

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

Yunhao Liu and Mo Li

Department of Computer Science and Engineering
Hong Kong University of Science and Technology
{liu, limo}@cs.ust.hk

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 per-node 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.

1. INTRODUCTION

Recent advances in wireless communication and micro system techniques have resulted in significant developments of wireless sensor networks (WSNs) [4, 6, 8]. A sensor network consists of a large number of low-power, cost-effective sensor nodes that interact with the physical world. 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. 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. Figure 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.

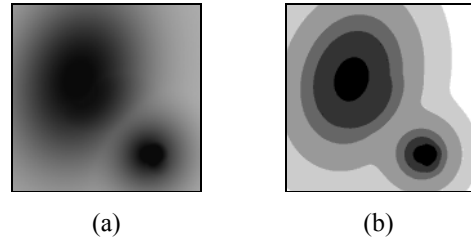


Fig. 1. Contour mapping. (a) A section of underwater depth measurement; (b) The isobath contour map of (a).

A naïve 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 [9, 16, 17]. These protocols aggregate data with similar readings at intermediate nodes, reducing the traffic overhead up to 40% [16]. 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 [16] 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 tradeoff 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. Our analysis proves that, Iso-Map reduces the traffic generation from $O(n)$ of existing protocols to $O(\sqrt{n})$. (2) By employing local measurement on sensors, the per-node 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 discusses the related work. 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 Section 5. Section 6 introduces our investigation of a practical Iso-Map application. And we conclude the work in Section 7.

2. 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 environmental changes can be easily captured.

Hellerstein et al. [4] 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. [16] 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 [17] 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 [9] 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 [13] 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.

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.

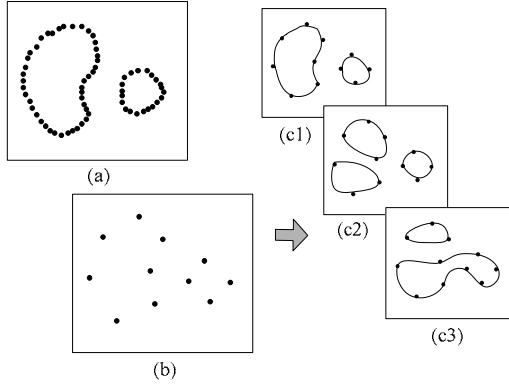


Fig. 2. Contour mapping from isoline nodes. (a) Dense deployment of sensor nodes leads to the isolines; (b) Sparse deployment of sensor nodes provides ambiguous information; (c1) – (c3) Three possible contour maps of (b).

It is not, however, trivial to construct the contour map based solely on isoline nodes' reports. Ideally, as illustrated in Figure 2(a), 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 Figure 2(b), in which the positions of isoline nodes provide only discrete "iso-positions". We cannot deduce how the isolines pass through these positions. For example, based on the data illustrated in Figure 2(b), the sink can interpret into different contour maps, such as the ones shown in Figures 2(c1), (c2) and (c3).

In this section, we will introduce the major operations of Iso-Map including *building network architecture*, *query dissemination and isoline node appointment*, *isoline node measurement*, and *contour map generation*.

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 that is adopted in many systems [4, 7]. We believe that assuming a concrete underlying networking strategy helps clearly state the idea, providing a fair platform for the comparison of performance of 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 [7], which allow each node to dynamically choose a parent from its neighboring nodes based on communication quality. MAC layer reliability of node transmissions can be easily added into this framework [10, 11].

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 $[v_L, v_H]$ and the granularity T of the contour map, which specifies the desired isolines in the contour map with the isolevels $v_i = v_L + i \cdot T \in [v_L, v_H]$. 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 v_p) is an isoline node if and only if: (1) its sensing value is within a predefined border region of the isolevel v_i specified in the query, i.e. $[v_i - \epsilon, v_i + \epsilon]$, and (2) one of its neighboring nodes q has a sensing value v_q , where v_i is between their sensing values, i.e. $v_p < v_i < v_q$, or $v_q < v_i < v_p$. The satisfying node has the isolevel of v_i .

Based on *Definition 3.1*, a node incurs local operations within its neighborhood. The two conditions guarantee that the isoline node is close to the isoline in terms of value and space.

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 $r = \langle v, p, d \rangle$, in which v represents the isolevel of the node, p represents the position of the sensor node and d 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 isoline node, and the position p can be obtained either from attached localization devices such as a GPS receiver or by one of existing algorithms[3, 14]. However, as we mentioned earlier, 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 isoline 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 of d is calculated by:

$$d = -grad(f) = -\nabla f = -\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)^T \quad 3.1$$

To estimate the gradient direction d , an isoline node first needs to approximate the local data map. To build the local data map in this design, each isoline 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.

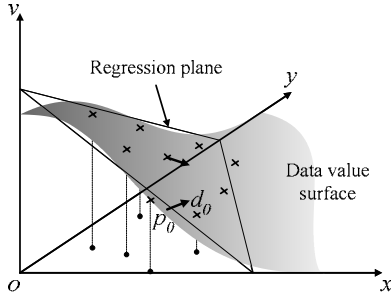


Fig. 3. Linear regression for spatial data modeling

Upon receiving the $\langle v, p \rangle$ tuples from neighboring nodes, the isoline 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.

Figure 3 illustrates the rationale of how the isoline node performs the linear regression and approximates the data value surface with the regression plane. Without loss of generality, we assume the isoline 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 their 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 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 isoline node computes the coefficients of the linear model by solving the equation:

$$Aw = b \quad 3.2$$

$$A = V^T V = \begin{pmatrix} 1 & \sum_{i=0}^n x_i & \sum_{i=0}^n y_i \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i y_i \\ \sum_{i=0}^n y_i & \sum_{i=0}^n x_i y_i & \sum_{i=0}^n y_i^2 \end{pmatrix}, V = \begin{pmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{pmatrix}$$

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

With the obtained plane of linear model approximation $v = L(x, y)$, the isoline node can calculate its gradient by introducing this approximation into formula 3.1:

$$d_0 = -\left(\frac{\partial L}{\partial x}, \frac{\partial L}{\partial y}\right)^T \Big|_{p_0} = -(c_1, c_2)^T \quad 3.3$$

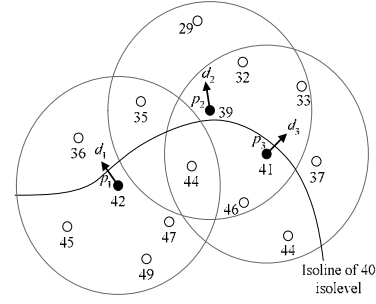


Fig. 4. Gradient directions of the isoline nodes

Figure 4 plots an example where the isoline nodes are at the isolevel of 40. Each isoline 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 isoline node reflects the local trend of data spatial variation and it well approximates the normal direction of the isoline passing by.

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 $v_i = v_L + i \cdot T \in [v_L, v_H]$. The sink separately constructs isolines of different isolevels, and the contour regions reside between them.

When constructing isolines of the isolevel v_i , the sink utilizes the reports with isolevel v_i from the isoline nodes residing along the isolines of v_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. Figure 5(a) shows that isoline nodes of the same isolevel report to the sink and Figure 5(b) depicts the reported iso-positions and corresponding gradient directions. The sink first builds a Voronoi diagram for the set of iso-positions, as shown in Figure 5(c). The Voronoi cell specifies the affecting area of each iso-position, where the sink constructs the local isoline segment according to the gradient direction d at that iso-position. For each cell, a straight line passing the iso-position 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. Figure 5(d) 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.

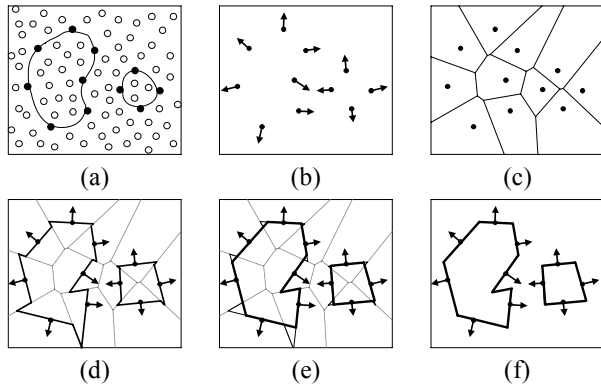


Fig.5. Illustration of the process of contour boundary deduction

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 $(180^\circ, 270^\circ)$. 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 $(90^\circ, 180^\circ)$. 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.

Figure 5(e) 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 iso-position and makes a tighter approximation. The approximated isolines that are eventually obtained are shown in Figure 5(f).

The sink initially builds isolines of the lowest isolevel v_L . These isolines enclose all contour regions above the isolevel v_L . Isolines are then sequentially constructed according to their isolevels.

4. DISCUSSION

Iso-Map utilizes the reports from isoline nodes to construct contour maps. Compared with existing works which rely on 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 $O(\sqrt{n})$, significantly reducing the traffic cost over the entire network. 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 [1] did to exclude some pathologically-shaped “monster curves” such as Peano’s space-filling curves, which hardly emerge as contour boundaries in practice [12].

Definition 4.1: A curve is well behaved if for any 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 [2]. In practice, most of the non-bizarre curves have Hausdorff dimensions of 1.

Theorem 4.1: For any constant number K isolines within an $n^{1/2} \times n^{1/2}$ square area, the total of their lengths L is $O(n^{1/2})$.

Proof. We first enumerate the K isolines and assume that the bounding factors for the K well behaved curves are c_1, c_2, \dots, c_K . We then split the square area into an $r \times r$ grid, with sides $d = n^{1/2}/r$ for each cell (see Figure 6). The i -th isoline intersects with some of the cells and the segment in each cell is bounded by the factor c_i . So:

$$l_i < \sum_{r^2} d \cdot c_i = r \cdot c_i \cdot n^{1/2} \quad 4.1$$

If we choose $c = \text{MAX}(c_1, c_2, \dots, c_K)$, formula 4.1 can be reduced to $l_i < r \cdot c \cdot n^{1/2}$. We sum up all isolines and get:

$$L = \sum_{i=1}^K l_i < Kcr \cdot n^{1/2} \quad 4.2$$

Since the constant factor of $Kc \cdot r$ is independent of n , formula 4.2 leads to the fact that the sum of isoline lengths is at the level of $O(n^{1/2})$. ♦

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 4.2: 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 in *Theorem 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:

$$S = \sum_{i=1}^K \int_{L_i} \varepsilon ds = \varepsilon \cdot \sum_{i=1}^K L_i = \varepsilon L \quad 4.3$$

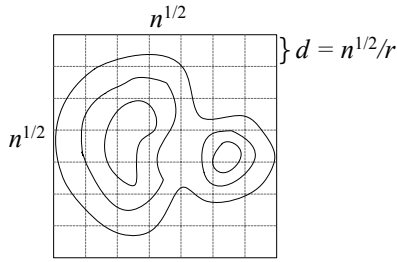


Fig. 6. The square field split into grids

According to formula 4.3, the number of isoline nodes scattered in the stripe is thus $p \cdot S = O(n^{1/2})$. ♦

From *Theorem 4.2*, 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 formula 3.2 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. Thus the computational overhead within the forwarding network is bounded by $O(n)$. Combining the above two parts, the computational overhead within the entire network is $O(deg \cdot n^{1/2} + n) = O(n)$.

Table 1 summarizes and compares Iso-Map with 4 existing approaches. Iso-Map incurs the lowest traffic cost and network computation when performing contour mapping. Note that among the 5 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 H.H. Harbor

TABLE 1: Overhead Comparison

Approach	Traffic Generation	Network Computation	Sensor Deployment
TinyDB [4]	$O(n)$	$O(n)$	Grid
eScan [17]	$O(n)$	$O(n^4)$	Free
INLR [16]	$O(n)$	$\Omega(n^{1.5})$	Grid
Suppression [9]	$O(n)$	$\Omega(nd)$	Grid
Iso-Map	$O(\sqrt{n})$	$O(n)$	Free

(see Section 6). 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 [5]. Perfect link layer is assumed in this simulation, in which the data delivery is guaranteed through performance based routing dynamics [7, 15] and MAC layer re-transmissions [10, 11]. 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.

5.1 Fidelity of Contour Mapping

We utilize a 400m \times 400m section of the underwater depth measurement as our testing data (refer to figure 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 \times 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.

Figures 7(a) – (c) depict the resulting contour maps of TinyDB under the above node densities. Figures 7(d) – (f) depict the resulting contour maps of Iso-Map. The isoline reports received at the sink are 112, 89 and 49. Clearly, both approaches degrade in precision as the node density decreases, but both still produce acceptable fidelity maps.

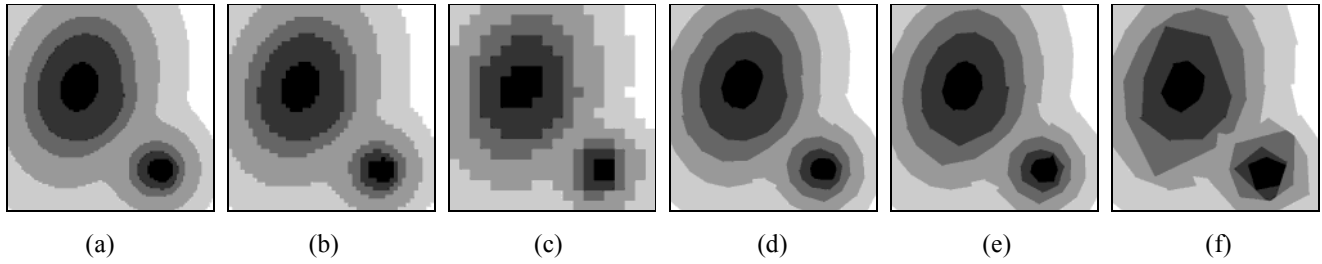


Fig. 7. Performance of isobath contour mapping. (a) – (c): The contour maps created by TinyDB algorithm, under different normalized densities of sensor nodes (4, 1 and 0.16); (d) – (f): The contour maps created by Iso-Map, under different normalized densities of sensor nodes (4, 1 and 0.16).

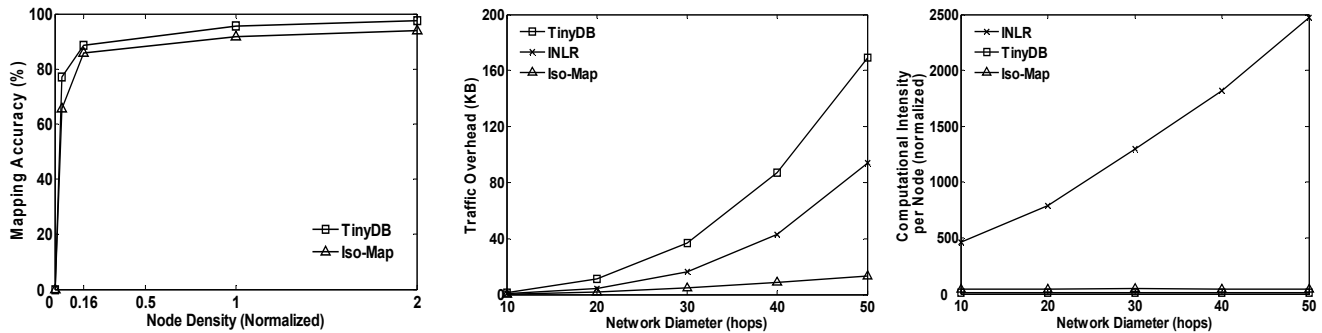


Fig. 8. (a) Contour mapping accuracy against node density; (b) Network traffic overhead against network diameter; (c) Computational intensity against network diameter.

Figure 8(a) 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 400m × 400m field. The mapping accuracy of both TinyDB and Iso-Map rapidly jumps to a high level above 80% as the deployed node density increases. In all cases, Iso-Map is slightly below TinyDB but with comparable accuracy.

5.2 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 [16], as well as with the well-known TinyDB protocol.

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. Figure 8(b) plots the traffic overhead of the three protocols in terms of kilobytes. Consistent with the theoretical analysis, the traffic overhead incurred by TinyDB and INLR grows rapidly while Iso-Map mainly relies on the isoline node reports, imposing much less traffic.

5.3 Node Computational Overhead

In the aggregation based protocols, intermediate nodes conduct heavy computations to aggregate different map segments. On the other hand, in the non-aggregation 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. Figure 8(c) plots the normalized computational intensity of the three protocols under different network sizes. As shown in Figure 8(c), 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.

6. APPLICATION SCENARIO

We conducted a field study on H.H. Harbor, which is currently the second largest harbor of coal transportation in China. It has experienced rapid development over the past 5 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, H.H. Harbor currently suffers from the increasingly severe problem of the silted sea route. H.H. Harbor has a sea route that is 19 nautical miles long and 800m wide at the entrance, including an inner route and an outer route. The sea route is designed to have a water depth of 13.5m 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 H.H. Harbor.

The uncertainty and the high instantaneous intensity of the siltation make monitoring the extent of siltation extremely expensive and difficult. The amount of siltation in H.H. 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.

We propose to deploy an echolocation sensor network on the sea surface to continuously monitor the water depth of the sea route. 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 also clearly locate the dangerous areas where the water depth is under alarm thresholds.

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 H.H. 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.

Our future work includes building a prototype system at H.H. Harbor and testing our Iso-Map protocol on this pro-

totype. We hope the implementation experience helps us further understand the efficiency and scalability of the Iso-Map design.

ACKNOWLEDGEMENT

This work is supported in part by the Hong Kong RGC grant HKUST6152/06E, the HKUST Digital Life Research Center Grant, the National Basic Research Program of China (973 Program) under grant No. 2006CB303000, and NSFC Key Project grant No. 60533110.

REFERENCES

- [1] C. Buragohain, D. Agrawal and S. Suri, "Distributed Navigation Algorithms for Sensor Networks," in Proceedings of IEEE INFOCOM, 2006.
- [2] L. Evans and R. Garipey, *Measure Theory and Fine Properties of Functions*: CRC Press, 1992.
- [3] D. Goldenberg, P. Bihler, M. Gao, J. Fang, B. Anderson, et al., "Localization in Sparse Networks using Sweeps," in Proceedings of ACM MobiCom, 2006.
- [4] J. M. Hellerstein, W. Hong, S. Madden and K. Stanek, "Beyond Average: Toward Sophisticated Sensing with Queries," in Proceedings of IPSN, 2003.
- [5] J. Hill and D. Culler, "Mica: A Wireless Platform For Deeply Embedded Networks," *IEEE Micro*, vol. 22, pp. 12 - 24, 2002.
- [6] M. Li and Y. Liu, "Underground Structure Monitoring with Wireless Sensor Networks," in Proceedings of IPSN, 2007.
- [7] S. Madden, M. J. Franklin and J. M. Hellerstein, "TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks," in Proceedings of OSDI, 2002.
- [8] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in Proceedings of WSN, 2002.
- [9] X. Meng, T. Nandagopal, L. Li and S. Lu, "Contour Maps: Monitoring and Diagnosis in Sensor Networks," *Computer Networks*, 2006.
- [10] J. Polastre, J. Hill and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in Proceedings of ACM SenSys, 2004.
- [11] I. Rhee, A. C. Warriar, M. Aia, J. Min and P. Patel, "Z-MAC: A hybrid MAC for wireless sensor networks," in Proceedings of ACM SenSys, 2005.
- [12] H. Sagan, *Space-Filling Curves*: Springer-Verlag, 1994.
- [13] I. Solis and K. Obraczka, "Efficient Continuous Mapping in Sensor Networks Using Isolines," in Proceedings of IEEE Mobiquitous, 2005.
- [14] R. Stoleru, T. He, J. A. Stankovic and D. Luebke, "High-Accuracy, Low-Cost Localization System for Wireless Sensor Network," in Proceedings of ACM SenSys, 2005.
- [15] A. Woo, T. Tong and D. Culler, "Taming the Underlying challenges of Reliable Multihop Routing in Sensor Networks," in Proceedings of ACM SenSys, 2003.
- [16] W. Xue, Q. Luo, L. Chen and Y. Liu, "Contour Map Matching For Event Detection in Sensor Networks," in Proceedings of ACM SIGMOD, 2006.
- [17] Y. J. Zhao, R. Govindan and D. Estrin, "Residual Energy Scan for Monitoring Sensor Networks," in Proceedings of WCNC, 2002.