Recursive Implementation of the Distributed Karhunen-Loève Transform

Alon Amar, *Member, IEEE*, Amir Leshem, *Senior Member, IEEE*, and Michael Gastpar, *Member, IEEE*

***Abstract*—In the distributed linear source coding problem, a set of distributed sensors observe subsets of a data vector with noise, and provide the fusion center linearly encoded data. The goal is to determine the encoding matrix of each sensor such that the fusion center can reconstruct the entire data vector with minimum mean square error. The recently proposed local Karhunen-Loève transform approach performs this task by optimally determining the encoding matrix of each sensor assuming the other matrices are fixed. This approach is implemented iteratively until convergence is reached. Herein, we propose a greedy algorithm. In each step, one of the encoding matrices is updated by appending an additional row. The algorithm selects in a greedy fashion a single sensor that provides the largest improvement in minimizing the mean square error. This algorithm terminates after a finite number of steps, that is, when all the encoding matrices reach their predefined encoded data size. We show that the algorithm can be implemented recursively, and compared to the iterative approach, the algorithm reduces the computational load from cubic dependency to quadratic dependency on the data size. This makes it a prime candidate for on-line and real-time implementations of the distributed Karhunen-Loève transform. Simulation results suggest that the mean square error performance of the suggested algorithm is equivalent to the iterative approach.**

***Index Terms*—Distributed compression, distributed Karhunen-Loève transform, distributed transforms, principal component analysis, source coding, transform coding.**

I. Introduction

WIRELESS sensor networks consisting of battery operated sensors with computation and communication capabilities attract much attention due to their wide range of applications including: military and civilian surveillance, environmental monitoring, health care, traffic control, and source localization [1]–[5].

Each sensor in the network has a limited battery lifetime and communications bandwidth due to the constraints on size and cost. It is therefore important that each sensor locally compresses its own observed data before relaying the compressed information to a fusion center. The fusion center can then reconstruct the entire data vector from the collected information. A key problem is the design of the compression at each sensor such that the fusion center will produce a reconstruction that will be optimal under a certain criterion, such as the mean square error (MSE). Several different approaches towards distributed compression have been pursued. For example, in the information-theoretic literature, distributed compression has received considerable attention following the landmark works of Slepian and Wolf [6], and Wyner and Ziv [7]. In this paper, however, we consider a different abstraction of compression. Specifically, we view compression as dimensionality reduction by linear projections. In the centralized (nondistributed) setting, the MSE optimal solution is the well-known Karhunen-Loève transform (KLT). There are intimate relations between this perspective and the information-theoretic one, an account of which can be found, e.g., in [8].

Recently, the classical KLT was extended beyond the centralized case in [9]: Suppose there are several spatially distributed sensors, each observing only a part of the entire data vector. The sensors cannot communicate with each other. Each sensor provides linearly encoded data to the fusion center, which is the result of linearly transforming its input data by an encoding matrix (the row dimension of this matrix specifies the size of the encoded data of the sensor). For this problem, upper and lower bounds on the possible MSE performance can be given in a straightforward fashion. An upper bound is the marginal KLT, where each sensor computes a KLT based only on its local statistics, without taking into account the correlations of its data with the data observed by the other sensors. A lower bound is the joint KLT, which cannot be applied to the entire data since the entire data vector is not observed by each sensor (detailed explanation of the joint KLT is discussed in [9, Section II-A]). The correlations between the subvectors determine the separation between the two bounds. The MSE performance of the distributed KLT (dKLT) is between these two bounds. In [9] the reconstruction of the entire data vector in the fusion center (termed as the local KLT approach) is based on a sensor-by-sensor perspective. The idea can be described as follows: Consider one of the sensors, and assume that the encoding matrices of all the other sensors are fixed and known to that sensor. This sensor then determines its optimal encoding matrix in such a way as to minimize the MSE in reconstructing the entire data vector at the fusion center based on the encoded data from all sensors. This approach is performed with an iterative algorithm, where in each iteration step, only one of the sensors updates its encoding matrix. The algorithm terminates when the difference between the MSEs in

Manuscript received August 23, 2009; accepted June 11, 2010. Date of publication July 08, 2010; date of current version September 15, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mathini Sellathurai. This work was supported in part by NWO-STW under the VICI program (Project 10382), and by the 3TU.CeDICT: Centre for Dependable ICT Systems.

A. Amar is with Faculty of Electrical Engineering, Delft University of Technology, Delft, 2628 CD, The Netherlands (e-mail: a.amar@tudelft.nl).

A. Leshem is with Faculty of Electrical Engineering, Delft University of Technology, Delft, 2628 CD, The Netherlands. He is also with the School of Engineering, Bar Ilan University, Ramat Gan, 52900, Israel (e-mail: leshem.amir1@gmail.com).

M. Gastpar is with Faculty of Electrical Engineering, Delft University of Technology, Delft, 2628 CD, The Netherlands. He is also with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720 USA (e-mail: gastpar@eecs.berkeley.edu).

Digital Object Identifier 10.1109/TSP.2010.2056922

two subsequent iteration steps is smaller than a predefined tolerance. In [9] it was shown that this algorithm converges at least to a local minimum of the MSE.

The dKLT is also studied in [10] where only the existence of second moments is required, and not the assumption of joint gaussianity of the observations. For a two-terminal scenario, an asymptotic analysis of the MSE performance of the dKLT has appeared in [11]. The dKLT is not an orthogonal transform and thus, its subspaces are not nested. In [12], a novel algorithm was presented that provides a distributed transform with nested subspaces, at the expense of increased MSE. The distributed transforms discussed here apply to a single layer of encoders that directly feed to a decoder. In [13], distributed linear transforms are designed for multilayer networks. In recent work [14], an alternative perspective on distributed transforms is developed: The goal is to determine the joint (centralized) KLT in multiple distributed rounds with minimal communication.

It should be emphasized that the dKLT discussed here is used for the reconstruction scenario, that is, the goal of the fusion center is to reconstruct the entire data vector from the compressed sensor observations with minimal MSE. A different problem, although related in a sense, is the estimation scenario, where the goal of the fusion center is to accurately estimate a parameter vector of interest from the compressed sensor observations [15]. The latter setup is discussed in [16] and [17] using linear estimators and ideal communication channels between the sensors and the fusion center, and is further extended in [18] to the important case of nonideal communication channels (for example, fading).

Herein, we propose a greedy algorithm which terminates after a finite number of steps. In each step, one of the encoding matrices is updated by appending an additional row; in a greedy fashion, the algorithm selects one sensor that provides the largest decrease in the MSE. This algorithm terminates when all the encoding matrices reach their predefined row dimensions, and thus, requires a fixed number of iteration steps, known ahead of time. As the second-order statistics of the observation are known to the fusion center, the calculations of the encoding matrices and the selection process are done at the processing unit of this fusion center, and then it distributes the results to the other sensor nodes. Simulation results suggest that the MSE performance of the proposed new algorithm is equivalent to the iterative local KLT. It is worth mentioning that in extension of [9], in our problem setup we assume that a disjoint part of the vector of interest is observed by each sensor in the presence of noise. (In [19] we presented the noiseless version of the greedy algorithm.) The reconstruction of the vector of interest in the fusion center is then performed using the noisy compressed data. A related problem which deals with reconstructing the data vector in the presence of noise is discussed for the joint (centralized) KLT setup in [20]–[23].

A key advantage of the proposed algorithm lies in its reduced complexity. For the iterative local KLT algorithm, the calculation of the encoding matrix in each iteration step involves multiplying and inverting matrices whose sizes are linearly related to the dimensions of the entire data vector, the data observed by the sensor, and those of all the encoding matrices (see [9, Eq. (14)–(15)]). Moreover, this algorithm involves the eigenvalue decomposition of a matrix with dimensions equal to the data size. Due to these matrix multiplications and inversions, and this eigenvalue decomposition, the complexity of the iterative algorithm increases cubically with respect to the data size. By contrast, we show that since the greedy algorithm involves only products of a matrix by a vector but not matrix inversions, and only computes the largest eigenvector, the algorithm requires fewer computations compared to the iterative approach, that is, the complexity increases quadratically with respect to the data size and not cubically. We also show that the proposed algorithm can be implemented recursively. In standard applications, joint second-order statistics may be known ahead of time and thus the KLT and dKLT can be calculated ahead of time, meaning that the computation complexity of finding the best transform is a minor issue, though if we consider correlations over time, the dimension of interest can easily be in the thousands, making fast algorithms interesting. Even more important are emerging applications, where it may be significant to compute (and recompute) optimal dimensionality reduction "on the fly," based on sequentially updated estimates of second-order statistics. In these cases, the computational complexity of determining optimal transforms can become an important issue, and thus, the proposed new algorithm may become an important tool.

The rest of this paper is organized as follows. In Section II we present the problem formulation. In Section III we summarize the main results of the iterative local KLT method. In Section IV we present the greedy algorithm. In Section V we develop recursive expressions for both the reconstructed data vector and its MSE obtained by the greedy method. In Section VI we compare the complexity load of the iterative method with the proposed greedy method. In Section VII we present numerical examples, and finally, in Section VIII we conclude the paper.

The following notation is used in the paper: uppercase bold fonts denote matrices, and lowercase bold fonts denote vectors. The superscripts $(\cdot)^T$, $(\cdot)^{-1}$ stand for transpose, and inverse, respectively. $\mathbf{I}_n$ is the $n \times n$ identity matrix, $\mathbf{0}_n$ is a $n \times 1$ vector with all elements equal to zero. Also, $\mathrm{Diag}(\mathbf{Z}_1, \cdots, \mathbf{Z}_N)$ is a block diagonal matrix where the matrices $\mathbf{Z}_1, \cdots, \mathbf{Z}_N$ are on the main diagonal, $\mathrm{diag}(z_1, \ldots, z_N)$ is a diagonal matrix with $z_1, \ldots, z_N$ on the main diagonal. $\mathrm{tr}(\mathbf{A})$ denote the trace of the matrix $\mathbf{A}$, and $[\mathbf{A}]_m$ represents the submatrix of $\mathbf{A}$ consisting of the first m rows of $\mathbf{A}$. Finally, $E[\mathbf{x}]$ represents the expectation of the random vector $\mathbf{x}$, $\hat{E}[\mathbf{x}|\mathbf{y}] \triangleq E[\mathbf{x}\mathbf{y}^T]E^{-1}[\mathbf{y}\mathbf{y}^T]\mathbf{y}$ is the linear minimum MSE estimator of $\mathbf{x}$ given $\mathbf{y}$ (and also the optimal estimator when both are jointly Gaussian), and its MSE is $\mathrm{cov}(\hat{E}[\mathbf{x}|\mathbf{y}], \mathbf{x}) = E[\mathbf{x}\mathbf{x}^T] - E[\mathbf{x}\mathbf{y}^T]E^{-1}[\mathbf{y}\mathbf{y}^T]E[\mathbf{y}\mathbf{x}^T]$.

II. Problem Formulation

For simplicity we consider the case of two sensors. The extension to a larger number of sensors is straightforward. Consider a $N \times 1$ real-valued random vector $\mathbf{r} \triangleq [r_1, \ldots, r_N]^T$, where $\mathbf{r} = \mathbf{x}+\mathbf{n}$. The $N \times 1$ real-valued random vector of interest is denoted by $\mathbf{x} \triangleq [x_1, \ldots, x_N]^T$, which has a zero mean and a covariance matrix $\mathbf{\Sigma_{xx}} \triangleq E[\mathbf{x}\mathbf{x}^T]$. The $N \times 1$ vector $\mathbf{n} \triangleq [n_1, \ldots, n_N]^T$, representing the noise, has a zero mean and a covariance matrix

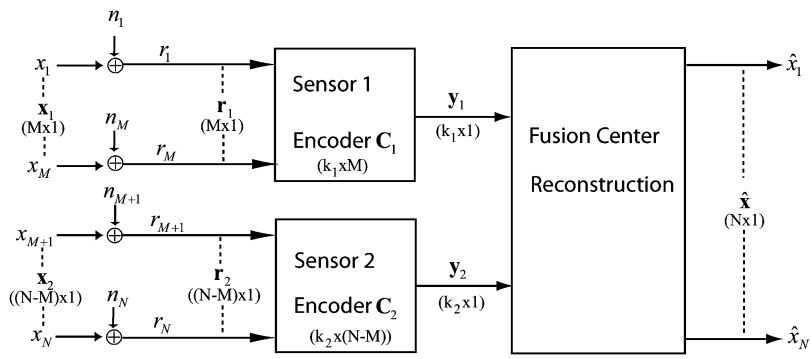

Fig. 1. The distributed KLT setup for the case of two sensors.

$\mathbf{\Sigma_{nn}} \triangleq E[\mathbf{nn}^T]$. We assume that the vector of interest $\mathbf{x}$ is uncorrelated with the noise vector $\mathbf{n}$. Therefore, $\mathbf{r}$ has a zero mean with a covariance matrix $\mathbf{\Sigma_{rr}} \triangleq E[\mathbf{rr}^T] = \mathbf{\Sigma_{xx}} + \mathbf{\Sigma_{nn}}$.

Each sensor samples a disjoint part of $\mathbf{r}$. The first sensor observes the first M components of $\mathbf{r}$, denoted by $\mathbf{r}_1 = [r_1, r_2, \ldots, r_M]^T$ where $\mathbf{r}_1 = \mathbf{x}_1 + \mathbf{n}_1$, with $\mathbf{x}_1 = [x_1, x_2, \ldots, x_M]^T$, and $\mathbf{n}_1 = [n_1, n_2, \ldots, n_M]^T$. The second sensor observes the next $N - M$ components of $\mathbf{r}$, denoted by $\mathbf{r}_2 = [r_{M+1}, \ldots, r_N]^T$ where $\mathbf{r}_2 = \mathbf{x}_2 + \mathbf{n}_2$, with $\mathbf{x}_2 = [x_{M+1}, x_{M+2}, \ldots, x_N]^T$, and $\mathbf{n}_2 = [n_{M+1}, n_{M+2}, \ldots, n_N]^T$ (Although we discuss real-valued random vectors, the extension to complex-valued random vectors is straightforward).

Each sensor individually sends to the fusion center a $k_j \times 1$ encoded data vector, denoted by $\mathbf{y}_j = \mathbf{C}_j \mathbf{r}_j$, $j = 1, 2$, where $k_j \leq M_j$ is a fixed (predefined) integer, $M_1 = M$, $M_2 = N - M$, and $\mathbf{C}_j$ is a $k_j \times M_j$ encoding matrix (see Fig. 1). The goal of the fusion center is to obtain a reconstruction of the vector of interest $\mathbf{x}$, denoted by $\hat{\mathbf{x}}$, such that the MSE, denoted by $D_x \triangleq E[\|\mathbf{x} - \hat{\mathbf{x}}\|^2]$, is minimized. The problem discussed in the context of the dKLT is: How to determine the encoding matrices $\mathbf{C}_j$ such that the MSE will be minimized?

III. The Iterative Local KLT Algorithm

The iterative local KLT algorithm was proposed in [9, Algorithm 1, p. 5186] as a suboptimal solution to the problem. In [9] it is assumed that the observations are noiseless, i.e., $\mathbf{r}_j = \mathbf{x}_j$, $j = 1, 2$. In this section, we briefly describe this algorithm. Consider a system with two sensors. Let $\hat{\mathbf{x}}(n)$ denote the estimate of $\mathbf{x}$ at the nth iteration step, and $D_x(n) \triangleq E[\|\hat{\mathbf{x}}(n) - \mathbf{x}\|^2]$ the MSE. Consider that at the $(n+1)$th iteration step sensor 2 has a fixed matrix $\mathbf{C}_2(n)$. Given $\mathbf{C}_2(n)$, the goal is to determine the optimal encoding matrix of sensor 1 at the $(n+1)$th iteration step, denoted by $\mathbf{C}_1(n+1)$, such that $D_x(n+1)$ is minimized.

Define the $(N - M) \times (M + k_2)$ matrix $\mathbf{A}$, and the $N \times N$ matrix $\mathbf{\Lambda}$

$$\mathbf{A} = \left[\mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_2}^T \quad \mathbf{\Sigma}_{\mathbf{x}_2\mathbf{x}_2}\mathbf{C}_2^T(n)\right] \times \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_1} & \mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_2}\mathbf{C}_2^T(n) \\ \mathbf{C}_2(n)\mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_2}^T & \mathbf{C}_2(n)\mathbf{\Sigma}_{\mathbf{x}_2\mathbf{x}_2}\mathbf{C}_2^T(n) \end{bmatrix}^{-1} \tag{1}$$

$$\begin{aligned}\mathbf{\Lambda} = &\begin{bmatrix} \mathbf{I}_M \\ \left([\mathbf{A}^T]_M\right)^T \end{bmatrix} \\ &\times \Big(\mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_1} - \mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_2}\mathbf{C}_2^T(n)\left(\mathbf{C}_2(n)\mathbf{\Sigma}_{\mathbf{x}_2\mathbf{x}_2}\mathbf{C}_2^T(n)\right)^{-1} \\ &\quad \times \mathbf{C}_2(n)\mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_2}^T\Big)\begin{bmatrix} \mathbf{I}_M \\ \left([\mathbf{A}^T]_M\right)^T \end{bmatrix}^T \\ &\triangleq \mathbf{R}\mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)\mathbf{R}^T\end{aligned} \tag{2}$$

where $\mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_1} \triangleq E[\mathbf{x}_1\mathbf{x}_1^T]$, $\mathbf{\Sigma}_{\mathbf{x}_2\mathbf{x}_2} \triangleq E[\mathbf{x}_2\mathbf{x}_2^T]$, and $\mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_2} \triangleq E[\mathbf{x}_1\mathbf{x}_2^T]$, $\lambda_1 \geq \cdots \lambda_M \geq \lambda_{M+1} = \cdots = \lambda_N = 0$ are the eigenvalues of $\mathbf{\Lambda}$, and $\mathbf{R}$ is the orthonormal eigenvectors matrix of $\mathbf{\Lambda}$.

As shown in [9], the optimal encoding matrix $\mathbf{C}_1(n+1)$, given $\mathbf{y}_2 = \mathbf{C}_2(n)\mathbf{x}_2$, is

$$\begin{aligned}\mathbf{C}_1(n+1) &= \left[[\mathbf{R}^T]_M \begin{bmatrix} \mathbf{I}_M \\ \left([\mathbf{A}^T]_M\right)^T \end{bmatrix}\right]_{k_1} \\ &= [\mathbf{R}^T]_{k_1}\begin{bmatrix} \mathbf{I}_M \\ \left([\mathbf{A}^T]_M\right)^T \end{bmatrix}\end{aligned} \tag{3}$$

Let $\mathbf{C}(n+1) \triangleq \mathrm{Diag}(\mathbf{C}_1(n+1), \mathbf{C}_2(n))$. The estimate of $\mathbf{x}$ and its associated MSE at the $(n+1)$th iteration step are then given as

$$\hat{\mathbf{x}}(n+1) = \mathbf{\Sigma_{xx}}\mathbf{C}^T(n+1)\left(\mathbf{C}(n+1)\mathbf{\Sigma_{xx}}\mathbf{C}^T(n+1)\right)^{-1} \times \mathbf{y}(n+1) \tag{4}$$

$$\begin{aligned}D_x(n+1) = \mathrm{tr}\big(&\mathbf{\Sigma_{xx}} - \mathbf{\Sigma_{xx}}\mathbf{C}^T(n+1) \\ &\times \left(\mathbf{C}(n+1)\mathbf{\Sigma_{xx}}\mathbf{C}^T(n+1)\right)^{-1} \\ &\times \mathbf{C}(n+1)\mathbf{\Sigma_{xx}}\big)\end{aligned} \tag{5}$$

where $\mathbf{y}(n+1) \triangleq \mathbf{C}(n+1)\mathbf{x}$. In the next iteration step the optimal matrix $\mathbf{C}_2(n+2)$ is determined given $\mathbf{C}_1(n+1)$, and so on. The iterative algorithm terminates when $|D_x(n+1) - D_x(n)| \leq \epsilon$, where ϵ is a predefined tolerance. In [9], [10] it was shown that the iterative local KLT method converges at least to a local minimum of the MSE.

Remark: The matrix $\mathbf{C}_1(n+1)$ is the optimal solution in two different senses: i) when the observation $\mathbf{x}$ is a Gaussian random vector; ii) if we assume that the reconstruction is linear given that the observation $\mathbf{x}$ is a general random vector with finite second-order moments.

IV. The Greedy Algorithm

The principal notion of the proposed algorithm is that at each step, we determine the single sensor that by adding another dimension to its compressed version, attains the largest reduction in the MSE for the reconstruction of the entire data vector. Only that sensor's compression matrix is updated in this step. In this sense it is a greedy algorithm. The number of steps required for such a procedure is $k_1 + k_2$, which is the total predefined size of the encoded data of all sensors.

We start by defining the following:

- $\mathbf{d}_i \triangleq [d_0, d_1, \ldots, d_i]^T$, $i = 1, \ldots, k_1 + k_2$ is the $i \times 1$ decision vector of the fusion center at the ith step, where

$$d_i = \ell, \quad \ell \in \{1, 2\}, \quad \text{if sensor } \ell \text{ is selected at the } i\text{th step.} \tag{6}$$

- p_i and q_i are the total numbers of 1's and 2's, respectively, in $\mathbf{d}_i$, where $p_i + q_i = i$.
- $\mathbf{C}_1^{(p_i)}$ and $\mathbf{C}_2^{(q_i)}$ are the $p_i \times M$, and $q_i \times (N - M)$ encoding matrices at the ith step of sensor 1 and sensor 2, respectively.[1]
- $\mathbf{u}_n^T$, $n = 1, \ldots, p_i$, and $\mathbf{v}_m^T$, $m = 1, \ldots, q_i$ are the nth row and mth row of $\mathbf{C}_1^{(p_i)}$ and $\mathbf{C}_2^{(q_i)}$, respectively.
- $\hat{\mathbf{x}}^{(i)}$ is the estimate of $\mathbf{x}$ at the ith step, and $D_x^{(i)} \triangleq E[\|\hat{\mathbf{x}}^{(i)} - \mathbf{x}\|^2]$ is its MSE.
- $\mathbf{y}_{1,i} = \mathbf{C}_1^{(p_i)}\mathbf{r}_1$ and $\mathbf{y}_{2,i} = \mathbf{C}_2^{(q_i)}\mathbf{r}_2$ are the outputs of sensor one and sensor two at the end of the ith step.
- $\mathbf{W}_i$ is a $N \times i$ matrix such that its mth column vector, denoted by $\mathbf{w}_m$, is defined as

$$\mathbf{w}_m \triangleq \begin{cases} \left[\mathbf{u}_{p_m}^T, \mathbf{0}_{N-M}^T\right]^T, & \text{if } d_m = 1 \\ \left[\mathbf{0}_M^T, \mathbf{v}_{q_m}^T\right]^T, & \text{if } d_m = 2 \end{cases} \tag{7}$$

- $\mathbf{z}_i \triangleq \mathbf{W}_i^T\mathbf{r}$ is the $i \times 1$ vector that consists the previous outputs of the sensors.

Consider the $(i+1)$th step. The fusion center needs to decide between two alternatives:

1) Letting sensor 1 add a new row vector, denoted by $\mathbf{u}_{p_{i+1}}^T$, such that $\mathbf{C}_1^{(p_{i+1})} = \left[\begin{smallmatrix}\mathbf{C}_1^{(p_i)} \\ \mathbf{u}_{p_{i+1}}^T\end{smallmatrix}\right]$.
2) Letting sensor 2 add a new row vector, denoted by $\mathbf{v}_{q_{i+1}}^T$, such that $\mathbf{C}_2^{(q_{i+1})} = \left[\begin{smallmatrix}\mathbf{C}_2^{(q_i)} \\ \mathbf{v}_{q_{i+1}}^T\end{smallmatrix}\right]$.

The decision criterion is as follows: Assume that sensor 1 (or sensor 2) is selected. Determine the optimal vector $\mathbf{u}_{p_{i+1}}$ (or $\mathbf{v}_{q_{i+1}}$), given $\mathbf{z}_i = \mathbf{W}_i^T\mathbf{r}$, such that the MSE in reconstructing $\mathbf{x}$ is minimized. Let $\varepsilon_{i+1,1}$ (or $\varepsilon_{i+1,2}$) denote the MSE in reconstructing $\mathbf{x}$ at the $(i+1)$th step assuming that sensor 1 (or sensor 2) is selected. Given $\varepsilon_{i+1,1}$ and $\varepsilon_{i+1,2}$, the decision of the fusion center at the $(i+1)$th step is based on selecting the sensor which provides a smaller MSE, that is

$$d_{i+1} = \arg\min_{\ell \in \{1,2\}} \{\varepsilon_{i+1,\ell}\}. \tag{8}$$

The MSE in reconstructing $\mathbf{x}$ at the $(i+1)$th step is then

$$D_x^{(i+1)} = \varepsilon_{i+1,\ell} \quad \text{if} \quad d_{i+1} = \ell, \quad \ell \in \{1, 2\}. \tag{9}$$

[1]Note that it is also possible that one of the matrices will be empty at this step.

This binary decision process, performed by the fusion center, continues until $p_i = k_1$ or $q_i = k_2$. Assume that this occurs when $q_i = k_2$ while $p_i < k_1$. The fusion center then performs a sequence of $k_1 - p_i$ steps where only sensor 1 further adds a new encoding vector at each step, based on the previous outputs of the two sensors.

The determination of the vector $\mathbf{u}_{p_{i+1}}$ is described in the following result (the determination of $\mathbf{v}_{q_{i+1}}$ is similar, and is obtained with minor changes). The result is a modification of the proof in [9, Theorem 2], and its derivation is explained in the Appendix. In short, in the proof presented in [9, Appendix], the determination of the complete matrix $\mathbf{C}_1$ (or $\mathbf{C}_2$) is based on the previous output $\mathbf{y}_2 = \mathbf{C}_2\mathbf{x}_2$ (or $\mathbf{y}_1 = \mathbf{C}_1\mathbf{x}_1$) (Recall that in [9] the observations are noiseless, i.e., $\mathbf{r}_j = \mathbf{x}_j$, $j = 1, 2$.) Here, the determination of $\mathbf{u}_{p_{i+1}}$ is based on all the previous outputs up to the $(i+1)$th step which is $\mathbf{z}_i = \mathbf{W}_i^T\mathbf{r}$.

Result 4.1: Define the $(N - M) \times (M + q_i)$ matrix $\mathbf{A}^{(q_i)}$,

$$\mathbf{A}^{(q_i)} \triangleq \mathbf{S}_i\mathbf{G}_i \tag{10}$$

where the $(N - M) \times (M + q_i)$ matrix $\mathbf{S}_i$ and the $(M + q_i) \times (M + q_i)$ matrix $\mathbf{G}_i$ are

$$\mathbf{S}_i \triangleq \left[\mathbf{\Sigma}_{\mathbf{r}_1\mathbf{x}_2}^T \quad \mathbf{\Sigma}_{\mathbf{x}_2\mathbf{r}_2}\left(\mathbf{C}_2^{(q_i)}\right)^T\right] \tag{11}$$

$$\mathbf{G}_i \triangleq \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{r}_1\mathbf{r}_1} & \mathbf{\Sigma}_{\mathbf{r}_1\mathbf{r}_2}\left(\mathbf{C}_2^{(q_i)}\right)^T \\ \mathbf{C}_2^{(q_i)}\mathbf{\Sigma}_{\mathbf{r}_1\mathbf{r}_2}^T & \mathbf{C}_2^{(q_i)}\mathbf{\Sigma}_{\mathbf{r}_2\mathbf{r}_2}\left(\mathbf{C}_2^{(q_i)}\right)^T \end{bmatrix}^{-1} \tag{12}$$

where $\mathbf{\Sigma}_{\mathbf{x}_2\mathbf{r}_1} \triangleq E[\mathbf{x}_2\mathbf{r}_1^T]$, $\mathbf{\Sigma}_{\mathbf{x}_2\mathbf{r}_2} \triangleq E[\mathbf{x}_2\mathbf{r}_2^T]$, $\mathbf{\Sigma}_{\mathbf{r}_1\mathbf{r}_1} \triangleq E[\mathbf{r}_1\mathbf{r}_1^T]$, $\mathbf{\Sigma}_{\mathbf{r}_1\mathbf{r}_2} \triangleq E[\mathbf{r}_1\mathbf{r}_2^T]$, and $\mathbf{\Sigma}_{\mathbf{r}_2\mathbf{r}_2} \triangleq E[\mathbf{r}_2\mathbf{r}_2^T]$. Also define the $N \times N$ matrix $\mathbf{\Sigma}_i$

$$\mathbf{\Sigma}_i \triangleq \mathbf{P}_i\mathbf{J}_i\mathbf{P}_i^T \tag{13}$$

where we define the $N \times M$ matrix $\mathbf{P}_i$ as

$$\mathbf{P}_i \triangleq \begin{bmatrix} \mathbf{I}_M \\ \left(\left[\left(\mathbf{A}^{(q_i)}\right)^T\right]_M\right)^T \end{bmatrix} \tag{14}$$

and where the $M \times M$ matrix $\mathbf{J}_i$ is defined as

$$\mathbf{J}_i \triangleq \mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_1} - \mathbf{\Sigma}_{\mathbf{x}_1\mathbf{r}}\mathbf{\Psi}_i\mathbf{\Sigma}_{\mathbf{x}_1\mathbf{r}}^T \tag{15}$$

$$\mathbf{\Psi}_i \triangleq \mathbf{W}_i\mathbf{H}_i\mathbf{W}_i^T \tag{16}$$

$$\mathbf{H}_i \triangleq \left(\mathbf{W}_i^T\mathbf{\Sigma}_{\mathbf{rr}}\mathbf{W}_i\right)^{-1}. \tag{17}$$

We note that $\mathbf{\Psi}_i$ is a $N \times N$ matrix, and $\mathbf{H}_i$ is a $i \times i$ matrix. Finally, we express the eigenvalue decomposition of $\mathbf{\Sigma}_i$ as

$$\mathbf{\Sigma}_i = \mathbf{Q}_i \mathrm{diag}(\lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,N})\mathbf{Q}_i^T \tag{18}$$

where $\lambda_{i,1} \geq \cdots \lambda_{i,M} \geq \lambda_{i,M+1} = \cdots = \lambda_{i,N} = 0$ are the (nonincreasingly ordered) eigenvectors of $\mathbf{\Sigma}_i$, and $\mathbf{Q}_i \triangleq [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N]$ collects the (orthonormal) eigenvectors of $\mathbf{\Sigma}_i$.

The vector $\mathbf{u}_{q_{i+1}}$ which leads to the minimization of the MSE in reconstructing $\mathbf{x}$ is

$$\mathbf{u}_{q_{i+1}}^T = \left[\left[\mathbf{Q}_i^T\right]_M \mathbf{P}_i\right]_1 = \left[\mathbf{Q}_i^T\right]_1 \mathbf{P}_i = \mathbf{q}_1^T\mathbf{P}_i \tag{19}$$

Observe that $\mathbf{q}_1$ is the largest (principal) eigenvector of $\mathbf{Q}_i$. The MSE in reconstructing $\mathbf{x}$ at the $(i+1)$th step is then given as

$$\begin{aligned}\varepsilon_{i+1,1} &= E\left[\|\mathbf{x}-\hat{\mathbf{x}}\|^2\right] \\ &= \mathrm{tr}\left(\mathrm{E}[\mathbf{x}\mathbf{x}^T] - E\left[\mathbf{x}(\mathbf{F}_{i+1}\mathbf{r})^T\right]\right. \\ &\qquad \left.\times\left(E\left[\mathbf{F}_{i+1}\mathbf{r}(\mathbf{F}_{i+1}\mathbf{r})^T\right]\right)^{-1} E[\mathbf{F}_{i+1}\mathbf{r}\mathbf{x}^T]\right) \\ &= \mathrm{tr}\left(\boldsymbol{\Sigma}_{\mathbf{xx}} - \boldsymbol{\Sigma}_{\mathbf{xr}}\mathbf{F}_{i+1}^T\left(\mathbf{F}_{i+1}\boldsymbol{\Sigma}_{\mathbf{rr}}\mathbf{F}_{i+1}^T\right)^{-1}\right. \\ &\qquad \left.\times\ \mathbf{F}_{i+1}\boldsymbol{\Sigma}_{\mathbf{rx}}\right) \end{aligned} \tag{20}$$

where $\boldsymbol{\Sigma}_{\mathbf{xr}} \triangleq E[\mathbf{x}\mathbf{r}^T]$, and $\mathbf{F}_{i+1}$ is the $(i+1)\times N$ matrix that consists the encoding matrices and is defined as

$$\mathbf{F}_{i+1} \triangleq \mathrm{Diag}\left(\mathbf{C}_1^{(p_{i+1})}, \mathbf{C}_2^{(q_i)}\right) \tag{21}$$

This completes the description of Result 4.1. ■

Remark: The vector $\mathbf{u}_{q_{i+1}}$ is the optimal solution in two different senses: i) when the observation $\mathbf{x}$ is a Gaussian random vector; ii) if the reconstruction in the fusion center is constrained to be linear given that the observation $\mathbf{x}$ is a general random vector with finite second-order moments.

Note that in order to determine $\mathbf{u}_{q_{i+1}}$ we only need to compute the largest (principal) eigenvector of $\boldsymbol{\Sigma}_i$, that is, the first column of $\mathbf{Q}_i$. This is in contrast to [9] where in each step all the eigenvectors of the $N\times N$ matrix $\boldsymbol{\Lambda}$ are computed (refer to (3)). Since we are interested only in this principal eigenvector we can adopt a known method for computing it, such as the power method or the inverse iteration (refer to [24]–[27] for further discussion on such methods). While the complexity of the eigenvalue decomposition is known to be $\mathcal{O}(N^3)$, the complexity of the power method, for example, is $\mathcal{O}(N^2)$. Therefore, compared to the iterative local KLT method we reduce the complexity from cubic to quadratic dependency on the data size N. In Section VI we show that the complexity of computing the matrix $\boldsymbol{\Sigma}_i$ also increases quadratically with respect to the data size.

V. Recursive Implementation

Another advantage of the proposed approach is that the estimated data vector and its associated MSE can be calculated recursively. We first show that using the results of the previous section, the matrices involved in calculating the largest eigenvector in each step can be expressed recursively.

A. Recursive Calculation of $\boldsymbol{\Sigma}_i$

We develop a recursive expression for the matrix $\boldsymbol{\Sigma}_i$ in (13). In order to perform this task, we first develop a recursive expression for matrix $\mathbf{P}_i$ in (14), and then for the matrix $\mathbf{J}_i$ in (15).

1) Recursive Expression of $\mathbf{P}_i$*:* We first consider the matrix $\mathbf{G}_i$ in (12) and define by[2] $\mathbf{G}_0 \triangleq \boldsymbol{\Sigma}_{\mathbf{r}_1\mathbf{r}_1}^{-1}$. Assuming that sensor 2 was selected at the previous step we can write $\mathbf{C}_2^{(q_i)} = \left[\begin{smallmatrix}\mathbf{C}_2^{(q_{i-1})} \\ \mathbf{v}_{q_i}^T\end{smallmatrix}\right]$ (note that if sensor 2 was not selected, then $\mathbf{G}_i$, $\mathbf{S}_i$, and $\mathbf{A}^{(q_i)}$ do not change.) We can now rewrite the inverse matrix $\mathbf{G}_i$ as

$$\begin{aligned}\mathbf{G}_i &= \begin{bmatrix}\mathbf{G}_{i-1}^{-1} & \mathbf{b}_i \\ \mathbf{b}_i^T & \mathbf{v}_{q_i}^T\boldsymbol{\Sigma}_{\mathbf{r}_2\mathbf{r}_2}\mathbf{v}_{q_i}\end{bmatrix}^{-1} \\ &= \begin{bmatrix}\mathbf{G}_{i-1} + \frac{1}{h_i}\mathbf{G}_{i-1}\mathbf{b}_i\mathbf{b}_i^T\mathbf{G}_{i-1}^T & -\frac{1}{h_i}\mathbf{G}_{i-1}\mathbf{b}_i \\ -\frac{1}{h_i}\mathbf{b}_i^T\mathbf{G}_{i-1}^T & \frac{1}{h_i}\end{bmatrix}\end{aligned} \tag{22}$$

where in the second passing we use the matrix inversion lemma [28]. Also, we define the $(M+q_i-1)\times 1$ vector $\mathbf{b}_i$, and the scalar h_i as

$$\mathbf{b}_i \triangleq \left[\boldsymbol{\Sigma}_{\mathbf{r}_1\mathbf{r}_2}^T \quad \boldsymbol{\Sigma}_{\mathbf{r}_2\mathbf{r}_2}\left(\mathbf{C}_2^{(q_{i-1})}\right)^T\right]^T \mathbf{v}_{q_i} \tag{23}$$

$$h_i \triangleq \mathbf{v}_{q_i}^T\boldsymbol{\Sigma}_{\mathbf{r}_2\mathbf{r}_2}\mathbf{v}_{q_i} - \mathbf{b}_i^T\mathbf{G}_{i-1}\mathbf{b}_i \tag{24}$$

Therefore, the calculations of $\mathbf{G}_i$ are implemented recursively with the need to perform an initial matrix inversion of the $M\times M$ matrix $\mathbf{G}_0$. By substituting (22) in (10), and noting that the matrix $\mathbf{S}_i$ in (11) can be also expressed recursively as $\mathbf{S}_i = [\mathbf{S}_{i-1}, \boldsymbol{\Sigma}_{\mathbf{x}_2\mathbf{r}_2}\mathbf{v}_{q_i}]$, we obtain after a few simple mathematical steps a recursive expression for $\mathbf{A}^{(q_i)}$ given as

$$\mathbf{A}^{(q_i)} = \left[\mathbf{A}^{(q_{i-1})} + \boldsymbol{\alpha}_i\mathbf{b}_i^T\mathbf{G}_{i-1}^T, \quad -\boldsymbol{\alpha}_i\right] \tag{25}$$

where $\boldsymbol{\alpha}_i$ is a $(N-M)\times 1$ vector defined as

$$\boldsymbol{\alpha}_i \triangleq \frac{1}{h_i}\left(\mathbf{A}^{(q_{i-1})}\mathbf{b}_i - \boldsymbol{\Sigma}_{\mathbf{x}_2\mathbf{r}_2}\mathbf{v}_{q_i}\right) \tag{26}$$

Note that using the result in (25) we obtain that,

$$\begin{aligned}\left[\left(\mathbf{A}^{(q_i)}\right)^T\right]_M &= \left[\left(\left[\mathbf{A}^{(q_{i-1})}\right]\right)^T\right]_M + \left[\left(\boldsymbol{\alpha}_i\mathbf{b}_i^T\mathbf{G}_{i-1}^T\right)^T\right]_M \\ &= \left[\left(\left[\mathbf{A}^{(q_{i-1})}\right]\right)^T\right]_M + [\mathbf{G}_{i-1}]_M\mathbf{b}_i\boldsymbol{\alpha}_i^T.\end{aligned} \tag{27}$$

Therefore, by substituting (27) in (14) we can express $\mathbf{P}_i$ as

$$\mathbf{P}_i = \mathbf{P}_{i-1} + \mathbf{T}_i \tag{28}$$

where we define the $N\times M$ matrix $\mathbf{T}_i$ as

$$\mathbf{T}_i \triangleq \boldsymbol{\rho}_i\boldsymbol{\theta}_i^T \tag{29}$$

and $\boldsymbol{\rho}_i$, $\boldsymbol{\theta}_i$ are $N\times 1$ and $M\times 1$ vectors defined as

$$\boldsymbol{\rho}_i \triangleq \left[\mathbf{0}_M^T, \quad \boldsymbol{\alpha}_i^T\right]^T \tag{30}$$

$$\boldsymbol{\theta}_i \triangleq [\mathbf{G}_{i-1}]_M\mathbf{b}_i. \tag{31}$$

Observe that the rank of $\mathbf{T}_i$ is one. This property will be helpful when considering the complexity load involved in the calculations of the algorithm. As mentioned, the above recursion expression for $\mathbf{A}^{(q_i)}$ is needed if sensor 2 was selected at the ith step, otherwise $\mathbf{A}^{(q_i)} = \mathbf{A}^{(q_{i-1})}$, and therefore $\mathbf{P}_i = \mathbf{P}_{i-1}$.

2) Recursive Expression of $\mathbf{J}_i$*:* Consider now the matrix $\mathbf{J}_i$ in (15). This matrix depends on the matrix $\boldsymbol{\Psi}_i$ given in (16) which in turn depends on $\mathbf{H}_i$ expressed in (17). Note that $\mathbf{W}_i =$

[2]Although the computation of this inverse increases cubically with respect to N it is done only once. Moreover, if this matrix has a special structure (e.g., Toeplitz) then the computation of this matrix only increases quadratically with respect to N. Another possibility is that the covariance matrix of the observed data is slowly changing, such that the inverse can be calculated using a one-rank update, for example.

$[\mathbf{W}_{i-1}, \mathbf{w}_i]$. We can rewrite the inverse matrix $\mathbf{H}_i$ for $i = 2, \ldots, k_1 + k_2$ as

$$\begin{aligned}\mathbf{H}_i &= \begin{bmatrix} \mathbf{H}_{i-1}^{-1} & \mathbf{e}_i \\ \mathbf{e}_i^T & \mathbf{w}_i^T \boldsymbol{\Sigma}_{\mathbf{rr}} \mathbf{w}_i \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \mathbf{H}_{i-1} + \frac{1}{s_i}\mathbf{H}_{i-1}\mathbf{e}_i\mathbf{e}_i^T\mathbf{H}_{i-1}^T & -\frac{1}{s_i}\mathbf{H}_{i-1}\mathbf{e}_i \\ -\frac{1}{s_i}\mathbf{e}_i^T\mathbf{H}_{i-1}^T & \frac{1}{s_i} \end{bmatrix} \end{aligned} \tag{32}$$

where in the second passing we use the matrix inversion lemma [28]. Also, we define the $(i-1) \times 1$ vector $\mathbf{e}_i$ and the scalar s_i as

$$\mathbf{e}_i \triangleq \mathbf{W}_{i-1}^T \boldsymbol{\Sigma}_{\mathbf{rr}} \mathbf{w}_i \tag{33}$$

$$s_i \triangleq \mathbf{w}_i^T \boldsymbol{\Sigma}_{\mathbf{rr}} \mathbf{w}_i - \mathbf{e}_i^T \mathbf{H}_{i-1} \mathbf{e}_i \tag{34}$$

Again, the calculations of $\mathbf{H}_i$ are implemented recursively with the need to calculate the initial scalar $\mathbf{H}_1 = (\mathbf{w}_1^T \boldsymbol{\Sigma}_{\mathbf{rr}} \mathbf{w}_1)^{-1}$.

Substituting (32) in (16) results in the following recursive expression for the matrix $\boldsymbol{\Psi}_i$

$$\boldsymbol{\Psi}_i = \boldsymbol{\Psi}_{i-1} + \frac{1}{s_i}\boldsymbol{\beta}_i\boldsymbol{\beta}_i^T \tag{35}$$

where $\boldsymbol{\beta}_i$ is a $N \times 1$ vector defined as

$$\boldsymbol{\beta}_i \triangleq \mathbf{W}_{i-1}\mathbf{H}_{i-1}\mathbf{e}_i - \mathbf{w}_i \tag{36}$$

By substituting (35) in (15) we can express the matrix $\mathbf{J}_i$ as

$$\mathbf{J}_i = \mathbf{J}_{i-1} - \xi_i \boldsymbol{\gamma}_i \boldsymbol{\gamma}_i^T \tag{37}$$

where we define the $M \times 1$ vector $\boldsymbol{\gamma}_i$ and the scalar ξ_i as

$$\boldsymbol{\gamma}_i \triangleq \boldsymbol{\Sigma}_{\mathbf{x}_1\mathbf{r}} \boldsymbol{\beta}_i \tag{38}$$

$$\xi_i \triangleq \frac{\|\boldsymbol{\gamma}_i\|^2}{s_i}. \tag{39}$$

Using the recursive expression for $\mathbf{P}_i$ and $\mathbf{J}_i$ we now obtain a recursive expression for $\boldsymbol{\Sigma}_i$ given in (13). By substituting (28) and (37) in (13) we write $\boldsymbol{\Sigma}_i$ as

$$\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_{i-1} + \mathbf{B}_i \tag{40}$$

where we define the $N \times N$ matrix

$$\mathbf{B}_i \triangleq \boldsymbol{\psi}_i\boldsymbol{\rho}_i^T + \boldsymbol{\rho}_i\boldsymbol{\psi}_i^T + \mu_i\boldsymbol{\rho}_i\boldsymbol{\rho}_i^T - \xi_i\boldsymbol{\omega}_i\boldsymbol{\omega}_i^T \tag{41}$$

and we define the $N \times 1$ vectors $\boldsymbol{\omega}_i$ and $\boldsymbol{\psi}_i$, and the scalar μ_i as

$$\boldsymbol{\omega}_i \triangleq \left(\mathbf{P}_{i-1} + \boldsymbol{\rho}_i\boldsymbol{\theta}_i^T\right)\boldsymbol{\gamma}_i \tag{42}$$

$$\boldsymbol{\psi}_i \triangleq \mathbf{P}_{i-1}\mathbf{J}_{i-1}\boldsymbol{\theta}_i \tag{43}$$

$$\mu_i \triangleq \boldsymbol{\theta}_i^T\mathbf{J}_{i-1}\boldsymbol{\theta}_i. \tag{44}$$

Observe that the rank of each of the terms in (41) is one, and thus each matrix product involves the multiplication of a vector by a vector which significantly reduces the complexity load as discussed in the next section.

B. Recursive Calculation of the Reconstructed Vector

We now present a recursive implementation of the reconstructed vector and its associated MSE. Assume that the fusion center is at the end of the $(i+1)$th step. The output of the two sensors is given by $\mathbf{z}_{i+1} = \mathbf{W}_{i+1}^T\mathbf{r}$ which can be written as $\mathbf{z}_{i+1} = [\mathbf{z}_i^T, z_{i+1}]^T$, where $\mathbf{z}_i = \mathbf{W}_i^T\mathbf{r}$ and $z_{i+1} = \mathbf{w}_{i+1}^T\mathbf{r}$. By writing $\mathbf{W}_{i+1} = [\mathbf{W}_i, \mathbf{w}_{i+1}]$ we obtain that the reconstructed data vector $\mathbf{x}$ at the $(i+1)$th step is given as

$$\begin{aligned}\hat{\mathbf{x}}^{(i+1)} &= \hat{E}[\mathbf{x}|\mathbf{z}_{i+1}] \\ &= [\boldsymbol{\Sigma}_{\mathbf{xx}}\mathbf{W}_i \quad \boldsymbol{\Sigma}_{\mathbf{xx}}\mathbf{w}_{i+1}] \\ &\quad \times \begin{bmatrix} \mathbf{W}_i^T\boldsymbol{\Sigma}_{\mathbf{rr}}\mathbf{W}_i & \mathbf{W}_i^T\boldsymbol{\Sigma}_{\mathbf{rr}}\mathbf{w}_{i+1} \\ \mathbf{w}_{i+1}^T\boldsymbol{\Sigma}_{\mathbf{rr}}\mathbf{W}_i & \mathbf{w}_{i+1}^T\boldsymbol{\Sigma}_{\mathbf{rr}}\mathbf{w}_{i+1} \end{bmatrix}^{-1} \mathbf{z}_{i+1} \\ &= \hat{\mathbf{x}}^{(i)} - \frac{\mathbf{D}_i\mathbf{w}_{i+1}}{\mathbf{w}_{i+1}^T\mathbf{D}_i\mathbf{w}_{i+1}}\varphi_{i+1} \end{aligned} \tag{45}$$

where in the last equality we used the matrix inversion lemma [28], and we defined

$$\varphi_{i+1} \triangleq \mathbf{w}_{i+1}^T\hat{x}^{(i)} - z_{i+1} \tag{46}$$

Also we define the reconstructed vector $\mathbf{x}$ at the ith step, and its MSE matrix as

$$\hat{\mathbf{x}}^{(i)} \triangleq \hat{E}[\mathbf{x}|\mathbf{z}_i] = \boldsymbol{\Sigma}_{\mathbf{xr}}\mathbf{W}_i\mathbf{H}_i\mathbf{z}_i \tag{47}$$

$$\mathbf{D}_i \triangleq \mathrm{cov}\left(\hat{E}[\mathbf{x}|\mathbf{z}_i], \mathbf{x}\right) = \boldsymbol{\Sigma}_{\mathbf{xx}} - \boldsymbol{\Sigma}_{\mathbf{xr}}\boldsymbol{\Psi}_i\boldsymbol{\Sigma}_{\mathbf{rx}} \tag{48}$$

where $\boldsymbol{\Psi}_i$ and $\mathbf{H}_i$ are defined in (16) and (17), respectively. Using the result in (35) it can be shown that $\mathbf{D}_i$ can be expressed recursively as

$$\mathbf{D}_i = \mathbf{D}_{i-1} - \frac{1}{s_i}\boldsymbol{\Sigma}_{\mathbf{xr}}\boldsymbol{\beta}_i\boldsymbol{\beta}_i^T\boldsymbol{\Sigma}_{\mathbf{rx}} \tag{49}$$

where $\boldsymbol{\beta}_i$ and s_i are defined in (36) and (34), respectively. Note that by exploiting the orthogonality principle we get that $E[\hat{\mathbf{x}}^{(i)}\varphi_{i+1}] = E[\hat{\mathbf{x}}^{(i)}(\hat{\mathbf{x}}^{(i)} - \mathbf{x})^T]\mathbf{w}_{i+1} = \mathbf{0}$, that is, the second term in (45) is the innovation at the $(i+1)$th step of the estimate $\hat{\mathbf{x}}^{(i)}$.

Similarly, using the matrix inversion lemma [28] we get that the MSE associated with $\hat{\mathbf{x}}^{(i+1)}$ is

$$\begin{aligned}D_x^{(i+1)} &= \mathrm{tr}(\mathbf{D}_{i+1}) \\ &= E\left[\left\|\hat{\mathbf{x}}^{(i)} - \mathbf{x}\right\|^2\right] + \frac{\mathbf{w}_{i+1}^T\mathbf{D}_i\mathbf{D}_i\mathbf{w}_{i+1}}{\left(\mathbf{w}_{i+1}^T\mathbf{D}_i\mathbf{w}_{i+1}\right)^2}E\left[\varphi_{i+1}^2\right] \\ &\quad - 2\frac{\mathbf{w}_{i+1}^T\mathbf{D}_i}{\mathbf{w}_{i+1}^T\mathbf{D}_i\mathbf{w}_{i+1}}E\left[\left(\hat{\mathbf{x}}^{(i+1)} - \mathbf{x}\right)\varphi_{i+1}\right] \\ &= D_x^{(i)} - \frac{\mathbf{w}_{i+1}^T\mathbf{D}_i\mathbf{D}_i\mathbf{w}_{i+1}}{\mathbf{w}_{i+1}^T\mathbf{D}_i\mathbf{w}_{i+1}} \end{aligned} \tag{50}$$

where we used the results that $E[(\hat{\mathbf{x}}^{(i+1)} - \mathbf{x})\varphi_{i+1}] = \mathbf{D}_i\mathbf{w}_{i+1}$ and $E[\varphi_{i+1}^2] = \mathbf{w}_{i+1}^T\mathbf{D}_i\mathbf{w}_{i+1}$. Therefore, the reconstruction of the data vector $\mathbf{x}$, and the calculation of its associated MSE, can be implemented recursively.

VI. Computational Complexity

We discuss the computational complexity of the iterative local KLT method and the greedy algorithm. We focus on the number of real multiplications which are required per iteration step of each method.

A. The Iterative Local KLT

The calculations in each iteration step of the iterative local KLT method involve three parts: matrix inversions, matrix multiplications and an eigenvalue decomposition.

The method involves the inversion of an $(N - M_j + k_j) \times (N - M_j + k_j)$ matrix and a $k_j \times k_j$ matrix, $j = 1, 2$ in computing the optimal encoding matrix in each iteration step [see (1) and (2)]. Recall that inverting an $N \times N$ matrix requires $\mathcal{O}(N^3)$ operations. Therefore, the total complexity of this step at a single iteration step is $\mathcal{O}((N - M + k_1)^3 + k_1^3)$ (given the encoding matrix of sensor 2 is fixed) or $\mathcal{O}((M + k_2)^3 + k_2^3)$ operations (given the encoding matrix of sensor 1 is fixed). This amount of operations increases cubicly with respect to N, and the size of the encoded data, $k_1 + k_2$. Also, the calculations of $\mathbf{A}$ and $\mathbf{\Lambda}$ in (1) and (2), respectively, involve the product of matrices whose dimensions are proportional to the data size (e.g., $\mathbf{\Lambda}$ requires to multiply an $N \times M$ matrix by an $M \times M$ matrix and this involves $\mathcal{O}(NM^2)$ operations). Finally, we also need to perform an eigenvalue decomposition of $\mathbf{\Lambda}$ which its complexity also increases cubicly with respect to N. In summary, the calculation of the iterative local KLT method per iteration step increases cubically with respect to the data size.

B. The Greedy Algorithm

The complexity of the greedy algorithm in each step involves three parts: the calculation of $\mathbf{\Sigma}_i$ [expressed in (40)], the calculation of its largest eigenvector, and obtaining the encoding vector as given in (19).

As mentioned earlier, given the matrix $\mathbf{\Sigma}_i$, we can use a known method for computing the largest eigenvector, $\mathbf{q}_1$, (e.g., power method [27] which has a quadratic complexity with respect to the data size). Also notice that according to (19), in order to obtain the encoding vector, we need to multiply the $N \times 1$ vector $\mathbf{q}_1$ with an $N \times M$ matrix $\mathbf{P}_i$. The complexity of this step is $\mathcal{O}(NM)$ operations which also increases quadratically with respect to the data size.

We now focus on the first part of the complexity of the greedy algorithm which is the calculation of $\mathbf{\Sigma}_i$. We show that this calculation also increases quadratically with respect to the data size. According to the recursive expression of $\mathbf{\Sigma}_i$ given in (40), we need to consider the computation of the updating matrix $\mathbf{B}_i$ which depends on the vectors $\boldsymbol{\omega}_i$, $\boldsymbol{\psi}_i$, and $\boldsymbol{\rho}_i$, and the scalars μ_i and ξ_i. Each of the four terms in (41) is an outer product of two $N \times 1$ vectors, and thus given these two vectors, the complexity of calculating each outer product is $\mathcal{O}(N^2)$ operations.

Next we consider the complexity of calculating $\boldsymbol{\omega}_i$, $\boldsymbol{\psi}_i$ and μ_i. We see that given the vectors $\boldsymbol{\rho}_i$, $\boldsymbol{\theta}_i$, and $\boldsymbol{\gamma}_i$, the complexity of calculating $\boldsymbol{\omega}_i$ in (42) involves the computation of the product $\boldsymbol{\rho}_i\boldsymbol{\theta}_i^T$, which requires $\mathcal{O}(NM)$ operations, and then the multiplication of the $N \times M$ matrix $\mathbf{P}_{i-1} + \boldsymbol{\rho}_i\boldsymbol{\theta}_i^T$ by the $M \times 1$ vector $\boldsymbol{\gamma}_i$, which also requires $\mathcal{O}(NM)$ operations. The calculation of $\boldsymbol{\psi}_i$ in (43) involves the multiplication of the $N \times M$ matrix $\mathbf{P}_{i-1}\mathbf{J}_{i-1}$ (known from the previous step) by the $M \times 1$ vector $\boldsymbol{\theta}_i$, which requires $\mathcal{O}(NM)$ operations. Finally, the calculation of the scalar μ_i in (44) involves the multiplication of the $M \times M$ matrix $\mathbf{J}_{i-1}$ by the $M \times 1$ vector $\boldsymbol{\theta}_i$, which requires $\mathcal{O}(M^2)$ operations, and then the multiplication of the $M \times 1$ vector $\mathbf{J}_{i-1}\boldsymbol{\theta}_i$ by the $M \times 1$ vector $\boldsymbol{\theta}_i$, which also requires $\mathcal{O}(M^2)$ operations. We therefore conclude that given the vectors $\boldsymbol{\rho}_i$, $\boldsymbol{\theta}_i$ and $\boldsymbol{\gamma}_i$ and the scalar ξ_i, the calculation of $\mathbf{B}_i$, which is used to update the matrix $\mathbf{\Sigma}_i$, increases quadratically with respect to the data size.

Finally, we show that the complexity of calculating the vectors $\boldsymbol{\rho}_i, \boldsymbol{\theta}_i, \boldsymbol{\gamma}_i$, and the scalar ξ_i also increases quadratically with respect to the data size. In order to calculate the vector $\boldsymbol{\rho}_i$ in (30), we need to calculate the vector $\boldsymbol{\alpha}_i$ given in (26). Calculating the vector $\boldsymbol{\alpha}_i$ involves: 1) the computation of $\mathbf{b}_i$ which according to (23) requires $\mathcal{O}(M(N-M))$ operations; 2) the computation of h_i which according to (24) requires $\mathcal{O}((N-M)^2)$ operations. The calculation of $\boldsymbol{\theta}_i$ in (31) involves the multiplication the $M \times (N-M)$ matrix $[\mathbf{G}_{i-1}]_M$ and the $(N-M) \times 1$ vector $\mathbf{b}_i$ which requires $\mathcal{O}(M(N-M))$ operations. To calculate the vector $\boldsymbol{\gamma}_i$ in (38) we need to first compute the vector $\boldsymbol{\beta}_i$ given in (36). The vector $\boldsymbol{\beta}_i$ involves the calculation of $\mathbf{e}_i$, which according to (33) requires the multiplication of the $(i-1) \times N$ matrix $\mathbf{W}_{i-1}^T\mathbf{\Sigma}_{\mathbf{rr}}$ (known from the previous step) by the $N \times 1$ vector $\mathbf{w}_i$ which requires $\mathcal{O}((i-1)N)$ operations. Given $\mathbf{e}_i$, the calculation of the vector $\boldsymbol{\beta}_i$ requires the multiplication of the $N \times (i-1)$ matrix $\mathbf{W}_{i-1}\mathbf{H}_{i-1}$ (known from the previous step) by the $(i-1) \times 1$ vector $\mathbf{e}_i$ which requires $\mathcal{O}((i-1)N)$ operations. Given $\boldsymbol{\beta}_i$, the calculation of $\boldsymbol{\gamma}_i$ in (38) involves the multiplication of the $M \times N$ matrix $\mathbf{\Sigma}_{\mathbf{x_1r}}$ by the $N \times 1$ vector $\boldsymbol{\beta}_i$ which requires $\mathcal{O}(MN)$ operations. Finally, the computation of the scalar ξ_i in (39) involves the computation of the norm of $\boldsymbol{\gamma}_i$ which requires $\mathcal{O}(N)$ operations, and the calculation of the scalar s_i, which according to (34) requires $\mathcal{O}(N^2 + (i-1)^2)$ operations.

Therefore, we showed that the complexity of computing the matrix $\mathbf{\Sigma}_i$ increases quadratically with respect to the data size.

VII. Numerical Examples

In this section, we compare the MSE performance of the proposed algorithm with the MSE performance of the joint KLT, marginal KLT, and the iterative local KLT. The discussion on the reconstruction of the data vector and the MSEs using the joint KLT and the marginal KLT is described in [9]. Since the joint KLT, marginal KLT, and the iterative local KLT are discussed in [9] for the noiseless case, we assume in the simulations that the vector of interest $\mathbf{x}$ is observed without noise. Also, in our simulations we apply the power method [27, Algorithm 4.4.1] to compute the largest eigenvector in each step of the greedy algorithm.

A. Example 1

In the first three simulations we consider the example given in [9, p. 5186]. Suppose $\mathbf{\Sigma}_{\mathbf{xx}}$ is a symmetric Toeplitz matrix with first row $(1, \rho, \ldots, \rho^{N-1})$, where ρ is the decay parameter of the Toeplitz covariance matrix, $\mathbf{x}_1$ contains the odd-indexed components of $\mathbf{x}$, and $\mathbf{x}_2$ the even-indexed components. Assume that $N = 40$, and $M = 20$. For the iterative local KLT we used a tolerance of $\epsilon = 0.01$.

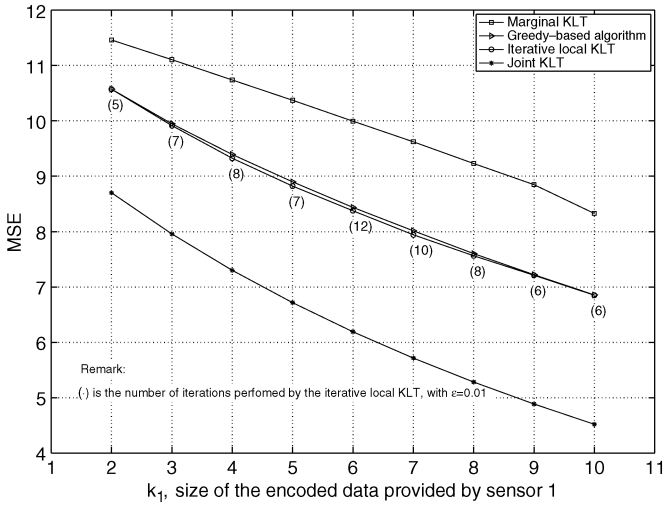

Fig. 2. MSE comparison of the joint KLT, marginal KLT, iterative local KLT, and the greedy approach versus the size of the encoded data of the first sensor for Example 1.

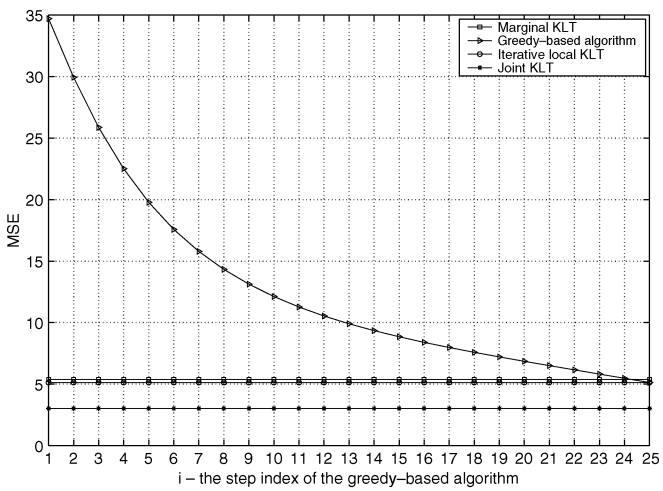

Fig. 3. MSE of the greedy approach versus the step index of the algorithm, compared with the MSE of the joint KLT, marginal KLT, and iterative local KLT for Example 1.

In the first comparison we calculated the MSE of each method as a function of the required number of encoded coefficients k_1, with $k_2 = 10$ fixed, and $\rho = 0.7$. We varied k_1 from 2 to 10 with a step of 1. The case of $k_1 = 10$ was considered in [9, Example 6, p. 5186]. The results are plotted in Fig. 2. The number in the parenthesis below the line associated with the results of the iterative local KLT is the number of iteration steps that were required for this method to reach the predefined tolerance. As can be seen the MSE of the proposed algorithm coincides with the MSE of the iterative local KLT for almost all values of k_1.

In the second comparison $k_1 = 10$ and $k_2 = 15$ with $\rho = 0.7$. In Fig. 3 we plotted the MSE of the greedy approach as a function of its step index, i. Also, plotted are the MSEs of the joint KLT, marginal KLT, and iterative local KLT. The number of iteration steps that were required for the iterative local KLT to reach the predefined tolerance is 5. As can be seen the MSE of the greedy approach decreases until it achieves the MSE of the iterative local KLT approach.

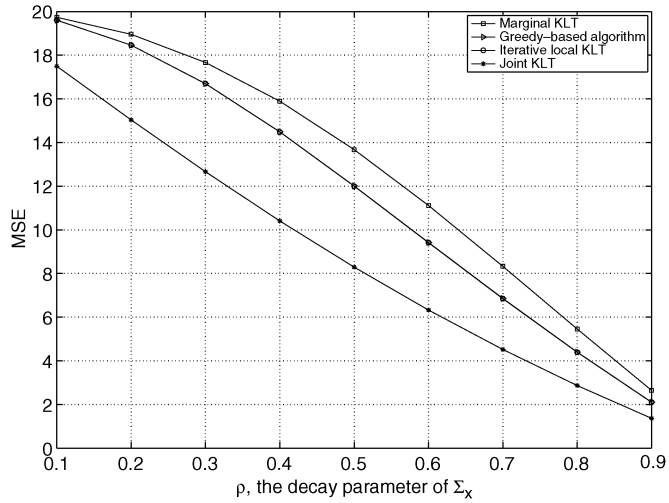

Fig. 4. MSE comparison of the joint KLT, marginal KLT, iterative local KLT, and the greedy approach versus ρ, the decay parameter of the Toeplitz covariance matrix for Example 1.

Finally, in the third comparison we evaluated the MSEs of the methods versus ρ, for $k_1 = k_2 = 10$. We varied ρ from 0.1 to 0.9 with a step of 0.1. The results are shown in Fig. 4. As can be seen the MSE performance of the proposed algorithm is the same as the MSE performance of the iterative algorithm.

B. Example 2

Consider now another example which will further demonstrate the gap between the MSE of the marginal KLT and the MSEs of the joint KLT, the iterative KLT, and the greedy algorithm. Assume that the first sensor observes the vector $\mathbf{x}_1$ which is a zero mean vector with covariance $\mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_1}$, while the second sensor observes a noisy version of $\mathbf{x}_1$ with an additive noise, that is, $\mathbf{x}_2 = \mathbf{x}_1 + \mathbf{n}_2$ where $\mathbf{n}_2$ is a zero mean white noise with variance η, independent with $\mathbf{x}_1$. This situation therefore corresponds to the case where the covariance matrix of the entire data vector $\mathbf{r}$ is given as

$$\mathbf{\Sigma}_{\mathbf{rr}} = \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_1} & \mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_1} \\ \mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_1} & \mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_1} + \eta\mathbf{I} \end{bmatrix}.$$

This example is applicable if we have communication limitations imposed such as capacity limits or battery constraints (that is, each sensor can provide to the fusion center only a limited number of coefficients). From a purely centralized perspective, the second sensor does not contribute to the reduction of the MSE, but due to communication limitations there is value in receiving some of its samples that are not covered by the first sensor.

Notice that η affects the similarity between the two observed vectors. That is, as η becomes smaller, the two sensors actually observe (almost) the same vector. For simplicity assume that $\mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_1} = \mathrm{diag}(\gamma_1, \gamma_2, \ldots, \gamma_M)$ where $\gamma_1 \geq \gamma_2 \geq \cdots \gamma_M$, that is, the elements of $\mathbf{x}_1$ are independent with zero mean and variance equal to γ_m, $m = 1, \ldots, M$. We also assume that these eigenvalues are given as follows: $\gamma_m = (1/2)(1 + ((1 - \exp(x(m)))/(1 + \exp(x(m)))))$, where $m = 1, \ldots, M$ and $x(m) = -3 + (6/19)(m - 1)$. This function considers a smooth

TABLE I
THE MSEs OF THE METHODS FOLLOWING THE RESULTS IN THE UPPER PLOT OF FIG. 5

MSE (marginal)	3.86 ($k_1 = 11$)	3.01 ($k_1 = 12$)	2.30 ($k_1 = 13$)	1.72 ($k_1 = 14$)	1.25 ($k_1 = 15$)
MSE (joint/iterative/greedy)	3.90 ($k_1 = 1$)	3.04 ($k_1 = 2$)	2.33 ($k_1 = 3$)	1.75 ($k_1 = 4$)	1.28 ($k_1 = 5$)

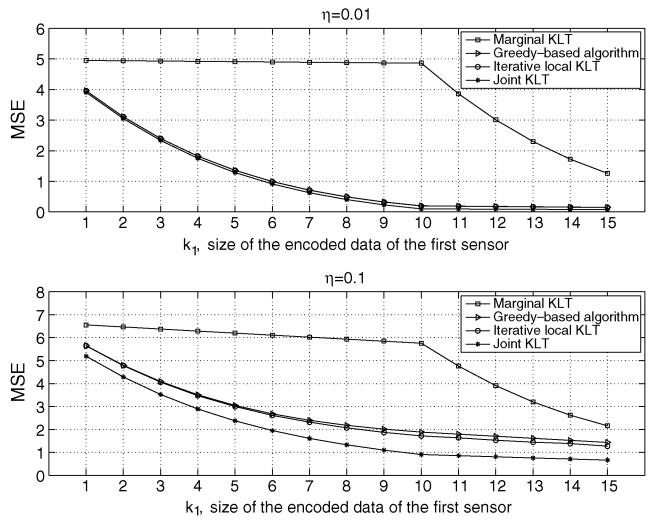

Fig. 5. MSE comparison of the joint KLT, marginal KLT, iterative local KLT, and the greedy approach versus the size of the encoded data of the first sensor for Example 2.

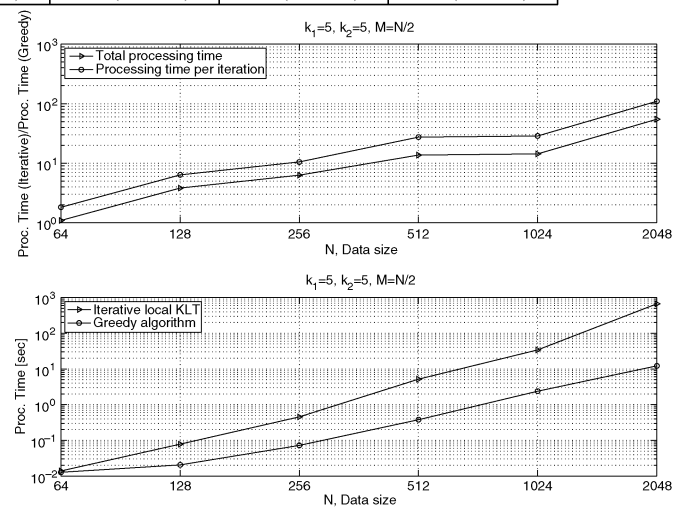

Fig. 6. Processing time comparison of the iterative local KLT and the greedy approach versus the data size.

change of the eigenvalues. Of course, it is also possible to consider other functions.

We calculated the MSE of each method as a function of the required number of encoded coefficients k_1, with $k_2 = 10$ fixed, $N = 40$ and $M = 20$. We varied k_1 from 5 to 15 with a step of 1. We considered the case when $\eta = 0.01$, and when $\eta = 0.1$. The results are plotted in Fig. 5. As can be seen, the MSE of the proposed algorithm is similar to the MSE of the iterative local KLT for almost all values of k_1. Also, as the variance of η decreases, the MSE of the iterative algorithm and the MSE of the proposed method become closer to the MSE of the joint KLT. The plots demonstrate that the gap between the MSE of the marginal KLT and the other methods can be very large.

In Table I we present specific values of the MSEs of the methods for the case $\eta = 0.01$ obtained in the last simulation. As can be seen, for example, the MSE of the marginal KLT for $k_1 = 11, 12, \ldots, 15$ is similar to the MSEs of the joint KLT, iterative KLT, and greedy algorithm for $k_1 = 1, 2, \ldots, 5$. This can be explained as follows. Consider the case when $k_1 = 11$. Since η is very small, the two sensors observe (almost) the same data vector. However, since the marginal KLT ignores the correlation between the sensors, then the first sensor provides the optimal $k_2 = 10$ encoding vectors which are similar to the first 10 encoding vectors provided by the first sensor. Only the 11th encoding vector of the first sensor provides an innovation to the vectors of the second sensor. Hence, although in total the two sensors provide 21 encoding vectors, only 11 vectors of them are distinct. Thus, this reduces to the case where there is a joint processing and the total number of encoding vectors is 11. Since $k_2 = 10$, this means that the first sensor only needs to provide one encoding vector, i.e. $k_1 = 1$. Therefore, the MSE of the marginal KLT for $k_1 = 11$ is similar to the MSEs of joint KLT, iterative KLT, and greedy algorithm for $k_1 = 1$ (as explained previously, for small η the MSEs of the iterative KLT and the greedy algorithm are similar to the MSE of the joint KLT). Similarly, we explain the results for $k_1 = 12, \ldots, 15$. The differences between the MSE of the marginal KLT (for $k_1 = 11, 12, \ldots, 15$) and the MSEs of the other methods decrease as η decreases.

C. Processing Time Comparison

We now discuss the tradeoff between the computational complexity and the reconstruction MSE error achieved by the greedy approach and the iterative local KLT. We compared the processing time (using MATLAB time commands), required for the iterative local KLT method and the greedy method to reach the MSE, as a function of the data size, N. We consider the setup of example 1: a Toeplitz covariance matrix with a decay parameter of $\rho = 0.7$, $M = N/2$, $k_2 = 5$, $k_1 = 5$, and $\epsilon = 0.01$. We used the following data sizes: $N = 64$, 128, 256, 512, 1012, 2048. For each value of N we calculate the MSE, number of iterations required for the iterative local KLT method, and the processing time (using MATLAB time commands) of the iterative local KLT method and the greedy method versus N. In Fig. 6 (upper subplot) we plotted the ratio between the processing time of the iterative local KLT and the processing time of the greedy algorithm, using the absolute processing times, and also by normalizing the total processing time of each method by its number of iterations, and in Fig. 6 (lower subplot) we plotted the absolute processing time of the iterative local KLT and the processing time of the greedy algorithm. In Table II we show the MSE and the corresponding number of iterations required by the iterative local KLT. It can be seen in Fig. 6 that the complexity of the greedy method is much smaller than the iterative

TABLE II
THE MSE AND THE NUMBER OF ITERATIONS OF THE ITERATIVE LOCAL KLT METHOD FOR $k_1 = 5$, $k_2 = 5$, $M = N/2$, AND $\epsilon = 10^{-2}$ VERSUS THE DATA SIZE ASSOCIATED WITH THE RESULTS OF FIG. 6

N	64	128	256	512	1024	2048
MSE	27.2	80.5	203.6	458.3	970	1994
Iter.	6	6	6	5	5	5

local KLT method (for $N = 2048$ the normalized processing time ratio is about 100).

VIII. CONCLUSION

In the distributed KLT problem, only parts of the entire data are observed by different sensors, which provide linearly encoded data to the fusion center. It is impossible to apply a centralized KLT to the entire data vector. The local KLT method proposed in [9] to reconstruct the data vector is an iterative procedure that depends on the predefined tolerance. Instead, we propose a greedy approach which is based on a predefined number of successive steps. In each step the fusion center selects the sensor for which the MSE in reconstructing the entire data vector will be the smallest, if letting this sensor increase the dimension of its current encoding matrix with a new and optimally determined encoding vector. The proposed approach requires fewer computations than the iterative approach and can also be implemented recursively. Therefore, we expect the proposed algorithm to be important in "on-line" implementations of the distributed KLT. Simulations demonstrate that the MSE of the proposed algorithm achieves the MSE of the iterative local KLT method.

APPENDIX

We present a proof of Result 4.1. As mentioned in Section IV, this result is based on some minor modifications of the proof given in [9, Theorem 2] as we explain herein. We start by noting that the goal is to minimize the MSE which can be expressed as

$$D_x = E\left[\|\mathbf{x} - \hat{\mathbf{x}}\|^2\right] = E\left[\|\mathbf{x}_1 - \hat{\mathbf{x}}_1\|^2\right] + E\left[\|\mathbf{x}_2 - \hat{\mathbf{x}}_2\|^2\right]. \tag{51}$$

We express $\mathbf{x}_2$ as a linear combination of the observation $\mathbf{r}_1$ and the encoded data $\mathbf{y}_{2,i}$ (refer to [9, Eq. (58), (66)] for a similar approach)

$$\begin{aligned}\mathbf{x}_2(\mathbf{r}_2, \mathbf{y}_{2,i}, \boldsymbol{\varepsilon}) &= \mathbf{A}^{(q_i)}\begin{bmatrix}\mathbf{r}_1\\ \mathbf{y}_{2,i}\end{bmatrix} + \boldsymbol{\varepsilon}\\ &= \left(\left[\left(\mathbf{A}^{(q_i)}\right)^T\right]_M\right)^T \mathbf{r}_1\\ &\quad + \left(\left[\left(\mathbf{A}^{(q_i)}\right)^T\right]_{q_i}\right)^T \mathbf{y}_{2,i} + \boldsymbol{\varepsilon}\end{aligned} \tag{52}$$

where $\mathbf{A}^{(q_i)}$ is expressed in (10), and $\boldsymbol{\varepsilon}$ is independent with $\mathbf{r}_1$ and $\mathbf{y}_{2,i}$. Substituting (52) into (51) yields

$$D_x = E\left[\|\mathbf{x}_1 - \hat{\mathbf{x}}_1\|^2\right] + E\left[\|\mathbf{x}_2(\mathbf{r}_2, \mathbf{y}_{2,i}, \boldsymbol{\varepsilon}) - \hat{\mathbf{x}}_2\|^2\right]. \tag{53}$$

Given the encoded data $\mathbf{y}_{2,i}$ the estimate $\hat{\mathbf{x}}_2$ which minimizes (53) is

$$\begin{aligned}\hat{\mathbf{x}}_2 &= \hat{E}\left[\mathbf{x}_2(\mathbf{r}_2, \mathbf{y}_{2,i}, \boldsymbol{\varepsilon})|\mathbf{y}_{2,i}\right]\\ &= \left(\left[\left(\mathbf{A}^{(q_i)}\right)^T\right]_M\right)^T \hat{E}[\mathbf{r}_1|\mathbf{y}_{2,i}]\\ &\quad + \left(\left[\left(\mathbf{A}^{(q_i)}\right)^T\right]_{q_i}\right)^T \mathbf{y}_{2,i}.\end{aligned} \tag{54}$$

Since $\mathbf{r}_1 = \mathbf{x}_1 + \mathbf{n}_1$ and $\mathbf{n}_1$ is independent with $\mathbf{y}_{2,i}$ we get that $\hat{E}[\mathbf{r}_1|\mathbf{y}_{2,i}] = \hat{E}[\mathbf{x}_1|\mathbf{y}_{2,i}] \triangleq \hat{\mathbf{x}}_1$. Substituting (54) into (53) results in

$$D_x = E\left[\|\mathbf{P}_i(\mathbf{x}_1 - \hat{\mathbf{x}}_1)\|^2\right] + E\left[\|\boldsymbol{\varepsilon}\|^2\right] \tag{55}$$

where $\mathbf{P}_i$ is defined in (14). We can now rewrite (55) as an expectation conditioned on the available encoded data $\mathbf{z}_i$ at the end of the ith step (similar to [9, Eq. (69)])

$$D_x = E\left[E\left[\|\mathbf{P}_i(\mathbf{x}_1 - \hat{\mathbf{x}}_1)\|^2 |\mathbf{z}_i\right]\right] + E\left[\|\boldsymbol{\varepsilon}\|^2\right]. \tag{56}$$

Similarly to [9, Eq. (59)] we express $\mathbf{P}_i\mathbf{x}_1$ as a linear combination of $\mathbf{z}_i$ and an independent random vector $\boldsymbol{\xi}$ as

$$\mathbf{P}_i\mathbf{x}_1 = \mathbf{B}\mathbf{z}_i + \boldsymbol{\xi} \tag{57}$$

Note that $\boldsymbol{\xi}$ is a zero mean real-valued random vector with a covariance matrix given in (13). Define the $M \times M$ matrix $\mathbf{C}_{1,i} \triangleq [\mathbf{Q}_i^T]_M \mathbf{P}_i$ where $\mathbf{Q}_i$ is presented in (18). Multiplying the inner term in (56) from the left by $[\mathbf{Q}_i^T]_M$ leads to

$$D_x = E\left[E\left[\|\mathbf{C}_{1,i}\mathbf{x}_1 - \mathbf{C}_{1,i}\hat{\mathbf{x}}_1\|^2|\mathbf{z}_i\right]\right] + E\left[\|\boldsymbol{\varepsilon}\|^2\right]. \tag{58}$$

Similarly to [9, Eq. (60)] it can be shown that $\mathbf{C}_{1,i}\mathbf{x}_1$ is a zero mean uncorrelated real-valued random vector with a covariance matrix given by $\mathrm{diag}(\lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,M})$. The result in (58) is minimized by selecting the desired number of rows in $\mathbf{C}_{1,i}$ (see [9, Eq. (72)–(76)]). In our case we are only interested in the first row denoted by $\mathbf{u}_{q_i+1}$. This completes the proof.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful and constructive comments which helped to improve and clarify the paper, and also thank Dr. Geert Leus for his useful remarks and suggestions.

REFERENCES

[1] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. 1st ACM Int. Workshop on Wireless Sens. Netw. Appl.*, Atlanta, GA, 2002, pp. 88–97.

[3] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP 2001)*, Salt Lake City, UT, May 7–11, 2001, pp. 2033–2036.

[4] A. Milenkovic, C. Otto, and E. Jovanov, "Wireless sensor networks for personal health monitoring: Issues and an implementation," *Comput. Commun.*, vol. 29, no. 13–14, pp. 2521–2533, Aug. 2006.
[5] J. C. Chen, Y. Kung, and R. E. Hudson, "Source localization and beamforming," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 30–39, Mar. 2002.
[6] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.
[7] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.
[8] V. K. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 9–21, Sep. 2001.
[9] M. Gastpar, P. L. Dragotti, and M. Vetterli, "The distributed Karhunen-Loève transform," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5177–5196, Dec. 2006.
[10] H. I. Nurdin, R. R. Mazumdar, and A. Bagchi, "Reduced dimension linear transform coding of distributed correlated signals with incomplete observations," *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2848–2858, Jun. 2009.
[11] O. Roy and M. Vetterli, "Dimensionality reduction for distributed estimation in the infinite dimensional regime," *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1655–1669, Apr. 2008.
[12] N. Goela and M. Gastpar, "Distributed Karhunen-Loève transform with nested subspaces," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP 2009)*, Taipei, Taiwan, Apr. 19–24, 2009, pp. 2405–2408.
[13] N. Goela and M. Gastpar, "Linear compressive networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT 2009)*, Seoul, Korea, Jun. 28–Jul. 3 2009, pp. 159–163.
[14] A. Wiesel and A. O. Hero, "Principal component analysis in decomposable Gaussian graphical models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP 2009)*, Taipei, Taiwan, Apr. 19–24, 2009, pp. 1537–1540.
[15] X. Jin-Jun, A. Ribeiro, Z. Q. Luo, and G. B. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 27–41, Jul. 2006.
[16] K. Zhang, X. R. Li, P. Zhang, and H. Li, "Optimal linear estimation fusion-Part IV: Sensor data compression," in *Proc. 6th Int. Conf. Inf. Fusion*, Australia, Jul. 2003.
[17] K. Zhang, "Best linear unbiased estimation fusion with constraints," Ph.D. dissertation, Univ. New Orleans, New Orleans, LA, 2003.
[18] I. D. Schizas, G. B. Giannakis, and Z. Q. Luo, "Distributed estimation using reduced-dimensionality sensor observations," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4284–4299, Aug. 2007.
[19] A. Amar, A. Leshem, and M. Gastpar, "A greedy approach for the distributed Karhune-Loeve transform," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP 2010)*, Dallas, TX, Mar. 14–19, 2010.
[20] Y. Yamashita and H. Ogawa, "Relative Karhunen-Loeve transform," *IEEE Trans. Signal Process.*, vol. 44, no. 2, pp. 371–378, Feb. 1996.
[21] Y. Hua and W. Liu, "Generalized Karhunen-Loeve transform," *IEEE Signal Process. Lett.*, vol. 5, no. 6, pp. 141–142, Jun. 1998.
[22] L. Scharf, "The SVD and reduced rank signal processing," *Signal Process.*, vol. 25, no. 2, pp. 113–133, Nov. 1991.
[23] A. Torokhti, S. Friedland, and P. Howlet, "Towards generic theory of data compression," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Nice, France, Jun. 24–29, 2007, pp. 291–295.
[24] D. P. Oleary, G. W. Stewart, and J. S. Vandergraft, "Estimating the largest eigenvalue of a positive definite matrix," *Math. Computat.*, vol. 33, no. 148, pp. 1289–1292, Oct. 1979.
[25] B. N. Parlett, H. Simon, and L. M. Stringer, "On estimating the largest eigenvalue with Lanczos algorithm," *Math. Computat.*, vol. 38, no. 157, pp. 153–165, Jan. 1982.
[26] P. Common and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc. IEEE*, vol. 78, no. 8, pp. 1327–1343, Aug. 1990.
[27] J. W. Demmel, *Applied Numerical Linear Algebra*. Philadelphia, PA: Soc. Indust. Appl. Math., 1997.
[28] H. L. Van Trees, *Detection, Estimation, and Modulation Theory—Part IV*. New York: Wiley, 2002.

Alon Amar (S'04–M'09) received the B.Sc. (*cum laude*) degree in electrical engineering from the Technion-Israel Institute of Technology, in 1997, and the M.Sc. degree in electrical engineering from Tel Aviv University, Tel Aviv, Israel, in 2003, and 2009, respectively.

He is currently a Postdoctoral Researcher with the Circuits and Systems Group, Faculty of Electrical Engineering, Mathematics and Computer Science of Delft University of Technology, Delft, The Netherlands. His main research interests include array signal processing with applications to source localization and direction finding, detection and estimation theory, and wireless sensor networks.

Amir Leshem (M'98–SM'06) received the B.Sc. degree (*cum laude*) in mathematics and physics, the M.Sc. degree (*cum laude*) in mathematics, and the Ph.D. degree in mathematics, all from the Hebrew University, Jerusalem, Israel.

He is one of the founders of the School of Electrical and Computer Engineering, Bar-Ilan University, Ramat Gan, Israel, where he is currently an Associate Professor and Head of the signal processing track. His main research interests include multichannel communication, applications of game theory to communication, array and statistical signal processing with applications to sensor arrays and networks, wireless communications, radio-astronomy, and brain research, set theory, logic, and foundations of mathematics.

Michael Gastpar (M'04) received the Dipl. El.-Ing. degree from the Swiss Federal Institute of Technology (ETH), Zurich, in 1997, the M.S. degree from the University of Illinois at Urbana-Champaign, Urbana, in 1999, and the Doctorat es Science degree from the Swiss Federal Institute of Technology (EPFL), Lausanne, in 2002, all in electrical engineering. He was also a student in engineering and philosophy at the University of Edinburgh, Edinburgh, U.K., and the University of Lausanne.

He is currently an Associate Professor with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, and a Professor with the Department of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, Delft, The Netherlands. He was a summer researcher with the Mathematics of Communications Department, Bell Labs, Lucent Technologies, Murray Hill, NJ. His research interests are in network information theory and related coding and signal processing techniques, with applications to sensor networks and neuroscience.

Prof. Gastpar won the 2002 EPFL Best Thesis Award, an NSF CAREER Award in 2004, and an Okawa Foundation Research Grant in 2008. He is an Information Theory Society Distinguished Lecturer (2009–2010). He is currently an Associate Editor for Shannon Theory for the IEEE TRANSACTIONS ON INFORMATION THEORY, and he has served as Technical Program Committee Co-Chair for the 2010 International Symposium on Information Theory, Austin, TX.