Fast Collision Checking: From Single Robots to Multi-Robot Teams

Joshua Bialkowski, Michael Otte, and Emilio Frazzoli

Collision checking is a critical bottle-neck in robotic motion planning and a key hurtle to enabling more sophisticated robotic systems [3]. Collision checking for a multi-robot team is even more difficult than for a single robot. In [1] we show that collision checking can be significantly reduced for a *single*-robot by using "safety certificates," see Figure 1. **We now extend this result to centralized *multi*-robot teams.**

The configuration space of a multi-robot team is a Cartesian product of the space of each robot (R robots, each planning in D-dimensions, yields a RD-dimensional configuration space). Collision checking vs. the environment can be accomplished piecewise per robot. We evaluate three safety certificate methods for multi-robot teams: *Basic Certificate*, *Partial Certificate*, and *Shared Projection*—see Figure 2.

Figures 3 and 4 depict results from experiments using these methods with RRT [4] for teams of 1-5 robots. Note, RRT* [2] gives similar results (these are omitted here due to space constraints). *Basic Certificate* suffers from a curse of dimensionality that limits its usefulness (Figure 3). Only *Shared Projection* provides significant runtime reductions for all team sizes (Figure 4). As in the single robot version of this work, there is eventually a graph size for which using certificates becomes more expensive than a tradition collision check. This appears to happen more quickly for *Shared Projection*, likely because *Shared Projection* adds R nodes to the secondary kd-tree per sample.

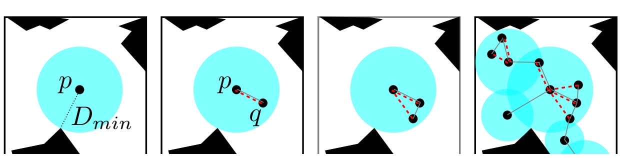
References

[1] J. Bialkowski, S. Karaman, M. Otte, and E. Frazzoli. Efficient collision checking in sampling-based motion planning. In *Proc. International Workshop on the Algorithmic Foundations of Robotics*, 2012.

[2] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *Int. Journal of Robotics Research*, 30(7):846–894, June 2011.

[3] S. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[4] S. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–400, 2001.

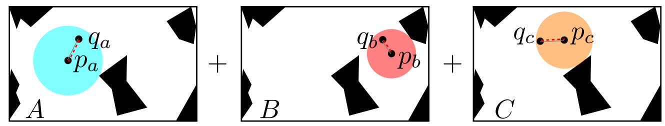


(A) (B) (C) (D)

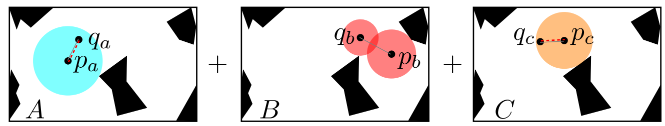
Collision checked nodes p store "safety certificates" (blue) defined by D_{min} the distance to the nearest obstacle (A). Future nodes q within a certificate can forgo collision checking (B). Pointer (red-dotted lines) are maintained to certifying nodes (C). The ratio of collision checks vs. all nodes approaches zero in the limit vs. graph size (D). See [1] for more details.

Fig. 1: Our Single-Robot Collision Certificate Method

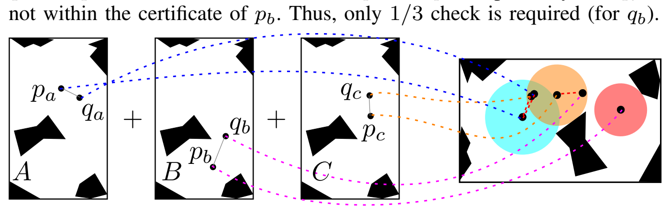
J. Bialkowski, M. Otte, and E. Frazzoli, Massachusetts Institute of Technology, Cambridge, MA. ottemw@mit.edu



Basic Certificate: Certificates are a Cartesian product of balls, one ball per robot. This is exactly [1] applied to a centralized multi-robot team. e.g., q is certified safe by p if the projections q_a, q_b and q_c are in the projected certificates of p_a, p_b and p_c (blue, red, and orange balls), respectively.



Partial Certificate: If a point is *not* certified as safe with respect to a subspace projection, then only a partial collision check is required. e.g, q_a and q_c are within the certificates of p_a and p_c, respectively, but q_b is not within the certificate of p_b. Thus, only $1/3$ check is required (for q_b).



Shared Projection: All robots collision check in the *same* D-dimensional projection (far right). This requires an *extra* kd-tree in the shared projected space, but time complexity only increases from $\mathcal{O}(RD\log(N))$ to $\mathcal{O}(RD\log(N) + D\log(R))$, where $D\log(R)$ is a constant. Pointers from configuration space node projections to their collision-checking projection counterparts are depicted with blue/magenta/orange dotted lines, respectively. Note that p_a certifies q_c and p_c certifies q_b.

Fig. 2: Our New Multi-Robot Collision Certificate Methods

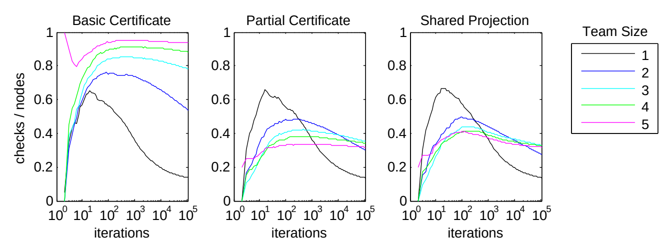


Fig. 3: Proportion of nodes requiring a collision check (mean value over 20 trials), lower values are better.

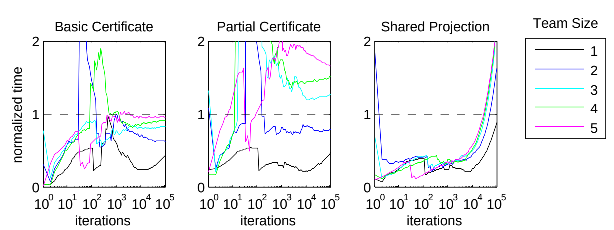


Fig. 4: Relative runtime of certificate methods vs. normal collision checking (mean over 20 trials), points below the dotted line are desired.