Preemptive Rapidly-Exploring Random Trees

Anonymous CVPR submission

Paper ID 9918

Abstract

Rapidly exploring random trees are a commonly employed method of efficiently searching non-convex and/or high-dimensional spaces, first introduced by Steven M. LaValle in 1998 [3]. Within the context of robot motion planning, RRTs can be used to handle problems with additional constraints, in the form of degrees of freedom, kinematic bounds, and obstacles. In all algorithms intended for real-time use in autonomous robots, computational time is an important consideration, and often planning must be done within a predefined time budget, even if at the cost of solution optimality. In doing so however, one wishes to still provide the most optimal solution discoverable within the time budget; as such, this paper sets out to consider the implementation of a time-budgeted RRT algorithm, taking inspiration from the preemption approach of Nistér's preemptive RANSAC implementation [6].

1. Introduction

1.1. Prior Work

1.1.1 Rapidly-Exploring Random Trees

LaValle's original implementation of the RRT algorithm was created to extend existing progress in randomized approaches to nonholonomic planning problems [3]. It considers path planning as a search in a metric space, X, for a continuous path between an initial state x_i to a final state x_{goal} or goal region X_{goal} within X: in standard problems the state space X corresponds to the configuration space C, in kinodynamic problems this changes to the tangent bundle of the configuration space, detailed in greater depth in LaValle's 2001 follow-on paper with Kuffner [4].

RRTs are constructed by beginning at x_i, and then for each vertex up to a defined count of vertices K, a random state within the configuration space is selected (rejecting illegal samples), and the nearest node to the random configuration is found, whereupon a new node is created a set incremental distance towards the random configuration from the current node, and the two nodes are connected by an edge; the former steps iterate until the total number of nodes allotted is met. This is method can be written in psuedocode as below, taking input arguments xI for graph's root, k for the number of vertices, and deltaT for the incremental distance from a given node towards the new random state. It returns an RRT graph T.

```
GENERATE_RRT(xI, k, deltaT)
    T.init(xI);
    for i=1 to k do
        xRand = RANDOM_STATE();
        xNear = NEAREST_NEIGHBOR(xRand, T);
        u = SELECT_INPUT(xRand, xNear);
        xNew = NEW_STATE(xNear, u, deltaT);
        T.add_vertex(xNew);
        T.add_edge(xNear, xNew, u);
    Return T
```

1.1.2 Preemption schemas, as applied to RANSAC

RANSAC, standing for random sample consensus, is an iterative approach to model parameter estimation from data containing outliers that must be excluded. The primary disadvantage of its iterative design is that there does not exist any upper bound to the computation time, and as such, when limits are externally enforced, results can be far from optimal. Nistér attempts to handle this by employing a preemption scheme in which observations are iteratively scored against the current best hypothesis, as given by his psuedocode description of the algorithm below (where $L_i(h) = \sum_{o=0}^{i} \rho(o, h)$, $\rho(o, h)$ is a scoring function taking an observation and hypothesis and returning a scalar log-likelihood value, and $f(i)$ is a decreasing preemption function that indicates the number of hypotheses kept at a given stage, determined by runtime constraints)[6]:

---

1. Randomly permute the observations.

2. Generate the hypotheses indexed by $h = 1, ..., f(1)$.

3. Compute the scores $L_1(h) = \rho(1, h)$ for $h = 1, ..., f(i)$. Set $i = 2$.

4. Reorder the hypotheses so that the range $h = 1, ..., f(i)$ contains the best $f(i)$ remaining hypotheses according to $L_{i-1}(h)$.

5. If $i > N$ or $f(i) = 1$, quit with the best remaining hypothesis as the preferred one. Otherwise, compute the scores $L_i(h) = \rho(i, h) + L_{i-1}(h)$ for $h = 1, ..., f(i)$, increase i and go to Step 4.

The main appeal of this approach over others considered within the paper is that hypotheses are able to be compared against each other throughout the entire process, rather than simply against some unchanging measure of quality.

2. Algorithm Development

2.1. Overview

The jumping-off point will be LaValle's original RRT algorithm [3][4], and a similar preemption scheme as proposed in the context of RANSAC [6], in which hypotheses are iteratively scored against observations, quitting with the best remaining hypothesis as the preferred one when the time / iteration budget is reached. As RRTs are not *per se* attempting to converge to a hypothesis, but rather simply seeking to fill space in a Monte-Carlo fashion, this concept will need to be modified somewhat. To increase the likelihood of the RRT reaching the goal region within an allotted time, we can seek to expand nodes in a stochastic fashion weighted by an objective function such as distance to the goal region, in a semi-depth-first-search manner. Alternatively, to carry over the element of scoring hypotheses against each other, rather than some absolute measure of quality, we can consider metrics such as path length to any given region within the overall space, as, owing to substructure optimality, if two separate branches reach the same region (as defined by a sufficiently fine n-dimensional grid to avoid artifacting of obstacles) and one has a shorter path length, any subsequent expansions of the two considered nodes will be more optimal by path length when the node considered is belonging to the path with shorter length.

Time permitting, and for easier benchmarking against current state-of-the-art, this scheme would also be integrated with other improvements / variants upon RRTs, that additionally decrease the convergence rate, or improve the solution optimality, such as A*-RRT*[1], informed RRT*[2], and RT-RRT*[5]. The solution will be evaluated by a receiver-operating-characteristics-style charting of optimality (as gauged in relation to the non-preemptive version of the RRT variant the preemption is implemented on, as well as potentially to a ground truth optimality) to the time (or iteration) budget allotted. To do so, a large set of environments and start / goal positions will need to be generated and tested against, for both the unmodified and modified RRT variant(s). Generating such a comparison metric will enable easy evaluation of the algorithm to others, that can be plotted as a point within the curve's space (time to convergence / budgeted time vs. path optimality), as well as an easy way to evaluate the practicality of the algorithm's implementation in hardware that would allow for greater/-lesser time budget allotments: i.e. what is the minimum hardware to achieve some optimality threshold.

2.2. Scoring of Node Hypotheses

Within the preemptive RANSAC algorithm described above...

3. Results

4. Conclusion

4.1. Notes to the editor

I was unfortunately under the impression this was due on the 2nd, not the 1st, due to the date stated in Blackboard, only noticing the day of that it was due through Gradescope immanently. As such this is far rougher of a draft than I would have wished to provide, missing multiple elements and truncated forms of others, and is not representative of the quality of work the final will look like. Ideally this version would have additionally included a motivation subsection, more depth on the two pieces of prior work, inclusion of short synopses of the other RRT improvements, and more background on preemption theory (though much of this may be integrated more into the algorithm development as it will likely fit better when motivated by the need to describe reasoning behind algorithm design decisions). Apologies for the mix-up to those who are reviewing, and best of luck with the remainder of your projects.

Smaller notes: as CMT does not appear to designate / provide anonymized paper IDs, I randomly selected 9918–I was uncertain if it would be needed for the anonymous review or if that would be managed in a secondary way (email lists or similar), and as such included a random one to be safe. Typesetting is not final, and is also a result of the limited time I completed the draft in.

References

[1] M. Brunner, B. Brüggemann, and D. Schulz. Hierarchical rough terrain motion planning using an optimal sampling-based method. *2013 IEEE International Conference on Robotics and Automation*, pages 5539–5544, 2013. 2

[2] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot. Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014*

IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2997–3004, 2014. 2

[3] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Iowa State University, 1998. 1, 2

[4] S. M. LaValle and J. J. K. Jr. Randomized kinodynamic planning. *The International Journal of Robotics Research (IJRR)*, 20(5):378–400, 2001. 1, 2

[5] K. Naderi, J. Rajamäki, and P. Hämäläinen. RT-RRT*: A real-time path planning algorithm based on RRT*. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, MIG '15, page 113–118, New York, NY, USA, 2015. Association for Computing Machinery. 2

[6] D. Nistér. Preemptive RANSAC for live structure and motion estimation. *Machine Vision and Applications*, 16:321–329, 2005. 1, 2