

基于地图非均匀划分的 Android 游戏智能体寻径算法

李红波 赵 宽 吴 渝

(重庆邮电大学计算机学院网络智能研究所 重庆 400065)

摘 要 针对当前 Android 手机 CPU 频率较低、内存较小等手机资源不足的问题,在分层寻径算法的基础上提出了基于地图非均匀划分的智能体寻径优化算法。该优化算法在预处理阶段对游戏地图进行非均匀划分,生成抽象地图,在线搜索阶段首先在抽象地图的关键点中进行寻径,再在每个子图中进行路径搜索,最终到达目标节点。实验结果表明,与 HPA* Enhancements 算法和 KM-A* 算法相比,提出的算法搜索时间更短,遍历点数更少,能够取得更好的效果。最后,在算法中考虑了地形因素,加入了地形因子,让算法更能适应 Android 手机游戏的需要。

关键词 Android 手机,分层寻径算法,非均匀划分,地形因子

中图分类号 TP391 **文献标识码** A

Routing Algorithm for Android Game Agents Based on Non-uniform Partition of Map

LI Hong-bo ZHAO Kuan WU Yu

(Institute of Web Intelligence, Department of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract For Android phones there are problems of insufficient mobile phone resources like the lower CPU frequency and smaller memory. To solve these problems, on the basis of the hierarchical routing algorithm, a routing algorithm for Android game agents based on non-uniform partition of map was presented. In this algorithm, the game map is non-uniformly partitioned to generate the abstract map in preprocessing phase. And in online searching phase, it first finds paths in key points of the abstract map, then searches paths in each sub-map until the destination node is found. The experiment results indicate that compared with the HPA* Enhancements and KM-A*, the algorithm has the advantages of shorter searching time, fewer traverse points and better results. Finally, a topographical factor is taken into consideration in order to adapt this algorithm to the need of the Android phone games.

Keywords Android phones, Hierarchical routing algorithm, Non-uniform partition, Topographical factor

1 引言

随着 Android 手机的普及,人们对手机游戏的需求量越来越大。一款优秀的手机游戏除了需要优美的场景外,还需要有游戏的 AI 设计,其中游戏中智能体的寻径问题是 AI 设计的重要组成部分。

目前,应用于游戏最多的寻径算法是 A* 算法^[1]。林巧民等将 A* 算法添加到 OIGE 游戏引擎寻径模块中,证明 A* 算法能够在游戏中取得很好的效果^[2]。然而,A* 算法理论上不受内存的大小和 CPU 时间片的限制,当地图过大,需要搜索的路径点数过多时,会消耗巨大的手机内存资源。因此,在 Android 手机上资源十分有限且性能低于 PC 机的情况下,传统的搜索算法并不适用。针对这个问题,学者们提出了很多改进的方案,李志建等人通过改变权值的方式控制算法的搜索精度和搜索效率,使 A* 算法的效率得到提高^[3];谭德荣等

人通过一种基于权值的对启发函数的改进,加快了算法的收敛速度^[4];李得伟等人通过逆向搜索和优化估价函数两种方法进行改进,取得了较好的效果^[5]。然而,以上算法是针对 A* 算法本身的改进,并未对地图进行预处理。Adi Botea 等人根据分层寻径的思想提出了 HPA* 算法^[6],其通过离线构造抽象地图的方法大大提高了搜索效率。M. Renee Jansen 和 Michael Buro 提出了 HPA* Enhancements 算法,其被证明能够取得更好的效果^[7]。然而以上两个算法只对地图进行了均匀的划分,没有考虑到地图障碍点的具体分布情况,在某个子图中也可能出现过多的无用搜索。李艳等人提出了 KM-A* 算法,较好地解决了这个问题^[8],但是在进行地图障碍点 k-means 聚类时需要事先人工给定 K 值,而 K 值往往很难判定。因此,本文采用了一种新的方法对地图进行划分,在分层寻径算法的基础上提出了优化寻径算法,并通过实验证明该算法优于 HPA* Enhancements 算法和 KM-A* 算法。

到稿日期:2013-01-15 返修日期:2013-03-18 本文受教育部新世纪优秀人才支持计划项目(教技司[2008]274号),重庆市原创动漫技术及软件(工具)技术研发团队(国科发高[2009]593号)资助。

李红波(1970—),男,硕士,高级工程师,CCF 会员,主要研究方向为数字媒体技术、机器视觉、增强现实,E-mail:lihongbo@cqupt.edu.cn;赵宽(1987—),男,硕士生,主要研究方向为数字媒体技术、增强现实、人工智能;吴渝(1970—),女,博士,教授,CCF 会员,主要研究方向为网络智能、数字媒体技术。

2 相关技术基础

2.1 HPA* 与 HPA* Enhancements 算法

HPA* 算法^[6]采用了分层寻径的思想,整个分为离线阶段和在线阶段。离线阶段将地图进行抽象化,在线阶段负责具体寻径。离线阶段将地图进行均等划分并找出每一个子图的关键点。在线搜索阶段分为两个部分,第一部分在关键点中搜索从起始节点到目标节点的抽象路径,第二部分在每一子图中使用 A* 算法寻径。HPA* Enhancements 算法^[7]在关键点之间采用 Dijkstra 算法寻径,这是因为实验证明在最坏情况下 A* 算法比 Dijkstra 算法表现更差。

HPA* 算法、HPA* Enhancements 算法与 A* 算法相比减少了遍历的节点数,但是对地图进行均匀划分没有考虑到不同地图的障碍分布情况。

2.2 KM-A* 算法

KM-A* 算法是一种基于 A* 算法和 KM-A* 聚类的计算游戏寻路算法^[8]。在预处理阶段采用 k-means 聚类算法^[9]对地图进行划分,寻径阶段首先在关键点之间进行寻径,接着在子图中进行寻径,找出总体接近最优的一条路径。

KM-A* 算法与 HPA* 算法相比降低了资源消耗,但是 k-means 算法有着自身的不足,如全局搜索能力不足、要预先确定 K 值、会受到初始聚类中心影响、难以处理分类属性数据、容易收敛于局部最优解以及聚类结果对初始代表点的选择非常敏感等^[9,10]。

3 基于 Android 手机游戏智能体寻径的优化算法

3.1 本文的研究思路

针对上文提出的在 Android 平台资源有限的情况下地图模型过大会导致消耗过多的手机资源以及过长的搜索时间这一问题,在分层寻径的思想上提出了优化的智能体寻径算法。本算法分为两个阶段,第一阶段为地图预处理阶段,将地图非均匀划分为大小不等的矩形子图;第二阶段是在线寻径阶段,首先找出每个子图的关键点并在关键点之间进行寻径,然后在各子图中进行寻径,最终找到从起始节点到目标节点的路径。

3.2 地图模型的生成

本文通过二维数组 $map[][]$ 对地图进行描述,以 $map[0][0]$ 作为地图的原点。地图模型以正方形网格为基础,数组中的元素 $map[i][j]$ 对应地图上正方形网格 P 的坐标 (j, i) ,当 $map[i][j]$ 为 0 时表示该方形网格 $P(j, i)$ 能够让智能体通过,为 1 表示时 $P(j, i)$ 有障碍物,不能让智能体通过。

3.3 地图预处理阶段

在地图预处理阶段将会把整张地图划分为由许多子图组成的抽象地图。

HPA* Enhancements 算法采用的方法是将地图均匀划分成由 $n * n$ 个网格组成的子图。这样做没有考虑到地图障碍点分布的具体情况。图 1 所示为 HPA* Enhancements 算法在随机地图上的划分结果。

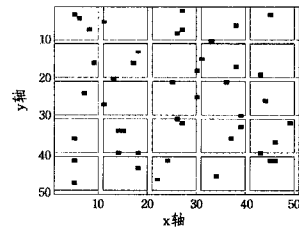


图 1 HPA* Enhancements 算法地图划分结果

KM-A* 算法在地图预处理阶段用 k-means 聚类算法对障碍点进行聚类,再根据聚类的结果对地图进行划分。这样做虽然考虑了地图中障碍点的分布情况,但是 k-means 算法需要人为确定 K 值并且当地图中障碍点过于分散时会划分成很多的子图,导致在线搜索阶段速度过慢。图 2 所示为 KM-A* 算法在随机地图上划分的结果。

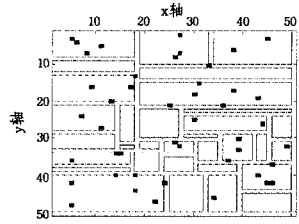


图 2 KM-A* 算法地图划分结果

本文充分考虑了上述两种算法的优缺点,在分层寻径算法的基础上提出了地图非均匀划分算法。算法的总体思路是首先找出地图中不包含障碍点的最大矩形连通区域并用矩形框标识出来,接着在剩下的区域中划分矩形区域。这种方法在考虑到地图障碍点具体分布情况的同时对同一地图划分的结果一致,不需要在划分前设定其他参数。非均匀地图划分算法的步骤如下:

步骤 1 从坐标原点开始遍历地图上所有节点。

1) 当该节点可以通过并且未被扩展时,进入步骤 2;

2) 如果不满足上述条件,继续遍历下一个节点。

步骤 2 将当前节点沿着 x 轴、 y 轴正方向扩展,寻找面积最大的矩形连通区域。

步骤 3 重复步骤 1、步骤 2,直到将地图上所有面积最大矩形连通区域划分出来。

步骤 4 遍历地图上划分后剩下的节点。

1) 如果当前节点未被扩展,则进入步骤 5;

2) 如果当前节点已被扩展,则继续遍历下一个节点。

步骤 5 将当前节点沿着 x 轴、 y 轴正方向扩展,在尚未被划分的区域中寻找最大矩形。

步骤 6 重复步骤 4、步骤 5,直到将地图上所有的点都划分到矩形区域中。

图 3 所示为非均匀划分后地图的各个子图。

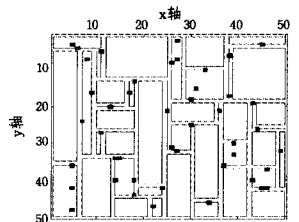


图 3 本文算法地图划分结果

3.4 在线搜索阶段

在线搜索阶段分为 3 个部分。首先,寻找关键点并建立关键点的连通矩阵;接着用 Dijkstra 算法在关键点中进行路径搜索;最后,在通过的子图中用 A* 算法进行寻径。

关键点是子图边界上的点以及起始节点和目标节点,其作用是将地图生成为由关键点组成的抽象地图以供算法的第一次路径搜索。在子图边界的关键点满足两个条件,首先是关键点所对应邻接子图上的点不是障碍点,其次是如果边界上连续非障碍点的个数小于某个特定的值(本文设为 6),则取这段非障碍点的中点节点为关键点,如果个数大于该值,则取这段非障碍点的两端为关键点。

关键点寻找算法的步骤如下:

步骤 1 将起始节点和目标节点设为关键点。

步骤 2 依次检查每个子图的上下左右 4 个边,如果边上有障碍点而邻接子图的对应点不是障碍点,则将该对应点设为临时障碍点。

步骤 3 设边界上障碍点的个数为 n ,则整个边界分为了 $n+1$ 段,判断每段长度如果大于指定的值,则取这段的两端为关键点,如果小于指定的值,则取这段的中点为关键点。

步骤 4 查看每个关键点在邻接子图上所对应的点,如果不为关键点,则将该点设为关键点。

地图上所有关键点找到后需要建立连通矩阵,在同一个子图中的所有关键点以及相邻子图对应关键点是连通的,权值为两个点之间的曼哈顿距离。

在线搜索第二阶段是在连通矩阵中用 Dijkstra 算法搜索从起始节点到目标节点的抽象路径并存储这些关键点;第三阶段是在关键点之间用 A* 算法寻径,最终找到从起始节点到目标节点的路径。图 4 所示为本文算法搜索的路径,灰色点为地图的关键点。

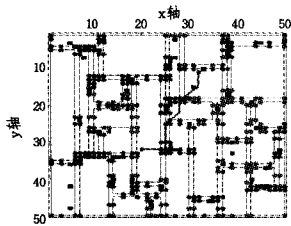


图 4 本文算法寻径结果

在线搜索阶段最坏的情况是遍历地图中的所有点,因此在 $n \times n$ 规模大小的地图上算法的时间复杂度为 $O(n^2)$ 。

3.5 地形因子的加入

在实际的游戏中有各种各样的地形,而智能体通过每种地形所花费的时间不尽相同。然而,HPA* Enhancements 算法与 KM-A* 算法并没有考虑到这一点。描述不同的地形以及在不同地形环境下智能体的路径选择对游戏的真实性和可玩性起着重要的作用。

在地图模型描述时元素的值除了 0 和 1 外,选取其他的整型数值来描述地形。本文以元素的值为 2 代表水塘为例对地图进行描述。本文算法的启发函数变为 $f(n) = g(n) + h(n) + d(n)$ 。 $d(n)$ 为地形因子代表通过这个节点所需花费的代价,当地图数组元素的值为 0 时 $d(n)$ 为 1,当地图数组元素

的值为 2 时 $d(n)$ 为 2。通过加入地形因子为智能体路径的选择提供了更多的依据。图 5 所示为加入地形因子后地图本文算法的路径搜索结果,五角星代表地形因子为 2 的点。

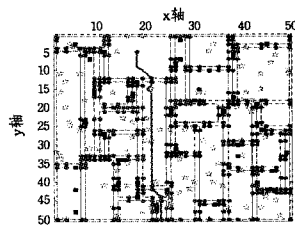


图 5 本文算法加入地形因子后的寻径结果

4 实验结果及分析

首先采用 Intel Xeon 2.40GHz CPU、4GB 内存的普通 PC 在 Matlab R2010b 平台上实现本文算法。将本文算法与 HPA* Enhancements 算法^[7]、KM-A* 算法^[8] 进行比较。

在不加入地形因子情况下考虑两种特殊的地图模型。

第 1 种情况:当地图上无障碍点时,HPA* Enhancements 算法依然将地图划分成由多个相同面积的子图组成的抽象图,KM-A* 算法和本文算法将地图划分成一块,在线搜索阶段 HPA* Enhancements 算法在各子图关键点之间搜索,然后在每个子图中搜索,而 KM-A* 算法和本文算法由于只有起始节点和目标节点两个关键点将直接进行搜索,因此效果均优于 HPA* Enhancements 算法。图 6—图 8 为 HPA* Enhancements 算法、KM-A* 算法、本文算法在地图没有障碍点的情况下的路径搜索情况。

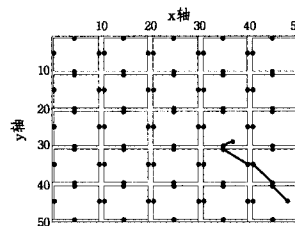


图 6 HPA* Enhancements 算法在无障碍点情况下的寻径结果

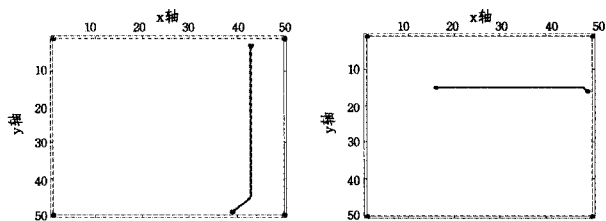


图 7 KM-A* 算法在无障碍点情况下的寻径结果

第 2 种情况:当起地图上障碍点过多、过于分散时,HPA* Enhancements 算法依然均等划分地图,但是 KM-A* 算法需要事先手动输入 K 值,当 K 值过小时整个地图划分成很少的几块,失去了地图划分的意义,当 K 值过大时整个地图划分,块数过多,导致关键点过多搜索速度过慢。本文采用的非均匀划分算法自动将地图进行划分,不会要求人工输入参数。

在 50×50 的大小范围内随机生成地图并随机插入起始

节点与目标节点。一共做了 1400 组实验,为了实验的公平性,将地形因子也加入到 HPA* Enhancements 算法和 KM-A* 算法中,提取了遍历点数、搜索时间以及智能体移动所花费的时间作为参照指标,计算各种指标的平均值。表 1 所列

表 1 各种方法的参数对比

算法	遍历点数	搜索时间/s	智能体移动时间/s
HPA* Enhancements	102.9006	0.0088	28.5940
KM-A*	89.7791	0.0089	29.9357
本文算法	86.3638	0.0078	27.1365

从表 1 可以看出,与 HPA* Enhancements 算法、KM-A* 算法相比,本文算法在遍历点数上,比 HPA* Enhancements 算法少 26.07%,比 KM-A* 算法少 3.80%;在搜索时间上,比 HPA* Enhancements 算法少 11.36%,比 KM-A* 算法少 12.36%;在智能体的移动时间上,比 HPA* Enhancements 算法少 5.10%,比 KM-A* 算法少了 9.35%。以上实验证明,本算法更加优秀。

最后将本文算法移植到 Android 平台上,采用小米手机 1(高通 MSM8260 1.5GHz CPU、1GB RAM、Adreno 220 GPU),利用 OpenGL ES 图形工具包在 Android 版本号为 4.0.4 的操作系统上实现了本文算法,其中黑色点代表障碍点,灰色点代表地形因子为 2 的点。结果如图 9 所示。

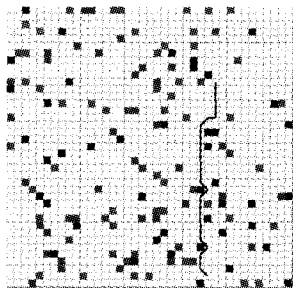


图 9 Android 手机上实现本文算法

结束语 本文研究一种适用于 Android 手机游戏的路径

搜索算法,主要针对 A* 搜索算法在地图模型过大时搜索节点过多而过度消耗手机资源的情况进行研究和改进。对地图进行不均匀划分,形成抽象地图,并分阶段在抽象地图和原始地图上进行搜索。实验结果表明,本文提出的算法减少了搜索时间和遍历点数,较好地解决了寻径算法消耗手机资源较高与 Android 手机资源较为紧张这一矛盾。然而,本算法只是针对静态地图上智能体的寻径,当地图上障碍点临时发生改变时本文算法可能无法得到寻径结果。下一步工作将会在本文算法的基础上充分考虑障碍点变化的因素,提出改进算法。

参考文献

- [1] 宋青,汪小帆. 最短路径算法加速技术研究综述[J]. 电子科技大学学报,2012,41(2):176-184
- [2] 林巧民,林萍,王汝传. 基于 OGRE 的智能游戏引擎的设计与实现[J]. 南京邮电大学学报:自然科学版,2010,30(2):71-76
- [3] 李志建,郑新奇,王淑晴,等. 改进 A* 算法及其在 GIS 路径搜索中的应用[J]. 系统仿真学报,2009,21(10):3116-3119
- [4] 谭德荣,严新平. 变权值加快收敛的路径寻优实时算法[J]. 交通运输工程学报,2004,4(1):118-120
- [5] 李得伟,韩宝明,韩宇. 一种逆向改进型 A* 路径搜索算法[J]. 系统仿真学报,2007,19(22):5175-5177
- [6] Botea A, Muller M, Schaeffer J. Near optimal hierarchical path-finding[J]. Journal of Game Development, 2004, 1: 7-28
- [7] Jansen M R, Buro M. HPA* enhancements[C]//Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment Conference, Stanford, California, USA, 2007: 84-87
- [8] Li Yan, Chen Cai, Li Wen-liang, et al. KM-Aa*: A pathfinding algorithm for computer games based on Aa* and K-means Clustering[C]//Proceedings of 2010 The 3rd International Conference on Computational Intelligence and Industrial Application, Volume 7, Wuhan, China, 2010: 391-394
- [9] 雷小锋,谢昆青,林帆,等. 一种基于 K-Means 局部最优性的搞笑聚类算法[J]. 软件学报,2008,19(7):1683-1692
- [10] 于海涛,贾美娟,王慧强,等. 基于人工鱼群的优化 K-means 聚类算法[J]. 计算机科学,2012,39(12):60-64

(上接第 303 页)

- [3] Balduzzi F, Febraro D. A Combining Fault Detection and Process Optimization in Manufacturing Systems Using First-Order Hybrid Petri Nets[C]//IEEE International Conference on Robotics and Automation. 2001:40-45
- [4] Lucas R F, Giua A, Piccaluga A. Modeling Production Systems with Inventory Using Hybrid Petri Nets [C]//IEEE International Conference on Control Applications. 2001:434-440
- [5] Giua A, Piloni M T, Seatzu C. Modelling and Simulation of a Bottling Plant Using Hybrid Petri Nets[J]. International Journal of Production Research, 2005, 43(7): 1375-1395
- [6] Dotoli M, Fanti M P, Iacobellis G. An Urban Traffic Network Model by First Order Hybrid Petri nets[C]//IEEE International Conference on Systems, Man and Cybernetics. 2008: 1929-1934

- [7] Dotoli M, Fanti M P. Fault Monitoring of Discrete Event Systems by First Order Hybrid Petri Nets[OL]. http://dee.poliba.it/labcontrolli/file_pdf/WPNM.pdf
- [8] Dotoli M, Fanti M P. A First-Order Hybrid Petri Net Model for Supply Chain Management[J]. IEEE Transactions on Automation Science and Engineering, 2009, 6(4): 744-758
- [9] Dotoli M, Fanti M. A Freeway Traffic Control Model by First Order Hybrid Petri Nets[C]//IEEE Conference on Automation Science and Engineering. 2011: 425-431
- [10] 周航,黄志球,胡军,等. 基于 Time Petri Nets 的实时系统资源冲突检测[J]. 计算机研究与发展, 2009, 46(9): 1578-1585
- [11] David R, All H. Discrete, Continuous and Hybrid Petri Nets [M]. Scientific Publishing Services Pvt. Ltd., 2009
- [12] 廖伟志,古天龙. 区间速率连续 Petri 网的有效冲突及其消解[J]. 计算机科学, 2006, 33(10): 221-224