

基于平均跳距的水下传感器网络节点定位算法^①

柳 凡, 杜秀娟, 冯振兴, 兰胜林

(青海师范大学 计算机学院, 西宁 810008)

摘 要: 在分析 DV-Hop 定位算法优缺点的基础上, 针对其平均跳距误差大的不足, 提出一种基于平均跳距的水下传感器网络节点定位算法。该算法对平均跳距进行两次改进。首先, 锚节点利用边界锚节点和其一跳邻居的锚节点计算平均每跳距离。其次, 未知节点收到各锚节点的平均距离广播分组包, 根据其到各锚节点的跳数按权重比例公式计算出最终的平均每跳距离。仿真实验表明, 经过两次改进, 该算法优于传统 DV-Hop 定位算法, 提高了定位精确度。

关键词: 水下传感器网络; 节点定位; DV-Hop 算法; 平均每跳距离

Localization Algorithm for Nodes in Underwater Sensor Network Based on Average Hop Distance

LIU Fan, DU Xiu-Juan, FENG Zhen-Xing, LAN Sheng-Lin

(School of Computer Science, Qinghai Normal University, Xining 810008, China)

Abstract: By analysing DV-Hop localization algorithm, for the larger error of average hop distance in the DV-Hop localization algorithm, we proposed a kind of localization algorithm for nodes in underwater sensor network based on average hop distance. The algorithm improves calculation average hop distance. First, the anchor node uses the boundary anchor nodes and its one-hop neighbors of anchor node to calculate average hop distance. Second, the unknown node receives broadcasting package including average hop distance from each anchor node, and then according to the hop-numbers to each anchor node with weight calculates the final average hop distance. Simulation results show that after two improvement the algorithm is superior to the traditional DV-Hop localization algorithm, and improves the location accuracy.

Key words: UWSN; node localization; DV-Hop algorithm; average hop distance

1 引言

近些年, 水下传感器网络(Underwater Sensor Network, UWSN)迅速成为研究领域的热点^[1,2,3,4]。其中一个很重要的原因是因为在水下应用方面它能够提供十分显著的优势。UWSN 能够更好的监测海洋环境, 为各种海洋活动提供更好的技术和全方位的信息。因此得到了各国政府及学术界的极大重视。而 UWSN 的节点定位是得到各种信息的前提条件之一。

水下传感器网络与地面传感器网络^[5]具有不同的特性。地面传感器网络采用电磁波通信。在水下, 由于水的吸收作用, 电磁波在水下传播时能量衰减严重,

且频率越高, 衰减越大。由此可见, 无线电波在水介质中传播距离极为有限, 无法在水下有效的工作。因此, 水下网络节点采用声波通信。由于应用环境的不同, 定位系统通常也要随着改变, 没有一种定位机制适用于所有应用环境。受传感器节点本身通信范围的影响, 节点之间通常需靠多跳的路由方式来进行通信。这时, 利用平均每跳距离和跳数的乘积来计算未知节点到信标节点的距离, 再利用三边定位法或极大似然法计算出它的地理位置信息。DV-Hop 算法^[5]就是利用这种特性来计算的一种典型定位算法。本文通过对 DV-Hop 算法中平均每跳距离的算法改进, 并让其

① 收稿时间:2013-08-31;收到修改稿时间:2013-10-14

适用于水下传感器网络,提高水下定位算法的精确度。

文章的其余部分结构如下。第 2 部分,简要回顾了一些相关的研究工作。第 3 部分,详细介绍 DV-Hop 算法。第 4 部分详细介绍了基于平均距离的水下传感器网络节点定位算法。第 5 部分,仿真实验结果分析。第 6 部分对全文进行总结及下一步研究工作。

2 相关研究

在传感器网络中,节点的位置信息对传感器网络的监测事件是非常重要的。而对于定位来说,事件发生的位置或获取信息的节点位置是至关重要的,没有位置信息的监测消息或活动是毫无意义的。在监测到事件之后,我们更关心事件发生的位置。传感器节点必须首先知道自己的地理位置信息,这是所有其它水下活动监测的前提。随机布放的传感器节点无法事先知道自身的位置信息,所以必须在布放之后进行实时地定位。传感器节点自身定位就是根据少数已知位置的节点,按照某种定位机制确定节点自身的位置。只有确认了节点的自身位置,才能监测到事件发生的具体位置。定位信息除了能够提供事件发生的地点外,还可以进行目标跟踪,实时监测目标的行动路线,预测目标的前进轨迹等等。

全球定位系统 GPS(global position system)是目前定位机制中经常用到的,通过电磁波进行定位,成本高昂,不适用于密集型的大量部署,而且由于电磁波在水下衰减严重,不适用于水下环境。在地面传感器网络中,知道自己位置信息的节点为信标节点,通常配备 GPS 来获取及时的位置信息^[6]。出于 UWSN 环境特性考虑,信标节点在网络中部署密度为 5%-20%,以此来控制成本,且只能部署于水面。

计算节点位置的基本方法分三类:一是三边测量法;二是三角测量法;三是极大似然估计。而定位算法通常分为两类:基于距离的定位算法和距离无关的定位算法。

在基于到达时间 TOA^[7]的定位机制中,已知信号的传播速度,然后根据信号传播时间来计算节点之间的距离,最后利用已有算法计算出节点的位置。基于到达时间差 TDOA^[8]的定位机制中,同时发射两种不同传播速度的无线信号,利用其到达时间差和两种信号的传播速度,计算两个节点之间的距离。基于信号到达角度 AOA^[9]的定位机制中,接收节点利用天线

阵列或多个超声波接收机感知发射节点信号的到达方向,计算接收节点和发射节点之间的相对方位或角度,再通过三角测量法计算出节点的位置;基于接收信号强度指示 RSSI^[10]的定位机制中,已知发射节点的发射信号强度,接收节点根据收到的信号强度,计算出信号的传播损耗,利用理论和经验模型将传输损耗转化为距离,在利用上述算法计算出节点的位置。

而基于距离无关的定位算法无需实际测量节点间的绝对距离或方位就能够计算节点的位置,目前提出的定位机制主要有质心算法^[11]、APIT 算法^[12]、DV-Hop 算法^[13]等等。上述算法,因为无需测量节点间的绝对距离或方位,从而降低了对节点硬件的要求,使得节点成本更适合于大规模传感器网络。所以这种定位机制定位性能受环境因素的影响小,虽然定位的误差相应有所增加,但是其定位精度能够满足大多无线传感器网络的应用要求。目前,在应用中较为普遍。

3 DV-Hop算法概述

DV-Hop 定位算法是一种类似于传统网络中的距离向量路由机制。它的基本思想是未知节点先计算自己到每个信标节点的最小跳数,然后估算平均每跳距离,利用最小跳数乘以平均每跳距离,得到未知节点与信标节点之间的估计距离,最后利用三边定位算法或极大似然法获得节点位置信息。

其定位过程分为三个阶段:

第一阶段:信标节点向邻居节点广播自身定位信息的分组,分组包含发送者,时间戳,自身位置信息,跳数。跳数字段的初始化为 0。未知节点记录到每个信标节点的最小跳数。若接收到同一个信标节点的较大跳数分组,舍弃;保留较小跳数的分组,同时转发给邻居节点,且跳数字段加 1。通过这个方法,网络中所有节点能够记录下到每个信标节点的最小跳数。

第二阶段:计算未知节点和信标节点的实际距离。

每个信标节点根据第一个阶段中记录的其他信标节点的位置信息和相距跳数,利用公式(1)估算平均每跳的实际距离,公式如下:

$$HopSize_i = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} h_j} \quad (1)$$

其中, (x_i, y_i) , (x_j, y_j) 是信标节点 i 、 j 的坐标, 是 h_j 信标节点 i 与 j ($i \neq j$) 之间的相距跳数。利用公式

(1)得到平均每跳距离, 信标节点广播带有生存期字段的分组到网络中, 未知节点仅记录第一个收到的平均每跳距离, 其余信标节点的平均每跳距离舍掉. 未知节点收到分组后, 利用第一阶段计算得到的最小跳数, 开始计算它到每个信标节点的距离.

第三阶段: 利用三边测量法或极大似然估计法, 计算节点自身位置.

未知节点利用第二阶段中平均每跳距离和最小跳数的乘积来计算它到各个信标节点的跳段距离, 利用三边测量法或极大似然估计法计算自身坐标.

DV-Hop 定位算法拓扑图, 举例如下:

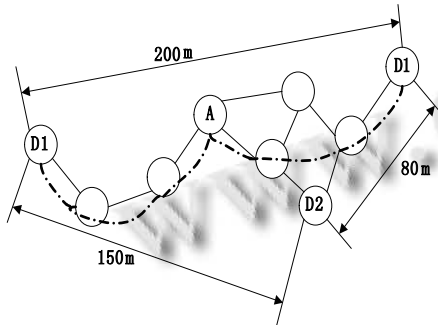


图 1 DV-Hop 定位算法拓扑图

图 1 中, $D1$, $D2$, $D3$ 为信标节点, 节点之间的距离、跳数, 如图所示, 未知节点为 A . 那么信标节点

$D2$ 计算的平均每跳距离为 $\frac{(80+150)}{(2+5)}$. 而 A 从 $D2$ 获

得平均每跳距离, 则节点 A 与三个信标节点之间的距离分别为 $D1: 3 \times 32.86$, $D2: 2 \times 32.86$, $D3: 3 \times 32.86$, 最后利用三边测量法计算出节点 A 的坐标.

以上定位机制, 对传感器节点的硬件要求低, 实现简单. 但它在计算未知节点到信标节点的平均每跳距离时存在较大误差. 下面我们通过改进平均每跳距离的算法提高其定位精度, 让其适用于水下传感器网络.

4 基于平均跳距的水下传感器网络节点定位算法

在本节中, 我们详细介绍了基于平均跳距的水下传感器网络节点定位算法. 我们首先介绍它的网络模型, 然后再详细介绍该机制是如何在水下工作的.

4.1 网络架构

为了实现大型的水下传感器网络定位任务, 我们

提出了一种网络架构, 它分为三种不同的节点. 其水下传感器网络模型如图 2 所示.

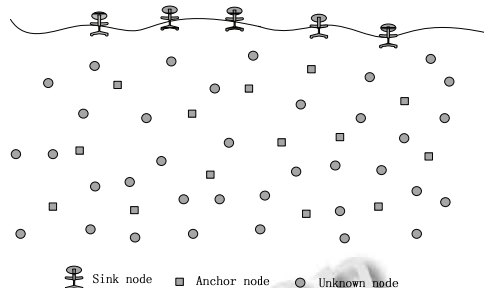


图 2 水下传感器网络模型

● 信标节点(Sink node): 分布在水表面, 节点配备 GPS 可以及时获取其位置信息. 在水下定位机制中, 它充当了“卫星节点”.

● 锚节点(Anchor node): 它可以直接与信标节点通信, 以此来确定自身的位置信息.

● 未知节点(Unknown node): 它只能跟锚节点进行通信, 具有较低的复杂度, 成本低廉, 能量有限, 不能肆意挥霍, 但适合大量部署.

在水下传感器网络中, 由于水流具有流动性, 所以未知节点的定位必须实时更新.

4.2 基于平均跳距的水下传感器网络节点定位算法概述

整个定位过程分为两部分: 锚节点定位和未知节点定位. 信标节点定位锚节点; 而锚节点通过本文两次改进后的机制来定位未知节点.

4.2.1 锚节点定位

在我们的水下传感器网络模型中, 信标节点处于媒介位置, 水下传感器节点的采集信息, 最终都要汇聚到信标节点, 再由信标节点传给陆上基站. 由三边测量法知, 三个信标节点可以确定位置基站位置, 但由于水流动等外界因素, 其定位误差加大. 增加一个信标节点可以计算多个基站位置信息, 取其平均值. 同理, 水下锚节点计算也是如此. 同时, 因每个信标节点配有 GPS, 考虑其昂贵成本. 综合考虑, 水面部署至少满足四个或四个以上的信标节点. 信标节点因为配备 GPS, 能够实时定位自己的位置信息, 通过声波信号与锚节点直接联系, 锚节点收集至少三个以上的信标节点位置信息, 利用三边测量法来计算锚节点的位置信息. 由于水具有流动的特性, 每个传感器节点需要定期得到自己的位置信息我们定义. 每个节点的定位周期为 T_1 .

信标节点分布于水面, 节点配有太阳能装置供其水下工作所耗能量, 出现因能量耗尽而节点损坏的情况较少; 再者, 信标节点配备四个以上, 因外界力量损坏某一节点, 仍然满足三边测量法对节点数的要求. 陆上基站对信标节点实时监控, 影响到水下监测, 会派人工 AUV 部署至缺失位置.

4.2.3 未知节点定位

在 UWSN 中, 节点需要大量部署, 由于低成本, 未知节点的能量、内存是有限的, 其上不能运行过于复杂的计算. 基于这种属性, 我们定义未知节点只跟它本地的一跳邻居节点进行联系. 通过他们自己和锚节点的本地消息传递, 来进行自我定位. 以便于它们参与网络的运行. 同时, 这种方式降低节点能耗, 延长网络生存周期.

通过锚节点定位未知节点, 我们通过两次改进 DV-Hop 算法中的平均每跳距离, 使其适用于 UWSN, 详细算法如下所述.

第一阶段: 计算未知节点到锚节点的最小跳数

锚节点通过信标节点得到自己的位置信息后, 广播其分组信息, 如图 3, 跳数字段初始化为 0, 逐跳 (hop-by-hop) 加 1. 广播过程中, 锚节点也要记录自己到其他锚节点的最小跳数, 未知节点记录到每个锚节点的最小跳数, 若收到来自同一节点的相同数据包或跳数字段较大的数据包, 则舍弃.

表 1 广播分组包

Node-id	Sender	Time-stamp	Location	Hops
---------	--------	------------	----------	------

第二阶段: 计算未知节点到锚节点的距离.

先计算锚节点的平均每跳距离 AHD (Average hop distance), 每个锚节点将得到的 AHD 广播分组给整个网络. 未知节点利用得到的 AHD 乘以跳数即可得到未知节点到锚节点的距离, 如公式 2.

$$Distance_{ij} = AHD \times N_{ij} \quad (2)$$

$Distance_{ij}$ 指锚节点 i 和未知节点 j 之间的距离; N_{ij} 指未知节点 j 到锚节点 i 的最小跳数. 由于网络的分布情况较为特殊, 我们不能单纯依靠未知节点最早收到锚节点的 AHD 来当作它的平均每跳距离. 两次改进工作, 如下.

1) 计算锚节点 i 的 AHD_i 是利用其锚节点 i 的所有一跳邻居的锚节点和其边界锚节点来计算 AHD_i . 公式 4 中, AHD_i 值由两部分节点数据组成, 锚节点

j 是锚节点 i 的一跳邻居节点; 锚节点 k 是其边界节点.

2) 由于节点分布的特殊性, 我们需综合考虑其周围环境. 未知节点接受其他锚节点发送来的 AHD_i , 根据其跳数倒数来计算其权重比例, 如公式 3, α_i 是对应锚节点 i 所占比重因子. 公式 5 中, N_{ij} 是未知节点 i 到锚节点 j 的跳数.

$$AHD = \sum_{i=1}^n \alpha_i AHD_i \quad (i=1, \dots, n \text{ 锚节点编号}) \quad (3)$$

$$AHD_i = \frac{\sum_{j \neq i} \left((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right) + \sum_{k \neq i} \left((x_i - x_k)^2 + (y_i - y_k)^2 + (z_i - z_k)^2 \right)}{\sum_{j \neq i} h_j + \sum_{k \neq i} h_k} \quad (4)$$

$$\alpha_i = \frac{\frac{1}{N_{ij}}}{\sum \frac{1}{N_{ij}}} \quad (5)$$

第三阶段: 利用未知节点到锚节点的距离, 通过极大似然法计算未知节点的位置信息.

未知节点收到三个以上的锚节点平均每跳距离. 其坐标分别为 (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , $\dots$, (x_n, y_n) , 设它们到节点 A 的距离分别为 d_1 , d_2 , d_3 , $\dots$, d_n , 设节点 A 的坐标为 (x, y) . 则有如下公式:

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \\ \vdots \\ (x_n - x)^2 + (y_n - y)^2 = d_n^2 \end{cases} \quad (6)$$

用第一个方程分别减去最后一个方程, 得:

$$\begin{cases} x_1^2 - x_n^2 - 2(x_1 - x_n)x + y_1^2 - y_n^2 - 2(y_1 - y_n)y = d_1^2 - d_n^2 \\ x_{n-1}^2 - x_n^2 - 2(x_{n-1} - x_n)x + y_{n-1}^2 - y_n^2 - 2(y_{n-1} - y_n)y = d_{n-1}^2 - d_n^2 \end{cases} \quad (7)$$

公式(7)的线性方程表示方式为: $AX = b$, 其中:

$$A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix} \quad (8)$$

$$b = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix} \quad (9)$$

$$X = \begin{bmatrix} x \\ y \end{bmatrix} \quad (10)$$

然后使用标准的最小均方差估计可以得到的节点 A 的坐标为: $\hat{X} = (A^T A)^{-1} A^T b$.

5 仿真与分析

在我们的仿真环境中, 500 个传感器节点随机分布在 $100\text{ m} \times 100\text{ m} \times 100\text{ m}$ 的 3D 空间内. 我们定义节点密度为节点的邻居个数. 通过控制节点密度来改变通信范围 R . 假设节点的通信范围遵循正态分布的平均值和均方差, 2%. 这个假定是合理的, 在水下距离测量技术, 很容易得到验证. 在我们的仿真实验中, 锚节点分布为 1%—5%. 我们设置定位周期 T_1 为 1s. 实验使用 matlab 和 NS2 进行仿真, 从横向、纵向两个方面进行性能比较, 并对结构进行分析.

从纵向而言, 采用 matlab 进行仿真分析, 比较了改进后 DV-Hop 算法和传统 DV-Hop 算法性能比较. 图 3(a)显示, DV-Hop 算法和基于平均距离的水下传感器网络节点定位算法都随着节点密度的增加而减少. 因为随着节点密度的增加, 锚节点与未知节点之间的通信更为直接, 得到的平均每跳距离更接近实际距离, 定位误差随之降低. 改进后, 定位误差明显比原算法提高. 当节点增大时, 锚节点的一跳邻居节点和其边界节点多, 跳数多, 求出的平均每跳距离精确度更高. 因此本文算法相比较与传统的 DV-Hop 算法更为精确.

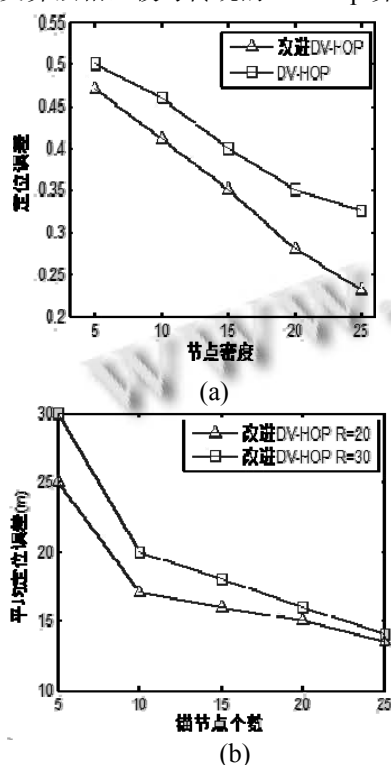


图3 改进后 DV-Hop 算法和传统算法比较

图 3(b)显示, 在不同锚节点条件下, 随着锚节点数目的增加, 改进后的算法计算平均每跳距离值更精确, 定位误差也随之减小. 从图中我们看到, 在不同的通信范围下, 锚节点数目少的时候, 定位误差相差大. 当锚节点继续增多时, 定位误差区域稳定, 是因为当锚节点密度达到一定数目后, 其一跳邻居锚节点和边界锚节点对其影响减弱. 锚节点数目少时, 误差大.

从横向而言, 采用 NS2 进行仿真实验, 比较了 SLMP^[14]和改进后 DV-Hop 算法. 图 4(a)所示, SLMP 算法没有考虑信标节点缺失损坏的情况, 所以其节点密度相当时, 定位的覆盖率相对于改进后的 DV-Hop 算法较窄. 图 4(b), 相同的节点密度, 改进 DV-Hop 算法定位误差较低. 因为改进算法覆盖率较广, 节点缺失会有替补, 在相同密度相同环境下, 节点间的信息交付相对较高, 定位误差降低.

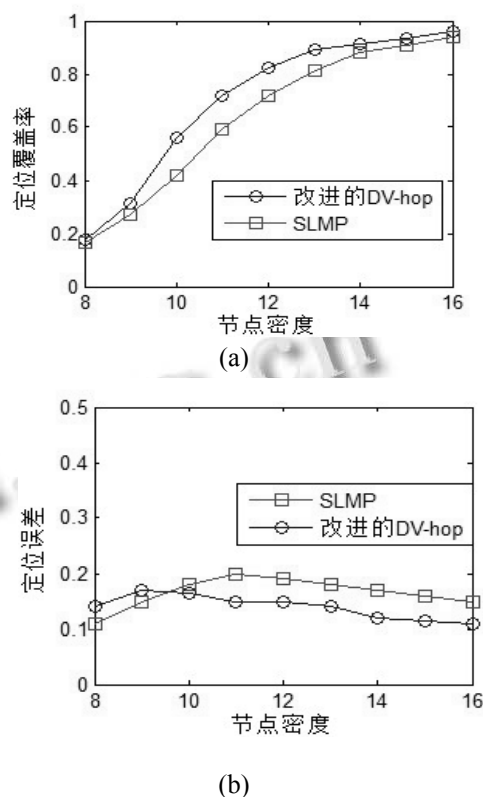


图4 改进后 DV-Hop 算法和 SLMP 比较

6 结论

基于平均距离的水下传感器网络节点定位算法, 通过锚节点的一跳邻居锚节点和其边界锚节点来计算出每个锚节点的 AHD, AHD 被广播至网络中. 考虑到网络分布的特殊性, 未知节点收到多个锚节点的

AHD, 通过锚节点到未知节点的跳数来计算其相应锚节点对未知节点的权重影响因子. 通过两次对平均每跳距离的改进, 经实验结果仿真分析, 该算法提高了定位精度.

该算法提高了定位精度, 但在节点能耗方面损耗较大. 在今后的工作中, 我们将对该算法的节点能耗做进一步研究.

参考文献

- 1 Du XJ, Huang KJ, Liu F. Micro-ANP: A novel network protocol architecture for underwater sensor network. ICSMIM. 2012,12.
- 2 Du XJ, Lan SL, Liu F, Feng ZX. Micro-ANP network protocol architecture and simulation implementation. TELKOMNIKA Indonesian Journal of Electrical Engineering, 2013, 11(4): 1757-1768.
- 3 Guo Z, Peng Z, Wang B, Cui JH, Wu J. Adaptive routing in underwater delay tolerant sensor networks. 2011 6th International ICST Conference on Communications and Networking in China. 2011. 1044-1051.
- 4 Cong YP, Yang G, Wei ZQ, Zhou W. Security in underwater sensor network. 2010 International Conference on Communications and Mobile Computing. 2010. 162-168.
- 5 孙利民, 李建中, 陈渝, 等. 无线传感网络. 北京: 清华大学出版社, 2005.
- 6 Partan J, Kurose J, Levine BN. A survey of practical issues in underwater networks. Proc. of ACM International Workshop on Underwater Networks (WUWNet). 2006. 17-24.
- 7 Harter A, Hopper A, Steggle P, et al. The anatomy of a contextaware application. Proc. of the 5th Annual ACM/IEEE Int'l Conf on Mobile Computing and Networking. Seattle, ACM Press, 1999. 59-68.
- 8 Girod L, Estrin D. Robust range estimation using acoustic and multimodal sensing. Proc. IEEE/RSJ Int'l Conf Intelligent Robots and Systems(IROS'01). Maui, Hawaii, USA. 2001. 1312-1320.
- 9 Priyantha NB, Miu AKL, Balakrishnan H, et al. The cricket compass for context-aware mobile applications. Proc. of the 7th Annual Int'l Conf on Mobile Computing and Networking. Rome: ACM Press, 2001. 1-14. <http://nms.lcs.mit.edu/papers/CricketCompass.pdf>.
- 10 Girod L, Bychovsky V, Elson J, et al. Locating tiny sensors in time and space: a case study. Proc. of the 2002 IEEE Int'l Conf on Computer Design: VLSI in Computers and Processors. Freiburg. IEEE Computer Society. 2002. 214-219. <http://lcs.cs.ucla.edu/Publications/papers/iccd-2002.pdf>.
- 11 Bulusu N, Heidemann J, Estrin D. GPS-less low cost outdoor localization for very small devices. IEEE Personal Communications Magazine, 2000, 7(5): 28-34.
- 12 Niculescu D, Nath B. DV based positioning in ad hoc networks. Journal of Telecommunication Systems, 2003, 22(1/4): 267-280.
- 13 We T, Huang C, Blum B M, et al. Range-free localization schemes for large scale sensor networks. Proc. of the Ninth Annual International Conference on Mobile Computing and Networking(MobiCom ZOOS). San Diego, California. Sep. 2003. 81-95.
- 14 Zhou Z, Jun-Hong C, Amvrossios B. Scalable Location with Mobility Prediction for Underwater Sensor Networks. INFOCOM 2008. The 27th Conference on Computer Communications. IEEE. 2008. 2198-2206.
- 6 Yan W, Zhu ZD, Hu R. A hybrid genetic/BP algorithm and its application for radar target classification. Aerospace and Electronics Conference, NAECON, Proc. the IEEE 1997 National. 1997. 981-984.
- 7 陆琼瑜, 童学锋. BP 算法改进的研究. 计算机工程与设计, 2007, 28(3).
- 8 Zhang JR, Zhang J, Lok TM, Lyu MR. A hybrid particle swarm optimization back propagation algorithm for feedforward neural network training. Applied Mathematics and Computation, 2007, 185(2): 1026-1037.

(上接第 137 页)