Distributed Multi-robot Collision Avoidance Using the Voronoi-based Method

Yiran Ding
School of Physics and Electronics, Hunan University, Changsha, Hunan Province, China, 410082
yrding@hnu.edu.cn

Abstract. Collision avoidance is a fundamental problem in multi-robot motion planning, enabling robots to coordinate in a safe and efficient manner. In this paper, we introduce several best-known distributed collision avoidance algorithms and present the Voronoi-based method in detail, which assigns optimal dominance regions for robots using the Voronoi diagram, constrains robot motions within the buffered Voronoi diagram, and follows right-hand rule when deadlock happens during navigating to the goals. A series of simulations are conducted to demonstrate the efficacy of collision avoidance guarantee under robot localization uncertainty while avoiding deadlock using our presented algorithm. We claim that such a method is superior to most of the existing distribution methods in a known environment in efficiency of implementation.

1. Introduction
Multi-robot coordination has drawn increasing attention in recent decades due to the increasingly stronger computational power and lower cost of robots. Robotic researchers benefit from getting better goal coverage, heterogeneous capabilities, better tolerance of individual failures, etc by using multirobot systems (MRSs). The broad applications of MRSs range from security surveillance to environment monitoring. For a team of mobile robots, individuals may be required to move to accomplish certain tasks or to optimize their poses to better finish the tasks. Therefore, the challenge in multi-robot motion planning arises as each robot may move simultaneously in a common area for finding its temporary goal. In order to guarantee safety and efficiency, a collision avoidance algorithm should be applied in addition to planning algorithms.

Collision avoidance is the task that a group of mobile robots avoids colliding with neighboring robots while navigated by a global planner in order to reach their goal destinations. A number of situations require robots to be capable of avoiding collision while planning online to interact with the environment in real-time, such as warehouse robots loading and unloading goods based on real-time demands. With a centralized solution, a master station (or robot) with strong computational power and communication capability collects real-time feedback from all (other) robots and sends back a global decision for the next time step. A centralized MRS requires easier motion planning compared to a distributed one, as the master station has access to real-time states of all robots. Jose et al, for instance, introduced a centralized scheme for multi-robot task allocation and collision-free path planning based on heuristic methods [1]. However, a centralized solution limits the team scalability due to the communication constraints in bandwidth and distance, and results in a fragile system as the failure of the master brings an invalid system. To overcome the weakness, in the majority of circumstances, a distributed MRS is used and each robot only has access to local information and makes a decision on its own. As a result, the system...
becomes more robust as the team is able to extend to a large swarm and the failure of an individual will only have a limited impact on the whole team.

One of the best-known collision avoidance strategies for robot motion planning is via velocity obstacles (VO), which represents the colliding velocities of the robot with given obstacles by the obstacles in the velocity space [2]. Such a method is extended to distributed multi-robot navigation scenarios later. Berg et al proposed the reciprocal velocity obstacle (RVO) for safe and oscillation-free navigation among autonomous robots and static and moving obstacles, based on which they further derive the optimal reciprocal collision avoidance (ORCA) [4] to provide a sufficient condition for multiple robots to avoid collisions among one another and guarantee collision-free navigation [3]. While Berg et al ignore robot kinematics and dynamics, Alonso et al proposed the non-holonomic optimal reciprocal collision avoidance (NH-ORCA) to further guarantee smooth and collision-free motions under non-holonomic constraints [5]. Guy et al extended the notion of VO and formulated the conditions for collision-free navigation as a quadratic optimization problem and used a discrete optimization method to efficiently compute the motion of each agent, a collision avoidance algorithm named Clearpath [6]. All above VO-based methods assume that each robot is able to measure the velocities of other robots in real-time, which is often difficult in practice.

Another class of collision avoidance algorithm is learning-based, taking an example of Long’s work in which a novel multi-scenario multi-stage training framework is proposed to exploit a robust policy gradient-based reinforcement learning algorithm trained in a large-scale robot system in a set of complex environments [7]. Such methods enable distributed collision avoidance at the sensor level, which directly maps the raw sensor data to collision-free steering commands. To implement a learning-based algorithm, training needs to be conducted offline for each robot prior to the tasks to generate an optimal collision avoidance policy.

In this paper, we introduce a Voronoi-based collision avoidance algorithm that does not require velocity measurements of agents or any offline training processes. This is done by robots partitioning the environment with the Voronoi diagram, a spatial partition investigating the domain for sites (robots) and planning the next goals only in their own Voronoi partitions, or Voronoi cells [8]. As robots move, Voronoi cells may also change, which requires robots to exchange their poses to neighbors so as to update their cells overtime. To account for robot localization uncertainty, we model robot locations as localization uncertainty regions, circular areas that contain all possible robot true locations to some confidence level and only allow each robot to move in a shrunk Voronoi cell, named a buffered Voronoi cell (BVC), based on the scale of the region [9]. This is similar to the convex uncertain Voronoi diagram which is proposed to maintain environment coverage under localization uncertainty and a collision avoidance region is proposed to guarantee collision-free [10]. As a result, robots are capable of avoiding collision in a probabilistic manner while navigating in the environment in parallel. A deadlock avoidance algorithm is also proposed to prevent robots from blocking their ways towards the goals while moving. Furthermore, we demonstrate the efficacy of the introduced method via a series of simulation experiments.

2. Voronoi-based Methods

A team of $n$ mobile robots $R = \{r_1, ..., r_n\}$ are tasked to move inside a convex open environment $E$. Robots are equipped with sensors that are able to localize themselves globally (e.g., via GPSs). We denote $q_i$ and $\hat{q}_i$ the true and estimated locations of $r_i$ in the global frame, respectively. We assume that robots move at maximum speed $v_{\text{max}}$ and robot states change at discrete times $t \in N$. We do not account for the robot’s kinematics and dynamics for simplicity. Thus, the maximum distance robots can travel within a time interval is $v_{\text{max}}t$.

2.1. Voronoi Diagram

Given a finite set of robots estimated positions $\{\hat{q}_1, ..., \hat{q}_n\} \in E$, a Voronoi cell $V_i$ consists of every point in $E$ whose distance to $\hat{q}_i$ is less than or equal to its distance to any other $\hat{q}_j$, i.e.,
where \( d(\cdot) \) denotes the Euclidean distance between two points. Note that \( V_i \) is a convex polygon and a pair of Voronoi neighbors share a common side. We call the tuple of cells \( V = (V_i)_{i \in n} \) the Voronoi diagram and \( \tilde{q}_i \) a Voronoi site, as demonstrated in Fig 1. Each Voronoi cell \( V_i \) should be updated at each time step \( t \) as \( \tilde{q}_i \) changes over time. Note that Voronoi cells do not overlap and that the union of all \( V_i \) covers \( E \) completely. Assuming \( q_i = \tilde{q}_i \), since Voronoi cells do not overlap, it is guaranteed that a team of robots does not collide with each other if each individual moves only within its current Voronoi cell at each time step. Thus, we select to utilize Voronoi-based methods for collision avoidance in multi-robot path-planning problems. We assume that robots are able to construct a Voronoi diagram over the team at the initial step. Thus, for a robot \( i \), we define its neighbor set \( N_i \) the set of all robots in the team that shares a Voronoi edge with \( r_i \). To update \( V_i \) over time in a distributed manner, \( r_i \) must exchange \( \tilde{q}_i \) with all robots in \( N_i \) at each time step \( t \).

Voronoi diagram and buffered Voronoi diagram

Fig. 1. Figure showing an example of a Voronoi diagram (blue lines) and a buffered Voronoi diagram (yellow lines) in a square area (black lines), with a set of six Voronoi sites, marked in blue dots.

2.2. Buffered Voronoi Diagram

Similar to (1), given a finite set of robots estimated positions \( \{\tilde{q}_1, ..., \tilde{q}_n\} \in E \), the buffered Voronoi cells (BVCs) are given by

\[
B_{i} = \{ x \in E \mid d(x, \tilde{q}_i) + s_i \leq d(x, \tilde{q}_j) \text{ for all } j \neq i \},
\]

where \( s_i \) denotes the safety distance, the minimal distance a robot should maintain from the Voronoi boundaries. We call the tuple of cells \( B = (B_i)_{i \in n} \) the buffered Voronoi diagram, shown in Fig 1. Inferred from (2), for a robot \( r_i \), its BVC \( B_i \) is constructed via shrinking the Voronoi cell \( V_i \) by a safety distance \( s_i \). While the Voronoi cells assign dominance regions for robots to dynamically allocating tasks such as information gathering optimally, the BVCs constraint the extent of the plane within which robots must move at each time step. By choosing an appropriate \( s_i \), robots are able to guarantee collision avoidance to each other to some confidence level while still approximate the optimal dominance region assignments, which will be discussed in Section III-A in detail.

Inferred from Zhou et al, we claim that

1) the BVC for a Voronoi site always exists;
2) for a Voronoi site, its BVC is contained in its Voronoi cell;
3) the BVCs never overlap [9].
3. Distributed Collision Avoidance

3.1. Localization Uncertainty Region

To construct the Voronoi cells, robots should be aware of their locations via GPS, motion capture systems, etc. In practice, errors exist in such localization due to sensor noise. In this section, we introduce the localization uncertainty region proposed similar to Chen’s previous work, a smallest circular region that contains all (or almost all) possible true locations of a robot given an estimated location [10].

We first assume that robot estimated locations are Gaussian distributed, i.e., \( P(\hat{q}_i) \sim \mathcal{N}(\mu, \Sigma) \), where \( \mu \) and \( \Sigma \) denote mean and covariance, respectively. While we assume that the covariance is time-invariant, robots may be heterogeneous in the localization covariance. The eigendecomposition of \( \Sigma \) is given by:

\[
\Sigma = R \Lambda R^{-1},
\]

where \( R \) is the square 2×2 matrix and \( \Lambda \) is the diagonal matrix with eigenvalues \( \lambda_1, \lambda_2 \) on the diagonal.

The localization uncertainty region of a robot \( r_j \) is defined to be \( B_j \) centered at \( \hat{q}_i \) with radius

\[
r_j = c \ max \lambda_j,
\]

where \( c \) is a positive constant. The probability of sensor \( s_j \) being located within this region is then

\[
p(q_j \in B_j) = \int_{B_j} \frac{\exp\{-\frac{1}{2}(x - \hat{q}_i)^T \Sigma^{-1} (x - \hat{q}_i)\}}{2\pi \det(\Sigma)^{\frac{1}{2}}} dx,
\]

To guarantee collision avoidance to a confidence level of 99.73%, for example, we can set \( c = 3 \). For non-Gaussian distributions, we assume that the spatial distribution of estimated locations is approximated from a sufficient amount of test data. A circular region is then created to cover all estimated locations to some confidence level. We denote the radius of the localization uncertainty region of \( r_j \) by \( l_j \).

For both classes of distributions, we set \( s_j = l_j + v_{\text{max}} t \) for BVC construction for a robot \( r_j \). This is to guarantee robot collision avoidance to desired confidence level while allowing a maximum moving distance during a time interval to ensure safety. In practice, we may also incorporate other safety factors with \( s_j \), such as deceleration distance for higher order dynamic systems, a time delay of reactions, robot sizes, etc.

3.2. Deadlock Avoidance

Robots should take action to avoid collision when two or more robots tend to move towards each other. A naive decision is to stop and yield to each other when two or more robots are too close. Robots may also choose to detour from any random directions. Both actions may result in deadlock, which happens when some of the vehicles block in others’ path, inducing at least one vehicle unlikely to reach its destination. This is because a “stop rule” does not enable robots to recover a safe distance and may cause permanent stops, while a “detour rule” does not guarantee there exists a direction in which a robot can avoid collision with all others.

Therefore, a right-handed (or left-handed) rule is introduced, outlined by Algorithm 1 [9] [10]. The strategy is twofold, aiming at breaking the deadlock condition both when a robot reaches a vertex and an edge of its BVC.

3.3. Distributed Collision Avoidance

In this section, we incorporate the Voronoi-based collision avoidance with the deadlock avoidance algorithm under the distributed multi-robot motion planning framework. The algorithm outlined in Algorithm 2 at each time step enables a team of robots to move in a distributed manner safely and effectively. At each time step, each robot first updates its Voronoi cell and BVC by exchanging estimated locations within the neighborhood, as outlined by lines 1-2. The communication only requires
local data exchange and is of low bandwidth, allowing fast and robust data streaming. Then each robot updates its goal and makes control decisions based on the distance towards the goal, as outlined by lines 39. By running this algorithm recursively overtime, robots are guaranteed to reach their goals since robots continue to move towards their goals while being able to avoid deadlock.

4. Results
We demonstrate the efficacy of the introduced collision avoidance strategy via a series of simulations in MATLAB. Initially, four pairs of robots are distributed at the edges of a 100m × 100m squared environment, each symmetric with respect to the square center. Robots are able to localize themselves with errors. Each robot makes its own decision, while communicating with only its neighbors for their current locations in order to maintain the Voronoi cell and BVC. Robots are going to swap locations between pairs with 10m/s simultaneously. As a result, all robots are expected to reach the environment center at the same time, blocking the way of each other. Tests will terminate if any robot collision happens.

We first test robot collision avoidance with a maximum localization error equal to 0.1m. As shown in Fig 2a, all robots eventually reach their goals, as a unique trajectory is shown connecting the initial location and the goal for each robot, indicating successful motion planning and collision avoidance using the BVC-based method. This is due to the fact that the robots are continuously driven to their goals while they are only allowed to move within their BVCs, which are never overlapping with each other, and that the deadlock avoidance algorithm guarantees their active moving while blocking with each other. This proves our argument that our introduced method allows a robot to plan motion safely and efficiently in an empty space while working in a distributed manner with only local estimated locations exchanged. Meanwhile, we find that a robot’s detours in the same direction while approaching the environmental center. This shows the effect of applying the right-hand rule for deadlock avoidance and illustrates the process that each robot avoids being blocked.

We also compare this test with another test, in which we set the robot’s maximum localization error equal to 0.5m. We find that each robot is still able to successfully reach its goal without colliding with another robot, which further demonstrates the efficacy and robustness of the introduced collision
avoidance algorithm. We also notice that it takes longer routes for robots to reach their goals since they allow larger distances between each other in order to guarantee safety when localization errors are larger. This coincides with our intuition that larger safety distances should be observed under higher localization uncertainty.

5. Conclusions

In this paper, we introduce a Voronoi-based collision avoidance algorithm allowing a group of robots to plan motion safely and efficiently under localization uncertainty in a distributed manner. The method utilizes the buffered Voronoi cell, a variant of Voronoi cell to guarantee collision-free while cooperating with each other and incorporate a right-hand rule to avoid deadlock to guarantee robots to reach their goals. This distributed algorithm only requires low bandwidth local communication, which allows the system to be infinitely scalable and robust. Furthermore, in contrast to most other reciprocal collision avoidance strategies, it requires only the easy-access position information, making it simpler to implement in real engineering projects. Our simulation results prove the arguments and show the efficacy of the introduced algorithm.

References

[1] K. Jose and D. K. Pratihar, “Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods,” Robotics and Autonomous Systems, vol. 80, pp. 34–42, 2016.

[2] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” The International Journal of Robotics Research, vol. 17, no. 7, pp. 760–772, 1998.

[3] J. Van den Berg, M. Lin, and D. Manocha, “Reciprocal velocity obstacles for real-time multi-agent navigation,” in 2008 IEEE International Conference on Robotics and Automation. IEEE, 2008, pp. 1928–1935.

[4] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, “Reciprocal n-body collision avoidance,” in Robotics research. Springer, 2011, pp. 3–19.

[5] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, “Optimal reciprocal collision avoidance for multiple non-holonomic robots,” in Distributed autonomous robotic systems. Springer, 2013, pp. 203–216.

[6] S. J. Guy, J. Chhugani, C. Kim, N. Satish, M. Lin, D. Manocha, and P. Dubey, “Clearpath: highly parallel collision avoidance for multi-agent simulation,” in Proceedings of the 2009 ACM
SIGGRAPH/Eurographics Symposium on Computer Animation, 2009, pp. 177–187.

[7] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan, “Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning,” in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 6252–6259.

[8] M. Erwig, “The graph voronoi diagram with applications,” Networks: An International Journal, vol. 36, no. 3, pp. 156–163, 2000.

[9] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, “Fast, online collision avoidance for dynamic vehicles using buffered voronoi cells,” IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 1047–1054, 2017.

[10] J. Chen and P. Dames, “Distributed and collision-free coverage control of a team of mobile sensors using the convex uncertain voronoi diagram,” in 2020 American Control Conference (ACC). IEEE, 2020, pp. 5307–5313.