Reconstruction of Large, Irregularly Sampled Multidimensional Images. A Tensor-Based Approach

Oleksii Vyacheslav Morozov*, *Member, IEEE*, Michael Unser, *Fellow, IEEE*, and Patrick Hunziker

***Abstract*—Many practical applications require the reconstruction of images from irregularly sampled data. The spline formalism offers an attractive framework for solving this problem; the currently available methods, however, are hard to deploy for large-scale interpolation problems in dimensions greater than two (3-D, 3-D+time) because of an exponential increase of their computational cost (curse of dimensionality). Here, we revisit the standard regularized least-squares formulation of the interpolation problem, and propose to perform the reconstruction in a uniform tensor-product *B*-spline basis as an alternative to the classical solution involving radial basis functions. Our analysis reveals that the underlying multilinear system of equations admits a tensor decomposition with an extreme sparsity of its one dimensional components. We exploit this property for implementing a parallel, memory-efficient system solver. We show that the computational complexity of the proposed algorithm is essentially linear in the number of measurements and that its dependency on the number of dimensions is significantly less than that of the original sparse matrix-based implementation. The net benefit is a substantial reduction in memory requirement and operation count when compared to standard matrix-based algorithms, so that even 4-D problems with millions of samples become computationally feasible on desktop PCs in reasonable time. After validating the proposed algorithm in 3-D and 4-D, we apply it to a concrete imaging problem: the reconstruction of medical ultrasound images (3-D+time) from a large set of irregularly sampled measurements, acquired by a fast rotating ultrasound transducer.**

***Index Terms*—*B*-spline, CANDECOMP, reconstruction from irregular samples, tensor, tensor decomposition, variational reconstruction, 4-D ultrasound reconstruction.**

I. Introduction

The problem of data reconstruction from irregularly sampled measurements is frequently encountered in the context of biomedical imaging. For instance, the use of acquisition schemes with nonuniform, non-Cartesian spatial sampling requires a reformatting of data samples to an evenly-spaced Cartesian grid, for further analysis on a computer (e.g., ultrasound scanline-conversion). Such images are sampled sequentially along a series of scan-lines, which are nonuniformly distributed in space and in addition, do not coincide in time, further complicating the reconstruction. Similar problems of nonuniform sampling in space and/or time typically occur in many macroscopic (CT, MRI, SPECT) as well as in microscopic imaging (confocal microscopy, raster microscopy). Motion of the imaged object and time-dependent changes of image intensity also contribute to the complexity of the reconstruction problem.

Nonuniform sampling can introduce an oversampling at some locations relative to the Cartesian target grid, making an optimal smoothing method in these oversampled areas desirable, whereas in other locations of the same image, severe undersampling may occur, requiring robust interpolation. In conventional approaches, this combination of smoothing, interpolation, and time-sequential acquisition—even of a single image—are typically handled separately, facilitating implementation, but introducing multiple error sources.

The generic signal reconstruction/nonuniform interpolation problem consists of finding a continuous representation of a d-dimensional signal from arbitrarily-sampled, noisy point measurements. In the absence of any constraints on the distribution of the sampled data, the problem is ill-posed and does not have a unique solution. A natural approach for resolving this ambiguity is to introduce some prior constraints on the solution and to formulate the reconstruction as a variational problem. The cost function to be minimized is typically chosen to be the sum of two components: 1) a data term, which quantifies the fitting error between the model and the measurements, and 2) a continuous-domain regularization functional, which introduces a penalty for nonsmooth, and thus, improbable solutions. When the latter functional is quadratic, the variational problem can be solved analytically and the optimal solution is represented as a linear combination of radial basis functions (RBFs) [1], [2]. If one further imposes that the reconstruction should be independent upon any particular choice of coordinate system/spatial units, our choice of solutions narrows down to the class of thin-plate splines [1], [3], which are popular in applications, especially in the context of landmark-based image registration where the number of data points is small [4]. While thin-plate splines have many attractive mathematical properties, they are notoriously hard to deploy for large-scale interpolation problems. The main difficulty is that the underlying system matrix is dense and poorly conditioned, especially as the number of samples increases. Moreover, the evaluation of the reconstructed function at a single location is computationally expensive with a complexity that is proportional to the number of data points. During recent years, various solutions have been proposed to overcome these limitations [5]–[8]. However, for cases of millions of data samples, the radial basis function (RBF) approach still poses significant difficulties. To our knowledge, current

Manuscript received June 25, 2010; revised August 28, 2010; accepted September 08, 2010. Date of publication September 27, 2010; date of current version February 02, 2011. *Asterisk indicates corresponding author.*

*O. V. Morozov is with the University Hospital of Basel, Physics in Medicine Group, CH-4031 Basel, Switzerland (e-mail: morozova@uhbs.ch).

P. Hunziker is with the University Hospital of Basel, Physics in Medicine Group, CH-4031 Basel, Switzerland (e-mail: hunzikerp@uhbs.ch).

M. Unser is with the Ecole Polytechnique Federale de Lausanne, Biomedical Imaging Group, Swiss Federal Institute of Technology Lausanne (EPFL), CH-1015 Lausanne, Switzerland (e-mail: michael.unser@epfl.ch).

Digital Object Identifier 10.1109/TMI.2010.2078832

RBF-based algorithms can handle up to 5 000 000 samples in 2-D and up to 250 000 samples in 3-D, running on conventional hardware [9]. There exist other approaches to the problem, which are either at risk to give unsatisfactory solutions or have limited computability similarly to the RBF [10]–[12].

The work of Arigovindan *et al.* [13] presents a computationally-efficient alternative to the RBF with much better numerical behavior. There, the use of the tensor product of 1-D *B*-splines as a basis for approximating the analytical solution of the optimal method was proposed. The main advantages of this approach are: the linear system arising from the formulation is well-conditioned; the system matrix is sparse and enjoys multiresolution properties that are exploited to derive an efficient Multigrid solver with a complexity mainly dependent upon the size of the reconstruction grid; the cost of resampling is minimal with a complexity $O((n+1)^d)$ per evaluated sample, where n is the degree of the applied *B*-spline function (for most frequent case of cubic *B*-spline $n = 3$), d is the number of dimensions of the reconstruction grid. The approach is extremely favorable for 2-D problems [13], but is still facing a computational bottleneck as the number of dimensions increases (curse of dimensionality). The intrinsic limitation is that the computational and storage requirements of the method, which are directly tied to the reconstruction grid, increase exponentially with the number of dimensions.

The authors of [14] and [15] adopted the approach in [13] and presented an algorithm based on subdivision of the target grid into overlapping blocks. Subproblems corresponding to each block were independently solved using the multigrid solver. The proposed algorithm was successfully applied to reconstruction of relatively large 3-D datasets. Note that in cases when a significant smoothing is imposed, the proposed block-wise solution is at risk of introducing discontinuities between neighboring blocks. In analogy to [13], the computational and storage requirements of the algorithm are still dependent linearly on the target grid size and exponentially on the number of dimensions, making it impractical for large-scale problems with dimensions greater than 3.

To overcome the computational bottleneck described in above and to be able to reconstruct large, arbitrarily-sampled, multidimensional datasets, we propose to revisit the *B*-spline-based variational approach. In particular, we introduce a tensor-based formalism that reveals the intrinsic structure of the underlying system of equations, and which suggests some efficient factorization along the dimensions of the data. We then design an efficient, multilinear solver that is capable of performing the reconstruction of such data with millions of data points in any number of dimensions using standard desktop PCs in reasonable time. This makes our approach more advantageous than the original approach proposed in [13] and its adoption [15]. We evaluate the proposed algorithm on 3-D and 4-D datasets while providing some performance metrics to document its computational advantages. Finally, we present a practical example of reconstruction of a large 4-D medical ultrasound dataset from irregularly sampled noisy measurements.

A. Summary of Notation

Scalars are denoted by lower-case Latin and Greek symbols, such as a and λ. Euclidean space of dimensionality d is denoted by $\mathbb{R}^d$. A vector in $\mathbb{R}^d$ is denoted by a lower case bold symbol, such as $\boldsymbol{t} \in \mathbb{R}^d$. Upper-case symbols with calligraphic font are used for designation of tensors, such as $\mathcal{A}$.

II. *B*-Spline Formulation of the Problem

Our task is to reconstruct an unknown, continuously-defined signal $s(\boldsymbol{t})$, $\boldsymbol{t} \in \mathbb{R}^d$ given a set $\{f_k = s(\boldsymbol{t}_k) + n_k\}_{k=1}^K$ of K noisy sample values at irregularly-spaced sampling locations $\boldsymbol{t}_k$. We will do so by fitting a function, which is represented as a linear combination of basis functions, to the data according to some regularized least-squares criterion. We choose to represent our signal in a *B*-spline basis which combines a number of advantages: continuous representation, finite support, good approximation properties, and fast computation, as described in [16].

In contrast to the common way of RBF-based scattered data interpolation where the centers of the basis functions $\varphi(\|\boldsymbol{t}-\boldsymbol{t}_k\|)$ coincide with the sampling locations $\boldsymbol{t}_k$, the proposed uniform *B*-spline reconstruction is based on representing the unknown solution as a weighted sum of *B*-spline basis functions located at the nodes of a uniform grid, covering the domain of definition of the signal. Thus, a continuous representation of the solution in d dimensions is given by the following expansion:

$$s(t_1,\ldots,t_d) = \sum_{(i_1,\cdots,i_d)\in\mathbb{Z}^d} c_{i_1,\cdots,i_d}\beta^n\left(\frac{t_1}{a}-i_1\right) \cdots \beta^n\left(\frac{t_d}{a}-i_d\right) \quad (1)$$

where a is the step size of the grid which controls the quality of the discretization; $\beta^n(t)$ is the *B*-spline of degree n.

The cost function to be minimized combines a usual least-squares data fitting term with a continuous-domain regularization functional $\|s\|_{D_p}^2$ that penalizes nonsmooth solutions; i.e.,

$$\xi(s,f) = \sum_{k=1}^{K} |s(\boldsymbol{t}_k) - f_k|^2 + \lambda\|s\|_{D_p}^2 \quad (2)$$

where λ is a tradeoff parameter chosen for a compromise between the quality of data fit and a non-oscillating behavior of the solution; $\|s\|_{D_p}$ is Duchon's semi-norm of order p which has the important property of being scale- and rotation-invariant [1], [3]. The general d-dimensional form of this regularization functional is

$$\|s\|_{D_p}^2 = \sum_{p_1+\cdots+p_d=p} \frac{p!}{p_1!\cdots p_d!} \times \int_{\mathbb{R}^d} \left(\frac{\partial^p s(\boldsymbol{t})}{\partial t_1^{p_1}\cdots\partial t_d^{p_d}}\right)^2 \mathrm{d}t_1\cdots\mathrm{d}t_d. \quad (3)$$

It involves the sum of $N_p = \binom{p+d-1}{d-1}$ distinct quadratic terms corresponding to all partial derivatives of order p. We recall that the global, unconstrained minimization of (2) together with (3) defines the classical thin-plate spline solution [1], [3]. The main difference here is that we are searching for a solution of the form (1) which results in a discretized version of the problem in a uniform *B*-spline basis. The problem therefore boils down to finding the *B*-spline coefficients $c_{i_1,\cdots,i_d}$ that minimize (2) and therefore uniquely specify the continuous-domain reconstruction $s(\boldsymbol{t})$. Note that the discrepancy with the theoretical thin-plate spline solution can be made arbitrarily small via the adjustment of the step size a—indeed, the discretization error is guaranteed to decay like $O(a^{n+1})$ where n is the degree of the *B*-spline basis in (1) with the constraint that $n \geq p$.

III. Tensor Characteristics of the Problem

We will now reformulate the approximation problem making extensive use of a tensor formalism (cf. Appendix A). The latter allows a convenient and natural description of multilinear algebra problems and offers advantages over alternative notations [17]. For the sake of simplicity, we focus on the case $d = 3$, keeping in mind that the reasoning and formulaes are readily transposable to any number of dimensions. To each direction of the reconstruction grid with extents $[N_X, N_Y, N_Z]$, we assign a vector space with corresponding dimensionality $X \subseteq \mathbb{R}^{N_X}, Y \subseteq \mathbb{R}^{N_Y}, Z \subseteq \mathbb{R}^{N_Z}$. In conjunction to this ordered space sequence, we define the measurement vector space $F \subseteq \mathbb{R}^K$. The weighting coefficients in the expansion (1) are represented as a tensor in $X \times Y \times Z \subseteq \mathbb{R}^{N_X \times N_Y \times N_Z}$ with components $\mathcal{C}^{xyz}$. The evaluation of (1) at the location of the kth measurement can be represented as a component of a vector in F

$$\mathcal{S}^k = \mathcal{B}^k_{xyz}\mathcal{C}^{xyz} = \left(\mathcal{E}^k_x \cdot \mathcal{G}^k_y \cdot \mathcal{H}^k_z\right) \cdot \mathcal{C}^{xyz} \tag{4}$$

with the convention that $\mathcal{S}^k = s(\boldsymbol{t}_k)$ and $\mathcal{C}^{xyz}$ represent the *B*-spline coefficients $c_{i_1,\cdots,i_d}$ in (1). The remaining tensors $\mathcal{E}^k_x, \mathcal{G}^k_y, \mathcal{H}^k_z$, which are the factors of $\mathcal{B}^k_{xyz}$, are obtained from the respective evaluation of the *B*-spline functions $\beta^n((t_{1k}/a) - x), \beta^n((t_{2k}/a) - y), \beta^n((t_{3k}/a) - z)$, at the sampling locations $\boldsymbol{t}_k$. Note that these vectors are sparse due to the finite support of the *B*-spline basis with at most $n + 1$ nonzero values.

The first quadratic term in (2) is then expressed as

$$\begin{aligned} \|\mathcal{S} - \mathcal{F}\|^2 &= (\mathcal{S}_k - \mathcal{F}_k) \cdot (\mathcal{S}^k - \mathcal{F}^k) \\ &= \mathcal{S}_k\mathcal{S}^k - 2\mathcal{S}_k\mathcal{F}^k + \mathcal{F}_k\mathcal{F}^k \\ &= \left(\mathcal{C}_{x_1y_1z_1}\mathcal{B}^{x_1y_1z_1}_k\right)\left(\mathcal{B}^k_{xyz}\mathcal{C}^{xyz}\right) \\ &\quad - 2\mathcal{C}_{xyz}\mathcal{B}^{xyz}_k\mathcal{F}^k + \mathcal{F}_k\mathcal{F}^k \end{aligned} \tag{5}$$

where $\mathcal{F}^k$ is the kth entry of the measurement vector $\mathcal{F}$.

It can be shown that the regularization functional in terms of the *B*-spline expansion coefficients reduces to a quadratic form which has a convolutional structure [13]. In our tensor notation, it is represented by

$$\mathcal{C}_{x_1y_1z_1} \cdot \mathcal{T}^{x_1y_1z_1}_{xyz} \cdot \mathcal{C}^{xyz} = \lambda\mathcal{I}^j \cdot \mathcal{C}_{x_1y_1z_1} \cdot \mathcal{P}^{x_1}_{jx} \cdot \mathcal{Q}^{y_1}_{jy} \cdot \mathcal{R}^{z_1}_{jz} \cdot \mathcal{C}^{xyz} \tag{6}$$

where $\mathcal{I}^j \in J \subseteq \mathbb{R}^{N_p}$ is a vector with all components equal to one. In 3-D with $p = 2$, $N_p = 6$. The set of tensors $\mathcal{P}^{x_1}_{jx}, \mathcal{Q}^{y_1}_{jy}, \mathcal{R}^{z_1}_{jz}$ represent N_p d-tuples of discrete separable convolution transforms along corresponding directions of the reconstruction grid [13].

Similarly to the case of matrices and vectors, we may compute the derivatives of the cost function with respect to the components of the *B*-spline tensor $\mathcal{C}^{xyz}$ [17]. After setting these derivatives to zero, we end up with the following linear system of equations:

$$\left(\mathcal{B}^{x_1y_1z_1}_k\mathcal{B}^k_{xyz} + \mathcal{T}^{x_1y_1z_1}_{xyz}\right) \cdot \mathcal{C}^{xyz} = \mathcal{B}^{x_1y_1z_1}_k \cdot \mathcal{F}^k \tag{7}$$

with the further tensor factorization

$$\mathcal{B}^{x_1y_1z_1}_k\mathcal{B}^k_{xyz} = \left(\mathcal{E}^{x_1}_k \cdot \mathcal{G}^{y_1}_k \cdot \mathcal{H}^{z_1}_k\right) \cdot \left(\mathcal{E}^k_x \cdot \mathcal{G}^k_y \cdot \mathcal{H}^k_z\right). \tag{8}$$

Note that the latter expression, which corresponds to the least-squares part of the cost function, involves a sum of outer products of first-order tensors. This type of decomposition is known in the field of multilinear algebra as CANonical DECOMPosition (CANDECOMP) [18], [19].

IV. Optimized Solver

At that point, we could, in principle, rely on the computational strategy proposed in [13], adapting it to the present tensor formalism for $d \geq 2$. Specifically, we might solve (7) by applying a tensor multigrid solver [17] which exploits the inherent multiresolution properties of the *B*-spline basis for the specification of appropriate restriction and prolongation operators [13], [16]. This allows to compute the solution with nearly linear complexity with respect to the number of grid nodes.

However, the approach proposed in [13] as well as the approach presented in [15] requires explicit computation of matrix coefficients. This may create a problem due to the fact that despite the extreme sparsity of the system tensor, the number of nonzero coefficients in higher dimensions can be so large as to exceed the storage capacities of ordinary desktop computers. For example, in four dimensions with a moderate reconstruction grid size of 128 × 128 × 128 × 16 (33 554 432 unknown *B*-spline coefficients), the total number of nonzero entries is 69 080 710 400 which is about 257 GB in single precision format. By taking advantage of symmetries, the storage requirement for the system matrix can be reduced to about 30 GB, in the best case, while requiring ~287 GB memory transfers, which still remains demanding.

For this reason we propose an alternative approach, based on computational tensor algebra as described in [17]. The idea is to fully exploit the tensor structure of (7) and (8). This approach overcomes the storage problem by efficiently recomputing the required tensor components on the fly.

Specifically, we propose to solve (7) by using a Tensor Krylov Solver (Appendix B), which is based on the iterative computation of the tensor product

$$\left(\mathcal{B}^{x_1y_1z_1}_k\mathcal{B}^k_{xyz} + \mathcal{T}^{x_1y_1z_1}_{xyz}\right) \cdot \mathcal{C}^{xyz} \tag{9}$$

where $\mathcal{C}^{xyz}$ is the current estimate of the *B*-spline coefficients with index (x, y, z). The crucial point in our implementation is to compute the above tensor product, which actually represents the gradient of criterion (2) with respect to $\mathcal{C}$, as efficiently as possible. First, we observe that the "regularization term" $\mathcal{T}_{xyz}^{x_1y_1z_1}\cdot\mathcal{C}^{xyz}$ is completely specified by a set of $(N_p \times d)$ 1-D FIR filters, with impulse responses of length $h = 2n + 1$ (typ., $h = 7$ for cubic splines). The cost of the regularization filtering in the time-domain is therefore $(N_p \times d)P(2h-1)$, taking advantage of separability, where P is the total number of the grid nodes (for 3-D: $P = N_X N_Y N_Z$). Note that, depending on the degree of the used *B*-spline basis, the number of filters and the size of the grid, it can be more efficient to implement the regularization filtering in the frequency domain using the fast Fourier transform (FFT). In 3-D and 4-D with the use of cubic *B*-splines the time-domain filtering offers a higher performance than FFT-based filtering, which is exploited in the current work.

To evaluate the second, "least-squares" part $\mathcal{B}_k^{x_1y_1z_1}\mathcal{B}_{xyz}^k \cdot \mathcal{C}^{xyz}$, we make use of (8) and rewrite the tensor product as

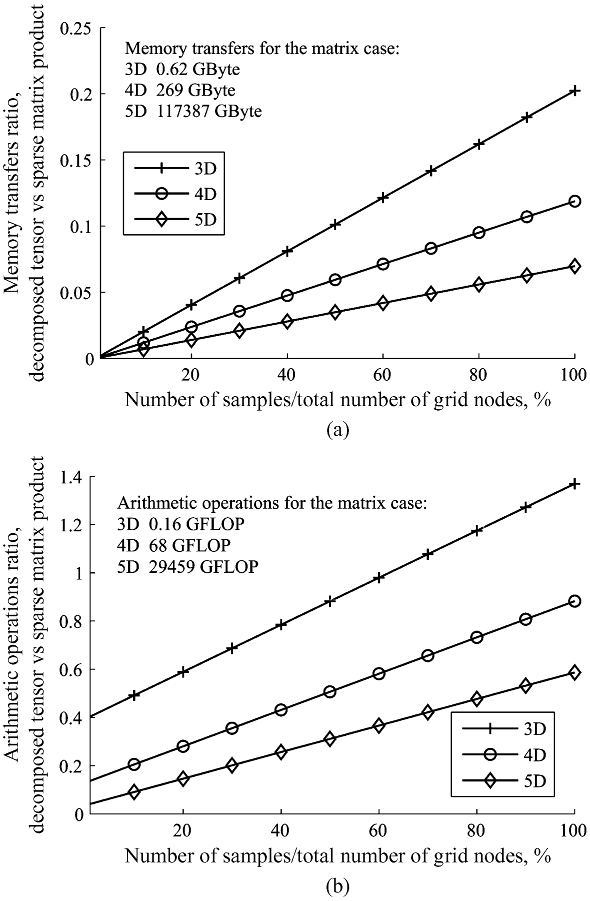
$$\left(\mathcal{E}_k^{x_1}\cdot\mathcal{E}_x^k\right)\cdot\left(\mathcal{G}_k^{y_1}\cdot\mathcal{G}_y^k\right)\cdot\left(\mathcal{H}_k^{z_1}\cdot\mathcal{H}_z^k\right)\cdot\mathcal{C}^{xyz} \qquad (10)$$

For each k, a nonzero block of products $(\mathcal{E}_k^{x_1}\cdot\mathcal{E}_x^k), (\mathcal{G}_k^{y_1}\cdot\mathcal{G}_y^k), (\mathcal{H}_k^{z_1}\cdot\mathcal{H}_z^k)$ represents a separable transformation applied to the corresponding dimension of a small hypercube in $\mathcal{C}$. The width of the d-dimensional hypercube is $n + 1$ and its bounds are dependent upon the spatial location of the given data sample. Thus, for computing (10), we successively loop through the K available samples, loading the corresponding kth hypercube from $\mathcal{C}$, applying to it the set of d 1-D separable transforms, and finally incrementing the respective hypercube in the resulting tensor.

Thanks to the decomposability of (10), we can avoid storing the system coefficients altogether. The components of tensor factors $\mathcal{E}_x^k, \mathcal{G}_y^k, \mathcal{H}_z^k$ in the decomposition can be recomputed in each iteration with a moderate cost. For a cubic *B*-spline model this cost is about 10 dK per iteration.

The advantages of the proposed, matrix-free computational scheme are the following. First, its complexity in terms of both memory transfers and arithmetic computations is linearly dependent upon the number of measurements (K), while the penalty incurred by the increase of the number of dimensions remains manageable (a factor $(n+1)^d$ instead of $\prod_{i=1}^{d} N_i(n+1)$ in the sparse matrix-based approach [13], where N_i are the extents of the grid). Second, the processing of dense multidimensional sub-blocks provides good data locality that fits well the requirements of current hardware with cache-based architecture. Third, computations can be efficiently parallelized by distributing signal measurements between multiple computational units.

Fig. 1 compares the efficiency of our method to that of an explicit implementation using sparse matrix multiplication. The quantities of interest are the memory transfers and number of operations required for the evaluation of the gradient (9). Note that in dimensions higher than three, the benefit of the tensor method is significant even for a quite large amount of signal measurements; this is due to the weaker exponential dependency on the number of dimensions. This property of the proposed method can allow efficient reconstruction of large 3-D/3-D+time and higher dimensional data (e.g. multidimensional spectral data, physical tensor fields) encountered in recent developments in sensing technologies.



(a)

(b)

Fig. 1. Relative requirements of decomposed tensor versus sparse matrix computation of the product (9) in single precision for memory transfers (a) and arithmetic computations (b). Reference numbers in GByte and GFLOP correspond to the sparse matrix case and do not depend on the number of data samples. Reconstruction grid has 64 nodes in all dimensions. Note that the higher the number of dimensions, the more advantageous is the use of the tensor product decomposition.

We did observe some degradation of the speed of convergence of the Krylov iterator when significant amounts of smoothing are required, even though the underlying linear system remains well conditioned. A typical example is shown in Fig. 2, which presents a result of the reconstruction of a 3-D CT data from 20% of samples with highest Laplacian values. The tradeoff factor in this experiment is 0.0034. In this particular case, we used the tensor conjugate gradient iterator (TCG) (see Appendix B), which was run for 20 iterations, until the relative residual norm reached 1.7×10^{-3}. Fig. 2(b) documents the convergence rate. In this case, the solver failed to compensate for the missing data in the regions with the lowest spatial variance, which is manifested by the presence of black spots in the reconstructed image. To overcome these difficulties, we propose to initialize the solver by providing it low frequency components of the solution. To this end, we employ a simple multiscale-based initialization. First, we obtain an approximation of the solution on the coarsest scale. Then, we interpolate the approximation to the next finer scale by means of the two scale relation filtering [16]. By successive transfer from coarse to the fine, we finally obtain the signal approximation that is used as initial condition for the fine-scale

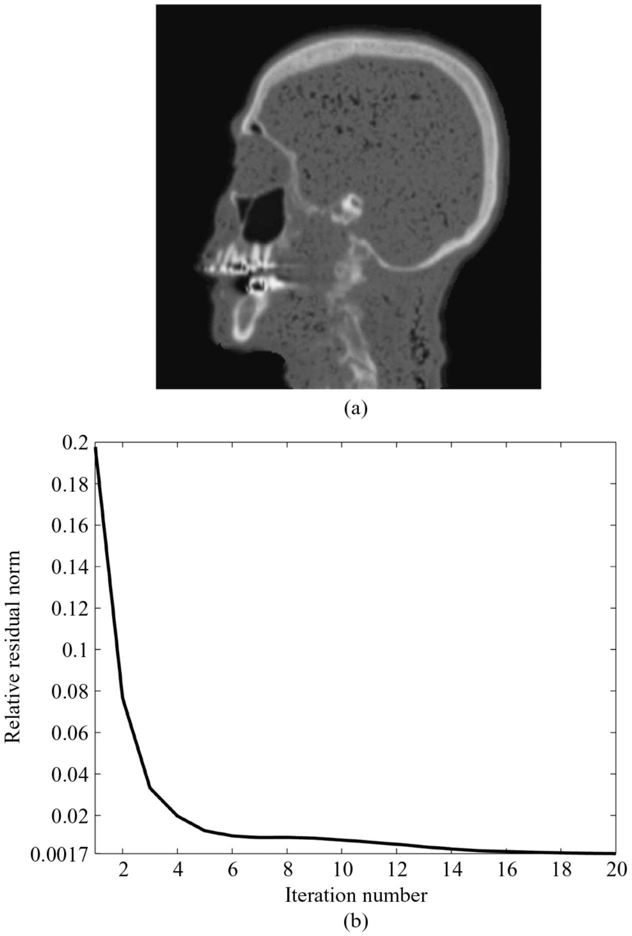


Fig. 2. Reconstruction of a 3-D data set from 20% of samples with highest Laplacian by applying non-preconditioned TCG iterator. (a) 2-D slice taken from the reconstructed image $(\mathrm{RMSE}(\%) = 3.63\%)$. (b) Convergence history of the solver. Note the presence of black spots in the reconstructed image which are located in the regions with lowest spatial variance.

solver. Note that, in contrast to [13], our multiscale approach does not require explicit computation and storing of the system coefficients. Instead, we use on the fly downsampling of the sparse factors in the decompositions (8) and (6), which allows us to avoid extra storage by an additional relatively small computational cost. Our practical experience showed that the presented initialization scheme is very effective. Fig. 3 presents a result of the multiscale-based initialized iteration for the same data set presented above. In this case, we used 3 coarser scales for initialization: 2 iterations on the third scale, 2 iterations on the second scale and 5 iterations on the first scale. Using this initial solution, the TCG solver was run for 6 iteration until the relative residual norm reached 1.9×10^{-3} value. The resulting solution has a root mean square error $\mathrm{RMSE} = 100 \cdot (\|I_0 - I_r\|/\|I_0\|) = 0.88\%$, where I_0 is the original image, I_r is the reconstructed image. Visually, the result of reconstruction is nearly indistinguishable from the original data.

Note that the presented multiscale initialization scheme can be used as a preconditioner for the TCG iterator. However, our practical experiments showed that this algorithm even used alone already offers good quality solutions.

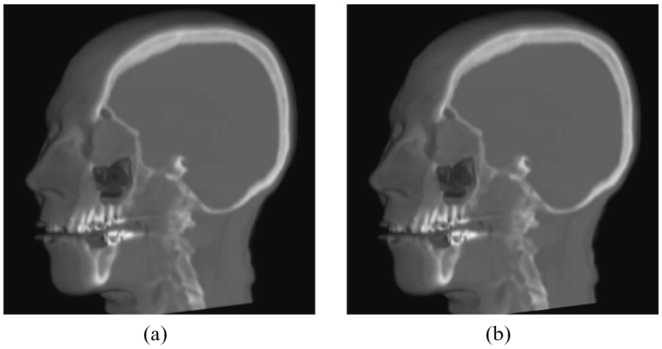


Fig. 3. Reconstruction of a 3-D data set from 20% of samples with highest Laplacian by applying TCG iterator with multiscale-based initialization. (a) Original image. (b) Reconstructed image $(\mathrm{RMSE} = 0.88\%)$.

V. Results

The proposed tensor-based algorithm was implemented in MathOberon, a programming language offering multidimensional abstractions for readable, compact and efficient parallel implementations [20]. The code was tested on a PC with Intel Core 2 Quad Q9400 2.67 GHz CPU equipped with 2 GB of DDR2 800 MHz memory.

A. *Evaluation in 3-D*

We evaluated the quality of reconstruction and computational performance of the proposed algorithm implementation on standard uniform 3-D datasets which are available online[1] The data were sampled by taking 20% of samples with the highest values of the Laplacian, similarly to the protocol in [15]. Table I summarizes the results of the evaluation.

In these experiments, in addition to RMSE, we used two error measures $\mathrm{RMSE}_1(\%) = 100 \cdot (\|I_0 - I_r\|/\sqrt{N} \cdot \max(I_0))$ and $\mathrm{RMSE}_2(\%) = (100/\max(\|\mathbf{g}_0\|)) \cdot \sqrt{\sum_{i=1}^{N} \|\mathbf{g}_i - \mathbf{g}_{0i}\|^2/N}$ where N is the total number of samples, $\mathbf{g}_0$ is the gradient field computed from the original data, $\mathbf{g}$ is the gradient field computed from the reconstructed data, $\mathbf{g}_{0i}$ and $\mathbf{g}_i$ are the values of the corresponding fields evaluated at the ith point of the grid. All error measures are computed using all the original uniform data samples. RMSE_1 is used to compare our results with those presented in [15] (values in brackets); to our knowledge, this latter algorithm is the most efficient published approach for reconstructing irregular scalar data in 3-D. We used the proposed multiscale initialization scheme with 8 TCG iterations on each coarser scale, which was sufficient to get a good initialization for the finest scale iteration. For all datasets in the tests, the minimal dimension extent for coarser scales was 16 nodes, which resulted in 3–4 scales. On the finest scale the TCG solver was run for 10 iterations. A further increase of the number of iterations did not lead to significant improvement of the error values. As the regularizer functional we used Duchon's semi-norm of order $p = 2$. The values of the regularization tradeoff λ were chosen by minimizing a 3-fold cross validation cost individually for each dataset. Fig. 4 displays the behavior of the cross validation cost in an interval around the optimum for the Tooth dataset. In the performed

[1] Available online at http://www.volvis.org

TABLE I
RESULTS OF THE VALIDATION OF THE PROPOSED ALGORITHM ON STANDARD 3-D DATASETS SAMPLED USING 20% OF HIGHEST LAPLACIAN SAMPLES

Dataset	Grid	λ	$RMSE$ %	$RMSE_1$ %	$RMSE_2$ %	Time min
Engine	$256 \times 256 \times 128$	$3.10 \cdot 10^{-3}$	2.75	0.58 (0.94)	1.40	1.12 (1.28)
Tooth	$256 \times 256 \times 160$	$6.25 \cdot 10^{-3}$	1.24	0.29 (0.18)	1.54	1.31 (1.88)
CT-Chest	$384 \times 384 \times 240$	$1.78 \cdot 10^{-3}$	1.41	0.32 (0.60)	1.09	4.38 (5.08)
Carp	$256 \times 256 \times 512$	$4.13 \cdot 10^{-4}$	1.96	0.30 (0.25)	0.34	4.81 (5.73)

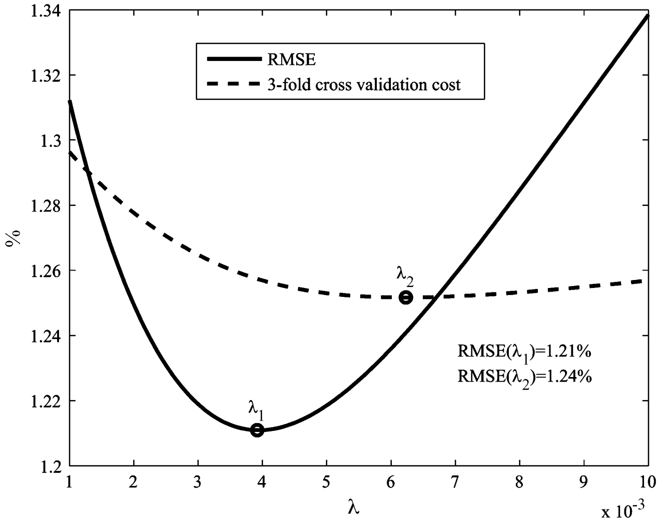


Fig. 4. Three-fold cross validation cost and RMSE evaluated for the Tooth dataset sampled using 20% of highest Laplacian samples.

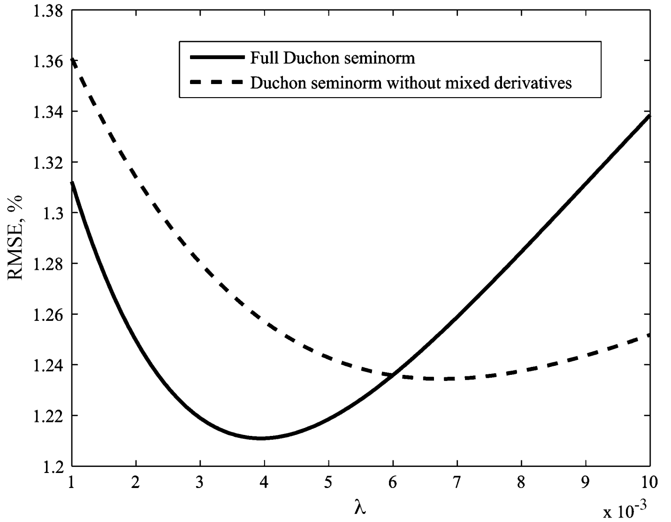


Fig. 6. Evaluation of RMSE for full Duchon's semi-norm and the regularizer (11) used for the reconstruction of the Tooth dataset.

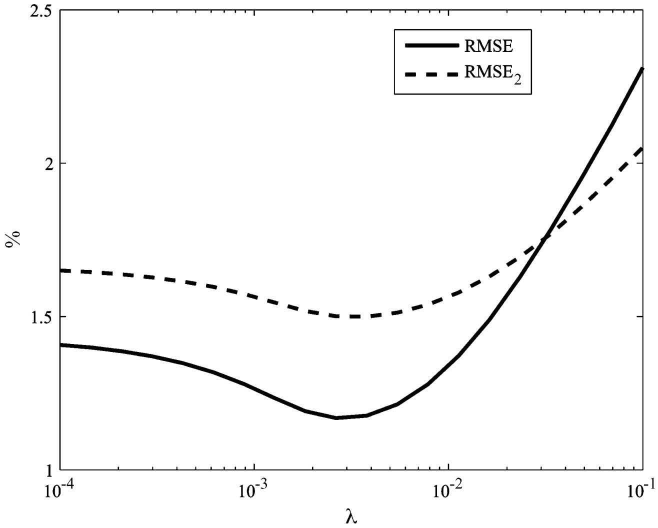


Fig. 5. Evaluation of RMSE and the gradient-based error measure RMSE_2 for the Tooth dataset sampled using 20% of highest Laplacian samples.

experiments the optimum of the cross validation cost for all datasets in the tests was rather close to the optimum of the true error. For finding the location of the optimum, we used the golden section search technique; 6–15 evaluations of the cost function were sufficient to identify the minimum.

Fig. 5 presents a comparison of the behavior of RMSE and the gradient-based error measure RMSE_2 with the change of the tradeoff parameter λ for the Tooth dataset. In this particular case, the optimum locations of these functions almost coincide.

Additionally, we performed tests using a regularizer of the form

$$R(s)=\lambda \int\limits_{\mathbb{R}^d} \left(\frac{\partial^2 s(\boldsymbol{t})}{\partial t_1^2}\right)^2+\left(\frac{\partial^2 s(\boldsymbol{t})}{\partial t_2^2}\right)^2+\left(\frac{\partial^2 s(\boldsymbol{t})}{\partial t_3^2}\right)^2 \mathrm{d}t_1\mathrm{d}t_2\mathrm{d}t_3 \tag{11}$$

which is Duchon's semi-norm of order $p = 2$ without mixed derivative terms. In contrast to the results reported in [15], our experiments showed that the reconstruction error using (11) was larger or in the best case comparable to the error obtained with full Duchon's semi-norm. Fig. 6 presents a comparison of the reconstruction error for full Duchon's semi-norm versus the error for regularizer (11) both evaluated for the Tooth dataset.

The results of our 3-D evaluation indicate that our implementation is at least as good or better in terms of reconstruction quality as the block-wise Multigrid-based implementation from [14] when using a single CPU core and a comparable time budget to do a fair comparison. The execution speed of our implementation was increased by 2 with the use of two cores. A further increase of the number of cores did not yield an additional performance improvement; this was due to the low speed of the memory bus on our hardware. With the use of a faster memory/bus architecture, a better scalability and thus higher performance is probably achievable with the same CPU.

B. Evaluation in 4-D

One of the main advantages of our algorithm over existing approaches is the possibility of efficient reconstruction of large datasets in dimensions higher than three even when the computational resources are limited. We tested our algorithm

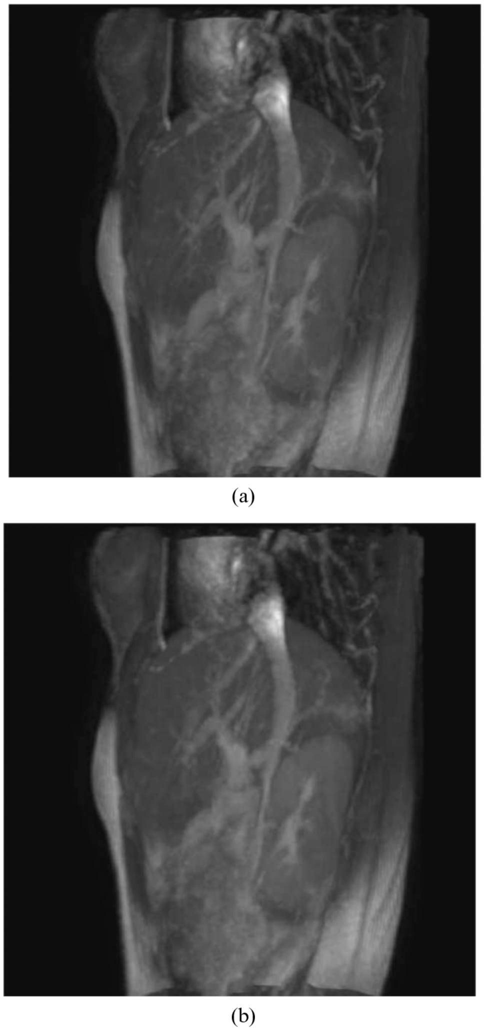


Fig. 7. Example of reconstruction of a 4-D MRI dataset from 30% of randomly selected samples. Presented images represent a maximum intensity projection at a single time point. (a) Original data. (b) Reconstructed data ($\mathrm{RMSE} = 8.12\%$, $\mathrm{RMSE}_1 = 1.29\%$).

implementation on a 4-D (space-time) MRI dataset kindly provided by the authors of [21]. We performed an evaluation with sampling the original data by taking 30% of random samples which for the grid size $184 \times 140 \times 30 \times 23$ corresponds to more than 5 million data points. In this experiment, we used tensor B-spline basis functions in (1) together with Duchon's semi-norm of order $p = 2$ and $d = 4$ as regularization functional. The tradeoff value was chosen in the same way as in 3-D tests by optimizing a three-fold cross validation cost and was set to $6.31 \cdot 10^{-4}$. We used the proposed multiscale initialization scheme with 4 coarser scales and 8 TCG iterations on each scale. The fine-scale TCG iterator was run for 20 iterations. The reconstruction time with two CPU cores was about 5.2 min. Fig. 7 presents a 3-D visualization of a single time frame from the obtained 4-D reconstruction. Fig. 8 shows RMSE computed at each time point of the data. Space-time visualization with a comparison of the obtained reconstruction with the original data is available online.[2]

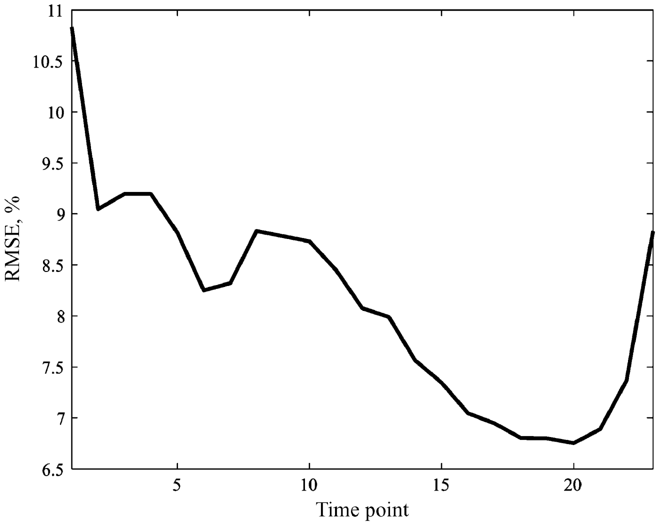


Fig. 8. RMSE computed at each time point of the reconstructed 4-D MRI data.

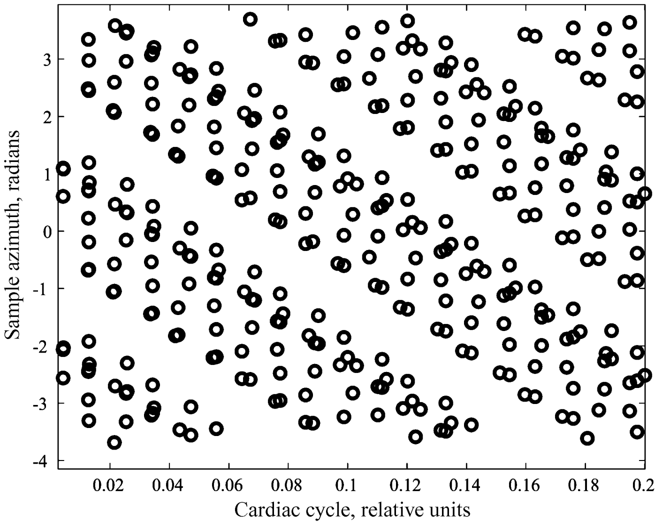


Fig. 9. Distribution of the azimuth versus cardiac cycle for a data set acquired by a fast continuously rotating ultrasound transducer.

C. Application to 4-D Medical Ultrasound

We applied the proposed algorithm to a concrete imaging problem: the reconstruction of 4-D (space-time) ultrasound data acquired using a fast continuously-rotating ultrasound transducer [22], [23]. The transducer, which is based on a conventional linear array with harmonic capabilities, samples 3-D volume, while continuously rotating at the high speed of 8 rotations per second, which allows temporal resolution of about 16 volumes per second [22]. A cone-shaped volume is scanned over several seconds. Since there is no synchronization between the heart rate and the continuous rotation, the acquired data is irregularly distributed over cardiac phase and angle, as schematically shown on Fig. 9. Note that in this case when the data is sampled irregularly in both space and time, the ability of the reconstruction method to account for the data coherence in all dimensions is an important prerequisite for achieving good quality reconstructions. Such ability of the proposed method

[2] Available online at http://www.computational.ch/downloads/mrirec4d.avi

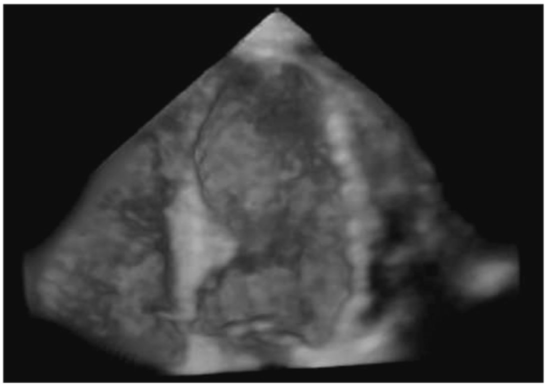


Fig. 10. Three-dimensional visualization of a single time frame from the reconstruction of a 4-D arbitrarily sampled data acquired by a fast continuously rotating ultrasound transducer.

distinguishes it from the conventional way of frame-by-frame reconstruction.

Our algorithm implementation was applied to a dataset kindly provided by the authors of [23]. This particular dataset represents the motion of the left ventricle of the heart. It consisted of more than 12 million ultrasound samples. The chosen reconstruction grid size is $128 \times 128 \times 128 \times 16$, with 16 nodes over the time axis corresponding to one cardiac cycle. The spacing of the reconstruction grid was chosen according to the sampling characteristics of the used 2-D rotating transducer. In this case, the spatial grid $128 \times 128 \times 128$ corresponds to a pixel size of about 1 mm, which is a typical axial resolution in conventional cardiac ultrasound imaging. As 4-D regularization functional, we used Duchon's semi-norm of order $p = 2$. The tradeoff value was chosen by optimizing a 3-fold cross validation cost and was set to $6.14 \cdot 10^{-2}$. For multiscale initialization, we used 4 coarser scales with 8 TCG iterations on each. On the finest scale, the TCG solver was stopped at the 30th iteration at a relative residual norm 5.4×10^{-3}. With the use of 2 CPU cores, the total reconstruction time was about 15 min. Fig. 10 shows a 3-D visualization of a single time frame taken from the obtained reconstruction. The reconstructed data was visually assessed by experienced doctors and approved to be of a good quality. Space-time visualization of the reconstructed left ventricle dataset is available online.[3]

VI. Conclusion

We presented a tensor-based approach for the efficient reconstruction of high-dimensional images from large sets of arbitrarily-sampled measurements. We proposed a tensor algebra framework to analyze the structure of the *B*-spline reconstruction problem which led to the identification of a series of 1-D factorizations of the system tensor. We then used this representation to develop an iterative solver that is computationally and memory efficient. The critical step of our algorithm is the computation of a tensor product (update of the solution), which, due to the inherent, sparse CANDECOMP structure of the problem, can be evaluated with a complexity that is proportional to the number of measurements with a dependence on the number of physical dimensions that is significantly less than previously published solutions (reduction of the impact of the curse of dimensionality). The proposed algorithm can be efficiently parallelized and implemented, exploiting all currently available computing technologies such as single instruction multiple data (SIMD), multi core, clusters of PCs and general purpose GPU (GPGPU). In particular, we have built an efficient multicore implementation of the proposed algorithm.

We first tested our approach on standard 3-D images using a nonuniformly subsampled portion of the data as input. We found it to be competitive with the best available Multigrid-based method [15] in the sense of providing a better or comparable reconstruction quality for a given computational budget. The proposed algorithm uses much less memory and allows computation of relatively large problems with the use of only 2 GB of memory available on our hardware. The implementation is parallel and can benefit from the use of multiple CPU cores. We also showed that the algorithm could handle more demanding tasks, such as the reconstruction of 4-D (space-time) dynamic MRI, and provide satisfactory results. In particular, we successfully applied it to a concrete large-scale imaging problem: the reconstruction of a 4-D (space-time) ultrasound signal from a large set of noisy arbitrarily sampled data acquired by a fast continuously rotating ultrasound transducer. Our ongoing work is directed towards further acceleration of the method by the use of a GPGPU platform and cloud-based distributed computations for even faster reconstructions.

Appendix A
Einstein's Tensor Notation

The tensor formulation of a linear approximation problem can offer advantages over the commonly-used matrix-vector formalism, especially when the system matrix involves outer products. By retaining the physical coordinate structure of the problem, it typically simplifies the analysis and specification of solutions in dimensions greater than one. Note that the role of the tensor indexing symbols is special for they each encode a specific (physical) dimension of the data. In Einstein's convention, when an index variable appears twice in a single term, once in an upper (superscript) and once in a lower (subscript) position, it implies that we are summing over all of its possible values. We will exemplify the notation by linking it to classical linear algebra. For instance, let $\mathcal{U}^n$ and $\mathcal{V}^n$ be two N-column vector tensors with $n \in \{1, \ldots, N\}$ corresponding to the vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^N$. Then,

$$\mathcal{U}_n \mathcal{V}^n = \mathbf{u}^T \mathbf{v}$$

with $\mathcal{U}_n = (\mathbf{u}^T)_n$ and $\mathcal{V}^n = (\mathbf{v})^n$. Similarly, given a $M \times N$ matrix tensor $\mathcal{A}_n^m$ and a N-column vector tensor $\mathcal{U}^n$, we represent the matrix product $\mathbf{Au}$ as

$$\mathcal{A}_n^m \mathcal{V}^n = (\mathbf{Av})^m.$$

Equivalently, we have that

$$\mathcal{A}_m^n \mathcal{V}_n = \mathcal{V}_n \mathcal{A}_m^n = (\mathbf{v}^T \mathbf{A}^T)_m$$

which corresponds to the transposed matrix product $\mathbf{u}^T \mathbf{A}^T$. In the present work, the spatial indexing is explicitly 3-D with xyz

[3] http://www.computational.ch/downloads/echorec4d.avi

(resp., $x_1y_1z_1$) taking the role of n (resp., m) in the formulas above.

Appendix B
Tensor Conjugate Gradient Solver

Krylov subspace solvers are among the most effective techniques for solving large linear systems of equations [24]. They operate iteratively by evaluating the residual of the system and then performing an update of the solution along an appropriate search direction. In our case, the problem reduces to solving the multi-linear system of (7) whose generic form is $\mathcal{A}^{x_1y_1z_1}_{xyz} \cdot \mathcal{C}^{xyz} = \mathcal{B}^{x_1y_1z_1}$. Since the corresponding system tensor is positive-definite, we apply a tensor variant of the Conjugate Gradient method—a special instance of a Krylov subspace solver—whose pseudo code is provided for completeness

$$\begin{aligned} \mathcal{R}^{xyz} &= \mathcal{B}^{xyz} - \mathcal{A}^{xyz}_{x_1y_1z_1} \cdot \mathcal{C}^{x_1y_1z_1} \qquad \text{(initial residual)} \\ \mathcal{P}^{xyz} &= \mathcal{R}^{xyz} \qquad \text{(initial search directions)} \\ \rho &= \mathcal{R}_{xyz}\mathcal{R}^{xyz}. \end{aligned}$$

Repeat loop until convergence

$$\begin{aligned} \mathcal{Q}^{xyz} &= \mathcal{A}^{xyz}_{x_1y_1z_1} \cdot \mathcal{P}^{x_1y_1z_1} \qquad \text{(tensor computation)} \\ \alpha &= \frac{\rho}{(\mathcal{Q}_{xyz}\mathcal{P}^{xyz})} \\ \mathcal{C}^{xyz} &= \mathcal{C}^{xyz} + \alpha \cdot \mathcal{P}^{xyz} \qquad \text{(solution update)} \\ \mathcal{R}^{xyz} &= \mathcal{R}^{xyz} - \alpha \cdot \mathcal{Q}^{xyz} \qquad \text{(residual update)} \\ \rho_1 &= \mathcal{R}_{xyz}\mathcal{R}^{xyz} \\ \mathcal{P}^{xyz} &= \mathcal{R}^{xyz} + \left(\frac{\rho_1}{\rho}\right) \cdot \mathcal{P}^{xyz} \qquad \text{(direction update)} \\ \rho &= \rho_1 \end{aligned}$$

End of loop.

Note that the computational cost of the majority of steps above is small; that, is linear in the number of unknowns (simple loop/summation over the tensor indexing variable xyz). The only step of the algorithm that is computer-intensive for large scale problems is the tensor computation that yields $\mathcal{Q}^{xyz}$. This is the part that is specific to our implementation and that is achieved according to the strategy outlined in Section IV.

Acknowledgment

The authors would like to thank P. Cattin and B. Krenning for providing 4-D data sets used for evaluation of presented reconstruction algorithm.

References

[1] G. Wahba, *Spline Models for Observational Data*. Philadelphia, PA: SIAM, 1990.

[2] M. D. Buhmann, *Radial Basis Functions: Theory and Implementations*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[3] J. Duchon, *Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces*, ser. Lecture Notes in Mathematics. Berlin, Germany: Springer, 1977, vol. 571.

[4] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, Jun. 1989.

[5] R. K. Beatson, J. B. Cherrie, and D. L. Ragozin, "Fast evaluation of radial basis functions: Methods for four-dimensional polyharmonic splines," *SIAM J. Math. Anal.*, vol. 32, no. 6, pp. 1272–1310, 2001.

[6] D. Achlioptas, F. Mcsherry, and B. Schoelkopf, "Sampling techniques for kernel methods," in *Annual Advances in Neural Information Processing Systems 14: Proceedings of the 2001 Conference*. Cambridge, MA: MIT Press, 2001, pp. 335–342.

[7] H. Zhang, M. Genton, and P. Liu, 2004, Compactly supported radial basis function kernels [Online]. Available: http://www4.stat.ncsu.edu/~hzhang/paper/csrbf.pdf

[8] E. Larsson and B. Fornberg, "Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions," *Comput. Math. Appl.*, vol. 49, pp. 103–130, 2003.

[9] R. K. Beatson, W. A. Light, and S. Billings, "Fast solution of the radial basis function interpolation equations: Domain decomposition methods," *SIAM J. Sci. Comput.*, vol. 22, no. 5, pp. 1717–1740, 2000.

[10] M. Bertram, J. C. Barnes, B. Hamann, K. I. Joy, H. Pottmann, and D. Wushour, "Piecewise optimal triangulation for the approximation of scattered data in the plane," *Comput. Aided Geom. Des.*, vol. 17, no. 8, pp. 767–787, 2000.

[11] Y. Jang, R. P. Botchen, A. Lauser, D. S. Ebert, K. P. Gaither, and T. Ertl, "Enhancing the interactive visualization of procedurally encoded multifield data with ellipsoidal basis functions," in *Comput. Graph. Forum*, 2006, vol. 25, no. 3, pp. 587–596.

[12] D. Juba and A. Varshney, Modelling and rendering large volume data with Gaussian radial basis functions Univ. Maryland, Tech. Rep, 2007.

[13] M. Arigovindan, M. Suehling, P. Hunziker, and M. Unser, "Variational image reconstruction from arbitrarily spaced samples: A fast multiresolution spline solution," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 450–460, Apr. 2005.

[14] E. Vucini, T. Moeller, and M. E. Groeller, "Efficient reconstruction from non-uniform point sets," *Visual Computer*, 2008.

[15] E. Vuçini, T. Möller, and M. E. Gröller, "On visualization and reconstruction from non-uniform point sets using b-splines," in *Computer Graphics Forum*, Jun. 2009, vol. 28, no. 3, pp. 1007–1014.

[16] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 16, no. 6, pp. 22–38, Nov. 1999.

[17] O. Morozov and P. Hunziker, "Solving tensor structured problems with computational tensor algebra," *CoRR*, vol. abs/1001.5460, 2010.

[18] R. A. Harshman, "Foundations of the PARAFAC procedure: Model and conditions for an "explanatory" multi-mode factor analysis," in *UCLA Working Papers in Phonetics*, 1970, vol. 16, pp. 1–84.

[19] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, Sep. 2009.

[20] F. Friedrich and J. Gutknecht, "Array-structured object types for mathematical programming," *Lecture Notes in Computer Science*, vol. 4228, pp. 195–210, 2006.

[21] M. Siebenthal, G. Szekely, U. Gamper, P. Boesiger, A. Lomax, and P. Cattin, "4D MR imaging of respiratory organ motion and its variability," *Phys. Med. Biol.*, vol. 52, pp. 1547–1564, 2007.

[22] K. Djoa and N. de Jong, "A fast rotating scanning unit for real-time three-dimensional echo data acquisition," *Ultrasound Med. Biol.*, vol. 26, no. 5, pp. 863–869, 2000.

[23] M. Voormolen and B. Krenning, "A new transducer for 3d harmonic imaging," in *Proc. IEEE Ultrason. Symp.*, 2002, pp. 1261–1264.

[24] Y. Saad and H. A. van der Vorst, "Iterative solution of linear systems in the 20th century," *J. Comput. Appl. Math.*, vol. 123, no. 1–2, pp. 1–33, 2000.