RoboCup Rescue 2023 Team Description Paper NuBot-Rescue

Chuang Cheng, Wenbang Deng, Ruihan Zeng, Hainan Pan, Hui Zhang, Huimin Lu, Junhao Xiao, Bailiang Chen, Chenghao Shi, Tengda Zhu, Shunjie Gong, Neng Wang, Tuochang Wu, Xuan Zhang, and Xieyuanli Chen

Info

Team Name:	NuBot-Rescue
Team Institution:	National University of Defense Tech.
Team Country:	China
Team Leader:	Chuang Cheng
Team URL:	https://nubot.trustie.net/intro

***Abstract*—Rescue robots are important for urban search and rescue. When disaster strikes, search and rescue robots minimize the risk of secondary damage while enabling human rescuers to quickly locate and extricate victims. Our team designs the NuBot rescue robot, including its mechanical structure, electronic architecture, and software system. Benefiting from the strong mechanical structure, our rescue robot has good mobility. It is durable and will not be trapped facing highly cluttered and unstructured terrains during urban searching and rescuing. We build the electronic architecture of our robot under industry standards, which can bear electromagnetic interferences and physical impacts from the intensive tasks. We develop the software system upon the Robot Operating System (ROS), using both self-developed programs and several basic open-source packages provided by the ROS community. We develop a complete software system for robotic rescuing, including localization, mapping, exploration, object recognition, etc. Our robot system has been successfully applied and tested in the RoboCup Rescue Robot League (RRL) competitions. Most recently, we won the Best in Class of Exploration and Mapping in the 2021, 2022 RoboCup RRL, and Search and Inspect in 2022 RoboCup RRL. We achieved the top 5 performance in the previous competitions and won the Best in Class small robot mobility in 2016 RoboCup RRL Leipzig Germany. Our team has also won the championships of RoboCup China Open RRL competitions since 2015. In this year, we equip a new-style LiDAR to obtain dense point clouds and reduce the dead zone of perception. We also propose a novel point cloud registration algorithm to achieve precise and robust simultaneous localization and mapping (SLAM).**

***Index Terms*—RoboCup Rescue, Team Description Paper, SLAM, Autonomy, ROS.**

I. Introduction

FACING the demand of urban search and rescue (USAR), various types of rescue robots are designed, e.g., wheeled, legged, and tracked robots. Because of the more robust mobility and adaptability to complex environments, tracked robots are widely used in USAR missions. We also design our rescue robot, NuBot, as a typical tracked robot, and will give a thorough introduction in this paper.

The base of our rescue robot mainly consists of six tracks and a robot arm. With the help of the customized powerful tracks, we manage to focus on improving the traversing ability of front and rear fins, so as to make the mobile platform more adaptive in the face of challanging terrains like stairs, 45° slopes, and stepfields. Our self-developed robot arm is equipped with powerful motors and mechanisms used to adjust the length of the arm. In consequence, our robot arm can be used to accomplish tough dexterity tasks, such as grasping, unscrewing, weightlifting, and unpacking. Additionally, with the cameras mounted on the robot arm, our robot can realize precise visual servo control and help the operator to cover more blind spots by manipulating the flexible robot arm to the interesting position.

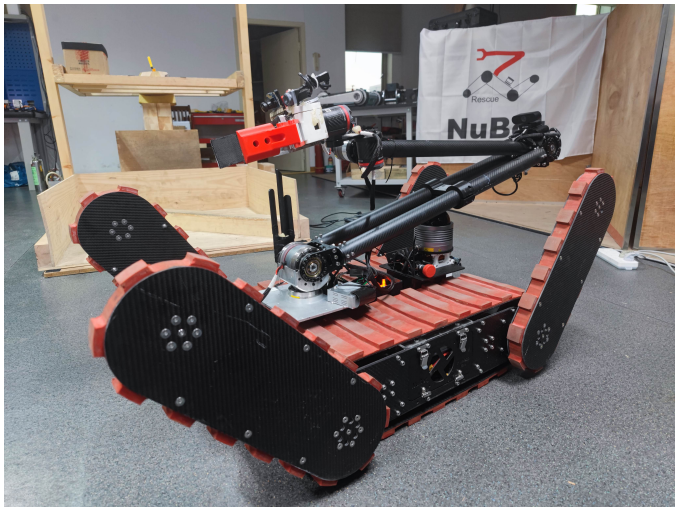


Fig. 1. Illustration of our rescue robot, NuBot.

To enhance the ability of perception, we mount a new-type 3D RoboSense light detection and ranging (LiDAR), an inertial measurement unit (IMU), a microphone, several USB cameras, and other extension accessories on our robot. By sending back the encoded real-time image information of onboard cameras, the operator can control the robot to traverse the complex post-disaster environments remotely and accurately. With the image information, we also manage to recognize the victims and the hazmats.

Besides the hardware mentioned above, we develop a comprehensive software system to enable the basic function and improve the level of intelligence of our rescue robot. Some highlights of our software system are autonomous exploration and mapping, semi-autonomous end-based robot arm control, autonomous narrow path traversing, and semi-autonomous fin-

Team NuBot-Rescue is supported by the College of Intelligence Science and Technology, National University of Defense Technology (NUDT).

based steep terrains traversing. All of these functions will be introduced in Section II-B.

The main improvements for this year's competition can be summarized as:

- We equip a new-type 3D LiDAR to realize accurate localization and dense mapping.
- A novel point cloud registration algorithm is proposed to improve the robustness and accuracy of localization.
- We design a lightweight robot arm with strong load capability.

In the previous years, we have published the following rescue-related papers: [1], [2], [3], [4], [5], [6].

II. System Description

Many impressive results about the design of the robot system for USAR missions have been achieved, and various robots have been developed for USAR tasks. In the early years, rescue robots were usually teleoperated by human operators. However, with robotic technology advancing significantly, the robot's autonomy level has been improved greatly. In order to adapt to the actual rescue environment, we also designed our rescue robot system to realize teleoperated operation, semi-autonomous, and fully autonomous operation [3].

A. Hardware

1) Locomotion: Our robot uses a tracked platform, as shown in Figure 1. The tracked robot with front and back sub-tracks (flippers) provides effective mobility, and it is the most common platform used in the RoboCup RRL competitions and real rescue missions.

2) Power (Batteries): Four 24V Li-PO batteries are equipped on the robots, where two for the onboard computer and sensors, and another two for motors.

3) Electronics: The robot contains the matched LiDAR driver, robot arm driver, and motor drivers to actuate corresponding electronics. Meanwhile, we mount an LED light in the front of the robot arm, so as to provide light source when the robot accesses dark environment.

4) Manipulation/ directed perception: We utilize a Sony joystick to manipulate the rescue robot moving. As for directed perception, visual perception is the most important source for environment observation and victim detection. Therefore a pan-tilt-zoom camera is mounted on the tracked platform. Besides, two low-cost USB video camera and a thermal image camera have been employed. The visual sensors are fused to detect and localize victims.

5) Sensors: Besides the cameras, the robot is also equipped with a 32-beam hemispherical RoboSense Bpearl LiDAR. This LiDAR can acquire $360° * 90°$ accurate and dense point clouds of surroundings. The closest detection range of this LiDAR is less than 10 centimeters, which is appropriate for using in the narrow post-disaster environments.

In order to measure the robot's pose when exploring in the unstructured and uneven terrains, a 9-DOF inertial sensor, Xsens MTI-300 has been integrated. MTI-300 is a miniature Inertial Measurement Unit (IMU) that outputs posture angles and provides a calibrated three-axis acceleration, angular velocity, and magnetic field strength.

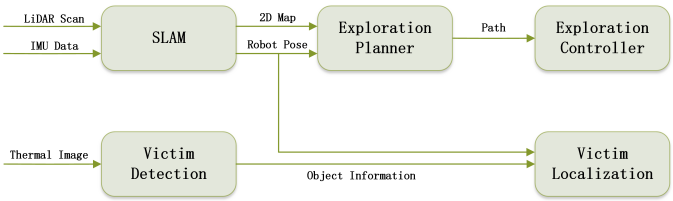


Fig. 2. The software architecture based on ROS. ROS nodes are represented by rectangles, topics by arrow-headed.

6) Computation: The robot uses a high-performance Microcomputer from Lenovo. The computer (with Intel Core i9 CPU) provides enough processing ability to deal with huge data and robustness when traversing challenging terrains.

7) Robot arm: The manipulator system has been greatly improved in this year. It has 6 degrees of freedom and is made of integrated joint motors. So it can realize the operation of the end workspace. In order to reduce weight, aluminum alloy and carbon fiber materials are used to form the connector of each joint, and the mechanical arm adopts a foldable configuration design to reduce the airborne volume, Meanwhile, we creatively present the scheme of the retractable manipulator. The length of the boom and jib of the manipulator can be adjusted at any time. The arm's length is 1.4m, and the maximum load is 10kg. When the boom and jib are fully extended, the total length is 2.2m, and the maximum load is 2.5kg. The manipulator's working range and end load can be adjusted according to different task requirements.

B. Software

Robot Operation System (ROS), which is the most popular robotic framework nowadays, is used to build the software of our rescue robot. It provides open-source tools, libraries, and drivers for robotics research and applications. Thus, ROS enables researchers to quickly and easily conduct experiments. The software architecture of our rescue robot is shown in Figure 2. Table IV in the Appendix shows the software we use.

1) low level control: The base of the robot is driven by 6 Tmotor motors, which are responsible for driving the two main tracks and four flippers, respectively. ROS Melodic is adopted for information processing and control at a low level. ROS establishes communication with the built-in motor driver through Socketcan driver, receives, and sends predefined CAN messages to realize the underlying control, and makes the angular velocity of the motor reach the target value.

2) communication protocol: Our data communication is under TCP/IPv4 communication protocol and in the form of ROS topics.

3) localization and mapping: Our robot realizes SLAM by implementing tightly-coupled SLAM method [7]. This method can fuse GNSS, IMU, and LiDAR data for accurate localization effectively and build globally consistent maps. To make use of our sensors data and improve the localization and mapping results, we implemented the following adjustments:

- Due to the drift of magnetometer measurement from the IMU, we use the stable absolute heading provided by dual

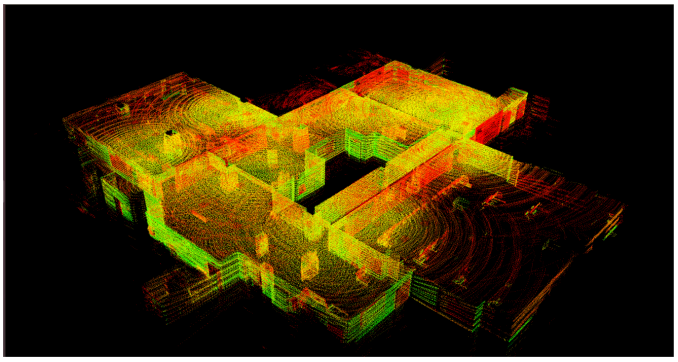


Fig. 3. The 3D mapping result of the Exploration and mapping task in 2022 RoboCup RRL.

antennas of GNSS to replace that of the IMU. Meanwhile, we utilize the angular velocity from the IMU to take the integral of the GNSS heading, so as to obtain a heading measurement at the same high rate as the IMU.

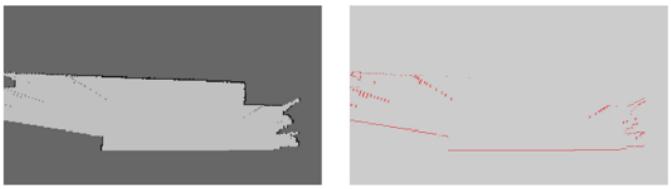
- We leverage scan context [8] to replace the original Euclidean distance-based loop closure detection method so that we can increase the possibility of accurate loop closure detection.
- We propose a point cloud registration method using multiplex dynamic graph attention networks to realize keypoint matching [9]. It contains a novel layer to construct the graph network dynamically, which captures the local information better and improves the distinctiveness of the learned descriptors. As a result, the accuracy of localization is improved.

Figure 3 shows the 3D mapping result of the Exploration and mapping task in 2021 RoboCup RRL generated by the tightly-coupled SLAM method. The result presents a globally consistent and accurate point cloud map.

4) Frontier-based exploration: The primary problem of autonomous exploration is: based on existing knowledge about the real world, where should the robot move to efficiently acquire new information?

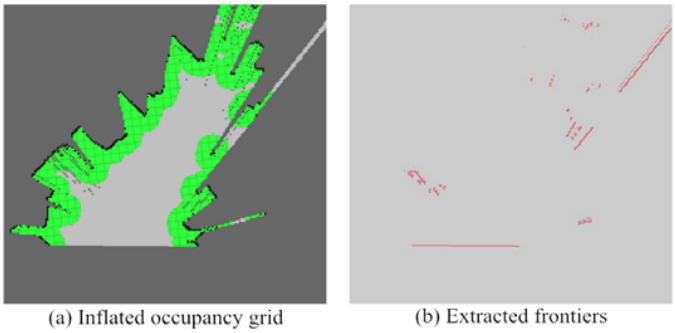
Most approaches use the occupancy grid. When a grid map has been built, all the grid can be divided into three categories: Free, Unknown, and Occupied. The primary idea is as follows: to get new information, the robot goes to a frontier that separates known regions from unknown regions. The frontier here is a cell in the occupancy grid marked as free but has a neighboring unknown cell. A segment of adjacent frontiers is considered a potential target if it is large enough for the robot to get through. The closest one is selected if more than one potential target is detected. Figure 4 shows the result of extracted frontiers.

A disadvantage of directly extracting the frontier from the original map is that the extracted frontier may be close to obstacles. To overcome this problem, we utilize the Inflated Obstacle method. This method will transform the cells within a certain distance to obstacles as Occupied. Thus the frontiers will not be extracted in these areas. Figure 5 shows the result of extracted frontiers in the occupancy grid with inflated obstacles.



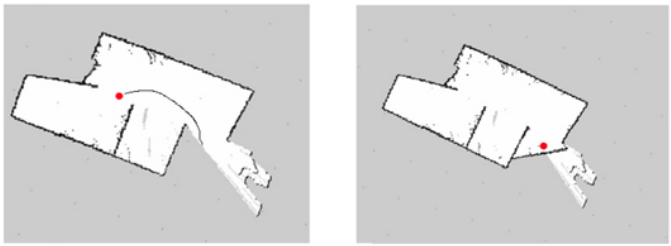
(a) Occupancy grid (b) Extracted frontiers

Fig. 4. Occupancy grid and extracted frontier. (a) The black area is Occupied, the white area is Free, and the gray area is Unknown. (b) The red points are detected as frontiers.



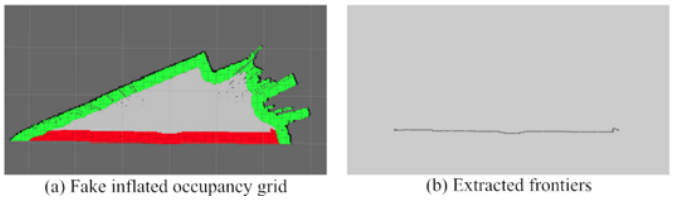
(a) Inflated occupancy grid (b) Extracted frontiers

Fig. 5. Occupancy grid with inflated obstacles and the extracted frontier. In (a), green areas are the inflated obstacles.



(a) Robot follows the path (b) Robot crashes with obstacle

Fig. 6. A example of the robot being stuck.



(a) Fake inflated occupancy grid (b) Extracted frontiers

Fig. 7. Occupancy grid with fake inflated obstacles and the extracted frontier.

Unknown areas could be occupied or free areas, which have not been known by robots. In the narrow rescue environment, the unknown areas' probability of being obstacles is quite high. Furthermore, the computational complexity of path planning will increase with the scale of the built map, so the frequency of path planning will not be high for autonomous UASR robots. If the robot takes one frontier as the target point, and the frequency of path planning is low, or the robot's moving distance to the target is small, the robot may crash with unknown obstacles.

Figure 6 shows an example where the robot may be stuck. The red point represents the robot's position, and the black

curve represents the planned path. In this situation, because of the low frequency of path planning, the robot will not re-plan the path before arriving at the target point, so the robot may crash with obstacles. The worst result is that the robot can not move anymore.

To deal with this problem, we proposed a method named Fake Inflated Obstacle. In this method, those extracted frontiers located between free areas and unknown areas are considered obstacles, then these obstacles are inflated, and finally, new frontiers can be searched between these inflated areas and free areas. Using this method, the distances between the target point and the actual obstacles are enough for our robot's safe movement. The extracted frontiers from fake inflated obstacles are shown in Figure 7.

5) path planning and navigation: When a target has been selected by the frontier-based method, the problem turns into optimal path planning. Following Javis and Byrne [10], we proposed the distance transform to find the closest way from an arbitrary starting point to a fixed target. The distance transform of an occupancy grid calculates the cost to reach the target cell for each free cell. The cost between two cells (without obstacles between them) can be the chessboard distance, city block distance, or the Euclidian distance. After the distance transform is applied for each grid cell, the shortest path from any cell to the target cell can be searched simply by following the steepest gradient.

Based on the SLAM and exploration planning algorithms mentioned above, the robot can build the 2D grid map and plan a path to the next frontier. Ideally, the robot can explore the environment autonomously after integrating a simple controller to compute the commands to drive itself. However, unlike virtual simulation environments or ideally indoor scenarios, real disaster sites are filled with unstructured terrains. Due to the inaccuracy of tracked vehicles, it is quite challenging to realize accurate robot control to follow the exploration path. A simple controller, which directly produces the velocity command by calculating the biases of the current position and orientation of the robot with the target, can not perform well in real-world experiments. Therefore, we propose a novel controller combining the exploration planer and multi-sensor information to overcome the influence of challenging terrains. The controller's inputs are LiDAR data, IMU data, the current position and orientation of the robot, and the target.

The experimental results show that using this kind of multi-sensor-based controller, the exploration efficiency and robustness can be improved. The robot can explore the full rescue environment of the 2016 RoboCup China Open RRL competition, as shown in the video posted on our website.

6) semi-autonomous exploration: In order to adapt to the actual rescue environment and make the operator easy to accomplish the USAR missions, we have added some semi-autonomous functions to the robot.

The first semi-autonomous function is the waypoint navigation. We collect waypoints during the run phase and use these waypoints to realize semi-autonomous navigation when the operation way of the robot is switched to semi-autonomous. We use the "timed elastic band" (TEB) approach [11] to reach the waypoints quickly. The TEB method takes the robot's state and the adjacent state's time as the optimized node. Constraints between the states, such as velocity, acceleration, and non-holonomic constraints, are used as the optimized edges.

7) semi-autonomous steep terrains traversing: The other semi-autonomous function we have implemented is flipper-based semi-autonomous traversing, using robotic inertial sensors and flippers to traverse steep terrains. According to the different postures of the robot during the movement, the flipper-based semi-autonomous traversing algorithm automatically adjusts the flippers to keep the robot in a safe posture. This semi-autonomous function frees the operator from the operation of the robot's flippers during the entire mission and simplifies the complexity of the operation.

8) autonomous narrow path traversing: In order to realize the rapid and collision-free traversing in the narrow path (MAN 1 of 2019 rrl_rulebook), we developed an autonomous traversing algorithm. This algorithm leverages point clouds from the LiDAR to acquire the edges and the corners in the narrow path. This algorithm can be done due to the regular environment. With these edges and corners, the robot can be aware of the position of the path, the turn, and itself relative to the center of the path.

9) victim detection: Reliable detection of human victims in unstructured post-disaster environments is a key issue for USAR robots, which is challenging in the rescue environment with low illumination, dust, and smoke when using the normal visible light camera. Therefore we use a thermal camera to recognize the simulated victims autonomously using a blob detection algorithm. After segmenting the thermal image with a threshold like 36°, the connected warm regions can be regarded as victims.

After detecting the victim successfully, the victim's position should be estimated on the built map. The image coordinate of the victim can be used to evaluate the victim's direction in the camera coordinate system. Because the victim should be located on the obstacles, using the victim's direction and the camera's orientation, the victim's position can be estimated by searching the nearest obstacle on the built map along the victim's direction.

10) arm control: We utilize the joystick to control the arm. By operating the joystick, the arm can be controlled in either a joint-based or an end-based way.

11) arm planning: The angle of each joint is obtained by reading the position information of each motor. The angle information of each arm's joint is used to visualize the manipulator and the inverse solution operation of the manipulator. Through the inverse solution operation, the end of the manipulator can complete the operation of fixed-point rotation and translation in any direction, and the position of the end fixed-point can be changed arbitrarily to complete different tasks. In this way, we can realize the semi-autonomous function of robot arm control.

C. Communication

Our robot can either use passive tether or wifi hardware to communicate with the operating station. We implement H3C WA6638i as the wifi hardware. It supports both 802.11ax/b/g/n

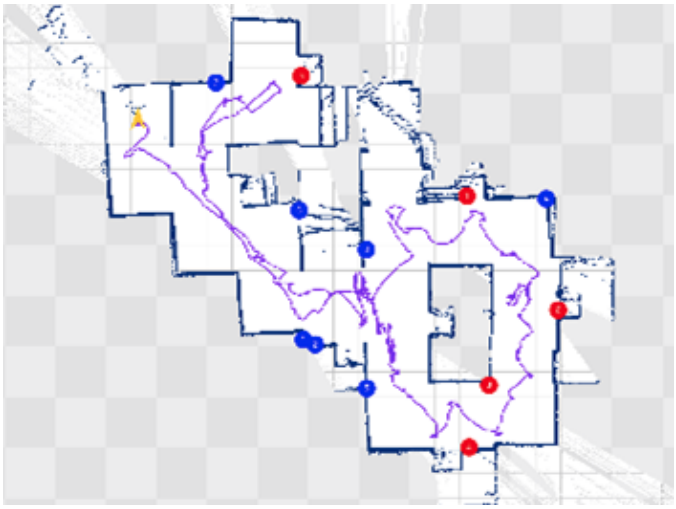


Fig. 8. The autonomous exploration results of our rescue robot participating the 2016 RoboCup China Open RRL. The starting pose of the robot is marked by yellow arrows, the detected victims marked by the red dot, and the recognized QR-codes marked by the blue dot. In the final competition, the robot discovered five victims correctly and autonomously.

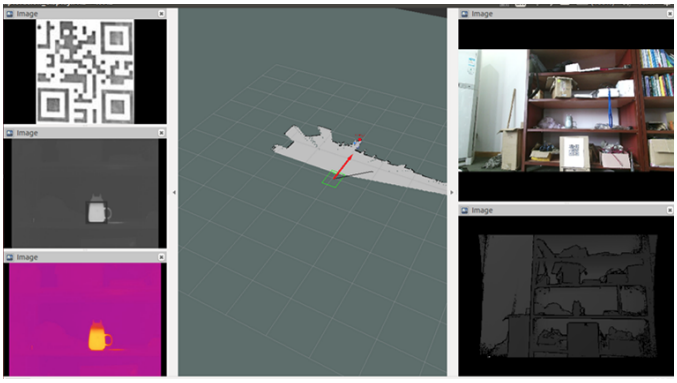


Fig. 9. The result of victim detection and localization, where the found victim is marked by red dot in the map.

2.4GHz and 802.11ax/ac/n/a 5GHz communication protocols. The transmitting power is 20dBm, and the gain of antennas on the robot is 3dBi. The SSID will be ”RRL_NuBot”.

D. Human-Robot Interface

To realize the remote operation, the operation station accesses the onboard computer remotely through the remote desktop tool NoMachine. Based on ROS, we developed a user interface suitable for our robot, see Figure 10. The user interface includes ① the parameter adjustment area, ② the visualization area, and ③ the sensor data display area. The parameter adjustment area can modify the relevant parameters in real-time, and the visualization area contains the 3D model of the robot, where the 3D model can display the actual posture of the robot and the robot arm. The sensor data display area shows the onboard camera images and CO_2 sensor data.

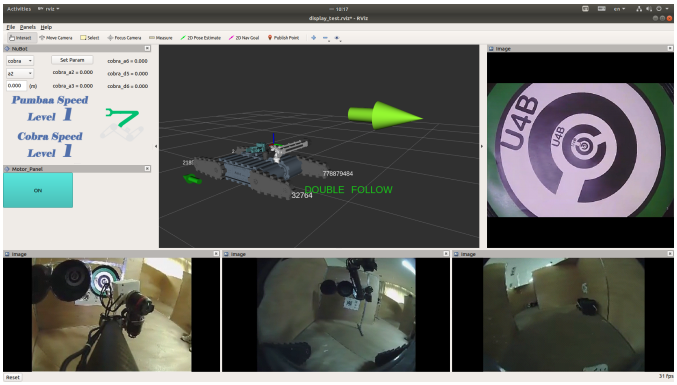


Fig. 10. The screenshot of user interface.

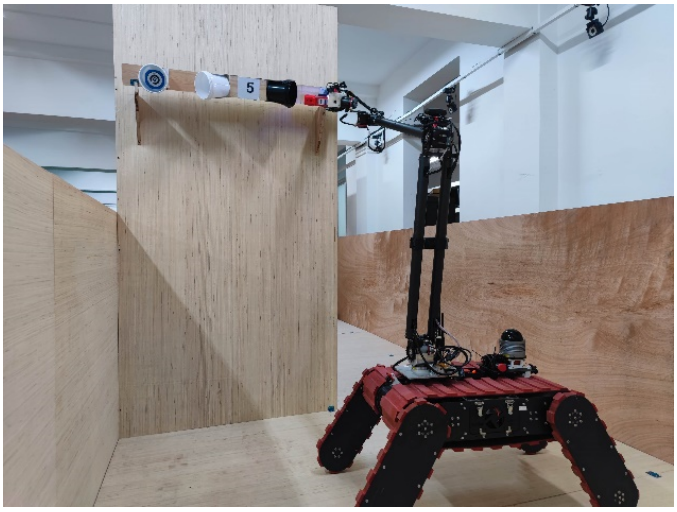


Fig. 11. The scene of implementing Search and Inspect.

III. Application

A. Set-up and Break-Down

The entire system consists of a tracked robot, an operation station, an antenna device, and a PS4 joystick. When setting up, the operation station needs to be connected to the robot via cable or wifi, and the remote desktop software is used to control the robot remotely. When breaking down, the robot will be powered off remotely. After that, all devices can be taken back.

B. Mission Strategy

We mainly rely on our tracked platform to traverse steep terrains and the flipper-based semi-autonomous exploration algorithm to simplify the operation for the Maneuvering and Mobility tasks. For the Dexterity tasks, we utilize the 6-DOF robot arm and the corresponding semi-autonomous end control algorithm to realize the robot arm’s flexible operation, see Figure 11. For the Exploration tasks, we will implement the SLAM and autonomous exploration methods mentioned in Section II-B, see Figure 3.

C. Experiments

In our lab, we have set up a test field, which contains Center (MAN 1), Align (MAN 2), Stepfields (MOB 3), Elevated Ramps(MOB 4), and Stair Debris (MOB 5) in the standard of 2019 rrl_rulebook. Thus, when we build our robot, we utilize these terrains to test the performance of our robot. Meanwhile, these terrains are used to train our operators.

D. Application in the Field

We present the results achieved when participating the RoboCup China Open 2016. We won the championship based on exploring the rescue environment autonomously in this competition.

Figure 8 shows the map of the explored arena and the marking of the found victims and recognized QR codes on the map (Figure 9). It should be noted that the robot successfully found five victims autonomously, and the total number of victims was eight in the final competition. The number of recognized QR codes correlates with the fraction of the arena explored by the robot during the competition. Videos of the final competition and experiments are available on our website.

The rescue robot we developed has also been to the earthquake site for experiments and data acquisition. Meanwhile, we have tried implementing our robot in explosion removal scenes, such as map construction, target detection, dangerous objects transferring and handling.

IV. Conclusion

Our rescue robot can achieve good performances in both telecontrolled and autonomous modes. Currently, we are going to participate in the 2023 RoboCup ChinaOpen. The RoboCup ChinaOpen will provide a chance for us to prepare and improve our robot system for participating the RoboCup 2023 .

Appendix A
Qualifying Videos

Our competition videos in RoboCup Rescue 2021 are listed below:

- Mapping: https://1drv.ms/v/s!AiP2j9W_Z84ximkPFK1s_aJHhMNE
- Linear Inspect: https://1drv.ms/v/s!AiP2j9W_Z84xinS0DJ8VbMTUpv5_

Appendix B
Team Members and Their Contributions

Currently, there are three supervisors and eleven team members in the NuBot-Rescue team. The supervisors are Hui Zhang, Huimin Lu, Junhao Xiao. The team members are as follows:

- Chuang Cheng — Mechanical design, software system.
- Wenbang Deng — software system.
- Bailiang Chen — Mechanical design.
- Chenghao Shi — Software system, software system.
- Hainan Pan — Drive system.
- Ruihan Zeng — Software system.
- Tengda Zhu — Software system.
- Shunjie Gong — Manipulator, mechanical design.
- Neng Wang — Manipulator, Software system.
- Tuochang Wu — Mechanical design.
- Xuan Zhang — Software system.
- Xieyuanli Chen — Software system.

Fig. 12. The overview of the mechanical structure.

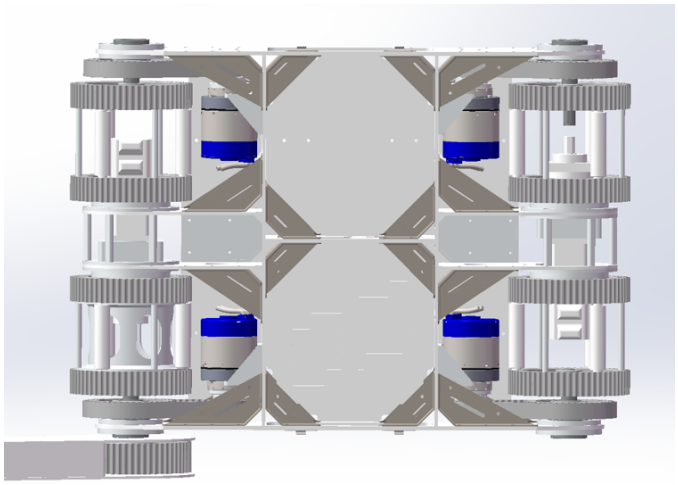


Fig. 13. The top view of the mechanical structure.

Appendix C
CAD Drawings

The mechanical structure of our previously designed robot platform is shown in Figure 12 and Figure 13. Figure 14 shows the CAD drawing of our scalable robot arm.

Appendix D
Lists

A. Systems List

We list the system parameters in Table I. To balance the traversing ability and the portability of the robot, we implemented the lightweight design and ultimately kept the system weight at 50 kilograms. Benefiting from the lightweight design, our robot has a maximum speed of 2 meters per second, and the payload can be more than 60 kilograms.

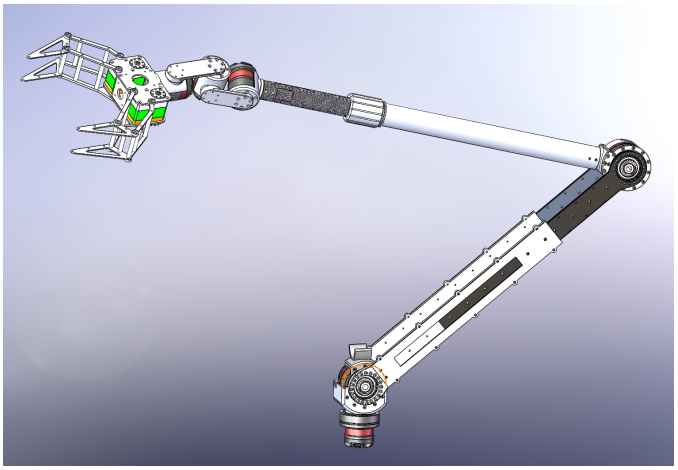


Fig. 14. The CAD drawing of our robot arm.

Fig. 15. Visually guided manipulator for precise operation.

B. Hardware Components List

The components of our Operation station and Robots are listed in Table II and Table III, respectively. The model of our operation station is Lenovo Legion R9000P2021H. This laptop possesses high performance and long battery endurance. Our robot carries six motors, and each motor is responsible for actuating a main track or a flipper. Meanwhile, all necessary sensors for USAR tasks, such as IMU, cameras, and CO_2 sensor, are mounted on the robot.

TABLE I
MANIPULATION SYSTEM

Attribute	Value
Name	NuBot
Locomotion	tracked
System Weight	50kg
Weight including transportation case	55kg
Transportation size	0.7 x 0.6 x 0.3 m
Typical operation size	0.7 x 0.6 x 0.2 m
Unpack and assembly time	60 min
Startup time (off to full operation)	5 min
Power consumption (idle/ typical/ max)	60 / 200 / 400 W
Battery endurance (idle/ normal/ heavy load)	120 / 60 / 30 min
Maximum speed (flat/ outdoor/ rubble pile)	2 / 1 / - m/s
Payload (typical, maximum)	10/ 60 kg
Arm: maximum operation height	2.2 cm
Arm: payload at full extend	2.5kg
Support: set of bat. chargers total weight	2.8kg
Support: set of bat. chargers power	220W (12-24V AC)
Support: Charge time batteries (80%/ 100%)	30 / 60 min
Support: Additional set of batteries weight	2kg
Cost	150,000 RMB

TABLE II
OPERATOR STATION

Attribute	Value
Name	NuBotOp
System Weight	2kg
Weight including transportation case	2.5kg
Transportation size	0.4 x 0.3 x 0.1 m
Typical operation size	0.4 x 0.3 x 0.35 m
Unpack and assembly time	1 min
Startup time (off to full operation)	2 min
Power consumption (idle/ typical/ max)	60 / 80 / 130
Battery endurance (idle/ normal/ heavy load)	5 / 4 / 3 h
Cost	10,000 RMB

TABLE III
HARDWARE COMPONENTS LIST

Part	Brand & Model	Unit Price	Num.
Drive motors	TMOTOR AK10-25	RMB 7000	2
Flipper motors	TMOTOR AK80-64	RMB 12000	4
Batteries	ACE(5300mAh)	RMB 660	4
Computing Unit	ThinkStation P340	RMB 2800	1
IMU	XSENS MTi-300	RMB 18000	1
Cameras	RERVISION USBFHD01M	RMB 260	3
3D LiDAR	RoboSense Bpearl	RMB 15000	1
Infrared Camera	InfiRay T2L-A4L	RMB 1700	1
CO_2 Sensor	SenseAir K30	RMB 600	1
Battery Chargers	SKYRC D100	RMB 600	2

C. Software List

Table IV shows the Software List of our robot system. To adapt to the developing software community, we transplanted our codes from ROS Kinetic to ROS Melodic. We also leverage a new SLAM algorithm [7] to improve the mapping ability.

TABLE IV
SOFTWARE LIST

Name	Version	License	Usage
Ubuntu	18.04	open	
ROS	Melodic	BSD	
PCL [12]	1.7	BSD	ICP
OpenCV-ORB [13]	2.4.8	BSD	Victim detection
LIO-SAM [7]	master	BSD	3D SLAM
NuBot Exploration[3]	1.0	closed source	Autonomous Exploration

ACKNOWLEDGMENT

The authors would like to thank the senior team members Zhiwen Zhang, Junqi Zhang, and Yao Li, who have obtained their Master's degrees and left the team, for their great contributions to the NuBot-Rescue team. Our team is also supported by projects of the National Science Foundation of China (No. 61403409 and No. 61503401).

REFERENCES

[1] X. Chen, H. Zhang, H. Lu, J. Xiao, Q. Qiu, and Y. Li, "Robust slam system based on monocular vision and lidar for robotic urban search and rescue," in *IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, Oct 2017, pp. 41–47.

[2] X. Chen, H. Lu, J. Xiao, H. Zhang, and P. Wang, "Robust relocalization based on active loop closure for real-time monocular slam," in *International Conference on Computer Vision Systems*. Springer, 2017, pp. 131–143.
[3] Y. Liu, Y. Zhong, X. Chen, P. Wang, H. Lu, J. Xiao, and H. Zhang, "The design of a fully autonomous robot system for urban search and rescue," in *IEEE International Conference on Information and Automation*, 2016.
[4] P. Wang, J. Xiao, H. Lu, H. Zhang, R. Yan, and S. Hong, "A novel human-robot interaction system based on 3d mapping and virtual reality," pp. 5888–5894, 2017.
[5] Q. Qiu, X. Chen, Z. Zeng, J. Xiao, and H. Zhang, "Target-based robot autonomous exploration in rescue environments," in *International Conference on Information and Automation*, 2018, pp. 609–614.
[6] W. Deng, K. Huang, X. Chen, Z. Zhou, C. Shi, R. Guo, and H. Zhang, "Semantic rgb-d slam for rescue robot navigation," *IEEE Access*, vol. 8, pp. 221 320–221 329, 2020.
[7] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5135–5142.
[8] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4802–4809.
[9] C. Shi, X. Chen, K. Huang, J. Xiao, H. Lu, and C. Stachniss, "Keypoint matching for point cloud registration using multiplex dynamic graph attention networks," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8221–8228, 2021.
[10] R. Jarvis and J. Byrne, "Robot navigation: Touching, seeing and knowing," in *Proceedings of the 1st Australian Conference on Artificial Intelligence*, vol. 69, 1986.
[11] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Efficient trajectory optimization using a sparse model," *2013 European Conference on Mobile Robots*, pp. 138–143, 2013.
[12] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.
[13] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 1, 2002, pp. I–900–I–903 vol.1.