

RESEARCH

Open Access



# Critical data points retrieving method for big sensory data in wireless sensor networks

Tongxin Zhu<sup>1</sup>, Siyao Cheng<sup>1\*</sup>, Zhipeng Cai<sup>2</sup> and Jianzhong Li<sup>1</sup>

## Abstract

With the development and widespread application of wireless sensor networks (WSNs), the amount of sensory data grows sharply and the volumes of some sensory data sets are larger than terabytes, petabytes, or exabytes, which have already exceeded the processing abilities of current WSNs. However, such big sensory data are not necessary for most applications of WSNs, and only a small subset containing critical data points may be enough for analysis, where the critical data points including the extremum and inflection data points of the monitored physical world during given period. Therefore, it is an efficient way to reduce the amount of the big sensory data set by only retrieving the critical data points during sensory data acquisition process. Since most of the traditional sensory data acquisition algorithms were only designed for discrete data and did not support to retrieve critical points from a continuously varying physical world, this paper will study such a problem. In order to solve it, we firstly provided the formal definition of the  $\delta$ -approximate critical points. Then, a data acquisition algorithm based on numerical analysis and Lagrange interpolation is proposed to acquire the critical points. The extensive theoretical analysis and simulation results are provided, which show that the proposed algorithm can achieve high accuracy for retrieving the  $\delta$ -approximate critical points from the monitored physical world.

**Keywords:** Wireless sensor networks, Adaptive sampling, Critical points

## 1 Introduction

The appearance of wireless sensor networks (WSNs) makes it possible to observe the complicated physical world with low cost. Nowadays, WSNs are widely used in many applications, including military defense [1–4], environment monitoring [5, 6], traffic monitoring [7, 8], and structural health monitoring [9–11]. Meanwhile, the amount of sensory data also grows fast with the wide use of sensor networks. For example, the climate data will exceed 100 PB in 2020 according to the report in [12]. For the Large Hadron Collider in Europe, if all sensory data were recorded, the total amount of the sensory data is nearly 50 EB per day. Similarly, the traffic data, including GPS data, the monitoring data captured by electronic eyes, and so on, also increase rapidly and have already exceeded petabyte annually. Fortunately, not all sensory values are necessary for the users' analysis in most applications. Some critical data points, such as maxima, minima,

and flection points in a monitoring process, may be the only ones required by the users. Therefore, the algorithms for retrieving the critical data points from the monitored physical world are quite important for WSNs.

Currently, it supposes that the sensor nodes sense and sample the data from the monitored environment with equal sampling frequency in most of traditional applications, and the sensory data generated by WSNs is regarded as a set of discrete values. Under such assumptions, a great number of query processing techniques on discrete sensory data have been proposed, including the curve query processing algorithms [13], aggregation query processing algorithms [14–17], top-k query processing algorithms [18–20], skyline [21, 22], and quantilen [23, 24] query processing algorithms. Although all the above algorithms are efficient for processing the discrete sensory data, they cannot meet the complicated query requirements given by users and do not support to retrieve the critical data points in current WSNs since only discrete data were considered. For example, in the air pollution monitoring application, the users want to estimate the extremum

\*Correspondence: csy@hit.edu.cn

<sup>1</sup>School of Computer Science and Tech, Harbin Institute of Technology, 150001 Harbin, China

Full list of author information is available at the end of the article

pollution values and obtain the period when these values appear. In the climate monitoring system, the users may want to know the convexity of the wind velocity or rainfall curve and acquire the inflection points of such curves. The reasons the above query requirements cannot be satisfied by the existing query processing techniques, which only consider the discrete sensory data, are given as follows. First, as pointed out by [25, 26], the monitored physical world varies continuously, and the discrete sensory dataset omits many critical points, such as maxima, minima, and flection points from the physical world, so that the queries of retrieving critical points cannot be answered since these points do not belong to the discrete sensory dataset. Second, to answer the queries about the convexity, monotonicity, or the positions of critical points, the original data needs to have the continuous first and second derivatives, which cannot be met by the discrete sensory dataset either.

To overcome the above problems, Cheng et al. [26] proposed an adaptive data acquisition algorithm to reconstruct the physical world as much as possible. For any given error bound  $\epsilon$ , the authors proved that the result returned by their algorithm is  $O(\epsilon)$  approximate to the monitored physical world, where  $\epsilon$  can be arbitrarily small. Since the sampling frequency of each sensor is adjusted adaptively according to the variation of the monitored physical world, the algorithm proposed by [26] samples few sensory data and saves lots of energy on condition that the observation error is guaranteed. However, since the aim of [26] is to reconstruct the physical world, the amount of sensory data acquired by the sensors is still quite large. Considering that many applications may only want to acquire the critical point information, such as maxima, minima, and flection points, according to the above analysis, therefore, the energy cost of collecting the sensory data from the monitored physical world can be further reduced.

Due to the above reasons, we will study the problem of retrieving the critical points, including the extremum and inflection points, in the sensor networks. In this paper, a novel sensory data acquisition algorithm is proposed based on numerical analysis techniques [27] and Lagrange interpolation [28] in order to retrieve the critical points approximately. Such algorithm can adjust the sampling frequency of sensors adaptively according to the variation of physical world in order to dramatically reduce the amount of sensory data. Furthermore, to evaluate the error of approximate critical points, the formal definitions of  $\delta$ -approximate extremum point and  $\delta$ -approximate inflection point are firstly provided, where  $\delta$  denotes the relative error between the approximate critical point and the exact one. The correctness of the algorithm is proved. In summary, the contributions of this paper are as follows.

1. The formal definitions of  $\delta$ -approximate extremum point and  $\delta$ -approximate inflection point are firstly proposed. The problem of acquiring critical points from the monitored physical world is also defined.
2. Two critical point aware data acquisition algorithms are proposed based on numerical analysis [27] and Lagrange interpolation [28] techniques. The algorithms can adjust the sampling frequency of the sensors automatically according to variation of the physical world. The correctness of the algorithms is proved and the complexities of the algorithms are analyzed.
3. The extensive simulations on the real data set are carried out. The experimental results show that both of the precision and recall rate of our proposed algorithms are quite high to retrieve  $\delta$ -approximate extremum point and  $\delta$ -approximate inflection point from the monitored physical world.

The organization of the paper is as follows. Section 2 gives the problem definition. Section 3 provides the mathematical foundations of the algorithms. Section 4 proposes two critical points aware data acquisition algorithms, to retrieve the  $\delta$ -approximate extremum point and  $\delta$ -approximate inflection point, respectively. Section 5 shows the experimental results. Section 6 discusses the related work of the paper. Finally, section 7 concludes the whole paper.

## 2 Problem definition

Let  $N$  denote the number of sensor nodes in a given WSN, and  $V = \{1, 2, \dots, N\}$  be the set of sensor nodes, where  $i$  ( $1 \leq i \leq N$ ) denotes the ID of a sensor node.

Suppose that  $t_s$  and  $t_f$  denote the start and final time in monitoring the physical world by a WSN, respectively. Therefore, the variation of the physical world monitored by sensor node  $i$  can be regarded as a curve. We use  $S_i$  ( $1 \leq i \leq N$ ) to denote such curve. According to the discussion in [26], the physical world always varies continuously and  $S_i$  is smooth enough to have a continuous fourth-order derivative, i.e.,  $S_i \in C^4[t_s, t_f]$ , where  $C^4[t_s, t_f]$  denotes the set of functions whose fourth-order derivative is continuous in  $[t_s, t_f]$ .

In this paper, the critical points considered by us are extremum points and inflection points of the physical curve  $S_i$  ( $1 \leq i \leq N$ ). Since  $S_i$  has the continuous fourth-order derivative according to the above analysis, the extremum points of  $S_i$  in range  $[t_s, t_f]$  can be denoted by  $\{x | S_i^{(1)}(x) = 0 \wedge x \in [t_s, t_f]\}$ ; similarly, the inflection points of  $S_i$  in  $[t_s, t_f]$  can be denoted by  $\{x | S_i^{(2)}(x) = 0 \wedge x \in [t_s, t_f]\}$ , where  $S_i^{(1)}$  and  $S_i^{(2)}$  are the first and second derivatives of  $S_i$ , respectively.

Since when the critical points will appear in the future is unknown, it requires that the sampling frequency should be infinite to obtain all the critical points of  $S_i$  exactly, which is almost impossible. Thus, we will study the sensory data acquisition algorithm to retrieve the critical points, including the extremum points and the inflection points, approximately. To evaluate the relative error between the approximate critical points and the exact critical points, the  $\delta$ -approximate extremum point and  $\delta$ -approximate inflection point are defined as follows.

**Definition 1. ( $\delta$ -approximate extremum points)**  $\hat{x}_i$  from  $\widehat{S}_i$  is called as a  $\delta$ -approximate extremum point if and only if  $\exists x_i \in [t_s, t_f]$  satisfying  $\frac{|\hat{x}_i - x_i|}{x_i} \leq \delta$  and  $S_i^{(1)}(x_i) = 0$ .

**Definition 2. ( $\delta$ -approximate inflection points)**  $\hat{x}_i$  is called as a  $\delta$ -approximate inflection point if and only if  $\exists x_i \in [t_s, t_f]$  satisfying that  $S_i^{(2)}(x_i) = 0$  and  $\frac{|\hat{x}_i - x_i|}{x_i} \leq \delta$ .

The intuition of our algorithm is to forecast the first and second derivatives of  $S_i$  using the current collected sensory data. If the first or second derivative is close to 0, then the sampling frequency increases. Otherwise, we reduce the sampling frequency in order to save energy. Since the physical world varies continuously, it is acceptable to use the history sensory data to forecast the future. The detailed algorithm is presented in Section 4, and the formal definition of retrieving the approximate critical points from physical world is given as follows.

**Input:**

1. The start and final time of monitoring,  $t_0 (= t_s)$  and  $t_f$
2. The maximum bound of the sampling frequency of a sensor node,  $f_{\max}$
3. The step-size increment  $t'$  and the decrease factor  $\alpha$

**Output:** The sets of approximate extremum and inflection points in  $[t_s, t_f]$ ,  $X_1$  and  $X_2$

We verify that the precision and the recall rate of our algorithms are quite high on condition that  $\delta$ -approximate extremum points and  $\delta$ -approximate inflection points are collected. To improve the readability of the whole paper, the table of the symbols used in this paper is given in Table 1.

**3 Mathematical foundations**

Let  $t_c$  denote the current sampling time,  $t_{c-1}$  denote the last sampling time before  $t_c$ , and  $t_{\frac{2c-1}{2}}$  be the median of  $t_c$  and  $t_{c-1}$ . For each sensor node  $i$  ( $1 \leq i \leq N$ ),  $S_i(t_{\frac{2c-1}{2}})$  should always be sampled besides  $S_i(t_{c-1})$  and

**Table 1** The meanings of symbols

Symbols	Meanings
$\delta$	The relative error of the approximate critical point
$N$	The number of sensor nodes in a given WSN
$V = \{1, 2, \dots, N\}$	The set of sensors
$i$	The ID of a sensor node
$S_i$	The real physical world curve at node $i$ 's location
$S_i(t)$	The sensed value of sensor node $i$ at time $t$
$S_i^{(k)}(t)$	The $k$ -order derivative of $S_i$ at time $t$ .
$\widehat{S}_i^{(k)}(t)$	The estimation of the $k$ -order derivative of $S_i$ at time $t$ .
$X_1$	The extremum point set returned by our algorithm.
$X_2$	The inflection point set returned by our algorithm.
$t_s$	The start time of an observation period
$t_f$	The final time of an observation period
$t_c$	The current data sampling time slot
$f_{\max}$	The maximum sampling frequency
$t'$	The step-size increment
$\alpha$	The decrease factor

$S_i(t_c)$  in  $[t_{c-1}, t_c]$ . Based on such operations and a three-point central difference formula [27], the first and second derivatives of  $S_i$  at  $t_{\frac{2c-1}{2}}$  can be estimated by the following two formulas.

$$\widehat{S}_i^{(1)}\left(t_{\frac{2c-1}{2}}\right) = \frac{S_i(t_c) - S_i(t_{c-1})}{2h} - \frac{h^2}{6} S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right) \tag{1}$$

$$\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right) = \frac{S_i(t_c) - 2S_i(t_{\frac{2c-1}{2}}) + S_i(t_{c-1})}{h^2} - \frac{h^2}{12} S_i^{(4)}\left(t_{\frac{2c-1}{2}}\right) \tag{2}$$

where  $h = t_{\frac{2c-1}{2}} - t_{c-1} = t_c - t_{\frac{2c-1}{2}}$  denotes the half length of the sampling interval.

The following theorem guarantees that the difference between  $\widehat{S}_i^{(1)}\left(t_{\frac{2c-1}{2}}\right)$  and  $S_i^{(1)}\left(t_{\frac{2c-1}{2}}\right)$  is determined by  $h^2$ .

**Theorem 1.**  $\left| S_i^{(1)}\left(t_{\frac{2c-1}{2}}\right) - \widehat{S}_i^{(1)}\left(t_{\frac{2c-1}{2}}\right) \right| = \left| \frac{h^2}{6} \left\{ S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right) - S_i^{(3)}(\xi) \right\} \right|$ , where  $\xi \in [t_{c-1}, t_c]$ .

**Proof.** Based on Taylor Series, we have

$$S_i(t_c) = S_i\left(t_{\frac{2c-1}{2}}\right) + h S_i^{(1)}\left(t_{\frac{2c-1}{2}}\right) + \frac{h^2}{2} S_i^{(2)}\left(t_{\frac{2c-1}{2}}\right) + \frac{h^3}{3!} S_i^{(3)}(\xi_1)$$

Similarly

$$S_i(t_{c-1}) = S_i\left(t_{\frac{2c-1}{2}}\right) - hS_i^{(1)}\left(t_{\frac{2c-1}{2}}\right) + \frac{h^2}{2}S_i^{(2)}\left(t_{\frac{2c-1}{2}}\right) - \frac{h^3}{3!}S_i^{(3)}(\xi_2)$$

where  $\xi_1 \in \left[t_{\frac{2c-1}{2}}, t_c\right]$  and  $\xi_2 \in \left[t_{c-1}, t_{\frac{2c-1}{2}}\right]$ . Therefore,

$$S_i(t_c) - S_i(t_{c-1}) = 2hS_i^{(1)}\left(t_{\frac{2c-1}{2}}\right) + \frac{h^3}{3!}\left(S_i^{(3)}(\xi_1) + S_i^{(3)}(\xi_2)\right) \quad (3)$$

Let  $M$  and  $m$  be the maximum and minimum values of  $S_i^{(3)}(t)$  for any  $t \in [t_{c-1}, t_c]$ . Since  $\xi_1, \xi_2 \in [t_{c-1}, t_c]$ ,  $m \leq \frac{S_i^{(3)}(\xi_1) + S_i^{(3)}(\xi_2)}{2} \leq M$ . Furthermore, since  $S_i$  has the continuous third-order derivative,  $\exists \xi \in [t_{c-1}, t_c]$  satisfies that  $S_i^{(3)}(\xi) = \frac{S_i^{(3)}(\xi_1) + S_i^{(3)}(\xi_2)}{2}$ . Therefore, according to Formula (3), we have

$$S_i^{(1)}\left(t_{\frac{2c-1}{2}}\right) = \frac{S_i(t_c) - S_i(t_{c-1})}{2h} - \frac{h^2}{6}S_i^{(3)}(\xi) \quad (4)$$

Comparing Formula (3) with Formula (1), we have

$$\left|S_i^{(1)}\left(t_{\frac{2c-1}{2}}\right) - \widehat{S}_i^{(1)}\left(t_{\frac{2c-1}{2}}\right)\right| = \left|\frac{h^2}{6}\left\{S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right) - S_i^{(3)}(\xi)\right\}\right|.$$

□

**Theorem 2.**  $\left|S_i^{(2)}\left(t_{\frac{2c-1}{2}}\right) - \widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right)\right| = \left|\frac{h^2}{12}\left(S_i^{(4)}\left(t_{\frac{2c-1}{2}}\right) - S_i^{(4)}(\xi)\right)\right|$ , where  $\xi \in [t_{c-1}, t_c]$ . □

The proof of Theorem 2 is similar as that of Theorem 1. These two theorems indicate that the difference between the exact critical points and the approximate ones calculated by Formulas (1) and (2) can be arbitrary small with the decline of  $h$ .

In Theorems 1 and 2,  $S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right)$  and  $S_i^{(4)}\left(t_{\frac{2c-1}{2}}\right)$  can be estimated by Lagrange interpolation. The sensory values collected by sensor node  $i$  ( $1 \leq i \leq N$ ) in the last and current sampling intervals, i.e.,  $S_i(t_{c-2})$ ,  $S_i\left(t_{\frac{2c-3}{2}}\right)$ ,  $S_i(t_{c-1})$ ,  $S_i\left(t_{\frac{2c-1}{2}}\right)$  and  $S_i(t_c)$ , are used.

Let  $L_1(t)$  and  $L_2(t)$  denote the Lagrange interpolation polynomials to estimate  $S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right)$  and  $S_i^{(4)}\left(t_{\frac{2c-1}{2}}\right)$ , respectively. The construction of  $L_1(t)$  and  $L_2(t)$  are shown in the Appendix. Therefore,  $S_i^{(3)}(t)$  can be estimated by the following formula, where  $t \in \left[t_{\frac{2c-3}{2}}, t_c\right]$ .

$$S_i^{(3)}(t) \approx L_1^{(3)}(t) = 6 \left[ \frac{S_i\left(t_{\frac{2c-3}{2}}\right)}{\left(t_{\frac{2c-3}{2}} - t_{c-1}\right)\left(t_{\frac{2c-3}{2}} - t_{\frac{2c-1}{2}}\right)\left(t_{\frac{2c-3}{2}} - t_c\right)} + \frac{S_i(t_{c-1})}{\left(t_{c-1} - t_{\frac{2c-3}{2}}\right)\left(t_{c-1} - t_{\frac{2c-1}{2}}\right)\left(t_{c-1} - t_c\right)} + \frac{S_i\left(t_{\frac{2c-1}{2}}\right)}{\left(t_{\frac{2c-1}{2}} - t_{\frac{2c-3}{2}}\right)\left(t_{\frac{2c-1}{2}} - t_{c-1}\right)\left(t_{\frac{2c-1}{2}} - t_c\right)} + \frac{S_i(t_c)}{\left(t_c - t_{\frac{2c-3}{2}}\right)\left(t_c - t_{c-1}\right)\left(t_c - t_{\frac{2c-1}{2}}\right)} \right] \quad (5)$$

Besides,  $S_i^{(4)}(t)$  can be estimated by the following formula, where  $t \in [t_{c-2}, t_c]$ .

$$S_i^{(4)}(t) \approx L_2^{(4)}(t) = 24 \left[ \frac{S_i(t_{c-2})}{\left(t_{c-2} - t_{\frac{2c-3}{2}}\right)\left(t_{c-2} - t_{c-1}\right)\left(t_{c-2} - t_{\frac{2c-1}{2}}\right)\left(t_{c-2} - t_c\right)} + \frac{S_i\left(t_{\frac{2c-3}{2}}\right)}{\left(t_{\frac{2c-3}{2}} - t_{c-2}\right)\left(t_{\frac{2c-3}{2}} - t_{c-1}\right)\left(t_{\frac{2c-3}{2}} - t_{\frac{2c-1}{2}}\right)\left(t_{\frac{2c-3}{2}} - t_c\right)} + \frac{S_i(t_{c-1})}{\left(t_{c-1} - t_{c-2}\right)\left(t_{c-1} - t_{\frac{2c-3}{2}}\right)\left(t_{c-1} - t_{\frac{2c-1}{2}}\right)\left(t_{c-1} - t_c\right)} + \frac{S_i\left(t_{\frac{2c-1}{2}}\right)}{\left(t_{\frac{2c-1}{2}} - t_{c-2}\right)\left(t_{\frac{2c-1}{2}} - t_{\frac{2c-3}{2}}\right)\left(t_{\frac{2c-1}{2}} - t_{c-1}\right)\left(t_{\frac{2c-1}{2}} - t_c\right)} + \frac{S_i(t_c)}{\left(t_c - t_{c-2}\right)\left(t_c - t_{\frac{2c-3}{2}}\right)\left(t_c - t_{c-1}\right)\left(t_c - t_{\frac{2c-1}{2}}\right)} \right] \quad (6)$$

The error of such estimation is bounded by Theorem 3, which is very small in practice.

**Theorem 3.** The errors generated by Formulas (5) and (6) equal to  $\gamma_1(t) = \frac{S_i^{(4)}(\xi)}{4} \left[4t - \left(t_{\frac{2c-3}{2}} + t_{c-1} + t_{\frac{2c-1}{2}} + t_c\right)\right]$  and  $\gamma_2(t) = \frac{S_i^{(5)}(\xi)}{5} \left[5t - \left(t_{c-2} + t_{\frac{2c-3}{2}} + t_{c-1} + t_{\frac{2c-1}{2}} + t_c\right)\right]$ , respectively.

**Proof.** According to the property of Lagrange interpolation [28], the interpolation remainder, denoted by  $R(x)$  satisfies that

$$R(x) = f(x) - L_f(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega(x) \quad (7)$$

where  $f(x)$  is a function whose  $n$ th-order derivative is continuous, and  $L_f(x)$  is  $n$ -degree Lagrange interpolation of  $f(x)$ ,  $x_0, x_1, \dots, x_n$  denote the interpolation points, and  $\xi \in [x_0, x_n]$ .

Therefore, the three-order interpolation remainder of  $L_1(t)$  satisfies that

$$\begin{aligned} \gamma_1(t) &= R^{(3)}(t) = \frac{S_i^{(4)}(\xi)}{4!} \omega_1^{(3)}(t) \\ &= \frac{S_i^{(4)}(\xi)}{4} \left[ 4t - \left( t_{\frac{2c-3}{2}} + t_{c-1} + t_{\frac{2c-1}{2}} + t_c \right) \right] \end{aligned} \quad (8)$$

which is the error generated during estimating  $S_i^{(3)}(t)$ .

The fourth-order interpolation remainder of  $L_2(t)$  satisfies that

$$\begin{aligned} \gamma_2(t) &= R^{(4)}(t) = \frac{S_i^{(5)}(\xi)}{5!} \omega_2^{(4)}(t) \\ &= \frac{S_i^{(5)}(\xi)}{5} \left[ 5t - \left( t_{c-2} + t_{\frac{2c-3}{2}} + t_{c-1} + t_{\frac{2c-1}{2}} + t_c \right) \right] \end{aligned} \quad (9)$$

which is the error generated during estimating  $S_i^{(4)}(t)$ . Thus, Theorem 3 is proved.  $\square$

Theorem 3 also verifies that the error generated by Lagrange interpolation estimation is also very small.

#### 4 Critical point aware data acquisition algorithm

According to the analysis in Section 3,  $t_c$  and  $t_{c-1}$  denote the current sampling time and the last one before  $t_c$ , respectively.  $t_{\frac{2c-1}{2}}$  is the median time slot of  $t_c$  and  $t_{c-1}$ . Let  $f_{\max}$  be the maximum sampling frequency that a sensor node can achieve.

Based on such symbols, the whole critical point aware data acquisition algorithm can be divided into two phases. The first one is the initial phase, the sampling frequency in such phase is set to be  $f_{\max}$  in order not to omit any critical points. The second one is the maintenance phase; the sampling frequency in such phase is determined according to the variation of physical world. Since the variation of the monitored physical world in the future is unknown, the posterior estimation is adopted, that is, the history sensory data collected in the current time will be used to estimate the variation of the monitored physical world in the future. Because the physical world always varies continuously, such estimation is acceptable and can achieve high precision.

Specifically, the critical point aware data acquisition algorithm consists of the following five steps.

**Step 1.** Sample sensory values at time  $t_0, t_{\frac{1}{2}}, t_1$ , where  $t_{\frac{1}{2}} = t_0 + \frac{1}{f_{\max}}$  and  $t_1 = t_{\frac{1}{2}} + \frac{1}{f_{\max}}$ . That is, we

initialize  $h$  with the minimum sampling interval  $\frac{1}{f_{\max}}$ . Initialize  $c$  with 2, then execute a loop until  $t_c > t_f$ .

**Step 2.** Sensor node  $i$  ( $1 \leq i \leq N$ ) samples the sensory values at time  $t_{\frac{2c-1}{2}}$  and  $t_c$ , where

$t_{\frac{2c-1}{2}} = t_{c-1} + h$  and  $t_c = t_{\frac{2c-1}{2}} + h$ . Using the sensory values sampled in the current and last sampling intervals, i.e.,  $S_i(t_{c-2}), S_i(t_{\frac{2c-3}{2}}), S_i(t_{c-1}), S_i(t_{\frac{2c-1}{2}})$  and  $S_i(t_c)$ , the Lagrange interpolation polynomial can be constructed. Therefore,  $S_i^{(3)}(t)$  and  $S_i^{(4)}(t)$  can be obtained according to Formulas (5) and (6) for  $t \in [t_{\frac{2c-3}{2}}, t_c]$  and  $t \in [t_{c-2}, t - c]$  respectively.

**Step 3.** Call the extremum point retrieving algorithm to obtain the extremum point in current sampling interval and determine the length of the next possible sampling interval  $h_1$ , the detailed method is shown in Section 4.1.

**Step 4.** Similarly, call the inflection points retrieving algorithm to collect the inflection point and determine the length of the next possible sampling interval  $h_2$ . The detailed algorithm is given in Section 4.2.

**Step 5.** Finally, select the minimum one returned by the above two steps to be the adopted length of the next sampling interval, i.e.,  $h = \min\{h_1, h_2\}$ , which avoids omitting critical points. Go to step 2 with increasing  $c$  by 1, and start a new loop until  $t_c > t_f$ .

The detailed critical point aware data acquisition algorithm is shown in Algorithm 1.

#### 4.1 Extremum point retrieving algorithm

The extremum point retrieving algorithm has four steps.

**Step 1.** Sensor  $i$  ( $1 \leq i \leq N$ ) estimates the the first derivative  $\widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}})$  based on Formula (1).

Meanwhile,  $\widehat{S}_i^{(2)}(t_{\frac{2c-1}{2}})$  should be computed according to Formula (2).

**Step 2.** Find extremum point in  $[t_{\frac{2c-3}{2}}, t_{\frac{2c-1}{2}}]$ . If  $\widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}}) = 0$ , return the extremum point  $t_{\frac{2c-1}{2}}$ .

Otherwise, compare  $\widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}})$  and  $\widehat{S}_i^{(1)}(t_{\frac{2c-3}{2}})$  calculated in last loop. If  $\widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}}) \times \widehat{S}_i^{(1)}(t_{\frac{2c-3}{2}}) < 0$ , there must be an extremum point in  $[t_{\frac{2c-3}{2}}, t_{\frac{2c-1}{2}}]$ . Then, retrieve the extremum point by curve tessellation techniques [29].

**Step 3.** There exists three cases that need to be considered when determining the length of the next sampling interval.

**Algorithm 1:** Critical point aware data acquisition algorithm

---

**Input:**  $t_0 (= t_s), f_{\max}, t',$  the decrease factor  $\alpha$ .  
**Output:** The set of extremum points  $X_1$  and the set of inflection points  $X_2$

---

- 1  $c \leftarrow 2, h \leftarrow \frac{1}{f_{\max}}$
- 2 Sample sensory values at  $\{t_{j/2} | 0 \leq j \leq 2 \wedge t_{j/2} - t_{(j-1)/2} = 1/f_{\max}\};$
- 3 **while**  $t_c < t_f$  **do**
- 4   Sample sensory values at  $t_{\frac{2c-1}{2}}, t_c;$
- 5   Construct third order Lagrange interpolation and estimate  $S_i^{(3)}(t)$  ( $t \in [t_{\frac{2c-3}{2}}, t_c]$ ) according to Formula (5);
- 6   Construct fourth order Lagrange interpolation and estimate  $S_i^{(4)}(t)$  ( $t \in [t_{c-2}, t_c]$ ) according to Formula (6);
- 7    $(y_1, h_1) =$  **extremum point retrieving**  
 $(S_i(t_{c-1}), S_i(t_{\frac{2c-1}{2}}), S_i(t_c), h, S_i^{(3)}(t), S_i^{(4)}(t));$
- 8    $(y_2, h_2) =$  **inflection point retrieving**  
 $(S_i(t_{c-1}), S_i(t_{\frac{2c-1}{2}}), S_i(t_c), h, S_i^{(3)}(t), S_i^{(4)}(t));$
- 9    $X_1 = X_1 \cup y_1, X_2 = X_2 \cup y_2, h = \min\{h_1, h_2\};$
- 10    $t_{\frac{2c+1}{2}} = t_c + h, t_{c+1} = t_c + 2h, c \leftarrow c + 1;$
- 11 Return  $X_1$  and  $X_2$

---

1. If  $\widehat{S}_i^{(2)}(t_{\frac{2c-1}{2}}) \times \widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}}) > 0$ , which means that the first derivative of the physical world increases or decreases monotonously; therefore, the extremum point may not be contained in the next sampling interval  $[t_c, t_{c+1}]$ , so that the length of the next sampling interval should be increased in order to save energy. In summary, let  $h$  and  $h_1$  denote the half length of the current and next sampling interval, then  $h_1 = h + t'$ , where  $t'$  is a given constant to denote the step size increment.
2. If  $\widehat{S}_i^{(2)}(t_{\frac{2c-1}{2}}) \times \widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}}) < 0$  and  $|\widehat{S}_i^{(2)}(t_{\frac{2c-1}{2}})| \times 2h > |\widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}})|$ , which means that the next extremum point is more likely to be included in the next sampling interval; thus, we decrease the length of sampling interval in order to catch the extremum point. Therefore,  $h_1$  is set to be  $h_1 = \max\{\frac{1}{f_{\max}}, \alpha h\}$ , where  $\alpha$  is a decreasing factor and  $0 < \alpha \leq 1$ .
3. Otherwise, the ratio between  $|\widehat{S}_i^{(2)}(t_{\frac{2c-1}{2}})|$  and  $|\widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}})|$  will be considered, if  $|\widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}})| \geq (2h + t') \times |\widehat{S}_i^{(2)}(t_{\frac{2c-1}{2}})|$ ,  $h_1 = h + t'$ .

In other situations,  $h_1$  maintains the last sampling interval  $h$ .

**Step 4.** Return  $h_1$  and the extremum point.

The detailed algorithm is shown in Algorithm 2.

**Algorithm 2:** Extremum point retrieving algorithm

---

**Input:**  $f_{\max}, t', \alpha, S_i(t_c), S_i(t_{\frac{2c-1}{2}}), S_i(t_{c-1}), h,$   
 $S_i^{(3)}(t_{\frac{2c-1}{2}}), S_i^{(4)}(t_{\frac{2c-1}{2}})$

**Output:**  $h_1, y_1$

- 1  $\widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}}) = \frac{S_i(t_c) - S_i(t_{c-1})}{2h} - \frac{h^2}{6} S_i^{(3)}(t_{\frac{2c-1}{2}});$
- 2  $\widehat{S}_i^{(2)}(t_{\frac{2c-1}{2}}) = \frac{S_i(t_c) - 2S_i(t_{\frac{2c-1}{2}}) + S_i(t_{c-1})}{h^2} - \frac{h^2}{12} S_i^{(4)}(t_{\frac{2c-1}{2}});$
- 3  $y_1 = \emptyset;$
- 4  $h_1 = h;$
- 5 **if**  $\widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}}) \times \widehat{S}_i^{(1)}(t_{\frac{2c-3}{2}}) < 0$  **then**
- 6    $y_1$  is determined by curve tessellation techniques  
 $[29]$  in  $[t_{\frac{2c-3}{2}}, t_{\frac{2c-1}{2}}];$
- 7 **if**  $\widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}}) = 0$  **then**
- 8    $y_1 = \{t_{\frac{2c-1}{2}}\};$
- 9 **if**  $\widehat{S}_i^{(2)}(t_{\frac{2c-1}{2}}) \times \widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}}) > 0$  **then**
- 10    $h_1 = h + t';$
- 11 **else**
- 12   **if**  $|\widehat{S}_i^{(2)}(t_{\frac{2c-1}{2}})| \times 2h > |\widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}})|$  **then**
- 13     $h_1 = \max\{\frac{1}{f_{\max}}, \alpha h\};$
- 14   **else**
- 15    **if**  $|\widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}})| \geq (2h + t') \times |\widehat{S}_i^{(2)}(t_{\frac{2c-1}{2}})|$  **then**
- 16     $h_1 = h + t';$
- 17 Return  $h_1$  and  $y_1;$

---

**4.2 Inflection point retrieving algorithm**

The inflection point retrieving algorithm also has four steps and can be constructed by similar method shown in above section.

**Step 1.** Sensor  $i$  ( $1 \leq i \leq N$ ) estimates the the first derivative  $\widehat{S}_i^{(1)}(t_{\frac{2c-1}{2}})$  and second derivative

$\widehat{S}_i^{(2)}(t_{\frac{2c-1}{2}})$  based on Formula (1) and Formula (2).

**Step 2.** Find inflection point in  $[t_{\frac{2c-3}{2}}, t_{\frac{2c-1}{2}}]$ . If

$\widehat{S}_i^{(2)}(t_{\frac{2c-1}{2}}) = 0$ , inflection point is  $t_{\frac{2c-1}{2}}$  and return

it. Otherwise, compare  $\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right)$  and  $\widehat{S}_i^{(2)}\left(t_{\frac{2c-3}{2}}\right)$  calculated in last loop. If  $\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right) \times \widehat{S}_i^{(2)}\left(t_{\frac{2c-3}{2}}\right) < 0$ , there must be an inflection point in  $\left[t_{\frac{2c-3}{2}}, t_{\frac{2c-1}{2}}\right]$  we have missed. Then, retrieve the inflection point by curve tessellation techniques [29].

**Step 3.** There exists three cases that need to be considered when determining the length of the next sampling interval.

1. If  $S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right) \times \widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right) > 0$ , which means that the second derivative of the physical world increases or decreases monotonously; therefore, the inflection point may not exist in the next sampling interval  $[t_c, t_{c+1}]$ . In order to save energy, increasing the length of the next sampling interval. That is, increase the half length of sampling interval  $h_2 = h + t'$ , where  $t'$  is a given constant to denote the step size increment.
2. If  $S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right) \times \widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right) < 0$  and  $\left|S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right)\right| \times 2h > \left|\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right)\right|$ , which means that the next inflection point is more likely to be included in the next sampling interval; thus, we decrease the length of sampling interval in order to catch the inflection point. Therefore,  $h_2$  is set to be  $h_2 = \max\left\{\frac{1}{f_{\max}}, \alpha h\right\}$ , where  $\alpha$  is a decreasing factor and  $0 < \alpha \leq 1$ .
3. Otherwise, the ratio between  $\left|S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right)\right|$  and  $\left|\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right)\right|$  will be considered, if  $\left|\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right)\right| \geq (2h + t') \times \left|S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right)\right|$ , increase  $h_2$  with  $t'$ ,  $h_2 = h + t'$ .

In other situations,  $h_2$  maintains the last sampling interval  $h$ .

**Step 4.** Return  $h_2$  and the inflection point.

The detailed algorithm is shown in Algorithm 3.

In each sampling interval, the complexity of the above algorithm is  $O(1)$  since it only needs to sample two sensory values and the first and second derivatives can be calculated in  $O(1)$ . Therefore, the total complexity of Algorithm 1 is determined by the number of sampling interval it includes.

In the best case, the half length of sampling interval,  $h$ , increases  $t'$  every loop. Therefore, there exists  $O\left(\sqrt{\frac{t_f - t_s}{t'}}\right)$  sampling intervals, so that the minimum complexity of the above algorithm is  $O\left(\sqrt{\frac{t_f - t_s}{t'}}\right)$ .

---

**Algorithm 3:** Inflection point retrieving algorithm

---

**Input:**  $f_{\max}, t', \alpha, S_i(t_c), S_i\left(t_{\frac{2c-1}{2}}\right), S_i(t_{c-1}), h,$   
 $S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right), S_i^{(4)}\left(t_{\frac{2c-1}{2}}\right)$

**Output:**  $h_2, y_2$

```

1  $\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right) = \frac{S_i(t_c) - 2S_i\left(t_{\frac{2c-1}{2}}\right) + S_i(t_{c-1})}{h^2} - \frac{h^2}{12} S_i^{(4)}\left(t_{\frac{2c-1}{2}}\right);$ 
2  $y_2 = \emptyset;$ 
3  $h_2 = h;$ 
4 if  $\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right) \times \widehat{S}_i^{(2)}\left(t_{\frac{2c-3}{2}}\right) < 0$  then
5    $y_2$  is determined by curve tessellation techniques
   [29] in  $\left[t_{\frac{2c-3}{2}}, t_{\frac{2c-1}{2}}\right];$ 
6 if  $\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right) == 0$  then
7    $y_2 = \left\{t_{\frac{2c-1}{2}}\right\};$ 
8 if  $S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right) \times \widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right) > 0$  then
9    $h_2 = h + t';$ 
10 else
11   if  $\left|S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right)\right| \times 2h > \left|\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right)\right|$  then
12      $h_2 = \max\left\{\frac{1}{f_{\max}}, \alpha h\right\};$ 
13   else
14     if  $\left|\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right)\right| \geq (2h + t') \times \left|S_i^{(3)}\left(t_{\frac{2c-1}{2}}\right)\right|$  then
15        $h_2 = h + t';$ 
16 Return  $h_2$  and  $y_2;$ 

```

---

In the worst case, the sampling frequency is always  $f_{\max}$ , so that the maximum complexity of the above algorithm is  $O(f_{\max}(t_f - t_s))$ . However, the worst case requires that there should exist lots of critical points in  $[t_s, t_f]$ , which is rarely happened. Therefore, in practice, the complexity of our algorithm is much better than the worst case.

**4.3 Discussion: the method for estimating missing critical points**

If the product of the first derivatives of  $t_{\frac{2c-3}{2}}$  and  $t_{\frac{2c-1}{2}}$  is less than 0, i.e.,  $\widehat{S}_i^{(1)}\left(t_{\frac{2c-1}{2}}\right) \times \widehat{S}_i^{(1)}\left(t_{\frac{2c-3}{2}}\right) < 0$ , then there exists an extremum point in range  $\left(t_{\frac{2c-3}{2}}, t_{\frac{2c-1}{2}}\right)$  which is not captured by our extremum point retrieving algorithm. Since such missing extremum points are also very important for users' observation, the method for estimating these missing critical points are required. The traditional methods, such as curve tessellation techniques [29], are too complicate to solve such problem, and not practical for WSNs. However, in some situations, the missing critical points can be obtained easily just as shown in the following Theorem 4.

**Theorem 4.** When  $\widehat{S}_i^{(1)}\left(t_{\frac{2c-1}{2}}\right) \times \widehat{S}_i^{(1)}\left(t_{\frac{2c-3}{2}}\right) < 0$  or  $\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right) \times \widehat{S}_i^{(2)}\left(t_{\frac{2c-3}{2}}\right) < 0$ ,  $t_{c-1}$  is a  $\delta$ -approximate critical point of  $S_i$ , if  $\frac{\max\left\{\left|t_{c-1}-t_{\frac{2c-3}{2}}\right|,\left|t_{c-1}-t_{\frac{2c-1}{2}}\right|\right\}}{t_{\frac{2c-3}{2}}} \leq \delta$ .

**Proof.** When  $\widehat{S}_i^{(1)}\left(t_{\frac{2c-1}{2}}\right) \times \widehat{S}_i^{(1)}\left(t_{\frac{2c-3}{2}}\right) < 0$  or  $\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right) \times \widehat{S}_i^{(2)}\left(t_{\frac{2c-3}{2}}\right) < 0$ , there must exist a critical point in  $\left[t_{\frac{2c-3}{2}}, t_{\frac{2c-1}{2}}\right]$ , since  $S_i$  has continuous fourth-order derivative. We denote the critical point as  $t$ ,  $t \in \left[t_{\frac{2c-3}{2}}, t_{\frac{2c-1}{2}}\right]$ . According to the definitions of  $\delta$ -approximate extremum and inflection points,  $t_{c-1}$  is a  $\delta$ -approximate critical point if and only if  $\frac{|t_{c-1}-t|}{t} \leq \delta$ .

Since  $t \in \left[t_{\frac{2c-3}{2}}, t_{\frac{2c-1}{2}}\right]$ , therefore,  $|t_{c-1} - t| \leq \max\left\{\left|t_{c-1}-t_{\frac{2c-3}{2}}\right|,\left|t_{c-1}-t_{\frac{2c-1}{2}}\right|\right\}$ . As the condition shows,  $\frac{\max\left\{\left|t_{c-1}-t_{\frac{2c-3}{2}}\right|,\left|t_{c-1}-t_{\frac{2c-1}{2}}\right|\right\}}{t_{\frac{2c-3}{2}}} \leq \delta$ , then

$$\frac{|t_{c-1} - t|}{t_{\frac{2c-3}{2}}} \leq \frac{\max\left\{\left|t_{c-1}-t_{\frac{2c-3}{2}}\right|,\left|t_{c-1}-t_{\frac{2c-1}{2}}\right|\right\}}{t_{\frac{2c-3}{2}}} \leq \delta.$$

Besides, since  $t \geq t_{\frac{2c-3}{2}}$ , then  $\frac{|t_{c-1}-t|}{t} \leq \frac{|t_{c-1}-t|}{t_{\frac{2c-3}{2}}} \leq \delta$ .

That is,  $\frac{|t_{c-1}-t|}{t} \leq \delta$ . Therefore,  $t_{c-1}$  is a  $\delta$ -approximate critical point.  $\square$

According to Theorem 4, the missing extremum points can be obtained by following three steps.

**Step 1.** Judge whether  $t_{c-1}$  is a  $\delta$ -approximate extremum point by Theorem 4.

**Step 2.** Use the the values  $\widehat{S}_i^{(1)}\left(t_{\frac{2c-3}{2}}\right)$ ,  $\widehat{S}_i^{(2)}\left(t_{\frac{2c-3}{2}}\right)$ ,  $\widehat{S}_i^{(1)}\left(t_{\frac{2c-1}{2}}\right)$  and  $\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right)$  to locate the interval where the approximate extremum point is in.

**Step 3.** Use the Lagrange interpolation polynomial we constructed before to estimate the sampling value of the estimated extremum point.

The detailed algorithm for estimating the missing extremum point is shown in Algorithm 4. As shown in the algorithm, we use the linear function to estimate approximate extremum point if the condition in Theorem 4 is not satisfied, and more complicated situation will be considered in our future works. For the missing inflection points, the similarly method can be adopted to retrieve them approximately.

The complexity of Algorithm 4 is  $O(1)$ . The algorithm only needs to conduct compares and additions which can be calculated in  $O(1)$ . So the method for estimating these missing critical points is efficient.

---

**Algorithm 4:** The algorithm of estimating the missing extremum points

---

**Input:**  $t_{c-2}, t_{\frac{2c-3}{2}}, t_{c-1}, t_{\frac{2c-1}{2}}, t_c, \widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right), \widehat{S}_i^{(1)}\left(t_{\frac{2c-1}{2}}\right), \widehat{S}_i^{(2)}\left(t_{\frac{2c-3}{2}}\right), \widehat{S}_i^{(1)}\left(t_{\frac{2c-3}{2}}\right)$

**Output:**  $x, y$

```

1  if  $\frac{\max\left\{\left|t_{c-1}-t_{\frac{2c-3}{2}}\right|,\left|t_{c-1}-t_{\frac{2c-1}{2}}\right|\right\}}{t_{\frac{2c-3}{2}}} \leq \delta$  then
2  |    $y = t_{c-1}$ ;
3   $a = \left|\frac{\widehat{S}_i^{(2)}\left(t_{\frac{2c-1}{2}}\right)}{\widehat{S}_i^{(1)}\left(t_{\frac{2c-1}{2}}\right)}\right|$ ;  $b = \left|\frac{\widehat{S}_i^{(2)}\left(t_{\frac{2c-3}{2}}\right)}{\widehat{S}_i^{(1)}\left(t_{\frac{2c-3}{2}}\right)}\right|$ ;
4  if  $a > b$  then
5  |    $y = t_{\frac{2c-3}{2}} + \frac{1}{4}\left(t_{\frac{2c-1}{2}} - t_{\frac{2c-3}{2}}\right)$ ;
6  else
7  |   if  $a == b$  then
8  |   |    $y = \frac{1}{2}\left(t_{\frac{2c-3}{2}} + t_{\frac{2c-1}{2}}\right)$ ;
9  |   else
10 |   |    $y = t_{\frac{2c-3}{2}} + \frac{3}{4}\left(t_{\frac{2c-1}{2}} - t_{\frac{2c-3}{2}}\right)$ ;
11 Return  $y$ ;
```

---

#### 4.4 Discussion: the in-network cooperation and communication algorithms

The above algorithms are all about single sensor's sensing strategies, which can reduce sensory data. Obviously, when in-network cooperation and communication is considered, the sensory data can be further reduced. Since sensory data is spatially-related, not all the sensors need to transmit their critical points. That is, when two sensors are close enough, the critical points of their sensory data may appear at the same time. In this situation, not all the critical points need to be transmitted since they may present the same area. We design an algorithm for the in-network cooperation and communication, which can further reduce the transmitting sensory data and the communication energy.

Firstly, we divide the sensors into clusters according to their positions. We divide the monitored area into squares, whose length of the side is  $D$ . The length of the side  $D$  is given by the users according to the specified applications. Each cluster has a cluster head, and all the sensor nodes in the cluster transmit their critical points to the cluster head. All cluster heads compose a spanning tree, where the base station is the root of the spanning tree and cluster heads transmit the critical points along the spanning tree. Secondly, each time a sensor samples a critical point, it transmit the critical point and its position to the cluster head. The cluster head will determine whether the critical points it received need to be reduced

according to the positions. When the distance of two critical points is less than  $L$ , only one of the critical points will be transmitted, where  $L$  is a parameter given by the users. The detailed algorithm of in-network cooperation and communication is shown in the Algorithm 5.

This algorithm employs the spatial correlation of the sensors, which can further reduce the transmitting energy. Only simple in-network cooperation and communication is considered in the algorithm, since this paper mainly focuses on requiring critical points by single sensor. In the future work, we will study the in-network cooperation to further reduce the energy consumption.

---

**Algorithm 5:** The algorithm of in-network cooperation and communication

---

**Input:**  $D, L$

**Output:** Critical points set  $S$

```

1 Divide the monitored area into squares, whose length
  of the side is  $D$ ;
2 Each square as a cluster selects a cluster head;
3 Organize all cluster heads as a spanning tree rooted
  at sink;
4 for each sensor  $i$  in the network do
5   if  $i$  capture a critical point then
6     Transmit the critical point and its position to
     the cluster head it belongs;
7 for each cluster head  $h$  do
8   if  $h$  receives more than two critical points then
9      $S_h = \{a | a \text{ is the critical point } h \text{ received}\}$ ;
10    for each  $a$  in  $S_h$  do
11       $l_a = \text{the distance between } a \text{ and any}$ 
      other critical point  $b$  in  $S_h$ ;
12      if  $l_a > L$  then
13         $S_h = S_h - a$ ;
14     $h$  transmits  $S_h$  to its parent.
15 Sink combine all  $S_h$  as critical points set  $S$ .
16 Return  $S$ ;
```

---

## 5 Experiment result

### 5.1 Experiment setting

We use a simulated network with 200 sensor nodes to evaluate the performance of our sensory data acquisition algorithms. The network is deployed into a rectangular region with  $200 \times 200$  m size. The transmission range of each sensor node is set to be 25 m.

The sensory data of the network comes from a real sensor system, where the TelosB mote ([http://www.willow.co.uk/TelosB\\_Datasheet.pdf](http://www.willow.co.uk/TelosB_Datasheet.pdf)) is used to acquire indoor temperature, humidity, and light intensity continuously with the frequency equaling to 1 Hz, and the light intensity is adopted in the experiments.

### 5.2 The performance of the algorithm

The first group of experiments is going to evaluate the recall rate and precision of our algorithm for retrieving  $\delta$ -approximate extremum and inflection points. The recall rate equals to the fraction of the exact  $\delta$ -approximate critical points that are returned. The precision is the fraction of the returned results that are the exact  $\delta$ -approximate critical points. They are important parameters to evaluate the accuracy of the proposed algorithm. In the following experiments, the recall rate and the precision of our algorithm were computed, respectively, while  $\delta$  increases from 0.005 to 0.025,  $\alpha = 0.5$ , and  $t' = 0.5$ . The experimental results are presented in Figs. 1 and 2.

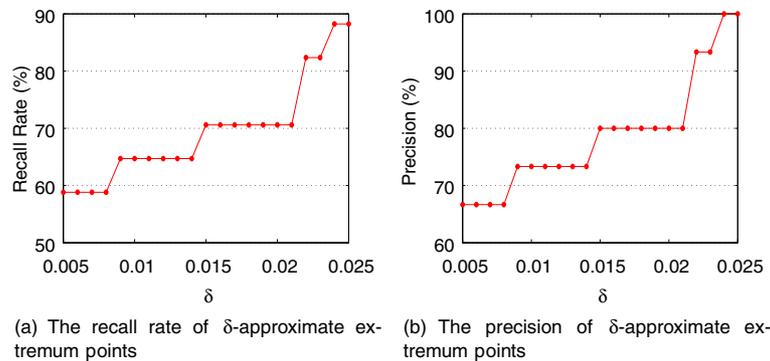
From Fig. 1a, b, we can see that the recall rate and precision of  $\delta$ -approximate extremum points increase with the growth of  $\delta$ , and both of them are close to a hundred percent even when  $\delta$  is quite small. For example, when  $\delta = 0.024$ , the precision of  $\delta$ -approximate extremum point is 100 % and the recall rate of  $\delta$ -approximate extremum point is close to 90 %. The results in Fig. 2a, b show that the recall rate and precision of  $\delta$ -approximate inflection points are also close to a hundred percent even when  $\delta$  is relatively small. The recall rate and precision of  $\delta$ -approximate critical points increase with the increase of  $\delta$ , since  $\delta$  is the relative error. When the error is loose, the algorithm can capture more approximate critical points which leads to higher recall rate and precision. Furthermore, when the relative error is not too large, the algorithm can capture almost all critical points.

In summary, our algorithms can achieve high accuracy in practice.

The second group of experiments is to investigate the impact of  $\alpha$  on the performance of our algorithm. In the following experiments, the recall rate, the precision, and the errors of the first and second derivatives generated by our algorithm are computed respectively while  $\alpha$  increases from 0 to 1,  $\delta = 0.015$ , and  $t' = 0.5$ . The experimental results are presented in Figs. 3 and 4.

Figure 3a, b shows the recall rate and precision of  $\delta$ -approximate extremum and inflection points. The recall rate and precision of  $\delta$ -approximate inflection points decrease with the growth of  $\alpha$ ; the reason is that the larger  $\alpha$  will omit more inflection points. Besides, the recall rate and precision of  $\delta$ -approximate inflection points are still every high even when  $\alpha$  is not very small. For example, the recall rate of  $\delta$ -approximate inflection points reaches 96 % when  $\alpha = 0.1$ . At the mean time, the recall and precision rate of  $\delta$ -approximate extremum points keep stable and high enough in practice with the variation of  $\alpha$ . Such recall rate and precision are quite high and can be acceptable in practice.

Figure 4a, b presents the errors of the first and second derivatives generated by our algorithm. The maximum, average and 0.9-quantile errors of the first and second



**Fig. 1** The relationship between the accuracies of  $\delta$ -approximate extremum points and  $\delta$

derivatives are quite small for different  $\alpha$ , which indicate the high accuracy of our algorithm and also explain why the recall rate and precision of our algorithm are quite high. For example, the average error of the estimated first derivative is less than  $10^{-3}$  and the average and 0.9-quantile errors of the estimated second derivative is about  $10^{-4}$ .

The third group of experiments is to investigate the impact of  $t'$  on the performance of our algorithm. In the experiments, the recall rate, the precision, and the errors of the first and second derivatives generated by our algorithm are calculated, while  $t'$  increases from 0.1 to 1,  $\alpha = 0.5$ , and  $\delta = 0.015$ . The experimental results are presented in Figs. 5 and 6.

Figure 5a, b shows that the recall rate of  $\delta$ -approximate inflection points is around 90% and the precision of  $\delta$ -approximate inflection points decreases with the growth of  $t'$ , since more  $\delta$ -approximate inflection points will be omitted when  $t'$  is larger. Although the recall rate and the precision of  $\delta$ -approximate extremum points are lower than  $\delta$ -approximate inflection points, we still capture a large portion of extremum points.

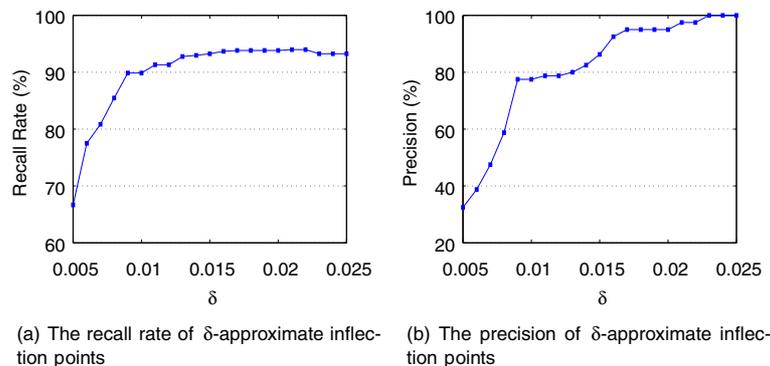
Figure 6a, b shows the maximum, average, and 0.9-quantile errors of the first and second derivatives

generated by our algorithm while  $t'$  increases. The figures show that all these errors are extremely small, which also verify that our algorithm can achieve high precision on identifying the critical points. For example, the maximum error of the estimated first derivative is only about less than  $2 \times 10^{-3}$ . For the estimated second derivative, most of the errors is less than  $0.5 \times 10^{-4}$ .

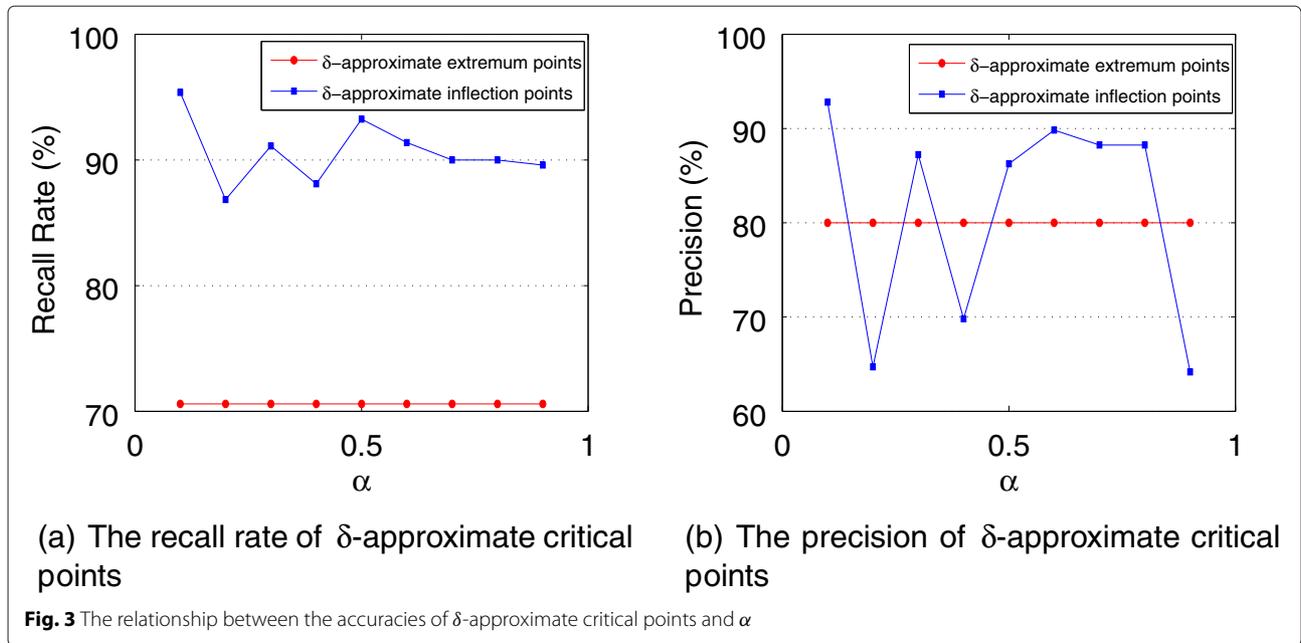
### 6 Related works

Currently, there exists few published works considering the adaptive sampling in sensor networks. Moreover, none of them could support the requirement of retrieving the critical points from the monitored physical world.

Some adaptive sampling algorithms are proposed for particular applications. For example, the algorithm in [30] is designed for target tracking. Each sensor uses sensory values sensed by its neighbours and itself to predict the target position and adjusts the sampling frequency adaptively. And [31] introduced an energy efficient algorithm to adjust the sampling frequency in the abnormal event detecting applications. It applies Fourier transform to predict the events and to adjust sampling frequency automatically. Since they are designed for particular applications, they have limitation in applying.

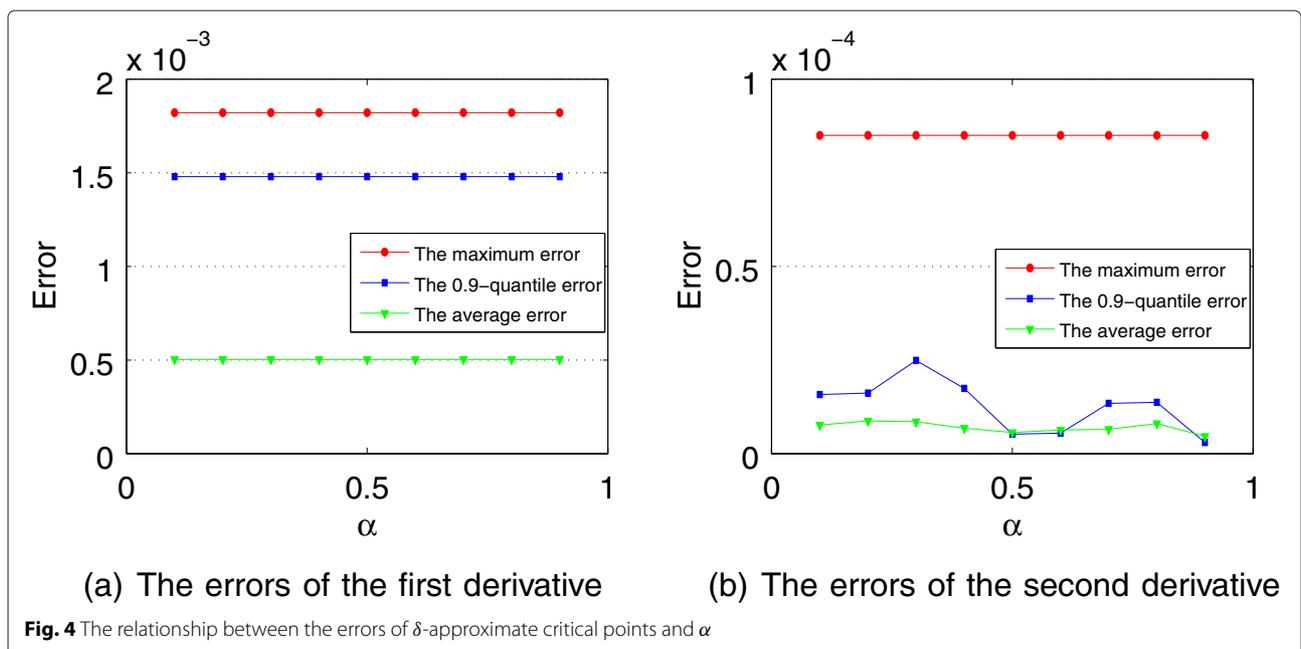


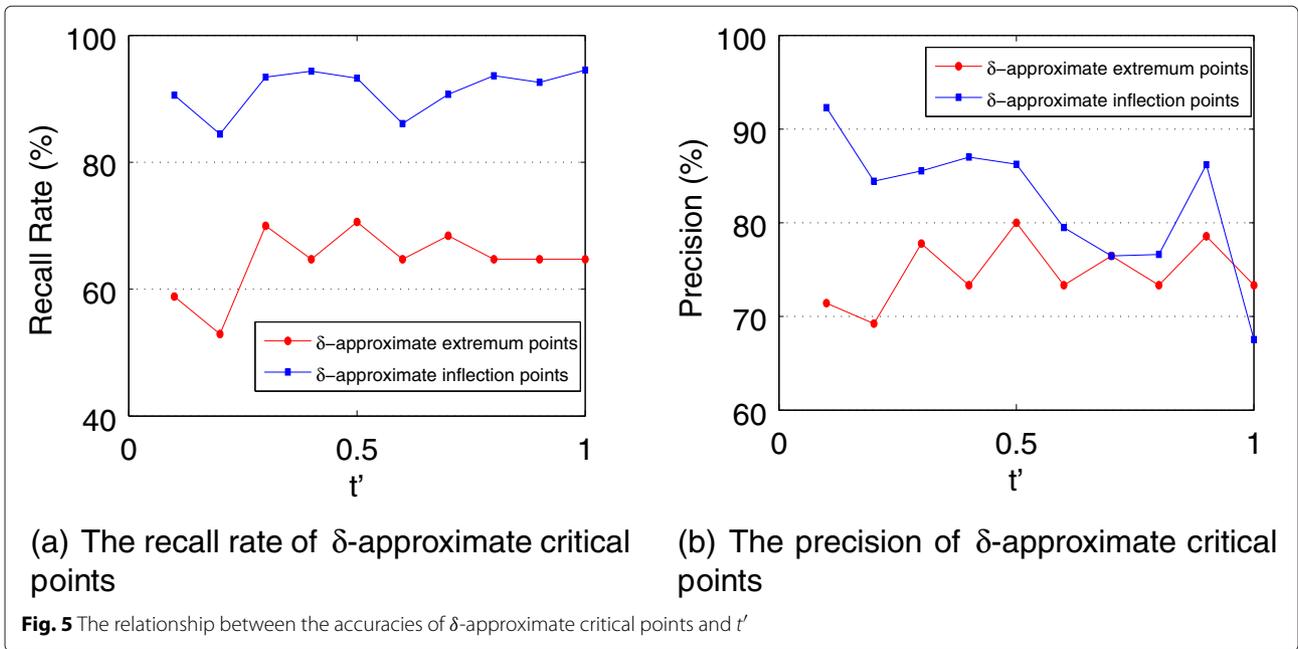
**Fig. 2** The relationship between the accuracies of  $\delta$ -approximate inflection points and  $\delta$



Most of works on adaptive sampling apply prediction models to estimate sensory values instead of sampling them. Jain and Chang [32] uses Kalman filter-based prediction model to predict sensory values, if the estimations beyond the acceptable range, they will adjust sampling frequency. However, the prediction ability of Kalman filter is limited, and the estimation error may be large. The prediction method in [33] is Box-Jenkins approach. The main idea of the work is to skip samplings from equal-sampling-frequency and use the forecast ones, which

can adjust the sampling frequency adaptively. However, since the method is based on equal-sampling-frequency, its accuracy is even worse than the EFS method. The work in [34] proposed a heuristic adaptive sampling algorithm for a glacial sensor network. Each sensor locally adjusts its sensing frequency based on the linear regression forecasting model. Such method reduces the energy cost of acquiring sensory data since the forecasting model is sufficiently utilized. However, the forecasting ability of the linear regression model is limited, and it



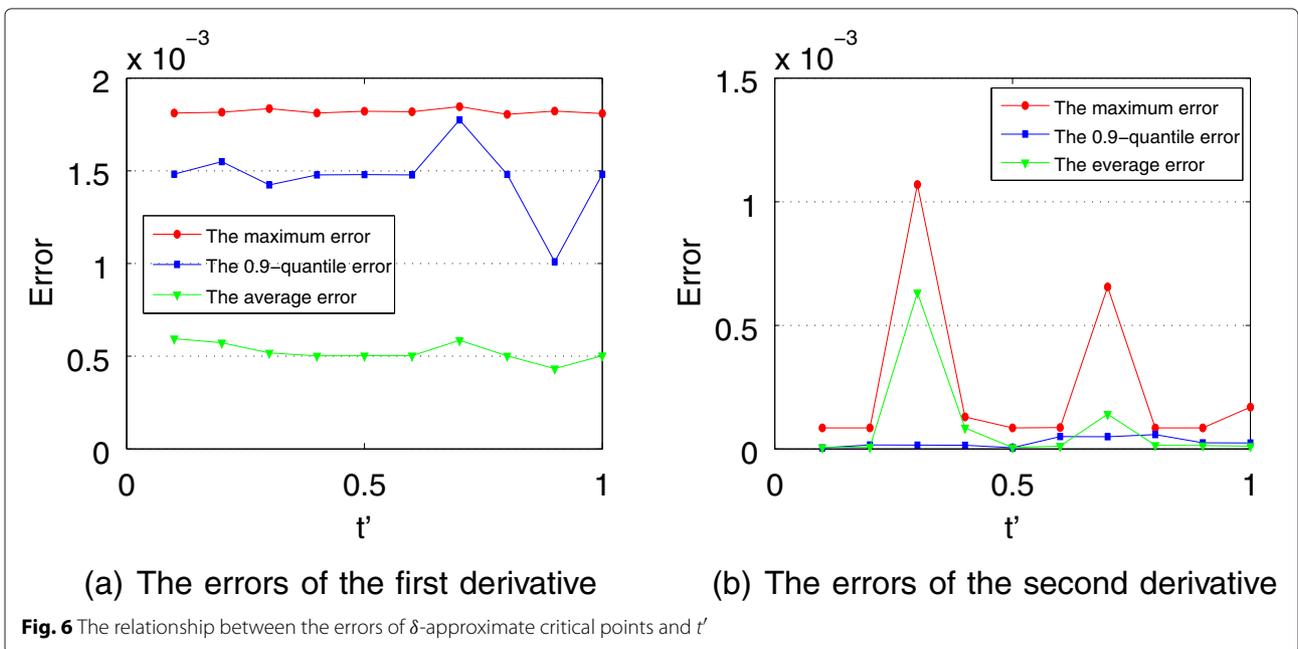


do not consider the problem of retrieving the critical points, either.

As for retrieving critical points in wireless sensor networks, some published works focus on it. However, few works focus on retrieving extremum and inflection points. The most common critical points they considering is top-k values.

The work in [20] proposed a novel approach for top-k query. The basic idea of it is to use filters for each node, which can filter the unnecessary sensor updates.

Silberstein et al. [19] studied the optimizing top-k queries in wireless sensor networks. The authors proposed to use samples of past sensor readings, which can reduce energy significantly. The work in [18] focuses on the location aware peak value query in sensor networks. It consider the top-k values' location which is not considered the traditional top-k query. The authors defined LAP- $(D, k)$  query which can find top-k values and the distance between any two values is larger than  $D$ . However, these papers only consider peak values. In face, besides peak



values, the extremum and inflection points are also critical points.

### 7 Conclusions

This paper studies the critical point aware data acquisition algorithm to obtain  $\delta$ -approximate extremum and inflection points from the physical world. We firstly provide the formal definition of  $\delta$ -approximate extremum and inflection points. Then, a data acquisition algorithm is proposed based on numerical analysis and Lagrange interpolation. Such algorithm can adjust the sampling frequency of each sensor node adaptively according to the variation of physical world. The correctness of the algorithm is proved and the its complexity is analyzed in detail. Finally, the extensive simulations are carried out, which show that our algorithm can achieve high accuracy of retrieving  $\delta$ -approximate extremum and inflection points from physical world.

### Appendix

The construction of the Lagrange interpolation polynomials are as follows.

The construction for  $L_1(t)$ : first, let  $l_{1k}(t)$  be a  $k$ -polynomial of  $t$  for  $0 \leq k \leq 3$ , where  $\{l_{1k}(t)|0 \leq k \leq 3\}$  satisfies that

$$l_{10}(t) = \frac{(t - t_{c-1}) \left(t - t_{\frac{2c-1}{2}}\right) (t - t_c)}{\left(t_{\frac{2c-3}{2}} - t_{c-1}\right) \left(t_{\frac{2c-3}{2}} - t_{\frac{2c-1}{2}}\right) \left(t_{\frac{2c-3}{2}} - t_c\right)}$$

$$l_{11}(t) = \frac{\left(t - t_{\frac{2c-3}{2}}\right) \left(t - t_{\frac{2c-1}{2}}\right) (t - t_c)}{\left(t_{c-1} - t_{\frac{2c-3}{2}}\right) \left(t_{c-1} - t_{\frac{2c-1}{2}}\right) (t_{c-1} - t_c)}$$

$$l_{12}(t) = \frac{\left(t - t_{\frac{2c-3}{2}}\right) (t - t_{c-1}) (t - t_c)}{\left(t_{\frac{2c-1}{2}} - t_{\frac{2c-3}{2}}\right) \left(t_{\frac{2c-1}{2}} - t_{c-1}\right) \left(t_{\frac{2c-1}{2}} - t_c\right)}$$

$$l_{13}(t) = \frac{\left(t - t_{\frac{2c-3}{2}}\right) (t - t_{c-1}) \left(t - t_{\frac{2c-1}{2}}\right)}{\left(t_c - t_{\frac{2c-3}{2}}\right) (t_c - t_{c-1}) \left(t_c - t_{\frac{2c-1}{2}}\right)}$$

Therefore,  $L_1(t) = S_i \left(t_{\frac{2c-3}{2}}\right) l_{10}(t) + S_i(t_{c-1}) l_{11}(t) + S_i \left(t_{\frac{2c-1}{2}}\right) l_{12}(t) + S_i(t_c) l_{13}(t)$ .

Similarly,  $L_2(t)$  is a fourth-order interpolation Lagrange polynomial and can be calculated by

$$L_2(t) = S_i(t_{c-2}) l_{20}(t) + S_i \left(t_{\frac{2c-3}{2}}\right) l_{21}(t) + S_i(t_{c-1}) l_{22}(t) + S_i \left(t_{\frac{2c-1}{2}}\right) l_{23}(t) + S_i(t_c) l_{24}(t), \text{ where}$$

$$l_{20}(t) = \frac{\left(t - t_{\frac{2c-3}{2}}\right) (t - t_{c-1}) \left(t - t_{\frac{2c-1}{2}}\right) (t - t_c)}{\prod_{j=1}^4 (t_{c-2} - t_{c-2+j/2})}$$

$$l_{21}(t) = \frac{(t - t_{c-2})(t - t_{c-1}) \left(t - t_{\frac{2c-1}{2}}\right) (t - t_c)}{\left(t_{\frac{2c-3}{2}} - t_{c-2}\right) \prod_{j=2}^4 \left(t_{\frac{2c-3}{2}} - t_{c-2+j/2}\right)}$$

$$l_{22}(t) = \frac{(t - t_{c-2}) \left(t - t_{\frac{2c-3}{2}}\right) \left(t - t_{\frac{2c-1}{2}}\right) (t - t_c)}{\prod_{j=0}^1 (t_{c-1} - t_{c-2+j/2}) \prod_{j=3}^4 (t_{c-1} - t_{c-2+j/2})}$$

$$l_{23}(t) = \frac{(t - t_{c-2}) \left(t - t_{\frac{2c-3}{2}}\right) (t - t_{c-1})(t - t_c)}{\left(t_{\frac{2c-1}{2}} - t_c\right) \prod_{j=0}^2 \left(t_{\frac{2c-1}{2}} - t_{c-2+j/2}\right)}$$

$$l_{24}(t) = \frac{(t - t_{c-2}) \left(t - t_{\frac{2c-3}{2}}\right) (t - t_{c-1}) \left(t - t_{\frac{2c-1}{2}}\right)}{(t_c - t_{c-2}) \left(t_c - t_{\frac{2c-3}{2}}\right) (t_c - t_{c-1}) \left(t_c - t_{\frac{2c-1}{2}}\right)}$$

### Competing interests

The authors declare that they have no competing interests.

### Acknowledgements

This work was supported in part by the National Basic Research Program of China (973 Program) under Grant No. 2012CB316200, the National Natural Science Foundation of China (NSFC) under Grant No. 61190115, 61370217, the Fundamental Research Funds for the Central Universities under grant No. HIT.K12TP201415, the National Science Foundation (NSF) under Grants No. CNS-1152001, CNS-1252292, the Research Fund for the Doctoral Program of Higher Education of China under grant No. 20132302120045, and the Natural Scientific Research Innovation Foundation in Harbin Institute of Technology under grant No. HIT.NSRIF. 2014070.

### Author details

<sup>1</sup>School of Computer Science and Tech, Harbin Institute of Technology, 150001 Harbin, China. <sup>2</sup>Department of Computer Science, Georgia State University, 30303 Atlanta, GA, USA.

Received: 24 September 2015 Accepted: 20 December 2015

Published online: 15 January 2016

### References

1. Darpa Sensit Program. <http://comlab.ecs.syr.edu/workshop2002/files/RichardButler.pdf>
2. S Kumar, D Shepherd, in *Proc. 4th Int. Conf. on Information Fusion*. SensIT: Sensor information technology for the warfighter (Elsevier, 2001), pp. 1–7
3. Z Cai, Z-Z Chen, G Lin, A 3.4713-approximation algorithm for the capacitated multicast tree routing problem. *Theor. Comput. Sci.* **410**(52), 5415–5424 (2008)
4. Z Cai, G Lin, G Xue, in *Proceedings of 11th Annual International Conference on Computing and Combinatorics 2005 (COCOON)*. Improved approximation algorithms for the capacitated multicast routing problem (Springer, Kunming, China, 2005), pp. 136–145
5. M Li, Y Liu, L Chen, Nonthreshold-based event detection for 3d environment monitoring in sensor networks. *IEEE Trans. Knowl. Data Eng.* **20**(12), 1699–1711 (2008)
6. Greenorbs. <http://greenorbs.org/>
7. K Liu, M Li, Y Liu, X Li, M Li, H Ma, in *Proceedings of the 18th Annual IEEE International Conference on Network Protocols, ICNP 2010, Kyoto, Japan, 5–8 October, 2010*. Exploring the hidden connectivity in urban vehicular networks (IEEE Computer Society, 2010), pp. 243–252

8. B Barbagli, L Bencini, I Magrini, G Manes, A Manes, in *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference, IWCMC 2011, Istanbul, Turkey, 4–8 July, 2011*. A real-time traffic monitoring based on wireless sensor network technologies (IEEE, 2011), pp. 820–825
9. Structural Health Monitoring of the Golden Gate Bridge. <http://www.cs.berkeley.edu/~binetude/ggb/>
10. S Kim, S Pakzad, DE Culler, J Demmel, G Fenves, S Glaser, M Turon, in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks, IPSN 2007, Cambridge, Massachusetts, USA, April 25–27, 2007*. Health monitoring of civil infrastructures using wireless sensor networks (ACM, 2007), pp. 254–263
11. S Kim, S Pakzad, DE Culler, J Demmel, G Fenves, S Glaser, M Turon, in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, SenSys 2006, Boulder, Colorado, USA, October 31 - November 3, 2006*. Wireless sensor networks for structural health monitoring (Springer, 2006), pp. 427–428
12. JT Overpeck, GA Meehl, S Bony, DR Easterling, Climate data challenges in the 21 st century. *Science* (Washington). **331**(6018), 700–702 (2011)
13. S Cheng, Z Cai, J Li, Curve query processing in wireless sensor networks. *IEEE Trans. Veh. Technol.* **64**(11), 5198–5209 (2015)
14. J Li, S Cheng, ( $\epsilon, \delta$ )-approximate aggregation algorithms in dynamic sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **23**(3), 385–396 (2012)
15. S Cheng, J Li, in *29th IEEE International Conference on Distributed Computing Systems (ICDCS 2009), 22–26 June 2009, Montreal, Québec, Canada*. Sampling based ( $\epsilon, \delta$ )-approximate aggregation algorithm in sensor networks (IEEE Computer Society, 2009), pp. 273–280
16. S Cheng, J Li, Q Ren, L Yu, in *INFOCOM 2010. 29th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 15–19 March 2010, San Diego, CA, USA*. Bernoulli sampling based ( $\epsilon, \delta$ )-approximate aggregation in large-scale sensor networks (IEEE, 2010), pp. 1181–1189
17. Z He, Z Cai, S Cheng, X Wang, Approximate aggregation for tracking quantiles in wireless sensor networks. *Theor. Comput. Sci.* **607**, 381–390 (2015)
18. S Cheng, J Li, L Yu, in *Proceedings of the IEEE INFOCOM 2012, Orlando, FL, USA, March 25–30, 2012*. Location aware peak value queries in sensor networks (IEEE, 2012), pp. 486–494
19. A Silberstein, R Braynard, CS Ellis, K Munagala, J Yang, in *Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, 3–8 April 2006, Atlanta, GA, USA*. A sampling-based approach to optimizing top-k queries in sensor networks (IEEE Computer Society, 2006), p. 68
20. M Wu, J Xu, X Tang, W Lee, Top-k monitoring in wireless sensor networks. *IEEE Trans. Knowl. Data Eng.* **19**(7), 962–976 (2007)
21. I Su, Y Chung, C Lee, Y Lin, Efficient skyline query processing in wireless sensor networks. *J. Parallel Distrib. Comput.* **70**(6), 680–698 (2010)
22. Z Wu, M Wang, L Yuan, H Jiang, in *Proceedings of the 2nd International Conference on BioMedical Engineering and Informatics (BMEI)*. Optimized routing structure based skyline query algorithm in wireless sensor network (IEEE Computer Society, Tianjin, China, 2009), pp. 1–5
23. J Cao, LE Li, A Chen, T Bu, in *Proceedings of the 1st ACM Workshop on Mobile Internet Through Cellular Networks*. Incremental tracking of multiple quantiles for network monitoring in cellular networks (ACM, New York, NY, USA, 2009), pp. 7–12
24. Z Huang, L Wang, K Yi, Y Liu, in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12–16, 2011*. Sampling based algorithms for quantile computation in sensor networks (ACM, 2011), pp. 745–756
25. J Li, S Cheng, H Gao, Z Cai, Approximate physical world reconstruction algorithms in sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **25**(12), 3099–3110 (2014)
26. S Cheng, J Li, Z Cai, in *Proceedings of the IEEE INFOCOM 2013, Turin, Italy, April 14–19, 2013*. O( $\epsilon$ )-approximation to physical world by sensor networks, (2013), pp. 3084–3092
27. C-E Froberg, CE Frhoberg, *Introduction to Numerical Analysis*. (Addison-Wesley Reading, Massachusetts, USA, 1969)
28. MJD Powell, *Approximation Theory and Methods*. (Cambridge university press, England, 1981)
29. T Lindgren, J Sanchez, J Hall, in *Graphics Gems III*. Curve tessellation criteria through sampling (ACM, San Diego, CA, USA, 1992), pp. 262–265
30. M Rahimi, R Safabakhsh, in *Proceedings of the IEEE International Conference on Wireless Communications, Networking and Information Security, WCNIS 2010, 25–27 June 2010, Beijing, China*. Adaptation of sampling in target tracking sensor networks (IEEE, 2010), pp. 301–305
31. C Alippi, G Anastasi, MD Francesco, M Roveri, An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors. *IEEE Trans. Instrum. Meas.* **59**(2), 335–344 (2010)
32. A Jain, EY Chang, in *Proceedings of the 1st Workshop on Data Management for Sensor Networks, in Conjunction with VLDB, DMSN 2004, Toronto, Canada, August 30, 2004*. Adaptive sampling for sensor networks (ACM, 2004), pp. 10–16
33. YW Law, S Chatterjea, J Jin, T Hanselmann, M Palaniswami, in *Proceedings of the International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, IWCMC 2009, Leipzig, Germany, June 21–24, 2009*. Energy-efficient data acquisition by adaptive sampling for wireless sensor networks (ACM, 2009), pp. 1146–1151
34. P Padhy, RK Dash, K Martinez, NR Jennings, in *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8–12, 2006*. A utility-based sensing and communication model for a glacial sensor network (ACM, 2006), pp. 1353–1360

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---