Preconditioning for Edge-Preserving Image Super-Resolution

Stéphane Pelletier, Jeremy R. Cooperstock

***Abstract*—We propose a simple preconditioning method for accelerating the solution of edge-preserving image super-resolution (SR) problems in which a linear shift-invariant (LSI) point spread function (PSF) is employed. Our technique involves reordering the high-resolution (HR) pixels in a similar manner to what is done in preconditioning methods for quadratic SR formulations. However, due to the edge-preserving requirements, the Hessian matrix of the cost function varies during the minimization process. We develop an efficient update scheme for the preconditioner in order to cope with this situation. Unlike some other acceleration strategies that round the displacement values between the low-resolution (LR) images on the HR grid, the proposed method does not sacrifice the optimality of the observation model. In addition, we describe a technique for preconditioning SR problems involving rational magnification factors. The use of such factors is motivated in part by the fact that, under certain circumstances, optimal SR zooms are non-integers. We show that by reordering the pixels of the LR images, the structure of the problem to solve is modified in such a way that preconditioners based on circulant operators can be used.**

***Index Terms*—image super-resolution, regularization, edge restoration, preconditioning.**

I. Introduction

When designing and deploying computer vision systems, engineers are often confronted with the limitations of available cameras with respect to the image quality they can provide. Similar to other sensing devices, cameras alter the measured signal and often yield images representing a degraded version of the original scene. Degradation sources include blur induced by such factors as the lens and sensors, frequency aliasing due to subsampling, signal quantization and circuitry reading noise [1]. As a result, fine details are often lost during the image acquisition process.

These factors drive the need for higher quality digital imaging across a spectrum of consumer and industrial applications. Higher quality may be achieved either by adoption of sensors with improved characteristics or through image processing operations, capable of overcoming the resolution limits of imaging systems. The latter approach, often preferable for reasons of cost, is exemplified by super-resolution (SR) algorithms. These refer to techniques for synthesizing a high-resolution (HR) image or video sequence from a set of degraded and aliased low-resolution (LR) ones [2], [3].

Although SR algorithms are well understood, they tend to be computationally expensive, often prohibitively so for time-critical operation. This motivates our investigation of techniques for accelerating computations associated with SR problems. In particular, we concentrate on preconditioning for edge-preserving SR algorithms, which offers the benefit of perceptually improved images with sharper edges.

The remainder of this article is organized as follows. First, in Section II, we review related literature. Following this background, we provide in Section III the formulation of the SR problem studied. More precisely, we describe the image formation model and the cost function to minimize.

In Section IV, we explain how to reorder the HR pixels in the problem formulation such that the minimization of the cost function can be accelerated through preconditioning. For this purpose, we examine the impact of a permutation scheme proposed for quadratic penalized SR problems [4], [5] on the structure of the Hessian matrix of our edge-preserving cost function. We also discuss the reordering of the LR pixels when rational magnification factors are employed. This is an extension of our initial work [6], in which we did not consider the preservation of edges. This application is motivated in part by the work of Lin and Shum [7] suggesting that, under certain circumstances, optimal magnification factors for SR are non-integers. Although Sroubek *et al.* [8] concurrently investigated the use of rational zooms in the blind SR framework, they did not consider preconditioning methods for SR.

In Section V, we discuss the minimization of the cost function using the preconditioned nonlinear conjugate gradient (PNCG) method. Since the structure of the Hessian matrix associated with our edge-preserving SR formulation varies during the minimization process, we propose an efficient update scheme for the preconditioner in order to cope with this situation. A complexity analysis is also provided to validate the proposed approach. Finally, we discuss the application of our acceleration technique to color images.

In Section VI, we provide a detailed analysis of the effectiveness of our approach under different conditions of practical interest. In particular, we study the convergence behavior of our method when suboptimal regularization parameter values are employed. This is highly important in practice, since several restoration processes must often be performed in order to determine the optimal values for these parameters. We also justify our choice of accelerating a specific edge-preserving cost function by providing a visual comparison of our results against those obtained using alternative SR formulations.

S. Pelletier and J. R. Cooperstock are with the Department of Electrical and Computer Engineering, McGill University, Montréal, Canada

Manuscript received August 9, 2010; revised April 7, 2011.

II. Background

The super-resolution literature includes excellent summaries of existing methods by Borman and Stevenson [9] and Park *et al.* [2]. The topic has been addressed in several special issues in the signal processing literature [10]–[12] and in two books by Chaudhuri [3], [13].

Of specific relevance to the topic of this paper, preconditioning methods for accelerating quadratic (non edge-preserving) SR formulations employing a LSI blur were presented by Nguyen *et al.* [4] and Bose *et al.* [5]. With these methods, the PSF does not need to be identical for all LR images. Issues related to the use of rational magnification factors were not discussed by these authors. In the problem formulation of both methods, the HR pixels are reordered to allow for the application of preconditioners derived from circulant matrices. We shall employ the same reordering scheme in our preconditioning technique, since the image formation models associated with quadratic and edge-preserving SR formulations are similar.

When the blur is identical for all LR images, other acceleration methods shift and interpolate the LR frames on a single HR grid, and then apply some iterative deblurring process to this temporary image in order to synthesize the HR image [14]–[19]. The efficiency of such approaches comes from the fact that the computations involving several LR images are replaced by computations on a single image. Obviously, this advantage becomes more significant as the number of LR images increases. However, the optimality of the HR images obtained in this manner is difficult to verify, since the interpolation process introduces errors in the input data.

Nonetheless, when the motion model is translational, Elad and Hel-Or [20] showed that an earlier technique proposed by Gillette *et al.* [17] for fusing the LR images can preserve the optimality of the SR reconstruction process in the maximum-likelihood sense. This optimality relies on the assumption that the translations between the LR frames follow the nearest neighbor displacement paradigm [21], i.e., that the displacement magnitudes are multiples of the size of a HR pixel. Farsiu *et al.* [22] later adapted this method to the robust SR case with an edge-preserving prior. Tanaka and Okutomi [23] also exploited this idea by grouping LR pixels whose positions are the same after discretization and computing the average of the pixel values in each group. The computations are then performed on this reduced set of averaged values instead of the original data. All the methods above assume that rounding the displacement values has a negligible effect on image quality, since it occurs on the HR grid.

Some methods employ preconditioning to accelerate the deblurring process after combining the LR images on the HR grid. R. Chan *et al.* [24] considered the special case where the number of LR images is equal to the square of the magnification factor. Their approach transforms the SR problem into a deblurring one by interleaving the pixels from the LR images to produce a single HR blurred image. They then employ the preconditioned conjugate gradient (PCG) method in conjunction with a cosine transform preconditioner to recover the HR image. Ng *et al.* [25] proposed a similar approach in which perturbations in the registration parameters are taken into account. This algorithm was also applied independently to individual color channels for color image SR [26]. Lin *et al.* [27] performed experiments using incomplete factorization preconditioners on the same problem. In all these techniques, a simple quadratic prior was employed to regularize the solution, which tends to reduce image quality around sharp edges.

III. Problem formulation

Before introducing our acceleration method for super-resolution, we first describe our assumptions regarding the image formation model that induces the loss of resolution from which we wish to recover. We then discuss the choice of cost function, and motivate the use of the Huber function based on its edge-preserving properties.

A. Image formation model

Similar to many other image processing algorithms, the objective of SR is to reverse the degradation process undergone by a scene during its capture by a digital device. Formally, we define the problem as that of recovering a HR image $\mathcal{X} \in \mathbb{R}^{N\times N}$ from L observed LR images $\mathcal{Y}_l \in \mathbb{R}^{M\times M}$, where $l = 1 \dots L$. The relationship between M and N is given by $N = QM$, where Q is the desired *magnification factor*.

To reconstruct $\mathcal{X}$, a discrete image formation model describing the mathematical relationship between the observed pixels and the unknown ones needs to be defined. For this purpose, let $\mathbf{x} \triangleq \text{vec}(\mathcal{X})$ be a vector of length N^2 whose elements are the pixel values of $\mathcal{X}$ in a given lexicographic order. Similarly, $\mathbf{y}_l \triangleq \text{vec}(\mathcal{Y}_l)$ is a vector of length M^2 representing the LR image $\mathcal{Y}_l$. The imaging process yielding an observed LR image $\mathbf{y}_l$ is often assumed to be linear and can thus be expressed in matrix form as:

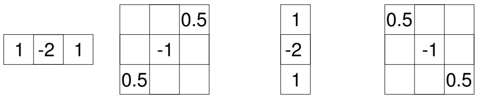
$$\mathbf{y}_l = \mathbf{G}_l\mathbf{x} + \mathbf{n}_l, \tag{1}$$

where $\mathbf{G}_l \in \mathbb{R}^{M^2\times N^2}$ is a matrix describing the imaging geometry and $\mathbf{n}_l$ is an additive noise vector. The matrix $\mathbf{G}_l$ can be factored as follows:

$$\mathbf{G}_l = \mathbf{D}\mathbf{U}_l\mathbf{B}_l\mathbf{S}_l, \tag{2}$$

where $\mathbf{D} \in \mathbb{R}^{M^2\times N^2}$ is a downsampling matrix, $\mathbf{U}_l \in \mathbb{R}^{N^2\times N^2}$ and $\mathbf{B}_l \in \mathbb{R}^{N^2\times N^2}$ are blur matrices associated with the sensor and the lens of the l^{th} camera respectively, and $\mathbf{S}_l \in \mathbb{R}^{N^2\times N^2}$ is a shift matrix representing the relative displacement between $\mathcal{Y}_l$ and $\mathcal{X}$.

In this work, we assume that the motion in the scene is purely translational. In particular, such a model can represent small vibrations of a camera or account for images obtained by slightly panning the camera while capturing a video sequence of a distant scene. Furthermore, we assume that the image formation matrices $\{\mathbf{G}_l\}_{l=1}^L$ are known. In particular, the shift matrices $\{\mathbf{S}_l\}_{l=1}^L$ are determined by the relative displacements between the LR frames, which can be computed using image registration algorithms [21], [28]. When the cameras are in focus, the lens blur matrices $\{\mathbf{B}_l\}_{l=1}^L$ are equal to the identity matrix. Otherwise, these matrices can be constructed



1	-2	1

		0.5
	-1	
0.5		

1
-2
1

0.5		
	-1	
		0.5

Fig. 1. Finite difference approximation masks to second-order derivatives in four directions used for regularization.

by estimating the PSF of the blurs induced by the lens. The downsampling matrix $\mathbf{D}$ and the image sensor blur matrices $\{\mathbf{U}_l\}_{l=1}^{L}$ are determined by the magnification factor Q and the *fillfactor*[1] of the image sensor in the camera. Finally, we shall assume that the PSF associated with the image sensor of the camera and its lens are both space-invariant.

Under these assumptions, each matrix $\mathbf{U}_l$, $\mathbf{B}_l$ and $\mathbf{S}_l$ has a block circulant with circulant block (BCCB) structure [29] when the periodic boundary condition is employed. More precisely, we have $\mathbf{U}_l, \mathbf{B}_l, \mathbf{S}_l \in \mathcal{BCCB}(N)$, where $\mathcal{BCCB}(N)$ denotes the set of $N \times N$ block circulant matrices with $N \times N$ circulant blocks.

B. Cost function

Since Bayesian methods offer great flexibility for modeling noise characteristics and a priori knowledge about the solution, they are very popular for solving ill-posed problems such as super-resolution [20], [22], [30]. To perform edge-preserving SR, we minimize a penalized least-squares cost function $\Phi : \mathbb{R}^{N^2} \to \mathbb{R}$ of the form:

$$\Phi(\mathbf{x}) = \frac{1}{2}\sum_{l=1}^{L} \|\mathbf{y}_l - \mathbf{G}_l\mathbf{x}\|_2^2 + \lambda \sum_{m=1}^{4}\sum_{k=1}^{N^2} \phi([\mathbf{C}_m\mathbf{x}]_k), \quad (3)$$

where $\lambda \in \mathbb{R}$ is a parameter, $\mathbf{C}_m \in \mathbb{R}^{N^2 \times N^2}$ is a linear operator, $\|\cdot\|_2$ is the usual Euclidean norm and $\phi : \mathbb{R} \to \mathbb{R}$ is called a *potential function*.

The first term of the sum favors solutions that are consistent with the observed data $\{\mathbf{y}_l\}_{l=1}^{L}$, whereas the second term penalizes solutions that do not agree with a priori assumptions about the structure of $\mathbf{x}$. The tradeoff between these two criteria is controlled by the value of λ, which can be chosen based on available information about the nature of $\mathbf{x}$ and the degree of perturbations in the data [31]. When no such information is available, one can try solving the problem for several values of λ [32], or use generalized cross-validation methods [33].

We employ finite difference approximations to second-order derivatives in four directions [34] as the image smoothness measure. The effect of multiplying matrices $\{\mathbf{C}_m\}_{m=1}^{4}$ in Equation (3) with $\mathbf{x}$ can be understood as filtering the associated image $\mathcal{X}$ with the four masks shown in Fig. 1.

The choice of ϕ is generally based on prior knowledge about the nature of $\mathcal{X}$. In this work, we employ the Huber function [35], which is defined as follows:

$$\phi(t) = \begin{cases} \frac{t^2}{2}, & |t| \leq \alpha \\ \alpha|t| - \frac{\alpha^2}{2}, & |t| > \alpha \end{cases}. \quad (4)$$

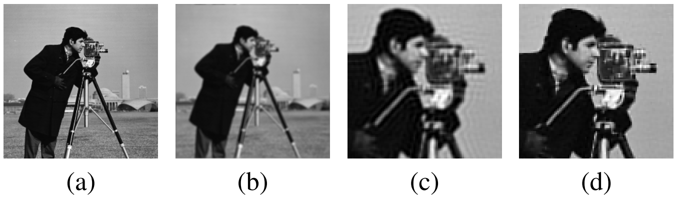


Fig. 2. Effect of the choice of the potential function $\phi(t)$ in Equation (3) on the restored image. (a) Original HR image. (b) One of the degraded LR images. (c) A region of the image restored using $\phi(t) = \frac{t^2}{2}$. (d) Corresponding region of the image restored using Equation (4) with $\alpha = 0.01$.

This function quadratically penalizes small discontinuities in the image, which are often associated with noise, whereas large discontinuities (actual edges) are only linearly penalized. The transition point between the quadratic and linear regions is determined by the parameter α. The higher this value, the more the Huber function becomes equivalent to a simple quadratic function.

To illustrate the advantage of using this penalty function, the image shown in Fig. 2(a) was blurred using a 5×5 Gaussian kernel with variance 4. The resulting blurred image was then downsampled by a factor of two using different spatial offsets, yielding a set of four LR images, one of which is shown in Fig. 2(b). These LR images were then restored twice by minimizing Equation (3) using the nonlinear conjugate gradient (NCG) method [36]. In the first experiment, the quadratic penalty term $\phi(t) = \frac{t^2}{2}$ was employed. In the second experiment, the penalty function (4) was used with $\alpha = 0.01$. The regularization parameter λ was set to 0.001 in both experiments. Corresponding regions of the images restored using the quadratic penalty term and Equation (4) are presented in Fig. 2(c) and Fig. 2(d), respectively. The advantages of using an edge-preserving regularization term are quite obvious from this example.

IV. Hessian matrix structure

The minimization of the cost function (3) can often be accelerated by employing a preconditioner. The efficiency of this approach depends on the structure of the Hessian matrix of the cost function, i.e., the square matrix of its second-order partial derivatives. When the potential function (4) is employed, this matrix is not mathematically defined at every $\mathbf{x}$, since the Huber penalty function is not twice continuously differentiable at $t = \pm\alpha$. However, since the probabilities of obtaining a minimizer $\mathbf{x}$ such that $|[\mathbf{C}_m\mathbf{x}]_k| = \alpha$ for some k are null [37], we assume that the Hessian matrix $\mathbf{H}(\mathbf{x})$ exists everywhere and is given by:

$$\mathbf{H}(\mathbf{x}) = \sum_{l=1}^{L} \mathbf{G}_l^T\mathbf{G}_l + \lambda \sum_{m=1}^{4} \mathbf{C}_m^T\mathbf{D}_m(\mathbf{x})\mathbf{C}_m, \quad (5)$$

where $\mathbf{D}_m(\mathbf{x}) \in \mathbb{R}^{N^2 \times N^2}$ is the diagonal matrix defined as

$$[\mathbf{D}_m(\mathbf{x})]_{k,k} = \begin{cases} 1, & |[\mathbf{C}_m\mathbf{x}]_k| \leq \alpha \\ 0, & \text{otherwise} \end{cases}. \quad (6)$$

Preconditioners based on BCCB matrices have been shown to be very effective for accelerating image deblurring problems.

[1] The fillfactor is the percentage of a pixel sensor surface that is sensitive to light.

However, the downsampling matrix $\mathbf{D}$ in Equation (2) impacts the structure of $\mathbf{G}_l$ in a way that prevents the direct application of a preconditioning strategy proposed for edge-preserving image deblurring [38]. This problem is similar to that encountered in other SR formulations employing a quadratic penalty term [4], [5]. To better illustrate the effect of $\mathbf{D}$ on the Hessian matrix, we consider a simple SR scenario in which a HR image $\mathcal{X} \in \mathbb{R}^{8\times 8}$ is computed from two LR images $\mathcal{Y}_1, \mathcal{Y}_2 \in \mathbb{R}^{4\times 4}$. The relative displacements of $\mathcal{Y}_1$ and $\mathcal{Y}_2$ with respect to $\mathcal{X}$ are given by $[0.3, 0.7]$ and $[0.5, 0.1]$ respectively, where the notation $[a, b]$ refers to a translation of a units along the first dimension and b units along the second one. This scenario corresponds to the use of the following parameter values: $N = 8$, $M = 4$, $Q = 2$ and $L = 2$. Assuming for simplicity that $\mathbf{B}_1$ and $\mathbf{B}_2$ are the identity matrix, i.e., lens blur is not considered, and that the periodic boundary condition is employed, the structure of the image formation model matrix $\mathbf{G} \triangleq [\mathbf{G}_1^T, \mathbf{G}_2^T]^T \in \mathbb{R}^{32\times 64}$ associated with this SR problem is shown in Fig. 3(a). In this image representation of $\mathbf{G}$, squares whose gray level intensities are the same denote matrix elements of equal value. Furthermore, the thick horizontal line denotes the separation between the two components of $\mathbf{G}$, namely $\mathbf{G}_1$ and $\mathbf{G}_2$. The structure of $\mathbf{G}^T\mathbf{G}$ is presented in Fig. 3(b) and the region indicated by the dark square is expanded in Fig. 3(c). As can be seen, $\mathbf{G}^T\mathbf{G}$ does not have a BCCB structure.

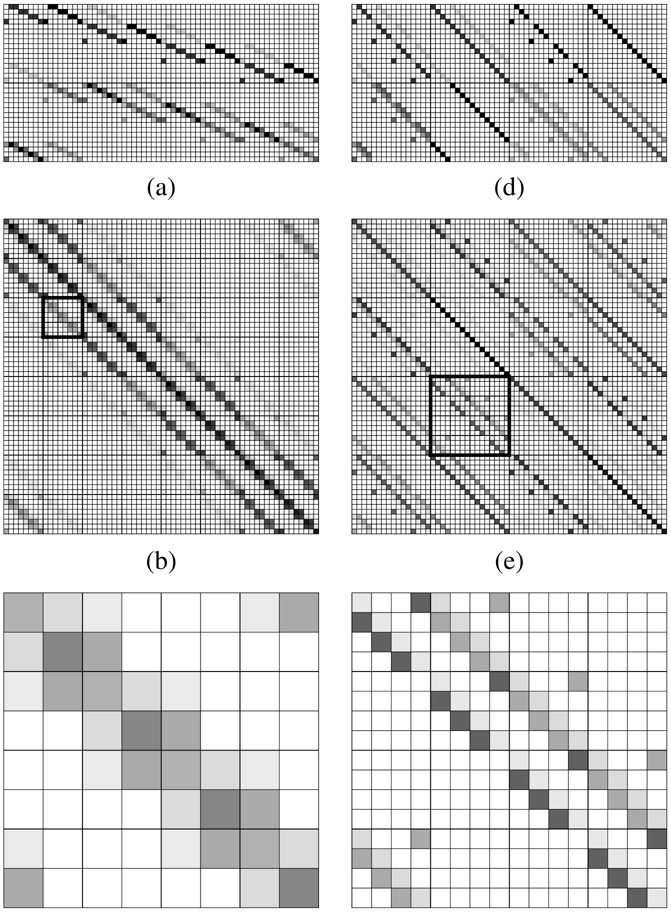


Fig. 3. Structures of $\mathbf{G} \triangleq [\mathbf{G}_1^T, \mathbf{G}_2^T]^T$, $\mathbf{G}^T\mathbf{G}$, $\tilde{\mathbf{G}} \triangleq [\tilde{\mathbf{G}}_1^T, \tilde{\mathbf{G}}_2^T]^T$ and $\tilde{\mathbf{G}}^T\tilde{\mathbf{G}}$ corresponding to a SR problem employing the periodic boundary condition with the following parameters: $N = 8$, $M = 4$, $Q = 2$ and $L = 2$. The permutation matrix in Equation (8) is $\mathbf{P} = \mathbf{Z}^{8,2}$. (a) Matrix $\mathbf{G} \in \mathbb{R}^{32\times 64}$. (b) Matrix $\mathbf{G}^T\mathbf{G} \in \mathbb{R}^{64\times 64}$. (c) Region indicated in (b). (d) Matrix $\tilde{\mathbf{G}} \in \mathbb{R}^{32\times 64}$. (e) Matrix $\tilde{\mathbf{G}}^T\tilde{\mathbf{G}} \in \mathbb{R}^{64\times 64}$. (f) BCCB block indicated in (e).

By reordering the columns of $\mathbf{G}_l$ and the elements of $\mathbf{x}$ in a similar manner, one can modify the structure of $\mathbf{G}_l$ without changing the correctness of the image formation model. The primary motivation for performing these permutations is to obtain a cost function whose Hessian matrix has a structure that is more suitable for preconditioning.

Mathematically, we can represent this reordering step using a permutation matrix $\mathbf{P} \in \mathbb{R}^{N^2\times N^2}$. Since $\mathbf{P}$ is a permutation matrix, it is easy to show that $\mathbf{P}^T\mathbf{P} = \mathbf{I}$, and thus, one can rewrite the image formation model of Equation (1) as:

$$\mathbf{y}_l = \tilde{\mathbf{G}}_l\tilde{\mathbf{x}} + \mathbf{n}_l, \tag{7}$$

where $\tilde{\mathbf{x}} \triangleq \mathbf{P}\mathbf{x}$ and

$$\tilde{\mathbf{G}}_l \triangleq \mathbf{G}_l\mathbf{P}^T. \tag{8}$$

Note that $\mathbf{P}\mathbf{x}$ permutes the elements of $\mathbf{x}$, i.e., the HR pixels, whereas $\mathbf{G}_l\mathbf{P}^T$ permutes the columns of $\mathbf{G}_l$. The new minimization problem is then given by

$$\tilde{\Phi}(\tilde{\mathbf{x}}) = \frac{1}{2}\sum_{l=1}^{L}\|\mathbf{y}_l - \tilde{\mathbf{G}}_l\tilde{\mathbf{x}}\|_2^2 + \lambda\sum_{m=1}^{4}\sum_{k=1}^{N^2}\phi([\tilde{\mathbf{C}}_m\tilde{\mathbf{x}}]_k), \tag{9}$$

where $\tilde{\mathbf{C}}_m \triangleq \mathbf{P}\mathbf{C}_m\mathbf{P}^T$ is a permuted version of the matrix operator $\mathbf{C}_m$ compatible with the structure of $\tilde{\mathbf{x}}$; the presence of $\mathbf{P}$ on the left of $\mathbf{C}_m$ in the definition of $\tilde{\mathbf{C}}_m$ will be justified later. Once the minimum $\tilde{\mathbf{x}}$ to this cost function is computed, the desired solution $\mathbf{x}$ can be recovered using the following relation:

$$\mathbf{x} = \mathbf{P}^T\tilde{\mathbf{x}}. \tag{10}$$

One can easily verify that the minimization of Equation (9) followed by the application of the relation (10) yields the same result as that obtained through the direct minimization of the original cost function (3).

The Hessian matrix $\tilde{\mathbf{H}}(\tilde{\mathbf{x}})$ of the new cost function (9) is given by:

$$\tilde{\mathbf{H}}(\tilde{\mathbf{x}}) = \sum_{l=1}^{L}\tilde{\mathbf{G}}_l^T\tilde{\mathbf{G}}_l + \lambda\sum_{m=1}^{4}\tilde{\mathbf{C}}_m^T\tilde{\mathbf{D}}_m(\tilde{\mathbf{x}})\tilde{\mathbf{C}}_m, \tag{11}$$

where $\tilde{\mathbf{D}}_m(\tilde{\mathbf{x}})$ is the $N^2\times N^2$ diagonal matrix defined as

$$[\tilde{\mathbf{D}}_m(\tilde{\mathbf{x}})]_{k,k} = \left\{\begin{array}{ll} 1, & |[\tilde{\mathbf{C}}_m\tilde{\mathbf{x}}]_k| \le \alpha \\ 0, & \text{otherwise} \end{array}\right. . \tag{12}$$

To determine $\mathbf{P}$, we use the reordering strategy proposed by Nguyen *et al.* [4], which corresponds to taking the polyphase decomposition [39] of the HR image.

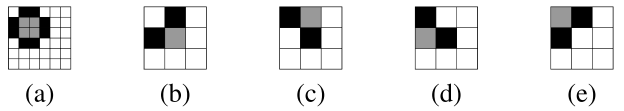


Fig. 4. Images obtained by applying matrices (b) $\mathbf{Z}^{6,2}_{1,1}$, (c) $\mathbf{Z}^{6,2}_{2,1}$, (d) $\mathbf{Z}^{6,2}_{1,2}$ and (e) $\mathbf{Z}^{6,2}_{2,2}$ to the image shown in (a).

A. Polyphase decomposition of $\mathcal{X}$

To describe the effects of polyphase decomposition when applied to images, we introduce the following definitions: let $\mathbf{Z}^{n,q}_{i,j} \in \mathbb{R}^{\frac{n^2}{q^2}\times n^2}$ be the matrix representation of a linear operator that downsamples an image $\mathcal{X} \in \mathbb{R}^{n\times n}$ by a factor of $q \in \mathbb{N}$ along both axes, starting with pixel $\mathcal{X}_{i,j}$. We assume that n is a multiple of q. Furthermore, let $\mathcal{Z}^{n,q}$ denote the set of all matrices $\mathbf{Z}^{n,q}_{i,j}$ such that $1 \leq i, j \leq q$. For example, multiplying the four matrices that belong to the set $\mathcal{Z}^{6,2}$, namely $\mathbf{Z}^{6,2}_{1,1}$, $\mathbf{Z}^{6,2}_{2,1}$, $\mathbf{Z}^{6,2}_{1,2}$ and $\mathbf{Z}^{6,2}_{2,2}$, by the vector form of the image shown in Fig. 4(a) would yield vectors representing the images shown in Figures 4(b)-(e).

The q^2-components polyphase decomposition of $\mathcal{X}$ is defined as $\mathbf{Z}^{n,q}\mathbf{x}$, where $\mathbf{x} \triangleq \mathrm{vec}(\mathcal{X})$ and $\mathbf{Z}^{n,q} \in \mathbb{R}^{n^2\times n^2}$ is the square matrix obtained by stacking on top of one another the q^2 matrices in the set $\mathcal{Z}^{n,q}$. It is easy to see that $\mathbf{Z}^{n,q}$ is a permutation matrix, since the polyphase decomposition of $\mathcal{X}$, i.e., $\mathbf{Z}^{n,q}\mathbf{x}$, simply reorders its elements.

Using the previous notation, the permutation matrix is given by $\mathbf{P} = \mathbf{Z}^{N,Q}$. The resulting matrix $\tilde{\mathbf{G}}_l$ has the following structure [5]:

$$\tilde{\mathbf{G}}_l = \left[\ \tilde{\mathbf{G}}_{l,1} \quad \tilde{\mathbf{G}}_{l,2} \quad \dots \quad \tilde{\mathbf{G}}_{l,Q^2}\ \right], \tag{13}$$

where $\tilde{\mathbf{G}}_{l,i} \in \mathcal{BCCB}(M)$. Each term $\tilde{\mathbf{G}}^T_l\tilde{\mathbf{G}}_l$ that appears in Equation (11) can then be partitioned as follows:

$$\tilde{\mathbf{G}}^T_l\tilde{\mathbf{G}}_l = \begin{bmatrix} \tilde{\mathbf{G}}^T_{l,1}\tilde{\mathbf{G}}_{l,1} & \dots & \tilde{\mathbf{G}}^T_{l,1}\tilde{\mathbf{G}}_{l,Q^2} \\ \vdots & & \vdots \\ \tilde{\mathbf{G}}^T_{l,Q^2}\tilde{\mathbf{G}}_{l,1} & \dots & \tilde{\mathbf{G}}^T_{l,Q^2}\tilde{\mathbf{G}}_{l,Q^2} \end{bmatrix}. \tag{14}$$

Given that all blocks in Equation (13) are BCCB matrices, any product $\tilde{\mathbf{G}}^T_{l,i}\tilde{\mathbf{G}}_{l,j}$ also has this structure, and thus, $\tilde{\mathbf{G}}^T_l\tilde{\mathbf{G}}_l$ is a symmetric block matrix with BCCB blocks. To illustrate this, the matrices $\tilde{\mathbf{G}} \triangleq [\tilde{\mathbf{G}}^T_1, \tilde{\mathbf{G}}^T_2]^T$ and $\tilde{\mathbf{G}}^T\tilde{\mathbf{G}}$ resulting from a reordering of the HR pixels in the simple SR scenario described before are shown in Fig. 3(d) and Fig. 3(e), respectively. In this example, the permutation matrix is $\mathbf{P} = \mathbf{Z}^{8,2}$. To better visualize the BCCB structure of the blocks of $\tilde{\mathbf{G}}^T\tilde{\mathbf{G}}$, the region identified in Fig. 3(e) is enlarged in Fig. 3(f).

We now examine the structure of each term $\tilde{\mathbf{C}}^T_m\tilde{\mathbf{D}}_m(\tilde{\mathbf{x}})\tilde{\mathbf{C}}_m$ in Equation (11). First, an important observation is that unlike $\mathbf{G}_l$, the matrix $\mathbf{C}_m$ in the Hessian matrix (5) of the original problem is already a BCCB matrix. Consequently, permuting the columns of $\mathbf{C}_m$ according to $\mathbf{P}$, which is required by the new ordering of the HR pixels, destroys this structure. For this reason, we also reorder the rows of $\mathbf{C}_m$, which explains the presence of $\mathbf{P}$ on the left of this matrix in the definition of $\tilde{\mathbf{C}}_m$ in Equation (9). Each matrix $\tilde{\mathbf{C}}_m$ has the following block structure:

$$\tilde{\mathbf{C}}_m = \begin{bmatrix} \tilde{\mathbf{C}}_{m,1,1} & \dots & \tilde{\mathbf{C}}_{m,1,Q^2} \\ \vdots & & \vdots \\ \tilde{\mathbf{C}}_{m,Q^2,1} & \dots & \tilde{\mathbf{C}}_{m,Q^2,Q^2} \end{bmatrix}, \tag{15}$$

where $\tilde{\mathbf{C}}_{m,q,i} \in \mathcal{BCCB}(M)$. Also, we partition the diagonal matrix $\tilde{\mathbf{D}}_m(\tilde{\mathbf{x}})$ in the Hessian matrix (11) as follows:

$$\tilde{\mathbf{D}}_m(\tilde{\mathbf{x}}) = \begin{bmatrix} \tilde{\mathbf{D}}_{m,1}(\tilde{\mathbf{x}}) & & \\ & \ddots & \\ & & \tilde{\mathbf{D}}_{m,Q^2}(\tilde{\mathbf{x}}) \end{bmatrix}, \tag{16}$$

where $\tilde{\mathbf{D}}_{m,q}(\tilde{\mathbf{x}}) \in \mathbb{R}^{M^2\times M^2}$ is a diagonal matrix. The permuted Hessian matrix $\tilde{\mathbf{H}}(\tilde{\mathbf{x}})$ can then be expressed as:

$$\tilde{\mathbf{H}}(\tilde{\mathbf{x}}) = \begin{bmatrix} \tilde{\mathbf{H}}_{1,1}(\tilde{\mathbf{x}}) & \dots & \tilde{\mathbf{H}}_{1,Q^2}(\tilde{\mathbf{x}}) \\ \vdots & & \vdots \\ \tilde{\mathbf{H}}_{Q^2,1}(\tilde{\mathbf{x}}) & \dots & \tilde{\mathbf{H}}_{Q^2,Q^2}(\tilde{\mathbf{x}}) \end{bmatrix}, \tag{17}$$

where

$$\tilde{\mathbf{H}}_{i,j}(\tilde{\mathbf{x}}) = \sum_{l=1}^{L} \tilde{\mathbf{G}}^T_{l,i}\tilde{\mathbf{G}}_{l,j} + \lambda \sum_{m=1}^{4}\sum_{q=1}^{Q^2} \tilde{\mathbf{C}}^T_{m,q,i}\tilde{\mathbf{D}}_{m,q}(\tilde{\mathbf{x}})\tilde{\mathbf{C}}_{m,q,j}. \tag{18}$$

Each block $\tilde{\mathbf{H}}_{i,j}(\tilde{\mathbf{x}})$ has a structure similar to the Hessian matrix encountered in edge-preserving image deblurring problems [38], [40]. Note that when a quadratic penalty function is employed in the regularization term of the cost function, the diagonal matrices $\tilde{\mathbf{D}}_{m,q}(\tilde{\mathbf{x}})$ in Equation (18) disappear and $\tilde{\mathbf{H}}_{i,j}(\tilde{\mathbf{x}})$ becomes a BCCB matrix. In this particular case, the preconditioning methods proposed by Nguyen *et al.* and Bose *et al.* can be employed. In edge-preserving SR scenarios, these diagonal matrices must be taken into account when constructing the preconditioner and a method for doing so will be presented in Section V. However, before describing this technique, we analyze the structure of the SR problem in the special case where rational magnification factors are employed.

B. Rational magnification factors

When the magnification factor Q is not an integer, one cannot find a permutation matrix $\mathbf{P}$ such that $\mathbf{G}_l\mathbf{P}^T$ is a block matrix with BCCB blocks. However, if Q is a rational number, i.e., if it can be written as $\frac{A}{B}$, where $A, B \in \mathbb{N}^*$, and M is chosen to be a multiple of B, then one can find two permutation matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ such that $\mathbf{P}_2\mathbf{G}_l\mathbf{P}^T_1$ has the desired structure. More precisely, these permutation matrices are given by

$$\mathbf{P}_1 = \mathbf{Z}^{N,A} \text{ and } \mathbf{P}_2 = \mathbf{Z}^{M,B}. \tag{19}$$

Note that since $\mathbf{P}_2$ reorders the rows of $\mathbf{G}_l$, the LR pixels in the image formation model must be permuted in a similar manner in order to preserve the consistency of the observation model.

TABLE I
SUMMARY OF A PNCG ITERATION.

	Operations	Description
$\mathbf{r}^{(i)}$	$= -\nabla\tilde{\Phi}(\tilde{\mathbf{x}}^{(i-1)})$	gradient
$\mathbf{p}^{(i)}$	$= [\mathbf{M}(\tilde{\mathbf{x}}^{(i-1)})]^{-1}\mathbf{r}^{(i)}$	preconditioner
β	$= \frac{<\mathbf{r}^{(i)},\mathbf{p}^{(i)}>}{<\mathbf{r}^{(i-1)},\mathbf{p}^{(i-1)}>}$	Fletcher-Reeves rule
$\mathbf{d}^{(i)}$	$= \mathbf{p}^{(i)} + \beta\mathbf{d}^{(i-1)}$	search direction
$\alpha^{(i)}$	$= \mathrm{argmin}_{\alpha}\, \Phi(\tilde{\mathbf{x}}^{(i-1)} + \alpha\mathbf{d}^{(i)})$	line search
$\tilde{\mathbf{x}}^{(i)}$	$= \tilde{\mathbf{x}}^{(i-1)} + \alpha^{(i)}\mathbf{d}^{(i)}$	solution update

Permuting the rows and columns of $\mathbf{G}_l$ results in a reordered matrix $\tilde{\mathbf{G}}_l \triangleq \mathbf{P}_2\mathbf{G}_l\mathbf{P}_1^T$ with a block structure [5]:

$$\tilde{\mathbf{G}}_l = \begin{bmatrix} \tilde{\mathbf{G}}_{l,1,1} & \dots & \tilde{\mathbf{G}}_{l,1,A^2} \\ \vdots & & \vdots \\ \tilde{\mathbf{G}}_{l,B^2,1} & \dots & \tilde{\mathbf{G}}_{l,B^2,A^2} \end{bmatrix}, \tag{20}$$

where $\tilde{\mathbf{G}}_{l,i,j} \in \mathcal{BCCB}(\frac{M}{B})$. Upon comparing Equations (13) and (20), one can see that the latter contains B^2 times as many rows of BCCB matrices as the former. This illustrates that a SR problem employing a rational magnification factor $\frac{A}{B}$ can be transformed into a SR problem with an integer factor $Q = A$ simply by applying the polyphase decomposition $\mathbf{Z}^{M,B}$ to every LR image. Consequently, the theory presented in the previous section can readily be applied to the rational magnification factor case upon replacing Q with A in all equations.

V. COST FUNCTION MINIMIZATION

The cost function (9) is convex, and thus, its minimization is equivalent to finding the zero of its gradient. Since the solution to this problem is nonlinear with respect to the observed data, iterative solvers must be employed. Although the method of steepest descent is one of the simplest iterative techniques available for this purpose, other minimization algorithms such as the nonlinear conjugate gradient (NCG) method are known to provide faster convergence [36]. In this work, we use the preconditioned nonlinear conjugate gradient (PNCG) method with the Fletcher-Reeves update rule [36]. Due to the presence of a preconditioner, this technique can yield faster convergence than the simpler NCG method. This iterative algorithm performs the operations described in Table I at every iteration i. The gradient of this cost function is given by

$$\nabla\tilde{\Phi}(\tilde{\mathbf{x}}) = \tilde{\mathbf{G}}^T(\tilde{\mathbf{G}}\tilde{\mathbf{x}} - \mathbf{y}) + \lambda\tilde{\mathbf{C}}^T\mathbf{z}(\tilde{\mathbf{x}}), \tag{21}$$

where $\mathbf{z} : \mathbb{R}^{N^2} \to \mathbb{R}^{4N^2}$ is defined as

$$z_k(\tilde{\mathbf{x}}) \triangleq \dot{\phi}([\tilde{\mathbf{C}}\tilde{\mathbf{x}}]_k), \tag{22}$$

and $\tilde{\mathbf{G}}$, $\tilde{\mathbf{C}}$ and $\mathbf{y}$ represent the respective concatenations of all elements in the sets $\{\tilde{\mathbf{G}}_l\}_{l=1}^L$, $\{\tilde{\mathbf{C}}_m\}_{m=1}^4$ and $\{\mathbf{y}_l\}_{l=1}^L$, as follows:

$$\tilde{\mathbf{G}} \triangleq \begin{bmatrix} \tilde{\mathbf{G}}_1 \\ \vdots \\ \tilde{\mathbf{G}}_L \end{bmatrix}, \tilde{\mathbf{C}} \triangleq \begin{bmatrix} \tilde{\mathbf{C}}_1 \\ \vdots \\ \tilde{\mathbf{C}}_4 \end{bmatrix} \text{ and } \mathbf{y} \triangleq \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_L \end{bmatrix}. \tag{23}$$

A. Line search step

At each PNCG iteration, optimizing the cost function along the current search direction $\mathbf{d}^{(i)}$ is achieved by performing a *line search*, which consists of determining the step size α minimizing

$$f(\alpha) \triangleq \tilde{\Phi}(\tilde{\mathbf{x}}^{(i-1)} + \alpha\mathbf{d}^{(i)}). \tag{24}$$

Although there exist general-purpose techniques for achieving this minimization [41], it is better to use a method that exploits the convexity of our cost function. In particular, we employ the line search technique described by Fessler and Booth [42], which is an adaptation of the iterative method proposed by Huber [35] for robust M-estimation.

B. Proposed preconditioner

To be effective, the proposed preconditioner should be a "good" approximation to the Hessian matrix of the permuted cost function (9). To describe our preconditioner $\mathbf{M}(\tilde{\mathbf{x}})$, we define the following quantities:

$$\mathbf{M}_{i,j}^{\{\mathbf{G}\}} \triangleq \sum_{l=1}^{L} \tilde{\mathbf{G}}_{l,i}^T\tilde{\mathbf{G}}_{l,j}, \tag{25}$$

$$\mathbf{M}_{i,j}^{\{\mathbf{C}\}} \triangleq \sum_{m=1}^{4}\sum_{q=1}^{Q^2} \tilde{\mathbf{C}}_{m,q,i}^T\tilde{\mathbf{C}}_{m,q,j}, \tag{26}$$

$$\beta(\tilde{\mathbf{x}}) \triangleq \frac{\lambda}{4N^2}\sum_{m=1}^{4}\sum_{r=1}^{N^2}[\tilde{\mathbf{D}}_m(\tilde{\mathbf{x}})]_{r,r}. \tag{27}$$

Our preconditioner is then given by:

$$\mathbf{M}(\tilde{\mathbf{x}}) \triangleq \begin{bmatrix} \mathbf{M}_{1,1}(\tilde{\mathbf{x}}) & \dots & \mathbf{M}_{1,Q^2}(\tilde{\mathbf{x}}) \\ \vdots & & \vdots \\ \mathbf{M}_{Q^2,1}(\tilde{\mathbf{x}}) & \dots & \mathbf{M}_{Q^2,Q^2}(\tilde{\mathbf{x}}) \end{bmatrix}, \tag{28}$$

where $\mathbf{M}_{i,j}(\tilde{\mathbf{x}}) \in \mathcal{BCCB}(M)$ is an approximation to $\tilde{\mathbf{H}}_{i,j}(\tilde{\mathbf{x}})$ defined as:

$$\mathbf{M}_{i,j}(\tilde{\mathbf{x}}) = \mathbf{M}_{i,j}^{\{\mathbf{G}\}} + \beta(\tilde{\mathbf{x}})\mathbf{M}_{i,j}^{\{\mathbf{C}\}}. \tag{29}$$

Note that unlike some other preconditioners proposed for non edge-preserving image restoration problems [43], $\mathbf{M}(\tilde{\mathbf{x}})$ is not the optimal BCCB approximation to $\tilde{\mathbf{H}}(\tilde{\mathbf{x}})$ with respect to any specific norm. Indeed, we found that constructing such an optimal preconditioner for the edge-preserving case at every PNCG iteration would be too demanding computationally. Nonetheless, the effectiveness of the proposed preconditioner will be demonstrated experimentally in Section VI.

C. Application of the preconditioner

In order for the proposed preconditioner to be effective, its application at every iteration of the PNCG algorithm, i.e., the computation of $\mathbf{p}^{(i)} = [\mathbf{M}(\tilde{\mathbf{x}}^{(i-1)})]^{-1}\mathbf{r}^{(i)}$ in Table I, must be performed efficiently. For this purpose, solving the following system of equations:

$$\mathbf{M}(\tilde{\mathbf{x}}^{(i-1)})\mathbf{p}^{(i)} = \mathbf{r}^{(i)} \tag{30}$$

is more efficient than calculating the inverse of the preconditioner and applying it to $\mathbf{r}^{(i)}$. Moreover, the structure of $\mathbf{M}(\tilde{\mathbf{x}})$ allows for the efficient computation of the expression above. Since each matrix $\mathbf{M}_{i,j}(\tilde{\mathbf{x}})$ has a BCCB structure, it can be diagonalized by the DFT, and thus, has the following factorization [29]:

$$\mathbf{M}_{i,j}(\tilde{\mathbf{x}}) = \mathbf{F}^{*}\mathbf{\Lambda}_{i,j}(\tilde{\mathbf{x}})\mathbf{F}, \tag{31}$$

where $\mathbf{F} \in \mathbb{C}^{M^2 \times M^2}$ is a matrix representing the application of the 2D DFT to an $M \times M$ signal in vector form and $\mathbf{\Lambda}_{i,j}(\tilde{\mathbf{x}}) \in \mathbb{C}^{M^2 \times M^2}$ is a diagonal matrix. This latter matrix is related to the first column of $\mathbf{M}_{i,j}(\tilde{\mathbf{x}})$ as follows:

$$\mathbf{\Lambda}_{i,j}(\tilde{\mathbf{x}}) \triangleq \mathcal{D}(\mathbf{F}[\mathbf{M}_{i,j}(\tilde{\mathbf{x}})]_{:,1}), \tag{32}$$

where $\mathcal{D}(\cdot)$ and $[\cdot]_{:,i}$ are operators defined such that $\mathcal{D}(\mathbf{a})$ is the diagonal matrix whose diagonal elements are the entries of $\mathbf{a}$ and $[\mathbf{A}]_{:,i}$ is the i^{th} column of $\mathbf{A}$. By combining the matrices $\mathbf{\Lambda}_{i,j}(\tilde{\mathbf{x}})$ in the following manner:

$$\mathbf{\Lambda}(\tilde{\mathbf{x}}) \triangleq \begin{bmatrix} \mathbf{\Lambda}_{1,1}(\tilde{\mathbf{x}}) & \dots & \mathbf{\Lambda}_{1,Q^2}(\tilde{\mathbf{x}}) \\ \vdots & & \vdots \\ \mathbf{\Lambda}_{Q^2,1}(\tilde{\mathbf{x}}) & \dots & \mathbf{\Lambda}_{Q^2,Q^2}(\tilde{\mathbf{x}}) \end{bmatrix}, \tag{33}$$

the preconditioner $\mathbf{M}(\tilde{\mathbf{x}})$ can be expressed as:

$$\mathbf{M}(\tilde{\mathbf{x}}) = \mathbf{F}_B^{*}\mathbf{\Lambda}(\tilde{\mathbf{x}})\mathbf{F}_B, \tag{34}$$

where $\mathbf{F}_B \in \mathbb{C}^{N^2 \times N^2}$ is the block-diagonal matrix formed by repeating the matrix $\mathbf{F}$ Q^2 times along the diagonal, as shown below:

$$\mathbf{F}_B \triangleq \begin{bmatrix} \mathbf{F} & & \\ & \ddots & \\ & & \mathbf{F} \end{bmatrix}. \tag{35}$$

Since $\mathbf{\Lambda}(\tilde{\mathbf{x}})$ is a block matrix with diagonal blocks, it can be permuted into a block-diagonal matrix $\tilde{\mathbf{\Lambda}}(\tilde{\mathbf{x}})$ as follows:

$$\tilde{\mathbf{\Lambda}}(\tilde{\mathbf{x}}) \triangleq \begin{bmatrix} \tilde{\mathbf{\Lambda}}_1(\tilde{\mathbf{x}}) & & \\ & \ddots & \\ & & \tilde{\mathbf{\Lambda}}_{M^2}(\tilde{\mathbf{x}}) \end{bmatrix}, \tag{36}$$

where each block $\tilde{\mathbf{\Lambda}}_i(\tilde{\mathbf{x}}) \in \mathbb{C}^{Q^2 \times Q^2}$ is a dense, Hermitian positive-definite matrix. Consequently, the preconditioner can be applied at every iteration of the PNCG algorithm by solving M^2 independent systems of Q^2 unknowns, in addition to computing the $2Q^2$ two-dimensional FFTs of size $M \times M$ associated with $\mathbf{F}_B$ and its conjugate transpose, which requires $\mathcal{O}(Q^2M^2\log M + M^2Q^4)$ operations. Since each block $\tilde{\mathbf{\Lambda}}_i(\tilde{\mathbf{x}})$ is a Hermitian matrix, it can be solved using the Cholesky factorization, which is roughly twice as efficient as the more general LU decomposition [41].

D. Construction of the preconditioner

Since our preconditioner depends on $\tilde{\mathbf{x}}$, it must be constructed at every PNCG iteration, and thus, the effectiveness of the proposed approach also depends on the speed at which $\mathbf{\Lambda}(\tilde{\mathbf{x}})$ can be updated. Employing Equation (32) to perform this task would require the computation of Q^4 extra FFTs at every PNCG iteration. To avoid these operations, the BCCB structures of $\mathbf{M}_{i,j}^{\{\mathbf{G}\}}$ and $\mathbf{M}_{i,j}^{\{\mathbf{C}\}}$ are exploited to rewrite $\mathbf{\Lambda}_{i,j}(\tilde{\mathbf{x}})$ as

$$\mathbf{\Lambda}_{i,j}(\tilde{\mathbf{x}}) = \mathbf{\Lambda}_{i,j}^{\{\mathbf{G}\}} + \beta(\tilde{\mathbf{x}})\mathbf{\Lambda}_{i,j}^{\{\mathbf{C}\}}, \tag{37}$$

where

$$\mathbf{\Lambda}_{i,j}^{\{\mathbf{G}\}} \triangleq \mathcal{D}(\mathbf{F}[\mathbf{M}_{i,j}^{\{\mathbf{G}\}}]_{:,1}) \tag{38}$$

and

$$\mathbf{\Lambda}_{i,j}^{\{\mathbf{C}\}} \triangleq \mathcal{D}(\mathbf{F}[\mathbf{M}_{i,j}^{\{\mathbf{C}\}}]_{:,1}). \tag{39}$$

The matrices $\mathbf{\Lambda}_{i,j}^{\{\mathbf{G}\}}$ and $\mathbf{\Lambda}_{i,j}^{\{\mathbf{C}\}}$ do not depend on $\tilde{\mathbf{x}}$, and thus, they can be precomputed. Therefore, updating $\mathbf{\Lambda}(\tilde{\mathbf{x}})$ only involves calculating $\beta(\tilde{\mathbf{x}})$ and constructing each block $\mathbf{\Lambda}_{i,j}(\tilde{\mathbf{x}})$ using Equation (37), which costs $\mathcal{O}(Q^4M^2)$ operations.

The computation of $\mathbf{\Lambda}_{i,j}^{\{\mathbf{G}\}}$ must be performed carefully. Perhaps the simplest way of doing so consists of first computing the Fourier diagonal matrices associated with each BCCB matrix $\tilde{\mathbf{G}}_{l,i}$, and then combining these diagonal matrices according to the factorization given by Equation (25). This approach is similar to that employed by Nguyen [44] to compute the Fourier coefficients of a preconditioner proposed for least-squares SR problems (see p.55 in the aforementioned reference). The computation of $\mathbf{\Lambda}_{i,j}^{\{\mathbf{G}\}}$ using this method requires LQ^2 two-dimensional FFTs of size $M \times M$, each of which costs $\mathcal{O}(M^2 \log M)$ operations. While this approach might be acceptable for low values of L, it can render the preconditioner ineffective when many LR images are employed.

For this reason, we recommend constructing the first column of $\mathbf{M}_{i,j}^{\{\mathbf{G}\}}$ directly, and then computing $\mathbf{\Lambda}_{i,j}^{\{\mathbf{G}\}}$ according to Equation (38). Computing this column is not computationally expensive, since the blur kernel associated with each matrix $\tilde{\mathbf{G}}_{l,i}$ has a small support. This approach requires Q^4 two-dimensional FFTs of size $M \times M$, which is better than the previous approach when $L > Q^2$.

Note that when accelerating the related problem of edge-preserving image deblurring, Bedini *et al.* [45] only compute their preconditioner once, namely for the case where $\beta(\tilde{\mathbf{x}}) = 1$, which corresponds to the absence of edges in the image. The authors justify this choice by the fact that natural images generally have a low number of edges. Since their preconditioner is also based on circulant operators, it can easily be adapted to our SR problem. In the following, we shall refer to this adapted approach as Bedini's method.

E. Complexity analysis

To obtain a better appreciation of the efficiency of the proposed method, the application cost of our preconditioner and its memory requirements must be compared to those of the other operations that need to be performed at every PNCG iteration. In particular, our preconditioner involves the storage of the diagonal elements of all matrices $\mathbf{\Lambda}_{i,j}^{\{\mathbf{G}\}}$ and $\mathbf{\Lambda}_{i,j}^{\{\mathbf{C}\}}$ defined in Equations (38) and (39), which yields a memory complexity of $\mathcal{O}(Q^4M^2)$. This is reasonable, since the space required to store the LR images grows as $\mathcal{O}(LM^2)$.

A significant part of the calculation at every iteration is associated with the computation of the gradient (21), which involves matrix-vector products with $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{C}}$. To analyze the computational complexity associated with these multiplications, we recall that each matrix $\mathbf{G}_l$ represents a convolution of the HR image with a blur kernel, followed by a downsampling operation. Furthermore, we assume for simplicity that each such kernel, which models the combined effects of matrices $\mathbf{U}_l$, $\mathbf{B}_l$ and $\mathbf{S}_l$, is represented by an $A \times A$ mask. The number of nonzero elements in $\mathbf{G}_l$ and its permuted version $\tilde{\mathbf{G}}_l$ is then A^2M^2, and thus, multiplying a vector with $\tilde{\mathbf{G}}$ costs $\mathcal{O}(LA^2M^2)$ operations. It is important to note that $\tilde{\mathbf{G}}$ does not need to be constructed explicitly; only the kernels associated with matrices $\{\mathbf{G}_l\}_{l=1}^{L}$ must be stored, which requires $\mathcal{O}(LA^2)$ memory space. Finally, it is easy to verify upon examining the structures of matrices $\{\mathbf{C}_m\}_{m=1}^{4}$ that the number of nonzero elements in each such matrix is $3N^2 = 3Q^2M^2$. Consequently, the computational complexity associated with the gradient step at every PNCG iteration is $\mathcal{O}((LA^2 + Q^2)M^2)$.

A situation of special interest arises when the only source of blur is light integration at the pixel sensors, which happens when all cameras are in focus. In this case, the value of A is roughly equal to the magnification factor Q, and thus, the cost of computing the gradient is $\mathcal{O}(LQ^2M^2)$. From Section V-C, we recall that applying our preconditioner requires $\mathcal{O}(Q^2M^2 \log M + M^2Q^4)$ operations per PNCG iteration. One can see that the computational complexity of the preconditioning step does not depend on the number L of LR images, unlike that associated with the calculation of the gradient. This suggests that preconditioning might be more effective when a higher number of LR images are employed. On the other hand, since the computational complexity with respect to Q of the preconditioning step grows faster than that of the gradient step, one might expect the effectiveness of the preconditioner to decrease for larger magnification factors. These two observations will be validated experimentally in Section VI.

F. Minimization of an augmented cost function

An alternative to minimizing Equation (9) with the PNCG method is to minimize the following *augmented* cost function:

$$\begin{aligned}\tilde{\Phi}(\tilde{\mathbf{x}}, \mathbf{b}) =& \frac{1}{2}\sum_{l=1}^{L} \|\mathbf{y}_l - \tilde{\mathbf{G}}_l\tilde{\mathbf{x}}\|_2^2 + \\ & \lambda \sum_{m=1}^{4} \Big(\|\tilde{\mathbf{C}}_m\tilde{\mathbf{x}} - \mathbf{b}_m\|_2^2 + \sum_{k=1}^{N^2} \psi([\mathbf{b}_m]_k) \Big),\end{aligned} \tag{40}$$

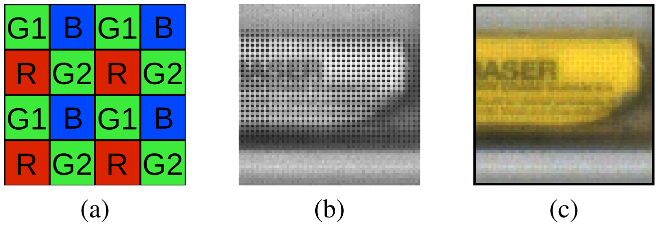
where $\mathbf{b} \triangleq [\mathbf{b}_1^T\mathbf{b}_2^T\mathbf{b}_3^T\mathbf{b}_4^T]^T$, $\mathbf{b}_m \in \mathbb{R}^{N^2}$ is a vector of auxiliary variables and ψ is a dual function constructed from ϕ by using the theory of convex conjugacy [46]. This half-quadratic reformulation, which was introduced by Geman and Yang [47], is said to be of the *additive* form and has the same minimum as Equation (9). In the following, we shall refer to this approach as the half-quadratic additive form (HQAF) method. The minimization of Equation (40) is achieved using an *alternate minimization* scheme [37]; given the iterate $(\tilde{\mathbf{x}}^{(i-1)}, \mathbf{b}^{(i-1)})$ obtained from iteration $i-1$, one computes $\mathbf{b}^{(i)}$ and $\tilde{\mathbf{x}}^{(i)}$ at iteration i such that

$$\tilde{\Phi}(\tilde{\mathbf{x}}^{(i-1)}, \mathbf{b}^{(i)}) \leq \tilde{\Phi}(\tilde{\mathbf{x}}^{(i-1)}, \mathbf{b}) \quad \forall \mathbf{b} \in \mathbb{R}^{4N}, \tag{41}$$

and

$$\tilde{\Phi}(\tilde{\mathbf{x}}^{(i)}, \mathbf{b}^{(i)}) \leq \tilde{\Phi}(\tilde{\mathbf{x}}, \mathbf{b}^{(i)}) \quad \forall \tilde{\mathbf{x}} \in \mathbb{R}^{N^2}. \tag{42}$$

The minimization of $\tilde{\Phi}(\tilde{\mathbf{x}}, \mathbf{b})$ with respect to $\mathbf{b}$ described in Equation (41) is simple, since it can be performed independently for each element of $\mathbf{b}$. Also, since $\tilde{\Phi}(\tilde{\mathbf{x}}, \mathbf{b})$ is a convex and quadratic function with respect to $\tilde{\mathbf{x}}$, its minimizer for any fixed configuration of $\mathbf{b}$ can be computed easily by solving a system of linear equations. Furthermore, since the Hessian matrix with respect to $\tilde{\mathbf{x}}$ of Equation (40) is a block matrix with BCCB blocks, this linear system can be solved efficiently using FFT transforms. However, despite the benefits of such an approach, this yields slower convergence than the PNCG method in many SR scenarios, as will be demonstrated experimentally in Section VI.



(a) (b) (c)

Fig. 5. (a) A typical Bayer mosaic pattern for a small 4×4 area of pixels. Each pixel sensor is sensitive to a single color component. (b) A 72×72 region of a raw image captured by a camera whose image sensor is covered with a Bayer CFA. (c) A 72×72 color image obtained by demosaicing (b) using the VNG method [48].

G. Color super-resolution

Most consumer cameras available today capture color images by employing a single CCD or CMOS image sensor covered with a color filter array (CFA). Consequently, when working with such cameras, the theory presented in the previous sections for grayscale images must be applied to color filtered images. A widely used CFA is the Bayer mosaic pattern, shown in Fig. 5(a) for a 4×4 image sensor. To maintain compatibility with our preconditioner, as explained below, we identify the green channel pixels using two components, even if the associated color filters have similar characteristics in practice. With this camera design, each pixel sensor measures a single color component, i.e., spectral band, resulting in Bayer pattern images similar to that shown in Fig. 5(b). The process by which these raw images are decoded and transformed into full-resolution color images is called *demosaicing*. Although standard interpolation methods (nearest neighbor, linear, cubic spline, etc.) can be used for demosaicing, techniques exploiting the correlation between the color bands generally produce better results. As one example, applying the Variable Number of Gradients (VNG) demosaicing method [48] to Fig. 5(b) yields the color image shown in Fig. 5(c).

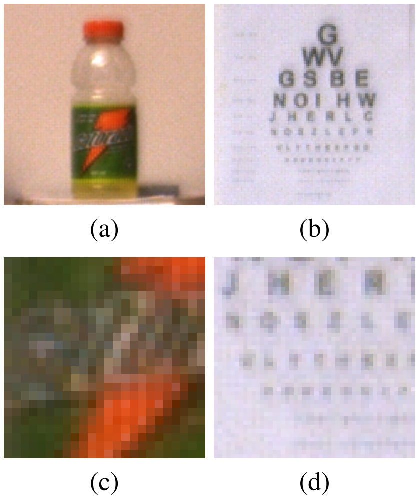
Upon examining the structure of the CFA shown in Fig. 5(a), one can see that a Bayer pattern image is obtained by downsampling each color channel by a factor of two along both image axes, and then interleaving the pixels from these decimated images. The image formation model associated with each color channel is similar to that described in Section III-A for monochrome images. However, the decimation matrix $\mathbf{D}$ now models the combined effect of the downsampling operators associated with the CFA and the desired SR magnification factor. In the remainder of this paper, we shall assume that the magnification factor Q is expressed relative to the size of the decimated color channel images. Under this assumption, demosaicing can be assimilated to a SR problem that employs a magnification factor of two. Indeed, combining the demosaicing and super-resolution tasks is known to yield HR images of better visual quality [22].

The simplest way of performing color SR is to compute the HR color channels independently, and then combine them to obtain a color image [19], [26]. In the particular case where a single Bayer pattern image needs to be demosaiced, this approach usually produces color artifacts, especially in the presence of severe aliasing [49]. One way to avoid this problem is to include in the cost function a color regularization term that penalizes mismatches between edge locations and orientations across the color channels [8], [50]. However, this technique is not optimal when chromatic aberration is significant, and a proper post-processing treatment that takes into account the resulting geometric deformation between the color channels might be preferable [51], [52].

In this work, we assume that the number of Bayer pattern images is sufficient to yield a complete coverage of the scene. That is, all three color components of every real world point are seen through the CFA by at least one camera. Under this condition, we observed that HR images are not significantly affected by color artifacts, and thus, for the sake of simplicity, we can reconstruct the HR color channels independently. To do so, we decompose the unknown HR image $\mathcal{X}$ into four channel images, namely $\mathcal{X}_R$, $\mathcal{X}_{G1}$, $\mathcal{X}_{G2}$ and $\mathcal{X}_B$. Splitting the green channel into two components, as illustrated in Fig. 5(a), is necessary to ensure that the permuted Hessian matrices of the associated cost functions have the structure described in Section IV-A. The final green image $\mathcal{X}_G$ is obtained by averaging $\mathcal{X}_{G1}$ and $\mathcal{X}_{G2}$. Each color channel reconstruction is achieved by minimizing a cost function of the form of Equation (9). The same regularization parameters are employed for both green channels. Henceforth, λ_R, λ_G and λ_B shall refer to the regularization parameters employed in the SR restoration of the red, green and blue channels, respectively. Similarly, α_R, α_G and α_B will denote the different Huber thresholds employed in these problems.

VI. Experiments

In this section, we examine the performance of our acceleration method by considering the problem of simultaneous super-resolution and demosaicing of color-filtered images. For this purpose, we use two sets of Bayer CFA images captured with a Point Grey Flea2 digital camera. During each video acquisition, the camera is slightly rotated manually about its vertical and horizontal axes, which results in small translational motion between the frames. Images from these two datasets are presented in Fig. 6(a) and Fig. 6(b). These images are obtained by demosaicing the original Bayer pattern images using the VNG algorithm.[2] Regions in Fig. 6(a) and Fig. 6(b) are enlarged in Fig. 6(c) and Fig. 6(d), respectively.



(a) (b)

(c) (d)

Fig. 6. Datasets employed in the experiments. The dimensions of the LR images in each dataset are: (a) M = 72 (b) M = 60. Enlarged regions of the images are shown in (c) and (d).

The frames in each set are registered with subpixel accuracy using the method proposed by Karybali *et al.* [28]. Many SR experiments are conducted on each dataset by varying both the number L of LR images used and the regularization parameters λ_R, λ_G and λ_B. The motivation for testing our preconditioner using different regularization weights is that one rarely knows the optimal values for these parameters beforehand. A common way of determining them is to solve the SR problem using different weights and thresholds, and then to choose the values that maximize the visual quality of the HR image. The choice of the regularization parameters can sometimes be automated by finding the "corner" in the Pareto curve,[3] also known as the L-curve, associated with the problem [53], [54]. Although the optimal choice of the regularization parameters is beyond the scope of this paper, it is important to study the convergence behavior of the solution process in situations where suboptimal parameter values are employed, since several restorations under such conditions are often necessary in practice.

In all experiments, the camera is assumed to be in focus, and thus, the PSF employed is a simple $Q \times Q$ square kernel representing the blurring effect of the pixel sensors. For each SR problem, the associated HR image is computed using the NCG, PNCG, HQAF and Bedini's methods. The iterative solvers are stopped when the cost function (9) goes below a predetermined threshold value. To allow for a fair comparison between the three methods above, the same stopping criterion is used with all of them when solving a specific SR problem. To determine this criterion, we first minimize the cost function by running a very large number of NCG, PNCG, HQAF and

[2] as implemented in the libdc1394-22 video capture library for IEEE 1394 devices

[3] The Pareto curve is a log-log plot of the norm of a regularized solution with respect to the norm of the corresponding residual.

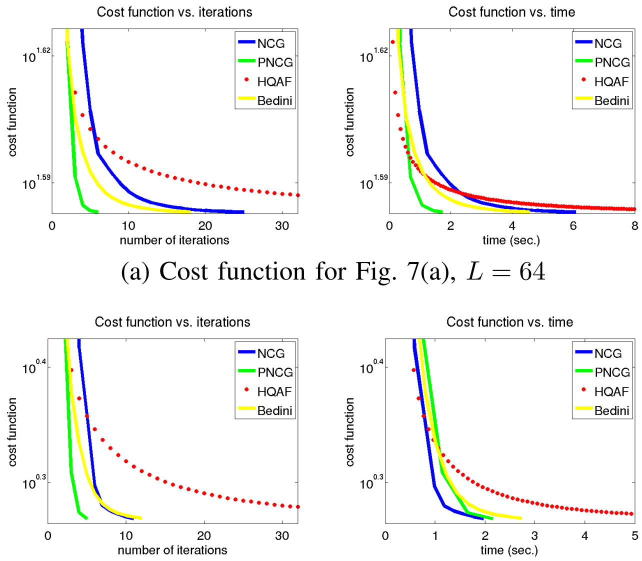
Bedini iterations. We then choose the worst result, i.e., the highest cost function value, as the threshold.

Performance comparisons between the minimization methods for the first and second datasets are presented in Fig. 7 and Fig. 8, respectively. For each method, we indicate the number of iterations required to solve the problem, along with the computation time. Note that for both measurements, each number corresponds to the sum of the values for all color channels.

Upon examining the performance comparison plots in Fig. 7 and Fig. 8, one can see that the proposed preconditioner is effective at reducing the number of iterations with respect to the NCG method in all experiments. In particular, this improvement tends to be more significant for larger values of L. As discussed in Section V-E, this is due to the fact that the computational complexity of the preconditioning step does not increase with L, unlike that associated with the calculation of the gradient. Furthermore, we observe that larger values of λ_R, λ_G and λ_B reduce the usefulness of the PNCG approach over the NCG method. The reason is that increasing the regularization weights improves the condition number of the Hessian matrix, which decreases the need for preconditioning. Finally, one can see that the number of HQAF iterations is higher than the number of PNCG and NCG iterations in all experiments. Our PNCG method also requires fewer iterations to converge than Bedini's method.

When examining the plots of computation time, we observe that the performance improvements achieved by the PNCG method over other methods are generally lower than those indicated by the iteration count. This discrepancy is due to the additional computation requirements from the extra preconditioning step in the PNCG algorithm. We also observe that the HQAF method outperforms the NCG method in some experiments, especially for larger values of L. Although the former requires many more iterations, each iteration has very low computational cost in comparison to those of the NCG method. Indeed, since we use the periodic boundary condition, each HQAF iteration needs only solve a Hermitian block matrix with BCCB blocks, which can be performed efficiently using FFTs and the Cholesky factorization.

For some rational magnification factors, using a preconditioner can result in significantly increased computation time, even if the number of iterations is reduced. Indeed, this is what we observe in five experiments of Fig. 8, where a magnification factor of $\frac{16}{5}$ is employed. This performance loss is due to the fact that the computational complexity with respect to Q of the preconditioning step grows faster than that of the gradient step, as discussed in Section V-E. Consequently, for large magnification factors, we expect the effectiveness of the preconditioner to decrease. This is not a significant issue with integer factors, since practical limits of SR only allow for the use of relatively small factors [7]. However, as explained in Section IV-B, SR problems employing rational magnification factors $\frac{A}{B}$ must be transformed into ones using integer factors $Q = A$ before preconditioning. In this case, the value of A can be large even if the initial rational factor is within the feasible range of SR, and thus, using a preconditioner can actually slow down the computations, especially for low values of L.



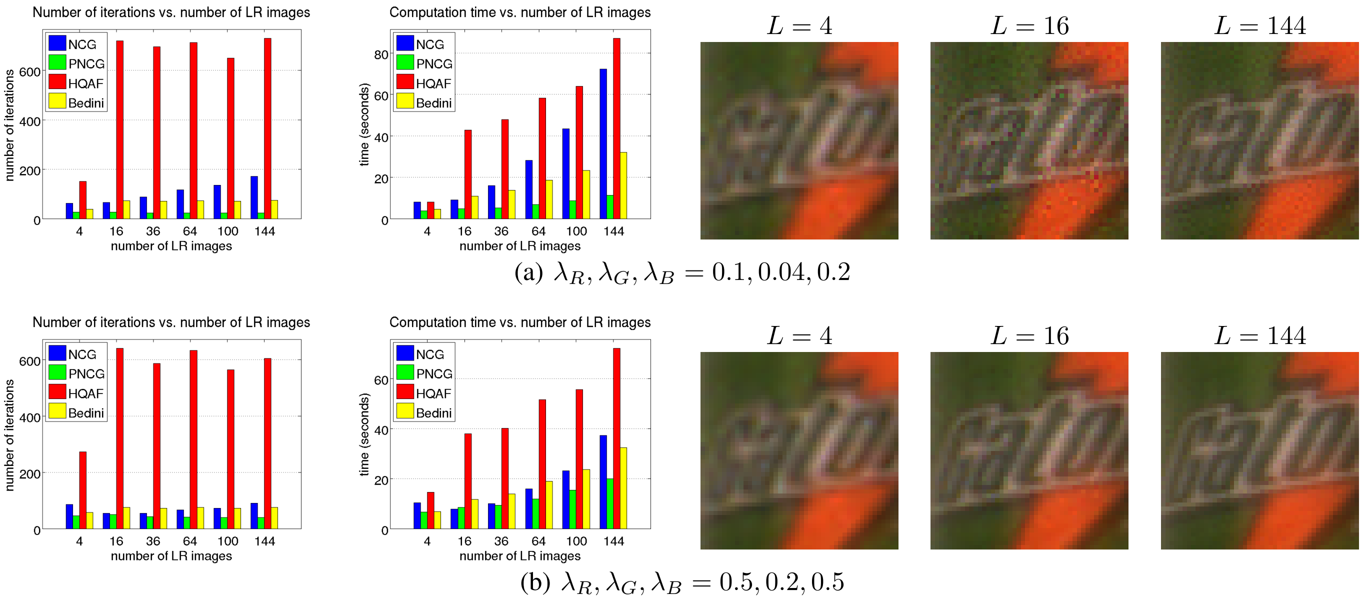
(a) Cost function for Fig. 7(a), $L = 64$

(b) Cost function for Fig. 8(a), $L = 16$

Fig. 9. Performance comparison of the NCG, PNCG, HQAF and Bedini's algorithms when minimizing the cost function (9) associated with the red channels of two SR problems.

To provide a better understanding of the relationship between the visual quality of the HR images and the parameter values, Fig. 7 and Fig. 8 also show a region of interest in the HR images synthesized in some of the experiments. Since each HR image is mathematically defined as the minimum of a given convex cost function, all four minimization methods yield the same solution, and thus, each image for a given experiment is shown only once. Obviously, this holds when the four algorithms converge, which was the case in all experiments. Interestingly, one can observe that HR images restored from four LR images are smoother than those obtained using sixteen LR images. This can be explained by the fact that using only four LR images with a magnification factor of $q = 4$ yields an underdetermined problem, and thus, the regularization prior has a greater smoothing effect on the final image. When the number of input images is increased to sixteen, the SR problem becomes better conditioned. However, since the observed LR images are noisy, inconsistencies are introduced in the observation model, which, in turn, amplifies noise in the recovered HR images. This is particularly visible in Fig. 7(a), where lower values for the regularization parameters are employed. By further increasing the number of LR images, one obtains HR images of superior visual quality. This improvement arises directly from the averaging of a greater amount of observation data. The fact that a greater reduction in computation time is achieved with the PNCG method with larger values of L is thus valuable, since using several LR images increases the visual quality of the HR image with our problem formulation.

In Fig. 9, we examine the convergence curves obtained when employing the PNCG, NCG, HQAF and Bedini's methods to minimize the cost function associated with some of the previous SR problems. Although the proposed approach converges in the smallest number of iterations, a fair comparison of the methods also requires consideration of their computation



(a) $\lambda_R, \lambda_G, \lambda_B = 0.1, 0.04, 0.2$

(b) $\lambda_R, \lambda_G, \lambda_B = 0.5, 0.2, 0.5$

Fig. 7. Performance comparison of the NCG, PNCG, HQAF and Bedini's algorithms for the simultaneous demosaicing and resolution enhancement by a factor $q = 4$ of the dataset shown in Fig. 6(a). Multiple experiments are performed by varying the number L of input LR images and the regularization parameters λ_R, λ_G and λ_B. Enlarged regions of some of the HR images obtained are shown on the right. In all experiments, the following parameter values are employed: $(\alpha_R, \alpha_G, \alpha_B) = (0.001, 0.001, 0.001)$.

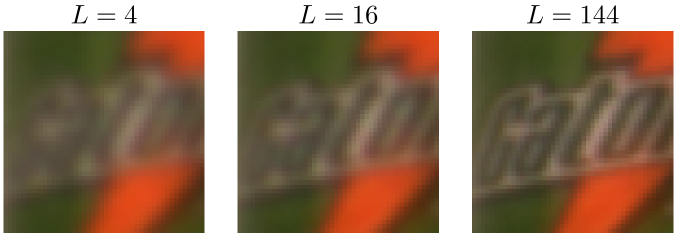


Fig. 10. Simultaneous demosaicing and resolution enhancement by a factor $q = 4$ of the dataset shown in Fig. 6(a) using the method of Tanaka and Okutomi, which employs a quadratic prior.

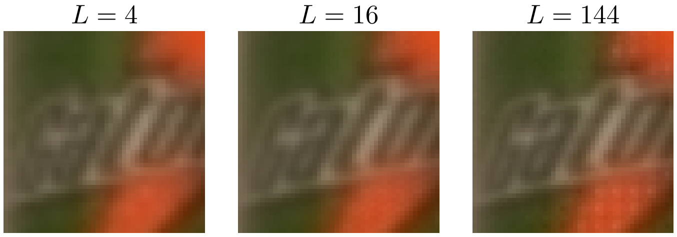


Fig. 11. Simultaneous demosaicing and resolution enhancement by a factor $q = 4$ of the dataset shown in Fig. 6(a) using Vandewalle's method [55].

time per iteration. In this regard, the proposed PNCG method outperforms the three others in the first experiment. However, in the second experiment, which corresponds to a SR problem employing a rational magnification factor, the NCG method converges faster than the PNCG method, as explained above.

We now justify our choice of employing the mathematical formulation of Equation (9) to perform SR by visually comparing our results against those obtained using other SR methods. For this purpose, we employ the image dataset of Fig. 6(a). First, we consider the acceleration method for SR proposed by Tanaka and Okutomi, which relies on a discretization of the displacements between the LR images. The cost function in their method has the same form as Equation (9) but employs a quadratic prior. Fig. 10 shows HR images obtained using our implementation of their method. In this experiment, the displacement values of the LR frames are rounded on the HR grid and the same weights as in Fig. 7(b) are employed, i.e., $\lambda_R, \lambda_G, \lambda_B = 0.5, 0.2, 0.5$. Since their method employs a quadratic prior, the resulting images are blurred when compared to those of Fig. 7(b).

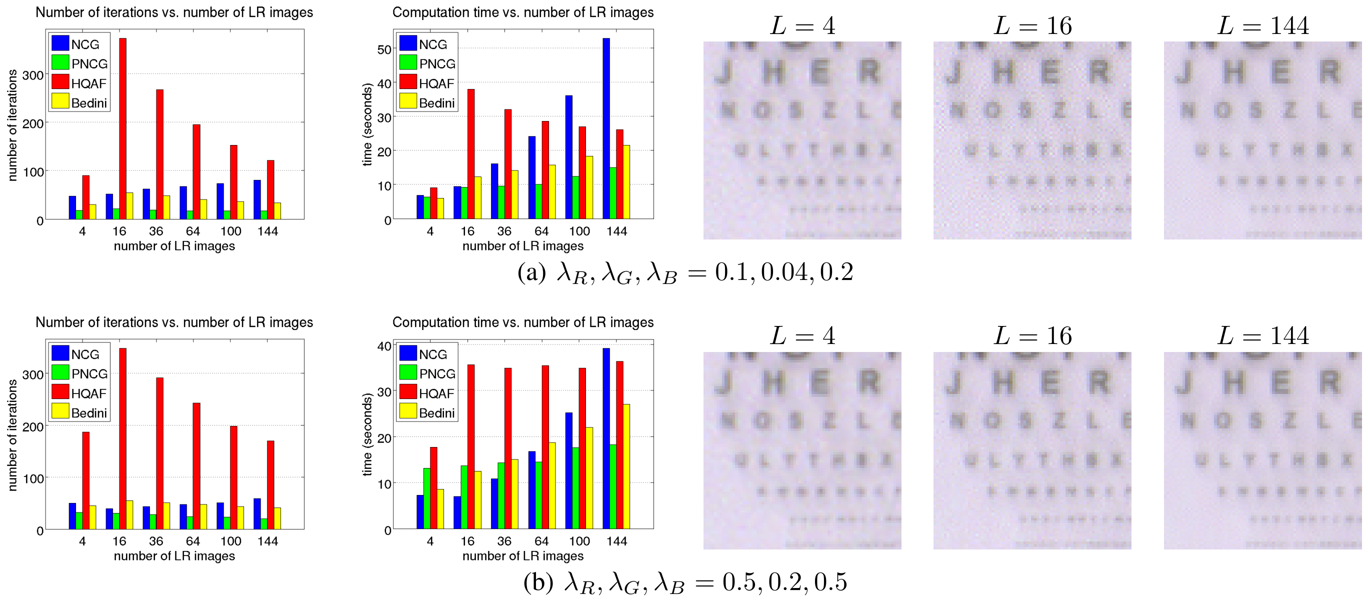
In Fig. 11, we show the HR images obtained by employing the method of Vandewalle *et al.* [55], which performs demosaicing and super-resolution simultaneously. The implementation[4] provided by the authors was employed. Their approach employs normalized convolution to interpolate a nonuniform set of samples obtained from the registered LR images. In these experiments, the demosaicing and SR steps were performed using their implementation. However, we used the same displacement values as in Fig. 7, since the registration algorithm provided by Vandewalle *et al.* yielded very poor results on our dataset. Comparing the images in Fig. 7 and Fig. 11, one can see that our proposed method better preserves sharp edges in the HR images. Also, we observe that image quality seems to deteriorate with Vandewalle *et al.*'s method when a large number of LR images is employed.[5]

As a final comparison, we present HR images in Fig. 12, obtained using the method of Farsiu *et al.* [50], which also employs an edge-preserving regularization term. For this purpose, the implementation[6] provided by the authors

[4] The code is available from http://lcav.epfl.ch/reproducible_research/VandewalleKAS07_1.

[5] However, we note that the code employed for running these experiments was not developed or tested by the authors with such over-determined cases in mind.

[6] The *MDSP Resolution Enhancement Software* is available from http://users.soe.ucsc.edu/~milanfar/software/superresolution.html.



(a) $\lambda_R, \lambda_G, \lambda_B = 0.1, 0.04, 0.2$

(b) $\lambda_R, \lambda_G, \lambda_B = 0.5, 0.2, 0.5$

Fig. 8. Performance comparison of the NCG, PNCG, HQAF and Bedini's algorithms for the simultaneous demosaicing and resolution enhancement by a factor $q = \frac{16}{5}$ of the dataset shown in Fig. 6(b). Multiple experiments are performed by varying the number L of input LR images and the regularization parameters λ_R, λ_G and λ_B. Enlarged regions of some of the HR images obtained are shown on the right. In all experiments, the following parameter values are employed: $(\alpha_R, \alpha_G, \alpha_B) = (0.001, 0.001, 0.001)$.

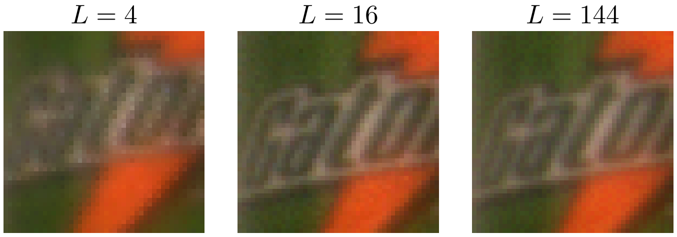


Fig. 12. Simultaneous demosaicing and resolution enhancement by a factor $q = 4$ of the dataset shown in Fig. 6(a) using the edge-preserving method presented by Farsiu [49].

was employed. We used the default values for most of the restoration parameters. However, special care had to be taken in the choice of the step size of the minimization algorithm. In particular, we found that smaller values were necessary to ensure convergence when increasing the number of input images. Note that our method does not suffer from this drawback, since the optimal step size value is computed by the PNCG algorithm at every iteration, as shown in Table I. Although Farsiu *et al.*'s method offers several regularization parameters to control the restoration process, we were unable to achieve the same image quality on this dataset as with the proposed method. This may be due to the fact that their method rounds the displacement values of the LR frames on the HR grid, which introduces errors in the image formation model. It might also be a consequence of the non-convexity of the cost function employed in their method, which causes the solution to converge to a global minima instead of the true solution. Although our proposed problem formulation offers less flexibility in choosing restoration parameters, it is not affected by these issues.

VII. Conclusions

In this paper, a method for accelerating the minimization of cost functions associated with edge-preserving image super-resolution problems was presented. A regularization prior based on the Huber-Markov function was employed to obtain sharp edges in the reconstructed images. When the PSF is spatially invariant, we showed that the proposed preconditioner can be effective for reducing computation time for such problems, even if the Hessian matrix of the associated nonquadratic cost function does not have a Toeplitz-like structure. For this purpose, an efficient method for constructing and updating the preconditioner at every iteration of the PNCG method was described.

Our acceleration method was validated by several experiments. In particular, we studied the convergence behavior of the PNCG solver when suboptimal parameter values are employed. This is important in practice, since one must often perform several restoration processes in order to determine the optimal values for the regularization parameters. We applied the proposed acceleration technique to Bayer pattern images and observed that the speed improvement achieved using the proposed preconditioner tends to be more significant when a larger amount of LR images is employed. This is valuable, since we further observed that using more images reduces the amount of noise and color artifacts in the HR images.

We also noted that increasing the regularization weights reduces the usefulness of the PNCG approach over the NCG method. Furthermore, the effectiveness of the PNCG method decreases for large magnification factors. In particular, we showed that employing a rational factor $\frac{A}{B}$ with a large numerator A reduces the performance of the PNCG method, despite a reduction of the number of iterations. For this reason, one should be careful in the choice of the values of A and B

when using rational factors.

Finally, we compared the visual quality of the HR images synthesized by our method against that achieved by alternative algorithms. Our results demonstrate that the proposed preconditioning approach is useful in practice, since the specific SR problem formulation it is designed to accelerate can yield HR images of superior visual quality than that achieved by other methods.

Acknowledgment

This work was generously supported by a Precarn scholarship.

References

[1] B. Jähne and H. Haußecker, Eds., *Computer Vision and Applications: A Guide for Students and Practitioners*. Orlando, FL, USA: Academic Press, Inc., 2000.

[2] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, May 2003.

[3] S. Chaudhuri, *Super-Resolution Imaging*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[4] N. Nguyen, P. Milanfar, and G. Golub, "Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement," *IEEE Trans. Image Processing*, vol. 10, no. 9, pp. 1299–1308, Sep. 2001.

[5] N. K. Bose, M. K. Ng, and A. C. Yau, "A fast algorithm for image super-resolution from blurred observations," *EURASIP Journal on Applied Signal Processing (special issue on Super-resolution)*, vol. 2006, pp. Article ID 35 726, 14 pages, 2006.

[6] S. Pelletier and J. R. Cooperstock, "Fast super-resolution for rational magnification factors," in *Proc. IEEE International Conf. Image Processing (ICIP)*, vol. 2, San Antonio, TX, Sep. 2007, pp. 65–68.

[7] Z. Lin and H.-Y. Shum, "Fundamental limits of reconstruction-based superresolution algorithms under local translation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 83–97, 2004.

[8] F. Sroubek, J. Flusser, and G. Cristóbal, "Super-resolution and blind deconvolution for rational factors with an application to color images," *Computer Journal*, vol. 52, no. 1, pp. 1–11, 2009.

[9] S. Borman and R. Stevenson, "Spatial resolution enhancement of low resolution image sequences - a comprehensive review with directions for future research," University of Notre Dame, Tech. Rep., 1998.

[10] M. G. Kang and S. Chaudhuri, "Super-resolution image reconstruction (guest editorial)," *IEEE Signal Processing Magazine, special issue on super-resolution*, vol. 20, no. 3, pp. 19–20, May 2003.

[11] M. Ng, T. Chan, M. G. Kang, and P. Milanfar, "Super-resolution imaging: analysis, algorithms, and applications (guest editorial)," *EURASIP Journal on Applied Signal Processing, special issue on super-resolution*, vol. 2006, pp. Article ID 90 531, 2 pages, 2006.

[12] R. C. Hardie, R. R. Schultz, and K. E. Barner, "Super-resolution enhancement of digital video (guest editorial)," *EURASIP Journal on Advances in Signal Processing, special issue on super-resolution enhancement of digital video*, vol. 2007, pp. Article ID 20 984, 3 pages, 2007.

[13] S. Chaudhuri and J. Manjunath, *Motion-Free Super-Resolution*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[14] A. Papoulis, "Generalized sampling theorem," *IEEE Trans. Circuits Syst.*, vol. 24, pp. 652–654, Nov. 1977.

[15] J. Clark, M. Palmer, and P. Lawrence, "A transformation method for the reconstruction of functions from nonuniformly spaced samples," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 33, no. 5, pp. 1151–1165, Oct. 1985.

[16] H. Ur and D. Gross, "Improved resolution from sub-pixel shifted pictures," *CVGIP: Graphical Models and Image Processing*, vol. 54, pp. 181–186, Mar. 1992.

[17] J. C. Gillete, T. M. Stadtmiller, and R. C. Hardie, "Aliasing reduction in staring infrared images utilizing subpixel techniques," *Opt. Eng.*, vol. 34, no. 11, pp. 3130–3137, Nov. 1995.

[18] M. Alam, J. Bognar, R. Hardie, and B. Yasuda, "Infrared image registration and high-resolution reconstruction using multiple translationally shifted aliased video frames," *IEEE Trans. Instrum. and Meas.*, vol. 49, pp. 915–923, 2000.

[19] N. Shah and A. Zakhor, "Resolution enhancement of color video sequences," *IEEE Trans. Image Processing*, vol. 8, no. 6, pp. 879 – 885, Jun. 1999.

[20] M. Elad and Y. Hel-Or, "Fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur," *IEEE Trans. Image Processing*, vol. 10, no. 8, pp. 1187–1193, Aug. 2001.

[21] J. Barron, D. J. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, pp. 43–77, 1994.

[22] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Trans. Image Processing*, vol. 13, no. 10, pp. 1327–1344, Oct. 2004.

[23] M. Tanaka and M. Okutomi, "A fast algorithm for reconstruction-based superresolution and evaluation of its accuracy," *Systems and Computers in Japan*, vol. 38, no. 7, pp. 44–52, 2007.

[24] R. H. Chan, T. F. Chan, M. K. Ng, W. C. Tang, and C. K. Wong, "Preconditioned iterative methods for high-resolution image reconstruction with multisensors," *Proceedings to the SPIE Symposium on Advanced Signal Processing: Algorithms, Architectures, and Implementations*, vol. 3461, pp. 348–357, Jul. 1998.

[25] M. K. Ng, R. H. Chan, T. F. Chan, and A. M. Yip, "Cosine transform preconditioners for high resolution image reconstruction," *Linear Algebra and its Applications*, vol. 316, no. 1–3, pp. 89–104, 2000.

[26] M. K. Ng, "An efficient parallel algorithm for high resolution color image reconstruction," in *ICPADS '00: Proceedings of the Seventh International Conference on Parallel and Distributed Systems: Workshops*. Washington, DC, USA: IEEE Computer Society, 2000, pp. 547–552.

[27] F.-R. Lin, W.-K. Ching, and M. K. Ng, "Preconditioning regularized least squares problems arising from high-resolution image reconstruction from low-resolution frames," *Linear Algebra and Its Applications*, vol. 391, pp. 149–168, 2004.

[28] I. G. Karybali, E. Z. Psarakis, K. Berberidis, and G. D. Evangelidis, "An efficient spatial domain technique for subpixel image registration," *Signal Processing: Image Communication*, vol. 23, no. 9, pp. 711–724, 2008.

[29] K. Chen, *Matrix Preconditioning Techniques and Applications*, ser. Cambridge Monographs on Applied and Computational Mathematics. Cambridge, UK: Cambridge Univ. Press, 2005, vol. 19.

[30] R. R. Schultz and R. L. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Trans. Image Processing*, vol. 5, no. 6, pp. 996–1011, Jun. 1996.

[31] J. Abbiss and P. Earwicker, "Compact operator equations, regularization and super-resolution," in *Mathematics in Signal Processing*. Oxford: Clarendon Press, 1987.

[32] L. Eldén, "An algorithm for the regularization of ill-conditioned, banded least squares problems," *SIAM J. Sci. Statist. Comput.*, vol. 5, no. 1, pp. 237–254, Mar. 1984.

[33] N. Nguyen, P. Milanfar, and G. H. Golub, "A computationally efficient superresolution image reconstruction algorithm." *IEEE Trans. Image Processing*, vol. 10, no. 4, pp. 573–583, Apr. 2001.

[34] R. R. Schultz and R. L. Stevenson, "A bayesian approach to image expansion for improved definition," *IEEE Trans. Image Processing*, vol. 3, no. 3, pp. 233–242, May 1994.

[35] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[36] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep., 1994.

[37] M. Nikolova and M. K. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM J. Sci. Comput.*, vol. 27, no. 3, pp. 937–966, 2005.

[38] S. Pelletier and J. R. Cooperstock, "Efficient image restoration with the Huber-Markov prior model." in *Proc. IEEE International Conf. Image Processing (ICIP)*, San Diego, CA, Oct. 2008.

[39] G. Zelniker and F. J. Taylor, *Advanced Digital Signal Processing: Theory and Applications*. New York, NY: Marcel Dekker, Inc., 1993.

[40] R. Pan and S. J. Reeves, "Efficient Huber-Markov edge-preserving image restoration," *IEEE Trans. Image Processing*, vol. 15, no. 12, pp. 3728–3735, Dec. 2006.

[41] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge, UK: Cambridge Univ. Press, 1992.

[42] J. A. Fessler and S. D. Booth, "Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction," *IEEE Trans. Image Processing*, vol. 8, no. 5, pp. 688–699, May 1999.

[43] T. F. Chan and J. A. Olkin, "Circulant preconditioners for Toeplitz-block matrices," *Numerical Algorithms*, vol. 6, no. 1, pp. 89–101, 1994.

[44] N. Nguyen, “Numerical algorithms for image superresolution,” Ph.D. dissertation, Stanford University, Jul. 2000.

[45] L. Bedini, G. M. D. Corso, and A. Tonazzini, “Preconditioned edge-preserving image deblurring and denoising,” *Pattern Recogn. Lett.*, vol. 22, no. 10, pp. 1083–1101, Aug. 2001.

[46] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*. Springer Verlag, Heidelberg, 1996, two volumes - 2nd printing.

[47] D. Geman and C. Yang, “Nonlinear image recovery with half-quadratic regularization,” *IEEE Trans. Image Processing*, vol. 4, no. 7, pp. 932–946, Jul. 1995.

[48] E. Chang, S. Cheung, and D. Y. Pan, “Color filter array recovery using a threshold-based variable number of gradients,” *Proc. SPIE: Sensors, Cameras, and Applications for Digital Photography*, vol. 3650, pp. 36–43, Mar. 1999.

[49] S. Farsiu, “A fast and robust framework for image fusion and enhancement,” Ph.D. dissertation, University of California, Santa Cruz, Dec. 2005.

[50] S. Farsiu, M. Elad, and P. Milanfar, “Multiframe demosaicing and super-resolution of color images,” *IEEE Trans. Image Processing*, vol. 15, no. 1, pp. 141–159, Jan. 2006.

[51] T. E. Boult and G. Wolberg, “Correcting chromatic aberrations using image warping,” in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 1992, pp. 684–687.

[52] J. Mallon and P. F. Whelan, “Calibration and removal of lateral chromatic aberration in images,” *Pattern Recogn. Lett.*, vol. 28, no. 1, pp. 125–135, 2007.

[53] N. Bose, S. Lertrattanapanich, and J. Koo, “Advances in superresolution using L-curve,” in *Proc. Int. Symp. Circuits and Systems*, vol. 2, May 2001, pp. 433–436.

[54] G. Hennenfent, E. van den Berg, M. P. Friedlander, and F. Herrmann, “New insights into one-norm solvers from the Pareto curve,” *Geophysics*, vol. 73, no. 4, pp. A23–A26, 2008.

[55] P. Vandewalle, K. Krichane, D. Alleysson, and S. Süsstrunk, “Joint demosaicing and super-resolution imaging from a set of unregistered aliased images,” in *Proc. IS&T/SPIE Electronic Imaging: Digital Photography III*, vol. 6502, 2007.

Stéphane Pelletier received a Ph.D. from the Department of Electrical and Computer Engineering at McGill University in 2010. Since then, he has been involved in the development of a real-time view interpolation software for training students at the McGill Medical Simulation Centre. He also collaborated with a company in the development of an autostereoscopic display. His work interests are in the area of image processing and photorealistic 3D model reconstruction from images.

Jeremy R. Cooperstock directs McGill University’s Shared Reality Lab, which focuses on computer mediation to facilitate high-fidelity human communication and the synthesis of perceptually engaging, multimodal, immersive environments. His accomplishments include the world’s first Internet streaming demonstrations of Dolby Digital 5.1, uncompressed 12-channel 96kHz/24bit, multiple simultaneous streams of uncompressed high-definition video, and a simulation environment that renders graphic, audio, and vibrotactile effects in response to footsteps. His work has been recognized by an award for Most Innovative Use of New Technology from ACM/IEEE Supercomputing and a Distinction Award from the Audio Engineering Society.