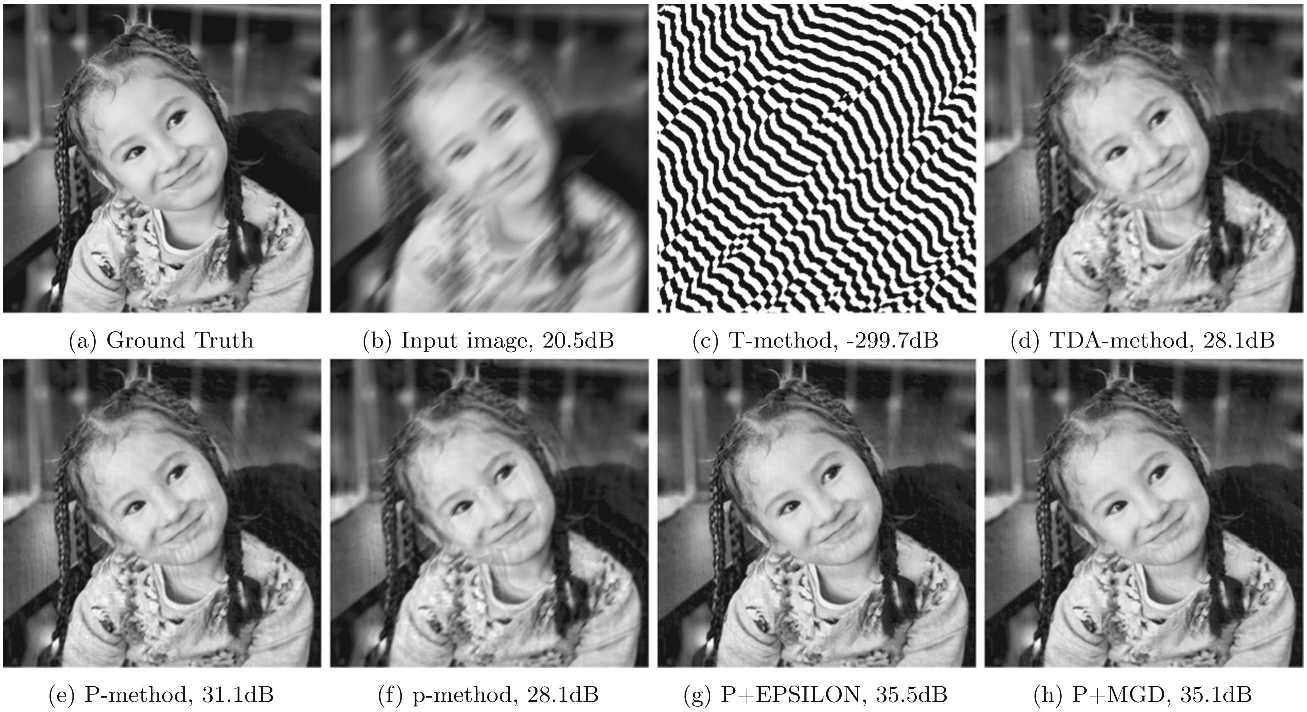


Fig. 6 Result of reversing a motion blur filter with different methods after 200 iterations. **a** Ground truth. **b** Blurred image. **c–f** Results using T-, TDA-, P-, and p-methods without acceleration. **g** P+EPSILON. **h** P+MGD

Table 4 Parameter settings for acceleration method. Irons, Epsilon, and Anderson accelerations are parameter-free

Accel.	T	TDA	P	p
NAG/MGD	$\alpha = 1, \beta = 0.9$	$\alpha = 1, \beta = 0.9$	$\alpha = 1, \beta = 0.9$	$\alpha = 1, \beta = 0.9$
ADAM	$\beta_1 = 0.9, \beta_2 = 0.999$	$\beta_1 = 0.9, \beta_2 = 0.999$	$\beta_1 = 0.9, \beta_2 = 0.999$	$\beta_1 = 0.9, \beta_2 = 0.999$
SGDR	$T = 5, \lambda_{min} = 1, \lambda_{max} = 2$	$T = 5, \lambda_{min} = 0, \lambda_{max} = 3$	$T = 5, \lambda_{min} = 0, \lambda_{max} = 1$	$T = 5, \lambda_{min} = 0, \lambda_{max} = 3$
CHB	$\alpha = 3, T = 32$	$\alpha = 3, T = 32$	$\alpha = 1, T = 32$	$\alpha = 3, T = 32$

4.3 Results of reversing 14 black box filters

We perform an extensive evaluation of the performance of the reverse filters and acceleration techniques by using 14 black box filters and 20 images. Experimental setup is detailed in the supplementary material. Let $p_k(i)$ and $p_0(i)$ represent, respectively, the PSNR value at the kth iteration and 0th iteration when reversing the effect of a filter on the ith image. We define the percentage of improvement in PSNR as:

$$\bar{p}_k(i) = \frac{p_k(i) - p_0(i)}{p_0(i)} \times 100 \tag{14}$$

Table 5 Test results measured by the percentage of improvement in PSNR. Only results from the best performing acceleration method are listed. Fixed-point acceleration methods are highlighted with italic

Filter	T		TDA		P		**p**	
WF [27]	29.9	*AAcc*	35.7	NAG	**40.2**	*IRONS*	35.6	NAG
Disk averaging filter	36.5	*AAcc*	38.7	MGD	**46.6**	*EPSILON*	38.7	MGD
Motion blur filter	36.3	*AAcc*	43.7	MGD	**51.7**	*EPSILON*	43.7	MGD
GF [26]	**200.9**	NAG	78.9	*AAcc*	95.7	*IRONS*	78.4	*AAcc*
GF+Gauss. filt.	**49.2**	MGD	16.7	MGD	21.4	*IRONS*	16.7	MGD
SSIF [28]	**1059.4**	*AAcc*	936.5	*AAcc*	1053.4	*IRONS*	948.0	*AAcc*
WLS [29]	**23.0**	SGDR	11.1	ADAM	12.0	ADAM	11.6	NAG
RTV [30]	**35.9**	SGDR	16.5	*IRONS*	20.6	*IRONS*	16.0	*IRONS*
Gauss. filt.	10.2	*AAcc*	8.9	MGD	**11.5**	*IRONS*	8.9	MGD
AMF [31]	**99.7**	*IRONS*	57.7	NAG	60.0	NAG	58.7	NAG
ILS [32]	**59.2**	NAG	29.4	NAG	31.6	*IRONS*	29.0	NAG
L0 [33]	**11.2**	*EPSILON*	6.0	*EPSILON*	6.1	*EPSILON*	6.1	*EPSILON*
BF [34]	**105.4**	*AAcc*	58.1	ADAM	60.8	ADAM	58.2	ADAM
LLF [35]	**38.1**	*AAcc*	25.4	*EPSILON*	24.6	*EPSILON*	26.9	*AAcc*

Bold values indicate the best performing method for each filter

For each image, there is a sequence $\bar{p}_k(i)$. Let $h(i) = \max_k\{\bar{p}_k(i)\}$ be the maximum percentage of improvement achieved for each image at a certain iteration index. We average the results over 20 images as the final measure of the performance:

$$p = \frac{1}{20}\sum_{i=1}^{20} h(i) \tag{15}$$

We compute the performance index p for each combination of reverse filtering methods and acceleration techniques. The best performing acceleration techniques are shown in Table 5. Other results are presented in supplementary material. We have the following observations.

- All reverse filters are improved to some extent by at least one acceleration technique.
- When the T-method converges, its accelerated versions achieve higher PSNR improvement than TDA-, P- and p-methods. When it does not converge, only Anderson acceleration makes it converge. Other acceleration techniques fail to make it converge.
- Overall, the T-method together with a particular acceleration has the best performance in 10 out of 14 filters. It is also the least complex method. Both fixed-point and gradient descent acceleration techniques contribute to the success of the T-method.
- Out of the 56 cases, gradient descent accelerations are the winners in 26 cases, while fixed-point accelerations are the winners for 30 cases. Anderson and EPSILON/IRONS accelerations perform well with the T-method and the P-method, respectively. Gradient descent accelerations perform well with TDA-method and p-method.
- It is a challenge task to reverse the effect of certain filters such as L0 and Gaussian filter with large standard deviation. The average maximum improvement in PSNR is around 10%.

5 Conclusion

Although fixed-point acceleration is well developed in numerical methods, its applications in signal and image processing are emerging and its application in reverse filtering presented in this paper is new. We have reformulated several reverse filters as fixed-point iterations. We have conducted extensive experiments to demonstrate the performance of such techniques and made a comparison with gradient descent accelerations. We show that (1) the Anderson acceleration is the only technique which can make the otherwise non-convergent T-method convergent, (2) when it is convergent, the T-method together with an acceleration technique (either fixed-point or gradient descent) in most cases outperforms other reverse filtering methods, and (3) effects of some image filters are more difficult to reverse and the achievable PSNR improvement is about 10% or less. While the success of Anderson acceleration in making a non-convergent T-method convergent highlights the importance of this work, it also calls for further study of the mathematical property behind the success.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11760-023-02584-1.

Author Contributions G.D conceived of the presented idea and developed the theory. F. G performed the experiments, computations and prepared the figures. All authors discussed the results, wrote and review the main manuscript text.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability The source code for the current study is available at https://github.com/fergaletto/FastReverseFilters

Declarations

Conflict of interest Authors have no competing interests to declare that are relevant to the content of this article.

Ethical approval Not applicable.

References

1. Tao, X., Zhou, C., Shen, X., Wang, J., Jia, J.: Zero-order reverse filtering. In Proc. IEEE ICCV, 222–230, (2017)
2. Dong, L., Zhou, J., Zou, C., Wang, Y.: Iterative first-order reverse image filtering. In Proc. ACM Turing Celebration Conf.-China, 1–5, (2019)
3. Milanfar, P.: Rendition: Reclaiming what a black box takes away, arXiv preprint arXiv:1804.08651, (2018)
4. Belyaev, A.G., Fayolle, P.-A.: Two iterative methods for reverse image filtering. Signal, Image and Video Processing **15**, 1565–1573 (2021)
5. Deng, G., Broadbridge, P.: Bregman inverse filter. Electron. Lett. **55**(4), 192–194 (2019)
6. Wang, L., Fayolle, P.-A., Belyaev, A.G.: Reverse Image Filtering with Clean and Noisy Filters. Signal, Image and Video Processing (2022)
7. Belyaev, A.G., Fayolle, P.-A.: Black-box image deblurring and defiltering. Signal Proc. Image Commun. **108**, 116833 (2022)
8. Galetto, F.J., Deng, G.: Reverse image filtering using total derivative approximation and accelerated gradient descent. IEEE Access **10**, 124928–124944 (2022)
9. Kochenderfer, M.J., Wheeler, T.A.: Algorithms for Optimization. The MIT Press, (2019)
10. Zhang, J., O'Donoghue, B., Boyd, S.P.: Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations. SIAM J. Optim. **30**, 3170–3197 (2020)
11. Chen, J., Gan, M., Zhu, Q., Narayan, P., Liu, Y.: Robust standard gradient descent algorithm for ARX models using Aitken acceleration technique. IEEE Trans Cybern **52**(9), 9646–9655 (2021)
12. Wadayama, T., Takabe, S.: Chebyshev periodical successive overrelaxation for accelerating fixed-point iterations. IEEE Signal Process. Lett. **28**, 907–911 (2021)
13. Li, Z., Li, J.: A fast Anderson-Chebyshev acceleration for nonlinear optimization. In Proc. 23rd Int. Conf. AI and Statistics, 1047–1057 (2020)
14. Bornemann, F., Laurie, D., Wagon, S., Waldvogel, J.: The SIAM 100-Digit Challenge A Study in High-Accuracy Numerical Computing. SIAM, (2004)
15. Ramière, I., Helfer, T.: Iterative residual-based vector methods to accelerate fixed point iterations. Comput. Math. with Appl. **70**, 2210–2226 (2015)
16. Brezinski, C.: Convergence acceleration during the 20th century. J. Comput. Appl. Math. **122**, 1–21 (2000)
17. Anderson, D.G.: Iterative procedures for nonlinear integral equations. J. ACM **12**, 547–560 (1965)
18. Aitken, A.: XXV.-On Bernoulli's numerical solution of algebraic equations, Proc. Roy. Soc., Edinburgh, pp., 289 – 305, (1927)
19. Irons, B.M., Tuck, R.C.: A version of the Aitken accelerator for computer iteration. Int. J. Numer. Methods Eng. **1**(3), 275–277 (1969)
20. Wynn, P.: Acceleration techniques for iterated vector and matrix problems. Math. Comput. **16**(79), 301–322 (1962)
21. Walker, H.F., Ni, P.: Anderson acceleration for fixed-point iterations. SIAM J. Numer. Aanal. **49**(4), 1715–1735 (2011)
22. Mann, W.R.: Mean value methods in iteration. Proc. Am. Math. Soc. **4**, 506–510 (1953)
23. Borwein, D., Borwein, J.: Fixed point iterations for real functions. J. Math. Anal. Appl. **157**, 112–126 (1991)
24. Wadayama, T., Takabe, S.: Chebyshev inertial iteration for accelerating fixed-point iterations, arXiv preprint arXiv:2001.03280, (2020)
25. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts, arXiv preprint arXiv:1608.03983, (2016)
26. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE Trans. Pattern Anal. Mach. Intell. **35**(6), 1397–1409 (2012)
27. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Pearson Higher Ed., (2011)
28. Deng, G., Galetto, F., Al-nasrawi, M., Waheed, W.: A guided edge-aware smoothing-sharpening filter based on patch interpolation model and generalized gamma distribution. IEEE Open J. Signal Proc. **2**, 119–135 (2021)
29. Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. ACM Trans. Graph. **27**(3), 1–10 (2008)
30. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. ACM Trans. Graph. **31**(6), 1–10 (2012)
31. Gastal, E.S., Oliveira, M.M.: Adaptive manifolds for real-time high-dimensional filtering. ACM Trans. Graph. **31**(4), 1–13 (2012)
32. Liu, W., Zhang, P., Huang, X., Yang, J., Shen, C., Reid, I.: Real-time image smoothing via iterative least squares. ACM Trans. Graph. **39**(3), 1–24 (2020)
33. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via L_0 gradient minimization. ACM Trans. Graph. **30**(6), 1–12 (2011)
34. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In Proc. IEEE ICCV, pp. 839–846 (1998)
35. Paris, S., Hasinoff, S.W., Kautz, J.: Local Laplacian filters: edge-aware image processing with a Laplacian pyramid. ACM Trans. Graph. **30**(4), 68 (2011)