RRT path planner with 3DOF local planner

Jade Yang and Elisha Sacks
Computer Science Department
Purdue University
West Lafayette, IN 47907
Email: eps@cs.purdue.edu

***Abstract*— We present a path planning algorithm for a polyhedral robot with six degrees of freedom (6DOF) and a static obstacle. The planner consists of a dual-tree RRT algorithm that uses a novel local planner. A local planner tests if two robot configurations can be connected by a simple path. Ours searches a 3DOF subspace of the robot/obstacle configuration space, whereas prior planners search the line segment that connects the two configurations. Empirical evidence suggests that the benefit of the 3DOF search outweighs the cost. Our planner outperforms prior planners on problems with narrow channels and performs comparably (shorter paths, similar running times) on other problems.**

I. Introduction

This paper presents a path planning algorithm for a polyhedral robot with six degrees of freedom (6DOF) and a static obstacle. Path planning with 6DOF is important for robot navigation and manipulation, design for assembly, virtual prototyping, and computer graphics. Although complete planning algorithms are available, they have proved impractical. The most successful technique to date is probabilistic planning. The drawback is that there is no efficient way to select enough samples to guarantee a specified success rate. Picking an optimal sampling rate is critical when the robot/obstacle configuration space contains narrow channels, as is common in structured environments. Too many samples make the planner inefficient, whereas too few make it unreliable.

We hypothesize that the problem lies in the local planner: the subroutine that tests if two robot configurations can be connected by a simple path. In current planners, the only candidate path is the line segment between the two configurations. Narrow channels cannot accommodate long segments, so the planner must traverse them in many local steps. At each step, many configurations are explored before a reachable one is found, since most directions are blocked.

We have developed a local planner that searches 3DOF subsets of the 6DOF configuration space. Each 3DOF subset consists of parallel translations and perpendicular rotations relative to a fixed plane. Planar motions are good building blocks for general motions. Any Euclidean motion (rotation plus translation) can be expressed as two planar motions. Hence an n-step plan from current planners is expressible as $2n$ planar motions. In return for this factor of two, we gain the ability to plan locally with 3DOF, instead of with line segments. Complete 3DOF path planning has proved superior to probabilistic planning for planar robots [1]. We build on that work to obtain a local planner for polyhedral parts.

The 3DOF local planner finds more paths than prior planners because it searches a larger subset of configuration space. The question is whether the benefit outweighs the cost. We present empirical evidence in the affirmative. A standard probabilistic algorithm (dual-tree RRT) with our local planner far outperforms prior planners on problems with narrow channels and performs comparably (shorter paths, similar running times) on other problems.

II. Prior work

The main approaches to robot path planning are deterministic, heuristic, and probabilistic [2]. The most promising approach is probabilistic planning, since deterministic planning is impractical with 6DOF and heuristic planning is unreliable. Probabilistic algorithms sample the robot/obstacle free space, which is the subset of configuration space where the robot is separated from the obstacle. The roadmap approach [3] generates random configurations, prunes the blocked ones, and links adjacent free ones into a graph. Path planning is performed by linking the initial and goal configurations to the graph then searching for a path between them. The RRT (rapidly exploring random tree) approach [4], [5] builds a roadmap between the start and goal, rather than over the entire free space.

Barraquand et al [6] unify the treatment of probabilistic algorithms and prove them probabilistically complete. Their theory predicts that narrow channels can require excessive numbers of samples, as has proved true. Prior work addresses this problem by varying the sampling distribution and the configuration space metric to favor narrow channels [7]–[12]. Other work replaces random sampling with grid search [13], [14]. The reported results show improved performance in some cases, but narrow channels remain problematic.

Sacks [1] presents a fast, complete planner for a planar robot with a static obstacle, as part of a heuristic planner for articulated robots. The planner performs a best-first search on an exact representation of configuration space. Extensive experimentation shows that the complete planner is much better than probabilistic planners. We adapt this algorithm to search 3DOF subsets of 6DOF configuration spaces.

III. Path planning algorithm

Our planner consists of a dual-tree RRT algorithm [4] with our local planner. We chose the RRT approach because it is the most popular. We chose the dual-tree variant because it

is simple, well tested, and performs well. Any probabilistic planner should benefit similarly from using our local planner.

A. RRT algorithm

The RRT algorithm generates a tree whose nodes are robot configurations and whose links are free space paths. The tree starts as a single node and grows as follows (Fig. 1). Pick a free configuration, q, according to a given distribution. Find the node whose configuration, r, is closest to q in a given metric. Compute s by taking a step of length ϵ along the line segment rq. If the line segment rs lies in free space, add s to the tree and link it to r by this path.

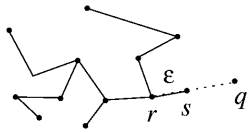


Fig. 1. RRT construction.

The dual-tree algorithm generates two RRT trees starting from the start and goal configurations. It alternately grows each tree and tries to connect the new node to the closest node in the other tree. The connection succeeds when the line segment between the two nodes lies in free space. We modify the algorithm to connect r to q using our local planner, instead of connecting r to s via a line segment. We also use our local planner to connect the two trees.

B. Local planner

Fig. 2 shows the local planning algorithm. The inputs are the robot and obstacle geometry, and the start and goal configurations. The robot is a polyhedron and the obstacle is a set of polyhedrons. The algorithm selects a robot motion plane, maps the geometry to the motion plane, and moves the mapped robot as close as possible to the goal. The cycle repeats until the robot reaches the goal or until no progress is made. We discuss the algorithm after presenting an example.

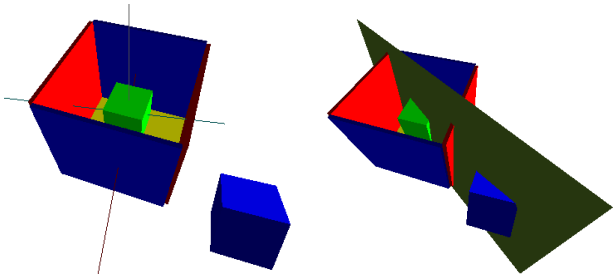
Input: robot, obstacle, start, goal.
1. Set current configuration to start configuration.
2. Select robot motion plane.
3. Map robot and obstacle to plane.
4. Find best path in plane.
5. If no progress, report failure.
6. If goal reached, return path.
7. Update current configuration and go to step 2.

Output: path or failure.

Fig. 2. Local path planning algorithm.

C. Example

Fig. 3a shows a cubical robot with a tank-shaped obstacle. The initial configuration is inside the tank and the goal configuration is outside. The planner maps the geometry to the plane shown in Fig. 3b and finds a planar motion to the goal position (Fig. 4). Fig. 5a shows the final configuration of this path, which becomes the new current configuration. The planner maps the geometry to the plane shown in Fig. 5b and finds a planar rotation to the goal orientation.



(a) (b)

Fig. 3. First iteration: (a) start and goal configurations; (b) motion plane.

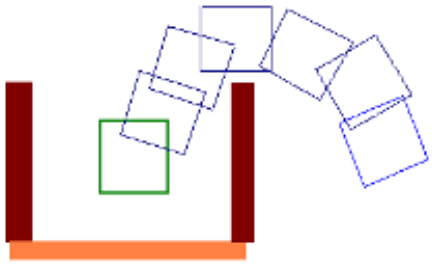
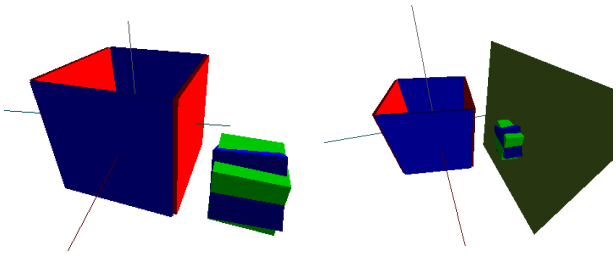


Fig. 4. Robot path in first motion plane.



(a) (b)

Fig. 5. Second iteration: (a) start and goal configurations; (b) motion plane.

D. Motion plane selection

The motion plane alternates between two choices, called the translation plane and the rotation plane, that are determined by the current configuration, c, and the goal configuration, g. Both planes pass through the current position for specificity, since orthogonal translation of the motion plane does not alter the associated three degrees of freedom. Write gc^{-1} as rotation by angle θ around axis n through the robot reference point followed by translation by t. The rotation plane normal is n. The translation plane normal is $n-(t\cdot n)t$, which is orthogonal to t and as close as possible to n in the least squares sense.

Without an obstacle, the robot can move from c to g via rotation by θ around n in the rotation plane followed

by translation by t in the translation plane. Our heuristic assumption is that the planner can make good progress in these planes even with an obstacle. It can rotate towards the goal orientation in the rotation plane, while translating around obstacles. It can translate towards the goal position in the translation plane, while translating and rotating around obstacles. When the robot gets close enough to the goal, the desired rotation and translation lie in free space and are achieved by a final iteration of the local planner.

E. Geometry mapping

Path planning within a motion plane employs configuration space construction. Although construction is feasible for polyhedrons with 3DOF, the computation is much faster for polygons. We map the polyhedral problem to a planar problem in two ways, both of which are incomplete. Our heuristic assumption is that this procedure works better than a complete polyhedral planner.

First, we project the robot and the obstacle onto the motion plane. The obstacle is clipped before being projected. Pick a coordinate system in which projection occurs along the z axis. The robot in configuration c is bounded by planes $z = z_1$ and $z = z_2$. The portion of the obstacle outside these planes is discarded because the robot cannot reach it via motion in the xy plane. If the planner planar finds a path for the projected geometry, it is free for the true geometry and we are done. Fig. 6 shows a two-dimensional example in which the horizontal axis represents the motion plane.

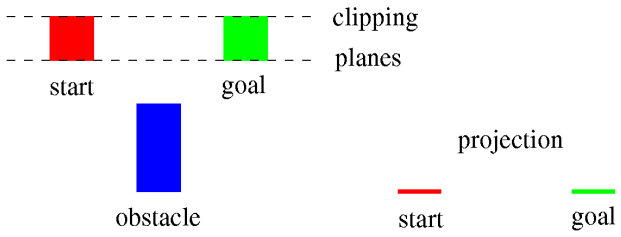


Fig. 6. Projection succeeds.

When projection yields no path, we slice the geometry with a plane through the robot centroid and parallel to the motion plane. If no path is found, we fail. If a path is found, we lift it to the true geometry. The path consists of Euclidean motions. We sweep the robot along each motion and test for collisions with the obstacle. If the lift succeeds, we are done. Fig. 7 shows an example where projection fails, but slicing succeeds.

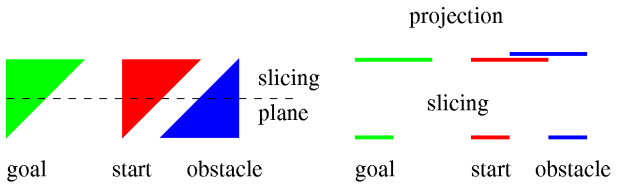


Fig. 7. Projection fails and slicing succeeds.

F. Planar path planning

The planar planner searches the 3DOF configuration space of the mapped robot and obstacle for a path from c to g. Fig. 8 show a path for a typical robot/tank motion plane. The planner tests if c and g lie in the same connected component of free space. If not, it reports that there is no path. If so, it finds a path via A* search. The search states are regions in free space and patches in contact space (the subset of configuration space where the robot touches the obstacle without overlap). The quality metric for the motion "rotation by θ and translation by t" is $||t||+|\theta r|$ with r the robot radius. The search terminates at the closest configuration to g in this metric. The path consists of line segments in free space and of curves in contact space. The algorithm is described by Sacks [1].

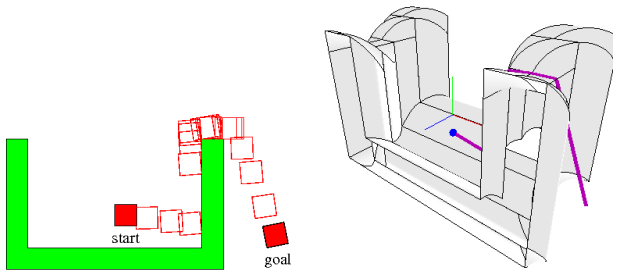


Fig. 8. Planar planning: (a) path snapshots; (b) configuration space path.

IV. Results

We present experimental results that compare our planner with a standard dual-tree RRT planner. The robot is a 16x16x16 cube. Some obstacles are random and others are handcrafted.

A. Random cubes

The first obstacle is comprised of 20 random cubes. The cube sizes vary from 1–40 and their configurations are random in a bounding cube of size 220. The robot start and goal configurations are random. Fig. 9 shows a typical test. The start configuration is red, the goal is blue, and intermediate steps are gray.

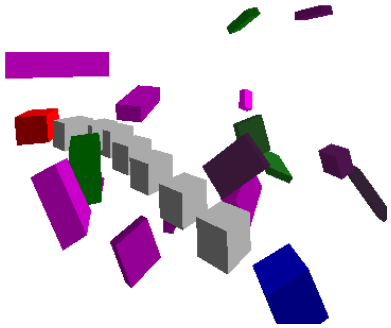
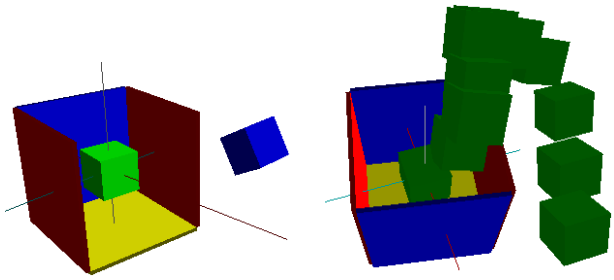


Fig. 9. Random test case.

B. Tank

Fig. 10 shows the second test. The obstacle is a tank of size 80. One wall of the tank is removed for the purpose of illustration. The robot starts at the center of the tank aligned with its walls. The goal configuration is selected randomly outside the tank.

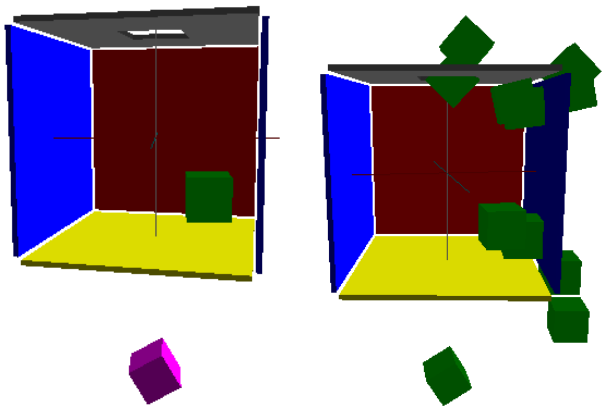


(a) (b)

Fig. 10. Tank test case: (a) start and goal; (b) path.

C. Box

Fig. 11 shows the third test. The obstacle is a box of size 80 with a window of size 20 on its top. One wall of the box is removed for the purpose of illustration. The robot starts at a random configuration inside the box; the goal is a random configuration outside.



(a) (b)

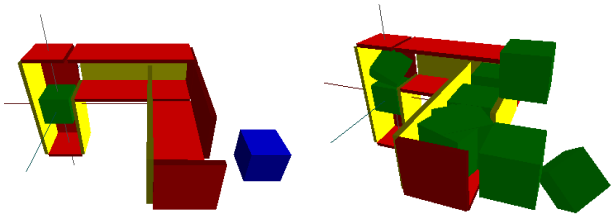
Fig. 11. Box test case: (a) start and goal; (b) path.

D. Tunnel

Fig. 12 shows the fourth test. The obstacle is a tunnel comprised of three segments of size 60x20x20 that are parallel to the three coordinate axes. The mouth of the first segment is sealed and the mouth of the third segment is open. Several sides are removed for the purpose of illustration. The robot starts in the first segment. The goal is a random configuration outside the tunnel.

E. Maze

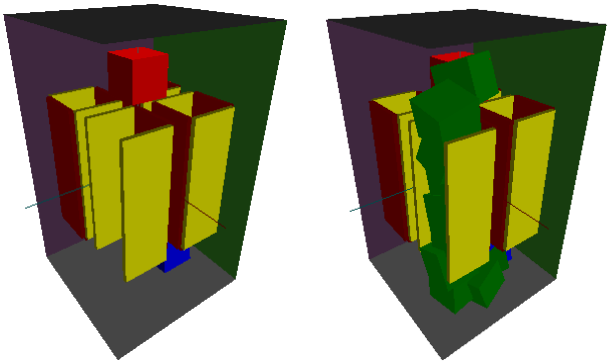
Fig. 13 shows the fifth test. The obstacle is a maze comprised of four 20x20x60 tunnels in a 70x70x120 room.



(a) (b)

Fig. 12. Tunnel test case: (a) start and goal; (b) path.

Adjacent tunnels are 10 units apart. The robot starts on one side of the maze. The goal is a random configuration on the other side. All four tunnels are open on the start side, but only one leads to the goal side.



(a) (b)

Fig. 13. Maze test case: (a) start and goal; (b) path.

F. Performance

Table I compares the performance of the two planners. The "old" columns refer to the standard dual-tree RRT algorithm; the "new" columns refer to our planner. The "success" columns list the number of paths found on 100 trials. The other columns are average values over these trials. The "nodes" entry is the number of attempts to link two RRT nodes in order to grow one tree or to connect the two trees. The number of motion planes that our planner examines is listed in parenthesis. The "time" entry is the running time in seconds on a 2GHz Pentium 4.

TABLE I

PLANNER PERFORMANCE.

	success		nodes		time	
test	old	new	old	new	old	new
20 cubes	100	100	97	4 (4)	4	8
tank	100	100	241	4 (7)	0.4	0.9
box	65	100	12112	11 (18)	324	9
tunnel	0	100	20000	63 (128)	618	19
maze	0	100	20000	29 (63)	951	205

The sample configurations (q in Fig. 1) are selected uniformly in the bounding box. For the standard algorithm, the step size (ϵ in Fig. 1) is 10 units of translation and 0.5 radians of rotation. This was the best step we could find

based on trial and error with values ranging from 10% to 400% of it. We terminated the RRT tests after 20,000 nodes. Changing the cutoff to 50,000 nodes increases the running time proportionally without increasing the success rate. The only exception is that the RRT can sometimes solve the maze in 2 CPU hours with the 10% step and 50,000 nodes.

All five tests show that our planner performs far fewer local planning steps than the standard planner. The final three tests show that our planner traverses narrow channels in a moderate number of steps, whereas the standard planner fails after many steps. The first two tests show that our planner is half as fast as the standard planner on problems without channels. We discuss this issue below.

V. Conclusion

We have presented a 6DOF RRT path planner with a 3DOF local planner. The local planner heuristically formulates a sequence of planar path planning problems, solves them with a complete configuration space search algorithm, and combines the results to obtain 6DOF paths. The overall planner is a hybrid of probabilistic, heuristic, and deterministic methods. The motivation for this work is narrow channels. Our test results show that prior planners perform poorly on narrow channels, whereas our planner performs well.

The first direction for future research is to see how our planner works on complex robot/obstacle geometry. The computational bottleneck is configuration space construction, which is polynomial in the geometric complexity. We expect that the number of motion planes is linear in the number of configuration space channels. We also expect that the number of channels is independent of the geometric complexity, since it is a topological property. A preliminary experiment supports this analysis. We increased the number of random obstacles from 20 to 100 in the first test. The number of motion planes stayed the same, while the time per plane increased by a factor of 10.

We will explore ways to limit the configuration space construction time. One option is to explore only a portion of the configuration space per motion plane. We might restrict the planner to an ϵ neighborhood of c, just as the original RRT planner takes an ϵ step from r to s. A second option is to use multi-resolution polyhedral models, as has proved effective in collision detection. A third option is to search 2DOF subsets of configuration space before (or instead of) 3DOF subsets. Prior work [1] shows that configuration space construction for planar parts with 2DOF is fast even for complex geometry. It remains to see if the number of motion planes is manageable.

Another research direction is to derive an efficient algorithm for selecting enough motion planes to guarantee a specified planning success rate. We conjecture that the success rate can be estimated by sampling a moderate number of motion planes, constructing their free spaces, and calculating certain metric properties.

Acknowledgments

This research benefited from discussions with Jim Cremer, Bruce Donald, Ananth Grama, Leo Joskowicz, Jean-Claude Latombe, Jean Paul Laumond, Matt Mason, and Brian Mirtich. It was supported by NSF grants CCR-0306214 and IIS-0082339.

References

[1] E. Sacks, "Path planning for planar articulated robots using configuration spaces and compliant motion," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, 2003.

[2] J.-C. Latombe, "Motion planning: A journey of robots, molecules, digital actors, and other artifacts," *International Journal of Robotics Research*, vol. 18, no. 11, pp. 1119–1128, 1999.

[3] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, 1996.

[4] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *IEEE International Conference on Robotics and Automation*, 2000.

[5] E. Mazer, J. M. Ahuactzin, and P. Bessière, "The ariadne's clew algorithm," *Journal of Artificial Intelligence Research*, vol. 9, pp. 295–316, 1998.

[6] J. Barraquand, L. Kavraki, J.-C. Latombe, T.-Y. Li, and P. Raghavan, "A random sampling scheme for path planning," *International Journal of Robotics Research*, vol. 16, no. 6, pp. 759–774, 1997.

[7] N. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo, "OBPRM: an obstacle-based PRM for 3d workspaces," in *Robotics: the algorithmic perspective*, P. Agarwal, L. E. Kavraki, and M. T. Mason, Eds. Natick, MA: A.K. Peters, 1998.

[8] D. Hsu, L. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin, "On finding narrow passages with probabilistic roadmap planners," in *Robotics: the algorithmic perspective*, P. Agarwal, L. E. Kavraki, and M. T. Mason, Eds. Natick, MA: A.K. Peters, 1998.

[9] T. Siméon, J. P. Laumond, and C. Nissoux, "Visibility based probabilistic roadmaps for motion planning," *Advanced Robotics Journal*, vol. 14, no. 6, 2000.

[10] P. Cheng and S. M. LaValle, "Reducing metric sensitivity in randomized trajectory design," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 43–48.

[11] C. Urmson and R. Simmons, "Approaches for heuristically biasing rrt growth," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[12] J. J. Kuffner, "Effective sampling and distance metrics for 3d rigid body path planning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004.

[13] S. R. Lindemann and S. M. LaValle, "Current issues in sampling-based motion planning," in *Eighth International Symposium on Robotics Research*, P. Dario and R. Chatila, Eds. Springer-Verlag, 2004.

[14] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *International Journal of Robotics Research*, vol. 24, 2004.