

Previous Research Experience

My research aims to address the question, '**How we might leverage advances by combining the reactivity of visual servoing with the generalization capability of machine learning to solve dynamic mobile manipulation tasks, such as dynamic grasping or handover tasks?**'. The definition of visual servoing is using visual data as input of real-time closed-loop control methods for controlling the motion of a dynamic system [1]. Even visual servoing is a mature area with many mobile and robot arms applications. However, designing new control strategies using machine learning is another direction for improvements on this topic, especially in mobile manipulators. My past research has helped to answer partially this question by implementing Position based Visual Servoing (PBVS) with dual 6-DOF robotics arms [2], and I have hands-on experience in some robotics grasping models such as Generative Grasping CNN (GG-CNN) [3] and Generative Residual Convolutional Neural Network (GR-ConvNet) [4], as well as point cloud data processing with DBSCAN and K-mean unsupervised clustering methods [5]. Moreover, I also develop a mobile robot platform called MiniROS by using ROS for education and research purposes [6]. In general, I am interested in exploring the intersection of machine learning and classical visual servoing to solve mobile manipulator dynamic tasks.

Details of my previous research work are available in my CV, but I would like to highlight my interest in the intersection of machine learning and visual servoing technique for solving dynamic mobile manipulator problems. The PBVS task I have implemented on dual robotics arms is preliminary work, and I will continue to eagerly continue pursuing and developing this control method by combining machine learning methods. I am also interested in continuing to research more in 3D vision by developing more efficient real-time machine learning robotics grasping models.

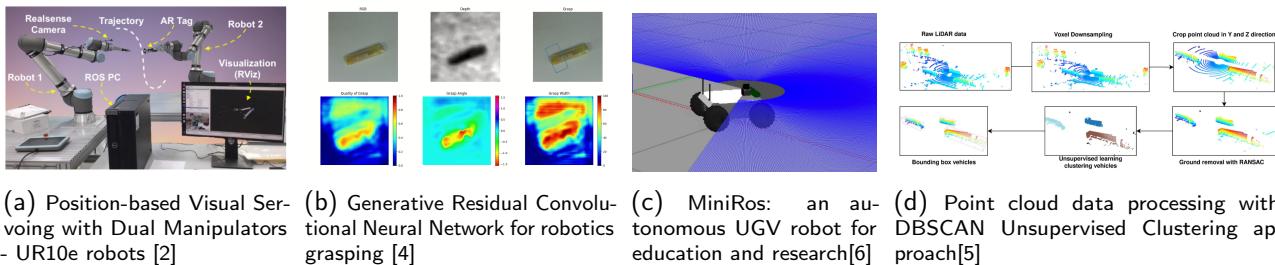


Figure 1: Our work on developing Position-based visual servoing based on ROS (a), evaluation of Convolutional Neural Network grasping models (b), developing for a wide range of mobile robot tasks using ROS (c) and point cloud data processing with different approaches (d).

Research Goals

One of the keystone of robotics is the dynamic mobile manipulation capability, which addresses the uncertainty of relative motions, incentives, and behaviors between robots, humans, and the environment. My expertise in robotics manipulators and mobile robots provides me with a solid foundation of develop new research in dynamic mobile manipulation tasks. My long-term research goal is to develop the methodology for robust control for real-time dynamic tasks with autonomous mobile manipulators. Solving dynamic tasks with a mobile manipulator introduces unique challenges in both reactive motion generation [7] and robot perception [3], [4],[8]. The sequential control architecture for mobile manipulation is limited to dynamic tasks in speed and gracefulness for example the mobile base must stop moving before arm motion activate, the need for reactive visual servoing control architecture should be capable of a broad of robot designs. Robustness and natural interaction can be enhanced by using a reactive visual servoing control approach for the robot arm and mobile base. From the robot perception perspective, most of the current research focuses on robot grasping in a stable environment on a robot manipulator, and it is unclear how the mobile manipulator can perform

grasping in dynamic tasks. My work contributes to overcoming these challenges by 1) developing a novel reactive visual servoing control technique based on visual features for mobile manipulators. 2) developing a novel grasping machine learning model that can integrate the visual servoing to grasp dynamic objects or handover tasks 3) developing a novel 3D vision model combine with Deep RL or differentiable motion planning for learning dual-arm mobile manipulation dynamic tasks.

Research Plan

The research will apply the following stages and methodologies:

- **Stage 1. Reactive visual servoing control for mobile manipulator.** Development of a novel reactive control approach based on visual features, it is able to steer the mobile manipulator robot towards the object of interest even in attendance of significant distraction in the background and obstacle avoidance.
- **Stage 2. Visual servoing CNN-based dynamic grasping.** Compare and evaluate different grasping machine learning models such as Generative Grasping CNN (GG-CNN) [3], Generative Residual Convolutional Neural Network (GR-ConvNet) [4] and ORientation AtteNtive Grasp synthEsis (ORANGE) [8]. Development of a novel grasping machine learning model can integrate the visual servoing based on Stage 1 to grasp dynamic objects or handover tasks. Apply this algorithm to a mobile manipulator such as a TIAGO robot. The real Cornell, Jacquard, EGAD! grasp datasets and a new synthetic grasping dataset will be created for benchmarking.
- **Stage 3. 3D vision for dual-arm mobile manipulator.** This stage aims to solve the more challenging dynamic tasks of dual-arm mobile manipulation. From the developed algorithm in Stage 2, development a novel 3D vision method and combine them with robot learning algorithms [9] to learn using real-world, imperfect 3D information such as point clouds. Evaluate 3D information processing neural networks such as PointNets [10] and among others then integrate them into novel Deep RL or differentiable motion planning algorithms for learning dual-arm mobile manipulation dynamic tasks.

In conclusion, I love developing algorithms that are simple, efficient, providing solid theoretical guarantees, and are beneficial in solving real-world dynamic problems with mobile manipulators.

References

- [1] F. Chaumette, "Visual servoing," in *Encyclopedia of Robotics*, M. H. Ang, O. Khatib, and B. Siciliano, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020, pp. 1–9. DOI: 10.1007/978-3-642-41610-1_104-1.
- [2] B. M. Tri, "Position-based visual servoing with dual manipulators," in *Preprint*, 2022. [Online]. Available: <https://triknight.github.io/images/fulls/PBVS-UR.pdf>.
- [3] D. Morrison, J. Leitner, and P. Corke, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," Jun. 2018. DOI: 10.15607/RSS.2018.XIV.021.
- [4] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9626–9633. DOI: 10.1109/IROS45743.2020.9340777.
- [5] B. M. Tri and V. B. Hien, "Lidar-based vehicle detection by using dbscan unsupervised clustering approach," in *6th International Conference on Control, Robotics and Informatics (ICCRI)*, 2023. [Online]. Available: https://triknight.github.io/images/fulls/Tri_2023_PrePrint.pdf.
- [6] B. M. Tri, H. Thanh Luan, D. X. Phu, T. Quang Nhu, and B. M. Duong, "Miniros: An autonomous ugv robot for education and research," in *2021 International Conference on System Science and Engineering (ICSSE)*, 2021, pp. 170–175. DOI: 10.1109/ICSSE52999.2021.9538463.
- [7] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1371–1394. DOI: 10.1007/978-3-540-30301-5_60.
- [8] C. Georgia, G. Nikolaos, M. Petros, and P. Jan, "Orientation attentive robotic grasp synthesis with augmented grasp map representation," in *Preprint*, 2021. [Online]. Available: <https://arxiv.org/abs/2006.05123>.
- [9] J. Peters, D. D. Lee, J. Kober, D. Nguyen-Tuong, J. A. Bagnell, and S. Schaal, "Robot learning," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Cham: Springer International Publishing, 2016, pp. 357–398. DOI: 10.1007/978-3-319-32552-1_15.
- [10] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85. DOI: 10.1109/CVPR.2017.16.

Tri Bien Minh

Portfolio: triknight.github.io
Github: github.com/triknight

Email: nvn.bienminhtri@gmail.com

Research Gate: Tri-Bien

Address: VietNam

EDUCATION

- **Karlsruhe University of Applied Sciences (Collaboration with the Vietnamese German University - VGU)**
• *M.Sc in Mechatronics System and Sensor Technology; GPA: 1.6* 2017
 - Thesis: *Design, Modeling and Control an Octocopter (Grade : 1.0 - Excellent)*
 - 100% Full tuition scholarship

- **Lac Hong University** Vietnam
• *B.Sc in Mechatronics Engineer; GPA: 7.97/10.0 (Top 5% students in class)* 2013
 - Team leader a university robot team in ABU Robocon, a robotic competition for Asia pacific universities from 2011-2013
 - Second prize in Nation Robocon Techshow competition with project Humanoid personal assistant robot in 2012

EXPERIENCE

- **Vietnamese German University (Collaboration with Frankfurt University of Applied Sciences)** Vietnam
• *Robotics Lab Engineer (Full-time)* Oct 2017 - Present
 - **Managing VGU's Robotics Lab & support research activities:** maintaining and managing laboratory equipment (Robot UR10e, Kuka Youbot, Turtlebot3, Realsense, Velodyne Lidar,...), materials, and computer systems through regular service and repair. Work on assigned research projects in the fields of Autonomous Robotic Systems, Computer Vision, Embedded Systems, and Machine Learning. Some projects are: developing and maintaining ROS packages for the autonomous mobile robot Turtlebot3, Youbot and MiniROS, position-based visual servoing, and integrating MoveIt as a motion planner for the UR10e robot arm. Create a simulation environment UR10e robot in Unity, and Robotic Toolbox.
 - **Research on the intersection of Machine Learning and Visual Servoing:** for solving dynamic robot problems, such as dynamic grasping and handover tasks. Implementing Position-based Visual Servoing (PBVS) with dual 6-DOF robotics arms. Execute and develop Machine Learning models for object detection, and grasping with various input data like RGB-image and point-cloud. I am also interested in continuing to research more in 3D vision by developing more efficient real-time machine learning robotics grasping models.
 - **Lab tutorial & supervise undergraduate students:** Collaborate with Prof. Dr. Peter Nauth and VGU Lecturers to prepare Lab tutorials: Embedded Intelligent System (ROS, OpenCV), Robotics and Autonomous Systems (ROS, Pytorch), Smart Systems in Automation (Python, UR PolyScope), Microcontroller (Atmel Studio), Digital Signal Processing (MatLab), Robotics Workshop (CAD and PCB Design) and supervising/co-supervising undergraduate students in robotics projects.

- **Nguyen Tat Thanh University** Vietnam
• *Lecture of Mechatronic Department* Nov 2013 - Jun 2017
 - **Prepared & delivered lectures to undergraduate students:** on topics of mechatronics and robotics.
 - **Designed robots, machines & teaching kit for education purposes:** Upper body humanoid robot (14-DoF), Ant-like robot (23 DoF), RC Humanoid robot (19 DoF), PLC-Modular Production Station, 3-Axes CNC Machine.
 - **Administration work:** monitored undergraduate teaching, internship, and supervised robotics projects and machine designed for undergraduate students.

- **Robert Bosch Engineering and Business Solutions** Vietnam
• *Intern. Mechanical Engineer* Feb 2016 - Aug 2016
 - **Designed the charger docking and locking mechanism for the electric motorbike:** in the "Bosch Green Challenge project", and got awarded "Certification of Innovation Activities and Development" for this design.
- **Pepperl and Fuchs Co., Ltd.** Vietnam
• *Intern. Process Engineer* Oct 2015 – Dec 2015
 - **Implemented PDCA (Plan-Do-Check-Action) process:** for ultrasonic welding sensors, and improvement of quality sensors in the manufacturing process. Designed a new kind of machine, and planned some automation processes.

PUBLICATIONS

- **Position-based Visual Servoing with Dual Manipulators (ongoing project).** Tri B. Minh*, PrePrint
- **LiDAR-based Vehicle Detection by using DBSCAN Unsupervised Clustering approach (accepted).**
Tri B. Minh*, Hien Vo Bich, 2023 6th International Conference on Control, Robotics and Informatics (ICCR 2023) PrePrint,
- **Development of a novel V-frame Octocopter: Design, Kinematic Analysis, and Simulation using PID controllers with Ziegler Nichols tuning method.** Tri B. Minh*, Hien Vo, Hua Thanh Luan, *International Journal of Intelligent Unmanned Systems* 2023 DOI: 10.1108/IJIUS-08-2021-0087,
- **MiniRos: an autonomous UGV robot for education and research.** Tri B. Minh*, H. Thanh Luan, D. X. Phu, T. Quang Nhu and B. M. Duong, 2021 International Conference on System Science and Engineering (ICSSE) pp. 170-175, DOI: 10.1109/ICSSE52999.2021.9538463,

- **Robot Gesture Control Using Online Feedback Data with Multi-Tracking Capture System.** Khang Hoang Vinh Nguyen, Tri Bien Minh, Van Chi Le and Phu Xuan Do *The 7th International Conference on Advanced Engineering - Theory and Applications AETA 2022 pp. 121-130, ISBN 1876-1119*
- **Adaptive Optimal Control for Upper Exoskeleton following Saturation Function.** Do Xuan Phu, Tri B. Minh, 2021 *24th International Conference on Mechatronics Technology (ICMT)*, DOI: 10.1109/ICMT53429.2021.9687228

ACADEMIC REFEREES

- **Prof. Dr. Peter Nauth:** Professor of Computer Engineering and Robotics, Frankfurt University of Applied Sciences, Frankfurt am Main, Germany — email: pnauth@fb2.fra-uas.de — [Personal website](#)
- **Associate Prof. Do Xuan Phu:** Associate Professor of Mechatronics and Sensor Systems Technology, Vietnamese-German University, Binh Duong, Vietnam — email: phu.dx@vgu.edu.vn — [Personal website](#)
- **Dr. Vo Bich Hien:** Senior lecturer of Department Electrical and Computer Engineering Vietnamese-German University, Binh Duong, Vietnam, email: hien.vb@vgu.edu.vn — [Google Scholar](#)

HONORS AND AWARDS

- Best Junior Researcher Award in Vietnamese German University - Academic year, 2020-2021
- 100% full tuition scholarship (Pepperl+Fuchs scholarship) - in Master course, 2015-2016
- Global Entrepreneurship Training under the Global Entrepreneurship Education Program (GEEP), 2017
- Youth exchange JENESYS 2.0 Scholarship (JICA 2014) - Japan, 2014
- Second prize in Nation Robocon Techshow competition with project Humanoid personal assistant robot, 2012

SKILLS SUMMARY

- **Programming:** Python, C++, MatLab
- **Frameworks:** ROS, Pytorch, TensorFlow, OpenCV, Open3D, Isaac Sim, OpenAI-Gym
- **Tools:** Software (Git, Docker), PCB Design(KiCad), 3D CAD Design(Solidworks)
- **Platforms:** MacOS, Linux, Windows, Arduino, Nvidia-Jetson, Raspberry Pi
- **Languages:** English: Professional Working Proficiency, Vietnamese: Native
- **Soft Skills:** Leadership, Event Management, TeamWork, Writing, Time Management

CERTIFICATE

- | | |
|--|-----------------------------------|
| • TensorFlow for Artificial Intelligence, Machine Learning, and Deep Learning | Coursera
Nov 2021 |
| • <i>Credential ID: C6WDSPX7BKVH</i> | |
| • Convolution Neural Network in TensorFlow | Coursera
Nov 2021 |
| • <i>Credential ID: JFFHZFB8QZEF</i> | |
| • Natural Language Processing in TensorFlow | Coursera
Nov 2021 |
| • <i>Credential ID: PRNTD5GJ9G5C</i> | |
| • SIMATIC S7-1500 Programming 1 in The TIA Portal (TIA-PRO1) | Siemens
Oct 2020 |
| • <i>Programming PLC S7-1500 with TIA Portal</i> | |
| • Deep Reinforcement Learning NanoDegree | Udacity
May 2020 |
| • <i>Credential ID: 466QEDKQ</i> | |
| • Certification of Innovation Activities and Development | BOSCH Vietnam
2016 |
| • <i>Docking and Locking for Electric bike in BOSCH Station</i> | |
| • Global Entrepreneurship Training | Handong Global University
2017 |
| • <i>Entrepreneurship Training</i> | |
| • JENESYS 2.0 Program | Japan
2014 |
| • <i>Japan-East Asia Network of Exchange for Students and Youths (JENESYS)</i> | |

VOLUNTEER EXPERIENCE

- | | |
|---|---------------------------|
| Founder at Robotlab Facebook and Website | Binh Duong, Vietnam |
| • <i>Conducted online and offline technical STEM training for students</i> | <i>Jan 2019 - Present</i> |
| • Member at Jenesys 2.0 (Japan-East Asia Network of Exchange for Students and Youths) | Japan |
| • <i>Students exchange programs that are intended to create a bridge between Japan and country in Asia</i> | <i>Jan 2014</i> |
| • Team Leader at a Robocon ABU(Asia-Pacific Robot Contest) University team | LHU, VietNam |
| • <i>Technical lead, facilitating open communication, encouraging member growth to reach the team goals</i> | <i>2011 - 2013</i> |

HAND-ON HARDWARE EXPERIMENTS

- **Robot platform:** UR10e, Kuka Youbot, Turtlebot 3, NAO, DJI Drone ..
- **Sensor:** Velodyne, IMU-Xsens, Houkyo Lidar, Intel Realsense, SICK Lidar-Camera, Torque-Force Sensor..
- **Embedded Computer:** Nvidia Jetson family, Raspi-Pi, NUC, Arduino..
- **Actuator:** Various of Servo motor, BLDC Motor, Linear motor, Motor driver,..



Master-Urkunde

Die Hochschule Karlsruhe-Technik und Wirtschaft verleiht
Herrn Tri Bien Minh,
geboren am 18. Juni 1991 in Tuy Hoa,
aufgrund der am 07. März 2017 bestandenen Master-Prüfung
den Hochschulgrad

Master of Science (M.Sc.)

im Studiengang

Mechatronics and Sensor Systems Technology

Der Rektor
In Vertretung

Prof. Dr. Dieter Höpfel
Prorektor





Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Master-Zeugnis

Herr Tri Bien Minh geboren am 18. Juni 1991 in Tuy Hoa,
hat das Studium im Studiengang

Mechatronics and Sensor Systems Technology

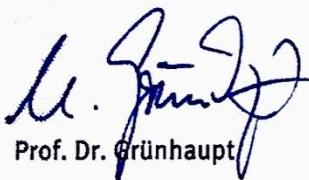
mit dem Hochschulgrad

Master of Science (M.Sc.)

am 07. März 2017 erfolgreich abgeschlossen.

Gesamtnote: gut (1,6)

Der Dekan



Prof. Dr. Grünhaupt

Prüfungsausschussvorsitzender



Prof. Dr. Westermann



Die Leistungen wurden wie folgt bewertet:

Fachprüfungen

Vietnamese German University

Sensors	gut	(2,3)
Electronics	befriedigend	(2,8)
Sensor Data Processing	gut	(2,1)
Sensor Manufacturing	sehr gut	(1,0)
Control Technologies	gut	(2,3)
Automation	ausreichend	(3,7)
Management and Communication	gut	(2,0)
Sensor Systems	sehr gut	(1,0)
Areas of Specialization	gut	(1,7)

Von der Hochschule Karlsruhe – Technik und Wirtschaft im Rahmen der Externenprüfung bewertet

Final Examination	sehr gut	(1,0)
Master-Thesis	sehr gut	(1,0)
Thema Master-Thesis:	Design, Modeling and Control of an Octocopter	

Position-based Visual Servoing with Dual Manipulators

Tri Bien¹

¹Electrical and Computer Engineering, Vietnamese German University

Abstract

The research focuses on designing a system for position-based visual servoing (PBVS) with dual 6-Degree of Freedom (DoF) UR10e robots. The main robot uses computer vision integrated position-based servoing algorithm for tracking the ARTag trajectories from another robot by using Realsense D435 camera. The PBVS algorithm allows the robot to compute the inverse Jacobian matrix from the elementary transform sequence (ETS) UR10e robot model and spatial velocity from the ARTag pose. The result of this algorithm is joint velocity control on the real UR10e robot, which helps the main robot simultaneously track the ARTag trajectories in real time. The resulting video can be accessed at: triknight.github.io/

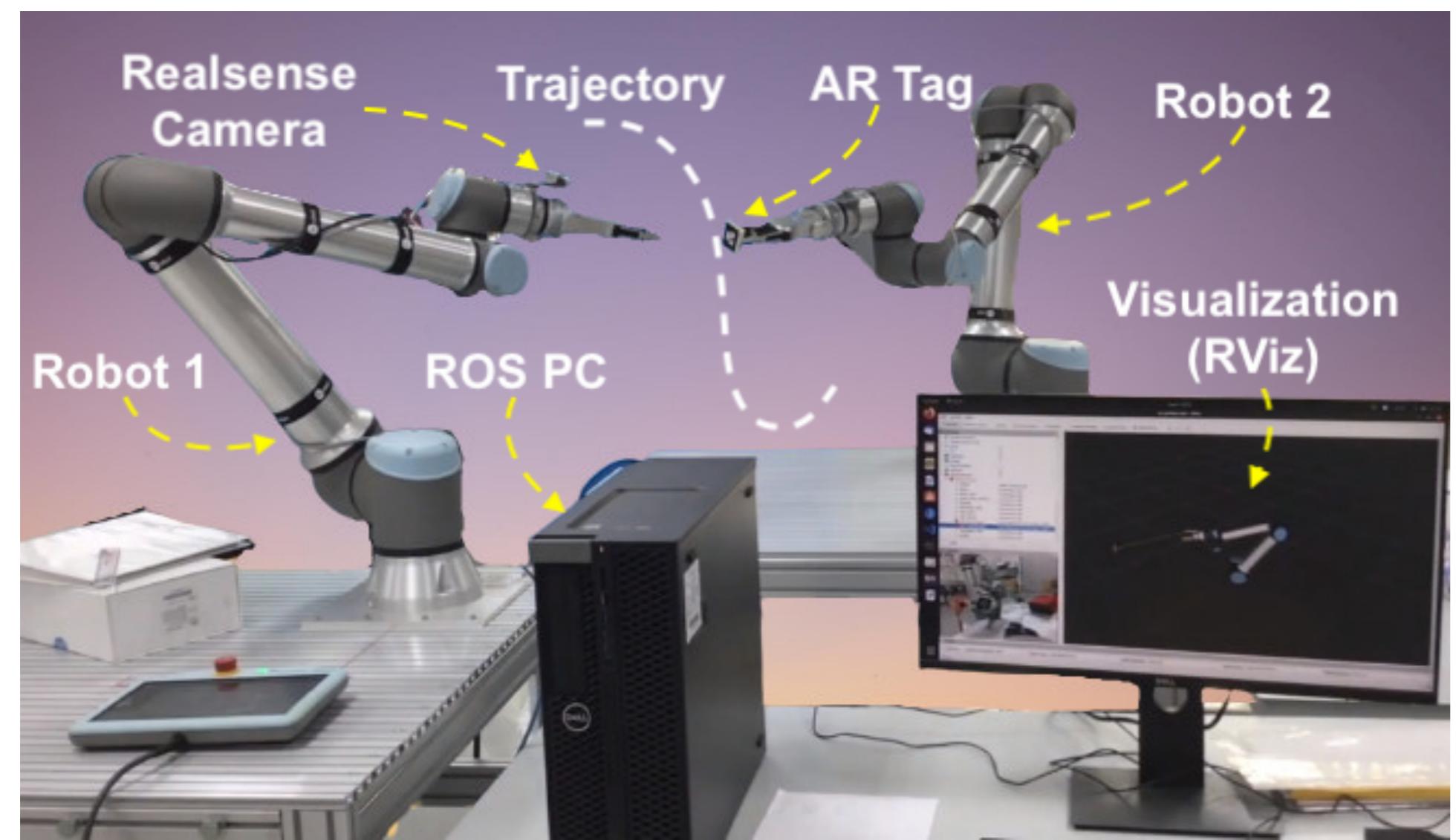


Figure 1. Hardware setup of position-based visual servoing with dual manipulators

The contributions of this paper are:

- Developed forward kinematics of the 6-DoF UR10e robot based on elementary transform sequence (ETS). The ETS model gets better robot kinematic representation, which can avoid the frame assignment constraints of Denavit and Hartenberg (DH) notation.
- Designed a position-based visual servoing (PBVS) with dual manipulators system by using ROS middle-ware, interfacing joint position, ARTag position, and controlling joints velocity of the real robot. Visualization of real robot operation data feedback by Rviz, and using Robotics Toolbox for simulation.
- Applied resolved-rate motion control (RRMC) into position-based servoing by calculating the inverse Jacobian matrix and spatial velocity from an error position vector of the end-effector.

System Overview

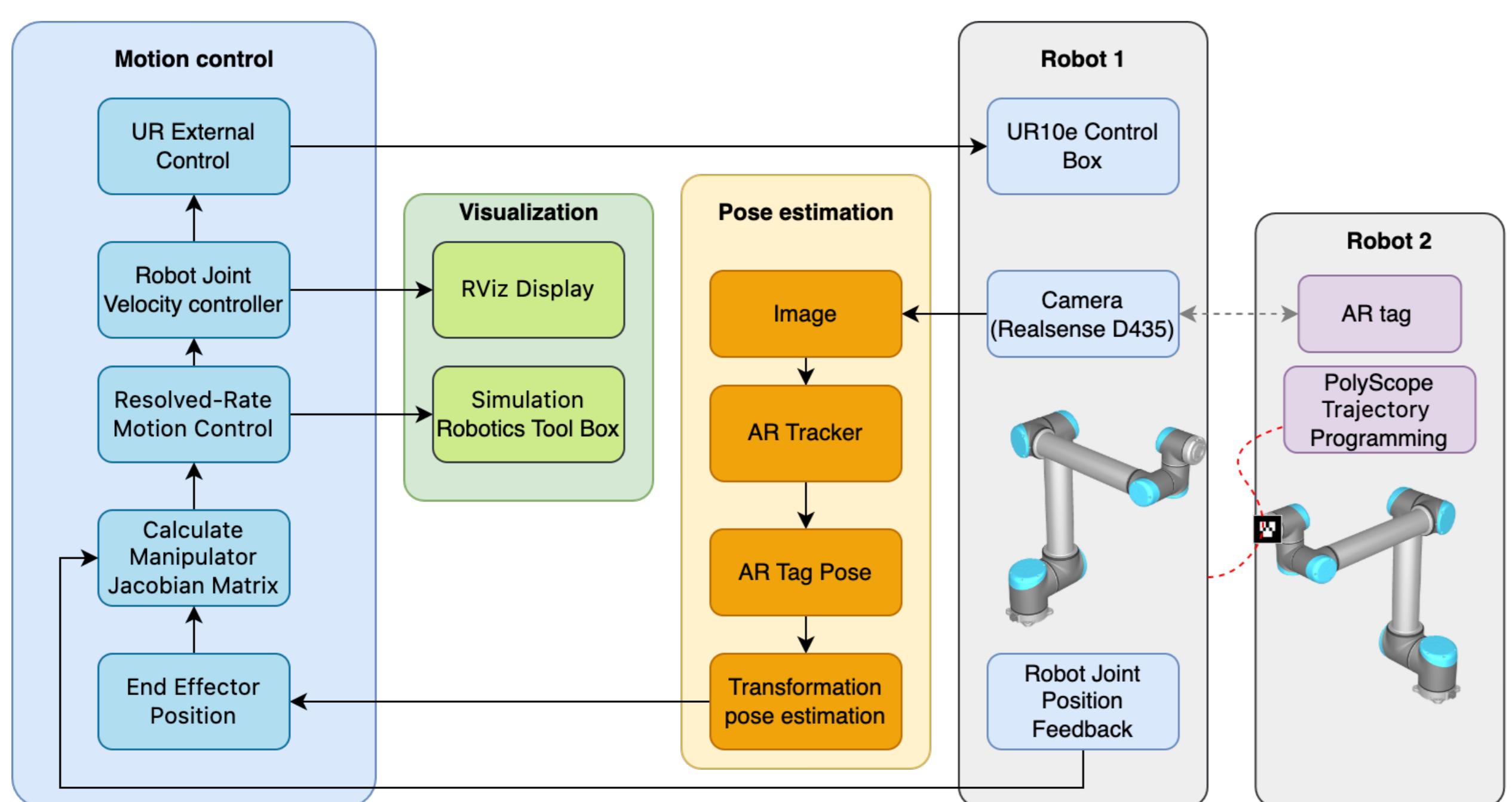


Figure 2. Position-based visual servoing with dual manipulators system

System Overview (Cont.)

The scheme of the developed PBVS with dual manipulators system is shown in Fig.2, which includes two robot arms. Robot 2 has been programming for grasping the ARTag and creating trajectory movement. Robot 1 has been connecting with a ROS computer with implemented PBVS algorithm. This algorithm can separate into three main components, which are pose estimation, motion control, and visualization.

- Pose estimation:** A fiducial marker -ARTag has been chosen to simplify object tracking and pose estimation. A new URDF model of the UR10e robot is created by integrating camera Realsense and gripper On-Robot RG2, which model can help the robot estimate the position from ARTag to the robot's end-effector via the open-source ARTag tracking ROS package [4].
- Motion control:** The position-based servoing algorithm includes two main steps. In the first step, the algorithm would be designed so that it would calculate the end-effector spatial velocity by running in as many iteration loops as necessary until the solution converges to the minimizing translation and rotation error vector between the end-effector current pose and ARTag pose. The second step, using resolved rate motion control calculates the required joint velocities to achieve the desired end-effector velocity by using an inverse Jacobian matrix with joint state feedback.
- Visualization:** The development of the visualization software is mostly done in ROS [5] using Rviz to visualize the robot, simulation in Python Robotics Toolbox [1] and IDE environment in Visual Studio using several open source libraries (OpenCV, ARTag tracking).

Elementary Transform Sequence (ETS)

The elementary transform sequence (ETS) of the 6-DoF UR10e manipulator in its home configurations. E_i represents an elementary transform and nT_m represents the pose of link frame m in the reference frame of link n.

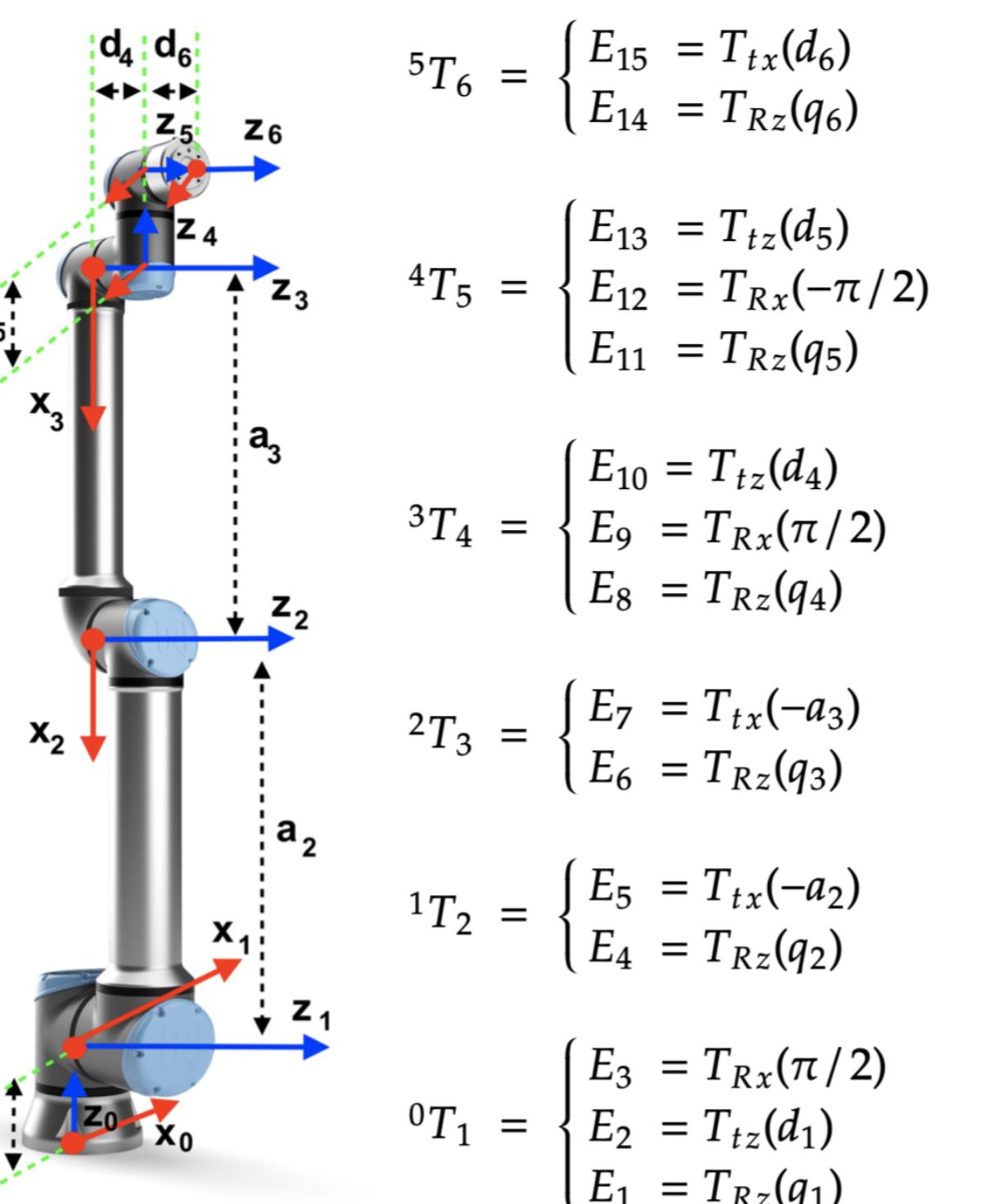


Figure 3. The Elementary Transform Sequence of the 6 DoF of UR10e robot

The dimension of the UR10e robot follows the table below. Where a_i, d_i are dimensions from joint i to joint $i - 1$, and q_i is the rotation around z_i axis of the robot joint.

Joint	a(m)	d(m)	α (rad)
Joint 1	0	0.1807	$\frac{\pi}{2}$
Joint 2	0.6127	0	0
Joint 3	0.57155	0	0
Joint 4	0	0.17415	$\frac{\pi}{2}$
Joint 5	0	0.11985	$-\frac{\pi}{2}$
Joint 6	0	0.11655	0

Table 1. These parameters reference the Universal Robot Denavit-Hartenberg parameters [6]

Position-Based Servoing

The elementary transform sequence (ETS) is another representation of the kinematics for a serial-link manipulator, which was introduced by Corke [2]. The forward kinematics of a manipulator presents a non-linear mapping between the robot's joint space and Cartesian task space.

$${}^0T_e(t) = \mathcal{K}(\mathbf{q}(t)) \quad (1)$$

Where ${}^0T_e \in SE(3)$ is a homogeneous transformation matrix describing the pose of end-effector in the world-coordinate frame, $\mathcal{K}(\cdot)$ is the product of a number of elementary transforms, and $\mathbf{q}(t) \in \mathbb{R}^n$ is the vector of joint generalized coordinates. Fig.3 is shown the ETS model of the UR10e and all dimension parameters as shown in Table.1.

Resolved-Rate Motion Control (RRMC)

End-effector spatial velocity in the world frame $\nu = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$ can be computed

$$\nu = \begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} J_v(q) \\ J_\omega(q) \end{pmatrix} \dot{q} = J\dot{q} \quad (2)$$

Where $J_v(q) \in \mathbb{R}^{3 \times n}$, $J_\omega(q) \in \mathbb{R}^{3 \times n}$ are the translational part and rotational part of the manipulator Jacobian matrix, and \dot{q} is the joint velocities. RRMC is an elegant motion method along an arbitrarily oriented straight line in space [7], from Eq.2 can re-arrangement:

$$\dot{q} = J(q)^{-1}\nu \quad (3)$$

UR10e has 6 degree-of-freedom (DOF), so the Jacobian matrix $J(q) \in \mathbb{R}^{6 \times 6}$ is square and invertible.

Position-based servoing (PBS)

PBS algorithms depend on an error vector that presents the translation and rotation from the end-effector's current pose to the ARTag pose [3].

$$\mathbf{e} = \begin{pmatrix} \tau({}^0T_e*) - \tau({}^0T_e) \\ \alpha(\rho({}^0T_e*) - \rho({}^0T_e)) \end{pmatrix} \in \mathbb{R}^6 \quad (4)$$

Where 0T_e and 0T_e* are forward kinematics and desired end-effector pose in the base frame of the robot, τ and ρ are translation and rotation of the end-effector pose, and $\alpha(\cdot)$ transforms a rotation matrix to Euler vector. At each time step, the PBS scheme is constructed by taking the error term from Eq.4 to set end-effector spatial velocity ν

$$\nu = k\mathbf{e} \quad (5)$$

Where k is a gain, which is typically a diagonal matrix. From Eq.5 and Eq.3 , we can calculate the joint velocities based on the Jacobian matrix and spatial velocity, then apply them to the real robot.

Conclusion and future work

This study developed position-based visual servoing algorithms for existing dual commercial industrial robot manipulators and encapsulated them into a ROS package. Encouraging outcomes have been presented based on unique real-world dual manipulators experiments, demonstrating the ability of the position visual servo control scheme.

The further application of this work is developing visual approaches focused on semantic understanding, vision-enabled intervention, and SLAM algorithms for archiving fully autonomous manipulation in industrial applications. Further study can apply different types of manipulators, such as underwater manipulation for ROV and AUV systems.

References

- [1] Peter Corke and Jesse Haviland. "Not your grandmother's toolbox – the Robotics Toolbox reinvented for Python". In: Proceedings - IEEE International Conference on Robotics and Automation (2021). DOI: [10.1109/ICRA48506.2021.9561366](https://doi.org/10.1109/ICRA48506.2021.9561366).
- [2] Peter I. Corke. "A Simple and Systematic Approach to Assigning Denavit–Hartenberg Parameters". In: IEEE Transactions on Robotics 23.3 (2007), pp. 590–594. DOI: [10.1109/TRO.2007.896765](https://doi.org/10.1109/TRO.2007.896765).
- [3] Jesse Haviland and Peter Corke. "Manipulator Differential Kinematics Part I: Kinematics, Velocity, and Applications". In: (July 2022). DOI: [10.48550/arXiv.2207.01796](https://doi.org/10.48550/arXiv.2207.01796).
- [4] Open source AR tag tracking library. http://wiki.ros.org/ar_track_alvar.
- [5] Morgan Quigley et al. "ROS: an open-source Robot Operating System". In: (May 2009).
- [6] Universal Robot DH parameters. <https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/>.
- [7] Daniel E. Whitney. "Resolved Motion Rate Control of Manipulators and Human Prostheses". In: IEEE Transactions on Man-Machine Systems 10.2 (1969), pp. 47–53. DOI: [10.1109/TMMS.1969.299896](https://doi.org/10.1109/TMMS.1969.299896).

LiDAR-based Vehicle Detection by using DBSCAN Unsupervised Clustering approach

1st Tri Bien Minh
Electrical and Computer Engineering
Vietnamese German University
Binh Duong, VietNam
tri.bm@vgu.edu.vn

2nd Hien Vo Bich
Electrical and Computer Engineering
Vietnamese German University
Binh Duong, VietNam
hien.vb@vgu.edu.vn

Abstract— Roadside LiDAR is helping to build intelligent and safe transportation. Object detection is a challenging and fundamental problem in computer vision. Moreover, the vehicle detection system is essential to Intelligent Transportation Systems (ITS). Many researchers in the transportation field spend an enormous amount of money to collect and analyze traffic data to optimize street systems. This research aims to develop a case study for a vehicle detection system in a complex roadway area based on LiDAR through an embedded system. For this purpose, an embedded GPU integrated (Nvidia JetsonTX2) with low power and high performance has been picked, which supports an unsupervised learning algorithm to be run simultaneously and a detection algorithm to be applied for point cloud recognition. We also discuss the structure of the architectures of LiDAR-based roadside systems, and lidar data processing for vehicle detection. In the real-scanned HDL-32E Velodyne 3D LiDAR dataset, our proposed method can achieve vehicle detection accuracy up to 82.7% in several real-scenes datasets. The future research directions to contribute resources beneficially to industry, academia, and government agencies for choosing appropriate LiDAR-based technologies for their vehicle monitoring systems.

Keywords— *LiDAR, vehicle detection, traffic monitoring, unsupervised learning, Jetson, DBSCAN, roadside lidar*

I. INTRODUCTION

As the number of vehicles has increased dramatically, this causes severe traffic congestion in big cities in many countries. One of the good solutions to reduce traffic congestion is a traffic monitoring system [1]. The Intelligent Transportation System (ITS) is used to gather traffic data such as the number of vehicles, vehicle speed, and types of vehicles. These collected data can be used for traffic analysis to enhance the safety of transportation, predict future transportation demands, and take advantage of roadway systems [2]. Vehicle detection is the main functionality of the traffic monitoring system. Due to dramatic technical challenges, various research papers have been considered regarding vehicle detection systems. The contributions of this article are summarized as follows.

- This article is using the architecture with computing at the edge using a low-power embedded computer Jetson TX2.

- Propose a collecting and processing hardware and processing point cloud data pipeline.
- Experiment with vehicle detection based on DBSCAN unsupervised learning methods with different scenes.

II. ARCHITECTURE OF THE LiDAR ROADSIDE SYSTEM

The system architecture of the LiDAR-based connected roadway infrastructure integrates roadside LiDAR sensors, traffic data processing system, and software-connected road users. The system is shown in Fig.1.

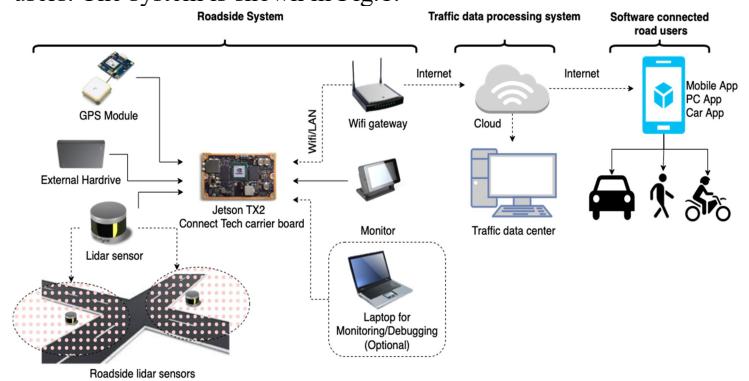


Fig. 1. The system architecture of the roadside LiDAR-based connected roadway infrastructure

A. Roadside LiDAR sensors

LiDAR sensors can be installed at two or four corners to cover the full coverage of the intersection area, which depends on the intersection size and geometry of the roadway. The 360-degree 3D LiDAR sensors were used for vehicle detection. In our work, the roadside LiDAR sensors used high-resolution HDL-32E and average-resolution VLP-16 LiDAR sensors from the Velodyne brand. The HDL-32E LiDAR can generate 360-degree 3D point cloud data of up to 1.39 million points per second by using 32 internal laser/detector, which detects a range of up to 100 m and an average accuracy of ± 2 cm. VLP-16 LiDAR has only 16 laser/detector, which can create 600,000 points per second, and the range is also up to 100 m, which lowers the accuracy to only ± 3 cm. The rotational speed of the LiDAR is about 5-20 rotations per second, which covers the vertical field of view (FOV) about 45-degree for HDL-32E

LiDAR and 30-degree for VLP-16 [3]. Maintaining the Integrity of the Specifications

B. Embedded computer

Embedded computer at each intersection is to process collecting data from LiDAR sensors. In this study, the Jetson TX2 has been the main processor unit. It has integrated the GPU with 256-CUDA cores, Quad-Core ARM Cortex A57, 8GB memory, and 32 GB disk space. Power saving is the unit key point of this module, which consumes only 7.5W~15W [4]. It is suitable for AI computing at the edge applications, especially computer vision applications.

C. External hard drives and Peripherals

Hard drives are used for storing and collecting data, which are optional and depend on the architecture of the systems. If the embedded computer can process the real-time data and send this data over the internet through the gateway, in this case the hard drives are not necessary. Otherwise, data can be stored on the hard disks for backup, diagnostic and debugging applications. The hard drives have not been used in our system, because the main approach in our system is toward computing at the edge, and uploading these data to the cloud server. Other peripherals such as laptops, monitors, light sensors, and GPS-module are helping the system work more reliably and easy to monitoring and debug systems.

D. Traffic data processing

Data collected from the roadside system have been sent to the cloud and traffic data center. The traffic data center is for data archiving, data integration, performing the traffic operation, traffic control with optimization decisions, and helping the human make the most suitable choices to control the system. This data center can also help the road users fetch, and visualize the information from the real situation in the roadway.

E. Software-connected road users

Users can get real-time road information about congestion status, numbers of vehicles on roads, the density of vehicles on the roads, average moving speed in consider areas, and related traffic information. The software can be developed on multiple platforms and devices such as smartphones, tablets, PC, and car HMI.

III. LIDAR DATA PROCESSING

The vehicle detection process and visualization LiDAR data processing steps are shown in Fig.2.

A. Point cloud Preprocessing step

LiDAR can generate a huge amount of 3D point clouds which require high computing powers. Therefore, the raw point cloud must be filtered before going to the ground point removal step. The crop point cloud is adopted to filter the sparse points within a distance greater than 50m from LiDAR, and remove the center lane area. Voxel grid filtering created a cubic grid and filtered the point cloud by only leaving a single point per voxel cube.

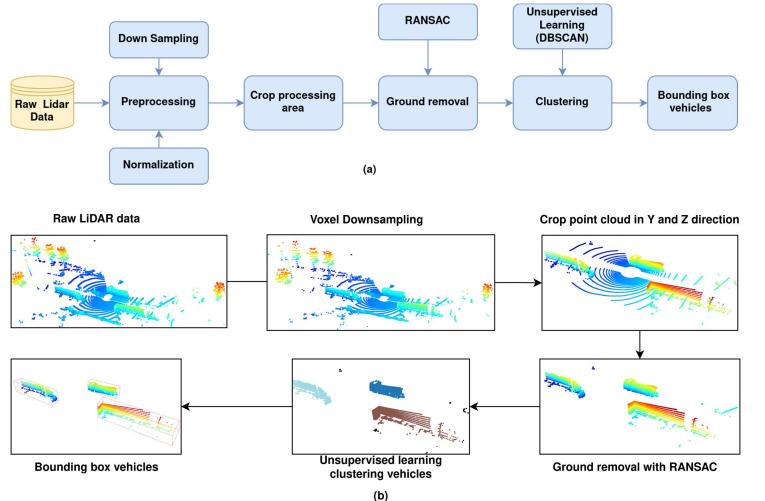


Fig. 2. Vehicles detection process (a) and visualization LiDAR data processing step (b)

Then the larger the cube length the lower the resolution of the point cloud, which is the number of points data also decreasing, which can be seen in Fig. 3.

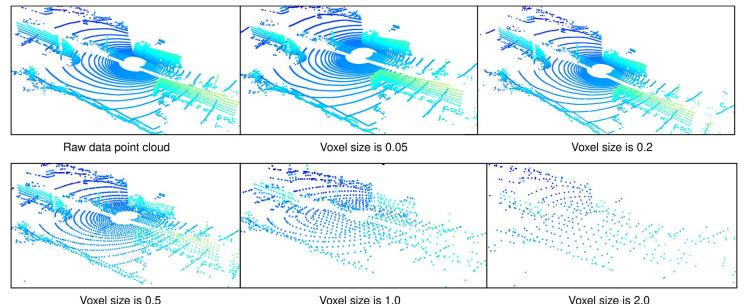


Fig. 3. Voxel down sampling with various voxel sizes between 0.05 to 2.0

B. Ground point removing

LiDAR with 45/30-degree vertical FOV can create a ground point cloud of the road when they are working. These ground points take up a large proportion of the raw point cloud, which intensely affects the subsequent processing. In this step, the RANSAC (Random Sample Consensus) method was used to filter the ground plane [6]. The RANSAC operates for a max number of iterations and returns the model with the best fit. Each iteration randomly picks a subsample of the point cloud data and fits a model through a point cloud plane. Then the iteration with the highest number of inliers or the lowest noise is used as the best model. The RANSAC algorithm for ground point removal is described below.

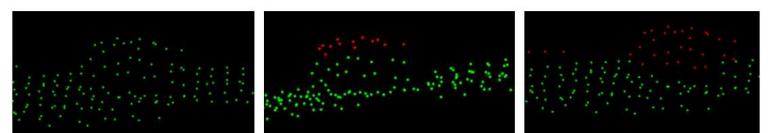


Fig. 4. Result of the RANSAC with number of iterations is remaining 100, and different threshold distance 1.5 (left), 0.7 (middle), 0.3 (right)

When changing the different threshold distances the inliers were changing, and the number of iterations can also affect the computing time and finding the best fit model. In our work, the good number for getting a good result is a threshold distance value of 0.3 and the iteration value of 100. The result of the RANSAC is shown in Fig. 4. The red point cloud is the outliers, and the green one is the inliers after running the algorithm.

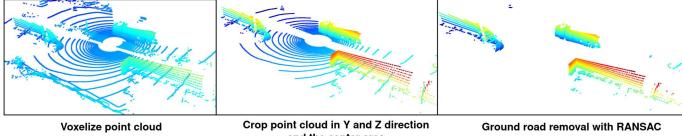


Fig. 5. Processing steps for RANSAC: Voxel down sampling to reduce point cloud data (left) Crop in the Y and Z-axis and center area (middle) Result of the RANSAC filter out the road plane (right)

Processing steps for RANSAC have followed steps, as shown in Fig. 5: Voxel down sampling to reduce point cloud data, next crop point cloud data in the Y and Z-axis and center area, then using RANSAC for plane segmentation to remove road plane.

C. DBSCAN Unsupervised clustering approach

To combine local point cloud clusters given a point cloud from LiDAR sensor. Clustering algorithms can be used for this objective. It includes a variety of techniques based on various distance units. For instance, Gaussian mixtures (Mahalanobis distance to centers) [7], Affinity propagation (graph distance) [8], Mean-shift (distance between points) [9], and Spectral clustering (graph distance) [10]. The researchers developed unsupervised learning techniques that could build feature detector layers without the need for labeled data. Deep learning interest was rekindled in part due to unsupervised learning. Unsupervised learning will likely become much more significant in the long run [11].

This study uses the Density-based spatial clustering of applications with the noise clustering method. Density-Based Clustering to the concept that a cluster in data space is a contiguous region of high point density, separated from other similar clusters by contiguous regions of low point density [12], clustering refers to unsupervised learning approaches that discover unique clusters in the data. The unsupervised learning DBSCAN algorithm for vehicle clustering is shown below.

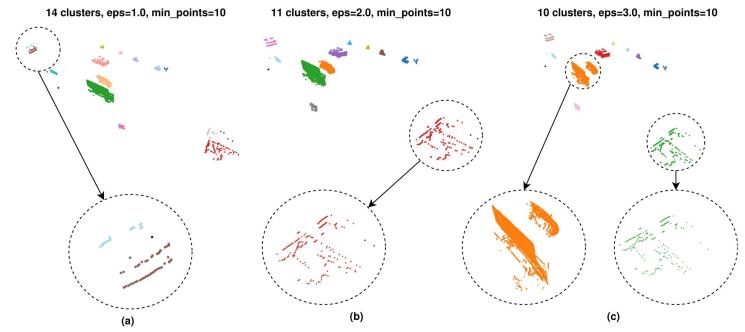


Fig. 6. Example DBSCAN with various eps value between 1.0 to 3.0 with the same minimum number of points is 10 points.

The DBSCAN is an unsupervised learning algorithm, so it does not need to train before running the clustering, this is the main advantage of this approach. But it has some drawbacks, Fig. 6 illustrates an example when changing the eps number and remaining the minimum point in each cluster, which has reduced the number of clusters when increasing the eps value. Following this experiment, DBSCAN is good for clustering the various vehicle, even small motor-bike vehicles. But it is not the optimal solution, in Fig. 6 (a) the $\text{eps} = 1.0$ then the neighbor in the car object is a mixture, and the algorithm returns two clusters in one object. Moreover, in Fig. 6 (b-c) the $\text{eps} = 2.0 - 3.0$, the number of clusters is decreased, and get more clustering errors by grouping two objects into one cluster. It is quite tricky to choose the optimal eps and minimum points in this algorithm.

IV. EXPERIMENTAL RESULT

This section shows the roadside hardware setup and the result of experiments. A mounting plate for 3D LiDAR and housing for Jetson TX2 integrated Connect Tech carrier board is designed and printed by using 3D printing machine with PLA material, which is illustrated in Fig. 7.



Fig. 7. 3D printing mounting object for LiDAR (left), housing Jetson TX2 (middle) and tripod mounting (right)

The roadside system of the experimental platform setting up is shown in Fig. 8. This system includes 3D Lidar HDL-32E, LCD monitoring, Ublox M8N GPS module, Jetson TX2 integrated with Connect Tech shield is the main embedded computer and 12V power supply. The system connects all the modules together and works properly in normal daylight conditions.



Fig. 8. Roadside system setup in the field test (left), setting lidar at the middle of the lane (right)

The roadside system is setup at the location, which longitude and latitude location are [11.014811, 106.662682], respectively, which is shown in Fig. 9.

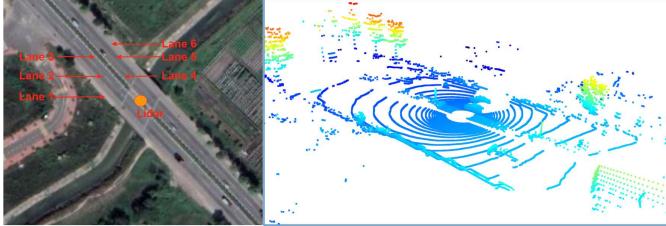


Fig. 9. Field site for collecting point cloud data in geometry location [11.014811, 106.662682] (left), sample LiDAR data frame (right)

In order to test the vehicle detection of the algorithm, we counted the number of down sampling points, crop and ground removal points, and the number of vehicle detection at each stage in multiple environments. The code was implemented using Python and the Open3D framework [13]. These data have been compared with the number of vehicles counted by humans.

Table 1. Result of vehicles detection in different scenes

Scene	Points number	Down sampling points number	Crop & Ground removal points number	Number of vehicle detection by algorithm	Number of vehicle counted by human
Scene 1	69120	28287	9201	3	4
Scene 2	69504	26677	8303	7	8
Scene 3	69120	26423	2679	8	10
Scene 4	69120	26789	1226	9	11
Scene 5	69504	26325	5053	11	12
Scene 6	69120	27643	6075	10	13
Total	415488	162144	32537	48	58

In the vehicle detection part, as shown in Table 1, the total of raw point cloud collecting was 415488 points, after down sampling the points number has 39.02% of the original data. After the crop and ground segmentation part, the number of points dramatically reduced by 92.16%, it only has 35537 points.

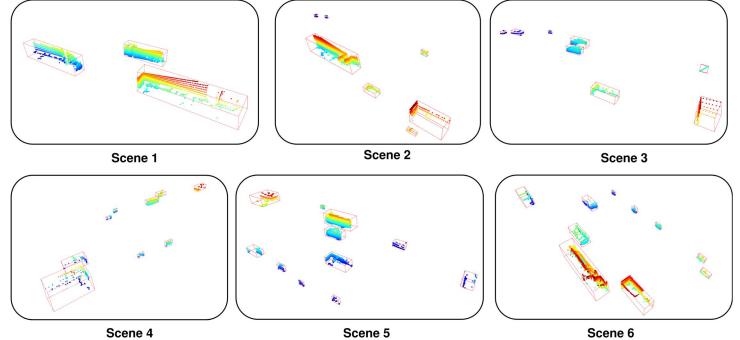


Fig. 10. Visualization of the result of vehicles detection in different scenes

The visualization of vehicle detection result in different 6 scenes, which is illustrated in Fig. 10. It is clear to see the unsupervised learning algorithm can detect mostly vehicles in the road, this can detect 48 over 58 vehicles, and vehicle detection precision was 82.7%, and some vehicles connected with bushes and cannot be separated. The four wheels vehicle detection precision was 80% over the dataset.

V. CONCLUSION

This study describes a vehicle detection procedure that mostly consists of four steps: To acquire objects above the ground by using LiDAR and get the raw point clouds, first down-sampled, crop point clouds data and ground points are eliminated, then clustering the vehicles by using the unsupervised learning DBSCAN approach. In the Velodyne HDL-32E dataset, our method achieved 82.7 % precision for total vehicle detection, and 80% precision for four wheels vehicle detection in six scenes. The error has come from the clustering process, some small vehicles such as bikes and motorbikes can be neglected, and occlusion between vehicles reduced the accuracy. Future research will take into account RGB cameras and other sensor data to increase detection accuracy. Moreover, deep learning methods are considered to deal with further detection and classification applications.

ACKNOWLEDGMENT

The authors wish to thank Mr. Linh V. Ngo for collecting data and field tests, the BugCar team in the VGU robotics laboratory for designing the 3D mounting housing, as well as providing us with the 3D Velodyne Lidar HDL-32E in a short time to run the application.

REFERENCES

- [1] M. Won, T. Park and S. H. Son, "Toward Mitigating Phantom Jam Using Vehicle-to-Vehicle Communication," *IEEE Transactions on Intelligent*

- Transportation Systems*, vol. vol. 18, no. no. 5, pp. pp. 1313-1324, (2017).
- [2] Wang, Zhao & Wang, Xing & Fang, Bin & Yu, Kun & Ma, Jie, "Vehicle detection based on point cloud intensity and distance clustering," *Journal of Physics: Conference Series*, vol. 1748. 042053., (2021).
- [3] Velodyne Lidar, "VLP-16 Lidar, HDL-32E," [Online]. Available: <https://velodynelidar.com>.
- [4] Nvidia, "Nvidia Homepage," [Online]. Available: <https://developer.nvidia.com/embedded/jetson-tx2>.
- [5] Bin Lv, e. al, "LiDAR-Enhanced Connected Infrastructures Sensing and Broadcasting High-Resolution Traffic Information Serving Smart Cities," *IEEE Access*, vol. vol. 7, pp. pp. 79895-79907, (2019).
- [6] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, p. 381–395, (1981).
- [7] P. Mahalanobis, "On the generalized distance in statistics," *Proc. Nat. Inst. Sci.*, p. 49–55, (1936).
- [8] Brendan J. Frey, Delbert Dueck, "Clustering by Passing Messages Between Data Points," *SCIENCE*, vol. 315, no. 5814, pp. 972-976.
- [9] Cheng, Yizong , "Mean Shift, Mode Seeking, and Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 790–799, (1995).
- [10] Ulrike von Luxburg, "A Tutorial on Spectral Clustering," *Statistics and Computing*, (2007).
- [11] LeCun, Yann & Bengio, Y. & Hinton, Geoffrey, "Deep Learning," *Nature*, vol. 521, no. 436-44, (2015).
- [12] Martin Ester, e. al "A density-based algorithm for discovering clusters in large spatial databases with noise," *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, p. 226–231, (1996).
- [13] Qian-Yi Zhou and Jaesik Park and Vladlen Koltun , "Open3D: A Modern Library for 3D Data Processing," *ArXiv*, vol. abs/1801.09847, (2018).
- [14] M. Won, "Intelligent Traffic Monitoring Systems for Vehicle Classification: A Survey," *IEEE Access*, vol. vol. 8, pp. pp. 73340-73358, (2020).

MiniRos: an autonomous UGV robot for education and research

Tri Bien Minh

*Electrical and Computer Engineering
Vietnamese-German University
Binh Duong, Viet Nam
tri.bm@vgu.edu.vn*

Hua Thanh Luan

*Research and Development Autonomous Robot
Robotlab
Binh Duong, Viet Nam
robotlabvn@gmail.com*

Do Xuan Phu

*Mechatronics and Sensor Systems Technology
Vietnamese-German University
Binh Duong, Viet Nam
phu.dx@vgu.edu.vn*

Tran Quang Nhu

*Electrical and Computer Engineering
Vietnamese-German University
Binh Duong, Viet Nam
nhu.tq@vgu.edu.vn*

Bui Minh Duong

*Electrical and Computer Engineering
Vietnamese-German University
Binh Duong, Viet Nam
duong.bm@vgu.edu.vn*

Abstract—MiniRos is a small autonomous UGV-Unmanned ground vehicle developed for education and research purposes. In this paper, we develop MiniRos with a kinematic skid-steering, high torque and compact structure drive system that can take research to the outdoor environment. Its small payload capacity and power system are suitable for various types of loads, customized to meet research needs. MiniRos is run on a Robot Operating System (ROS), which has been developed to be compatible with standard robotics programming environments. The robot senses the environment with various kinds of sensors like depth cameras, 2D lidar scanners and performs all processing on embedded computers. The robot can build a 2D map with various SLAM algorithms such as Kart SLAM, Gmapping, Hector SLAM, and navigation in the built map. Moreover, the study evaluated both the computational and the memory requirements based on each SLAM algorithm within different sizes of maps. Finally, skid-steering type robot MiniRos simulation on Gazebo environment is available as open-source, this is a useful tool for researchers and educators.

Index Terms—Autonomous robot, UGV-Unmanned ground vehicle, mobile robot, Robot Operating System, SLAM

I. INTRODUCTION

Autonomous outdoor robots are assured to become one of the most impactful applications of autonomy. Skid-steered mobile robots are widely utilized in a variety of applications, including terrain mapping, exploration, as well as defense and home services. The steering system for a skid steered mobile robot (vehicle) makes the robot mechanically rigid, robust, and simple for indoor or outdoor environments [1].

A modern autonomous robotics curriculum should train students from the individual components to the system-level interactions. The time required to develop all of the hardware stuff and assembly is important, and these tasks are not straight related to the desired autonomous subject matter [2].

To solve this problem, we propose MiniRos - an autonomous robot platform as shown in Fig. 1. The minimal autonomy configuration uses Jetson Nano , Pixhawk, Rplidar A1M8, depth camera Realsense D435, and MiniRos motor



Fig. 1. MiniRos in 3D design

driver is developed by us. Furthermore, the extended autonomy version has mostly the same configuration with minimal version, but this is using higher computing power with Jetson TX2 and reliable sensor, Hokuyo URG-04LX lidar, stereo Zed camera as shown in Table I. Moreover, another objective was that the education applications were able to use the robots for academic research applications. Having a robot platform that can be used by students and researchers helps to raise the experience of students towards the world of robotics research.

This paper is organized as follows. In section II, we discuss the hardware architecture, four-wheel skid-steered mobile robot kinematic, and Gazebo model of the MiniRos. Section III presents simultaneous localization and mapping (SLAM) with SLAM computational experiments. MiniRos navigation and these elements are presented in section IV. Finally, we conclude the study and discuss future research directions in section V.

II. THE MINIROS

A. Hardware Architecture

MiniRos is an autonomous vehicle designed with the objectives of modularity, ease of development in programming, working outdoor environment, and modularity and ease of

Table I. MiniRos autonomy configuration

Components	Minimal Autonomy	Extended Autonomy
Main frame	laser cutting	laser cutting
Embedded Computer	Jetson Nano™	Jetson TX2™
Controller	Pixhawk	Pixhawk
Lidar	Rplidar A1M8	Hokuyo URG-04LX
Camera	Realsense D435	Zed Stereo
Motor driver	MiniRos drive	MiniRos drive
DC motor	Planetary	Planetary
Gear ratio	26.9:1	26.9:1
Communication	Wireless dongle	Wireless dongle
Li-ion Battery	12V 4.4Ah	12V 4.4Ah

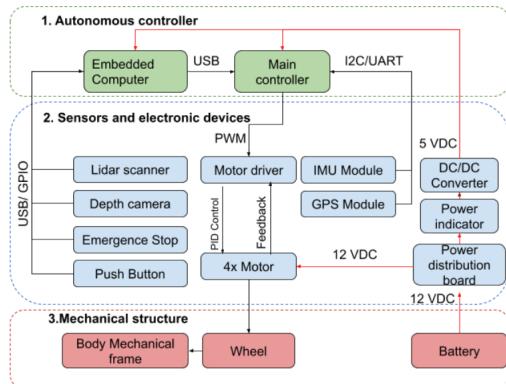


Fig. 2. MiniRos an autonomous UGV robot hardware layer

construction. It includes two configurations as shown in Table I.

- The **minimal autonomy configuration**.
- The **extended autonomy configuration**.

MiniRos hardware layer includes main following components below, shown in Fig. 2

1) *Mechanical structure*: The MiniRos is a skid drive four-wheeled mobile robot. Four 12V brushed DC motors are used to motorize the robot. Motors have a 26.9:1 metal gearbox ratio and an integrated quadrature encoder of 13 pulses per revolution of the motor shaft.

2) *Controller*: Minibot minimal configuration uses embedded computer Nvidia Jetson Nano as the main processor, this is included: GPU: 128-core Nvidia Maxwell, A57 Quad-core ARM, and 4 GB DDR4 memory. On the other hand, the extended configuration is using a more powerful core processor Jetson TX2 with 256-core NVIDIA Pascal, A57 Quad-Core ARM, and 8GB DDR4 memory. The main controller board uses the Pixhawk (32-bit ARM Cortex M4, CPU Speed 168 Mhz, 256 KB RAM and 2 MB Flash) to interface with the peripheral device, motor driver, IMU, GPS module, radio controller, battery indicator.

3) *Motor driver*: MiniRos motor drive was designed by using KiCad software. The main microcontroller in the motor driver PCB is ATmega-168PA, which is an 8-bit AVR RISC-

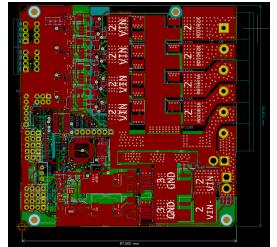


Fig. 3. MiniRos Motor driver integrated PID algorithms (left) MiniRos drive PCB Layout (right)

based architecture microcontroller combining 16KB ISP flash memory, 512B EEPROM, 1KB SRAM, 23 General-purpose input/output, 32 general purpose registers [3]. The main clock to operate the microcontroller is an external 16 MHz crystal clock generator. This internal timer module is handled to generate pulses for the DC motor's speed control. Several communication techniques are supported by the microcontroller such as I²C, RS232, and SPI. The motor driver is using bootloader Arduino platform and programming on C/C++ language via Visual Studio Code IDE with Platform IO add-in, but it is possible to use any programming languages including Java, Assembly.

The motor driver includes dual H-Bridge control DC motor, and each H-Bridge contains four power MOSFET gates. The MOSFET gate is HEXFET® IRF3205 [4] Power MOSFET from International Rectifier, which is fast switching speed and ruggedized device design with Continuous Drain Current at 10V can reach 110 A ($T_c = 25^\circ C$).

The MiniRos motor driver allows controlling four DC motors according to a speed command. This driver snapshot and PCB layout are shown in Fig. 3. It needs two different power supplies: 12V for the DC motors and 5V for the electronics components. This receives the speed command through the PWM signal from the mainboard, the control DC motor with "Brushed BiPolar" mode, which is control speed of the motor by sending PWM values normally between 1000 μs and 2000 μs . In value 1000 μs motor runs maximum speed in Counter Clock Wise to 2000 μs maximum speed in Clock Wise and 1500 μs motor stop. The motor driver is integrated PID control algorithms by getting the encoder data and controlling the motor speed by using PWM signal.

4) *Sensor*: MiniRos are using various kinds of sensors to get information from the environment, such as laser scanner Rplidar A1M8 or Hokuyo URG-04LX to build a map and scan the object. Moreover, the depth camera Intel Realsense D435 or stereo Zed camera is streaming the raw data and depth image to the robot, and this helps the robot to detect and recognize the object in the working area. Depending on the request of the user, a robot can attach more sensors like 3D Lidar, high precision GPS, IMU module if applicable.

B. Kinematic Skid-steering MiniRos

Model assumptions: Global frame at the O origin and x , y axes, and local frame locale at the robot body frame. Robot

moves on the 2D plane and the linear velocity concerning the robot local frame can write:

$$v = (v_x, v_y, 0)^T \quad (1)$$

where $v = (v_x, v_y)$ is the translational velocity of the robot in relation to its local frame, and the robot's angular velocity:

$$\omega = (0, 0, \omega_z)^T \quad (2)$$

where ω_z is the angular velocity rotation about z-axis

We can call $p = (X, Y, \theta)^T$ is state vector describing local coordinate robot frame reference to the global frame. Then the generalized velocity state vector can denoted $\dot{p} = (\dot{X}, \dot{Y}, \dot{\theta})^T$. This velocity state vector references with the linear velocity and angular velocity by equation below [5]:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} \quad (3)$$

Robot is designed with four wheels, front right and rear right, front left and rear left as shown in Fig. 4. When the robot turns, the Instantaneous Center of Rotation (ICR) of the right-side tread, left-side tread, and the robot body are denoted ICR_R, ICR_L, ICR , respectively. With local coordinates of each ICR bellow:

$$\begin{aligned} ICR_r &= (x_{ICR_l}, y_{ICR_r}) \\ ICR_l &= (x_{ICR_l}, y_{ICR_l}) \\ ICR &= (x_{ICR}, y_{ICR}) \end{aligned} \quad (4)$$

Wheel rotational velocity denote ω_i with $i = 1, 2, 3, 4$, respectively. From model assumption: $\omega_l = \omega_1 = \omega_2$ and $\omega_r = \omega_3 = \omega_4$. The direct kinematics in the plane can be stated this way:

$$\begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} = \vartheta \begin{bmatrix} \omega_l r \\ \omega_r r \end{bmatrix} \quad (5)$$

where ω_r, ω_l are wheel angular velocities for right and left tread, r is the radius of the wheel, ϑ The matrix is determined by the ICR coordinates and correction factors of each tread.

It is known that all ICR_r, ICR_l, ICR lie on a line parallel to the local y axis [6]. The geometrical relation between robot's transitional and rotational velocities with ICR positions is indicated by:

$$y_{ICR} = \frac{v_x}{\omega_z} \quad (6)$$

$$y_{ICR_l} = \frac{v_x - \omega_l r}{\omega_z} \quad (7)$$

$$y_{ICR_r} = \frac{v_x - \omega_r r}{\omega_z} \quad (8)$$

$$x_{ICR} = x_l = x_r = -\frac{v_y}{\omega_z} \quad (9)$$

From Eqs.(6) to (9) the element of matrix ϑ can be writen:

$$\vartheta = \frac{1}{y_{ICR_l} - y_{ICR_r}} \cdot \begin{bmatrix} y_{ICR_r} & -y_{ICR_l} \\ x_{ICR} & -x_{ICR} \\ -1 & 1 \end{bmatrix} \quad (10)$$

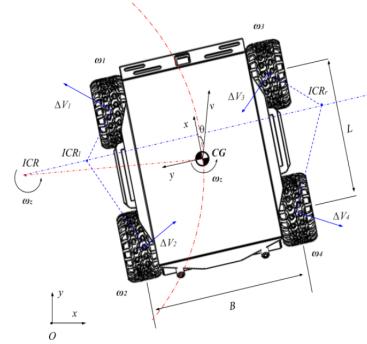


Fig. 4. Kinematic Skid-steering MiniRos

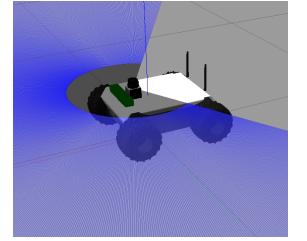
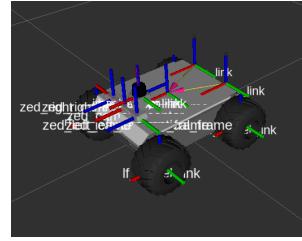


Fig. 5. Snapshot of the MiniRos model rendered in the ROS-Rviz environment (left) and Gazebo simulated environment (right)

In this case, the robot is symmetrical so the ICRs lie symmetrically on the x-axis and $X_{ICR} = 0$. A matrix ϑ gets the following form:

$$\vartheta = \frac{1}{2y_0} \cdot \begin{bmatrix} y_0 & -y_0 \\ 0 & 0 \\ -1 & 1 \end{bmatrix} \quad (11)$$

where $y_0 = y_{ICR_r} = -y_{ICR_l}$ is the instantaneous tread ICR value along y-axis. For the symmetrical model $v_l = \omega_l r$, $v_r = \omega_r r$. The following equations can be represented as:

$$\begin{cases} v_x = \frac{\omega_l r + \omega_r r}{2} = \frac{v_l + v_r}{2} \\ v_y = 0 \\ \omega_z = \frac{-\omega_l r + \omega_r r}{2y_0} = \frac{-v_l + v_r}{2y_0} \end{cases} \quad (12)$$

C. MiniRos Gazebo model

Gazebo is a robot simulator for rapidly design robots, testing algorithms, and training machine learning systems using a realistic environment. Gazebo is suitable for robots simulation with complex indoor and outdoor environments [7]. Moreover, Gazebo is free to use with the support from the robotics community. The MiniRos was designed by using a CAD assembly file from Solidworks software. Then, all parts are converted to *.dae files and applied physical parameters in Gazebo environment. Robot model consists of five main parts: a base link robot, an IMU link, a lidar link, a depth camera link, four links of wheels connect to the main body (left-front, right-front, left-back, right-back) as shown in Fig. 5.

Xacro (XML Macros) is an XML macro language. Xacro creates XML files that are shorter and more understandable.

MiniRos model builds on xacro code and open-source on the *miniroos_description*¹ package.

III. 2D SIMULTANEOUS LOCALIZATION AND MAPPING

Simultaneous localization and mapping (SLAM) is the problem of building a map of the environment while simultaneously determining the robot's position related to this map. This is the problem of the mobile robot navigating an unknown environment. In time navigation the environment, the robot explores to acquire a map and robot localizes itself using its map at the same time. [8]

Let's denote time by t and the robot location by x_t . The path of the robot moving is given as: $X_T = x_0, x_1, x_2, \dots, x_T$.

Which is T is some terminal time(T might be ∞). The location of the robot at time t denote x_t . When x_0 is the initial location.

Odometry provides the related information between two consecutive locations. The odometry data can be collected from the robot's control signal or wheel encoders signal. Odometry at time t denote u_t , that is motion the between time $t - 1$ and time t :

$$U_T = u_1, u_2, u_3, \dots, u_T \quad (13)$$

The environment includes landmarks, surfaces, objects. The map of the environment denotes m . The robot measurements Z_T provide information between features in m and the robot location x_t . At each point of time, the robot takes a measurement. Then the sequence:

$$Z_T = z_1, z_2, z_3, \dots, z_T \quad (14)$$

A. OpenSLAM_Gmapping

The MiniRos mapping with Rao-Blackwellized particle filters (RBPF) [9]. This SLAM method have already implemented on ROS with the name *OpenSLAM_Gmapping* package [10]. The following factorization is used by the RBPF for SLAM as shown in Eq[15]

$$\begin{aligned} p(x_{1:t}, m | z_{1:t}, u_{1:t-1}) &= \\ p(m | x_{1:t}, z_{1:t}) \cdot p(x_{1:t} | z_{1:t}, u_{1:t-1}) \end{aligned} \quad (15)$$

The RBPF are using sampling importance re-sampling (SIR) method [9]. It is updating the set of samples that represents the posterior about the robot path and the map based on the odometry and sensor data. A robot pose is estimated in each particle, denoting $x_t^{[k]}$, and a set of Kalman filters with mean $\mu_{t,n}^k$ and covariance $\Sigma_{t,n}^k$ one for each map feature m_j , which $[k]$ is the index of the sample.

At any point in time, the RBPF maintains k particles: $X_t^{[k]}, \mu_{t,1}^k, \dots, \mu_{t,N}^k, \Sigma_{t,1}^k, \dots, \Sigma_{t,N}^k$. Here $n(1 \leq n \leq N)$ the index of the landmark.

In the beginning, all particles of the robot location starting at the zero on the map. The particles update then process as follows four steps: Sampling → Important weighting → Re-sampling → Map estimation. Finally, the compute the map

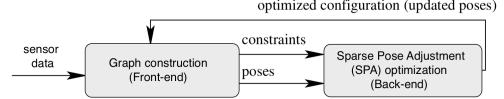


Fig. 6. Simple Karto graph-based SLAM system

$p(m^{[k]} | x_{0:t}, z_{1:t})$ given that the observations and the trajectory with corresponding a particle.

B. Hector SLAM

Hector SLAM associates a robust scan matching approach using a 3D attitude estimation system based on inertial sensing with a LIDAR system [11]. It uses a multi-resolution grid and an efficient estimation of map gradients that can provide reliable localization and mapping capabilities in challenging geographical environments. Gauss-Newton approach is used to improve the alignment of scan beam endpoints. State estimation of the robot is combined with the inertial measurement system (IMU) by applying Extended Kalman Filter (EKF). Moreover, this SLAM algorithms supports multi-resolution map representation, different maps are simultaneously updated and kept in memory which are using the pose estimates generated by the alignment progress. This method ensures that maps remain consistent across map sizes while avoiding expensive down-sampling processes.

C. Karto SLAM

Karto SLAM is a graph optimization method that uses Sparse Pose Adjustment (SPA) based on a nonlinear optimization system to solve 2D pose graphs [12]. A simple Karto graph-based SLAM includes two main parts as shown in Fig. 6: the front-end deals directly with sensor data, while the back-end corrects the robot's poses in order to obtain a consistent map of the environment given the constraints. SPA relies on sparse non-iterative Cholesky decomposition and efficient linear matrix construction to precisely show and solve large sparse pose graphs. A robot pose graph is a collection of robot poses that are linked by nonlinear constraints derived from the measurement of features in neighboring poses.

D. SLAM Computational Experiments

In this section, the study present experiments where this study compare three state-of-the art SLAM approaches.

- OpenSLAM_Gmapping [10]
- Hector SLAM [11]
- Karto graph-based SLAM [12]

The study evaluates both the computational and the memory requirements based on a small map and a large map. The small map is TurtleBot3 World [13] with an approximate area of $30m^2$, which includes small objects and barrier cover around the robot. The large map is Indoor Structure World from Clearpath Robotics [14] with an approximate area of $490m^2$, which includes tables, chairs, walls, rooms, doorways, wooden frames, sandbags, windows, rubble, and more. All experiments are executed on a Ryzen 3 3100 @4 cores-8 threads running

¹https://github.com/robotlabvn/miniroos/tree/master/miniroos_simulation

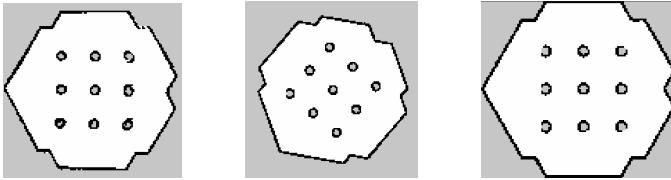


Fig. 7. TurtleBot3 World small map was built based on Gmapping (left), Hector SLAM (middle), Karto SLAM (right)

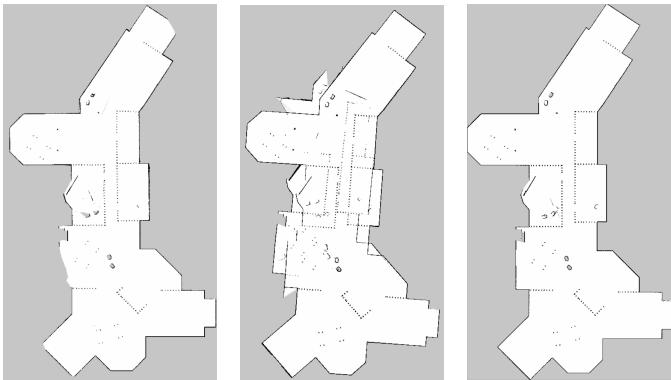


Fig. 8. Indoor Structure World large map was built based on Gmapping (left), Hector SLAM (middle), Karto SLAM (right)

at 3.6GHz with 8GB DDR4 memory. To get the similarity simulation experiment, each SLAM approach is running with only one thread of the CPU.

The result of each SLAM algorithm in a small map and a large map as shown in Fig. 7 and Fig. 8, respectively. Robot is moving with an average linear velocity around $0.4m/s$ and angular velocity around $0.8rad/s$. The map quality of each SLAM method has almost the same quality. However, when the robot mapping the large map, the Hector SLAM appears the "ghosting" in the middle of the map, whereas the laser data measurement is not matched with other parts in the map, and the odometry is affected by a high friction environment.

Each SLAM method performance computational and memory requirements differ based on the size of each map, as shown in Fig. 9 and Fig. 10. Each SLAM algorithms takes approximately 96s and 283s to finish the small map and large map, respectively. In a small map, Gmapping is the highest consume CPU processing power, which fluctuates around 60% CPU, the second highest is Hector SLAM which oscillates

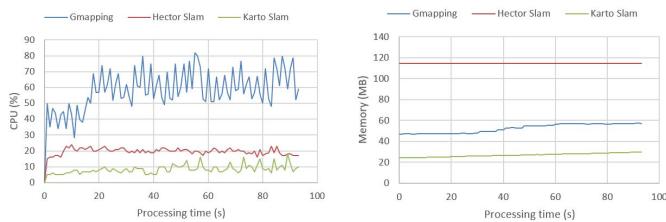


Fig. 9. TurtleBot3 World small map with computational and memory requirements of each SLAM methods

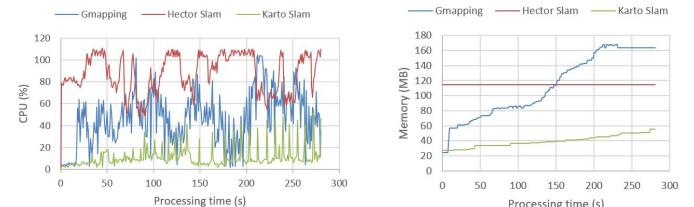


Fig. 10. Indoor Structure World large map with computational and memory requirements of each SLAM methods

slightly around 20% CPU, the less consuming CPU is Karto SLAM with less than 20% CPU power. On the other hand, in the large environment, the Hector SLAM and Gmapping required more CPU consumption, which dramatically oscillates around 80% CPU and 50% CPU, respectively. The main advantage of Hector SLAM is consistent 120MB memory size even small or large maps. Karto SLAM with more environment landmarks, the slightly increasing CPU and memory requirements, below 40% CPU and 60MB memory. Moreover, this graph SLAM optimization method has a greater improvement than other SLAM methods in a large environment.

IV. MINIROS NAVIGATION

After building the map by using the SLAM package, the MiniRos gets sensor data from odometry, velocity commands, and distance sensor streams to send velocity to a mobile base. The robot can easily avoid walls and autonomously navigate based on the request of the user. This process is done by using the ROS Navigation Stack [15]². The Navigation Stack requires to be configured for the dynamics and the shape of the robot achieve at a high-level [16]. The Navigation Stack of the robot can be described on a simple block diagram as shown in Fig. 12 and the main function of each element is presented below:

A. Robot data information

Navigation Stack needs robot's map, sensor, odometry information and transform configuration to send the velocity commands to base controller. Miniros publishes the list of topic below to Navigation Stack: */tf* is transformation frame, */odom* is odometry data, */scan* is laser scan data, */map* is the 2D map generated by the robot, */goal* is the user command pose.

B. Costmap

The costmap includes local costmap and global costmap, and this uses to store obstacles information in the world. Local costmap is used for obstacle avoidance, and local planning, and the global is used for global planing over a complete environment. The costmap is initialized with the static map, normally this is provided by user. When a robot is moving, new scan data publish to maintain and update the robot's local and global position in the environment. The costmap in this Navigation Stack is a two-dimensional

²<http://wiki.ros.org/navigation>

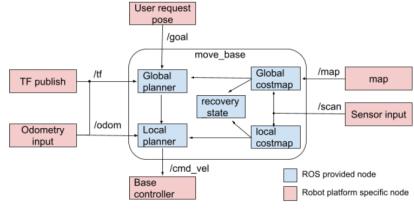


Fig. 11. Simple block diagram navigation of robot

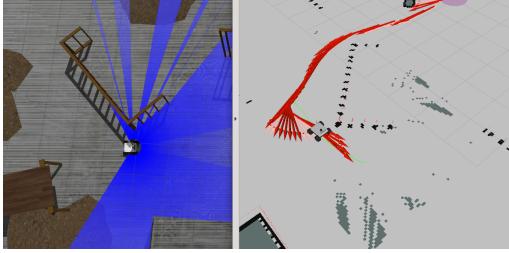


Fig. 12. Experiment MiniRos running navigation to avoid obstacles

structure. In some cases, the three-dimensional can be used with the Voxel grid. Miniros costmap configuration is stored in *miniros_navigation* packages [1] that includes five configuration files: *costmap_common.yaml*, *costmap_exploration.yaml*, *costmap_global_laser.yaml*, *costmap_global_static.yaml*, *costmap_local.yaml*.

C. Global and Local Planner

The Global planner is provided the cost information, the obstacle in the costmap, the robot's localization, and the user request pose. From these data, this generates a high-level plan for the robot moving to reach the goal, normally A* algorithm using for global planner [16]. The Local planner is able for create velocity commands for the robot to move toward the goal to avoid obstacles. The Dynamic Window Approach(DWA) is used for generating local planners based on cost function [17]. Following configuration file for MiniRos planner *planner.yaml*³. MiniRos can successfully point-to-point traverse the environment by using Navigation Stack to generate an optimal path as shown in Fig. 12.

V. CONCLUSION

The study have published "MiniRos: an autonomous UGV robot for education and research". In the first step, the study have targeted an autonomous robot application here, which have provided the MiniRos simulation environment on Gazebo with SLAM and Navigation package included. Moreover, the study considers this model with hardware architecture and kinematic skid-steering, which can be extended to autonomy in another kind of UGV with more payload and working in an outdoor environment. The MiniRos platform provides an excellent test-bed for illustrating a wide variety of robotics research projects. In the next step, our research direction is

currently working to finish the real hardware and on a series of exciting research directions including simulation multi-robot navigation, decentralized, and collaborative multi-robot manipulation.

All materials are accessible under the open-source ROS package, and the materials can be found at the Github repository <https://github.com/robotlabvn/miniros>. It can also be useful for the robotics community for research and education purposes.

VI. ACKNOWLEDGMENT

We would like to extend our thanks to the engineer of the robotics laboratory of Vietnamese-German University and robotics community Robotlab.vn for their help in offering our resources in running the study.

REFERENCES

- [1] J. Yi, D. Song, J. Zhang, and Z. Goodwin, "Adaptive trajectory tracking control of skid-steered mobile robots," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 2605–2610.
- [2] L. Paull, J. Tani, and Ahn, "Duckietown: An open, inexpensive and flexible platform for autonomy education and research," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1497–1504.
- [3] Microchip, "Atmega168a/328/p," Available at <https://www.microchip.com/>
- [4] I. Rectifier, "Data sheet irf3205," Available at <https://www.infineon.com/>
- [5] T. Wang, Y. Wu, J. Liang, C. Han, J. Chen, and Q. Zhao, "Analysis and experimental kinematics of a skid-steering wheeled robot based on a laser scanner sensor," *Sensors*, vol. 15, pp. 9681–9702, 05 2015.
- [6] J. Martínez, A. Mandow, J. Morales, S. Pedraza, and A. García, "Approximating kinematics for tracked mobile robots," *The International Journal of Robotics Research*, vol. 24, pp. 867–878, 10 2005.
- [7] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.
- [8] C. Stachniss, J. Leonard, and S. Thrun, "Simultaneous localization and mapping," pp. 1153–1176, 01 2016.
- [9] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks." Morgan Kaufmann Publishers Inc., 2000, p. 176–183.
- [10] W. B. Giorgio Grisetti, Cyril Stachniss, "Openslam gmapping," Available at <https://openslam.org.github.io/gmapping>
- [11] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 11 2011.
- [12] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2d mapping," 10 2010, pp. 22–29.
- [13] Robotis, "Turtlebot3 package," Available at <https://emanual.robotis.com/>
- [14] C. Robotics, "Gazebo simulation environments," Available at <https://clearpathrobotics.com>
- [15] E. Marder-Eppstein, "Ros 2d navigation stack," Available at <http://wiki.ros.org/navigation>
- [16] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 300–307.
- [17] D. Fox and W. Burgard, "The dynamic window approach to collision avoidance," *Robotics & Automation Magazine, IEEE*, vol. 4, pp. 23 – 33, 04 1997.

³https://github.com/robotlabvn/miniros/tree/master/miniros_navigation/

Certificate

SITRAIN Digital Industry Academy

Bien Minh Tri

Working at Vietnamese-German University

completed the course

SIMATIC S7 – 1500 Programming 1 In The TIA Portal

TIA-PRO1 (5 days)

from 05th October to 04th November 2020.

Main subjects of the course:

- Overview and significant performance characteristics of the SIMATIC S7 system family
- The components of the TIA Portal: STEP 7, WinCC, Start-drive
- Program execution in automation systems
- STEP 7 block types and program structuring
- Binary and digital operations in the function block and ladder diagram (FBD/LAD)
- Programming of parameterizable blocks
- Data management with data blocks
- Programming organization blocks
- Test tools for system information, troubleshooting, and diagnostics
- Hardware configuration and parameterization of the SIMATIC S7 modules, a PROFINET IO system (ET-200), a Touch Panel.
- Program Cycle, Process Image and Process Image Partition
- Explanation and using different organization blocks
- Deeper understanding of contents through practical exercises on the SIMATIC S7-1500 system model

HCMC, 06th November 2020



Serial no. : SMSB2020/21-0286
siemens.com/sitrain

Certificate of Appreciation

This certificate is awarded to

Bien Minh Son Department: RBVH/LESS

To appreciate your contributions toward

INNOVATION ACTIVITIES AND DEVELOPMENT
OF OUR COMPANY

Date: 29th June 2016

Henn





Nov 20, 2021

TRI MINH BIEN

has successfully completed

**Introduction to TensorFlow for Artificial
Intelligence, Machine Learning, and Deep Learning**

an online non-credit course authorized by DeepLearning.AI and offered through Coursera

Laurence Moroney

Laurence Moroney
Lead AI Advocate, Google

COURSE
CERTIFICATE



Verify at coursera.org/verify/C6WDSPX7BKVH

Coursera has confirmed the identity of this individual and their participation in the course.



Nov 28, 2021

TRI MINH BIEN

has successfully completed

Convolutional Neural Networks in TensorFlow

an online non-credit course authorized by DeepLearning.AI and offered through Coursera

A handwritten signature in black ink that reads "Laurence Moroney".

Laurence Moroney
Lead AI Advocate, Google

COURSE CERTIFICATE



Verify at coursera.org/verify/FFHZFB8QZEF

Coursera has confirmed the identity of this individual and their participation in the course.

VERIFIED CERTIFICATE OF COMPLETION

May 19, 2020



Tri Bien Minh

Has successfully completed the

Deep Reinforcement Learning Nanodegree

NANODEGREE PROGRAM



Sebastian Thrun
Founder, Udacity

Co-Created with

Unity NVIDIA Deep Learning Institute

Udacity has confirmed the participation of this individual in this program.
Confirm program completion at confirm.udacity.com/466QEDKQ.

Best Junior Researcher Award

Vietnamese - German University
proudly presents Best Junior Researcher Award to

Mr. Biên Minh Trí

Lab Engineer, Faculty of Engineering

in recognition of excellent achievements
in research activities during the academic year 2020/2021



SOCIALIST REPUBLIC OF VIETNAM

RECTOR
LAC HONG UNIVERSITY

has conferred
THE DEGREE OF ENGINEER
MECHATRONICS

Upon: **Mr. Biên Minh Tri**

Date of birth: 18 June 1991

Year of graduation: 2013

Degree classification: Good

Mode of study: Full-time

Dongnai, 01 October 2013

Reg. No: 13CK0059

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

HIỆU TRƯỞNG
TRƯỜNG ĐẠI HỌC LẠC HỒNG

cấp
BẰNG KỸ SƯ



Cho:

Ngày sinh:

Năm tốt nghiệp:

Xếp loại tốt nghiệp:

Hình thức đào tạo: Chính quy

18/06/1991

2013

Khá

Đồng Nai, ngày 01 tháng 10 năm 2013

Số hiệu: 130375

Số vào sổ cấp bằng: 13CK0059



NGND-TS Đỗ Hữu Tài



CERTIFICATE OF PARTICIPATION

This is to certify that

No: C/OSB-BY/0043

BIEN MINH TRI

successfully participated in the

TRAINING ON BAYSPEC HYPERSPECTRAL SYSTEM

on 6th to 9th JANUARY 2019

AT VIETNAMESE - GERMAN UNIVERSITY (VGU),

THU DAU MOT CITY, BINH DUONG PROVINCE, VIETNAM.

A handwritten signature in black ink, appearing to read "K. Veliayudam".

Training Instructor
Kumar Veliayudam
BaySpec Hyperspectral Expert &
Technical Manager



Training Organizer
APPLICATION TECHNOLOGY &
TELECOMMUNICATION SERVICES JSC (ATCOM)

CERTIFICATE OF RECOGNITION

We hereby express our sincere appreciation to

Mr. Bien Minh Tri

FOR THE CONTRIBUTION AND DEDICATION TO
THE STEM WORKSHOP: "EXPERIENCING ENGINEERING"
AT VIETNAMESE-GERMAN UNIVERSITY

SIGNED BY:



Prof. Dr. Tomas Benz

Benz

DATE: 13/03/2019



Certificate of Completion

Global Entrepreneurship Training 2017 in Vietnam for Sustainable Economic Growth

Under the Global Entrepreneurship Education Program(GEEP)
Handong Global University sponsored by the United Nations Educational,
Scientific and Cultural Organization (UNESCO) and Republic of Korea,
and the Global Citizenship Education Program of the United Nations Academic Impact Korea

It is certified that **Bien Minh Tri** has shown
great devotion and met all of the requirements
with the highest expectation for
the **Global Entrepreneurship Training 2017 in Vietnam**
for Sustainable Economic Growth
offered by Handong Global University in Pohang, Korea,
United Nations Academic Impact Korea,
and Nguyen Tat Thanh University in Ho Chi Minh, Vietnam.

With all the honors, rights, and privileges pertaining therein,
to be presented on the 12th of January
in the year of our Lord, 2017

Dr. Soon Heung Chang
President, Handong Global University

Dr. Young-Gil Kim
Chairman, United Nations Academic Impact Korea



Assoc. Prof. Dr. Nguyen Manh Hung
President, Nguyen Tat Thanh University



JENESYS^{2.0}

Certificate

Mr. BIEN MINH TRI

Science and Technology 18th Batch

From June 24 to July 1, 2014

*This is to certify that the above participant of
JENESYS 2.0 has successfully completed the
programme conducted by Japan International
Cooperation Center.*

山野幸子

Sachiko YAMANO

President

Japan International Cooperation Center

Date: June 30, 2014



Share knowledge and experience.
For our world. For the future.
JAPAN INTERNATIONAL COOPERATION CENTER

Training Certificate

Mr Bien Minh Tri

Faculty of Engineering

has participated in the training

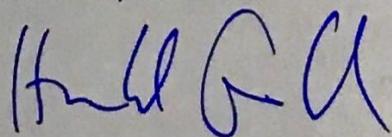
Principles of Good Scientific Practice

with specific topics

- Principles of good scientific research
- Principles of VGU
- International practice examples (German DFG)
- Drafting principles of Scientific Good Practices for VGU

at Vietnamese-German University, Thu Dau Mot City, Binh Duong Province, Vietnam,
on 8th October 2020.

Hannover, 02.11.2020



Dr. Harald Gilch



CERTIFICATE

BIEN MINH TRI

Has completed the course in:

TEACHING FOR QUALITY LEARNING

Organised by Nam Viet Institute of Education Sciences,
June 07th and 14th 2019.

Ho Chi Minh City, July 16th, 2019
INSTRUCTOR

Certificate No: 134/2019

Dr. Kim D. Nguyen



CHỨNG NHẬN

BIỆN MINH TRÍ

Đã hoàn thành khoá học:

CHIẾN LƯỢC GIẢNG DẠY GIÚP SINH VIÊN
HỌC TẬP CHẤT LƯỢNG

Do Viện Khoa học Giáo dục Nam Việt tổ chức,
ngày 07 và 14 tháng 6 năm 2019.

Tp. Hồ Chí Minh, ngày 16 tháng 7 năm 2019



Số chứng nhận: 134/2019

TS. Vương Văn Cho



GIẤY CHỨNG NHẬN

Chứng nhận

Ông: Biện Minh Trí

Trường Đại học Việt Đức (VGDU)

Đã tham dự
Khóa huấn luyện đào tạo sử dụng UAV eBee+

Tại Bình Dương
Từ ngày 07 đến ngày 09 tháng 01 năm 2019



Nguyễn Hữu Tùng
Trưởng Phòng Kỹ thuật

Nguyễn Anh Tuấn
Giám đốc Công ty ATCOM



Vietnamese - German University

Certificate of Contribution

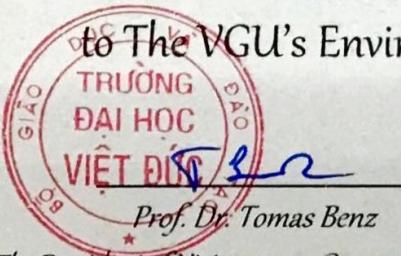


Presented to:

Biên Minh Trí

Faculty of Engineering

In appreciation of your contribution:
GREEN Inspiration video



The President of Vietnamese - German University

