NORTHEASTERN

A 3D Reactive Navigation Algorithm for Mobile Robots by Using Tentacle-Based Sampling

Neşet Ünver Akmandor and Taşkın Padır

The Fourth IEEE International Conference on Robotic Computing

RIVeR

Institute for Experiential Robotics
at Northeastern University

robotics
collaborative

Motivation

Active research problems in autonomous motion and path planning in 3D:

- Localization
- Mapping
- Safe path planning/navigation

Challenges:

- Unknown map
- Dynamic environment

Reference: Google Image

Problem Definition

Task:

- Find a navigable path from a start to a goal position
- Ensure multiple objectives such as; closest proximity to the goal, collision-free path and minimum navigation time.

Assumptions:

- No prior global map info
- Global robot and goal positions are known

State-of-art

Path and Motion Planning

- Optimization based [7]
  - Stuck at local minima
- Compound
  - Sampling + Optimization [1]
  - Graph Search + Optimization [5,6,9]
  - Required replanning →high processing power

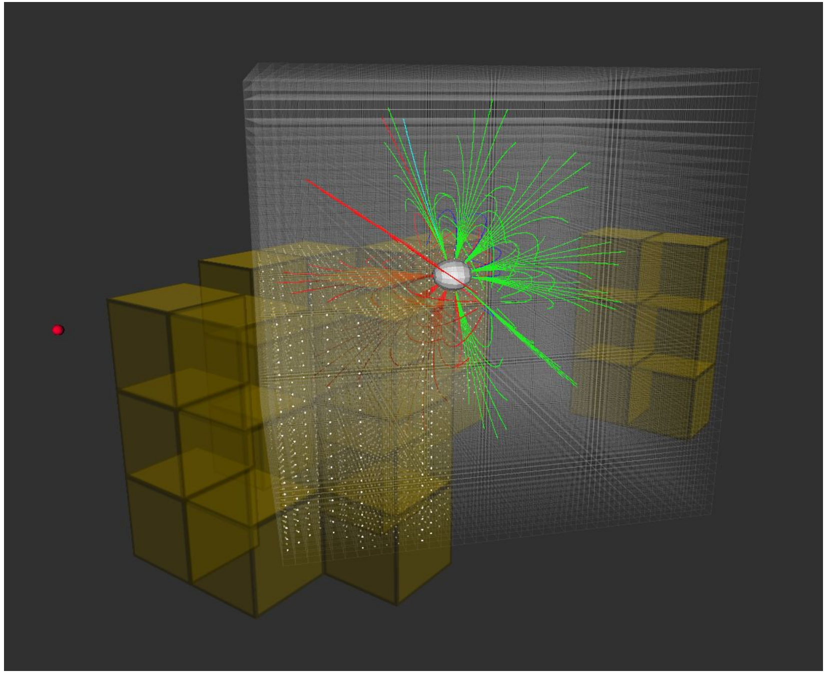
Reactive Navigation

- Graph Search + Reactive obstacle avoidance [3]
- Vision based reactive control [4]
- Tentacle based algorithms [12-21]
  - Fast evaluation of possible paths, but 2D...

3D Reactive Navigation Framework

Robot-Centered 3D Grid

- Formed by user defined (dimension and number) cubic voxels
- Mapping point cloud data (probabilistic occupancy value) to its respective voxel
  - Cartesian coordinate ↔Linearized grid index
- No global map but keeping FIFO history of point cloud



3D Reactive Navigation Framework

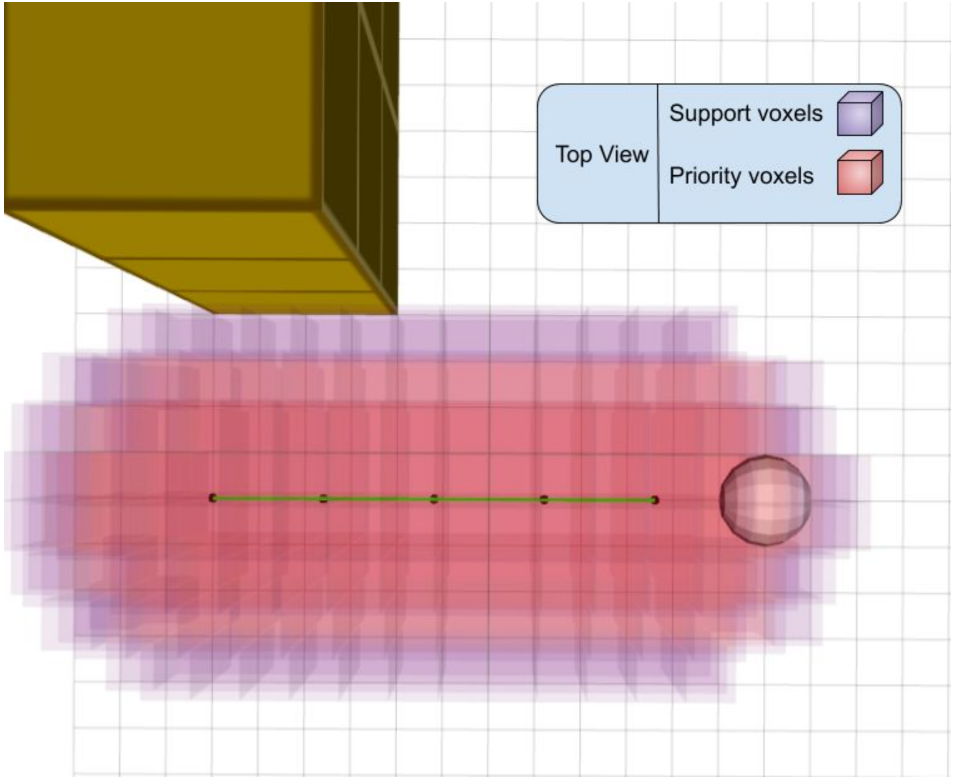
Tentacles

- Pre-calculated paths formed by sampling points
- Fixed to robot’s frame starting from the center of 3D grid
- Not necessary to generate tentacles by considering dynamics or kinematics
- Considering feasibility of the selected path by motion execution block
- Sensing the environment (collision checking)

3D Reactive Navigation Framework

Support and Priority Voxels

- Classified based on the distance
  - [sampling point - voxel center]
- Each tentacle has its own set
- Higher collision risk →higher weight



3D Reactive Navigation Framework

Tentacle Evaluation

Every cycle, each tentacle is evaluated by 5 heuristic metrics (normalized):

1. **Navigability** (navigable, non-navigable, temporarily-navigable)
2. **Clearance** (proximity of obstacle on the tentacle)
3. **Nearby clutter** (based on Priority and Support voxel weights)
4. **Goal closeness**
5. **Smoothness** (based on previously selected tentacle)

3D Reactive Navigation Framework

Tentacle Selection and Execution

- No calculated cost for non-navigable tentacles
- Cost function = weighted sum of four heuristic functions (2, 3, 4, 5)
- Selected tentacle = min Cost
- Not directly send first sampling point to motion controller
    - Interpolate by considering kinematic constraints (lateral, angular velocities, etc.)

3D Reactive Navigation Framework

Parameters of the algorithm

Adjusted based on the robot model, sensor specifications and the navigation task.

- Robot parameters: Volumetric info and kinematic constraints
- Offline parameters: General form of tentacles and robot centered grid
- Online parameters: Navigation policy by heuristic function weights (greediness towards the goal, timidness while avoiding obstacles, etc.)

Results

Average computation time statistics

- Initialization and main iteration steps
- Parameters: voxel dimension and number of tentacles
- Result: Capable of running 10-60 Hz

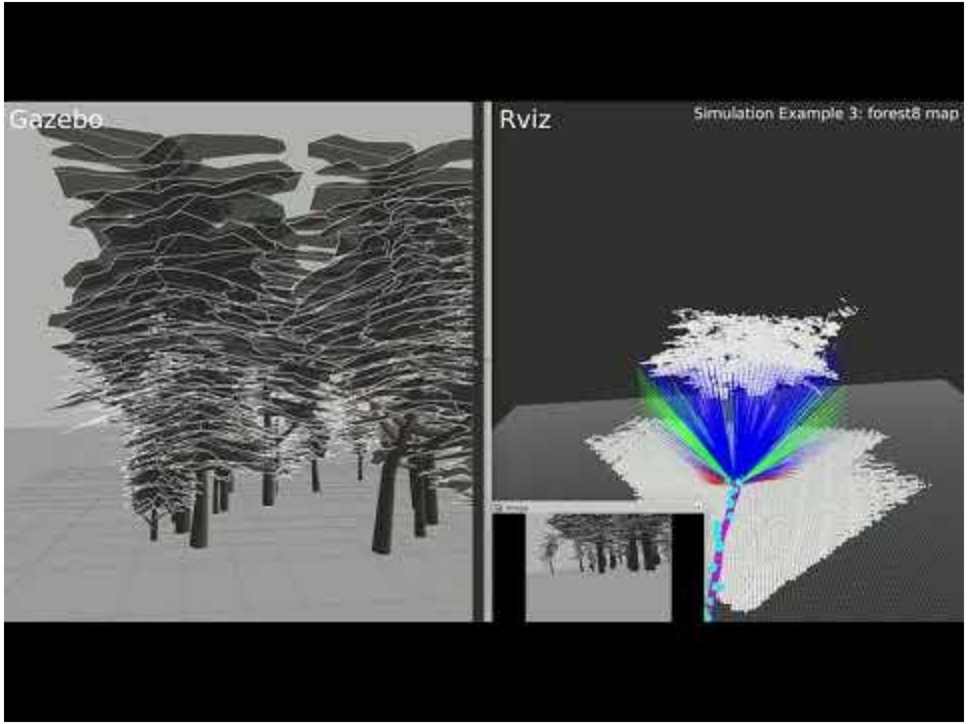
Benchmarks

- [2017, Usenko et al.] within the 10 maps (cylinders map + 9 forest maps)
- Robustness: all configurations are run 10 times for each map without changing any parameter
- Same offline params, empirically tuned online params
- Result: No fail, higher success rate and faster navigation

Summary

- Reactive navigation algorithm for 3D environments
  - No required prior global map
  - Tentacles: pre-determined group of points, to sample the space around
  - Robot-centered grid: keep the latest occupancy information.
  - Tentacle evaluation: 5 heuristic functions
  - Classify offline and online parameters to enhance the reactive navigation performance
  - Outperforms the state-of-art method in terms of success rate and navigation duration
- Open-source: https://github.com/RIVeR-Lab/tentabot

Thank you...Questions?



References

[1] Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang, and S. Shen, “Autonomous aerial navigation using monocular visual-inertial fusion,” Journal of Field Robotics, vol. 35, no. 1, pp. 23–51, 2018.

[3] M. Beul, D. Droeschel, M. Nieuwenhuisen, J. Quenzel, S. Houben, and S. Behnke, “Fast autonomous flight in warehouses for inventory applications,” IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3121–3128, 2018.

[4] H. D. Escobar-Alvarez, N. Johnson, T. Hebble, K. Klingebiel, S. A. Quintero, J. Regenstein, and N. A. Browning, “R-ADVANCE: Rapid adaptive prediction for vision-based autonomous navigation, control, and evasion,” Journal of Field Robotics, vol. 35, no. 1, pp. 91–100, 2018.

[5] H. Oleynikova, C. Lanegger, Z. Taylor, M. Pantic, A. Millane, R. Siegwart, and J. Nieto, “An open-source system for vision-based micro-aerial vehicle mapping, planning, and flight in cluttered environments,” arXiv preprint arXiv:1812.03892, 2019.

References

[6] F. Gao, W. Wu, Y. Lin, and S. Shen, “Online safe trajectory generation for quadrotors using fast marching method and Bernstein basis polynomial,” in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 344–351.

[7] V. Usenko, L. von Stumberg, A. Pangercic, and D. Cremers, “Realtime trajectory replanning for MAVs using uniform B-splines and a 3D circular buffer,” in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 215–222.

[9] K. Mohta et al., “Fast, autonomous flight in GPS-denied and cluttered environments,” Journal of Field Robotics, vol. 35, no. 1, pp. 101–120, 2018.