Hierarchical Tucker Tensor Optimization - Applications to Tensor Completion

Curt Da Silva
Seismic Laboratory for Imaging and Modeling
& Department of Mathematics
University of British Columbia
Email: curtd@math.ubc.ca

Felix J. Herrmann
Seismic Laboratory for Imaging and Modeling
& Department of Earth and Ocean Sciences
University of British Columbia
Vancouver, BC, Canada
Email: fherrmann@eos.ubc.ca

***Abstract*—In this work, we develop an optimization framework for problems whose solutions are well-approximated by *Hierarchical Tucker* (HT) tensors, an efficient structured tensor format based on recursive subspace factorizations. Using the differential geometric tools presented here, we construct standard optimization algorithms such as Steepest Descent and Conjugate Gradient for interpolating tensors in HT format. We also empirically examine the importance of one's choice of data organization in the success of tensor recovery by drawing upon insights from the matrix completion literature. Using these algorithms, we recover various seismic data sets with randomly missing sources.**

I. Introduction

Matrix completion has seen a large amount of development in recent years, resulting in algorithms that are very space and time efficient and theoretical guarantees which closely agree with empirical recovery rates. The success of completing a matrix with randomly missing entries via rank-minimizing optimization is a result of assuming a low-rank model on the underlying solution, coupled with a subsampling operator that tends to increase the rank of the underlying matrix.

We use extended notions of low-rank in the case of interpolating a *tensor* with missing entries. Our model is a structured tensor format known as the *Hierarchical Tucker* (HT) format, which efficiently represents a high-dimensional tensor by means of a Kronecker splitting of subspaces, with the set of all such tensors parametrizing a smooth manifold in $\mathbb{R}^{n_1 \times n_2 \times \dots n_d}$. We extend the largely theoretical results of [1] by imposing a Riemannian metric on the resulting quotient manifold, from which we can derive the Riemannian gradient and develop solvers for minimizing smooth functions defined on this manifold. We will use these efficient, SVD-free solvers in order to interpolate tensors that have a large portion of their entries removed and empirically examine the effect of data organization on the success of recovery for our test seismic cases. Our manifold-optimization approach for completing tensors with missing entries follows a similar spirit to [2]. We present the results of several interpolated seismic frequency slices and demonstrate our ability to recover tensors even amidst high levels of subsampling.

II. Hierarchical Tucker Tensor Format

An important choice of dimension separation, ensuring that the resulting HT tensor is low-rank, is that of a *dimension tree*.

Definition 1. A *dimension tree* for a $d-$dimensional tensor is a nontrivial binary tree such that

- The root node, t_{root}, has the label $\{1, \dots, d\}$
- The labels for the children of each non-leaf node form a partition of the parent's label, i.e.

$$t_l \sqcup t_r = t, \quad t \notin L$$

where t_l, t_r are the left and right children of the node t, respectively, and L is the set of all leaves of T.

Suppose that we have chosen a set of (positive integer) hierarchical ranks $(k_t)_{t \in T}$ assigned to each node of a dimension tree T, with $k_{\mathrm{t}_{\text{root}}} = 1$. Then we have the following

Definition 2. Let $\mathbb{R}_*^{n \times p}$ and $\mathbb{R}_*^{p \times q \times r}$ denote the set of all $n \times p$ matrices of full rank and $p \times q \times r$ 3-tensors of full *multilinear* rank, respectively.

A $d-$tensor X is said to be in *Hierarchical Tucker* format with associated dimension tree T and hierarchical ranks $(k_t)_{t \in T}$ if there exist parameter matrices/tensors $x = (U_t, B_t)$ with $U_t \in \mathbb{R}_*^{n_t \times k_t}$, $B_t \in \mathbb{R}_*^{k_r \times k_l \times k_t}$ such that $\phi(x) = X$, where

$$\begin{aligned} \operatorname{vec} \phi(x) &= (U_{t_l} \otimes U_{t_r})(B^{(k_l, k_r)}) && t = \mathrm{t}_{\text{root}} \\ U_t &= (U_{t_l} \otimes U_{t_r})(B^{(k_l, k_r)}) && t \notin L \cup \mathrm{t}_{\text{root}} \end{aligned} \tag{1}$$

where k_t is the rank associated to node t and k_l, k_r are the ranks associated to nodes t_l, t_r, respectively. We say that the parameter matrices x are in *Orthogonal Hierarchical Tucker* (OHT) format if (U_t, B_t) also satisfy

$$\begin{aligned} U_t^T U_t &= I_{k_t} \quad \text{for } t \in L \\ (B_t^{(k_l, k_r)})^T B_t^{(k_l, k_r)} &= I_{k_t} \quad \text{for } t \notin L \cup \mathrm{t}_{\text{root}} \end{aligned}$$

Let $\mathcal{H}_{\mathrm{T,k}}$ denote the set of all tensors expressible in HT format with dimension tree T and hierarchical ranks $(k_t)_{t \in T}$.

Note that the intermediate matrices U_t in (1) for $t \notin L$ do not need to be stored: only the matrices U_t for $t \in L$ and so-called *transfer tensors* B_t for $t \notin L$ need to be stored to specify the tensor X completely. Let

$$\mathcal{M} = \bigtimes_{t \in L} \mathbb{R}_*^{n_t \times k_t} \times \bigtimes_{t \in T \setminus L} \mathbb{R}_*^{k_{t_r} \times k_{t_l} \times k_t}$$

be the space of admissible HT parameters. ϕ given in (1) is a smooth function from $\mathcal{M}$ to its image $\mathcal{H}_{\mathrm{T,k}} \subset \mathbb{R}^{n_1 \times \dots n_d}$ that is *not* injective. From the optimization point of view, any optimization problem defined on $\mathcal{H}_{\mathrm{T,k}}$ and parametrized by $\mathcal{M}$ and (1) will have minimizers which are *not* isolated. We will characterize this non-uniqueness, and its remedy, below.

III. Quotient Geometry of the HT format

There is an ambiguity in the representative parameters for a given HT tensor X, which is characterized in [1] as follows. Let $\mathbf{G}$ be the Lie group

$$\mathbf{G} = \{\mathcal{A} = (A_t)_{t \in T} : A_t \in GL(k_t)\ A_{\mathrm{t_{root}}} = 1\}$$

acting on $\mathcal{M}$ via the right action

$$\theta_{\mathcal{A}}(U_t, B_t) := (U_t A_t, (A_{t_r}^{-1}, A_{t_l}^{-1}, A_t^T) \circ B_t)$$

where $(A_1, A_2, A_3) \circ C$ is the multilinear product that premultiplies C by A_i in the i-th dimension. Note that $\phi(x) = \phi(y)$ if and only if there exists a unique $\mathcal{A} \in \mathbf{G}$ such that $y = \theta_{\mathcal{A}}(x)$. The quotient manifold has a unique smooth structure such that $\pi : \mathcal{M} \to \mathcal{M} / \mathbf{G}$ is a smooth submersion. The quotient manifold $\mathcal{M} / \mathbf{G}$ is really our manifold of interest for the purposes of solving optimization problems, since each equivalence class $\pi(x)$ is identified with unique values of $\phi(x)$.

The authors in [1] introduce the following horizontal space

$$\mathcal{H}_x \mathcal{M} := \left\{ (U_t^h, B_t^h) : \begin{array}{c} (U_t^h)^T U_t = 0_{k_t \times k_t} \text{ for } t \in L \\ (B_t^h)^{(k_t)} Q_t (B_t^{(k_t)})^T = 0_{k_t \times k_t} \text{ for } t \notin L \cup \mathrm{t_{root}} \end{array} \right\} \tag{2}$$

where $Q_t = (U_{t_l}^T U_{t_l} \otimes U_{t_r}^T U_{t_r})$, which is shown to be invariant under the action of θ. Eq. (2) allows us to uniquely identify vector fields on $\mathcal{M} / \mathbf{G}$ with horizontal vector fields in $\mathcal{M}$.

For the purposes of interpolation, we are interested in the computing the best fit of our data within the space of HT models, which involves solving a corresponding optimization program on $\mathcal{H}_{\mathrm{T,k}}$. There is a large body of existing research on solving optimization problems on matrix manifolds (see [3] for a comprehensive introduction). Before we can develop such optimization methods, we must first specify a well-defined Riemannian metric on the quotient manifold $\mathcal{M} / \mathbf{G}$.

Fix $x = (U_t, B_t)$, $\eta_x = (\delta U_t, \delta B_t), \zeta_x = (\delta V_t, \delta C_t) \in \mathcal{H}_x \mathcal{M}$. Let $P_t = U_t^T U_t$ for each $t \in T \setminus \mathrm{t_{root}}$, Q_t as above, and let, by abuse of notation, $\delta B_t := \delta B_t^{(k_l, k_r)}$ and similarly for δC_t. One can show that, for the following inner product,

$$\begin{aligned} g_x(\eta_x, \zeta_x) := & \sum_{t \in T} \mathrm{tr}(P_t^{-1} \delta U_t^T \delta V_t) \\ & + \sum_{t \notin L \cup \mathrm{t_{root}}} \mathrm{tr}(P_t^{-1} (\delta B_t)^T Q_t \delta C_t) \\ & + \mathrm{vec}(\delta B_{\mathrm{t_{root}}})^T Q_{\mathrm{t_{root}}} \mathrm{vec}(\delta C_{\mathrm{t_{root}}}) \end{aligned} \tag{3}$$

it holds that $g_x(\eta_x, \zeta_x) = g_{\theta_{\mathcal{A}}(x)}(\eta_{\theta_{\mathcal{A}}(x)}, \zeta_{\theta_{\mathcal{A}}(x)})$ for every $\mathcal{A} \in \mathbf{G}$. Therefore the metric g restricted to vectors in the horizontal space does *not* depend on the representative point for the equivalence class, $x' \in \pi(x)$. Since each $U_t^T U_t$ is symmetric positive definite for each $t \in T \setminus \mathrm{t_{root}}$ and varies smoothly with x, it is easy to see that g_x varies smoothly with x as well. This yields a Riemannian metric that is well-defined on the quotient manifold $\mathcal{M} / \mathbf{G}$ (see 3.6.2 in [3]). Our optimization algorithm will then be implemented on the total space $\mathcal{M}$ rather than the abstract quotient $\mathcal{M} / \mathbf{G}$, with the understanding that points $x \in \mathcal{M}$ will represent their equivalence class $\pi(x) \in \mathcal{M} / \mathbf{G}$ (see [3] for more details).

Require: $x = (U_t, B_t)$, $Z \in \mathbb{R}^{n_1 \times \dots n_d}$
  $\delta U_{\mathrm{t_{root}}} \leftarrow Z$
  for each $t \in T \setminus L$, visiting each node before its children **do**
    $\delta U_{t_l} \leftarrow \frac{\partial U_t}{\partial U_{t_l}}^* \delta U_t$, $\delta U_{t_r} \leftarrow \frac{\partial U_t}{\partial U_{t_r}}^* \delta U_t$,
    $\delta B_t \leftarrow \frac{\partial U_t}{\partial B_t}^* \delta U_t$
  end for
  return $D\phi(x)^* Z = P_{\mathcal{H}_x \mathcal{M}}((\delta U_t)_{t \in L}, (\delta B_t)_{t \in T \setminus L})$

Fig. 1. Algorithm for computing $D\phi(x)^* Z$

When we restrict our parameter matrices to be in OHT, one can see that since $U_t^T U_t = I_{k_t}$ for every $t \in T \setminus \mathrm{t_{root}}$, and so the inner product (3) reduces to the standard Euclidean one. For this reason, and to ensure that the resulting projections on to $\mathcal{H}_x \mathcal{M}$ can be performed efficiently, we restrict our parameters $x = (U_t, B_t)$ to be OHT in the sequel. This is not a hindrance from a theoretical point of view, because any non-orthogonalized parameter set x can be efficiently orthogonalized via Proposition 3 to a parameter set x' such that $\phi(x) = \phi(x')$. It can be shown that the resulting quotient space of orthogonalized parameters is diffeomorphic to $\mathcal{H}_{\mathrm{T,k}}$.

A. Riemannian Gradient

Using this Riemannian metric, we can compute the Riemannian gradient of a smooth function $f : \mathcal{H}_{\mathrm{T,k}} \to \mathbb{R}$ as follows. Let $x \in \mathcal{M}$. Then by the fundamental theorem of linear algebra, since $\mathrm{im}\ D\phi(x) = T_{\phi(x)}\mathcal{H}$, $\ker D\phi(x)^* = T_{\phi(x)}^{\perp}$

Our Riemannian gradient in this case can be easily seen as $Z = D\phi(x)^* \mathrm{grad} f(\phi(x))$, since for any $\xi \in \mathcal{H}_x \mathcal{M}$,

$$\begin{aligned} \langle Z, \xi \rangle &= \langle D\phi(x)^* \mathrm{grad} f(\phi(x)), \xi \rangle \\ &= \langle P_{T_{\phi(x)}\mathcal{H}} \mathrm{grad} f(\phi(x)), D\phi(x)[\xi] \rangle \\ &= Df(\phi(x)) \circ D\phi(x)[\xi] \\ &= Df(\phi(x))[\xi] \end{aligned}$$

The adjoint of $D\phi(x)$ can be computed using that, for $t \in T$,

$$\delta U_t = \frac{\partial U_t}{\partial U_{t_l}} \delta U_{t_l} + \frac{\partial U_t}{\partial U_r} \delta U_{t_r} + \frac{\partial U_t}{\partial B_t} \delta B_t$$

and $D\phi(x)[\xi] = \mathrm{vec}(\delta U_{\mathrm{t_{root}}})$. The adjoint of this recursion, followed by a projection on to (2), gives us Figure 1.

Since U_t in (1) is linear in each variable, one can write out the partial derivatives of U_t with respect to U_{t_l}, U_{t_r} and B_t by considering the possible matricizations of U_t

$$\begin{aligned} U_t^{(k_r)} &= U_r B_t^{(k_r)} (U_{t_l} \otimes I_{k_t})^T \\ U_t^{(k_l)} &= U_l B_t^{(k_l)} (U_{t_r} \otimes I_{k_t})^T \end{aligned}$$

and using the matrix calculus product rule

$$\frac{\partial(AB)}{\partial X} = (B^T \otimes I_{M_A})\frac{\partial A}{\partial X} + (I_{N_B} \otimes A)\frac{\partial B}{\partial X}$$

to isolate for the corresponding differential. We will not go into the full derivation here due to space constraints. The final result is a very simple set of MATLAB commands, which uses code from the SPOT framework [4] and from the hTucker toolbox [5], that requires only matrix-matrix multiplications and permutations of relatively small matrices, which can be performed efficiently.

IV. Optimization Algorithms

Let $\mathcal{M}$ be the space of parameters for the OHT format with the corresponding Lie group of orthogonal matrices $\mathcal{G} \leq \mathbf{G}$ acting on $\mathcal{M}$ via θ. For the purpose of interpolation, we are interested in solving

$$x^* = \underset{x=(U_t,B_t)}{\arg\min}\, f(x) = \|A\phi(x) - b\|_2^2 \quad (4)$$
$$\text{s.t.} \quad U_t^T U_t = I_{k_t}, (B_t^{(k_l,k_r)})^T B_t^{(k_l,k_r)} = I_{k_t}$$

where A is our subsampling operator and b is our subsampled data. For a Steepest Descent-type method, we have a means to compute the Riemannian gradient of f at a point x, which we will denote g_x. In order to move along $-g_x$ for some step size t, we need a retraction on $\mathcal{M}$, which is a first-order approximation to the exponential mapping on $\mathcal{M}$.

Proposition 3. Let $x = (U_t, B_t) \in \mathcal{M}$, $\eta = (\delta U_t, \delta B_t) \in T_x\mathcal{M}$. Then the reorthogonalization mapping R, introduced in [6], and defined by

$$R_x(\eta) = \begin{cases} \text{qf}(U_t + \delta U_t) & \text{if } t \in L \\ \text{qf}((R_{t_l} \otimes R_{t_r})(B_t + \delta B_t)) & \text{if } t \notin \text{t}_{\text{root}} \cup L \\ (R_{t_l} \otimes R_{t_r})(B_t + \delta B_t) & \text{if } t = \text{t}_{\text{root}} \end{cases}$$

where $\text{qf}(X), R_t$ are the Q-factor from the QR factorization of X and R_t is the R-factor from the QR factorization associated to node t, is a retraction on $T\mathcal{M}$.

$R_x(\eta)$ can be computed very efficiently, in the sense that one avoids operating on the full tensor space $\mathbb{R}^{n_1 \times n_2 \times \ldots n_d}$ and instead one performs QR factorizations on relatively small matrices. Since R is a retraction on the tangent bundle $T\mathcal{M}$ and $\mathcal{H}_x\mathcal{M}$ in (2) is a θ-invariant horizontal distribution on $\mathcal{M}$, by 4.1.2 in [3] we have that the mapping $\tilde{R}_{\pi(x)}(\xi_{\pi(x)}) = \pi(R_x(\xi_x))$ is a well-defined retraction on $T(\mathcal{M}/\mathcal{G})$.

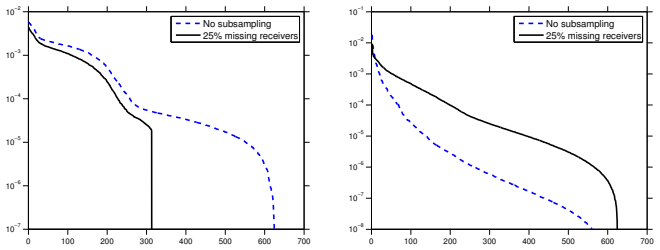
Using this retraction, we formulate the steepest descent algorithm using an Armijo line search in a straightforward manner, presented in Figure 2. We can easily modify this framework to implement other first-order methods such as CG, which we will use for our numerical examples.

V. Multidimensional Subsampling

As we use seismic data examples for our recovery, it should be noted that 3D seismic data is five dimensional, with two source coordinates (x, y), two receiver coordinates (x, y), and time, from which we extract a single, 4D frequency slice by taking the Fourier transform in time and fixing a frequency. Owing to the symmetric nature of seismic data between sources and receivers, we have essentially two choices of underlying dimension tree, both depicted in Figure 3. Namely, we can choose between placing the (src x, src y) dimensions in the rows and (rec x, rec y) dimensions in the columns, or placing the (src x, rec x) dimensions in the rows and (src y, rec y) dimensions in the columns (each choice specifies the rest of the dimension tree). In the case when we are, say, randomly missing sources, the former organization of data has the effect that subsampling will tend to remove rows of this matrix, and hence the singular values will not increase and in fact *are set to zero* at the low end (the worst-case scenario for the purposes of rank-minimizing recovery, e.g. see [7]). On the other hand, the latter organization of data results in a subsampling operator that randomly removes blocks from the underlying matrix, which is a much more favourable situation from a low-rank recovery perspective, as we can see from the singular values of the resulting matrix. The same situation holds for matricizations in the singleton dimensions, adding further degrees of regularity to the computed solution compared to standard matrix completion. Our choice of dimension tree is of great importance in the success of our recovery.

Require: Initial guess $x_0 = (U_t, B_t)$, $0 < c < 1$ sufficient decrease parameter, $0 < \theta < 1$ step size decrease
for $k = 0, 1, 2, \ldots$ until convergence **do**
  $\mathbf{X}_k \leftarrow \phi(x_k)$
  $f_k \leftarrow f(\mathbf{X}_k)$
  $g_k \leftarrow \nabla_x f(\phi(x_k))$ //Riemannian gradient of f at x_k
  $\alpha \leftarrow 1$ //Armijo line search
  while $f(\phi(R_{x_k}(-\alpha g_k))) - f_k > -c\alpha\langle g_k, g_k\rangle$ **do**
    $\alpha \leftarrow \alpha \cdot \theta$
  end while
  $x_{k+1} \leftarrow R_{x_k}(-\alpha g_k)$
end for

Fig. 2. Steepest descent for optimizing a function f over the manifold $\mathcal{H}_{\text{T,k}}$

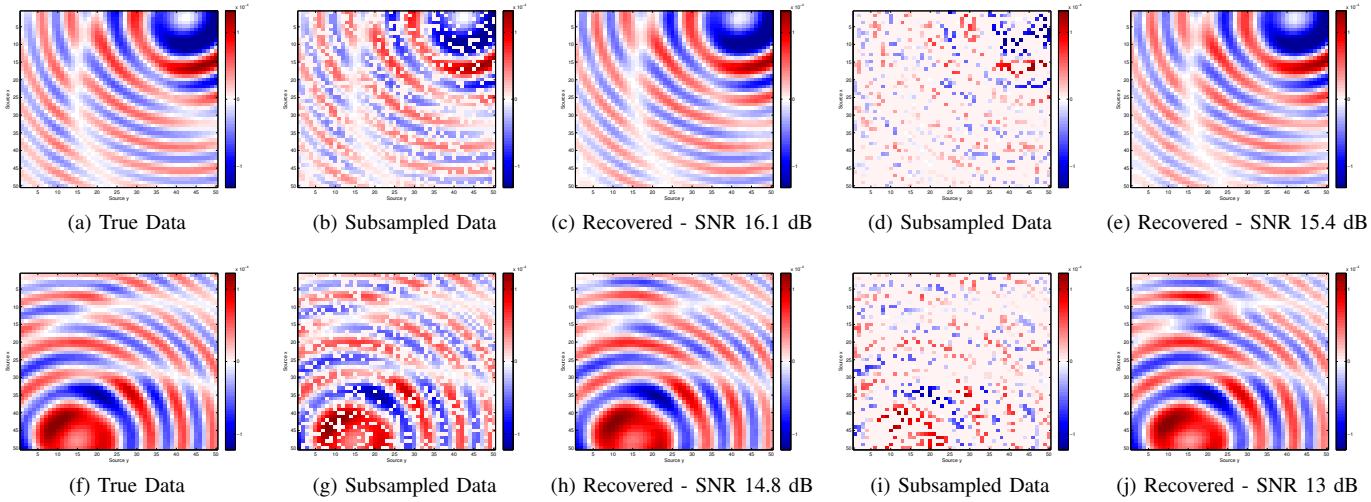


(a) (b)

Fig. 3. Singular values for the *Left*: (src x, src y) Matricization *Right*: (src x, rec x) Matricization of a test data set. *Blue*: Without subsampling, *Black*: With subsampling

VI. Numerical Experiments

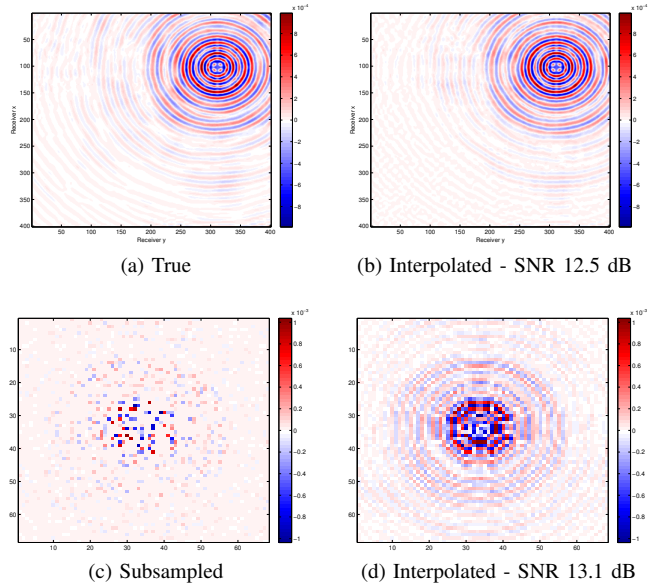
In the following examples, we apply our algorithms to interpolate seismic frequency slices from two test sets. In the first set, we use data generated from a simple single-reflector model, while the second set has been provided to us by British Gas (BG), generated from an unknown model. For our solver, we implement nonlinear CG in this OHT framework, using



(a) True Data (b) Subsampled Data (c) Recovered - SNR 16.1 dB (d) Subsampled Data (e) Recovered - SNR 15.4 dB

(f) True Data (g) Subsampled Data (h) Recovered - SNR 14.8 dB (i) Subsampled Data (j) Recovered - SNR 13 dB

Fig. 4. *Top*: (Rec x, Rec y) = (5,45). (b), (c), (g), (h) are results for 25% source subsampling, (d), (e),(i), (j) are results for 75% source subsampling



(a) True (b) Interpolated - SNR 12.5 dB

(c) Subsampled (d) Interpolated - SNR 13.1 dB

Fig. 5. Interpolated BG Data for 75% missing sources. *Top*: Fixed, unknown source image, *Bottom*: Fixed receiver image

the orthogonalization retraction in (3) and projection onto the horizontal space (2) as a vector transport.

The simple data set has size $D \in \mathbb{R}^{50\times50\times50\times50}$ and we randomly remove source (x, y) pairs from the data set before recovery. We run the resulting algorithm for 200 iterations starting from a random initial guess, which produces the results in Figure 4. Even amidst high levels of missing sources, the HT construction is able to sufficiently regularize the interpolation process to successfully recover each slice for fixed receiver coordinates (known as a common receiver gather in seismic circles).

The BG data set originally has 68 x 68 sources corresponding to 401 x 401 receivers, from which we remove a subset of the sources randomly and interpolate using our CG method. We show a common source gather and a common receiver gather for 75% missing sources in Figure 5. We summarize our results in Figure 6 for interpolating this volume from varying amounts of missing sources.

Missing Sources	SNR - Known	SNR - Interpolated
25%	15.4 dB	14.4 dB
50%	15.7 dB	14.1 dB
75%	17.4 dB	11.6 dB

Fig. 6. SNRs of the data volume restricted to known source locations and interpolated source locations after recovery.

VII. Conclusion

In this work, we have extended the largely theoretical results of [1] to a practical algorithmic framework for solving optimization problems whose solutions lie on a Hierarchical Tucker manifold of fixed dimension tree and hierarchical rank. Our methods easily allow us to interpolate tensors exhibiting this hierarchical low-rank structure from a subset of their entries. There is a large open question as to how one can formulate precise recovery results for this problem to the sufficiently comprehensive level of the recovery results present in the Compressive Sensing and Matrix Completion literature, a question that we leave for future research.

The authors would like to thank the sponsors of the SINBAD consortium for their continued support and particularly BG for providing the test data set.

References

[1] A. Uschmajew and B. Vandereycken, "The geometry of algorithms using hierarchical tensors," *preprint, http://sma.epfl.ch/~vanderey/papers/geom_htucker.pdf*, 2012.

[2] B. Vandereycken, "Low-rank matrix completion by riemannian optimization—extended version," *arXiv.org*, Sep. 2012.

[3] P. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton Univ Press, 2008.

[4] E. van den Berg and M. Friedlander. (2009) The spot operator toolbox. [Online]. Available: http://www.cs.ubc.ca/labs/scl/spot/index.html

[5] D. Kressner and C. Tobler, "htucker–a matlab toolbox for tensors in hierarchical tucker format," *MATHICSE, EPF Lausanne, available at http://sma.epfl.ch/~anchpcommon/publications/htucker.pdf*, 2012.

[6] C. TOBLER, "Low rank tensor methods for linear systems and eigenvalue problems," Ph.D. dissertation, ETH Zürich, 2012.

[7] E. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717–772, 2009.