

障碍空间中的移动对象位置预测

李实吉 秦小麟 施峻严

(南京航空航天大学计算机科学与技术学院 南京 210016)

摘要 移动对象的运动基本是在障碍空间里进行的。近几年,已有较多针对障碍空间中范围查询、近邻查询、聚类查询等的研究,但是目前尚没有对障碍空间中移动对象的位置预测进行研究。为此,研究了障碍空间中移动对象的位置预测;采用灰模型与线性模型相结合的办法,提出了一种基于R树的高效的剪枝算法;根据移动对象的运动规律性,提出了几条剪枝策略,从而大大减少了需要处理的障碍对象个数。最后,通过实验验证了算法的准确性和高效性。

关键词 障碍空间,移动对象,预测,灰模型

中图分类号 TP311.131 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.07.045

Location Prediction of Moving Objects in Obstructed Space

LI Shi-ji QIN Xiao-lin SHI Jun-yan

(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract Most moving objects are typically influenced by obstacles. Recently, existing researches are mainly on range query, nearest neighbor query and spatial clustering query in obstructed space. However, there is no research on location prediction of moving object in obstructed space. This paper focused on location prediction of moving object in the presence of obstacles. We proposed a efficient pruning method based on R-tree using gray model and line model. Considering the trajectories of moving object are regular, we proposed several pruning strategies which can greatly reduce the number of searched obstacles. Finally, our experiment results demonstrate the accuracy and efficiency of the proposed approach.

Keywords Obstructed space, Moving object, Prediction, Gray model

1 引言

近年来,移动对象^[1]的研究取得了许多进展。移动对象的运动空间一般被抽象成理想的欧氏空间和受限空间,其中,受限空间又包括路网空间和障碍物空间等,现实中的移动对象基本都在这两类空间里进行,本文关注障碍空间。比如战场空间就可以抽象成为障碍空间,对海战场来说,障碍就是海岸、岛屿、礁浅、沉船、渔礁、水下钢桩、水雷等,战场中的对象可以抽象为移动对象。目前,已经有较多针对障碍空间中范围查询、近邻查询、聚类查询等的研究,但是尚没有对障碍空间中移动对象的位置预测进行相关研究。在战场中,为了帮助指挥人员理解战场情况、预测态势演化,并辅助指挥人员对战场情况进行监控,预测移动对象的位置显得特别重要。

目前,在理想的欧式空间和路网空间中,关于位置预测已经有较多的研究。线性回归预测模型是最早提出的一种位置预测模型,这类方法本质上都是对移动对象的运动习惯进行仿真,构造复杂的回归函数,主要缺陷在于模型和运动函数构造复杂。灰色理论提出一种处理不完全信息的理论方法。灰色预测^[2]就是利用灰色系统所做的预测,灰色预测使用微分方程所描述的动态方程,比如基于灰色系统的理论模型 GM

(1,1)。以神经网络为基础的轨迹预测方法也不断被提出^[3]。在移动预测问题上,多阶Markov模型和以多阶 Markov 模型为基础的回退模型也被广泛研究^[4]。近几年,移动对象位置预测方法中引入了数据挖掘的思想,其中以基于频繁轨迹的位置预测技术^[5]为代表。元胞自动机(GCA)模型^[6]考虑了网络约束对移动对象的影响,是一种在公路网中对移动对象进行位置预测的模型。

在对障碍空间的研究中,文献[7]首先系统地研究了障碍范围查询、最近邻查询、e-距离 Join、最近对查询等常见空间查询。文献[8]研究了障碍空间中的连续障碍最近邻查询。文献[9]研究了障碍空间中的移动 k 近邻查询。文献[10]研究了连续障碍范围查询。文献[11]给出了一种障碍空间中不确定对象的连续最近邻查询方法。文献[12]研究了障碍空间中的自适应的聚类算法。文献[13]给出了一种障碍反向最近邻查询处理方法。文献[14]研究了障碍空间中的连续最近邻搜索。文献[15]给出了一种障碍空间中的反 k 最近邻查询方法。

目前有一些针对室内空间移动对象预测的算法研究^[18,19],其也需对障碍物进行处理。室内空间也是一种受限空间,主要特点为空间有限并且被墙分割成不同的区域,区域

到稿日期:2013-09-26 返修日期:2014-03-03 本文受国家自然科学基金项目(61373015),国家教育部高等学校博士学科点专项科研基金资助项目(20103218110017),江苏高校优势学科建设工程资助项目(PAPD),中央高校基本科研业务费专项基金项目(NP2013307)资助。

李实吉(1986—),男,硕士生,主要研究方向为空间/时空数据库、移动对象数据库,E-mail:lshjiii@nuaa.edu.cn;秦小麟(1953—),男,教授,博士生导师,主要研究方向为空间/时空数据库、分布式数据管理与安全;施峻严(1990—),男,硕士生,主要研究方向为移动对象数据库、RFID定位。

之间通过门、走廊等实现连通,包括二维到多维(即单层与多层),各维度之间只有简单的通道(楼梯、电梯等)。而本文关注的障碍空间指的是在二维理想欧氏空间(无限自由平面)的基础上,存在着障碍物(比如湖泊、沼泽等),移动对象必须绕过障碍物运动。其与二维室内空间的主要区别在于空间的有限与无限,障碍物种类不一样,形成的障碍空间也不一样。在室内的运动模式也与在障碍空间中不一样,同样是做预测,室内空间比较关注移动对象在某个区域(房间等)内,而障碍空间直接关注移动对象的具体位置。所以室内空间移动对象预测的研究对障碍空间只具有参考意义,而不能直接使用。

目前还没有对障碍空间中的移动对象的位置预测进行研究。在移动对象针对未来的查询中,预测移动对象未来的位置是非常重要的。在障碍空间中,如何预测移动对象的位置是如今亟待解决的问题。

本文给出一种基于 R 树的高效的预测障碍空间中移动对象位置的方法,提出了九方位图剪枝的方法,大大减小了搜索 R 树的代价,利用移动对象运动的规律性使在不久的将来速度和方向一般变化较小,由移动对象最近障碍邻居增量地得到局部预测可视图 VG,并由 VG 计算出移动对象的预测位置。

2 问题定义

在给定的空间中存在着障碍物和移动对象,移动对象不能穿过障碍物内部运动。设障碍物集合为 $O = \{o_1, o_2, \dots, o_m\}$,不失一般性,本文假设障碍物为矩形,且障碍物与障碍物之间不相交。设移动对象集合为 $MP = \{mp_1, mp_2, \dots, mp_n\}$,其中的每个 mp_i 由一系列快照 (id, t_j, p_j) 组成,其中 id 是指移动对象的标识, t_j 是指采样时间戳, p_j 是指在采样时刻的位置。本文只考虑二维移动对象,即 p_j 为二维点坐标 (x_j, y_j) ,设 $mp_i = \{(id, t_1, p_1), (id, t_2, p_2), \dots, (id, t_k, p_k)\}$ 。本文研究的位置预测问题是在包含障碍物集 O 的空间中,估算某个移动对象 mp_i 在未来一段时间 interval 之后可能移动到的位置。

移动对象运动过程中会碰到障碍物阻挡,需要绕着障碍物移动。如图 1 所示,移动对象 mp 的前进过程方向上有障碍物 O ,所以 mp 必须绕过障碍物 O ,可以经由 $p_1 \rightarrow p_3$ 或者 $p_2 \rightarrow p_4$ 绕过障碍物 O 。在图 1 中,移动对象 mp 当前运动方向偏向 p_2 ,在做预测的时候认为 mp 将会选择 $p_2 \rightarrow p_4$ 绕过障碍物 O 。

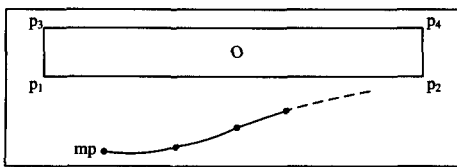


图 1 障碍空间中运动

本文假设障碍物为矩形,主要考虑是一般用 R 树来索引障碍物空间,用最小包围矩形 MBR 来表示障碍物。本文关注的不是具体的障碍,而是 MBR。当然,如果换成具体形状的障碍物,本文算法也很容易扩展,就是把本文提出的算法作为过滤步骤首先执行,再加一步精练操作,就能得到结果。

给定一个二维空间的障碍物集合 O (包含互不相交的矩形)、一个起始点 p_{start} 和一个目的点 p_{end} ,要寻找一条从起点 p_{start} 到目的点 p_{end} 的最短路径,这条路径不能通过 O 中的任何障碍物,最短路径问题在计算几何^[16]里已经被广泛研究。

图 2 显示了一个包含两个障碍物的可视图 VG 例子。可视图 VG 包含 O 中所有障碍物的顶点以及 p_{start}, p_{end} 。VG 中的两个点如果是可见的(即两个点的连线没有穿过任何障碍物),那么以这两个点为顶点的边也在 VG 中(障碍物的边也在 VG 中)。最短路径的求解实际上就是求可视图 VG 中的最短路径。

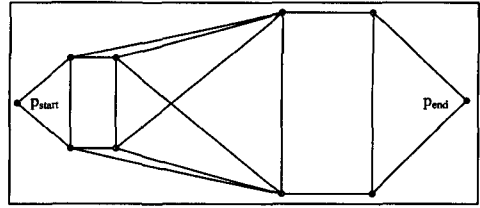


图 2 可视图 VG

做预测需要考虑所有路径,可见构建可视图 VG 至关重要。由文献[6,16]可知,求解全局可视图非常困难(复杂度很高),考虑到实际情况,可以不求全局可视图,只求解对结果有影响的局部可视图。下一小节将介绍两种求解局部可视图过程中使用的剪枝策略。

3 基于九方位图的剪枝策略

在 MBR 的基础上可得到九方位图。设 (x, y) 为移动对象当前点坐标,障碍 MBR 的坐标范围为 x 轴 $[x_1, x_2]$, y 轴 $[y_1, y_2]$,如图 3 所示。九方位是移动对象相对于 MBR 的方位,由 x 轴(左,中,右)与 y 轴(上,中,下)正交得到 9 个方位。在这 9 个方位里,(4 内部)对预测算法是无意义的(移动对象无法通过障碍)。

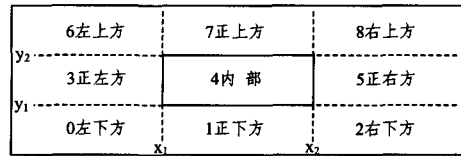


图 3 九方位图

下面定义移动对象的运动方向,如图 4 所示,在二维坐标系内,弧度范围为 $[-\pi, \pi]$, x 轴正方向为弧度 0,负方向为弧度 π 与 $-\pi$ 的重合处, y 轴正方向弧度为 $\pi/2$,负方向为 $-\pi/2$ 。移动对象的运动方向为与 x 轴正方向所成夹角的弧度 rad,由 x 轴弧度为 0 开始逆时针旋转弧度增大到 π ,顺时针旋转则减小到 $-\pi$ 。

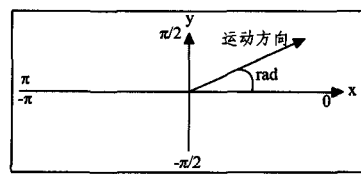


图 4 移动对象运动方向示意图

结合移动对象的当前运动方向,可以得到剪枝策略。根据现实世界的场景,在不久的将来移动对象的运动方向保持不变或者变化较小,可称之为方向保持性。考虑图 3 的障碍物场景,假设移动对象当前位置在(0 左下方),即障碍物在移动对象的右上方。考虑图 4 所示的运动方向(第一象限, $[0, \pi/2]$),移动对象有可能碰到障碍物,则需考虑改变方向,决定移动对象改变后的方向是障碍 MBR 左上方和右下方的顶点(即 $(x_1, y_2), (x_2, y_1)$ 两点)连成的线。若移动对象运动方向为图 4 中剩下的 3 个象限 $([\pi/2, \pi], [-\pi, 0])$,由于移动对象

具有方向保持性,因此图 3 中的障碍物对移动对象没有影响,则可以剪枝。考虑其余的情况,即可得表 1 的剪枝策略(PruneCondition1),表 1 中第 2、3 列表示移动对象当前位置所在的范围,第 4 列的弧度范围为移动对象当前运动方向。前面的例子即是序号为 0 的行。

表 1 九方位图坐标范围与剪枝策略 1

序号	方位名	坐标范围	剪枝策略(与运动方向相关)
0	左下方	$x \leq x_1$ $y < y_1$	$[\pi/2, \pi], [-\pi, 0]$ 剪枝 $[0, \pi/2]$ 考虑 $(x_2, y_1), (x_1, y_2)$
1	(正)下方	$x_1 < x < x_2$ $y < y_1$	$[-\pi, 0]$ 剪枝 $[0, \pi]$ 考虑 $(x_2, y_1), (x_1, y_1)$
2	右下方	$x_2 \leq x$ $y < y_1$	$[0, \pi/2], [-\pi, 0]$ 剪枝 $[\pi/2, \pi]$ 考虑 $(x_2, y_2), (x_1, y_1)$
3	(正)左方	$x < x_1$ $y_1 < y < y_2$	$[\pi/2, \pi], [-\pi, -\pi/2]$ 剪枝 $[-\pi/2, \pi/2]$ 考虑 $(x_1, y_1), (x_1, y_2)$
5	(正)右方	$x_2 < x$ $y_1 < y < y_2$	$[-\pi/2, \pi/2]$ 剪枝 $[\pi/2, \pi], [-\pi, -\pi/2]$ 考虑 $(x_2, y_2), (x_2, y_1)$
6	左上方	$x \leq x_1$ $y_2 \leq y$	$[0, \pi], [-\pi, -\pi/2]$ 剪枝 $[-\pi/2, 0]$ 考虑 $(x_1, y_1), (x_2, y_2)$
7	(正)上方	$x_1 < x < x_2$ $y_2 \leq y$	$[0, \pi]$ 剪枝 $[-\pi, 0]$ 考虑 $(x_1, y_2), (x_2, y_2)$
8	右上方	$x_2 \leq x$ $y_2 \leq y$	$[0, \pi], [-\pi/2, 0]$ 剪枝 $[-\pi, -\pi/2]$ 考虑 $(x_1, y_2), (x_2, y_1)$

表 1 中的剪枝策略实际上是直接使用象限来进行剪枝,只要障碍物不在移动对象前进的象限中,就可以直接被剪枝。由于象限的角度范围不用再计算,可以直接按表 1 的规则判断得出剪枝结果。

如果障碍物在移动对象前进的象限中,根据表 1 的策略可知不能直接剪枝,即表 1 中需要特别考虑的情况。再考虑上面序号为 0 的行的例子,移动对象的运动方向为第一象限,障碍物也在第一象限中,这时决定移动对象改变后的方向是障碍 MBR 左上方和右下方的顶点(即 $(x_1, y_2), (x_2, y_1)$ 两点,可见表 1)连成的线。计算移动对象当前运动方向是否和障碍物相交,即是否与表 1 中所考虑的两点连成的线相交。如果不相交,障碍物不会影响移动对象的运动,可以直接剪枝,即剪枝策略 2(PruneCondition2)。如果相交,那此障碍物就必须放到局部可视图中进行计算。

4 障碍空间中的移动对象预测方法

给定一个障碍物集 O 、移动对象数据集 MP ,对于其中的某个移动对象 mp 和一个时间段 $interval$,一个预测结果返回 $interval$ 时间后的移动对象 mp 的位置(绕过障碍物集)。在进行更进一步的细节讨论之前,先给出算法的总体思路。若移动对象目的地明确,则求出最短路径来预测将来位置。若目的地不明确,则按下面步骤求解:(i) 求出移动对象当前位置、运动方向、运动速度、预测距离;(ii) 计算局部预测可视图 VG ;(iii) 若 VG 为空,则采用 $GM(1, 1)$ 模型进行预测,否则根据夹角较小策略计算预测路径,求出预测位置。

在现实生活中,有些情况下移动对象的目的地是明确的。对于目的地明确的移动对象,采用最短路径来做预测。根据当前位置和目的地求出局部可视图,采用某种算法求出最短路径,根据一定的规则求出速度(比如采取当前速度),求出预测距离(速度 * 预测时间),由预测距离就可在最短路径上计算出移动对象的将来位置。

对于目的地不明确的移动对象,下面给出了一个基于 R 树的算法。假设障碍物集是用 R 树进行索引的。在现实世

界中,移动对象运动遵循各自的规律,并且在不久的将来,移动对象的速度和方向一般保持不变或者变化较小,本文基于这个观察前提做预测。

4.1 局部预测障碍物集

下面给出求局部预测障碍物集的算法。这里的局部预测障碍物指的是对预测有影响的障碍物。假设移动对象保持当前的速度和方向向前运动,与之相交的障碍物就会被视为对预测有影响的障碍物,移动对象要绕过此障碍物。

根据移动对象的当前位置、运动方向、预测距离,找出障碍物集 O 中对预测有影响的 MBR,组成集合 O' 。集合 O' 是由离移动对象当前位置最近的且对预测有影响的障碍物组成的。假设障碍物集用 R 树做索引,使用一个队列 Q 保存即将访问的节点。首先,根节点里的 MBR 进入 Q ,并按与移动对象当前位置的距离大小升序排列。从 Q 中取出最小距离的节点进行扩展。如果此节点满足剪枝条件 1(九方位图)和剪枝条件 2(与运动方向不交叉),则直接剪枝,不进行扩展;否则,把它的孩子节点加到 Q 中,保持 Q 按升序排列。下一个距离最近的节点从堆中被取出,继续扩展。如果找到的是叶子节点,且无法被剪枝,就加到结果集 O' 中,并根据该节点修改移动对象的运动方向。算法终止的条件是当前最小距离大于预测距离超过预测距离的障碍物对预测没有影响。

算法 1 LocalPredObs($R, pos, rad, dis, \&obs$)

输入:障碍物集的索引树 R ,移动对象当前位置 pos ,移动对象当前运动方向 rad ,移动对象预测距离 dis (当前速度 * 预测时间)

输出:局部预测障碍物集 obs

1. Insert root's MBR into Q //保持 Q 升序排列
2. While Q non-empty
3. De-queue m from Q // m 与当前位置距离 $dis(m, pos)$ 最近
4. If $dis(m, pos) > dis$
5. Break while //中断 while 循环
6. If m not satisfy PruneCondition1 and PruneCondition2 //见第 2 节
7. If m not a leaf
8. Insert m 's children into Q
9. Else
10. If $dis(m, pos) \leq dis$
11. Add m into obs
12. End LocalPredObs

4.2 计算预测路径

假设局部预测障碍物集 O' 已经求出,并且局部预测可视图 VG 也已经求出。预测路径是可视图中路径长度大于预测距离并且与移动对象当前运动方向的夹角最小的路径。这里的夹角是指移动对象当前位置与路径终点的射线与移动对象当前运动方向的夹角,如图 5 中的 rad_1 与 rad_2 。如果只有一个障碍物,那么只有两条路径,且路径无需扩展,取夹角较小的路径。如果有多个障碍物,那么需要扩展路径,直到当前路径长度大于预测距离或者当前路径无法扩展。

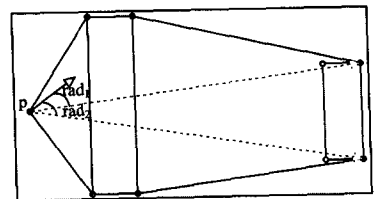


图 5 移动对象预测运动方向与当前运动方向的夹角

创建一个局部可视图 VG,包括移动对象的当前位置 pos 和 O' 。从移动对象当前位置 pos 出发,将与 pos 直接相连的边加到一个路径队列 Q 中。此队列保存的是从 pos 出发的路径,并且按照对象当前位置与路径终点的射线与移动对象当前运动方向的夹角升序排列。取出夹角最小的路径,扩展该路径,计算夹角值,按升序插入 Q。继续取出差值最小的路径进行扩展。算法的终止条件是取出的当前路径不需要扩展(前方无新的障碍物)或者是当前路径长度大于预测距离。把此路径当作预测轨迹,根据预测距离,就可以计算出预测位置。

算法 2 ComputePredPath(VG, pos, rad, dis, &path)

输入:局部预测可视图 VG,移动对象当前位置 pos,移动对象当前运动方向 rad,移动对象预测距离 dis(当前速度 * 预测时间)
输出:预测路径 path

1. Insert each edge(pos, p) into Q//保持 Q 升序排列
2. While Q non-empty
3. De-queue CurPath from Q// CurPath 与当前运动方向夹角最小
4. If length(CurPath) > dis or path cannot expand
5. path = CurPath
6. Break while //中断 while 循环
7. Expand path and insert new path into Q
8. End ComputePredPath

4.3 障碍空间中移动对象位置预测算法 PreLocationInObs

根据上面两个算法,现在给出障碍空间中移动对象位置预测的完整算法。

首先计算出移动对象的当前运动方向 rad (与 x 轴的夹角)。移动对象是采用 (id, t_i, p_i) 的快照形式存储的,假设当前位置为 (id, t_k, p_k) ,则上一个位置为 (id, t_{k-1}, p_{k-1}) ,那么可以根据 p_k 与 p_{k-1} 计算出方向。假设 p_k 与 p_{k-1} 的坐标分别为 (x_k, y_k) 与 (x_{k-1}, y_{k-1}) ,则运动方向为射线 $p_{k-1}p_k$ 与 x 轴正方向的夹角 rad 。计算当前速度 $speed$,并计算预测距离 dis (当前速度 * 预测时间)。

```
dx = xk - xk-1; dy = yk - yk-1; dt = tk - tk-1;
dxy = sqrt(dx * dx + dy * dy); /* sqrt 为求平方根函数 */
rad = acos(dx/dxy); /* acos 为求反余切函数 */
speed = dxy/dt;
dis = speed * interval; /* interval 为预测时间 */
```

然后调用算法 LocalPredObs 求局部预测障碍物集 O' 。若集合 O' 为空,则说明当前运动在给定的规则下不受障碍物影响,那么采用 GM(1, 1)模型进行预测。若集合 O' 不为空,则根据移动对象当前位置 pos 和局部障碍物集 O' 计算出局部预测可视图 VG。调用算法 ComputePredPath 计算出预测路径。把此路径当作预测轨迹,根据预测距离,计算出预测位置。

算法 3 PreLocationInObs(R, mp, interval, &location)

输入:障碍物集的索引树 R,某个移动对象 mp,预测时间 interval
输出:移动对象 mp 在 interval 时间后的位置 location

1. compute pos, rad, dis//当前位置,当前方向,预测距离
2. LocalPredObs(R, pos, rad, dis, &obs)
3. if obs is empty //局部障碍物集合为空
4. GM(1, 1)//采用灰模型进行预测
5. else
6. ComputePredPath(VG, pos, rad, dis, &path)
7. ComputePreLocation(path, dis, &location)//根据预测距离 pre-

Dis 在 prePath 中算出 location,起点为 0

8. End PreLocationInObs

5 实验与分析

实验主要考察提出的算法的预测准确性以及算法的高效性。实验环境为 Intel Core i3-2120 CPU、4G 内存与运行 32 位 Windows XP 系统的 lenovo ThinkVision PC。本文算法采用 C 语言实现,Microsoft Visual C++ 6.0 编译。

本文使用 GSTD^[17]产生的模拟数据集验证所提出的算法。GSTD 产生包含 (id, t_i, p_i) 快照的移动对象 mp ,起始点为 (id, t_1, p_1) 。GSTD 假定运动空间为单元正方形 $[0, 1]^2$,这个空间为正方形,并假定左下角坐标为 $(0, 0)$,右上角坐标为 $(1, 1)$ 。移动对象在这个空间中运动,当运动超过这个范围时,有 3 种处理策略:(1)不处理,即认为该移动对象丢失,类似于监控;(2)穿越边界,从另一边回来。将左右(上下)边界视为同一边界,移动对象向左(上)边界运动到出界,即从右(下)边界进来向左(上)运动,类似于球体。(3)调整策略。当运动超出范围时,则采用一定的方法将位置调整到 $[0, 1]^2$ 空间中。本文采用第一种策略,比较接近现实情况。运动时间从 0 到 1(即单元间隔)。GSTD 可以产生不同数量、不同最大速度、不同分布的移动对象与不同数量的随机分布的障碍物集,是一个被广泛使用的移动对象数据产生器。这里移动对象最大速度的单位为(单元长度/单元间隔),意思是移动对象每单元间隔移动多少个单元长度,这里的单元长度为 1,单元间隔也为 1,是 GSTD 数据产生器设定的。移动对象分布的取值为高斯(Gaussian)分布、偏态(skewed)分布和随机(random)分布 3 种。障碍物集为互不相交的矩形,使用 R 树索引障碍物集。实验结果显示本文提出的算法处理的准确性与高效性。

目前尚没有对障碍空间中移动对象的位置预测进行相关研究,本文的实验只考虑所提出的算法的准确性与高效性。根据 GSTD 所产生的数据集,分析得出影响预测准确性(算法准确性)的因素有预测时间的长短、移动对象的最大运动速度、障碍物的数量、不同的分布等。影响算法的高效性的因素有障碍物的数量等。本文根据这几个影响因素设计并做了相关实验。

首先考察算法的预测准确性(算法准确性)。每次用 GSTD 产生 1000 个移动对象,每个移动对象有 100 快照,每次都产生 3 种不同分布的数据进行实验。预测误差用来衡量实验统计距离误差在 20%且角度误差在 $\pi/6$ 以下的移动对象的数量。

距离误差 = (当前位置与预测位置的距离) / (当前位置与预测时刻的实际位置的距离)

角度误差 = 当前位置与预测位置的射线和当前位置与预测时刻实际位置的射线的夹角

图 6 给出了实际轨迹与预测轨迹在运动空间中的投影。这里固定障碍物数量为 100 个,固定移动对象最大速度为 0.2 单位,固定预测时间长短为 ΔT (采样间隔)。从实验结果中选取了两个有代表性的移动对象的部分轨迹进行显示。图 6 中阴影矩形为 GSTD 随机产生的障碍物,两条轨迹中一条绕过障碍物,一条没有遇到障碍物。实线代表实际轨迹,虚线代

表预测轨迹。从图 6 中可以看出,预测结果还是比较准确的。以下将通过统计结果来衡量预测结果的准确性。

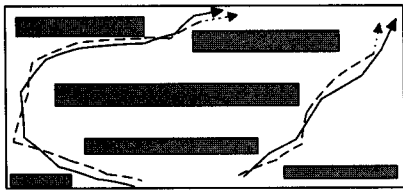


图 6 实际轨迹与预测轨迹

图 7 给出预测时间长短对预测准确性的影响。这里固定障碍物数量为 100,固定移动对象最大速度为 0.2 单位。从图 7 中可以看出,在移动对象快照采样时间间隔 ΔT 之内,预测效果最好;超出 ΔT 之后,预测效果下降。这是由于利用灰模型 GM(1,1) 进行预测的最佳范围是从当前时间 t 到按采样时间间隔 ΔT 的下一个时间点,即区间 $(t, t + \Delta T)$,其中, t 是当前时间, ΔT 为采样间隔。

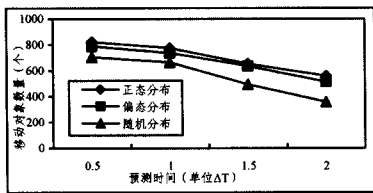


图 7 预测时间长短对预测准确性的影响

图 8 给出障碍物数量变化对预测准确性的影响。这里固定预测时间长短为 ΔT ,固定移动对象最大速度为 0.2 单位。从图 8 中可以看出,随着障碍物数量的增加,预测准确性越来越差。这是因为随着障碍物的增多,移动对象前进过程中碰到的障碍物也越来越多,方向的变化越来越频繁。本文提出的预测算法采用的方向改变策略比较单一,没有考虑到方向改变的多样性,因此准确性就会越来越差。

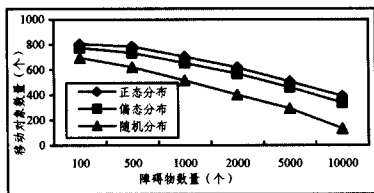


图 8 障碍物数量对预测准确性的影响

图 9 给出移动对象最大速度变化对预测准确性的影响。这里固定预测时间长短为 ΔT ,固定障碍物数量为 100。从图 9 中可以看出,随着移动对象最大速度的增加,预测准确性越来越差。这是因为随着最大速度的增加,移动对象速度的变化范围越来越大,而本文的预测算法没有考虑移动对象的速度变化,而只是简单地认为移动对象保持当前的速度,因此准确性就会越来越差。

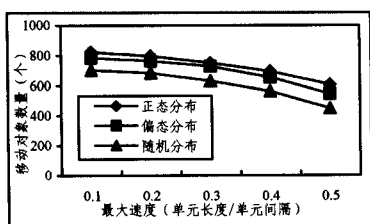


图 9 最大速度对预测准确性的影响

综合比较图 7—图 9 可以观察到,正态分布与偏态分布的预测准确性相当,而对随机分布的预测准确性比较差。这是因为正态分布与偏态分布的数据集中在某个区域,随机分布则不具有这种性质,而本文的算法认为移动对象会保持当前的速度和方向,而不是随机变化的。

综上所述,本文提出的算法在预测时间小于等于 ΔT 、障碍物数量较少、移动对象最大速度较小、移动对象运动在一定分布而不是随机分布的情况下,预测准确性较高。

然后考察算法的高效性。用 GSTD 产生 100 个移动对象,每个移动对象有 50 个快照,固定最大速度为 0.2 单位,固定预测时间长短为 ΔT ,产生 3 种不同分布的数据进行实验。图 10 给出障碍物数量变化对算法运行时间的影响。从图 10 中可以看出,随着障碍物数量的增加,算法运行时间增加缓慢。这说明采用 R 树及两个剪枝策略起到了很好的效果,使算法能够高速运行。

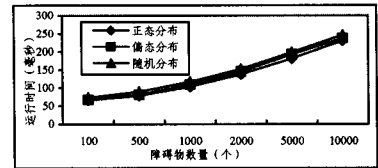


图 10 障碍物数量对算法运行时间的影响

结束语 本文提出了一种基于 R 树的高效的剪枝算法,它将空间划分成九方位图的方法用于障碍空间,根据移动对象运动规律来预测移动对象未来的位置,大幅度减少了需要处理的障碍物的数量。使用 GSTD 产生的多种分布方式的模拟数据的实验结果,表明了所提算法的高效性和准确性。

然而还有以下几个方面需要研究:

(1) 本文采用线性模型和 GM(1,1) 模型进行预测,是在默认不久的将来移动对象一般会保持速度和方向的前提下做出选择的,这与现实世界是有差距的。另外,本文采用固定策略,即认为在碰到障碍物时,移动对象会选择与当前方向夹角较小的路径前进。实际生活中,移动对象改变方向和速度是有一定的概率的。所以引进概率模型或者选择其它更合理的预测模型可能会改善预测结果精度。

(2) 可以研究与预测相关的查询。本文研究的预测是针对单个移动对象的,不涉及多个移动对象之间的操作。而预测态势演化要考虑的却是多个移动对象之间的操作,比如预测障碍范围查询、预测最近邻查询、预测 e-距离 Join、预测最近对查询等常见的查询。

参考文献

- [1] 秦小麟. 移动空间数据类型和操作的初步研究[J]. 计算机科学, 2000, 27(1): 65-68
- [2] 邓聚龙. 灰色预测与决策[M]. 武汉: 华中理工大学出版社, 1989: 13-116
- [3] Hill T, O'Connor M, Remus W. Neural network models for time series forecasts[J]. Management science, 1996, 42(7): 1082-1092
- [4] Song L, Kotz D, Jain R, et al. Evaluating location predictors with extensive Wi-Fi mobility data[C]// Twenty-third Annual Joint

Conference of the IEEE Computer and Communications Societies(INFOCOM 2004). IEEE, 2004, 2:1414-1424

[5] Agrawal R, Imieliński T, Swami A. Mining association rules between sets of items in large databases[J]. ACM SIGMOD Record, ACM, 1993, 22(2):207-216

[6] Chen J, Meng X, Guo Y, et al. Modeling and predicting future trajectories of moving objects in a constrained network[C]//7th International Conference on Mobile Data Management, 2006 (MDM 2006). IEEE, 2006:156-156

[7] Zhang J, Papadias D, Mouratidis K, et al. Spatial queries in the presence of obstacles[M]//Advances in Database Technology-EDBT 2004. Springer Berlin Heidelberg, 2004:366-384

[8] Gao Y, Zheng B. Continuous obstructed nearest neighbor queries in spatial databases[C]//Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. ACM, 2009:577-590

[9] Li C, Gu Y, Li F, et al. Moving k-nearest neighbor query over obstructed regions[C]//Web Conference(APWEB), 2010 12th International Asia-Pacific. IEEE, 2010:29-35

[10] Li Z, Gao Y, Lu Y. Continuous obstructed range queries in spatio-temporal databases[C]//2011 International Conference on System Science, Engineering Design and Manufacturing Informatization(ICSEM). IEEE, 2011, 2:267-270

[11] 李传文, 谷峪, 李芳芳, 等. 一种障碍空间中不确定对象的连续最

近邻查询方法[J]. 计算机学报, 2010, 8:006

[12] Liu Q, Deng M, Shi Y. Adaptive spatial clustering in the presence of obstacles and facilitators[J]. Computers & Geosciences, 2013, 56:104-118

[13] Gao Y, Yang J, Chen G, et al. On efficient obstructed reverse nearest neighbor query processing[C]//Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2011:191-200

[14] Gao Y, Zheng B, Chen G, et al. Continuous nearest-neighbor search in the presence of obstacles[J]. ACM Transactions on Database Systems(TODS), 2011, 36(2):9

[15] 于晓楠, 谷峪, 张天成, 等. 一种障碍空间中的反 k 最近邻查询方法[J]. 计算机学报, 2011, 10:1917-1925

[16] De Berg M, Van Kreveld M, Overmars M, et al. Computational geometry[M]. Springer, 1997:305-315

[17] Pfoser D, Theodoridis Y. Generating semantics-based trajectories of moving objects[J]. Computers, Environment and Urban Systems, 2003, 27(3):243-263

[18] 王志良, 杨溢, 杨扬, 等. 一种周期时变马尔可夫室内位置预测模型[J]. 智能系统学报, 2009, 4(6):521-527

[19] Tran K, Phung D, Adams B, et al. Indoor location prediction using multiple wireless received signal strengths[C]//Proceedings of the 7th Australasian Data Mining Conference-Volume 87. Australian Computer Society, Inc., 2008:187-192

(上接第 186 页)

图 4 对比了在系统中次用户数量不同的情况下的干扰噪声比性能。在对比中可以看出, 优化平均能效的方案, 即本文所提最优方案在整个次用户数量变化区域具有鲁棒性, 具有很强的抑制干扰的能力, 而文献[10]所提方案仅仅在次用户数达到 30 时性能才能够达到本方案的水平, 说明了本方案具有次用户数量方面的鲁棒性。

结束语 本文提出了考虑平均能效的认知无线电次用户频谱接入方案, 此方案对子载波信道和各个子载波信道的功率进行了分配, 同时优化了系统的平均能效和传输速率, 提出的次优算法虽然在性能上有一定的损失, 但实现较为方便, 运算复杂度低。

参 考 文 献

[1] Fu L, Zhang Y J, Huang J. Energy efficient transmissions in MI-MO cognitive radio networks[C]//Proc. Conference on Information Sciences and Systems (CISS). Princeton, New Jersey, USA, Mar. 2012

[2] Huang Y, Li Q, Ma W-K, et al. Robust multicast beamforming for spectrum sharing-based cognitive radios[J]. IEEE Trans. Signal Process., 2012, 60(1):527-533

[3] Miao G, Li G Y, Talwar S. Low-complexity energy-efficient scheduling for uplink OFDMA[J]. IEEE Trans. Commun.,

2012, 60(1):112-120

[4] Xiong C, Li G Y, Zhang S-Q, et al. Energy efficient resource allocation in OFDMA networks[J]. IEEE Trans. Commun., 2012, 60(12):3767-3778

[5] Yang J, Uluks S. Optimal packet scheduling in an energy harvesting communication system[J]. IEEE Trans. Commun., 2012, 60(1):220-230

[6] Tutuncuoglu K, Yener A. Optimum transmission policies for battery limited energy harvesting nodes[J]. IEEE Trans. Wireless Commun., 2012, 11(3):1180-1189

[7] Dhillon H S, Li Y, Nuggeshalli P, et al. Fundamentals of base station availability in cellular networks with energy harvesting[C]//Proc. 2013 IEEE Globecom. 2013

[8] Park S, Kim H, Hong D. Cognitive radio networks with energy harvesting[J]. IEEE Trans. Wireless Commun., 2013, 12(3):1386-1397

[9] Park S, Heo J, Kim B, et al. Optimal mode selection for cognitive radio sensor networks with RF energy harvesting[C]//Proc. 2012 IEEE International Symposium on Personal Indoor and Mobile Radio Communications. 2012:2155-2159

[10] Sultan A. Sensing and transmit energy optimization for an energy harvesting cognitive radio[J]. IEEE Wireless Commun. Lett., 2012, 1(5):500-503