无线传感器网络中 ε-近似区域聚集算法

高静，李建中，刘禹

(哈尔滨工业大学 计算机科学与技术学院，黑龙江 哈尔滨 150001)

摘 要：提出了能够满足任意误差和任意查询区域的 ε-近似区域聚集算法。针对聚集函数 SUM，提出了动态规划算法计算达到任意误差的最小数据传输量；针对聚集函数 MAX/MIN，提出的算法通过只传输可能成为查询结果的数据来降低能量的消耗。在真实数据集上进行的实验表明，算法在满足任意区域和任意精度的同时，能够有效地减少能量消耗。

关键词：无线传感器网络；聚集；ε-近似；空间窗口

中图分类号：TP393.01 **文献标识码**：A **文章编号**：1000-436X(2012)02-0099-11

ε-approximate spatial-window aggregation algorithm in wireless sensor networks

GAO Jing, LI Jian-zhong, LIU Yu

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

Abstract: An efficient ε-approximate spatial-window aggregate query processing technique was proposed to approximate aggregate values over arbitrary regions with arbitrary accuracy. A dynamic programming algorithm was devised to compute minimum number of data to refine approximate summation to reach arbitrary accuracy. The proposed algorithm was efficient to compute minimum/maximum values by only transmitting values promising to be in the exact results in order to reduce energy consumption. The experiment using real-world data demonstrates that the algorithms provide high quality results in arbitrary region and reach arbitrary accuracy with low energy cost.

Key words: wireless sensor networks; aggregation; ε-approximate; spatial window

1 引言

近年来，随着传感器技术、嵌入式计算技术以及无线通信技术的飞速发展，由大量具备感知、计算和通信能力的传感器节点组成的无线传感器网络[1,2]已经广泛应用于医疗监护、军事侦察、环境和交通监测、空间探索和灾难救助等领域。

在许多应用中，感知数据的聚集结果，例如最大值、最小值和平均值等，对用户获得监测区域的概况信息有着十分重要的意义。例如，在煤矿监测系统中，用户需要知道瓦斯浓度最大的区域，以防止爆炸的发生。

在传感器网络中，现有的聚集方法主要分为精确聚集[3~6]和近似聚集[7~20]2 类。早期的聚集方法总是将精确的聚集结果返回给用户。大部分精确聚集方法采用基于聚集树[6]的网内聚集思想。该方法将传感器网络中所有节点组织成一棵以基站为根的生成树，内节点可以在网内进行局部的聚集操作，最终基站将获得整个网络的精确聚集结果。虽然该方法利用网内聚集节省了大量能量，然而，该方法

收稿日期：2010-06-17；**修回日期**：2011-01-29

基金项目：国家自然科学基金重点资助项目(61033015)；国家自然科学基金资助项目(60933001, 60831160525)

Foundation Items: The Key Program of the National Natural Science Foundation of China (61033015); The National Natural Science Foundation of China (60933001, 60831160525)

同样也存在以下几个缺点：首先，生成树的结构是不稳定的，一旦树中的某个节点能量耗尽或者某条链路断开，以此节点为根的整个子树都将失效；其次，这种方法需要全网的数据都参与计算，必然会造成较大的能量开销；最后，这种方法对于获得全网的聚集结果比较有效，而对于查询区域可变的聚集查询，该方法不能很好地支持。随着查询区域的变化，聚集树的结构也要随之变化，整个过程会耗费大量的能量和带宽。

由于物理世界的不确定性以及感知器件的不稳定性，使得感知数据本身带有一定的噪声。即便是精确的聚集查询处理算法，也无法保证用户获得准确的聚集结果。而且，在很多应用中，用户是可以忍受一定的误差的。为了降低能量开销，人们展开了传感器网络中近似聚集算法的研究。这些近似聚集方法[16~19]主要利用感知数据的时空相关性建立数据模型，利用模型来估计网内感知数据，以减少数据传输量。与精确的聚集方法相比，近似聚集算法不需要传感器网络内所有的数据都参与聚集计算，能够有效地降低能量开销。但是现有的近似聚集方法也存在着以下一些共性问题。

首先，现有的近似聚集方法都针对预设的固定误差，但是在现实的传感器网络应用中，不同的用户的误差需求可以是任意的。如果要应用户的需求改变预设的固定误差, 则需要对数据模型进行相应的更新甚至重构, 这会消耗大量的能量和带宽。另外，当用户指定误差小于某一阈值之后，模型维护的代价显著提高，甚至接近收集全部原始数据的代价，这就丧失了近似方法的有效性。

其次，现有的近似聚集方法也不能有效地支持查询区域可变的聚集查询。现有的近似聚集方法根据全网数据进行建模，一旦查询区域变化，相应的模型也要随之进行调整甚至完全重建，模型调整的过程中需要向基站传输大量的数据，造成严重的能量消耗。

在现实的传感器网络应用中，不同的用户的误差需求是可变的。例如，在环境检测领域，普通用户只需要知道监测区域是否发生污染，此时可以设定比较宽松的误差界限。而环境学家则需要了解具体的污染程度，需要严格的误差界限以便确定污染等级。近似的聚集方法无法同时有效地满足所有用户的误差需求。另外，在有些情况下，用户并不想知道整个网络的聚集结果，而是对网络中某个特定的区域感兴趣，并且不同用户感兴趣的查询区域也是不同的。例如，在室内环境检测领域，某用户只想了解某个特定房间的最高温度，而不是整个建筑的总体信息，另一个用户则需要其他某个房间的温度信息。对于这类区域可变的聚集查询，现有的方法是不适用的。

鉴于已有的聚集方法不能处理上述问题，本文提出了 ε-近似区域聚集算法。该算法扩展了文献[21]中提出的 ε-近似查询处理架构，能够同时满足用户对任意区域和任意误差的聚集查询需求。针对聚集函数 SUM，本文提出了 EA-Sum (ε-approximate sum) 算法，该算法利用了动态规划的思想确定查询区域的最优传输策略，即计算每个分组达到任意误差所需要传输的最少的数据个数。对于聚集函数 MIN，提出了 EA-Min (ε-approximate min)算法。同时，本文证明了对于任意给定的查询区域，EA-Sum、算法与 EA-Min 算法能够满足任意的精度需求。

本文的主要贡献如下。

1) 扩展了 ε-近似查询处理架构，使其能够支持区域可变的近似聚集查询。

2) 针对 SUM 和 MIN 聚集操作提出了 ε-近似区域聚集查询算法，即 EA-Sum 算法与 EA-Min 算法。上述算法有以下 2 个优点：第一，算法能够有效地满足任意的误差要求，可以在查询处理的过程中对误差进行调整；第二，算法不仅支持全网数据的聚集查询，还可以有效地支持任意区域的查询，提高了聚集查询的灵活性。

3) 在真实的数据集上对 EA-Sum 算法与 EA-Min 算法的性能进行了测试和分析。实验结果表明本方法能够有效地满足任意查询区域和任意误差的要求。

本文的章节安排如下：第 2 节简要介绍了 ε-近似查询处理架构；第 3 节给出了 ε-近似区域聚集查询的形式化定义以及本文的问题定义；第 4 节和第 5 节分别给出了聚集操作 SUM 的 ε-近似查询算法和聚集操作 MIN 的 ε-近似查询算法；第 6 节通过实验验证了算法的有效性；第 7 节是结束语。

2 ε-近似查询处理架构

2.1 传感器网络模型

给定传感器网络 S，网络中传感器节点的个数为 N，N 个节点分为 C 组，每个分组中节点的个数

分别为 m_1，m_2，…，m_C，$\sum_{i=1}^{c} m_i = N$。在每个分组中，有一个节点作为群首节点，任意 2 个节点之间的通信不会超过 λ 跳。由于感知数据具有空间相关性，本文将地理位置相近的节点分为一组。在每个采样周期中，传感器节点的感知数据形式为，其中，h 是节点的位置标识符，其作用相当于节点编号，每个节点都有唯一的位置标识符。t 是感知数据的时间戳。$(v_1,v_2,\cdots,v_d)$表示传感器节点的感知数据值，v_i 是某个观察属性的数据值，例如温度、湿度、电压等。为了表述方便，在下文中，将节点的感知数据记为 $o=(\theta_1,\theta_2,\cdots,\theta_{d+2})^{\mathrm{T}}$。在当前采样周期中，设分组 i 中包含节点，分组 i 中所有节点的感知数据组成集合 $D_i=(o_{h_1},o_{h_2},\cdots,o_{h_k})$，整个查询区域的数据集合 $D_w=\bigcup_{i=1}^{c} D_i$。

2.2　ε-近似查询处理架构

在 ε-近似查询处理架构中，当查询被提交到网络中之后，每个节点将当前的感知数据传输给群首节点，群首节点执行 data shuffling 算法，将接收到的数据进行置换，然后将置换后数据集合的前缀发送给基站，基站根据接收到的部分数据，进行数据恢复得到每个节点的近似数据。

ε-近似查询处理过程由一系列子查询 $\rho_1,\rho_2,\cdots,\rho_q$ 组成。基站或任意节点都可以发出子查询。每个子查询 ρ_i 取得目标集合 $D=(o_{h_1},o_{h_2},\cdots,o_{h_k})$ 中的数据片段 Δ_i，ρ_i 将 Δ_i 与前面子查询的结果 O_{i-1} 整合成新的查询结果 O_i，。根据部分查询结果 O_i，基站可以生成数据集合 D 的近似集合 $\hat{D}=(\hat{o}_{h_1},\hat{o}_{h_2},\cdots,\hat{o}_{h_k})$，集合 $\hat{D}$ 中含有 D 中每个节点 o_{h_i} 的近似值 $\hat{o}_{h_i}$。通过发出新的子查询，获得新的数据片段，基站可以不断地提高拟合的近似数据集合 $\hat{D}$ 的精度，进而满足用户任意的误差要求。

ε-近似查询处理架构如图 1 所示。基站 s_0 发出的查询经过 R 跳到达节点 s_q，在节点 s_q 处发现目标数据集合 $D(D'$ 是另一个数据集合)，s_q 发出子查询 $\rho_1,\rho_2,\cdots$，获得数据片段 $\Delta_1,\Delta_2,\cdots$。s_q 将数据片段 Δ_i 直接返回基站，或者在节点 s_q 处生成部分查询结果 $O_1,O_2,\cdots$，然后将查询结果传输至基站。

下面分别介绍数据恢复和 data shuffling 的执行过程。

数据恢复的执行过程如下所示：在接收到部分查询结果 O 后，$O\subseteq D$ 且 $|O|\geqslant 2$，根据分段线性近似，利用式(1)基站可以得到集合 D 的近似结果 $\hat{D}$，对所有的 $o_h\in D\,(h=h_1,h_2,\cdots,h_k,k=|D|)$，都可以得到 $\hat{o}_h\in\hat{D}$，有

$$\hat{o}_h=\begin{cases}\dfrac{h-v}{u-v}o_u+\dfrac{u-h}{u-v}o_v, & o_h\notin O\\ o_h, \text{其他}\end{cases} \tag{1}$$

其中，$o_u\in O$，$o_v\in O$，u 和 v 是集合 O 中距离 h 最近的 2 个位置标识。

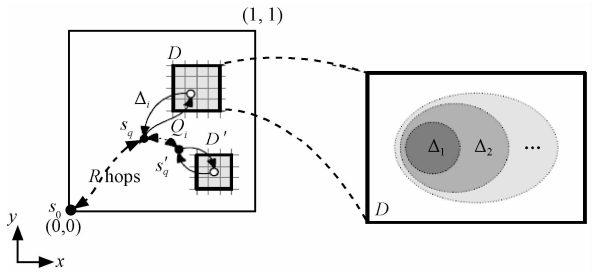

图 1　ε-近似查询处理架构[21]

集合 D 与 $\hat{D}$ 中每一个数据项之间的误差

$$\varepsilon_h=\|\hat{o}_h-o_h\|_\infty=\max_{j=1}^{d+2}|\hat{\theta}_j-\theta_j| \tag{2}$$

其中，$\|\cdot\|_\infty$ 表示无穷范数，即向量中所有元素绝对值的最大值，θ_i 是 o_h 的第 i 个元素。使用无穷范式作为误差的衡量标准，那么，近似集合 $\hat{D}$ 与真实集合 D 之间的误差如式(3)所示，

$$L_\infty(D,\hat{D})=\max_{i=1}^{k}\|\hat{o}_{h_i}-o_{h_i}\|_\infty=\max_{i=1}^{k}\max_{j=1}^{d+2}|\hat{\theta}_{ij}-\theta_{ij}| \tag{3}$$

其中，θ_{ij} 是 o_i 的第 j 个元素，$o_i\in D$。

Data shuffling 算法的主要思想如下：在查询处理过程中，每个分组中的节点都将当前的感知数据传输到临时的群首节点[1]。群首节点执行 data shuffling 算法，将分组内的感知数据集合 $D=(o_1,o_2,\cdots,o_k)$ 中的元素进行置换。当子查询到来时，群首节点按照置换结果 $\pi(D)$ 的顺序将元素传输给基站。data shuffling 算法的运行过程如下：初始时，集合 $\pi(D)$ 为空；首先，将位置标号最小和最大的节点数据 o_1 和 o_k 放入有序集合 $\pi(D)$中，此时 $O=\pi(D)=\{o_1,o_k\}$。然后，根据式(1)对集合 D 中剩余数据进行分段线

1 为了避免固定群首造成的能量失衡，本文使用临时的群首。每隔固定的周期基站重新选取一次群首节点。群首节点可以随机选择，也可以选取剩余能量最大的节点。

性近似，根据式(2)计算每个节点的误差，找到误差最大的节点 i，将 o_i 放入集合 $\pi(D)$中，此时 $O=\pi(D)=\{o_1,o_k,o_i\}$；然后按照上面步骤分别使用 $\{o_1,o_i\}$ 和 $\{o_i,o_k\}$ 对集合 $D[1:i]=(o_1,o_2,\cdots,o_i)$ 和 $D[i:k]=(o_i,o_{i+1},\cdots,o_k)$ 进行近似，分别计算 2 次近似的误差，找到误差最大的点 j，将 o_j 加入集合 $\pi(D)$;重复上述过程,直到D中所有元素都加入$\pi(D)$为止。

图 2 描述了使用 data shuffling 算法的执行过程。数据集 D={ o_1,o_2, o_3, o_4, o_5, o_6}中含有 6 个元素。初始时如图 2(a)所示，先将位置标号 h 最小和最大的点放入集合 O 中；然后计算 o_2, o_3, o_4, o_5 对近似直线 $\overline{o_1o_6}$ 的误差，找到误差最大的点 o_4，将其放入集合 O 中，如图 2(b)所示，此时集合 O_2=(o_1,o_6,o_4)，当前数据近似的误差不超过 ε_4；如图 2(c)所示，分别计算 D(1:4)和 D(4:6)相对 $\overline{o_1o_4}$ 和 $\overline{o_4o_6}$ 的近似误差，找到其中误差最大的点 o_3，将其加入 O 中，此时 O_3=(o_1,o_6, o_4, o_3)，近似误差不超过 ε_3；重复上述计算过程，找到当前近似误差最大的点 o_2,将其放入集合 O 中，如图 2(d)所示。最终，data shuffling 算法输出集合 $o=\pi(D)=\{o_1,o_6,o_4,o_3,o_2,o_5\}$。

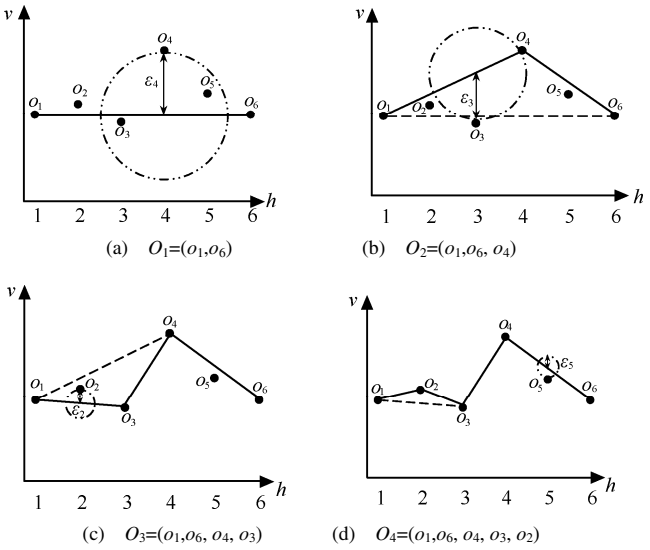

图 2 data shuffling 示例

在文献[21]中证明，按照有序集合 $\pi(D)$的顺序返回查询结果，当基站获得 $\pi(D)$的某个前缀 O 之后，其中，$|\pi(D)|=k$ 、$|O|=b$ 且 $2\leqslant b\leqslant k$，使用 O 生成的近似集合 $\hat{D}$ 与目标数据集合 D 的误差满足不等式

$$L_\infty(D,\hat{D})\leqslant\left\|\hat{x}-x\right\|_\infty$$

其中，x 是 $\pi(D)$中第 b+1 个元素。在文献[21]中还证明近似误差 $L_\infty(D,\hat{D})$ 随着集合 O 的增加而下降，当 O=$\pi(D)$时，$L_\infty(D,\hat{D})=0$。

设集合 $\pi(D)=\{a_1,a_2,\cdots,a_k\}$，$a_i$ 表示节点 i 的感知数据。当接收到子查询 ρ_1 后，将集合 $\pi(D)[1:b_1]=(a_1,a_2,\cdots,a_{b_1})$ 传输至基站，基站使用当前部分结果 $O_1=\pi(D)[1:b_1]$ 根据式(1)对目标数据集合 D 进行近似，得到 $\hat{D}$，近似误差 ε_1 满足 $\varepsilon_1=L_\infty(D,\hat{D})\leqslant\left\|\hat{a}_{b_1+1}-a_{b_1+1}\right\|_\infty$。基站可以根据接收到的结果 O_1 计算出误差 $\left\|\hat{a}_{b_1}-a_{b_1}\right\|_\infty$，由于 $\left\|\hat{a}_{b_1+1}-a_{b_1+1}\right\|_\infty\leqslant\left\|\hat{a}_{b_1}-a_{b_1}\right\|_\infty$，可以用 $\varepsilon_1=\left\|\hat{a}_{b_1}-a_{b_1}\right\|_\infty$ 作为近似集合 $\hat{D}$ 的误差上界。如果当前误差 ε_1 不能满足用户的要求，则基站继续发出子查询 ρ_2，ρ_2 返回集合 $\pi(D)[b_1+1:b_2]=(a_{b_1+1},a_{b_1+2},\cdots,a_{b_2})$，基站使用 ρ_2 返回的结果集合 $\pi(D)[b_1:b_2]$ 与之前所有子查询的结果集合 O_1 合成新的部分结果集合 O_2，$O_2=O_1\bigcup\pi(D)[b_1+1:b_2]=\pi(D)[1:b_2]=(a_1,a_2,\cdots,a_{b_2})$，然后，基站使用 O_2 近似恢复得到新的 $\hat{D}$，当前误差 $\varepsilon_2=L_\infty(D,\hat{D})=\left\|\hat{a}_{b_2}-a_{b_2}\right\|_\infty\leqslant\varepsilon_1$。若 ε_2 仍不满足用户要求，则继续发出子查询，直到取得满足精度要求的数据为止。

ε-近似查询处理架构不仅能够有效地达到任意的误差，支持误差的调整，还可以实现数据的重用。当用户动态调整误差时，只需要传输一些新数据，而完全不需要重传已经传输的数据，大大减少了网络的能量开销。

3 ε-近似区域聚集查询及问题定义

3.1 ε-近似区域聚集查询的定义

在无线传感器网络中，空间窗口查询[22]是一类查询指定地理区域内节点数据的查询方法。在本文中，区域聚集查询(SWAQ, spatial-window aggregate query)可以抽象为如下形式的 SQL 语句：

```
SELECT   {agg(expr)}
FROM   spatial window W
WHERE   selPreds
GROUP BY   { attrs }
HAVING   {havingPreds}
```

EPOCH DURTION　*t*

除了 EPOCH DURTION 子句以外，其他子句的语义与传统 SQL 相同。SELECT 子句指定要执行的聚集操作或者查询的属性。FROM 子句指定执行查询的区域，即在指定区域 W 内的节点才参与聚集计算，在本文中，查询区域 W 使用二维的矩形窗口表示。WHERE 子句指明参与查询的节点需要满足的条件。GROUP BY 子句指定感知数据的分组属性。HAVING 子句过滤掉聚集结果不满足条件的元组。EPOCH DURTION 指明采样周期。

区域聚集查询 SWAQ 返回的结果包括：该 SWAQ 查询的查询号；聚集计算的结果；以及聚集结果的当前误差。

本文假设基站存储网络中节点的位置信息和分组信息。当用户发出区域聚集查询时，根据查询指定的空间窗口 *W*，基站可以计算出每个分组中查询所涉及的节点编号的集合 $Q_1,Q_2,\cdots,Q_C$。设 $|Q_i|=n$，$i=1,2,\cdots,C$。在当前采样周期中，查询所涉及的分组 *i* 中的数据集合为 $D_i=(o_{h_1},o_{h_2},\cdots,o_{h_{n_i}})$，整个查询区域 W 的数据集合 $D_w=\bigcup_{i=1}^{C}D_i$。在执行子查询 ρ_i 后，基站已经收集到每个分组 *i* 的数据集合定义为 O_i，使用 O_i 可以近似恢复分组 *i* 中的数据 $\hat{D}_i$。根据式(1)，基站可以恢复得到整个网络的近似数据集合 $\hat{D}_w=\bigcup_{i=1}^{C}\hat{D}_i$。基站可以根据当前恢复的数据进行聚集运算，得到聚集结果的误差 ε_o，然后将聚集结果及误差返回给用户。

若当前的误差 ε_o 不满足用户要求，则需要进行 REFINE 操作，REFINE 操作的形式如下所示：

$$REFINE(query\_id, \varepsilon_e)$$

其中，*query_id* 表示已经执行的 SWAQ 查询的查询号，ε_e 是期望误差。在执行 REFINE(q_i, ε_e)操作时，基站需要发出新的子查询 ρ_{i+1}，在接收到子查询 ρ_{i+1} 之后，每个分组继续向基站传输部分数据，基站根据新接收到的数据更新数据恢复 $\hat{D}_w$，然后使用更新后的恢复数据 $\hat{D}_w$ 重新计算聚集结果，使得更新后的聚集结果的误差满足期望误差 ε_e。

下面使用一个例子来说明查询的具体过程。

实例 1　假设建筑分为 A、B、C、D 4 个区域，每个区域分别用二维矩形窗口{(0,0),(0.5,0.5)}、{(0.5,0),(1,0.5)}、{(0.5,0.5),(1,1)}和{(0,0.5),(0.5,1)}表示。每个区域由若干个功能不同的房间组成。用户希望查询区域 A 中所有平均温度大于某阈值的会议室。则可以发出如下 SWAQ 查询：

SELECT　avg(temperature),　room
FROM　{(0,0), (0.5,0.5)}
WHERE　roomType=conference room
GROUP BY　room
HAVING　avg(temperature)>threshold
EPOCH DURTION　30s

该查询向用户返回的结果包括：查询号 q_i；满足条件的房间号；该房间的近似平均温度；以及近似的误差。若某一时刻平均温度的误差 ε_o 不满足用户的要求，用户给定新的误差 ε_e，基站执行 REFINE(q_i, ε_e)操作，该操作向用户返回满足期望误差的聚集结果。

3.2　问题定义

在执行 REFINE 操作的过程中，基站需要发出新的子查询 ρ_{i+1}，在接收到子查询 ρ_{i+1} 之后，每个分组继续向基站传输部分数据，基站根据新接收到的数据更新数据恢复，然后使用更新后的恢复数据重新计算聚集结果。

定义 1　(传输方案)。传输方案，表示每个分组 *i* 继续传输 l_i 个数据，使得基站恢复数据的聚集结果满足期望误差。

定义 2　(最优传输策略)。设满足期望误差 ε_e 的所有传输方案的集合为 $P=\{L_1,L_2,\cdots,L_m\}$，每个传输方案的代价 $c(L)=\sum_{i=1}^{c}l_i$，最优传输策略 L^*是传输代价最小的传输方案。

在 REFINE 操作的过程中，为了尽量减少能量开销，需要计算每个分组向基站传输的最少的数据个数，确定最优的数据传输策略。

本文使用的主要符号以及含义如表 1 所示。

4　聚集操作 SUM 的 ε-近似区域查询算法

在当前采样周期内，当执行子查询 ρ_i 之后，基站已经收集到每个分组的数据集合为 O_i，$1\leqslant i\leqslant C$。基站利用 O_i 恢复数据得到查询区域的近似数据集合 $\hat{D}_i=\{\hat{o}_{i1},\hat{o}_{i2}\cdots,\hat{o}_{in_i}\}$，分组 *i* 数据恢复之后的当前误差为 $\varepsilon_i=L_\infty(D_i,\hat{D}_i)$，$1\leqslant i\leqslant C$。直接对恢复后的

数据进行求和操作，得到聚集结果 $SUM=\sum_{i=1}^{C}\sum_{j=1}^{n_i}\hat{o}_{ij}$，当前的总误差为

$$\varepsilon_o=\sum_{i=1}^{C}(n_i-|O_i|)\varepsilon_i$$

表 1　本文使用符号及其描述

符号	描述
N	网络的节点总数
C	网络的分组个数
m_i	第 i 个分组中的节点个数，$\sum_{i=1}^{c}n_i=N$
O_i	当前基站已经收集到的第 i 组的数据的集合
ε_i	基站收集到第 i 分组的数据集合 O_i 后，利用其进行数据恢复所得的最大误差
ε_o	当前的聚集结果的误差
ε_e	期望达到的误差
l_i	为了使总误差达到 ε_e，需要收集的分组 i 的数据集合
D_i	分组 i 中查询所涉及的节点的感知数据集合
D_w	整个查询区域 W 的数据集合，$D_w=\bigcup_{i=1}^{c}D_i$
n_i	第 i 个分组中在查询区域 W 中的节点个数

若当前误差 ε_o 不满足要求，即 $\varepsilon_o>\varepsilon_e$，基站发出新的子查询，当基站接收到第 i 个分组传输的数据 Δ_i' 之后，利用当前集合 $O_i\cup\Delta_i'$ 进行数据恢复，第 i 组的误差由 ε_i 减小为 ε_i'。在新的恢复数据上进行 SUM 聚集运算，当前的总体误差变为

$$\tilde{\varepsilon}^i=\sum_{i=1}^{C}(n_i-|O_i|)\varepsilon_i-(n_i-|O_i|)\varepsilon_i+(n_i-|O_i|-|\Delta_i|)\varepsilon_i'$$

假设 $|\Delta_i'|=k$，令函数 $\delta(i,k)$ 表示第 i 个分组传输 k 个数据之后总体误差的变化量，则

$$\begin{aligned}\delta(i,k)&=\varepsilon_o-\tilde{\varepsilon}^i=(n_i-|O_i|)\varepsilon_i-(n_i-|O_i|-|\Delta_i|)\varepsilon_i'\\&=(n_i-|O_i|)(\varepsilon_i-\varepsilon_i')+k\varepsilon_i'\end{aligned}\tag{4}$$

即第 i 个分组继续传输 k 个数据，总体误差减小 $\delta(i,k)$。

令函数 $opt(i,\varepsilon)$ 表示若当前误差 $\varepsilon_o=\varepsilon$，为了达到期望误差要求，第 i，i+1，…，C 组需要传输的数据总数的最小值。那么，为了满足期望误差，网络需要传输的数据总数的最小值为 $opt(1,\varepsilon_e)$，$opt(1,\varepsilon_e)$ 是最优传输策略 L^*的代价，也是本问题的优化目标。

当 $\varepsilon_e\leqslant\varepsilon<\varepsilon_o$、$1\leqslant i<C$ 时函数 $opt(i,\varepsilon)$ 有下面等式成立：

$$opt(i,\varepsilon)=\min_{0\leqslant k\leqslant n_i-|O_i|}\{opt(i+1,\varepsilon-\delta(i,k))+k\}\tag{5}$$

当 $\varepsilon=\varepsilon_o$ 时，$opt(i,\varepsilon_o)=0$。当 i=C 时，有

$$opt(C,\varepsilon)=\arg\min_{k}\delta(C,k)，\ \delta(C,k)\geqslant\varepsilon_o-\varepsilon\tag{6}$$

式(5)的直观思想如下：如果 $opt(i,\varepsilon)$ 是第 i，i+1，…，C 组的部分最优解，即为了达到误差 ε，第 i，i+1，…，C 组总共最少需传输 $opt(i,\varepsilon)$ 个数据。若在全局最优解中第 i 组需要传输 k 个数据，使得总体误差减少 $\delta(i,k)$，那么第 i+1，i+2，…，C 组为了达到误差 $\varepsilon-\delta(i,k)$ 最少需传输 $opt(i+1,\varepsilon-\delta(i,k))$ 个数据，即 $opt(i+1,\varepsilon-\delta(i,k))$ 也是第 i+1，i+2，…，C 组的部分最优解。

EA-Sum 算法描述如下所示。

算法 1　EA-Sum

procedure　SWAQ -Sum

输入：形如 SWAQ 的区域聚集查询 q

输出：查询号，查询区域内的 SUM 查询结果，当前误差 ε_o

1) 基站根据用户指定的空间窗口 W 计算查询所涉及的节点集合

2) 基站向查询区域发出子查询

3) 在接收到分组 i 的部分查询结果 O_i 后，基站根据式(1)、式(2)和式(3)计算每个分组 i 的近似数据集合 $\hat{D}_i$ 和误差 ε_i

4) 根据恢复的数据计算聚集值 $SUM=\sum_{i=1}^{C}\sum_{j=1}^{n_i}o_{ij}$ 和当前总误差 $\varepsilon_o=\sum_{i=1}^{C}(n_i-|O_i|)\varepsilon_i$

5) return　SUM, ε_o, qid

procedure　REFINE(qid, ε_e)

输入：SWAQ 查询编号 qid; 期望达到的误差 ε_e

输出: 最优数据传输策略

1) for　i=1　to　C

2)　　for　k=1　to　n_i

3)　　根据式(4)计算 $\delta(i,k)$ 的值

4) for each　$\varepsilon_e\leqslant\varepsilon<\varepsilon_o$　根据式(6)计算 $opt(C,\varepsilon)$

5) for　i=C　to　1

6) for　$\varepsilon=\varepsilon_o$　to　ε_e

7)　　利用式(5)计算 $opt(i,\varepsilon)$

8)　　矩阵 $\boldsymbol{S}$ 用来记录分组 i 的最优传输数

据个数 $\boldsymbol{S}(i,\varepsilon)=\arg\min_{k}{}_{0\le k\le n_i-|O_i|}\{opt(i+1,\varepsilon-\delta(i,k))+k\}$

// 12~17 行：计算每个分组需要传输的数据个数

9) l_1=$\boldsymbol{S}(1,\varepsilon_e)$

10) $\varepsilon'=\varepsilon_e$

11) for i=2 to C

12) $\varepsilon'=\varepsilon'-\delta(i-1,l_i-1)$

13) l_i=$\boldsymbol{S}(i,\varepsilon_e)$

14) return $L^*=\{l_1, l_2,\cdots,l_C\}$

定理 1　EA-Sum 算法的时间复杂性是 $O(mN)$，其中，$m=(\varepsilon_o-\varepsilon_e)/ACCURACY$，$ACCURACY$ 为感知数据的精度。

证明　算法在执行 SWAQ 过程中，需要对查询区域内的感知数据进行恢复，设查询所涉及的节点集合为 Q，数据恢复的时间复杂性是 $O(|Q|)$。算法在执行 REFINE 的过程中，影响时间复杂性的主要因素是 $opt(i,j)$ 的计算，因此算法的运行时间主要花费在第 5~8 行。第 7 行计算 $opt(i,\varepsilon)$ 需要 $O(n_i-|O_i|)$ 次运算。第 6 行的内层循环所需的次数为 m，$m=(\varepsilon_o-\varepsilon_e)/ACCURACY$，其中，$ACCURACY$ 为感知数据的精度。第 5 行的外层循环次数为 C。所以，算法总的运行时间为 $\sum_{i=1}^{C}(n_i-|O_i|)m<mN$。由于所以算法的时间复杂性为 $O(mN)$。

算法空间开销主要花费在存储 $opt(i,\varepsilon)$ 和 $S(i,\varepsilon)$，空间复杂性为 $O(mC)$。然而，算法实际消耗的空间远远小于 $O(mC)$，$opt(i,\varepsilon)$ 中相同行的相邻元素值通常是相同的，因此，只需要记录相同值的元素所在的区间端点即可。在实验部分可以看出，算法实际消耗的空间与节点总个数 N 呈线性关系。

5　聚集操作 MIN/MAX 的 ε-近似查询算法

在当前采样周期内，当执行子查询 ρ_i 之后，基站已经收集到每个分组的数据集合为 O_i，$1\le i\le C$。基站利用 O_i 恢复数据得到近似数据集合 $\hat{D}_i=\{\hat{o}_{i1}, \hat{o}_{i2}\cdots, \hat{o}_{ir_i}\}$，每个分组数据恢复之后的误差为 $\varepsilon_i=L_\infty(D_i,\hat{D}_i)$，$1\le i\le C$。对所有分组进行数据恢复，可以得到查询区域内所有的传感器节点的近似感知数据集合 $\hat{D}_w=\bigcup_{i=1}^{C}\hat{D}_i$。基站接收到的分组数据为准确值，通过恢复得到的数据为近似值。因此，查询所处理的数据对象包含准确值和近似值 2 种数据类型。设数据集合 $\hat{D}_w=\{x_1, x_2,\cdots,x_N\}$，其中，$x_i=\begin{cases}\theta_i, & \text{如果基站获得第}i\text{号节点的准确值}\\ \tilde{\theta}_i, & \text{否则，基站恢复得到第}i\text{号节点的近似值}\end{cases}$。

每个经过数据恢复的近似值 $\hat{\theta}_i$ 的误差为 ε_i，近似值 $\hat{\theta}_i$ 对应的准确值一定在以 $\hat{\theta}_i$ 为中心的长度为 ε_i 的区间内，因此，可以将 $\hat{\theta}_i$ 看作区间，$\hat{\theta}_i=[\hat{\theta}_i-\varepsilon_i, \hat{\theta}_i+\varepsilon_i]$，使用二元组 $\langle\hat{\theta}_i,\varepsilon_i\rangle$ 表示。

下面以 MIN 查询为例，描述查询处理的具体过程。对于属性 A，查询区域内所有的传感器节点的感知数据组成输入数据集合 $\hat{D}_w=\{x_1,x_2,\cdots,x_N\}$。若集合 $\hat{D}_w$ 中所有的数据都是准确的，则容易找到准确的最小值，将其返回给用户即可。若查询所处理的数据同时包含准确值和近似值，则首先需要确定最小值所在区间的上界 U 和下界 L。U 是所有准确值和近似值区间左端点的最小值，

$$U=\min_{i=1,2,\cdots,N}\{u_i\} \tag{7}$$

其中，$u_i=\begin{cases}\theta_i, & \text{基站获得节点}i\text{的准确值}\\ \hat{\theta}_i-\varepsilon_i, & \text{基站获得节点}i\text{的近似值}\ \hat{\theta}_i\end{cases}$。

L 是所有准确值和近似值区间右端点的最小值：

$$L=\min_{i=1,2,\cdots,N}\{l_i\} \tag{8}$$

其中，$l_i=\begin{cases}\theta_i, & \text{基站获得节点}i\text{的准确值}\\ \hat{\theta}_i+\varepsilon_i, & \text{基站获得节点}i\text{的近似值}\ \hat{\theta}_i\end{cases}$。

最小值一定在区间 [L,U] 内，所以可以用 $\hat{\theta}_{\min}=\frac{U+L}{2}$ 来近似最小值，近似结果与真实的最小值的误差为所有可能出现最小值节点组成的集合 *Candidate*。

如果根据当前的近似数据集合 $\hat{D}_w$ 计算的最小值不满足期望误差，则从集合 *Candidate* 中继续收集数据，用新收集到的数据更新区间[L,U]和集合 *Candidate*，直到最小值满足期望误差为止。

具体算法描述如下所示。

算法 2　EA-Min

procedure　SWAQ-Min

输入：形如 SWAQ 的区域聚集查询 q

输出：查询号 qid；查询区域内的最小值 $\hat{\theta}_{\min}$，当前误差 ε_{min}

1) 基站根据用户指定的空间窗口 W 计算查询所涉及的节点集合

2) 基站向查询区域发出子查询

3) 在接收到分组 i 的部分查询结果 O_i 后，基站根据式(1)、式(2)和式(3)计算每个分组 i 的近似数据集合 $\hat{D}_i$ 和误差 ε_i

4) 根据式(7)计算最小值所在区间的右端点 U

5) 根据式(8)计算最小值所在区间的左端点 L 和可能出现最小值的节点集合 Candidate

6) 计算当前误差和最小值 $\hat{\theta}_{\min}=\dfrac{U+L}{2}$

7) return qid，$\hat{\theta}_{\min}$，$\varepsilon_{\min}$

procedure REFINE(qid, ε_e)

输入：SWAQ 查询编号 qid，期望达到的误差 ε_e

输出：满足期望误差的最小值 $\hat{\theta}_{\min}$，集合 *Candidate*

1) while $\varepsilon_{\min}>\varepsilon_e$ do

2) 计算集合 $X_c=\{x_i \mid x_i\in\hat{D}_w, i\in Candidate\}$ 中最小值 $x_{\min}$ 以及 $x_{\min}$ 所在的节点编号 m

3) 发出子查询取回编号为 m 的节点的感知数据值 x_{m}

4) 使用 x_{m} 更新数据集合 $\hat{D}_w$

5) 根据式(7)和式(8)计算 $\hat{D}$ 中最小值所在的区间[*L*,*U*]以及集合 *Candidate*

6) 计算和 $\hat{\theta}_{\min}=\dfrac{U+L}{2}$

7) end while

8) return $\hat{\theta}_{\min}$ 和集合 *Candidate*

6 实验结果及分析

实验数据采用 Intel Lab Data[2]，该数据由布置在 Intel Berkeley Research Lab 的 54 个 mica2 传感器节点在 36 天收集的温度、湿度、光强和电压值组成。数据采集的周期是 31s，整个数据集包含大约 230 万条元组。数据的模式包含感知周期、节点编号、温度、湿度、光强、电压等属性。算法采用 C++实现。

传感器节点的部署和分组情况如图 3 所示，按照地理位置相近的原则，将节点分为 6 组。

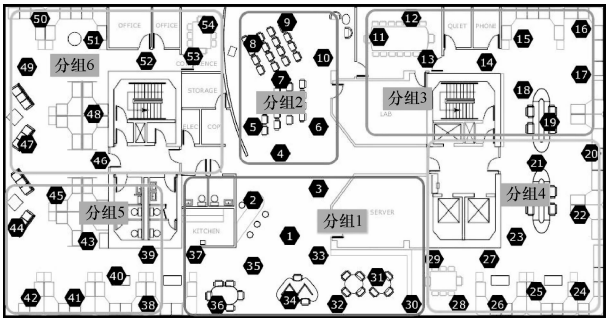

图 3 Intel Lab Data 的部署及分组

实验假设在每个感知周期开始时，每个分组向基站传输一个数据分组，每个数据分组最多包含 3 个节点的数据信息，为了表述方便，实验中选取的空间窗口 W 为全网的数据。

6.1 EA-Sum 算法的性能

首先，通过一组实验测试 EA-Sum 算法达到不同误差所需要传输的数据量。从图 4 中可以看出，随着误差要求的下降，传输的数据个数显著降低。当相对误差达到 8%时，不需要再传输任何数据。另外，数据传输的个数不仅与误差要求有关，还与数据的分布有关。若分组内的数据数值相关性较强，数据恢复时产生的误差较小。由于数据的时变性，即使达到相同的误差所需要传输的数据个数的差距可能很大，如图 4 所示，达到 1%的相对误差最少需要传输 3 个数据，最多需要传输 12 个数据。

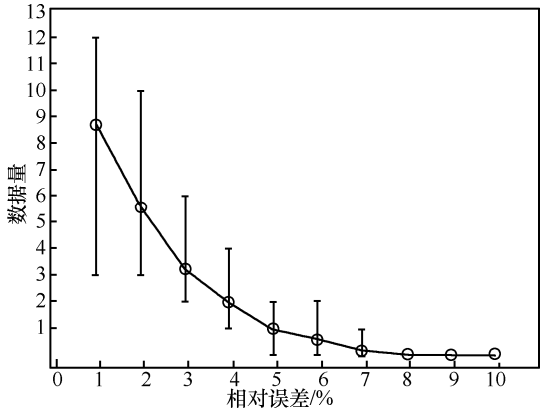

图 4 达到不同误差所需传输的数据个数

EA-Sum 算法采用动态规划思想实现，在算法运行过程中，需要存储最优解矩阵。矩阵每行相邻元素值通常是相同的，每行的数据值可以分解为若干个区间，每个区间内的数据值是相同的。因此，对于每一个区间只需记录其起始位置和区间的元素值即可。由图 5 可以看出，存储最优解矩阵所需的空间与总节点数之间呈线性关系，一般不超过总节点数 N 的 2.5 倍。

2 Intel Lab Data. http://www.select.cs.cmu.edu/data/labapp3/index.html

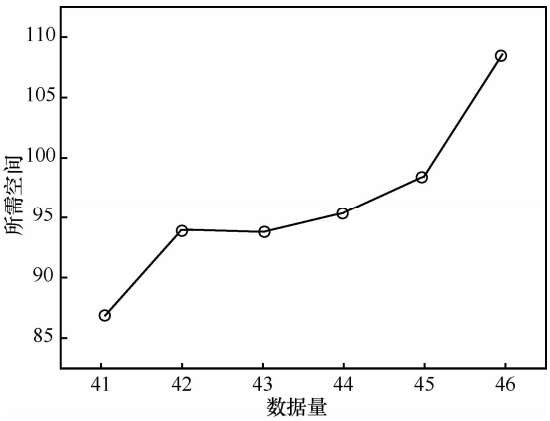

图 5　存储最优解矩阵所需的空间与总数据量之间的关系

图 6 中的实验考察算法处理区域聚集的能力。随机生成了查询所涉及的节点（用来表示任意的查询区域），然后考察算法的数据传输量。由图 6 看出，在达到相同误差的前提下，算法在处理不同数目的节点所需传输的的数据量基本相同。对比图 4 和图 6 可见，误差界限对数据传输量的影响远远大于查询区域的影响。

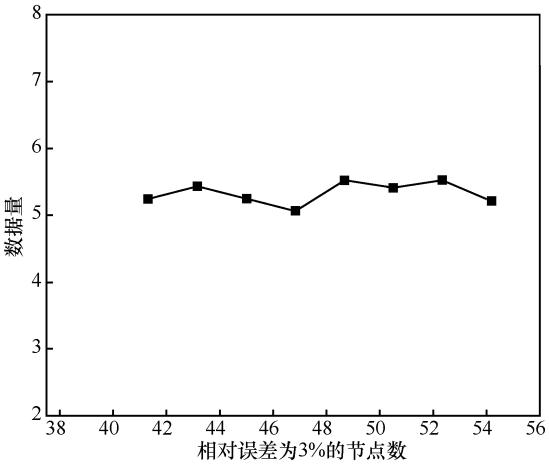

图 6　不同节点数所需的数据传输量

6.2　EA-Min 算法的性能

图 7 表明 EA-Min 算传输的数据个数与所达到的误差之间的关系。可以看出，EA-Min 算法所需传输的数据个数随着误差的增加显著下降。当相对误差为 8%~9%时，基本不需要传输任何数据。平均情况下，传输 8 个数据即可得到确定的最小值。

在图 8 中，依然使用随机生成了查询节点来表示任意的查询区域，然后考察算法的数据传输量。由图 8 可以看出，在达到相同误差的前提下，EA-Min 算法在不同数目的节点上所需传输的的数据量相似。

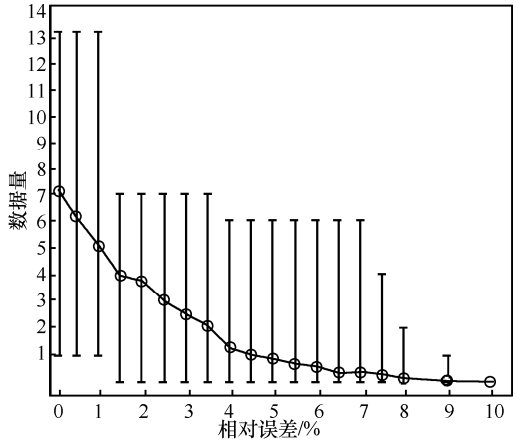

图 7　达到不同误差所需传输的数据个数

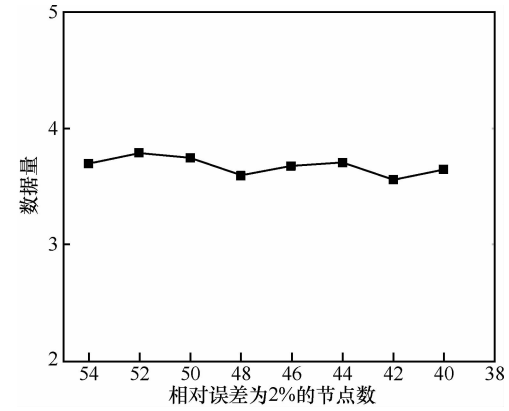

图 8　不同节点数所需的数据传输量

图 9 主要对比 EA-Min 算法和抽样方法在传输相同数据量情况下的绝对误差。由图 9 可以看出，随着传输数据量的增加，EA-Min 算法所达到的误差平稳下降。在传输相同数据量的情况下，EA-Min 算法能够达到更小的误差。由于抽样方法是在样本空间中随机抽取数据，对所得结果的误差没有任何保证，而 EA-Min 算法总是有目标地选择可能成为最小值的数据，因此 EA-Min 算法更为有效。

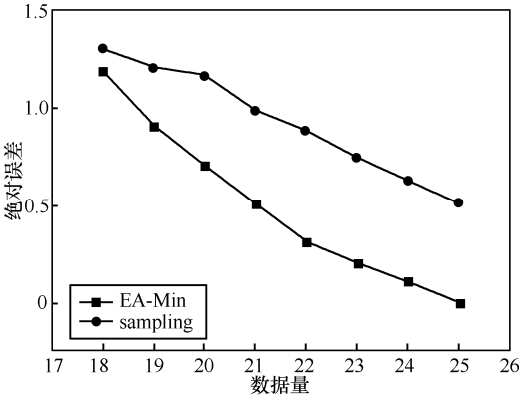

图 9　EA-Min 算法与抽样算法的误差比较

7 结束语

本文提出了 ε-近似区域聚集算法。与现有的聚集方法不同，该算法能够处理传感器网络中任意区域、满足任意误差的聚集查询，并且可以对误差进行灵活的调整。针对聚集操作 SUM 和 MIN，分别提出了 EA-Sum 和 EA-Min 算法。EA-Sum 算法使用动态规划思想计算用户指定的查询区域内满足任意误差的最优传输策略。EA-Min 算法可以有效地计算查询区域内满足任意误差的聚集结果。实验结果表明，EA-Sum 和 EA-Min 算法在满足任意区域、任意误差的同时，能够有效地减少数据传输量，降低网络的能量开销。

参考文献：

[1] AKYILDIZ I F, SU W, SANKARASUBRAMANIAM Y, *et al*. Wireless sensor networks: a survey[J]. Computer Networks, 2002, 38 (4): 393 – 422.

[2] 孙利民, 李建中, 陈渝等. 无线传感器网络[M]. 北京：清华大学出版社, 2005.
SUN L M, LI J Z, CHEN Y, *et al*. Wireless Sensor Networks[M]. Beijing: Tsinghua University Press, 2005.

[3] ZHAO J, GOVINDAN R, ESTRIN D. Computing aggregates for monitoring wireless sensor networks[A]. Proc of the 1st IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)[C]. Anchorage, Alaska, USA, 2003. 139-148.

[4] MADDEN S, FRANKLIN M J, HELLERSTEIN J M, *et al*. The design of an acquisitional query processor for sensor networks[A]. Proc of the 2003 ACM SIGMOD International Conference on Management of Data[C]. San Diego, California, USA, 2003. 491-502.

[5] ZHAO J, GOVINDAN R. Understanding packet delivery performance in dense wireless sensor networks[A]. Proc of the 1st International Conference on Embedded Networked Sensor Systems(SenSys)[C]. Los Angeles, California, USA, 2003.1-13.

[6] MADDEN S, FRANKLIN M J, HELLERSTEIN J M, *et al*. Tag: a tiny aggregation service for ad hoc sensor networks[A]. Proc of OSDI Conf[C]. Boston, United States, 2002. 131-146.

[7] OLSTON C, JIANG J, WIDOM J. Adaptive filters for continuous queries over distributed data streams[A]. Proc of ACM SIGMOD Conference[C]. San Diego, California, 2003. 563-574.

[8] CONSIDINE J, LI F, KOLLIOS K, *et al*. Approximate aggregation techniques for sensor databases[A]. Proc of TEEE Intl Conf on Data Engineer (ICDE)[C]. Boston, USA, 2004. 449-460.

[9] KOLLIOS G, BYERS J, CONSIDINE J, *et al*. Aggregation in sensor networks[J]. IEEE Data Engineering Bulletin, 2005, 28(1): 26-32.

[10] CONSIDINE J, HADJIELEFTHERIOU M, LI F, *et al*. Robust approximate aggregation in sensor data management systems[J]. ACM Transactions on Database Systems, 2009, 34(1): 1-35.

[11] NATH S, GIBBONS P, SESHAN S, *et al*. Synopsis diffusion for robust aggregation in sensor networks[A]. Proc of the 1st International Conference on Embedded Networked Sensor Systems(SenSys)[C]. Baltimore, Maryland, USA, 2004.250-262.

[12] DELIGIANNAKIS A, KOTIDIS Y, ROUSSOPOULOS N. Hierarchical in-network data aggregation with quality guarantees[A]. Proc of the 2004 Intl Conf on Extending Database Technology(EDBT)[C]. Boston, USA, 2004. 658-675.

[13] DELIGIANNAKIS A, KOTIDIS Y, ROUSSOPOULOS N. Processing approximate aggregate queries in wireless sensor networks[J]. Information Systems, 2006, 31(8): 770-792.

[14] DELIGIANNAKIS A, KOTIDIS Y, ROUSSOPOULOS N. Dissemination of compressed historical information in sensor networks[J]. VLDB Journal, 2007, 16(4): 439-461.

[15] CORMODE G, GAROFALAKIS M, MUTHUKRISHNAN S, *et al*. Holistic aggregates in a networked world: distributed tracking of approximate quantiles[A]. Proc of the 2005 ACM SIGMOD International Conference on Management of Data[C]. Baltimore, Maryland, USA, 2005. 25-36.

[16] HARTL G, LI B. Infer: a Bayesian inference approach towards energy efficient data collection in dense sensor networks[A]. Proc of 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)[C]. Columbus, Ohio, USA, 2005. 371-380.

[17] CHU D, DESHPANDE A, HELLERSTEIN J, *et al*. Approximate data collection in sensor networks using probabilistic models[A]. Proc of the 22nd International Conference on Data Engineering (ICDE)[C]. New York, United States, 2006. 48-59.

[18] SILBERSTEIN A, PUGGIONI G, GELFAND A, *et al*. Suppression and failures in sensor networks: a bayesian approach[A]. Proc of 35th International Conference on Very Large Data Bases (VLDB)[C]. Vienna, Austria, 2007.842-853.

[19] DELIGIANNAKIS A, KOTIDIS Y, ROSSOPOULOS N. Processing approximate aggregation queries in wireless senor networks[J]. Information Systems, 2006, 31(8):770-792.

[20] STERN M, BUCHMANN E, B HM K. A wavelet transform for efficient consolidation of sensor relations with quality guarantees[A]. Proc of 35th International Conference on Very Large Data Bases (VLDB)[C]. Lyon, France, 2009. 157-168.

[21] LIU Y, LI J Z, GAO H, *et al*. Enabling ε-approximate querying in

sensor networks[A]. Proc of 35th International Conference on Very Large Data Bases (VLDB)[C]. Lyon, France, 2009. 169-180.

[22] XU Y, LEE W, XU J, *et al*. Processing window queries in wireless sensor networks[A]. Proc of TEEE Intl Conf on Data Engineer (ICDE)[C]. Atlanta, GA, USA, 2006.70.

作者简介：

高静（1985-），女，黑龙江大兴安岭人，哈尔滨工业大学博士生，主要研究方向为无线传感器网络查询处理。

李建中（1950-），男，黑龙江哈尔滨人，哈尔滨工业大学教授、博士生导师，主要研究方向为海量数据管理、无线传感器网络和信息物理融合系统等。

刘禹（1981-），男，黑龙江哈尔滨人，哈尔滨工业大学博士生，主要研究方向为传感器网络、CPS 和移动对象数据管理。

（上接第 98 页）

[10] 刘宏伟, 谢维信, 喻建平等. 基于身份的公平不可否认协议[J]. 通信学报, 2009, 30(7):119-123.
LIU H W, XIN W X, YU J P, *et al*. Fare non-repudiation protocol based on identity-based cryptography[J]. Journal on Communications, 2009, 30(7):119-123.

作者简介：

罗长远（1973-），男，河南信阳人，博士，信息工程大学副教授、硕士生导师，主要研究方向为装备工程和无线通信系统安全。

霍士伟（1985-），男，河北邯郸人，硕士，西安通信学院助教，主要研究方向为普适计算安全和无线网络安全。

邢洪智（1986-），男，河北石家庄人，信息工程大学硕士生，主要研究方向为无线网络安全和移动 IPv6。