Scattered data approximation by regular grid weighted smoothing

BIBIN FRANCIS, SANJAY VISWANATH and MUTHUVEL ARIGOVINDAN*

Imaging Systems Laboratory (ISL), Department of Electrical Engineering, Indian Institute of Science,
Bangalore 560012, India
e-mail: francisbiningeorge@gmail.com; sanjayviswanath89@gmail.com; mvel@ee.iisc.ernet.in

MS received 26 May 2016; revised 29 June 2017; accepted 22 July 2017; published online 7 February 2018

Abstract. Scattered data approximation refers to the computation of a multi-dimensional function from measurements obtained from scattered spatial locations. For this problem, the class of methods that adopt a roughness minimization are the best performing ones. These methods are called variational methods and they are capable of handling contrasting levels of sample density. These methods express the required solution as a continuous model containing a weighted sum of thin-plate spline or radial basis functions with centres aligned to the measurement locations, and the weights are specified by a linear system of equations. The main hurdle in this type of method is that the linear system is ill-conditioned. Further, getting the weights that are parameters of the continuous model representing the solution is only a part of the effort. Getting a regular grid image requires re-sampling of the continuous model, which is typically expensive. We develop a computationally efficient and numerically stable method based on roughness minimization. The method leads to an algorithm that uses standard regular grid array operations only, which makes it attractive for parallelization. We demonstrate experimentally that we get these computational advantages only with a little compromise in performance when compared with thin-plate spline methods.

Keywords. Scattered data approximation; variational reconstruction; conjugate gradient algorithm; multi-resolution approach; thin-plate splines; radial basis functions.

1. Introduction

Scattered data approximation (SDA) refers to the problem of reconstructing a standard digital image from spatially scattered measurements and occurs frequently in many areas, including medical imaging, biological imaging, astronomy, geology, object modelling and so on. Acquisition of samples at regular intervals is hindered by obstacles in the medium, limitations of the imaging equipment or the cost of acquiring the samples on a regular grid, thus leading to acquisition of scattered or non-uniform samples. Examples of work involving non-uniform sampling and reconstruction include image reconstruction in astronomy [1], ultrasound imaging [2], super-resolution imaging [3], geophysics [4], surface reconstruction [5], seismic imaging [6] and communication systems [7]. In ultrasound imaging [2], the acquisition device samples the signal uniformly in polar coordinates, which are then mapped to Cartesian coordinates, resulting in irregular sampling on the Cartesian grid. On the other hand, in seismic imaging [6], it is the cost of deploying sensors and receivers at regular intervals over a large geographical area that limits uniform data acquisition.

All SDA methods approach the problem by estimating a continuously defined function that passes through the measured points. Most of these methods fall into either of two categories, namely *local* and *global* methods. Given a set of measurements $f_1, f_2, \dots, f_{N_s}$ at arbitrarily spaced locations $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_s}$ with $\mathbf{x}_i \in \mathbb{R}^d$ where d is the dimension, *global* methods approach the reconstruction problem by estimating a continuously defined function $S(\mathbf{x})$ satisfying $S(\mathbf{x}_i) = f_i$, for $i = 1, \dots, N_s$, and then re-sample $S(\mathbf{x})$ in regular d -dimensional grid to get d -dimensional array, which becomes the required digital image. Specifically, if $\hat{S}$ is the required d -dimensional image array, then its value at $\mathbf{j}^{\text{th}}$ pixel index $\hat{S}(\mathbf{j})$ is computed as $\hat{S}(\mathbf{j}) = S(\mathbf{j}\delta)$, where $\mathbf{j}$ is the d -dimensional pixel index and δ is the re-sampling step size. Local methods work in the following way: at first, for each grid point index $\mathbf{j}$, a function $S^{(\mathbf{j})}(\mathbf{x})$ is constructed satisfying $S^{(\mathbf{j})}(\mathbf{x}_i) = f_i$ for only measurements that are in a neighbourhood of $\mathbf{j}\delta$; then the required pixel value $\hat{S}(\mathbf{j})$ is computed as $\hat{S}(\mathbf{j}) = S^{(\mathbf{j})}(\mathbf{j}\delta)$.

In figure 1, a toy 2D reconstruction grid of size 6×6 is shown along with the scattered non-grid measurement locations. The intersections of vertical and horizontal lines in figure 1 denote the regular grid locations, whose pixel values are to be evaluated from the scattered measurements,

*For correspondence

Figure 1. Figure showing the non-grid measurements and the corresponding reconstruction grid.

whose locations are denoted by circles. The spacing between two regular grid locations, defined as the re-sampling step size, is denoted by δ . The SDA problem is to estimate the pixel values at regular grid locations from the available set of scattered measurements.

1.1 Local approximation methods

These include local polynomial fitting [8, 9], kriging [10] and mesh-based methods [11].

In local polynomial fitting [8, 9], around each regular grid location, a window of pre-defined size is used to select the measurement samples for computing the coefficients of a polynomial. The size of the window is a trade-off between computational complexity and the accuracy of reconstruction. Since the fitting process needs to be repeated for every regular grid location, this method is not computationally efficient. Further, the accuracy of reconstruction degrades rapidly with lower sampling densities or larger sampling gaps [12].

Mesh-based methods [11, 13] start with triangulation of the data set followed by fitting a local polynomial to the triangulated region using the available measurements and partial derivative information. However, accurate evaluation of partial derivatives is hindered by the irregular distribution of the measurements. In addition, the performance of these methods degrades in the presence of measurement noise or when the sampling densities are low.

In kriging [10, 14], a variogram is fitted to the measurements. The pixel values at regular grid locations are then estimated as a weighted linear combination of the

measurements inside a suitable window. The weights are estimated so as to minimize the variance. This method is robust to measurement noise. However, the accuracy of the reconstruction depends heavily on the variogram. Also, its computational complexity is higher than those of the local polynomial fitting and mesh-based methods.

All of these methods lack a systematic way to handle the variations in sampling density. One way to handle it is to set the window size to match the lowest sample density. However, this approach fails to capture the fine variations in the measurements at locations of high sample density. Another way is to make the window size adaptive. However, there are no prior works that offer a systematic way of doing this in a computationally affordable way.

1.2 Global approximation methods

They include inverse distance weighted method [15], reconstruction in shift invariant spaces [16] and variational methods [17]. Inverse distance weighted method [15] estimates the function value at each regular grid location as a weighted linear combination of all the measurements. The inverse of the Euclidean distance between the measurement location and the regular grid location is used as the weight function. Since this method does not make any assumption on the sample density, it can handle variations in sampling density. However, the reconstructed function may not be continuous at the data points. Further, it is not robust to noise [18].

Reconstruction in shift-invariant spaces [16] is carried out based on the assumption that the underlying function is continuous and belongs to the wavelet scaling function space, which is also known as shift-invariant space [19]. A function in this space can be expressed in the form $\sum_{\mathbf{k} \in \mathcal{N}^d} c_{\mathbf{k}} \phi(\mathbf{x}/h - \mathbf{k})$, where $\phi(\cdot)$ is the generating function, h is the step size and $\mathcal{N}^d$ is the set of d -dimensional integer vectors. In order to ensure that $c_{\mathbf{k}}$ is uniquely determined, the largest sampling gap should be within a limit determined by h . Alternatively, for the given sample set, h should be larger than the limit guided by the largest sampling gap. Hence, this method fails to capture signal variations in the regions where the sample density is higher.

1.2a Variational methods: In variational methods, the reconstruction problem is posed in a constrained optimization framework. Here, the roughness of the reconstructed function is minimized while ensuring that the reconstructed function passes exactly through the measurements, which we shall refer to as the *interpolation constraints*. The roughness of the reconstructed function is measured in terms of derivative norms, which are evaluated using derivative operators.

When second-derivative operators are used to compute the derivative norm, the variational reconstruction problem can be posed as [17]

$$\min \int \int \left[\left(\frac{\partial^2 S(\mathbf{x})}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 S(\mathbf{x})}{\partial xy} \right)^2 + \left(\frac{\partial^2 S(\mathbf{x})}{\partial y^2} \right)^2 \right] dx dy \quad (1)$$

$$\text{such that } S(\mathbf{x}_i) = f_i, \quad \text{for } i = 1, 2, \dots, N_s, \quad (2)$$

where f_i is the measurement at location $\mathbf{x}_i = (x_i, y_i)$ and $S(\mathbf{x})$ is the continuous function to be determined. The constraint (Eq. (2)) mandates the function value at the measurement locations to be equal to the measurement values.

The general solution to this optimization problem is given by

$$S(\mathbf{x}) = \sum_{i=1}^{N_s} w_i \Phi(\|\mathbf{x} - \mathbf{x}_i\|) + a_0 + a_1 x + a_2 y. \quad (3)$$

Here, $\Phi(\cdot)$ is the thin-plate spline and is given by

$$\Phi(r) = r^2 \log(r), \quad r > 0. \quad (4)$$

Notice that the reconstructed function $S(\mathbf{x})$ involves the sum of a weighted linear combination of thin-plate splines positioned at the sampling locations $\{\mathbf{x}_i\}$ and a polynomial of degree one. The reconstruction problem is now to evaluate the weights $\{w_i\}$ and the coefficients of the polynomial a_0 , a_1 and a_2 . They are chosen to ensure that Eq. (2) is satisfied.

By virtue of its formulation, the variational method can handle the variations in sampling density efficiently. Large sample gaps are filled smoothly by utilizing the information from the neighbouring samples by virtue of roughness minimization. At the same time, fine function variations in areas of high sampling density are captured.

A modification of the variational method formulates the reconstruction problem as an unconstrained optimization problem [20]. Here, the interpolation constraints are replaced by a fidelity term measuring the closeness of fit of the reconstructed function to the data. This fidelity term is augmented by the derivative norm with a weighing factor λ . The resulting cost function is

$$\sum_{i=1}^{N_s} |S(\mathbf{x}_i) - f_i|^2 + \lambda \int \int \left[\left(\frac{\partial^2 S(\mathbf{x})}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 S(\mathbf{x})}{\partial xy} \right)^2 + \left(\frac{\partial^2 S(\mathbf{x})}{\partial y^2} \right)^2 \right] dx dy. \quad (5)$$

Similar to the variational method, this method can handle sampling density variations efficiently. Moreover, it can handle noisy measurements, as it replaces the interpolation constraints.

The main drawback of these constrained and unconstrained formulations of the variational reconstruction problem is their computational complexity. This is due to the re-sampling that

is required for evaluating the function values at regular grid locations. Further, the weights are evaluated by inverting a large dense matrix. The computational complexity of inverting this matrix is $O(N_s^3)$. Also, this matrix tends to be ill-conditioned and numerically breaks down for a large number of samples [21]. To reduce the computational cost, the weights are evaluated iteratively in [22]. Further, its speed of convergence is improved using a pre-conditioner in [23]. However, they do not fully address the numerical instability associated with the variational method.

To address the numerical instability issue, the thin-plate splines in (3) are replaced by functions whose value depends on the radial distance from the centre. Such functions are known as the radial basis functions (RBF) [24, 25] and they lead to a linear system of equations that is well-conditioned. The commonly used RBF are multi-quadrics, Gaussian, inverse quadrics, inverse multi-quadrics and so on [26]. Unlike the thin-plate spline, RBF guarantee stable and unique reconstruction when the measurement locations are distinct [20]. However, similar to the variational method, computationally intensive steps such as inversion of a large dense matrix and re-sampling to obtain the final discrete image are required in the RBF approach. Further, the accuracy of reconstruction depends heavily on the parameters of the RBF and choosing them appropriately is a challenging task [27, 28].

To reduce the computational complexity of the RBF approach, partition of unity (PoU) approach was proposed in [29]. It proceeds by partitioning the set of measurement locations into overlapping sub-domains. A local reconstruction is obtained for each of these sub-domains using the RBF approach, which are then stitched together to get the global reconstructed function. While this method reduces the computational complexity compared with the RBF approach, its reconstruction accuracy is also reduced. Recent works in this direction involve employing compact RBF [30] and improving the procedure for partitioning set of measurement locations [31]. However, they do not address the loss in reconstruction accuracy due to partitioning.

Subspace variational method circumvents the numerical problems encountered in variational methods by combining the variational method with the reconstruction in shift-invariant subspace method. In this method, the reconstructed function is restricted to a weighted linear combination of shifted and scaled B-splines located at the regular grid points [2, 21, 32]. The reconstruction problem is now to evaluate the weights associated with the B-splines. They are chosen to minimize the unconstrained variational cost function in Eq. (5). The weights are evaluated by inverting a sparse system of linear equations, which is well-conditioned. Further, the values of the reconstructed function at the regular grid locations can be obtained from a simple convolution of the weights with the B-splines, resulting in significant reduction in computational complexity.

However, this method requires storing an $N \times N$ sparse matrix, where N is the number of regular grid locations. For example, if the reconstruction grid size is 512×512 , then an array of size $2^{18} \times 2^{18}$ needs to be stored. This can be prohibitively high, particularly for large reconstruction grids.

1.3 Contributions

In this paper, we propose a novel method for SDA with reduced computational complexity and memory requirements. We first transform the given non-uniform data into another form of non-uniform data containing the following: a set of sample locations constrained to be a subset of the reconstruction grid (albeit non-uniform) and corresponding re-estimated sample values, and a set of weights quantifying the inaccuracy caused by re-estimation procedure. Then, an unconstrained cost function similar to that in (5) is defined in terms of the transformed non-uniform data, where the variable of the cost is the required regular grid discrete image. Since the cost is formulated directly on the regular grid discrete image, the computational problem becomes well conditioned. Further, because of the specific form of the transformed data set, the need for storing large matrices is also avoided. The computation of the minimum of the cost function is achieved using iterations that require standard digital filtering and array multiplications only with the overall complexity of order $O(N)$ where N is the grid size. Compared with other representative methods, we achieve a significant reduction in the complexity without significantly compromising the quality of reconstruction as shown in the simulation results.

2. The proposed method

2.1 Transforming the scattered data

Let $\{\mathbf{x}_i\}_{i=1}^{N_s}$ be the given non-grid measurement locations together with the corresponding function values $\{f_i\}_{i=1}^{N_s}$. The goal here is to transform this data set such that the transformed data set allows matrix-free reconstruction. The transformation procedure will re-estimate the sample values for a new set of non-uniform locations, where the new locations become a subset of the reconstruction grid (albeit non-uniform). The transformation procedure will also compute some weights that quantify the error incurred by this re-estimation, which will be used in the regularized reconstruction.

Let the 2D reconstruction grid contain N grid points. For a given reconstruction grid point $\mathbf{r}$, let $\mathcal{I}_{\mathbf{r}}$ be a set of indices such that for each $j \in \mathcal{I}_{\mathbf{r}}$, the corresponding sample $\mathbf{x}_j$ is within a unit rectangular neighbourhood centred around the grid point $\mathbf{r}$. Now define

$$\tilde{f}(\mathbf{r}) = \frac{\sum_{j \in \mathcal{I}_{\mathbf{r}}} w_j f_j}{\sum_{j \in \mathcal{I}_{\mathbf{r}}} w_j}, \quad (6)$$

where the weight w_j is given by $w_j = \phi(\|\mathbf{r} - \mathbf{x}_j\|^2)$ with ϕ being an appropriate monotonically decreasing function to be specified later. For a given grid point $\mathbf{r}$, the quantity $\tilde{f}(\mathbf{r})$ is the distance-weighted interpolation from points that lie within the unit neighbourhood around $\mathbf{r}$.

We now propose to designate $\tilde{f}(\mathbf{r})$ as the derived measurement at location $\mathbf{r}$ and use this for variational reconstruction, instead of the original measurements. However, since $\tilde{f}(\mathbf{r})$ is not an actual measurement, we propose using $C(\mathbf{r}) = \sum_{\mathcal{I}_{\mathbf{r}}} w_j$ as a score for $\tilde{f}(\mathbf{r})$, which can serve as a measure of the confidence level. For any grid point $\mathbf{r}'$ for which there are no original measurement points, we assign $\tilde{f}(\mathbf{r}') = 0$ and set the corresponding confidence measure to zero, i.e., we set $C(\mathbf{r}') = 0$. After evaluating $\tilde{f}$ and C for all grid points, the regular grid image pair $\{\tilde{f}(\mathbf{r}), C(\mathbf{r})\}$ will have the information present in given nonuniform data $\{\mathbf{x}_i, f_i\}_{i=1}^{N_s}$ embedded in it. In other words, we have proposed a mechanism to represent the given nonuniform data in terms of a regular grid image pair. Figure 2 illustrates this described scheme using a toy example. As compared with figure 1, this figure has additional solid circles on the grid locations that obtain derived measurements (non-zero $C(\mathbf{r}')$) from the original measurements (denoted by hollow circles).

It should be emphasized that the pair $\{\tilde{f}(\mathbf{r}), C(\mathbf{r})\}$ is still a non-uniform sample set: a non-zero value of $C(\mathbf{r})$ denotes the availability of re-estimated samples values for those locations, and a zero value denotes unavailability. The only differences is that the samples values have been re-estimated for a new set of non-uniform locations that constitute a subset of the reconstruction grid. The non-zero values of $C(\mathbf{r})$ quantify the error involved in this re-estimation. Since the non-uniform locations become a subset of the reconstruction grid, it is possible to represent the reconstruction equation without any matrix. This will become evident from the next subsection.

Figure 2. Showing the non-grid measurements and the reconstruction grid with mapped scattered measurements.

Now, it remains to specify the weight function $\phi(r)$. We propose to use the following function:

$$\phi(r) = \frac{\tanh(\alpha r)}{\alpha r}. \quad (7)$$

An interesting property of this function is that it behaves like $1/r$ for large values of r and $\phi(r) \rightarrow 1$ as $r \rightarrow 0$. Further, for any value of r except zero, the closeness of $\phi(r)$ to $1/r$ is improved by larger values of α . We observed experimentally that this function gives better results compared with using $1/r$ directly, as done in the inverse weighted interpolation methods [15]. It is now clear why $C(\mathbf{r}) = \sum_{j \in \mathcal{I}_r} \phi(\|\mathbf{r} - \mathbf{x}_j\|)$ can be considered as a confidence measure for $\tilde{f}(\mathbf{r})$: since $C(\mathbf{r})$ measures both closeness and number of original measurements used to get $\tilde{f}(\mathbf{r})$, it can serve as the confidence measure.

In actual implementation, it is more efficient to start with zero-initialized arrays for $\tilde{f}(\mathbf{r})$ and $C(\mathbf{r})$, and then scan through each given point in $\{\mathbf{x}_i, f_i\}_{i=1}^{N_s}$ and make the corresponding updates in the arrays $\tilde{f}(\mathbf{r})$ and $C(\mathbf{r})$. Such an algorithm is given in **Algorithm 1**.

Input:

N_s : Number of measurements.

Scattered measurement locations (non-integer): $\{\mathbf{x}_i\}_{i=1}^{N_s}$.

Measurements: $\{f_i\}_{i=1}^{N_s}$.

α : parameter in the weighing function.

Output:

$\tilde{f}(\mathbf{r})$: $N \times N$ regular grid image estimated from the measurements.

$C(\mathbf{r})$: $N \times N$ weight image indicating the confidence in the estimated pixel values of $\tilde{f}(\mathbf{r})$.

$\mathbf{r}$ denotes array index (integer vector) representing grid points.

$C(\mathbf{r})$ and $\tilde{f}(\mathbf{r})$ are initialized as $N \times N$ images with all pixel values set to zero.

for $i = 1, \dots, N_s$ **do**

if $\mathbf{x}_i$ is an integer vector **then**

$C(\mathbf{x}_i) = C(\mathbf{x}_i) + 1$

$\tilde{f}(\mathbf{x}_i) = \tilde{f}(\mathbf{x}_i) + f_i$

else

 Determine the four nearest regular grid locations of $\mathbf{x}_i$ denoted by $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4\}$

for $j = 1, \dots, 4$ **do**

$r = \|\mathbf{r}_j - \mathbf{x}_i\|^2$

$w = \frac{\tanh(\alpha r)}{\alpha r}$

$\tilde{f}(\mathbf{r}_j) \leftarrow \tilde{f}(\mathbf{r}_j) + w f_i$

$C(\mathbf{r}_j) \leftarrow C(\mathbf{r}_j) + w$

end

end

end

Divide $\tilde{f}(\mathbf{r})$ by $C(\mathbf{r})$ whenever $C(\mathbf{r})$ is nonzero.

Algorithm 1: Estimating the regular grid image $C(\mathbf{r})$ and $\tilde{f}(\mathbf{r})$ from the measurements.

2.2 Reconstruction by roughness minimization

From the transferred measurement image $\tilde{f}(\mathbf{r})$ and the associated weight image $C(\mathbf{r})$, we propose to obtain the required image $u(\mathbf{r})$ as a minimum of the following cost:

$$J(u) = \sum_{\mathbf{r}} C(\mathbf{r}) (\tilde{f}(\mathbf{r}) - u(\mathbf{r}))^2 + \lambda \sum_{\mathbf{r}} \left((d_{xx} * u(\mathbf{r}))^2 + 2(d_{xy} * u(\mathbf{r}))^2 + (d_{yy} * u(\mathbf{r}))^2 \right). \quad (8)$$

Note that $C(\mathbf{r})$ indicates the confidence in the estimate of the sample value at pixel location $\mathbf{r}$. Here, $\{d_{xx}, d_{yy}, d_{xy}\}$ is the set of discrete masks (3×3 discrete images) implementing the derivative operators $\{\frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2}, \frac{\partial^2}{\partial x \partial y}\}$ and $*$ denotes convolution. They are given by

$$d_{xx} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix}, d_{yy} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad (9)$$

$$d_{xy} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

We call the first part as the weighted data fitting term. The equivalence of this term to the first part of Eq. (5) is clear in two ways. The non-uniformity of $\{\mathbf{x}_i, f_i\}_{i=1}^{N_s}$ is taken care by the fact that $C(\mathbf{r})$ is zero for those values of $\mathbf{r}$ for which there are no points in $\{\mathbf{x}_i\}_{i=1}^{N_s}$ within the unity neighbourhood. The inaccuracies from transferring the points in $\{\mathbf{x}_i\}_{i=1}^{N_s}$ to the nearest grid point is taken care by the non-zero values in $C(\mathbf{r})$. Even though the cost function is defined on the entire reconstruction grid, the weighing term evaluates the accuracy of the reconstruction using only those sample locations where the measurements are available. The reconstruction error is equivalent to the least square error between non-grid measurements and the value of the function at these sample locations. Since the cost function is defined on the reconstruction grid, the main advantage in this formulation is that we directly solve for the regular grid values. Hence the need for determining a continuous function model and then re-sampling on the required grid is eliminated.

The second term, which is roughness term (also known as the regularization term) is implemented by convolving u with discrete second-derivative operators. The squared amplitude of this discrete derivative operation gives the roughness measure; λ is the trade-off

parameter between the accuracy of fit and smoothness of the reconstructed image. As λ increases, the reconstructed image becomes more and more smooth. The smoothness term is used as prior information to overcome the indeterminacy (ill-posedness) that might arise if there are grid points that do not have any measurements within the unit neighbourhood; λ should be sufficiently large so that the space between the large sampling gaps is filled adequately using the information from the nearby sample values.

The cost function (Eq. (8)) is a multi-variable quadratic function, where each required grid point value (pixels of $u(\mathbf{r})$) is a variable. In standard optimization literature, quadratic functions are typically expressed in vector-matrix notation, whereas the cost given in Eq. (8) is expressed in terms of image operations. To get the cost in vector-matrix format, let $\mathbf{u}$ represent a vector containing the pixels of $u(\mathbf{r})$ scanned in lexicographic order and let $\tilde{\mathbf{f}}$ represent a vector containing the pixels of $\tilde{f}(\mathbf{r})$. Then the cost can be expressed as

$$J(\mathbf{u}) = (\mathbf{u} - \tilde{\mathbf{f}})^T \mathbf{C} (\mathbf{u} - \tilde{\mathbf{f}}) + \lambda \mathbf{u}^T (\mathbf{D}_{xx}^T \mathbf{D}_{xx} + \mathbf{D}_{yy}^T \mathbf{D}_{yy} + \mathbf{D}_{xy}^T \mathbf{D}_{xy}) \mathbf{u}, \quad (10)$$

where $\mathbf{C}$ is the diagonal matrix containing the elements of $C(\mathbf{r})$ scanned in lexicographic order as its diagonal elements. Further, $\{\mathbf{D}_{xx}, \mathbf{D}_{yy}, \mathbf{D}_{xy}\}$ are the matrices corresponding to the filters $\{d_{xx}, d_{yy}, d_{xy}\}$. The cost can also be compactly expressed as follows:

$$J(\mathbf{u}) = \mathbf{u}^T \underbrace{\left[\mathbf{C} + \lambda (\mathbf{D}_{xx}^T \mathbf{D}_{xx} + \mathbf{D}_{yy}^T \mathbf{D}_{yy} + \mathbf{D}_{xy}^T \mathbf{D}_{xy}) \right]}_{\mathbf{Q}} \mathbf{u} - 2\mathbf{u}^T \underbrace{\mathbf{C}\tilde{\mathbf{f}}}_{\mathbf{b}}. \quad (11)$$

As evident in the following section, minimizing this cost function will be implemented using iterations that involve repeated multiplication of the matrix $\mathbf{Q}$ with optimization variable $\mathbf{u}$. The good news is that $\mathbf{Q}$ has a special structure such that building $\mathbf{Q}$ explicitly is not required. This is because $\mathbf{Q}$ is a sum of a diagonal matrix $\mathbf{C}$ and circulant matrices $\{\mathbf{D}_{xx}^T \mathbf{D}_{xx}, \mathbf{D}_{yy}^T \mathbf{D}_{yy}, \mathbf{D}_{xy}^T \mathbf{D}_{xy}\}$. Evaluating the product $\mathbf{C}\mathbf{u}$ is equivalent to evaluating the image product $C(\mathbf{r})u(\mathbf{r})$. Similarly, evaluating the product $\mathbf{D}_{xx}^T \mathbf{D}_{xx}\mathbf{u}$ is equivalent to evaluating the convolution $d_{xx}(-\mathbf{r}) * d_{xx}(\mathbf{r}) * u(\mathbf{r})$. Hence minimization of this cost function can be achieved using convolution and array multiplications only, thereby eliminating the need for building $\mathbf{Q}$ explicitly. Note that elimination of matrix building becomes possible only because the data fitting term in the cost given in Eq. (8) does not have terms linking

different pixels, which is due to the transformation described in section 2.1.

2.3 Minimization algorithm

The minimum of cost (Eq. (10)) can be uniquely determined by equating the gradient to zero. The gradient vector of the cost in vector-matrix form is given by

$$\mathbf{g}_u = \mathbf{Q}\mathbf{u} - \mathbf{b} = \mathbf{0}. \quad (12)$$

Eq. (12) represents a large sparse positive definite linear system of equations. Such a linear system can be efficiently solved by the method of conjugate gradient (CG) [33]. The efficiency of CG method can be further revamped by pre-conditioning. Pre-conditioning provides a way to speed up CG iteration, if a computationally inexpensive approximation for the inverse of the system matrix $\mathbf{Q}$ is available. Here, we propose a pre-conditioned conjugate gradient (PCG) method that exploits the specific structure of our problem. For clarity of presentation, we first provide a conceptual introduction to PCG method and then specify the proposed iterative scheme.

CG method is an extension of the method of steepest descent. Given a current estimate of the minimum $\mathbf{u}_k$, the method of steepest descent proceeds by doing a single variable minimization with argument to J restricted to be of the form $\mathbf{u}_k - \alpha \mathbf{g}_k$, where $\mathbf{g}_k$ is the gradient vector of J evaluated at $\mathbf{u}_k$, which is given by $\mathbf{g}_k = \mathbf{Q}\mathbf{u}_k - \mathbf{b}$. Here α is the minimization variable and if α_k denotes the minimum of this single variable minimization, the next improved estimate for the solution is determined as $\mathbf{u}_{k+1} = \mathbf{u}_k - \alpha_k \mathbf{g}_k$. CG adds a simple modification: it performs minimization of the form $\alpha_k = \arg\min_{\alpha} J(\mathbf{u}_k + \alpha \mathbf{d}_k)$, where $\mathbf{d}_k$ is the search direction obtained as perturbation of the gradient, where the perturbation is done along the previous search direction. In other words, the search direction is evaluated as $\mathbf{d}_k = -\mathbf{g}_k + \beta_k \mathbf{d}_{k-1}$. This perturbation is done such that $\mathbf{d}_{k-1}^T \mathbf{Q} \mathbf{d}_k = 0$, which leads to the formula $\beta_k = \frac{\mathbf{g}_k^T \mathbf{Q} \mathbf{d}_{k-1}}{\mathbf{d}_{k-1}^T \mathbf{Q} \mathbf{d}_{k-1}}$. Exploiting the fact that J is quadratic, we get a closed form expression for α_k : $\alpha_k = -\frac{\mathbf{g}_k^T \mathbf{Q} \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{Q} \mathbf{d}_k}$. Because of the recursive nature of the method, the relation $\mathbf{d}_{k-1}^T \mathbf{Q} \mathbf{d}_k = 0$ leads to the following relation that is known as $\mathbf{Q}$ -orthogonality: $\mathbf{d}_j^T \mathbf{Q} \mathbf{d}_k = 0 \forall j \in \{0, 1, \dots, k-1\}$. This further leads to other interesting algebraic properties among the gradients and search directions that manifest into the following property for CG: it completely exploits the distribution of eigenvalues of $\mathbf{Q}$ to get the optimal convergence rate without having the need to compute

the eigenvalues. Further, $\mathbf{Q}$ -orthogonality property leads to the following alternative formulas for α_k and β_k :

$$\alpha_k = -\frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{d}_k^T \mathbf{Q} \mathbf{d}_k}, \quad \beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k}.$$

Let $\mathbf{P}$ be an approximation for $\mathbf{Q}^{-1}$ and let $\mathbf{P}' = (\mathbf{P})^{1/2}$. The PCG method can be obtained as follows: we substitute $\mathbf{u} = \mathbf{P}'\mathbf{u}'$, and let $J'(\mathbf{u}') = J(\mathbf{P}\mathbf{u}')$ be the transformed cost function. We then write CG steps for $J'(\mathbf{u}')$ in terms of the variable $\mathbf{u}'$. We then transform the equations of the resulting steps to get the update rule in terms of $\mathbf{u}$. After these operations, we finally get the PCG step as follows: $\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha \mathbf{d}_k$ where $\alpha_k = -\frac{\mathbf{g}_{k+1}^T \mathbf{P} \mathbf{g}_k}{\mathbf{d}_k^T \mathbf{Q} \mathbf{d}_k}$; $\mathbf{d}_{k+1} = -\mathbf{P} \mathbf{g}_{k+1} + \beta_k \mathbf{d}_k$ where $\beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{P} \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{P} \mathbf{g}_k}$.

For our problem, we use the inverse of diagonal of the system matrix $\mathbf{Q}$ for the pre-conditioner $\mathbf{P}$. Referring to Eq. (11), we find that the part of $\mathbf{Q}$ contributed by the data fitting term, the matrix $\mathbf{C}$, is already diagonal. Hence $\mathbf{P} = [\mathbf{C} + \lambda \text{diag}(\mathbf{D}_{xx}^T \mathbf{D}_{xx} + \mathbf{D}_{yy}^T \mathbf{D}_{yy} + \mathbf{D}_{xy}^T \mathbf{D}_{xy})]^{-1}$, where $\text{diag}(\cdot)$ refers to the diagonal parts of its argument. Since $\mathbf{D}_{xx}$, $\mathbf{D}_{yy}$ and $\mathbf{D}_{xy}$ are circulant matrices corresponding to filters given in Eq. (9), the required diagonal part can be readily determined, which is $20\mathbf{I}$, where $\mathbf{I}$ is identity matrix. Hence the inverse diagonal of $\mathbf{Q}$ is given by

$$\mathbf{P} = [\mathbf{C} + 20\lambda \mathbf{I}]^{-1}. \quad (13)$$

Each step of PCG algorithm involves the evaluation of gradient, step length and search direction. The gradient in (14) involves the convolution with the derivative filters d_{xx} , d_{yy} and d_{xy} . These operations are implemented by directly convolving the search direction with the derivative filters. For a reconstruction grid with N grid points, the convolution operation requires $5N$ multiplications and $21N$ additions. Hence the gradient in (14) requires $6N$ multiplications and $22N$ additions for evaluation. The step length evaluation requires $5N$ multiplication and $2N$ addition operations. Furthermore, the search direction evaluation requires $5N$ multiplications and $5N$ additions. Hence, the total computational cost for each PCG is $16N$ multiplications and $29N$ additions.

Except for the initialization, the foregoing discussion completes the conceptual design of the minimization algorithm. However, for practical implementation, we need the algorithm to be specified in terms of steps that satisfy the following: (i) the steps should be expressed in terms of image operations instead of matrix-vector operations and (ii) the steps should show intermediate computations so that any specific computation is performed only once. The panel named **Algorithm 2** provides this, assuming that the initialization denoted by $u_0(\mathbf{r})$ is given.

$PCG(u_0(\mathbf{r}), \tilde{\mathbf{f}}(\mathbf{r}), \mathbf{C}(\mathbf{r}), p(\mathbf{r}), \lambda, \text{tolerance})$

Input:

$u_0(\mathbf{r})$: Initial guess about the solution at the regular grid location $\mathbf{r}$.

$\tilde{\mathbf{f}}(\mathbf{r})$: Estimated pixel value at the regular grid location $\mathbf{r}$ from the measurements.

$\mathbf{C}(\mathbf{r})$: Confidence in the estimated regular grid pixel values at location $\mathbf{r}$.

$p(\mathbf{r})$: Pixel value of the pre-conditioner at the pixel location $\mathbf{r}$.

λ : Trade-off parameter between the accuracy of the fit and the smoothness of the reconstructed image.

tolerance: Stopping criterion for the iteration.

Initialization:

$$\begin{aligned} g_0(\mathbf{r}) &= \mathbf{C}(\mathbf{r})u_0(\mathbf{r}) + \\ &\lambda \left(d_{xx}^T * d_{xx} + 2 * d_{xy}^T * d_{xy} + d_{yy}^T * d_{yy} \right) * u_0(\mathbf{r}) \\ &\quad - \mathbf{C}(\mathbf{r})\tilde{\mathbf{f}}(\mathbf{r}) \end{aligned} \quad (14)$$

$$\gamma_0 = \langle g_0(\mathbf{r}), p(\mathbf{r})g_0(\mathbf{r}) \rangle \quad (15)$$

$$d_0(\mathbf{r}) = -p(\mathbf{r})g_0(\mathbf{r}) \quad (16)$$

$$\text{relative_change} = 1, \quad k = 0 \quad (17)$$

while *relative_change* > *tolerance* **do**

$$\begin{aligned} s_k(\mathbf{r}) &= \mathbf{C}(\mathbf{r})d_k(\mathbf{r}) + \\ &\lambda \left(d_{xx}^T * d_{xx} + 2 * d_{xy}^T * d_{xy} + d_{yy}^T * d_{yy} \right) * d_k(\mathbf{r}) \end{aligned} \quad (18)$$

$$\alpha_k = \frac{\gamma_k}{\langle d_k(\mathbf{r}), s_k(\mathbf{r}) \rangle} \quad (19)$$

$$u_{k+1}(\mathbf{r}) = u_k(\mathbf{r}) + \alpha_k d_k(\mathbf{r}) \quad (20)$$

$$g_{k+1}(\mathbf{r}) = g_k(\mathbf{r}) + \alpha_k s_k(\mathbf{r}) \quad (21)$$

$$\gamma_{k+1} = \langle g_{k+1}(\mathbf{r}), p(\mathbf{r})g_{k+1}(\mathbf{r}) \rangle \quad (22)$$

$$\beta_k = \frac{\gamma_{k+1}}{\gamma_k} \quad (23)$$

$$d_{k+1}(\mathbf{r}) = -p(\mathbf{r})g_{k+1}(\mathbf{r}) + \beta_k d_k(\mathbf{r}) \quad (24)$$

$$\text{relative_change} = \frac{\|u_{k+1}(\mathbf{r}) - u_k(\mathbf{r})\|}{\|u_k(\mathbf{r})\|}, \quad (25)$$

$$k = k + 1$$

end

Algorithm 2: PCG algorithm.

Here $\mathbf{r}$ represents the regular grid image index. It is used to discriminate image from scalar quantities.

Now what remains to be specified is the way to find the initialization for PCG algorithm. To this end, we use a multi-resolution strategy. Multi-resolution analysis (MRA)

represents the signal at multiples scales, which are effectively implemented using pyramidal decomposition. For a given choice of scale J , pyramidal decomposition at level $k = 0, 1, \dots, J$ is obtained by convolving the image at level $k - 1$ with a low pass filter, followed by a down-sampling operation. At level $k = 0$, the algorithm is initialized with the original image. The energy compaction in MRA can be exploited to generate a good approximation to the original image [34] and for solving a dense system of linear equations [35].

At each scale, first we define an ensemble of pairs $\{\tilde{f}^{(i)}(\mathbf{r}), C^{(i)}(\mathbf{r})\}_{i=0}^{J-1}$ where $\tilde{f}^{(0)}(\mathbf{r}) = \tilde{f}(\mathbf{r})$ and $C^{(0)}(\mathbf{r}) = C(\mathbf{r})$ and J represents number of scales used. The subsequent images are computed recursively as follows: the image $\tilde{f}^{(i)}(\mathbf{r})$ (resp. $C^{(i)}(\mathbf{r})$) is obtained from $\tilde{f}^{(i-1)}(\mathbf{r})$ (resp. $C^{(i-1)}(\mathbf{r})$) by convolving with a 2D smoothing mask (denoted by $h(\mathbf{r})$) and then skipping the samples from odd-numbered rows and odd-numbered columns. Note that the size of $\tilde{f}^{(i)}(\mathbf{r})$ is less than the size of $\tilde{f}^{(i-1)}(\mathbf{r})$ by a factor of two along each dimension. We denote this operation by $\tilde{f}^{(i)}(\mathbf{r}) = [\tilde{f}^{(i-1)}(\mathbf{r}) * h(\mathbf{r})]_{\downarrow 2}$ and $C^{(i)}(\mathbf{r}) = [C^{(i-1)}(\mathbf{r}) * h(\mathbf{r})]_{\downarrow 2}$ where $\downarrow 2$ indicates the down-sampling operations. The index i is called the scale, and the image $\tilde{f}^{(i)}(\mathbf{r})$ (respectively $C^{(i)}(\mathbf{r})$) is called the representation of $\tilde{f}(\mathbf{r})$ (respectively $C(\mathbf{r})$) at scale i . Now we define the cost at scale i as follows:

$$J^{(i)}(u^{(i)}) = \sum_{\mathbf{r}} C^{(i)}(\mathbf{r}) (\tilde{f}^{(i)}(\mathbf{r}) - u^{(i)}(\mathbf{r}))^2 + \lambda \sum_{\mathbf{r}} \left((d_{xx} * u^{(i)}(\mathbf{r}))^2 + 2(d_{xy} * u^{(i)}(\mathbf{r}))^2 + (d_{yy} * u^{(i)}(\mathbf{r}))^2 \right). \quad (26)$$

Note that the number of variables in $J^{(i)}(u^{(i)})$ (number of pixels in $\tilde{f}^{(i)}(\mathbf{r})$, $C^{(i)}(\mathbf{r})$ and $u^{(i)}(\mathbf{r})$) is 4 times less than that of $J^{(i-1)}(u^{(i-1)})$. Similar to the expression given in Eq. (11), this scale-dependent cost can be expressed in the vector-matrix form as follows:

$$J^{(i)}(\mathbf{u}^{(i)}) = \mathbf{u}^{(i)T} \mathbf{Q}^{(i)} \mathbf{u}^{(i)} - 2\mathbf{u}^{(i)T} \mathbf{b}^{(i)} \quad (27)$$

where $\mathbf{Q}^{(i)} = [C^{(i)} + \lambda(\mathbf{D}_{xx}^T \mathbf{D}_{xx}^{(i)} + \mathbf{D}_{yy}^T \mathbf{D}_{yy}^{(i)} + \mathbf{D}_{xy}^T \mathbf{D}_{xy}^{(i)})]$ and $\mathbf{b}^{(i)} = C^{(i)} \tilde{\mathbf{f}}^{(i)}$. Here the matrices $\{C^{(i)}, \mathbf{D}_{xx}^{(i)}, \mathbf{D}_{yy}^{(i)}, \mathbf{D}_{xy}^{(i)}\}$ are defined in a similar way, but now they are pertinent to the scale i . The direct solution for the minimum of $J^{(i)}(\mathbf{u}^{(i)})$ in vector-matrix form is given by $\mathbf{u}_{opt}^{(i)} = [\mathbf{Q}^{(i)}]^{-1} \mathbf{b}^{(i)}$.

The multi-resolution method proceeds as follows. We first compute the minimum of $J^{(j-1)}(\mathbf{u}^{(j-1)})$ by direct matrix inversion: $\mathbf{u}_{opt}^{(j-1)} = [\mathbf{Q}^{(j-1)}]^{-1} \mathbf{b}^{(j-1)}$. This is feasible, because $j - 1$ represents the coarsest level and hence the number of variables is low. We expand this by a factor of two by interpolation as follows: along each dimension, we insert zeros in between each pixel such that the image size is doubled along each dimension; we then convolve with the same smoothing mask. We denote this by $u_0^{(j-2)}(\mathbf{r})$ and

represent this operation by $u_0^{(j-2)}(\mathbf{r}) = h(\mathbf{r}) * ([u_{opt}^{(j-1)}(\mathbf{r})]_{\uparrow 2})$. We use this as an initialization for minimizing the next level cost $J^{(j-2)}(u^{(j-2)})$ where the PCG algorithm specified in the panel **Algorithm 1** will be used. This process will be repeated until we get the minimum of $J^{(0)}(u^{(0)}) = J(u)$, which is the desired solution. The panel **Algorithm 3** provides a symbolic specification of this algorithm.

Although the proposed method appears to be extremely simple, its reconstruction accuracy is comparable to that of the gold-standard method in the field, the thin-plate spline method. Further, our method outperforms a recent method belonging to the line of methods developed as numerically efficient alternatives to the thin-plate spline method [31] both in terms of speed and accuracy. These claims will be substantiated in section 4. Moreover, since our method is purely based on regular grid array operations, our method is very attractive for parallelized implementations.

The approach adopted here was inspired from the work of Unser [36], but there are two factors that make the proposed method significantly different from their method: (i) our method has no restriction on the distribution of the input sample locations whereas Unser's multi-grid method [36] restricts input sample locations to be a subset of the regular grid and (ii) the multi-grid method [36] builds multi-resolution on the equation to be solved as opposed to our method where we build multi-resolution on the cost function. Because of the second factor, their method [36] leads to reconstruction quality that is not comparable to that of thin-plate spline method, whereas the proposed method gives reconstruction quality that is comparable to that of the thin-plate spline method.

Complete algorithm with multi-resolution:

Input: $\tilde{f}(\mathbf{r})$, $C(\mathbf{r})$, $h(\mathbf{r})$, λ , *tolerance*

Define: $\tilde{f}^{(0)}(\mathbf{r}) = \tilde{f}(\mathbf{r})$, $C^{(0)}(\mathbf{r}) = C(\mathbf{r})$

for $i = 1, \dots, j - 1$ **do**

$$\tilde{f}^{(i)}(\mathbf{r}) = [\tilde{f}^{(i-1)}(\mathbf{r}) * h(\mathbf{r})]_{\downarrow 2}$$

$$C^{(i)}(\mathbf{r}) = [C^{(i-1)}(\mathbf{r}) * h(\mathbf{r})]_{\downarrow 2}$$

end

Build: $\mathbf{Q}^{(j-1)}, \mathbf{b}^{(j-1)}$

Compute: $\mathbf{u}_{opt}^{(j-1)} = [\mathbf{Q}^{(j-1)}]^{-1} \mathbf{b}^{(j-1)}$

Assemble: Get image $u_{opt}^{(j-1)}(\mathbf{r})$ from vector $\mathbf{u}_{opt}^{(j-1)}$

for $i = 1, \dots, j - 1$ **do**

$$u_0^{(j-1-i)}(\mathbf{r}) = h(\mathbf{r}) * ([u_{opt}^{(j-i)}(\mathbf{r})]_{\uparrow 2})$$

$$p(\mathbf{r}) = 1 / (C^{(j-1-i)}(\mathbf{r}) + 20\lambda)$$

$$u_{opt}^{(j-1-i)}(\mathbf{r}) = PCG(u_0^{(j-1-i)}(\mathbf{r}),$$

$$\tilde{\mathbf{f}}^{(j-1-i)}(\mathbf{r}), C^{(j-1-i)}(\mathbf{r}), p(\mathbf{r}), \lambda, \text{tolerance}) \quad (28)$$

end

Output: $u_{opt}^{(0)}(\mathbf{r})$

Algorithm 3: Multi-resolution PCG.

3. Experimental results

In this section, we carry out reconstruction trials to study the effectiveness of the proposed SDA technique. Further, we benchmark the proposed method against the thin-plate method. For the thin-plate spline method, we used GMRES method [37] for solving the linear system of equations. We generate the test measurements either from synthetic phantom or from standard digital images. The performance of an algorithm is measured in terms of the reconstruction accuracy and the time required to produce the reconstructed image. We also consider noisy measurements in our study to evaluate the robustness of the proposed algorithm to the measurement noise. In the case of noisy data, the smoothing parameter λ is tuned heuristically with the following strategy: starting from the lowest possible value, a series of reconstructions are performed with regular increments in λ .

The result that does not have any noise-related artifacts with the lowest possible value of λ is chosen.

To quantitatively evaluate the reconstruction accuracy, we employ the following relative reconstruction error metric, which computes the error between the reconstructed image and the original image.

$$\epsilon = \frac{\|I_{org} - I_{rec}\|_2}{\|I_{org}\|_2} \quad (29)$$

where I_{org} is the original image and I_{rec} is the reconstructed image from the measurements.

We first applied our algorithm on a 256×256 synthetic phantom image, which has a pixel value of 255 inside an annular disc and a pixel value of zero outside. The phantom image is sampled uniformly along the radial as well as the angular direction with a radial spacing of 5 pixels and an

Figure 3. Evaluation using synthetic phantom image: (a) phantom image, (b) sample locations, (c) reconstruction by RGWS (proposed method) and (d) thin-plate spline reconstruction.

angular increment of 18° . This sampling procedure generated 500 samples. The reconstruction of the phantom image from these samples is challenging due to the sparse

Figure 4. Comparison of reconstruction error with that of the thin-plate spline method: noiseless case.

distribution of the measurements. Figure 3a is the original phantom image. Figure 3b shows the sample locations, while the reconstructions with the proposed method and thin-plate spline are shown in figure 3c and d, respectively. The relative reconstruction error for thin-plate spline and the proposed method is evaluated using the metric in Eq. (29) and is found to be 0.02 and 0.03, respectively. The closeness of the relative reconstruction error values indicates the ability of the proposed algorithm to produce results close to those by the thin-plate spline method. Further, the experiment reveals the ability of the proposed method to handle large sampling gaps just like the thin-plate spline method.

In our next experiment, we applied our algorithm on randomly spaced samples obtained from two well-known test images, namely the pepper image and the brain image. For each sample location, the sample value is computed using a linear interpolation of nearest grid values. The

Figure 5. Evaluation using 20% of noiseless pepper image samples: (a) original image, (b) sample locations, (c) reconstruction by RGWS (proposed method) and (d) thin-plate spline reconstruction.

Figure 6. Evaluation using 20% of noiseless brain image samples: (a) original image, (b) sample locations, (c) reconstruction by RGWS (proposed method) and (d) thin-plate spline reconstruction.

Figure 7. Comparison of reconstruction error with thin-plate spline method: noisy case with 20 dB SNR.

reconstruction was performed for a series of data sets with sample density ranging from 5% to 65% and reconstruction

errors for both thin-plate spline method and proposed reconstruction were evaluated. For each value of sample density, the reconstruction error was evaluated 10 times by performing reconstructions on 10 sets of independently generated random sample locations. The relative reconstruction error curves corresponding to pepper image are shown in figure 4. A similar reconstruction error plot is observed for brain image. We see that both the proposed and thin-plate methods have similar reconstruction error. This result confirms that our method is capable of producing results close to those by the thin-plate spline method and its ability to effectively tackle the large sampling gap. Furthermore, we see that the proposed method does not suffer from numerical instabilities even at high sampling densities, unlike the thin-plate method. Figures 5 and 6 display the reconstructed images corresponding to 20% sample density for pepper and brain images, respectively.

Figure 8. Evaluation using 20% of noisy brain image samples: (a) original image, (b) sample locations, (c) reconstruction by RGWS (proposed method) and (d) thin-plate spline reconstruction.

We see that the reconstructed image using the proposed method is as good as the thin-plate spline reconstruction in both cases.

In our next experiment, we evaluated the performance of the proposed method in the presence of noise and compared it to that of thin-plate spline. For this, we added white Gaussian noise (AWGN) to the measurements such that the signal to noise ratio was 20 dB. Here as well, we considered the sampling densities in the ranges 5 – 65%. The relative reconstruction error curve is shown in figure 7. The reconstructed images with 20% noisy measurements from pepper and brain images for thin-plate spline and proposed method are shown in figures 8 and 9, respectively. These results indicate that the proposed method exhibits the same noise-robustness of thin-plate spline reconstruction.

Now we compare the reconstruction speed of the proposed and the thin-plate spline method. We use the pepper image with its pixel density being varied from 5% to 65%. The input measurements are generated as before. Figure 10 compares the reconstruction time for different reconstruction methods. We see that the time required for thin-plate spline reconstruction grows rapidly with the sampling density. This is due to the time required for generating the coefficient matrix and for solving the system of equations using GMRES. However, in the proposed method, the reconstruction time decreases as the sampling density increases. This is because of the improvement in the condition number of the system matrix.

In our next experiment, we compared the relative reconstruction error and time for the proposed method with the PoU approach in [31]. For this, we varied the sampling

Figure 9. Evaluation using 20% of noisy pepper image samples: (a) original image, (b) sample locations, (c) reconstruction by RGWS (proposed method) and (d) thin-plate spline reconstruction.

Figure 10. Comparison of reconstruction time between thin-plate spline method and RGWS (proposed method).

density from 5% to 65% of the 256×256 pepper image. As before, sample locations are generated from a uniform

distribution. The measurement locations along with measurements are given as input to both the proposed and the PoU approach. The experiment is repeated 10 times to average out the effect of random sampling. The relative reconstruction error curve is shown in figure 11 and time taken for reconstruction is shown in figure 12.

The reconstruction error plot clearly shows that the proposed method produces better reconstruction than the PoU approach. Thus our method produces reconstruction comparable to that of the thin-plate spline method while taking less time and computations than the state of the art PoU approximation to thin-plate spline as shown by the reconstruction time plot.

The PoU method [31], with which we compared the proposed method earlier, is one of the recent advancements in developing computationally affordable SDA. Hence, in the following, we provide even more extensive comparison

Figure 11. Comparison of relative reconstruction error between PoU method and RGWS (proposed method).

Figure 12. Comparison of reconstruction time between PoU method and RGWS (proposed method).

using various data sets. For this purpose, we used eight test images of size 256×256 , which are displayed in figure 13. The images with label ‘CIL’ are from Cell Image Library [38]. The sample locations were drawn from a uniform distribution and the corresponding measurements were

corrupted by additive white Gaussian noise. The noise variance was set to maintain the signal to noise ratio at 20 dB. The input sampling density was varied from 20% to 60% with an increment of 10%. The comparison results are reported in tables 1 and 2. As evident from the table, our method outperforms the PoU method both in terms of speed and reconstruction accuracy.

These reconstruction examples emphasize the fact that the proposed method is much more computationally efficient than the thin-plate spline method with only a little compromise in the reconstruction accuracy. The factors contributing to the computational efficiency are the following: (i) the linear system of equations to be solved is better-conditioned; (ii) there is no expensive re-sampling required in the proposed method as done in the thin-plate spline method; (iii) in the proposed method, the condition number of the system to be solved improves with increase in the sample density as opposed to the thin-spline method, where the condition number of the system gets worse; (iv) the matrix representing the linear system of equations is not required to be stored explicitly; (v) the iterations involved in solving the linear system of equations for getting the reconstruction require only regular grid digital filtering and array multiplication; (vi) our method allows multi-resolution, leading to fast reconstruction. The only constraint in applying the proposed method is that the reconstruction step size should be known *a priori*. If it is not known, we can apply the proposed method with the step size equal to the smallest gap between the given samples. In that case, the size of the linear system to be solved in the proposed method will be much larger than that of the thin-plate spline method. However, due to factors (i), (ii), (v) and (vi), the proposed method will still be faster than the thin-plate spline method.

Figure 13. Test images: (a) Lena, (b) Boat, (c) Barbara, (d) Baboon, (e) CIL 10016, (f) CIL 248, (g) CIL 1585 and (h) CIL 7437.

Table 1. Table listing the relative reconstruction error between proposed approach (RGWS) and PoU approach.

Percentage of samples										
	20		30		40		50		60	
	RGWS	PoU	RGWS	PoU	RGWS	PoU	RGWS	PoU	RGWS	PoU
Lena	0.1090	0.1275	0.0979	0.1135	0.0925	0.1052	0.0880	0.0988	0.0848	0.0927
Baboon	0.1433	0.1526	0.1333	0.1457	0.1261	0.1399	0.1217	0.1351	0.1179	0.1318
Barbara	0.1185	0.1402	0.1079	0.1252	0.1023	0.1154	0.0974	0.1090	0.0933	0.1025
Boat	0.1210	0.1406	0.1109	0.1293	0.1049	0.1213	0.1003	0.1159	0.0961	0.1095
CIL 248	0.1677	0.2212	0.1551	0.2000	0.1393	0.1848	0.1331	0.1747	0.1287	0.1659
CIL 1585	0.1777	0.2464	0.1557	0.2178	0.1418	0.2008	0.1356	0.1960	0.1291	0.1805
CIL 7437	0.1311	0.1563	0.1242	0.1473	0.1110	0.1338	0.1066	0.1270	0.1025	0.1214
CIL 10016	0.1702	0.2401	0.1509	0.2118	0.1377	0.1953	0.1306	0.1795	0.1248	0.1680

Table 2. Reconstruction time in seconds for proposed approach (RGWS) and PoU.

Percentage of samples										
	20		30		40		50		60	
	RGWS	PoU	RGWS	PoU	RGWS	PoU	RGWS	PoU	RGWS	PoU
Lena	2.4000	5.5200	1.9000	6.2500	1.3100	7.7400	1.0900	9.8000	1.0000	10.0400
Baboon	2.6100	5.5200	1.7800	6.0100	1.5200	7.6700	1.2800	9.4300	1.2700	11.2400
Barbara	2.6900	5.4700	1.9200	5.9600	1.4300	7.6800	1.0600	9.4300	0.9400	11.2000
Boat	2.1700	5.6000	1.6700	6.0100	1.3600	7.6400	1.0300	9.5400	1.0300	11.0300
CIL 248	2.4300	5.4800	1.5700	5.9700	1.3600	7.5700	1.1200	9.4900	0.9300	11.1900
CIL 1585	2.2700	5.6200	1.7200	6.1800	1.3500	7.7000	1.0300	9.3300	0.9200	11.0600
CIL 7437	2.6000	5.4300	1.7300	6.0500	1.5000	7.5900	1.2100	9.3100	0.9500	11.1800
CIL 10016	2.7400	5.6500	1.7100	6.3600	1.6800	7.7400	1.2800	9.3200	0.9200	11.5100

4. Conclusion

We proposed a new method for SDA using digital filtering. The proposed method draws inspiration from variation-based reconstruction using thin-plate spline. The reconstruction problem is posed as an unconstrained optimization in terms of regular grid images estimated from non-grid measurements. The solution to the optimization problem is computed using the CG algorithm. To provide a good initial estimate for the CG algorithm, multi-resolution approach is used. While yielding a reconstruction quality that is close to that of the thin-plate function method, the method has four major advantages when compared with the thin-plate spline method: (i) the linear system of equations involved is well-conditioned; (ii) there is no expensive re-sampling step since the method directly solves for the required regular grid pixels; (iii) the matrix representing the linear system of equation is not required to be stored explicitly and (iv) the iterations involved in solving the linear system of equations for getting the reconstruction require only regular grid digital filtering and array multiplication. While the first two advantages can be found in subspace variational methods, the last two, which make the proposed algorithm extremely attractive for parallelization, is unique to our approach.

Acknowledgement

This work was supported by the Young Scientist Scheme of Department of Science and Technology, Government of India, under Grant SB/FTP/ETA-240/2013.

References

- [1] Vio R, Strohmer T and Wamsteker W 2000 On the reconstruction of irregularly sampled time series. *Publ. Astron. Soc. Pac.* 112(767): 74–90
- [2] Arigovindan M, Suhling M, Jansen C, Hunziker P and Unser M 2007 Full motion and flow field recovery from echo Doppler data. *IEEE Trans. Med. Imag.* 26(1): 31–45
- [3] Park S C, Park M K and Kang M G 2003 Super-resolution image reconstruction: a technical overview. *IEEE Sig. Process. Mag.* 20(3): 21–36
- [4] Rauth M and Strohmer T 1998 Smooth approximation of potential fields from noisy scattered data. *Geophysics* 63(1): 85–94
- [5] Faugeras O and Keriven R 2002 Variational principles, surface evolution, PDEs, level set methods and the stereo

- problem. In: *5th IEEE EMBS International Summer School on Biomedical Imaging*, 83 pp
- [6] Dijndam A, Sehonewille M and Hindriks C 1999 Reconstruction of band-limited signals, irregularly sampled along one spatial direction. *Geophysics* 64(2): 524–538
 - [7] Rashidi M 2010 Non-uniform sampling and reconstruction of multi-band signals and its application in wideband spectrum sensing of cognitive radio. *arXiv preprint arXiv:1010.2158*
 - [8] Fenn M and Steidl G 2006 Robust local approximation of scattered data. In: *Geometric properties for incomplete data* Springer, Netherlands 317–334
 - [9] Schall O, Belyaev A and Seidel H P 2005 Robust filtering of noisy scattered point data. In: *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings* IEEE 71–144
 - [10] Stein M L 2012 *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media
 - [11] Lai M J 1996 Scattered data interpolation and approximation using bivariate C1 piecewise cubic polynomials. *Comput. Aided Geom. Des.* 13(1): 81–88
 - [12] Arigovindan M 2005 Variational reconstruction of vector and scalar images from non-uniform samples. *These de doctorat, École Polytechnique Fédérale de Lausanne (EPFL)* 94: 96
 - [13] Cazals F and Giesen J 2006 Effective computational geometry for curves and surfaces. In: *Delaunay triangulation based surface reconstruction*. Springer 231–276
 - [14] Rauth M 1995 Application of 2D methods for scattered data approximation to geophysical data sets. In: *Sampling Theory and Applications, Proceedings of the SAMPTA'95 Workshop, Jurmala (Riga), Latvia* 38–43
 - [15] Shepard D 1968 A two-dimensional interpolation function for irregularly-spaced data. In: *Proceedings of the 1968 23rd ACM National Conference* 517–524
 - [16] Strohmer T and Tanner J 2006 Fast reconstruction methods for bandlimited functions from periodic nonuniform sampling. *SIAM J. Num. Anal.* 44(3): 1073–1094
 - [17] Duchon J 1977 Splines minimizing rotation-invariant seminorms in Sobolev spaces. 85–100
 - [18] Franke R and Nielson G 1980 Smooth interpolation of large sets of scattered data. *Int. J. Num. Methods Eng.* 15(11): 1691–1704
 - [19] Strang G and Nguyen T 1996 *Wavelets and filter banks*. SIAM. Philadelphia, USA
 - [20] Micchelli C A 1984 Interpolation of scattered data: distance matrices and conditionally positive definite functions. In: *Approximation theory and spline functions* Springer 143–145. Berlin, Germany
 - [21] Arigovindan M, Suhling M and Unser M 2005 Variational image reconstruction from arbitrarily spaced samples: a fast multiresolution spline solution. *IEEE Trans. Image Process.* 14(4): 450–460
 - [22] Beatson R K, Cherrie J B and Mouat C T 1999 Fast fitting of radial basis functions: methods based on preconditioned GMRES iteration. *Adv. Comput. Math.* 11(2): 253–270
 - [23] Carr J C, Fright W R and Beatson R K 1997 Surface interpolation with radial basis functions for medical imaging. *IEEE Trans. Med. Imag.* 16(1): 96–107
 - [24] Fasshauer G E 2007 *Meshfree approximation methods with Matlab*. vol. 6 World Scientific Publishing Co Inc, Singapore
 - [25] Buhmann M D 2003 *Radial basis functions: theory and implementations*. Cambridge University Press, Cambridge, England
 - [26] Baxter B 1992 *The interpolation theory of radial basis functions*. Cambridge, England
 - [27] Mongillo M 2011 Choosing basis functions and shape parameters for radial basis function methods. *SIAM Undergrad. Res. Online* 4: 190–209
 - [28] Fasshauer G E and McCourt M J 2012 Stable evaluation of Gaussian radial basis function interpolants. *SIAM J. Sci. Comput.* 34(2): A737–A762
 - [29] Wendland H 2006 Computational aspects of radial basis function approximation. *Stud. Comput. Math.* 12: 231–256
 - [30] Cavoretto R 2013 Partition of unity algorithm for two-dimensional interpolation using compactly supported radial basis functions. *Commun. Appl. Commun. Appl. Ind. Math.* 3(2)
 - [31] Cavoretto R, De Rossi A and Perracchione E 2015 Partition of unity interpolation on multivariate convex domains. *Int. J. Model. Simul. Sci. Comput.* 6(04): 1550034
 - [32] Morozov O V, Unser M and Hunziker P 2011 Reconstruction of large, irregularly sampled multidimensional images. A Tensor-Based Approach. *IEEE Trans. Med. Imag.* 30(2): 366–374
 - [33] Shewchuk J R et al. 1994 An introduction to the conjugate gradient method without the agonizing pain
 - [34] Salomon D 2004 *Data compression: the complete reference*. Springer Science & Business Media, New York, USA
 - [35] Harten A and Yad-Shalom I 1994 Fast multiresolution algorithms for matrix-vector multiplication. *SIAM J. Num. Anal.* 31(4): 1191–1218
 - [36] Unser M 1995 Multigrid adaptive image processing. In: *Proceedings of the International Conference on Image Processing* IEEE vol. 1 49–52
 - [37] Saad Y and Schultz M H 1986 GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* 7(3): 856–869
 - [38] Brandner D and Withers G *The Cell Image Library, CIL: 248, 1585, 7437 and 10016*. Available at <http://www.cellimagelibrary.org>