Fast solver for some computational imaging problems: A regularized weighted least-squares approach

Bo Zhang, Sherif Makram-Ebeid, Raphael Prevost, Guillaume Pizaine

Fast Solver for Some Computational Imaging Problems: A Regularized Weighted Least-Squares Approach

B. Zhang[a,*], S. Makram-Ebeid[a], R. Prevost[a], G. Pizaine[a]

[a] *Medisys, Philips Research, Suresnes France*

Abstract

In this paper we propose to solve a range of computational imaging problems under a unified perspective of a regularized weighted least-squares (RWLS) framework. These problems include data smoothing and completion, edge-preserving filtering, gradient-vector flow estimation, and image registration. Although originally very different, they are special cases of the RWLS model using different data weightings and regularization penalties. Numerically, we propose a preconditioned conjugate gradient scheme which is particularly efficient in solving RWLS problems. We provide a detailed analysis of the system conditioning justifying our choice of the preconditioner that improves the convergence. This numerical solver, which is simple, scalable and parallelizable, is found to outperform most of the existing schemes for these imaging problems in terms of convergence rate.

Keywords: Regularized weighted least-squares, Preconditioned conjugate gradient, Preconditioning, Condition number

1. Introduction

In this paper we propose to solve some classical imaging problems with a unified quadratic optimization perspective. These topics include data smoothing and completion, edge-preserving filtering, gradient-vector flow estimation, and image registration. We are particularly interested in high-performance numerical solvers for these problems, as they are widely used as

*Corresponding author

Email address: bo.wang-zhang@philips.com (B. Zhang)

building blocks of numerous applications in many domains such as computer vision and medical imaging [1].

Concretely, we look at the framework of the regularized weighted least-squares (RWLS):

$$\arg\min_{u:\mathbb{R}^D\to\mathbb{R}} J(u) := \int_{\mathbb{R}^D} w(\mathbf{x})(u(\mathbf{x}) - u_0(\mathbf{x}))^2\, d\mathbf{x} + \gamma^{2\alpha} \int_{\mathbb{R}^D} |\mathbf{L}_\alpha u(\mathbf{x})|^2\, d\mathbf{x} \quad (1)$$

Here, $u_0(\mathbf{x}) \in \mathbb{R}$ represents the observed measurement at point $\mathbf{x} \in \mathbb{R}^D$, and $u(\mathbf{x})$ the data to estimate. $w(\mathbf{x}) \geq 0$ are non-negative weights, which can be considered as confidence levels of the measurements. $\mathbf{L}_\alpha u : \mathbb{R}^D \mapsto \mathbb{R}^n$ represents some regularization operator $\mathbf{L}_\alpha$ applying a penalty on u. We will restrict $\mathbf{L}_\alpha$ to be a linear fractional differential operator of order $\alpha > 0$, such as the gradient and the Laplacian. Further, $\gamma > 0$ is a trade-off parameter between the weighted data-fidelity term and the regularity-penalty term.

In this work we will focus on the discrete RWLS problem, or the discrete counterpart of Eq. (1):

$$\arg\min_{\mathbf{u}\in\mathbb{R}^N} J(\mathbf{u}) := \|\mathbf{W}^{\frac{1}{2}}(\mathbf{u} - \mathbf{u}_0)\|^2 + \gamma^{2\alpha}\|\mathbf{L}_\alpha \mathbf{u}\|^2 \quad (2)$$

Here, $\mathbf{u}_0 \in \mathbb{R}^N$ is the vector of the measurements of length N. For multi-dimensional measurements, the data are assumed to be vectorized in the lexicographical order. $\mathbf{W} \in \mathbb{R}^{N\times N}$ stands for a diagonal weighting matrix with the weights on its diagonal $\mathbf{W}_{i,i} = w_i \geq 0$ for $i = 0, \ldots, N-1$. $\mathbf{L}_\alpha$ in the discrete setting will be a matrix representing the differential operator and we keep the same notation. It will be clear (see Section 4) that each of our aforementioned imaging problems fits Eq. (2) by choosing a particular set of weights $\mathbf{W}$ and a particular regularization operator $\mathbf{L}_\alpha$.

Our main contribution here is proposing an efficient preconditioned conjugate gradient (PCG) scheme which solves RWLS, and hence the above imaging problems. We provide a detailed analysis of the system conditioning justifying our choice of the preconditioner that improves the convergence. Surprisingly, this simple solver is found to outperform most of the state-of-the-art numerical schemes proposed for those problems. In particular, the convergence rate of PCG is spectacular, with a gain up to an order of magnitude observed in some of our experiments. Additionally, the PCG has the advantages of being easily implementable, scalable and parallelizable.

This paper is organized as follows. Section 2 describes in detail the RWLS framework, and the proposed PCG solver. Section 3 analyzes the choice of

the preconditioner by showing its potential in reducing the condition number of the problem and hence improving the convergence rate. Then, Section 4 presents the different imaging problems revisited and solved by the RWLS approach. We show the superior performance of our method compared to various existing schemes. We also discuss an extension of the RWLS model in Section 5. Our conclusions are drawn in Section 6. Finally, mathematical details are deferred to the appendices.

2. Regularized Weighted Least-Squares and a PCG Solver

2.1. Notations in the 1D case

Let us use an example in the 1-dimensional (1D) RWLS to introduce our notations and present our main results. The setting can be easily extended to multi-dimensional cases (Section 2.5).

Consider the following RWLS in the continuous setting where the regularization operator is the first derivative (i.e., $\mathbf{L}_\alpha = d/dx$ with $\alpha = 1$):

$$\arg\min_u J(\mathbf{u}) = \int_{\mathbb{R}} w(x)(u(x) - u_0(x))^2\, dx + \gamma^2 \int_{\mathbb{R}} u'(x)^2\, dx \tag{3}$$

This choice makes Eq. (3) a Dirichlet regularized regression problem. Its solution is the stationary point to the associated Euler-Lagrange equation:

$$w(x)u(x) - \gamma^2 u''(x) = w(x)u_0(x), \quad x \in \mathbb{R} \tag{4}$$

In the discrete version, the operator $\mathbf{L}_1$ will be represented by a first-order finite-difference matrix. For example, let $\mathbf{L}_1$ be the following circulant matrix (Eq. (5)), which corresponds to a filter $g_1 = [1, -1]/h_1$ with a periodic boundary condition.

$$\mathbf{L}_1 := \frac{1}{h_1} \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 1 \\ 1 & 0 & \cdots & 0 & -1 \end{bmatrix} \tag{5}$$

Here $h_1 > 0$ represents the finite-difference spacing.

To solve Eq. (2), one sets the gradient of $J(\mathbf{u})$ to zero, and obtain a linear system which is no more than the discrete counterpart of Eq. (4):

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \quad \text{where } \mathbf{A} := \mathbf{W} + \gamma^2 \mathbf{L}_1^* \mathbf{L}_1 \text{ and } \mathbf{b} := \mathbf{W}\mathbf{u}_0 \tag{6}$$

We used $\mathbf{L}_1^*$ to denote the conjugate transpose of $\mathbf{L}_1$. It follows that $(-\mathbf{L}_1^*\mathbf{L}_1)$ is a Hermitian matrix which represents a second-order differential filter [2] $g_2 = [1, -2, 1]/h_1^2$. In addition, $\mathbf{A}$ is Hermitian and semi-positive definite.

$\mathbf{L}_1$ is diagonalizable by the fast Fourier transform (FFT) matrix $\mathbf{F}$ and its k-th eigenvalue is $\lambda_k = (e^{-j\omega_k} - 1)/h_1$ with $\omega_k := 2\pi k/N$:

$$\mathbf{L}_1 = \mathbf{F}^*\Lambda\mathbf{F}, \quad \Lambda := \mathrm{diag}[\lambda_0, \lambda_1, \ldots, \lambda_{N-1}]$$

Due to the orthonormality of FFT, one has $\mathbf{F}^*\mathbf{F} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. Therefore $\mathbf{A}$ can be rewritten as:

$$\mathbf{A} = \mathbf{W} + \gamma^2\mathbf{F}^*\tilde{\Lambda}\mathbf{F}, \quad \tilde{\Lambda} := |\Lambda|^2 := \Lambda^*\Lambda$$

where $\tilde{\Lambda}$ is the diagonal matrix of the eigenvalues $\tilde{\lambda}_k$ of $\mathbf{L}_1^*\mathbf{L}_1$ which are given by $\tilde{\lambda}_k := |\lambda_k|^2 = [\frac{2}{h_1}\sin(\omega_k/2)]^2$.

The FFT choice above is clearly due to the assumed periodic boundary condition. More generally, the Hermitian matrix $\mathbf{L}_1^*\mathbf{L}_1$ always possesses an orthonormal eigen-decomposition:

$$\mathbf{L}_1^*\mathbf{L}_1 = \mathbf{B}^*|\Lambda|^2\mathbf{B}, \quad |\Lambda|^2 := \mathrm{diag}[|\lambda_0|^2, |\lambda_1|^2, \ldots, |\lambda_{N-1}|^2]$$

where $\mathbf{B}$ is some orthonormal matrix, and $(|\lambda_k|^2)_{k=0,\ldots,N-1}$ are the eigenvalues of $\mathbf{L}_1^*\mathbf{L}_1$ written in the modulus form to emphasize their non-negative nature. In practice, the basis $\mathbf{B}$ will represent trigonometric transforms, i.e. FFT, DCT (discrete cosine transform), and DST (discrete sine transform), with a periodic, an even symmetric, and an odd-symmetric boundary conditions [2] respectively imposed on the matrix $\mathbf{L}_1^*\mathbf{L}_1$.

Consequently, for any order $\alpha > 0$ one can define $\mathbf{L}_\alpha^*\mathbf{L}_\alpha$ to be a fractional differential operator such that:

$$\mathbf{L}_\alpha^*\mathbf{L}_\alpha := \mathbf{B}^*|\Lambda|^{2\alpha}\mathbf{B}, \quad |\Lambda|^{2\alpha} := \mathrm{diag}[|\lambda_0|^{2\alpha}, |\lambda_1|^{2\alpha}, \ldots, |\lambda_{N-1}|^{2\alpha}] \tag{7}$$

We will keep noting the spectrum of $\mathbf{L}_\alpha^*\mathbf{L}_\alpha$ by

$$\tilde{\Lambda} := |\Lambda|^{2\alpha}, \quad \tilde{\lambda}_k := |\lambda_k|^{2\alpha} \tag{8}$$

In the subsequent presentation, we will concentrate on the periodic boundary condition (i.e., $\mathbf{B} = \mathbf{F}$).

Similar to Eq. (6), for an arbitrary $\alpha > 0$, the minimizer of Eq. (2) is the solution to the following linear system:

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \quad \text{where } \mathbf{A} := \mathbf{W} + \gamma^{2\alpha}\mathbf{L}_\alpha^*\mathbf{L}_\alpha \text{ and } \mathbf{b} := \mathbf{W}\mathbf{u}_0 \tag{9}$$

where $\mathbf{A}$ is Hermitian and semi-positive definite, and can be written as:

$$\mathbf{A} = \mathbf{W} + \gamma^{2\alpha}\mathbf{F}^*\tilde{\Lambda}\mathbf{F} \tag{10}$$

The k-th eigenvalue of $\mathbf{L}_\alpha^*\mathbf{L}_\alpha$ is given by $\tilde{\lambda}_k = |\lambda_k|^{2\alpha} = [\frac{2}{h_1}\sin(\omega_k/2)]^{2\alpha}$. These definitions will be extended to multi-dimensional case in Section 2.5.

2.2. Case of constant weights: a linear filtering

We keep considering the 1D RWLS problem. If the weights are everywhere constant (say $\mathbf{W} = \bar{w}\mathbf{I}$ for some constant $\bar{w}$), $\mathbf{A}$ has an explicit inverse. The solution is given by a linear filtering:

$$\mathbf{u} = \mathbf{A}^{-1}\mathbf{b} = \mathbf{F}^*\bar{w}\left(\bar{w}\mathbf{I} + \gamma^{2\alpha}\tilde{\Lambda}\right)^{-1}\mathbf{F}\mathbf{u}_0 \tag{11}$$

In plain words, Eq. (11) signifies:

(i) take the Fourier transform of $\mathbf{u}_0$;
(ii) weight the spectrum by $S_k := \bar{w}/(\bar{w} + \gamma^{2\alpha}\tilde{\lambda}_k)$ in a pointwise manner;
(iii) take the inverse Fourier transform.

The weights S_k correspond to the spectrum of a low-pass filter: S_k attains its maximum at the zero frequency ($k = 0$) and starts to drop down as k increases. It attains the half of the maximum at the frequency k such that $\tilde{\lambda}_k = \bar{w}/\gamma^{2\alpha}$. Examples of the spectrum for different α are shown in Fig. 1.

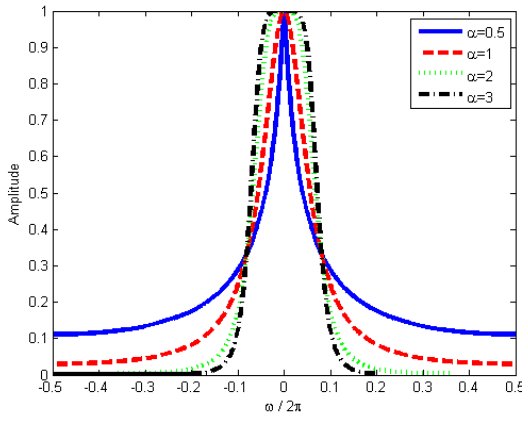


Figure 1: The spectrum S_k for $\alpha = 0.5, 1, 2, 3$. We set $h_1 = 1$, $\bar{w} = 0.5$, and $\gamma = 2$.

2.3. Case of non-constant weights: interpretation of a controlled diffusion

Regarding the case of non-constant weights, $\mathbf{A}$ no longer has an explicit inverse in general. However some asymptotic analysis sheds light on the expected behavior of the solution.

Suppose that the weights are nowhere zero, $\alpha = 1$ and a sufficiently small γ such that one can consider the first-order approximation of the solution:

$$\begin{aligned} \mathbf{u} &= \mathbf{A}^{-1}\mathbf{b} = (\mathbf{W} + \gamma^2 \mathbf{L}_1^* \mathbf{L}_1)^{-1} \mathbf{W}\mathbf{u}_0 \\ &= (\mathbf{I} + \gamma^2 \mathbf{W}^{-1} \mathbf{L}_1^* \mathbf{L}_1)^{-1} \mathbf{u}_0 \\ &\approx (\mathbf{I} - \gamma^2 \mathbf{W}^{-1} \mathbf{L}_1^* \mathbf{L}_1)\mathbf{u}_0 = \mathbf{u}_0 - \gamma^2 \mathbf{W}^{-1} \mathbf{L}_1^* \mathbf{L}_1 \mathbf{u}_0 \end{aligned} \tag{12}$$

It can be seen that Eq. (12) represents a step of diffusion on $\mathbf{u}_0$ where the step length is controlled by $\gamma^2 \mathbf{W}^{-1}$. Clearly, a data point associated with a large weight has a small step length and will undergo little change, while a point with a small weight (or large step) will tend to be blurred out by the diffusion process.

2.4. A PCG solver for the RWLS problem

Generally we resort to a PCG scheme for iteratively solving the linear system $\mathbf{A}\mathbf{u} = \mathbf{b}$. Let us point out that $\mathbf{A}$ is strictly positive definite if the null space of $\mathbf{W}$ and that of $\mathbf{L}_\alpha$ do not intersect. This is the case in Eq. (10) as long as the weights are not all zero.

Our preconditioner is formulated as

$$\mathbf{M} := \left(\nu \mathbf{I} + \gamma^{2\alpha} \mathbf{F}^* \tilde{\Lambda} \mathbf{F}\right)^{-1} = \mathbf{F}^* \mathbf{H} \mathbf{F} \tag{13}$$

where

$$\mathbf{H} := \left(\nu \mathbf{I} + \gamma^{2\alpha} \tilde{\Lambda}\right)^{-1} \tag{14}$$

The scalar parameter $\nu > 0$ is tunable. Note that Eq. (13) can be interpreted as a filter with the spectrum defined by Eq. (14). The PCG iteratively solves the system $\mathbf{M}\mathbf{A}\mathbf{u} = \mathbf{M}\mathbf{b}$ which is equivalent to $\mathbf{A}\mathbf{u} = \mathbf{b}$ but has a better convergence rate when $\mathbf{M}$ is properly chosen. The analysis and the choice of the preconditioner are detailed in Section 3.

In practice, PCG method does not need to explicitly store the huge matrices $\mathbf{A}$ and $\mathbf{M}$. Instead, given any vector $\mathbf{x}$ one only needs to implement the matrix-vector applications $\mathbf{A}\mathbf{x}$ and $\mathbf{M}\mathbf{x}$. These operations only involve the FFT, and pointwise additions and multiplications in the Fourier and in

the signal domains. Moreover, the eigenvalues $\tilde{\Lambda}$ can be pre-computed for a given regularization operator. As a consequence, the PCG solver runs very fast and is inherently scalable and parallelizable.

2.5. Multi-dimensional extension

There is no unique way of extending Eq. (2) to multi-dimensional situations. For any extension, we only need to care about the form of $\mathbf{L}_\alpha^*\mathbf{L}_\alpha$ in Eq. (9). In this paper, we choose $\mathbf{L}_\alpha^*\mathbf{L}_\alpha$ to be any discrete version of the (negative) fractional Laplacian of order α:

$$(-\Delta)^\alpha := \left(-\frac{\partial^2}{\partial x_1^2} - \frac{\partial^2}{\partial x_2^2} - \cdots - \frac{\partial^2}{\partial x_D^2}\right)^\alpha \tag{15}$$

where D is the dimension. Note that $(-\Delta)^\alpha$ should be understood in the Fourier sense. It is associated with the spectrum $|\boldsymbol{\omega}_s|^{2\alpha}$ where $\boldsymbol{\omega}_s$ is the spatial frequency in the continuous domain.

The rationale of this choice relies on its consistency with two important regularization kinds, i.e., the continuous RWLS energy (Eq. (1)) where the regularization term is either set to: (a) the Dirichlet energy, or to (b) the thin-plate spline bending energy with a periodic boundary condition. In both cases, the Laplacian operator shows up in the associated Euler-Lagrange equations.

For a D-dimension dataset of size $N_1 \times N_2 \times \cdots \times N_D$, the eigenvalues of $\mathbf{L}_\alpha^*\mathbf{L}_\alpha$ are written in the multi-index form:

$$\tilde{\lambda}_{\mathbf{k}} = \left(\sum_{d=1}^{D} |\lambda_{k_d}^{(d)}|^2\right)^\alpha, \quad \mathbf{k} \in \{(k_1, k_2, \cdots, k_D) : k_d = 0, 1, \ldots, N_d - 1\} \tag{16}$$

where $\lambda_{k_d}^{(d)}$ is the k_d-th eigenvalue of the matrix representing the 1D first-order derivative along the dimension d, i.e., $\partial/\partial x_d$. Under the periodic boundary condition, one has

$$\tilde{\lambda}_{\mathbf{k}} = \left(\sum_{d=1}^{D} \left[\frac{2}{h_d} \sin\left(\frac{\omega_{k_d}}{2}\right)\right]^2\right)^\alpha, \quad \omega_{k_d} := 2\pi k_d / N_d \tag{17}$$

3. Convergence Improvement Through Preconditioning

For a positive-definite linear system $\mathbf{A}\mathbf{u} = \mathbf{b}$, the condition number of the matrix $\mathbf{A}$ is defined as the ratio between its maximum eigenvalue $\lambda_{\max}(\mathbf{A})$

and its minimum one $\lambda_{\min}(\mathbf{A})$, i.e., $\kappa(\mathbf{A}) := \lambda_{\max}(\mathbf{A})/\lambda_{\min}(\mathbf{A})$. It is known that the convergence of a conjugate gradient method is faster if the condition number is smaller (i.e., closer to 1) [3], or in other words, if the spectrum of $\mathbf{A}$ is more compact. In this sense, $\kappa(\mathbf{A})$ can also be deemed as a measure of the spectral compactness. The maximal compactness is achieved by a scalar matrix $\mathbf{A} = c\mathbf{I}$ ($c > 0$) where one has $\kappa(\mathbf{A}) = 1$.

The purpose of a preconditioner is to improve the conditioning of a linear system. In our case the preconditioner $\mathbf{M}$ is written in the form of Eq. (13). In the subsequent sections, we make detailed analysis of the system conditioning before and after introducing the preconditioner $\mathbf{M}$ (Section 3.1), and provide suggestions of choosing the parameter ν (Section 3.2).

3.1. A spectral-compactness measure

Exact computation of the condition number turns out to be prohibitive for large data, as we have to evaluate the eigenvalues of a huge-size matrix. Alternatively, we will introduce another spectral compactness measure based on the coefficient of variation (CV) of the spectrum. Although different from the condition number, the CV is found to be directly related to a certain upper bound of the condition number. It is in this sense that the CV can be deemed as an estimator of the system conditioning. Moreover, this measure possesses an explicit expression that can be easily evaluated.

3.1.1. Notations and preliminary facts

Let us first establish some notations and preliminary results. Given an N by N matrix $\mathbf{G}$ with real eigenvalues, we define:

$$\mu(\mathbf{G}) := \frac{1}{N}\mathrm{tr}(\mathbf{G}) \tag{18}$$

$$\sigma^2(\mathbf{G}) := \mu(\mathbf{G}^2) - \mu^2(\mathbf{G}) \tag{19}$$

$$\tau(\mathbf{G}) := \frac{\sigma(\mathbf{G})}{\mu(\mathbf{G})} \tag{20}$$

where $\mathrm{tr}(\mathbf{G})$ denotes the trace of $\mathbf{G}$. It can be seen that Eq. (18), Eq. (19) and Eq. (20) represent the average, the variance and the CV of the spectrum of $\mathbf{G}$.

We recall the form of the preconditioner in Eq. (13). The diagonal matrix $\mathbf{H}$ (Eq. (14)) contains on its diagonal the eigenvalues of $\mathbf{M}$ written as

$$\eta_k = (\nu + \gamma^{2\alpha}\tilde{\lambda}_k)^{-1}, \quad k = 0, \ldots, N-1 \tag{21}$$

We use $\bar{w}$ to denote the average of the weights, and σ_w^2 for the sample variance of the weights:

$$\bar{w} := \frac{1}{N}\sum_{i=0}^{N-1} w_i \tag{22}$$

$$\sigma_w^2 := \frac{1}{N}\sum_{i=0}^{N-1}(w_i - \bar{w})^2 = \frac{1}{N}\sum_{i=0}^{N-1} w_i^2 - \bar{w}^2 \tag{23}$$

We also define

$$\hat{\mathbf{W}} := \mathbf{F}\mathbf{W}\mathbf{F}^* = [\mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_{N-1}] \tag{24}$$

where $\mathbf{q}_i$ is the i-th column vector. We use $q_{j,i}$ to denote the j-th element of $\mathbf{q}_i$. Note that since $\mathbf{W}$ is diagonal, $\hat{\mathbf{W}}$ has a periodic structure such that it can be written as a Kronecker product of circulant matrices.

Eigenvalues of $\mathbf{A}$ are real positive, and so do those of $\mathbf{MA}$ since $\mathbf{M}$ is a symmetric positive definite matrix. We note for any $m > 0$:

$$\beta_m(\mathbf{A}) := \begin{cases} \frac{\mu(\mathbf{A})+m\cdot\sigma(\mathbf{A})}{\mu(\mathbf{A})-m\cdot\sigma(\mathbf{A})} & \text{if } \mu(\mathbf{A}) > m\cdot\sigma(\mathbf{A}) \\ +\infty & \text{otherwise} \end{cases} \tag{25}$$

$\beta_m(\mathbf{A})$ will become an upper bound of $\kappa(\mathbf{A})$ as m increases. As a result, one way to compare the system conditioning before and after preconditioning is to assess $\beta_m(\mathbf{A})$ and $\beta_m(\mathbf{MA})$ for a certain m. We do not bother to find the optimal m that results in the tightest bounds as we have the following relation:

$$\forall \mathbf{A}_1, \mathbf{A}_2, \quad \tau(\mathbf{A}_1) \geq \tau(\mathbf{A}_2) \Longleftrightarrow \beta_m(\mathbf{A}_1) \geq \beta_m(\mathbf{A}_2) \text{ for any } m > 0 \tag{26}$$

Eq. (26) implies that comparing β_m's and comparing τ's are equivalent since Eq. (26) holds for any m. Comparing the CV's no longer involves this parameter and consequently, to tell that a preconditioner $\mathbf{M}$ is potentially effective, one only needs to verify the relation $\tau(\mathbf{MA}) < \tau(\mathbf{A})$.

3.1.2. CV's before and after preconditioning

Propositions 1 and 2 show the CV of $\mathbf{A}$ and that of $\mathbf{MA}$, respectively. Reader can find the proofs in the appendices.

Proposition 1 *The CV of the matrix* $\mathbf{A}$ *is given by* $\tau(\mathbf{A}) = \frac{\sigma(\mathbf{A})}{\mu(\mathbf{A})}$ *where*

$$\begin{aligned} \mu(\mathbf{A}) &= \bar{w} + \gamma^{2\alpha}\mu(\tilde{\Lambda}) \quad (27) \\ \sigma^2(\mathbf{A}) &= \sigma_w^2 + \gamma^{4\alpha}\sigma^2(\tilde{\Lambda}) \quad (28) \end{aligned}$$

Proposition 2 *The CV of the matrix* $\mathbf{MA}$ *is given by* $\tau(\mathbf{MA}) = \frac{\sigma(\mathbf{MA})}{\mu(\mathbf{MA})}$ *where*

$$\begin{aligned} \mu(\mathbf{MA}) &= 1 + (\bar{w} - \nu)\mu(\mathbf{H}) \quad (29) \\ \sigma^2(\mathbf{MA}) &= \frac{1}{N}\sum_{i=0}^{N-1}\eta_i \sum_{j=0}^{N-1}\eta_j |q_{j,i}|^2 + (\bar{w}-\nu)^2\sigma^2(\mathbf{H}) - \bar{w}^2\mu(\mathbf{H}^2) \quad (30) \end{aligned}$$

with $\mathbf{H}$, η_i *and* $q_{j,i}$ *defined in Eq. (14), Eq. (21) and Eq. (24) respectively.*

We also introduce an upper bound of $\sigma^2(\mathbf{MA})$ which is noted as $\tilde{\sigma}^2(\mathbf{MA})$ as precised in Proposition 3. The CV computed from this upper bound is written as $\tilde{\tau}(\mathbf{MA}) := \tilde{\sigma}(\mathbf{MA})/\mu(\mathbf{MA})$.

Proposition 3 *We have the relation* $\tilde{\sigma}^2(\mathbf{MA}) \geq \sigma^2(\mathbf{MA})$ *where*

$$\tilde{\sigma}^2(\mathbf{MA}) := \sigma_w^2\mu(\mathbf{H}^2) + (\bar{w}-\nu)^2\sigma^2(\mathbf{H}) \quad (31)$$

The purpose of introducing this approximation is to simplify Eq. (30) by avoiding computing the high-order harmonics $q_{j,i}$. Clearly, any preconditioner $\mathbf{M}$ verifying $\tilde{\tau}(\mathbf{MA}) < \tau(\mathbf{A})$ automatically satisfies $\tau(\mathbf{MA}) < \tau(\mathbf{A})$. One can see that only the first and the second statistical moments of the weights are involved in the quantities $\tau(\mathbf{A})$ and $\tilde{\tau}(\mathbf{MA})$.

In practice, we are more interested in the asymptotic behavior of the CV's for the case of large datasets. In Eq. (17), if we let $N_d \to +\infty$ and write

$$\tilde{\lambda}(\boldsymbol{\omega}) := \left(\sum_{d=1}^{D}\left[\frac{2}{h_d}\sin\left(\frac{\omega_d}{2}\right)\right]^2\right)^{\alpha}, \quad \boldsymbol{\omega} \in \mathcal{A} := \{(\omega_1, \omega_2, \cdots, \omega_D) : \omega_d \in [0, 2\pi)\} \quad (32)$$

Eq. (27) and Eq. (28) become respectively:

$$\begin{aligned} \mu(\mathbf{A}) &= \bar{w} + \frac{\gamma^{2\alpha}}{(2\pi)^D}\int_{\mathcal{A}}\tilde{\lambda}(\boldsymbol{\omega})\, d\boldsymbol{\omega} \quad (33) \\ \sigma^2(\mathbf{A}) &= \sigma_w^2 + \gamma^{4\alpha}\left[\frac{1}{(2\pi)^D}\int_{\mathcal{A}}\tilde{\lambda}(\boldsymbol{\omega})^2\, d\boldsymbol{\omega} - \left(\frac{1}{(2\pi)^D}\int_{\mathcal{A}}\tilde{\lambda}(\boldsymbol{\omega})\, d\boldsymbol{\omega}\right)^2\right] \quad (34) \end{aligned}$$

Likewise, Eq. (29) and Eq. (31) respectively tend to:

$$\mu(\mathbf{MA}) = 1 + \frac{\bar{w} - \nu}{(2\pi)^D} \int_{\mathcal{A}} (\nu + \gamma^{2\alpha} \tilde{\lambda}(\boldsymbol{\omega}))^{-1} \, d\boldsymbol{\omega} \tag{35}$$

$$\begin{aligned} \tilde{\sigma}^2(\mathbf{MA}) &= \frac{\sigma_w^2 + (\bar{w} - \nu)^2}{(2\pi)^D} \int_{\mathcal{A}} (\nu + \gamma^{2\alpha} \tilde{\lambda}(\boldsymbol{\omega}))^{-2} \, d\boldsymbol{\omega} \\ &- \frac{(\bar{w} - \nu)^2}{(2\pi)^{2D}} \left[\int_{\mathcal{A}} (\nu + \gamma^{2\alpha} \tilde{\lambda}(\boldsymbol{\omega}))^{-1} \, d\boldsymbol{\omega} \right]^2 \end{aligned} \tag{36}$$

3.2. Choosing the parameter ν

To understand the behavior of our preconditioners, we compute the ratio $R_{CV} := \tilde{\tau}(\mathbf{MA})/\tau(\mathbf{A})$ for the 2D case (derived from Eq. (33) - (36)) as a function of $\nu/\bar{w}$ for different weight moments and for different values of γ and α. Without loss of generality, we suppose the weights to be normalized within $[0, 1]$. The weight mean and variance are uniformly sampled in the area $A = \{(\bar{w}, \sigma_w^2) : 0 < \bar{w} \leq 1 \text{ and } 0 \leq \sigma_w^2 \leq \bar{w}(1 - \bar{w})\}$. Note that the variance upper bound comes from the Bhatia-Davis inequality [4]. α and γ are sampled within $[1, 2]$ and $[0.1, 5]$ respectively. Each combination of a weight mean, a weight variance, a value of γ and a value of α produces a curve of R_{CV} (as a function of $\nu/\bar{w}$). We have about 3×10^4 combinations in total.

Fig. 2 shows some of them for $\alpha = 1$ and $\gamma = 1$, curves for the other cases being similar. One can see that the highest efficiency of the preconditioners (i.e., the valleys of R_{CV}) is achieved for ν typically ranging from $\bar{w}$ up to several multiples of $\bar{w}$. Indeed, the optimal ν is found within $[\bar{w}, 5\bar{w}]$ for more than 94% among all cases. For a given RWLS problem, ultimately one could use some line search methods in this range to find the optimal ν which minimizes $\tilde{\tau}(\mathbf{MA})$. Practically, we choose $\nu = \bar{w}$ in all our experiments of Section 4. Out of all the combinations above, this value of ν makes $\tilde{\tau}(\mathbf{MA})$ strictly lower than $\tau(\mathbf{A})$ (i.e., $R_{CV} < 1$) for more than 90% cases. In other words, the preconditioner $\mathbf{M}$ is effective in most of the time. We point out that this choice also makes $\mathbf{M}^{-1}$ the best approximation to the matrix $\mathbf{A}$ in the least squares sense.

4. Applications

4.1. Data smoothing and completion

Data smoothing with missing values refers to filling unobserved pixels with some estimates computed from the observed pixels. This problem is

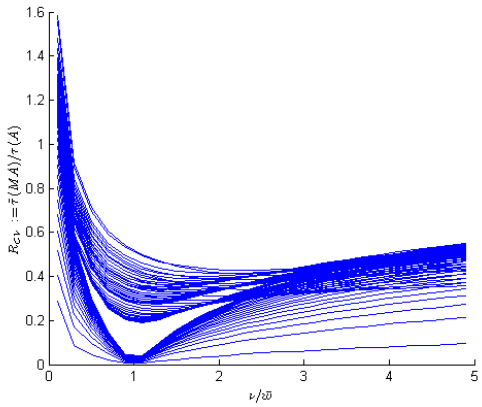


Figure 2: $R_{CV} := \tilde{\tau}(\mathbf{MA})/\tau(\mathbf{A})$ as a function of $\nu/\bar{w}$ which ranges from 0.1 to 5. R_{CV} quantifies the improvement of the system conditioning: the condition number is expected to decrease for $R_{CV} < 1$. We set $D = 2$, $h_d = 1$, $\alpha = 1$ and $\gamma = 1$. In the figure, 60 curves are plotted which correspond to the same number of parameters $(\bar{w}, \sigma_w^2)$ uniformly sampled in the area $A = \{(\bar{w}, \sigma_w^2) : 0 < \bar{w} \leq 1 \text{ and } 0 \leq \sigma_w^2 \leq \bar{w}(1 - \bar{w})\}$.

straightforwardly addressed by the RWLS (Eq. (2)) where the weights are typically binary by taking zero values for unobserved areas and values of 1 for the observed ones. This application is studied in detail by Garcia [5] where the issues of robustness and of choice of γ are emphasized. Numerically, the author proposed Keller's preconditioned gradient descent solver [6] for RWLS.

Fig. 3 compares the performances of Keller's approach and the PCG on a 2-dimensional (2D) image smoothing example. The original image (of size 256×256) is composed of a mixture of Gaussian functions. The observed image is generated by first adding a Gaussian noise, followed by dropping off about 2/3 data points. The missing data are first picked out randomly, then cropped out by 4 squares of size 50×50 each. We set $\alpha = 2$ and $\gamma = 1$. An even symmetry boundary condition is assumed and the DCT is employed to be the diagonalization basis of $\mathbf{L}_\alpha^* \mathbf{L}_\alpha$ as in [5].

Compared to the PCG result, the square masks are still partially visible in the estimate of Keller after 100 iterations. This shows that the PCG diffuses faster the observed information into the unfilled areas. The energy evolutions confirm that PCG converges much faster than Keller's iterations. The PCG is also more accurate: after 100 replications, the mean squared error (MSE) of the PCG estimates is evaluated to be 0.015, in comparison to 0.48 for Keller.

We also measured the execution time of both methods for indicative purpose. To achieve the same MSE target of 0.2 in this example, our approach took 0.9 seconds in contrast to Keller's method which took 11.5 seconds. The time was measured on a PC with 3.33 GHz Intel Xeon® CPU. Both methods are implemented in Matlab®.

Fig. 4 shows a more realistic case where we try to restore an image with 1/3 randomly sampled data available. Here the underlying image contains both low-frequency information and discontinuities. As we have shown in Section 2.3, one can view the weights in RWLS as actors of boundary conditions in a diffusion process. The boundary conditions are not strictly imposed but handled in a soft way through these weights. Eq. (12) shows that pixels receiving large fidelity weights prevent the diffusion induced by the regularity term at those pixels. In this sense, RWLS is very flexible as the weights automatically manage boundary conditions of any spatial shapes.

Here the observed pixels serve as our boundary condition which are associated with weights $w_i = 1$, and the unobserved pixels with zero weights. We set $\gamma^2 = 0.1$. Therefore, our RWLS will interpolate the unobserved areas while keeping observed values almost unchanged. Note that this procedure is quite similar to the interpolation based on partial differential equations [7].

4.2. Edge-preserving filtering

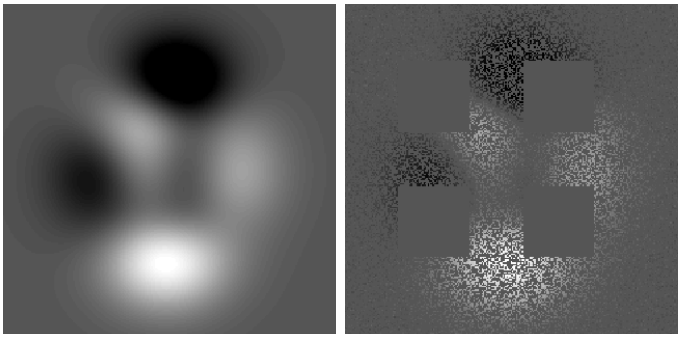
In the purpose of regularizing image while preserving discontinuities, one may apply the RWLS framework by using large fidelity weights on the pixels with discontinuities and small ones for homogeneous regions. According to Eq. (12), $\gamma^2\mathbf{W}^{-1}$ plays a similar role as the diffusion-rate function in the Perona-Malik anisotropic diffusion model [8].

Fig. 5 shows a filtering example using RWLS using $\gamma = 0.5$ and $\alpha = 1$. Our weights are defined as follows:

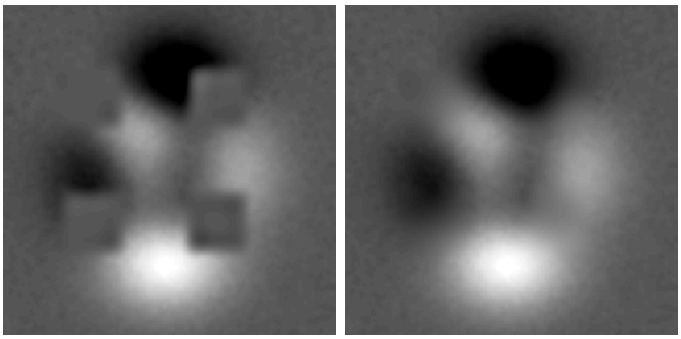
$$w_i := 1 - \exp(-|\nabla I(\mathbf{x}_i)|^2/K^2)$$

The parameter K can be considered as a reference edge-saliency level above which the diffusion will be hampered. Parallely, homogeneous areas encircled by the salient edges will be associated with small weights closed to zero. It follows that the remaining Dirichlet regularity term in Eq. (2) favors an approximation with harmonic functions on those areas[1]. Consequently, this

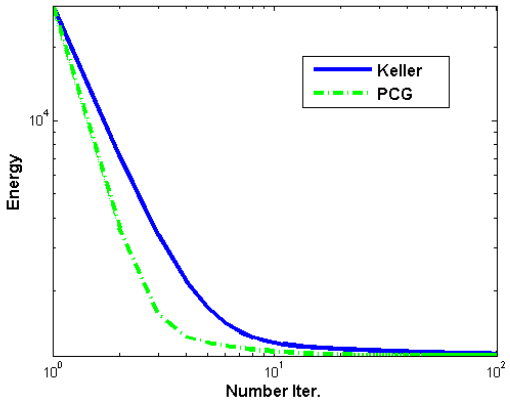
[1]This is due to the fact that Dirichlet penalty leads to Laplace equation as Euler-Lagrange equation with harmonic functions as solutions.



(a) (b)



(c) (d)



(e)

Figure 3: 2D image smoothing with missing values. (a) original image (size: 256×256, range: $[-6.5, 8.1]$); (b) observed image (by adding on the original image a zero-mean Gaussian noise of $\sigma = 0.25$, then removing 2/3 data through a random mask with 4 squares of 50×50 each); (c) result of Keller's method with 100 iterations; (d) result of PCG method with 100 iterations; (e) the evolutions of energy. Parameters: $h_d = 1$, $\alpha = 2$, the diagonalization basis is DCT, and $\gamma = 1$. MSE of the PCG estimates is 0.015 and that of Keller is 0.48. Targeting the same MSE of 0.2, PCG took 0.9 secs compared to 11.5 secs for Keller's approach on a 3.33 GHz Intel Xeon® CPU.

(a) (b) (c)

Figure 4: Image restoration with missing values. (a) original image of Lena (size: 512×512); (b) observed image (by removing 2/3 data through a random mask); (c) result of PCG method with 20 iterations; Parameters: $h_d = 1$, $\alpha = 1$, the diagonalization basis is DCT, and $\gamma^2 = 0.1$. The weights are set to 1 on observed pixels and 0 elsewhere.

results in piecewise smooth images.

4.3. Gradient-vector flow estimation

The gradient-vector flow (GVF) [9] is widely used in deformable-model based applications in order to extend the attraction range of external forces (see [1] and references therein). GVF in 2D continuous setting consists in solving the following variational problem:

$$\underset{\mathbf{u}:=[\mathbf{u}_1,\mathbf{u}_2]^T}{\arg\min} \int_\Omega g(|\nabla v(\mathbf{x})|)|\mathbf{u}(\mathbf{x}) - \nabla v(\mathbf{x})|^2 \, d\mathbf{x} + \gamma^2 \int_\Omega |\nabla \mathbf{u}_1(\mathbf{x})|^2 + |\nabla \mathbf{u}_2(\mathbf{x})|^2 \, d\mathbf{x} \tag{37}$$

Here Ω is the image domain, $\mathbf{x} = [x_1, x_2]^T \in \Omega$, $\nabla v(\mathbf{x})$ is the observed gradient field, and $\mathbf{u}(\mathbf{x}) := [\mathbf{u}_1(\mathbf{x}), \mathbf{u}_2(\mathbf{x})]^T$ is the sought extended field. g is an "edge" weighting function which is typically monotone increasing and smooth.

It can be seen that the components $\mathbf{u}_1$ and $\mathbf{u}_2$ are decoupled in the optimization problem (37). Solving (37) is equivalent to minimizing for $\mathbf{u}_1$ and $\mathbf{u}_2$ independently:

$$\underset{\mathbf{u}_l}{\arg\min} \int_\Omega g(|\nabla v(\mathbf{x})|) \left[\mathbf{u}_l(\mathbf{x}) - \frac{\partial v}{\partial x_l}(\mathbf{x})\right]^2 d\mathbf{x} + \gamma^2 \int_\Omega |\nabla \mathbf{u}_l(\mathbf{x})|^2 \, d\mathbf{x} \tag{38}$$

with $l = 1$ or 2. The Euler-Lagrange equation of Eq. (38) is given by $g(|\nabla v(\mathbf{x})|)(\mathbf{u}_l(\mathbf{x}) - \frac{\partial y}{\partial x_l}(\mathbf{x})) - \gamma^2 \Delta \mathbf{u}_l(\mathbf{x}) = 0$. Therefore, it is straightforward

Figure 5: Edge-preserving filtering by RWLS on the image of Lena (range: $[0, 255]$). (a) Original Lena image; (b) filtered result with $K = 10$; (c) filtered result with $K = 30$; (d) filtered result with $K = 50$. From (b) to (d), we set $\gamma = 0.5$, $\alpha = 1$, $h_d = 1$ and a maximum number of 20 PCG iterations. The diagonalization basis is DCT.

to convert Eq. (38) to our discrete RWLS framework of Eq. (2) by setting:

$$w_i := g(|\nabla v(\mathbf{x}[i])|), \quad \mathbf{u}_0[i] := \frac{\partial v}{\partial x_l}(\mathbf{x}[i]), \quad \mathbf{L}_\alpha^* \mathbf{L}_\alpha = -\Delta, \quad \alpha = 1$$

where $\mathbf{x}[i]$ is the position of the i-th pixel.

A vast number of schemes have been proposed to solve Eq. (37) such as the gradient-descent [9], operator-splitting based schemes [10], augmented Lagrangian method [11], and the alternating-direction methods [12]. Boukerroui [12] also provides a comparative study of the different numerical approaches.

In our comparative study, we include the alternating direction explicit scheme (ADES) solver which is recommended in [12], and the augmented Lagrangian (AL) based approach [11]. For ADES we followed the advice in [12] by including a left-right domain flipping in our iterations.

Our comparison is based on a synthetic image of a centered disk (Fig. 6(a)) as in [12]. The gradient of this image (i.e., ∇v), which is nonzero only in a local vicinity of the disk boundary, is used to construct the edge-weighting function:

$$g(t) := 1 - e^{-t^2}$$

We set $\gamma^2 = 1.5$ for all three methods. In ADES, the time step in the iteration scheme is set to 10 (see [12] for details).

As [12], our goal is to compare the orientation accuracy of the estimated vector fields. To build the ground truth, we compute analytically the orientation at each pixel of the image with respect to the disk center. These orientations are color-coded for $[-\pi, \pi)$ and shown in Fig. 6(c). The orientations of $\mathbf{u}(\mathbf{x})$ derived from the different GVF estimates are demonstrated in Fig. 6(d), (e) and (f), as well as their absolute residual in Fig. 6(g), (h) and (i) for the three approaches at the end of 15 iterations.

Visually, some bias for pixels far from the image center can be seen in ADES, and some inaccurate estimates in the AL method lie around the image corners. Generally, we found that more iterations are necessary for ADES to reduce the bias, and for AL to diffuse the local gradient information sufficiently far away to the image boundaries and corners. Finally, the PCG result looks the closest to the ground truth.

Quantitatively, the evolutions of the root mean squared error (RMSE) of the orientations of $\mathbf{u}(\mathbf{x})$ (measured in degrees) for the three methods are shown in Fig. 7. One can see that the PCG needs very few iterations to

reduce most of the error. This best performance is followed by ADES, and then AL. At the end of the iterations, we have RMSE = 0.71 degrees for PCG, 4.17 degrees for ADES, and 11.1 degrees for AL.

In terms of execution time, in order to attain an RMSE less than 5 degrees, it took 17.4 seconds for ADES, 1.04 seconds for AL, and 0.16 seconds for PCG. The time was evaluated on the same PC as in section 4.1.

4.4. Image registration

Registration between two 2D images consists in seeking a smooth deformation field that approximately maps an observed image to a reference. We are particularly interested in Demons registration algorithm [13], which is extensively employed in medical imaging [14][15] for its simplicity and fast performance.

The deformation field is derived as a solution of a variational problem such that the field is the best tradeoff between an image similarity measure and a field regularity measure. Here we consider the following minimization problem using the sum of squared difference (SSD) as our similarity term:

$$\arg\min_{\mathbf{u}} J(\mathbf{u}) = \sum_i (v(\mathbf{x}_i + \mathbf{u}_i) - v_0(\mathbf{x}_i))^2 + \gamma^{2\alpha} \|\mathbf{L}_\alpha \mathbf{u}\|^2 \tag{39}$$

where v and v_0 are the observed image and the reference image respectively, and $\mathbf{u}$ the sought deformation field.

We will cast the problem Eq. (39) into our RWLS framework using the approximation of [14]. For this purpose, we first adopt a first-order approximation of the SSD measure, or in other words the assumption of a small deformation $\mathbf{u}$.

$$\begin{aligned} \mathrm{SD}_i &:= (v(\mathbf{x}_i + \mathbf{u}_i) - v_0(\mathbf{x}_i))^2 \approx (\nabla v(\mathbf{x}_i)^T \mathbf{u}_i + v(\mathbf{x}_i) - v_0(\mathbf{x}_i))^2 \\ &= (v(\mathbf{x}_i) - v_0(\mathbf{x}_i))^2 + 2(v(\mathbf{x}_i) - v_0(\mathbf{x}_i)) \nabla v(\mathbf{x}_i)^T \mathbf{u}_i + \\ &\quad \mathbf{u}_i^T \nabla v(\mathbf{x}_i) \nabla v(\mathbf{x}_i)^T \mathbf{u}_i \end{aligned} \tag{40}$$

We then approximate the Hessian tensor $\nabla v(\mathbf{x}_i) \nabla v(\mathbf{x}_i)^T$ in Eq. (40) by the optimal scalar matrix in the least squares sense $\frac{1}{2}\mathrm{Tr}(\nabla v(\mathbf{x}_i) \nabla v(\mathbf{x}_i)^T)\mathbf{I}$. Completing the squares in Eq. (40), the problem Eq. (39) is converted into:

$$\arg\min_{\mathbf{u}} \sum_i |\nabla v(\mathbf{x}_i)|^2 \|\mathbf{u}_i - \hat{\mathbf{u}}_i\|^2 + 2\gamma^{2\alpha} \|\mathbf{L}_\alpha \mathbf{u}\|^2 \tag{41}$$

$$\text{with} \quad \hat{\mathbf{u}}_i = 2 \frac{v_0(\mathbf{x}_i) - v(\mathbf{x}_i)}{|\nabla v(\mathbf{x}_i)|^2} \nabla v(\mathbf{x}_i) \tag{42}$$

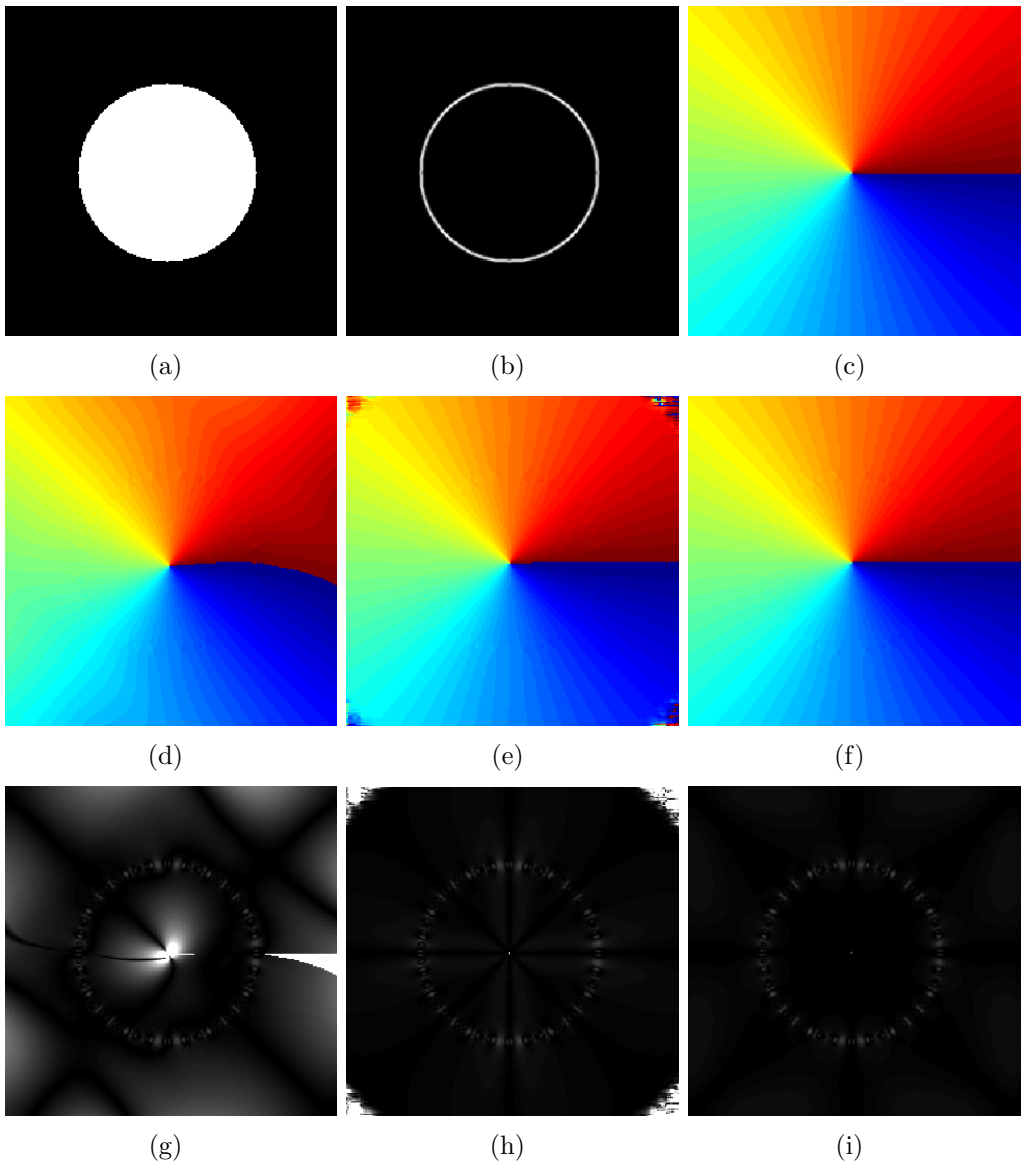


Figure 6: GVF estimation on a synthetic image. (a) The image of a centered disk (image size 257×257, disk radius is 64 pixels); (b) the gradient magnitude of the disk image; (c) the orientation analytically computed at each pixel w.r.t. the disk center (our ground-truth) and color-coded for $[-\pi, \pi)$; (d) the orientations of $\mathbf{u}(\mathbf{x})$ derived from the ADES scheme ($\tau = 10$); (e) the orientations of $\mathbf{u}(\mathbf{x})$ derived from the AL scheme; (f) the orientations of $\mathbf{u}(\mathbf{x})$ derived from the proposed PCG scheme; (g) the absolute residual of the angle estimates using ADES; (h) the absolute residual of the angle estimates using AL; (i) the absolute residual of the angle estimates using PCG. (g), (h) and (i) are shown in the scale of $[0, 20]$ degrees. The number of iterations is 15, and we set $h_d = 1$ and $\gamma^2 = 1.5$. The diagonalization basis is DCT.

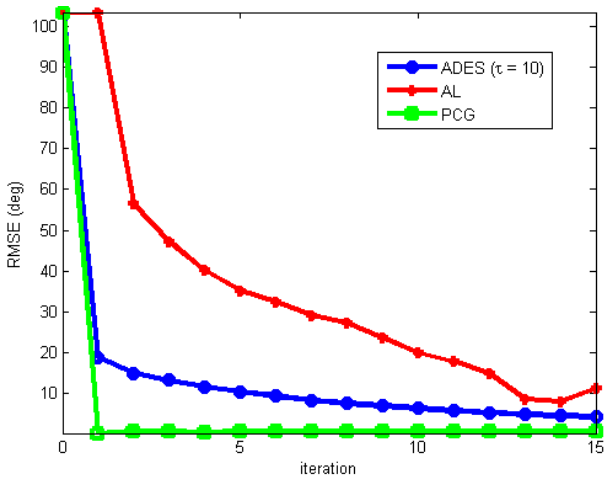


Figure 7: RMSE of the orientation estimates as a function of number of iterations. Targeting the same RMSE of 5 degrees, it took 17.4 secs for ADES, 1.04 secs for AL, and 0.16 secs for PCG on a 3.33 GHz Intel Xeon® CPU. The PCG method reduces the error much faster than the competing numerical schemes.

It can be seen that Eq. (41) fits our RWLS framework with the weights given by $w_i := |\nabla v(\mathbf{x}_i)|^2$. This dependency of the weights on the gradient is very intuitive: the deformation of the image is entirely encoded by the pixels having high gradients (i.e., discontinuities like contours and edges), while a small displacement of a pixel inside a homogeneous area is totally invisible.

Let us also point out that the first order approximation in (40) is valid for small displacement $\mathbf{u}$. In registrating two images with large deformation, we basically need to solve a sequence of problems (39) where in each step, v is replaced by the estimate from the previous warped result. Consequently, the final field is estimated as a successive composition of the small deformation fields.

In parallel, Demons registration consists first in computing the field Eq. (43) then followed by a Gaussian smoothing on both $\hat{\mathbf{u}}$ and the cumulated composite deformation field [13][15]. Compared to Eq. (42), Eq. (43) possesses an additional term in the denominator for preventing instabilities due to small gradient values. Note that this is not a problem for the RWLS model as small gradients are weighted by small values in Eq. (41).

$$\hat{\mathbf{u}}_i = \frac{v_0(\mathbf{x}_i) - v(\mathbf{x}_i)}{|\nabla v(\mathbf{x}_i)|^2 + \epsilon(v_0(\mathbf{x}_i) - v(\mathbf{x}_i))^2} \nabla v(\mathbf{x}_i) \tag{43}$$

Fig. 8 shows an example comparing Demons method and the RWLS approach on a synthetic case where we register a disk onto a peanut shape. The top three rows correspond to Demons warping progression using three increasing smoothing Gaussian widths ($\sigma = 1$, $\sigma = 3$, and $\sigma = 6$). The value of ϵ is set to 0.01. The last row demonstrates the warping results given by RWLS. The corresponding deformation field estimates are presented in Fig. 9. Fig. 9 clearly shows that Demon requires a large enough Gaussian width to guarantee the regularity of the warping (in our example $\sigma \geq 3$). However, as the Gaussian width increases the displacement magnitudes shrink implying that many more warpings are needed and hence can be time consuming.

Fig. 10 presents the evolution of the registration errors (defined as the norm of the difference between the warped image and the reference) as a function of the number of warpings. For a fixed error target, considerably fewer warpings are required in our PCG-based approach. The error reduction is slow in Demon's iterations and eventually almost stagnating.

5. Generalization of RWLS and the related works

In this section, we will briefly discuss a generalization of the RWLS framework and its potential in real applications. We will consider its extension by weighting both the data term and the regularization term. In addition, we will consider a general linear regularization operator $\mathbf{L}$:

$$\underset{u:\mathbb{R}^D \to \mathbb{R}}{\arg\min}\, J(u) := \int_{\mathbb{R}^D} w(\mathbf{x})(u(\mathbf{x}) - u_0(\mathbf{x}))^2\, d\mathbf{x} + \gamma \int_{\mathbb{R}^D} v(\mathbf{x})|\mathbf{L}u(\mathbf{x})|^2\, d\mathbf{x} \quad (44)$$

Here, $v(\mathbf{x}) \geq 0$ are the weights applied pointwisely on the regularization penalty. It follows that the generalized discrete RWLS can be written as:

$$\underset{\mathbf{u}\in\mathbb{R}^N}{\arg\min}\, J(\mathbf{u}) := \|\mathbf{W}^{\frac{1}{2}}(\mathbf{u} - \mathbf{u}_0)\|^2 + \gamma\|\mathbf{V}^{\frac{1}{2}}\mathbf{L}\mathbf{u}\|^2 \quad (45)$$

where the diagonal matrix $\mathbf{V} \in \mathbb{R}^{N\times N}$ bears the weights $v_i \geq 0$ on its diagonal elements. The minimizer is the solution to the following linear system:

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \quad \text{where } \mathbf{A} := (\mathbf{W} + \gamma\mathbf{L}^*\mathbf{V}\mathbf{L}) \text{ and } \mathbf{b} := \mathbf{W}\mathbf{u}_0 \quad (46)$$

Mimicking the standard RWLS case, we can solve the system by PCG with a preconditioner $\mathbf{M}$ given by

$$\mathbf{M} := (\bar{w}\mathbf{I} + \gamma\bar{v}\mathbf{L}^*\mathbf{L})^{-1} \quad (47)$$

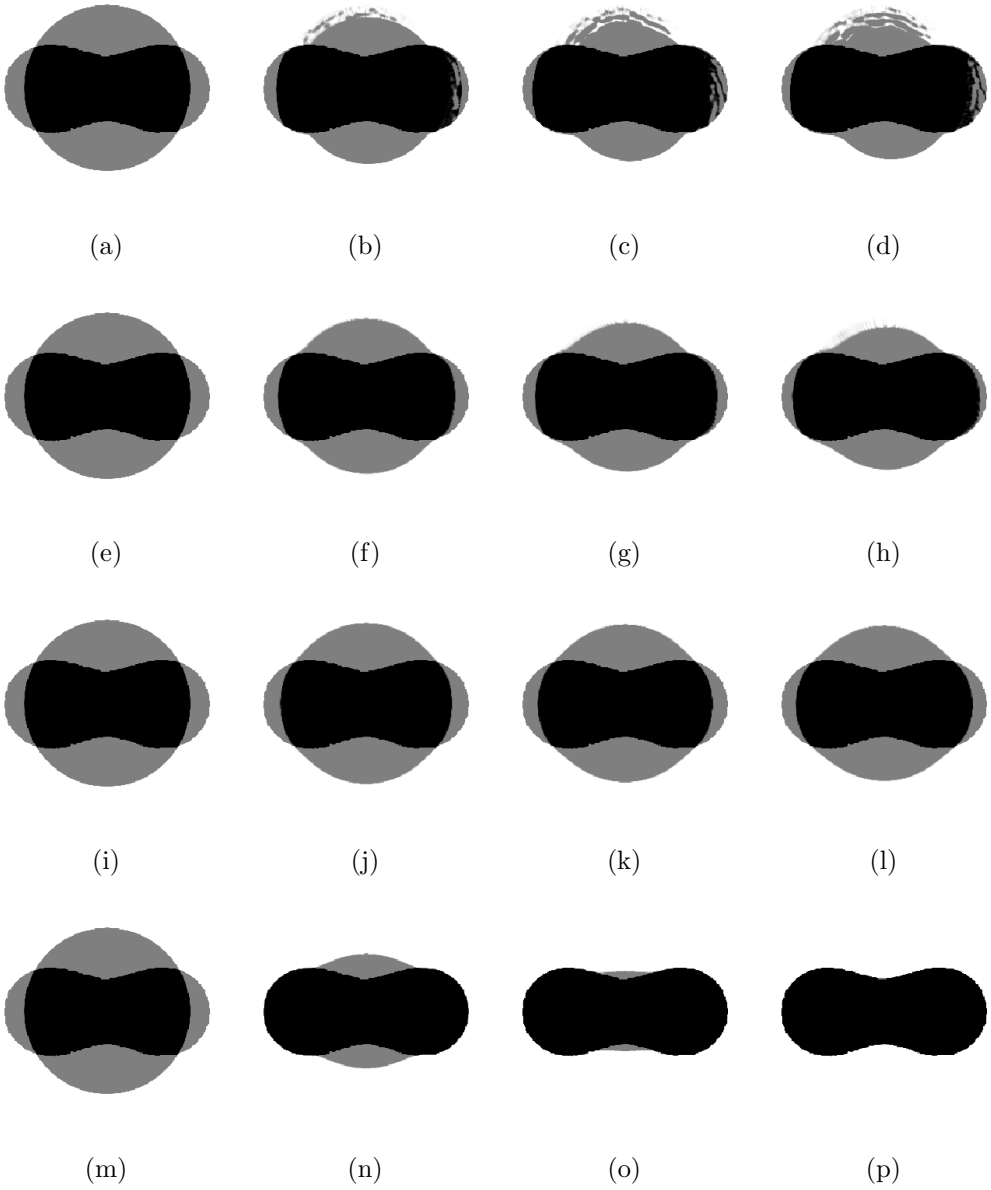


Figure 8: Registration by Demons algorithm and the proposed RWLS approach. The first image of every row (i.e., (a), (e), (i) and (m)) represents the initial state that a disk will be progressively warped to the shape of peanut. The two shapes are superimposed for display purpose. (b) to (d): Demons algorithm at 10, 20 and 30 warpings (smoothing Gaussian standard deviation $\sigma = 1$); (f) to (h): Demons algorithm at 10, 20 and 30 warpings (smoothing Gaussian standard deviation $\sigma = 3$); (j) to (l): Demons algorithm at 10, 20 and 30 warpings (smoothing Gaussian standard deviation $\sigma = 6$); (n) to (p): RWLS approach with PCG at 10, 20 and 30 warpings ($h_d = 1$, $\alpha = 1$ and $\gamma = 1$. The diagonalization basis is FFT).

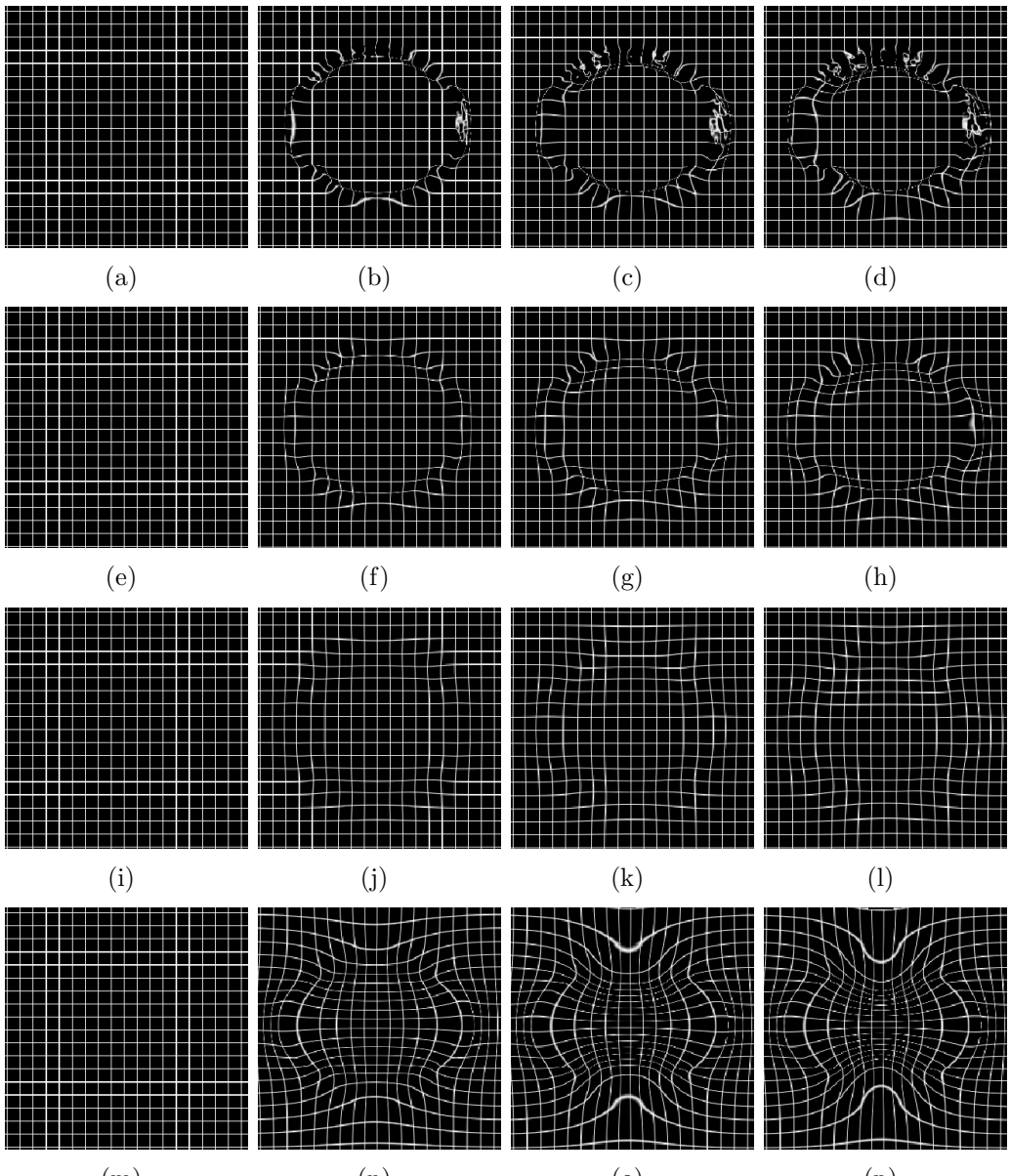
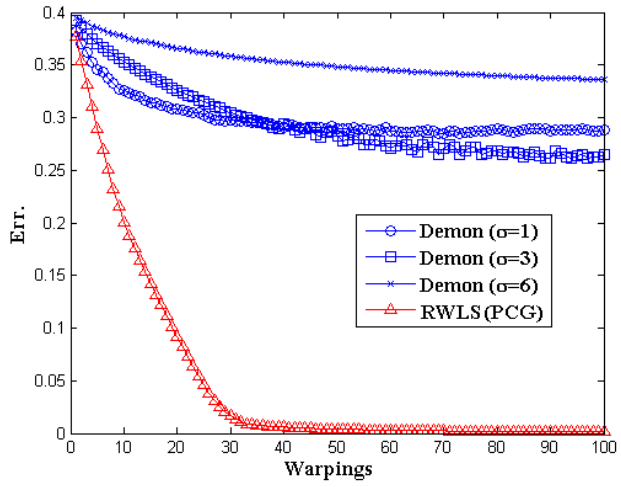


Figure 9: Warping fields of registration. Images are displayed in the same order as in Fig. 8.



(a)

Figure 10: Evolution of registration error as a function of the number of warpings. The error drops much faster in our PCG-based RWLS approach, compared to the Gaussian-filter-based Demons registrations.

where $\bar{v}$ represents the average value of the weights v_i.

The advantage of considering the extended RWLS framework is that, by properly designing the weights v, one may address other penalties such as L^1-minimization by this quadratic optimization framework. This point has been exploited by several works in the literature. For example, Daubechies et al. [16] proposed a sparse solution solver in compressed-sensing applications by casting an L^1-norm minimization into a sequence of weighted L^2-norm minimization problems. In [17], a half-quadratic algorithm [18][19] was applied to solve the total-variation (TV) minimization. The method converts the TV-regularization term $\int |\nabla u(\mathbf{x})|\, d\mathbf{x}$ into a weighted quadratic term through the weights $v(\mathbf{x})$:

$$\arg\min_{u,v>0} \text{Data-Term}(u) + \int \left[v(\mathbf{x}) |\nabla u(\mathbf{x})|^2 + \frac{1}{4v(\mathbf{x})} \right] d\mathbf{x}$$

At each iterative step, the algorithm first minimizes u by fixing v, and then solves the weights v while fixing u. Note that the solution of v given u has a simple form: $v(\mathbf{x}) = 1/(2|\nabla u(\mathbf{x})|)$. The updated u and v serve as initializations at the next iteration.

The generalized RWLS model allows us to deal with even richer appli-

cations under this flexible framework. Here we show its potential on a toy example of texture restoration application in Fig. 11. We simulated a periodic textural pattern by setting random amplitudes on 4 DCT coefficients and zeroing the rest. This image is shown by Fig. 11(a). Then in the spatial domain, we added a white noise (PSNR = 30) and randomly kept only 5.6% of the samples. The sampling mask is presented in Fig. 11(b). Our goal is to estimate the underlying pattern given the heavily down-sampled image.

This textural restoration problem can be formulated as:

$$\arg\min_{u} \int w(\mathbf{x})(u(\mathbf{x}) - u_0(\mathbf{x}))^2 \, d\mathbf{x} + \gamma \int |\mathbf{L}u(\boldsymbol{\omega})| \, d\boldsymbol{\omega} \tag{48}$$

where $\mathbf{L}$ represents the DCT transform, and u_0 the observed noisy and down-sampled image. The weights $w(\mathbf{x})$ are binary which take value of 1 only at the positions of the spatial sampling. The penalty of the L^1-norm on $\mathbf{L}u$ promotes a solution with sparse DCT coefficients. Using the same technique of half-quadratic algorithm, Eq. (48) is converted into Eq. (49) by introducing the weights v:

$$\arg\min_{u,v>0} \int w(\mathbf{x})(u(\mathbf{x}) - u_0(\mathbf{x}))^2 \, d\mathbf{x} + \gamma \int \left[v(\boldsymbol{\omega})\mathbf{L}u(\boldsymbol{\omega})^2 + \frac{1}{4v(\boldsymbol{\omega})} \right] d\boldsymbol{\omega} \tag{49}$$

While fixing the weights v, Eq. (49) is a generalized RWLS. The solution of u can be obtained by our PCG schema. While fixing u, the weights v can be explicitly computed. Consequently, we alternatively minimize u and v at each iteration and use the results as initial points for the next iteration. Fig. 11(d) shows the restored texture pattern despite the heavily degraded observations.

6. Conclusion

In this paper, we proposed to solve a range of computational imaging problems under the perspective of a regularized weighted least-squares model using a PCG approach. We provided a detailed convergence analysis justifying our choice of the preconditioner that improves the system conditioning. This numerical solver, which is simple, scalable and parallelizable, is found to outperform most of the competing schemes proposed in the literature in terms of the convergence rate. We also discussed an extended RWLS formulation by introducing weights on the regularization term, making the model

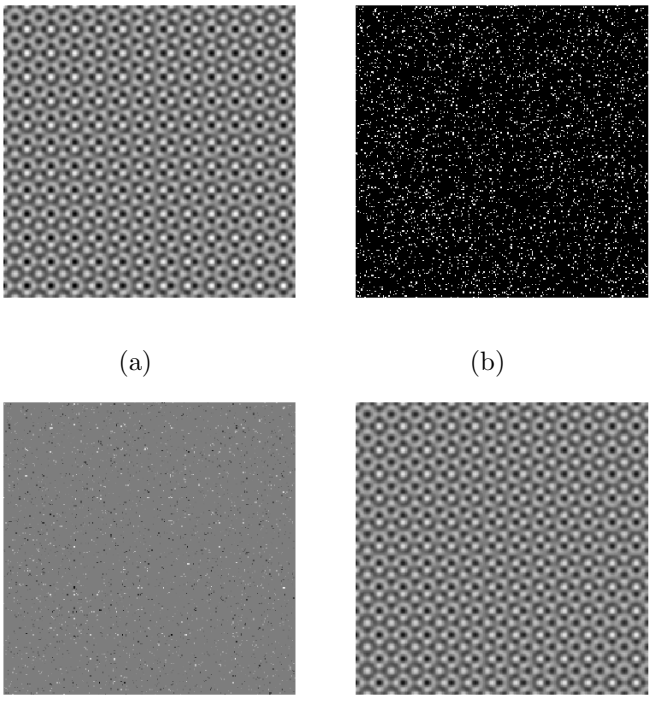


Figure 11: (a) Original image obtained from 4 DCT coefficients with random amplitudes (image size 256×256). (b) The spatial sampling mask which keeps only 5.6% spatial samples. (c) The down-sampled image is then further degraded by adding a white noise (PSNR = 30). (d) The restored image from the heavily down-sampled and noisy image (c). We have set a total of 15 iterations, and $\gamma = 0.05$. At each iteration, we used the PCG of the generalized RWLS to solve u while fixing v, and then explicitly compute v while fixing u by $v(\boldsymbol{\omega}) = 1/(2\sqrt{\mathbf{L}u(\boldsymbol{\omega})^2 + \epsilon})$ with $\epsilon = 10^{-4}$ for preventing zero-division.

even more flexible. We believe that a lot more imaging applications could join this framework and benefit from the efficient PCG scheme. This is the area of our future investigations.

Appendix A. Proof of Proposition 1

Applying the trace operator on $\mathbf{A}$ in Eq. (10), we immediately establish Eq. (27). To show Eq. (28), we compute $\mu(\mathbf{A}^2)$:

$$\mu(\mathbf{A}^2) = \mu(\mathbf{W}^2) + 2\gamma^{2\alpha}\mu(\mathbf{W}\mathbf{F}^*\tilde{\Lambda}\mathbf{F}) + \gamma^{4\alpha}\mu(\tilde{\Lambda}^2) \tag{A.1}$$

Eq.(24) implies that $\mu(\hat{\mathbf{W}}) = \mu(\mathbf{W}) = \bar{w}$. Due to the circulant structure in $\hat{\mathbf{W}}$, one can show that the diagonal elements of $\hat{\mathbf{W}}$ are all equal to $\bar{w}$. Therefore, we have

$$\mu(\mathbf{W}\mathbf{F}^*\tilde{\Lambda}\mathbf{F}) = \mu(\hat{\mathbf{W}}\tilde{\Lambda}) = \bar{w}\mu(\tilde{\Lambda}) \tag{A.2}$$

Inserting Eq. (A.2) into Eq. (A.1) and using the definition Eq. (19), we obtain Eq. (28). This ends the proof.

Appendix B. Proof of Proposition 2

Recall the definitions of $\mathbf{M}$ (i.e., Eq. (13)), of $\mathbf{H}$ (i.e., Eq. (14)) and of $\hat{\mathbf{W}}$ (i.e., Eq.(24)). We have

$$\begin{aligned} \mu(\mathbf{M}\mathbf{A}) &= \mu\left[\mathbf{H}(\hat{\mathbf{W}} + \gamma^{2\alpha}\tilde{\Lambda})\right] \\ &= \mu\left[\mathbf{H}(\mathbf{H}^{-1} + \hat{\mathbf{W}} - \nu\mathbf{I})\right] \\ &= 1 + (\bar{w} - \nu)\mu(\mathbf{H}) \end{aligned} \tag{B.1}$$

where we used the fact that $(\hat{\mathbf{W}} - \nu\mathbf{I})$ has all its diagonal elements equal to $(\bar{w} - \nu)$. This shows Eq. (29)

To show Eq. (30), we first show that

$$\mu((\mathbf{H}\hat{\mathbf{W}})^2) = \frac{1}{N}\sum_{i=0}^{N-1}\eta_i\sum_{j=0}^{N-1}\eta_j|q_{j,i}|^2 \tag{B.2}$$

we can write $\hat{\mathbf{W}}\mathbf{H} = [\eta_0\mathbf{q}_0, \eta_1\mathbf{q}_1, \ldots, \eta_{N-1}\mathbf{q}_{N-1}]$. Using the fact that $\hat{\mathbf{W}}^* = \hat{\mathbf{W}}$, we have

$$\hat{\mathbf{W}}\mathbf{H}\hat{\mathbf{W}} = \hat{\mathbf{W}}\mathbf{H}\hat{\mathbf{W}}^* = \sum_{i=0}^{N-1}\eta_i\mathbf{q}_i\mathbf{q}_i^*$$

Therefore, the j-th diagonal element of the matrix $\hat{\mathbf{W}}\mathbf{H}\hat{\mathbf{W}}$ is given by

$$(\hat{\mathbf{W}}\mathbf{H}\hat{\mathbf{W}})_{j,j} = \sum_{i=0}^{N-1} \eta_i q_{j,i} q_{j,i}^* = \sum_{i=0}^{N-1} \eta_i |q_{j,i}|^2$$

Consequently,

$$\mu((\mathbf{H}\hat{\mathbf{W}})^2) = \mu(\mathbf{H}(\hat{\mathbf{W}}\mathbf{H}\hat{\mathbf{W}})) = \frac{1}{N}\sum_{j=0}^{N-1}\eta_j \sum_{i=0}^{N-1}\eta_i |q_{j,i}|^2$$

This shows Eq. (B.2).

Now, $\mu((\mathbf{MA})^2)$ can be computed as

$$\begin{aligned} & \mu((\mathbf{MA})^2) = \mu((\mathbf{H}(\mathbf{H}^{-1} + \hat{\mathbf{W}} - \nu\mathbf{I}))^2) \\ = \;& \mu(\mathbf{I} + 2\mathbf{H}\hat{\mathbf{W}} - 2\nu\mathbf{H} + \nu^2\mathbf{H}^2) - 2\nu\mu(\mathbf{H}^2\hat{\mathbf{W}}) + \mu((\mathbf{H}\hat{\mathbf{W}})^2) \\ = \;& 1 + 2(\bar{w} - \nu)\mu(\mathbf{H}) + (\bar{w} - \nu)^2\mu(\mathbf{H}^2) - \bar{w}^2\mu(\mathbf{H}^2) + \mu((\mathbf{H}\hat{\mathbf{W}})^2) \end{aligned} \quad \text{(B.3)}$$

Inserting Eq. (B.2) into Eq. (B.3), we obtain Eq. (30) by the definition Eq. (19). This ends the proof.

Appendix C. Proof of Proposition 3

First, we point out that for any matrix $\mathbf{G}$ with real eigenvalues, we have $\mathrm{tr}(\mathbf{G}^2) \leq \mathrm{tr}(\mathbf{G}^*\mathbf{G})$. In effect, if we use $g_{i,j}$ to denote the element of $\mathbf{G}$ at the i-th row and the j-th column, one has

$$\begin{aligned} \mathrm{tr}(\mathbf{G}^2) &\leq \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |g_{i,j}g_{j,i}| = \sum_{i=0}^{N-1} |g_{i,i}|^2 + 2\sum_{i=0}^{N-1}\sum_{j<i}^{N-1} |g_{i,j}g_{j,i}| \\ &\leq \sum_{i=0}^{N-1} |g_{i,i}|^2 + \sum_{i=0}^{N-1}\sum_{j<i}^{N-1}(|g_{i,j}|^2 + |g_{j,i}|^2) = \sum_{i=0}^{N-1} |g_{i,i}|^2 + \sum_{i=0}^{N-1}\sum_{j\neq i}^{N-1} |g_{i,j}|^2 \\ &= \mathrm{tr}(\mathbf{G}^*\mathbf{G}) \end{aligned}$$

Using this result and replacing $\mathbf{G}$ by $\mathbf{MA}$, we have

$$\sigma^2(\mathbf{MA}) = \mu((\mathbf{MA})^2) - \mu(\mathbf{MA})^2 \leq \mu(\mathbf{A}^*\mathbf{M}^*\mathbf{MA}) - \mu(\mathbf{MA})^2 \qquad \text{(C.1)}$$

The term $\mu(\mathbf{A}^*\mathbf{M}^*\mathbf{M}\mathbf{A})$ is computed as follows:

$$\begin{aligned}
\mu(\mathbf{A}^*\mathbf{M}^*\mathbf{M}\mathbf{A}) &= \mu((\mathbf{H}^{-1} - \nu\mathbf{I} + \hat{\mathbf{W}})\mathbf{H}^2(\mathbf{H}^{-1} - \nu\mathbf{I} + \hat{\mathbf{W}})) \\
&= 1 + 2(\bar{w} - \nu)\mu(\mathbf{H}) + (\nu^2 - 2\nu\bar{w})\mu(\mathbf{H}^2) + \mu(\mathbf{H}^2\hat{\mathbf{W}}^2) \\
&= 1 + 2(\bar{w} - \nu)\mu(\mathbf{H}) + (\bar{w} - \nu)^2\mu(\mathbf{H}^2) + (\mu(\mathbf{W}^2) - \bar{w}^2)\mu(\mathbf{H}^2) \\
&= 1 + 2(\bar{w} - \nu)\mu(\mathbf{H}) + (\bar{w} - \nu)^2\mu(\mathbf{H}^2) + \sigma_w^2\mu(\mathbf{H}^2) \qquad \text{(C.2)}
\end{aligned}$$

where we have used $\hat{\mathbf{W}}^* = \hat{\mathbf{W}}$ and that $\hat{\mathbf{W}}^2 = \hat{\mathbf{W}}^*\hat{\mathbf{W}}$ has all diagonal elements equal to $\mu(\mathbf{W}^2)$.

Inserting Eq. (C.2) into Eq. (C.1) we obtain Eq. (31). This ends the proof.

References

[1] M. Sonka, J. M. Fitzpatrick, Handbook of Medical Imaging Vol. 2 Medical Image Processing and Analysis, SPIE, 2004.

[2] G. Strang, The discrete cosine transform, SIAM Rev. 41 (1999) 135–147.

[3] J. R. Shewchuk, An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, Tech. rep., CMU (1994).

[4] R. Bhatia, C. Davis, A Better Bound on the Variance, The American Mathematical Monthly 107 (4) (2000) 353–357.

[5] D. Garcia, Robust smoothing of gridded data in one and higher dimensions with missing values, Comput. Stat. Data An. 54 (2010) 1167–1178.

[6] H. B. Keller, On the solution of singular and semidefinite linear systems by iteration, SIAM B: Numer. Anal. 2 (1965) 281–290.

[7] I. Galic, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, H.-P. Seidel, Image compression with anisotropic diffusion, J. Math. Imaging 31 (2008) 255–269.

[8] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE T. Pattern Anal. 12 (7) (1990) 629–639.

[9] C. Xu, J. L. Prince, Snakes, shapes, and gradient vector flow, IEEE Transactions on Image Processing 7 (3) (1998) 359–369.

[10] D. Barash, T. Schlick, M. Israeli, R. Kimmel, Multiplicative operator splitting in nonlinear diffusion: from spatial splitting to multiple timesteps, Journal of Mathematical Imaging and Vision 19 (1) (2003) 33–48.

[11] J. Lu, W. Zuo, X. Zhao, D. Zhang, An augmented Lagrangian method for fast gradient vector flow computation, in: IEEE Internat. Conf. Imag. Proc., 2011, pp. 1525–1528.

[12] D. Boukerroui, Efficient numerical schemes for gradient vector flow, Pattern Recogn. 45 (1) (2012) 626–636.

[13] J.-P. Thirion, Image matching as a diffusion process: an analogy with Maxwell's demons, Med. Image Anal. 2 (3) (1998) 243–260.

[14] X. Pennec, P. Cachier, N. Ayache, Understanding the "Demon's Algorithm": 3D Non-rigid Registration by Gradient Descent, in: MICCAI, 1999, pp. 597–606.

[15] T. Vercauteren, X. Pennec, A. Perchant, N. Ayache, Non-parametric Diffeomorphic Image Registration with the Demons Algorithm, in: MICCAI, 2007, pp. 319–326.

[16] I. Daubechies, R. DeVore, M. Fornasier, C. S. Gunturk, Iteratively reweighted least squares minimization for sparse recovery, Communications on Pure and Applied Mathematics 63 (1) (2010) 1–38.

[17] D. Bertaccini, R. H. Chan, S. Morigi, F. Sgallari, Lecture Notes in Computer Science, Vol. 6667, 2012, Ch. An Adaptive Norm Algorithm for Image Restoration, pp. 194–205.

[18] M. Nikolova, R. Chan, The equivalence of half-quadratic minimization and the gradient linearization iteration, IEEE Transactions on Image Processing 16 (6) (2007) 1623–1627.

[19] D. Geman, C. Yang, Nonlinear image recovery with half-quadratic regularization and FFTs, IEEE Transactions on Image Processing 4 (7) (1995) 932–946.