Hierarchical Recursive Search across 3D-Scene Graphs Utilizing Multi-Resolution Bernstein Curves

Rohan Bosworth

***Abstract*— 3D Dynamic Scene Graphs (DSG) have emerged as a powerful new method of representing an environment in a hierarchical fashion. A DSG is a layered representation where nodes represent physical entities and edges represent relations between entities. Recent strides have been made in real time localization and mapping of a DSG, however, little research has been conducted on planning across DSGs. This paper proposes a recursive planning methodology for DSGs. Planning is conducted at the coarsest resolution and the results of the course resolution search are improved by results determined recursively from the finest resolution. This paper also proposes a fine resolution planner that utilizes the derivation of Bezier Curves and multi-resolution format to reduce time required to perform motion planning.**

I. Introduction

In this paper, we propose a methodology for motion planning across a 3D Dynamic Scene Graph (DSG) that can be performed in real time. DSGs [14] have been recently proposed as a novel method for environmental representation that allows for a hierarchical representation. DSGs have several layers representing different layers of abstraction and with edges representing relations in the environment. Hydra [5] proposes methods of optimizing 3D Scene graphs using embedded deformation graphs that allow for concurrent improvement of all levels of a scene graph. Hierarchical motion planning algorithms [16] utilizing multiple resolutions that interact to improve search speed and results have been recently developed for a gridmap environment. Given the discretized environment may not represent all obstacles, a feasibility check using Bayes Filtering is proposed to be performed after planning to ensure the path is effective.

In this paper, we also propose a continuous motion planning methodology using Bezier curves. Yin et al. [18] proposes a method of imposing quintic polynomials onto a set of pre-determined waypoints following the conclusion of a search. Zhou et al. [19] proposes a search-based planning method where a Bezier curve is imposed onto the points following the search. Gao et al. [4] proposes optimization for time using a fast-marching method followed by a curve imposition. Chen et al. [6] proposes the use of a safe corridor where rectangular hulls (safe corridor) are constructed around search based planning results. Preiss et al. [13] demonstrates that safe corridors can be combined with the Downwash method to determine trajectories for several robots. Liu et al. [10] proposes incorporating smooth motion into the planning stage, eliminating the need for the imposition of curved trajectories after the planning is complete. Liu et. al. proposes smooth trajectory generation by developing the action space

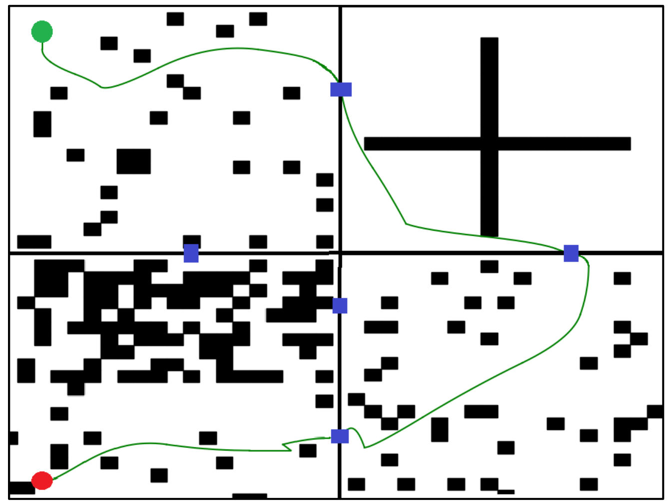

Figure 1 - Completed DSG trajectory generation in cluttered environment (blue rectangles represent doors)

from changes in acceleration or jerk instead of position. Following the path generation, the resulting curve is smoother by minimizing the cumulative jerk and eliminating cusps in the acceleration, for a jerk-controlled system.

II. Contribution 1 -Muti-Resolution DSG Search

We propose a planning methodology for DSGs that can be combined with real time localization and mapping methods and, if provided a map, guarantees optimality.

A. Action Space

When constructing the action space from a DSG, the edges between nodes in the DSG become the nodes in the state space between which the robot can travel at a coarse resolution. The reasoning for this design is that two nodes of a DSG could have multiple connections so there could be multiple costs for traveling between two nodes. Thus, the state space is the set of all edges in a DSG while the action space is the set of all edges that can be drawn to connect the edges in a DSG. For example, the set of all doors would be the state space and the set of all transitions between doors is the action space. Thus, a node in the action space is constructed as a tuple *(a, b)* where *a* represents a starting or current edge of the DSG the robot is on and *b* represents a concluding edge. For example, an open list is initialized which is a Priority Queue that contains the set of all nodes in the coarse resolution action space that can be connected to prior expanded action space nodes. When the open list is initialized, a set of tuples (S, b_i) are added to the where S is the starting node in the state space and b_i is the set of all doors adjacent to the start state in the state space. In the DSG representation, b_i would be a set of edges adjacent to the starting room whereas *S* would be an edge constructed within the state space to represent the specific starting position. *S* is treated as a door in the middle of a room node of a DSG.

B. Heuristic Functions

Depending on the DSG structure a heuristic could be constructed. If information about global room position is provided Euclidian distance could be utilized. However, since that information is not always provided or may not guide to the goal node, the proposed method does not use a heuristic. While this usually may lead to a slower search, there are much fewer nodes at the higher levels of a DSG.

C. Recursive Search

After a node is removed from the open list, it is expanded at the finest resolution of the DSG. For example, if the first node removed from the open list is *(a, b_i)*, the room with mesh *X*, which contains both a and b_i is selected from the DSG. A fine resolution search is performed where a is the starting point, b_i is the goal point, and X is the environment. The proposed method utilizes a Multi-Heuristic, Multi-Resolution, Bezier Curve Search. Once the low level search is concluded, the true cost of the node *(a, b)* is known, g_i. Thus, as all of the successor nodes of *(a, b)* are added to the open list, *f* is defined as the sum of the prior cost to reach node *(a, b)* and g_i. By expanding the open list in this manner, exploration of nodes that are not promising is discouraged. Moreover, the use of real cost values guarantees optimality as if the proposed method is allowed to run until conclusion, all cost values will be determined from the fine resolution planner.

III. Contribution 2 -Hierarchical Bezier Search Planner

A. Motion Model

Given a set of predetermined acceleration constants for the action space U, the position and velocity can be solved for given the quadratic Bezier Curve. *U* is a predetermined set of acceleration constants that correspond to the possible acceleration vectors for the system. This is similar to traditional A* where *U* would be a set of possible movements or actions the robot could take. Since the first point of the Bezier Curve, P_1, must be equal to the starting position of the robot, P_0, $P_1 = P_0$. The remaining control points can then be solved for given constant velocity.

Algorithm 1 DSG Search

```
function runSearch()
    InitEnv              ▷ 2D Array with ID's for free, occupied, and semantic
    currentNode ← (startNode, startNode)   ▷ Start node is stationary so
transition represented with same node listed twice
    openList ← PriotityQueue
    closedList ← [ ]
    while (open is not empty) do
        neighbors ← (currNode[1], adjacentNodes)        ▷ Every state is a
transition between two physical poses
        openList ← (getCost(currentNode), neighbors)  ▷ Current cost of
a node is stored with that node - similar to traditional A* the current cost
is determined as the sum of all prior costs
        newNode ← popNode(openList)
        g ← runFineResolutionSearch(newNode)        ▷ Fine resolution
search can vary but this method proposes Hierarchical Bezier Search
        closedList ← (g + getCost(currentNode), newNode)
        if newNode[1] = goalDoor then
            ExitLoop
        end if
    end while
end function
```

$$P_2 = \frac{V_0 + 2P_1}{2} \quad (1)$$

$$P_3 = \frac{U - 2P_1 + 4P_2}{2} \quad (2)$$

B. Obstacle Detection

A Bezier Curve is contained by the convex hull of its control points. Rather than sampling points along the curve to check for collisions, the points inside of the convex hull of the control points can be checked. Given three control points, a rectangular hull can be constructed. Then, triangle rasterization can be performed on all points in the rectangular hull to determine which fall inside the triangular hull of the control points. Given the Bezier Curve also falls inside the triangular hull, all points inside the triangular hull can be checked for obstacles, allowing for rapid determination regarding whether the Bezier curve intersects any obstacles.

C. Hierarchical Search

Given state space s, a constant x and a set $N = [1, 2, \dots n]$, s is sampled every x^N units to create a smaller, scaled set of state spaces, *X*. The motion model determined above is then applied to every map in *X* in a round robin fashion. Different resolutions are also paired with different heuristics with the finest resolution utilizing an admissible heuristic. As such, heuristics corresponding to power use can be utilized. By scaling the state space, the search speed can be enhanced without compromising the accuracy of the search.

IV. Experimental Verification

To verify the efficacy of the proposed method, a matched pair experiment was run. Random DSG environments were generated with four rooms, random placement of obstacles, random door locations, and random start and goal locations. Both the proposed method and a traditional A* method were then tasked with solving the environment and their respective planning times were recorded. The traditional A* method is required to reconstruct a gridmap. When these planning times are compared in a matched-pair experiment, the hierarchical method is over 12% faster than the traditional method.

Moreover, in observing Figure 1, it can be noted that all rooms are stored separately due to the DSG structure. Thus every coarse resolution step is a transition between door nodes. For example, the first node expanded on Figure 1 was a transition between the start node and a door in the bottom-left room. This methodology allowed the hierarchical planner to not waste time exploring the cluttered environment between the start and goal node.

V. Conclusion

The method proposed in this paper demonstrates an entry into planning across DSGs as well as kinematically feasible trajectory generation for mobile robots. Further exploration regarding real time recursive planning and exploration simultaneous to hierarchical planning in real world environments could continue to show the promise of DSGs.

VI. Acknowledgements

I would like to thank Arash Asgharivaskasi, Shusen Lin, and Dr. Nikolay Atanasov for their contributions.

REFERENCES

[1] Aine, S., Swaminathan, S., Narayanan, V., Hwang, V., & Likhachev, M. (2016). Multi-heuristic A. The International Journal of Robotics Research, 35(1–3), 224–243. doi:10.1177/0278364915594029

[2] Chen, J., Su, K., & Shen, S. (2015, December). Real-time safe trajectory generation for quadrotor flight in cluttered environments. 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO). Presented at the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai. doi:10.1109/robio.2015.7419013

[3] Costanzi, R., Fanelli, F., Meli, E., Ridolfi, A., & Allotta, B. (2016). Generic path planning algorithm for mobile robots based on Bézier curves. IFAC-PapersOnLine, 49(15), 145–150. doi:10.1016/j.ifacol.2016.07.723

[4] Gao, F., Wu, W., Lin, Y., & Shen, S. (2018, May). Online safe trajectory generation for quadrotors using fast marching method and Bernstein basis polynomial. 2018 IEEE International Conference on Robotics and Automation (ICRA). Presented at the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD. doi:10.1109/icra.2018.8462878

[5] Hughes, N., Chang, Y., & Carlone, L. (2022, June 27). Hydra: A real-time spatial perception system for 3D scene graph construction and optimization. Robotics: Science and Systems XVIII. Presented at the Robotics: Science and Systems 2022. doi:10.15607/rss.2022.xviii.050

[6] Chen, J., Liu, T., & Shen, S. (2016, May). Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments. 2016 IEEE International Conference on Robotics and Automation (ICRA). Presented at the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden. doi:10.1109/icra.2016.7487283

[7] Klancar, G., Seder, M., Blazic, S., Skrjanc, I., & Petrovic, I. (2021). Drivable path planning using hybrid search algorithm based on E* and Bernstein–Bézier motion primitives. IEEE Transactions on Systems, Man, and Cybernetics. Systems, 51(8), 4868–4882. doi:10.1109/tsmc.2019.2945110

[8] Li, J., & Yang, C. (2020, October 13). AUV path planning based on improved RRT and bezier curve optimization. 2020 IEEE International Conference on Mechatronics and Automation (ICMA). Presented at the 2020 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China. doi:10.1109/icma49215.2020.9233842

[9] Li, X., Gao, X., Zhang, W., & Hao, L. (2022). Smooth and collision-free trajectory generation in cluttered environments using cubic B-spline form. Mechanism and Machine Theory, 169(104606), 104606. doi:10.1016/j.mechmachtheory.2021.104606

[10] Liu, S., Atanasov, N., Mohta, K., & Kumar, V. (2017, September). Search-based motion planning for quadrotors using linear quadratic minimum time control. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Presented at the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC. doi:10.1109/iros.2017.8206119

[11] Liu, Y., Wang, Y., Wang, X., & Tao, C. (2022, November 24). A model for safe lane changing of connected vehicles based on quintic polynomial Trajectory planning. 2022 7th International Conference on Intelligent Informatics and Biomedical Science (ICIIBMS). Presented at the 2022 7th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Nara, Japan. doi:10.1109/iciibms55689.2022.9971621

[12] Lu, X., Wan, J., Zhong, Y., & Wang, J. (2022, April 22). Dual redundant UAV path planning and mission analysis based on dubins curves. 2022 3rd International Conference on Geology, Mapping and Remote Sensing (ICGMRS). Presented at the 2022 3rd International Conference on Geology, Mapping and Remote Sensing (ICGMRS), Zhoushan, China. doi:10.1109/icgmrs55602.2022.9849361

[13] Preiss, J. A., Honig, W., Ayanian, N., & Sukhatme, G. S. (2017, September). Downwash-aware trajectory planning for large quadrotor teams. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Presented at the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC. doi:10.1109/iros.2017.8202165

[14] Rosinol, A., Gupta, A., Abate, M., Shi, J., & Carlone, L. (2020). 3D Dynamic Scene Graphs: Actionable Spatial Perception with Places, Objects, and Humans. Robotics: Science and Systems (RSS). https://doi.org/10.15607/RSS.2020.XVI.079

[15] Saxena, D. M., Kusnur, T., & Likhachev, M. (2022, May 23). AMRA*: Anytime Multi-Resolution Multi-Heuristic A. 2022 International Conference on Robotics and Automation (ICRA). Presented at the 2022 IEEE International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA. doi:10.1109/icra46639.2022.9812359

[16] Stadler, Martina, Liu, Katherine and Roy, Nicholas. 2021. "Online High-Level Model Estimation for Efficient Hierarchical Robot Navigation." 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[17] Yang, R., Zhang, M., Hansen, N., Xu, H., & Wang, X. (2021). Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. doi:10.48550/ARXIV.2107.03996

[18] Yin, J., Yang, G., Zhao, F., & Qiu, H. (2015, August). Motion planning implemented in ROS for Omni-directional Wheeled Mobile Robot. 2015 IEEE International Conference on Information and Automation. Presented at the 2015 IEEE International Conference on Information and Automation (ICIA), Lijiang, China. doi:10.1109/icinfa.2015.7279741

[19] Zhou, F. & Song, Baoye & Tian, Guohui. (2011). Bézier curve based smooth path planning for mobile robot. Journal of Information and Computational Science. 8. 2441-2450.