On the computational bottleneck in sampling-based robot motion planning

Michal Kleinbort
Tel Aviv University

ABSTRACT

The complexity of nearest-neighbor search dominates the asymptotic running time of many sampling-based motion-planning algorithms. However, collision detection is often considered to be the computational bottleneck in practice. Examining various asymptotically optimal planning algorithms, we characterize settings, which we call *NN-sensitive*, in which the *practical* computational role of nearest-neighbor search is far from being negligible, i.e., the portion of running time taken up by nearest-neighbor search is comparable to, or sometimes even greater than the portion of time taken up by collision detection. This reinforces and substantiates the claim that motion-planning algorithms could significantly benefit from efficient and possibly specially-tailored nearest-neighbor data structures.

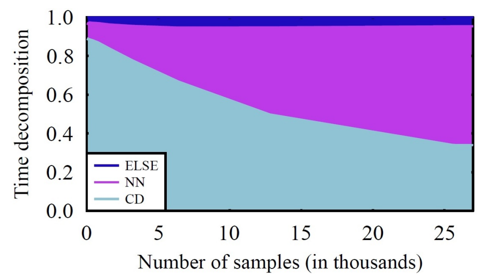

Figure 1: Running-time breakdown of the main primitive operations used in an NN-sensitive algorithm as a function of the number of samples. The portion of time spent on NN (purple) dominates that of CD (light-blue), for more than approximately 15K samples.

1 INTRODUCTION

Given a robot $\mathcal{R}$ moving in a workspace $\mathcal{W}$ cluttered with obstacles, motion-planning (MP) algorithms are used to efficiently plan a path for $\mathcal{R}$, while avoiding collision with the obstacles [1]. Although motivated originally by motion planning for robots, the MP problem has found many more applications including modeling of molecular motions, animating digital actors, designing modern surgical tools, and generating crowd simulations.

Prevalent algorithms abstract $\mathcal{R}$ as a point in a (possibly high-dimensional) space called the *configuration space* (C-space) $\mathcal{X}$ and plan a path (curve) in this space. A point, or a configuration, in $\mathcal{X}$ represents a placement of $\mathcal{R}$ that is either collision-free or not, subdividing $\mathcal{X}$ into the sets $\mathcal{X}_{\text{free}}$ and $\mathcal{X}_{\text{forb}}$, respectively. *Sampling-based* algorithms study the structure of $\mathcal{X}$ by constructing a graph, called a *roadmap*, which approximates the connectivity of $\mathcal{X}_{\text{free}}$. The nodes of the graph are collision-free configurations sampled at random. Two (nearby) nodes are connected by an edge if the straight line segment connecting their configurations is collision-free.

Sampling-based MP algorithms are typically implemented using two primitive operations: *Collision detection* (CD), which is primarily used to determine whether a configuration is collision-free or not, and *Nearest-neighbor* (NN) search, which is used to efficiently return the nearest neighbor (or neighbors) of a given configuration. CD is also used to test if the straight line segment connecting two configurations lies in $\mathcal{X}_{\text{free}}$—a procedure referred to as *local planning* (LP). We consider both CD and LP calls when measuring the time spent on collision-detection operations.

2 CONTRIBUTION

The complexity of NN search dominates the asymptotic running time of many sampling-based MP algorithms. However, the main computational bottleneck in practical settings is typically considered to be LP [1]. We argue that this may not always be the case and describe settings, which we call *NN-sensitive*, where the (computational) role of NN search after *finite* running-time is far from negligible and merits the use of advanced and specially-tailored data structures. NN-sensitivity may be due to (i) planners that *algorithmically* shift the computational weight to NN search; (ii) scenarios in which certain planners perform mostly NN search; or (iii) parameters' values for which certain planners spend the same order of running time on NN and CD.

Specifically, we focus on asymptotically (near) optimal (AO) MP algorithms, that is, algorithms that are guaranteed to converge the optimal-cost solution if a solution exists. In [3] we prove that even for these algorithms, which typically perform more CD queries than their non-AO variants, the complexity of NN search will dominate the asymptotic running time. Moreover, we study the ratio between the overall time spent on NN search and CD after N configurations were sampled. We observe situations where NN takes up to 100% more time than CD in scenarios based on the Open Motion Planning Library (OMPL), as demonstrated in Figure 1, and even more on certain synthetic high-dimensional C-spaces (see [3] for a detailed description and additional experiments).

To substantiate our claims, in [2] we use a specially-tailored NN data structure allowing for efficient NN queries within an NN-sensitive algorithm. Our experiments show a significant speedup (by a factor of three) in the overall running time of the algorithm, as well as an significant improvement in the time to converge to a high-quality solution.

REFERENCES

[1] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. 2005. *Principles of Robot Motion: Theory, Algorithms, and Implementation.* MIT Press.
[2] Michal Kleinbort, Oren Salzman, and Dan Halperin. 2015. Efficient high-quality motion planning by fast all-pairs r-nearest-neighbors. In *ICRA 2015.* 2985–2990.
[3] Michal Kleinbort, Oren Salzman, and Dan Halperin. 2016. Collision detection or nearest-neighbor search? On the computational bottleneck in sampling-based motion planning. In *WAFR 2016.* To appear.