Decentralized and Compressed Data Storage for Mobile Crowdsensing

Siwang Zhou, Xiao Zhang, Yonghe Liu, Hongbo Jiang, and Keqin Li, *Fellow, IEEE*

***Abstract*—Sensing data acquired with crowdsensing are generally stored at central cloud servers, since massive data are involved and sensing devices do not have enough space to store them. Although each sensing device only has limited storage capacity, the total size of storage across thousands of devices can be considerable. In view of this, this article addresses decentralized storage problem in mobile crowdsensing system, providing an alternative to cloud-based data storage. By investigating a virtual sensor model, the movement of a participant in the target sensing area is formulated as a random sampling over the data field related to this area. With a particular encoding algorithm, the data field is compressed into only one measurement along with a random sampling process. Each participant stores its own measurements as if various compressed snapshots of the data field are separately stored by different participants. We further investigate a recovery algorithm, reconstructing the original data field by carefully decoding enough measurements. Extensive experiments validate the proposed storage scheme under various crowdsensing scenarios, and our scheme achieves excellent performance in terms of recruitment overhead, decoding time, and decoding accuracy.**

***Index Terms*—Compressed sensing, data recovery, data storage, mobile crowdsensing.**

I. Introduction

NETWORK-BASED applications, including the popular mobile crowdsensing (MCS) systems, usually employ cloud-based storage management [9], [11], [23]. In MCS campaigns, the participants, such as human beings, unmanned aerial vehicles, vehicles and vessels, are required to collect data from their sensing devices [1], [24]. The target sensing area generally relates to a large amount of environmental monitoring data, whereas the sensing devices cannot provide enough storage capacity for crowdsensing tasks. Therefore the participants have to report their data over the network to some collectors typically located in central cloud servers.

Manuscript received 20 November 2022; revised 17 June 2023; accepted 10 July 2023. Date of publication 13 July 2023; date of current version 4 April 2024. This work was supported by the National Science Foundation of China under Grant 62172153. Recommended for acceptance by D. Yang. *(Corresponding author: Siwang Zhou.)*

Siwang Zhou, Xiao Zhang, and Hongbo Jiang are with the College of Computer Science and Electrical Engineering, Hunan University, Changsha 410082, China (e-mail: swzhou@hnu.edu.cn; xiaozhang@hnu.edu.cn; hongbojiang2004@gmail.com).

Yonghe Liu is with the Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019 USA (e-mail: yonghe@cse.uta.edu).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TMC.2023.3294969

Cloud servers are often supposed to have enough space for storing all the data uploaded from MCS systems. However, in some MCS scenarios, the users may need to inquire just a subset of data related to certain interested hot-spot areas. Uploading the entire data field of the whole target area obviously wastes valuable storage space of the cloud server in this situation. Moreover, it will introduce additional network load if numerous participants keep transmitting their sensing data to cloud servers via wireless network infrastructures. In addition, sending data frequently can chew up a lot of electricity of sensing devices, while mobile sensing devices have only limited battery power. There is an urgent need to exploit a new MCS system, where cloud servers focus on computation what they do best, without having to store the data field all the time.

It is noteworthy that, although a participant only has very limited storage capacity, the total storage of an extensive number of participants can be reasonably large. For instance, human beings participate in a MCS campaign, utilizing their smartphones to monitor the information from a target sensing area. Let us assume that one participant uses only 0.01% of its 128 gigabytes of storage to conduct crowdsensing tasks since the smartphone is used mostly for work and life. Even with such small storage space of a participant, the storage of MCS system can still reach as many as 12 terabytes if this system recruits one million participants. It can be fairly said that MCS systems do have certain storage capacities of their own.

However, few researchers study the storage method in MCS systems outside of a cloud server. It is not a trivial task to utilize little pieces of storage space scattered over numerous participants. This is like a normal disk is broken up into a number of tiny storage fragments. To complicate matters further, the participants often move around in the target sensing area, and these storage fragments do not even have a stable spatial distribution. This may be the reason that central cloud-based storage has to be employed in MCS systems. It should be noted that, the storage without employment of central servers has been exploited in traditional wireless sensor networks by utilizing multi-hop routing [14], [19], [31]. Sensor networks contain a large number of nodes deployed in designated locations, and these nodes communicate with each other via a certain routing protocol. Unfortunately, the participants are, for the most part, total strangers moving by themselves in the target sensing area. It is inappropriate for MCS systems to simply adopt the storage strategy employed in sensor networks.

This article investigates a participant-based, decentralized and compressed data storage in MCS systems to utilize the

fragmented storage across different participants. Our motivation is based on an important understanding: one random movement of a participant over the target sensing area can be considered as a random sampling for the data field on this area. We can therefore propose a unique encoding algorithm, compressing the entire data field into one value, termed as measurement, along with a random sampling process. The measurement is like a snapshot of the data field, which is stored by the corresponding participant with the space of just one measurement. Note that this measurement may be a very ambiguous snapshot due to the fact that the data field is represented by only one encoded measurement. However, if enough number of participants store their measurements, one can recover the original data field with desired accuracy by exploring a reasonable decoding algorithm.

Our contributions of this article are summarized below.

- We propose a novel virtual sensor model to abstract the target sensing area, where geographical observation sites are considered as virtual sensor nodes. One random movement of a participant in the monitoring area can then be formulated as one random sampling of the data field corresponding to this area.
- We develop a participant-based compressed storage scheme by designing a special encoding algorithm. Based on one random sampling owing to a participant's movements, the entire data field is compressed into only one measurement, stored by the corresponding participant. We further develop block partition and data exchange strategies, increasing the number of the measurements without affecting the encoding performance.
- We present a data reconstruction scheme with an attention mechanism, ensuring the original data field, when necessary, can be recovered from our decentralized MCS storage system with desired accuracy. We further present the mathematical foundation for successful data recovery, which is validated through extensive experiments.

The remainder of this article is organized as following. Section II reviews the related work. Section III introduces the virtual sensor model, based on which encoding algorithms are further presented. Section IV presents the data recovery algorithm followed by the mathematical foundation for data recovery in Section V. In Section VI we evaluate the performance of the proposed scheme through extensive experiments and conclude in Section VII.

II. Related Work

Mobile crowdsensing is becoming increasingly attractive as a solution for environmental monitoring with the pervasive presence of mobile sensing devices and ever increasing wireless communication capacity [5], [8], [16], [27]. As common for most environmental monitoring systems, the potential large data volume, coupled with limited storage capacity of the mobile sensing devices, has incentivised MCS systems to rely on cloud-based storage. In a typical MCS system, acquired environmental data by a participant's device are reported using wireless network links to the central data collector, generally located within a cloud server [9], [11], [23]. In order to acquire the data field on an area, central cloud servers are also needed to schedule the participants to visit designated observation sites [13], [21], [25]. In contrast, this article investigates decentralized MCS storage scheme, where the entire data field is stored into the sensing devices of the participants independently of any cloud servers.

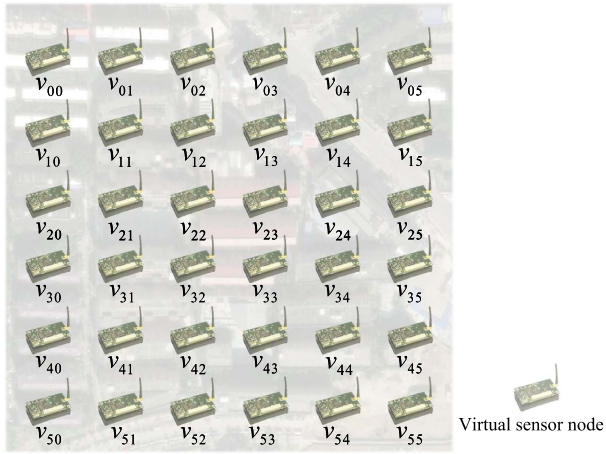


Fig. 1. Virtual sensor model.

Data compression technology is provided to decrease the size of data volume, thus the storage space can be saved. A plethora of compression strategies have been investigated in recent years, including compression algorithms for document data [12], [28], the compression standard for image data with Joint Photographic Experts Group [18], compression standards for audio and video data with Moving Picture Experts Group [2], and compression algorithms toward high-energy single-cycle pulses and streaming spectrum data [20], [26]. Data sharing technology is also introduced in [22] for sharing the data generated by sensing devices, potentially compressing the size of data by avoiding redundancy. Unfortunately, in MCS scenarios, there is not a consistent one-to-one match between each participant and each observation site, and the participant does not even have a normal data set of the target sensing area, since one can hardly expect this participant to visit all observation sites. Therefore conventional data compression and data sharing algorithms are extremely difficult to be applied to MCS systems.

The research on MCS originates from wireless sensor network. Several distributed storage methods without employment of central servers have been proposed in wireless sensor networks [10], [14], [19], [31]. For fault tolerance in case of network failure or the central sink node failure, sensors may temporarily store the data by themselves. The authors in [19] recommend that every sensor node disseminates its data throughout the network by using multi-hop routing. In this way, each node can receive the data from all other nodes in the network. These data are then compressed and the compressed result is stored at these nodes. This method is further improved in [31] to save data dissemination cost. However, data dissemination based decentralized storage is not applicable to MCS, since most of the participants are total strangers and do not communicate with each other. Perhaps for this reason, nearly all the MCS systems still employ cloud storage except reference [32], which exploits a distributed

storage method to temporarily store the data when network outage is encountered. However, this storage strategy only consider the extreme scenario where wireless connectivity between the participants and the server is temporarily interrupted. In this article, we further consider a more general case, i.e., the MCS system itself is seen as a natural participant-based decentralized storage system. The entire data field is stored into a number of participants outside of a cloud server.

III. Compressed Data Storage

This section formulates the movements of the participants as random samplings by defining a virtual sensor model. Then encoding algorithms with block partition and data exchange strategies are proposed. Each participant independently performs encoding and stores its own measurements, without any support from central cloud servers.

A. Data Sampling by Employing Virtual Sensor Model

1) Virtual Sensor Model: We consider the geographical target sensing area as a static virtual sensor model. An example of virtual sensor model is schematically shown in Fig. 1, where v_{ij} denotes the virtual sensor node located at the coordinates of i, j.

Our idea is gotten from classical wireless sensor network, where sensor nodes are deployed at the observation sites in the target area for collecting environmental information. We notice that environmental information is, in fact, irrelevant to physical sensor nodes. Environmental information, if anything, is relevant to the corresponding observation site. Environmental information is always there, even if one does not deploy any physical sensor node. With this in mind, the concept of virtual sensor node is defined in Definition 3.1.

Definition 3.1 (Virtual Sensor Node). A geographical observation site is a virtual sensor node. The information of the observation site is sensed by the corresponding virtual node, just as a real node is deployed at this observation site.

From Definition 3.1, an observation site is imagined as a virtual sensor node, and the target sensing area consists of a number of virtual nodes. Virtual sensor node is like a physical one, but it has many differences from a real one, as illustrated in Properties 3.1 and 3.2.

Property 3.1 (Storage Capacity). The virtual sensor node, like a real physical sensor, is capable of sensing the information of the observation site. However, the virtual node has no storage capacity, and can not store any sensing data.

Property 3.2 (Sensing Time). The sensing time of the virtual sensor node relies on the activities of the participants. When some participant visits an observation site, the corresponding virtual node senses once.

It can be seen from Property 3.1 that, virtual node is memoryless, since it is a virtual one of our imagination. Property 3.2 gives the relationship between the proposed virtual sensor model and the participants in terms of sensing time. When a participant visits an observation site, it acquires and stores the data sensed by the corresponding virtual sensor node.

2) Random Data Sampling: Based on the proposed virtual sensor model, random movements of the participants in the target sensing area can be easily formulated into a set of data sampling processes.

Fig. 2 illustrates the relationship between these two issues. As shown in Fig. 2(a), six participants move randomly in an area, each with its own moving trajectory. The trajectory of a participant corresponds to a random data samplings over the data field related to this area. The sampling processes are shown in Fig. 2(b)–(g), respectively. From Fig. 2, with the data sampling process, each participant obtains its own sampled result of the data field. In crowdsensing scenario, any participant can not be expected to move through all observation sites. A participant may only randomly visit a percentage of visual nodes. That is, the sampled result is a random subset of the data field.

Fortunately, the data in the neighboring observation sites are generally correlated to each other. By exploiting the data correlation, it is possible to deduce the entire data field from a sampled subset of the data with a reasonable encoding algorithm and the corresponding recovery strategy. That is, the participant can acquire the information of the entire data field with some degree of accuracy, although it only conducts random local sampling. Along with the increase of the participants, the target sensing area can be wholly covered by their trajectories, i.e., the data field may be fully sampled with enough participants.

Algorithm 1: Encoding Algorithm.

Input: $\mathbb{V}_l$;
Output: y_l;
1: Initialize $y_l = 0$;
2: **while** $v_{ij} \in \mathbb{V}_l$ **do**
3: $y_l \leftarrow y_l + \phi_{ij} x_{ij}$;
4: **end while**

B. Participant-Based Compressed Data Storage

In this section we introduce an encoding method, with which the data field is compressed into one measurement along with a random sampling process. Each participant stores its own measurement, not the original sampled result. Block partition and data exchanging strategies are further investigated to increase the number of the measurements, without requiring more participants.

1) Encoding Algorithm: Assume that X is the data field related to a target sensing area, x_{ij} denotes the datum sensed by virtual sensor node v_{ij} on the sampling path of participant P_l, $\mathbb{V}_l$ is the set of all virtual nodes visited by P_l, ϕ_{ij} is a random number generated by P_l when virtual node v_{ij} is visited, and y_l is the encoded result, i.e., measurement. The encoding algorithm is shown in Algorithm 1.

We use Fig. 3 as an example to illustrate the encoding process. With Algorithm 1, at every virtual sensor node, participant P_l encodes the sensory data and stores the encoded measurement. For instance, in the first step of the sampling process, $y_l = \phi_{50} x_{50}$, where x_{50} is the original environmental datum sensed by v_{50}, and in the second step, $y_l = \phi_{50} x_{50} + \phi_{41} x_{41}$. P_l performs this encoding operation until it completes the sampling process. At

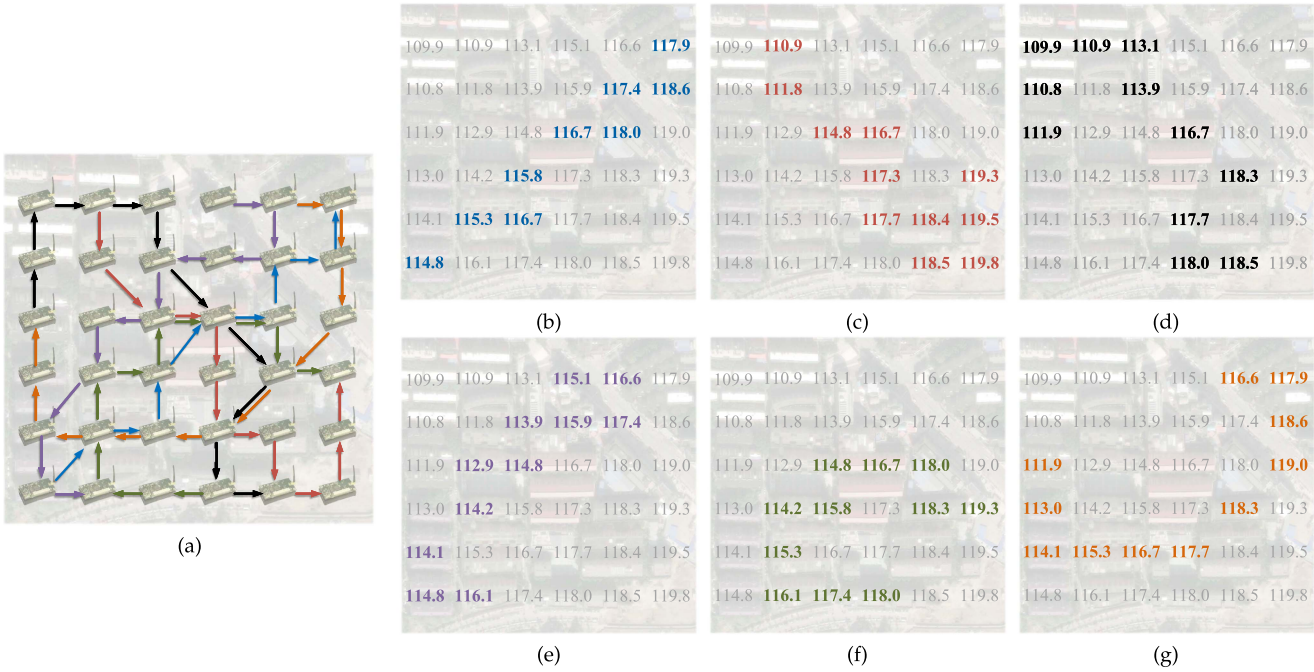


Fig. 2. Random sampling process. (a) Moving trajectories of six participants shown in blue, red, black, purple, green, and orange, respectively. (b)–(g) Sampling results of six participants shown in the corresponding colors.

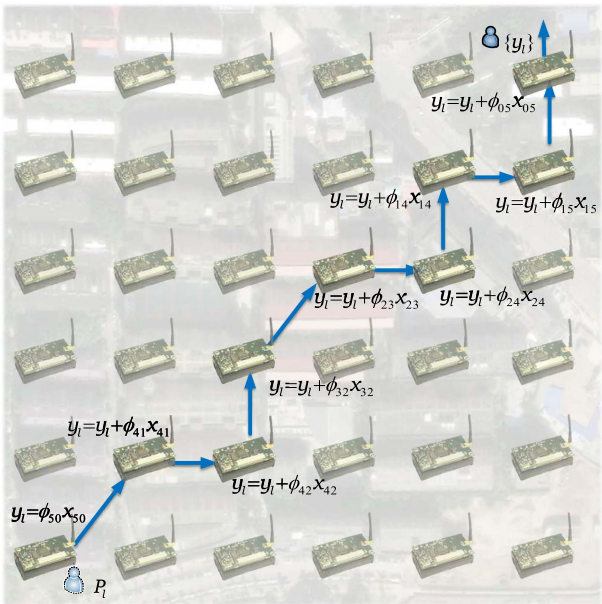


Fig. 3. Participant P_l runs the encoding algorithm along with a random sampling process, and stores the encoded result, measurement y_l.

the end of the process, it has

$$y_l = \phi_{50}x_{50} + \phi_{41}x_{41} + \phi_{42}x_{42} + \phi_{32}x_{32} + \phi_{23}x_{23} + \phi_{24}x_{24} + \phi_{14}x_{14} + \phi_{15}x_{15} + \phi_{05}x_{05}. \quad (1)$$

Denote that T_l is the trajectory matrix of P_l in the sampling process shown in Fig. 3, ϕ_l is a matrix consisting of random numbers, $\Phi_l = \phi_l \circ T_l$, $M_l(i,j)$ is an element at i-th row and j-th column of matrix M_l, and "$\circ$" means Hadamard Product. Then, measurement y_l shown in (1) is calculated as (2)–(4).

$$\tilde{X}_l = \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{T_l} \circ \underbrace{\begin{pmatrix} x_{00} & x_{01} & \cdots & x_{05} \\ x_{10} & x_{11} & \cdots & x_{15} \\ \vdots & \vdots & \ddots & \vdots \\ x_{50} & x_{51} & \cdots & x_{55} \end{pmatrix}}_{X}, \quad (2)$$

$$M_l = \underbrace{\begin{pmatrix} \phi_{00} & \phi_{01} & \cdots & \phi_{05} \\ \phi_{10} & \phi_{11} & \cdots & \phi_{15} \\ \vdots & \vdots & \ddots & \\ \phi_{50} & \phi_{51} & \cdots & \phi_{55} \end{pmatrix}}_{\phi_l} \circ \tilde{X}_l$$

$$= \phi_l \circ (T_l \circ X)$$

$$= (\phi_l \circ T_l) \circ X$$

$$= \Phi_l \circ X, \quad (3)$$

$$y_l = \sum_i \sum_j M_l(i,j). \quad (4)$$

The derivation from (2) to (4) indicates that, our encoding algorithm has considered the data correlation. By constructing matrix Φ_l, measurement y_l can acquire the information

Algorithm 2: Encoding Algorithm With Block Partition.

Input: $\mathbb{V}_l$;
Output: $\{y_l^k\}$;
1: Initialize $y_l^k = 0, 0 \le k < \varpi$;
2: **while** $v_{ij} \in \mathbb{V}_l$ **do**
3: **for** $k = 0$ to $\varpi - 1$ **do**
4: **if** $v_{ij} \in \mathbb{V}^k$ **then**
5: $y_l^k \leftarrow y_l^k + \phi_{ij} x_{ij}$;
6: **end if**
7: **end for**
8: **end while**

of the entire data field, X, with $\Phi_l \circ X$. That is, what participant P_l encodes is X, not just the sampled subset of the data, $\{x_{50}, x_{41}, x_{42}, x_{32}, x_{23}, x_{24}, x_{14}, x_{15}, x_{05}\}$. We ascribe this conclusion to the randomness of Φ_l, where $\Phi_l = \phi_l \circ T_l$ and trajectory matrix T_l abstracts the random movement of P_l. $\Phi_l \circ X$ is like a random sampling for X with Φ_l, data correlation of X can then be exploited, and y_l features the information of the entire data field, X. When enough of the measurements are generated by different Φ_l related to various participants, the complete information of X could be represented jointly by these measurements.

2) Block Partition Strategy: This subsection proposes a block partition strategy, aiming at increasing the number of measurements without requiring more participants. Block partition is usually used in image processing for reducing the computational complexity [7], [30]. Here we apply the idea of blocking to MCS scenarios.

Suppose that the target sensing area is partitioned into ϖ blocks, $\mathbb{V}^k$ denotes the set of virtual nodes in k-th block, and y_l^k is the k-th measurement of P_l. An enhanced version of the encoding algorithm with additional block partition strategy is shown in Algorithm 2.

Fig. 4 illustrates the scenario of block partition, where two measurements, $\{y_l^1, y_l^2\}$, are generated by the same participant, P_l. According to Algorithm 2, one has

$$y_l^1 = \phi_{23} x_{23} + \phi_{24} x_{24} + \phi_{14} x_{14} + \phi_{15} x_{15} + \phi_{05} x_{05}, \quad (5)$$

and

$$y_l^2 = \phi_{50} x_{50} + \phi_{41} x_{41} + \phi_{42} x_{42} + \phi_{32} x_{32}. \quad (6)$$

Taking measurement y_l^1 as an example, it can be further derived from

$$y_l^1 = \sum_{i,j} M_l^1(i,j), \quad (7)$$

where

$$M_l^1 = \left(\phi_l \circ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right) \circ X$$

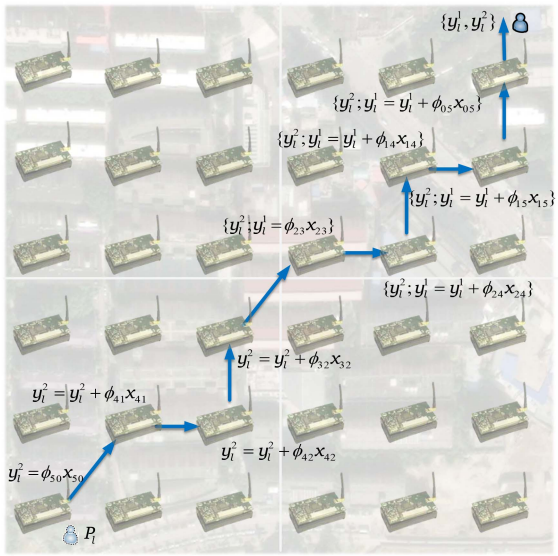


Fig. 4. Encoding with block partition, where the trajectory of P_l is seen as two sampling processes, and P_l stores two measurements, $\{y_l^1, y_l^2\}$.

$$= \Phi_l^1 \circ X. \quad (8)$$

By investigating a measurement matrix Φ_l^1 to exploit the data correlation, measurement y_l^1 features the entire data field, X, not its subset, $\{x_{23}, x_{24}, x_{14}, x_{15}, x_{05}\}$. The same analysis is true for measurement y_l^2.

Block partition increases the number of the measurements by reducing the size of the data subset used to compute the measurements. Randomness is still the hypothesis ensuring the validity of the encoding operation. When a participant conduct a random movement in the target sensing area, its local trajectories in some blocks are random as well. The movement in a local block can also be seen as a random sampling for the data field corresponding to the whole area.

3) Data Exchange: Data exchange is used to further increase the number of the measurements for a certain participant. Consider a common scenario: When two participants meet at some observation site, they have the opportunity to establish transient communication by using short range communication interfaces such as Bluetooth. The participants can then exchange their data during this opportunistic contact.

Suppose that $\mathbb{D}_l$ and $\mathbb{D}_a$ are data sets acquired by P_l and P_a on their moving paths, respectively. x_{ij} is the r-th element in $\mathbb{D}_l$, and v_{ij} is the corresponding virtual node. The encoding algorithm by employing block partition and data exchange is described in Algorithm 3.

We use Fig. 5 to illustrate the encoding algorithm with data exchange. When P_l meets P_a at virtual node v_{23}, it acquires the data exchanged from P_a, $\{x_{01}, x_{11}, x_{22}\}$. These data are temporarily stored until P_l completes its sampling process. P_l then performs encoding operation and generates three measurements, $\{y_l^0, y_l^1, y_l^2\}$, by blocks. Notice that P_l does not visit any nodes in 0-th block, but it can achieve the measurements related to 0-th block, thanks to the data exchange strategy. From the P_a's point

Algorithm 3: Encoding Algorithm With Block Partition and Data Exchange.

Input: $\mathbb{V}_l$;
Output: $\{y_l^k\}$;
Initialize $\mathbb{D}_l = \emptyset$, $y_l^k = 0, 0 \le k < \varpi$;
while $v_{ij} \in \mathbb{V}_l$ **do**
  $\mathbb{D}_l \leftarrow \mathbb{D}_l \cup x_{ij}$;
  if P_l meets P_a at v_{ij} **then**
    $\mathbb{D}_l \leftarrow \mathbb{D}_l \cup \mathbb{D}_a$;
  end if
end while
for $r = 0$ to $|\mathbb{D}_l| - 1$ **do**
  for $k = 0$ to ϖ-1 **do**
    if $v_{ij} \in \mathbb{V}^k$ **then**
      $y_l^k \leftarrow y_l^k + \phi_{ij} x_{ij}$;
    end if
  end for
end for

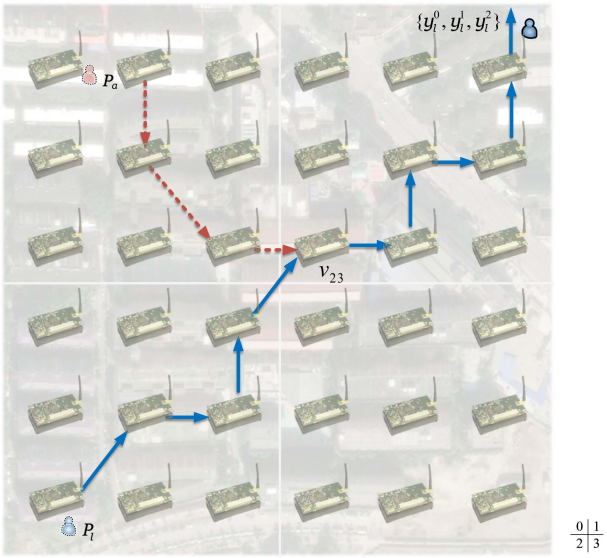


Fig. 5. Encoding with block partition and data exchange. P_l stores three measurements, $\{y_l^0, y_l^1, y_l^2\}$.

of view, it can also obtain the data from P_l with data exchange when they meet at v_{23}.

For a participant, by employing data exchange, the number of measurements can be further increased. The participants exchange their data via short range wireless communication when they meet at some common observation sites, with almost no added cellular network overhead.

IV. Data Recovery From Encoded Measurements

This section investigates a data recovery algorithm, targeting at recovering the original data field from a set of M measurements. When decentralized storage is implemented by the participants independently, data recovery algorithm runs on a cloud server.

A. Under-Determined System of Measurement Equations

1) Measurement Equations: This subsection formulates the encoding operations into a system of measurement equations. Suppose that matrix X represents the original data field consisting of N data. According to Algorithm 1, one measurement is generated by performing an encoding operation on X with matrix Φ_l corresponding to participant P_l. Let $vec(\cdot)$ denote a function that reshapes a matrix into a row vector. $\boldsymbol{x} = (vec(X))^\intercal$, $\Phi(l,:) = vec(\Phi_l)$, where $\Phi(l,:)$ is the l-th row of matrix Φ. Encoding equation shown in (4) of Section III-B1 can be rewritten into a more general form,

$$y_l = \Phi(l,:)\boldsymbol{x}. \tag{9}$$

M measurements, $\{y_0, y_1, \ldots, y_{M-1}\}$, consist of a system of M measurement equations,

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{M-1} \end{pmatrix} = \Phi\boldsymbol{x}. \tag{10}$$

Here we term Φ as measurement matrix of this system.

In Algorithms 2 and 3, block partition is employed, and the measurements are generated by block. Suppose that the target sensing area is partitioned into ϖ blocks, the original data field corresponding to k-th block is expressed as $\boldsymbol{x}_k$, $\cup\{\boldsymbol{x}_k\}_{0\le k<\varpi} = \boldsymbol{x}$, $|\boldsymbol{x}_k| = n$, and $\varpi n = N$. By considering block partition, (10) is further rewritten as

$$\begin{pmatrix} \boldsymbol{Y}_0 \\ \boldsymbol{Y}_1 \\ \vdots \\ \boldsymbol{Y}_{\varpi-1} \end{pmatrix} = \Phi \begin{pmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_{\varpi-1} \end{pmatrix}, \tag{11}$$

where $\sum_{k=0}^{\varpi-1} |\boldsymbol{Y}_k| = M$, measurement matrix $\Phi =$

$$\begin{pmatrix} \Phi_0 & & & \\ & \Phi_1 & & 0 \\ & & \ddots & \\ 0 & & & \Phi_{\varpi-1} \end{pmatrix}. \tag{12}$$

As can be seen, Φ is a measurement matrix with block diagonal structure caused by block partition strategy, where Φ_k corresponds to the k-th block of the target sensing area. This indicates that (11) consists of ϖ set of equations in the form of (10).

2) Attention Mechanism: Considering a common MCS scenario, where the organizer may pay more attention to some special regions of the target sensing area, we present an attention mechanism to improve the efficiency of measurement equations.

With block partition, measurements are encoded by block. This gives ones an opportunity, allowing encoding resources to be deliberately assigned to the designated blocks, or allocate more measurements to the blocks of being interested. We term it as attention mechanism. Fig. 6 gives an instance to illustrate the attention mechanism. The organizer may think the fourth block is of more importance, and hope the recovered data field has

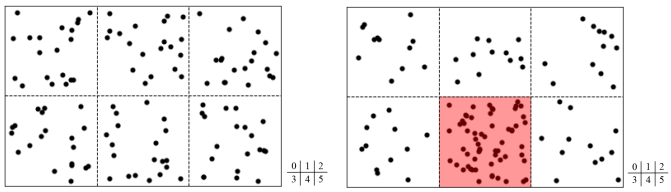


Fig. 6. Two methods of measurement allocation, where a dot represents a measurement. (a) Uniform allocation. (b) Allocation with more attention on the fourth block.

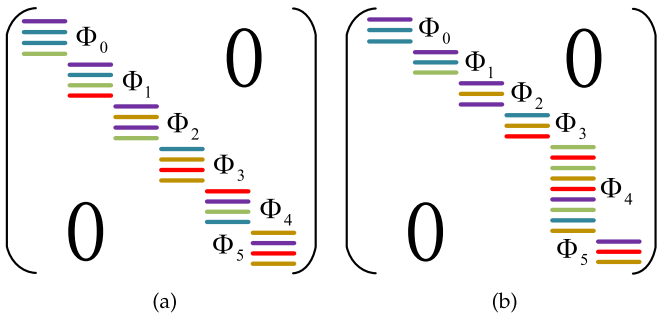


Fig. 7. Comparison of two block diagonal matrices. (a) All sub-matrices have the same number of rows. (b) Sub-matrix Φ_4, corresponding to the fourth block of being interested, is with more rows.

higher accuracy on this block. Instead of uniform allocation, where each block is allocated the same measurements, it is reasonable to allocate more measurements to the fourth block, while the total number of measurements remains unchanged.

To put the attention mechanism into effect, we adjust the size of the sub-matrices of Φ shown in (12). Simply designed measurement matrix Φ corresponds to uniform allocation of measurements. An example is shown in Fig. 7(a), where each sub-matrix has the same 4 rows. With attention mechanism, the size of Φ_4 increases from 4 rows to 9 rows, as illustrated in Fig. 7(b), since the fourth block may be an interested hot-spot area. Here the total number of rows of Φ keeps unchanged.

It should be noted that (10) or (11) may represent an under-determined system of equations. There are M linear equations but the number of variables is N. Clearly, if M is equal to or greater than N, then this set of equations will be easy to solve. However, general MCS campaigns often recruit the participants with minimum number to save the recruitment cost. That is, normally, M is far smaller than N in MCS scenarios. In this way, (11) may have countless solutions, which can be problematic.

B. Decoding Algorithm Based on Compressed Sensing Theory

Fortunately, classical Compressed Sensing theory can be used to solve the under-determined equations [4], [6], [29]. Compressed Sensing is originally introduced to represent a signal with fewer information than the Shannon-Nyquist limit. In Compressed Sensing, signal recovery with only a small amount of information is also formulated as an under-determined problem. We can leverage this method from Compressed Sensing in our proposed scheme to reconstruct the original data field.

Let $\boldsymbol{y} = (y_0\ y_1\ \ldots\ y_{M-1})^\intercal$, ψ represent a sparse basis, and $\boldsymbol{\alpha}$ be the result of $\boldsymbol{x}$ in the transform domain of ψ. The under-determined system of measurement equations, $\boldsymbol{y} = \Phi\boldsymbol{x}$, can be derived as $\boldsymbol{y} = \Psi\boldsymbol{\alpha}$, where Ψ is called as sensing matrix, $\Psi = \Phi\psi$, and $\boldsymbol{x} = \psi\boldsymbol{\alpha}$. According to Compressed Sensing theory, although $M \ll N$, if Φ has the mutual coherence with ψ, one can compute $\boldsymbol{\alpha}$ by solving the following optimization problem

$$\min_{\boldsymbol{\alpha}} \| \boldsymbol{\alpha} \|_1 \quad \text{subject to} \quad \| \boldsymbol{y} - \Psi\boldsymbol{\alpha} \|_2^2 \leq \lambda, \tag{13}$$

where λ is a small constant, $\| \cdot \|_1$ denotes the 1-norm, and $\| \cdot \|_2$ is the 2-norm. The author in [6] quantifies the error bound of (13) by pointing out that, M measurements with $M = O(S\log(N))$ are just as good as knowing the S biggest coefficients of $\boldsymbol{\alpha}$.

Several fast algorithms have been introduced to solve this optimization problem. Iteration-based DAMP presented in [15] is one of the best performing algorithms. Block partition-based measurement equations constructed in MCS scenarios can be considered as a set of ϖ equations, each of which is capable of being formulated as an optimization equation shown in (13). Intuitively, a Compressed Sensing algorithm, such as DAMP, can be directly used to reconstruct each block, and the entire data field is then recovered by simply concatenating all blocks together. However, the data subsets related to each block of the target area are correlated to each other. Using the same Compressed Sensing algorithm will not exploit this correlation and inevitably will introduce blocking artifacts.

We propose an extended version of the-state-of-art DAMP, termed as B-DAMP, specially targeting at data recovery of MCS with block partition. The proposed algorithm is divided into three stages, namely approximating, estimation of the residual, and denoising process. Let $\mathcal{D}_{\hat{\sigma}^t}(\cdot)$ be a common denoising function, and $\hat{\sigma}^t = \| \boldsymbol{z}_i^t \|_2 / \sqrt{m_k}$ represent an estimate of the standard deviation of that noise. div denotes the operation of partial derivative, $\mathrm{div}\mathcal{D}_{\hat{\sigma}^{t-1}}$ is the divergence of denoiser $\hat{\sigma}$, Φ_k^* is the inverse matrix of Φ_k, and m_k is the number of measurements related to k-th block. The three stages are described below.

- *Approximating:* For the approximation corresponding to k-th recovered block at t-th iteration, its $(t+1)$-th approximation, $\hat{\boldsymbol{x}}_k^{t+1}$, is calculated as

$$\hat{\boldsymbol{x}}_k^{t+1} = \hat{\boldsymbol{x}}_k^t + \Phi_k^* \boldsymbol{z}_k^t. \tag{14}$$

- *Estimation of the residual:* For $\hat{\boldsymbol{x}}_k^t$, one can further estimate its residual $\boldsymbol{z}_k^t$,

$$\boldsymbol{z}_k^t = \boldsymbol{y}_k - \Phi_k \hat{\boldsymbol{x}}_k^t + \boldsymbol{z}_k^{t-1} \mathrm{div}\mathcal{D}_{\hat{\sigma}^{t-1}} \left(\hat{\boldsymbol{x}}^{t-1} + \Phi_k^* \boldsymbol{z}_k^{t-1} \right) / m_k. \tag{15}$$

- *Denoising:* Concatenating ϖ blocks, $\{\hat{\boldsymbol{x}}_k^t\}$, into an entire data set, $\hat{\boldsymbol{x}}^{t+1}$, one then performs denoising operation,

$$\hat{\boldsymbol{x}}^t = \mathcal{D}_{\hat{\sigma}^t} \left(\hat{\boldsymbol{x}}^t \right). \tag{16}$$

Based on these three stages, the proposed block-based B-DAMP algorithm is illustrated in Algorithm 4. The key idea here is that, at each iteration, the denoising operation, $\mathcal{D}_{\hat{\sigma}^t}(\cdot)$, is performed on the entire data set, $\hat{\boldsymbol{x}}$, not separate blocks, $\hat{\boldsymbol{x}}_k$. In this way, our B-DAMP enjoys both the benefit of ameliorating blocking artifacts and the advantage of high-performance data

Algorithm 4: B-DAMP Decoding Algorithm.

Input: $\boldsymbol{y}_k$, Φ, ϖ, $Iter$, $Threshold$;
Output: $\hat{\boldsymbol{x}}^t$;
1: Initialize $\hat{\boldsymbol{x}}^0 = 0$, $t = 0$, $\boldsymbol{z}_k^0 = \boldsymbol{y}_k$;
2: **while** $t < Iter$ **do**
3: Partition $\hat{\boldsymbol{x}}^t$ into ϖ blocks;
4: **for** $k = 0$ to ϖ-1 **do**
5: Calculate $\hat{\boldsymbol{x}}_k^{t+1}$ according to (14);
6: **end for**
7: $t \leftarrow t + 1$;
8: **for** $k = 0$ to ϖ-1 **do**
9: Calculate $\boldsymbol{z}_k^t$ according to (15);
10: **end for**
11: Concatenate ϖ blocks into the entire data set, $\hat{\boldsymbol{x}}^t$;
12: Update $\hat{\boldsymbol{x}}^t$ according to (16);
13: $error \leftarrow \frac{\|\hat{\boldsymbol{x}}^t - \hat{\boldsymbol{x}}^{t-1}\|_2}{\|\hat{\boldsymbol{x}}^t\|_2}$;
14: **if** $error < Threshold$ **then**
15: break;
16: **end if**
17: **end while**

recovery in the original DAMP. In Algorithm 4, the number of iterations $Iter$ and the desired accuracy $Threshold$ are predefined.

So far we have introduced four algorithms. Algorithms 1, 2, and 3 are encoding algorithms while Algorithm 4 is a decoding algorithm. The participants perform encoding operation independently by employing an encoding algorithm and store the encoded measurements. Decoding algorithm runs on a cloud server for reconstructing the original data by receiving enough number of measurements from the corresponding participants.

V. Performance Guarantee of Compressed Data Storage

In this section, we mathematically demonstrate that our compressed storage scheme can guarantee successful data recovery with enough number of measurements. We also present in-depth discussion on the decentralization performance.

A. Mathematical Foundation

Mathematically, measurement matrix Φ, shown in (12), is the key to the performance of our participant-based data storage. The proposed scheme includes decentralized data storage and cloud-based data recovery, both of which are designed on the basis of Φ. By constructing block diagonal matrix Φ of size $M \times N$, the data field with N-point data are encoded into M distinctive measurements, stored by different participants. Based on matrix Φ, data recovery in MCS system is formulated as a Compressed Sensing optimization problem, and a fast decoding algorithm can then be employed to find the solution.

To make the Compressed Sensing optimization problem have an optimal solution, sensing matrix Ψ has to obey the so-called restricted isometry property. Let ψ be a sparse basis. For natural signal $\boldsymbol{x}$, one has $\boldsymbol{x} = \psi\boldsymbol{\alpha}$ and $\Psi = \Phi\psi$. It had been proven

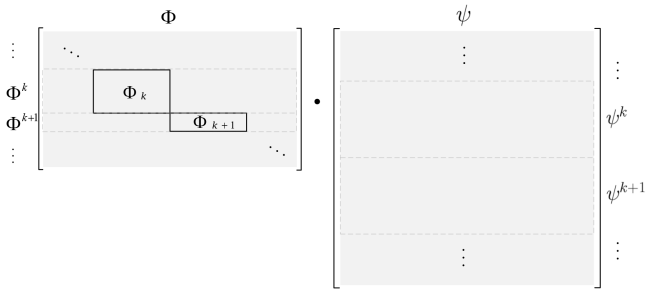


Fig. 8. For matrix Φ of size $M \times N$, its row matrix Φ^k and diagonal sub-matrix Φ_k are with $m_k \times N$ and $m_k \times n$, respectively. The size of ψ is $N \times N$, and its row matrix ψ^k is with $n \times N$.

in [4] that, if Φ is a randomly generated matrix, then it has mutual coherence with ψ, thus Ψ obeys the above-mentioned restricted isometry property with a high probability. Let T^k represent m_k trajectories of the participants in k-th block of the target sensing area, ϕ^k is a matrix consisting of $m_k \times n$ random numbers. Then measurement matrix Φ_k related to k-th block is formulated as $\Phi_k = \phi_k \circ T^k$. Note that the movements of the participants in the target sensing area tend to be random in nature. The distribution of 0 and 1 in trajectory matrix T^k is then random accordingly. Thus Φ_k can be seen as a randomly generated matrix, and is coherent with any given sparse basis.

In terms of the mutual coherence, we next prove that block diagonal matrix Φ including ϖ sub-matrices, where Φ_k is the k-th sub-matrix of size $m_k \times n$, also satisfies the condition of mutual coherence for restricted isometry property. Suppose that the size of Φ is $M \times N$, $\sum_{k=0}^{\varpi-1} m_k = M$, $\varpi n = N$, ψ represents the sparse basis of size $N \times N$, and φ is with $n \times n$.

Theorem 5.1. Let c be a pre-defined non-negative constant satisfying $c \leq 1$. If the mutual coherence between Φ_k and φ is bounded by a positive constant γ, that is, $\mu(\Phi_k, \varphi) < \gamma$, then block diagonal matrix Φ also has mutual coherence with ψ as

$$\mu(\Phi, \psi) < c\gamma. \tag{17}$$

Proof. Denote $\Phi_k(i)$ be the i-th row of Φ_k, and $\varphi(j)$ be the j-th column of sparsity basis φ. One has

$$\mu(\Phi_k, \varphi) = \max_{0 \leq i < m_k, 0 \leq j < n} \frac{|\langle \Phi_k(i), \varphi(j) \rangle|}{\|\Phi_k(i)\|_2 \|\varphi(j)\|_2} < \gamma. \tag{18}$$

Dividing matrix Φ into ϖ blocks, one can then calculate the mutual coherence between Φ and ψ as

$$\mu(\Phi, \psi) = \max_{0 \leq k < \varpi, 0 \leq i < m_k, 0 \leq j < N} \frac{|\langle \Phi^k(i), \psi(j) \rangle|}{\|\Phi^k(i)\|_2 \|\psi(j)\|_2}, \tag{19}$$

where Φ^k denotes the k-th block of matrix Φ, which is drawn with dashed lines in Fig. 8.

Considering that Φ is a ϖ-block diagonal matrix, all elements of Φ^k, except these corresponding to Φ_k, are zeroes. In this way, one has $|\langle \Phi^k(i), \psi(j) \rangle| = |\langle \Phi_k(i), \psi^k(j) \rangle|$, where ψ^k denotes the k-th block of matrix ψ shown in Fig. 8. According to (18), one has $|\langle \Phi_k(i), \psi^k(j) \rangle| < \gamma \cdot \|\Phi_k(i)\|_2 \|\psi^k(j)\|_2$. As a result, one can achieve

$$|\langle \Phi^k(i), \psi(j) \rangle| < \gamma \cdot \|\Phi_k(i)\|_2 \|\psi^k(j)\|_2. \tag{20}$$

Since $\|\Phi_k(i)\|_2 = \|\Phi^k(i)\|_2$, by applying (20) to (19), one further has

$$\mu(\Phi, \psi) < \max_{0 \le k < \varpi, 0 \le j < N} \frac{\|\psi^k(j)\|_2}{\|\psi(j)\|_2} \cdot \gamma. \tag{21}$$

Let $c = \max_{0 \le k < \varpi, 0 \le j < N} \frac{\|\psi^k(j)\|_2}{\|\psi_j\|_2}$, then

$$\mu(\Phi, \psi) < c\gamma. \tag{22}$$

From Theorem V.1, matrix Φ constructed with our storage scheme satisfies mathematical mutual coherence with any given sparse transform basis. Given enough measurements, i.e., $M = O(S\log(N))$ illustrated in Compressed Sensing theory, the under-determined system of equations formulated from the proposed participant-based storage has an optimal solution with an error bound. In other words, the original data field can be recovered with desired accuracy by recruiting a certain number of participants. We are going to further validate this claim in the experimental study.

B. In-Depth Discussion

Our decentralized storage scheme is in particular suitable for MCS with energy consumption in mind by introducing a two-part structure, i.e., compressed storage and data recovery. Specifically, the two-part structure takes an asymmetric design. Compressed storage is extremely simple, where the encoding is just a linear operation, and thus the participants do not impose extra energy consumption. However, data recovery has a high computational complexity, since original data have to be reconstructed from a number of simply encoded measurements. Fortunately, data recovery is implemented at a cloud server, generally with enough energy and computational resources.

The proposed decentralized storage scheme is independent of any specific participants. It is a practical possibility that some participants may leave MCS systems at any time for various reasons. In this event, the measurements stored in these unreliable participants may be discarded as they withdraw from the systems. We argue that, this has no effect on the performance of our scheme. The reason is, every measurement, no matter where it is stored, features the same data field. In this way, all participants are equivalent to one another. As long as enough participants are still recruited in MCS system, or to be more exact, the number of measurements M satisfies $M = O(S\log(N))$, the original data field can be recovered with desired accuracy.

Randomness is the mathematical prerequisite for our decentralized and compressed storage, and block partition strategy improves its performance. Thanks to the randomness of the movements of the participants, we design a measurement matrix that has mutual coherence with a sparse transform basis. The data correlation can then be exploited, and the data field is encoded into a measurement. Based on block partition, data exchange and attention mechanism are further investigated to reduce the number of participants and improve decoding accuracy. At the same time, block partition also decreases the computational complexity of data recovery, as it is usually employed in the field of image processing.

One of the most important advantages of proposed scheme is to provide a great alternative to the popular cloud storage by utilizing little pieces of storage space scattered over numerous participants. The disadvantage may be the query latency. With the proposed decentralized scheme, data are not stored at cloud server, but are stored across numerous individual participants. In a sense, these participants form a large-scale distributed database, which could cause a relatively longer query latency in some cases. The proposed scheme is designed for uni-modal data, however, as sensor technology advances, one mobile device is capable of sensing and collecting multi-modal data, i.e., several kinds of data for collaboratively providing complete information of the target sensing data. Although multi-modal data can be considered as multiple uni-modal data, the data with multi-modality may have strong correlation that needs to be considered. Exploiting multi-modal correlation to improve decentralized data storage can be a challenging task.

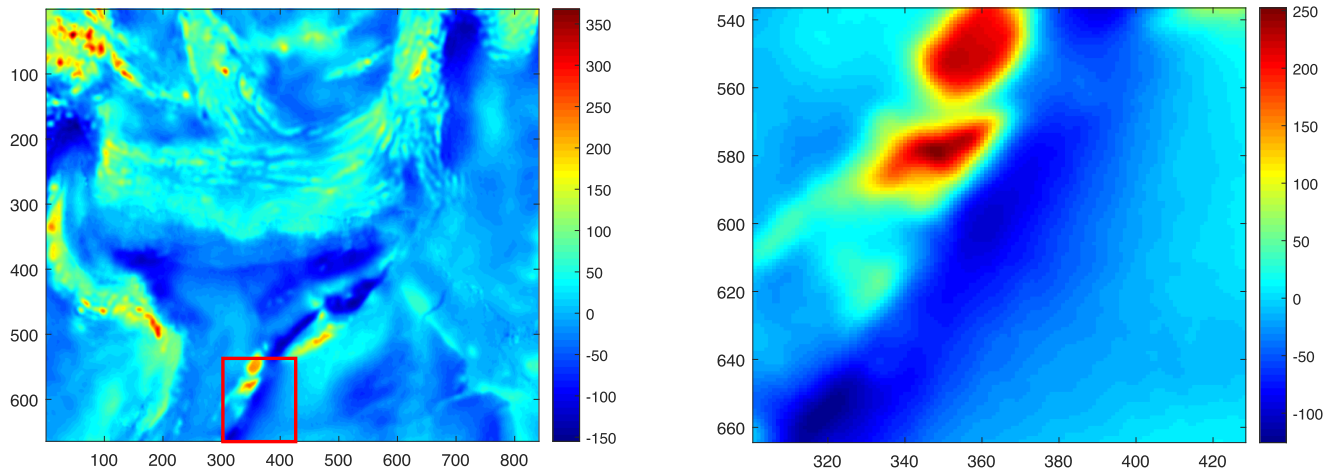
VI. Experimental Results

This section first introduces the experimental scenario and defines performance indexes. Extensive experiments are then designed to evaluate the proposed scheme by using real gravity data. The data set and our source codes are available at https://github.com/siwangzhou/pdds.

A. Experimental Scenario

The target sensing area consists of 40000 observation sites, which follow a 200×200 grid distribution. Virtual sensor nodes are imagined as being deployed at the corresponding observation sites, and a virtual node is related to a gravity datum. The participants with their sensing devices move through these virtual nodes based on the popular Metropolis-Hastings random walk algorithm presented in [3]. The minimum number of steps of the participants is set to 200, and the maximum number is 1200. A participant randomly selects a site on the area to start its moving process, and at the same time it enters this area at a randomly chosen time point. Along with the process of random walk, the participant collects the data from virtual sensor nodes. Each participant independently runs encoding algorithm and stores its own measurements. For block-based sampling strategy, the crowdsensing system recruits 500 participants, while for sampling scenarios without block partition, the number is increased to 5000. After the participants finish their random walks, only the trajectories at the central grid area of size 128×128 are retained for avoiding the impact of the boundary of the area.

Gravity anomaly data on the earth presented in [17] are used to evaluate the proposed storage scheme. A public data set of gravity anomaly is shown in Fig. 9, where Fig. 9(b) is a part of Fig. 9(a). In the experiment, the data shown in Fig. 9(b) are related to the virtual sensor nodes deployed in central target area of size 128×128. The values of gravity data range from -125.6 to 252.4. To facilitate the encoding and decoding operations, the range of data values is changed from 0 to 378 in the experiments, without influencing the performance evaluation.



(a) Gravity anomaly of size 841×664 (milligal)

(b) Experimental data of size 128×128 (milligal)

Fig. 9. Visual map of experimental data set.

Recruitment overhead, coverage ratio, decoding error and decoding time of the decoding algorithm are used as performance indexes. Recruitment overhead is used to indicate the number of participants participating in decoding process in the crowdsensing system. Coverage ratio is the proportion of virtual nodes visited by the participants in the target sensing area. Assume that $\boldsymbol{x}$ represents the original data field and $\hat{\boldsymbol{x}}$ corresponds to the recovered one. The size of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are all N, i.e., the number of the virtual nodes deployed in the target area is N. Let p_l denote the number of measurements stored by participant P_l, and $\{s_l\}$ be the set of virtual nodes participant P_l visits. Then, coverage ratio, cr, is defined as

$$cr = \frac{|\cup\{s_l\}|}{N}. \tag{23}$$

Here $\cup\{s_l\}$ denotes the union set of $\{s_l\}$ for all participants, and $|\cdot|$ is the number of the elements in this union. Decoding error, $error$, is defined as a relative square error,

$$error(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2}{\|\boldsymbol{x}\|_2}. \tag{24}$$

Let mae be mean absolute error, where $mae(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_1}{N}$. Set the threshold of mae to 1. If mae is less than 1, then the original data field is said to be successfully recovered. Suppose that one runs data reconstruction algorithm $totalnumber$ times, where the number of successful data recovery is $number$. The proportion of successful data recovery, sdr, is then defined as

$$sdr = \frac{number}{totalnumber}. \tag{25}$$

Decoding rate is defined as

$$dr = \frac{M}{N}, \tag{26}$$

where $M = \sum_l p_l$, i.e., M is the total number of measurements stored by the participants involving in the decoding process.

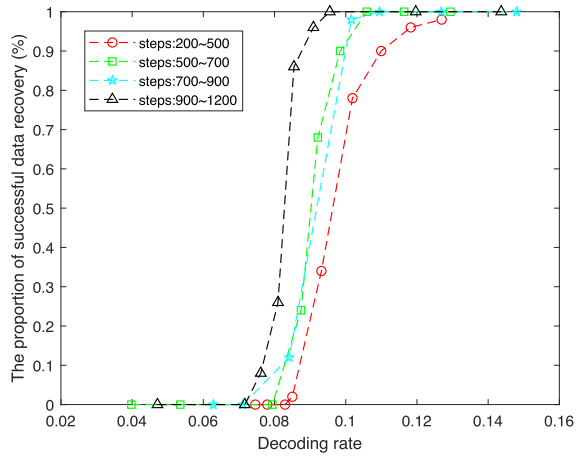
The experiments are implemented using the MATLAB R2015b simulator on a server platform configured with 256 GB of memory and two 3.2 GHz Intel(R) CPUs.

B. Performance Analysis

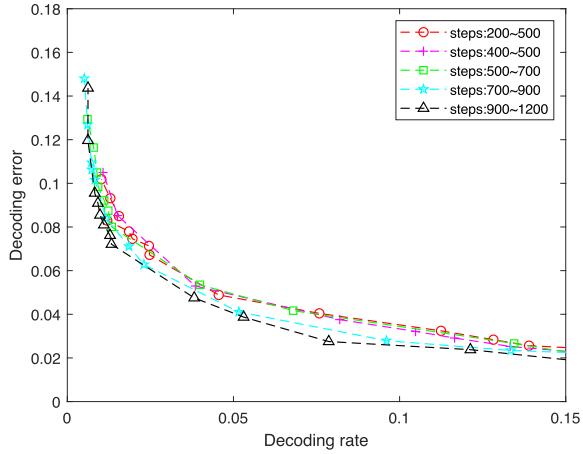
This section analyses the performance of the proposed scheme under several MCS scenarios by observing the impact of several factors, including the steps of the participants, the probability of data exchange, and various encoding and decoding strategies.

1) Impact of Steps of the Participants: This subsection evaluates the impact on performance of the proposed scheme in terms of the participant's scope of activities in the target sensing area by setting various step numbers. Here, the probability of data exchange is set to 50%. That is, when two participants meet at an observation site, they exchange their data with a 50% probability. Various probabilities will also be tested in the next subsection.

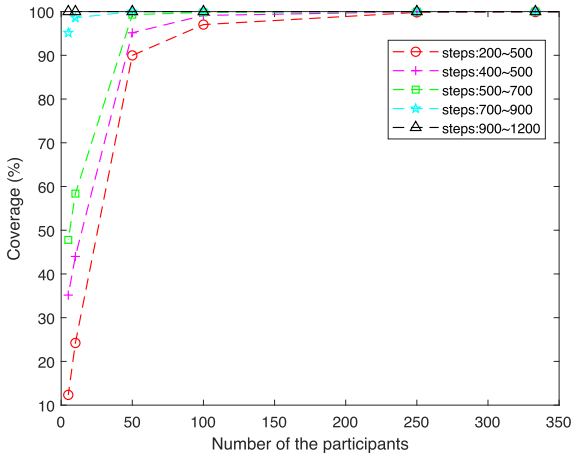
Theoretically, if the number of measurements, M, satisfies $M = O(S\log(N))$, the original data field can be recovered with desired accuracy. We further validate it in experimental environments by using decoding rate, $dr = M/N$, as an index. Fig. 10(a) and (b) evaluate the recovery accuracy by investigating the probability of successful data recovery and decoding error, respectively. Here $N = 128 \times 128 = 16384$. From Fig. 10(a), when dr exceeds 0.12, the proportion of successfully data recovery could be one hundred percent at various scenarios. The scenario, where the step ranges from 900 to 1200, has better performance than other three scenarios with less steps. This shows our decoding algorithm can achieve an accuracy improvement along with the increase of the number of steps. From Fig. 10(b), the number of steps has very small impact on decoding error. As the decoding rate increases, decoding error in all five scenarios continues to fall. This accords with the principle of decoding algorithm, as dr is proportional to the number of the measurements, M. The scenario where the steps ranging from 900 to 1200 has slightly lower decoding error than other four



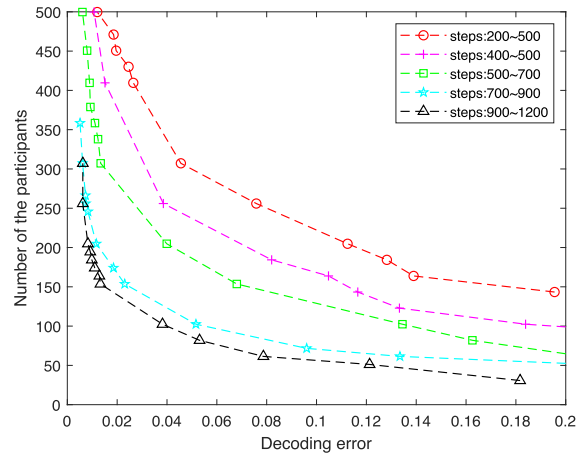
(a) The probability of successful data recovery



(b) Recovery accuracy



(c) Coverage rate



(d) Recruitment overhead

Fig. 10. Impact on performance in terms of various steps.

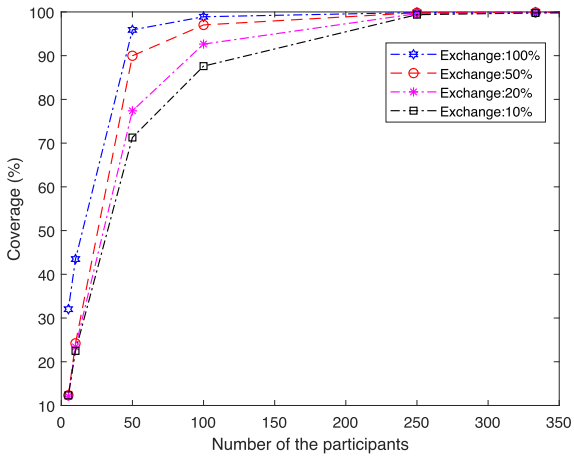
scenarios. This result agrees with that shown in Fig. 10(a), where more steps indicate a relatively higher probability of successful data recovery.

Fig. 10(c) shows that, along with the increase of the steps of the participants, the coverage rate increases significantly. When the crowdsensing system only recruits 20 participants, the coverage rate is about 25% with the steps ranging from 200 to 500. Once the range of steps changes from 900 to 1200, the coverage rate is up to 100%. This is because the participants have more opportunities to exchange their data as they go further in the target sensing area. Note that the scope of the participants' activities corresponds to the crowdsensing cost, so we have to see another side of the double-edged sword: the more steps a participant moves, the more cost the crowdsensing system has.

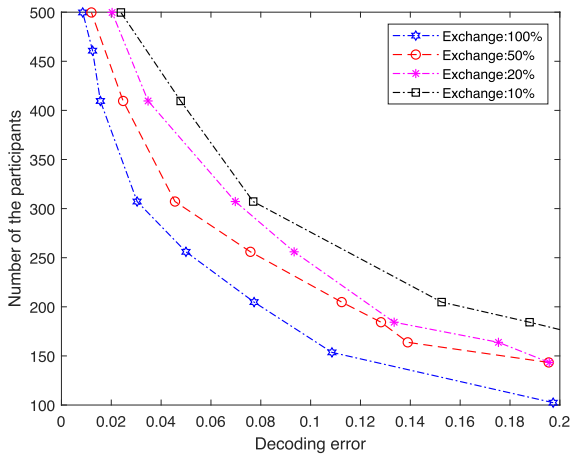
Fig. 10(d) illustrates the relationship between the recruitment overhead and the steps the participants move in the target area. Along with the increase of the decoding accuracy, the numbers of the participants at various steps all increase accordingly. This is a reasonable result, since more participants indicates more measurements being stored. When a participant goes further into the target area, it traverses across more blocks. With the proposed block partition strategy, it can generate and store more measurements. In this way, with the same measurements, or the same decoding error, the scenario where the steps range from 900–1200 requires far less participants than that with the steps ranging from 200–500.

2) Impact of Data Exchange: This subsection evaluates the performance of the proposed scheme on the scenarios with various probabilities of data exchange. In the experiments, the step number of the participants is set from 200 to 500.

Fig. 11(a) illustrates that, at the same number of the participants, as the exchanging probability increases from 10% to 100%, the coverage rate increases accordingly. When 50 participants are recruited in the crowdsensing system, the coverage rate is up to more than 95% with 100 percent probability of exchanging data compared to about 70% with 10 percent probability. This experimental result is reasonable. When two participants, at possible opportunities, to exchange their data with each other, the same data is stored across these two participants. This is as if one participant visits the observation sites on the walking path of the other one. The coverage rate can then be increased with the same number of the participants.

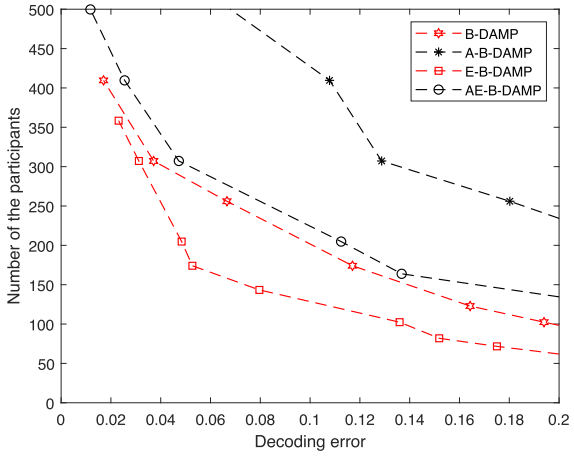


(a) Coverage rate

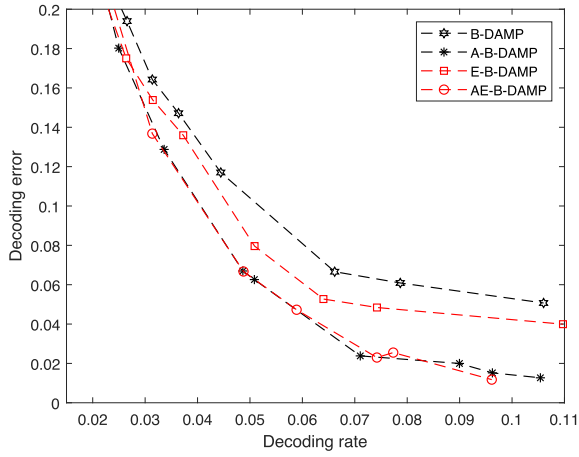


(b) Recruitment overhead

Fig. 11. Impact on performance in terms of exchanging probability.



(a) Recruitment overhead



(b) Decoding accuracy

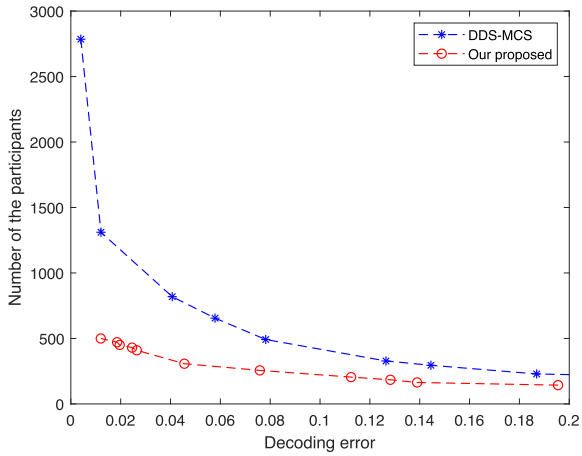
Fig. 12. Impact on performance in terms of attention mechanism and data exchange.

Fig. 11(b) gives the relationship between the number of the participants and decoding accuracy in the scenarios with four data exchanging probabilities. When two participants meet at a certain observation site, data exchange is performed with certain probability. Observing Fig. 11(b), the crowdsensing scenario with 100 percent probability of exchanging data has the best performance, while the worst performance is for the scenario of 10 percent probability of exchanging data in terms of the number of the participants. That is, to achieve the same decoding accuracy, significantly fewer participants are required for the crowdsensing system if higher probability of exchanging data can be employed.

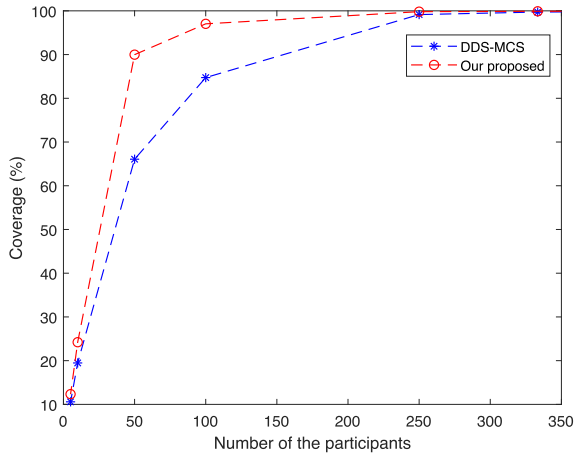
3) Ablation Study: This subsection performs an ablation study to evaluate the impact of our attention mechanism and data exchange strategy. A-B-DAMP denotes the proposed storage by employing B-DAMP algorithm with attention mechanism, E-B-DAMP denotes that with data exchange strategy, and AE-B-DAMP denotes that with both attention mechanism and data exchange. In the experiments, we set the probability of data exchange to 50%, and the range of steps is from 200 to 500.

Fig. 12(a) shows that, by employing data exchange, E-B-DAMP and AE-B-DAMP have significantly better performance in terms of recruitment overhead than their counterparts, B-DAMP and A-B-DAMP, respectively. For instance, to achieve the accuracy of 0.05 of decoding error, only about 150 participants have to be recruited for E-B-DAMP method, while for B-DAMP the crowdsensing system requires more than 250 participants. This indicates that the improvement of recruitment overhead mainly benefits from block partition-based data exchange strategy. It can also be seen from Fig. 12(a) that, E-B-DAMP requires fewer participants than AE-B-DAMP, although both approaches employ data exchanges. The reason is that AE-B-DAMP is based on attention mechanism, which has to choose reasonable measurements from participants in order to achieve higher accuracy.

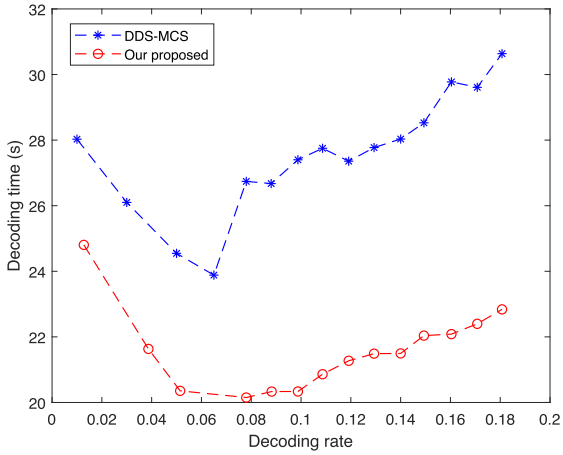
From Fig. 12(b), the methods employing attention mechanism, A-B-DAMP and AE-B-DAMP, have lower decoding error



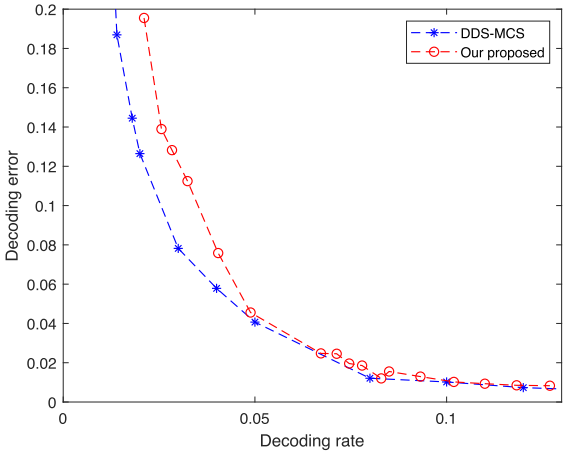
(a) Recruitment overhead (the less the better)



(b) Coverage rate (the higher the better)



(c) Decoding time (the less the better)



(d) Decoding error (the less the better)

Fig. 13. Performance comparison with the completing DDS-MCS method.

than their counterparts, B-DAMP and E-B-DAMP, respectively. Taking decoding rate of 0.05 as an example, the decoding error is about 0.04 for AE-B-DAMP, while for E-B-DAMP the decoding error is more than 0.08. This indicates that our attention mechanism can decrease decoding error, i.e., improving the accuracy of data recovery. The main reason for this improvement is that, with attention mechanism, we can choose the participants whose trajectories cover the blocks with significant data anomaly. This ensures the efficient use of sampling resources and thus achieves accuracy improvement.

C. Comparison With the Competing Method

To the best of our knowledge, nearly all the MCS systems employ cloud storage except reference [32], which introduces a distributed storage method and is most similar to our scheme. We call it DDS-MCS for simplicity and choose it as the competing method for comparing with our proposed AE-B-DAMP scheme.

The experimental result is shown in Fig. 13, where exchanging probability is set to 50% and step number ranges from 200 to 500. Our scheme has significantly less participants, much higher coverage rate, and much faster decoding speed, while still achieving almost the same decoding error when the decoding rate exceeds 0.05.

Fig. 13(a) shows that our scheme requires far smaller number of the participants, thus saving recruitment overhead for crowdsensing. The significant improvement is attributed to the proposed block-based encoding strategy. By employing block partition, each participant runs encoding algorithm and generates more measurements. In this way, at the same number of measurements, our scheme requires much fewer participants. As can be seen from Fig. 13(a), when the relative square error of data recovery, i.e., decoding error $error$, is about 0.05, the proposed scheme only requires 250 participants, while for DDS-MCS, the number of the participants is more than 650. Along with the decrease of the decoding error, our scheme even presents much lower recruitment overhead.

Fig. 13(b) illustrates that our scheme achieves far higher coverage rate than the competing one. High coverage rate benefits from the proposed data exchange strategy. When two participants meet in a certain observation site, they take the opportunity to exchange their data. It is as if one participant

is visiting the sites that the other participant has visited. For example, by recruiting the same 50 participants, our scheme covers more than 90% observation sites while the competing DDS-MCS covers only about 60%. In other words, our scheme can achieve the same coverage rate by recruiting fewer participants. With achieved higher coverage rate, the crowdsensing system has greater freedom when selecting the participants for data recovery.

Fig. 13(c) compares the decoding time of the proposed scheme and the competing DDS-MCS. The performance improvement is mainly due to block partition, which implies a divide-and-conquer strategy and is originally employed in image processing for reducing computation complexity. From Fig. 13(c), it is obvious that our scheme can recover the original data field much faster than DDS-MCS at various decoding rate. Both schemes take longer to recover the original data field as the decoding rate increases, since the decoding algorithms incur higher computation complexity along with the increase of decoding rate. However, our proposed decoding algorithm always runs faster than the competing one in any decoding rate. With our scheme, the entire data field is recovered by blocks, which indicates marked drop in computation overhead, hence the gain in speed.

Fig. 13(d) shows that, our scheme has almost the same decoding error as DDS-MCS when decoding rate exceeds 0.05, thanks to the block-based B-DAMP technique and attention mechanism. As we know, block partition, if unrestricted, may reduce the accuracy of decoding algorithm due to blocking artifacts. We observe from Fig. 9 in Section VI-A that, the distribution of gravity data is not uniform, and there exists gravity anomaly subarea. By utilizing block partition, more sampling resources are assigned to the subarea existing gravity anomaly data and less to these relatively uniform subarea. Therefore when the decoding rate exceeds a certain threshold, our scheme can complete a reasonable allocation of sampling resources, and thus eliminate the adverse impact of block partition.

VII. Conclusion

In contrast to the popular cloud-based storage, this article proposes a decentralized and compressed data storage scheme for MCS systems. Our scheme takes a two-part structure combined with an asymmetric design: data storage with simple encoding operation and data recovery with higher computational complexity. Data storage is accomplished independently by numerous participants without any support from cloud servers, while data recovery, when necessary, is implemented on a cloud with enough computational resources by using a compressed sensing based decoding algorithm. We further investigate block partition strategy as well as block-based data exchange and attention mechanism, improving the encoding and decoding efficiency. The mathematical foundation and extensive experimental results validate the performance of the proposed decentralized storage scheme.

With the decentralized storage scheme, the participants potentially store sensitive data of other participants, introducing security as a key problem. Although existing security strategies can be utilized to add additional security by considering the compressed data as a normal data, separating security from the storage process may be less efficient. We will in the future integrate security into the encoding algorithm and develop a crowdsensing platform, combing the secure storage strategy with the real scenario to validate its performance.

References

[1] S. B. Azmy, N. Zorba, and H. S. Hassanein, "Quality estimation for scarce scenarios within mobile crowdsensing systems," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10955–10968, Nov. 2020.

[2] J. Bormans, J. Gelissen, and A. Perkis, "MPEG-21: The 21st century multimedia framework," *IEEE Signal Process. Mag.*, vol. 20, no. 2, pp. 53–62, Mar. 2003.

[3] S. Boyd, P. Diaconis, and X. Lin, "Fastest mixing Markov chain on a graph," *SIAM Rev.*, vol. 46, no. 4, pp. 667–689, 2004.

[4] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, 2007, Art. no. 969.

[5] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE Commun. Surv. Tuts.*, vol. 21, no. 3, pp. 2419–2465, Third Quarter 2019.

[6] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[7] L. Gan, "Block compressed sensing of natural images," in *Proc. Int. Conf. Digit. Signal Process.*, 2007, pp. 403–406.

[8] R. Ganjavi and A. R. Sharafat, "Edge-assisted public key homomorphic encryption for preserving privacy in mobile crowdsensing," *IEEE Trans. Serv. Comput.*, vol. 16, no. 2, pp. 1107–1117, Mar./Apr. 2023.

[9] G. Gao, J. Wu, M. Xiao, and G. Chen, "Combinatorial multi-armed bandit based unknown worker recruitment in heterogeneous crowdsensing," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 179–188.

[10] B. Gong, P. Cheng, Z. Chen, N. Liu, L. Gui, and F. De Hoog, "Spatiotemporal compressive network coding for energy-efficient distributed data storage in wireless sensor networks," *IEEE Commun. Lett.*, vol. 19, no. 5, pp. 803–806, May 2015.

[11] J. Hu et al., "Towards demand-driven dynamic incentive for mobile crowdsensing systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4907–4918, Jul. 2020.

[12] L. Kozma, "Review of compact data structures-a practical approach by Gonzalo Navarro," *ACM SIGACT News*, vol. 49, no. 3, pp. 9–13, 2018.

[13] L. Li et al., "Privacy preserving participant recruitment for coverage maximization in location aware mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 9, pp. 3250–3262, Sep. 2022.

[14] F. Liu, M. Lin, Y. Hu, C. Luo, and F. Wu, "Design and analysis of compressive data persistence in large-scale wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 10, pp. 2685–2698, Oct. 2015.

[15] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5117–5144, Sep. 2016.

[16] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. Shen, "Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1317–1331, Jun. 2020.

[17] D. T. Sandwell, R. D. Müller, W. H. F. Smith, E. Garcia, and R. Francis, "New global marine gravity model from CryoSat-2 and Jason-1 reveals buried tectonic structure," *Science*, vol. 346, no. 6205, pp. 65–67, 2014.

[18] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 36–58, Sep. 2001.

[19] A. Talari and N. Rahnavard, "CStorage: Decentralized compressive data storage in wireless sensor networks," *AdHoc Netw.*, vol. 37, no. 2, pp. 475–485, 2016.

[20] M.-S. Tsai, A.-Y. Liang, C.-L. Tsai, P.-W. Lai, M.-W. Lin, and M.-C. Chen, "Nonlinear compression toward high-energy single-cycle pulses by cascaded focus and compression," *Sci. Adv.*, vol. 8, no. 31, 2022, Art. no. eabo1945.

[21] L. Wang, W. Liu, D. Zhang, Y. Wang, E. Wang, and Y. Yang, "Cell selection with deep reinforcement learning in sparse mobile crowdsensing," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst.*, Vianna, Austria, 2018, pp. 1543–1546.

[22] T. Wu et al., "Blockchain-based anonymous data sharing with accountability for Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 5461–5475, Mar. 2023.

[23] M. Xiao, G. Gao, J. Wu, S. Zhang, and L. Huang, "Privacy-preserving user recruitment protocol for mobile crowdsensing," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 519–532, Apr. 2020.
[24] G. Yang, Z. Shi, S. He, and J. Zhang, "Socially privacy-preserving data collection for crowdsensing," *IEEE Trans. Veh. Technol*, vol. 69, no. 1, pp. 851–861, Jan. 2020.
[25] Q. Yuan, H. Zhou, Z. Liu, J. Li, F. Yang, and X. Shen, "CESense: Cost-effective urban environment sensing in vehicular sensor networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3235–3246, Sep. 2019.
[26] Y. Zeng, R. Calvo-Palomino, D. Giustiniano, G. Bovet, and S. Banerjee, "Adaptive uplink data compression in spectrum crowdsensing systems," *IEEE/ACM Trans. Netw.*, early access, Jan. 30, 2023, doi: 10.1109/TNET.2023.3239378.
[27] C. Zhang, L. Zhu, C. Xu, J. Ni, C. Huang, and X. Shen, "Location privacy-preserving task recommendation with geometric range query in mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4410–4425, Dec. 2022.
[28] F. Zhang et al., "TADOC: Text analytics directly on compression," *VLDB J.*, vol. 30, no. 2, pp. 163–188, 2021.
[29] S. Zhou, X. Deng, C. Li, Y. Liu, and H. Jiang, "Recognition-oriented image compressive sensing with deep learning," *IEEE Trans. Multimedia*, vol. 25, pp. 2022–2032, Jan. 2022.
[30] S. Zhou, Y. He, Y. Liu, C. Li, and J. Zhang, "Multi-channel deep networks for block-based image compressive sensing," *IEEE Trans. Multimedia*, vol. 23, pp. 2627–2640, 2020.
[31] S. Zhou, Y. He, S. Xiang, K. Li, and Y. Liu, "Region-based compressive networked storage with lazy encoding," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 6, pp. 1390–1402, 2019.
[32] S. Zhou, Y. Lian, D. Liu, H. Jiang, Y. Liu, and K. Li, "Compressive sensing based distributed data storage for mobile crowdsensing," *ACM Trans. Sensor Netw.*, vol. 18, no. 2, 2022, Art. no. 25.