Article

Above-Threshold Queries of Environmental Conditions Based on Bilinear Interpolation in Wireless Sensor Networks

Yaxi Liu , Wei Huangfu * , Haijun Zhang and Keping Long

Beijing Advanced Innovation Center for Materials Genome Engineering, Beijing Engineering and Technology Research Center for Convergence Networks and Ubiquitous Services, School of Computer and Communication Engineering, University of Science and Technology Beijing(USTB), Beijing 100083, China; b20160295@xs.ustb.edu.cn (Y.L.); haijunzhang@ieee.org (H.Z.); longkeping@ustb.edu.cn (K.L.)

* Correspondence: huangfuwei@ustb.edu.cn; Tel.: +86-139-1116-3520

Received: 23 September 2018; Accepted: 23 November 2018; Published: 30 November 2018

Abstract: Wireless sensor networks can be regarded as sensor database systems, which permit users to query sensor data of interest. Among various spatial database queries, we focus the area-wise aggregate queries in the region where the sensor values are above a predefined threshold, which are summarized as above-threshold queries. In this paper, we propose a novel Bilinear Interpolation-Based (BIB) algorithm, which utilizes the bilinear interpolation to estimate the environmental variables inside a grid with the known sensor values at the vertexes, to support the above-threshold queries for regularly-deployed sensor networks and provide the closed-form solution of the above-threshold ratio. We designate experiments with both the artificially-constructed environment data and the real temperature data. Experiment results manifest that the proposed BIB algorithm shows a good performance in estimating the above-threshold ratios to support the above-threshold queries in an accurate and efficient manner.

Keywords: wireless sensor networks; spatial database queries; window queries; above-threshold queries; bilinear interpolation

1. Introduction

Wireless sensor networks are sharply emerging as one of the most significant technologies to bridge the gap between the physical world and the digital world [1–3]. In order to sense spatial natural phenomena in the physical world remotely, wireless sensor networks usually consist of many smart energy-efficient devices, namely sensor nodes that are equipped with inexpensive embedded sensors, processors, memories, radio communication modules, and power supplies to gather environmental conditions, ranging from temperature, precipitation, smoke, soil humidity to atmospheric pressure, to name a few. Wireless sensor networks provide us prolific spatiotemporal environment data and will play important roles in many applications, including environmental monitoring, target tracking, precision agriculture, and so forth [4–9].

Since data are constantly being sent from sensors, wireless sensor networks can be regarded as sensor database systems, such as TinyDB [10]. The ability to apply various kinds of data-querying techniques to sensors and process data is highly desirable. Various methods and techniques of data storage and data query have been proposed, which permit users to manage sensor data of interest [11,12].

Spatial database queries, which gather environmental variables, from the perspective of geography, inside a particular geographical region of interest (ROI), have become one of the primary technologies due to the spatial characteristics of the collected environmental conditions [13,14]. As one

of the various kinds of spatial database queries, window queries mainly refer to those that retrieve the sensor data inside a specified geographical window, e.g., a two-dimensional rectangular region $\{(X, Y) \mid X_{min} \leq X \leq X_{max}, Y_{min} \leq Y \leq Y_{max}\}$ [15]. Some complicated spatial queries are used to determine the region in which the sensor values satisfy a predefined condition [16–18]. For example, in Ref. [19], in order to find out the relationships between the weather and the habitats of animal species, scientists should query the species densities in the rainy regions and those in the sunny regions. Assume that when the humidity function $H(X, Y)$ is above the rainy threshold $TH_{humidity}$, it begins to rain, and vice versa. Thus, the queries should determine the region where the sensor values are above a given threshold from humidity sensors, denoted as $\{(X, Y) \mid H(X, Y) \geq TH_{humidity}\}$.

Aggregate queries, another critical type of database query, refer to the summaries of the current sensor values by implementing the basic database aggregates such as COUNT, MIN, MAX, SUM, AVERAGE, etc. [20,21]. Queries are often combined with aggregation operations to support complex analysis of environmental data [19,22]. Due to the spatial characteristics of the environmental conditions, queries should support the area-wise aggregate operation. In Ref. [23], Aslan proposed a forest fire detection and monitoring framework. In such a case, we will pay more attention to the total area where the temperature values are above a predefined threshold, i.e., the above-threshold area. An alarm should be triggered, for example, if the area of the hot subregion where the temperature is above 40 °C is greater than 5% of the total region. Here, the query will retrieve the area $|\{(X, Y) \mid T(X, Y) \geq 40\text{ °C}\}|$, and we will run the aggregate query on the spatial region determined by the sensor values, like “SELECT AREA(*) from temperature_table where T > 40”. We summarize such area-wise aggregate queries in the region where the sensor values are above a predefined threshold as above-threshold queries in this paper.

Furthermore, there are two deployment patterns in sensor networks to achieve particular environmental monitoring: random deployment and the regular (deterministic) one [24]. Although the regular deployment scheme is considered to be too ideal to implement in practical applications, it usually has better network coverage and connectivity than the random deployment [24–26]. Furthermore, it is practically developed in some applications such as forest fire prevention and high temperature alert [23] and offers us a basis to further our research. Thus, we will only focus on the regular deployment for wireless sensor networks.

For the regular deployment of wireless sensor networks, there are some related techniques to support the above-threshold queries. The Nearest Neighbor (NN) algorithm is a technique to estimate the environmental conditions only by the sensor data provided by the nearest sensors, which leads to a Voronoi diagram [27], to enable the accumulation of the areas of polygons, inside which the estimated environmental conditions are beyond the threshold. The Marching Square (MS) algorithm is a computer graphics algorithm to generate contours for a two-dimensional scalar field, referring to environmental conditions here, and thus also supports the above-threshold queries based on the isobars [28,29]. Both algorithms are extensively studied. However, the NN algorithm, only taking the information about whether the sensor values are greater than the threshold or not into account, is robust, but inaccurate. The MS algorithm performs the linear interpolation to estimate the environmental conditions on the edges based on the sensor values, which usually produces more accurate results. Nevertheless, the MS algorithm determines the above-threshold subregion by a straightforward connection of all the points on the edges by interpolation, and thus, the environmental conditions inside the region are not adequately considered.

In this paper, we propose a novel Bilinear Interpolation-Based (BIB) algorithm to support the above-threshold queries for regularly-deployed sensor networks. We utilize the bilinear interpolation to estimate the environmental variables inside a square with the known sensor values from sensor nodes located at its four vertexes. The information of environmental variables is taken into full consideration by performing interpolation inside the grid. Thus, the area of the above-threshold subregion can be obtained by mathematical formulation, and finally, we propose the closed-form solution of the above-threshold ratio. In order to evaluate the performance of different above-threshold

query algorithms, we first use the Kriging interpolation, which can be regarded as the gold standard to compare the interpolation weights of various algorithms. The results of theory analysis show that the proposed BIB above-threshold algorithm is an approximation of the Kriging algorithm, while the NN algorithm has a large calculation error in geographic data analysis. Then, we perform the artificially-constructed environment data experiment and real temperature data experiment to draw the boundaries and check the metrics of False Positives (FP), False Negatives (FN), and Mean Squared Error (MSE). Experiment results show that the proposed BIB algorithm has a good performance in estimating the above-threshold ratios to support the above-threshold queries in an accurate and efficient manner.

The rest of the paper is organized as follows. Section 2 discusses related works on the applications of the regular deployment sensor networks and enumerates the database management techniques for window queries and aggregate queries. Section 3 presents the system model and formulates the problem. Section 4 introduces the bilinear interpolation-based above-threshold query algorithm. Section 5 describes the experiments and discusses the results. Finally, Section 6 concludes this paper.

2. Related Work

Quite extensive distributed database management techniques have been proposed to support the spatial queries, mainly the window queries. Berchtold et al. [30] proposed a pyramid technique to index data in the high-dimensional database system, which is efficient for window queries. In Ref. [31], Ooi et al. presented another tunable index scheme called iMinMax for high-dimensional window queries, which was less complex and performed better than the pyramid technique. However, there are two major issues impeding the window queries: the lack of explicit window semantics and the low implementation efficiency. To solve this problem, Li et al. [22] established a window query technique framework by defining window semantics to avoid the intra-operator buffers and the reprocessing tuples. However, the previous works have not yet explored the spatial property in sensor network databases. Xu et al. [32] investigated a framework of spatial window query in dynamic sensor networks. To further simplify the query process, Xu et al. [33] proposed an infrastructure-free window query technique in sensor networks. Ku et al. [34] presented an approach to reduce the access latency of the spatial queries, e.g., window queries, by utilizing the neighbor information in wireless broadcast environments. The geographical location information can guide the window queries due to the spatial correlation, and the inspiration of the spatial-aware approach can also be used in wireless sensor networks. The existing database management techniques mainly focus on the geographical coordinate-type window queries, i.e., the traditional window queries, rather than the spatial region determined by environmental values. The environmental value-determined spatial region was also studied [19,23]. Note that the window queries are not only restricted to the spatial coordinators, but also applied to the environmental conditions. The latter window queries to retrieve the sensor values inside a predefined interval were discussed [16–18]. It is obvious that such queries degenerate into above-threshold ones if the lower bound of the query window is the threshold and the upper bound is assigned to the positive infinite.

The spatial queries combined with aggregate operation, which are closely related to the above-threshold query techniques in this paper, have also been studied and used in some environmental monitoring applications. In order to figure out the influence factors of climate changes, Wang et al. [35] established a window query algorithm to support various aggregate queries on three-dimensional time-varying datasets, such as the AVERAGE, MIN, and MAX operations.

The deployment is significant for sensor networks. A regular deployment pattern is important in some scenarios such as disaster alert [24–26]. In Ref. [36], Bulusu proposed a GPS (Global Positioning System)-less low-cost outdoor localization model for small devices, e.g., sensors, which are placed in a regular mesh structure. Wang et al. [37] proposed a deployment algorithm in an arbitrarily-shaped region and used the regular sensor patterns in a region with boundaries and obstacles. In Ref. [38], the typical regularly-deployed oilfield with an area of 8×8 km required approximately a 50–100 m

distance between the sensors in seismic analysis. In Ref. [23], Aslan proposed a forest fire detection and monitoring framework and implemented it in regularly-deployed wireless sensor networks. Two popular layout models—square layout, i.e., temperature sensors are placed at the corners of squares, and hexagonal layout, i.e., sensors are placed at the corners of the hexagons—were adopted.

To estimate the environmental conditions at a given location to support spatial queries, the NN and MS algorithms are usually used. The former estimates the environmental conditions only by the nearest sensors, while the latter generates the isobars by the linear interpolation on the edges. Both are applicable to the above-threshold queries. The NN algorithm is simple, and its computation workload and local communication traffic load are small. The MS algorithm is more accurate by taking the environmental conditions on the edges into account. However, neither adequately exploits the sensor values to better the interpolation of the environmental conditions inside the region.

Thus, the idea to fully exploit the sensor values to better the interpolation inside the region inspires us to propose a novel BIB algorithm for the above-threshold queries. With small incrementation of the computation workload and the local communication traffic, the proposed BIB algorithm performs interpolation inside the region and deduces a closed-form above-threshold query solution. It is not only more accurate than the NN and MS algorithms, but also accordant with the gold standard in the geographical sense.

3. System Model and Problem Formulation

In this section, we present the system model and formulate the problem.

3.1. System Model

Consider a wireless sensor network deployed in a predefined square region R , consisting of N sensor nodes, $e_1, e_2, \dots, e_N$. Each sensor node senses and then transfers the environment condition data to the database server. Let $f(x, y)$ be the environmental variable at the position with the coordinate (x, y) . For a given threshold T , the above-threshold (sub-)region query is used to evaluate the above-threshold region R_T , which is defined as:

$$R_T \triangleq \{(x, y) \mid f(x, y) \geq T, (x, y) \in R\}. \quad (1)$$

The above-threshold ratio is then defined as:

$$C_T \triangleq \frac{|R_T|}{|R|} \quad (2)$$

where $|X|$ is the cardinality, i.e., the area of the region X .

Suppose that the region R is split into many regular non-overlapping rectangle grids, as depicted in Figure 1. All the rectangle subregions are listed as $\mathbf{S} = \{S_1, S_2, \dots, S_m\}$, where m is the total number of rectangle subregions.

There are four sensor nodes distributed at four vertexes of S_i . Sensors can gather environmental variables, which can be used to estimate the above-threshold ratio. Notice that adjacent rectangles may share some sensor nodes.

Figure 1. Illustration of the system.

Then, the area-wise above-threshold ratio is given by:

$$C_T = \frac{\sum_{i=1}^m |S_{Ti}|}{\sum_{i=1}^m |S_i|} = \frac{\sum_{i=1}^m |S_{Ti}|}{|R|} \quad (3)$$

where S_{Ti} is defined as

$$S_{Ti} \triangleq \{(x, y) \mid f(x, y) \geq T, (x, y) \in S_i\}. \quad (4)$$

3.2. Problem Formulation

Our goal is to estimate the above-threshold ratio C_T . From Equation (3), we can obtain C_T by summing the above-threshold area limited inside each S_i . Thus, the next objective is to calculate $|S_{Ti}|$ according to the coordinates and the sensor values of the four vertexes in S_i .

Without loss of generality and for the sake of simplicity, we only consider one of the rectangle subregions and rewrite such a region S_i as S . The vertex coordinates of rectangle S are $A(x_A, y_A)$, $B(x_B, y_B)$, $C(x_C, y_C)$, and $D(x_D, y_D)$, and the sensor values at four vertexes are f_A , f_B , f_C , and f_D , respectively. Let W and H be the length and the height of S , as shown in Figure 2.

Figure 2. Illustration of rectangle subregion characteristics.

Now, the goal is to calculate the above-threshold area inside S with length W and height H under the condition of known coordinates and the sensor values of four vertexes. The above-threshold region S_T in S is defined as:

$$S_T \triangleq \{(x, y) \mid f(x, y) \geq T, (x, y) \in S\}. \quad (5)$$

4. Bilinear Interpolation-Based Above-Threshold Query Algorithm

In this section, we propose the bilinear interpolation-based algorithm and describe the nearest neighbor algorithm and the marching square algorithm. Furthermore, we introduce the evaluative metrics of mean squared error, false positive, and false negative.

4.1. Coordinate Transformation and Boundary of the Above-Threshold Region

In order to simplify the calculation, we perform a translation and an affine transformation to convert the coordinate (x, y) inside S to $(u = \frac{x-x_A}{W}, v = \frac{y-y_A}{H})$. Now, the region S is converted into a unit square with $u_A = 0$ and $v_A = 0$. Thus, we have $S = \{(u, v) \mid 0 \leq u \leq 1, 0 \leq v \leq 1\}$. To obtain the original above-threshold area, the result in this subsection should be multiplied by $W \cdot H$ due to the area unit $dx \cdot dy = W \cdot H \cdot du \cdot dv$.

We utilize the bilinear interpolation to estimate the environmental variables inside S . Bilinear interpolation is a widespread interpolation algorithm for two variables on a two-dimensional grid. It is an extension of linear interpolation for one variable. The key idea of bilinear interpolation is to perform linear interpolation first in one direction and then again in the other direction. Although each step is linear, the bilinear interpolation as a whole is not linear, but rather, quadratic.

The sensor value at arbitrary point (u, v) inside S can be interpolated by the sensor values of four vertexes based on bilinear interpolation. Denote $\hat{f}(u, v)$ as the estimated environmental variable. Thus,

$$\begin{aligned} \hat{f}(u, 0) &= (1-u)f_A + uf_B, \\ \hat{f}(u, 1) &= (1-u)f_D + uf_C, \\ \hat{f}(u, v) &= (1-v)\hat{f}(u, 0) + v\hat{f}(u, 1) \\ &= a_{11}uv + a_{10}u + a_{01}v + a_{00} \end{aligned} \quad (6)$$

where:

$$\begin{aligned} a_{11} &= f_A + f_C - f_B - f_D, \\ a_{10} &= -f_A + f_B, \\ a_{01} &= -f_A + f_D, \\ a_{00} &= f_A. \end{aligned} \quad (7)$$

For the four-vertex interpolation in $S = \{(u, v) \mid 0 \leq u \leq 1, 0 \leq v \leq 1\}$, the estimated sensor value of arbitrary coordinate (u, v) in the square is given by:

$$\hat{f}(u, v) = \mathbf{W}^T(u, v) \cdot \begin{pmatrix} f_A \\ f_B \\ f_C \\ f_D \end{pmatrix} \quad (8)$$

where $W(u, v)$ is a 4×1 matrix, which implies the interpolation weights of four vertexes. We can conclude from Equations (6) and (7) that:

$$W_{\text{BIB}}(u, v) = \begin{pmatrix} uv - u - v + 1 \\ u - uv \\ uv \\ v - uv \end{pmatrix}. \quad (9)$$

Denote the estimated above-threshold region $\hat{S}_T$ as $\{(u, v) | \hat{f}(u, v) \geq T, (u, v) \in S\}$ and the estimated above-threshold ratio $\hat{C}_T$ as $\frac{1}{|R|} \cdot \sum_{i=1}^m |\hat{S}_{Ti}|$. The boundary of $\hat{S}_T$, which lies between the above-threshold part and the below-threshold one, is given by:

$$\hat{f}(u, v) = a_{11}uv + a_{10}u + a_{01}v + a_{00} = T. \quad (10)$$

The boundary $\hat{f}(u, v) = T$ is hyperbolic if $a_{11} \neq 0$ and is degraded to a straight line if $a_{11} = 0$. Various situations for the boundaries and the above-threshold regions with the BIB algorithm are shown in Figure 3. The solid black circles on the vertexes represent the above-threshold values, and the hollow ones represent the below-threshold values. The threshold T is assigned as zero. The black lines are the boundaries where $\hat{f}(u, v) = T$, and the shaded areas are the above-threshold regions where $\hat{f}(u, v) \geq T$. In such a concrete case, the sensor values at points A, B, C , and D are assigned to $\pm 3, \pm 2, \pm 4$, and ± 1 , respectively, depending on whether the sensor values are above or below the threshold.

Figure 3. Illustrations of the boundaries and the above-threshold regions with the Bilinear Interpolation-Based (BIB) algorithm.

4.2. Above-Threshold Area and Above-Threshold Ratio

The above-threshold area $|\hat{S}_T|$ can be discussed for different cases. If the boundary is hyperbolic ($a_{11} \neq 0$), the expression of $\hat{f}(u, v)$ can be transformed to:

$$\hat{f}(u, v) = a_{11}\hat{u}\hat{v} + Q \quad (11)$$

where:

$$\hat{u} = u + \frac{a_{01}}{a_{11}}, \quad (12)$$

$$\hat{v} = v + \frac{a_{10}}{a_{11}}, \quad (13)$$

$$Q = \frac{a_{00}a_{11} - a_{01}a_{10}}{a_{11}}. \quad (14)$$

The boundary $\hat{f}(u, v) = a_{11}\hat{u}\hat{v} + Q = T$ is equivalent to $\hat{u}\hat{v} = E$ where $E = \frac{T-Q}{a_{11}}$. Let G_E denote the region $\hat{u}\hat{v} \geq E$, defined as:

$$G_E = \{(\hat{u}, \hat{v}) | \hat{u}\hat{v} \geq E\}. \quad (15)$$

If $a_{11} > 0$, $\hat{S}_T$ is the intersection of the region G_E and the region $\hat{S}$, or precisely, $\hat{S}_T = \hat{S} \cap G_E$. Otherwise, if $a_{11} < 0$, we have $\hat{S}_T = \hat{S} \setminus G_E$. Without loss of generality, we only discuss the case of $a_{11} > 0$. The illustrations of the boundaries and the above-threshold regions in the $\hat{u}\hat{v}$ plane if $a_{11} > 0$ are shown in Figure 4.

Figure 4. Illustrations of the boundaries and the regions G_E in the $\hat{u}\hat{v}$ plane if $a_{11} > 0$.

We denote an auxiliary function $k(E, \hat{u}_0, \hat{v}_0)$ as

$$k(E, \hat{u}_0, \hat{v}_0) = \int_0^{\hat{u}_0} \int_0^{\hat{v}_0} \mathbb{1}_{G_E}(\hat{u}, \hat{v}) d\hat{u} d\hat{v} \quad (16)$$

where $\mathbb{1}_{\mathbb{X}}(u, v)$ is the characteristic function of the set $\mathbb{X}$, which satisfies:

$$\mathbb{1}_{\mathbb{X}}(u, v) = \begin{cases} 1 & \text{if } (u, v) \in \mathbb{X}, \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

Furthermore, let us define:

$$\mathbb{U} = \begin{cases} [0, \hat{u}_0] & \hat{u}_0 \geq 0, \\ [\hat{u}_0, 0] & \hat{u}_0 < 0, \end{cases} \quad (18)$$

$$\mathbb{V} = \begin{cases} [0, \hat{v}_0] & \hat{v}_0 \geq 0, \\ [\hat{v}_0, 0] & \hat{v}_0 < 0. \end{cases} \quad (19)$$

Denote the symbol “+” as the positive sign of E , $\hat{u}_0$, or $\hat{v}_0$ and the symbol “−” as the negative one. For example, the notation “+++” of the symbol $k^{+++}(E, \hat{u}_0, \hat{v}_0)$ indicates that all the independent variables E , $\hat{u}_0$, and $\hat{v}_0$ are positive.

There are two situations if $E > 0$, $\hat{u}_0 > 0$ and $\hat{v}_0 > 0$. If $\hat{u}_0\hat{v}_0 \geq E$, the function value $k^{+++}(E, \hat{u}_0, \hat{v}_0) = |G_E \cap (\mathbb{U} \times \mathbb{V})|$ is the area of the shaded region shown in Figure 5a. Otherwise, the function value $k^{+++}(E, \hat{u}_0, \hat{v}_0) = 0$, as shown in Figure 5b.

Thus, we have:

$$k^{+++}(E, \hat{u}_0, \hat{v}_0) = \int_0^{\hat{u}_0} \int_0^{\hat{v}_0} \mathbb{1}_{G_E}(\hat{u}, \hat{v}) \, d\hat{u} \, d\hat{v} = \begin{cases} \hat{u}_0\hat{v}_0 - E - E \ln \frac{\hat{u}_0\hat{v}_0}{E} & \hat{u}_0\hat{v}_0 \geq E, \\ 0 & \hat{u}_0\hat{v}_0 < E. \end{cases} \quad (20)$$

Similarly, the result of $E > 0$, $\hat{u}_0 < 0$, and $\hat{v}_0 < 0$ can be calculated by:

$$k^{+--}(E, \hat{u}_0, \hat{v}_0) = \int_{\hat{u}_0}^0 \int_{\hat{v}_0}^0 \mathbb{1}_{G_E}(\hat{u}, \hat{v}) \, d\hat{u} \, d\hat{v} = \begin{cases} \hat{u}_0\hat{v}_0 - E - E \ln \frac{\hat{u}_0\hat{v}_0}{E} & \hat{u}_0\hat{v}_0 \geq E, \\ 0 & \hat{u}_0\hat{v}_0 < E. \end{cases} \quad (21)$$

In the same way, we have:

$$k^{+-+}(E, \hat{u}_0, \hat{v}_0) = 0, \quad (22)$$

$$k^{++-}(E, \hat{u}_0, \hat{v}_0) = 0. \quad (23)$$

So far, all the cases of $E > 0$ have been taken into consideration.

If $E < 0$, $\hat{u}_0 > 0$ and $\hat{v}_0 > 0$, we always have $\hat{u}_0\hat{v}_0 > E$, then:

$$k^{-++}(E, \hat{u}_0, \hat{v}_0) = \hat{u}_0\hat{v}_0. \quad (24)$$

Similarly, if $E > 0$, $\hat{u}_0 < 0$ and $\hat{v}_0 < 0$, we have:

$$k^{---}(E, \hat{u}_0, \hat{v}_0) = \hat{u}_0\hat{v}_0. \quad (25)$$

There are two situations for $E < 0$, $\hat{u}_0 < 0$ and $\hat{v}_0 > 0$. If $\hat{u}_0\hat{v}_0 \geq E$, the function value $k^{--+}(E, \hat{u}_0, \hat{v}_0) = \hat{u}_0\hat{v}_0$, as shown in Figure 6a. Otherwise, the function value $k^{--+}(E, \hat{u}_0, \hat{v}_0) = -|\mathbb{U} \times \mathbb{V} - G_E|$.

Figure 5. Illustrations of function values $k^{+++}(E, \hat{u}_0, \hat{v}_0)$ if $E > 0$, $\hat{u}_0 > 0$, and $\hat{v}_0 > 0$.

Figure 6. Illustrations of function values $k^{--}(E, \hat{u}_0, \hat{v}_0)$ if $E < 0$, $\hat{u}_0 < 0$, and $\hat{v}_0 > 0$.

Therefore,

$$\begin{aligned}
 k^{--}(E, \hat{u}_0, \hat{v}_0) &= \int_0^{\hat{u}_0} \int_0^{\hat{v}_0} \mathbb{1}_{G_E}(\hat{u}, \hat{v}) \, d\hat{u} \, d\hat{v} = - \int_{\hat{u}_0}^0 \int_0^{\hat{v}_0} \mathbb{1}_{G_E}(\hat{u}, \hat{v}) \, d\hat{u} \, d\hat{v} \\
 &= \begin{cases} -E - E \ln \frac{\hat{u}_0 \hat{v}_0}{E} & \hat{u}_0 \hat{v}_0 \geq E, \\ \hat{u}_0 \hat{v}_0 & \hat{u}_0 \hat{v}_0 < E. \end{cases} \quad (26)
 \end{aligned}$$

Similarly, the result of $E < 0$, $\hat{u}_0 > 0$, and $\hat{v}_0 < 0$ is given by:

$$\begin{aligned}
 k^{--}(E, \hat{u}_0, \hat{v}_0) &= \int_0^{\hat{u}_0} \int_0^{\hat{v}_0} \mathbb{1}_{G_E}(\hat{u}, \hat{v}) \, d\hat{u} \, d\hat{v} = - \int_0^{\hat{u}_0} \int_{\hat{v}_0}^0 \mathbb{1}_{G_E}(\hat{u}, \hat{v}) \, d\hat{u} \, d\hat{v} \\
 &= \begin{cases} -E - E \ln \frac{\hat{u}_0 \hat{v}_0}{E} & \hat{u}_0 \hat{v}_0 \geq E, \\ \hat{u}_0 \hat{v}_0 & \hat{u}_0 \hat{v}_0 < E. \end{cases} \quad (27)
 \end{aligned}$$

If $E = 0$, we can see from Figure 4c that $k^{0++}(E, \hat{u}_0, \hat{v}_0) = k^{0--}(E, \hat{u}_0, \hat{v}_0) = \hat{u}_0 \hat{v}_0$ and $k^{0-+}(E, \hat{u}_0, \hat{v}_0) = k^{0+-}(E, \hat{u}_0, \hat{v}_0) = 0$.

The function value $k(E, \hat{u}_0, \hat{v}_0)$ in different cases is listed in Table 1. The symbol “*” can be either “+”, “−”, or “0”, implying the positive or the negative sign of $\hat{u}_0$ or $\hat{v}_0$.

Table 1. The function value $k(E, \hat{u}_0, \hat{v}_0)$ in different cases.

E	$\hat{u}_0$	$\hat{v}_0$	$k(E, \hat{u}_0, \hat{v}_0)$
+	+	+	$\begin{cases} \hat{u}_0 \hat{v}_0 - E - E \ln \frac{\hat{u}_0 \hat{v}_0}{E} & (\hat{u}_0 \hat{v}_0 \geq E) \\ 0 & (\hat{u}_0 \hat{v}_0 < E) \end{cases}$
+	−	+	0
+	−	−	$\begin{cases} \hat{u}_0 \hat{v}_0 - E - E \ln \frac{\hat{u}_0 \hat{v}_0}{E} & (\hat{u}_0 \hat{v}_0 \geq E) \\ 0 & (\hat{u}_0 \hat{v}_0 < E) \end{cases}$
+	+	−	0
+	0	*	0
+	*	0	0
−	+	+	$\hat{u}_0 \hat{v}_0$
−	−	+	$\begin{cases} -E - E \ln \frac{\hat{u}_0 \hat{v}_0}{E} & (\hat{u}_0 \hat{v}_0 \geq E) \\ \hat{u}_0 \hat{v}_0 & (\hat{u}_0 \hat{v}_0 < E) \end{cases}$
−	−	−	$\hat{u}_0 \hat{v}_0$
−	+	−	$\begin{cases} -E - E \ln \frac{\hat{u}_0 \hat{v}_0}{E} & (\hat{u}_0 \hat{v}_0 \geq E) \\ \hat{u}_0 \hat{v}_0 & (\hat{u}_0 \hat{v}_0 < E) \end{cases}$
−	0	*	0
−	*	0	0
0	+	+	$\hat{u}_0 \hat{v}_0$
0	−	+	0
0	−	−	$\hat{u}_0 \hat{v}_0$
0	+	−	0
0	0	*	0
0	*	0	0

In the coordinate system $(\hat{u}, \hat{v})$, the original coordinates of four vertexes are converted into:

$$\hat{A} = (\hat{u}_A, \hat{v}_A) = \left(\frac{a_{01}}{a_{11}}, \frac{a_{10}}{a_{11}} \right), \quad (28)$$

$$\hat{B} = (\hat{u}_B, \hat{v}_B) = \left(1 + \frac{a_{01}}{a_{11}}, \frac{a_{10}}{a_{11}} \right), \quad (29)$$

$$\hat{C} = (\hat{u}_C, \hat{v}_C) = \left(1 + \frac{a_{01}}{a_{11}}, 1 + \frac{a_{10}}{a_{11}} \right), \quad (30)$$

$$\hat{D} = (\hat{u}_D, \hat{v}_D) = \left(\frac{a_{01}}{a_{11}}, 1 + \frac{a_{10}}{a_{11}} \right). \quad (31)$$

For similarity, let us define $H_E(P) \triangleq k(E, u_P, v_P)$. When considering various practical situations, no matter where the square S lies, the above-threshold area $|\hat{S}_T|$ is always provided by:

$$\begin{aligned}
 |\hat{S}_T| &= \left| \int_{\hat{u}_A}^{\hat{u}_C} \int_{\hat{v}_A}^{\hat{v}_C} \mathbb{1}_{G_E}(\hat{u}, \hat{v}) \, d\hat{u} \, d\hat{v} \right| \\
 &= \left| \left(\int_{\hat{u}_A}^0 \int_{\hat{v}_A}^0 + \int_0^{\hat{u}_C} \int_0^{\hat{v}_C} + \int_{\hat{u}_A}^0 \int_0^{\hat{v}_C} + \int_0^{\hat{u}_C} \int_{\hat{v}_A}^0 \right) \mathbb{1}_{G_E}(\hat{u}, \hat{v}) \, d\hat{u} \, d\hat{v} \right| \\
 &= \left| \int_0^{\hat{u}_A} \int_0^{\hat{v}_A} \mathbb{1}_{G_E}(\hat{u}, \hat{v}) \, d\hat{u} \, d\hat{v} + \int_0^{\hat{u}_C} \int_0^{\hat{v}_C} \mathbb{1}_{G_E}(\hat{u}, \hat{v}) \, d\hat{u} \, d\hat{v} \right. \\
 &\quad \left. - \int_0^{\hat{u}_A} \int_0^{\hat{v}_C} \mathbb{1}_{G_E}(\hat{u}, \hat{v}) \, d\hat{u} \, d\hat{v} - \int_0^{\hat{u}_C} \int_0^{\hat{v}_A} \mathbb{1}_{G_E}(\hat{u}, \hat{v}) \, d\hat{u} \, d\hat{v} \right| \\
 &= |H_E(\hat{A}) - H_E(\hat{B}) + H_E(\hat{C}) - H_E(\hat{D})|.
 \end{aligned} \tag{32}$$

When $a_{11} < 0$, the above-threshold area $|\hat{S}_T|$ is the total area $|S|$, subtracting the area $|S_T|$ for $a_{11} > 0$. We have:

$$|\hat{S}_T| = 1 - |H_E(\hat{A}) - H_E(\hat{B}) + H_E(\hat{C}) - H_E(\hat{D})|. \tag{33}$$

If $a_{11} = 0$, the boundary is degraded into a straight line and can be transformed to:

$$\hat{f}(u, v) = a_{10}u + a_{01}v + a_{00} = T. \tag{34}$$

Then, the above-threshold area $|\hat{S}_T|$ can be calculated according to the following four steps:

Step 1. If $\hat{f}(u_A, v_A) \geq T$, add $A(u_A, v_A)$ to the ordered set U .

Step 2. If the edge $\overline{AB}$ is intersected with the boundary $\hat{f}(u, v) = T$ at the point $F(u_F, v_F)$ that satisfies $\hat{f}(u_F, v_F) = T$, append $F(u_F, v_F)$ to the ordered set U .

Step 3. Similarly, judge if the point B , the intersection between $\overline{BC}$ and $\hat{f}(u, v) = T$, the point C , the intersection between $\overline{CD}$ and $\hat{f}(u, v) = T$, the point D , and the intersection between $\overline{DA}$ and $\hat{f}(u, v) = T$ should be added to the ordered set U according to the principles in Step 1 and Step 2.

Step 4. If ordered set $U = \emptyset$, $|\hat{S}_T| = 0$. Otherwise, $|\hat{S}_T|$ is equal to the area of the polygon of which the vertexes are listed in the ordered set U .

So far, the above-threshold area $|\hat{S}_T|$ inside the subregion S is obtained by the bilinear interpolation for all the cases.

We propose a bilinear interpolation-based algorithm to estimate the above-threshold ratio in a predefined region, as shown in Algorithm 1.

Algorithm 1 BIB algorithm.

Input: Region R , threshold T , the four coordinates (x_A, y_A) , (x_B, y_B) , (x_C, y_C) , and (x_D, y_D) , and the four sensor values f_A, f_B, f_C , and f_D of all S_i , the heights and the widths of all S_i

Output: $\hat{C}_T$

```

1: for  $i \leftarrow 1, m$  do
2:    $S \leftarrow S_i$ .
3:   Calculate  $a_{11}, a_{10}, a_{01}$ , and  $a_{00}$  with reference to  $S$ .
4:   if  $a_{11} \neq 0$  then
5:      $\hat{u} \leftarrow u + \frac{a_{01}}{a_{11}}$ 
6:      $\hat{v} \leftarrow v + \frac{a_{10}}{a_{11}}$ 
7:      $Q \leftarrow \frac{a_{00}a_{11} - a_{01}a_{10}}{a_{11}}$ 
8:      $E \leftarrow \frac{T-Q}{a_{11}}$ 
9:     Calculate  $H_E(\hat{A}), H_E(\hat{B}), H_E(\hat{C}), H_E(\hat{D})$ .
10:     $S' \leftarrow H_E(\hat{A}) - H_E(\hat{B}) + H_E(\hat{C}) - H_E(\hat{D})$ .
11:    if  $a_{11} > 0$  then
12:       $|\hat{S}_T| \leftarrow W \cdot H \cdot |S'|$ 
13:    else
14:       $|\hat{S}_T| \leftarrow W \cdot H \cdot (1 - |S'|)$ 
15:    end if
16:  else
17:     $U \leftarrow \{\}$ 
18:    for  $j \leftarrow [A, \overline{AB}, B, \overline{BC}, C, \overline{CD}, D, \overline{DA}]$  do
19:      if  $j$  in  $[A, B, C, D]$  then
20:        if  $\hat{f}(u_j, v_j) \geq T$  then
21:          Add  $j$  to  $U$ .
22:        end if
23:      end if
24:      if  $j$  in  $[\overline{AB}, \overline{BC}, \overline{CD}, \overline{DA}]$  then
25:        if  $j$  is intersected with the line  $\hat{f}(u, v) = T$  then
26:          Add the intersection to the ordered set  $U$ .
27:        end if
28:      end if
29:    end for
30:    if  $U = \emptyset, |\hat{S}_T| = 0$  then
31:       $|\hat{S}_T| \leftarrow 0$ 
32:    else
33:       $|\hat{S}_T|$  is equal to the area of the polygon of which the vertexes are listed in set  $U$ .
34:    end if
35:  end if
36:   $|\hat{S}_{Ti}| \leftarrow |\hat{S}_T|$ 
37: end for
38:  $\hat{C}_T \leftarrow \frac{1}{|R|} \cdot \sum_{i=1}^m |\hat{S}_{Ti}|$ 

```

4.3. Nearest Neighbor Algorithm and Marching Square Algorithm

Both the NN and MS algorithm utilize the sensor values of four vertexes in the rectangle subregion.

In the NN algorithm, the sensor value of arbitrary coordinate (u, v) in subregion S is determined by that of the nearest vertex, and the interpolation weight $W_{NN}(u, v)$ in Equation (8) can be expressed as:

$$W_{NN}(u, v) = \begin{pmatrix} \mathbb{1}_{[0, \frac{1}{2}] \times [0, \frac{1}{2}]}(u, v) \\ \mathbb{1}_{(\frac{1}{2}, 1] \times [0, \frac{1}{2}]}(u, v) \\ \mathbb{1}_{(\frac{1}{2}, 1] \times (\frac{1}{2}, 1]}(u, v) \\ \mathbb{1}_{[0, \frac{1}{2}] \times (\frac{1}{2}, 1]}(u, v) \end{pmatrix}. \quad (35)$$

The illustrations of the boundaries and the above-threshold regions with the NN algorithm are shown in Figure 7.

Figure 7. Illustrations of the boundaries and the above-threshold regions with the NN algorithm.

The MS algorithm is a computer graphics algorithm, and thus, its interpolation weight does not exist. The illustrations of the boundaries and the above-threshold regions with the MS algorithm are depicted in Figure 8. Note that there are two ambiguous situations for above-threshold calculation and boundary description. The MS algorithm is possibly considered as an approximation of the BIB algorithm, which transforms the hyperbolic boundaries by the BIB algorithm into the straight lines.

Note that in order to estimate the above-threshold coverage ratio in region R , we sum the above-threshold area limited inside each rectangle grid S_i for the three algorithms. Thus, the algorithm complexities of the three mentioned algorithms are both $O(M)$, where M is the number of grids.

Figure 8. Illustrations of the boundaries and the above-threshold regions with the Marching Square (MS) algorithm.

4.4. Evaluative Metrics

The MSE in mathematical statistics refers to the mean value of the squares of the differences between the estimated values and true values, which is given by:

$$MSE = \frac{1}{N_e} \sum_{t=1}^{N_e} (observed_t - predicted_t)^2 \quad (36)$$

where $observed_t$ represents the exact above-threshold ratio in the t th experiment, $predicted_t$ is the estimated above-threshold ratio obtained by different algorithms in the t th experiment, and N_e is the number of experiments. A lesser MSE is better.

In statistical hypothesis testing, the FP and FN are known as type I and type II errors, where FP corresponds to the situation when the exact value is false, while the estimated value is true and FN is opposite. True Positive (TP) represents that the exact value and the estimated value are both true, and True Negative (TN) represents both false. In the above-threshold queries, the conditions of FP, FN, TP, and TN are listed in Table 2. The probability of FP and FN in a single experiment can be expressed as:

$$FP = \frac{1}{m} \sum_{i=1}^m \frac{|(S - S_{Ti}) \cap \hat{S}_{Ti}|}{|S|}, \quad (37)$$

$$FN = \frac{1}{m} \sum_{i=1}^m \frac{|S_{Ti} \cap (S - \hat{S}_{Ti})|}{|S|}. \quad (38)$$

Table 2. False positive, false negative, true positive, and true negative.

Exact Sensor Value \ Estimated Sensor Value	Estimated Sensor Value	
	Positive (Above T)	Negative (Under T)
Positive (Above T)	TP	FN
Negative (Under T)	FP	TN

5. Experiments and Discussion

In this section, we first testify to the correctness of the proposed algorithm by theory analysis. Then, we describe the experiments with both the artificially-constructed environment data and the real temperature data. All programming and benchmarking were done in Python (Version 3.6) with the Numpy [39] and pyKriging [40] libraries running on a computer with a 2600-MHz Intel processor and 8-gigabyte memories. Furthermore, we illustrate the possible implementation frameworks in wireless sensor network and analyze the algorithm complexity. Finally, we provide the evaluation and the comparison of the NN, MS, and BIB algorithms.

5.1. Theory Analysis

In geography, the Kriging algorithm is one of the most wide-spread interpolation algorithms and can be regarded as the gold standard [41].

We use the local four-point Kriging (Kr4) interpolation to validate the performance of the proposed algorithm. Assume that the Kr4 interpolation is controlled by the four vertexes of the square. The semi-variable function is linear, and the nugget is zero. The interpolation weight $W_{Kr4}(u, v)$ in Equation (8) can be calculated as:

$$W_{Kr4}(u, v) = \begin{pmatrix} a_1 b_1 + a_2 b_2 + a_3 b_3 + a_4 b_4 + \frac{1}{4} \\ a_2 b_1 + a_1 b_2 + a_2 b_3 + a_3 b_4 + \frac{1}{4} \\ a_3 b_1 + a_2 b_2 + a_1 b_3 + a_2 b_4 + \frac{1}{4} \\ a_2 b_1 + a_3 b_2 + a_2 b_3 + a_1 b_4 + \frac{1}{4} \end{pmatrix} \quad (39)$$

where:

$$\begin{aligned} a_1 &= -\frac{1}{4} - \frac{3\sqrt{2}}{8}, \quad a_2 = \frac{1}{4} + \frac{\sqrt{2}}{8}, \quad a_3 = -\frac{1}{4} + \frac{\sqrt{2}}{8}, \\ b_1 &= \sqrt{u^2 + v^2}, \quad b_2 = \sqrt{(u-1)^2 + v^2}, \\ b_3 &= \sqrt{(u-1)^2 + (v-1)^2}, \quad b_4 = \sqrt{u^2 + (v-1)^2}. \end{aligned} \quad (40)$$

We denote the i th elements in $W_{Kr4}(u, v)$, $W_{BIB}(u, v)$ and $W_{NN}(u, v)$ as $W_{Kr4}^i(u, v)$, $W_{BIB}^i(u, v)$, and $W_{NN}^i(u, v)$, respectively, and these values are shown in Figure 9 for $i = 1$. We can conclude that the weights $W_{Kr4}^1(u, v)$ and $W_{BIB}^1(u, v)$ are almost identical, while the weight $W_{NN}^1(u, v)$ is quite different from the other two.

Figure 9. Function images of $W_{Kr4}^1(u, v)$, $W_{BIB}^1(u, v)$, and $W_{NN}^1(u, v)$.

The differences of weights, i.e., $|W_{BIB}^1(u, v) - W_{Kr4}^1(u, v)|$ and $|W_{NN}^1(u, v) - W_{Kr4}^1(u, v)|$, are shown in Figure 10. The largest weight error $\max_{u,v} |W_{BIB}^1(u, v) - W_{Kr4}^1(u, v)|$ is less than 0.03 in the BIB algorithm, while $\max_{u,v} |W_{NN}^1(u, v) - W_{Kr4}^1(u, v)|$ is more than 0.5 in the NN algorithm. The same

conclusion holds for the other weights $W_{Kr4}^i(u, v)$, $W_{BIB}^i(u, v)$, and $W_{NN}^i(u, v)$, $i = 2, 3, 4$. Therefore, the interpolation result of the BIB algorithm almost approximates that of Kr4 algorithm. Note that the above-threshold area is difficult to be obtained by the Kr4 algorithm due to the complex calculation of the integral inside a complex boundary. Moreover, the NN algorithm has a large weight error in geographic data analysis.

Figure 10. Function images of $|W_{BIB}^1(u, v) - W_{Kr4}^1(u, v)|$ and $|W_{NN}^1(u, v) - W_{Kr4}^1(u, v)|$.

5.2. Experiment on Artificially-Constructed Environment Data

In order to evaluate different above-threshold query algorithms, we test them with artificial data generated by the Kriging algorithm. The interpolated values by the Kriging algorithm are modeled as a Gaussian process governed by full prior covariances.

Assume that the Kriging interpolation is controlled by N_c points P_i , $1 \leq i \leq N_c$, randomly distributed in R , and the value f at P_i is an independent random variable uniformly distributed in the interval $[a, b]$. Typical artificial environmental variables are generated by N_c control points inside a 300×300 region. We denote the central 100×100 region as the ROI in which the sensor values are depicted in Figure 11 in a single experiment. The greater the number of the control points is, the richer the details of the variances of artificial environmental variables are.

Figure 11. Artificial environmental variables for simulation experiments.

To compare different algorithms with the same number of sensors, we assume that there are in total $N = (n + 1) \times (n + 1)$ sensors and $M = n \times n$ grids.

We performed simulation experiments to contrast the three algorithms, NN, MS, and BIB, to estimate the above-threshold ratios. The simulation configuration is given in Table 3.

Table 3. Simulation configuration.

Parameter	Value
Side length of R	100
N_c	100
a	−20
b	20
N_e	2000

The real boundary is shown in Figure 12a, and the boundaries generated by the three algorithms in ROI when $n = 3$ are drawn in Figure 12b–d. Obviously, the boundaries by the BIB and MS algorithms are more accurate than those by the NN algorithm. Furthermore, the boundaries by the BIB algorithm are smoother than those by the other algorithms.

Figure 12. The real boundary and the boundaries generated by the three mentioned algorithms in ROI when $n = 3$ in the artificially-constructed environment data experiment. (a) The real boundary. (b) The boundaries by the NN algorithm. (c) The boundaries by the MS algorithm. (d) The boundaries by the BIB algorithm.

In order to analyze the three aforementioned algorithms qualitatively, we used the metrics FP, FN, and MSE to evaluate their performance for N_e experiments. The average FP, FN, and MSE versus the number of grids M for the three mentioned algorithms are shown in Figure 13.

Figure 13. The average FP, FN, and MSE versus the grid numbers M for the three mentioned algorithms in the artificially-constructed environment data experiment.

The threshold $T = \frac{f_{\max} + f_{\min}}{2}$, where $f_{\max}$ and $f_{\min}$ are the maximum and the minimum of the variables. First, all the FP, FN, and MSE metrics obtained by the three algorithms tend to a definite value, zero, along with the increasing of the number of sensors. This implies all algorithms are feasible and more sensors help to reduce the query error. Second, the NN algorithm performs worse than

the others. This is mainly caused by the rough estimating of the variables inside the ROI. Since the proposed BIB algorithm is an approximation of the Kr4 algorithm and the MS algorithm is an approximation of the BIB algorithm, the MS and BIB algorithms both performed well.

5.3. Experiment of Real Temperature Data

WorldClim [42] is a set of gridded global climate data with a spatial resolution from 1–340 km² over 30 years, which was created by Steve Fick and Robert Hijmans. WorldClim provides various environmental variables, including minimum temperature (°C), maximum temperature (°C), average temperature (°C), precipitation (mm), solar radiation (kJm⁻²day⁻¹), wind speed (ms⁻¹) and water vapor pressure (kPa). These data can be used for mapping and spatial modeling.

The global average monthly temperature data, that is 12 sets of temperature data, obtained by WorldClim from 1970–2000, were used in the following experiments. We first randomly selected an ROI that did not contain a water surface, ranging from 42–78° E and from 49–65° N, to observe the boundaries. The hypsometric map of the ROI is depicted in Figure 14.

Figure 14. The hypsometric map of the ROI.

Rectangle divisions were done to the ROI, and $M = n \times n$ denotes the grid numbers. The real boundary and the boundaries generated by the three algorithms in ROI when $n = 20$ are depicted in Figure 15.

Obviously, the boundaries by the BIB algorithm are more accurate than those by the NN and MS algorithms. Furthermore, the boundaries by the BIB algorithm are smoother than those by the other algorithms.

Figure 15. The real boundary and the boundaries generated by the three mentioned algorithms in ROI when $n = 20$ in the real temperature data experiment.

Then, we randomly selected 20 ROIs that did not contain a water surface with a length of 10° both in the latitude and longitude direction in Eurasia to demonstrate the universality of the proposed algorithm. We assumed that in the case that the distance between the sampling points was 30 s, i.e., the most precise spatial resolution, the average above-threshold ratio obtained by the NN algorithm was the exact above-threshold ratio. The average FP, FN, and MSE versus the number of grids M for the three mentioned algorithms in 20 ROIs are shown in Figure 16. The threshold T was assigned to the median of the temperature data. Although the effects of the results cannot be favorably compared with those in the artificial experiment due to the discontinuity points such as the mountain segregation, the shapes and the trends of the curves in the real temperature data experiment are the same as those in the artificially-constructed environment data experiment. Above all, the proposed BIB algorithm shows a good performance in estimating the above-threshold ratio and solving the above-threshold query equation within the given ROI.

Figure 16. The average FP, FN, and MSE versus the grid numbers M for the three mentioned algorithms in the real temperature data experiment.

5.4. Possible Implementation Framework and Computation/Communication Load Analysis

A naive implementation framework to support the aforementioned above-threshold queries with the BIB, NN, or MS algorithms is to collect all the sensor data and then process the data in the center server. All the positions of the nodes were assumed to be obtained with node location techniques. A possible data collection route tree to transfer the sensor data from nodes is shown in Figure 17a. In such a naive implementation, the nodes only send out their data, and the center server needs to execute the query, which is thus named the centralized query framework. The traffic loads in the sensor network of different algorithms were equal. However, the center server needed more computation for the BIB and MS algorithms to perform the interpolation than that for the NN algorithm, which in fact simply counted the number of the data above the threshold.

However, in most practical scenarios, there would be only a small proportion of the sensor values above the threshold. We herein propose a distributed implementation framework to support the aforementioned above-threshold queries.

In this distributed implementation framework to support the NN algorithm, the sensor will transfer its sensor value if and only if it is triggered by the above-threshold event, i.e., its sensor value is beyond the threshold, as shown in Figure 17b. The nodes in the route path from the triggered node to the sink node help to forward the data hop by hop. Therefore, the center server only counted the data from the triggered nodes, and the values of other nodes were assumed to be below the threshold. Now, the network traffic was significantly reduced if the above-threshold events occurred sparsely.

Figure 17. Possible implementation frameworks in wireless sensor networks. (a) Centralized query framework. (b) Distributed implementation framework for the NN algorithm. (c) Distributed implementation framework for the MS or BIB algorithm.

It is more complicated to support the MS and BIB algorithms since both need local co-operation. We first grouped the sensor nodes into overlapped clusters. All four nodes in the vertexes of a square grid make a cluster. One node possibly belongs to a different cluster since it may be shared by neighbor grids. There was a cluster head for each cluster, as shown in Figure 17c. When a node, say, s_1 , is triggered by the above-threshold event, it will send its sensor value to notify the cluster heads of all the clusters to which it belongs. Each cluster head notified, say s_4 , will command its cluster members to submit their sensor values to it. Once the header collects all the sensor values in the cluster, it will perform the computation and transfer the fused result to the sink hop by hop. Therefore, the MS and BIB algorithms needed more local traffic and more computation to exchange their sensor values for the interpolation.

5.5. Comparison and Discussion

We evaluate and compare the three mentioned algorithms, NN, MS, and BIB, according to various evaluative metrics in Table 4. We can see from Figures 12 and 15 that the curve boundaries by the BIB algorithm are more accurate and smoother than the straight line boundaries by the other two algorithms. We can see from Figures 13 and 16 that all algorithms are feasible and that NN algorithm performed worse than the others. The largest weight error is less than 0.03 in the BIB algorithm, while it is more than 0.5 in the NN algorithm. Therefore, the interpolation result of the BIB algorithm almost approximates that of the Kr4 algorithm. Since the MS algorithm is possibly considered as an approximation of the BIB algorithm, the same conclusion holds for it. Both the BIB and the MS algorithms have small weight errors in geographic data analysis, while the NN algorithm have the larger one. The interpolation weights $W_{BIB}(u, v)$ and $W_{NN}(u, v)$ are given by Equations (9) and (35), respectively. The MS algorithm is a computer graphics algorithm, and thus, its interpolation weight does not exist. We can see from Figure 8 that there were two ambiguous situations for above-threshold area calculation and boundary description in the MS algorithm. For the distributed implementation framework, once the header collects all the sensor values in the cluster, it will perform the computation and transfer the fused result to the sink hop by hop in the MS and BIB algorithms. The center server only counts the data from the triggered nodes and the values of other nodes were assumed to be below the threshold in NN algorithm. Therefore, the MS and BIB algorithms need more local traffic and more computation to exchange their sensor values for the interpolation. The traffic and computation are modest in the MS and BIB algorithms due to the small above-threshold clusters. Taking all the factors into consideration, the proposed BIB algorithm has an excellent comprehensive performance compared with the other algorithms.

Table 4. Algorithm comparison.

Index	Algorithm		
	NN	MS	BIB
Boundary Accuracy	Worse	Good	Best
Boundary Continuity	Worse	Good	Best
FP	Worse	Good	Good
FN	Worse	Good	Good
MSE	Worse	Good	Good
Geographic Interpretation	Worse	Good	Good
Closed-form Solution	Yes	No	Yes
Ambiguity	No	Sometimes	No
Computation Workload	Little	Medium	Medium
Communication Traffic	Little	Medium	Medium

6. Conclusions

Wireless sensor networks can be regarded as sensor database systems, which permit users to query sensor data of interest. In this paper, we proposed a novel Bilinear Interpolation-Based (BIB) algorithm to support the above-threshold queries. It is mainly designed for regularly-deployed sensor networks. We utilize the bilinear interpolation to estimate the environmental variables inside a square with the known sensor values from sensor nodes located at its four vertexes. Thus, the area of the above-threshold subregion can be obtained by mathematical formulation, and finally, we proposed the closed-form solution of the above-threshold ratio.

In order to evaluate the performance of different above-threshold query algorithms, we first used the interpolation weights of Kr4 interpolation, which is commonly used in geometry research, as a golden standard to compare various algorithms. The weights of the proposed BIB algorithm are approximate to those of the Kr4 interpolation, which means BIB is valid from the perspective of geography.

We also described experiments both with the artificially-constructed environment data and with real temperature data. Experiment results manifest that the proposed BIB algorithm shows a good performance in estimating the above-threshold ratio.

In summary, with a closed-form formulation, the proposed BIB algorithm supports the above-threshold queries in an efficient and accurate manner.

Author Contributions: Conceptualization, Y.L. and W.H.; formal analysis, Y.L. and W.H.; investigation, W.H.; methodology, Y.L. and W.H.; software, Y.L. and W.H.; supervision, H.Z. and K.L.; validation, Y.L. and W.H.; visualization, Y.L. and W.H.; writing, original draft, Y.L. and W.H.; writing, review and editing, Y.L., W.H., H.Z., and K.L.

Funding: This research is funded by the Joint Foundation of MoE (Ministry of Education) and China Mobile Group (No. MCM20160103).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fadel, E.; Gungor, V.C.; Nassef, L.; Akkari, N.; Malik, M.A.; Almasri, S.; Akyildiz, I.F. A survey on wireless sensor networks for smart grid. *Comput. Commun.* **2015**, *71*, 22–33. [\[CrossRef\]](#)
2. Khan, I.; Belqasmi, F.; Glitho, R.; Crespi, N.; Morrow, M.; Polakos, P. Wireless sensor network virtualization: A survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 553–576. [\[CrossRef\]](#)
3. Rawat, P.; Singh, K.D.; Chaouchi, H.; Bonnini, J.M. Wireless sensor networks: A survey on recent developments and potential synergies. *J. Supercomput.* **2014**, *68*, 1–48. [\[CrossRef\]](#)
4. Aponte-Luis, J.; Gómez-Galán, J.A.; Gómez-Bravo, F.; Sánchez-Raya, M.; Alcina-Espigado, J.; Teixido-Rovira, P.M. An efficient wireless sensor network for industrial monitoring and control. *Sensors* **2018**, *18*, 182. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Mainwaring, A.; Culler, D.; Polastre, J.; Szewczyk, R.; Anderson, J. Wireless sensor networks for habitat monitoring. In Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, GA, USA, 28 September 2002. [\[CrossRef\]](#)
6. Abbasi, A.Z.; Islam, N.; Shaikh, Z.A. A review of wireless sensors and networks applications in agriculture. *Comput. Stand. Interfaces* **2014**, *36*, 263–270. [\[CrossRef\]](#)
7. Noel, A.; Abdaoui, A.; Badawy, A.; Elfouly, T.; Ahmed, M.; Shehata, M. Structural health monitoring using wireless sensor networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1403–1423. [\[CrossRef\]](#)
8. Srbinovska, M.; Gavrovski, C.; Dimcev, V.; Krkoleva, A.; Borozan, V. Environmental parameters monitoring in precision agriculture using wireless sensor networks. *J. Clean. Prod.* **2015**, *88*, 297–307. [\[CrossRef\]](#)
9. Ueyama, J.; Faiçal, B.S.; Mano, L.Y.; Bayer, G.; Pessin, G.; Gomes, P.H. Enhancing reliability in wireless sensor networks for adaptive river monitoring systems: Reflections on their long-term deployment in Brazil. *Comput. Environ. Urban Syst.* **2017**, *65*, 41–52. [\[CrossRef\]](#)

10. Kim, D.O.; Liu, L.; Shin, I.S.; Kim, J.J.; Han, K.J. Spatial TinyDB: A spatial sensor database system for the USN environment. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 512368. [[CrossRef](#)]
11. Diallo, O.; Rodrigues, J.J.; Sene, M.; Lloret, J. Distributed database management techniques for wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 604–620. [[CrossRef](#)]
12. Cheng, S.; Cai, Z.; Li, J. Curve query processing in wireless sensor networks. *IEEE Trans. Veh. Technol.* **2015**, *64*, 5198–5209. [[CrossRef](#)]
13. Eldawy, A.; Mokbel, M.F. Spatialhadoop: A mapreduce framework for spatial data. In Proceedings of the IEEE 31st International Conference on Data Engineering, Seoul, Korea, 13–17 April 2015. [[CrossRef](#)]
14. Tang, J.; Zhang, B.; Zhou, Y.; Wang, L. An energy-aware spatial index tree for multi-region attribute query aggregation processing in wireless sensor networks. *IEEE Access* **2017**, *5*, 2080–2095. [[CrossRef](#)]
15. Pagel, B.U.; Six, H.W.; Winter, M. Window query-optimal clustering of spatial objects. In Proceedings of the Fourteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, San Jose, CA, USA, 22–25 May 1995. [[CrossRef](#)]
16. Molina-Pico, A.; Cuesta-Frau, D.; Araujo, A.; Alejandre, J.; Rozas, A. Forest monitoring and wildland early fire detection by a hierarchical wireless sensor network. *J. Sens.* **2016**, *2016*, 8325845. [[CrossRef](#)]
17. Ferentinos, K.P.; Katsoulas, N.; Tzounis, A.; Bartzanas, T.; Kittas, C. Wireless sensor networks for greenhouse climate and plant condition assessment. *Biosyst. Eng.* **2017**, *153*, 70–81. [[CrossRef](#)]
18. Ojha, T.; Misra, S.; Raghuwanshi, N.S. Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. *Comput. Electron. Agric.* **2015**, *118*, 66–84. [[CrossRef](#)]
19. Yang, X.; Lim, H.B.; Özsu, T.M.; Tan, K.L. In-network execution of monitoring queries in sensor networks. In Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, Beijing, China, 11–14 June 2007. [[CrossRef](#)]
20. Madden, S.; Szewczyk, R.; Franklin, M.J.; Culler, D. Supporting aggregate queries over ad-hoc wireless sensor networks. In Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications, Callicoon, NY, USA, 20–21 June 2002. [[CrossRef](#)]
21. Li, J.; Cheng, S.; Cai, Z.; Yu, J.; Wang, C.; Li, Y. Approximate holistic aggregation in wireless sensor networks. *ACM Trans. Sens. Netw.* **2017**, *13*, 11. [[CrossRef](#)]
22. Li, J.; Maier, D.; Tufte, K.; Papadimos, V.; Tucker, P.A. Semantics and evaluation techniques for window aggregates in data streams. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, MD, USA, 14–16 June 2005. [[CrossRef](#)]
23. Aslan, Y.E.; Korpeoglu, I.; Ulusoy, Ö. A framework for use of wireless sensor networks in forest fire detection and monitoring. *Comput. Environ. Urban Syst.* **2012**, *36*, 614–625. [[CrossRef](#)]
24. Yun, Z.; Bai, X.; Xuan, D.; Lai, T.H.; Jia, W. Optimal deployment patterns for full coverage and k-connectivity wireless sensor networks. *IEEE/ACM Trans. Netw.* **2010**, *18*, 934–947. [[CrossRef](#)]
25. Hossain, A.; Biswas, P.K.; Chakrabarti, S. Sensing models and its impact on network coverage in wireless sensor network. In Proceedings of the IEEE Region 10 and the Third international Conference on Industrial and Information Systems, Kharagpur, India, 8–10 December 2008. [[CrossRef](#)]
26. Lin, C.Y.; Peng, W.C.; Tseng, Y.C. Efficient in-network moving object tracking in wireless sensor networks. *IEEE Trans. Mob. Comput.* **2006**, *5*, 1044–1056. [[CrossRef](#)]
27. Aurenhammer, F. Voronoi diagrams: A survey of a fundamental geometric data structure. *ACM Comput. Surv.* **1991**, *23*, 345–405. [[CrossRef](#)]
28. Maple, C. Geometric design and space planning using the marching squares and marching cube algorithms. In Proceedings of the IEEE International Conference on Geometric Modeling and Graphics, London, UK, 16–18 July 2003. [[CrossRef](#)]
29. Mantz, H.; Jacobs, K.; Mecke, K. Utilizing minkowski functionals for image analysis: A marching square algorithm. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, 12015. [[CrossRef](#)]
30. Berchtold, S.; Böhm, C.; Kriegel, H.P. The pyramid-technique: Towards breaking the curse of dimensionality. In Proceedings of the SIGMOD '98 Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, Seattle, WA, USA, 1–4 June 1998. [[CrossRef](#)]
31. Ooi, B.C.; Tan, K.L.; Yu, C.; Bressan, S. Indexing the edges—A simple and yet efficient approach to high-dimensional indexing. In Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, Dallas, TX, USA, 15–18 May 2000. [[CrossRef](#)]

32. Xu, Y.; Lee, W.C. Window query processing in highly dynamic geosensor networks: Issues and solutions. *Geosens. Netw.* **2004**, *31*–52. [[CrossRef](#)]
33. Xu, Y.; Lee, W.C.; Xu, J.; Mitchell, G. Processing window queries in wireless sensor networks. In Proceedings of the 22nd International Conference on Data Engineering, Atlanta, GA, USA, 3–7 April 2006. [[CrossRef](#)]
34. Ku, W.S.; Zimmermann, R.; Wang, H. Location-based spatial query processing with data sharing in wireless broadcast environments. *IEEE Trans. Mob. Comput.* **2008**, *7*, 778–791. [[CrossRef](#)]
35. Wang, H.; Deng, J.; Yuan, G. Window query and analysis on massive spatio-temporal data. *IERI Procedia* **2014**, *10*, 138–143. [[CrossRef](#)]
36. Bulusu, N.; Heidemann, J.; Estrin, D. GPS-less low-cost outdoor localization for very small devices. *IEEE Pers. Commun.* **2000**, *7*, 28–34. [[CrossRef](#)]
37. Wang, Y.C.; Hu, C.C.; Tseng, Y.C. Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks. In Proceedings of the First International Conference on Wireless Internet, Budapest, Hungary, 10–15 July 2005. [[CrossRef](#)]
38. Heidemann, J.; Ye, W.; Wills, J.; Syed, A.; Li, Y. Research challenges and applications for underwater sensor networking. In Proceedings of the Wireless Communications and Networking Conference, Las Vegas, NV, USA, 3–6 April 2006. [[CrossRef](#)]
39. Walt, S.V.D.; Colbert, S.C.; Varoquaux, G. The NumPy array: A structure for efficient numerical computation. *Comput. Sci. Eng.* **2011**, *13*, 22–30. [[CrossRef](#)]
40. Murphy, B.S. PyKrig: Development of a Kriging Toolkit for Python. In Proceedings of the 2014 AGU Fall Meeting, San Francisco, CA, USA, 15–19 December 2014.
41. Meyer, T.H. The discontinuous nature of Kriging interpolation for digital terrain modeling. *Cartogr. Geogr. Inf. Sci.* **2004**, *31*, 209–216. [[CrossRef](#)]
42. Fick, S.E.; Hijmans, R.J. WorldClim 2: New 1-km spatial resolution climate surfaces for global land areas. *Int. J. Climatol.* **2017**, *37*, 4302–4315. [[CrossRef](#)]

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).