A Manipulator-based Trace Explosives Detection System for Baggage

Zhiqian Zhou[1], Xiao Li[1], Zhiwen Zeng[1], Huimin Lu[1*], Zongtan Zhou[1] and Zhiqiang Zheng[1]

Abstract— The explosives detection system plays an important role in the counter-terrorism field. For the lack of automated trace explosives detection system, we design a manipulator-based intelligent trace explosives detection system for baggage, named AnBot. As far as we know, it is the first automated trace explosives detection system in China. To avoid reducing the detector's sensitivity and selectivity to explosives, AnBot uses a UR3 manipulator to set up its sampling subsystem. To set up the system well and quickly, we build a ROS-based software framework for AnBot. Finally, the first version of AnBot was designed to accomplish the automated trace explosives detection process by itself. On the one hand, it promotes trace explosives detectors' further widespread use. On the other hand, the proposed system makes it possible to incorporate the trace explosives detection system into the intelligent security network.

I. INTRODUCTION

In the past 20 years, the number of terrorist activities experienced a dramatical increase [1], which becomes a growing headache for every country. According to data released by the Global Terrorism Database (GTD)[1], terrorist bombing is the main form of terrorism. In 2017, the proportion of terrorist bombings in terrorist attacks was more than 50%. In addition, since terrorist bombings are more prone to occur in crowded public places, such as airports and stations, they often cause large casualties to ordinary people. Therefore, it is extremely important to enhance explosives detection in key sites. Generally speaking, the baggage is the most popular choice to convey explosives. Therefore, baggage inspection is the most common and important way to detect explosives. Actually, the development of explosives detection systems and equipment for baggage has become an important research topic in the security field.

Various explosives detectors can be divided into two categories: bulk detection and trace detection. The former, including X-ray imaging [2], nuclear quadrupole resonance (NQR) [3] and neutron techniques [4], is usually used to get the size and shape of suspicious items. However, it can't determine which type the suspicious item is. On the other hand, the latter, including ion mobility spectroscopy (IMS) [5], mass spectroscopy (MS) [6], [7], terahertz spectroscopy [8] and so on, shows very high sensitivity and selectivity to explosives. The high sensitivity and selectivity earns trace detection more attention and use in the field of explosives detection, especially IMS and MS. In 2013, more than 50,000 handheld IMS analyzers and more than 10,000 bench-top analyzers were used as explosives detectors in airports worldwide.

However, most of trace detection devices show a common drawback. All trace explosives detection devices only have the detection function and don't have any active sampling device. On the one hand, it increases the use cost of trace detection devices to a certain extent and limits to its further promotion and application. On the other hand, it is also an obstacle to the intelligent security network, which requires as less manual interventions as possible. Therefore, it is of great value to design an automated explosives detection system.

The critical difference between an automated explosives detection system and a common detector is the automated sampling system. In [9], Fulghum et al. designed a walk-through explosive-trace detection portal for passengers. It collects particles of explosives from the human aerodynamic wake and uses the IMS technique to detect explosives. The system requires $1m^3/s$ airflow, resulting in a relatively low sensitivity. [10] designed an aerodynamic shoe sampling system for trace explosives detection, which faces the similar problem. Yasuaki Takada et al. designed an automated trace-explosives detection system on the basis of MS technique [11], [12]. It uses an air jets to remove the particles adhering to the detection surface and collects them with a cyclone concentrator. Though it is really a great work, it still stays in the laboratory until now. To design an automated sampling system is still a problem.

Meanwhile, various manipulators are used in different tasks, including sampling, picking and placing, machine tending and so on. In DARPA Robotics Challenge (DRC)[2], some missions are related closely to the robotic arm. The main reason for the popularity of manipulators is that they always show great flexibility, high stability and excellent accuracy. Besides, the motion planning problem for low dimensional manipulator has been well solved. An open source project named Optimal Motion Planning Library (OMPL) [13] provides various sampling-based motion planning algorithms for manipulators. A motion planning library named Moveit! is built for any ROS-supported robot[3]. These open-source

*This work is supported by National Key R&D Program of China (No.YFC20170806503).

1. Department of Automation, National University of Defense Technology, Changsha 410073, P. R. China
E-mail: lhmnew@nudt.edu.cn

[1] http://www.start.umd.edu/gtd

[2] http://drc.mit.edu/index.html

[3] https://moveit.ros.org/

projects make it possible to apply manipulators easily and quickly in various applications.

Though the manipulator is so popular, we could not find any case integrating the manipulator with explosives detection. The advantage of using manipulator for explosives sampling is obvious. It is similar with the manual sampling process, and the manipulator sampling system will never reduce the detector's sensitivity and selectivity to explosives. Therefore, we think it is a meaningful and practical work to design a manipulator-based trace explosives detection system for baggage. Now, the first version has been designed and is named AnBot.

This paper is organized as follows. In Section II, the overall architecture of AnBot is depicted. Section III describes the mechanical design of AnBot, including its static base and dynamic sampling subsystem. Then, Section IV introduces its software based on ROS, including some critical techniques. Section V concludes the whole paper and Section VI summarizes the future work.

II. Overall Architecture

Fig.1 shows the overall architecture of AnBot. The whole system includes five subsystems with different color borders. A brief introduction for each subsystem is provided in the following text.

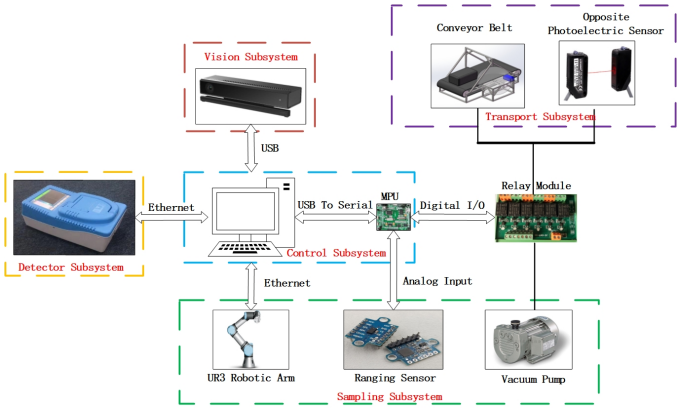


Fig. 1: The overall architecture of AnBot.

The control subsystem is made up of a PC and a microprocessor unit (MPU). The computer is the main processor of the whole system and does the main calculations including the processing of visual information and the planning of the robotic arm. The MPU completes some simple tasks. It reads the data from the ranging sensor, obtains the state of the conveyor belt and opposite photoelectric sensors and sends control commands to the conveyor belt and the vacuum pump. At last, we choose Robot Operating System (ROS) to build up AnBot's software.

The sampling subsystem includes 3 components: a robotic arm named UR3, a ranging sensor and a vacuum pump. The UR3 is an industrial robotic arm from UNIVERSAL ROBOTS[4]. A ranging sensor is fixed on the end of UR3 to determine the distance to the target surface. Besides, there is a vacuum pump equipped with a vacuum sucker to suck the sampling paper. The realization of sampling will be introduced in section III.

The vision subsystem is built to detect the sampling point on the baggage surface and determine its 3D position. The sampling point is transformed to the target point for the sampling subsystem later. In our vision subsystem, a Kinect v2 camera is used to get the 3D information of the environment.

The transport subsystem is built based on a conveyor belt. The conveyor belt is driven by a motor and a frequency converter. By sending "start" signal and "stop" signal to the frequency converter, we can control the start and stop of the conveyor belt. Meanwhile, two pairs of opposite photoelectric sensors are fixed on both sides of the conveyor belt to locate baggage on the belt.

The detector subsystem includes an explosives detector. A ZA-800BX from Shenzhen ZOAN GAOKE Electronics Company is chosen for AnBot[5]. It has the ability to verify various kinds of military, civil and indigenous explosives, such as nitroglycerine nitrification, ammonium nitrate black powder, ammonium nitrate black powder. For our demand to build up an intelligent system based on a PC, the detector is added with basic interfaces with computers.

III. Mechanical Design

This section describes the mechanical design of AnBot. When designing the platform, we consider two criteria. On the one hand, the whole system should meet the rules and regulations of security check, namely its size, high reliability and so on. On the other hand, the sampling subsystem should comply with the operating specification of the explosives detector.

A. Static Base

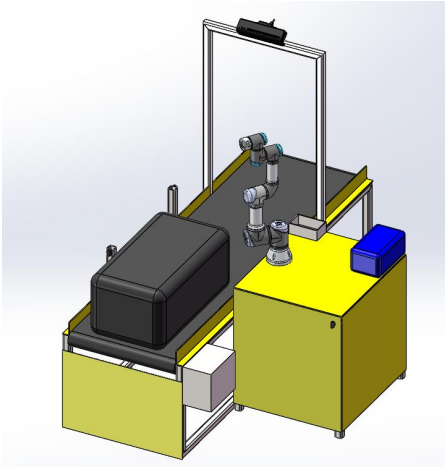


Fig. 2: The 3D model of AnBot.

As shown in Fig.2, the static base is based on a conveyor mechanism with the length of 2.0m, the width of 0.6m and the height of 0.53m. Behind the conveyor belt,

[4] http://www.universal-robots.cn/site/products-ur3-robot

[5] http://www.szzoan.com/ProductDetail/1360401.html#

there is enough space to contain various equipment with a large size, such as the vacuum bump, the controller box of UR3 and the computer mainframe. A worktop is fixed firmly at the right side of the conveyor belt. Now, the explosives detector, the UR3 and the sampling paper container are fixed on the worktop. Besides, there are two pairs of opposite photoelectric sensors in the system to locate baggage on the conveyor belt. It is obvious that the available sampling area lies in front of the UR3. Hence, two transmitters are fixed on the left side of the conveyor belt and two receivers are fixed at the leg of the worktop. Another non-ignorable component of the static base is a simple but firm gantry, on which a RGB-D camera is set. In order to ensure that the camera's field of view covers the target area, the camera is tilted with a certain angle. Fig.3 is an image from the Kinect v2 camera when the baggage locates in the given area.

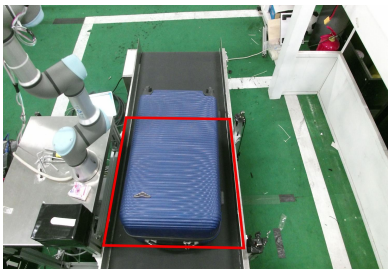


Fig. 3: The field of view of the Kinect camera and the borders of the ideal sampling area are painted red.

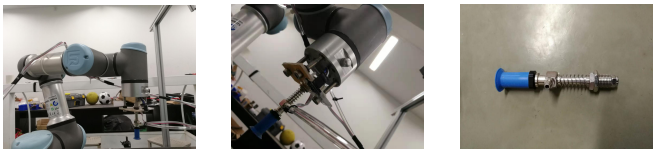
B. Dynamic Sampling Subsystem

The sampling subsystem is designed to sample on the baggage surface. The standard manual sampling includes 3 steps: picking up a piece of sampling paper, wiping the baggage surface with the sampling paper and sending the paper into the explosives detector. Fig.4 shows an explosives detector and its dedicated sampling paper with the width of $18mm$ and the length of $58mm$. To accomplish this mission, we design the peculiar subsystem, which is shown in Fig.5. The UR3 arm is its base frame and completes its motion function. It has six joints and its motion range is nearly $500mm$. A vacuum sucker and a ranging sensor are fixed on a board on the UR3's end. The vacuum pump is used to create a vacuum environment in the vacuum tube. The other end of the vacuum tube is the vacuum sucker. With the low pressure in the vacuum tube, the sucker is able to suck the paper. It is able to separate a piece of sampling paper from a pile of sampling paper. Another critical component of the sucker is the spring, which enables the sucker to deform along the spring. The ranging sensor senses the distance from the board to the target surface, which is used to judge if the sucker gets in touch with the target surface.

Though the AnBot's sampling shares the same process with the manual sampling, there is a micro improvement to protect the sampling subsystem when the sucker touches the sampling paper surface or the baggage surface. At first, the sucker moves right above the target point. Secondly, the sucker is pressed vertically to the point. Once the distance is less than a given threshold, the catch action or the touch action is completed. Finally, the sucker leaves vertically from the surface.

(a) (b)

Fig. 4: (a) An explosives detector and (b) its sampling paper.



(a) (b) (c)

Fig. 5: The sampling subsystem: (a) the UR3 robotic arm, (b) the improved end and (c) the vacuum sucker.

IV. Software Based on ROS

Since this is the first version of AnBot and we want to promote AnBot to be applied in various sites, it is extremely important to design a highly modular software framework for convenient improvements. Considering the hardware, the software should provide convenient interface with the UR3 manipulator. Therefore, ROS becomes the first choice for its high-modularity. Fig.6 shows the software framework of AnBot, which is divided into 3 main parts: the Kinect Camera Node and the Vision Node; the Control Node; the UR Driver Node and the Motion Plan Node. Except from the five nodes, a node is created for prior hand-eye calibration. They will be detailed in the following sub-sections.

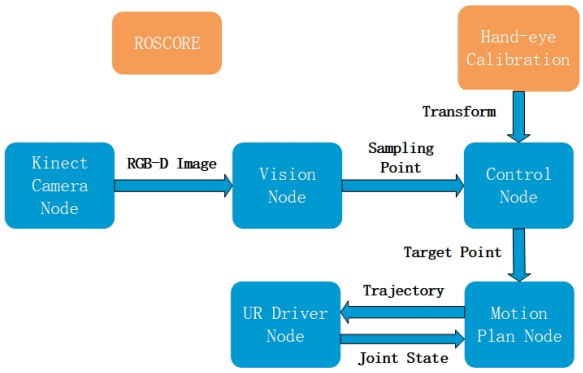


Fig. 6: The software framework based on ROS.

A. Vision Node

The Vision Node is built to find a proper sampling point for the robotic arm. Obviously, the 3D information of the point is needed. Therefore, a Kinect v2 camera is selected as the "eye" of AnBot. The Kinect Camera Node drives the camera and provides point cloud data for the Vision Node. Next, we will explain how we get the sampling point quickly and accurately.

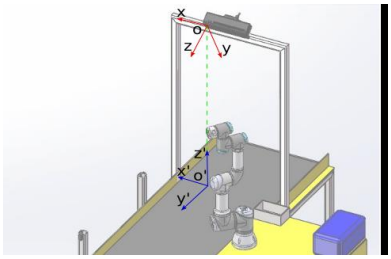


Fig. 7: The camera coordinate is painted red and the conveyor belt coordinate is painted blue.

The vision algorithm mainly includes three stages: down-sampling, segmentation and plane detection. The first stage is to perform spatial down-sampling of point cloud data. The data is divided into many 3D grids of $5mm \times 5mm \times 5mm$. For every 3D grid, its position is the mean of all point clouds contained by it. To avoid background interference, the second stage is to segment the ideal sampling area. As mentioned in Section III.A, the sampling area lies on the conveyor belt and the camera is fixed on the gantry. A feasible and simple way is to segment point clouds according to their position information. With given fixed inclination angle and height, it is possible to transform point clouds from the camera coordinate to the conveyor belt coordinate, which are shown in Fig.7. With given threshold of coordinates, these point clouds right above the belt are retained. Later, these selected point clouds are inversely converted to the camera coordinate. Then, since our system focus its eyes on suitcases, they always have a flat surface. Therefore, we use the RANSAC algorithm [14], a plane detection algorithm, to search the target surface. With the detected plane and a given motion range of UR3, a sampling point can be determined easily in the camera coordinate. Fig.8 shows the result of our vision algorithm and its average processing time is nearly 0.1484s. The test is carried out on a laptop with the 8th generation Core i7 as its processor.

(a) (b)

Fig. 8: The result of our vision algorithm: (a) the original RGB image (b) the detected baggage surface.

B. Motion Plan Node

The Motion Plan Node aims to plan a safe and collision-free trajectory for the sampling subsystem. Considering Moveit! collects most motion planning algorithm and provides convenient interfaces with the UR3 manipulator, AnBot chooses Moveit! as its motion planning platform.

In practice, the system has a simple usage environment. Most of obstacles are static and only the baggage is uncertain. Furthermore, the baggage also has a regular size and shape and keeps still during the sampling process. It means that we could avoid collisions with static obstacles by improving the manipulator model. As shown in Fig.9, the manipulator model is added with the operator board, the detector, the sampling paper container, the gantry and part of the conveyor belt.

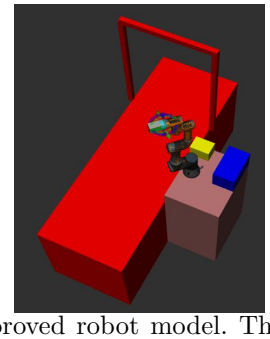
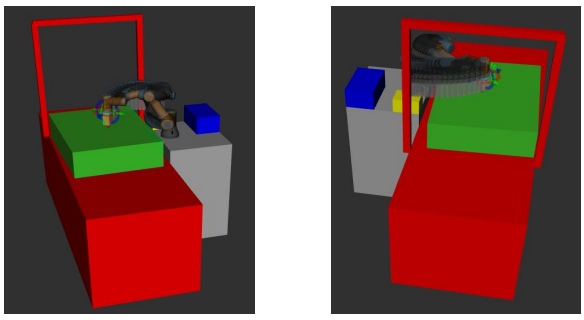


Fig. 9: The improved robot model. The blue box is an explosives detector, the yellow box is a sampling paper container and the red box is the transport belt.

As for the uncertain baggage, its approximate size can be easily determined by the vision subsystem. With an existing interface from Moveit!, it is practical to insert an approximate baggage model into the planning scene of AnBot. Once the sampling process ends, the approximate baggage model could also be removed from the scene.

Another important component of Moveit! is its motion planner. Until now, it has integrated a variety of motion planning algorithms, including sampling-based motion planning algorithms in OMPL. Due to the low failure rate and a relatively short planning time of RRT Connect algorithm [15], it is used in the Motion Plan Node. For it is a sampling based motion planning algorithm, its planned trajectories are different from each other. Fig.10 shows a trajectory from the sampling paper container to the sampling point.



(a) (b)

Fig. 10: The planned trajectory from two angles of view. The green box is an approximate model of baggage.

C. Hand-eye Calibration

To link the vision subsystem and the sampling subsystem, hand-eye calibration is an indispensable and extremely important work. The aim of hand-eye calibration is to obtain the coordinate transformation from the camera coordinate to the base coordinate, which is used to transform the sampling point to the target point.

As shown in Fig.11, the hand-eye calibration system is made up of three parts: the Kinect v2 camera, the UR3

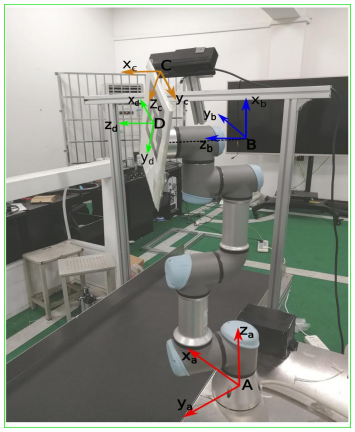


Fig. 11: Four coordinates in the hand-ere calibration system. The base coordinate is painted red, the end coordinate is painted blue, the camera coordinate is painted yellow and the board coordinate is painted green. To show four coordinates clearly, we move the end coordinate along the dashed line.

manipulator and a calibration board. The calibration board is fixed at the end of the manipulator. The calibration board is from an open-source library named ArUco, which provides convenient solutions to detect its markers. Besides, the library creates a ROS package named "aruco_ros" as its interface with ROS. There are four coordinates in the system. The first one is the base coordinate with its origin at the center of UR3's base link. The second coordinate is the end coordinate, which takes the center of the end as its origin. The third one is the camera coordinate and the last one is the board coordinate. They are represented by A,B,C and D respectively and the transformation matrix from c_1 to c_2 is represented by $T_{c_1c_2}$, where c_1 and c_2 is the symbol of above four coordinates. For example, T_{AB} is the transformation from the base coordinate to the end coordinate. According to the definition of transformation matrices, we could get equation (1).

$$T_{c_1c_3} = T_{c_1c_2}T_{c_2c_3}; \tag{1}$$

Therefore, the transformation matrix from the end coordinate to the board coordinate can be described as:

$$T_{DB} = T_{DC}T_{CA}T_{AB}; \tag{2}$$

In equation (2), T_{DB} and T_{CA} are unknown, and T_{AB} could be calculated with the joint states. As for T_{DC}, it is able to be computed with the "aruco_ros" package.

If the manipulator has different joint states at m moments, represented by $t_i, i = 1, 2, ..., m$, we use a subscript to distinguish the same coordinate at different moment. For example, A_1 is the base coordinate at t_1 and B_m denotes the end coordinate at t_m. Assuming the calibration board is fixed firmly enough, the transformation matrix from the end coordinate to the board coordinate is the same for m moments. On the basis of (2), (3) is deduced easily.

$$T_{D_iC_i}T_{C_iA_i}T_{A_iB_i} = T_{D_jC_j}T_{C_jA_j}T_{A_jB_j}, i, j = 1, 2, .., m; \tag{3}$$

For the transformation matrix is non-singular, (3) could be written as:

$$T^{-1}_{D_jC_j}T_{D_iC_i}T_{C_iA_i} = T_{C_jA_j}T_{A_jB_j}T^{-1}_{A_iB_i}, i, j = 1, 2, .., m; \tag{4}$$

Then, because T_{CA} is our optimization objective, T_{AB} and T_{DC} are known, (4) becomes a typical equation in the form of $AX = XB$. [16] provide practical ways to solve these equation. The transformation matrix is represented by a translation vector T ($T = (x, y, z)$) and $Z - Y - X$ euler angles (α, β, γ). For it is hard to obtain the accurate transformation matixa, we repeat the hand-eye calibration for 7 times and the statistical results are shown in Table 1. The unit of length is meter (m) and the unit of angle is radian (rad).

TABLE I: Statistical results of hand-eye calibrations.

Para	$x(m)$	$y(m)$	$z(m)$	$\alpha(rad)$	$\beta(rad)$	$\gamma(rad)$
mean	-0.785	-0.008	1.004	-0.596	0.014	-0.778
std.	0.008	0.016	0.014	0.006	0.008	0.012

D. Control Node

The Control Node aims to control the whole detection process, which is depicted in Algorithm 1.

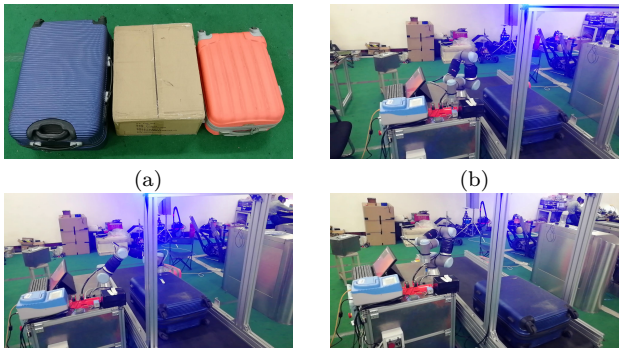
Algorithm 1

```
repeat
    repeat
        run the conveyor belt
    until the baggage lies in the ideal sampling area
    stop the conveyor belt
    sample at the baggage
    detect explosives
until there are explosives in the baggage
```

At first, the transport subsystem conveys the baggage into the ideal sampling area. In this step, two pairs of opposite photoelectric sensors are used to locate the baggage. Once a piece of baggage arrives the sampling area, the conveyor belt is stopped and the system turns into the second stage. The vision subsystem detects the practical sampling point on the baggage surface based on the results of prior hand-eye calibration. At this time, the Motion Plan Node plans a feasible and collision-free trajectory to the target point for the sampling subsystem. The third stage is to detect explosives with the explosives detection subsystem. After receiving the detection command, the detector detects whether there is an explosive on the sampling paper and returns the result to the PC. The whole system will run continually until it detects suspicious baggage. Three pieces of baggage with different size are tested in our experiments. These experiments show AnBot is able to complete explosives detection for baggage by itself. Some videos could be found at https://youtu.be/MyIGnlqlnn0.

V. CONCLUSIONS

In summary, we presented the whole design of our manipulator-based automated explosives detection system with its overall architecture, mechanical platform,



(a) (b) (c) (d)

Fig. 12: (a) Three pieces of baggage, (b) AnBot sucks a piece of sampling paper, (c) AnBot samples on the baggage surface and (d) Anbot removes the baggage after explosives detection.

and software framework based on ROS in this paper. The core of the system is its sampling subsystem based on a UR3 manipulator. A non-negligible advantage of the system is that it will not reduce the detector's sensitivity and selectivity. To accomplish the automated explosives detection procedure, we solved some key problems such as baggage recognition, hand-eye calibration and manipulator motion planning. Finally, the first version of AnBot was designed and the experimental results show that it is potential to be used in the real scene.

As far as we know, the system is the first automated explosives detection system in China. It integrates the manipulator with trace explosives detection. We expect this work to be valuable in the explosives detection field. On the one hand, the unmanned process of the entire process can reduce the cost of using trace explosives detectors and promote the further widespread use of trace explosives detectors. On the other hand, the intelligent system can initially realize the unmanned detection of trace explosives. It means that it is possible to incorporate the trace detection of explosives into the intelligent security network, which is of significance in the security field.

VI. FUTURE WORK

As the first version of our intelligent explosives detection system, AnBot has some obvious shortcomings. First of all, the explosives detector only owns basic control interfaces with the computer, because the current trace explosives detector is an existing commercial product. The detector is not initially developed for our automated detection system. In the future, we will work with the manufacturer to develop a trace explosives detector that is more suitable for automatic detection in the actual application process. Secondly, in terms of the robotic arm, the current moving range of the manipulator is limited, and the existing motion blind zone is large. In the future, we will try to use a manipulator with larger size. Finally, in terms of sampling point selection, the current method is to randomly select sampling points within the surface of the baggage and the moving range of the robotic arm. However, the handle of the suitcase is more likely to contact explosives. Therefore, we will try to use the machine learning method to identify and locate the handle of the baggage or other special positions, and use them as sampling points.

References

[1] A. M. Lopes, J. A. T. Machado, and M. E. Mata, "Analysis of global terrorism dynamics by means of entropy and state space portrait," Nonlinear Dynamics, vol. 85, no. 3, pp. 1547–1560, 2016.

[2] N. C. Murray and K. Riordan, "Evaluation of automatic explosive detection systems," in Institute of Electrical & Electronics Engineers International Carnahan Conference on Security Technology, 1995.

[3] A. N. Garroway, M. L. Buess, J. P. Yesinowski, J. B. Miller, and R. A. Krauss, "Explosives detection by nuclear quadrupole resonance (nqr)," in Cargo Inspection Technologies, 1994.

[4] B. Király, L. Oláh, and J. Csikai, "Neutron-based techniques for detection of explosives and drugs," Radiation Physics & Chemistry, vol. 61, no. 3, pp. 781–784, 2001.

[5] H. Borsdorf and G. A. Eiceman, "Ion mobility spectrometry: Principles and applications," Applied Spectroscopy Reviews, vol. 41, no. 4, pp. 323–375, 2006.

[6] M. Christopher, I. Amos, B. V. Pond, D. L. Huestis, M. J. Coggiola, and O. Harald, "Detection of explosives and explosives-related compounds by single photon laser ionization time-of-flight mass spectrometry," Analytical Chemistry, vol. 78, no. 11, pp. 3807–3814, 2006.

[7] C. R Graham, O. Zheng, T. Zoltan, and J. M. Wiseman, "Detection technologies. ambient mass spectrometry," Science, vol. 311, no. 5767, pp. 1566–1570, 2006.

[8] M. R. Leahy-Hoppa, M. J. Fitch, and O. Robert, "Terahertz spectroscopy techniques for explosives detection," Analytical & Bioanalytical Chemistry, vol. 395, no. 2, pp. 247–57, 2009.

[9] M. R. Fulghum, M. J. Hargather, and G. S. Settles, "Integrated impactor/detector for a high-throughput explosive-trace detection portal," IEEE Sensors Journal, vol. 13, no. 4, pp. 1252–1258, April 2013.

[10] M. Staymates, G. Gillen, J. Grandner, and S. Lukow, "Design and characterization of an aerodynamic shoe sampling system for screening trace explosive materials," in Aps Division of Fluid Dynamics Meeting, 2011.

[11] Y. Takada, H. Nagano, Y. Kawaguchi, H. Kashima, M. Sugaya, K. Terada, Y. Hashimoto, and M. Sakairi, "Automated trace-explosives detection for passenger and baggage screening," IEEE Sensors Journal, vol. 16, no. 5, pp. 1119–1129, 2016.

[12] H. Yuichiro, N. Hisashi, T. Yasuaki, K. Hideo, S. Masakazu, T. Koichi, and S. Minoru, "Real-time explosive particle detection using a cyclone particle concentrator," Rapid Communications in Mass Spectrometry, vol. 28, no. 12, pp. 1376–1380, 2014.

[13] I. A. Sucan, M. Moll, and E. E. Kavraki, "The open motion planning library," IEEE Robotics & Automation Magazine, vol. 19, no. 4, pp. 72–82, 2012.

[14] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," Computer Graphics Forum, vol. 26, no. 2, pp. 214–226, 2010.

[15] 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, April 2000, pp. 995–1001 vol.2.

[16] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," IEEE Transactions on Robotics and Automation, vol. 5, no. 3, pp. 345–358, June 1989.