

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

H04W 64/00 (2009.01)

H04W 84/18 (2009.01)



[12] 发明专利申请公布说明书

[21] 申请号 200910087901.8

[43] 公开日 2009年12月2日

[11] 公开号 CN 101594677A

[22] 申请日 2009.6.25

[21] 申请号 200910087901.8

[71] 申请人 北京航空航天大学

地址 100083 北京市海淀区学院路37号

[72] 发明人 何兵 张继 赵沁平 郝爱民
王莉莉

[74] 专利代理机构 北京科迪生专利代理有限责任公司

代理人 李新华 徐开翟

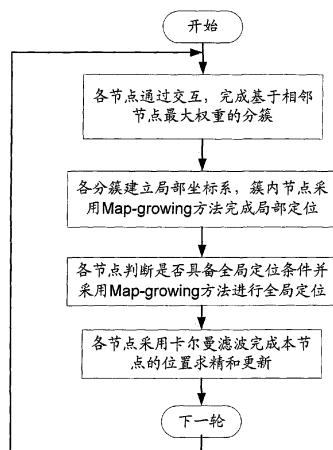
权利要求书1页 说明书12页 附图5页

[54] 发明名称

一种基于分簇的非规则 Ad hoc 网络节点自定位系统

[57] 摘要

基于分簇的非规则 Ad hoc 网络节点无线自定位系统，其特点在于采用锚节点和普通节点组成无线自定位系统，锚节点为静止节点，且其全局坐标经过精确标定，是作为整个系统的定位基准的节点。本发明方法主要包括以下流程(如附图所示)(1)各节点通过交互，完成基于相邻节点最大权重的分簇；(2)各分簇建立局部坐标系，簇内节点采用 Map-growing 方法完成局部定位；(3)各节点判断是否具备全局定位条件并采用 Map-growing 方法进行全局定位；(4)各节点采用卡尔曼滤波完成本节点的位置求精和更新。



1、基于分簇的非规则 Ad hoc 网络节点无线自定位系统，其特征在于通过以下步骤实现：

- (1) 各节点通过交互，完成基于相邻节点最大权重的分簇；
- (2) 各分簇建立局部坐标系，簇内节点采用 Map-growing 方法完成局部定位；
- (3) 各节点判断是否具备全局定位条件并采用 Map-growing 方法进行全局定位；
- (4) 各节点采用卡尔曼滤波完成本节点的位置求精和更新。

2、根据权利要求 1 所述的基于分簇的非规则 Ad hoc 网络节点无线自定位系统，其特征在于：所述步骤（1）中的各节点通过交互，完成基于相邻节点最大权重的分簇方法如下：

- (1.1) 各节点根据自身所有相邻节点的情况计算本节点权重；
- (1.2) 各节点以本节点所有相邻节点为集合形成初始分簇，并自动成为相应初始分簇的簇头；
- (1.3) 各初始分簇的簇头节点根据簇内节点权重计算本分簇权重；
- (1.4) 各初始分簇的簇头节点获得各相邻节点作为簇头节点的初始分簇权重，如果本节点的分簇权重高于所有相邻节点作为簇头的分簇权重，则本初始分簇保留并作为最终分簇，否则取消本节点的分簇资格；
- (1.5) 对未能进入最终分簇的节点采用加权 k 近邻算法选择加入最终分簇。

3、根据权利要求 1 所述的基于分簇的非规则 Ad hoc 网络节点无线自定位系统：所述步骤（2）中建立局部坐标系及簇内节点局部定位方法如下：

- (2.1) 各分簇簇头节点发起建立该分簇的初始局部坐标系，并由该初始局部坐标系的三个节点发起局部的 Map-growing 算法流程，进行局部定位消息的发送和定位处理；
- (2.2) 各分簇簇头节点根据接收到的本分簇内的节点定位信息，选出定位精度高且已完成全局和局部定位的节点，一旦满足局部坐标系的标定条件，则进行局部坐标系标定，而后再将标定后的信息发送给所有分簇内节点。

4、根据权利要求 1 所述的基于分簇的非规则 Ad hoc 网络节点无线自定位系统，其特征在于：所述步骤（3）的全局定位方法如下：各分簇内节点一旦接收到局部坐标系的全局标定信息，则根据标定信息将局部坐标转换到全局坐标，从而最终完成自身的全局定位。

5、根据权利要求 4 所述的基于分簇的非规则 Ad hoc 网络节点无线自定位系统，其特征在于：所述局部坐标系的全局标定信息，包括：3 个已全局定位节点的全局坐标及由局部定位结果计算得到的本节点到上述 3 个节点的距离信息。

一种基于分簇的非规则 Ad hoc 网络节点自定位系统

技术领域

本发明涉及一种基于分簇的非规则 Ad hoc 网络节点自定位系统，属于无线网络定位技术领域。

背景技术

Ad hoc 网络的前身是分组无线网 (Packet Radio Network)，早在 1972 年，美国 DARPA 就启动了分组无线网项目 PRNET，研究如何在战场环境下利用分组无线网进行数据通信。在此之后，DARPA 于 1983 年启动了高残存性自适应网络项目 SURAN (Survivable Adaptive Network)，研究如何将 PRNET 的研究成果加以扩展，以支持更大规模的网络。1994 年，DARPA 又启动了全球移动信息系统 GloMo (Global Mobile Information Systems) 项目，旨在对能够满足军事应用需要的、可快速展开的、高抗毁性的移动信息系统进行全面深入的研究。成立于 1991 年 5 月的 IEEE802.11 标准委员会采用了“Ad hoc 网络”一词来描述这种特殊的自组织、对等式多跳移动通信网络，Ad hoc 网络由此诞生。IETF 也将 Ad hoc 网络称为 MANET (移动 Ad hoc 网络)。

与其它传统的通信网络相比，Ad hoc 网络具有以下显著特点，主要包括：无中心和自组织性；自动配置；动态变化的网络拓扑；受限的无线传输带宽；移动终端的局限性；安全性较差；网络的可扩展性不强；多跳路由；存在单向的无线信道；特殊的信道共享方式；供电问题突出等。

基于 Ad hoc 网络的无线自定位技术是目前研究的重点问题之一，其意义在于移动终端的位置信息可以作为军用通信及各作战单位协同作战的依据；可以解决工程应用中传感器的定位问题；可以应用于医疗急救及一些突发性事件；还可以用于开发基于位置信息的无线 Ad hoc 网络路由协议。

一般的无线节点自定位过程可分为三个阶段，如图 1 所示：

(1) 测量移动节点之间的距离。节点的定位问题一般采用三边测量法计算，因此距离的测定十分重要。可以利用传统的距离获取方法（如 TOA[10]、TDOA 等）计算每对移动节点之间的距离；也可以通过 AOA 的方法获得角度信息，再应用三角测量定位的方法计算距离；或者利用线性规划的方法，根据 RSSI[11]测量的节点之间的关系确定节点间距离。

(2) 确定节点的坐标系统。无论是基于距离测量的定位方法的还是与距离无关的定位方法都需要建立一个全局统一的坐标系统，以确定节点的二维或三维位置坐标。

(3) 计算节点位置。可以根据第一步测量的信息采用相应的定位算法，如三边测量定位算法、三角测量定位算法或最大似然估计法等。

典型的无线自定位系统可分为以下三类

(1) 基于距离的定位算法和与距离无关的定位系统

根据定位过程中是否测量实际节点间的距离又可分为：基于距离的（Range-based）定位和与距离无关（Range-free）的定位。前者需要测量相邻节点间的绝对距离或方位，并利用节点间的实际距离来计算普通节点的位置；后者无需测量节点间的绝对距离或方位，而是利用节点间的估计距离计算普通节点位置。

(2) 递增式的定位算法和并发式的定位系统

根据节点定位的先后次序不同，又可分为：递增式（Incremental）定位和并发式（Concurrent）定位。递增式的定位通常从锚节点开始，锚节点附近的节点首先定位并依次向外延伸，然后各节点逐次进行定位。递增式定位的主要缺点是定位过程中会累积和传播测量误差。并发式的定位算法中所有的节点同时进行位置计算。

(3) 基于锚节点的定位算法和无锚节点的定位系统

根据定位过程中是否使用锚节点又可分为：基于锚节点的（Beacon-based）定位和无锚节点（Beacon-free）的定位。前者在定位过程中以锚节点作为定位基准节点，各节点定位后产生整体统一的坐标系；后者只关心节点间的相对位置，在定位过程中无需锚节点，各节点先以自身作为基准节点，将邻近的节点纳入自己的坐标系中，相邻的坐标系统依次转换合并，最后产生全局的坐标系。

典型的无线自定位算法主要有：

(1) Convex Position 算法又称凸规划定位算法，是一种集中式的定位算法。该算法的基本思想是将节点之间的通信连接关系看成节点位置之间的几何约束，把整个网络模型化为一个凸集，将节点的定位问题转化为凸约束的优化问题，并使用半定规划和线性规划的方法进行全局优化。该算法首先根据普通节点与锚节点之间的通信关系和节点无线通信范围，计算出普通节点可能的存在区域，确定相应的矩形空间，然后以矩形的中心点坐标作为普通节点的位置坐标。如图 2 所示。

凸规划的算法简单，但位置信息的估计误差大，而且由于使用了集中式的算法，中心节点计算负载大、功耗高、网络通信代价大。

(2) Bounding Box 算法，其基本思路是假定所有的节点都位于区域 Q 内， Q 进一步分为 2^n 个单元，然后采用离散的通信模型，即如果节点通信半径为 r ，则节点的通信范围是

以 $2r$ 为边长的正方形通信区域，如图 3 所示。

于是，如果普通节点有 k 个邻居锚节点，则可以取矩形区域的交集作为节点的定位估计。

在该算法的定位过程中，锚节点发送广播消息，消息只传播到一跳邻居，每个节点都要与其邻居节点通信一次，因而整个网络发送的消息数为 $N \times C$ 。该算法的计算开销小，只需要有限的乘法和加法运算和很少的网络通信量。由于该算法使用了分布式处理，因而网络覆盖速度很快，同时由于该算法是基于锚节点的，所以位置估计的精度随着锚节点的数量增加而提高。

(3) DV-hop 算法的基本思想是将普通节点到锚节点之间的距离用网络平均每跳距离和两者之间的跳数乘积表示。该算法首先使用距离矢量交换协议，使网络中所有节点获得距锚节点的跳数，然后锚节点计算网络平均每跳距离值，并将其广播至网络中。当普通节点接收到该值后，则根据跳数计算与锚节点的距离，当普通节点获得与三个或三个以上锚节点的距离时，则执行三边测量定位法，如图 4 所示。

DV-hop 算法不需要节点具备测距能力，算法简单，易于实现，对于各向同性的密集网络，可以得到合理的平均每跳距离，并达到适当的定位精度。但对于拓扑不规则的网络，定位精度会急剧下降。

(4) Map-growing 算法首先在网络中选择一个节点，要求该节点的连通度不小于其邻居节点，然后选择该节点的两个邻居节点形成一个良好三角形（三角形的每一个内角都大于 30° ），利用该三角形可以建立如图 5 所示的局部坐标系。局部坐标系建立之后，以这三个节点作为锚节点向其邻居节点广播其自身坐标，邻居节点收到足够进行定位计算的信息后计算自身坐标，并将计算结果再次向外广播，从而将定位的节点逐步覆盖到整个网络。

Map-growing 算法通过建立独立的局部坐标系进行节点定位，该算法能够适应网络拓扑不规则的情况，当网络规模较大时，由于累积误差的影响，距离坐标系中心节点较远的节点定位误差较大。

(5) GPS-free 算法是一种无锚节点位置信息的分布式处理算法，该算法分为三个部分：

局部相对坐标的确定：首先网络中各个节点向它的邻居节点广播自己的位置信息，从而发现其邻居节点并记入节点矢量 K_i ；然后测量自身与这些邻居节点之间的距离并记入距离矢量 D_i ，最后发送 K_i ， D_i 给它的邻居节点，这样每个节点都知道它的一跳节点和二跳节点与自己的距离关系；最后各节点借助这些距离关系建立自己的局部相对坐标系如图 6 所示。网络中任一节点 i 以自己为中心定义自己的相对坐标系，它的一跳邻居节点 p 作为该坐标系的 x 轴方向，另一个邻居节点 q 的分量 q_y 作为 y 轴方向，其他的节点根据距离信息逐个加入该

坐标系统。

全局坐标的统一：由于网络中每个节点分别建立自己的坐标系统，因而各自相对坐标系方向不一致，需要进行坐标系的统一。

误差估计：GPS-free 算法采用最小二乘法进行误差估计。

(6) MDS-MAP 算法属于集中式算法，如图 7 所示，分为三个步骤：

首先，用最短路径算法，粗略的估计网络中每对节点之间的距离，这些最短路径距离用来为 MDS 方法构造距离矩阵。

其次，对距离矩阵应用典型的 MDS 方法，取最大的两个特征值和特征向量构造一个二维的相对位置坐标。

最后，如果网络中有三个或更多的锚节点，则可根据它们的位置信息将相对坐标转化为绝对坐标。

该算法的精度依赖于多方面的因素，包括网络连通性、局部距离测量的误差、路径的长度和两个相邻坐标系统公共节点的数量。

上述定位系统主要存在以下一些问题：

1) 对非规则网络 Ad hoc 不能依据一些传统的基于规则网络的无线节点自定位算法进行定位（如 DV-hop 算法）。原因在于 DV-hop 算法是一种距离无关的定位算法，仅通过粗略的估计计算平均每跳之间的距离，这在非规则 Ad hoc 网络中可能造成很大的定位误差。

2) 对于一些适合非规则 Ad hoc 网络的无线定位算法（如 Map-growing 算法等），此类算法能够适应网络拓扑不规则的情况，但当网络规模较大时，由于累积误差的影响，对远离局部坐标系中心节点的定位误差会急剧增大。

3) 由于非规则 Ad hoc 网络中无线节点具有移动性，因而需要一种适合非规则 Ad hoc 网络特点的节点位置求精及更新机制。

针对以上问题，本发明提出了一种基于分簇和滤波机制的无线自定位算法及系统，其核心内容可以总结为：整个系统由锚节点和普通节点组成，就普通节点而言，同时存在两个定位系统，分别是局部定位系统和全局定位系统。全系统定位首先基于分簇策略构造若干节点分布相对规则的分区，并建立局部坐标系，然后采用 Map-growing 方法同时进行局部和全局定位，再通过局部向全局的坐标系转换获得全局坐标；此外，本发明还基于卡尔曼滤波的方法跟踪节点位置，建立了一套全系统节点位置求精和更新机制。

发明内容

本发明要解决的技术问题是：克服现有技术的不足，提供一个适应非规则 Ad hoc 网络特点的，具有较高定位精度的无线自定位系统。

本发明采用的技术方案：基于分簇的非规则 Ad hoc 网络节点无线自定位系统，其特点在于采用锚节点和普通节点组成无线自定位系统如图 8 所示，锚节点为静止节点，且其全局坐标经过精确标定，是作为整个系统的定位基准的节点，本定位系统中的锚节点个数需大于等于 3 个。本系统内各节点均需具备针对相邻节点的测距能力和与相邻节点间的无线通信能力。各节点内的模块包括测距模块、无线通信模块、分簇模块、自定位模块及位置信息管理模块，如图 9 所示，其中测距模块负责测量与相邻节点间的距离，无线通信模块负责与相邻节点通信交换定位信息，分簇模块负责发起与确立分区，自定位模块负责根据分簇及测距结果完成节点的局部及全局定位，位置信息管理模块负责位置信息的保存更新和管理。本系统具体通过以下步骤实现普通节点的自定位，如图 10 所示：

- (1) 各节点通过交互，完成基于相邻节点最大权重的分簇；
- (2) 各分簇建立局部坐标系，簇内节点采用 Map-growing 方法完成局部定位；
- (3) 各节点判断是否具备全局定位条件并采用 Map-growing 方法进行全局定位；
- (4) 各节点采用卡尔曼滤波完成本节点的位置求精和更新。

本发明与现有技术相比的有益效果是：

(1) 相对现有技术，本发明通过分簇策略，使得分簇内的节点分布较为规则，通过合理构建坐标系保证分簇内局部定位的精度较高，同时分簇策略可使分簇以较大概率包含有临近初始坐标系或锚节点的簇内节点，从而从而有效控制 Map-growing 方法所造成的误差积累和扩散，保证局部坐标系向全局坐标系转换的精度。

(2) 针对 Ad hoc 网络中，节点的运动一般较慢的特点，采用基于卡尔曼滤波的节点位置求精和更新机制。相对节点定位的波动，节点运动过程体现为一个慢变化过程，采用卡尔曼滤波可以有效地降低定位波动对定位精度的影响。

(3) 相对单一全局定位的 Map-growing 方法，本发明方法可以有效控制定位范围由起始坐标系向外围扩展过程中所造成的误差积累，提高系统的定位精度和鲁棒性。。此外，Ad hoc 网络中，节点的运动一般较慢，因此相对节点定位的波动，其运动过程体现为一个慢变化过程，而卡尔曼滤波可以有效地降低定位波动对定位精度的影响。

附图说明

图 1 为节点自定位过程图示；

图 2 为凸规划方法图示；

图 3 为 Bounding Box 算法图示

图 4 为 DV-hop 方法图示；

图 5 为 Map-growing 算法图示；

图 6 为 GPS-free 算法示意图;

图 7 为 MDS-MAP 算法示意图;

图 8 为基于分簇的非规则 Ad hoc 网络节点无线自定位系统组成, 其中圆框内的节点为锚节点, 其余节点为普通节点;

图 9 为系统内节点模块组成;

图 10 为本系统定位流程;

图 11 为静止节点定位实验中的节点分布情况;

图 12 为静止节点定位实验中 DV-distance 系统定位结果;

图 13 为静止节点定位实验中 Map-growing 系统的定位结果;

图 14 为静止节点定位实验中本系统的定位结果;

图 15 为单一节点运动情况下的定位实验中节点运动情况;

图 16 为单一节点运动情况下的定位实验中移动节点 13 的定位方差曲线;

图 17 为单一节点运动情况下的定位实验中节点 1~8 的定位方差曲线;

图 18 为单一节点运动情况下的定位实验中节点 9~16 的定位方差曲线。

具体实施方式

下面结合附图及具体实施方式对本发明进一步详细说明。

(1) 各节点通过交互, 完成基于相邻节点最大权重的分簇。

这一流程由分簇模块完成;

第一步: 各节点根据自身所有相邻节点的情况计算本节点权重, 具体流程如下:

假设节点 N 拥有 K 个相邻节点, 记为 $\{n_k \quad k=1,2,\dots,K\}$, 并设 N 到 n_k 的测得距离为 d_k 。

定义 N 与 n_k 间的定位因子 $p(N, n_k) = T(n_k) e^{-\sigma d_k^2} \quad k=1,2,\dots,K$ 。其中 $T(n_k)$ 为 n_k 的类型参数, 当 n_k 为锚节点时, $T(n_k) = 1$, 否则 $T(n_k) = e^{-\sigma D^2}$, D 为预设常数 (节点作用半径), σ 为距离加权因子;

从 $p(N, n_k)$ 中选择 3 个最大值, 设为 $p(N, n_{k1}), p(N, n_{k2}), p(N, n_{k3})$, 并定义节点权重为: $G(N) = p(N, n_{k1}) \cdot p(N, n_{k2}) \cdot p(N, n_{k3})$ 。 $G(N)$ 描述的是 N 节点的全局定位精度, 如果 N 的相邻节点中存在三个锚节点, 且距离较近, 则 $G(N)$ 较大, 反之, 如果 N 的相邻节点中存在较少的锚节点或与锚节点距离较远, 甚至没有锚节点, 则 $G(N)$ 较小, 当 N 的相邻节点不足三个时, $G(N) = 0$ 。

第二步: 各节点以本节点所有相邻节点为集合形成初始分簇, 并自动成为相应初始分簇的簇头。

第三步：各初始分簇的簇头节点根据簇内节点权重计算本分簇权重，具体流程如下：

设节点 N 为簇头的初始节点拥有 K 个相邻节点，记为 $\{n_k \ k=1,2,\dots,K\}$ ， n_k 节点的权重为 $G(n_k)$ ，则该初始分簇的权重为 $Z(N) = \sum_{k=1}^K G(n_k)$ 。分簇权重综合反映了分簇内的节点数目及节点的全局定位能力，如果分簇内的节点权重较高，意味着分簇的全局定位能力较强，反映为 $Z(N)$ 较大，反之则 $Z(N)$ 较小。

第四步：各初始分簇的簇头节点获得各相邻节点作为簇头节点的初始分簇权重，如果本节点的分簇权重高于所有相邻节点作为簇头的分簇权重，则本初始分簇保留并作为最终分簇，否则取消本节点的分簇资格；

第五步：对未能进入最终分簇的节点采用加权 k 近邻算法选择加入最终分簇，具体流程如下：

采用的加权 k 近邻算法，设最终形成的分簇为 $\{N_m\} \ m=1,2,\dots,M$ ，而节点 N 为未能进入最终分簇的节点，且拥有 K 个相邻节点，记为 $\{n_k \ k=1,2,\dots,K\}$ ， N 与 n_k 间的定位因子为 $p(N,n_k)$ ；

定义 N 与 N_m 间的隶属度函数： $g_m(N) = \sum_k p(N,n_k) | n_k \in N_m$ ，若 $g_j(N) = \max_m g_m(N)$ ，

则选择 N_j 作为 N 将加入的最终分簇；

上述加权 k 近邻方法充分考虑到 N 结点加入最终分簇后的全局定位能力，即加入哪个最终分簇对 N 结点的全局定位最为有利。

(2) 各分簇建立局部坐标系，簇内节点采用 Map-growing 方法完成局部定位。

这一流程由自定位模块完成；

第一步：各节点遵照 Map-growing 算法进行全局定位消息发送和定位处理，与此同时，各分簇簇头节点发起建立该分簇的初始局部坐标系，并由该初始局部坐标系的三个节点发起局部的 Map-growing 算法流程，进行局部定位消息的发送和定位处理；

上述流程中，分簇内初始坐标系统的建立与 Map-growing 算法类似，本发明重新定义了节点选择策略。设 N 为分簇头，拥有相邻节点集合 $\{n_k \ k=1,2,\dots,K\}$ ，并设 N 到 n_k 的距离为 d_k ，选择其他两个节点 $n_i \in \{n_k\}$ ， $n_j \in \{n_k\}$ ， n_i 与 n_j 间的距离设为 $d_{i,j}$ ，满足 d_i ， d_j ， $d_{i,j}$ 均接近预设距离 D ，该原则的实质是希望得到一个三角形，该三角形具有比较适宜定位的边长（三节点不应非常靠近），同时又比较接近等边三角形。具体的，从 $\{n_k \ k=1,2,\dots,K\}$ 中选择两个节点 n_i, n_j ，形成一组 (N, n_i, n_j) ，定义该组合适宜作为初始坐标系统的权重为：

$V(N, n_i, n_j) = (d_i - D)^2 + (d_j - D)^2 + (d_{ij} - D)^2$ ，最终选择 $n_i \in \{n_k\}$ ， $n_j \in \{n_k\}$ ，满足 $V(N, n_i, n_j) = \text{Max}(V)$ ，至此可构成一个局部坐标系统，包含三个标定节点，分别为簇头 N ，标定节点 n_i 和 n_j ；

第二步：各分簇簇头节点根据接收到的本分簇内的节点定位信息，选出定位精度高且已完成全局和局部定位的节点，一旦满足局部坐标系的标定条件，则进行局部坐标系标定，而后再将标定后的信息发送给所有分簇内节点；

局部定位方法：如果待定位的节点接收到从簇头节点发来的 3 个局部标定节点的局部坐标：簇头 N 为 (x_1, y_1) ，标定节点 n_i 为 (x_2, y_2) ，标定节点 n_j 为 (x_3, y_3) ，且需定位节点已测得到上述三个节点的距离为 d_1 ， d_2 ， d_3 ，则由最小均方差估计准则有： $\hat{S}_{\text{局部}} = (A^T A)^{-1} A^T b$ ，其中： $S_{\text{局部}} = [x, y]^T$ 为待定位节点的真实局部坐标， $\hat{S}_{\text{局部}}$ 为定位系统可计算获得的 $S_{\text{局部}}$ 的最小均方估计，

$$A = \begin{bmatrix} 2(x_1 - x_3) & 2(y_1 - y_3) \\ 2(x_2 - x_3) & 2(y_2 - y_3) \end{bmatrix} \quad b = \begin{bmatrix} x_1^2 - x_3^2 + y_1^2 - y_3^2 + d_3^2 - d_1^2 \\ x_2^2 - x_3^2 + y_2^2 - y_3^2 + d_3^2 - d_2^2 \end{bmatrix}.$$

(3) 各节点判断是否具备全局定位条件并采用 Map-growing 方法进行全局定位。

这一流程由自定位模块完成；

各分簇内节点一旦接收到局部坐标系的全局标定信息（包括 3 个已全局定位节点的全局坐标及测定的本节点到上述 3 个节点的距离信息），则根据标定信息将局部坐标转换到全局坐标，从而最终完成自身的全局定位；

簇头节点 N 为局部坐标系统的原点，有 $N_x = 0$ ， $N_y = 0$ ，对局部坐标系标定节点 n_i ，

其局部坐标为 $n_{i,x} = 0$ 、 $n_{i,y} = d_i$ ，对另一标定节点 n_j ，其局部坐标为 $n_{i,x} = \frac{d_i^2 + d_j^2 - d_{i,j}^2}{2d_i}$ ，

$$n_{i,y} = \sqrt{d_j^2 - n_{i,x}^2}；$$

全局定位方法：设待定位节点的局部坐标为 (x, y) ，3 个已全局定位节点的全局坐标： N 为 (X_1, Y_1) ，标定节点 n_i 为 (X_2, Y_2) ，标定节点 n_j 为 (X_3, Y_3) ，计算待定位节点 N 到上述 3 个节点到的距离为 d_1 ， d_2 ， d_3 ，

$$d_1 = \sqrt{(x - N_x)^2 + (y - N_y)^2}$$

$$d_2 = \sqrt{(x - n_{i,x})^2 + (y - n_{i,y})^2}$$

$$d_3 = \sqrt{(x - n_{j,x})^2 + (y - n_{j,y})^2}$$

由最小均方差估计准则有： $\hat{S}_{\text{全局}} = (A^T A)^{-1} A^T b$ ，其中： $S_{\text{全局}} = [X, Y]^T$ 为节点 N 的真实全局坐标， $\hat{S}_{\text{全局}}$ 为定位系统可计算获得的 $S_{\text{全局}}$ 的最小均方估计，

$$A = \begin{bmatrix} 2(X_1 - X_3) & 2(Y_1 - Y_3) \\ 2(X_2 - X_3) & 2(Y_2 - Y_3) \end{bmatrix} \quad b = \begin{bmatrix} X_1^2 - X_3^2 + Y_1^2 - Y_3^2 + d_3^2 - d_1^2 \\ X_2^2 - X_3^2 + Y_2^2 - Y_3^2 + d_3^2 - d_2^2 \end{bmatrix}。$$

(4) 各节点采用卡尔曼滤波完成本节点的位置求精和更新。

在节点定位跟踪的过程中，可借助卡尔曼滤波完成节点位置的求精和更新机制。具体流程如下：

第一步：初始化卡尔曼滤波器：设置 0 时刻的节点位置速度状态估计为

$$\hat{s}(0|0) = [s_x(0) \quad s_{vx}(0) \quad s_y(0) \quad s_{vy}(0)]^T，\text{ 相应的协方差矩阵为}$$

$$P(0,0) = \begin{bmatrix} \sigma_c^2 & 0 \\ 0 & \sigma_c^2 \end{bmatrix}，\text{ 其中 } \sigma_c^2 \text{ 为数值较大的常数；}$$

第二步：由状态转移模型，根据 $t-1$ 时刻的节点位置速度状态估计可以获得 t 时刻的状态预测及相应的预测协方差阵；

$$\hat{s}(t|t-1) = F(t-1)\hat{s}(t-1|t-1)$$

$$P(t|t-1) = F(t-1)P(t-1|t-1)F(t-1)^T + Q$$

其中： $\hat{s}(t-1|t-1)$ 为 $t-1$ 时刻获得的节点位置速度状态估计， $P(t-1|t-1)$ 为 $\hat{s}(t-1|t-1)$ 对应的协方差矩阵， $\hat{s}(t|t-1)$ 为根据 $t-1$ 时刻的节点位置速度状态估计得到的 t 时刻的状态预测， $P(t|t-1)$ 为 $\hat{s}(t|t-1)$ 对应的协方差矩阵；

第三步：由 t 时刻的状态预测 $\hat{s}(t|t-1)$ 及相应的预测协方差阵 $P(t|t-1)$ ，并由 t 时刻的节点位置观测信息 $z(t)$ ，可获得 t 时节点位置速度状态估计；

$$P^{-1}(t|t) = P^{-1}(t|t-1) + H^T R^{-1} H$$

$$\hat{s}(t|t) = P(t|t)P^{-1}(t|t-1)\hat{s}(t|t-1) + P(t|t)H^T R^{-1} z(t)$$

其中： $\hat{s}(t|t)$ 为 t 时刻获得的节点位置速度状态估计， $P(t|t)$ 为 $\hat{s}(t|t)$ 对应的协方差矩阵；

第四步： $t = t+1$ 返回第二步，重复迭代过程。

由上述流程可以看出，对于节点的定位信息，首先由前一时刻节点的状态估计信息并根据状态转移模型获得当前时刻的一步状态预测信息，然后又用当前时刻节点定位所获得的定位信息作为观测修正一步状态预测信息，获得当前时刻的更新后的状态估计信息，由此循环往复，即可完成节点位置信息的求精和更新机制。换句话说，用观测修正预测的过程为求精过程，用状态估计获得状态预测的过程为更新过程。

卡尔曼滤波的本质是针对状态向量状态转移过程的自适应低通滤波器，可以有效地降低因节点间测距信息抖动所造成的定位误差。此外，由于非规则 Ad hoc 网络中的信息节点的移动速度一般不会很快，因此可以采用卡尔曼滤波模型中的常速模型较好的跟踪状态转移的过程。

节点自定位算法的 NS2 仿真模型：由 UC Berkeley 开发的 NS2 (Network Simulator, Version 2) 是一种面向对象的网络仿真器，NS2 作为一种针对网络技术的、开源的软件模拟平台，包含有丰富的组件和模块，已经成为目前广泛使用的网络模拟软件，故本发明选择 NS2 作为本系统底层支撑的网络仿真平台以验证系统的性能。

针对静止节点的定位实验：节点分布如图 11 所示，图中红圈内设置了 4 个锚节点，实验中假设锚节点的位置已知。锚节点周围设置了一些普通节点，以验证并对比分析锚节点直接定位性能，此类节点简称为“周围节点”；远离锚节点的区域也设置了一些普通节点，以验证并对比分析必须借助过渡节点定位的性能，此类节点简称为“远离节点”；在周围节点和远离节点间设置了一些“过渡节点”，以保证远离锚节点的普通节点可以根据这些过渡节点的信息完成自身的定位。

实验中各节点的作用半径为 50m；节点间测距误差认为是零均值高斯白噪声，方差设置为实际距离的 0%（无误差），10%（较小误差）和 30%（较大误差）三种情况。本测试统计了三种定位方法针对 30 个普通节点的定位方差，以及所需进行的平均交互次数及收发信息次数。

无误差情况下，三种方法的定位结果如图 12-14 所示，其中，图 12 为 DV-distance 系统的定位结果，图 13 为 Map-growing 系统的定位结果，图 14 为本方明系统的定位结果。

由定位结果可以看出，无误差情况下，Map-growing 系统和本系统均可以准确定位所有的普通节点，而 DV-distance 系统仅能准确定位锚节点周围的普通节点，过渡节点和远离节点存在定位误差。

较小误差情况下，三种系统的定位性能如表 1 所示。

表 1 较小误差情况下三种系统的平均定位方差

节点	DV-distance	Map-growing	本系统	节点	DV-distance	Map-growing	本系统
1	31.42	31.86	14.31	16	55.71	11.46	24.75
2	18.90	19.10	9.06	17	129.05	13.13	25.79

3	26.53	28.33	17.08	18	124.93	18.34	35.39
4	30.02	31.72	15.16	19	112.75	27.02	12.86
5	24.14	24.31	11.59	20	80.90	13.65	27.49
6	34.70	35.15	11.61	21	478.75	582.16	182.77
7	33.38	33.75	11.75	22	173.89	113.89	137.21
8	31.71	32.13	14.04	23	213.65	206.85	186.78
9	25.83	25.27	12.07	24	165.52	50.32	52.98
10	234.60	53.44	36.01	25	191.98	69.60	65.44
11	29.63	29.14	13.53	26	298.10	101.54	106.58
12	30.45	30.31	14.19	27	160.10	44.18	43.37
13	65.78	15.56	8.08	28	98.13	178.84	56.08
14	34.23	34.76	16.07	29	126.38	141.92	45.08
15	126.48	11.58	39.40	30	302.36	1264.43	265.69

由测试结果可以看出，在较小误差情况下，DV-distance 系统的定位精度最差，本系统与 Map-growing 系统相比，除个别“过渡节点”外，整体定位精度优于 Map-growing 方法，尤其针对“远离节点”情况，本系统的定位精度和鲁棒性具有明显优势。

较大误差情况下，三种系统的定位性能如表 2 所示。

表 2 较大误差情况下三种系统的定位方差

节点	DV-distance	Map-growing	本系统	节点	DV-distance	Map-growing	本系统
1	114.78	110.33	65.98	16	188.79	111.74	118.59
2	69.05	66.15	43.76	17	251.52	278.00	145.11
3	96.90	98.10	75.85	18	253.90	64.23	63.71
4	109.67	109.85	71.41	19	252.07	99.80	51.92
5	88.19	84.20	55.40	20	231.92	226.83	134.12
6	126.63	121.72	62.91	21	813.23	1332.44	652.90
7	121.56	116.85	64.30	22	551.2	2191.22	652.41
8	115.75	111.26	63.71	23	750.81	2500.43	752.36

9	94.37	87.49	54.95	24	549.40	1019.07	148.04
10	376.44	185.20	139.22	25	602.28	992.24	259.01
11	108.26	100.92	56.69	26	616.79	2186.14	752.60
12	111.25	104.96	55.64	27	442.80	397.75	99.78
13	203.69	67.89	43.28	28	337.18	321.25	213.40
14	125.21	120.36	62.03	29	393.67	348.17	147.89
15	229.56	157.15	153.40	30	793.04	2713.51	952.69

由测试结果可以看出，在较大误差情况下，DV-distance 系统针对“周围节点”及“过渡节点”的定位精度不如 Map-growing 系统及本系统，但对“远离节点”的定位精度与本系统相当。

本系统与 Map-growing 系统相比，除个别“过渡节点”外，整体定位精度较高。此外，测试结果表明，Map-growing 系统在进行“远离节点”的定位时，鲁棒性较低。

针对单一节点运动情况下的定位实验，设计的节点运动情况如图 15 所示：设计单节点移动，主要考虑以下几方面的因素，首先可以避免其他节点同时运动所产生的各种干扰，从而可以更好地分析节点运动对系统定位造成的影响；其次可以更好地验证本系统所采用的节点位置求精及更新机制的有效性和性能，保证测试结果的可靠性。

实验中各节点的作用半径为 50m，节点间测距误差为零均值高斯白噪声，方差设置为实际距离的 10%。节点 13 的运动速度为 $v_x = -2.5m/s$ ， $v_y = 1m/s$ ；测试时间从 0 时刻到 40s。图 16 给出的是节点 13 的定位方差曲线，图 17-18 给出的是节点 1~16 的定位方差曲线。图 16-18 中横坐标均为测试时刻，纵坐标均为定位方差；图 16-18 中，双虚线为 DV-distance 系统的结果，点划线为 Map-growing 系统的结果，虚线为本系统不采用求精更新机制情况下的结果，实线为本系统采用求精更新机制情况下的结果。

由测试结果可以看出，本系统针对节点移动情况下的定位精度优于 DV-distance 系统和 Map-growing 系统，而采用求精更新机制可以进一步提高定位的精度和鲁棒性。

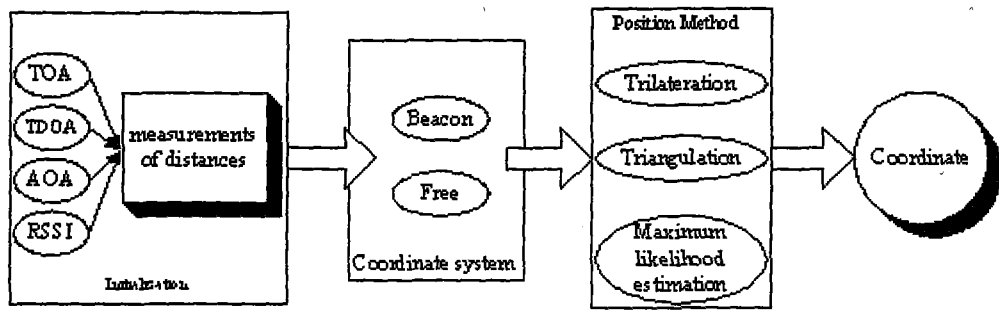


图 1

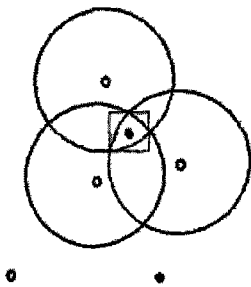


图 2

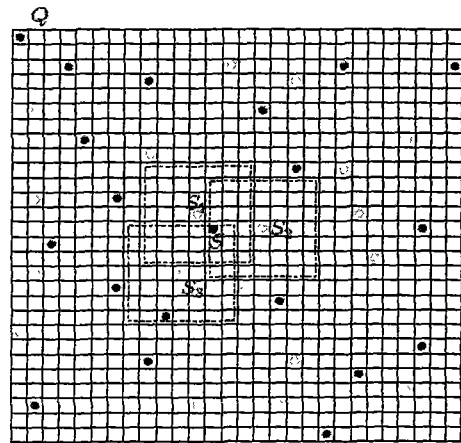


图 3

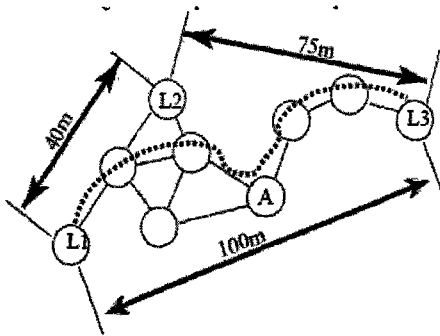


图 4

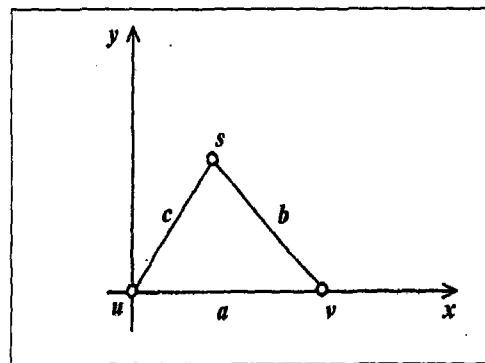


图 5

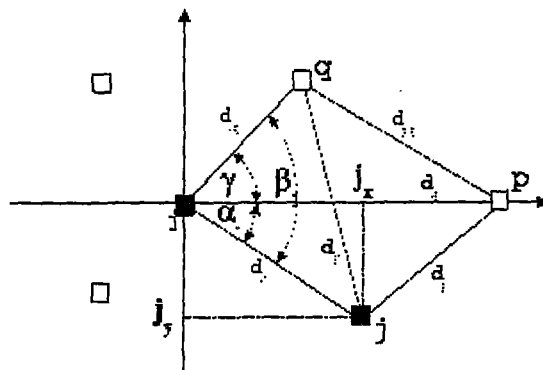


图 6

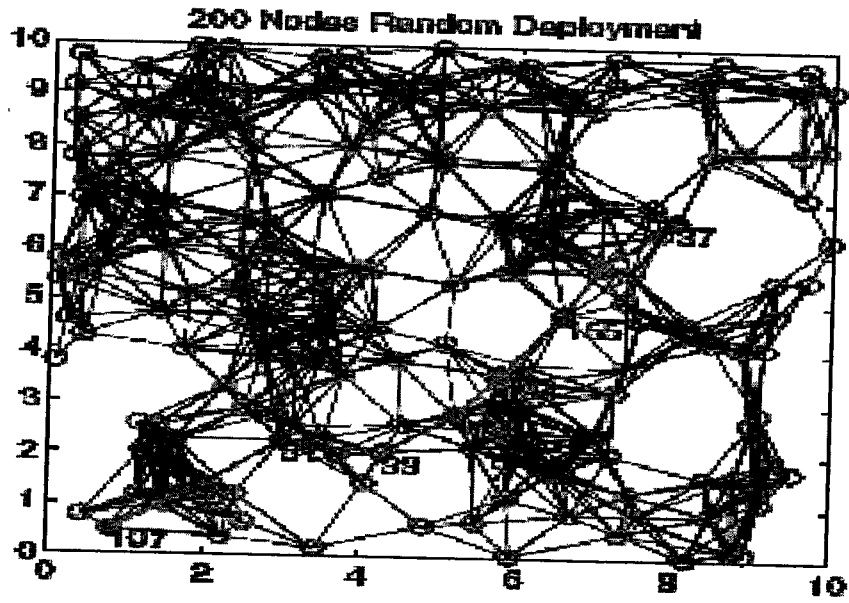


图 7

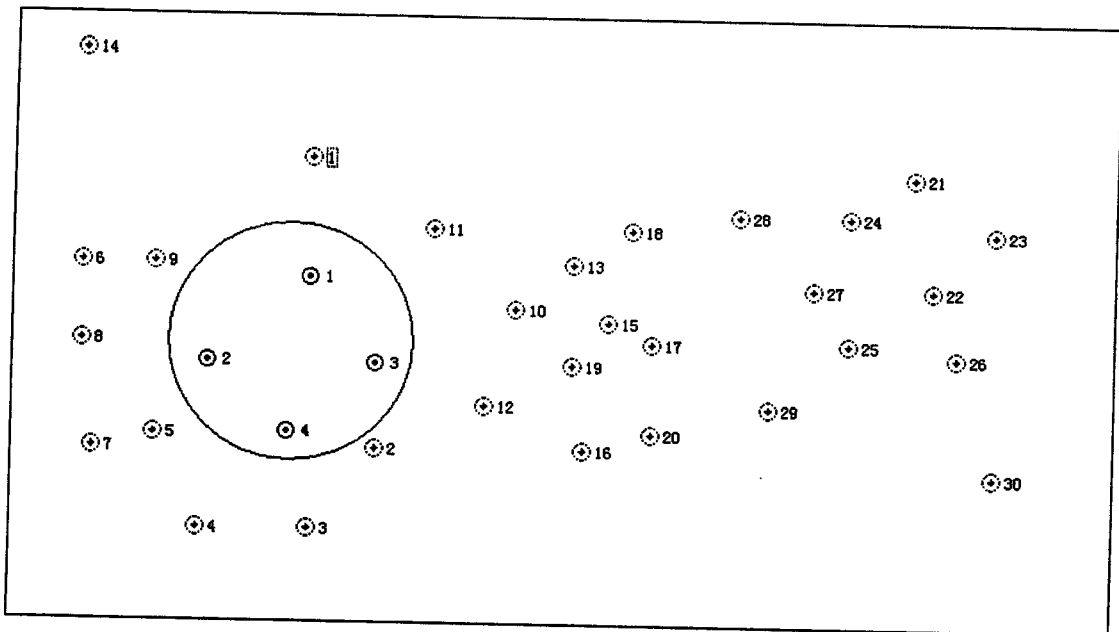


图 8

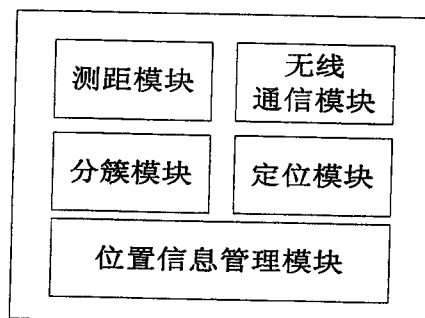


图 9

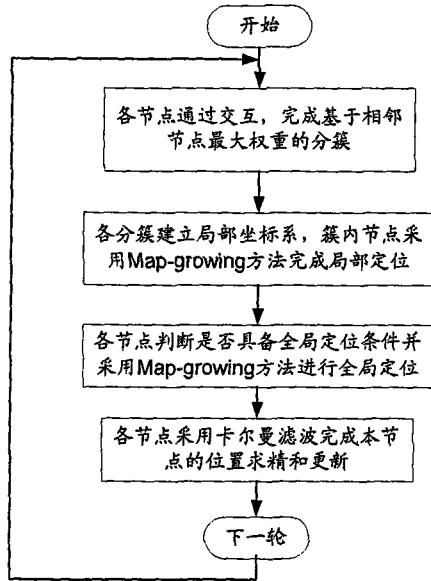


图 10

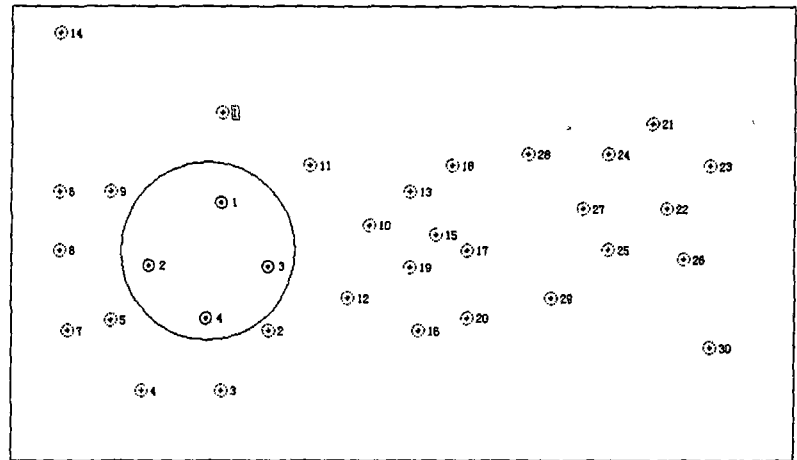


图 11

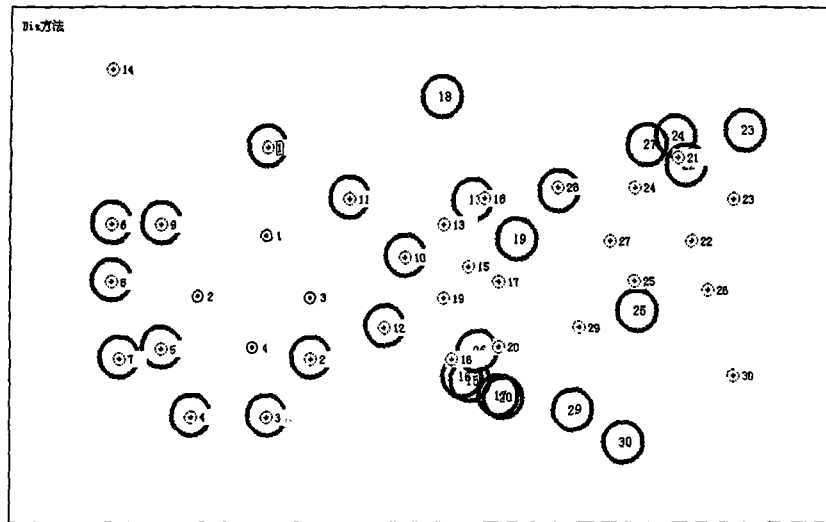


图 12

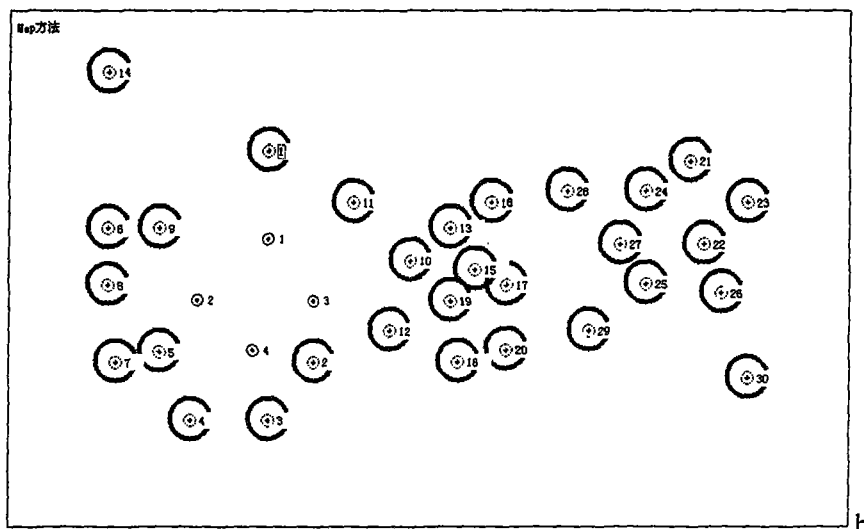


图 13

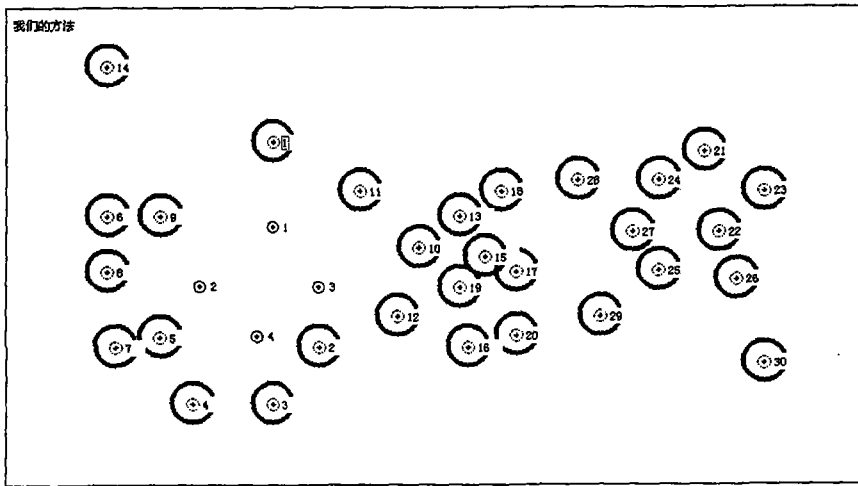


图 14

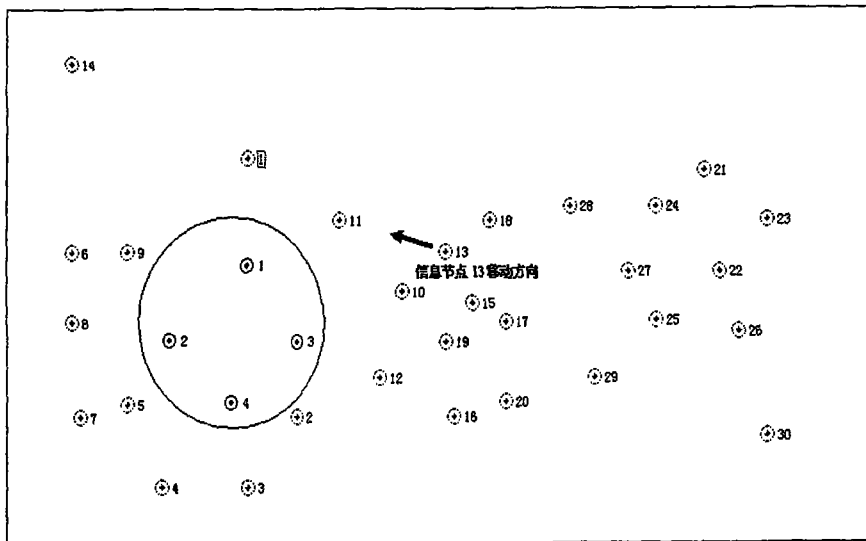


图 15

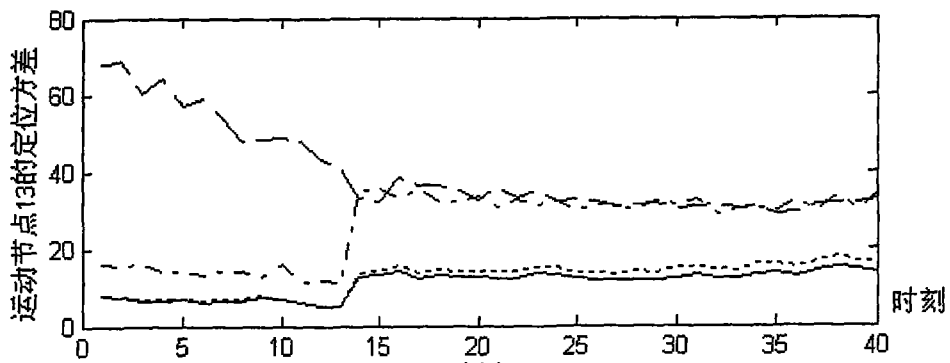


图 16

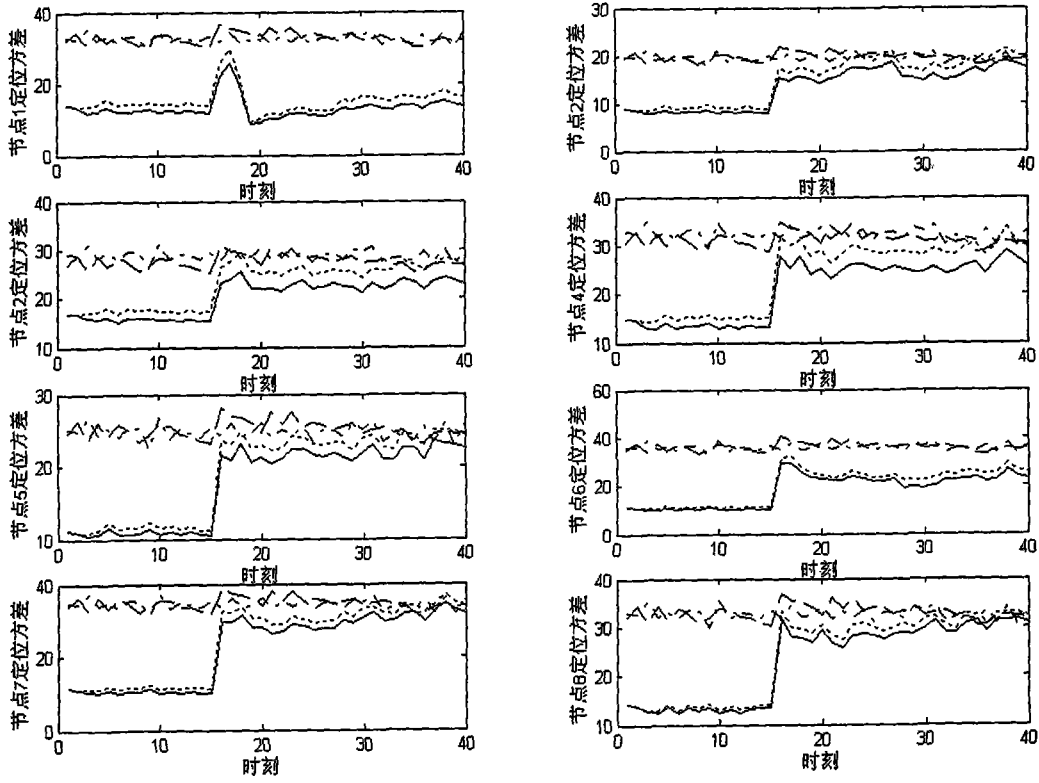


图 17

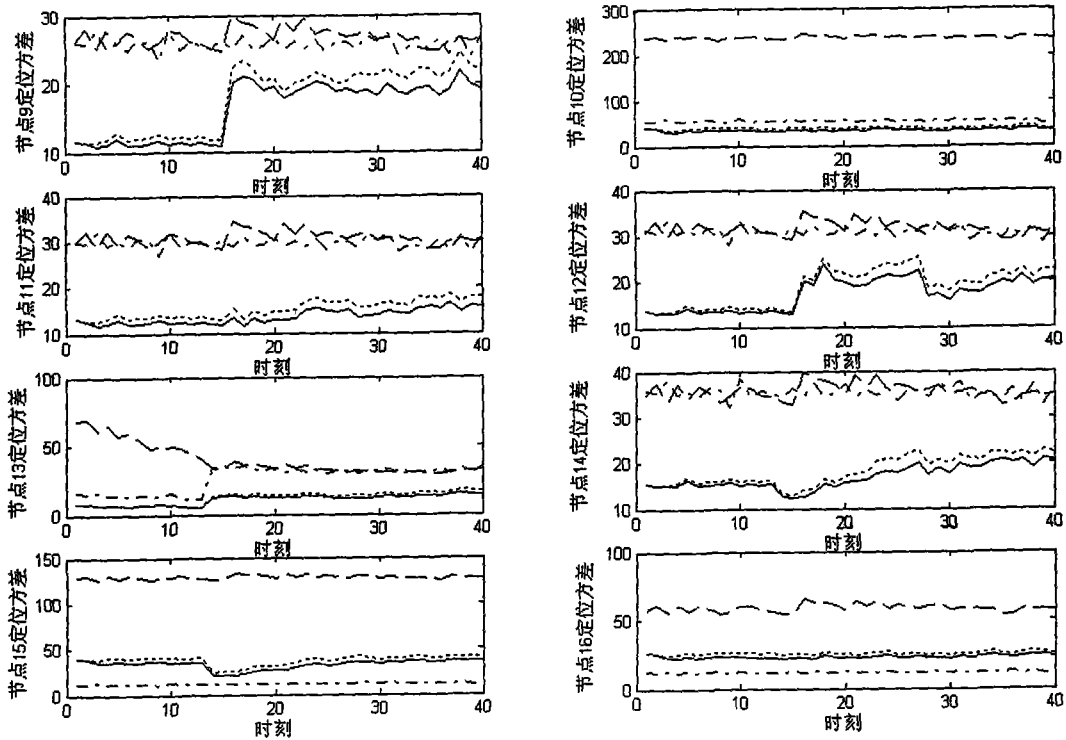


图 18