

改进的 RRT-Connect 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 modongcheng88@163.com)

摘要:针对当组态空间内存在大量的窄道时,快速搜索随机树算法(RRT)难以取得连通路径的问题,提出了一种改进的 RRT-Connect 算法。该算法利用改进的桥梁检测算法来识别和采样窄道,使得路径规划在窄道内能轻易取得连通性;同时将 RRT-Connect 算法与任意时间算法相结合,显著地减少了 RRT-Connect 算法的移动代价。每个算法分别运行 100 次,与 RRT-Connect 算法相比,改进后的算法成功次数由 34 提高到 93,规划时间由 9.3 s 减少到 4.2 s。双足机器人的仿真实验结果表明,该算法能在窄道内取得优化路径,同时可以有效地提高路径规划的效率。

关键词:快速搜索随机树;桥梁检测;任意时间算法;路径规划;窄道;双足机器人

中图分类号: TP242 **文献标志码:** A

Improved RRT-Connect path planning algorithm for biped robot

MO Dongcheng*, LIU Guodong

(School of IoT Engineering, Jiangnan University, Wuxi Jiangsu 214122, China)

Abstract: When lots of narrow passages are contained in the configuration space, the algorithm of Rapidly-exploring Random Tree (RRT) can hardly get the path. In order to deal with the problem, an improved RRT-Connect algorithm was presented. An improved bridge test algorithm was employed to identify and sample narrow passages, so it could be easy to get the connectivity. Combining RRT-Connect with anytime algorithm, the cost of RRT-Connect could be reduced obviously. Each algorithm was run 100 times. Compared with RRT-Connect, the successes' number of the improved algorithm was increased from 34 to 93, and the path planning time was decreased from 9.3 s to 4.2 s. The biped robot simulation results demonstrate that the algorithm can get the optimal path inside the narrow passages; meanwhile, it improves the efficiency.

Key words: Rapidly-exploring Random Tree (RRT); bridge test; anytime algorithm; path planning; narrow passage; biped robot

0 引言

在过去的十年里,路径规划^[1-3]在机器人学、制造业、计算机动画等领域得到了广泛的研究。由于传统的确定性路径规划算法在高维组态空间里性能会急剧下降,使得以随机采样为基础的算法不断得到应用和改进。快速搜索随机树(Rapidly-exploring Random Tree, RRT)采用随机采样的规划方法,不需要预处理且搜索速度快(其高维空间中速度优势尤为明显),因此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,出现了许多改进的 RRT 算法,如:偏向 RRT、双向 RRT 和 RRT-Connect^[4-5],ERRT(Extend RRT)算法^[6],DRRT(Dynamic RRT)算法^[7],MP-RRT(Multipartite RRT)算法^[8]等。改进后的算法在路径规划方面的性能得到了显著的提升,其中 RRT-Connect 算法性能尤为突出。该方法基于两个基本思想:1)连接采样点的延伸函数试图移动更长的距离;2)RRT 从初始组态和目标组态同时延伸。但当组态空间内存在大量窄道,且路径规划必须通过窄道时,其性能会显著下降。由于 RRT-Connect 算法采用的是均匀的随机采样策略,其显著缺点是在识别到狭窄的通道前浪费了大量的计算时间在开阔的地带,且相对于整个组态空间而言窄道的通道面积很小,即使是两个很近的节点但由于属于不同的部分,RRT-Connect 算法也很难将它们连接取得连通性。

为解决上述问题,提出了一种改进的 RRT-Connect 算法。将 RRT-Connect 算法与桥梁检测(Bridge Test)^[9]算法相结合。Bridge Test 算法易于推广到高维空间,不需要复杂的几何处理就能很容易地提高窄道内的采样密度,将其与 RRT-Connect 算法结合可以解决 RRT-Connect 算法在窄道内采样难的问题。同时任意时间(anytime)算法^[10-11]是一种解的质量随着时间的增加而逐步提高的算法,在规定的时间内可不断减少 RRT-Connect 的移动代价。实验的结果表明算法的有效性。

1 双足机器人模型与步行

1.1 双足机器人的模型

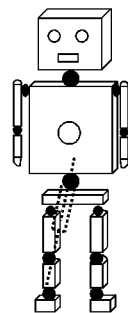


图1 机器人三维线性倒立摆

将三维空间的机器人^[12-13]近似为一个由集中了所有质量的点和连接该质点与支撑点的无质量的腿组成的倒立摆,如图1所示。在伸缩力 f 的作用下腿可以伸缩,可分解为 x, y, z 三个方向上的分量:

$$\begin{cases} f_x = (x/r)f \\ f_y = (y/r)f \\ f_z = (z/r)f \end{cases} \quad (1)$$

收稿日期:2013-03-07;修回日期:2013-04-01。

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:机器人的路径规划、智能控制;刘国栋(1950-),男,辽宁沈阳人,教授,博士,主要研究方向:机器人系统、智能控制。

其中 r 为支撑点和质心之间的距离。其质心在水平面 $k_x x + k_y y + z_c$ 上的运动方程为:

$$\begin{cases} \ddot{x} = \frac{g}{z_c} x \\ \ddot{y} = \frac{g}{z_c} y \end{cases} \quad (2)$$

由于约束面的斜率参数 k_x, k_y 没有包含在上面两式中,约束面的倾斜不影响质心的水平运动。这样的摆动称为三维线性倒立摆^[14-15]。

1.2 双足机器人的步行

利用步行单元的对称性,沿 x 轴的初始条件为 $(-\bar{x}, \bar{V}_x)$, 终止位置为 \bar{x} 。根据线性倒立摆的解析式有

$$\bar{x} = -\bar{x}C + T_c \bar{V}_x S \quad (3)$$

其中: $T_c = \sqrt{\frac{Z_c}{g}}$, $C \equiv \cosh \frac{T_{\text{sup}}}{T_c}$, $S \equiv \sinh \frac{T_{\text{sup}}}{T_c}$ 。

从式(3)中求得终止速度为

$$\bar{V}_x = \bar{x}(C+1)/(T_c S) \quad (4)$$

类似地,对步行单元的 y 轴分量,初始条件为 $(\bar{y}, -\bar{v}_y)$, 终止位置为 \bar{y} 。终止的位置和速度为:

$$\bar{y} = \bar{y}C + T_c (-\bar{v}_y) S \quad (5)$$

$$\bar{V}_y = \bar{y}(C-1)/(T_c S) \quad (6)$$

利用步行单元可以很容易生成步行轨迹,为了改变行走方向,需要在步行参数中追加有关方向的信息。每步踏足的方向用 S_r 表示,第 n 步的落脚点 $(p_x^{(n)}, p_y^{(n)})$ 由式(7)确定:

$$\begin{bmatrix} p_x^{(n)} \\ p_y^{(n)} \end{bmatrix} = \begin{bmatrix} p_x^{(n-1)} \\ p_y^{(n-1)} \end{bmatrix} + \begin{bmatrix} \cos s_r^{(n)} & -\sin s_r^{(n)} \\ \sin s_r^{(n)} & \cos s_r^{(n)} \end{bmatrix} \begin{bmatrix} s_x^{(n)} \\ s_y^{(n)} \end{bmatrix} - (-1)^n s_r^{(n)} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (7)$$

第 n 步的步行单元给定如下:

$$\begin{bmatrix} \bar{x}^{(n)} \\ \bar{y}^{(n)} \end{bmatrix} = \begin{bmatrix} \cos s_r^{(n+1)} & -\sin s_r^{(n+1)} \\ \sin s_r^{(n+1)} & \cos s_r^{(n+1)} \end{bmatrix} \begin{bmatrix} s_x^{(n+1)}/2 \\ s_y^{(n+1)}/2 \end{bmatrix} \quad (8)$$

而步行单元的速度按下式计算:

$$\begin{bmatrix} \bar{v}_x^{(n)} \\ \bar{v}_y^{(n)} \end{bmatrix} = \begin{bmatrix} \cos s_r^{(n+1)} & -\sin s_r^{(n+1)} \\ \sin s_r^{(n+1)} & \cos s_r^{(n+1)} \end{bmatrix} \cdot \begin{bmatrix} (1+C)/(T_c S) \bar{x}^{(n)} \\ (C-1)/(T_c S) \bar{y}^{(n)} \end{bmatrix} \quad (9)$$

这样就可以生成沿着任意方向行走的模式。

2 改进的桥梁检测算法

为清楚起见,首先给出一些定义和术语。对于有着 n 个自由度的机器人, n 维拓扑空间描述了机器人所有的位置和姿态叫作组态空间,用 C 表示;所有未与障碍物发生碰撞的组态叫作自由组态,用 F 表示;障碍物空间 B 定义为 F 的补集: $B = C/F$ 。因此,路径规划的问题可以如下描述:给出一对初始目标组态: q_{init} 和 q_{goal} , 找到一条连续无碰撞路径 $\tau: [0, 1] \rightarrow F$, 这样 $\tau(0) = q_{\text{init}}$, $\tau(1) = q_{\text{goal}}$ 。

窄道在 n 维组态空间内至少有一个限制方向,机器人沿着这个方向步行只要有一个小的扰动就有可能与障碍物发生碰撞。在窄道中利用桥梁检测算法可增加采样密度来获得窄道的连通性。桥梁检测算法只需检测三个采样组态: 短线 s 的两个端点和一个中点。短线 s 的两个端点只要在 C 的障碍物里且中点不在障碍物里,则这个中点通过桥梁检测可被选为路径节点。由于短线 s 类似一座桥跨越障碍物,所以就将短线 s 叫作“桥”。

初始的桥梁检测算法在 C 中随机地采样第一个桥端点 q_i ,

然后根据一个特殊的概率密度函数 λ_q 随机采样邻近 q_i 的第二个桥端点 q_s , 其中概率密度函数 λ_q 是个离散对称的高斯分布。由于需要对每个自由度都要设计一个不同的高斯方差,且高斯方差依据窄道的宽度得到,这就使得计算难度很大。考虑到桥的两端点 q_i 和 q_s 间的距离偏差很远,可利用该特性来改进该算法获得第二个桥端点 q_s 。改进的算法步骤如下:

步骤1 k 取值为 1 到 K , 分别采样点得到 $q_i (i = 1:K)$;

步骤2 选取 q_i 在障碍物中的点 q_i , 即为第一个桥端点;

步骤3 $d_q = q_c/l, q_s = q_i + (-1)^n d_q$;

步骤4 当检测到 q_s 在障碍物中时, q_s 即为第二个端点;

步骤5 $q_m = (q_i + q_s)/2$, 若 q_m 是无碰撞点, 即为所需的采样点。

改进的地方主要体现在步骤3, 其中 q_c 是任意的采样点, l 是窄道的最大宽度。这样 d_q 就是个很小的补偿。 n 是任意的整数, 这就意味着 q_s 离 q_i 很近, 当 n 为奇数时就在其前面, n 为偶数时则在后面。

2.1 窄道的采样点分布概率

设窄道采样点分布的概率密度为 X_a , 桥的端点 x 和 x' 为两随机变量。第一个端点 x 均匀分布在障碍物 $B = C/F$ 中, 所以如果 x 在 B 中, 则第一个端点的密度 $f_x(x)$ 是非零的。可假定 B 的体积为 1, 则当 x 在 B 中时, $f_x(x) = 1$, 否则为 $f_x(x) = 0$ 。根据前面提到的概率密度函数 λ_q 来选定第二个端点 x' 。设第二个端点的密度函数为 $I(q)$, 则由第一个端点得到第二个端点的条件密度函数为

$$f_{x'/x}(x'/x) = \lambda_x(x') I(x') / Z_x \quad (10)$$

其中 $Z_x = \int_C \lambda_x(x') I(x') dx'$ 是个归一化常数。在 x 条件下的采样点 p 的概率分布为

$$X_a(p) = \int_C f_{x'/x}(x'/x) f_x(x) dx \quad (11)$$

其中 p 是“桥”的中点, 则 $x' = 2p - x$ 。将 x' , $f_{x'/x}(x'/x)$, $f_x(x)$ 的表达式代入式(11)中, 则可得到采样点的分布概率为

$$X_a(p) = \int_B \lambda_x(2p-x) I(2p-x) / Z_x dx \quad (12)$$

上述提到 λ_q 是个离散对称的高斯分布, 其中心在 x , 当 $x' = 2p - x$ 靠近 x 时, λ_q 的值会很大。而且只有当 $I(2p-x) = 1$ 时, 式(11)的积为非零。对于窄道中的采样点 p 越满足上述的条件, 其采样的概率值越大。

如图2所示(其中左边为窄道, 右边为开阔地带), 很明显的在窄道里取得的采样点通过桥梁检测要容易得多。而在开阔地带, 桥梁检测算法需要花费更多的计算时间来使得采样点通过检测。如果开阔地带也采用桥梁检测算法采样的话, 整个路径规划的效率会大大降低, 所以在开阔地带需结合 RRT-Connect 算法来采样。

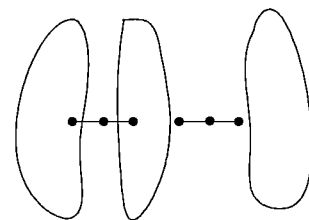


图2 桥梁检测的描述

3 改进的 RRT-CONNECT 算法

RRT-Connect 算法基于两个基本思想: 启发函数试图移动更长的距离, RRT 从 q_{init} 和 q_{goal} 两端同时生长。Connect 启

发函数是个贪婪函数,它反复迭代 EXTEND 直到碰到障碍物或到达目标区域。这个操作类似人工势场函数在随机势场内的作用,将快速搜索和 RRT 均匀探索的特性结合起来且避免局部极小的缺陷。 T_a 和 T_b 代表两棵树,它们一直保持生长,直到相遇并找到解决路径。每次迭代, T_a 都指向随机组态 q_{rand} , 产生新的节点 q_{new} 。选择 T_b 中靠近 q_{new} 的节点,将其 q_{new} 与相连。若能到达 q_{new} ,则 T_a 和 T_b 认为是能连接的;若不能相连,在下次迭代中交换两树的作用,这样每棵树都有平等的机会延伸。因此,RRT-Connect 算法很好地维持了探索和连接的平衡。

任意时间算法是一种反复求精算法,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改进 RRT-Connect 算法。最开始不考虑移动代价,快速地生成一个初始路径,记录下这次的移动代价计为 C_s 。通过限制节点使下次生成的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 C_s ($0 \leq \varepsilon < 1$),这样就能确保下一次路径规划的移动代价比本次少 εC_s 。其中 C_s 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。下面列出了改进的 RRT-Connect 算法的 Main 函数的伪代码。改进的地方主要体现在第7)步,其中 $T.Cn$ 是初始的移动代价, $T.Cs$ 是改进后的移动代价。节点的选择主要依据式(13):

$$Cost(q) = T.db \cdot Dis(q, q_{target}) + T.cb \cdot c(q_{start}, q) \quad (13)$$

其中: $T.db$ 是距离偏置参数, $T.cb$ 是移动代价偏置参数。算法开始时设定 $T.db = 1$, $T.cb = 0$, 节点的选择则先由节点到目标点的距离决定。每次成功搜索后, $T.db$ 不断减少 0.1, 而 $T.cb$ 不断地增加 0.1。这样在规定时间内,算法在选择优化节点时考虑移动代价的比重就会不断增加,从而达到不断减少移动代价的效果。

Main()

- 1) $T.db = 1$; $T.cb = 0$; $T.Cs = \infty$;
- 2) forever
- 3) ReinitializeRRT (T);
- 4) $T.Cn = \text{GrowRRT} (T)$;
- 5) if ($T.Cn \neq \text{null}$)
- 6) PostCurrentSolution (T);
- 7) $T.Cs = (1 - \varepsilon) * T.Cn$;
- 8) $T.db = T.db - 0.1$;
- 9) if ($T.db < 0$)
- 10) $T.db = 0$;
- 11) $T.cb = T.cb + 0.1$;
- 12) if ($T.cb > 1$)
- 13) $T.cb = 1$;

4 两算法采样分布的结合

桥梁检测算法可以有效地提升在窄道 P 中的采样密度。密度 X_a 主要偏向 P ,但在开阔无碰撞的区域 $F \setminus P$ 却是不利的,采样点很少。一个好的路线图必须充分覆盖整个组态,为弥补 X_a 的不足,将其与均匀分布 X_U 结合。 X_U 大部分采样分布在 $F \setminus P$ 中,这两采样分布可以很好地互补。将这两采样分布相结合产生一个混合采样分布:

$$X_H = (1 - \omega)X_a + \omega \cdot X_U \quad (14)$$

其中: ω ($0 \leq \omega \leq 1$) 是个权值, ω 的选择主要依据在窄道中采样的困难程度和需要覆盖 F 的采样点的数量。目前,主要人为地尽量满足第一个条件即优先满足密度 X_a 。因为,窄道的采样在整个组态采样里是最困难的。混合采样的关键不是简单地将两分布结合,而是保留了两分布在各自擅长区域的采样能力。

5 实验结果与分析

5.1 计算机仿真实验

为验证算法的有效性,用 Matlab 分别在二维和三维环境里运行该算法。其中权值 ω 为满足密度 X_a , 比例设为 1:5, 即在 X_a 中每产生 5 个采样点就在 X_U 中产生一个采样点。

图3是算法在二维环境中的仿真结果。将移动区域限制在 1×3 的环境中,其中 X 轴是 -0.5 到 0.5 , Y 轴是 -1.5 到 1.5 ;起始点的坐标为 $(0.4, -1.3)$, 目标点的坐标为 $(-0.4, 1.3)$ 。图中黑色的长方形块代表障碍物,其中有 3 个窄道可以通过。黑色的曲线代表算法的最终规划路径。可以看出图中算法可以穿过窄道找到目标点,并规划出路径,证明了算法的有效性。

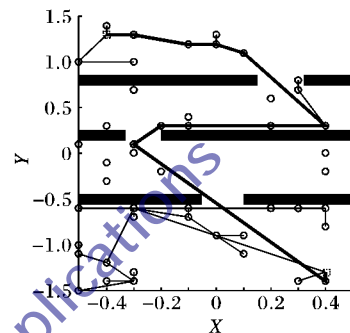


图3 算法在二维环境中的路径规划

图4是算法在三维环境中的仿真结果。移动区域限制在 $2 \times 1 \times 1$ 的空间内,其中 X 轴是 -1 到 1 , Y 轴是 -0.5 到 0.5 , Z 轴是 -0.5 到 0.5 。起始点坐标为 $(0, -0.7, 0)$, 目标点的坐标为 $(0, 0.7, 0)$ 。图中两块 1×1 灰色半透明正方形块为障碍物,每块障碍物上都有一个 0.2×0.2 的正方形窄道。其中较细的黑线(线宽为 2)代表初始的规划路径,最粗的黑线(线宽为 6)代表 RRT-Connect 算法经过任意时间算法的改进并结合桥梁检测算法后的规划路径,可以看出改进后算法的移动代价明显降低。

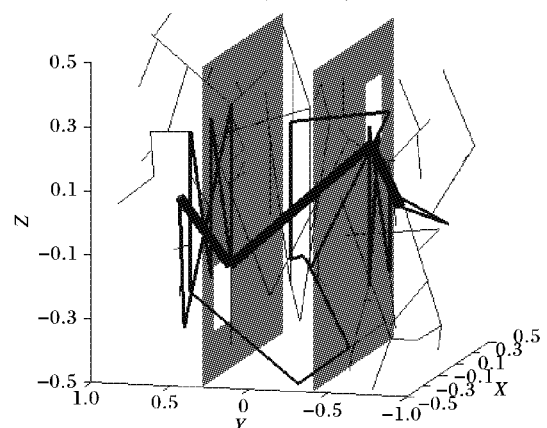


图4 算法在三维环境中的路径规划

表1显示了改进后的 RRT-Connect 算法与初始 RRT-Connect 算法在有窄道的三维环境内分别运行 100 次的性能对比。

表1 两算法的性能对比

算法	成功数	平均时间/s
RRT-Connect	34	9.3
改进的 RRT-Connect	93	4.2

分别抽取两算法成功数中的前20个,统计其中各个算法每次所花费的采样节点数。图5显示了比较结果,可以看出与RRT-Connect算法相比,改进后的算法由于结合任意时间算法,其节点数不断优化减少,最后接近50时达到最小值。

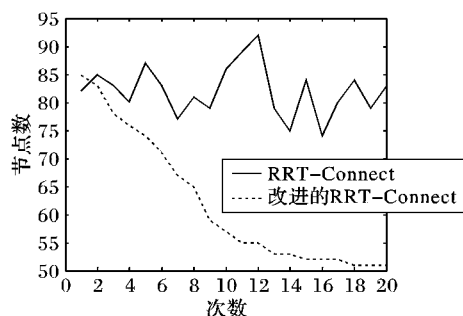


图5 两算法的搜索节点数比较

5.2 实际的双足机器人实验

如图6所示,设定初始的出发点和目标点,控制实验室双足机器人跟踪规划路径。实验结果表明,生成的路径可使双足机器人通过窄道,证明了算法的可行性。

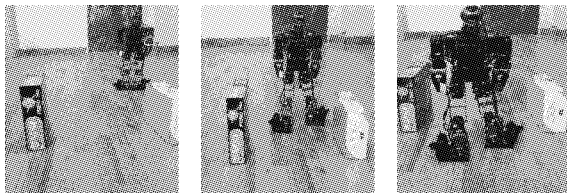


图6 双足机器人通过窄道

6 结语

本文提出了一种改进的RRT-Connect算法,将RRT-Connect算法和桥梁检测算法结合起来,解决了RRT-Connect算法在狭窄的通道内采样规划路径难的问题;并用任意时间算法进一步改进RRT-Connect算法,减少了路径规划的移动代价。但该算法还有一些问题未能得到很好的解决,如桥梁检测算法采样窄道时会产生一些不必要的探索树,这样就降低了算法的性能。这些问题将是下个阶段努力的方向。

参考文献:

- [1] 夏泽洋,陈恩,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] 郑慧杰,刘弘,郑向伟. 基于改进群搜索优化算法的群体路径规划方法[J]. 计算机应用, 2012, 32(8): 2223-2226.
- [3] 张彤,肖南峰. 仿人机器人实时路径规划方法研究[J]. 计算机工
- [4] LAVALLE S M, KUFFNER J J, Jr. Rapidly-exploring random trees: progress and prospects [C]// Proceedings of the 4th International Workshop on the Algorithmic Foundations of Robotics: Algorithmic and Computational Robotics: New Directions. Natick, MA, USA: A. K. Peters, 2000: 293-308.
- [5] LAVALLE S M, KUFFNER J. RRT-Connect: an efficient approach to single-query path planning [C]// Proceedings of the 2000 IEEE International Conference on Robotics & Automation. Piscataway: IEEE, 2000, 4: 995-1001.
- [6] BRUCE J, VELOSO M. Real-time randomized path planning for robot navigation [C]// Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway: IEEE, 2002: 2383-2388.
- [7] FERGUSON D. Replanning with RRTs [C]// Proceeding of the 2006 IEEE International Conference on Robotics & Automation. Piscataway: IEEE, 2006, 5: 1243-1248.
- [8] ZUCKER M, KUFFNER J J, Jr. Multipartite RRTs for rapid replanning in dynamic environments [C]// Proceeding of the 2007 IEEE International Conference on Robotics & Automation. Piscataway: IEEE, 2007, 4: 1603-1609.
- [9] ZHEN S, DVAID H, JIANG T T, et al. Narrow passage sampling for probabilistic roadmap planning[J]. IEEE Transactions on Robotics, 2005, 21(6): 1105-1115.
- [10] JEON J H, KARAMAN S, FRAZZOLI E. Anytime computation of time-optimal off-road vehicle maneuvers using the RRT* [C]// Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference. Piscataway: IEEE, 2011: 3276-3282.
- [11] FERGUSON D, STENTZ A. Anytime RRTs [C]// Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway: IEEE, 2006: 5369-5375.
- [12] 夏泽洋,陈恩. 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.
- [13] (日) 梶田秀司. 仿人机器人[M]. 管贻生,译. 北京: 清华大学出版社, 2007.
- [14] 于国晨,刘永信,李晓红. 基于三维线性倒立摆的仿人机器人步态规划[J]. 计算机应用, 2012, 32(9): 2643-2647.
- [15] 李龙澍,王唯翔,王凡. 基于三维线性倒立摆的双足机器人步态规划[J]. 计算机技术与发展, 2011, 21(6): 66-69.
- [16] 程与应用, 2009, 45(27): 4-6.
- [17] DABBLE F, SEBASTIAN F. Supervised term weighting for automated text categorization [C]// SAC '03: Proceedings of the 2003 ACM Symposium on Applied Computing. New York: ACM, 2003. 784-788.
- [18] ZHANG C, FAN X H, CHEN X N. Hot topic detection on Chinese short text [C]// Advanced Research on Computer Education, Simulation and Modeling: Communications in Computer and Information Science, CCIS 176. Berlin: Springer-Verlag, 2011: 207-212.
- [19] 王细薇,樊兴华,赵军. 一种基于特征扩展的中文短文本分类方法[J]. 计算机应用, 2009, 29(3): 843-845.
- [20] 徐易. 基于短文本的分类算法研究[D]. 上海: 上海交通大学, 2010.
- [21] 王细薇,张凯. 一种改进的基于共现关系的短文本特征扩展算法研究[J]. 河南城建学院学报, 2012, 21(4): 48-50.
- [22] 胡佳妮,郭军,徐蔚然. 一种基于短文本的独立语义特征抽取算法[C]// 2007年全国网络与信息安全技术研讨会论文集. [出版地不详]: 中国通信学会, 2007: 332-335.
- [23] LAN M, TAN C L, SU J, et al. Supervised and traditional term weighting method for automatic text categorization [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(4): 721-735.
- [24] QUAN X J, LIU W Y. Term weighting schemes for question categorization [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(5): 1009-1020.
- [25] 林智勇,郝志峰,杨晓伟. 不平衡数据分类的研究现状[J]. 计算机应用研究, 2008, 25(2): 332-336.
- [26] 崔争艳. 中文短文本分类的相关技术研究[D]. 河南: 河南大学, 2011.
- [27] YANG Y M, LIU X. A re-examination of text categorization methods [C]// SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 1999: 42-49.

(上接第2282页)