COLLABORATIVE SENSOR CACHING VIA SEQUENTIAL COMPRESSED SENSING

Yi-Jen Yang[†], *Ming-Hsun Yang*[‡], *Y.-W. Peter Hong*[†⋆], *and Jwo-Yuh Wu*[‡⋆]

[†]Institute of Communications Engineering, National Tsing Hua University, Taiwan
[‡]Institute of Communications Engineering, National Chiao Tung University, Taiwan
[⋆]MOST Joint Research Center for AI Technology and AII Vista Healthcare

ABSTRACT

This work proposes a collaborative sensor caching and data reconstruction method based on the sequential compressed sensing framework. Here, multiple caches are assumed to exist in the wireless sensor network to store the most recent data gathered from sensors within their respective coverage areas. To reduce the cache size and the data-acquisition overhead, each cache accesses measurements only from a small subset of sensors. This work proposes a collaborative sparse-signal reconstruction method that exploits the presence of sensors simultaneously accessible by multiple caches as anchor nodes to introduce dependency in the reconstruction. The reconstruction is based on the use of the alternating direction method of multipliers (ADMM), which enables distributed implementation of the algorithm. Simulations are provided to demonstrate the effectiveness of the proposed scheme.

Index Terms— Caching, wireless sensor networks, compressed sensing, alternating direction method of multipliers.

1. INTRODUCTION

Wireless sensor networks (WSNs) and internet-of-things consist of a large number of sensors or devices that are deployed to gather information from the environment [1]. The gathered information must often be forwarded back to the data-gathering node or made easily accessible by users. Due to the large-scale and wide deployment of sensors, constant query of information directly from the sensors may be costly. This can be reduced by placing caches optimally within the network to reduce the data access cost, to increase the flexibility, and also to avoid frequent activation of the sensors.

Sensor caching problems have been examined in the literature, e.g., in [2–7], mostly from the networking perspective to reduce data access costs. Specifically, [2] and [3] studied the optimal placement of caches (or storages) to minimize the total cost of gathering data, storage, computation, and replying requests; and [4] proposed the use of cooperative caching among sensors to improve the accessibility of the sensory data. More recently, [5] proposed an adaptive caching mechanism to facilitate sensor virtualization under the sensor-cloud architecture, and [6] utilized caching to improve the efficiency of energy usage in energy harvesting WSNs. Furthermore, [7] examined the joint storage allocation and content placement, and proposed a hierarchical cache-enabled C-RAN to handle the tremendous data traffic of IoT sensing services. While sensor caching has been well-studied from the networking perspective, its integration with advanced signal processing techniques has been less explored.

The main objective of this work is to propose an efficient collaborative sensor caching and data reconstruction method based on the sequential compressive sensing (CS) framework [8]. CS has been adopted as an effective sub-Nyquist signal processing paradigm for WSNs, e.g., in [9], and thus can also serve as an effective tool for reducing the cost and storage of sensor caching. In particular, suppose that multiple caches are deployed in the network to reduce the users' data access cost by buffering data from sensors within its vicinity. To reduce the data-acquisition cost, a cache gathers information only from a subset of sensors at each time instant. The cache only stores data that fall within a most recent time window. By exploiting the sparse representation of the spatially-temporally correlated sensing field, sensor data can be retrieved locally at each cache using CS based reconstruction. In this work, we propose a collaborative reconstruction method where caches can utilize sensors that fall within their common coverage area as anchor nodes to introduce dependency among the reconstruction at different cache locations and, in this way, improve the reconstruction accuracy. Here, we utilize the iterative reweighted ℓ_1-minimization (IRW-ℓ_1) algorithm [10] to perform the reconstruction, and employ on top of that the alternating direction method of multipliers (ADMM) method to distribute the computation among multiple caches. Simulations are provided to demonstrate the effectiveness of the proposed collaborative reconstruction algorithm.

2. SYSTEM MODEL

Let us consider a WSN with N sensors, denoted by the set $\mathcal{N} = \{1, \ldots, N\}$, and C distributed caches, denoted by the

This work is supported in part by Ministry of Science and Technology, Taiwan, under grant MOST 107-2634-F-007-005.

set $\mathcal{C} = \{1, \dots, C\}$. The sensors are deployed to periodically monitor the physical environment whereas the caches are placed to buffer the most recent observations made by sensors in their vicinity. The set of sensors that are accessible by cache c is denoted by $\mathcal{N}_c \triangleq \{i_{c,1}, \dots, i_{c,|\mathcal{N}_c|}\}$, where $|\mathcal{N}_c|$ is the cardinality of the set and $\cup_{c=1}^{C} \mathcal{N}_c = \mathcal{N}$. Note that $\mathcal{N}_1, \dots, \mathcal{N}_c$ are not necessarily disjoint. In fact, the nodes that are simultaneously accessible by multiple caches can serve as anchor for collaborative reconstruction. The size of cache c (i.e., the maximum number of real-valued scalars that can be stored in the cache) is denoted by L_c.

Specifically, by letting $x_i(t)$ be the observation made by sensor i at time t, the observations that are to be stored in cache c at time t can be represented by the $|\mathcal{N}_c| \times W_c$ matrix

$$\mathbf{X}_c(t) \triangleq \begin{bmatrix} x_{i_{c,1}}(t - W_c + 1) & \cdots & x_{i_{c,1}}(t) \\ \vdots & \ddots & \vdots \\ x_{i_{c,|\mathcal{N}_c|}}(t - W_c + 1) & \cdots & x_{i_{c,|\mathcal{N}_c|}}(t) \end{bmatrix} \quad (1)$$

$$= \left[\underline{\mathbf{x}}_c(t - W_c + 1), \cdots, \underline{\mathbf{x}}_c(t)\right] \quad (2)$$

$$= \left[\mathbf{x}_{i_{c,1}}(t), \cdots, \mathbf{x}_{i_{c,|\mathcal{N}_c|}}(t)\right]^T, \quad (3)$$

where W_c is the number of most recent observations stored in cache c, $\underline{\mathbf{x}}_c(t) \triangleq [x_{i_{c,1}}(t), \cdots, x_{i_{c,|\mathcal{N}_c|}}(t)]^T$ is the vector of sensor observations gathered by cache c at time t, and $\mathbf{x}_i(t) \triangleq [x_i(t - W_c + 1), \cdots, x_i(t)]^T$ is the vector of observations made by sensor $i \in \mathcal{N}_c$ over time slots $t - W_c + 1$ to t.

Suppose that there exists sparsifying bases $\mathbf{\Psi}_{c,S} \in \mathbb{R}^{|\mathcal{N}_c| \times |\mathcal{N}_c|}$ and $\mathbf{\Psi}_{c,T} \in \mathbb{R}^{W_c \times W_c}$ in the spatial and temporal domains, respectively, such that $\underline{\mathbf{x}}_c(t) = \mathbf{\Psi}_{c,S} \boldsymbol{\theta}_{c,S}(t)$ and $\mathbf{x}_{i_c}(t) = \mathbf{\Psi}_{c,T} \boldsymbol{\theta}_{i_c,T}(t)$, for all t and for all $i_c \in \mathcal{N}_c$, where $\boldsymbol{\theta}_{c,S}(t) \in \mathbb{R}^{|\mathcal{N}_c|}$ and $\boldsymbol{\theta}_{i_c,T}(t) \in \mathbb{R}^{W_c}$ are sparse vectors. Then, by the theory of Kronecker compressive sensing (KCS) [11] and by the derivations in [8], the spatial and temporal sparsifying bases can be combined into a single transformation matrix to obtain

$$\mathbf{x}_{c,\text{vec}}(t) = \text{vec}(\mathbf{X}_c(t)) = \mathbf{\Psi}_c \mathbf{z}_{c,\text{vec}}(t), \quad (4)$$

where $\mathbf{z}_c(t) = \text{vec}(\mathbf{\Theta}_{c,S}(t) \mathbf{\Psi}_{c,T}^{-T}) \in \mathbb{R}^{|\mathcal{N}_c| W_c}$ with $\mathbf{\Theta}_{c,S}(t) \triangleq \left[\boldsymbol{\theta}_{c,S}(t - W_c + 1), \cdots, \boldsymbol{\theta}_{c,S}(t)\right]$ and $\mathbf{\Psi}_c \triangleq \mathbf{\Psi}_{c,T} \otimes \mathbf{\Psi}_{c,S} \in \mathbb{R}^{|\mathcal{N}_c| W_c \times |\mathcal{N}_c| W_c}$ is a Kronecker sparsifying basis [11].

To reduce communication and storage costs, we assume that each cache, say cache c, only samples observations from $M_c (\ll |\mathcal{N}_c|)$ randomly selected sensors in the set $\mathcal{N}_c$ at time instant t. The set of sampled sensors is denoted by $\mathcal{A}_c(t)$, and the collected measurements can be written as

$$\underline{\mathbf{y}}_c(t) = [y_{c,1}(t) \cdots y_{c,M_c}(t)]^T = \underline{\mathbf{\Phi}}_c(t) \underline{\mathbf{x}}_c(t), \quad (5)$$

where $\underline{\mathbf{\Phi}}_c(t) \in \mathbb{R}^{M_c \times |\mathcal{N}_c|}$ is a sensor identity matrix with each row being a canonical vector with only one entry equal to 1 and all others equal to 0. The position of the 1 indicates the sensor that is sampled. The measurements gathered by cache c over the most recent W_c time slots are stored in the cache at time t and can be stacked into the vector

$$\mathbf{y}_{c,\text{vec}}(t) \triangleq [\underline{\mathbf{y}}_c^T(t - W_c + 1), \cdots, \underline{\mathbf{y}}_c^T(t)]^T \quad (6)$$

$$= \mathbf{\Phi}_c(t) \mathbf{x}_{c,\text{vec}}(t), \quad (7)$$

where $\mathbf{\Phi}_c(t) = \text{diag}(\underline{\mathbf{\Phi}}_c(t - W_c + 1), \cdots, \underline{\mathbf{\Phi}}_c(t))$ is a block diagonal matrix. Since the cache size is L_c, the time window W_c and the number of sampled measurements M_c must be chosen such that $M_c W_c = L_c$. Here, we assume that M_c is fixed over time, but the members of the set may vary. Moreover, let $M_{\text{tot}} = |\cup_{c=1}^{C} \mathcal{A}_c(t)|$ be the total number of sensors sampled at time t. Suppose that, at time t, cache c receives demand for the stored sensor information. Then, in response to the demand, the caches jointly exploit sparsity of the observations $\mathbf{y}_{c,\text{vec}}(t)$, for all c, to perform collaborative reconstruction of the sensor measurements $\mathbf{x}_{c,\text{vec}}(t)$, for all c.

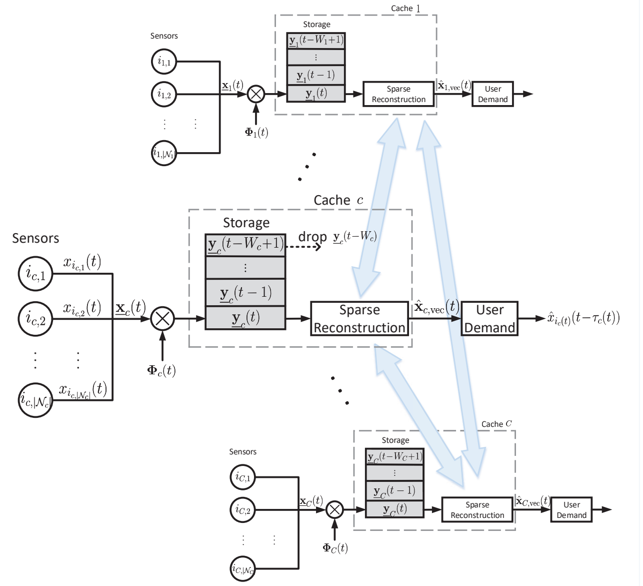
Cache 1
Storage
Sensors
$i_{1,1}$
$i_{1,2}$
$i_{1,|\mathcal{N}_1|}$
$\underline{\mathbf{x}}_1(t)$
$\mathbf{\Phi}_1(t)$
$\underline{\mathbf{y}}_1(t-W_1+1)$
$\underline{\mathbf{y}}_1(t-1)$
$\underline{\mathbf{y}}_1(t)$
Sparse Reconstruction
$\hat{\mathbf{x}}_{1,\text{vec}}(t)$
User Demand
Cache c
Storage
Sensors
$i_{c,1}$
$i_{c,2}$
$i_{c,|\mathcal{N}_c|}$
$x_{i_{c,1}}(t)$
$x_{i_{c,2}}(t)$
$x_{i_{c,|\mathcal{N}_c|}}(t)$
$\underline{\mathbf{x}}_c(t)$
$\mathbf{\Phi}_c(t)$
$\underline{\mathbf{y}}_c(t-W_c+1)$
drop $\underline{\mathbf{y}}_c(t-W_c)$
$\underline{\mathbf{y}}_c(t-1)$
$\underline{\mathbf{y}}_c(t)$
Sparse Reconstruction
$\hat{\mathbf{x}}_{c,\text{vec}}(t)$
User Demand
$\hat{x}_{i_c(t)}(t-\tau_c(t))$
Cache C
Storage
Sensors
$i_{C,1}$
$i_{C,2}$
$i_{C,|\mathcal{N}_C|}$
$\underline{\mathbf{x}}_C(t)$
$\mathbf{\Phi}_C(t)$
$\underline{\mathbf{y}}_C(t-W_C+1)$
$\underline{\mathbf{y}}_C(t-1)$
$\underline{\mathbf{y}}_C(t)$
Sparse Reconstruction
$\hat{\mathbf{x}}_{C,\text{vec}}(t)$
User Demand

Fig. 1. Illustration of a multi-cache system with collaborative reconstruction of sparse sensor signals.

3. MULTI-CACHE COLLABORATIVE RECONSTRUCTION VIA ADMM

In this section, we formulate and solve the collaborative reconstruction problem for the distributed caches in the network. The key idea is to treat the sensors that are simultaneously accessible by different caches as anchor nodes to further improve the reconstruction and then utilize the ADMM algorithm to distribute the computation.

Following the iterative reweighted ℓ_1-minimization (IRW-ℓ_1) problem [10], we formulate the collaborative reconstruction problem as

$$\min_{\{\tilde{\mathbf{z}}_{c,\mathrm{vec}}\}_{c=1}^{C}} \sum_{c=1}^{C} \|\mathbf{G}_c^{(k)}(t)\tilde{\mathbf{z}}_{c,\mathrm{vec}}\|_1 \tag{8a}$$

$$\text{subject to } \mathbf{y}_{c,\mathrm{vec}}(t) = \mathbf{\Phi}_c(t)\mathbf{\Psi}_c\tilde{\mathbf{z}}_{c,\mathrm{vec}},\ \forall c \tag{8b}$$

$$\mathbf{\Upsilon}_{c',c}\mathbf{\Psi}_c\tilde{\mathbf{z}}_{c,\mathrm{vec}} = \mathbf{\Upsilon}_{c,c'}\mathbf{\Psi}_{c'}\tilde{\mathbf{z}}_{c',\mathrm{vec}},\ \forall c, c', \tag{8c}$$

where $\mathbf{\Upsilon}_{c,c'} = \mathrm{diag}(\underline{\mathbf{\Upsilon}}_{c,c'}, \ldots, \underline{\mathbf{\Upsilon}}_{c,c'}) \in \mathbb{R}^{|\mathcal{N}_c \cap \mathcal{N}_{c'}|W_c \times |\mathcal{N}_{c'}|W_c}$ is a block diagonal matrix with $\underline{\mathbf{\Upsilon}}_{c,c'} \in \mathbb{R}^{|\mathcal{N}_c \cap \mathcal{N}_{c'}| \times |\mathcal{N}_{c'}|}$ being a matrix of canonical row vectors with a 1 in each row corresponding to a sensor in $\mathcal{N}_{c'}$ that is simultaneously accessible by cache c. Notice that the sensors simultaneously accessible by different caches are utilized as the anchor for collaborative reconstruction. In the objective function of (8), $\mathbf{G}_c^{(k)}(t) \triangleq \mathrm{diag}(g_{c,1}^{(k)}(t), \ldots, g_{c,|\mathcal{N}_c|W_c}^{(k)}(t))$ is a diagonal matrix of positive weights utilized in iteration k. Suppose that $\hat{\mathbf{z}}_{c,\mathrm{vec}}^{(k)}(t)$ is the solution obtained in the k-th iteration. Then, the weights are updated such that

$$g_{c,i}^{(k+1)}(t) = \left(|\hat{z}_{c,i,\mathrm{vec}}^{(k)}(t)| + \epsilon_0\right)^{-1}, \tag{9}$$

for $i = 1, \ldots, |\mathcal{N}_c|W_c$, where $\hat{z}_{c,i,\mathrm{vec}}^{(k)}(t)$ is the i-th entry of $\hat{\mathbf{z}}_{c,\mathrm{vec}}^{(k)}(t)$. The IRW-$\ell_1$ algorithm is terminated when the weights converge or when the maximum number of iterations $k_{\max}$ is reached. The solution to (8) is denoted by $\hat{\mathbf{z}}_{c,\mathrm{vec}}(t)$, for $c = 1, \ldots, C$. Then, the reconstructed sensor observations at cache c is $\hat{\mathbf{x}}_{c,\mathrm{vec}}(t) = \mathbf{\Psi}_c\hat{\mathbf{z}}_{c,\mathrm{vec}}(t)$.

To enable distributed implementation of the reconstruction algorithm, we adopt the ADMM algorithm to decompose the problem into subproblems that can be solved separately by the local caches. For ease of description, we consider here only the two-cache case, i.e., the case where $C = 2$. Extensions to the general case with $C > 2$ can be done following the consensus ADMM approach presented in [12].

Specifically, let us reformulate the two cache problem as follows:

$$\min_{\tilde{\mathbf{z}}_{1,\mathrm{vec}}, \tilde{\mathbf{z}}_{2,\mathrm{vec}}} \|\mathbf{G}_1^{(k)}(t)\tilde{\mathbf{z}}_{1,\mathrm{vec}}\|_1 + \|\mathbf{G}_2^{(k)}(t)\tilde{\mathbf{z}}_{2,\mathrm{vec}}\|_1 \tag{10a}$$

subject to

$$\begin{bmatrix} \mathbf{\Phi}_1(t)\mathbf{\Psi}_1 \\ \mathbf{0} \\ \mathbf{\Upsilon}_{1,2}\mathbf{\Psi}_1 \end{bmatrix} \tilde{\mathbf{z}}_{1,\mathrm{vec}} + \begin{bmatrix} \mathbf{0} \\ \mathbf{\Phi}_2(t)\mathbf{\Psi}_2 \\ -\mathbf{\Upsilon}_{1,2}\mathbf{\Psi}_2 \end{bmatrix} \tilde{\mathbf{z}}_{2,\mathrm{vec}} = \begin{bmatrix} \mathbf{y}_{1,\mathrm{vec}}(t) \\ \mathbf{y}_{2,\mathrm{vec}}(t) \\ \mathbf{0} \end{bmatrix}. \tag{10b}$$

By letting

$$\mathbf{A} \triangleq \begin{bmatrix} \mathbf{\Phi}_1(t)\mathbf{\Psi}_1 \\ \mathbf{0} \\ \mathbf{\Upsilon}_{1,2}\mathbf{\Psi}_1 \end{bmatrix}, \mathbf{B} \triangleq \begin{bmatrix} \mathbf{0} \\ \mathbf{\Phi}_2(t)\mathbf{\Psi}_2 \\ -\mathbf{\Upsilon}_{1,2}\mathbf{\Psi}_2 \end{bmatrix}, \mathbf{c} \triangleq \begin{bmatrix} \mathbf{y}_{1,\mathrm{vec}}(t) \\ \mathbf{y}_{2,\mathrm{vec}}(t) \\ \mathbf{0} \end{bmatrix}, \tag{11}$$

the augmented Lagrangian of the problem can be written as

$$\begin{aligned} &\mathcal{L}_\rho(\tilde{\mathbf{z}}_{1,\mathrm{vec}}, \tilde{\mathbf{z}}_{2,\mathrm{vec}}, \boldsymbol{\lambda}) \\ &= \sum_{c=1}^{2} \|\mathbf{G}_c^{(k)}(t)\tilde{\mathbf{z}}_{c,\mathrm{vec}}\|_1 + \boldsymbol{\lambda}^T(\mathbf{A}\tilde{\mathbf{z}}_{1,\mathrm{vec}} + \mathbf{B}\tilde{\mathbf{z}}_{2,\mathrm{vec}} - \mathbf{c}) \\ &\quad + \frac{\rho}{2}\|\mathbf{A}\tilde{\mathbf{z}}_{1,\mathrm{vec}} + \mathbf{B}\tilde{\mathbf{z}}_{2,\mathrm{vec}} - \mathbf{c}\|_2^2 \qquad (12) \\ &= \|\mathbf{G}_1^{(k)}(t)\tilde{\mathbf{z}}_{1,\mathrm{vec}}\|_1 + \|\mathbf{G}_2^{(k)}(t)\tilde{\mathbf{z}}_{2,\mathrm{vec}}\|_1 \\ &\quad + \frac{\rho}{2}\|\mathbf{A}\tilde{\mathbf{z}}_{1,\mathrm{vec}} + \mathbf{B}\tilde{\mathbf{z}}_{2,\mathrm{vec}} - \mathbf{c} + \mathbf{u}\|_2^2 - \frac{\rho}{2}\|\mathbf{u}\|_2^2 \qquad (13) \end{aligned}$$

where $\mathbf{u} \triangleq (1/\rho)\boldsymbol{\lambda}$. The resulting ADMM updates are

$$\begin{aligned} \tilde{\mathbf{z}}_{1,\mathrm{vec}}^{(r+1)} &:= \arg\min_{\tilde{\mathbf{z}}_{1,\mathrm{vec}}} \Big\{\|\mathbf{G}^{(k)}(t)_1\tilde{\mathbf{z}}_{1,\mathrm{vec}}\|_1 \\ &\quad + \frac{\rho}{2}\|\mathbf{A}\tilde{\mathbf{z}}_{1,\mathrm{vec}} + \mathbf{B}\tilde{\mathbf{z}}_{2,\mathrm{vec}}^{(r)} - \mathbf{c} + \mathbf{u}^{(r)}\|_2^2\Big\} \qquad (14a) \\ \tilde{\mathbf{z}}_{2,\mathrm{vec}}^{(r+1)} &:= \arg\min_{\tilde{\mathbf{z}}_{2,\mathrm{vec}}} \Big\{\|\mathbf{G}_2^{(k)}(t)\tilde{\mathbf{z}}_{2,\mathrm{vec}}\|_1 \\ &\quad + \frac{\rho}{2}\|\mathbf{A}\tilde{\mathbf{z}}_{1,\mathrm{vec}}^{(r+1)} + \mathbf{B}\tilde{\mathbf{z}}_{2,\mathrm{vec}} - \mathbf{c} + \mathbf{u}^{(r)}\|_2^2\Big\} \qquad (14b) \\ \mathbf{u}^{(r+1)} &:= \mathbf{u}^{(r)} + \rho\big(\mathbf{A}\tilde{\mathbf{z}}_{1,\mathrm{vec}}^{(r+1)} + \mathbf{B}\tilde{\mathbf{z}}_{2,\mathrm{vec}}^{(r+1)} - \mathbf{c}\big), \qquad (14c) \end{aligned}$$

where $\rho > 0$. The updates are performed until the primal and the dual residuals, i.e., $\|\mathbf{A}\tilde{\mathbf{z}}_{1,\mathrm{vec}}^{(r+1)} + \mathbf{B}\tilde{\mathbf{z}}_{2,\mathrm{vec}}^{(r+1)} - \mathbf{c}\|_2^2$ and $\|\tilde{\mathbf{z}}_{c,\mathrm{vec}}^{(r+1)} - \tilde{\mathbf{z}}_{c,\mathrm{vec}}^{(r)}\|_2^2$, converge to zero as r increases.

Notice that (14a) can be reformulated as

$$\min_{\tilde{\mathbf{z}}_{1,\mathrm{vec}}, \mathbf{v}_1} \frac{1}{2}\|\mathbf{A}\tilde{\mathbf{z}}_{1,\mathrm{vec}} - \boldsymbol{\mu}_1^{(r)}\|_2^2 + \frac{1}{\rho}\|\mathbf{v}_1\|_1 \tag{15a}$$

$$\text{subject to } \mathbf{G}_1^{(k)}(t)\tilde{\mathbf{z}}_{1,\mathrm{vec}} - \mathbf{v}_1 = 0, \tag{15b}$$

where $\boldsymbol{\mu}_1^{(r)} \triangleq -\mathbf{B}\tilde{\mathbf{z}}_{2,\mathrm{vec}}^{(r)} + \mathbf{c} - \mathbf{u}^{(r)}$. This problem can again be solved using the ADMM algorithm by first taking the augmented Lagrangian with regularization parameter $\rho_1 > 0$ (similar to the role of ρ in (12)), and by following the closed-form update equations given below:

$$\begin{aligned} \tilde{\mathbf{z}}_{1,\mathrm{vec}}^{(r+1,s+1)} &:= \big(\mathbf{A}^T\mathbf{A} + \rho_1\mathbf{G}_1^{(k)}(t)^T\mathbf{G}_1^{(k)}(t)\big)^{-1}\big[\mathbf{A}^T\boldsymbol{\mu}_1^{(r)} \\ &\quad + \rho_1\mathbf{G}_1^{(k)}(t)^T(\mathbf{v}_1^{(s)} - \mathbf{u}_1^{(s)})\big] \qquad (16a) \\ \mathbf{v}_1^{(s+1)} &:= S_{(1/\rho)/\rho_1}(\mathbf{G}_1^{(k)}(t)\tilde{\mathbf{z}}_{1,\mathrm{vec}}^{(r+1,s+1)} + \mathbf{u}_1^{(s)}) \qquad (16b) \\ \mathbf{u}_1^{(s+1)} &:= \mathbf{u}_1^{(s)} + \mathbf{G}_1(t)\tilde{\mathbf{z}}_{1,\mathrm{vec}}^{(r+1,s+1)} - \mathbf{v}_1^{(s+1)}, \qquad (16c) \end{aligned}$$

where $S_M(\mathbf{a}) \triangleq (\mathbf{a} - M\mathbf{1})_+ - (-\mathbf{a} - M\mathbf{1})_+$ represents element-wise soft-thresholding. The problem in (14b) can be solved following a similar set of update equations as in (15).

4. SIMULATIONS

In this section, simulations are provided to demonstrate the effectiveness of the proposed CS-based collaborative cache

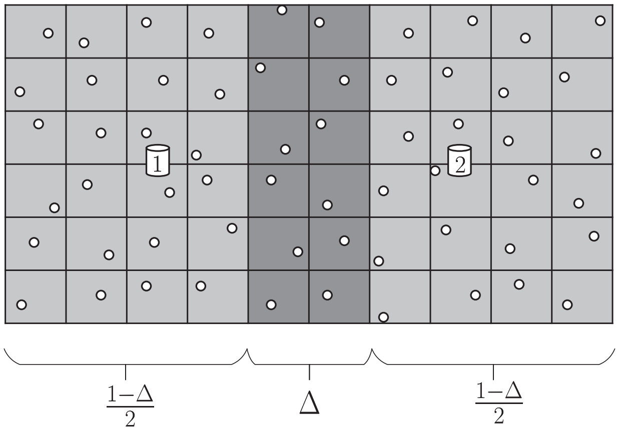

Fig. 2. Illustration of the sensor deployment with two caches placed at positions $(25, 30)$ and $(75, 30)$, respectively.

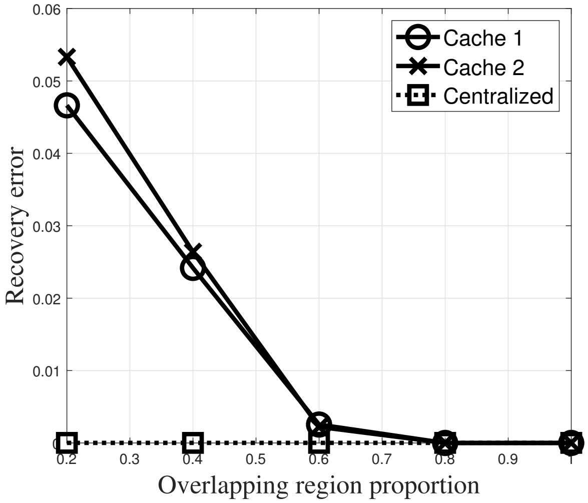

Fig. 3. Normalized MSE versus overlapping portion Δ for the case with $M_{\text{tot}} = 10$ and $W_1 = W_2 = 8$.

reconstruction method. The sensor data is generated following the simulation settings in [8] with an average over 10 network realizations. Specifically, in our experiments, $N = 60$ sensors are deployed over a 100×60 rectangular region. The field is divided into 10×10 square regions and a sensor node is deployed randomly according to the uniform distribution in each square region. An illustration is given in Fig. 2. Here, two caches are placed at positions $(25, 30)$ and $(75, 30)$, respectively. Caches 1 and 2 are able to access sensors in the left and right $\frac{1+\Delta}{2}$ portions of the field. The coverage of the two caches overlap over the middle Δ portion of the field. The M_{tot} sampled measurements are chosen proportionally from the three (overlapping and non-overlapping) regions.

Each sensor observation is assumed to be an aggregate of 6 underlying Gaussian sources distributed randomly according to the uniform distribution throughout the field. Hence, sensor i's observation at time t can be written as $x_i(t) = \sum_{s=1}^{6} e^{-(d_{i,s}/\eta_1)^{\eta_2}} \beta_s(t)$ where η_1 is the correlation length, η_2 is the order used to control the smoothness of the random field, $d_{i,s}$ is the distance between sensor i and source s, and $\beta_s(t)$ is the value of the s-th Gaussian source at time t. Here, we set $\eta_1 = 100$ and $\eta_2 = 2$. Moreover, to incorporate temporal correlation, we assume that $\boldsymbol{\beta}_s \triangleq (\beta_s(1), \ldots, \beta_s(T))^T = \boldsymbol{\lambda}_s + \boldsymbol{\pi}_s$, where the first term has continuous-valued entries that are generated by filtered Gauss-Markov processes, and the second term has discrete-valued entries obtained from a finite-state Markov chain. We refer the readers to [8] for further details on the sensor data generation. However, different from [8], we further truncate the vector $\mathbf{z}_{c,\text{vec}}(t) = \boldsymbol{\Psi}_c^{-1}\mathbf{x}_{c,\text{vec}}(t)$ so that it only preserves the 8 largest entries. These entries contain over 99% of the total signal power.

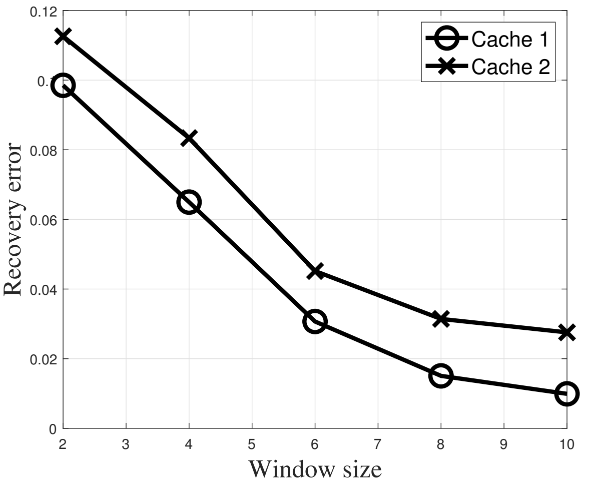

Fig. 4. Normalized MSE versus window size $W_1 = W_2 = W$ for the case where $M_{\text{tot}} = 10$ and $\Delta = 0.4$.

In Fig. 3, we plot the normalized mean-square error (MSE) of the reconstructed sensor data within each cache's coverage versus the overlapping portion Δ. We compare between the proposed ADMM method and the centralized reconstruction where the data from both caches are available. Here, ADMM is run for at least 100 iterations to guarantee good convergence. We can see that the recovery error decreases as the overlapping region increases since more sensors are available in the overlapping region to serve as anchor nodes to strenghten the dependency between the reconstruction at the two caches. In fact, as the overlapping region increases, the performance gradually converges to that of centralized reconstruction.

In Fig. 4, we show the normalized MSE versus the window size $W_1 = W_2 = W$ for the case where $M_{\text{tot}} = 10$. We can see that the normalized MSE decreases as the window size increases since, in this case, more observations are available in the cache to perform the reconstruction.

5. REFERENCES

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292 – 2330, 2008.

[2] L. Ying, Z. Liu, D. Towsley, and C. H. Xia, "Distributed operator placement and data caching in large-scale sensor networks," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, Apr. 2008, pp. 977–985.

[3] B. Sheng, Q. Li, and W. Mao, "Optimize storage placement in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 10, pp. 1437–1450, 2010.

[4] N. Dimokas, D. Katsaros, L. Tassiulas, and Y. Manolopoulos, "High performance, low complexity cooperative caching for wireless sensor networks," *Wireless Networks*, vol. 17, no. 3, pp. 717–737, Apr. 2011.

[5] S. Chatterjee and S. Misra, "Dynamic and adaptive data caching mechanism for virtualization within sensor-cloud," in *IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS)*, Dec. 2014, pp. 1–6.

[6] D. Niyato, D. I. Kim, P. Wang, and L. Song, "A novel caching mechanism for internet of things (IoT) sensing service with energy harvesting," in *Proceedings of IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[7] J. Yao and N. Ansari, "Joint content placement and storage allocation in C-RANs for IoT sensing service," *IEEE Internet of Things Journal*, accepted, 2018.

[8] M. Leinonen, M. Codreanu, and M. Juntti, "Sequential compressed sensing with progressive signal reconstruction in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1622–1635, 2015.

[9] A. Y. Yang, M. Gastpar, R. Bajcsy, and S. S. Sastry, "Distributed sensor perception via sparse representation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1077–1088, 2010.

[10] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ-1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, Dec 2008.

[11] M. F Duarte and R. G Baraniuk, "Kronecker compressive sensing," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 494–504, 2012.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.