SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

August 2015, Vol. 58 082306:1–082306:13
doi: 10.1007/s11432-014-5244-z

Efficient InSAR phase noise reduction via total variation regularization

LUO XiaoMei[1,2*], WANG XiangFeng[3], SUO ZhiYong[4] & LI ZhenFang[4]

[1]*State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an* 710071*, China;*
[2]*Department of Electronic Information Engineering, Nanchang University, Nanchang* 330031*, China;*
[3]*Shanghai Key Laboratory of Trustworthy Computing, Software Engineering Institute, East China Normal University, Shanghai* 200062*, China;*
[4]*National Laboratory of Radar Signal Processing, Xidian University, Xi'an* 710017*, China*

Received July 29, 2014; accepted September 1, 2014; published online March 5, 2015

Abstract We consider the phase noise filtering problem for Interferometric Synthetic Aperture Radar (InSAR) using a total variation regularized complex linear least squares formulation. Although the original formulation is convex, solving it directly with the standard CVX package is time consuming due to the large problem size. In this paper, we introduce the effective and efficient alternating direction method of multipliers (ADMM) to solve the equivalent well-defined complex formulation for the real and imaginary parts of the optimization variables. Both the iteration complexity and the computational complexity of the ADMM are established in the forms of theorems for our InSAR phase noise problem. Simulation results based on simulated and measured data show that this new InSAR phase noise reduction method not only is 3 orders of magnitude faster than the standard CVX solver, but also has a much better performance than the several existing phase filtering methods.

Keywords interferometric synthetic aperture radar, phase noise reduction, total variation regularization, least squares formulation, alternating direction method of multipliers

Citation Luo X M, Wang X F, Suo Z Y, et al. Efficient InSAR phase noise reduction via total variation regularization. Sci China Inf Sci, 2015, 58: 082306(13), doi: 10.1007/s11432-014-5244-z

1 Introduction

Typical InSAR interferogram data is deteriorated by various noises. It is well known that phase noise reduction [1–8] of InSAR phase images is an important step prior to phase unwrapping [9,10]. The mean filter [1,2] is a commonly used spatial domain filter for InSAR noise reduction. Though simple and fast, it will blur edges and fine details in the filtered images. Two classical popular filtering methods are the Lee filter [3] (see [4] for an improved version) and the Goldstein filter [5] (see [6] for improved versions). Though the Lee filter can achieve a higher filtering accuracy by selecting independent and identically distributed (i.i.d.) samples along the fringe orientation in the scenario with speckle noise as well as phase noise [3], it is computationally less efficient due to its local phase unwrapping. The Goldstein filter has strong smoothing capability and fast operation, however, it sometimes can not represent phase noise intensity by its coherent values. A weighted maximum likelihood estimator based on nonlocal techniques

* Corresponding author (email: xxmluo@gmail.com)

is applied to estimate the InSAR data [7]. Though this nonlocal nonlinear filtering method is effective for denoising, the attenuation of thin and dark details still exist in the regularized images. An improved nonlocal nonlinear filtering method is proposed in [8].

Despite the availability of various algorithms for InSAR phase noise filtering, a general systematic approach is still lacking. In this paper, we study the phase noise filtering problem in InSAR based on the Bayesian statistical formulation [11]. The main contributions of this paper are as follows. Firstly, we propose a general systematic design method for InSAR noise reduction. Specifically, a convex Bayesian approach to process InSAR data is proposed via a complex linear least squares minimization and a total variation regularization [12,13]. Secondly, due to the size of the problem, it is time consuming to solve this convex formulation only with the standard CVX package [14]. A new phase filtering formulation is given and ADMM is introduced to effectively and efficiently solve the equivalent well-defined reformulation. Thirdly, in order to analyze the performance of the ADMM, both the iteration and the computational complexities of this algorithm are established by theorems. Fourthly, we provide numerical examples based on both simulated and measured data. Simulation results show that this new InSAR phase noise reduction method not only is 3 orders of magnitude faster than the standard CVX solver, but also has a superior performance than several existing phase filtering methods.

2 Phase filtering formulation

Consider a complex interferogram whose size is $M \times M$. If we stack this InSAR data according to the columns, the complex interferogram model with additive noise [15] can be denoted as follows:

$$\boldsymbol{y}(k) = \boldsymbol{x}(k) + \boldsymbol{n}(k),\ k = 1, 2, \ldots, N, \tag{1}$$

where k denotes the pixel's series number, $\boldsymbol{x}(k)$ and $\boldsymbol{y}(k)$ denote the ideal and measured InSAR phase (complex form), respectively, $\boldsymbol{n}(k)$ denotes additive noise and N equals M^2.

InSAR phase noise filtering is to estimate $\boldsymbol{x}(k)$ from $\boldsymbol{y}(k)$. According to (1), the Bayesian estimator [11] of $\boldsymbol{x}$ is simply given by the least squares estimator, i.e.,

$$\hat{\boldsymbol{x}}_{\text{Bayesian}} := \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2. \tag{2}$$

Without any regularization, the above estimator is simply $\hat{\boldsymbol{x}}_{\text{Bayesian}} = \boldsymbol{y}$ itself, which is clearly unacceptable. We must impose some prior information (or structures) on the recovered image.

As we have known, SAR data usually has sparsity [16,17] while InSAR data commonly does not have such property. For most of the interferograms, there exists sparsity (few zeros) in the gradient vectors of the complex pixels. Hence, we use a total variation regularization term to promote sparsity of the gradient [12], leading to the following convex formulation:

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 + \lambda_1 \sum_{i=1}^{N} \sqrt{(\nabla x_{1i}^h)^2 + (\nabla x_{1i}^v)^2} + \lambda_2 \sum_{i=1}^{N} \sqrt{(\nabla x_{2i}^h)^2 + (\nabla x_{2i}^v)^2}, \tag{3}$$

where ∇x_{1i} and ∇x_{2i} denote the gradients of the real and imaginary parts of the pixel x_i, respectively, while the superscripts h and v denote the horizonal and vertical derivatives, respectively. λ_1 and λ_2 are two penalty parameters not only balancing the corresponding regularization and data fidelity, respectively, but also controlling the sparsity of the image [12]. Usually, the larger the parameters λ_1 and λ_2, the more sparser the image.

It is obvious that the optimization problem (3) is a convex minimization problem, but the difficulty of solving the problem lies in its non-smoothness. Although the CVX package can be introduced to solve (3), a new algorithm must be designed competent to the high computational complexity caused by the CVX.

3 A total variation (TV) filter and ADMM

Recall the formulation (3), it is a regularized least squares optimization problem. For total variation regularization, we use the splitting technique [18] to attain the variable-decoupled convex optimization problem with equality constraints. Further we present the well-known *Alternating Direction Method of Multipliers* (ADMM) [19,20] to solve the new reformulation of the optimization problem (3).

First, the objective function of the optimization problem (3) can be written as

$$\frac{1}{2}\|\boldsymbol{y}_c-\boldsymbol{x}_c\|_2^2+\lambda_1\sum_{i=1}^{N}\sqrt{(\nabla x_{1i}^h)^2+(\nabla x_{1i}^v)^2}+\lambda_2\sum_{i=1}^{N}\sqrt{(\nabla x_{2i}^h)^2+(\nabla x_{2i}^v)^2}, \tag{4}$$

where $\boldsymbol{x}_c=[\boldsymbol{x}_1^{\mathrm{T}}\ \boldsymbol{x}_2^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{y}_c=[\boldsymbol{y}_1^{\mathrm{T}}\ \boldsymbol{y}_2^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ denote the real and imaginary parts of $\boldsymbol{x}$, respectively.

Second, introducing two new variables $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ and using the splitting technique, we can rewrite (3) as

$$\begin{aligned}&\min_{\boldsymbol{x}_c}\ h(\boldsymbol{x}_c):=\frac{1}{2}\left\|\boldsymbol{y}_c-\boldsymbol{x}_c\right\|_2^2+\lambda_1\left\|\boldsymbol{v}_1\right\|_{1,2}+\lambda_2\left\|\boldsymbol{v}_2\right\|_{1,2},\\&\text{s.t.}\ \ \boldsymbol{v}_1=\nabla\boldsymbol{x}_1,\ \ \boldsymbol{v}_2=\nabla\boldsymbol{x}_2,\end{aligned} \tag{5}$$

where the notation $\|\cdot\|_{1,2}$ norm is defined as

$$\|\boldsymbol{u}\|_{1,2}=\sum_{i=1}^{N}\sqrt{(\boldsymbol{u}_i^h)^2+(\boldsymbol{u}_i^v)^2}$$

for any real-valued image $\boldsymbol{u}=(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_N)\in\mathbb{R}^{2N}$ with $\boldsymbol{u}_i=(u_i^h,u_i^v)\in\mathbb{R}^2$.

In order to well solve the optimization variables $\boldsymbol{x}_c,\boldsymbol{v}_1,\boldsymbol{\mu}_1,\boldsymbol{v}_2,\boldsymbol{\mu}_2$, we do not simply resort to Lagrangian function [21]. Under the augmented Lagrangian framework [22], the pursuit of $\boldsymbol{x}_c,\boldsymbol{v}_1,\boldsymbol{\mu}_1,\boldsymbol{v}_2,\boldsymbol{\mu}_2$ is equivalent to solving the following optimization problem

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}_c,\boldsymbol{v}_1,\boldsymbol{\mu}_1,\boldsymbol{v}_2,\boldsymbol{\mu}_2)=&\lambda_1\|\boldsymbol{v}_1\|_{1,2}+\lambda_2\|\boldsymbol{v}_2\|_{1,2}+\frac{1}{2}\|\boldsymbol{y}_c-\boldsymbol{x}_c\|_2^2+\boldsymbol{\mu}_1^{\mathrm{T}}(\boldsymbol{v}_1-\nabla\boldsymbol{x}_1)+\boldsymbol{\mu}_2^{\mathrm{T}}(\boldsymbol{v}_2-\nabla\boldsymbol{x}_2)\\&+\frac{\beta_1}{2}\|\boldsymbol{v}_1-\nabla\boldsymbol{x}_1\|_2^2+\frac{\beta_2}{2}\|\boldsymbol{v}_2-\nabla\boldsymbol{x}_2\|_2^2,\end{aligned} \tag{6}$$

where $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ denote the Lagrangian multipliers and $\beta_1>0$, $\beta_2>0$ are penalty parameters of the linear constraint violations.

Combined with the method of multiplier, the optimization problem (6) can be decoupled into separable structures by introducing Gauss-Seidel iteration scheme into the algorithm framework. This indicates that the subproblems of $\boldsymbol{x}_c$ and $\boldsymbol{v}$ should be separately. Generally, it is much easier to solve the separated subproblems about $\boldsymbol{x}_c$ and $\boldsymbol{v}$ which may have closed-form solutions. Hence, the basic iteration scheme of two blocks ADMM can be naturally obtained as follows:

$$\boldsymbol{v}_1^{k+1}=\arg\min_{\boldsymbol{v}_1}\lambda_1\|\boldsymbol{v}_1\|_{1,2}+\frac{\beta_1}{2}\|\boldsymbol{v}_1-\nabla\boldsymbol{x}_1^k+\frac{\boldsymbol{\mu}_1^k}{\beta_1}\|_2^2, \tag{7}$$

$$\boldsymbol{v}_2^{k+1}=\arg\min_{\boldsymbol{v}_2}\lambda_2\|\boldsymbol{v}_2\|_{1,2}+\frac{\beta_2}{2}\|\boldsymbol{v}_2-\nabla\boldsymbol{x}_2^k+\frac{\boldsymbol{\mu}_2^k}{\beta_2}\|_2^2, \tag{8}$$

$$\boldsymbol{x}_c^{k+1}=\arg\min_{\boldsymbol{x}_c}\|\boldsymbol{y}_c-\boldsymbol{x}_c\|_2^2+\beta_1\|\boldsymbol{v}_1^{k+1}-\nabla\boldsymbol{x}_1+\frac{\boldsymbol{\mu}_1^k}{\beta_1}\|_2^2+\beta_2\|\boldsymbol{v}_2^{k+1}-\nabla\boldsymbol{x}_2+\frac{\boldsymbol{\mu}_2^k}{\beta_2}\|_2^2, \tag{9}$$

$$\boldsymbol{\mu}_1^{k+1}=\boldsymbol{\mu}_1^k+\beta_1(\boldsymbol{v}_1^{k+1}-\nabla\boldsymbol{x}_1^{k+1}), \tag{10}$$

$$\boldsymbol{\mu}_2^{k+1}=\boldsymbol{\mu}_2^k+\beta_2(\boldsymbol{v}_2^{k+1}-\nabla\boldsymbol{x}_2^{k+1}). \tag{11}$$

Eqs. (7)–(9) all have closed-form solutions as follows:

$$\boldsymbol{v}_1^{k+1}=\left(\nabla\boldsymbol{x}_1^k-\frac{\boldsymbol{\mu}_1^k}{\beta_1}\right)-\min\left(\frac{\lambda_1}{\beta_1},\left|\nabla\boldsymbol{x}_1^k-\frac{\boldsymbol{\mu}_1^k}{\beta_1}\right|\right)\cdot\frac{\nabla\boldsymbol{x}_1^k-\frac{\boldsymbol{\mu}_1^k}{\beta_1}}{|\nabla\boldsymbol{x}_1^k-\frac{\boldsymbol{\mu}_1^k}{\beta_1}|}, \tag{12}$$

$$\boldsymbol{v}_2^{k+1} = \left(\nabla \boldsymbol{x}_2^k - \frac{\boldsymbol{\mu}_2^k}{\beta_2}\right) - \min\left(\frac{\lambda_2}{\beta_2}, \left|\nabla \boldsymbol{x}_2^k - \frac{\boldsymbol{\mu}_2^k}{\beta_2}\right|\right) \cdot \frac{\nabla \boldsymbol{x}_2^k - \frac{\boldsymbol{\mu}_2^k}{\beta_2}}{|\nabla \boldsymbol{x}_2^k - \frac{\boldsymbol{\mu}_2^k}{\beta_2}|}, \tag{13}$$

$$\begin{aligned}\boldsymbol{x}_c^{k+1} = & \left[\boldsymbol{I} + \beta_1(\boldsymbol{B}\boldsymbol{A}_1)^{\mathrm{T}}(\boldsymbol{B}\boldsymbol{A}_1) + \beta_2(\boldsymbol{B}\boldsymbol{A}_2)^{\mathrm{T}}(\boldsymbol{B}\boldsymbol{A}_2)\right]^{-1} \\ & \cdot\left[\boldsymbol{y}_c + \beta_1(\boldsymbol{B}\boldsymbol{A}_1)^{\mathrm{T}}\boldsymbol{v}_1^{k+1} + \beta_2(\boldsymbol{B}\boldsymbol{A}_2)^{\mathrm{T}}\boldsymbol{v}_2^{k+1} + (\boldsymbol{B}\boldsymbol{A}_1)^{\mathrm{T}}\boldsymbol{\mu}_1^k + (\boldsymbol{B}\boldsymbol{A}_2)^{\mathrm{T}}\boldsymbol{\mu}_2^k\right],\end{aligned} \tag{14}$$

where $\nabla \boldsymbol{x}_1$ and $\nabla \boldsymbol{x}_2$ denote the real and imaginary parts of the gradient vector of $\boldsymbol{x}$, respectively, $\boldsymbol{A}_1 = [\boldsymbol{I}\ \boldsymbol{0}]$, $\boldsymbol{A}_2 = [\boldsymbol{0}\ \boldsymbol{I}]$ and $\boldsymbol{B}$ is the gradient generating matrix satisfying $\nabla \boldsymbol{x}_1 = \boldsymbol{B}\boldsymbol{A}_1\boldsymbol{x}_c$ and $\nabla \boldsymbol{x}_2 = \boldsymbol{B}\boldsymbol{A}_2\boldsymbol{x}_c$. The proof of the close-form solutions (12)–(14) is shown in Appendix A.

ADMM was first introduced in [19,20] and has been widely introduced to various research fields [18,23]. Further, ADMM has theoretically analyzed in [24–26]. The general convergence and iteration complexity results are shown as follows,

(1) Residual convergence: $\boldsymbol{v}_1^k - \nabla \boldsymbol{x}_1^k \to 0$ and $\boldsymbol{v}_2^k - \nabla \boldsymbol{x}_2^k \to 0$ as $k \to \infty$.

(2) Objective convergence: $\lim_{k\to\infty} h(\boldsymbol{x}_c^k) = p^*$, where p^* is the optimal value of (5).

(3) Iterate convergence: the iterates $\boldsymbol{x}_c^k$, $\boldsymbol{v}_1^k$, $\boldsymbol{\mu}_1^k$, $\boldsymbol{v}_2^k$ and $\boldsymbol{\mu}_2^k$ converge to their respective optimal solutions.

In order to obtain the global computational complexity of the optimization problem (6), we first give its iteration complexity result, which means that the obtained point $(\hat{\boldsymbol{v}}, \hat{\boldsymbol{x}}_c, \hat{\boldsymbol{\mu}})$ satisfies the following ϵ-optimality condition

$$\mathcal{L}(\hat{\boldsymbol{v}}, \hat{\boldsymbol{x}}_c, \boldsymbol{\mu}) - \mathcal{L}(\boldsymbol{v}, \boldsymbol{x}_c, \hat{\boldsymbol{\mu}}) \leqslant \epsilon,\ \forall\, \boldsymbol{v},\ \boldsymbol{x}_c,\ \boldsymbol{\mu}, \tag{15}$$

where $\mathcal{L}$ is the augmented Lagrangian function, $\boldsymbol{v} = [\boldsymbol{v}_1^{\mathrm{T}}\ \boldsymbol{v}_2^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{\mu} = [\boldsymbol{\mu}_1^{\mathrm{T}}\ \boldsymbol{\mu}_2^{\mathrm{T}}]^{\mathrm{T}}$, $\hat{\boldsymbol{v}} = [\hat{\boldsymbol{v}}_1^{\mathrm{T}}\ \hat{\boldsymbol{v}}_2^{\mathrm{T}}]^{\mathrm{T}}$ and $\hat{\boldsymbol{\mu}} = [\hat{\boldsymbol{\mu}}_1^{\mathrm{T}}\ \hat{\boldsymbol{\mu}}_2^{\mathrm{T}}]^{\mathrm{T}}$. The iteration complexity of ADMM was first established in [26]. And in the following theorem, we give the iteration complexity of the optimization problem (6).

Theorem 1. Let $\{\boldsymbol{v}^k, \boldsymbol{x}_c^k, \boldsymbol{\mu}^k\}$ be the sequence generated by ADMM for (6). In the $(K+1)$th iteration, for all $\boldsymbol{v}, \boldsymbol{x}_c, \boldsymbol{\mu}$, an ergodic point $(\hat{\boldsymbol{v}}^{K+1}, \hat{\boldsymbol{x}}_c^{K+1}, \hat{\boldsymbol{\mu}}^{K+1})$ defined by $\{\boldsymbol{v}^i, \boldsymbol{x}_c^i, \boldsymbol{\mu}^i\}_{i=1}^{K+1}$ satisfies

$$\mathcal{L}(\hat{\boldsymbol{v}}^{K+1}, \hat{\boldsymbol{x}}_c^{K+1}, \boldsymbol{\mu}) - \mathcal{L}(\boldsymbol{v}, \boldsymbol{x}_c, \hat{\boldsymbol{\mu}}^{K+1}) \leqslant \frac{C}{K+1}. \tag{16}$$

Proof. The proof is shown in Appendix B.

The above theorem shows the iteration complexity of ADMM for our optimization formulation in (6), and it is the whole perspective of ADMM convergent speed. If the per iteration computation can be calculated, we can obtain the entire computational complexity in order to get an ϵ-optimal solution. In the following, we present the related computational complexity result.

Theorem 2. The computational complexity of (7)–(11) is $O^*(28N^4)$, where O^* denotes that lower order terms are ignored and N denotes the number of pixels in a filtering image.

Proof. The proof is shown in Appendix C.

For the linear filters based on sample average, all the used samples satisfy i.i.d. condition. Even if the samples fail to satisfy this condition due to terrain fluctuation, we can compensate them by some frequency estimation method, which leads to small influence on frequency resolution. From (12)–(14), it is easy to know that our designed filtering method is linear. The general analysis of convergence and iteration complexity results of ADMM for a convex problem shown above indicate that we can obtain the accurate solution of the optimization problem (3) by using (7)–(9) in iteration scheme. According to the optimization problem (3), phase image resolution is affected by the parameters λ_1 and λ_2. With the increment of the λ_1 and λ_2, more noise will be filtered from the phase image while the phase image resolution is almost unaffected. But if the λ_1 and λ_2 are selected too large, the filtered image may be over-filtered and can produce a blocky effect. This indicates that phase image resolution may be degraded with too large values of λ_1 and λ_2. We can also obtain this result from the numerical experiments using both simulated and measured data in the following section.

Table 1 Comparison of residue counts and MSE

Methods	Residue counts (whole wrapped image)	MSE/rad^2 (whole wrapped image)	Residue counts (small wrapped patch)	MSE/rad^2 (small unwrapped patch)
Original image	4498	0.36	56	0.3615
Mean	5810	0.32	0	0.0150
Goldstein (0.5)	684	0.12	0	0.0599
NL-InSAR	5952	0.49	0	0.0181
Lee	1914	0.16	0	0.0174
Total variation ($\lambda = 0.5$)	900	0.1154	0	0.0262
Total variation ($\lambda = 1.5$)	3978	0.35	0	0.0172
Total variation (λ, Eq. (19))	800	0.114	0	0.0182

4 Numerical experiments

One commonly used metric to evaluate the effectiveness of phase noise reduction methods is the residue count [5]. Usually, a low count of residues indicates a good phase noise reduction performance. However, considering the over-filtering cases, we need to use another two metrics to access the noise reduction performance. One metric is the mean-square error (MSE) which has two definitions, one each for the wrapped phase image and the unwrapped image. The MSE for the wrapped phase is defined as [27]

$$\mathrm{MSE} = \mathrm{E}\big[|\arg\exp(\mathrm{j}\hat{\varphi} - \mathrm{j}\varphi_{\mathrm{ideal}})|^2\big], \tag{17}$$

where E represents the statistics expectation, φ_{ideal} is the ideal InSAR phase and $\hat{\varphi}$ is the filtered phase.

The MSE for the unwrapped phase is defined as

$$\mathrm{RMSE} = \mathrm{E}\big[|\hat{\phi} - \phi_{\mathrm{ideal}}|^2\big], \tag{18}$$

where ϕ_{ideal} is the ideal unwrapped phase and $\hat{\phi}$ is the filtered unwrapped phase.

The smaller the MSE, the better the noise reduction. The other metric to access the noise reduction is the fringe continuity. The more continuous the fringes in the filtered phase, the better the noise reduction.

Scenario 1. Generate an unwrapped phase by using the 'peaks' function in MATLAB, xy_phase = $a \cdot \mathrm{peaks}(N)$, where $a = 24,\ N = 900$. Assume that the noise variance in the complex phase model (1) is 0.6 and that $\lambda_1 = \lambda_2 = \lambda$ in the optimization problem (3). We use our filtering method and the existing filtering methods to execute phase noise reduction. A slide window size of 5×5 is used for all the filtering methods except the Lee filter size of 9×9. In particular, we have tested the mean filter, Goldstein filter, NL-InSAR filter [8], Lee filter and our TV filter. For the latter, we have chosen $\lambda_1 = \lambda_2 = \lambda = 0.5,\ 1.5$ or according to

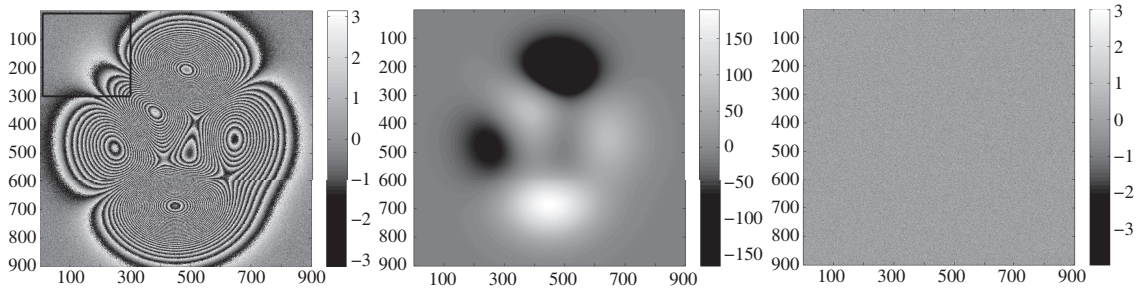
$$\lambda_1 = \lambda_2 = \lambda = \begin{cases} 0.85, & \bar{\gamma} < 0.3, \\ 0.80, & 0.3 < \bar{\gamma} < 0.5, \\ 0.55, & \text{others}, \end{cases} \tag{19}$$

where $\bar{\gamma}$ represents the mean local coherence. λ is a positive parameter which is used to balance the two terms in the optimization problem (3). The selection principle of λ is that λ should be proportional to the local variance of noise of each patch. This indicates that we can select λ inversely proportional to the mean local coherence of each patch [6]. In this experiment, however, we select three λ values according to three segments of the mean local coherence of the wrapped phase for simplicity.

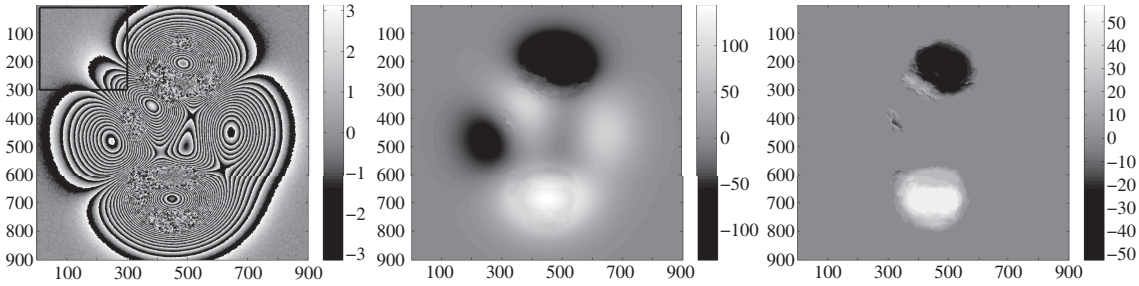
The wrapped phase images by different filtering methods, the unwrapped images by the weighted least squares unwrapped method *mainlpno.c* [28] and difference images are shown respectively below.

Comparison of residue counts and MSE is listed in Table 1.

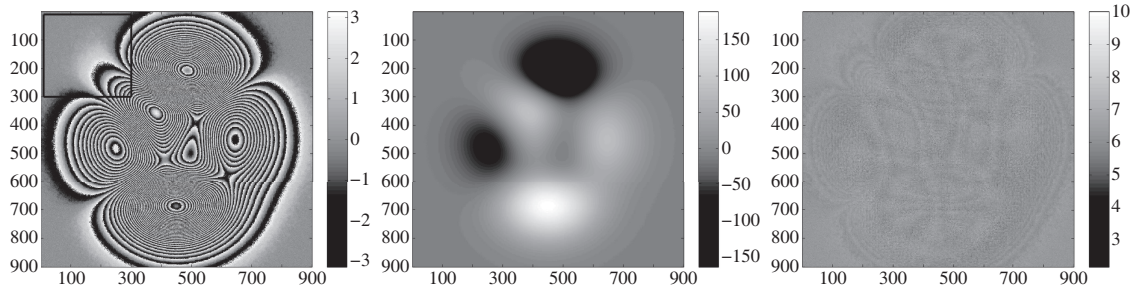
From the filtered images shown in Figure 1 and the residue counts and MSE listed in Table 1, the advantages of our filtering method is though the residue counts of the wrapped phase of the adaptive TV


100 200 300 400 500 600 700 800 900
100 300 500 700 900
3 2 1 0 −1 −2 −3
150 100 50 0 −50 −100 −150
3 2 1 0 −1 −2 −3

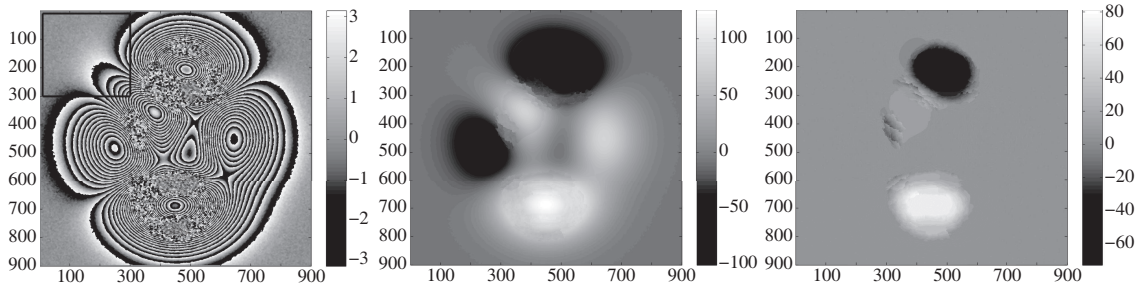
(a)


100 200 300 400 500 600 700 800 900
100 300 500 700 900
3 2 1 0 −1 −2 −3
100 50 0 −50 −100
50 40 30 20 10 0 −10 −20 −30 −40 −50

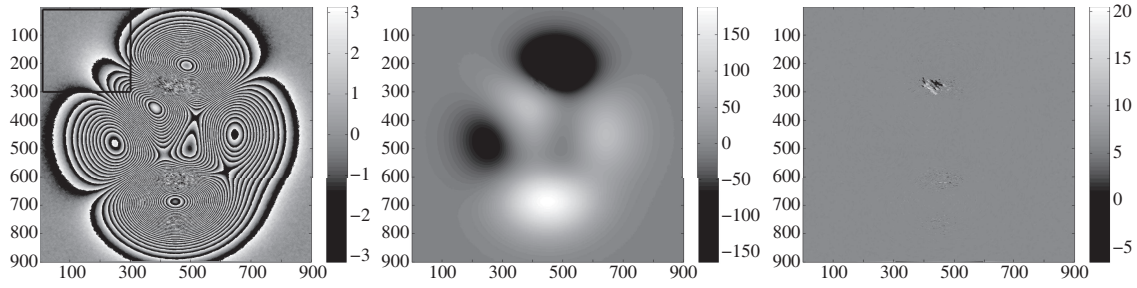
(b)


100 200 300 400 500 600 700 800 900
100 300 500 700 900
3 2 1 0 −1 −2 −3
150 100 50 0 −50 −100 −150
10 9 8 7 6 5 4 3

(c)


100 200 300 400 500 600 700 800 900
100 300 500 700 900
3 2 1 0 −1 −2 −3
100 50 0 −50 −100
80 60 40 20 0 −20 −40 −60

(d)


100 200 300 400 500 600 700 800 900
100 300 500 700 900
3 2 1 0 −1 −2 −3
150 100 50 0 −50 −100 −150
20 15 10 5 0 −5

(e)

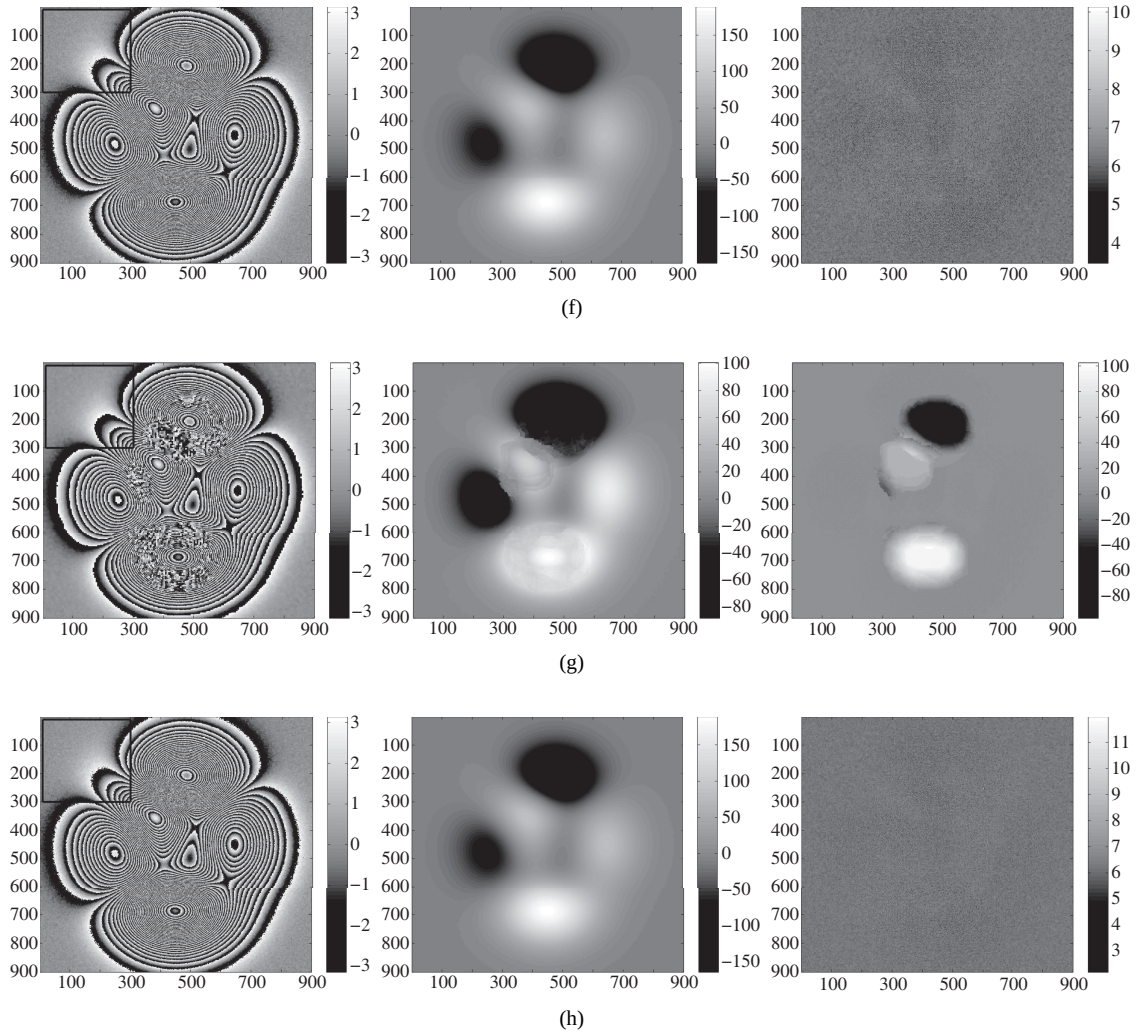


Figure 1 Filtered results of wrapped phases, unwrapping phases and difference phases based on simulated data. (a) Original wrapped phase, original unwrapped phase, original difference phase. (b) Wrapped phase, mean filter; unwrapped phase, mean filter; difference phase, mean filter. (c) Wrapped phase, Goldstein filter; unwrapped phase, Goldstein filter; difference phase, Goldstein filter. (d) Wrapped phase, NL-InSAR filter; unwrapped phase, NL-InSAR filter; difference phase, NL-InSAR filter. (e) Wrapped phase, Lee filter; wrapped phase, Lee filter; difference phase, Lee filter. (f) Wrapped phase, TV filter, $\lambda = 0.5$; unwrapped phase, TV filter, $\lambda = 0.5$; difference phase, TV filter, $\lambda = 0.5$. (g) Wrapped phase, TV filter, $\lambda = 0.5$; unwrapped phase, TV filter, $\lambda = 0.5$; difference phase, TV filter, $\lambda = 0.5$. (h) Wrapped phase, TV filter, λ satisfying (18); unwrapped phase, TV filter, λ satisfying (18); difference phase, TV filter, λ satisfying (18).

method is a little bit higher than those of Goldstein filter, the MSE is the least. That is, compared with the ideal phase image, fringe distortion of our method after filtering is the least. The existence of MSE of each unwrapped phase image shows the performance of the corresponding filtering algorithm. Due to the low MSE of the unwrapped phase image, our filtering method proves better in the subsequent generation of DEM and accurate detection of deformation. Furthermore, the less the variance of the residual phase, the higher the accuracy of the subsequent processing. For each wrapped phase image, we can extract the flatter patch from the whole image. Such patch can be approximately regarded as the phase of a flat scene for DEM or as the D-InSAR phase of no deformation scene. If we unwrap it and consider the residue counts and MSE in it, the whole processed results show the potential application of our proposed method though the MSE of our method is a litter bit higher than that of NL-InSAR filter or Lee filter.

Scenario 2. In this scenario, a measured noisy InSAR phase image whose size is 882×882 is used to compare the performance of mean filter, Goldstein filter, Lee filter, NL-InSAR and our TV filter. For the

Table 2 Comparison of residue counts of a wrapped phase image

Methods	Residue counts (whole wrapped image)	Residue counts (small wrapped patch)
Original image	58670	24779
Mean	552	152
Goldstein (0.5)	16170	7141
NL-InSAR	388	93
Lee	398	30
Total variation ($\lambda = 0.79$)	421	98
Total variation ($\lambda = 0.9$)	385	26
Total variation ($\lambda = 2.5$)	96	2
Total variation (λ, Eq. (20))	351	20

latter, we have chosen $\lambda = 0.79, 0.9, 2.5$ or simply according to

$$\lambda = \begin{cases} 1.1, & \bar{\gamma} < 0.4, \\ 0.8, & 0.4 < \bar{\gamma} < 0.7, \\ 0.75, & \text{others}, \end{cases} \tag{20}$$

where $\bar{\gamma}$ represents the mean local coherence [5].

In practice, the noise in (1) can be approximately equivalent to a complex zero mean white Gaussian noise with the known covariance depending on the coherence γ [29], i.e.

$$\mathrm{E}\{\boldsymbol{n}(k)\} = \mathbf{0}, \tag{21}$$

$$\mathrm{E}\{\boldsymbol{n}(k)\boldsymbol{n}(j)^H\} = \mathbf{I} \cdot \frac{1}{2}\left[\frac{1}{2|\gamma|^2} - \frac{1}{2}\right] \cdot \delta(k, j), \tag{22}$$

where

$$\delta(k, j) = \begin{cases} 1, & \text{if } k = j, \\ 0, & \text{else}, \end{cases} \tag{23}$$

is the discrete Dirac impulse.

This indicates that we should normalize the noise $\boldsymbol{n}(k)$ with its standard deviation before using the ADMM algorithm.

The wrapped phase images by different filtering methods, the unwrapped images by the code *mainlpno.c* and difference images are shown respectively as follows.

Comparison of residue counts of a wrapped phase image is listed in Table 2.

From the filtered interferograms shown in Figure 2, we can see that the TV filter is superior to other filtering methods since it not only filters higher amount of noise but also maintains the fringe completeness efficiently. In addition, the filtered results in Figure 2 show that interferogram can be over-filtered and give a mosaic effect when λ is too large such as $\lambda = 2.5$. The residue counts listed in Table 2 further illustrate the advantage of our filtering method. For each wrapped phase image, we can extract the flatter patch from the whole image. Such patch can also be approximately regarded as the phase of a flat scene for DEM or as the D-InSAR phase of no deformation scene as that in scenario 1. The processed result of it obviously shows the potential application of our proposed method.

Assume that the simulation is implemented using Matlab 7.10.0 (R2010a) on an HP PC with 2.50 GHz and 2.50 GHz Intel (R) Xeon (R) dual core CPU. CPU time comparison of the methods is listed in Table 3.

Table 3 shows the computational efficiency of the proposed ADMM implementation of the TV filter. The CPU time is significantly less than that of the NL-InSAR filter and that of the CVX implementation of the TV filter.

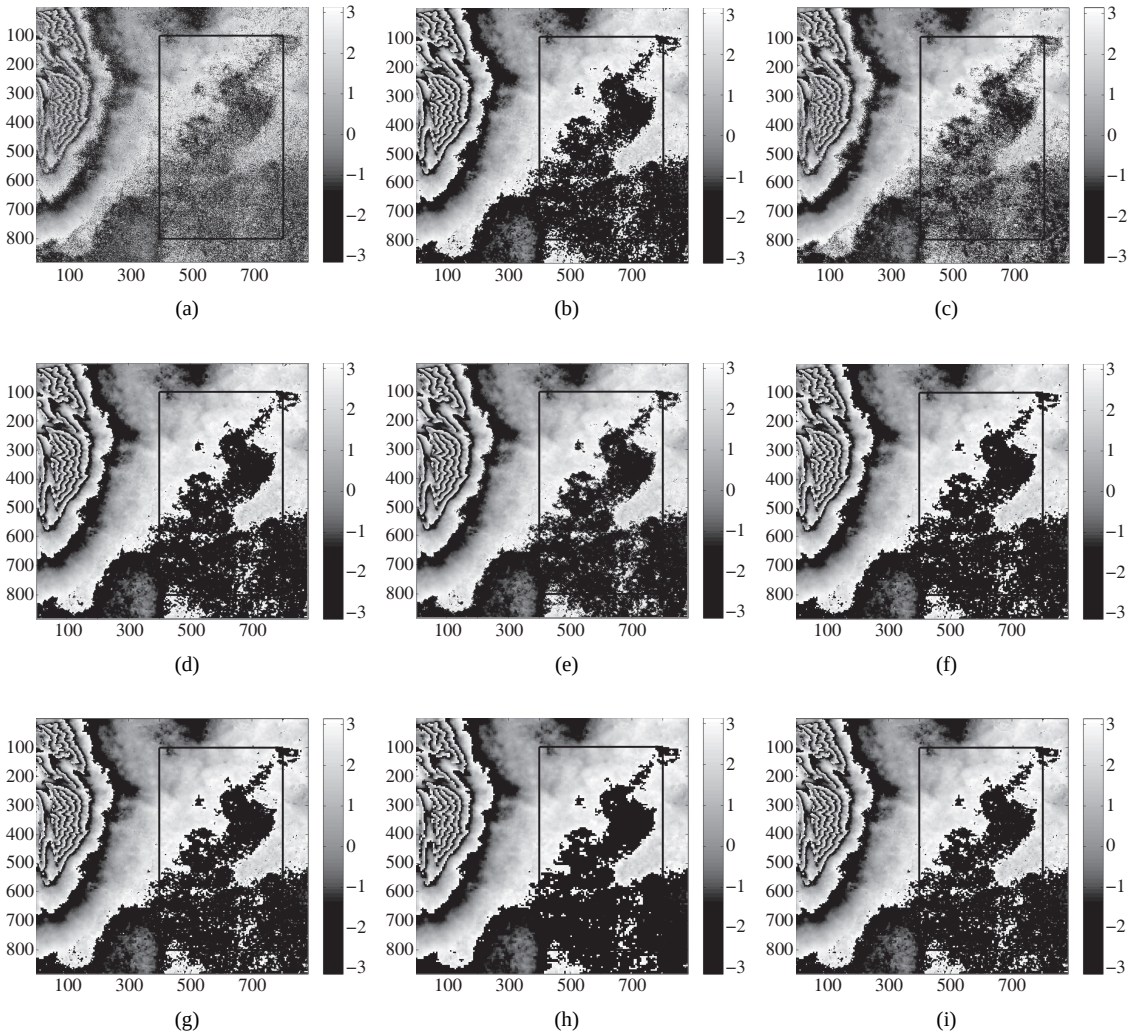


Figure 2 Filtered results based on measured data. (a) Original image; (b) mean filter; (c) Goldstein filter, $\lambda = 0.5$; (d) NL-InSAR filter; (e) Lee filter; (f) TV filter, $\lambda = 0.79$; (g) TV filter, $\lambda = 0.9$; (h) TV filter, $\lambda = 2.5$; (i) TV filter, adptive, λ satisfying (19).

Table 3 CPU time comparison

Type	Time (s)
Mean	0.78
Goldstein ($\lambda = 0.5$)	11.68
NL-InSAR	234.5042
Lee	75.36
TV filter, solving (3) with CVX	83290.36
TV filter, solving (7)–(11) with ADMM	10.98

5 Conclusion

In this paper, we present a novel phase filtering method. Based on total variation technique in sparse optimization, we have established a convex optimization problem from the original filtering problem of an InSAR phase image. Further we also design the ADMM algorithm to compute this optimization problem. Both the iteration complexity and the computational complexity of this algorithm are established in theorems for our InSAR phase noise reduction. Compared with other filtering methods, our filtering

method is much better in suppressing the noises and eliminating the residuals. Meanwhile, the TV method shows the best performance in fringe conservation and much less CPU time than that of the NL-InSAR filter and the CVX implementation of the TV filter. The analysis of the flatter patch with simulated and measured data shows potential application of our proposed method on a flat scene for DEM or a D-InSAR phase of no deformation scene.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant Nos. 41001282, 61462061, 61471276, 11401315, 61321064). The authors are grateful to Professor Luo ZhiQuan (Tom) of University of Minnesota for his helpful suggestions.

References

1 Eichel P H, Ghiglia D C, Jakowatz C V, et al. Spotlight SAR interferometry for terrain elevation mapping and interferometric change detection. Sandia National Laboratories SAND-93-2072. 1996
2 Chen R P, Yu W D, Wang R, et al. Interferometric phase denoising by pyramid nonlocal means filter. IEEE Geosci Remote Sens Lett, 2013, 10: 826–830
3 Lee J S. A new technique for noise filtering of SAR interferogram phase images. IEEE Trans Geosci Remot Sen, 1998, 36: 1456–1465
4 Fu S H, Long X J, Yang X, et al. Directionally adaptive filter for synthetic aperture radar interferometric phase image. IEEE Trans Geosci Remot Sen, 2013, 51: 552–559
5 Goldenstein R M, Radar C L. Radar interferogram filtering for geophysical applications. Geophys Reserv Lett, 1998, 25: 4035–4038
6 Baran L, Stewart M P, Kampes B M, et al. A modification to the goldstein radar interferogram filter. IEEE Trans Geosci Remot Sen, 2003, 41: 2114–2118
7 Deledalle C, Denis L, Tupin F. NL-InSAR: nonlocal interferogram estimation. IEEE Trans Geosci Remot Sen, 2011, 49: 1441–1452
8 Li J W, Li Z F, Bao Z, et al. Noise filtering of high-resolution interferograms over vegetation and urban areas with a refined nonlocal filter. IEEE Geosci Remote Sens Lett, 2015, 12: 77–81
9 Andriyan S B, Akira H. Interferometric SAR image restoration using Monte Carlo metropolis method. Sci China Inf Sci, 2010, 52: 1399–1408
10 Suo Z Y, Li Z F, Bao Z, et al. SAR-GMTI investigation in hybrid along and cross-track baseline InSAR. Sci China Ser-F: Inf Sci, 2009, 52: 1399–1408
11 Steven M K. Fundamentals of Statistical Signal Processing. Volume I: Estinimation Theory. Englewood Cliffs: Prentice-Hall, 1998
12 Yang J F, Zhang Y, Yin W T. A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data. IEEE J Sel Top Signal Process, 2010, 4: 288–297
13 Li M. A fast algorithm for color image enhancement with total variation regularization. Sci China Inf Sci, 2010, 53: 1913–1916
14 Grant M, Boyd S. CVX: Matlab Software for Disciplined Convex Programming. Version 1.21. 2010
15 López-Martínez C, Fábregas X. Modeling and reduction of SAR interferometric phase noise in the wavelet domain. IEEE Trans Geosci Remot Sen, 2002, 40: 2553–2566
16 Zeng C, Wang M H, Liao G S, et al. Sparse synthetic aperture radar imaging with optimized azimuthal aperture. Sci China Inf Sci, 2012, 55: 1852–1859
17 Xu G, Sheng J L, Zhang L, et al. Performance improvement in multi-ship imaging for ScanSAR based on sparse representation. Sci China Inf Sci, 2012, 55: 1860–1875
18 Yang J F, Zhang Y. Alternating direction algorithms for l_1-problems in compressive sensing. SIAM J Sci Comput, 2011, 33: 250–278
19 Gabay D, Mercier B. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Comput Math Appl, 1976, 2: 17–40
20 Glowinski R, Marrocco A. On the approximation and resolution of one class of problems of Dirichlet nonlinearity by finite elements and penalization-duality. Revue Fr Autom Inf Rech Oper, 1975, R-2: 41–76
21 Zhao G H, Shen F F, Wang Z Y, et al. A high quality image reconstruction method based on nonconvex decoding. Sci China Inf Sci, 2013, 56: 112103
22 Bertsekas D P, Bertsekas D P. Constrained Optimization and Lagrange Multiplier Method. New York: Athena Scientific, 1996
23 Yuan X M. Alternating direction method for covariance selection models. J Sci Comput, 2012, 51: 261–273

24 Cai X J, Gu G Y, He B S, et al. A proximal point algorithm revisit on the alternating direction method of multipliers. Sci China Math, 2013, 56: 2179–2186

25 Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn, 2011, 3: 1–122

26 He B S, Yuan X M. On the $O(1/n)$ convergence rate of the douglas-rachford alternating direction method. SIAM J Numer Anal, 2012, 50: 700–709

27 Cai B, Liang D, Dong Z. A new adaptive multiresolution noise-filtering approach for SAR interferometric phase images. IEEE Geosci Remote Sens Lett, 2008, 5: 266–270

28 Ghiglia D C, Pritt M D. Two-Dimensional Phase Unwrapping Theory, Algorithms, and Software. Wiley-Interscience, 1998

29 Loffeld O, Nies H, Knedlik S, et al. Phase unwrapping for SAR interferometry—a data fusion approach by Kalman filtering. IEEE Trans Geosci Remot Sen, 2008, 46: 47–58

Appendix A Derive (12)–(14)

Derive (12).

$$\boldsymbol{v}_1^{k+1} = \arg\min_{\boldsymbol{v}_1} \lambda_1\|\boldsymbol{v}_1\|_{1,2} + \frac{\beta_1}{2}\|\boldsymbol{v}_1 - \nabla\boldsymbol{x}_1^k + \frac{\boldsymbol{\mu}_1^k}{\beta_1}\|_2^2.$$

Solution:

$$\boldsymbol{v}_1^{k+1} = \text{shrink}_{\frac{\lambda_1}{\beta_1}}(\nabla\boldsymbol{x}_1^k - \frac{\boldsymbol{\mu}_1^k}{\beta_1}) = \left(\nabla\boldsymbol{x}_1^k - \frac{\boldsymbol{\mu}_1^k}{\beta_1}\right) - \min\left(\frac{\lambda_1}{\beta_1}, \left|\nabla\boldsymbol{x}_1^k - \frac{\boldsymbol{\mu}_1^k}{\beta_1}\right|\right) \cdot \frac{\nabla\boldsymbol{x}_1^k - \frac{\boldsymbol{\mu}_1^k}{\beta_1}}{|\nabla\boldsymbol{x}_1^k - \frac{\boldsymbol{\mu}_1^k}{\beta_1}|}, \tag{A1}$$

where the first equation can be derived[1)].

Derive (13).

$$\boldsymbol{v}_2^{k+1} = \arg\min_{\boldsymbol{v}_2} \lambda_2\|\boldsymbol{v}_2\|_{1,2} + \frac{\beta_2}{2}\|\boldsymbol{v}_2 - \nabla\boldsymbol{x}_2^k + \frac{\boldsymbol{\mu}_2^k}{\beta_2}\|_2^2.$$

Solution:

$$\boldsymbol{v}_2^{k+1} = \text{shrink}_{\frac{\lambda_2}{\beta_2}}(\nabla\boldsymbol{x}_2^k - \frac{\boldsymbol{\mu}_2^k}{\beta_2}) = \left(\nabla\boldsymbol{x}_2^k - \frac{\boldsymbol{\mu}_2^k}{\beta_2}\right) - \min\left(\frac{\lambda_2}{\beta_2}, \left|\nabla\boldsymbol{x}_2^k - \frac{\boldsymbol{\mu}_2^k}{\beta_2}\right|\right) \cdot \frac{\nabla\boldsymbol{x}_2^k - \frac{\boldsymbol{\mu}_2^k}{\beta_2}}{|\nabla\boldsymbol{x}_2^k - \frac{\boldsymbol{\mu}_2^k}{\beta_2}|}. \tag{A2}$$

Derive (14).

$$\boldsymbol{x}_c^{k+1} = \arg\min_{\boldsymbol{x}_c} \|\boldsymbol{y}_c - \boldsymbol{x}_c\|_2^2 + \beta_1\|\boldsymbol{v}_1^{k+1} - \nabla\boldsymbol{x}_1 + \frac{\boldsymbol{\mu}_1^k}{\beta_1}\|_2^2 + \beta_2\|\boldsymbol{v}_2^{k+1} - \nabla\boldsymbol{x}_2 + \frac{\boldsymbol{\mu}_2^k}{\beta_2}\|_2^2,$$

where $\nabla\boldsymbol{x}_1$ and $\nabla\boldsymbol{x}_2$ denote the real and imaginary parts of the gradient vector of $\boldsymbol{x}$, respectively, $\boldsymbol{A}_1 = [\boldsymbol{I}\ \boldsymbol{0}]$, $\boldsymbol{A}_2 = [\boldsymbol{0}\ \boldsymbol{I}]$ and $\boldsymbol{B}$ is a gradient generating matrix satisfying $\nabla\boldsymbol{x}_1 = \boldsymbol{B}\boldsymbol{A}_1\boldsymbol{x}_c$ and $\nabla\boldsymbol{x}_2 = \boldsymbol{B}\boldsymbol{A}_2\boldsymbol{x}_c$.

Solution:

Assume that $f(\boldsymbol{x}_c) = \|\boldsymbol{y}_c - \boldsymbol{x}_c\|_2^2 + \beta_1\|\boldsymbol{v}_1^{k+1} - \nabla\boldsymbol{x}_1 + \frac{\boldsymbol{\mu}_1^k}{\beta_1}\|_2^2 + \beta_2\|\boldsymbol{v}_2^{k+1} - \nabla\boldsymbol{x}_2 + \frac{\boldsymbol{\mu}_2^k}{\beta_2}\|_2^2$, the gradient vector of this function is as follows:

$$\boldsymbol{0} = -2(\boldsymbol{y}_c - \boldsymbol{x}_c) - 2\beta_1(\boldsymbol{B}\boldsymbol{A}_1)^{\mathrm{T}}\left(\boldsymbol{v}_1^{k+1} - \boldsymbol{B}\boldsymbol{A}_1\boldsymbol{x}_c + \frac{\boldsymbol{\mu}_1^k}{\beta_1}\right) - 2\beta_2(\boldsymbol{B}\boldsymbol{A}_2)^{\mathrm{T}}\left(\boldsymbol{v}_2^{k+1} - \boldsymbol{B}\boldsymbol{A}_2\boldsymbol{x}_c + \frac{\boldsymbol{\mu}_2^k}{\beta_2}\right). \tag{A3}$$

After arranging the previous equation, it is easy to get that

$$\begin{aligned}\boldsymbol{x}_c = &\left[\boldsymbol{I} + \beta_1(\boldsymbol{B}\boldsymbol{A}_1)^{\mathrm{T}}(\boldsymbol{B}\boldsymbol{A}_1) + \beta_2(\boldsymbol{B}\boldsymbol{A}_2)^{\mathrm{T}}(\boldsymbol{B}\boldsymbol{A}_2)\right]^{-1} \\ &\cdot\left[\boldsymbol{y}_c + \beta_1(\boldsymbol{B}\boldsymbol{A}_1)^{\mathrm{T}}\boldsymbol{v}_1^{k+1} + (\boldsymbol{B}\boldsymbol{A}_1)^{\mathrm{T}}\boldsymbol{\mu}_1^k + \beta_2(\boldsymbol{B}\boldsymbol{A}_2)^{\mathrm{T}}\boldsymbol{v}_2^{k+1} + (\boldsymbol{B}\boldsymbol{A}_2)^{\mathrm{T}}\boldsymbol{\mu}_2^k\right].\end{aligned} \tag{A4}$$

Appendix B Proof of the iteration complexity of ADMM algorithm

Let

$$\boldsymbol{v} = \begin{bmatrix}\boldsymbol{v}_1 \\ \boldsymbol{v}_2\end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix}\boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2\end{bmatrix}. \tag{B1}$$

The augmented Lagrangian function of (6) can be equivalently transformed into

$$\mathcal{L}(\boldsymbol{v}, \boldsymbol{x}_c, \boldsymbol{\mu}) = \lambda_1\|\boldsymbol{v}_1\|_{1,2} + \lambda_2\|\boldsymbol{v}_2\|_{1,2} + \frac{1}{2}\|\boldsymbol{y}_c - \boldsymbol{x}_c\|_2^2 + \boldsymbol{\mu}_1^{\mathrm{T}}(\boldsymbol{v}_1 - \nabla\boldsymbol{x}_1) + \boldsymbol{\mu}_2^{\mathrm{T}}(\boldsymbol{v}_2 - \nabla\boldsymbol{x}_2)$$

1) Han D R, Yuan X M, Zhang W X. An augmented-lagrangian-based parallel splitting method for separable convex minimization with applications to image processing. Math Comput, 2014, 83: 2263–2291.

$$+\frac{\beta_1}{2}\|\boldsymbol{v}_1-\nabla\boldsymbol{x}_1\|_2^2+\frac{\beta_2}{2}\|\boldsymbol{v}_2-\nabla\boldsymbol{x}_2\|_2^2. \tag{B2}$$

The optimality conditions of the proposed method are as follows

$$\begin{aligned}
0 \ &\in \ \lambda_1\partial\|\boldsymbol{v}_1^{k+1}\|_{1,2}+\beta_1\left(\boldsymbol{v}_1^{k+1}-\boldsymbol{BA}_1\boldsymbol{x}_1^k+\frac{\boldsymbol{\mu}_1^k}{\beta_1}\right),\\
0 \ &\in \ \lambda_2\partial\|\boldsymbol{v}_2^{k+1}\|_{1,2}+\beta_2\left(\boldsymbol{v}_2^{k+1}-\boldsymbol{BA}_2\boldsymbol{x}_2^k+\frac{\boldsymbol{\mu}_2^k}{\beta_2}\right),\\
0 \ &= \ \left(\boldsymbol{y}_1-\boldsymbol{x}_1^{k+1}\right)+\beta_1(\boldsymbol{BA}_1)^{\mathrm{T}}\left(\boldsymbol{v}_1^{k+1}-\boldsymbol{BA}_1\boldsymbol{x}_1^{k+1}+\frac{\boldsymbol{\mu}_1^k}{\beta_1}\right),\\
0 \ &= \ \left(\boldsymbol{y}_2-\boldsymbol{x}_2^{k+1}\right)+\beta_2(\boldsymbol{BA}_2)^{\mathrm{T}}\left(\boldsymbol{v}_2^{k+1}-\boldsymbol{BA}_2\boldsymbol{x}_2^{k+1}+\frac{\boldsymbol{\mu}_2^k}{\beta_2}\right),\\
0 \ &= \ \boldsymbol{v}_1^{k+1}-\boldsymbol{BA}_1\boldsymbol{x}_1^{k+1}+\frac{1}{\beta_1}\left(\boldsymbol{\mu}_1^k-\boldsymbol{\mu}_1^{k+1}\right),\\
0 \ &= \ \boldsymbol{v}_2^{k+1}-\boldsymbol{BA}_2\boldsymbol{x}_2^{k+1}+\frac{1}{\beta_2}\left(\boldsymbol{\mu}_2^k-\boldsymbol{\mu}_2^{k+1}\right).
\end{aligned} \tag{B3}$$

Define

$$\begin{cases}
\tilde{\boldsymbol{\mu}}_1^{k+1}=\boldsymbol{\mu}_1^k+\beta_1(\boldsymbol{v}_1^{k+1}-\boldsymbol{BA}_1\boldsymbol{x}_1^k),\\
\tilde{\boldsymbol{\mu}}_2^{k+1}=\boldsymbol{\mu}_2^k+\beta_2(\boldsymbol{v}_2^{k+1}-\boldsymbol{BA}_2\boldsymbol{x}_2^k).
\end{cases} \tag{B4}$$

The optimality conditions can be rewritten as follows

$$\begin{cases}
0\in\lambda_1\partial\|\boldsymbol{v}_1^{k+1}\|_{1,2}+\tilde{\boldsymbol{\mu}}_1^{k+1},\\
0\in\lambda_2\partial\|\boldsymbol{v}_2^{k+1}\|_{1,2}+\tilde{\boldsymbol{\mu}}_2^{k+1},\\
0\in\left(\boldsymbol{y}_1-\boldsymbol{x}_1^{k+1}\right)+(\mathbf{BA}_1)^{\mathrm{T}}\tilde{\boldsymbol{\mu}}_1^{k+1}+\beta_1(\boldsymbol{BA}_1)^{\mathrm{T}}\boldsymbol{BA}_1\left(\boldsymbol{x}_1^k-\boldsymbol{x}_1^{k+1}\right),\\
0\in\left(\boldsymbol{y}_2-\boldsymbol{x}_2^{k+1}\right)+(\boldsymbol{BA}_2)^{\mathrm{T}}\tilde{\boldsymbol{\mu}}_2^{k+1}+\beta_2(\boldsymbol{BA}_2)^{\mathrm{T}}\boldsymbol{BA}_2\left(\boldsymbol{x}_2^k-\boldsymbol{x}_2^{k+1}\right),\\
0\in\boldsymbol{v}_1^{k+1}-\boldsymbol{BA}_1\boldsymbol{x}_1^{k+1}+\frac{1}{\beta_1}\left(\boldsymbol{\mu}_1^k-\boldsymbol{\mu}_1^{k+1}\right),\\
0\in\boldsymbol{v}_2^{k+1}-\boldsymbol{BA}_2\boldsymbol{x}_2^{k+1}+\frac{1}{\beta_2}\left(\boldsymbol{\mu}_2^k-\boldsymbol{\mu}_2^{k+1}\right).
\end{cases} \tag{B5}$$

By the definition of $\mathcal{L}$ and the convexity property, we obtain

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{v}^{k+1},\boldsymbol{x}_c^{k+1},\boldsymbol{\mu})-\mathcal{L}_(\boldsymbol{v},\boldsymbol{x}_c,\tilde{\boldsymbol{\mu}}^{k+1})\\
=\ &(\lambda_1\|\boldsymbol{v}_1^{k+1}\|_{1,2}-\lambda_1\|\boldsymbol{v}_1\|_{1,2})+(\lambda_2\|\boldsymbol{v}_2^{k+1}\|_{1,2}-\lambda_2\|\boldsymbol{v}_2\|_{1,2})+\frac{1}{2}\|\boldsymbol{x}_c^{k+1}-\boldsymbol{y}_c\|_2^2-\frac{1}{2}\|\boldsymbol{x}_c-\boldsymbol{y}_c\|_2^2\\
&+\left[\boldsymbol{\mu}_1^{\mathrm{T}}\left(\boldsymbol{v}_1^{k+1}-\mathbf{BA}_1\boldsymbol{x}_1^{k+1}\right)-\left(\tilde{\boldsymbol{\mu}}_1^{k+1}\right)^{\mathrm{T}}\left(\boldsymbol{v}_1-\boldsymbol{BA}_1\boldsymbol{x}_1\right)\right]+\left[\boldsymbol{\mu}_2^{\mathrm{T}}\left(\boldsymbol{v}_2^{k+1}-\boldsymbol{BA}_2\boldsymbol{x}_2^{k+1}\right)-\left(\tilde{\boldsymbol{\mu}}_2^{k+1}\right)^{\mathrm{T}}\left(\boldsymbol{v}_2-\boldsymbol{BA}_2\boldsymbol{x}_2\right)\right]\\
\leqslant\ &\langle\lambda_1\partial\|\boldsymbol{v}_1^{k+1}\|_{1,2},\boldsymbol{v}_1^{k+1}-\boldsymbol{v}_1\rangle+\langle\lambda_2\partial\|\boldsymbol{v}_2^{k+1}\|_{1,2},\boldsymbol{v}_2^{k+1}-\boldsymbol{v}_2\rangle+\left[\langle\boldsymbol{x}_1^{k+1}-\boldsymbol{y}_1,\boldsymbol{x}_1^{k+1}-\boldsymbol{x}_1\rangle-\left(\boldsymbol{v}_1-\boldsymbol{v}_1^{k+1}\right)^{\mathrm{T}}\tilde{\boldsymbol{\mu}}_1^{k+1}\right.\\
&\left.+\left(\boldsymbol{x}_1-\boldsymbol{x}_1^{k+1}\right)^{\mathrm{T}}(\boldsymbol{BA}_1)^{\mathrm{T}}\tilde{\boldsymbol{\mu}}_1^{k+1}+(\boldsymbol{\mu}_1-\tilde{\boldsymbol{\mu}}_1^{k+1})^{\mathrm{T}}(\boldsymbol{v}_1^{k+1}-\boldsymbol{BA}_1\boldsymbol{x}_1^{k+1})\right]\\
&+\left[\langle\boldsymbol{x}_2^{k+1}-\boldsymbol{y}_2,\boldsymbol{x}_2^{k+1}-\boldsymbol{x}_2\rangle-\left(\boldsymbol{v}_2-\boldsymbol{v}_2^{k+1}\right)^{\mathrm{T}}\tilde{\boldsymbol{\mu}}_2^{k+1}\right.\\
&\left.+\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{k+1}\right)^{\mathrm{T}}(\boldsymbol{BA}_2)^{\mathrm{T}}\tilde{\boldsymbol{\mu}}_2^{k+1}+(\boldsymbol{\mu}_2-\tilde{\boldsymbol{\mu}}_2^{k+1})^{\mathrm{T}}(\boldsymbol{v}_2^{k+1}-\boldsymbol{BA}_2\boldsymbol{x}_2^{k+1})\right].
\end{aligned} \tag{B6}$$

Further substituting the optimality conditions (B5) into the above inequality yields

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{v}^{k+1},\boldsymbol{x}_c^{k+1},\boldsymbol{\mu})-\mathcal{L}(\boldsymbol{v},\boldsymbol{x}_c,\tilde{\boldsymbol{\mu}}^{k+1})\\
\leqslant\ &\beta_1\left(\boldsymbol{x}_1-\boldsymbol{x}_1^{k+1}\right)^{\mathrm{T}}(\boldsymbol{BA}_1)^{\mathrm{T}}\boldsymbol{BA}_1\left(\boldsymbol{x}_1^{k+1}-\boldsymbol{x}_1^k\right)+\frac{1}{\beta_1}\left(\boldsymbol{\mu}_1-\tilde{\boldsymbol{\mu}}_1^{k+1}\right)^{\mathrm{T}}\left(\boldsymbol{\mu}_1^{k+1}-\boldsymbol{\mu}_1^k\right)\\
&+\beta_2\left(\boldsymbol{x}_2-\boldsymbol{x}_2^{k+1}\right)^{\mathrm{T}}(\boldsymbol{BA}_2)^{\mathrm{T}}\boldsymbol{BA}_2\left(\boldsymbol{x}_2^{k+1}-\boldsymbol{x}_2^k\right)+\frac{1}{\beta_2}\left(\boldsymbol{\mu}_2-\tilde{\boldsymbol{\mu}}_2^{k+1}\right)^{\mathrm{T}}\left(\boldsymbol{\mu}_2^{k+1}-\boldsymbol{\mu}_2^k\right)\\
=\ &\frac{\beta_1}{2}(\|\boldsymbol{x}_1-\boldsymbol{x}_1^k\|^2_{(\boldsymbol{BA}_1)^{\mathrm{T}}\boldsymbol{BA}_1}-\|\boldsymbol{x}_1-\boldsymbol{x}_1^{k+1}\|^2_{(\boldsymbol{BA}_1)^{\mathrm{T}}\boldsymbol{BA}_1}-\|\boldsymbol{x}_1^k-\boldsymbol{x}_1^{k+1}\|^2_{(\boldsymbol{BA}_1)^{\mathrm{T}}\boldsymbol{BA}_1})\\
&+\frac{1}{2\beta_1}(\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_1^k\|_2^2-\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_1^{k+1}\|_2^2)+\left(\|\boldsymbol{\mu}_1^{k+1}-\tilde{\boldsymbol{\mu}}_1^{k+1}\|_2^2-\|\boldsymbol{\mu}_1^k-\tilde{\boldsymbol{\mu}}_1^{k+1}\|_2^2\right)\\
&+\frac{\beta_2}{2}(\|\boldsymbol{x}_2-\boldsymbol{x}_2^k\|^2_{(\boldsymbol{BA}_2)^{\mathrm{T}}\boldsymbol{BA}_2}-\|\boldsymbol{x}_2-\boldsymbol{x}_2^{k+1}\|^2_{(\boldsymbol{BA}_2)^{\mathrm{T}}\boldsymbol{BA}_2}-\|\boldsymbol{x}_2^k-\boldsymbol{x}_2^{k+1}\|^2_{(\boldsymbol{BA}_2)^{\mathrm{T}}\boldsymbol{BA}_2})\\
&+\frac{1}{2\beta_2}(\|\boldsymbol{\mu}_2-\boldsymbol{\mu}_2^k\|_2^2-\|\boldsymbol{\mu}_2-\boldsymbol{\mu}_2^{k+1}\|_2^2)+\left(\|\boldsymbol{\mu}_2^{k+1}-\tilde{\boldsymbol{\mu}}_2^{k+1}\|_2^2-\|\boldsymbol{\mu}_2^k-\tilde{\boldsymbol{\mu}}_2^{k+1}\|_2^2\right).
\end{aligned} \tag{B7}$$

It is easy to prove that $\|\boldsymbol{\mu}_i^{k+1}-\tilde{\boldsymbol{\mu}}_i^{k+1}\|_2^2-\|\boldsymbol{\mu}_i^{k}-\tilde{\boldsymbol{\mu}}_i^{k+1}\|_2^2\leqslant 0,\ i=1,2$, so that

$$\begin{aligned}&\mathcal{L}(\boldsymbol{v}^{k+1},\boldsymbol{x}_c^{k+1},\boldsymbol{\mu})-\mathcal{L}(\boldsymbol{v},\boldsymbol{x}_c,\tilde{\boldsymbol{\mu}}^{k+1})\\ \leqslant\ &\frac{\beta_1}{2}\left(\|\boldsymbol{x}_1-\boldsymbol{x}_1^k\|^2_{(\boldsymbol{BA}_1)^{\mathrm{T}}\boldsymbol{BA}_1}-\|\boldsymbol{x}_1-\boldsymbol{x}_1^{k+1}\|^2_{(\boldsymbol{BA}_1)^{\mathrm{T}}\boldsymbol{BA}_1}\right)+\frac{1}{2\beta_1}\left(\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_1^k\|_2^2-\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_1^{k+1}\|_2^2\right)\\ &+\frac{\beta_2}{2}\left(\|\boldsymbol{x}_2-\boldsymbol{x}_2^k\|^2_{(\boldsymbol{BA}_2)^{\mathrm{T}}\boldsymbol{BA}_2}-\|\boldsymbol{x}_2-\boldsymbol{x}_2^{k+1}\|^2_{(\boldsymbol{BA}_2)^{\mathrm{T}}\boldsymbol{BA}_2}\right)+\frac{1}{2\beta_2}\left(\|\boldsymbol{\mu}_2-\boldsymbol{\mu}_2^k\|_2^2-\|\boldsymbol{\mu}_2-\boldsymbol{\mu}_2^{k+1}\|_2^2\right).\end{aligned}\tag{B8}$$

Summing from $k=0$ to $k=K$ yields

$$\begin{aligned}\sum_{k=0}^{K}\mathcal{L}(\boldsymbol{v}^{k+1},\boldsymbol{x}_c^{k+1},\boldsymbol{\mu})-\ \sum_{k=0}^{K}\mathcal{L}(\boldsymbol{v},\boldsymbol{x}_c,\tilde{\boldsymbol{\mu}}^{k+1})\leqslant\ &\frac{\beta_1}{2}\|\boldsymbol{x}_1-\boldsymbol{x}_1^0\|^2_{(\boldsymbol{BA}_1)^{\mathrm{T}}\boldsymbol{BA}_1}+\frac{1}{2\beta_1}\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_1^0\|_2^2\\ &+\frac{\beta_2}{2}\|\boldsymbol{x}_2-\boldsymbol{x}_2^0\|^2_{(\boldsymbol{BA}_2)^{\mathrm{T}}\boldsymbol{BA}_2}+\frac{1}{2\beta_2}\|\boldsymbol{\mu}_2-\boldsymbol{\mu}_2^0\|_2^2.\end{aligned}\tag{B9}$$

Define $\hat{\boldsymbol{v}}^{K+1}=\frac{1}{K+1}\sum_{k=1}^{K+1}\boldsymbol{v}^k$, $\hat{\boldsymbol{x}}_c^{K+1}=\frac{1}{K+1}\sum_{k=1}^{K+1}\boldsymbol{x}_c^k$ and $\hat{\boldsymbol{\mu}}^{K+1}=\frac{1}{K+1}\sum_{k=1}^{K+1}\tilde{\boldsymbol{\mu}}^k$. Using the convexity of the objective function, we obtain the final result: for all $\boldsymbol{v},\boldsymbol{x}_c,\boldsymbol{\mu}$

$$\mathcal{L}(\hat{\boldsymbol{v}}^{K+1},\hat{\boldsymbol{x}}_c^{K+1},\boldsymbol{\mu})-\mathcal{L}(\boldsymbol{v},\boldsymbol{x}_c,\hat{\boldsymbol{\mu}}^{K+1})\leqslant\frac{C}{K+1},\tag{B10}$$

for some constant $C>0$, as desired.

Appendix C Proof of the computational complexity of ADMM algorithm

For the computational complexity of ADMM (7)–(11), considering the basic operations of arithmetic operations $+,-,\times,\div$ and sorting.

First, we give the computational complexity of $\nabla\boldsymbol{x}_1^k=\boldsymbol{BA}_1\boldsymbol{x}_c^k$. Assume the number of pixels in a filtering image to be N, the size of $\boldsymbol{x}_c^k$ is $2N\times 1$, the size of $\boldsymbol{A}_1$ is $N\times 2N$ and the size of $\boldsymbol{B}$ is $(2N^2-2N)\times N$. Therefore, the computation complexity of $\nabla\boldsymbol{x}_1^k$ per iteration is $O((2N^2-2N)\times 2N\times N+2N\times(2N^2-2N))$, that is, $O^*(4N^4)$, where O^* denotes that lower order terms are ignored.

Second, due to the size of $\boldsymbol{BA}_1$ $(2N^2-2N)\times 2N$, the multiplication computational time of $(\boldsymbol{BA}_1)^{\mathrm{T}}\boldsymbol{BA}_1$ per iteration is $O(2N\times 2N\times(2N^2-2N))$. The same is for $(\boldsymbol{BA}_2)^{\mathrm{T}}\boldsymbol{BA}_2$. Due to the size of $\boldsymbol{v}_1^{k+1}$ $(2N^2-2N)\times 1$, the multiplication computational time of $(\boldsymbol{BA}_1)^{\mathrm{T}}\boldsymbol{v}_1^{k+1}$ per iteration is $O(2N\times 1\times(2N^2-2N))$. The same is for $(\boldsymbol{BA}_2)^{\mathrm{T}}\boldsymbol{v}_2^{k+1}$. Due to the size of $\boldsymbol{\mu}_1^k$ $(2N^2-2N)\times 1$, the multiplication time of $(\boldsymbol{BA}_1)^{\mathrm{T}}\boldsymbol{\mu}_1^k$ per iteration is $O(2N\times 1\times(2N^2-2N))$. The same is for $(\boldsymbol{BA}_2)^{\mathrm{T}}\boldsymbol{\mu}_2^k$. Therefore, the computational complexity of $\boldsymbol{x}_c^{k+1}$ per iteration is $O^*(16N^4)$.

Third, due to the size of $\frac{\boldsymbol{\mu}_1^k}{\beta_1}$ $(2N^2-2N)\times 1$, the $\div$ time of $\frac{\boldsymbol{\mu}_1^k}{\beta_1}$ per iteration is $O(2N^2-2N)$. And the computational complexity of $\|\nabla\boldsymbol{x}_1^k-\frac{\boldsymbol{\mu}_1^k}{\beta}\|$ per iteration is $O(2N^2-2N+1)$. Assume that we apply bubble method to attain the minimum value, the swapping time of $\min(\frac{\lambda_1}{\beta 1},\|\nabla\boldsymbol{x}_1^k-\frac{\boldsymbol{\mu}_1^k}{\beta}\|)$ per iteration is $O((2N^2-2N)^2)$. And the computational complexity of $(\nabla\boldsymbol{x}_1^k-\frac{\boldsymbol{\mu}_1^k}{\beta})/\|\nabla\boldsymbol{x}_1^k-\frac{\boldsymbol{\mu}_1^k}{\beta}\|$ per iteration is $O(2N^2-2N)$. Therefore, the computational complexity of $\boldsymbol{v}_1^{k+1}$ per iteration is $O^*(4N^4)$. The same is for $\boldsymbol{v}_2^{k+1}$.

At last, due to the size of $\boldsymbol{v}_1^{k+1}$ $(2N^2-2N)\times 1$, the computational complexity of $\boldsymbol{\mu}_1^{k+1}$ is $O^*(2N^2)$. The same is for $\boldsymbol{\mu}_2^{k+1}$.

Therefore, the whole computational complexity of ADMM algorithm (7)–(11) per iteration is for $O^*(28N^4)$.