# A Real-time Safe Path Planning System For Cooperative Robot

Yan He[1], Tao Lu[2] and Yinghao Cai[2]
[1] School of Artificial Intelligence, University of Chinese Academy of Science, Beijing,China
[2] State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Science, Beijing,China
{heyan2019}@ia.ac.cn, {tao.lu}@ia.ac.cn, {yinghao.cai}@ia.ac.cn

***Abstract*—The cooperative robot is requested to co-work with humans efficiently and safely in an environment with flexible arrangements. Safe path planning is the most important issue which must be resolved first in the process of manipulation. In this paper, we present a safe path planning system that could plan the manipulation path in real-time based on the environmental changement and guarantee safety when the robot interacts with the environment and humans. In this system, we first build a real-time obstacle Octomap from the environment RGB-D images, which could effectively difference the robot area from others in the environment and eliminate the robot influence during the map building. And then, we adopt the RRT-Connect method to plan the safe path in the Octomap. When the planning path is obstructed by the dynamic objects, the system will replan the new safe path based on the changed Octomap. The experimental results show that our system can effectively avoid obstacles in a dynamic environment and safely reach the manipulation destination.**

***Index Terms*—Cooperative Robot, Safe Path Planning, Octomap, RRT-Connect**

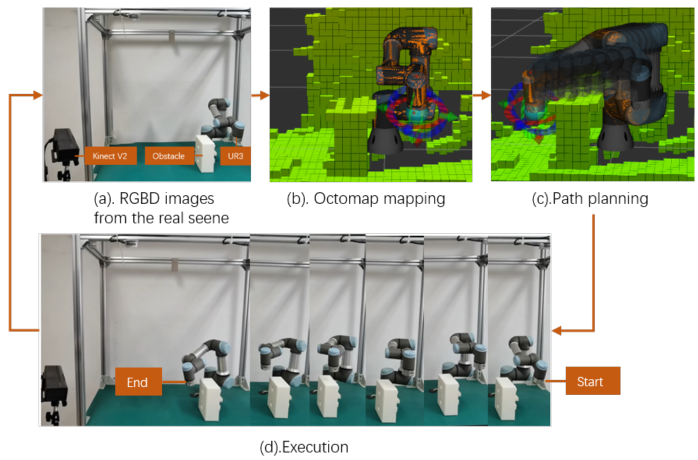



Fig. 1: Overview of the safe path planning system.

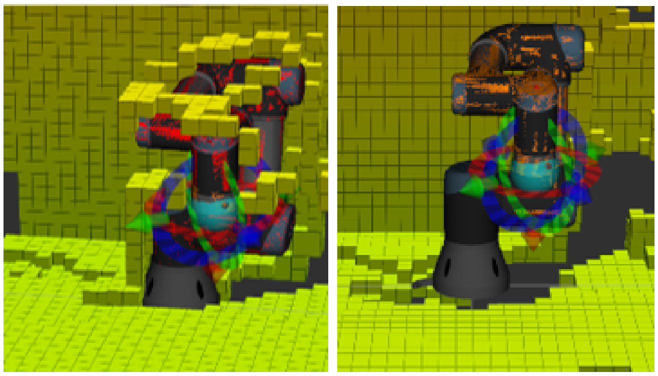
## I. Introduction

The cooperative robot is designed to work with humans in the same working space. At present, most cooperative robots guarantee safety by stopping when they contact unanticipated objects or humans during manipulating. The sensors which are used to percept mainly include joint torque sensors and skin pressure sensors. These sensors are now relatively expensive and also have to be exposed to obstacles [1] [2]. Visual sensors could percept the changement of the environment and predict the dangers before the contaction between the robot and other objects [3]. The early research is limited by camera performance, and the monocular camera is used as a visual sensor. However, the monocular camera can not obtain the depth information of the environment. Tan et al. proposed a visual system to capture the motion [4] of the sitting operator (upper body only) by having the operator wear a colour mark. The method relies on colour consistency and is not suitable for uneven ambient lighting conditions. Detection of depth information based on binocular or depth camera is a more commonly used environmental perception method. Schiavi [5] and Fischer [6] respectively put forward the method of obstacle detection based on 3D depth information collected by ToF camera and lidar. ToF camera provides high-performance solutions, but pixel resolution level is insufficient for deep image acquisition [7]. To address these challenges, we propose a safe path planning system with an RGBD camera to avoid obstacles in real-time. By converting the point cloud into an Octomap, the proposed system plans the safe path based on the RRT-Connect method in the Octomap. The whole system is built under the ROS(Robot Operating System) framework.

## II. SYSTEM DETAILS

In this section, we introduce the path planning system in detail, as shown in Fig. 1.

Firstly, we use Octomap to build a 3D obstacle map of the environment. Octomap [8] is an efficient probabilistic 3D mapping framework based on octrees in which each node of octrees represents whether the current space of the cube is occupied by the obstacle. We obtain point cloud information through an RGB-D camera and turn it into an Octomap of the environment as shown in Fig. 2a. While in this map, the robot area is also transferred as obstacles. If the map is directly created, the robot arm will be regarded as a part of the environmental obstacle. Therefore, it is necessary to filter out the points that represent the robot. To address this problem, we adopt the method of enveloping the ball based on the robot model. Set spheres at specific intervals in the URDF model of the robotic arm, so that these spheres can wrap the robot model. The Octomap filtered out robot is shown in Fig. 2b. Secondly, we customize the motion planning



(a) Octomap with robot as obstacle (b) Octomap filtered out robot

Fig. 2: The Octomap of working space.

algorithm based on RRT-Connect [9] through MoveIt in ROS. The critical problem of path planning is to avoid collisions. Let C be the configuration space of the robotic arm, then $C \epsilon R^n$, where n is the DOFs of the robot. For a robotic arm, a group of joint configurations corresponds to a robotic arm's pose state, which is expressed as q=$[\theta_1, \theta_2, ..., \theta_n]$,$\theta_1 \sim \theta_n$ represents the position of the n joints. The configuration space C can be divided into a free configuration space $C_{free}$ that does not overlap with the robotic arm configuration q and an obstacle configuration space $C_{obs}$ that overlaps with q. They are complementary. Therefore, the path planning problem can be described as: Find a set of continuous configuration P in the free configuration space such that $P \in C_{free}$, and the starting point is the initial configuration $q_{init}$, and the ending point is the target configuration $q_{goal}$. The obtained P is the collision-free feasible path of the robot. The characteristic of robotic arm path planning is that it needs to search for paths in a high-dimensional joint space, which often leads to a large amount of calculation. Here we adopts the RRT-Connect path planning method based on a sampling search. It can quickly and effectively search the state space. Algorithm 1 shows the RRT-Connect algorithm flow.

Two trees, $\tau_a$ and $\tau_b$ are maintained at all times until they become connected, and a solution is found as shown in Fig.3. In each iteration, one tree is extended, and an attempt is made to connect the nearest vertex of the other tree to the new vertex. Then, the roles are reversed by swapping the two trees. This causes both trees to explore the configuration space in which these bodies do not collide with any static obstacles, while trying to establish a connection between them. The growth of two RRTs was also proposed in [10] for kinodynamic planning; however, in each iteration, both trees are incrementally extended toward a random configuration. The current algorithm also attempts to grow the trees towards each other, which has been found to yield much better performance.

## III. EXPERIMENTS AND RESULTS

We test our system in an environment with dynamic obstacles. The experiments are conducted using real hardware: the UR3 robot arm, one Kinect V2. And the algorithms are all running in the ROS framework.

**Algorithm 1** RRT-Connect algorithm

$BUILD\_RRT(q_{init})$
$\tau.init(q_{init})$;
**for** $k = 1$ to $K$ **do**
  $q_{rand} \leftarrow RANDOM\_CONFIG()$;
  $EXTEND(\tau, q_{rand})$;
**end for**
Return $\tau$
$EXTEND(\tau, q)$
$q_{near} \leftarrow NEAREST\_NEIGHBOR(q, \tau)$;
**if** $NEW\_CONFIG(q, q_{near}, q_{new})$ **then**
  $\tau.add\_vertex(q_{new})$;
  $\tau.add\_edge(q_{near}, q_{new})$;
  **if** $q_{new} = q$ **then**
    Return Reached;
  **else**
    Return Advanced;
  **end if**
**end if**
Return Trapped;

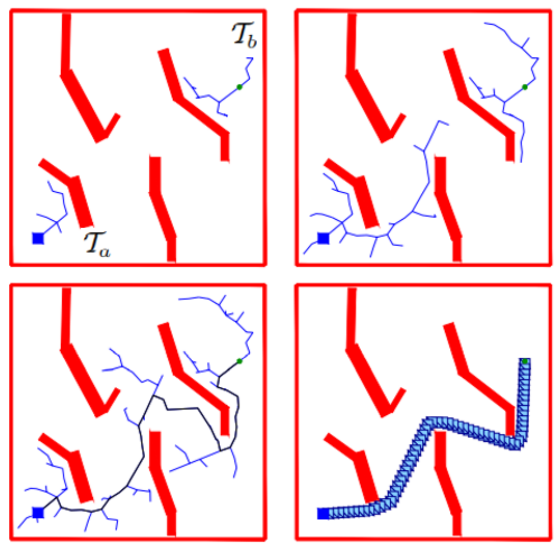



Fig. 3: Growing two trees towards each other.

### A. Building Real-time Obstacle Scene Map Based on Octomap

We use the Kinect V2 to obtain the obstacle depth information, then converting the obstacle depth information into the point cloud information, publishing the point cloud data to a specified topic, then calling the srv component under the ROS to convert in real-time, and publishing it to another Octomap topic. The built Octomap in different environment condition is shown in Fig.4.

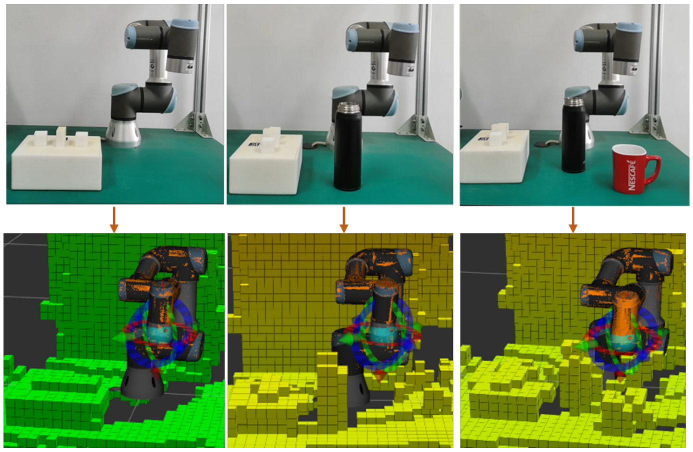


Fig. 4: The built Octomap in different condition.

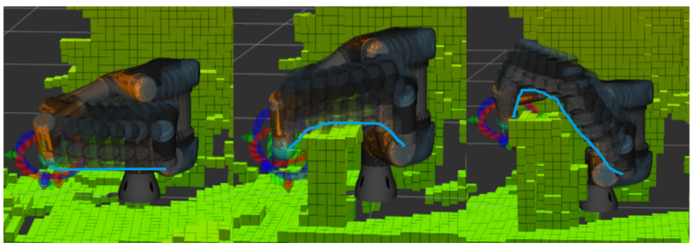


Fig. 5: Real-time safe path planning of robot in Octomap.

### B. *Experiment of Global Autonomous Path Planning*

The whole system was tested by planning and executing the trajectory between the two pre-defined points A and B in the workspace. We first don't place any obstacles, let the robotic arm move from point A to point B, then set an obstacle, and then let the robotic arm move from point A to point B to test the obstacle avoidance effect of our algorithm, and finally change Obstacle position, test the effect of real-time obstacle avoidance. When the Octomap is constructed, the RRT-connect algorithm is called to plan the obstacle avoidance path. The planning path results in simulation are shown in Fig.5.The execution process of planned safe path in real scene is shown in Fig.6.

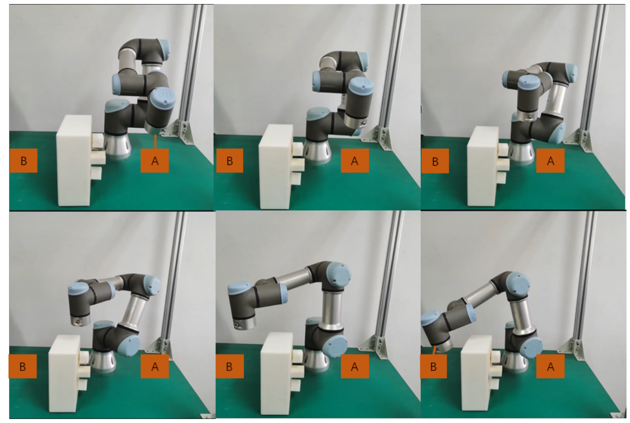



Fig. 6: Real-time safe path planning of robot in real scene.

## IV. CONLUSION

In this paper, we present a safe path planning system that could plan the manipulation path in real-time based on the environmental changement and guarantee safety when the robot interacts with the environment and humans. Through the real-time Octomap, the system replans the safe path according to the changement of the objects. The experimental results show the effectiveness and safety of our system in a dynamic environment.

## REFERENCES

[1] L. Baccelliere, N. Kashiri, L. Muratore, A. Laurenzi, M. Kamedula, A. Margan, S. Cordasco, J. Malzahn, and N. G. Tsagarakis, "Development of a human size and strength compliant bi-manual platform for realistic heavy manipulation tasks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5594–5601.

[2] S. Haddadin, *Towards safe robots: approaching Asimov's 1st law*. Springer, 2013, vol. 90.

[3] B. Siciliano and O. Khatib, *Springer handbook of robotics*. springer, 2016.

[4] J. T. C. Tan and T. Arai, "Triple stereo vision system for safety monitoring of human-robot collaboration in cellular manufacturing," in *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)*. IEEE, 2011, pp. 1–6.

[5] R. Schiavi, A. Bicchi, and F. Flacco, "Integration of active and passive compliance control for safe human-robot coexistence," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 259–264.

[6] M. Fischer and D. Henrich, "3d collision detection for industrial robots and unknown obstacles using multiple depth images," in *Advances in robotics research*. Springer, 2009, pp. 111–122.

[7] M. Yekkehfallah, M. Yang, Z. Cai, L. Li, and C. Wang, "Accurate 3d localization using rgb-tof camera and imu for industrial mobile robots," *Robotica*, pp. 1–18.

[8] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[9] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 995–1001.

[10] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.