A GREEDY APPROACH TO THE DISTRIBUTED KARHUNEN-LOÈVE TRANSFORM

Alon Amar[1], Amir Leshem[2], and Michael Gastpar[3]

1. Faculty of EEMCS, Delft University, Delft, 2628 CD, The Netherlands
2. School of Engineering, Bar Ilan University, Ramat Gan, 52900, Israel
3. Department of EECS, University of California, Berkeley, CA, U.S.A.

ABSTRACT

In the distributed linear source coding problem a set of distributed sensors observe subsets of a data vector, and provide the fusion center with linearly encoded data. The goal is to determine the encoding matrix of each sensor such that the fusion center reconstructs the entire data vector with minimum mean square error (MSE). The recently proposed local Karhunen Loève transform (KLT) 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-based non-iterative algorithm. In each step, one of the encoding matrices is updated by appending an additional row. The algorithm selects in a greedy fashion one sensor that provides the largest improvement in MSE, and terminates when all the encoding matrices reach their predefined encoded data size. The algorithm can be implemented recursively, and it reduces the complexity from cubic dependency on the data size, using the iterative method, to quadratic dependency. This makes it a prime candidate for on-line and real-time implementations of the distributed KLT. Simulation results show that for many covariance matrix types, the MSE performance of the suggested algorithm is equivalent to the iterative approach.

Index Terms— Source coding, distributed Karhunen Loève transform, principal component analysis.

1. INTRODUCTION

Wireless sensor networks attract much attention due to their wide range of applications including military and civilian surveillance, environmental monitoring, and source localization [1].

A key problem is the design of the compression at each sensor such that the fusion center will produce a reconstruction of the observed data vector that will be optimal under certain criterion, such as the MSE. This should be distinguished from the estimation scenario which is a different, though related in a sense, where the goal of the fusion center is to accurately estimate a parameter vector of interest from the compressed sensor observations [2, 3]. 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 [4], and Wyner and Ziv [5]. Herein, however, we consider a different abstraction of compression. Specifically, we view compression as dimensionality reduction by linear projections. In the centralized (non-distributed) setting, this is a classical problem often referred to as the KLT.

Recently, the classical KLT was extended to the distributed case [6, 7, 8]: Suppose there are several spatially distributed sensors, each observes only a disjoint part of the entire data vector. The sensors cannot communicate with each other. Each sensor provides to the fusion center encoded data, which is the result of linearly transforming its input data by an encoding matrix.

The problem is how to determine the encoding matrices in the distributed setup. The recently proposed local KLT approach is performed as follows [6]: 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 re-determines its new encoding matrix. The algorithm terminates when the difference between the MSEs in two subsequent iteration steps is smaller than a predefined tolerance. The algorithm converges at least to a local minimum of the MSE [6].

Herein, we propose a greedy-based non-iterative algorithm. 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 steps, known ahead of time. Simulation results suggest that the MSE performance of the proposed new algorithm is equivalent to the iterative local KLT.

This work was supported in part by NWO-STW under the VICI program (DTC.5893, and STW-10459), and also supported by 3TU.CeDICT: Centre for Dependable ICT Systems.

The proposed algorithm can be implemented recursively, and compared to the iterative approach, the algorithm reduces the computation load from cubic dependency on the data size, using the iterative method, to quadratic dependency. Due to space limitations, details are deferred to [9]. In many of the emerging applications, it may be important 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.

2. PROBLEM FORMULATION

For simplicity we consider the case of two sensors. The extension to a larger number of sensors is straightforward. Assume that each sensor samples a disjoint part of a $N \times 1$ Gaussian real-valued random vector, denoted by $\mathbf{x} \stackrel{\Delta}{=} [x_1, \ldots, x_N]^T$, which has zero mean and covariance $\mathbf{\Sigma}_x \stackrel{\Delta}{=} E[\mathbf{x}\mathbf{x}^T]$. The first sensor observes $\mathbf{x}_1 = [x_1, x_2, \ldots, x_M]^T$, and the second sensor observes $\mathbf{x}_2 = [x_{M+1}, \ldots, x_N]^T$.

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

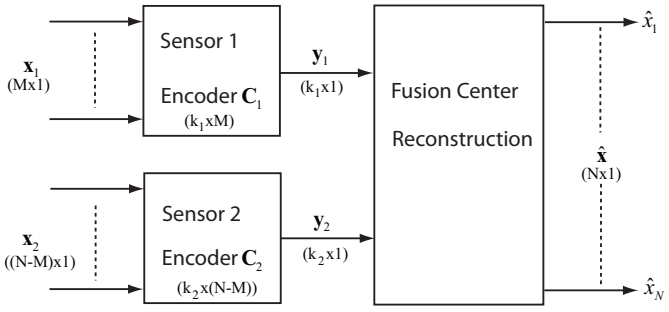

Fig. 1. The distributed KLT setup for the case of two sensors.

3. THE ITERATIVE LOCAL KLT ALGORITHM

We briefly describe the iterative local KLT algorithm [6, Algorithm 1] which was proposed as a suboptimal solution to the problem. Let $\hat{\mathbf{x}}(n)$ denote the estimate of $\mathbf{x}$ at the n-th iteration step, and $D_x(n) \stackrel{\Delta}{=} E[\|\hat{\mathbf{x}}(n) - \mathbf{x}\|^2]$ the MSE. Consider that at the $(n+1)$-th iteration step sensor 2 has 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 $\mathbf{\Lambda}$,

$$\begin{aligned} \mathbf{A} &= \begin{pmatrix} \mathbf{\Sigma}_{12}^T & \mathbf{\Sigma}_{22}\mathbf{C}_2^T(n) \end{pmatrix} \\ &\quad \begin{pmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12}\mathbf{C}_2^T(n) \\ \mathbf{C}_2(n)\mathbf{\Sigma}_{12}^T & \mathbf{C}_2(n)\mathbf{\Sigma}_{22}\mathbf{C}_2^T(n) \end{pmatrix}^{-1} \qquad (1) \\ \mathbf{\Lambda} &= \mathbf{G}(\mathbf{A}; M)(\mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{C}_2^T(n) \\ &\quad (\mathbf{C}_2(n)\mathbf{\Sigma}_{22}\mathbf{C}_2^T(n))^{-1}\mathbf{C}_2(n)\mathbf{\Sigma}_{12}^T)\mathbf{G}(\mathbf{A}; M)^T \\ &\stackrel{\Delta}{=} \mathbf{R}\text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)\mathbf{R}^T \qquad (2) \end{aligned}$$

where $\mathbf{G}(\mathbf{A}; M) \stackrel{\Delta}{=} \begin{pmatrix} \mathbf{I}_M & \left([\mathbf{A}^T]_M\right)^T \end{pmatrix}^T$, and $[\mathbf{A}^T]_M$ is the sub matrix of $\mathbf{A}^T$ consisting the first m rows of $\mathbf{A}$. Also, $\mathbf{\Sigma}_{11} = E[\mathbf{x}_1\mathbf{x}_1^T]$, $\mathbf{\Sigma}_{22} = E[\mathbf{x}_2\mathbf{x}_2^T]$, and $\mathbf{\Sigma}_{12} = E[\mathbf{x}_1\mathbf{x}_2^T]$, $\lambda_1 \geq \cdots \lambda_M \geq \lambda_{M+1} = \cdots = \lambda_N = 0$ are the eigenvectors of $\mathbf{\Lambda}$, and $\mathbf{R}$ is the orthonormal eigenvectors matrix of $\mathbf{\Lambda}$.

The optimal encoding matrix $\mathbf{C}_1(n+1)$ is [6]

$$\mathbf{C}_1(n+1) = \left[\left([\mathbf{\Lambda}^T]_M\right)^T \mathbf{G}(\mathbf{A}; M)\right]_{k_1} \qquad (3)$$

Let $\mathbf{C}(n+1) \stackrel{\Delta}{=} \text{Diag}(\mathbf{C}_1(n+1), \mathbf{C}_2(n))$. The estimate of $\mathbf{x}$ and its MSE at the $(n+1)$-th iteration step are

$$\begin{aligned} \hat{\mathbf{x}}(n+1) &= \mathbf{\Sigma}_x(\mathbf{C}(n+1))^T \qquad (4) \\ &\quad (\mathbf{C}(n+1)\mathbf{\Sigma}_x(\mathbf{C}(n+1))^T)^{-1}\mathbf{y}(n+1) \\ D_x(n+1) &= \text{tr}(\mathbf{\Sigma}_x - \mathbf{\Sigma}_x(\mathbf{C}(n+1))^T \\ &\quad (\mathbf{C}(n+1)\mathbf{\Sigma}_x(\mathbf{C}(n+1))^T)^{-1} \\ &\quad \mathbf{\Sigma}_x\mathbf{C}(n+1)) \qquad (5) \end{aligned}$$

where $\mathbf{y}(n+1) \stackrel{\Delta}{=} \mathbf{C}(n+1)\mathbf{x}$. 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.

4. THE GREEDY-BASED ALGORITHM

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

We start by defining the following:

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

$$d_i \stackrel{\Delta}{=} \begin{cases} 1, & \text{if sensor 1 is selected at the } i\text{-th step} \\ 2, & \text{if sensor 2 is selected at the } i\text{-th step} \end{cases} \qquad (6)$$

- 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 i-th step of sensor 1 and sensor 2, respectively.
- $\mathbf{u}_n$, $n = 1, \ldots, p_i$, and $\mathbf{v}_m$, $m = 1, \ldots, q_i$ are the n-th row and m-th 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 i-th step, and $D_x^{(i)} \triangleq E[\|\hat{\mathbf{x}}^{(i)} - \mathbf{x}\|^2]$ is its MSE.
- $\mathbf{W}_i$ is the $N \times i$ matrix where its m-th column vector, $\mathbf{w}_m$, is

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

- $\mathbf{y}_i \triangleq \mathbf{W}_i^T \mathbf{x}$ is the $i \times 1$ vector that consists of 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 $\mathbf{C}_1^{(p_{i+1})} = \begin{bmatrix} (\mathbf{C}_1^{(p_i)})^T & \mathbf{u}_{p_{i+1}} \end{bmatrix}^T$.
2. Letting sensor 2 add a new row vector, denoted by $\mathbf{v}_{q_{i+1}}^T$, such that $\mathbf{C}_2^{(q_{i+1})} = \begin{bmatrix} (\mathbf{C}_2^{(q_i)})^T & \mathbf{v}_{q_{i+1}} \end{bmatrix}^T$.

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 $\mathbf{y}_i = \mathbf{W}_i^T \mathbf{x}$, 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,

$$d_{i+1} = \begin{cases} 1 & , \varepsilon_{i+1,1} \le \varepsilon_{i+1,2} \\ 2 & , \varepsilon_{i+1,1} > \varepsilon_{i+1,2} \end{cases} \tag{8}$$

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

$$D_x^{(i+1)} = \begin{cases} \varepsilon_{i+1,1} & , \text{if } d_{i+1} = 1 \\ \varepsilon_{i+1,2} & , \text{if } d_{i+1} = 2 \end{cases} \tag{9}$$

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 which is a modification of the proof in [6, Theorem 2] ($\mathbf{v}_{q_{i+1}}$ is obtained similarly with minor changes).

Result 4.1 *Define the $(N-M) \times M$ matrix $\mathbf{A}^{(q_i)}$, and the $N \times N$ matrix $\mathbf{\Sigma}_i$ as*

$$\begin{aligned} \mathbf{A}^{(q_i)} &= \begin{pmatrix} \mathbf{\Sigma}_{12}^T & \mathbf{\Sigma}_{22}(\mathbf{C}_2^{(q_i)})^T \end{pmatrix} \\ &\quad \begin{pmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12}(\mathbf{C}_2^{(q_i)})^T \\ \mathbf{C}_2^{(q_i)}\mathbf{\Sigma}_{12}^T & \mathbf{C}_2^{(q_i)}\mathbf{\Sigma}_{22}(\mathbf{C}_2^{(q_i)})^T \end{pmatrix}^{-1} \qquad (10) \\ \mathbf{\Sigma}_i &= \mathbf{G}(\mathbf{A}^{(q_i)}; M)\,(\mathbf{\Sigma}_{11} - [\mathbf{\Sigma}]_M \mathbf{W}_i \\ &\quad (\mathbf{W}_i^T \mathbf{\Sigma}_x \mathbf{W}_i)^{-1} \mathbf{W}_i^T ([\mathbf{\Sigma}]_M)^T)\, \mathbf{G}(\mathbf{A}^{(q_i)}; M)^T \\ &\triangleq \mathbf{Q}_i diag(\lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,N}) \mathbf{Q}_i^T \qquad (11) \end{aligned}$$

where $\lambda_{i,1} \ge \cdots \lambda_{i,M} \ge \lambda_{i,M+1} = \cdots = \lambda_{i,N} = 0$ are the (non increasingly ordered) eigenvectors of $\mathbf{\Sigma}_i$, and $\mathbf{Q}_i$ is the (orthonormal) eigenvectors of $\mathbf{\Sigma}_i$.

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

$$\mathbf{u}_{q_{i+1}} = \left[\left[\mathbf{Q}_i^T\right]_M \mathbf{G}(\mathbf{A}^{(q_i)}; M) \right]_1 \tag{12}$$

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

$$\varepsilon_{i+1,1} = tr\left(\mathbf{\Sigma}_x - \mathbf{\Sigma}_x \mathbf{F}_{i+1}^T (\mathbf{F}_{i+1} \mathbf{\Sigma}_x \mathbf{F}_{i+1}^T)^{-1} \mathbf{F}_{i+1} \mathbf{\Sigma}_x\right) \tag{13}$$

where $\mathbf{F}_{i+1} \triangleq Diag(\mathbf{C}_1^{(p_{i+1})}, \mathbf{C}_2^{(q_i)})$ is the $(i+1) \times N$ matrix that consists the encoding matrices.

5. NUMERICAL EXAMPLES

We compare the MSE of the proposed algorithm with the MSEs of the joint KLT, marginal KLT, and the iterative local KLT. We consider the example in [6, p.5186]. Suppose $\mathbf{\Sigma}_x$ is a symmetric Toeplitz matrix with first row $(1, \rho, \ldots, \rho^{N-1})$, $\mathbf{x}_1$ and $\mathbf{x}_2$ contain the odd-indexed and the even-indexed components of $\mathbf{x}$, respectively. Also, $N = 40$ and $M = 20$, and $\rho = 0.7$. For the iterative local KLT we used a tolerance of $\epsilon = 0.01$.

We calculated the MSE of each method as a function of the required number of approximations k_1, with $k_2 = 10$ fixed. We varied k_1 from 2 to 10 with a step of 1. The case of $k_1 = 10$ was considered in [6, Example 6]. The results are plotted in Figure 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 coincide with the MSE of the iterative local KLT for almost all values of k_1.

Consider the case when $k_1 = 10$ and $k_2 = 15$. In Figure 3 we plotted the MSE of the greedy-based approach as a function of its step index, i. Also, plotted are the MSEs of the joint KLT, marginal KLT, and iterative local KLT. As can be seen the MSE of the greed-based approach decreases until it achieves the MSE of the iterative local KLT approach.

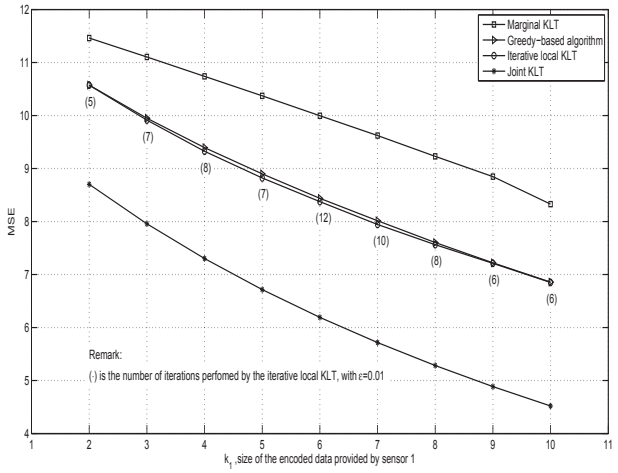

Fig. 2. MSE comparison of the joint KLT, marginal KLT, iterative local KLT, and the greedy-based approach versus the size of the encoded data of the first sensor.

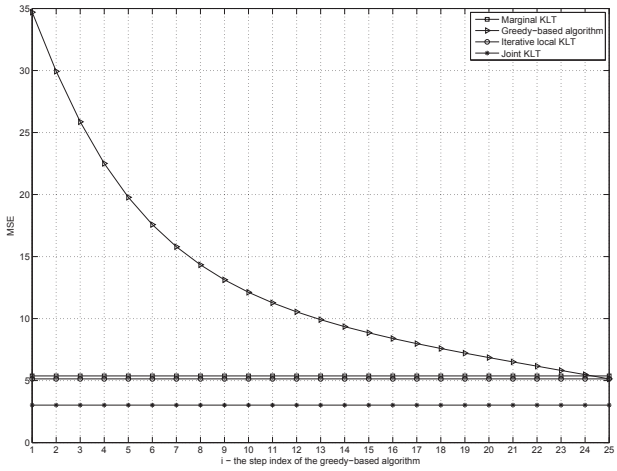

Fig. 3. MSE of the greedy-based approach versus the step index of the algorithm, compared with the MSE of the joint KLT, marginal KLT, and iterative local KLT.

6. CONCLUSIONS

In the distributed KLT problem several sensors observe disjoint parts of the entire data, and then provide linearly encoded data to the fusion center. It is impossible to apply a centralized KLT to the entire data vector. The local KLT is an iterative method that depends on the predefined tolerance. Instead, we propose a non iterative greedy-based approach which is based on successive steps. In each step the fusion center selects the sensor that by letting it increases its encoding matrix with a new, optimally determined, encoding vector, the MSE in reconstructing the entire data vector will be the smallest. The proposed approach reduces the complexity compared to the iterative approach, and can also be implemented recursively. We thus 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.

7. REFERENCES

[1] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks", *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-114, Aug. 2002.

[2] X. Jin-Jun, A. Ribeiro, Z. Q. Luo, and G. B. Giannakis, "Distributed compression-estimation using wireless sensor networks", *IEEE Signal Processing Magazine*, Vol. 23, No. 4, pp. 27-41, Jul. 2006.

[3] K. Zhang, X. R. Li, P. Zhang, and H. Li, "Optimal linear estimation fusion-Part IV: Sensor data compression", *Proceedings of the Sixth International Conference of Information Fusion*, Australia, July 2003.

[4] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources", *IEEE Transactions on Information Theory*, Vol. 19, No. 4, pp. 471-480, Jul. 1973.

[5] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", *IEEE Transactions on Information Theory*, Vol. 22, No. 1, pp. 1–10, Jan. 1976.

[6] M. Gastpar, P. L. Dragotti and M. Vetterli, "The distributed Karhunen-Loève transform", *IEEE Transactions on Information Theory*, Vol. 52, No. 12, pp. 5177–5196, Dec. 2006.

[7] O. Roy, and M. Vetterli, "Dimensionality reduction for distributed estimation in the infinite dimensional regime", *IEEE Transactions on Information Theory*, Vol. 54, No. 4, pp. 1655–1669, Apr. 2008.

[8] H. I. Nurdin, R. R. Mazumdar and A. Bagchi, "Reduced dimension linear transform coding of distributed correlated signals with incomplete observations", *IEEE Transactions on Information Theory*, Vol. 55, No. 6, pp. 2848–2858, Jun. 2009.

[9] A. Amar, A. Leshem, and M. Gastpar, "Recursive implementation of the distributed Karhunen Loève transform", *in preparation*.