

Iso-Map: Energy-Efficient Contour Mapping in Wireless Sensor Networks

Mo Li, *Member, IEEE*, and Yunhao Liu, *Senior Member, IEEE*

Abstract—Contour mapping is a crucial part of many wireless sensor network applications. Many efforts have been made to avoid collecting data from all the sensors in the network and producing maps at the sink, which is proven to be inefficient. The existing approaches (often aggregation based), however, suffer from heavy transmission traffic and incur large computational overheads on each sensor node. We propose Iso-Map, an energy-efficient protocol for contour mapping, which builds contour maps based solely on the reports collected from intelligently selected “isoline nodes” in wireless sensor networks. Iso-Map achieves high-quality contour mapping while significantly reducing the generated traffic from $O(n)$ to $O(\sqrt{n})$, where n is the total number of sensor nodes in the field. The pernode computation overhead is also restrained as a constant. We conduct comprehensive trace-driven simulations to verify this protocol, and demonstrate that Iso-Map outperforms the previous approaches in the sense that it produces contour maps of high fidelity with significantly reduced energy cost.

Index Terms—Distributed applications, query processing, terrain mapping, wireless sensor networks.

1 INTRODUCTION

RECENT advances in wireless communication and microsystem techniques have resulted in significant developments of wireless sensor networks (WSNs). A sensor network consists of a large number of low-power, cost-effective sensor nodes that interact with the physical world [5], [7], [10]. The increasing studies of wireless sensor networks aim to enable computers to better serve people by using instrumented sensors to automatically monitor the physical environment.

Contour mapping has been widely recognized as a comprehensive method to visualize sensor fields [8], [11], [14]. A contour map of an attribute (e.g., height) shows a topographic map that displays the layered distribution of the attribute value over the field. It often consists of a set of contour regions outlined by isolines of different isolevels. Fig. 1 plots a section of underwater depth measurement and the corresponding isobath contour map.

For many applications, contour mapping provides background information for the sink to detect and analyze environmental happenings in a global view of the features in the field. Such a view is often difficult to achieve by individual sensor nodes with constrained resources and insufficient knowledge.

A naive approach for contour mapping is to collect sensory data from all the sensors in the monitored field and then construct the contour map at the sink. Obviously, delivering a huge amount of data back to the sink incurs heavy traffic, which rapidly depletes the energy of sensor

nodes. To address this problem, several aggregation based protocols have been proposed [15], [27], [28]. These protocols aggregate data with similar readings at intermediate nodes, reducing the traffic overhead up to 40 percent [27]. We believe the aggregation based protocols cannot further improve the scalability of the network based on the following observations. First, as long as all sensors are required to report to the sink, the number of generated reports is always $O(n)$, where n is the total number of sensor nodes. Second, the aggregation operations insert a heavy computation overhead to the intermediate nodes. For example, INLR [27] requires each intermediate node to carry out multiple integrals in order to estimate the similarity of two contour regions.

In order to address the inherent limitations of aggregation based approaches, we propose Iso-Map. By intelligently selecting a small portion of the nodes to generate and report data, Iso-Map is able to construct contour maps with comparable accuracy while significantly reducing network traffic and computation overhead. Although the basic idea beyond Iso-Map is comprehensible, several challenges exist in its design. For example, partial utilization of the network information reduces the network traffic, but naturally leads to the degradation of the mapping fidelity. Thus, careful node selection policies and an effective algorithm to recover the contour map from the partial information are necessary. We also need to balance the trade off between the traffic savings and the mapping fidelity. In addition, we aim to avoid heavy computational overhead in the intermediate nodes so that the design is scalable for resource constrained sensor devices.

The major contributions of this work are as follows: 1) We design a novel algorithm to construct contour maps from a critical set of nodes, which we call *isoline nodes*. By restraining the traffic generation within the isoline nodes, Iso-Map significantly reduces the network traffic while still constructing high-quality contour maps that are comparable to the best ones ever achieved through existing protocols.

• The authors are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. E-mail: {limo, liu}@cse.ust.hk.

Manuscript received 11 Aug. 2008; revised 17 Feb. 2009; accepted 30 May 2009; published online 29 June 2009.

Recommended for acceptance by M. Garofalakis.

For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2008-08-0417. Digital Object Identifier no. 10.1109/TKDE.2009.157.

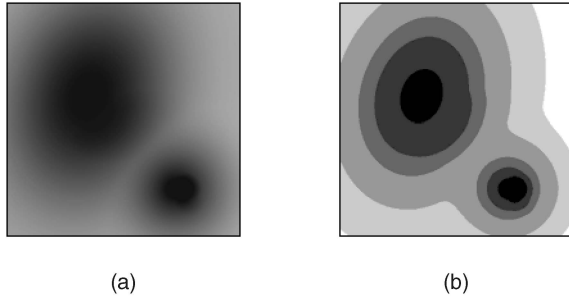


Fig. 1. Contour mapping. (a) A section of underwater depth measurement and (b) the isobath contour map of (a).

Our analysis proves that, Iso-Map reduces the traffic generation from $O(n)$ of existing protocols to $O(\sqrt{n})$, which substantially suppresses the traffic flows across the network. 2) By employing local measurement and lightweight in-network filtering, the pernode computational overhead is constrained as a constant and does not grow with the network size. 3) We conduct a field study on a practical Iso-Map application, and based on the collected real world data, we conducted a trace-driven simulation which confirms the superior performance of Iso-Map compared with existing protocols. Another strength of this design is that Iso-Map is orthogonal with many other designs, enabling further traffic savings to be achieved together with other approaches.

The remainder of this paper is organized as follows: Section 2 introduces our investigation of a practical Iso-Map application. Section 3 presents the Iso-Map design, illustrating the flow of its operations. In Section 4, we mathematically analyze the communicational and computational overhead of Iso-Map and compare with that of previous protocols. We present simulation results and evaluate the performance of Iso-Map in comparison with other protocols in Section 5. In Section 6, we discuss the related work and we conclude this work in Section 7.

2 APPLICATION SCENARIO

We conducted a field study on Huanghua Harbor, which is currently the second largest harbor of coal transportation in China. It has experienced rapid development over the past five years, and its coal transporting capability has increased from 1.6 million tons per year in 2002 to 6.7 million tons per year in 2006. However, Huanghua Harbor currently suffers from the increasingly severe problem of the silted sea route. As illustrated in Fig. 2, Huanghua Harbor has a sea route that is 19 nautical miles long and 800 m wide at the entrance, including an inner route and an outer route. The sea route is designed to have a water depth of 13.5 m to allow for the passage of ships that weigh over 50 thousand tons. Since the sea route has been in operation, it has always been threatened by the movement of silt from the short sea area within 14 nautical miles outside the route entrance. In the event that the sea route is silted up, ships of large tonnages must wait to prevent grounding, and ships of small tonnages need be piloted into the harbor. Monitoring the extent of siltation reliably is critical in order to ensure the safe operation of Huanghua Harbor.

The uncertainty and the high instantaneous intensity of the siltation make monitoring the extent of siltation

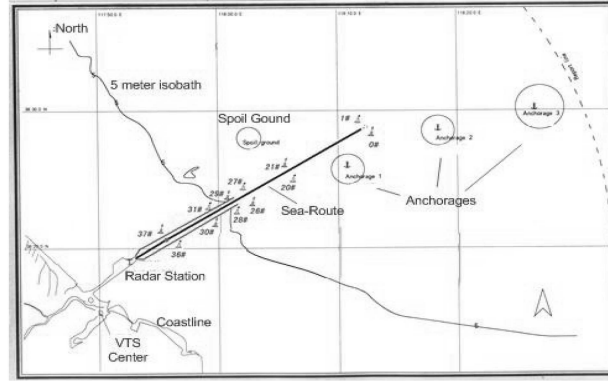


Fig. 2. The monitoring field of Huanghua Harbor.

extremely expensive and difficult. The amount of siltation in Huanghua Harbor is affected by many factors, among which tide and wind blow are the most dominating. While the tide produces a periodical influence on the movement of silt, the sudden blowing of wind brings more incidental and intensive influences. For example, records show that strong winds with wind forces of 9 to 10 on the Beaufort scale hit Huanghua Harbor from 10th Oct. to 13th Oct. in 2003. The stormy tide brought a siltation of 970,000 m^3 to the sea route, which suddenly decreased the water depth from 9.5 m to 5.7 m and blocked most of the ships weighing more than 35 thousand tons. The harbor administration hired three boats equipped with active sonars to cruise the 380 km^2 short sea area around the harbor for several days, creating underwater contour maps for ships to find possible pathways and to set future cleaning plans. According to the record, drawing the underwater contour maps cost more than 18 million US dollars per year. Even so, the monitoring granularity is low in terms of time and space, especially under stormy weather conditions, which creates intensive siltation and prevents boats from routine cruising.

We propose to deploy an echolocation sensor network on the sea surface to continuously monitor the water depth of the sea route. The sensor nodes can be deployed with buoys and tied with ropes to the bottom of the sea (as illustrated in Fig. 3). The precise depth measurement at each spot is not needed. Instead, Iso-Map can be utilized to build an isobath contour map to visualize the depth level of the sea area. The contour map depicts the contour sea zones above different depth levels. Based on this contour map, we can easily guide ships of different tonnages. With the map, we can

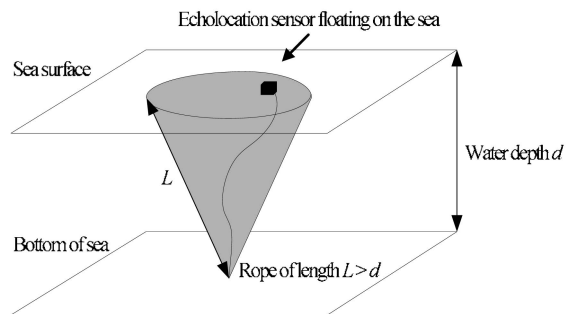


Fig. 3. The sensor node deployed on the sea surface.

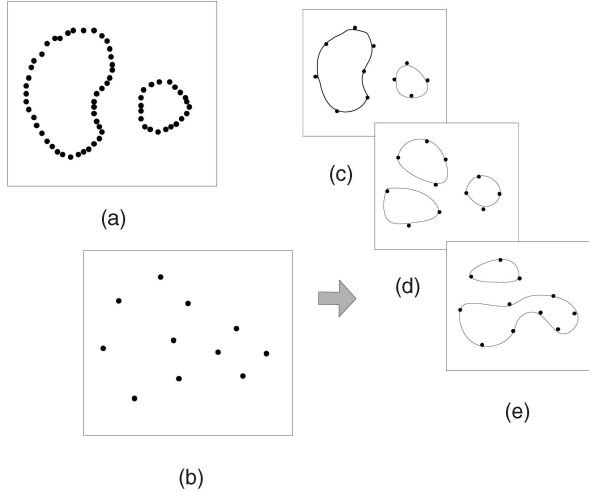


Fig. 4. Contour mapping from isoline nodes. (a) Dense deployment of sensor nodes leads to the isolines. (b) Sparse deployment of sensor nodes provides ambiguous information. (c), (d) and (e) Three possible contour maps of (b).

also clearly locate the dangerous areas where the water depth is under alarm thresholds. The method of contour mapping by the sensor network significantly eases the task of siltation monitoring and reduces the expenses. Using less than 1 million US dollars, we can afford to deploy more than 40,000 sensor nodes over the 380 km² sea area, with a density of one sensor node per 100 m × 100 m. We are currently launching this project, and all data used in the simulations are from the real world records.

3 ISO-MAP DESIGN

The basic idea of Iso-Map is to create the contour map based on a selected set of nodes, known as the *isoline nodes*. Isoline nodes are the sensor nodes residing on the isolines around contour regions. A more formal definition of isoline node will be given later. Intuitively, since isoline nodes correspond to the perimeter of contour regions, the number of reports from isoline nodes can be largely restricted compared with the network size. Later, we mathematically show that the traffic generated from isoline nodes is at the level of $O(\sqrt{n})$, where n is the total number of nodes in the monitored field.

It is not, however, trivial to construct the contour map based solely on isoline nodes' reports. Ideally, as illustrated in Fig. 4a, when sensor nodes are densely deployed, the positions of isoline nodes clearly outline the contour regions. In more practical scenarios, however, sensor nodes are usually deployed sparsely, as shown in Fig. 4b, in which the positions of isoline nodes provide only discrete "isopositions." We cannot deduce how the isolines pass through these positions. For example, based on the data illustrated in Fig. 4b, the sink can interpret into different contour maps, such as the ones shown in Figs. 4c, 4d, and 4e.

In this section, we will first introduce the major operations of Iso-Map including building network architecture, query dissemination and isoline node appointment, isoline node measurement, and contour map generation, and then discuss the in-network filtering for further traffic reductions.

3.1 Building Network Architecture

Iso-Map first builds the routing structure in the sensor network, through which the sink insert queries into the network and collects reports. Although we do not rely on any particular underlying network architecture, for this work, we assume a tree-based routing scheme [13] that is adopted in many systems [8]. We believe that assuming a concrete underlying networking strategy helps us clearly state the idea, providing a fair platform for the comparison of performance between different approaches. In the tree-based routing scheme, a spanning tree rooted at the sink is constructed over the communication graph. Each node is assigned a level, which specifies its hop count distance from the sink. The parent node is one level lower than its children nodes. Nodes in different levels forward packets during different time slots. Topology maintenance mechanisms can be employed [13], which allow each node to dynamically choose a parent from its neighboring nodes based on the quality of communication. MAC layer reliability of node transmissions can be easily added into this framework [18], [20].

3.2 Query Dissemination and Isoline Node Appointment

Initially, the sink disseminates a query through the routing tree for contour mapping over the targeted field. The query message specifies the data space $[\nu_L, \nu_H]$ and the granularity T of the contour map, which specifies the desired isolines in the contour map with the isolevels $\nu_i = \nu_L + i \cdot T \in [\nu_L, \nu_H]$. Upon receiving this query, each sensor node accordingly determines whether it is an isoline node.

Definition 3.1. A sensor node p (with sensing value ν_p) is an *isoline node* if and only if: 1) its sensing value is within a predefined border region of the isolevel ν_i specified in the query, i.e., $[\nu_i - \varepsilon, \nu_i + \varepsilon]$, and 2) one of its neighboring nodes q has a sensing value ν_q , where ν_i is between their sensing values, i.e., $\nu_p < \nu_i < \nu_q$, or $\nu_q < \nu_i < \nu_p$. The satisfying node has the isolevel of ν_i .

Based on Definition 3.1, a node only incurs local operations within its neighborhood. It first appoints itself as a candidate isoline node if its sensing value falls into the border region of the query. Then, the candidate isoline node checks its local neighborhood and identifies itself as an isoline node if the second condition is satisfied. The two conditions guarantee that the isoline node is close to the isoline in terms of value and space. Apparently, a larger tolerance on the border region of the sensing value specified by ε will broaden our selection of isoline nodes, yet lead to unexpected errors on the mapped isolines. Normally, ε is selected as a fraction of the isoline granularity T . In our later analysis and experiments, ε is selected as $0.05 \cdot T$. Nevertheless, we leave such a parameter adjustable by concrete applications.

3.3 Isoline Node Measurement

Once the isoline nodes are appointed, they make local measurements and generate reports to send back to the sink. Each isoline node generates a 3-tuple report $r = \langle \nu, p, d \rangle$, in which ν represents the isolevel of the node, p represents the position of the sensor node, and d

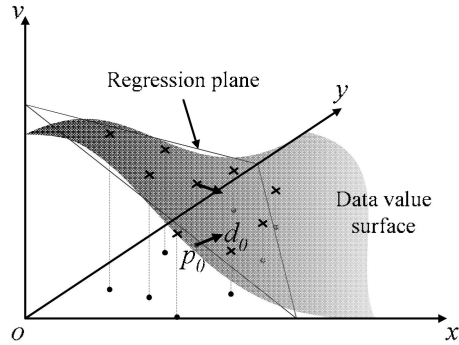


Fig. 5. Linear regression for spatial data modeling.

represents the gradient direction of the attribute value at the sensor node. Clearly, the isolevel v can be obtained when the node determines that it is an isolevel node, and the position p can be obtained either from attached localization devices such as a GPS receiver or by one of existing algorithms [6], [16], [25]. However, as illustrated in Fig. 4, having only p and v is often not sufficient for the sink to construct the contour map. To address this problem, we introduce the new parameter gradient direction d .

Each isolevel node performs local modeling on sensing values within its neighborhood and obtains an estimation of the gradient direction d . The spatial data value distribution is mapped into the (x, y, v) space, where the coordinate (x, y) represents the position and $v = f(x, y)$ describes the distribution surface of the data value in this space. The gradient direction d denotes the direction where the data value most degrades in the space. The vector d is calculated by:

$$d = -\text{grad}(f) = -\nabla f = -\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)^T. \quad (1)$$

To estimate the gradient direction d , an isolevel node first needs to approximate the local data map. To build the local data map in this design, each isolevel node sends queries to its neighboring sensor nodes for their positions and sensory values. The query scope can be adjusted within k -hop neighbors for different sensor deployment densities or to achieve different levels of estimation precision. Upon receiving the $\langle v, p \rangle$ tuples from neighboring nodes, the isolevel node approximates the local data map through regression analysis. Indeed, many regression models can be employed to construct the approximated data value surface on the local data map, among which linear regression is a simple and widely used one. The computational simplicity of the linear regression model makes it a natural choice for the resource constrained sensor devices.

Fig. 5 illustrates the rationale of how the isolevel node performs the linear regression and approximates the data value surface with the regression plane. Without loss of generality, we assume the isolevel node position is $p_0(x_0, y_0)$ and the sensory value is v_0 . The positions of its n neighboring sensors are $p_1(x_1, y_1), p_2(x_2, y_2), \dots, p_n(x_n, y_n)$ and the sensory values are v_1, v_2, \dots, v_n , respectively.

A linear model $v = L(x, y) = c_0 + c_1x + c_2y$ describes the regression plane of the $n + 1$ points in the data value

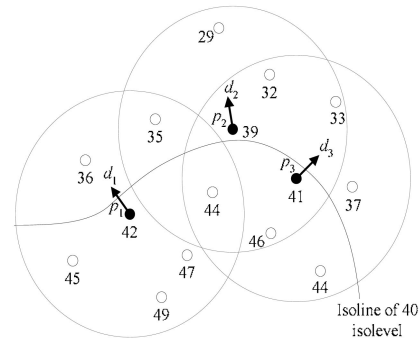


Fig. 6. The example of calculated gradient directions of three isolevel nodes.

space built on (x, y, v) . With the $n + 1$ points $(x_0, y_0, v_0), (x_1, y_1, v_1), \dots, (x_n, y_n, v_n)$, the isolevel node computes the coefficients of the linear model by solving the equation:

$$Aw = b, \quad (2)$$

where

$$A = V^T V = \begin{pmatrix} 1 & \sum_{i=0}^n x_i & \sum_{i=0}^n y_i \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i y_i \\ \sum_{i=0}^n y_i & \sum_{i=0}^n x_i y_i & \sum_{i=0}^n y_i^2 \end{pmatrix},$$

$$V = \begin{pmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{pmatrix},$$

$$b = V^T \lambda = \begin{pmatrix} \sum_{i=0}^n v_i \\ \sum_{i=0}^n x_i v_i \\ \sum_{i=0}^n y_i v_i \end{pmatrix}, \lambda = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_n \end{pmatrix} \text{ and } w = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix}.$$

With the obtained plane of linear model approximation $v = L(x, y)$, the isolevel node can calculate its gradient by introducing this approximation into (1):

$$d_0 = -\left(\frac{\partial L}{\partial x}, \frac{\partial L}{\partial y}\right)^T \Big|_{p_0} = -(c_1, c_2)^T. \quad (3)$$

Fig. 6 plots an example where the isolevel nodes are at the isolevel of 40. Each isolevel node calculates the gradient direction from the regression within its neighborhood. We mark the calculated gradient directions in the figure. The calculated gradient direction of each isolevel node reflects the local trend of data spatial variation and it well approximates the normal direction of the isolevel passing by. Fig. 7 shows the statistics on the error between the calculated gradient direction and the normal direction of isolevels. As the average node degree increases, the error drops rapidly.

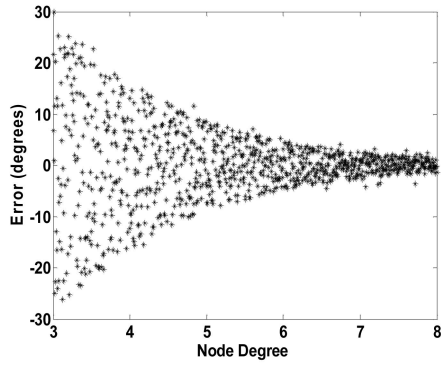


Fig. 7. The error between calculated gradient direction and the normal direction of isolines.

Note that generally, for a random deployment of sensors, a connected WSN results in an average node degree at least above 7 [1]. As shown in Fig. 7, this suppresses the error to within $\pm 5^\circ$. Later, the sink will utilize this parameter to measure the local features of isolines.

3.4 Contour Map Generation

Upon receiving isoline node reports, the sink constructs the contour map which is delineated by isolines of different isolevels, say $\nu_i = \nu_L + i \cdot T \in [\nu_L, \nu_H]$. The sink separately constructs isolines of different isolevels, and the contour regions reside between them.

When constructing isolines of the isolevel ν_i , the sink utilizes the reports with isolevel ν_i from the isoline nodes residing along the isolines of ν_i . Since the data gradient direction d at each reported position approximates the normal direction of isolines, it helps to construct local segments of isolines. Fig. 8a shows that isoline nodes of the same isolevel report to the sink and Fig. 8b depicts the reported isopositions and corresponding gradient directions. The sink first builds a Voronoi diagram for the set of

isopositions, as shown in Fig. 8c. The Voronoi cell specifies the affecting area of each isoposition, where the sink constructs the local isoline segment according to the gradient direction d at that isoposition. For each cell, a straight line passing the isoposition and perpendicular to its gradient direction d is drawn. It intersects with cell borders and partitions the cell into two parts. The part in the gradient direction is the outer part and the opposite one is the inner part. The separating line acts as a local boundary in each Voronoi cell, which we call the type-1 boundary. The sink then merges the inner parts in different Voronoi cells and complements the boundaries to separate contour regions from outer area. The complementary boundaries along the cell borders are called type-2 boundaries. Fig. 8d illustrates this step. As shown, after this step, well-approximated contour regions are outlined by the concatenated local boundaries, though it appears a bit rough.

The sink then regulates the approximation by smoothing the pinnacles based on the following two rules. *Rule 1.* The type-1 boundary is prolonged at the end where it intersects with a type-2 boundary and their internal angle is within (180 degree, 270 degree). If it intersects with the type-1 boundary in the adjacent Voronoi cell, the pinnacle area outside of it should be removed and accepted as the new boundary. Otherwise, no change is made. *Rule 2.* The type-1 boundary is prolonged at the end where it intersects with a type-2 boundary and their internal angle is within (90 degree, 180 degree). If it intersects with the type-1 boundary in the adjacent Voronoi cell, the concave area inside of it should be included and accepted as the new boundary. Otherwise, no change is made. Fig. 8e illustrates how the two rules are applied to regulate the approximation. The regulation process under the two rules substantially achieves better readjustments on the affecting area of each isoposition and makes a tighter approximation. The approximated isolines that are eventually obtained are shown in Fig. 8f.

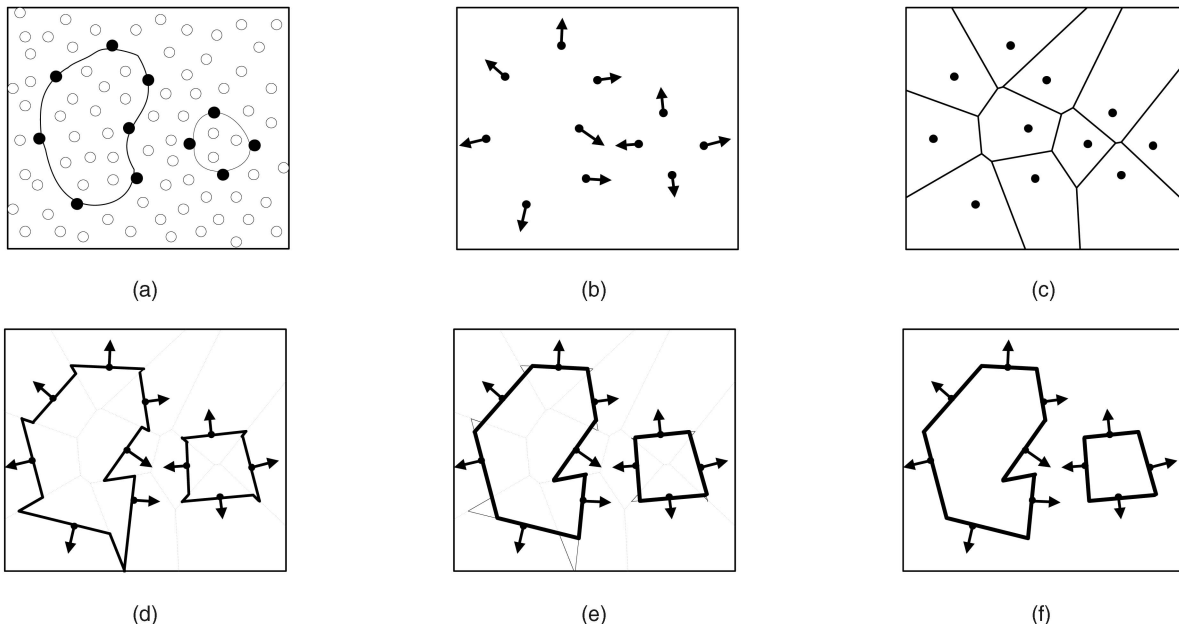


Fig. 8. Illustration of the process of contour boundary deduction.

When building the isolines of different isolevels, the sink initially builds isolines of the lowest isolevel, and the isolines of isolevel ν_L restrict the boundaries for all contour regions above the isolevel ν_L . When the contour regions of higher isolevels intersect with such a boundary, only the area inside the boundary is kept. Based on this recursive rule, Isolines are then sequentially constructed according to their isolevels.

3.5 In-Network Filtering

Until now, we have seen that Iso-Map constructs contour mapping with reduced message reporting. Now, we show how to further make trade off between traffic overhead and mapping precision.

We note that the precision of the contour mapping is related to the density of isoline node reports. As we previously mentioned, Iso-Map provides contour mapping with acceptable fidelity even when sensor nodes are sparsely deployed. When the network has a high density and we do not have special requirements on the mapping precision, it is not necessary to deliver all isoline node reports at a cost of heightened traffic overhead. Iso-Map employs in-network filtering in the routing process to control the report density and aims to achieve an optimum.

A parameterized method is used in the in-network filtering process. When the intermediate node receives the reports from its descendant nodes, it investigates the relationship between reports of different isoline nodes. Two parameters, *angular separation* (s_a) and *distance separation* (s_d) are utilized to evaluate the relationship between two different reports, where s_a describes the angular separation between the gradient d in the two reports and s_d describes the distance separation between the positions of the two reports. The intermediate node calculates the two parameters for each pair. If both s_a and s_d are smaller than the predetermined threshold values, one of the reports is considered redundant and dropped. Thus, the predefined threshold values act as a filter to control the report density. More tolerant thresholds lead to smaller traffic cost but result in a lower fidelity of the approximations. Such a filtering process is recursively applied to all the generated reports along the paths where they are forwarded to the sink. Intermediate nodes store and compare the filtered isoline reports from their descendant nodes. In Section 5, our simulation gives an analysis on the setting of the two parameters.

By introducing the evaluation of angular separation s_a , Iso-Map adjusts the report density without violating the uniformity of the reports. The isopositions along an isoline are filtered evenly according to their gradient directions. Thus, the degradation of precision on the constructed contour map is evenly distributed along the contour boundaries without breakages at extreme points. Fig. 9a and 9b compare the contour regions built under different report densities. Note that, although more reports better help the sink construct a precise contour map, evenly filtering some of the reports indeed does not degrade the result by much.

4 DISCUSSION

Iso-Map utilizes the reports from isoline nodes to construct contour maps. Compared with existing works which rely on

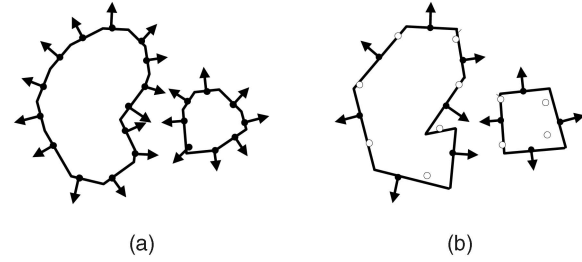


Fig. 9. The contour regions built under different report densities.

the aggregation of sensory readings from all nodes in the field, Iso-Map largely restrains the scale of sensor reporting. We will first conduct a theoretical analysis on the incurred traffic scale and prove that Iso-Map reduces the number of reports from $O(n)$ to $O(\sqrt{n})$. Such suppression on data generation dramatically reduces the traffic overhead across the network as one reporting source indeed brings many hop by hop data deliveries along its routing path to the sink. We further show that Iso-Map considerably reduces the computational overhead introduced to the nodes. Indeed, Iso-Map outperforms existing approaches in terms of both communicational and computational complexity.

4.1 Network Traffic

To study the network traffic incurred by Iso-Map, we first simplify our analysis to a continuous domain, where sensor nodes cover the field with infinite density. The isoline nodes are then represented by continuous isolines. We prove that the total length of a constant number of isolines is $O(n^{1/2})$, given that all isolines are “well behaved” and do not intersect each other. It is natural that different isolines do not intersect each other due to the principle of contour mapping. We impose the constraint of “well behaved” curves as [2] did to exclude some pathologically shaped “monster curves” such as Peano’s space-filling curves, which hardly emerge as contour boundaries in practice [21].

Definition 4.1. A curve is well behaved if for square box of any side x that intersects the curve, the length of the curve inside the box is less than cx for some constant $c > 1$.

The definition is equivalent to observing that the curve has a Hausdorff dimension of 1 [4]. In practice, most of the nonbizarre curves have Hausdorff dimensions of 1. Such a definition directly leads to the following observation. For any constant number K isolines within an $n^{1/2} \times n^{1/2}$ square area, the total of their lengths L is less than $cn^{1/2}$ which is of $O(n^{1/2})$ size. Now, we extend our analysis into a more practical scenario, where sensor nodes are uniformly deployed over the square field in a discrete manner. We assume that the density of nodes is p , and each isoline triggers a stripe of isoline nodes along it with a small width of ε (ε corresponds to the node communication radius, which is small enough compared with the size of the field). In fact, the continuous scenario discussed above is an extreme case of this when $p \rightarrow \infty$ and $\varepsilon \rightarrow 0$.

Theorem 4.1. For any constant number K contour regions within a square area of n sensor nodes, the number of isoline nodes is $O(n^{1/2})$.

Proof. The side of the square area is calculated to be $(n/p)^{1/2}$.

We then snatch the K isolines from the K contour regions. As shown according to Definition 4.1, the total length L of the K isolines is $O((n/p)^{1/2}) = O(n^{1/2})$. The area of the stripe is approximated by the path integral through these isolines:

$$S = \sum_{i=1}^K \int_{L_i} \varepsilon ds = \varepsilon \cdot \sum_{i=1}^K L_i = \varepsilon L. \quad (6)$$

According to (6), the number of isoline nodes scattered in the stripe is thus $p \cdot S = O(n^{1/2})$. \square

According to Theorem 4.1, the generated traffic from isoline nodes is thus limited to $O(n^{1/2})$.

4.2 Computational Overhead

We analyze the computational overhead of 1) the isoline nodes for local measurements on the 4-tuple parameters and 2) the intermediate nodes which carry out in-network filtering to reduce the traffic of reports.

The local measurements conducted by each isoline node require only local information within the neighborhood. The computational overhead is bounded by the node degrees. From the calculating process described in Section 3.3, we observe that the main computational workload comes from solving the regression equation of (2) which indeed incurs $O(deg)$ calculations, where deg is the average degree of each node in the network. Therefore, the total computational overhead among all isoline nodes is bounded by $O(deg \cdot n^{1/2})$.

The intermediate nodes which forward the isoline node reports normally simply relay the reports without any computational workload, except with in-network filtering. They will compare reports from different children nodes and drop the likely redundant ones. Each comparison between two reports incurs the calculation of their s_a and s_d values. If we focus on each generated isoline node report, it will be compared at most once with each of the other reports before it is delivered to the sink, regardless of which intermediate nodes these comparisons are carried out at. Thus, the computational overhead within the forwarding network is bounded by $O(N_{rep}^2) = O(n)$, where N_{rep} refers to the number of isoline node reports and is $O(n^{1/2})$ according to the analysis in the previous section. Combining the above two parts, the computational overhead within the entire network is $O(deg \cdot n^{1/2} + n) = O(n)$.

4.3 Comparison with Existing Approaches

In this section, we draw a comparative study with existing approaches. TinyDB [8] is the first work targeting the application of contour mapping. In its aggregate-free version, all sensor nodes are required to report and a simple algorithm is employed without data aggregation. In TinyDB, the number of sensor reports is n and the computation within the network is proportional to the network size, $O(n)$. The eScan [28] creates the residual energy map based on the aggregation of all sensor node reports. Thus, the number of sensor reports is also n . The aggregation algorithm provided in eScan merges different scans with $O(n^3)$ operations in the worst case for each sensor, so the total amount of computation within the network is bounded by $O(n^4)$. INLR [27] requires sensor

TABLE 1
Overhead Comparison of Different Approaches

Approach	Traffic Generation	Network Computation	Sensor Deployment
TinyDB	$O(n)$	$O(n)$	Grid
eScan	$O(n)$	$O(n^4)$	Free
INLR	$O(n)$	$\Omega(n^{1.5})$	Grid
Data Suppression	$O(n)$	$\Omega(nd)$	Grid
Iso-Map	$O(\sqrt{n})$	$O(n)$	Free

reports from all nodes for the in-network contour map construction. By the model based partial map aggregation, the network computational overhead of INLR reaches at least $\Omega(n^{1.5})$. The data suppression protocol [15] requires a subset of node reporting which is proportional to the total number of nodes in the network, so the generated traffic is $O(n)$. Each node is required to measure the data similarity with its 2-hop neighbors, so the computational overhead in the network is no less than $\Omega(nd)$, where d is the node degree of the 2-hop neighborhood.

Table 1 summarizes and compares Iso-Map with the four existing approaches. Iso-Map incurs the lowest traffic cost and network computation when performing contour mapping. Note that among the five approaches, only the Iso-Map and eScan protocols have no requirement on the sensor deployment. The TinyDB, INLR, and Data Suppression protocols basically rely on a regular deployment of sensor nodes into grids. They use sink interpolation to deal with irregular node deployment, which potentially degrades the fidelity of the resulting contour map.

5 PERFORMANCE EVALUATION

We implemented the Iso-Map protocol and conducted trace driven simulations to evaluate its performance. We utilized a real map of underwater depth as our testing data which is obtained from sonar measurements in Huanghua Harbor. Basically, n sensor nodes are uniformly deployed to monitor the depth values over a normalized $n^{1/2} \times n^{1/2}$ surveillance field with a density of 1. The radio range of sensor nodes determines the average degree of each node. Experimentally, we find that to keep a connected communication graph, the radio range should be no less than 1.5, which results in an average node degree of 7. This corresponds to a reasonable deployment of one node per 400 m² in practice, if we set up a 30 m radio range for the MICA2 motes [9]. Perfect link layer is assumed in this simulation, in which the data delivery is guaranteed through performance based routing dynamics [13], [26] and MAC layer retransmissions [18], [20]. For our Iso-Map approach, we select the border range of isoline value ε to be 0.05T, i.e., 5 percent of the value range between two consecutive isolevels. We first evaluate the produced fidelity of Iso-Map under various settings. Then we study the network overhead incurred by Iso-Map on the construction of the contour map, including communicational overhead as well as computational overhead. Finally, we bridge the

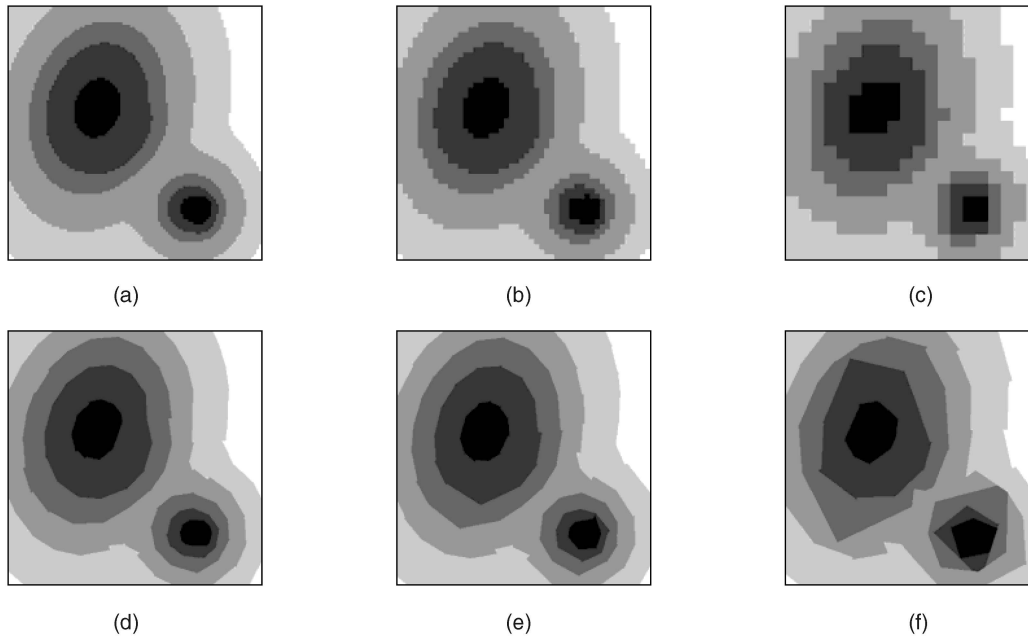


Fig. 10. Performance of isobath contour mapping. (a)-(c): The contour maps created by TinyDB algorithm, under different normalized densities of sensor nodes (4, 1, and 0.16); (d)-(f): the contour maps created by Iso-Map, under different normalized densities of sensor nodes (4, 1, and 0.16).

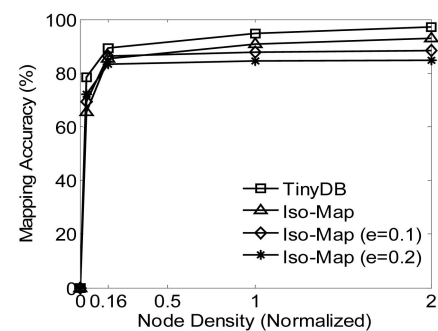
network overhead with energy consumptions of sensor nodes and evaluate the energy efficiency.

We utilize a $400\text{ m} \times 400\text{ m}$ section of the underwater depth measurement as our testing data (refer to Fig. 1 for the measurement and its contour map). We compare the resulting fidelity of Iso-Map with that of TinyDB, which achieves the best fidelity compared with all other existing approaches. Since the TinyDB protocol requires a grid deployment of sensor nodes, when simulating the TinyDB protocol, we deploy the sensor nodes into grids instead of randomly. For both approaches, node density is the dominating factor affecting the fidelity of the contour mapping. Thus, we simulate different node densities of deployment to reflect the impact. We study the cases with 400 nodes, 2,500 nodes and 10,000 nodes separately. If we normalize the field size to be 50×50 units, the normalized node densities are 0.16, 1, and 4, respectively. In practice, all three cases correspond to reasonable node densities for different applications requiring more or less surveillance precision.

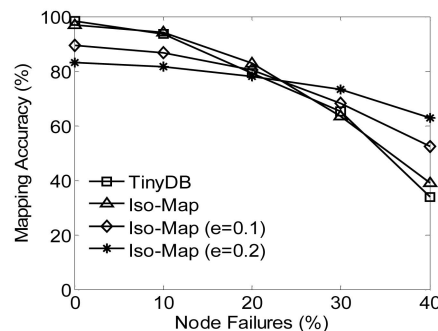
Figs. 10a, 10b, 10c depict the resulting contour maps of TinyDB under the above node densities. Figs. 10d, 10e, 10f depict the resulting contour maps of Iso-Map. For the Iso-Map protocol, we choose the two parameters *angular separation* $s_a = 30$ degree and *distance separation* $s_d = 4$ for in-network filtering. The isoline reports received at the sink are 112, 89, and 49. The number of received reports is not linear to the node density since in-network filtering helps raze out most of the redundant reports, especially under dense node reporting. Clearly, both approaches degrade in precision as the node density decreases, but both still produce acceptable fidelity maps.

Fig. 11a plots how the mapping accuracy is affected by the deployed node density. Here the mapping accuracy is measured as the ratio of the accurately mapped area in the resulting contour map to the whole area. The normalized density of 1 corresponds to deploying 2,500 nodes in the

$400\text{ m} \times 400\text{ m}$ field. The mapping accuracy of both TinyDB and Iso-Map rapidly jumps to a high level above 80 percent as the deployed node density increases. In all cases, Iso-Map is slightly below TinyDB but with comparable accuracy. We also compare the different settings for the ϵ value that determine different border range of isolines. The result shows that a rough border range definition helps to select



(a)



(b)

Fig. 11. Contour mapping accuracy against (a) node density and (b) node failures.

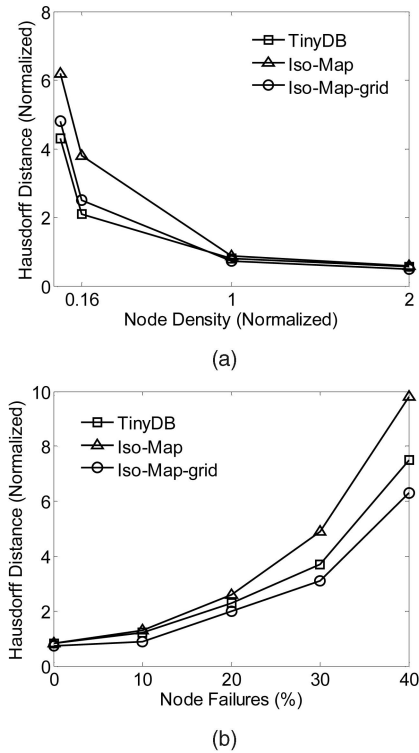


Fig. 12. Hausdorff Distance between the real isolines and estimated isolines against (a) node density and (b) node failures.

adequate isoline nodes when the node density is low, leading to better fidelity in such cases. However, when the network has enough node density, such a setting leads to worse fidelity due to the errors on the isolevel measurement.

Fig. 11b shows that the accuracy of both two protocols degrades as the ratio of node failures increases. TinyDB employs sink interpolation to recover the map from lossy isobars, which leads to the degradation of the accuracy. Iso-Map suffers from the loss of isoline node reports, which enlarges the distortion of mapped isolines. Overall, the two protocols perform similarly under node failures. More than 40 percent node failures make both of them unusable. Similar with the previous result, when the border range of isolines ε is large, the Iso-Map approach is more tolerable to the node failures because of the redundant isoline nodes selected. However, the best fidelity achievable is lowered down due to the errors on the isolevel measurement.

In Fig. 12, we use the Hausdorff Distance to evaluate the isoline accuracy. Hausdorff Distance [17] measures the maximum departure between two curves, thus providing an accuracy metric on the irregularity of the estimated isolines to the real ones. In Fig. 12, the Hausdorff Distance is normalized with the 50×50 unit field. Similar with Fig. 11, the irregularity of both two protocols grows intensive as the node density decreases and as the ratio of node failures increases. In this experiment, we test Iso-Map in both random and grid sensor deployments. We find that Iso-Map indeed benefits from the grid sensor deployment. Compared with the random sensor deployment, Iso-Map achieves a more regular output on the estimated isolines. The irregularity of the output becomes excessively intensive when the network is very sparse. In TinyDB, the irregularity is relatively stable, i.e., proportional to the grid size of

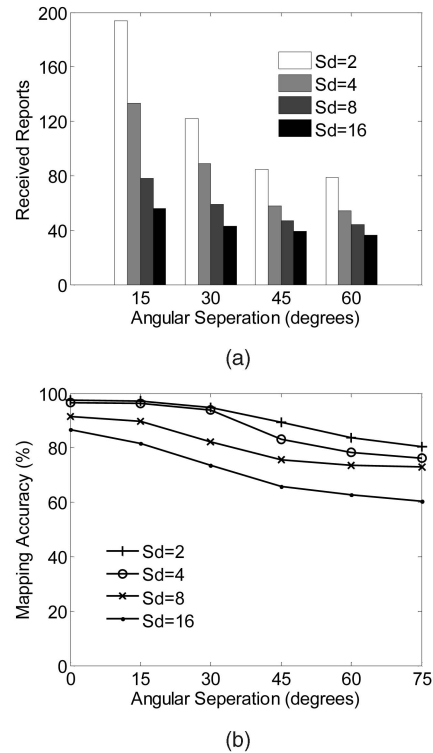


Fig. 13. Contour mapping accuracy against (a) node density and (b) node failures.

deployment. Thus as the density of sensors are decreased, such irregularity linearly increases (to the square root of the node density). However, TinyDB is more vulnerable with sensor failures. TinyDB is of higher error when both approaches are implemented on the grid deployment, especially when the failure rate is high.

5.1 Network Traffic Overhead

It is well known that the network traffic consumes the largest portion of the sensor energy and is considered the most important metric used to evaluate the energy efficiency of a WSN. In this section, we contrast Iso-Map with the most recent work INLR [27], as well as with the well-known TinyDB protocol.

We first investigate the impact of in-network filtering on the reduction of the number of reports. Fig. 13 plots how different settings of s_a and s_d result in different extents of filtering, where 2,500 nodes are scattered over the 50×50 field with a normalized density of 1. It is obvious that higher tolerances of s_a and s_d lead to larger reductions of the reports (see Fig. 13a) but with a lower mapping accuracy (see Fig. 13b). Such a feature provides Iso-Map with flexibility to trade accuracy with traffic. In later simulation runs, we choose the setting of $s_a = 30^\circ$ and $s_d = 4$, which achieves substantial savings of network traffic while keeping a high accuracy of contour mapping.

We vary the network diameter so that three protocols are simulated over the fields of different sizes. With a constant node density of 1, the network diameter varies from 10 to 50 hops. Each parameter in a report uses two bytes, such as the sensory value, position, gradient, etc. Fig. 14a plots the traffic overhead of the three protocols in terms of kilobytes. Consistent with the theoretical analysis, the traffic overhead

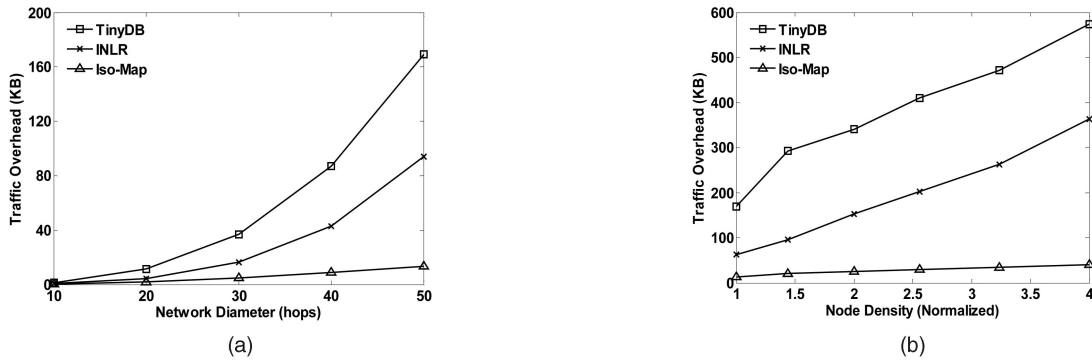


Fig. 14. Network traffic overhead against (a) network diameter and (b) node density.

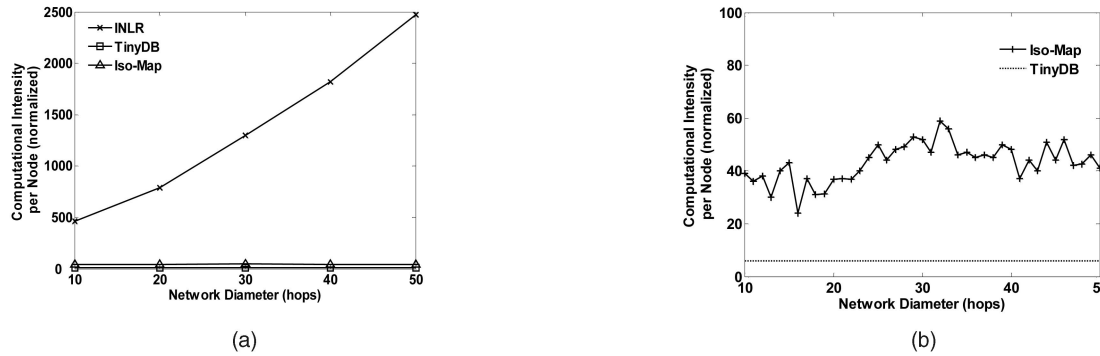


Fig. 15. The computational intensity against network diameter. (a) Comparison on three protocols and (b) an amplified view of Iso-Map.

incurred by TinyDB and INLR grows rapidly while Iso-Map mainly relies on the isoline node reports, imposing much less traffic.

We then vary the node density and again, we can see that Iso-Map outperforms TinyDB and INLR, as shown in Fig. 14b. Although all three protocols incur traffic overhead proportional to the node density, Iso-Map has a much smaller growing factor. The combinational view in Fig. 14 exhibits the dominating scalability of Iso-Map.

5.2 Node Computational Overhead

In the aggregation based protocols, intermediate nodes conduct heavy computations to aggregate different map segments. On the other hand, in the nonaggregation protocols, such as TinyDB, etc., reports are delivered to the sink without aggregation, which means the intermediate nodes simply store and forward packets. Thus, TinyDB actually gives a lower bound on the average computational overhead of each node.

We compare the computational overhead per node in TinyDB, INLR, and Iso-Map. Fig. 15 plots the computational intensity of the three protocols under different network sizes. The computational intensity of each protocol is normalized with the operational overhead of each arithmetic operation. As shown in Fig. 15a, TinyDB and Iso-Map constrain the computational intensity at a low level, while INLR introduces a relatively huge amount of computations on each sensor node, and such overhead grows with the network size. Compared with INLR, the difference between TinyDB and Iso-Map becomes negligible. Fig. 15b exhibits an amplified view of Iso-Map, showing that the pernode computational intensity does not grow with the network

size. Indeed, Iso-Map scales well as the network size increases with each sensor node bearing a constant computational overhead.

5.3 Energy Efficiency

We bridge the communicational and computational overhead with the energy consumption of the sensor nodes. We presume our sensor platform to be Mica2 mote, which is currently the de facto standard platform for sensor networks. Its 8 MHz/8 bit Microcontroller ATmega128 consumes an active power of 33 mW and provides computation at 242 MIPS/W. Its CC1000 transceiver has a data transfer rate of 38.4 Kbps and consumes 29 mW power for receiving and 42 mW power for transmitting (at 0 dBm) [9], [19], [24]. We transform the communicational and computational overhead into energy consumption according to the above capability data. Fig. 16 plots the pernode energy consumption for contour mapping under the three different protocols. Iso-Map significantly reduces the energy cost compared with TinyDB and INLR. More importantly, while in TinyDB and INLR, the pernode energy cost increases with the network size, Iso-Map minimizes this effect, which provides higher scalability for large scale sensor deployment.

6 RELATED WORK

Contour mapping has been widely proposed as a comprehensive method for visualizing sensor fields. Much research on sensor network monitoring can utilize contour mapping to provide a global view of the monitored fields from which the occurrence and development of environment changes can be easily captured [3], [12], [14], [23].

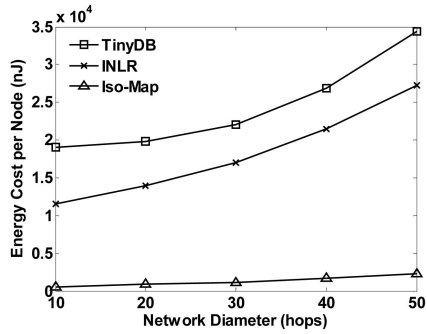


Fig. 16. The pernode energy consumption for contour mapping.

Hellerstein et al. [8] propose the first framework for contour mapping integrated in the TinyDB system. In TinyDB, sensor nodes are deployed into grids. Each sensor node builds a representation of its local cell and delivers it back to the sink. The sink accordingly constructs an isobar contour map based on the received representative values of different grids. Possible in-network aggregation is suggested in this paper; different isobars may be aggregated in the transmission if their attribute values are similar. However, there is no detailed description for the aggregation algorithm in this paper. Xue et al. [27] further develop an in-network aggregation algorithm, INLR, for the isobar contour mapping to reduce the traffic overhead. INLR makes contour regions from close sensor reports of similar readings and delivers contour regions back to the sink. A numerical data model is built for each contour region to describe the distribution of attribute values within the region. INLR aggregates contour regions according to their data model during the delivery. The sink constructs the contour map from the received contour regions. eScan [28] is a similar work that monitors the residual energy of sensor nodes by constructing contour maps of the network. An eScan is defined as a collection of (VALUE, COVERAGE) tuples and each tuple describes a region of COVERAGE where each node has its residual energy within VALUE = (min, max). A tuple initially consists of only an individual sensor node and gets aggregated with other tuples with adjacent COVERAGE and similar VALUE. The sink eventually collects different tuples and creates the eScan contour map based on them. Although the above protocols achieve contour mapping with reduced traffic cost through in-network aggregation, they do not reduce the scale of the generated traffic. The traffic generated from all sensor nodes is still high, and the traffic generation none the less scales proportional to the node number of the network, $O(n)$.

The recently proposed protocol in [15] performs aggregation from the data suppression of sensor nodes to reduce the traffic overhead. The sensor node suppresses its data if there is another sensor node “nearby” transmitting similar data and the transmitted data is considered as a representation of the local field. Upon receiving a subset of sensor readings, the sink performs interpolation and smoothing to obtain the approximation of the contour map. The data suppression protocol reduces the generation of sensor reporting, and thus, reduces the traffic overhead. The fidelity of the resulting contour map is highly related to the rate of data suppression in the network. Limited data suppression can be performed to achieve an acceptable

contour map approximation. As stated in the paper, the suppression algorithm ensures that the range spanned by suppressed nodes is bounded within the 2-hop neighborhood, so the traffic generation is indeed lowered by a factor of the node degree within 2-hop neighborhood. Nevertheless, the traffic generation scales linearly with the number of nodes in the network.

Isoline aggregation [22] shares some similarities with our work. It proposes to reduce the traffic overhead by restricting sensor reporting from nodes near the isolines. However, the paper neither specifies how the sensor nodes detect the isolines passing by nor how the sink recovers the isolines from the discrete reports from sensor nodes.

7 CONCLUSIONS AND FUTURE WORK

We propose Iso-Map, which achieves energy-efficient contour mapping by collecting reports from isoline nodes only. Our theoretical analysis shows that Iso-Map outperforms previous protocols in terms of communicational and computational cost in the network. Iso-Map reduces the generated traffic from $O(n)$ of existing protocols to $O(\sqrt{n})$. We also use trace-driven simulations to compare Iso-Map with existing protocols, and the results show that Iso-Map achieves high fidelity maps with significantly reduced overhead. The scalability of Iso-Map is superior, which makes Iso-Map feasible for the large-scale deployed sensor networks.

We conducted a field study at Huanghua Harbor and investigated the practical application scenario of monitoring the siltation of the sea route. We analyze the advantages and feasibility of deploying an echolocation sensor network for this scenario. We show that it will be of great benefit to utilize Iso-Map to construct contour maps over the sensor network in order to monitor the siltation instead of hiring boats that constantly cruise over the sea area, as is currently done.

Our future work includes building a prototype system at Huanghua Harbor and testing our Iso-Map protocol on this prototype. We hope the implementation experience helps us further understand the efficiency and scalability of the Iso-Map design.

ACKNOWLEDGEMENTS

This work was supported in part by the NSFC/RGC Joint Research Scheme N_HKUST 602/08, the National Basic Research Program of China (973 Program) under grant No. 2006CB303000, the National High Technology Research and Development Program of China (863 Program) under grant No. 2007AA01Z180, China NSFC Grants 60933011, the National Science and Technology Major Project of China under Grant No. 2009ZX03006-001, and the Science and Technology Planning Project of Guangdong Province, China under Grant No. 2009A080207002. The preliminary result of this paper was published in the *Proceedings of IEEE International Conference on Distributed Computing Systems* in 2007.

REFERENCES

- [1] C. Bettstetter, “On the Minimum Node Degree and Connectivity of a Wireless Multihop Network,” *Proc. ACM MobiHoc*, 2002.

- [2] C. Buragohain, D. Agrawal, and S. Suri, "Distributed Navigation Algorithms for Sensor Networks," *Proc. IEEE INFOCOM*, 2006.
- [3] D. Estrin, "Embedded Networked Sensing for Environmental Monitoring," *Keynote, Circuits and Systems Workshop*, Slides available at <http://lecs.cs.ucla.edu/estrin/talks/CAS-JPL-Sept02.ppt>, 2002.
- [4] L. Evans and R. Garipey, *Measure Theory and Fine Properties of Functions*. CRC Press, 1992.
- [5] B. Gedik, L. Liu, and P.S. Yu, "ASAP: An Adaptive Sampling Approach to Data Collection in Sensor Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 12, pp. 1766-1783, Dec. 2007.
- [6] D. Goldenberg, P. Bihler, M. Gao, J. Fang, B. Anderson, A.S. Morse, and Y.R. Yang, "Localization in Sparse Networks Using Sweeps," *Proc. ACM MobiCom*, 2006.
- [7] T. He, J.A. Stankovic, M. Marley, C. Lu, T. Abdelzaher, S.H. Son, and G. Tao, "Feedback Control-Based Dynamic Resource Management in Distributed Real-Time Systems," *Proc. IEEE Real-Time Systems Symp. (RTSS)*, 2001.
- [8] J.M. Hellerstein, W. Hong, S. Madden, and K. Stanek, "Beyond Average: Toward Sophisticated Sensing with Queries," *Proc. IEEE/ACM Information Processing in Sensor Networks (IPSN)*, 2003.
- [9] J. Hill and D. Culler, "Mica: A Wireless Platform For Deeply Embedded Networks," *Micro*, vol. 22, no. 6, pp. 12-24, Nov./Dec. 2002.
- [10] M. Li and Y. Liu, "Underground Coal Mine Monitoring with Wireless Sensor Networks," *ACM Trans. Sensor Networks*, vol. 5, no. 2, article 10, Mar. 2009.
- [11] M. Li, Y. Liu, and L. Chen, "Non-Threshold Based Event Detection for 3D Environment Monitoring in Sensor Networks," *IEEE Trans. Knowledge and Data Eng.*, vol. 20, no. 12, pp. 1699-1711, Dec. 2008.
- [12] S. Li, Y. Lin, S. Son, J. Stankovic, and Y. Wei, "Event Detection Services Using Data Service Middleware in Distributed Sensor Networks," *Telecomm. Systems J.*, vol. 26, pp. 351-368, 2004.
- [13] S. Madden, M.J. Franklin, and J.M. Hellerstein, "TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks," *Proc. Symp. Operating Systems Design and Implementation (OSDI)*, 2002.
- [14] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," *Proc. ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA)*, 2002.
- [15] X. Meng, T. Nandagopal, L. Li, and S. Lu, "Contour Maps: Monitoring and Diagnosis in Sensor Networks," *Proc. Computer Networks*, 2006.
- [16] D. Moore, J. Leonard, D. Rus, and S.J. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," *Proc. ACM SenSys*, 2004.
- [17] J. Munkres, *Topology*, 2nd ed. Prentice Hall, 2000.
- [18] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *Proc. ACM SenSys*, 2004.
- [19] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," *Proc. IEEE/ACM Information Processing in Sensor Networks (IPSN)*, 2006.
- [20] I. Rhee, A.C. Warriar, M. Aia, J. Min, and P. Patel, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," *Proc. ACM SenSys*, 2005.
- [21] H. Sagan, *Space-Filling Curves*. Springer-Verlag, 1994.
- [22] I. Solis and K. Obraczka, "Efficient Continuous Mapping in Sensor Networks Using Isolines," *Proc. IEEE Mobiquitous*, 2005.
- [23] S. Srinivasan and K. Ramamritham, "Contour Estimation Using Collaborating Mobile Sensors," *Proc. Workshop Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS)*, 2006.
- [24] M. Srivastava, "Sensor Node Platforms & Energy Issues," *Proc. Mobicom '02*, tutorial, 2002.
- [25] R. Stoleru, T. He, J.A. Stankovic, and D. Luebke, "High-Accuracy, Low-Cost Localization System for Wireless Sensor Network," *Proc. ACM SenSys*, 2005.
- [26] A. Woo, T. Tong, and D. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," *Proc. ACM SenSys*, 2003.
- [27] W. Xue, Q. Luo, L. Chen, and Y. Liu, "Contour Map Matching For Event Detection in Sensor Networks," *Proc. ACM SIGMOD*, 2006.
- [28] Y.J. Zhao, R. Govindan, and D. Estrin, "Residual Energy Scan for Monitoring Sensor Networks," *Proc. Wireless Comm. and Networking Conf. (WCNC)*, 2002.



Mo Li (M'06) received the BS degree in the Department of Computer Science and Technology from Tsinghua University, China, in 2004, and the PhD degree in the Department of Computer Science and Engineering from Hong Kong University of Science and Technology, in 2009. He is currently a research assistant professor in Fok Ying Tung Graduate School, Hong Kong University of Science and Technology. His research interests include distributed computing, wireless sensor networks, and pervasive computing. He won the ACM Hong Kong Chapter Prof. Francis Chin Research Award in 2009. He is a member of the IEEE and ACM.



Yunhao Liu (SM'06) received the BS degree in the Automation Department from Tsinghua University, China, in 1995, the MA degree from Beijing Foreign Studies University, China, in 1997, and the MS and PhD degrees in computer science and engineering from Michigan State University in 2003 and 2004, respectively. He is now with the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. He is also a member of the Tsinghua EMC Chair Professor Group. His research interests include wireless and sensor networking, peer-to-peer computing, and pervasive computing. He is a senior member of the IEEE and a member of the ACM. He and Mo Li won the Grand Prize of Hong Kong Best Innovation and Research Award 2007.

► **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.**