IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 7, 2020, accepted April 6, 2020, date of publication April 9, 2020, date of current version April 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986825

PAADA: Physical-World-Aware Approximate Data Acquisition for Image Sensor Networks

QIAN MA, XIA LI, GUANYU LI, BO NING, MEI BAI, AND XITE WANG
College of Information Science and Technology, Dalian Maritime University, Dalian 116000, China

Corresponding author: Qian Ma (maqian@dlmu.edu.cn)

This work was supported in part by the China Postdoctoral Science Foundation under Project 2019M661077, in part by the National Natural Science Foundation of China under Grant 6197020349, Grant 61602076, Grant 61702072, Grant 61976032, Grant 61772102, and Grant 61751205, in part by the Natural Science Foundation of Liaoning Province under Grant 20180540003, in part by the CERNET Innovation Project under Grant NGII20190902, and in part by the National Key Research and Development Program of China under Grant 2017YFC1404606.

ABSTRACT To observe the complicated physical world, sensor networks are widely used for data collection. Moreover, due to the limited energy, storage and computation capacity of sensor nodes, approximate data acquisition in adaptive sampling manner is a wide choice. Nevertheless, existing data acquisition methods are most designed for univariate data (e.g., temperature), and thus not applicable to image data with high dimensions and complex structures. In this paper, we propose a framework of *Physical-world-Aware Adaptive Data Acquisition* (PAADA) for image sensor networks, to sample data adaptively with pre-specified error bound. First, based on the convolutional autoencoders (CAEs), PAADA compresses the high-dimensional image data into a feature vector with a handful of hidden variables which compactly capture the key features of the image data. Second, PAADA designs a *Physical-world-aware Adaptive Sampling* (PAS) algorithm based on the Hermitee interpolation. Under the feature space, the PAS algorithm adjusts the sampling frequency automatically by considering the change trend of the feature vector. In addition, the feature vectors at non-sampling time points can be recovered with $O(\epsilon)$ approximation guarantee to the ground truths. Next, PAADA recovers the image data at non-sampling time points based on the recovered feature vectors. Finally, for each sensor, PAADA returns an image series composed of sampled images (at sampling time points) and approximate images (at non-sampling time points). Experiments on real-world datasets demonstrate that the proposed PAADA has high performance in both accuracy and energy consumption.

INDEX TERMS Image sensor networks, adaptive data acquisition, physical-world-aware approximation.

I. INTRODUCTION

The physical world presents an incredibly rich set of input modalities, e.g., acoustics, image, motion, vibration, heat, light, etc. For collecting information from the complicated physical world, a large number of sensing devices are deployed to acquire data continuously from monitored environments in many real applications [1]–[3]. In particular, in the area of big data [4], wireless networks in combination with image sensors (i.e., image sensor networks) [5], [6] open an up multitude of previously unthinkable sensing application, e.g., surveillance [7], [8], home automation [9] and wildlife monitoring [10].

Currently, given an image sensor network (ISN), most existing works acquire image data from the physical world based on the *equi-frequency sampling* (EFS) methods [11], [12]. However, the EFS methods may distort the observed physical world when the sampling frequency is low, which results in the misunderstanding of the physical world. One possible solution is to increase the sampling frequency. However, for an ISN, collecting a large amount of raw data with high sampling frequency may lead to several undesirable problems. First, the lifetime of the ISN may be short with much more energy consumption of sensor nodes. It is reported in [13] that the lifetime of a sensor network can be increased extraordinarily from 1 month to more than 18 months by lowering the data sampling rate of sensor nodes. Second, the data quality may deteriorate due to the packet loss caused by network congestion. With a high sampling frequency, the data collected intensively while the bandwidth of sensor nodes is limited, which may potentially result in network congestion. Third, the image data collected by ISN

The associate editor coordinating the review of this manuscript and approving it for publication was Mu Zhou.

are usually high-dimensional and large-scale, which further burdens data transmission and storage of the sensor nodes. In summary, the data collection with high sampling frequency is impractical for ISN due to the problems introduced above.

To address this issue, one possible way is *adaptive sampling*, i.e., to let the sensors skip sampling whenever, based on existing samples, the future readings we intend to skip can be estimated accurately. Currently, several adaptive sampling methods [14]–[17] have been proposed for acquiring data from the physical world. However, most existing algorithms are built based on EFS methods and focus on saving energy with certain accuracy loss. This may overlook some critical points and lead to distortion of the observed physical world, even provide incorrect information to the upper applications. Moreover, existing algorithms cannot handle the high-dimensional image data as they are proposed for univariate (one-dimensional) data.

To overcome the shortages of existing adaptive sampling methods, we propose a framework of *Physical-world-Aware Adaptive Data Acquisition* for image sensor networks, named PAADA, to support approximate data acquisition with a pre-specified error bound. In PAADA, we first compress the high-dimensional image into a *feature vector* (consisting of a few hidden variables) based on the *convolutional autoencoders* (CAEs), where the hidden variables can capture the key features of the original image data. Then in the low-dimensional feature space, the next sampling time point is determined adaptively. Moreover, the data at the non-sampling time points between two consecutive sampling time points are recovered approximately. Finally, the sampled and approximate images are concatenated to form the final image series.

Our contributions in this paper are summarized as follows.

- We propose a novel framework, namely PAADA, for physical-world-aware approximate image data acquisition under the energy, computation and storage limitations of the image sensor nodes.
- Owing to the high dimensionality of the image data, we propose a data compression algorithm based on the CAEs model. After data compression, the feature vectors of the original image data are used to later data processing, which significantly reduces the computational cost.
- A physical-world-aware data sampling algorithm is proposed based on Hermitee interpolation. By sampling discrete image data adaptively with considering the change trend of continuously-varying physical world, the proposed algorithm extends the lifetime of the ISN with controllable accuracy sacrifice.
- Based on the sampled data, we propose a data approximation algorithm to recover the image data at non-sampling time points by employing the Hermitee interpolation and the decode part of the trained CAEs model. With the guarantee of $O(\epsilon)$-approximation to physical world on feature level, the images are recovered with high accuracy.
- Finally, we conduct extensive experiments over two real-world datasets to evaluate the performance of the proposed PAADA framework.

Generally, to convert data into valuable information, there are five steps: data acquisition, data preprocessing, data storage, data modeling and analysis, and data visualization. It is notable that this work focuses on the image data acquisition with considerations of energy consumption and data accuracy simultaneously for an ISN. In other words, the proposed PAADA can be considered as an assemble serving for data collection. Moreover, the proposed PAADA returns the images at each time point where the images at non-sampling time points are recovered approximately rather than sampling from the real physical world. Thus there is no increase of data size and our proposed PAADA does not incur extra burden for the following data processing tasks. On the other hand, it is notable that this paper focuses on saving energy from the task of data sampling within each single sensor node in ISN. The main energy consumption is usually considered from data transmission and data reception. However, the energy consumption of data sensing is not always insignificant [18], [19], especially for energy-hungry sensors, e.g., image sensors, gas sensors, which also consume a large amount of energy during sensing data compared to that of data transmission [20], [21].

The reminder of this paper is structured as follows. In Section II, we introduce the preliminaries. In Section III, we overview the PAADA and then introduce the data compression technique in Section IV. After introducing the solution for physical-world-aware adaptive sampling in Section V, we present the data approximation algorithm in Section VI. We report the experimental results in Section VII, and finally conclude the paper in Section VIII.

II. PRELIMINARIES

A. PROBLEM STATEMENT

Suppose an image sensor network S consists of n sensor nodes $S = \{s_1, s_2, \cdots, s_n\}$.

Definition 1: An **image** $I_i(t)$ is the data collected by sensor s_i at time t to describe the observed physical world. Generally, an image is usually represented by a matrix of pixels values with size $(H \times W) \in \mathbb{R}^d$.

Definition 2: The **feature vector** $F_i(t) = \left(F_{i,1}(t), F_{i,2}(t), \cdots, F_{i,h}(t)\right)$ is the h-dimensional hidden representation (embedding) of the image $I_i(t)$, where the redundancy information in the original image is removed while essential aspects of the data are preserved.

Definition 3: An **image series** $\mathcal{I}_i = \left\{I_i(t_s), I_i(t_1), \cdots, I\left(t_f\right)\right\}$ is a series of images collected by sensor s_i at each time point t in range $\left[t_s, t_f\right]$, where t_s $(t_s = t_0)$ and t_f are the start and end time points of users observing the physical world by the image sensor network S.

As introduced earlier, the sensor node s_i does not sample data at each time point with the constraints of remaining energy and network bandwidth. Let $T_i^s =$

$\left\{t^s_{i,1}, t^s_{i,2}, \cdots, t^s_{i,c}, \cdots,\right\}$ $\left(t_s \leq t^s_{i,c} \leq t_f\right)$ be the sampling time points, and $T^u_i = \left\{t^u_{i,1}, t^u_{i,2}, \cdots, t^u_{i,l}, \cdots,\right\}$ $\left(t_s \leq t^u_{i,l} \leq t_f\right)$ be the non-sampling time points of s_i in range $\left[t_s, t_f\right]$. At non-sampling time points, the image data are not sampled by S and thus need to be approximated by estimation.

Definition 4: An **approximate image** $\widehat{I_i}(t)$ is the estimation of $I_i(t)$ where t is a non-sampling time point.

Definition 5: An **approximate feature vector** $\widehat{F_i}(t)$ is the estimation of $F_i(t)$ at a non-sampling time point t.

Definition 6: An **approximate image series** $\widehat{\mathcal{I}_i} = \left\{\widehat{I_i}(t_0), \widehat{I_i}(t_1), \cdots, \widehat{I_i}\left(t_f\right)\right\}$ is the approximation of the image series $\mathcal{I}_i$, where the data at sampling time points are the images essentially collected by sensor s_i while the data at non-sampling time points are the approximate images estimated based on the observed images.

Problem: Given an image sensor network S where each sensor collects image data from the physical world at some discrete sampling time points, our goal is to recover the image data at non-sampling time points by approximation and return an approximate image series $\widehat{\mathcal{I}_i}$ for each sensor node $s_i \in S$.

For tackling the above problem, we propose a new framework, called *Physical-world-Aware Approximate Data Acquisition* (PAADA), to obtain data from physical world based on an image sensor network. Note that the data acquisition of various sensor nodes in a given ISN are independent. For each sensor $s_i \in S$, PAADA aims to tackle the following issues:

- Compressing the high-dimensional image data $I_i(t)$ into a low-dimensional feature vector $F_i(t)$ to dramatically reduce the complexity of further data processing, where $t_s \leq t \leq t_f$.
- Determining the next sampling time $t^s_{i,c+1} \in T^s_i$ at the current sampling time $t^s_{i,c} \in T^s_i$.
- Approximating the image data at each non-sampling time point $t^u_{i,j} \in T^u_i$.

Example 1: The dataset PRID[1] provides the image sequences extracted from multiple person trajectories from two cameras. Figure 1 shows the flowchart of PAADA to process an example segment of image series in PRID dataset. As shown, at the sampling time point $t^s_{i,1}$, the image is captured by the camera. Then, PAADA computes the next sampling time point $t^s_{i,2}$ based on $t^s_{i,1}$ for sampling data adaptively to save energy and storage resource. Next, PAADA recovers the data at the non-sampling time points between $t^s_{i,1}$ and $t^s_{i,2}$ with high accuracy. With the same line, the sampling and approximation processes are repeated in $\left[t^s_{i,2}, t^s_{i,3}\right]$. Finally, an approximate image series of sensor s_i is expected to be returned.

B. RELATED WORK

Since the abilities on sensing, computation, storage and communication of sensor nodes in ISN are quite limited, adaptive sampling in data acquisition is essential to reduce energy consumption and obtain high accuracy of sensed data. Nowadays, many adaptive sampling methods have been proposed for acquiring data from the physical world by WSNs.

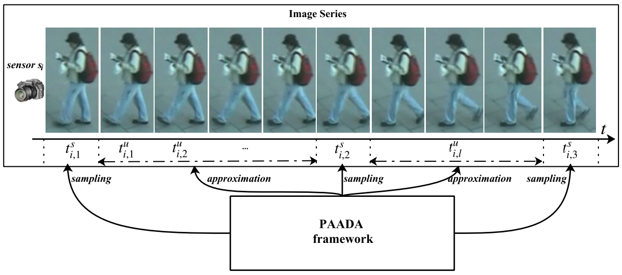


FIGURE 1. The flow chart of the proposed PAADA.

EDSAS [17] adapts the sampling rate of a sensor node by employing temporal data correlations based on its historical data segment. *LiteSense*, proposed in [22], is an adaptive sampling scheme, oriented to WSNs aiming at improving the trade-off between capturing data accurately and saving energy. Reference [20] proposes an energy aware adaptive sampling (ESAS) approach which combines an adaptive sampling algorithm with an energy management technique optimized for energy harvesting WSN. However, the adaptive sampling algorithm adopted in [20] is traditional rather than a novel proposed one. In the same way, [23] combines an adaptive sampling and a novel transmission reduction technique into a single energy efficient algorithm. In [24], the sampling rate is minimized based on the battery level, energy harvesting level and characteristics of the gathered data. Reference [25] rules sensing frequency of sensor node based on the battery power levels of sensor nodes, disregarding the variability of the observed parameter.

The methods in [26]–[29] are proposed for periodic sensor networks. To prolong the lifetime of a periodic sensor network, [26] proposes an adaptive data collection protocol (ADaC) where the sampling rates of sensor nodes are adapted based on the similarity of the data between periods by using Euclidean distance. In [27], each sensor adapts its sampling rate based on the dependence of measurements variance to the physical changing dynamics. Specifically, the authors use the ANOVA model to determine whether there are any significant differences between the means of different data sets collected in successive periods. The ADiDaG framework proposed in [28] is applied in distributed manner for periodic WSN applications. It adapts the sensor sampling rate based on Longest Common Subsequence (LCS). The suggested method in [29] gives every node the capacity to recognize data repetition between gathered data over time, by employing overlap similarity functions and providing sampling with an adaptive rate.

Moreover, there are several adaptive sampling approaches for various real-world applications. AdaSense [30] determines the optimal sampling rates of sensor nodes based on genetic programming to reduce data acquired in activity

[1] https://www.tugraz.at/institute/icg/research/team-bischof/lrs/downloads/prid11/

detection and multi-activity classification. DDASA [19] is a data-driven adaptive sampling algorithm proposed for sensor networks used in automated water quality monitoring. The sampling frequency of each sensor node is adapted based on the observed water quality. Reference [31] is proposed for landslide monitoring system, where the sampling rate dynamically varies depending on the sensor/network-level contexts. However, there are only three sampling rates predefined manually, which insufficient obviously. In [32], an adaptive sampling approach based on the input characteristics for air monitoring sensor networks is proposed. The algorithm adapts the sampling period based on the similarity between the current and the previous sensed data.

Disadvantages: However, most of the adaptive sampling approaches introduce above face issues summarized as follows.

(1) Most existing works focus on saving energy to prolong the lifetime of the sensing networks, rather than acquiring data accurately to recover the real world physical world. The accuracies of them are even worse than EFS (Equi-Frequency Sampling) methods.

(2) The basic idea of most existing methods is that a sensor node increases its sampling rate when it notices an obvious change of measurements, while it decreases its sampling rate when the change is lower than a threshold. However, the threshold used to filter the abrupt change is difficult to determine. Moreover, most existing works determine the change trend of the physical world by comparison current data with a segment of collected historical data, which is not accurate. In other words, the loss of the data acquisition by adaptive sampling is computed as the difference between the data acquired in different data segments, rather than the difference between the acquired data with the their ground truths.

(3) All these algorithms are focus on the data with single variable (i.e., just for one single time series), which are not competent for high-dimensional image data.

The data acquisition algorithm proposed in [33] is the most related one to our work. It adjusts the sampling frequency automatically based on Hermite/Spline interpolation by considering the changing trend of the physical world. Moreover, the accuracy of data acquisition can achieve certain approximation to the real physical world. However, as mentioned above, it is also only suitable for univariate data and thus cannot handle the high-dimensional image data. In this paper, we propose the PAADA framework to obtain the high-dimensional image data from the real physical world with high accuracy in an energy-efficient manner.

III. THE PAADA FRAMEWORK

In this section, we give an overview of the proposed PAADA framework. As shown in Figure 2, the proposed PAADA consists of three components:

(C1) Data Compression compresses the original image data into low-dimensional feature vectors (which contains h hidden variables) based on the *convolutional autoencoders* (CAEs). The solution contains two stages: training stage and testing stage. In the training stage, given a set of images as the input, a CAEs model is trained to extract feature vectors which can be used to reconstruct the input. Afterwards, for any image $I_i(t)$ in the testing stage, the feature vector $F_i(t)$ of $I_i(t)$ is obtained from the learned CAEs model directly.

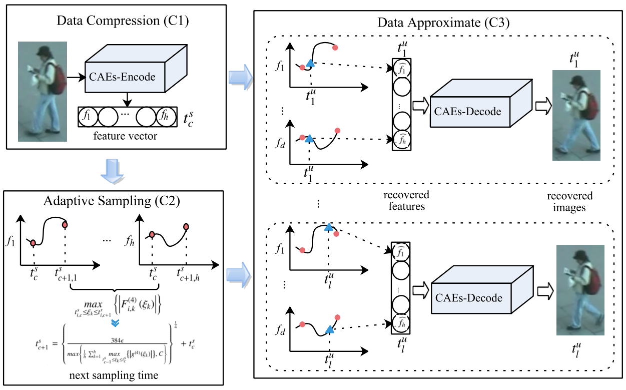

Data Compression (C1)
CAEs-Encode
f_1
f_h
t_c^s
feature vector
Adaptive Sampling (C2)
f_1
f_h
t_c^s
$t_{c+1,1}^s$
t_c^s
$t_{c+1,h}^s$
$\max_{t_{i,c}^s \le \xi_k \le t_{i,c+1}^s} \left\{ \left| F_{i,k}^{(4)}(\xi_k) \right| \right\}$
$t_{c+1}^s = \left\{ \frac{384\epsilon}{\max\left\{ \frac{1}{h} \sum_{k=1}^{h} \max_{t_{c-1}^s \le \xi_k \le t_c^s} \{ |g^{(4)}(\xi_k)| \}, C \right\}} \right\}^{\frac{1}{4}} + t_c^s$
next sampling time
Data Approximate (C3)
f_1
f_d
t_1^u
$\widehat{f_1}$
$\widehat{f_h}$
CAEs-Decode
recovered features
recovered images
t_l^u

FIGURE 2. The framework of the proposed PAADA.

(C2) Adaptive Sampling determines the future sampling time points adaptively under the low-dimensional feature space. As introduced earlier, at current time point $t_{i,c}^s$, the image $I\left(t_{i,c}^s\right)$ collected by sensor node $s_i \in S$ is transformed to a h-dimensional feature vector $F_i\left(t_{i,c}^s\right) = \left(F_{i,1}\left(t_{i,c}^s\right), F_{i,2}\left(t_{i,c}^s\right), \cdots, F_{i,h}\left(t_{i,c}^s\right)\right)$. Under the feature space, we compute the next sampling time point $t_{i,c+1}^s$ by exploring the characteristics of Hermitee interpolation [34].

(C3) Data Approximation recovers the images at non-sampling time points since data are not acquired from the physical world by the sensor node s_i. At a non-sampling time point $t_{i,j}^u \in T_i^u$, the approximate feature vector $\widehat{F}_i\left(t_{i,j}^u\right)$ of image $I_i\left(t_{i,j}^u\right)$ is estimated based on the Hermitee curve generated in the adaptive sampling component. Further, the approximate image $\widehat{I}_i\left(t_{i,j}^u\right)$ is recovered based on $\widehat{F}_i\left(t_{i,j}^u\right)$ by employing the decode part of the CAEs model learned in the data compression component.

Finally, PAADA returns an approximate image series $\widehat{\mathcal{I}}_i$ for each sensor $s_i \in S$. Algorithm 1 provides the pseudo-code of PAADA. In PAADA, the time window $\left[t_s, t_f\right]$ (given by users) is divided into multiple time intervals by the discrete sampling time points. At the current sampling time point $t_{i,c}^s$, each sensor node $s_i \in s$ samples data and computes its next sampling time point $t_{i,c+1}^s$ by using the historical data in time interval $\left[t_{i,c-1}^s, t_{i,c}^s\right]$ (Line 13-18). As shown in Line 18, PAADA needs to store the historical data at four time points, i.e., $t_{i,c-1}^t$, $t_{i,c-1}^t + \Delta t$, t', $t_{i,c}^s - \Delta t$, where $t_{i,c-1}^s$ is the last sampling time point before $t_{i,c}^s$ (see details in Section V). At the next sampling time point $t_{i,c+1}^s$, sensor s_i samples data and recovers the image data which are not

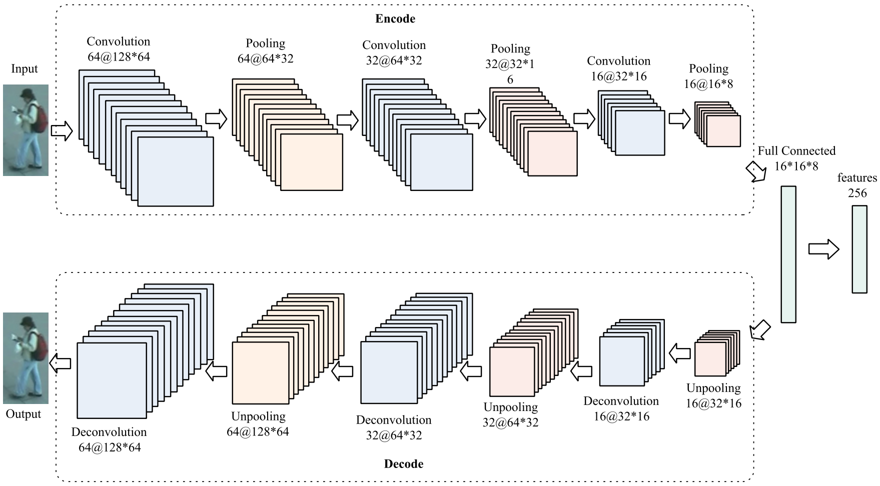


FIGURE 3. The architecture of the CAE model.

sampled at non-sampling time points in $\left[t_{i,c}^{s}, t_{i,c+1}^{s}\right]$ (Line 23-28). Repeating the process introduced above for each time interval $\left[t_{i,c}^{s}, t_{i,c+1}^{s}\right] \subseteq \left[t_s, t_f\right]$, the approximate image series for each sensor $s_i \in S$ is obtained. Next, we introduce each component in detail in the following sections.

IV. DATA COMPRESSION

In PAADA, we adopt the deep convolutional autoencoders (CAEs) to compress the high-dimensional image data since the CAEs model is the state-of-the-art tool for feature extraction from unlabeled images [35]. The architecture of the CAEs model in this paper is shown in Figure 3. We take an image (with size 128×64) in PRID dataset as an example for illustration. As shown, the model contains two parts: *encode* and *decode*.

The encode part consists of three convolutional and pooling layers, respectively. The first convolutional layer filters the $128 \times 64 \times 1$ input image with 64 kernels of size 3×3 and a stride of 1 pixel. The second convolutional layer takes as input the (pooled) output of the first convolutional layer and filters it with 32 kernels of size $3 \times 3 \times 1$. The third convolutional layer has 16 kernels of size $3 \times 3 \times 1$ connected to the (pooled) outputs of the second convolutional layer. The objective of the *convolution* operation is to extract the high-level *features*, e.g., edges, gradient orientation, from the input image. Note that between two convolutional layers, there is a pooling layer to streamline the underlying computation by reducing the spatial size of the convolved features. We adopt the max pooling as it performs a lot better than average pooling in most cases. As for the activation function in each convolution layer, we adopt the most-frequently-used ReLU [36].

After a series of convolution process, the non-linear combinations of the high-level features (as represented by the output of the final convolution layer) are learned by adding two fully-connected layers. Thus far, we have converted a high-dimensional input image $I_i(t)$ into a suitable form of feature vector (embedding) $F_i(t)$ with low dimension.

The decode part which consists of three deconvolutional and unpooling layers respectively has symmetric architecture with the encode. The objective of decode is to transform the extracted feature vector $F_i(t)$ going in the opposite direction of the convolutions, and then recover the input image, i.e., obtain the recovered image $\widehat{I_i}(t)$. With the idea that if the feature vector $F_i(t)$ captures the key features of the original input $I_i(t)$, it allows a good reconstruction $\widehat{I_i}(t)$. Thus the loss function is to minimize the reconstruction error between $\widehat{I_i}(t)$ and $I_i(t)$ as shown below.

$$Min\ \mathcal{L}(\theta, \phi; I_i(t)) = \left\| I_i(t) - \widehat{I_i}(t) \right\|^2, \qquad (1)$$

where $\left\| I_i(t) - \widehat{I_i}(t) \right\|^2$ denotes the mean square error (MSE) distortion between the original image $I_i(t)$ and the recovered image $\widehat{I_i}(t)$. θ and ϕ are the parameters during data transformation in the encode and decode, respectively.

To compress the high-dimensional image data into feature vectors based on CAEs model, there are two, i.e., training and testing, stages. Given a set of images as the input, a CAEs model is learned (i.e., the parameters θ and ϕ are optimized) in the training stage by optimizing the loss function. Moreover, we adopt Adam [37] as the optimizer and use a batch size of 128 to train the model up to 1000 iterations. The learning rate is kept as a fixed value of 0.001. In addition, since the image series are sampled continuously with time passing, the CAEs model can be retrained by adding new sampled data periodically. Next, in the testing stage, given an image $I_i(t)$ as the input, its feature vector $F_i(t)$ (low-dimensional embedding) is obtained by a series of data transformation in the encode part of the learned CAEs model.

Complexity Analysis: Since the trained CAEs model is used for feature extraction repeatedly and retrained periodically in off-line, the time complexity of model training is not included. In the testing stage, we use the encode part of the trained CAEs model to obtain the feature vector of each input image $I_i(t)$. The time complexity of a convolutional (containing pooling) layer is related to the size of input image and the numbers of input and output channels in the

Algorithm 1 PAADA($S, t_s, t_f, \epsilon, f_{max}$)

```
Input : The image sensor network $S = \{s_1, s_2, \cdots, s_n\}$, the start time $t_s$ and
        end time $t_f$, the user-specified error threshold $\epsilon$, the maximum sampling
        frequency $f_{max}$.
Output: The collection of image series $\{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_n\}$ of $S$.
foreach sensor node $s_i$ do
    $\widehat{\mathcal{I}}_i = \Phi$, $\Delta t = \frac{1}{f_{max}}$
    // initial the current and next sampling time points
    $t = t_{i,c-1}^s = t_s, t' = t_s + 2\Delta t, t_{i,c}^s = t_s + 4\Delta t$
    while $t < t_f$ do
        if
        $t == t_{i,c-1}^s \; or \; t == \left(t_{i,c-1}^s + \Delta t\right) \; or \; t == t' \; or \; t == \left(t_{i,c}^s - \Delta t\right)$
        then
            // sampling the image data at time $t$
            $\widehat{\mathcal{I}}_i \leftarrow \{I_i(t)\}$
            // compressing the image data into feature vector
            $F_i(t)$=DataCompression($I_i(t)$)
            $t = t + \Delta t$
        if $t == t_{i,c}^s$ then
            // sampling the image data at time $t$
            $\widehat{\mathcal{I}}_i \leftarrow \left\{I_i\left(t_{i,c}^s\right)\right\}$
            // compressing the image data into feature vector
            $F_i\left(t_{i,c}^s\right)$=DataCompression($I_i\left(t_{i,c}^s\right)$)
            // determining the next sampling time point
            $t_{i,c+1}^s$ = AdaptiveSampling($t_{i,c-1}^s, t', t_{i,c}^s, F_i\left(t_{i,c-1}^s\right),$
            $F_i\left(t_{i,c-1}^s + \Delta t\right), F_i(t'), F_i\left(t_{i,c}^s\right), F_i\left(t_{i,c}^s - \Delta t\right), \epsilon$)
            // selecting $t'$ randomly from $\left(t_{i,c}^s, t_{i,c+1}^s\right)$ used for next
            sampling time determination
            $t'$ =RandSelect($\left(t_{i,c}^s, t_{i,c+1}^s\right)$)
            $t = t + \Delta t$
        else if $t == t_{i,c+1}^s$ then
            Sampling the image data $I_i\left(t_{c+1}^s\right)$
            $\widehat{\mathcal{I}}_i \leftarrow \left\{I_i\left(t_{c+1}^s\right)\right\}$
            // compressing the image data into feature vector
            $F_i\left(t_{i,c+1}^s\right)$=DataCompression($I_i\left(t_{i,c+1}^s\right)$)
            // Approximating the image data at the non-sampling time
            points between $t_{i,c}^s$ and $t_{i,c+1}^s$, i.e.,
            $t_{i,j}^u \in \left(t_{i,c}^s + \Delta t, t'\right) \cup \left(t', t_{i,c+1}^s - \Delta t\right)$
            $\left\{\widehat{I}_i\left(t_{i,j}^u\right)\right\}$=DataApproximation($t_{i,c}^s, t_{i,c}^s + \Delta t, t_{i,c+1}^s - \Delta t,$
            $t_{i,c+1}^s, F_i\left(t_{i,c}^s\right), F_i\left(t_{i,c}^s + \Delta t\right), F_i\left(t_{i,c+1}^s - \Delta t\right), F_i\left(t_{i,c+1}^s\right)$)
            // adding the data into $\widehat{\mathcal{I}}_i$
            $\widehat{\mathcal{I}}_i \leftarrow \left\{\widehat{I}_i\left(t_{i,j}^u\right)\right\}\left(t_{i,j}^u \in \left(t_{i,c}^s + \Delta t, t'\right) \cup \left(t', t_{i,c+1}^s - \Delta t\right)\right)$
            $t_{i,c-1}^s = t_{i,c}^s, t_{i,c}^s = t_{i,c+1}^s$
        else
            $t = t + \Delta t$
return $\{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_n\}$
```

convolutional layer. These parameters are known and fixed once the structure of CAEs model is designed and the data set is determined. Thus for a sensor s_i, if it samples the image data at time t, then the time complexity of data compression can be considered as a constant, denoted by $O(c)$ in this paper. For the image sensor network S which consists of n sensors, the time complexity of data compression is $O(cn)$.

V. PHYSICAL-WORLD-AWARE ADAPTIVE SAMPLING

In this section, we propose a *Physical-world-aware Adaptive Sampling* (PSA) algorithm, to acquire image data from the physical world in an adaptive manner. Moreover, to reduce the computational consumption of data processing, the algorithm is applied over the low-dimensional feature vectors (derived based on the CAEs model introduced in Section IV).

For each sensor node $s_i \in S$, let $t_{i,c}^s$ and $t_{i,c+1}^s$ be two consecutive sampling time points. Suppose in $\left[t_{i,c}^s, t_{i,c+1}^s\right]$, the feature series of the image series $\mathcal{I}_i$ is $\mathcal{F}_i = \left\{F_i\left(t_{i,c}^s\right), F_i\left(t_{i,1}^u\right), \cdots, F_i\left(t_{i,l'}^u\right), F_i\left(t_{i,c+1}^s\right)\right\}$ where $\left\{t_{i,1}^u, \cdots, t_{i,1'}^u\right\}$ are the non-sampling time points between $t_{i,c}^s$ and $t_{i,c+1}^s$, while the approximate feature series of the approximate image series $\widehat{\mathcal{I}}_i$ is $\widehat{\mathcal{F}}_i = \left\{F_i\left(t_{i,c}^s\right), \widehat{F}_i\left(t_{i,1}^u\right), \cdots, \widehat{F}_i\left(t_{i,l'}^u\right), F_i\left(t_{i,c+1}^s\right)\right\}$.

Definition 7: $\widehat{\mathcal{F}}_i$ is $O(\epsilon)$-approximate to $\mathcal{F}_i$ in $\left[t_{i,c}^s, t_{i,c+1}^s\right]$ if and only if there exist a constant C satisfying that $\left|\widehat{F}_i\left(t_{i,j}^u\right) - F_i\left(t_{i,j}^u\right)\right| \leq C\epsilon$ for any non-sampling time point $t_{i,j}^u \in \left[t_{i,c}^s, t_{i,c+1}^s\right]$, where ϵ is a user-specified error bound.

Specifically, based on Definition 2 and Definition 5, both $F_i\left(t_{i,j}^u\right)$ and $\widehat{F}_i\left(t_{i,j}^u\right)$ are h-dimensional vectors, i.e., $F_i\left(t_{i,j}^u\right) = \left(F_{i,1}\left(t_{i,j}^u\right), F_{i,2}\left(t_{i,j}^u\right), \cdots, F_{i,h}\left(t_{i,j}^u\right)\right)$, and $\widehat{F}_i\left(t_{i,j}^u\right) = \left(\widehat{F}_{i,1}\left(t_{i,j}^u\right), \widehat{F}_{i,2}\left(t_{i,j}^u\right), \cdots, \widehat{F}_{i,h}\left(t_{i,j}^u\right)\right)$. We define that $\left|\widehat{F}_i\left(t_{i,j}^u\right) - F_i\left(t_{i,j}^u\right)\right| = \frac{1}{h}\sum_{k=1}^{h}\left|\widehat{F}_{i,k}\left(t_{i,j}^u\right) - F_{i,k}\left(t_{i,j}^u\right)\right|$.

Based on the above definition, for any sensor $s_i \in S$, the physical-world-aware adaptive sampling problem aims to predict the next sampling time point $t_{i,c+1}^s$ at the current sampling time point $t_{i,c}^s$, where the approximate feature series $\widehat{\mathcal{F}}_i$ is $O(\epsilon)$-approximation to $\mathcal{F}_i$ in $\left[t_{i,c}^s, t_{i,c+1}^s\right]$.

A. MATHEMATICAL FOUNDATIONS

Given a sensor node $s_i \in S$, when t is an independent variable, the feature vector $F_i(t)$ is actually a collection of curves where each curve $F_{i,k}(t)$ is the continuous curve reflecting the evolution of the kth feature with time t. At time $t_{i,c}^s$ and $t_{i,c+1}^s$, s_i samples the image data $I_i\left(t_{i,c}^s\right)$ and $I_i\left(t_{i,c+1}^s\right)$, respectively. Moreover, s_i computes the corresponding feature vectors $F_i\left(t_{i,c}^s\right)$ and $F_i\left(t_{i,c+1}^s\right)$ based on the learned CAEs model, respectively. Accordingly, the first derivative $F_i^{(1)}\left(t_{i,c}^s\right)$ and $F_i^{(1)}\left(t_{i,c+1}^s\right)$ are computed. With these values, based on *Hermitee interpolation*, each curve $F_{i,k}(t) \in F_i(t)$ $(1 \leq k \leq h)$ is computed by

$$\widehat{F}_{i,k}(t) = F_{i,k}\left(t_{i,c}^s\right)\phi_1(t) + F_{i,k}\left(t_{i,c+1}^s\right)\phi_2(t) + F_{i,k}^{(1)}\left(t_{i,c}^s\right)\phi_3(t) + F_{i,k}^{(1)}\left(t_{i,c+1}^s\right)\phi_4(t), \tag{2}$$

where $t \in \left[t_{i,c}^s, t_{i,c+1}^s\right]$, $\phi_1(t) = \frac{\left(t_{i,c+1}^s - t\right)^2\left(2t - 3t_{i,c}^s + t_{i,c+1}^s\right)}{\left(t_{i,c+1}^s - t_{i,c}^s\right)^3}$, $\phi_2(t) = \frac{\left(t - t_{i,c}^s\right)^2\left(3t_{i,c+1}^s - 2t - t_{i,c}^s\right)}{\left(t_{i,c+1}^s - t_{i,c}^s\right)^3}$, $\phi_3(t) = \frac{\left(t - t_{i,c}^s\right)\left(t - t_{i,c+1}^s\right)^2}{\left(t_{i,c+1}^s - t_{i,c}^s\right)^2}$, and $\phi_4(t) = \frac{\left(t - t_{i,c+1}^s\right)\left(t - t_{i,c}^s\right)^2}{\left(t_{i,c+1}^s - t_{i,c}^s\right)^2}$.

Theorem 1: For $\forall t \in \left[t^s_{i,c}, t^s_{i,c+1}\right]$, $\left|\widehat{F}_i(t) - F_i(t)\right| \leq \epsilon$, if $t^s_{i,c+1} - t^s_{i,c} \leq \left\{\frac{384\epsilon}{\frac{1}{h}\sum_{k=1}^{h} \max\limits_{t^s_{i,c} \leq \xi_k \leq t^s_{i,c+1}} \left\{\left|F^{(4)}_{i,k}(\xi_k)\right|\right\}}\right\}^{\frac{1}{4}}$, where $\epsilon > 0$ is a user-specified error bound.

Proof. Let

$$G(x) = \frac{1}{h}\sum_{k=1}^{h}\left[\widehat{F}_{i,k}(x) - F_{i,k}(x) - \frac{w(x)}{w(t)}\left(\widehat{F}_{i,k}(t) - F_{i,k}(t)\right)\right],$$

where $w(t) = \left(t - t^s_{i,c}\right)^2\left(t - t^s_{i,c+1}\right)^2, x, t \in \left[t^s_{i,c}, t^s_{i,c+1}\right]$.

Based on Equation (2), we have $G\left(t^s_{i,c}\right) = G\left(t^s_{i,c+1}\right) = G(t) = 0$, and $G^{(1)}\left(t^s_{i,c}\right) = G^{(1)}\left(t^s_{i,c+1}\right) = 0$. Thus, there exists $\zeta \in \left[t^s_{i,c}, t^s_{i,c+1}\right]$ satisfying that $G^{(4)}(\zeta) = 0$ by repeatedly using Rolle Theorem [38]. Further, based on the definition of $G(x)$, we have

$$G^{(4)}(\zeta) = \frac{1}{h}\sum_{k=1}^{h}\left[F^{(4)}_{i,k}(\zeta) - \frac{24}{w(t)}\left(\widehat{F}_{i,k}(t) - F_{i,k}(t)\right)\right] = 0,$$

and thereby,

$$\begin{aligned}
&\frac{1}{h}\sum_{k=1}^{h}\left|\widehat{F}_{i,k}(t) - F_{i,k}(t)\right| \\
&\quad = \frac{1}{h}\sum_{k=1}^{h}\left|\frac{w(t)}{24}F^{(4)}_{i,k}(\zeta)\right| \\
&\quad = \frac{1}{24h}\sum_{k=1}^{h}\left|\left(t - t^s_{i,c}\right)^2\left(t - t^s_{i,c+1}\right)^2\right|\left|F^{(4)}_{i,k}(\zeta)\right|.
\end{aligned}$$

Since $t \in \left[t^s_{i,c}, t^s_{i,c+1}\right]$, $\left|\left(t - t^s_{i,c}\right)\left(t - t^s_{i,c+1}\right)\right| \leq \frac{1}{4}\left(t^s_{i,c+1} - t^s_{i,c}\right)^2$, we have

$$\begin{aligned}
&\frac{1}{h}\sum_{k=1}^{h}\left|\widehat{F}_{i,k}(t) - F_{i,k}(t)\right| \\
&\quad \leq \frac{1}{24h}\sum_{k=1}^{h}\left[\frac{1}{4}\left(t^s_{i,c+1} - t^s_{i,c}\right)^2\right]^2 \max_{t^s_{i,c} \leq \xi_k \leq t^s_{i,c+1}}\left\{\left|F^{(4)}_{i,k}(\xi_k)\right|\right\} \\
&\quad \leq \frac{1}{384h}\left(t^s_{i,c+1} - t^s_{i,c}\right)^4\sum_{k=1}^{h}\max_{t^s_{i,c} \leq \xi_k \leq t^s_{i,c+1}}\left\{\left|F^{(4)}_{i,k}(\xi_k)\right|\right\}. \qquad (3)
\end{aligned}$$

Combining $\left|\widehat{F}_i(t) - F_i(t)\right| = \frac{1}{h}\sum_{k=1}^{h}\left|\widehat{F}_{i,k}(t) - F_{i,k}(t)\right|$ with Equation (3), there is

$$\begin{aligned}
\left|\widehat{F}_i(t) - F_i(t)\right| &\leq \frac{1}{384h}\left(t^s_{i,c+1} - t^s_{i,c}\right)^4\sum_{k=1}^{h}\max_{t^s_{i,c} \leq \xi_k \leq t^s_{i,c+1}}\left\{\left|F^{(4)}_{i,k}(\xi_k)\right|\right\} \\
&\leq \epsilon.
\end{aligned}$$

Thus $t^s_{i,c+1} - t^s_{i,c} \leq \left\{\frac{384h\epsilon}{\sum_{k=1}^{h}\max\limits_{t^s_{i,c} \leq \xi_k \leq t^s_{i,c+1}}\left\{\left|F^{(4)}_{i,k}(\xi_k)\right|\right\}}\right\}^{\frac{1}{4}}$. The theorem is proved. □

Algorithm 2 AdaptiveSamping($t^s_{i,c-1}, t', t^s_{i,c}, F_i\left(t^s_{i,c-1}\right), F_i\left(t^s_{i,c-1} + \Delta t\right), F_i\left(t'\right), F_i\left(t^s_{i,c}\right), F_i\left(t^s_{i,c} - \Delta t\right), \epsilon$)

```
Input : The last sampling time point $t^s_{i,c-1}$, the current sampling time point $t^s_{i,c}$,
        the randomly selected time point $t' \in \left(t^s_{i,c-1}, t^s_{i,c}\right)$, the feature vectors
        at the time points introduced above, the user-specified error threshold $\epsilon$.
Output: Next sampling time point $t^s_{i,c+1}$
1 foreach feature $F_{i,k}(t)$ $(1 \leq k \leq h)$ in $F_i(t)$ do
2     $g_{i,k}(t) = \mathbf{LI}\left(F_{i,k}\left(t^s_{i,c-1}\right), F_{i,k}\left(t^s_{i,c-1}+\Delta t\right), F_{i,k}\left(t'\right), I_i\left(t^s_{i,c}-\Delta t\right),\right.$
3         $\left.F_{i,k}\left(t^s_{i,c}\right)\right)$
4     $\max_{t^s_{i,c} \leq \xi_k \leq t^s_{i,c+1}}\left\{\left|F^{(4)}_{i,k}(\xi_k)\right|\right\} = \max_{t^s_{i,c-1} \leq \xi_k \leq t^s_{i,c}}\left|g^{(4)}(\xi_k)\right|$
5 $t^s_{c+1} = \left\{\frac{384\epsilon}{max\left\{\frac{1}{h}\sum_{k=1}^{h}\max_{t^s_{c-1} \leq \xi_k \leq t^s_c}\left\{\left|g^{(4)}(\xi_k)\right|\right\}, C\right\}}\right\}^{\frac{1}{4}} + t^s_c$
6 return $t^s_{c+1}$
```

B. ALGORITHM

Based on Theorem 1, to determine the next sampling time point $t^s_{i,c+1}$ at the current sampling time point $t^s_{i,c}$, the main issue needs to be tackled is to estimate the maximum forth derivative of each feature, i.e. $\max\limits_{t^s_{i,c} \leq \xi_k \leq t^s_{i,c+1}}\left\{\left|F^{(4)}_{i,k}(\xi_k)\right|\right\}(1 \leq k \leq h)$ in range $\left[t^s_{i,c}, t^s_{i,c+1}\right]$. Since only the data at $t^s_{i,c}$ is sampled, while the future data in $\left(t^s_{i,c}, t^s_{i,c+1}\right]$ are unknown, we estimate the $\max\limits_{t^s_{i,c} \leq \xi_k \leq t^s_{i,c+1}}\left\{\left|F^{(4)}_{i,k}(\xi_k)\right|\right\}$ based on the historical data in range $\left[t^s_{i,c-1}, t^s_{i,c}\right]$, where $t^s_{i,c-1}$ is the last sampling time point before $t^s_{i,c}$. The rationale behind the strategy is that $\max\limits_{t^s_{i,c} \leq \xi_k \leq t^s_{i,c+1}}\left\{\left|F^{(4)}_{i,k}(\xi_k)\right|\right\}$ in range $\left[t^s_{i,c}, t^s_{i,c+1}\right]$ is close to that in range $\left[t^s_{i,c-1}, t^s_{i,c}\right]$, as the monitored physical world usually changes continuously and smoothly.

Given the current sampling time point $t^s_{i,c}$ and the data at historical time points $t^s_{i,c-1}$, $t^s_{i,c-1} + \Delta t$, t', $t^s_{i,c} - \Delta t$, Algorithm 2 shows the pseudo-code of the PAS algorithm. First, for each feature $F_{i,k}$ $(1 \leq k \leq h)$ in the feature vector $F_i(t)$, PAS constructs a quartic interpolation function $g_{i,k}(t)$ by Lagrange interpolation [39], where $t \in \left[t^s_{i,c-1}, t^s_{i,c}\right]$ (Line 2). Next, the maximum forth derive of $F_{i,k}(t)$ in range $\left[t^s_{i,c}, t^s_{i,c+1}\right]$ is estimated by using the quartic curve $g_{i,k}(t)$ in range $\left[t^s_{i,c-1}, t^s_{i,c}\right]$ (Line 3). Finally, PSA averages the h maximum forth derives $\frac{1}{h}\sum_{k=1}^{h}\max\limits_{t^s_{i,c} \leq \xi_k \leq t^s_{i,c+1}}\left\{\left|F^{(4)}_{i,k}(\xi_k)\right|\right\}$ and determines the next sampling time point $t^s_{c+1} = \left\{\frac{384\epsilon}{max\left\{\frac{1}{h}\sum_{k=1}^{h}\max\limits_{t^s_{c-1} \leq \xi_k \leq t^s_c}\left\{\left|g^{(4)}(\xi_k)\right|\right\}, C\right\}}\right\}^{\frac{1}{4}} + t^s_c$ based on Theorem 1 (Line 4).

Theorem 2: $\widehat{\mathcal{F}_i}$ is $O(\epsilon)$-approximate to $\mathcal{F}_i$ in $\left[t_s, t_f\right]$.

Proof. $\widehat{\mathcal{F}_i}$ is $O(\epsilon)$-approximate to $\mathcal{F}_i$ in $\left[t_s, t_f\right]$ if and only if $\widehat{\mathcal{F}_i}$ is $O(\epsilon)$-approximate to $\mathcal{F}_i$ in each time interval $\left[t^s_{i,c}, t^s_{i,c+1}\right] \subseteq \left[t_s, t_f\right]$. Based on Equation (3) in the proof of Theorem 1, we have

$$\left|\widehat{F}_i(t) - F_i(t)\right| = \frac{1}{h}\sum_{k=1}^{h}\left|\widehat{F}_{i,k}(t) - F_{i,k}(t)\right| \leq \frac{1}{384h}\left(t^s_{i,c+1} - t^s_{i,c}\right)^4 \sum_{k=1}^{h} \max_{t^s_{i,c}\leq\xi_k\leq t^s_{i,c+1}}\left\{\left|F^{(4)}_{i,k}(\xi_k)\right|\right\}.$$

Moreover, $t^s_{i,c+1} - t^s_{i,c} = \left\{\frac{384\epsilon}{max\left\{\frac{1}{h}\sum_{k=1}^{h}\max_{t^s_{c-1}\leq\xi_k\leq t^s_c}\left\{\left|g^{(4)}(\xi_k)\right|\right\}, C\right\}}\right\}^{\frac{1}{4}}$ based on the PSA algorithm. Thus, there is

$$\left|\widehat{F}_i(t) - F_i(t)\right| \leq \epsilon \frac{\frac{1}{h}\sum_{k=1}^{h}\max_{t^s_{i,c}\leq\xi_k\leq t^s_{i,c+1}}\left\{\left|F^{(4)}_{i,k}(\xi_k)\right|\right\}}{max\left\{\frac{1}{h}\sum_{k=1}^{h}\max_{t^s_{c-1}\leq\xi_k\leq t^s_c}\left\{\left|g^{(4)}(\xi_k)\right|\right\}, C\right\}}.$$

Since $F^{(4)}_i(t)$ is continuous in most applications, $F^{(4)}_i(t)$ is bounded in range $\left[t^s_{i,c}, t^s_{i,c+1}\right]$. By adding $max\left\{\frac{1}{h}\sum_{k=1}^{h}\max_{t^s_{c-1}\leq\xi_k\leq t^s_c}\left\{\left|g^{(4)}(\xi_k)\right|\right\}, C\right\} \geq C$, we have $\left|\widehat{F}_i(t) - F_i(t)\right| = O(\epsilon)$. The conclusion is proved since for any time interval $\left[t^s_{i,c}, t^s_{i,c+1}\right] \subseteq \left[t_s, t_f\right]$, $\widehat{\mathcal{F}_i}$ is $O(\epsilon)$-approximate to $\mathcal{F}_i$. □

Algorithm 3 DataApproximation($t^s_{i,c}$, $t^s_{i,c} + \Delta t$, $t^s_{i,c+1} - \Delta t$, $t^s_{i,c+1}$, $F_i\left(t^s_{i,c}\right)$, $F_i\left(t^s_{i,c} + \Delta t\right)$, $F_i\left(t^s_{i,c+1} - \Delta t\right)$, $F_i\left(t^s_{i,c+1}\right)$)

```
Input : The sampling time points $t^s_{i,c}, t^s_{i,c} + \Delta t, t^s_{i,c} - \Delta t, t^s_{i,c+1}$ and the
        corresponding feature vectors at these time points.
Output: Recovered image $\left\{\widehat{I}_i\left(t^u_{i,j}\right)\right\}$ at each non-sampling time point
        $t^u_{i,j} \in \left[t^s_{i,c}, t^s_{i,c+1}\right]$.
//Constructing the Hermite interpolation curves in range $\left[t^s_{i,c}, t^s_{i,c+1}\right]$
foreach feature (element) $F_{i,k}(t) \in F_i(t)$ do
    $F^{(1)}_{i,k}\left(t^s_{i,c}\right) = \frac{F_{i,k}\left(t^s_{i,c}+\Delta t\right) - F_{i,k}\left(t^s_{i,c}\right)}{\Delta t}$
    $F^{(1)}_{i,k}\left(t^s_{i,c+1}\right) = \frac{F_{i,k}\left(t^s_{i,c+1}\right) - F_{i,k}\left(t^s_{i,c+1}-\Delta t\right)}{\Delta t}$
    $\widehat{F}_{i,k}(t) = F_{i,k}\left(t^s_{i,c}\right)\phi_1(t) + F_{i,k}\left(t^s_{i,c+1}\right)\phi_2(t) + F^{(1)}_{i,k}\left(t^s_{i,c}\right)\phi_3(t) + F^{(1)}_{i,k}\left(t^s_{i,c+1}\right)\phi_4(t)$
foreach non-sampling time point $t^u_{i,j}$ in range $\left[t^s_{i,c}, t^s_{i,c+1}\right]$ do
    $\widehat{F}_{i,k}\left(t^u_{i,j}\right) = F_{i,k}\left(t^s_{i,c}\right)\phi_1\left(t^u_{i,j}\right) + F_{i,k}\left(t^s_{i,c+1}\right)\phi_2\left(t^u_{i,j}\right) + F^{(1)}_{i,k}\left(t^s_{i,c}\right)\phi_3\left(t^u_{i,j}\right) + F^{(1)}_{i,k}\left(t^s_{i,c+1}\right)\phi_4\left(t^u_{i,j}\right)$
    $\widehat{F}_i\left(t^u_{i,j}\right) = \left\{\widehat{F}_{i,1}\left(t^u_{i,j}\right), \cdots, \widehat{F}_{i,h}\left(t^u_{i,j}\right)\right\}$
    $\widehat{I}_i\left(t^u_{i,j}\right) = CAEs(\widehat{F}_i\left(t^u_{i,j}\right))$
return the collection of approximate images $\left\{\widehat{I}_i\left(t^u_{i,j}\right)\right\}$
```

Complexity Analysis: As introduced in Algorithm 2, at a sampling time point, to determine the next sampling time point for a sensor node $s_i \in S$, the proposed PAS algorithm needs to estimate the $\max_{t^s_{i,c}\leq\xi_k\leq t^s_{i,c+1}}\left\{\left|F^{(4)}_{i,k}(\xi_k)\right|\right\}$ by constructing a quartic interpolation function for each feature $F_{i,k}(t)$ $(1 \leq k \leq h)$ in $F_i(t)$. Thus the time complexity of adaptive sampling is $O(hn)$ where h and n are the numbers of features in a feature vector and sensor nodes in the ISN, respectively.

VI. DATA APPROXIMATION

As introduced earlier, each sensor $s_i \in S$ samples data only at time points $t^s_{i,c}, t^s_{i,c} + \Delta t, t', t_{i,c+1} - \Delta t, t_{i,c+1}$ in each time interval $\left[t^s_{i,c}, t^s_{i,c+1}\right] \subseteq \left[t_s, t_f\right]$, while at other (non-sampling) time points $t^u_{i,j} \in \left[t_s, t_f\right]$, sensor s_i does not sample data. Thus, the image data at these non-sampling time points need to be approximate by estimation.

Recall that for a non-sampling time point $t^u_{i,j} \in \left[t^s_{i,c}, t^s_{i,c+1}\right]$, its feature vector is $F_i\left(t^u_{i,j}\right) = \left\{F_{i,1}\left(t^u_{i,j}\right), F_{i,2}\left(t^u_{i,j}\right), \cdots, F_{i,h}\left(t^u_{i,j}\right)\right\}$. Referring to Equation (2), for each feature in the feature vector, we can derive a Hermitee curve $\widehat{F}_{i,k}(t)$ where $t \in \left[t^s_{i,c}, t^s_{i,c+1}\right]$, $F^{(1)}_{i,k}\left(t^s_{i,c}\right) = \frac{F_{i,k}\left(t^s_{i,c}+\Delta t\right) - F_{i,k}\left(t^s_{i,c}\right)}{\Delta t}$, and $F^{(1)}_{i,k}\left(t^s_{i,c+1}\right) = \frac{F_{i,k}\left(t^s_{i,c+1}\right) - F_{i,k}\left(t^s_{i,c+1}-\Delta t\right)}{\Delta t}$. Thus the approximate feature $\widehat{F}_{i,k}\left(t^u_{i,j}\right)$ of $F_{i,k}\left(t^u_{i,j}\right)$ $\left(t^u_{i,j} \in \left[t^s_{i,c}, t^s_{i,c+1}\right]\right)$ can be computed based on the Hermite curve, and thereby the approximate feature vector $\widehat{F}_i\left(t^u_{i,j}\right)$ is derived. Moreover, the approximate feature vector $\widehat{F}_i\left(t^u_{i,j}\right)$ is $O(\epsilon)$-approximate to $F_i\left(t^u_{i,j}\right)$, i.e., $\left|\widehat{F}_i\left(t^u_{i,j}\right) - F_i\left(t^u_{i,j}\right)\right| \leq C\epsilon$ where $t^u_{i,j} \in \left[t^s_{i,c}, t^s_{i,c+1}\right]$.

After feature approximation, by taking the $\widehat{F}_i\left(t^u_{i,j}\right)$ as the input, the trained CAEs model (introduced in Section IV) outputs the approximate $\widehat{I}_i\left(t^u_{i,j}\right)$ after a series of data transformation going through in the decode part of the model. Finally, for each time interval $\left[t^s_{i,c}, t^s_{i,c+1}\right] \subseteq \left[t_s, t_f\right]$, the approximate image $\widehat{I}_i\left(t^u_{i,j}\right)$ at any non-sampling time point $t^u_{i,j} \in \left[t^s_{i,c}, t^s_{i,c+1}\right]$ is obtained. Afterwards, the approximate image series $\mathcal{I}_i$ of sensor $s_i \in S$ is derived. Algorithm 3 shows the pseudo-code of the data approximation.

Complexity Analysis: Based on Algorithm 3, the time complexity of the feature approximation for a sensor node s_i is $O(h)$ since a feature vector consists of h hidden variables. The time complexity of image approximation based on the decode part of the CAEs model is $O(c)$ as introduced earlier. Thus the time complexity of data approximation is $O((h + c)n)$

In summary, the time complexity of three components, i.e., data compression, adaptive sampling and data approximation, in PAADA are $O(cn)$, $O(hn)$ and $O((h + c)n)$, respectively, where c is a constant and h is the number of hidden variables in a feature vector which can be also regarded as a constant due to $d \ll n$. Moreover, the components of data compression and adaptive sampling are used at sampling time points, while the component of data approximation is used at non-sampling time points. Therefore, the time complexity of the proposed PAADA is $O\left((h+c)\,n\right)$, which increases linearly with the number of sensor nodes in the ISN.

VII. EXPERIMENTS

We evaluate the performance of the proposed algorithms empirically in this section. All the methods are implemented in Python and the experiments are conducted on a PC with 64-bit Windows operating system, 16GB memory and 3.2GHz CPU. Moreover, we use a NVIDIA GTX 1080 Ti GPU with 3584 cores and 11GB memory.

A. EXPERIMENTAL SETTINGS

1) DATASETS

To evaluate the performance of REMAIN, we adopt two real-world datasets from the *Person Re-identification Datasets*[2] as follows:

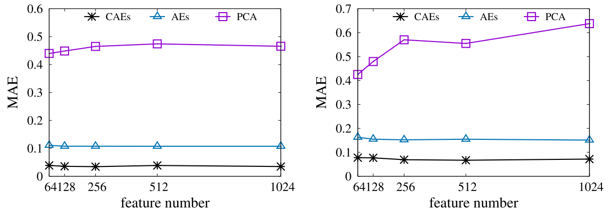
(1) *PRID* [40] consists of the image series which record the movement of persons. The images are extracted from trajectories recorded by two different, static surveillance cameras. We adopt the data from camera A which contains the image series of 298 persons, where each series consists of 100 images on average.

(2) *iLIDS-VID* [41], [42] involves 600 image sequences of 300 distinct individuals extracted from iLIDS MCTS dataset. We select 100 image sequences from the dataset to conduct our experiments, where each sequence contains 75 images averagely.

In both of the datasets, each original image has size 64 × 128. For model training, we split the datasets into training set and testing set with ratio 80/20, where the training set is used to learn an effective CAEs model to compress the high-dimensional image data into low-dimensional feature vectors, and the testing set is used to evaluate the performance of the proposed algorithms.

2) ALGORITHMS FOR COMPARISON

To the best of our knowledge, there is few existing adaptive data acquisition algorithm is competent for high-dimensional image data. In this paper, we first compress the high-dimensional image data into low-dimensional feature vectors. Afterwards, the adaptive sampling algorithms are applied over the feature vectors. Thus, over the same feature vectors obtained by CAEs model (which is adopted in PAADA for data compression), we experimentally compare the proposed PAADA with existing adaptive sampling algorithms EDSAS [17] and DDASA [19] since they are the latest methods. The reason of choosing CAEs model for image data compression is that it performs better than existing widely adopted data compression algorithms PCA [43] and AEs model [44], [45] (as shown in Section VII-B).



(a) PRID (b) iLIDS-VID

FIGURE 4. Performance on data compression.

3) PERFORMANCE METRIC

We adopt the Mean Absolute Error (MAE) to evaluate the accuracy of the acquired approximate image series. The lower MAE is, the performance of data acquisition is better.

Moreover, to evaluate the energy consumption of the proposed PAADA, we suppose that $6.53uW$ power is consumed for sampling an image by an CMOS image sensor which is widely used in imaging system [46]. Then the total energy consumed is the summation of energy consumption of multiple sampling times.

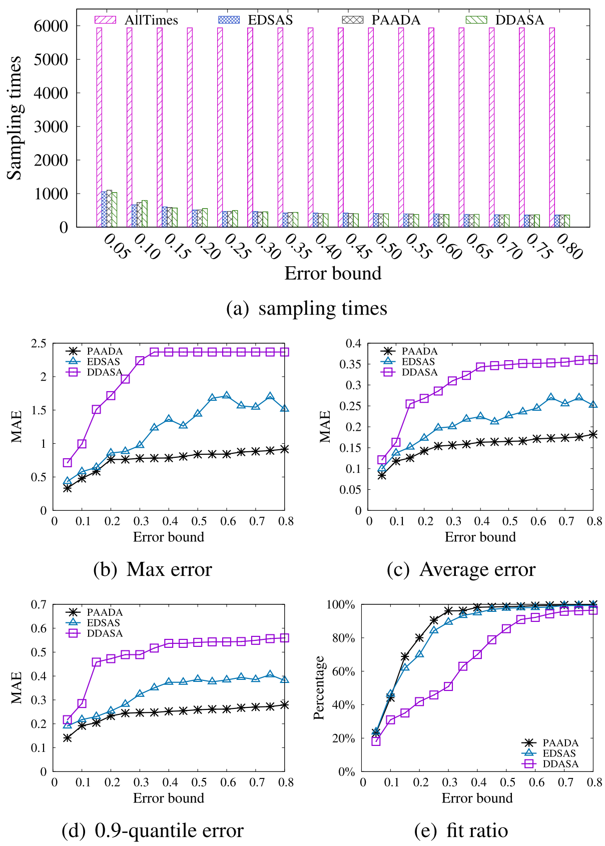
B. DATA COMPRESSION EVALUATION

As introduced earlier, various adaptive sampling algorithms apply over feature vectors obtained by the CAEs model. We choose the CAEs model because the obtained feature vectors are more effective than those obtained by existing data compression algorithms. We adopt the reconstruction error between the input (image data) and output (reconstructed image data) of the data compression model to evaluate the effectiveness of the feature vectors. The rationale behind is that more key features are captured, the reconstructed images are closer to the ground truths. Moreover, the structure of the CAEs adopted in this paper is shown in Figure. 3, where there are three convolutional and deconvolutional layers, respectively. In the paper, we implement the AEs model which contains three encoder and decoder layers, respectively. Additionally, the structure of AEs model for the two datasets is $[8192 \rightarrow 4096 \rightarrow 2048 \rightarrow \mathit{feature\_num} \rightarrow 2048 \rightarrow 4096 \rightarrow 8192]$, where the encode and decode parts are symmetric.

As shown in Figure. 4, the reconstructed error of CAEs model (adopted by PAADA) is the lowest, which verifies that the feature vectors obtained by CAEs model are the best. Since the PCA compresses data by exploring the linear correlations amongst attributes of the data,[3] it is not surprising that its performance is the worst in handling image data with complex (non-linear) correlations and structures. By comparison, the performance of AEs model is better than that of PCA but worse than that of CAEs model. Since we capture the features of the input image data in a fine-grained

[2] http://robustsystems.coe.neu.edu/sites/robustsystems.coe.neu.edu/files/systems/projectpages/reiddataset.html

[3] Note that each pixel is a an attribute.



(a) sampling times

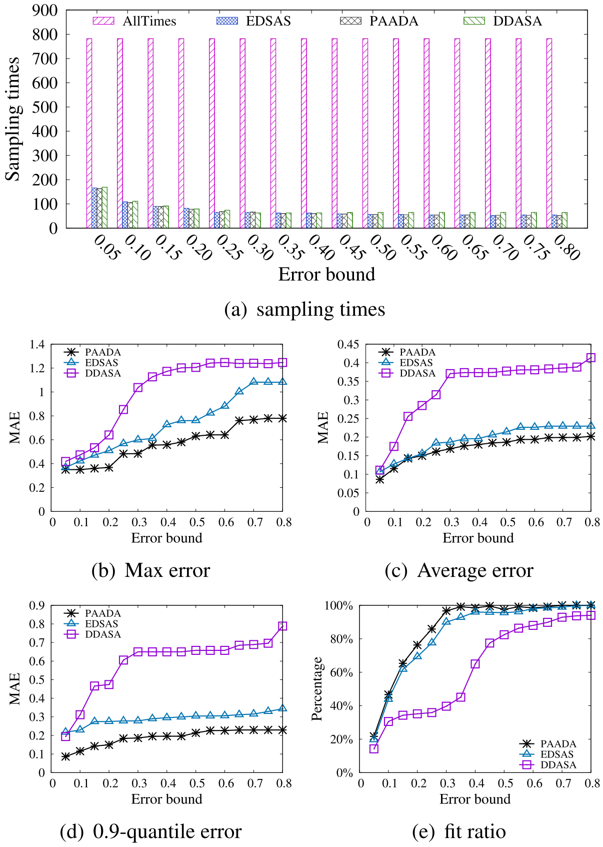
(b) Max error

(c) Average error

(d) 0.9-quantile error

(e) fit ratio

FIGURE 5. Performance on feature vector recovery over PRID dataset.



(a) sampling times

(b) Max error

(c) Average error

(d) 0.9-quantile error

(e) fit ratio

FIGURE 6. Performance on feature vector recovery over iLIDS dataset.

manner by using CAEs model, more local features can be captured accurately.

C. COMPARISON WITH EXISTING TECHNIQUES

In this section, we compare the proposed PAADA with existing adaptive sampling methods, i.e., EDSAS and DDASA, on the PRID and iLIDS-VID datasets.

To obtain the approximate image series, we first recover the feature vectors at non-sampling time points based on the Hermite interpolation by using the feature vectors acquired by PAADA, EDSAS and DDASA, respectively. Afterwards, the approximate images are recovered based on the decode part of the trained CAEs model by using the approximate feature vectors. Note that the error bound in DDASA (denoted by t in [19] and denoted by t_D in this paper for distinguishing the time point t) is over the average value of a segment historical data. To make t_D is in the same order of magnitude with ϵ in PAADA and EDSAS, the threshold t_D in DDASA is computed as $t_D = \frac{\epsilon}{\frac{1}{N}\sum_{j=1}^{N} F_i(t)}$ where N is the length of sliding-window (set as 20 in this paper).

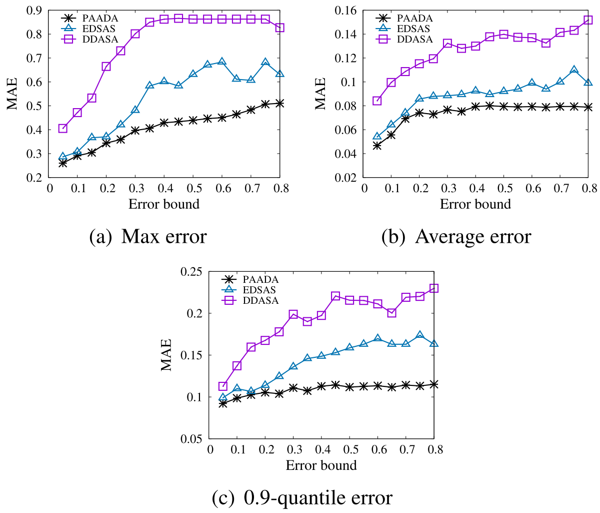
1) PERFORMANCE ON FEATURE VECTOR RECOVERY

In this subsection, we first evaluate the performances of various adaptive sampling algorithms on feature vector recovery.

First, Figure 5(a) shows the number of sampling times by varying error bound over PRID dataset. As a large error bound means a great tolerance of data quality, the sampling frequency is correspondingly low. Thus the numbers of sampling times of all three algorithms decrease with the increase of the error bound. On the other hand, as shown in Figure 5(a), the numbers of sampling times of three adaptive sampling algorithms are close with each other, and largely smaller than the size of all time points (denoted by 'AllTimes'). Thus the energy consumption is reduced and the lifetime of network is lengthened.

Second, as shown in Figure 5(b)-(d), with the increase of ϵ, the errors (i.e., max error, average error and 0.9-quantile error) of all adaptive sampling algorithms correspondingly increase. As the error bound is larger, the number of sampling times is lower, and thereby the recovery accuracy decreases. However, the errors of the proposed PAADA are the lowest and increase slowly while ϵ becomes large . Figure 5(b) shows that the max errors of DDASA and EDSAS are extremely large while the max error of PAADA is quite small. The reason is that DDASA and EDSAS consider the changing trend of physical world by a set of past-period data which may be sparse (i.e., a few data points are sampled in the sliding time window) and thus has the real physical world distortion problem. For the average error and 0.9-quantile error, similar results can be observed in Figure 5(c) and Figure 5(d).

Finally, as introduced in Theorem 2, the feature vectors at the non-sampling time points recovered by Hermite interpolation are $O(\epsilon)$-approximate to the ground truths. As shown

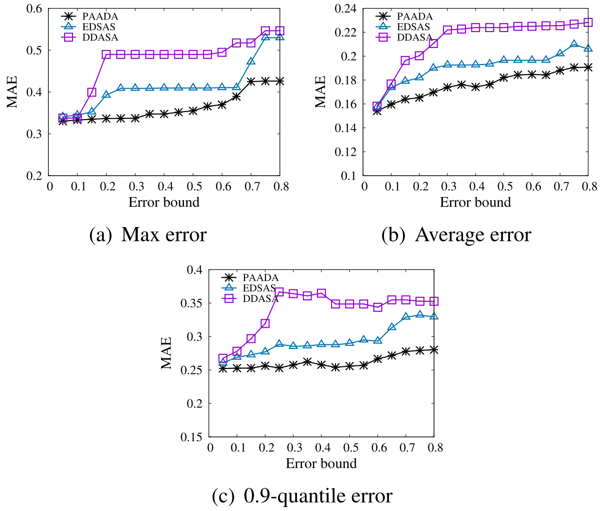


(a) Max error

(b) Average error

(c) 0.9-quantile error

FIGURE 7. Performance on image recovery over PRID dataset.



(a) Max error

(b) Average error

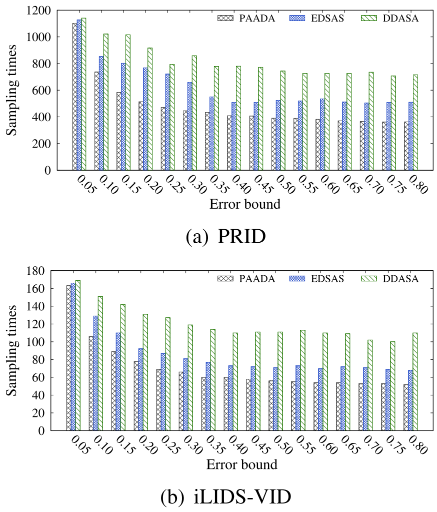
(c) 0.9-quantile error

FIGURE 8. Performance on image recovery over iLIDS-VID dataset.

in Figure 5(b), the max errors of adaptive sampling algorithms may be larger than given error bound ϵ since the sampling interval is determined by posterior estimation. From Figure 5(d), we can see that when $\epsilon \geq 0.25$, the 0.9-quantile error of PAADA is smaller than the given error bound ϵ. By comparison, the 0.9-quantile errors of EDSAS and DDASA are smaller than ϵ when $\epsilon \geq 0.35$ and $\epsilon \geq 0.55$, respectively. Therefore, PAADA guarantees more time points satisfying the precision requirement. Moreover, Figure 5(e) presents the percentage of recovered features satisfying ϵ. As shown, the percentage of recovered features satisfying ϵ in PAADA is the greatest, which verifies again that the proposed PAADA performs better at most time points in accuracy than existing EDSAS and DDASA. In the same line, similar results can be observed in Figure 6 where the performances are evaluated over the dataset iLIDS-VID.

2) PERFORMANCE ON IMAGE RECOVERY

In this subsection, we evaluate the performances of various methods on image recovery where the default error bound $\epsilon = 0.2$. Figure 7 and Figure 8 illustrate the max, average and 0.9-quantile errors of approximate image series estimated based on data at sampling time points obtained by various adaptive sampling algorithms. As shown, the lowest MAEs are achieved by the proposed PAADA, since more recovered feature vectors at non-sampling time points satisfy the precision requirement, while existing EDSAS and DDASA do not aim to reconstruct the physical world. Moreover, with the increase of error bound, it is not surprising that the recovery errors rises. The reasons are as follows. First, the number of sampling times decreases as ϵ increases (as shown in Figure 5(a) and Figure 8(a)). Second, the accuracies of recovered feature vectors become worse with the increase of ϵ as illustrated in Section VII-C.1.



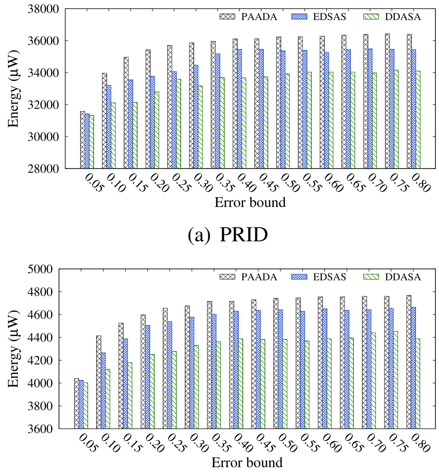
(a) PRID

(b) iLIDS-VID

FIGURE 9. The number of sampling times with the same image recover accuracy.

3) PERFORMANCE ON ENERGY CONSUMPTION

As introduced earlier, we can observe from Figure 5(a) and Figure 6(a) that the numbers of sampling times of EDSAS, DDASA and PAADA are much lower than the size of all times. Thus, it is reasonable to believe that these adaptive sampling algorithms can significantly reduce the energy consumed for data acquisition. Moreover, with almost the same number of sampling times, the accuracies of feature recovery and image recovery based on EDSAS and DDASA are obviously lower than that of PAADA (shown in Figure 5~8). To further show our contribution on energy saving, in this section, we conduct the experiments about the energy consumption of the three adaptive sampling algorithms (i.e., EDSAS, DDASA and PAADA). First, by adjusting the parameters of the EDSAS and DDASA, we let their image recovery accuracies are almost the same with that of PAADA. Then we compare their numbers of sampling times under the scenario of the comparable image recovery accuracy. As shown in Figure 9, to achieve the comparable image recovery accuracy, EDSAS and DDASA need to



(a) PRID

(b) iLIDS-VID

FIGURE 10. The saving energy with the same image recover accuracy.

sample more data, while the number of sampling times of PAADA is the fewest. Thus the proposed PAADA saves more energy.

Furthermore, Figure 10 shows the energy saved based on various adaptive sampling algorithms. For example, in the time interval with 5938 time points, the image data at 514 time points are sampled from the real physical world by PAADA (with $\epsilon = 0.2$), while the image data at the remaining 5424 time points are estimated by approximation. Thus the energy consumed for data acquisition at these 5425 non-sampling time points are saved . As a result, 35418.72 μW energy are saved based on PAADA. As shown in Figure 10, with the comparable image recovery accuracy, the performance of our PAADA is the best.

In summary, compared with EDSAS and DDASA, the energy consumption of the proposed PAADA is the lowest with comparable image recovery accuracy, while the image recovery accuracy is the highest with almost the same number of sampling times. Therefore, the proposed PAADA has high performance in both image accuracy and energy saving.

VIII. CONCLUSION

In this paper, we have studied the data acquisition problem for high-dimensional image data from image sensor networks. Considering the limited energy, storage and computation capacity of sensor nodes, we propose a physical-world-aware adaptive data acquisition framework, called PAADA. To the best of our knowledge, this is the first study on adaptive sampling for image sensor networks. Instead of employing the adaptive sampling algorithm over the original image data, PAADA first compresses the high-dimension image data into low-dimensional feature vectors. Then PAADA adjusts the sampling frequency automatically based on the change trend of the feature vector and the error bound given by users. To obtain the image series of each sensor node, PAADA first recovers the feature vectors at non-sampling time points based on the Hermite interpolation which guarantees that the approximate feature series can achieve $O(\epsilon)$-approximate to the physical world. Afterwards, the approximate image series are recovered based on the approximate feature series and the decode part of the trained CAEs model. Experimental results on two real-world datasets demonstrate the superiority of the proposed PAADA on accuracy and energy consumption.

REFERENCES

[1] L. Zhang, R. Wang, and L. Cui, "Real-time traffic monitoring with magnetic sensor networks," *J. Inf. Sci. Eng.*, vol. 27, no. 4, pp. 1473–1486, 2011.

[2] G. Xu, W. Shen, and X. Wang, "Marine environment monitoring using wireless sensor networks: A systematic review," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2014, pp. 13–18.

[3] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proc. 6th Int. Symp. Inf. Process. Sensor Netw.*, Apr. 2007, pp. 254–263.

[4] L. Yang, Y. Yang, G. B. Mgaya, B. Zhang, L. Chen, and H. Liu, "Novel fast networking approaches mining underlying structures from investment big data," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Jan. 14, 2020, doi: 10.1109/TSMC.2019.2961378.

[5] I. Downes, L. B. Rad, and H. Aghajan, "Development of a mote for wireless image sensor networks," in *Proc. Cognit. Syst. Interact. Sensors (COGIS)*, Paris, France, vol. 3, 2006, p. 1–8.

[6] I. Akyildiz, T. Melodia, and K. Chowdury, "Wireless multimedia sensor networks: A survey," *IEEE Wireless Commun.*, vol. 14, no. 6, pp. 32–39, Dec. 2007.

[7] L. Jiao, Y. Wu, G. Wu, E. Y. Chang, and Y.-F. Wang, "Anatomy of a multicamera video surveillance system," *Multimedia Syst.*, vol. 10, no. 2, pp. 144–163, Aug. 2004.

[8] T. Okazaki and T. Fukase, "Vehicle surveillance system, vehicle surveillance method, and program," U.S. Patent 10 157 541, 2018.

[9] N. Marten, "Home automation weather detection," U.S. Patent 9 948 477, 2018.

[10] J. Zhang, X. Luo, C. Chen, Z. Liu, and S. Cao, "A wildlife monitoring system based on wireless image sensor networks," *Sens. Transducers*, vol. 180, no. 10, p. 104, 2014.

[11] Z. Cai, S. Ji, J. He, and A. G. Bourgeois, "Optimal distributed data collection for asynchronous cognitive radio networks," in *Proc. IEEE 32nd Int. Conf. Distrib. Comput. Syst.*, Jun. 2012, pp. 245–254.

[12] S. Ji and Z. Cai, "Distributed data collection and its capacity in asynchronous wireless sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2113–2121.

[13] C. Guestrin, P. Bodi, R. Thibau, M. Paski, and S. Madde, "Distributed regression: An efficient framework for modeling sensor network data," in *Proc. 3rd Int. Symp. Inf. Process. sensor Netw. IPSN*, 2004, pp. 1–10.

[14] S. Chatterjea and P. J. M. Havinga, "An adaptive and autonomous sensor sampling frequency control scheme for energy-efficient data acquisition in wireless sensor networks," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst.*, Jun. 2008, pp. 60–78.

[15] Y. W. Law, S. Chatterjea, J. Jin, T. Hanselmann, and M. Palaniswami, "Energy-efficient data acquisition by adaptive sampling for wireless sensor networks," in *Proc. Int. Conf. Wireless Commun. Mobile Comput. Connecting World Wirelessly IWCMC*, 2009, pp. 1146–1151.

[16] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proc. 13th Int. Conf. Very Large Data Bases-Volume*, Aug. 2004, pp. 588–599.

[17] M. Gupta, L. V. Shum, E. Bodanese, and S. Hailes, "Design and evaluation of an adaptive sampling strategy for a wireless air pollution sensor network," in *Proc. IEEE 36th Conf. Local Comput. Netw.*, Oct. 2011, pp. 1003–1010.

[18] M. Razzaque and S. Dobson, "Energy-efficient sensing in wireless sensor networks using compressed sensing," *Sensors*, vol. 14, no. 2, pp. 2822–2859, 2014.

[19] T. Shu, M. Xia, J. Chen, and C. de Silva, "An energy efficient adaptive sampling algorithm in a sensor network for automated water quality monitoring," *Sensors*, vol. 17, no. 11, p. 2551, 2017.

[20] B. Srbinovski, M. Magno, F. Edwards-Murphy, V. Pakrashi, and E. Popovici, "An energy aware adaptive sampling algorithm for energy harvesting WSN with energy hungry sensors," *Sensors*, vol. 16, no. 4, p. 448, 2016.

[21] V. Jelicic, M. Magno, D. Brunelli, G. Paci, and L. Benini, "Context-adaptive multimodal wireless sensor network for energy-efficient gas monitoring," *IEEE Sensors J.*, vol. 13, no. 1, pp. 328–338, Jan. 2013.

[22] J. M. C. Silva, K. A. Bispo, P. Carvalho, and S. R. Lima, "LiteSense: An adaptive sensing scheme for WSNs," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2017, pp. 1209–1212.

[23] G. B. Tayeh, A. Makhoul, D. Laiymani, and J. Demerjian, "A distributed real-time data prediction and adaptive sensing approach for wireless sensor networks," *Pervas. Mobile Comput.*, vol. 49, pp. 62–75, Sep. 2018.

[24] J. Yang, T. S. Rosing, and S. S. Tilak, "Leveraging application context for efficient sensing," in *Proc. IEEE 9th Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process. (ISSNIP)*, Apr. 2014, pp. 1–6.

[25] J. Yang, X. Wu, and J. Wu, "Adaptive sensing scheduling for energy harvesting sensors with finite battery," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 98–103.

[26] A. K. M. Al-Qurabat and A. Kadhum Idrees, "Adaptive data collection protocol for extending lifetime of periodic sensor networks," *Qalaai Zanist Sci. J.*, vol. 2, no. 2, pp. 83–92, Apr. 2017.

[27] A. Makhoul, H. Harb, and D. Laiymani, "Residual energy-based adaptive data collection approach for periodic sensor networks," *Ad Hoc Netw.*, vol. 35, pp. 149–160, Dec. 2015.

[28] A. K. Idrees, H. Harb, A. Jaber, O. Zahwe, and M. A. Taam, "Adaptive distributed energy-saving data gathering technique for wireless sensor networks," in *Proc. IEEE 13th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2017, pp. 55–62.

[29] H. Harb, A. Makhoul, A. Jaber, R. Tawil, and O. Bazzi, "Adaptive data collection approach based on sets similarity function for saving energy in periodic sensor networks," *Int. J. Inf. Technol. Manage.*, vol. 15, no. 4, p. 346, 2016.

[30] X. Qi, M. Keally, G. Zhou, Y. Li, and Z. Ren, "AdaSense: Adapting sampling rates for activity recognition in body sensor networks," in *Proc. IEEE 19th Real-Time Embedded Technol. Appl. Symp. (RTAS)*, Apr. 2013, pp. 163–172.

[31] R. Prabha, M. V. Ramesh, V. P. Rangan, P. V. Ushakumari, and T. Hemalatha, "Energy efficient data acquisition techniques using context aware sensing for landslide monitoring systems," *IEEE Sensors J.*, vol. 17, no. 18, pp. 6006–6018, Sep. 2017.

[32] Y. Jon, "Adaptive sampling in wireless sensor networks for air monitoring system," M.S. thesis, Dept. Inf. Technol., Univ. Uppsala, Uppsala, Sweden, 2016, pp. 1–42.

[33] S. Cheng, J. Li, and Z. Cai, "O(ε)-approximation to physical world by sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 3084–3092.

[34] M. Zamani, "A simple 2D interpolation model for analysis of nonlinear data," *Natural Sci.*, vol. 2, no. 6, pp. 641–645, 2010.

[35] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. Int. Conf. Artif. Neural Netw.*, Apr. 2011, pp. 52–59.

[36] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[37] M. Emtiyaz Khan, D. Nielsen, V. Tangkaratt, W. Lin, Y. Gal, and A. Srivastava, "Fast and scalable Bayesian deep learning by weight-perturbation in adam," 2018, *arXiv:1806.04854*. [Online]. Available: http://arxiv.org/abs/1806.04854

[38] J.-C. Evard and F. Jafari, "A complex rolle's theorem," *Amer. Math. Monthly*, vol. 99, no. 9, pp. 858–861, 1992.

[39] J.-P. Berrut and L. N. Trefethen, "Barycentric Lagrange interpolation," *SIAM Rev.*, vol. 46, no. 3, pp. 501–517, Jan. 2004.

[40] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof, "Person re-identification by descriptive and discriminative classification," in *Proc. Scand. Conf. Image Anal.*, May 2011, pp. 91–102.

[41] M. Li, X. Zhu, and S. Gong, "Unsupervised person re-identification by deep learning tracklet association," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 737–753.

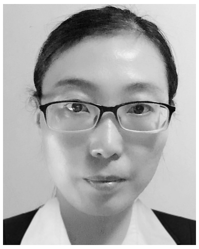
[42] X. Ma, X. Zhu, S. Gong, X. Xie, J. Hu, K.-M. Lam, and Y. Zhong, "Person re-identification by unsupervised video matching," *Pattern Recognit.*, vol. 65, pp. 197–210, May 2017.

[43] L. Wei, R. Zhou, J. Yin, C. Zhu, X. Zhang, and H. Liu, "Latent graph-regularized inductive robust principal component analysis," *Knowl.-Based Syst.*, vol. 177, pp. 68–81, Aug. 2019.

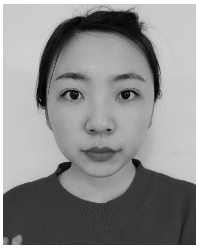
[44] R. Al-Hmouz, W. Pedrycz, A. Balamash, and A. Morfeq, "Logic-driven autoencoders," *Knowl.-Based Syst.*, vol. 183, Nov. 2019, Art. no. 104874.

[45] X. Gao, C. Zhou, F. Chao, L. Yang, C.-M. Lin, T. Xu, C. Shang, and Q. Shen, "A data-driven robotic chinese calligraphy system using convolutional auto-encoder and differential evolution," *Knowl.-Based Syst.*, vol. 182, Oct. 2019, Art. no. 104802.

[46] I. Cevik, X. Huang, H. Yu, M. Yan, and S. Ay, "An ultra-low power CMOS image sensor with on-chip energy harvesting and power management capability," *Sensors*, vol. 15, no. 3, pp. 5531–5554, 2015.

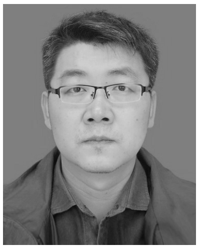


QIAN MA received the Ph.D. degree in computer software and theory from Northeastern University, China, in 2019. She is currently a Postdoctoral Staff of Dalian Maritime University, China. Her research interests are data management and data cleaning.



XIA LI received the B.S. degree in computer science and technology from Liaoning Technical University, China, in 2019. She is currently pursuing the master's degree with the College of Information Science and Technology, Dalian Maritime University, China. Her research interest is data cleaning.

GUANYU LI received the Ph.D. degree in management science and engineering from the Dalian University of Technology, China, in 2010. He is currently a Professor with Dalian Maritime University. His research interests include intelligent information processing, the Internet of Things, and semantic web.



BO NING received the Ph.D. degree in computer science and technology from Northeastern University, China, in 2009. He is currently an Associate Professor with Dalian Maritime University. His research interests include data management and privacy preserving.

MEI BAI received the Ph.D. degree in computer science and technology from Northeastern University, China, in 2016. She is currently an Associate Professor with Dalian Maritime University. Her research interests include data management, cloud computing, and query processing and optimization.

XITE WANG received the Ph.D. degree in computer science and technology from Northeastern University, China, in 2015. He is currently an Associate Professor with Dalian Maritime University. His research interests include big-data management and parallel data processing.