Journal of Computer Applications
计算机应用,2013,33(1):199-201,206
ISSN 1001-9081
CODEN JYIIDU
2013-01-01
http://www.joca.cn

文章编号:1001-9081(2013)01-0199-03
doi:10.3724/SP.J.1087.2013.00199

改进的快速探索随机树双足机器人路径规划算法

莫栋成*,刘国栋
(江南大学 物联网工程学院,江苏 无锡 214122)
(*通信作者电子邮箱 412055606@qq.com)

摘 要: 针对快速探索随机树(RRT)算法进行路径规划时随机性大且未考虑移动代价的问题,提出了任意时间快速探索随机树算法。生成一组快速探索随机树,之后每个树都重新使用上个树的信息来不断改进树的延伸。为进一步优化算法,使用节点缓存来生成一个引力函数来减少移动代价。最终的算法能够快速地生成初始路径,在规划时间内不断地改进路径且通过使用阈值来确保后面路径都比上次的移动代价更小。双足机器人仿真实验中,改进后的算法与初始的算法相比,搜索的节点数由883减少到704,效率提高了近25%。实验结果表明了改进算法的有效性。

关键词: 快速搜索随机树;路径规划;双足机器人;任意时间算法;移动代价
中图分类号: TP24;TP18 **文献标志码:** A

Improved path planning algorithm of rapidly-exploring random tree for biped robot

MO Dongcheng*, LIU Guodong
(*School of Internet of Things Engineering, Jiangnan University, Wuxi Jiangsu* 214122, *China*)

Abstract: To solve the problems that the Rapidly-exploring Random Tree (RRT) path planning is unstable and not taking cost into consideration, an anytime RRT Algorithm was proposed. The algorithm produced an initial solution very quickly, and then improved its growth by reusing information from the previous trees. Besides, to improve the algorithm, a biased distribution was produced to save the cost by using a waypoint cache. The resulted approach produced an initial solution very quickly, and then improved the quality of this solution within given time. It was guaranteed that subsequent solution would be less costly than all previous ones by using the bound. In the biped robot's simulation experiment, compared to the initial algorithm, the number of search nodes created by the improved algorithm decreases from 883 to 704 and the efficiency increases approximately 25%. The simulation result demonstrates the effectiveness of the improved algorithm.

Key words: Rapidly-exploring Random Tree (RRT); path planning; biped robot; anytime algorithm; cost

0 引言

路径规划[1-3]是指在具有障碍物的环境中,按照一定的评价标准,寻找一条从起始状态到目标状态的无碰撞路径。Jean[4]和LaValle[5]指出规划问题的复杂度与机器人的自由度成指数关系,与环境中的障碍物的规模成多项式关系。近年来快速探索随机树(Rapidly-exploring Random Tree,RRT)[6]算法作为一种快速的搜索方法在路径规划领域得到了广泛的研究。快速探索随机树算法由于采用随机采样的规划方法,不需要预处理,搜索速度快,其在高维空间中速度优势尤为明显,因此该算法得到了很多研究者的青睐。早期主要采用单棵RRT树进行搜索,为了进一步提高搜索速度,LaValle等提出了偏向RRT、双向RRT搜索算法[7]。由于利用随机采样产生扩展随机树,导致机器人在完成同一任务规划时会产生不同的路径,这样影响RRT算法的稳定性。针对这一问题,ERRT(Extend RRT)算法、DRRT(Dynamic RRT)算法和MP-RRT(Multipartite RRT)算法相继被提出,都通过重复使用上周期随机树的信息对RRT算法做了改进,一定程度上提高了动态环境下RRT算法的稳定性。虽然上述算法一定程度上改进了RRT算法,但固有规划方式限制了其进一步应用:1)在全局空间均匀地随机搜索,导致算法无谓的移动代价较大;2)先全局搜索构建随机树,再规划路径,导致算法通常只能应用在已知环境中,实时应用性较差;3)路径的搜索树由随机采样点生成,导致规划出的路径经常不是最优路径。

为了改进这些问题,本文提出了一种改进的以样本为基础的任意时间快速探索随机树算法(anytime RRTs Algorithm)[8-9]。该算法将任意时间算法和RRT算法结合起来且在此基础上又加入启发引力函数(waypoint cache)来引导搜索。通过实验验证了该算法的有效性。

1 快速探索随机树算法

令Tk为一个有K个节点的RRT,Q为Tk的节点,$Cfree$为与障碍物无碰撞的机器人自由状态空间。定义$Qinit$为初始状态即起点,$Qgoal$为目标状态即终点。令$Qrand$为空间中一个随机选取的位姿状态,然后找出Tk中距离$Qrand$最近的节点$Qnear$。设$p,q \in Cfree$,令$D(Qnear,Qrand) \leqslant D(Q,Qrand)$。在$Qrand$与$Qnear$连线上求$Qnew$,$Qnew$必须满足$Qnew \in Cfree$且$D(Qnear,Qrand) = \rho$的条件,其中$\rho > 0$,为RRT生长的最小单位长度,称为步长。如果存在$Qnew$,则$Tk$增加一个新节点;否则重新选取$Qrand$,重复以上过程。初始的RRT算法的构建过程如图1所示。

收稿日期:2012-07-12;修回日期:2012-08-13。

作者简介:莫栋成(1990-),男,江苏盐城人,硕士研究生,主要研究方向:机器人、人工智能;刘国栋(1950-),男,辽宁沈阳人,教授,博士生导师,主要研究方向:智能控制、机器人。

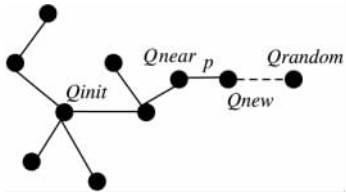

图1 RRT算法的构建

2 改进的任意时间快速探索随机树算法

任意时间(anytime)算法是一种解的质量随着时间的增加而逐步提高的算法。从本质上来讲,任意时间算法是一种反复求精算法,它可以很快地生成一个不精确解,然后经过若干次重复过程逐步提高解的质量。由于它可以在任意时刻中断并能输出一定质量的解,所以称之为任意时间算法。。

可以用这个基本思想来构建一个改进的任意时间RRT算法。最开始不考虑移动代价,快速地生成一个初始的RRT路径,然后记录下这次的移动代价计为Cs。通过限制节点使下次生成的RRT路径的移动代价比上次小,只将那些对规划路径可能有贡献的节点加到树上。也可以用$(1-\varepsilon)$来乘Cs($0 \leqslant \varepsilon < 1$),这样就能确保下一次的路径规划的移动代价至少要比这次的要小ε倍。其中Cs不断被新的规划路径的移动代价更新,重复这个过程直到规划时间结束。改进任意时间RRT算法(ExtendToRandom和Main函数)的主要伪代码如下所示:

```
SelCost(rrt T, configuration q, configuration random)
  return T.db* Distance(q,q_random) + T.cb* T.c(q_start,q);
ExtendToRandom(rrt T, configuration q_random)
  Qnear = kNearestNeighbors(q_random,k,T);
  while Qnear is not empty
  remove q_tree with minimum SelCost(T,q_tree) from Qnear;
  Qext = GenerateExtensions(q_tree,q_random);
  q_new = arg min_{q∈Qext} c(q_tree,q);
  T.c(q_start,q_new) = T.c(q_start,q_tree) + c(q_tree,q_new);
  if (T.c(q_start,q_new) + h(q_new,q_goal) ≤ T.Cs)
    return q_new;
  return null;
Main()
  T.db = 1; T.cb = 0; T.Cs = ∞;
  forever
  ReinitializeRRT(T);
  T.Cn = GrowRRT(T);
  if (T.Cn!=null)
    PostCurrentSolution(T);
    T.Cs = (1-ε)* T.Cn;
    T.db = T.db - &_d;
  if (T.db<0)
    T.db=0;
    T.cb = T.cb + &_c;
  if (T.cb>1)
    T.cb=1;
```

2.1 节点的采样

将导航节点缓存(waypoint cache)作为优化器。它是个大小一致的状态矩阵,当找到一个更好的规划时,规划的所有状态会任意地放入优化器中,这样可以知道之后在哪里会再次发现更好路径。

假如只关心生成路径的移动代价比上限值Cs小,那么就可以使用这个上限值来影响RRT算法的采样过程。只对那些可能会提供满足上限值的组态空间采样,而不需要对整个组态空间随机采样。给出一个节点$Qrandom$,可以通过计算启发成本来判断$Qnew$能不能成为规划的一部分。

比较$h(Qstart,Qrandom)+h(Qrandom,Qgoal)$和$Cs$的大小。如果

$$h(Qstart,Qrandom)+h(Qrandom,Qgoal) < Cs \quad (1)$$

则考虑$Qrandom$节点;反之则不考虑$Qrandom$节点。

2.2 节点的选择

基本的RRT算法是选择在树中离$Qrandom$最近的节点。但是,如Urmson等[10]说的,如果在选择过程中将移动代价考虑进去,规划会更好。所选择的规划方法就是以他们的思想为基础的,但是是使用偏差因素来影响代价的。最开始节点的选择是按距离的,接下来的选择则是将距离和移动代价结合起来考虑的,最终的选择是完全地考虑移动代价。使用距离参数d_b和代价参数c_b,c_b开始时计算在树中离$Qrandom$最近的k个邻近节点,然后这些节点根据选择这些节点做规划所需要花费代价的大小排序:

$$SelCost(q) = db*\ Distance(q,Qrandom) + cb*\ c(Qstart,q) \quad (2)$$

其中$c(Qstart,q)$是$Qstart$到q的当前路径的移动代价。图2说明了节点选择的过程(黑色区域代表障碍物,灰色区域代表路径经过时代价更大)。白点和两个黑点都是最近的节点,但却最先选择白点,因为经过它的路径规划所花费的代价比其他两个黑点要小。

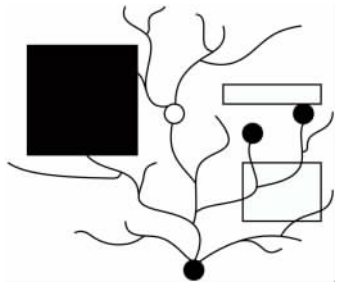

图2 节点的选择过程

2.3 节点的延伸

当选择树中一个节点作延伸时,要考虑它附近的组态空间的特性来产生一个新的,移动代价低的树的延伸分支。以样本节点$Qrandom$为大致方向,从节点$Qtree$生成一些可能的延伸分支,然后在这些分支里挑出移动代价最小。这样就能得到新的节点$Qnew$,假如它能满足条件(3):

$$c(Qstart,Qtree)+c(Qtree,Qnew)+h(Qnew,Qgoal) \leqslant Cs \quad (3)$$

那么就将$Qnew$增加到树上。其中:$c(Qstart,Qtree)$是树中$Qstart$到$Qtree$当前路径所花费的代价,$c(Qtree,Qnew)$是$Qtree$到$Qnew$构建分支的代价。

2.4 双足机器人的模型和姿态

双足机器人的模型[11-12]是基于机器人的物理结构构造的,为了简化双足机器人的模型,便于仿真和计算,本文在Matlab里只构建了双足机器人的躯干和下肢,如图3所示。其中:ZMP(Zero Moment Point)表示机器人的零力矩点,即作用在机器人足底的合力所通过的作用点;CoM(Center of Mass)表示机器人的质心。描述机器人的双足步行能力就是相对于站立脚定位抬起脚。由于双足机器人步行时是交替地移动双腿,那么规划的过程也可以选择两种姿态:左脚和右脚。为了简化计算,采用了一组离线轨迹运动[13]。假设机器人抬起脚会放在一些固定的区域。图4显示了当站在左脚上时,右脚的一组离散姿态。

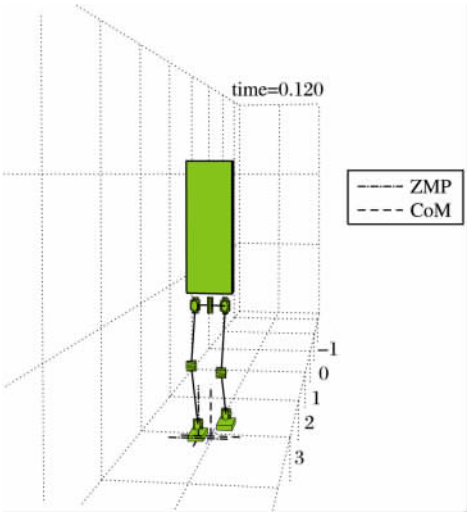

图 3　双足机器人的简单模型

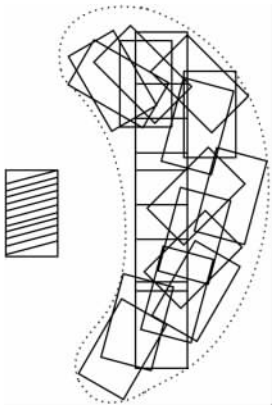

图 4　左脚的离线稳定姿态位置

3　仿真结果与分析

为了比较这两个算法分别运行了基本的 RRT 算法和改进的任意时间 RRT 算法。每个算法都运行了 100 次，最后平均结果。对于基本的 RRT 算法的节点选择设置 $p(\text{goal}) = 0.6$，$p(\text{random}) = 0.4$。对于改进的任意时间 RRT 算法，设置 $p(\text{goal}) = 0.6$，$p(\text{random}) = 0.2$，$p(\text{Waypoint Cache}) = 0.2$。改进的任意时间 RRT 算法刚开始都是快速地生成一个标准路径，然后改进的任意时间 RRT 算法在每次迭代后通过 $\delta_d = 0.1$ 来减少 d_b 和通过 $\delta_d = 0.1$ 来增加 $c_b \cdot \varepsilon$ 设为 0.05，这样就能确保每次成功的规划路径所花费的代价都比上个路径所花费的代价要小 5%。

为了更直观地比较这两个算法，在一个 400 × 600 的环境里障碍物随机摆放(黑色代表障碍物)且在这个环境里，路径经过的移动代价是不一致的(灰色区域代表移动代价更高)。每个方法所花费的时间为 2 s，单个 RRT 可以长达 0.5 s。

图 5 ~ 6 分别显示了基本的 RRT 算法和改进的任意时间 RRT 算法(黑色部分代表障碍物，灰色部分代表移动代价更高)。可以看出: 图 5 中的基本 RRT 算法没有考虑路径规划的移动代价问题; 图 6 显示了改进的任意时间 RRT 算法最大化地减少了规划的移动代价; 图 7 更加明显地显示了两个算法路径规划时移动代价的比较。可以看出基本的 RRT 算法规划路径时移动代价的不稳定性; 相反，改进的任意时间 RRT 算法的移动代价逐渐减少，最后达到最小值。表 1 显示了这两个算法的性能，可以看出改进的任意时间 RRT 算法比基本的 RRT 算法的搜索节点少了 179 个，搜索效率大大提高。

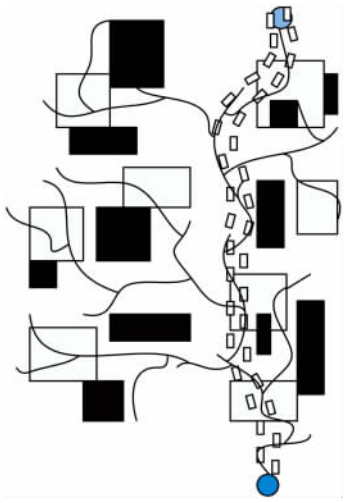

图 5　基本的 RRT 算法的路径规划

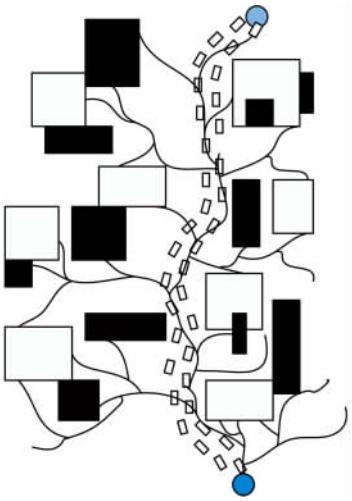

图 6　改进的任意时间 RRT 算法的路径规划

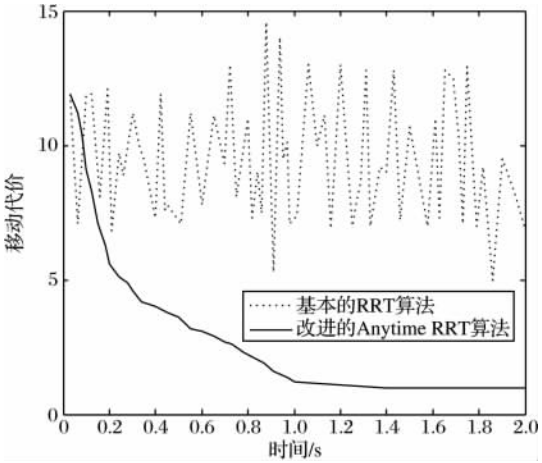

图 7　两种算法路径规划的移动代价比较

表 1　两个规划算法性能的对比

算法	时间/s	算法运行次数	平均值
基本的 RRT 算法	30	100	883
改进的任意时间 RRT 算法	30	100	704

4　结语

本文提出了双足机器人路径规划的一种改进的任意时间 RRT 算法。该算法通过每个树都重新使用上个树的信息来不断改进树的延伸和路径规划的性能，且使用了导航节点

(下转第 206 页)

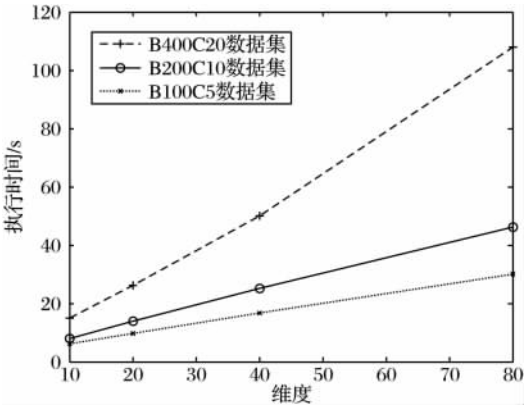

图 3 在维度上的可扩展性(流速为每秒 100 个数据点)

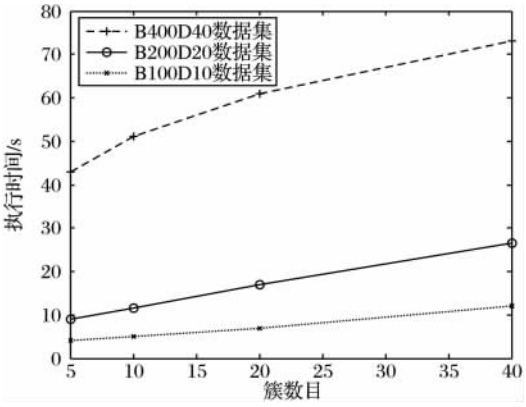

图 4 在簇数目上的可扩展性(流速为每秒 100 个数据点)

4 结语

虽然 CluStream 算法所提出的框架对数据流的聚类具有很大的意义,但该算法聚类效果不佳,在本文所提的算法中,用数据点进行物元化后的关联函数值来代替传统的衡量数据点归属的距离指标,很好地解决了 CluStream 算法中存在的问题。但是,由于要存储各簇的经典域和节域,当自然簇数目较多时,该算法的空间复杂度会随着自然簇的增多而呈线性增长,因此,如何对自然簇数目较多的数据流进行有效聚类,将是下一步要解决的问题。

参考文献:

[1] GUHA S, MISHRA N, MOTWANI R, *et al*. Clustering data streams [C]// Proceedings of the 41st Annual Symposium on Foundations of Computer Science. Washington, DC: IEEE Computer Society, 2000: 359 – 366.

[2] O'CALLAGHAN L, MISHRA N, MEYERSON A, *et al*. Streaming-data algorithms for high-quality clustering [C]// Proceedings of the 18th International Conference on Data Engineering. Washington, DC: IEEE Computer Society, 2002: 685 – 694.

[3] AGGARWAL C C, HAN J, WANG J, *et al*. A framework for clustering evolving data streams [C]// Proceedings of the 29th International Conference on Very Large Data Bases. Berlin: VLDB Endowment, 2003: 81 – 92.

[4] AGGARWAL C C, HAN J, WANG J, *et al*. A framework for projected clustering of high dimensional data streams [C]// Proceedings of the 30th International Conference on Very Large Data Bases. Toronto: VLDB Endowment, 2004: 852 – 863.

[5] CAO F, ESTER M, QIAN W N, *et al*. Density-based clustering over an evolving data stream with noise [C]// Proceedings of the 6th SIAM International Conference on Data Mining. Philadelphia: SIAM Press, 2006: 328 – 339.

[6] 朱蔚恒,印鉴,谢益煌. 基于数据流的任意形状聚类算法[J]. 软件学报,2006,17(3):379 – 387.

[7] 张晨,金澈清,周傲英. 一种不确定数据流聚类算法[J]. 软件学报,2010,21(9):2173 – 2182.

[8] 王述云,胡运发,范颖捷,等. 基于距离与熵的混合属性数据流聚类算法[J]. 小型微型计算机系统,2010,31(12):2365 – 2371.

[9] 蔡文,杨春燕,陈文伟,等. 可拓集与可拓数据挖掘[M]. 北京:科学出版社,2008:39 – 43.

[10] LI Q X. The interval elementary dependent function based on interval side-distance [C]// Proceedings of IEEE 2008 ISECS International Colloquium on Computing, Communication, Control, and Management. Washington, DC: IEEE Computer Society, 2008: 674 – 678.

[11] 李桥兴,刘思峰. 基于区间距和区间侧距的初等关联函数构造[J]. 哈尔滨工业大学学报,2006,38(7):1097 – 1100.

[12] 李桥兴. 一元多维位值公式及一元多维初等关联函数构造方法[J]. 兰州大学学报:自然科学版,2010,46(2):86 – 90.

[13] 杨春燕,蔡文. 基于可拓集的可拓分类知识获取研究[J]. 数学的实践与认识,2008,38(16):184 – 191.

[14] 蔡文,杨春燕. 可拓学的应用研究、普及与推广(综述)[J]. 数学的实践与认识,2010,40(7):214 – 220.

(上接第 201 页)

缓存来生成一个更普遍的引力函数来减少移动代价。仿真实验表明该算法的有效性。但该算法的启发引力函数还没达到最优,以后的工作还需对其改进,以使启发引力函数能更高效地指引树的生长。该算法只是在理论上得到了实现,还未能应用在实际的双足机器人上,这也将是以后努力的方向。

参考文献:

[1] 夏泽洋,陈恳,熊璟,等. 仿人机器人运动规划研究进展[J]. 高技术通讯,2007,17(10):1092 – 1099.

[2] FU C L, CHEN K. Gait synthesis and sensory control of stair climbing for a humanoid robot [J]. IEEE Transactions on Industrial Electronics, 2008, 55(5): 2111 – 2120.

[3] 石为人,黄兴华,周伟. 基于改进人工势场法的移动机器人路径规划[J]. 计算机应用,2010,30(8):2021 – 2023.

[4] JEAN F. Complexity of nonholonomic motion planning [J]. International Journal of Control, 2001, 74(8): 776 – 782.

[5] LaVALLE S M. Motion planning: the essentials [J]. IEEE Robotics and Automation Society Magazine, 2011, 18(1): 79 – 89.

[6] 康亮,赵春霞,郭剑辉. 未知环境下改进的基于 RRT 算法的移动机器人路径规划[J]. 模式识别与人工智能,2009,22(3):337 – 343.

[7] LAVALLE S M, KUFFNER Jr, J J. Rapidly-exploring random trees: progress and prospects [C]// Proceedings of the Fourth Workshop on the Algorithmic Foundations of Robotics. Natick, MA: A K Peters, 2000: 45 – 59.

[8] 张振荣,刘惊雷,张伟. 一种生成最优联盟结构的任意时间算法[J]. 计算机工程,2011,37(2):185 – 187.

[9] 郭亚军,鲁汉榕. 任意时间算法的性能描述[J]. 武汉交通科技大学学报:交通科学与工程版,2000,24(5):562 – 565.

[10] URMSON C, SIMMONS R. Approaches for heuristically biasing RRT growth [C]// IROS 2003: Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway: IEEE Press, 2003: 1178 – 1183.

[11] 夏泽洋,陈恳. 仿人机器人足迹规划建模及算法实现[J]. 机器人,2008,30(3):231 – 237.

[12] 胡金东,刘国栋. 双足机器人步行模式的在线全身修正[J]. 计算机应用,2011,31(1):286 – 288,292.

[13] 梶田秀司. 仿人机器人[M]. 管贻生,译. 北京:清华大学出版社,2007.