

运动规划的高效配置空间构建与优化

潘佳^{1*}, Dinesh Manocha²

摘要: 配置空间是算法机器人学领域使用较为广泛的一种基本概念。机器人学、计算机辅助设计与相关领域的诸多应用都可归类为配置空间方面的计算问题。本文将对近期与配置空间相关的两项重大挑战的解决方案成果进行探讨: ①如何高效计算高维配置空间的近似表达; ②如何在高维配置空间内高效执行几何学接近度与运动规划查询。基于机器学习与几何学近似技术, 笔者在此提出几种新型配置空间构建算法。上述算法对多个配置样本进行碰撞查询。得出的碰撞查询结果将用来计算配置空间的近似表达, 可快速聚合至准确的配置空间; 同时还提出了基于并行图形处理器 (GPU) 的算法, 以便加速配置空间优化与搜索计算的性能情况。笔者特别设计出了基于 GPU 的并行 k 最近邻算法与并行碰撞检测算法, 并使用这些算法来加快运动规划。

关键词: 配置空间, 运动规划, 图形处理器 (GPU) 并行算法

1 前言

智能机器人在工业领域与日常生活领域中起着越来越重要的作用。在工业领域, 由于劳力成本的不断上升, 致使制造企业在工厂中使用越来越多的机器人。例如, 相比较 2012 年, 中国的平均最低薪资水平已经增长了 20% 有余, 同时制造业的机器人的供应也增长了 51% [1]。欧美地区情况类似。人们设计使用智能机器人, 以提升生产能力, 并且使制造企业在价格与质量方面更具竞争力。具有代表性的最新成果之一便是名为 “Baxter” 的机器人 [2], 该机器人配备的软件可以让机器人自身从人为

示范动作中学习不同工作任务, 识别不同目标, 并对外部因素进行智能化反应。智能机器人还有望协助人们的日常生活。在未来, 智能机器人将有望完成不同工作任务, 包括: ①做家务、关怀支持, 如烹饪、洗衣等; ②健康生活支持, 如与老年人聊天、照顾残疾人士等; ③在危险工作条件下进行劳作, 如化工厂等 [3]。当前已研发出了几个成功的辅助机器人原型机。例如, Willow Garage 机器人制造公司推出的 PR2 机器人可以照顾残疾人士的生活, 如照顾四肢瘫痪的残疾人士 [4]; HRP-4 人形机器人可以模仿人类动作, 并使用语言与人们进行交流 [5]。除了在工业与日常生活领域的应用, 现代智能机器人还可用于其他一些领域, 包括无人驾驶车辆 [6]、外科手术治疗 [7]、紧急救援与灾害救援 [8] 以及完成军事任务等 [9]。

过去 10 年间, 智能机器人设计与可用性方面的巨大改善与提升基于诸多相关领域取得的突破与进展, 包括计算机视觉、人工智能、机器学习、控制、传感器系统、机械系统, 上述不同因素构成了智能机器人系统的不同部分 (图 1)。例如, 即时定位与地图构建 (SLAM) 算法可使机器人在未知环境下准确追踪自身位置 [10]。此外, 在先进视觉技术的协助下, 当前机器人可从背景点云中识别并区分目标 [11]。相比较传统的工业机器人而言, 现代智能机器人的一大重要特征便是其高级规划与导航。高级规划与导航的主要作用是根据需要执行任务的高级说明来计算低级指令; 然后将这些低级指令提供给机器人执行系统。高级规划与导航部分包括许多不同子部分 (图 1), 并且人们在这些领域进行了大量工作与研究,

¹ Department of Computer Science, The University of Hong Kong, Hong Kong, China; ² Department of Computer Science, University of N. Carolina, Chapel Hill, NC 27599-3175, USA

* Correspondence author. E-mail: jpan@cs.hku.hk

Received 12 March 2015; received in revised form 20 March 2015; accepted 25 March 2015

© The Author(s) 2015. Published by Engineering Sciences Press. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

引用本文: Jia Pan and Dinesh Manocha. Efficient Configuration Space Construction and Optimization for Motion Planning. *Engineering*, DOI10.15302/J-ENG-2015009

如任务规划 [12]、观测信息反馈 [13,14]、最优控制 [15] 和适应性控制 [16]。

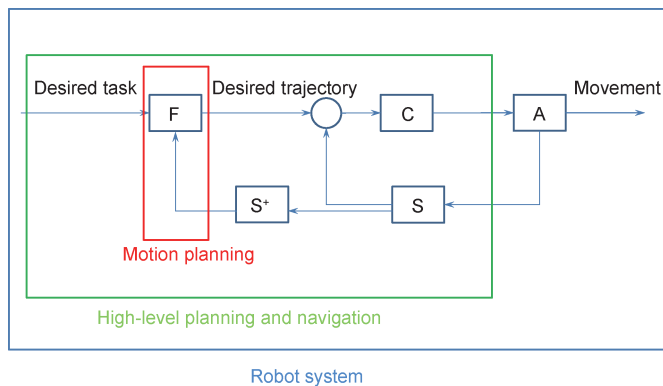


图 1. 机器人系统内的重要硬件与软件子系统。①前馈系统 (F)，包括任务规划、导航策略、运动规划与轨迹生成；②控制系统 (C)，包括运动学、动力学与控制算法；③执行系统 (A)，包括电动机、伺服系统、传动机构等；④传感器系统 (S)，包括各种传感器，如摄影机、激光、IMU，以及相关低级传感器数据处理算法，如信号处理、估算与融合。⑤传感器后处理系统 (S⁺)，包括定位、绘图等。机器人系统的主要软件部分为高级规划与导航，如果需要执行特定任务，可用来确定发送给执行系统的操作指令。高级规划与导航的一项重要部分便是运动规划，运动规划注重根据环境说明来计算轨迹。

高级规划与导航部分最为重要的子系统之一便是运动规划系统，该系统能够让机器人从初始位置安全移动至目标位置，并且不会碰撞到环境中的静止物体与移动物体。快速在线运动规划算法对许多应用而言至关重要。例如，在制造业领域，我们越来越需要移动机器人来协助人们进行重复性无附加值的任务操作 [17]。为了确保机器人与人类之间的协作安全，机器人需要具有实时运动规划功能，避免发生意外事故。对于许多传统制造业流程而言，在线运动规划在改善自动化水平方面也尤为重要。例如，对于自动化箱体抓取而言，高效的在线运动规划对抓取性能十分重要 [18]。此外，对于医院与家庭服务机器人而言，快速运动规划也不可缺少。动作规划问题可在三维 (3D) 工作空间内直接呈现并得到解决，如借助广泛运用的势场算法 [19]。然而，这些工作空间解决方案无法直接处理不同几何尺寸且具有机械约束因素的机器人。为了解决这些难题，可在一种名为配置空间 (C 空间) 的新型空间内呈现并解决运动规划问题 [20–22]。在配置空间内，将 3D 工作空间内具有复杂几何外形的机器人绘制为一个点，并且机器人的轨迹与高维配置空间内的连续曲线保持一致 (图 5)。根据配置空间公式，运动规划问题可通过两个步骤得到解决：

- (1) 构建配置空间的表达；
- (2) 根据计算的表达进行优化操作。

该动作规划流水线以配置空间为基础，运用过程十分成功，并广泛用于现实生活中需要得到优化规划解决方案的诸多规划应用中。当前研究人员已推出了配置

空间的多种不同表达，包括多面体 [23]、半代数集 [24, 25]、随机位图 [26] 和树图 [27]；并对不同配置空间表达提出了不同的优化方法，包括计算到配置空间内闭集边界的最短路径与最小距离等。此外，同样的流水线还可应用于一些动作规划算法，仅用于计算可行路径（如不会违反其他限制条件的无碰撞路径）。例如，不同变体形式的快速搜索随机树图 (RRT) [27] 可使用不同的启发式教学法将搜索向目标配置进行引导，与此同时，生成一个搜索树结构来作为配置空间的近似表达。该方法可视为上述流水线的变体形式，其中配置空间构建可与优化计算交替使用。

然而，基于 C 空间的该算法流水线在计算方面仍面临诸多挑战，其中两个最为重要的方面如下。

(1) 高效计算配置空间的近似表达或准确表达还比较困难，尤其针对高维配置空间的高自由度机器人。该配置空间的近似问题存在指数复杂性。

(2) 许多机器人应用程序需要进行实时规划，以便在具有移动障碍物的人类环境中让机器人进行可靠与高效操作，但是对配置空间的计算表达进行优化耗时较长。

本文将讨论配置空间面临的上述挑战的近期解决方案。首先笔者将展示如何将配置空间构建问题转换为机器学习问题，然后使用主动学习对近似配置进行高效与全面计算 (第 4 节)。然后，笔者使用基于并行图形处理器 (GPU) 的算法来加速配置空间内的优化计算操作，这样可以在许多挑战性环境中进行实时规划计算 (第 5 节)。

2 背景与相关工作

2.1 配置空间

配置空间是经典力学领域用来说明并分析许多重要系统运动的一个关键概念 [28]。通常来说，配置 q 作为独立参数的矢量，专门用来说明系统状态；C 空间的配置空间是某个指定系统的所有可能性配置的合集。例如，对于 n 点粒子系统，配置是一个可对所有粒子位置进行说明的矢量，并且相应的 C 空间为 \mathbf{R}^{3n} ；3D 刚体的配置包含其位置与方向，如果允许进行旋转与平移，那么配置空间为 $\mathbf{SE}(3)$ ，如果允许进行平移，则为 \mathbf{R}^3 ；关节式物体的配置为其自身所有关节角度的矢量。

机器人 A 的配置空间包括两个部分：无碰撞空间 $C_{\text{free}} = \{q : A(q) \cap B = \emptyset\}$ 以及碰撞空间或障碍空间 $C_{\text{obs}} = \{q : A(q) \cap B \neq \emptyset\}$ 。式中， B 相当于环境中障碍物的几何表达， $A(q)$ 相当于带有配置 q 的 A 。 C_{obs} 为闭集，其边界表示为接触空间 $C_{\text{cont}} = \partial C_{\text{obs}}$ ，与配置集相对应，其中 A

与 B 彼此之间仅接触，不穿透。图 2 显示的是两个物体的 C 空间，其中对 C_{cont} 使用橘色曲线进行了高亮显示。

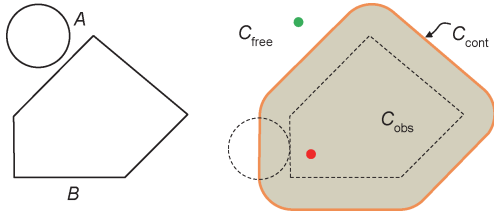


图 2. 两个物体的配置空间。橘色曲线高亮显示了 A 与 B 的接触空间 C_{cont} ；橘色曲线内/上的点为 C_{obs} ，橘色曲线外的点为 C_{free} ；红色与绿色点分别表示 C_{obs} 与 C_{free} 配置。

在特殊情况下，当 A 与 B 均为刚性物体时，机器人 A 仅进行平移运动， C_{obs} 等于 A 与 B 之间的闵可夫斯基和： $C_{\text{obs}} = A \oplus (-B) = \{x = x_A + x_B | x_A \in A, x_B \in -B\}$ 。闵可夫斯基和的范例之一详见图 2。当机器人 A 可以进行一般运动时（如平移与旋转）， C_{obs} 的几何学则更为复杂，其 2D 范例如图 3 所示。

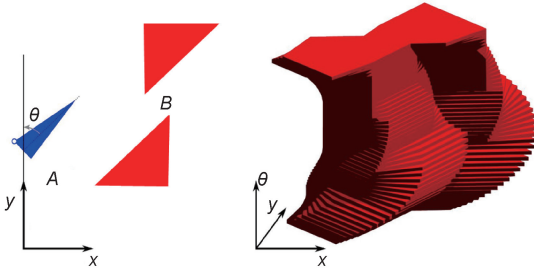


图 3. 2D 刚性物体 A 与 B 之间的 C_{obs} 。对于各旋转角度 $\theta \in [0, 2\pi)$ ，可以计算 $A(\theta)$ 与 $-B$ 之间的闵可夫斯基和，其中， $A(\theta)$ 表示将 A 在原点旋转 θ 角度后的最终形状。当把所有 θ 角度的闵可夫斯基和进行堆叠时，便可得出 A 与 B 之间的 C_{obs} 。

根据配置空间的概念，3D 工作空间内的运动规划问题可简化为 C 空间内点机器人的路径规划。例如，在 C_{free} 内找到与机器人指定原始配置与目标配置进行连接的曲线。

2.2 配置空间构建

在配置空间内进行运动规划计算之前，先决条件之一便是在适宜表达中计算 C 空间的几何学（如图像或表面）。由于 C 空间 $= C_{\text{free}} \cup C_{\text{obs}}$ ，并且 $C_{\text{free}} \cap C_{\text{obs}} = \emptyset$ ，我们仅需构建 C_{free} 或 C_{obs} 的表达。另外一个类似解决方案便是计算 C_{cont} 、 C_{free} 与 C_{obs} 之间的边界。

先前就配置空间构建的工作可通过两种不同方法实现，以几何学为基础和以拓扑学为基础的方法。以几何学为基础的方法对配置空间的准确几何表达进行计算，以拓扑学为基础的方法则捕捉配置空间的连通性。

由于涉及高维配置空间 C_{obs} 边界计算方面的组合复

杂性，以几何学为基础的方法通常会在低维配置空间方面受到限制。先前大多数工作与研究都侧重于特殊情况，也就是 A 物体与 B 物体均为刚体，并且仅进行平移运动。正如小节 2.1 所述，最终的 C_{obs} 为 A 与 $-B$ 之间的闵可夫斯基和。即便在这种特殊情况下，计算 C_{obs} 所涉及的计算复杂性也面临较高难度：当 A 与 B 均为凸面物体时，复杂性为 $O(mn)$ ，当 A 与 B 均为非凸面物体时，复杂性为 $O(m^3n^3)$ [29]，其中， m 与 n 分别为 A 与 B 的三角形数量。除了复杂性较高，大多数用于计算闵可夫斯基和的方法都会面临 3D 几何算法方面带来的挑战，尤其是当上述方法无法保证数值误差且易受退化影响（如无法可靠处理多边形集合或带孔网状物）时。

最近提出的一些方法 [30–32] 可以用来高效可靠计算近似闵可夫斯基和，但是这些方法面临稳定性方面的问题，并且在处理复杂物体方面也有较高复杂性。除了闵可夫斯基和的方法，还有其他一些方法也可用来计算 C_{obs} 。例如，Varadhan 等 [33] 借助适应性网格并计算 C_{obs} 近似值的方式来计算旋转与平移的 2D 物体；Zhang 等 [34] 使用单元分解方法来计算 4D C 空间的近似值。

以拓扑学为基础的方法则捕捉配置空间的连通性。先前大多数方法与研究通过使用采样技术 [26, 27] 试图捕捉 C_{free} 的连通性。其基本理念便是在 C_{free} 内生成随机样本（称为里程碑），然后使用位图结构或树形结构森林将这些样本组织起来（图 4）。由于 C_{free} 的拓扑学较复杂，并且可能包含多个部分或较小较窄通道，难以通过随机采样方式捕捉完整连通性。通过使用不同取样方法 [35–40] 来改善连通性计算方面已做了大量工作。近期的研究工作试图捕捉 C_{free} 与 C_{obs} 的拓扑学 [41]。与几何学为基础的方法相比，以拓扑学为基础的方法可更加快速计算近似 C 空间表达。然而，拓扑学方法在较窄通道方面的应用不太理想，同时在高自由度机器人的应用方面效率也较低。

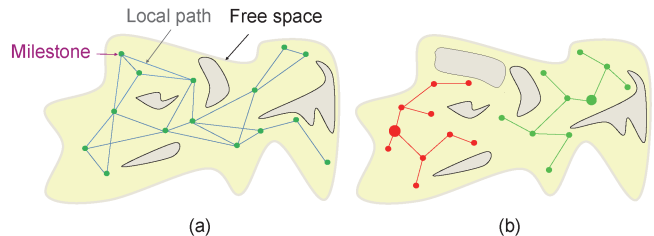


图 4. 用于配置空间计算的以拓扑学为基础的方法。(a) 使用位图结构捕捉 C_{free} ；(b) 使用树形结构森林捕捉 C_{free} 。

2.3 配置空间优化

一旦计算配置空间得以准确或近似表达，下一步需要对该 C 空间表达进行优化操作。例如，运动规划的目

标便是捕捉 C 空间内的轨迹，如图 5 所示。轨迹应满足下列约束条件：①轨迹应在 C_{free} 内；②轨迹应切实可行。例如，针对人形机器人而言，当机器人沿轨迹移动时不应摔倒。此外，在量度标准方面对轨迹进行优化则效果更佳。例如，最佳轨迹路线应距离最短，耗时最短，且与障碍物的距离保持最大。因此，运动规划可呈现为 C 空间内的约束优化问题。类似构想与方法还可应用于不同的条件，如侵入深度计算 [39, 42–44]。

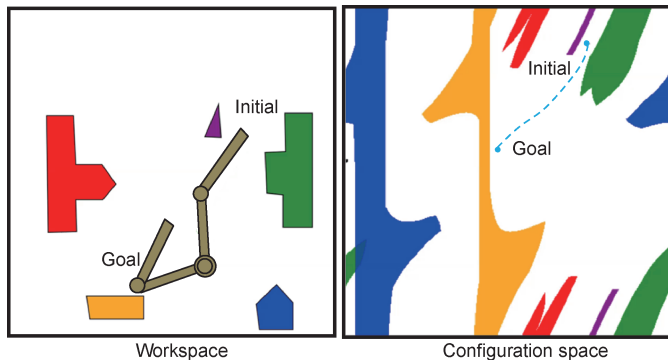


图 5. 工作空间与配置空间的运动规划。左图显示的是工作空间内的障碍物与双联机器人（初始设置与目标设置）。右图显示的是与左图工作空间一致的配置空间，其中使用不同颜色来说明工作空间与配置空间内障碍物的对应情况。蓝色曲线为连接至原始配置与目标配置的轨迹，并且也是运动规划算法的结果。

C 空间优化的计算通常代价较高，尤其是对于带有复杂结构与拓扑学的高维 C 空间。为了展示 C 空间优化问题面临的计算方面的挑战，笔者在此对 C 空间的运动规划进行举例说明。理论上，使用 C 空间准确表达的运动规划其计算复杂性较高。对于任何规划问题，如果计算找到了解决方案，或正确报告了不存在解决方案，那么规划算法可视为“完整”。PSPACE-hard [45] 与 PSPACE-complete[24] 已被证明是完整的规划算法，并且 kinodynamic 运动规划（如带有简单运动学或动力学约束的运动规划）已被证明为 NEXPTIME-hard [25]。对于带有一般微分约束的运动规划，其判定性仍处于未知状态 [46]。当使用 C 空间的近似表达时（如使用位图或树形结构来粗略估计 C_{free} 连通性），存在近似运动规划算法，该算法可保证概率完整性 [26, 27] 与 / 或渐进最优性 [47]。这些近似运动规划算法的复杂性通常受限于 $O(n \ln n)$ ，式中， n 代表 C 空间近似表达中使用的配置样本数 [47]。当 C_{free} 通道较窄并且 / 或维度较高时 [48]，由于 n 数值较大，这些近似算法的绩效仍远低于实时方法。

在过去的几十年间，人们已提出了诸多与 C 空间有关的不同规划方法，包括以优化为基础的规划算法，如 CHOMP [49] 和 TrajOPT [50]，以及以搜索为基础的算法，如 Anytime A* [51]。对于高自由度机器人的运动规划而言，大多数实用方法都以随机算法为基础，包括随机路

线图 (PRM) [26] 和 RRT[27]。

3 概述

对配置空间相关的挑战的解决方案包含两个部分：配置空间构建与配置空间优化。

3.1 高效配置空间构建

文献 [52] 的第一部分呈现了一种新型算法，在使用机器学习技术的情况下高效估算高维配置空间。主要的基本概念便是在配置空间内生成样本，然后使用这些样本来估算接触空间 C_{cont} ，其方法便是对表面进行分离，这些表面能正确分离所有碰撞与无碰撞样本。使用 SVM 分类法来计算分离表面。通过权衡增量与主动学习技术，该方法可极大降低所需的样本量。当样本量增加时，通过该方法计算的近似接触表面可以快速收敛至准确接触表面；笔者还会提供近似接触空间内预期误差的界限与范围，并评估该算法在高维基准方面的绩效。

为了构建配置空间的表达，笔者使用离线学习算法，如图 6 左侧方框所示。首先对两个指定物体在 C 空间的子空间内生成一个统一样本小集。下一步通过对两个物体进行准确碰撞检查的方式来证明这些配置是否处于 C_{free} 内或 C_{obs} 内。使用 $c(q) \in \{-1, +1\}$ 来表示配置 q 的碰撞状态。例如，如果 $q \in C_{\text{obs}}$ ，那么 $c(q) = +1$ ；如果 $q \in C_{\text{free}}$ ，那么 $c(q) = -1$ 。考虑到所有配置样本的碰撞状态，借助分类器来计算接触空间的大致估值 LCS_0 （图 6(b)），其中 LCS 表示学习接触空间。接下来在 C 空间内选择新样本，并通过使用主动学习法来进一步提升初始表达的准确性 LCS_0 。在主动学习期间，可以选择距离先前样本（探测）（图 6(c)）较远的样本，或距离 LCS_0 （探测）（图 6(d)）较近的样本。在生成新样本后，根据增量机器学习技术，计算更新的近似 LCS_1 （图 6(e)）。重复该操作，生成一系列带有更高准确度的近似表达 LCS_0, LCS_1, \dots 重复操作该迭代过程，直至当前近似值可以正确预测所有新样本的碰撞状态。最终结果 LCS （图 6(f)）与接触空间的光滑表面近似保持一致。

3.2 配置空间的高效优化

通过笔者第一部分工作计算出的近似接触空间可直接用于运动规划，如可在轨迹优化框架内将其用做无碰撞约束 [53]。然而，该约束仅针对高度非凸面情况，因此需要解决最终的优化问题仍然面临较大挑战，这也是笔者进行工作考虑的一个方面。解决运动规划问题的另一方法便是使用以样本为基础的配置空间近似法，该方

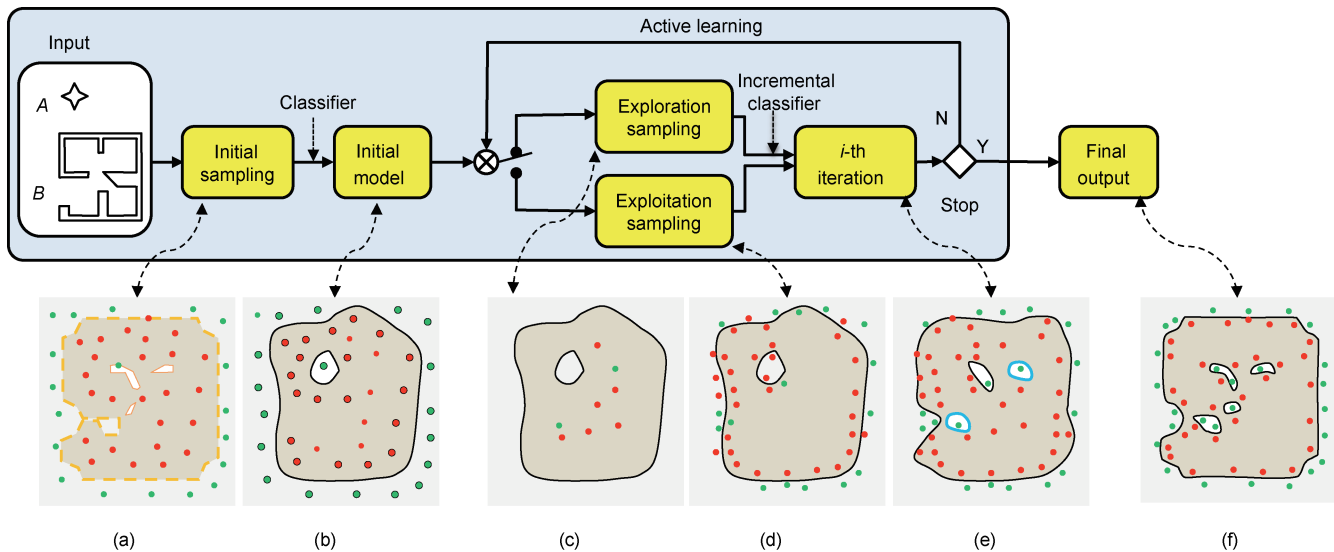


图 6. 该图显示的是 C_{cont} 近似的离线计算流水线。绿点表示无碰撞配置样本，红点表示碰撞样本。

法已应用至许多传统规划算法中，如 PRM。文献 [54] 的第二部分呈现了一种高自由度机器人实时运动规划的新型并行算法，该算法开发出了价值 \$400 的 GPU 的计算能力。当前的 GPU 为可编程多核处理器，支持成千上万的并行线程。笔者使用 GPU 进行 PRM 与惰性规划器的实时计算，所呈现的高效并行策略可以构建路线图，该路线图包括样本生成、碰撞检测、连接附近样本与局部规划。根据图片搜索，也可通过并行方式进行查询阶段。为了设计出高效的独立查询规划器，笔者采用推迟碰撞检测与本地规划的惰性策略，还将呈现一些新型以层级为基础的碰撞检测算法来提高整体性能。

笔者选择 PRM 算法作为并行规划的基本方法，因为 PRM 算法最适合利用 GPU 的多核与数据并行。PRM 算法分两个阶段：路线图构建和路线图查询。路线图构建包括 4 个主要步骤：①在配置空间内生成样本；②通过进行离散碰撞查询，计算与自由空间内样本相对应的里程碑；③找到与各里程碑最近的其他里程碑；④使用局部规划来连接附近的里程碑，同时形成路线图。查询阶段包括两个部分：①将查询的原始配置与目标配置连接至路线图；②对路线图进行图形搜索算法，并找到碰撞自由路径。

PRM 算法的个别部分，如碰撞查询的并行结果并不是令人非常满意 [55]。我们可以使用多核 GPU 来大幅提升其他部分的性能。针对 GPU 的 PRM 算法的框架详见图 7。笔者通过高效方式将 PRM 算法的 6 个步骤分别进行并行。首先，多核 GPU 的各线程生成机器人的随机配置，同时将某些配置与障碍物进行碰撞。所有无碰撞样本均为里程碑，并且成为路线图图表的顶点。接下来，各 GPU 线程对单个里程碑的 k 最近邻进行计算，并收集

所有近邻对。然后各线程检验是否可以通过局部规划方式将这些近邻对进行连接。如果里程碑近邻对之间不存在无碰撞路径，则向路线图添加边缘。一旦建成路线图，将查询并行连接至路线图，使用并行图搜索算法来找到路径。

最终基于 GPU 的框架对多查询版本的规划问题十分高效。该计算代价最高的步骤便是局部规划算法；因此，新型碰撞检测算法被用来提升其性能。为了加速单查询算法，笔者引入一种解决方案，该方案使用惰性策略，推迟局部规划的碰撞检测。换言之，该算法将与最近邻相对应的所有边缘进行连接，同时搜索原始配置与最终配置之间的路径。然后，在构成这些路径的边缘处执行局部规划。

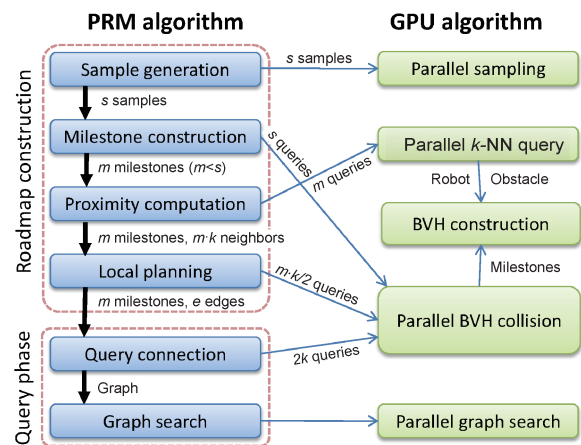


图 7. 算法中的 PRM 概述与并行部分。

4 基于学习的近似接触空间构建

在该节中，笔者的算法可呈现于接触空间的离线学

习与 LCS 的计算。该算法的不同阶段详见图 6。

4.1 初步采样

在 C 空间中进行均匀采样，获取一组配置点。并非对整个 C 空间进行取样，而是在包含 C_{cont} 的子空间内生成样本。考虑到 A 与 B 物体，将接触空间 C_{cont} 包含在它们的包围体 $BV(A)$ 与 $BV(B)$ 的碰撞空间内。

4.2 计算 LCS_0

在拥有一组 C_{obs} ($BV(A)$ 与 $BV(B)$) 的 k 样本情况下，对 A 与 B 进行准确的碰撞查询，检查这些样本是否处于碰撞空间内。目标是从这些配置中学习近似表达 LCS_0 。 LCS_0 与决策函数 $f(\mathbf{q}) = 0$ 一致，完全由 C 空间内配置 \mathbf{S} 集确定。将 $f(\mathbf{q})$ 视为分类器，并使用分类器来预测，指定的配置 \mathbf{q} 不发生碰撞 ($f(\mathbf{q}) < 0$) 还是发生碰撞 ($f(\mathbf{q}) > 0$)。 \mathbf{S} 相当于支撑矢量，支撑矢量是用于学习的配置样本的一个小子集。直觉上， \mathbf{S} 是距离 C_{cont} 最近的样本。

笔者使用 SVM 分类器 [56] 从 k 配置的初步采样中学习 LCS_0 。使用硬间隔 SVM 作为基础样本，这些基础样本可分离至无碰撞空间与碰撞空间内。直观上，SVM 使用函数从输入空间至更高（可能无限）维度特征空间来绘制指定样本 $\{\mathbf{q}_i\}$ 。SVM 对线性分离超平面进行计算，该超平面具有参数 \mathbf{w} 与 b 特性。超平面的最大间隔位于更高维度特征空间内。超平面相当于输入空间内的非线性分离表面。 \mathbf{w} 是超平面的法向量，参数 b 决定沿法向量原点的超平面偏移。在特征空间内，超平面与最近样本点之间的距离叫做“间隔”，最佳分离超平面应将该距离最大化。可通过解决下列优化问题来实现最大间距：

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & c_i (\mathbf{w} \cdot \phi(\mathbf{q}_i) + b) \geq 1, \quad 1 \leq i \leq k \end{aligned} \quad (1)$$

若 $c_i \in \{-1; +1\}$ 是各样本 \mathbf{q}_i 的碰撞状态，将 $K(\mathbf{q}_i, \mathbf{q}_j) = \phi(\mathbf{q}_i)^T \phi(\mathbf{q}_j)$ 确定为核函数（如用来计算特征空间内积的函数），在结果中使用 RBF 核（径向基函数核）。

式 (1) 的解是输入空间内的非线性表面，该表面可对无碰撞配置与碰撞配置进行分离。该解可通过下列方式进行表达：

$$f(\mathbf{q}) = \mathbf{w}^* \cdot \phi(\mathbf{q}) + b^* = \sum_{i=1}^k \alpha_i c_i K(\mathbf{q}_i, \mathbf{q}) + b^* \quad (2)$$

式 (2) 中， \mathbf{w}^* 与 b^* 为式 (1) 的解，并且 $\alpha_i \geq 0$ 。与非零 α_i 一致的矢量 \mathbf{q}_i 被称为支撑矢量，本文表示为 \mathbf{S} 。直观而言，支撑矢量为分离超平面 $f(\mathbf{q}) = 0$ 最近的这些样本，

如图 6 (b) 中的较大红点与绿点所示。因此， LCS_0 包含隐函数 $f_{LCS_0}(\mathbf{q}) = f(\mathbf{q})$ 与样本集 $\mathbf{S}_{LCS_0} = \mathbf{S}$ （如支撑矢量），可用来估算准确的接触空间。

4.3 使用主动学习法来改善 LCS_0

使用主动学习法来改善 LCS_0 。目标是主动选择新样本，以便通过将这些样本合并至 LCS_1 中来获得更佳的近似接触空间表达 LCS_0 。结合探索法与开发法 [57]，通过抛偏倚硬币的方式来确定使用探索法还是开发法，其中有一定概率硬币正面朝上（最初概率 0.5）。如果正面朝上，则使用探索法；反之，则使用开发法。通过探索样本的碰撞状态部分（可由当前 LCS_i 进行正确预测）来对硬币正面朝上的概率进行调整。使用新样本来更新 LCS_0 ，并生成一个新的近似 LCS_1 （或从 LCS_i 至 LCS_{i+1} 进行改善）。重复主动学习步骤操作，直至当前 LCS_i 正确预测新样本，或最终结果（作为 LCS ）有足够准确度来估算 C_{cont} 。

4.4 增量学习

采用增量学习技术从 LCS_i 高效计算 LCS_{i+1} ，而非通过使用先前样本重新开始计算新决策函数。增量学习使用一小组新样本来更新 LCS_i 。 LCS_i 的决策函数作为生成 LCS_{i+1} 的初始猜测值。增量 SVM [58] 可以更新使用 SVM 生成的当前结果；其关键在于，在添加新样本的同时，保留所有先前样本（式 (1)）的最优性条件（如库恩-塔克条件）。这一点可通过调整式 (2) 中的系数 α_i 和 b 及支撑矢量集 \mathbf{S} 的方式来实现。系数调整与支撑矢量变更可通过式 (2) 中的目标函数梯度进行指导。

4.5 终止主动学习

满足下列任一条件，主动学习终止。

(1) 当前近似 LCS_i 可以正确预测探索与开发期间生成的所有新样本的碰撞状态。

(2) 主动学习迭代中使用的样本总数超过用户指定限值。

第一个需要保证的条件便是用于学习 LCS 的所有配置必须保持一致（如 LCS 能够正确预测这些配置）。这意味着当前 LCS 非常近似于基本接触空间。第二个条件便是控制近似 C_{cont} 的准确性。由于使用更多样本，更加近似 C_{cont} 。

5 基于 GPU 的实时配置空间优化

在该节，笔者将呈现如何使用 GPU 来加速配置空间

的优化速度的相关细节。

5.1 层级计算

笔者为环境中的机器人与各障碍物创建包围体层次 (BVH)，加速碰撞查询。使用文献 [59] 介绍的基于 GPU 的构建算法，对于特定三角形表达，该算法可以构建 GPU 内并行的按坐标轴排列的包围盒 (AABB) 或有向包围盒 (OBB)。对于碰撞检测，笔者使用 OBB 层级，因为它可以在 GPU 的类似构架上实现更高的剔除效率与改善性能。这些层级保存在 GPU 内存中，笔者可以对不同配置采取适当改造。

5.2 路线图构建

路线图构建阶段试图捕捉自由配置空间的连通性，这也是 PRM 算法主要计算的密集部分。

(1) 样本生成：首先需要在配置空间内生成随机样本。由于样本属于独立形式，需要安排足够的并行线程来使用 GPU 与 MD5 密码杂凑函数 [60]，在实践中，该函数可在无需共享种子的情况下实现良好的随机性。

(2) 里程碑计算：对于在先前步骤中生成的各配置，我们需要检查其是否为里程碑。例如，位于自由空间内，并且不与障碍物发生碰撞的配置。在借助 BVH 来测试样本确定的配置中障碍物与机器人之间重叠的情况下，笔者使用层级碰撞检测法。通过在两个 BVH 中使用遍历算法的方式，在各线程中进行碰撞检测。遍历算法从两个 BVH 根节点开始，并测试 OBB 节点是否存在递归方式的重叠。如果两个节点重叠，那么其他结果的所有可能性配对进行递归检测来检查交叉情况。

通过使用并行层级构建算法 [59]，笔者也会使用 GPU 来计算机器人与障碍物的实际 BVH 结构。由于机器人的几何对象根据配置进行移动，其 BVH 仅对初始配置有效。为了避免重新计算各配置的 BVH，在进行重叠测试之前，笔者借助当前配置样本来改变机器人 BVH 的各节点。因此，仅改变碰撞检测期间实际所需的那些节点。

(3) 邻近计算：对于计算的各里程碑，我们需要找到其 k 最近邻 (k -NN)。通常存在两种 k -NN 算法：准确 k -NN 与近似 k -NN，通过进行稍许松弛，近似 k -NN 速度更快。该邻近算法以范围查询为基础，范围查询使用配置空间中点的 BVH 结构。

对于 3 自由度欧几里得空间，首先使用并行算法 [59] 来对所有里程碑构建 BVH 结构。对于各配置 q ，将其包围在按坐标轴排列的 ε 盒中：带有 q 的该盒作为中心， 2ε 作为边长。接下来遍历 BVH 树，找到 ε 盒中的所有叶节点（如配置）。这样就简化成了 q 的范围查询。对于

非欧几里得自由度，我们复制样本并将其局部转换至欧几里得空间内。例如，假设一个自由度为旋转度 $\alpha \in [0, 2\pi]$ 。在 α 的 2π 距离内添加另一个样本 $\alpha^* \in [-\pi, 3\pi]$ 。如果 3 自由度都在旋转，则需要为每个里程碑添加另外 7 个样本。一旦范围查询结束，从所有查询结果中选择 k 最近邻的样本；这样就可以了解到准确的最近邻。通过使用分解策略，该方法可延伸至高维空间内的 k -NN 搜索的处理方面，详见笔者最近的研究工作 [61]。

为了进一步提升高维空间内邻近计算的性能，我们开发了一种新型 k -NN 算法，该算法使用局部敏感散列法 (LSH) 与布谷鸟散列来高效计算 GPU 上并行的 k -NN。更多详情请参阅笔者近期的研究工作 [62]。

(4) 局部规划：局部规划可检查两个里程碑之间是否存在与路线图边缘相一致局部路径，这也是整个 PRM 算法代价最高的部分。假设拥有 n_m 里程碑，并且每个里程碑具有最多 n_k 最近邻。那么算法在最多 $n_m \cdot n_k$ 倍数情况下进行局部规划。如果使用离散碰撞检测 (DCD)，那么需要执行最多 $n_l = n_m \cdot n_k \cdot n_l$ 碰撞查询，这对于复杂基准而言频率过高。对于多查询问题，该成本费用可在多个查询上进行分摊，因为路线图仅构建一次。对于单查询问题，计算整个路线图则代价过高。

因此，在单查询情况下，除非绝对必要，笔者都采用惰性策略来推迟局部规划。如果进行查询，我们会从初始配置到最终配置期间计算路线图内的若干不同候选路径，但是仅检查候选路径上路线图边缘的局部规划。局部规划可判定某些边缘无效，在这种情况下，将无效边缘从路线图中删除。如果存在无效边缘的候选路径，那么算法已找到了无碰撞解决方案。否则，我们在更新的路线图上再次计算候选路径，并重复上述操作。这种惰性策略可以极大提升单查询的性能。

5.3 查询阶段

查询阶段包括两部分：将查询连接至路线图，执行图搜索来找到路径。

(1) 查询连接：如果是单查询的初始-目标配置，则将配置连接至路线图。对于上述两个配置，笔者在路线图上找到 k 最近里程碑，并在能够被局部规划连接的查询与里程碑之间添加边缘。笔者从路线图构建阶段使用同样算法，除非为了增加找到路径的概率而将使用的 k 增大 2 至 3 倍。

(2) 图搜索：搜索算法试图在连接初始配置与目标配置的路线图上找到目标。笔者使用深度优先搜索 (DFS) 或广度优先搜索 (BFS) 进行图搜索。对于多查询情况，在借助 DFS 的同时，GPU 线程遍历路线图进行一次查

询，最终结果为无碰撞路径。对于单查询情况，在借助 BFS 的同时，笔者利用所有 GPU 线程来找到路径并进行一次查询。对于远离初始节点同样步骤数目的节点，笔者将它们的未访问邻添加至并行队列中。换言之，不同的 GPU 核遍历图的不同部分。该方法面临的主要难题便是，随着 BFS 遍历不断进行，需要动态生成该工作流程，并且不同核上的计算负荷变化极为明显。为了解决负荷均衡与任务分配的问题，需要保持所有核的并行，笔者在文献 [63] 中使用轻质负荷均衡方法。

6 实验

在该节，在解决配置空间相关的两个问题的过程中，笔者探索了所使用方法的性能。

6.1 配置空间构建

笔者使用一组复杂基准来评估该算法的接触空间近似结果。

首先，计算 2D 物体的接触空间，如图 6 中的学习流水线输入所示。在该实验中， A 物体仅进行平移运动， B 物体固定不动。最终配置空间拥有两个自由度，通过基于学习的方法计算的 LCS 系列如图 8 所示。可以看到，在经过几百样本的数次迭代之后，学习的 LCS 可以充分估测两个物体之间的准确接触空间。

接下来，计算同样一对 2D 物体的接触空间，但是当前允许 A 物体在 2D 平面进行旋转。最终的配置空间拥有 3 个自由度，并且通过该方法生成的 LCS 表面系列如图 9 所示。与 2D 接触空间情况一样，学习的接触空间可以快速充分逼近至准确的接触空间。

最终，计算一对 3D 物体的接触空间，如图 10 (a) 所示，其中 A 物体仅进行平移运动，因此相应的配置空间为 3D 模式。同样，笔者观察到学习的接触空间序列快速收敛至准确的接触空间。

6.2 配置空间优化

笔者阐述了该算法的实施详情，并对该算法就一组

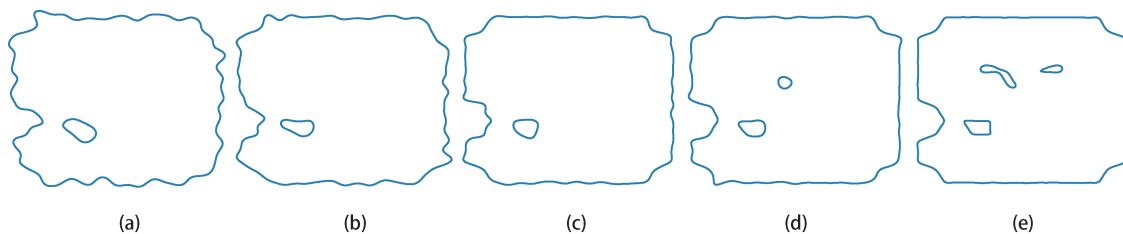


图 8. 借助主动学习法，对 2D 星星模型与房间模型之间的 2D 接触空间进行 LCS 计算，如图 6 所示的学习流水线输入。对与 LCS_i 相应的支撑矢量数量进行了高亮显示。(a) $i=0$, $|S|=37$; (b) $i=4$, $|S|=75$; (c) $i=8$, $|S|=198$; (d) $i=14$, $|S|=327$; (e) $i=20$; $|S|=654$ 。

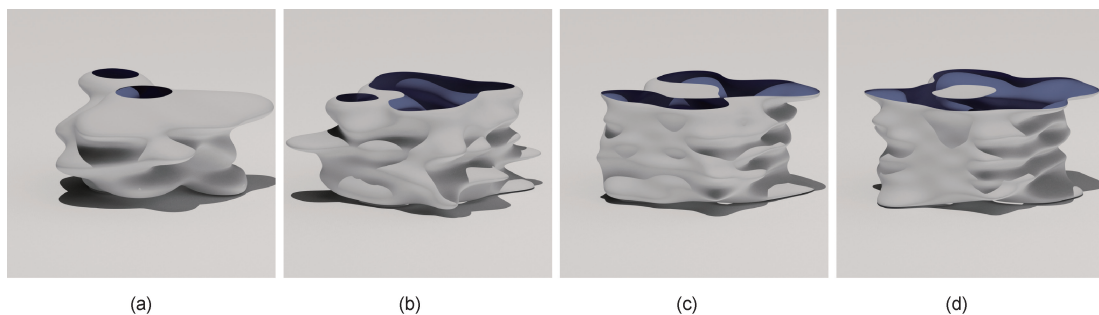


图 9. 借助主动学习法，对 2D 星星模型与房间模型之间的 3D 接触空间进行 LCS 计算。如图 6 所示的学习流水线输入。对与 LCS_i 相应的支撑矢量数量进行了高亮显示。垂直轴代表 C 空间的旋转分量。(a) LCS_0 , $|S|=88$; (b) LCS_5 , $|S|=174$; (c) LCS_9 , $|S|=237$; (d) LCS_{12} , $|S|=248$ 。

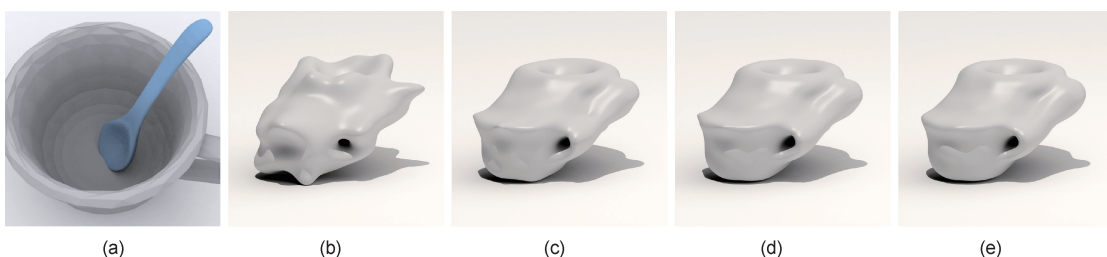


图 10. 借助主动学习法，对 3D 杯子与勺子之间的 3D 接触空间进行 LCS 计算。笔者提供与 LCS_i 一致的支撑矢量数目。如图所示，该算法可在数个迭代内计算得到一个很好的近似值。

基准的实施性能进行了评估。借助 Intel Core i7 CPU（约 \$600），在 3.2 GHz CPU 与 6 GB 内存条件下，将报告的所有时间控制在机器上进行呈现。笔者在借助视频内存为 1 GB 的 NVIDIA GTX 285 GPU（约 \$380）情况下执行该算法。

对于多查询规划问题，笔者在 GPU（G-PRM）上执行 PRM 算法；对于单查询问题而言，笔者执行惰性版本（GL-PRM）。笔者将这些情况与 OOPSMP 文库中执行的 PRM 及 RRT 算法进行对比 [64]，OOPSMP 文库是一种在 CPU 上进行运动规划算法的常见文库。使用的基准详见图 11。对比操作设计如下：对于各基准，找到一个适当设置，其中 C-PRM 找到解决方案，然后使用相对数量的样本来运行 G-PRM；接下来，按照 G-PRM 的同样设置来运行 GL-PRM，同时按照 C-PRM 的同样设置来运行 C-RRT。

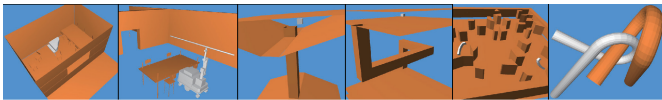


图 11. 依照下列顺序，用于该算法的基准场景：钢琴（2484 三角形）、直升机（2484 三角形）、maze3d1（40 三角形）、maze3d2（40 三角形）、maze3d3（970 三角形）与 alpha puzzle（2016 三角形）。

表 1 显示的是两种算法之间的时间对比情况。通常而言，G-PRM 比 C-PRM 快约 10 倍，并且对于单查询问题，GL-PRM 可以再次实现 10 倍加速。甚至在动态场景下，G-PRM 也快于 C-PRM。

当前的 C-RRT 与 C-PRM 均为单核版本。然而，甚至多核版本的 PRM 也仅能将时间效率提升最多 4 倍，因为在一个 8 核 CPU 上难以线性测量层级计算与最近邻计算。相比较 CPU 算法，笔者的 GPU 算法仍然可以将性能提升 1~2 个数量级。

图 12 显示的是 G-PRM 与 GL-PRM 不同阶段的时间分类。两种算法性能之间的差异十分明显：在 G-PRM 中，局部规划处于瓶颈状态，并支配时间，然而在 GL-PRM 中，

表 1. 左侧两列对 OOPSMP 中的 PRM 与 RRT 算法的性能进行了评估。右侧两列对笔者基于 GPU 的算法性能进行了评估。

	C-PRM	C-RRT	G-PRM	GL-PRM
Piano	6.53 s	19.44 s	1.71 s	111.23 ms
Helicopter	8.20 s	20.94 s	2.22 s	129.33 ms
Maze3d1	138.00 s	21.18 s	14.78 s	71.24 ms
Maze3d2	69.76 s	17.40 s	14.47 s	408.60 ms
Maze3d3	8.45 s	4.30 s	1.40 s	96.37 ms
Alpha1.5	65.73 s	2.80 s	12.86 s	1446.00 ms

图搜索耗时较长，因为局部规划在惰性模式与输出敏感模式中进行操作。在 GL-PRM 中，三个部分最为耗时：里程碑构建、邻近计算与图搜索，因为它们必须进行大量碰撞查询。如果环境较为杂乱并且模型的几何学较为复杂，那么里程碑构建过程将会更慢（图 12 中的 alpha puzzle）。如果环境属于空旷空间并且具有许多里程碑，那么邻近计算将会到达瓶颈（图 12 中的 maze3d2）。如果惰性策略无法猜到正确路径，由于大量碰撞查询（图 12 中的 maze3d3），那么图搜索将进行大量计算工作。然而，在所有这些环境中，GL-RPM 优于其他方法。

笔者测试了 maze3d3 基准上的 G-PRM 与 GL-PRM 的可伸缩性，结果详见图 13 所示。显然，GL-PRM 快于 G-PRM，并且两种算法在该基准上都实现了近线性伸缩。然而，根据观察，随着样本数量的增加，GL-PRM 快于 G-PRM 的速度放缓。这是因为当样本数量增加时，邻近计算的代价越来越大，并且在样本数量接近 100 万时非常耗时。

笔者的方法可延伸至关节体的高效规划，并且可实现 PR2 抓取操作的实时性能，如图 14 所示。

7 总结

本研究解决了该领域中的一些重要难题，如与配置空间相关的两个计算难题：配置空间构建与配置空间

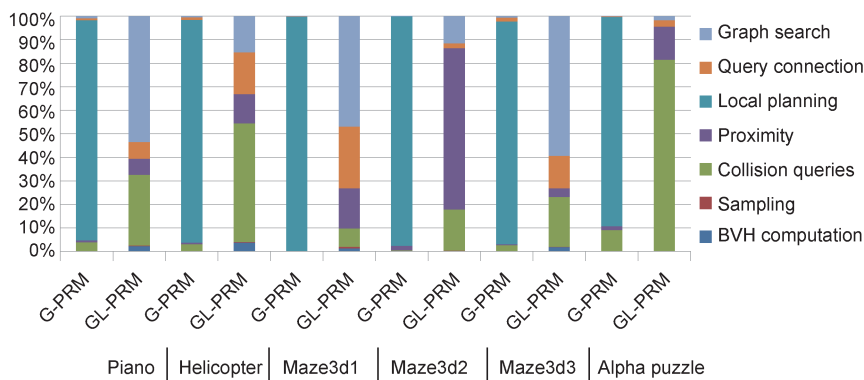


图 12. 时间分割：G-PRM 与 GL-PRM 不同部分花费的时间片段。

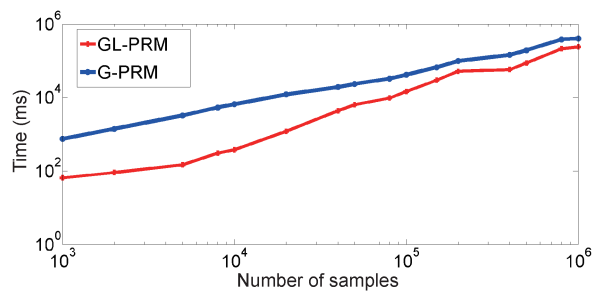


图 13. G-PRM 与 GL-PRM 的可伸缩性。

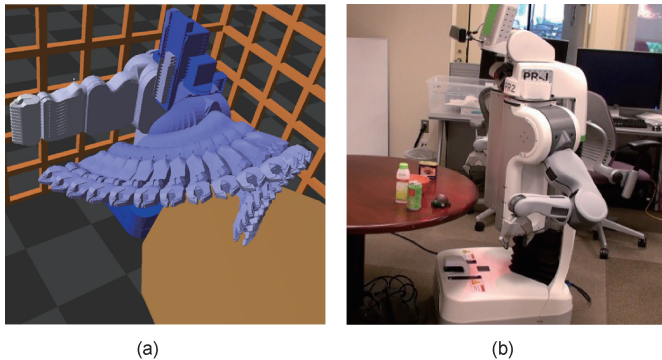


图 14. 笔者基于 GPU 的运动规划器可在不到 1 s 时间内计算 PR2 的无碰撞路径。(a) PR2 模拟；(b) 实时 PR2。

的高效优化。由于配置空间的相关研究仍在继续，我们希望能在这些领域取得进一步进展，还有更多工作需要钻研。

概述本文主要研究成果：首先讲述了配置空间近似的一种新方法。笔者的主要理念是根据机器学习分类器，尤其是支撑矢量机器，对配置空间进行取样，并估算接触空间。其次，在预先计算期间，使用主动学习技术来选择样本。接下来，引入了一种在 GPU 上进行的全动作规划算法。该算法可以利用 PRM 算法中的所有并行，包括 PRM 框架提供的高级并行与 PRM 算法不同部分内的低级并行，如碰撞检测与图搜索。这使得笔者首创了借助 GPU 在一般环境下进行的实时运动规划与全局导航。

致谢

该研究得到了美国陆军研究办公室、美国国家科学基金会、Willow Garage 公司与香港大学基础研究基金项目的部分支持。

Compliance with ethics guidelines

Jia Pan and Dinesh Manocha declare that they have no con-

flikt of interest or financial conflicts to disclose.

References

1. G. Litzemberger. Professional service robots: Continued increase. *Statistical Department, International Federation of Robotics, Tech. Rep.*, 2012
2. E. Guizzo, E. Ackerman. How Rethink Robotics built its new Baxter robot worker. *IEEE Spectrum*, <http://spectrum.ieee.org/robotics/industrial-robots/rethink-robotics-baxter-robot-factory-worker>, 2012
3. K. Yamazaki, et al. Home-assistant robot for an aging society. *Proc. IEEE*, 2012, 100(8): 2429–2441
4. S. Cousins. Robots for humanity. <http://www.willowgarage.com/blog/2011/07/13/robots-humanity>, 2012
5. S. Kajita, et al. Cybernetic human hrp-4c: A humanoid robot with human-like proportions. In: C. Pradalier, R. Siegwart, G. Hirzinger, eds. *Robotics Research, ser. Springer Tracts in Advanced Robotics*, vol 70. Berlin: Springer, 2011, 70: 301–314
6. M. Montemerlo, et al. Junior: The Stanford entry in the urban challenge. *J. Field Robot.*, 2008, 25(9): 569–597
7. M. Bonfe, et al. Towards automated surgical robotics: A requirements engineering approach. In: *Proceedings of IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechanics*, 2012: 56–61
8. E. Guizzo. Robots enter fukushima reactors, detect high radiation. *IEEE Spectrum*, 2011. <http://spectrum.ieee.org/automaton/robotics/industrial-robots/robots-enter-fukushima-reactors-detect-high-radiation>
9. E. Ackerman. Latest alphasdog robot prototypes get less noisy, more brainy, 2012. <http://spectrum.ieee.org/automaton/robotics/military-robots/latest-ls3-alphasdog-prototypes-get-less-noisy-more-brainy>
10. S. Thrun, W. Burgard, D. Fox. *Probabilistic Robotics*. Boston, MA: The MIT Press, 2005
11. R. B. Rusu, N. Blodow, Z. C. Marton, M. Beetz. Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments. In: *Proceedings of International Conference on Intelligent Robots and Systems*, 2009: 1–9
12. T. Lozano-Pérez, J. L. Jones, E. Mazer, P. A. O’Donnell. Task-level planning of pick-and-place robot motions. *Computer*, 1989, 22(3): 21–29
13. L. Kaelbling, T. Lozano-Pérez. Unifying perception, estimation and action for mobile manipulation via belief space planning. In: *Proceedings of IEEE International Conference on Robotics and Automation*, 2012: 2952–2959
14. L. Kaelbling, T. Lozano-Pérez. Hierarchical task and motion planning in the now. In: *Proceedings of IEEE International Conference on Robotics and Automation*, 2011: 1470–1477
15. R. F. Stengel. *Optimal Control and Estimation (Dover Books on Advanced Mathematics)*. New York: Dover Publications, 1994
16. K. J. Astrom, B. Wittenmark. *Adaptive Control*. 2nd ed. Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1994
17. V. Unhelkar, J. Perez, J. Boerkoel, J. Bix, S. Bartscher, J. Shah. Towards control and sensing for an autonomous mobile robotic assistant navigating assembly lines. In: *IEEE International Conference on Robotics and Automation*, 2014: 4161–4167
18. L. P. Ellekilde, H. G. Petersen. Motion planning efficient trajectories for industrial bin-picking. *Int. J. Robot. Res.*, 2013, 32(9–10): 991–1004
19. O. Khatib. Real-time obstacle avoidance for manipulators and mobile robot. *Int. J. Robot. Res.*, 1986, 5(1): 90–98

20. T. Lozano-Pérez, M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 1979, 22(10): 560–570
21. T. Lozano-Pérez. Automatic planning of manipulator transfer movements. *IEEE Trans. Syst. Man Cybern.*, 1981, 11(10): 681–698
22. T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Trans. Comput.*, 1983, C-32(2): 108–120
23. B. Chazelle. Approximation and decomposition of shapes. In: J. T. Schwartz, C. K. Yap, eds. *Algorithmic and Geometric Aspects of Robotics*. Hillsdale: Lawrence Erlbaum Associates, 1987: 145–185
24. J. F. Canny. Some algebraic and geometric computations in PSPACE. In: *Proceedings of ACM symposium on Theory of Computing*, 1988: 460–467
25. J. F. Canny, J. Reif, B. Donald, P. Xavier. On the complexity of kinodynamic planning. In: *Proceedings of Symposium on Foundations of Computer Science*, 1988: 306–316
26. L. Kavraki, P. Svestka, J. C. Latombe, M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.*, 1996, 12(4): 566–580
27. J. J. Kuffner, S. LaValle. RRT-connect: An efficient approach to single-query path planning. In: *Proceedings of IEEE International Conference on Robotics and Automation*, 2000: 995–1001
28. V. I. Arnold. *Mathematical Methods of Classical Mechanics*. New York: Springer-Verlag, 1989
29. D. Halperin. Robust geometric computing in motion. *Int. J. Robot. Res.*, 2002, 21(3): 219–232
30. J. M. Lien. Covering Minkowski sum boundary using points with applications. *Comput. Aided Geom. Des.*, 2008, 25(8): 652–666
31. J. M. Lien, N. M. Amato. Approximate convex decomposition of polyhedral. In: *Proceedings of the ACM Symposium on Solid and Physical Modeling*, 2007: 121–131
32. J. M. Lien. A simple method for computing Minkowski sum boundary in 3D using collision detection. In: *Algorithmic Foundation of Robotics VIII, Springer Tracts in Advanced Robotics*, vol 57. Berlin: Springer, 2009: 401–415
33. G. Varadhan, Y. J. Kim, S. Krishnan, D. Manocha. Topology preserving approximation of free configuration space. In: *Proceedings of International Conference on Robotics and Automation*, 2006: 3041–3048
34. L. Zhang, Y. J. Kim, D. Manocha. A hybrid approach for complete motion planning. In: *Proceedings of International Conference on Intelligent Robots and Systems*, 2007: 7–14
35. N. M. Amato, O. B. Bayazit, L. K. Dale, C. Johns, D. Vallejo. OBPRM: An obstacle-based prm for 3D workspaces. In: *Proceedings of Workshop on the Algorithmic Foundations of Robotics on Robotics*, 1998: 155–168
36. V. Boor, M. Overmars, A. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In: *Proceedings of IEEE International Conference on Robotics and Automation*, 1999: 1018–1023
37. D. Hsu, L. E. Kavraki, J. C. Latombe, R. Motwani, S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In: *Proceedings of Workshop on the Algorithmic Foundations of Robotics on Robotics*, 1998: 141–153
38. S. Rodriguez, X. Tang, J. M. Lien, N. M. Amato. An obstacle-based rapidly-exploring random tree. In: *Proceedings of IEEE International Conference on Robotics and Automation*, 2006: 895–900
39. L. Zhang, D. Manocha. An efficient retraction-based RRT planner. In: *Proceedings of IEEE International Conference on Robotics and Automation*, 2008: 3743–3750
40. Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, J. H. Reif. Narrow passage sampling for probabilistic roadmap planners. *IEEE Trans. Robot.*, 2005, 21(6): 1105–1115
41. J. Denny, N. M. Amato. Toggle PRM: Simultaneous mapping of C-free and C-obstacle—A study in 2D. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011: 2632–2639
42. L. Zhang, Y. J. Kim, G. Varadhan, D. Manocha. Generalized penetration depth computation. *Comput. Aided Des.*, 2007, 39(8): 625–638
43. L. Zhang, Y. J. Kim, D. Manocha. A fast and practical algorithm for generalized penetration depth computation. In: *Proceedings of Robotics: Science and Systems*, 2007
44. C. Je, M. Tang, Y. Lee, M. Lee, Y. J. Kim. Polydepth: Realtime penetration depth computation using iterative contact-space projection. *ACM Transactions on Graphics*, 2012, 31(3): 5:1–5:14
45. J. H. Reif. Complexity of the mover’s problem and generalizations. In: *Proceedings of Annual Symposium on Foundations of Computer Science*, 1979: 421–427
46. P. Cheng, G. Pappas, V. Kumar. Decidability of motion planning with differential constraints. In: *Proceedings of International Conference on Robotics and Automation*, 2007: 1826–1831
47. S. Karaman, E. Frazzoli. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.*, 2011, 30(7): 846–894
48. D. Hsu, J. C. Latombe, H. Kurniawati. On the probabilistic foundations of probabilistic roadmap planning. *Int. J. Robot. Res.*, 2006, 25(7): 627–643
49. N. Ratliff, M. Zucker, J. A. D. Bagnell, S. Srinivasa. CHOMP: Gradient optimization techniques for efficient motion planning. In: *Proceedings of International Conference on Robotics and Automation*, 2009: 489–494
50. J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, P. Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. In: *Proceedings of Robotics: Science and Systems*, 2013
51. M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, S. Thrun. Anytime dynamic A*: An anytime, replanning algorithm. In: *Proceedings of the International Conference on Automated Planning and Scheduling*, 2005
52. J. Pan, X. Zhang, D. Manocha. Efficient penetration depth approximation using active learning. *ACM Transactions on Graphics*, 2013, 32(6): 191:1–191:12
53. J. Schulman, et al. Motion planning with sequential convex optimization and convex collision checking. *Int. J. Robot. Res.*, 2014, 33(9): 1251–1270
54. J. Pan, C. Lauterbach, D. Manocha. g-Planner: Real-time motion planning and global navigation using GPUs. In: *Proceedings of AAAI Conference on Artificial Intelligence*, 2010: 1245–1251
55. N. M. Amato, L. Dale. Probabilistic roadmap methods are embarrassingly parallel. In: *Proceedings of IEEE International Conference on Robotics and Automation*, 1999: 688–694
56. V. N. Vapnik. *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995
57. S. J. Huang, R. Jin, Z. H. Zhou. Active learning by querying informative and representative examples. In: *Proceedings of Advances in Neural Information Processing Systems*, 2010: 892–900
58. M. Karasuyama, I. Takeuchi. Multiple incremental decremental learning of support vector machines. *IEEE Trans. Neural Netw.*, 2010, 21(7): 1048–1059
59. C. Lauterbach, M. Garland, S. Sengupta, D. Luebke, D. Manocha. Fast BVH construction on GPUs. *Comput. Graph. Forum*, 2009, 28(2): 375–384
60. S. Tzeng, L. Y. Wei. Parallel white noise generation on a GPU via cryptographic hash. In: *Proceedings of the Symposium on Interactive 3D Graphics and Games*, 2008: 79–87
61. J. Pan, C. Lauterbach, D. Manocha. Efficient nearest-neighbor computation for

- GPU-based motion planning. In: *Proceedings of International Conference on Intelligent Robots and Systems*, 2010: 2243–2248
62. J. Pan, D. Manocha. Bi-level locality sensitive hashing for k-nearest neighbor computation. In: *Proceedings of International Conference on Data Engineering*, 2012: 378–389
63. C. Lauterbach, Q. Mo, D. Manocha. gProximity: Hierarchical GPU-based operations for collision and distance queries. *Comput. Graph. Forum*, 2010, 29(2): 419–428