

Integrating earthquake early warning system and a smart robot for post-earthquake automated inspection and emergency response

Tzu-Hsuan Lin (✉ cometlin@gmail.com)

National Central University <https://orcid.org/0000-0001-7833-5394>

Jing-Ting Huang

National Central University

Alan Putranto

National Central University

Research Article

Keywords: Earthquake hazard, Earthquake early warning system, Emergency response, Inspection, Smart robot

Posted Date: May 7th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-450256/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Version of Record: A version of this preprint was published at Natural Hazards on August 13th, 2021. See the published version at <https://doi.org/10.1007/s11069-021-04969-2>.

Integrating earthquake early warning system and a smart robot for post-earthquake automated inspection and emergency response

Tzu-Hsuan Lin¹, Jing-Ting Huang², Alan Putranto³

^{1,2,3} Department of Civil Engineering, National Central University, Taoyuan 32011, Taiwan

Corresponding author

E-mail address: cornetlin@gmail.com (T.-H. Lin)

Abstract

The natural hazard, mainly earthquake, has caused substantial economic losses and human life loss to many countries. Taiwan, which is located on the western Circum-Pacific seismic belt, has encountered the problem as mentioned earlier in Meishan, Hsinchu-Taichung, and Chi-Chi earthquakes a few years ago. In this study, the researchers propose a novel robot-event integrated system capable of doing the automated inspection and emergency response due to a significant earthquake. When the household's earthquake warning receiving device picks up an alert, its built-in wireless communications system will send a signal to the robot. The robot commences inspection of the indoor area via real-time image recognition and tracking. It will approach them upon detecting fallen people, regulating their movements via a robot operating system (ROS) monitoring interface. The robot is designed to operate in a house that remains standing with acceptable damage in which the furniture might falling and injure the occupants after an earthquake hit. The indoor experiment conducted to verify the robot system and operation with a designed condition such as fallen and non-fallen people as a detected object. The robot tested to deliver food or medicine for fallen people while waiting for rescuers to arrive. Tests indicate that the proposed smart robot has prospective implementation to the real-world application with more research and development. The smart robot integrated with an earthquake early warning system has a promising approach to the temporary care of people affected by earthquakes.

Keywords Earthquake hazard, Earthquake early warning system, Emergency response, Inspection, Smart robot

1. Introduction

For the last few decades, the natural hazard such earthquake has caused huge economic losses and human life loss. The increasing worldwide population in high hazard and vulnerability areas became the main factor of the abovementioned problems. Lessons learned from major earthquake in Taiwan such as Meishan earthquake in 1906 ($M_w = 7.06$), Hsinchu-Taichung earthquake in 1935 ($M_w = 7.06$), and Chi-Chi earthquake in 1999 ($M_w = 7.45$) (Chen and Chang 2017). The total death of those major earthquakes reported in (Wu et al. 2013) is 1,258, 3,276, 2,455 deaths, respectively. Most earthquakes hit and affect buildings and infrastructures in Taiwan due to Taiwan's location on the western Circum-Pacific seismic belt under NW-SE compression with a 7 cm/year convergence rate (Wu et al. 1999). The losses push Taiwan's researchers to develop mitigation-based technology to reduce future earthquake hazards. The Central Weather Bureau (CWB) of Taiwan developed and implemented an earthquake early warning system (EWS). Three physical bases for earthquake EWS are described as follows: a) a strong ground movement due to the earthquake is caused by shear (S) waves and surface wave, b) typical crustal P-wave (primary wave) velocity is about 6-8 km/s in which S- and surface waves progressed half of P-waves' speed, and 3) seismic wave velocities have slower movement than transmitted signals from devices such as telegram, telephone, or radio about 300,000 km/s (Wu et al. 2007). The current established earthquake EWS is a ten-second alert system in advance of an imminent tremor in Taiwan's eastern regions. The earthquake EWS involves lining of a 620 km fibre-optic cable network along eastern Taiwan's coastal areas.

The earthquake EWS has detected earthquake occurrence during 2006-2012 is 207 earthquakes around Taiwan. 196 of 207 earthquakes were successfully detected with a 95% success rate while 5% offshore and magnitude smaller than five could not be detected with virtual subnetwork (VSN) system (Wu et al. 2013). Thus, the current earthquake EWS plays a vital role in reducing and mitigating earthquake disaster impact. Recent technology can improve not only one-way alerting devices but multiple devices to warn people at one time. Therefore, it is essential to provide especially older and disabled people living alone with higher risk with services such as earthquake alerts, post-earthquake inspections, supplies, and standard relief approaches. For instance, a compound earthquake early-warning system – taking advantage of the slower velocity of some major seismic waves such as S waves and surface waves – could alert older people living alone tens of seconds before an earthquake is felt locally, via lights, sounds, or electronic bulletins, so that they can take early action and thus reduce their injury risk. On the other hand, in the period of post-hazardous disaster due to natural catastrophe is usually found difficult to conducts search and rescue of victims. Recent developments in the field of disaster technology have led to a renewed interest in robot-based assistive.

Over the past few decades most research in post-hazardous disaster has emphasized the use of technical solution-based robot technology to assist rescue operations and investigate the hazardous disaster impact. On 27 October 2005, three days after Hurricane Wilma, a category five storm cause landfall at Cape Romano, Florida. A robotic team from the University of South Florida deployed an uncrewed sea surface vehicle (USV) and a rotary-wing micro aerial vehicle (MAV) to assist port post-disaster assessment (Murphy et al. 2008). March-October, 2011, a joint US-Japan team applied to ground and underwater robots for inspection after the great eastern Japan earthquake with M_w 9.0-9.1 (Matsuno et al. 2014). In July 2012, a group consisted of humans-robot deployed by NIFTi work together with the Italian National Fire Corps (CNVVF) to inspect Mirandola red area. Mirandola, Emilia-Romagna region, Northern Italy, was the area which has numerous historical building and experienced damages due to two major earthquake occurred in May 2012 (Kruijff et al. 2012). The human-robot application mentioned before shows great concern and huge contribution of human-robot operation for inspection in post-disaster of United States, East Asia, and Europe (Cubber et al. 2017). The robots mentioned before were designed to operate in outdoor conditions with human-robot mechanism operation. However, far too little attention has been paid to the indoor robot service development to monitor resident condition that affected by the natural catastrophe.

Therefore, researchers have been developed a wheeled smart robot furnished with five DOF robot arm for indoor daily service application and found capable of helping disabled and older people (Balaguer et al. 2005). The robot design has a locking mechanism to connect the robot arm to a wheelchair which has electronic equipment on board to control its movement. Meanwhile, another robot has successfully of handling even more sophisticated household work. With a seven-axis robotic arm on a mobile platform and visual capability could deliver drinks directly to people (Reiser et al. 2009), taking advantage of RGB-D camera to track the positions of people's mouths, make it possible as robot feeding assistance (Fang et al. 2018). Some existing mobile robots can also detect fallen people in real-time via a system comprising a Kinect depth camera, cloud-point cutting, retrieval of layer features, and a support vector machine (SMV) (Volkhardt et al. 2013; Antonello et al. 2017; Maldonado-Bascón et al. 2019). Another approach, which integrates a Kinect depth camera with a convolutional neural network (CNN), rebuilds 3D human gestures based on the 2D model to ascertain whether a fallen object is human was conducted (Solbach and Tsotsos 2017). Moreover, researchers have been developed the robot to detect fallen people in real-time with a low-cost furnished with an RGB-D camera, a Raspberry Pi 3 embedded single-board computer, and a robot operating system (ROS). This system enquires about fallen people via a present audio frequency and then sends its photos and audio frequency data to a mobile phone app, alerting victims' family members or other caregivers to come to their rescue (Ciabattini et al. 2017; 2018). The studies presented thus far provide evidence that the application of robot fallen people detection is very useful in term of unexpected situation such as earthquake hit which may cause the resident unconscious and fall down on floor.

To the best of the authors' knowledge, most studies in robot fallen people detection only focus on detects and reports fallen people to family members or other caregivers. However, much uncertainty still exists about the causes and effects of fallen people with the robot that detect them. So far, the success of the robot-based application to detect fallen people never discusses any trigger that can lead to people's fallen or provide a particular treatment to the victim.

Therefore, the following research proposes a novel and robust robot-event integrated system to anticipate unconscious or fallen people due to the earthquake hit. The robot is designed to automatically changes daily service mode into inspection and emergency response mode through a specific trigger of an earthquake magnitude threshold value. The main purpose of this study is to develop a smart robot capable of anticipating the resident experiencing unconscious or difficulty moving around after an earthquake occurs. This research is the first research that develops a robot prototype integrated with an earthquake EWS for indoor service operation to reduce earthquake risk and temporary care of people affected by earthquakes.

2. System Architecture

The proposed system employs an NVIDIA Jetson TX2 embedded development board, a robot operating system (ROS) created by the researchers, and a kinetic official version, all within the Ubuntu 16.04 environment to optimize compatibility and system stability. Various automatic functions were developed in the same environment, including an earthquake early warning interface, a two-wheel differential-driven chassis, mapping and navigation functions, a 6-DOF robotic arm from delivery of relief supplies, a real time image-based object tracker, sensor for data collection, and other automatic functions. The system's functional architecture is presented in Figure 1.

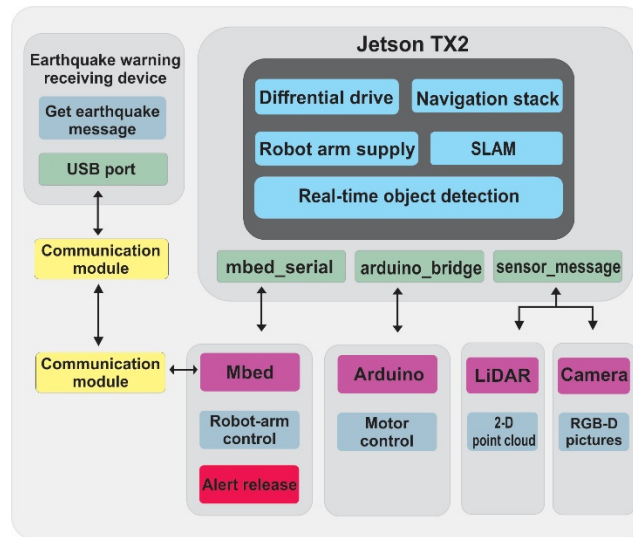


Fig. 1 Architecture of the proposed robot's functions

In practice, after receiving an earthquake warning, the system's earthquake warning receiving device will send a message via its built-in wireless communications device to the robot. When it happens, after activating its buzzer and LED light, the robot will open its simultaneous localization and mapping (SLAM) system for indoor fixed-point cruising, and conduct real-time image identification and tracking via a deep-learning-integrated RGB-D depth camera and Jetson TX2 AI computing device. Upon detecting fallen people, it will approach them, which will be monitored with ROS interface. Since fallen people may be still conscious, but unable to move freely, the robotic arm will distribute food or medicine directly to fallen peoples' faces for them, taking advantage of depth camera and pre-trained Mobilenet-SSD deep learning model to identify the exact positions required, before rescuers arrive. This process is described in Figure 2.

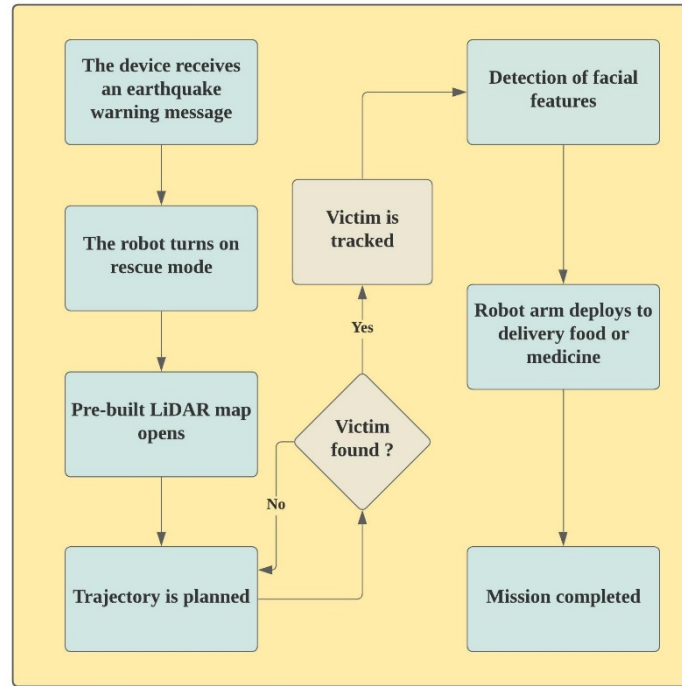


Fig. 2 Operating flow of the proposed system

The system's earthquake early warning interface relies on WebSocket, via an application programming interface (API) connection from compound earthquake early warning platform to the server. A master encryption key retrieved from the WebSocket server will be used to produce a set of encryption keys for return transmission to the Web Client, simulated using a PC in this case. The simulated Web Client then adds account and password information to each encryption key before sending it to the WebSocket server, which will examine whether that key belongs to a legal user before sending a message regarding login success/failure to the Web Client in JSON format. From that point, the Web Client is ready to receive an earthquake message. After receiving such a message, the Web Client sends it via its low-power wide-area network (LoRa) to the robot's LoRa, driven by an Mbed control board; and this action activates the warning light and buzzer that inform the user of the upcoming earthquake. This process can be carried out either within the robot itself, or via transmissions between an existing earthquake early warning device and the robot, as shown in Figure 3.

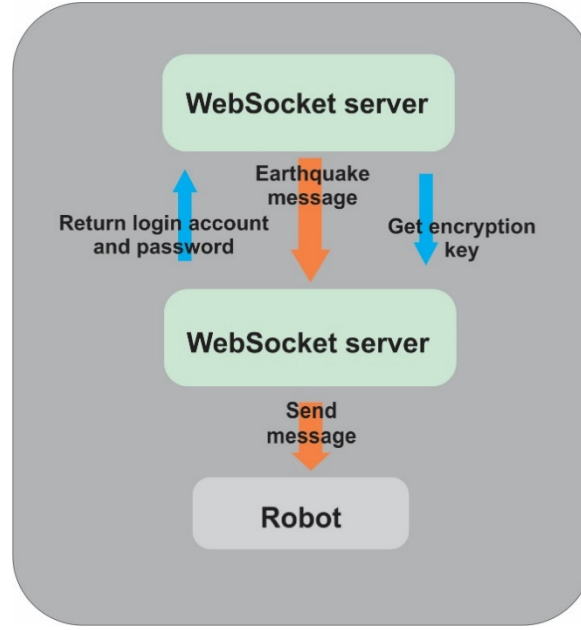


Fig. 3 Architecture of the proposed system's earthquake interface

3. Research Methods

3.1. Determination of the earthquake magnitude threshold

Earthquake magnitude threshold (EMT) is a value that define as a trigger value of robot operation. The value is defined based earthquakes record for the past few years. The earthquake magnitude referring to the previous research and collected from earthquake report in CWB website online. The previous research and present data of earthquake magnitude provided to estimate the threshold value based earthquake intensity around Taiwan for the past few years. The earthquakes magnitude data in this research collected from 2013-2020. The detail will be discussed in result and discussion section.

3.2. Robot operation

The proposed system's chassis employs Arduino as a control system, and features two encoder-furnished DC motors. Its motion is illustrated in Figure 4. The radius of the distance between the two wheels stands at d . The corresponding linear velocities of the left and right wheels reach V_L and V_R , respectively, where V is the chassis' instant linear velocity, and R is its motion radius. For circular motion, the angular velocity of the left and angle wheels is the same, and therefore is also the angular velocity of chassis motion, i.e.,

$$V = \omega R = \frac{V_L + V_R}{2} \quad (1)$$

where ω is its instant angle. Eq. (1) further implies that the chassis' motion mode has the following three states:

1. when $V_L > V_R$ or $V_L < V_R$, the robot will move in a circular arc;
2. when $V_L = V_R$, the robot move in a straight line; and
3. when $V_L = -V_R$, the robot will make a circumferential motion with itself at the centre.

Upon release of the required values of the robot's linear velocity and angular velocity by the Jetson TX2 upper device, communication between the ROS node and the Arduino slave computer can commence via an ROS-Arduino bridge, triggering the latter to implement a kinematic solution that includes velocity values for the left and right wheels, and sending those values to the PID controller. The PID algorithm [Ang et al. 2005] can be written as:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau - K_D \frac{d}{dt} y(t) \quad (2)$$

where $K_P e(t)$ is the proportional gain; $K_I \int_0^t e(\tau) d\tau$ is the integral gain; $K_D \frac{d}{dt} y(t)$ is the derivative gain of process variable, $e(t) = r(t) - y(t)$; $r(t)$ is the reference signal or set point. This algorithm serves as a fundamental basis of PID control, and can cause the chassis to move in a designated pattern by adjusting the rotational velocity of electric machine. When the wheels rotate, the encoder in electric machine counts the number of encoder pulses received in a unit of time, and calculate actual rotation velocity for feedback to Arduino and the upper device's ROS via serial port, before integrating the velocity reported in return with the encoder, and thereby estimating the robot's current location and posture.

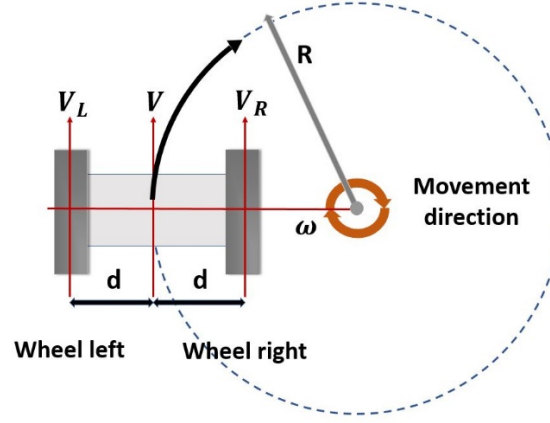


Fig. 4 Motions of the differential-wheel chassis

A SLAM approach was selected, given the robot's purpose, it is likely to start in an environment rendered unfamiliar by earthquake-related debris, and thus be required to determine its location and posture via map features detected while moving, and then conduct mapping incrementally. In other words, it must position itself and map its environs simultaneously. Specifically, the system incorporated GMapping, a popular 2D SLAM algorithm that first calls for the robot to scan its environment with LiDAR and detect which areas are accessible vs. inaccessible to it. GMapping then determines the robot's locations, and undertakes mapping via Rao-Blackwellized particle filter (RBpf) and Extended Kalman Filter (EKF) algorithms. The mathematical model of SLAM mapping, based on Bayes' Theorem (Mullane et al. 2008), is:

$$p(x_{1:t}, m | z_{1:t}, u_{1:t-1}) = p(x_{1:t} | z_{1:t}, u_{1:t-1}) p(m | x_{1:t}, z_{1:t}) \quad (3)$$

Where *a posteriori* probability of t time in SLAM is *a posteriori* probability of $t-1$ time and environmental and motion states; where $u_{1:t}$ is the historical control data from the chassis' odometer; $z_{1:t}$ is the LiDAR data; m is the mapping data; and $x_{1:t}$ is the current position of the robot. Specifically, Eq. (3) shows that z and u can derive x , which – when combined with z – can generate m , thereby estimating the movement track of the robot and providing an up-to-date map based on its known tracks.

The navigation stack was used to navigate the robot from one point to another point. It is a 2D navigation package for obtaining mileage information, sensor data, and target position, and which outputs safe velocity commands for the Arduino motion platform, consisting mainly of positioning and path planning. For the former, the study embraced the adaptive Monte Carlo localization (AMCL) approach, which combines adaptive Kullback-Leibler divergence (KLD) for updating particles with traditional Monte Carlo localization. Unlike the robotic mapping method described above, in which sensor were used for data collection and the map was derived from both sensor data and mapping data; then, the move-base package in the navigation stack is used to obtain further data on the surroundings (such as scanning results) and to generate a global or regional cost map, which helps the robot avoid obstacles and arrive at its designated position safely.

Path planning for the move base can be carried out via either a global or a regional approach, using an A* algorithm (Yao et al. 2010) for the former and a dynamic window approach (DWA) algorithm for the latter, according to the framework shown in Figure 5.

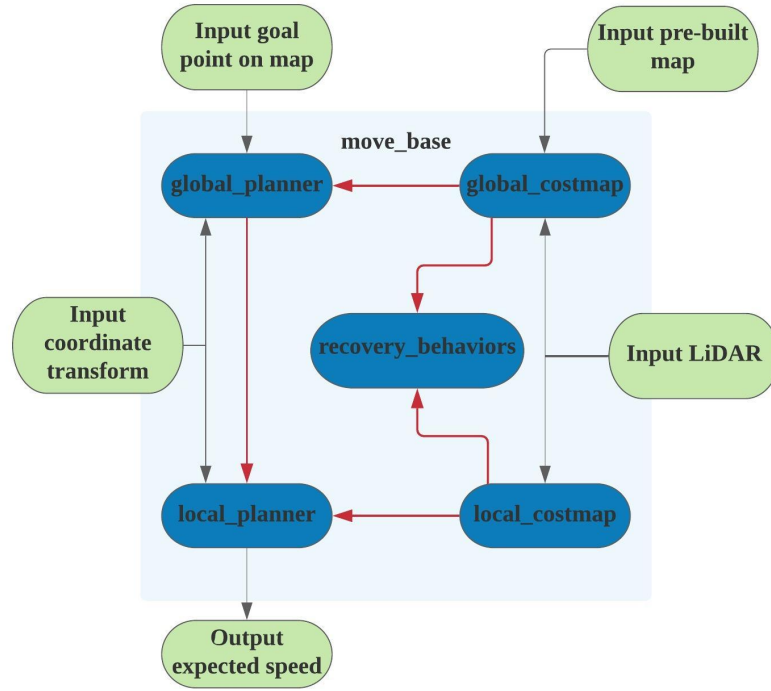


Fig. 5 System framework of move base

From among the multiple motion-planning algorithms in the ROS's MoveIt motion-planning database, the researchers selected Trac-IK Kinematics Solver to plan the motion of their robotic arm. A modified version of the Kinematics and Dynamics Library (KDL) motion compiler, Trac-IK boasts speedier and even more stable performance, and better outcomes when handling joint-limited numerical solutions, all without incurring extra computing time.

After computing the radians of the robotic arm's various joints with Trac-IK, the TX2 master computer can communicate with the control board of the Mbed slave computer via rosserial mbed's open-source ROS node. The latter can convert radian information from the former into pulse width modulation (PWM) values that the servo motor and direct motor should rotate to, inducing the robotic arm to move to the designated position. Figure 6 presents the arm's control flow.

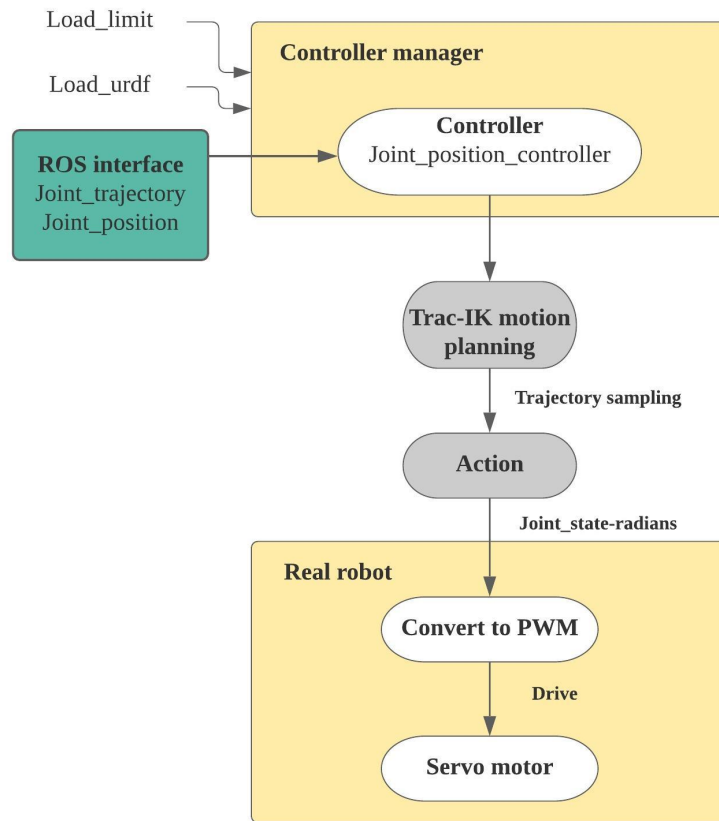


Fig. 6 Control flow for the robotic arm

The robot's embedded Jetson TX2 platform, coupled with a realsense depth camera, forms the core of its real-time object-tracking capability. TensorFlow was used as the framework for the in-depth learning and training of a Mobilenet-SSD V2 advanced training model, which in turn was used to construct the transfer-learning model. Mobilenet-SSD is a lightweight neural-network model capable of making inferences without a high-performance GPU. Specifically, the transfer-learning method of data training involves, first, loading the weighting of a model already trained in advance, and modifying the last several layers of output of the neural network, before using that network to accelerate training with existing data using feature-weighting derived from previous data training. This approach produces good identification results even without a good dataset.

The three general types of targets of this training – fallen people, non-fallen people, and people's mouths – were applied to the deep-learning model in the robot's navigation system, using 3D spatial coordinates derived from point-cloud messages released by the depth camera. Thus, the robot could be directed to proceed to a particular point in the x and y plane-coordinate system of the 3D coordinates.

Another deep-learning model was applied to the robotic arm, retrieving the 3D spatial coordinates of the victims' mouths from the point cloud, and driving the arm to the 3D coordinates where emergency supplies could be found. The architecture of this overall flow is presented in Figure 7.

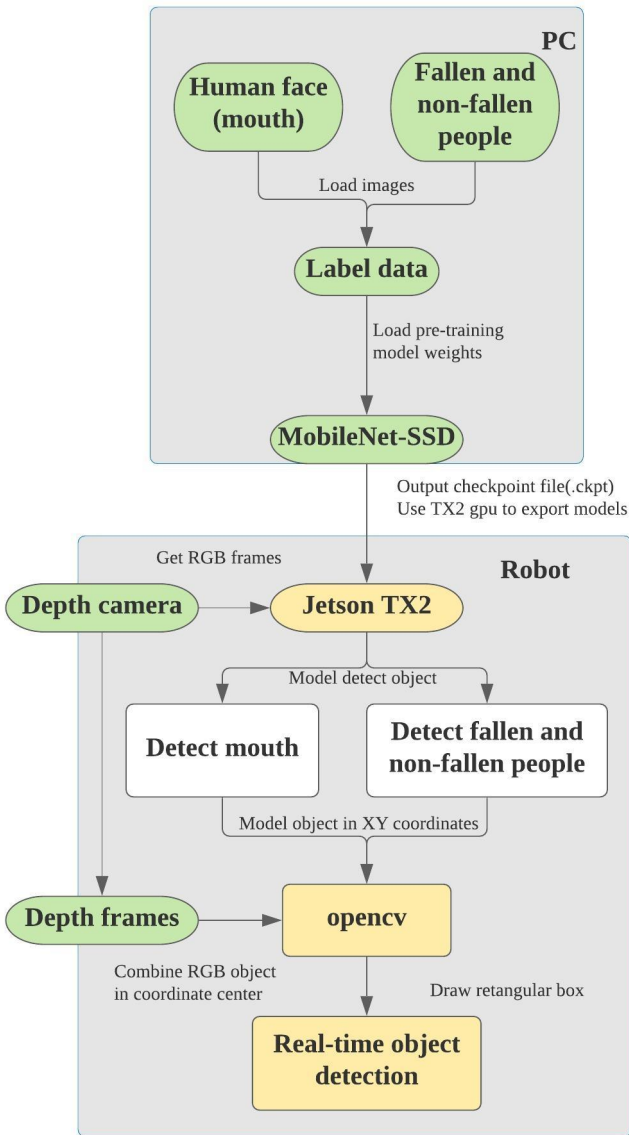


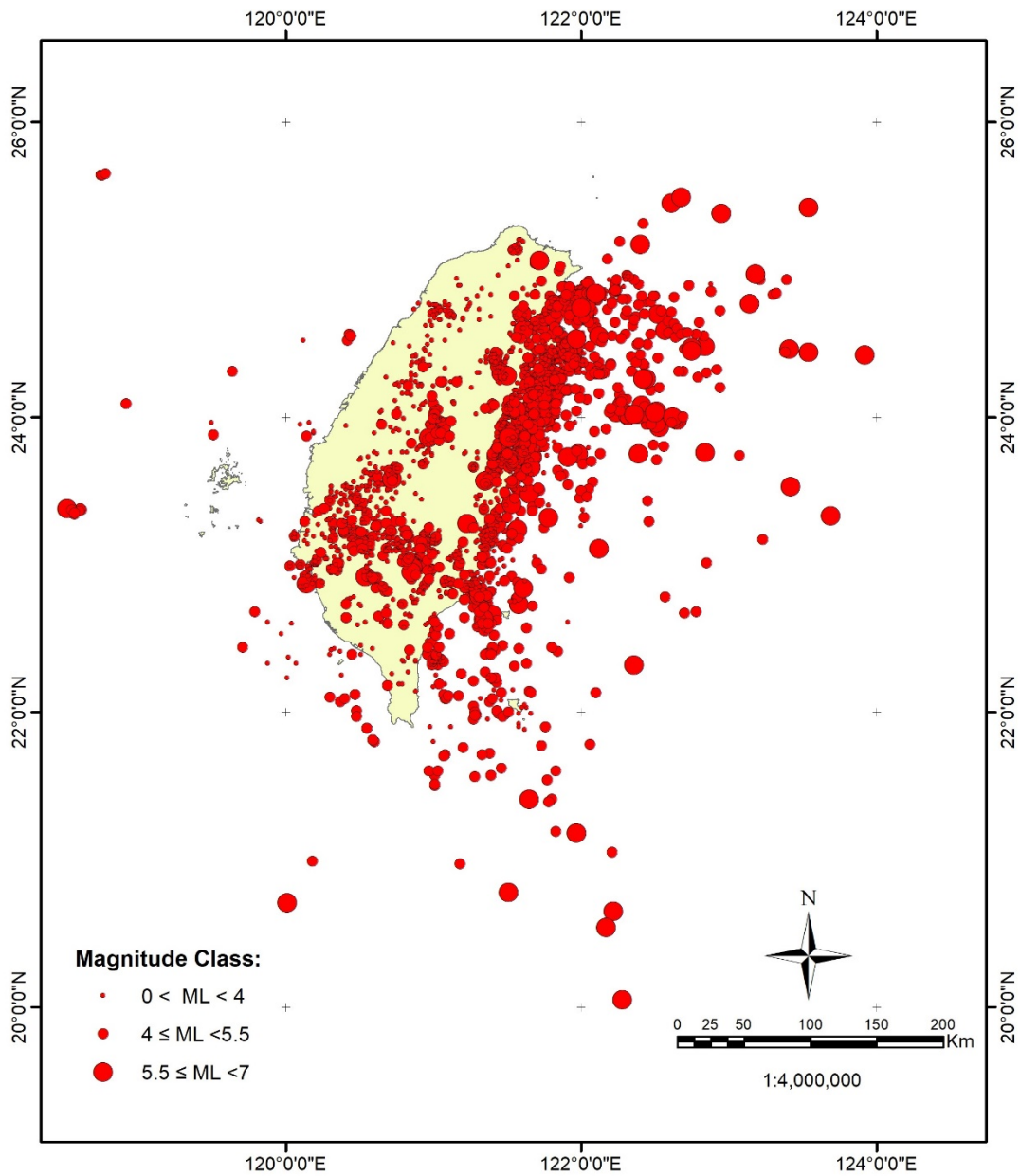
Fig. 7 Flow chart of real-time image identification

4. Result and Discussion

4.1 Earthquake magnitude threshold

From 2006 to 2012, the VSN system successfully detecting, processing, and reporting earthquake occurrence with a 95% success rate for hypocentre and magnitude M_L range 4.5 to 6.9 around Taiwan (Wu et al. 2013). Based on the Richter scale, an earthquake with M_L more than five is classified as a moderate earthquake. In this research, magnitude M_L during the period of 2013 to 2020 were collected from CWB Seismological Centre online system. The earthquake magnitude is classified into three class, the first is earthquakes with magnitude $0 < M_L < 4$; the second is earthquakes with magnitude $4 \leq M_L < 5.5$; and the last is earthquakes with magnitude $5.5 \leq M_L < 7$. The epicentre distribution of earthquakes around Taiwan is provided in Figure 8. There are 4755 earthquakes recorded and reported. During 2013 to 2020, 5.13% of the total earthquake's occurrence classified as moderate to strong earthquakes. Most of those earthquakes occur at a depth of less than 70 km. The detailed information is shown in Figure 9. This research defined the earthquake magnitude M_L more than four as the threshold and as a triggering indicator of automated robot operation to conduct search and temporary care of people affected by earthquakes. The effect of

244 earthquake intensity with magnitude M_L more than four is usually felt by everyone and potentially damaged the building. The
245 integration between earthquake EWS and the smart robot will be discussed in the next section.



246
247 **Fig. 8** Epicentre distribution of the earthquakes during 2013 to 2020 around Taiwan

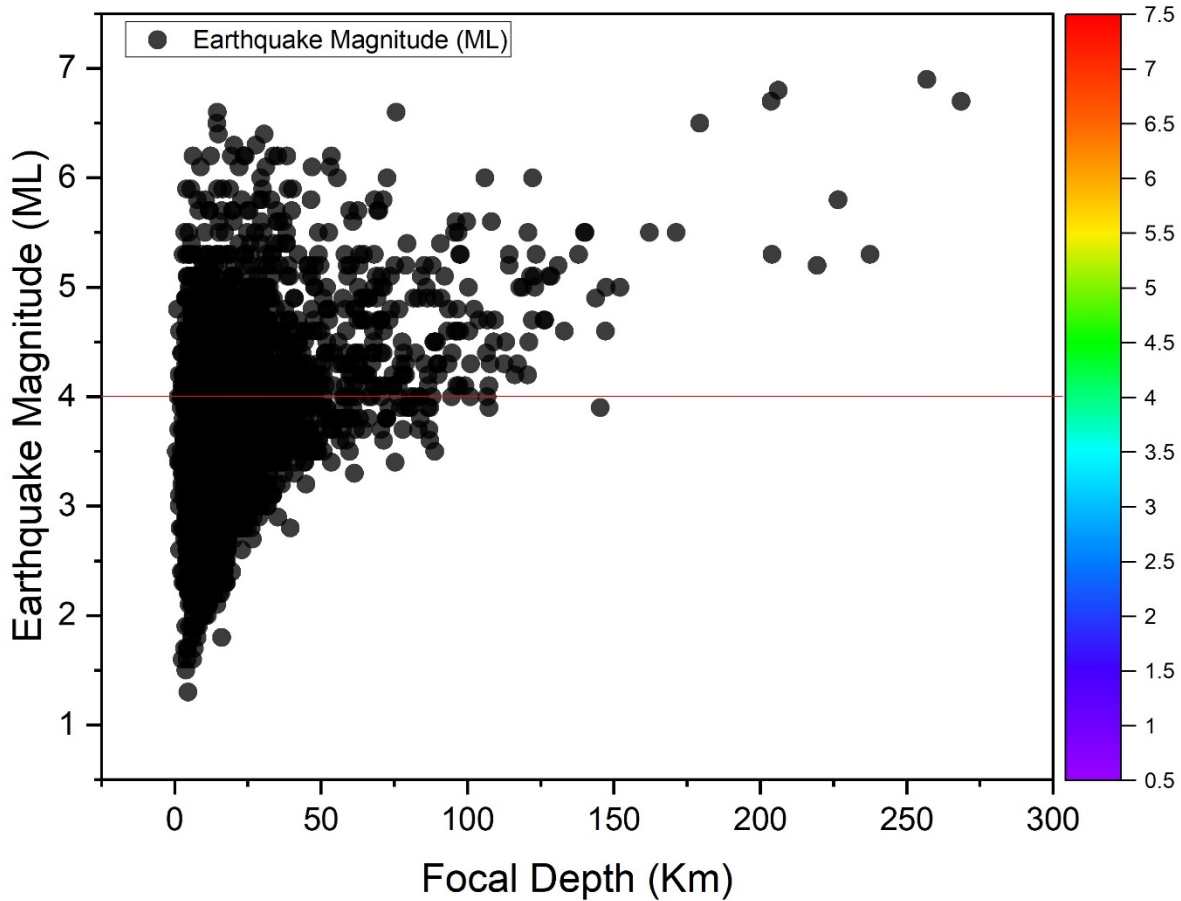


Fig. 9 Focal depth distribution of earthquake magnitude during 2013 to 2020

4.2 System verification of robot operation

Figure 10 shows the robot's overall architecture. Its physical architecture, on the left-hand side, features crawler tracks coupled via three-layer partition plates; a Jetson TX2 controller integrating multiple sensors; and two micro-control development boards. The right-hand side of the same figure presents a 3D schematic, hereafter referred to as the emulation model, of the ROS in unified robot description format (URDF). Once the emulation model is complete and the robot is operating under its ROS, the host computer can monitor the operation of the actual robot and the emulation model simultaneously.

A web client (PC) was employed to emulate the reception device for an earthquake early warning system, as shown in Figure 11. Upon receiving early warning of an earthquake larger than magnitude 4 on the Richter scale, the PC relays that message via the host computer LoRa to the robot, whose LED bulb will transform from green to red, signalling its entry into rescue mode. Its buzzer will also sound for 15 seconds, alerting residents to prepare themselves for a strong earthquake. The threshold value as a trigger in the system can be changed to any value according to user needs.

The robot is expected to complete its SLAM before receiving its first early earthquake warning, so that an indoor plane map will be available to aid its navigation when an earthquake hits. Real-time monitoring of the robot's mapping and movements can be carried out via the monitoring interface of ROS's built-in 3D visualization simulator, Rviz. Specifically, this interface can display topics including robot's movement path; odometer readings reflecting feedback from the moving chassis; the point cloud; and the emulation model. The present study embraced two mobile-mapping approaches: remote keyboard control (see Fig. 12) and remote coordinates release. To ensure the availability of a complete map for automated navigation and inspection, defects in various areas of map were found quickly via manual operation, and rectified through remote control of the robot.

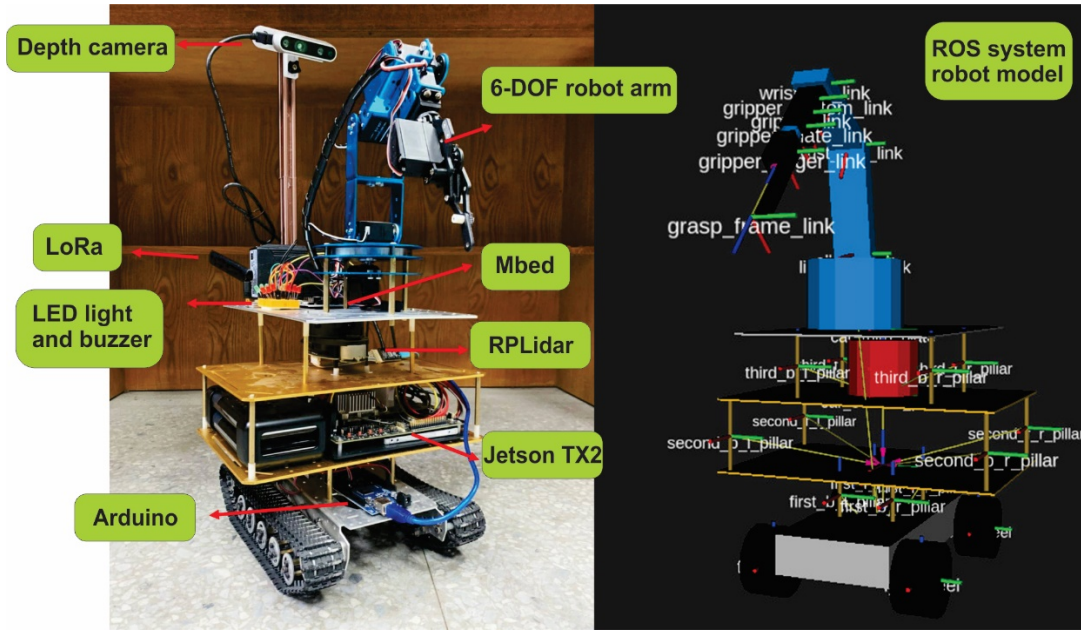


Fig. 10 Overall architecture of the robot

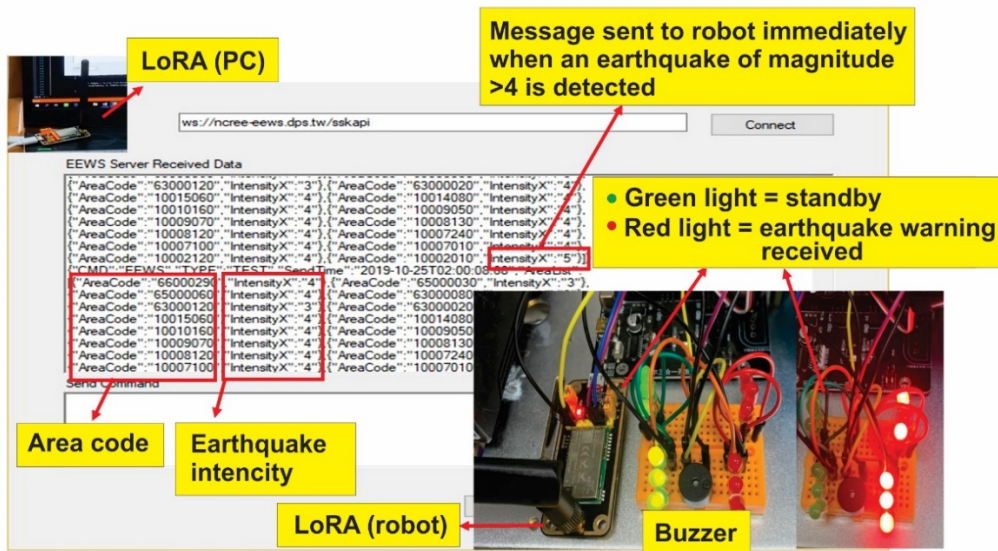


Fig. 11 Early earthquake warning report interface and robot's alarm device

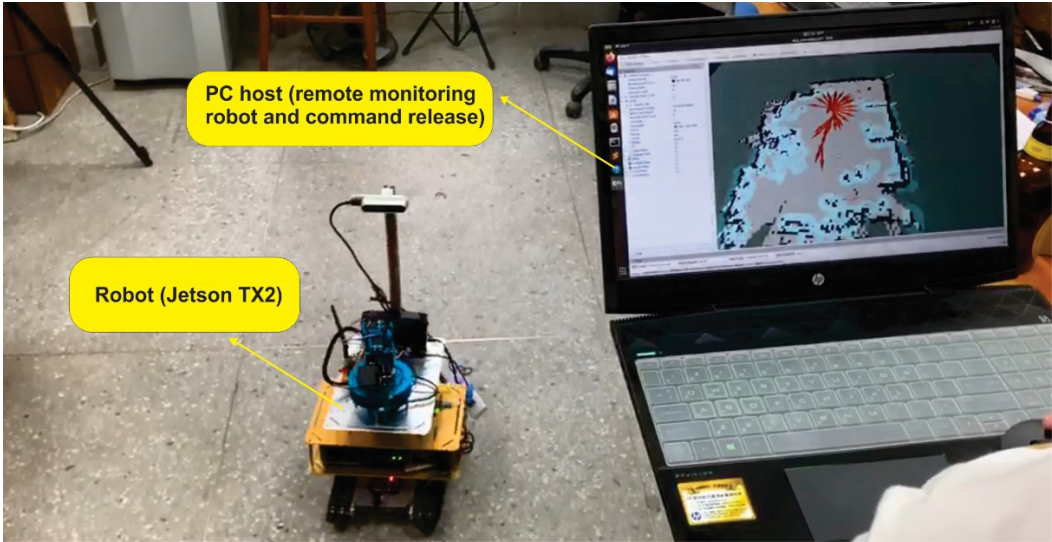


Fig. 12 Remote keyboard control of the robot’s mapping actions

The other approach to manual mapping, remote release of coordinates and use of a local costmap for partial obstacle avoidance, is shown in Figure 13. In it, the green arrow near the top represents manually released navigation coordinates, including x, y, z coordinate points and angles of rotation roll, pitch, and yaw.

Following the release of coordinate points, the robot will automatically plan a navigable path – for example, the one shown in blue in the middle of diagram, to its target position, based on the mileage messages, indicated by the red arrow, that are fed back by the robot chassis repeatedly. After mapping is completed, it is stored for application in subsequent fixed-point navigation. Figure 14 shows the completed mapping of the experimental site.

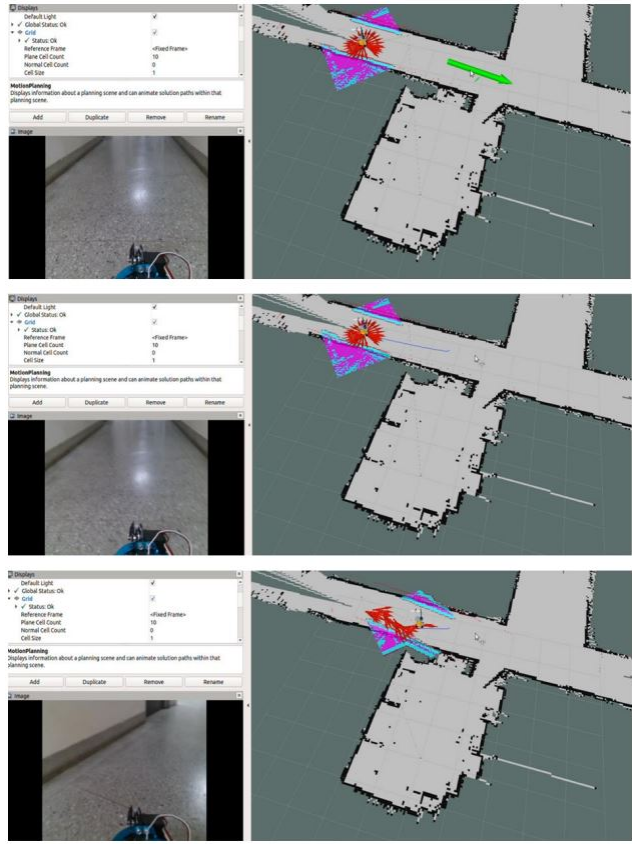


Fig. 13 Remotely released coordinates approach to control of the robot’s mapping actions

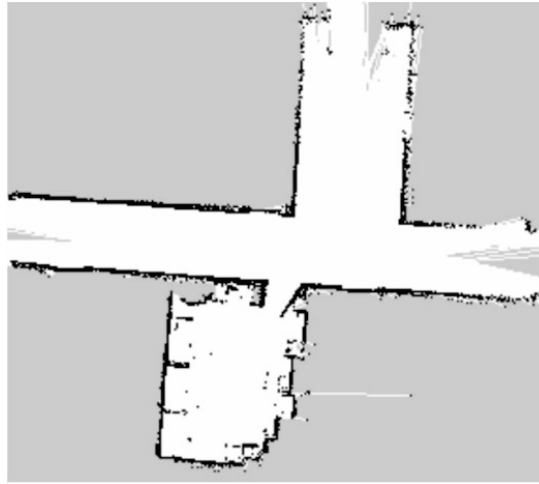


Fig. 14 Complete mapping of the experiment site

The completed map of the experimental site (Fig. 14) was applied to navigation. This process differed from applying the existing plan for navigation using a local costmap for partial obstacle avoidance, in that a global costmap was used for global obstacle avoidance. Information on generated areas was added to the constructed map, enabling the robot to be aware of fixed obstacles and keep its distance from them to avoid collision, supplemented by a local costmap for partial obstacle avoidance. When a new obstacle appeared on the map, it was inflated for avoidance by the robot. However, before navigation, positioning was carried out using the AMCL algorithm, so that the robot could understand its location on the map (Fig. 15). The robot spreads path direction signs around the area to facilitate this positioning before the user releases the coordinate points it needs for cruising. As indicated in Figure 16 by four round red spots, when the robot proceeds to point 1, path directions will converge around it, indicating successful positioning, before the robot proceeds to the other locations in sequentially before the user releases the coordinate points it needs for cruising.

In the relevant phase of this study's experiments, the robot was moved around effectively via remote keyboard control; conducted SLAM for automatic obstacle avoidance via manual remote release of coordinate points; and used the map it had constructed for indoor navigation, avoiding obstacles by changing path automatically whenever new obstacles appeared between it and its target point. The robot proved unable to avoid obstacles less than 20 centimeters in height, as its LiDAR sensor was 20 centimeters above the ground. However, this did not negatively impact its disaster-relief mission of inspecting fixed sites according to coordinate points set in advance.

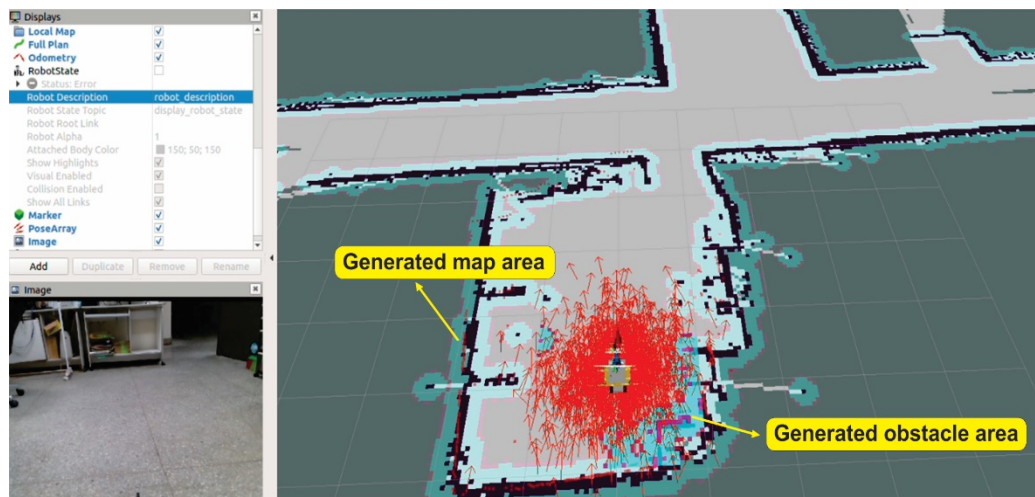


Fig. 15 AMCL positioning

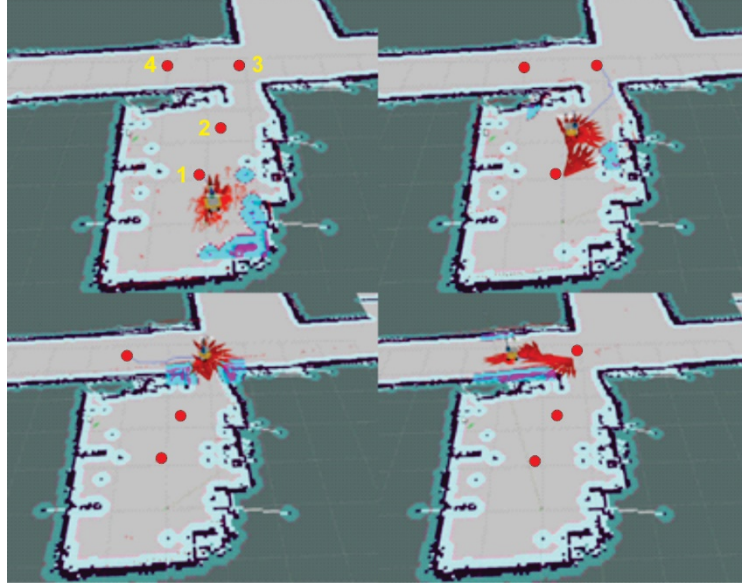


Fig. 16 Robot proceeds to various spots

4.3 Real-time Object Tracking, and Delivery of Supplies by The Robotic Arm

This study divided real-time object tracking into two neural-network models. The first, applied in navigation, was aimed at identifying fallen people while the robot was moving between any two points, so that it could stop to help them before continuing its journey to its original target location. The second model was used when the robot reached any victim, to identify his/her mouth and guide the robot arm to it to deliver relief supplies.

This study's basis model was constructed using 1) a Mobilenet-SSD neural network, 2) a transfer-learning approach to data training, 3) an open-source LabelImg Python image labeling tool, and 4) manual labeling of around 5,000 photos of fallen and non-fallen people and mouth features. The transfer-learning approach was selected due to the insufficient volume of available data, given its efficiency under such conditions.

Applications of neural networks to identify people are quite mature, as evidenced by their widespread use for various purposes. In our experiment, in which cross-entropy was used as the loss function, the robot was able to distinguish fallen from non-fallen people (Fig. 17). However, the final loss of 0.976 suggests that there is considerable room for improvement in the identification rate.



Fig. 17 Identification results, fallen and non-fallen people

Having been trained to detect fallen and non-fallen people, the robot begins to search for the former with the assistance of its navigation system, the robot will automatically release the coordinates of the point to be inspected only after activation of the navigation system. After detecting a fallen person, the robot will plan a new path to proceed to his/her location, rather than continuing along its originally planned route. Because, in map terms, a fallen person represents a new obstacle, a certain distance will be maintained between any such persons and the robot, via the latter's partial-avoidance function. After arriving at the fallen person's position, the robot will return to its pre-set cruising coordinate point, as shown in Figure 18. The experiments demonstrated that the study's robot could reach every coordinate point required to effectively search the experimental space for fallen people. Also, the robot was able to carry out the required supply mission, using the 3D coordinates of the victim's mouth as retrieved by a depth camera. After activation of the supply function, the robotic arm first proceeds to its preset posture. Here, it is important to note that the arm was installed in front of the depth camera, to maximize the latter's field of view; but this required that the arm's joints be set near the bottom of the platform, which limited its stretch.

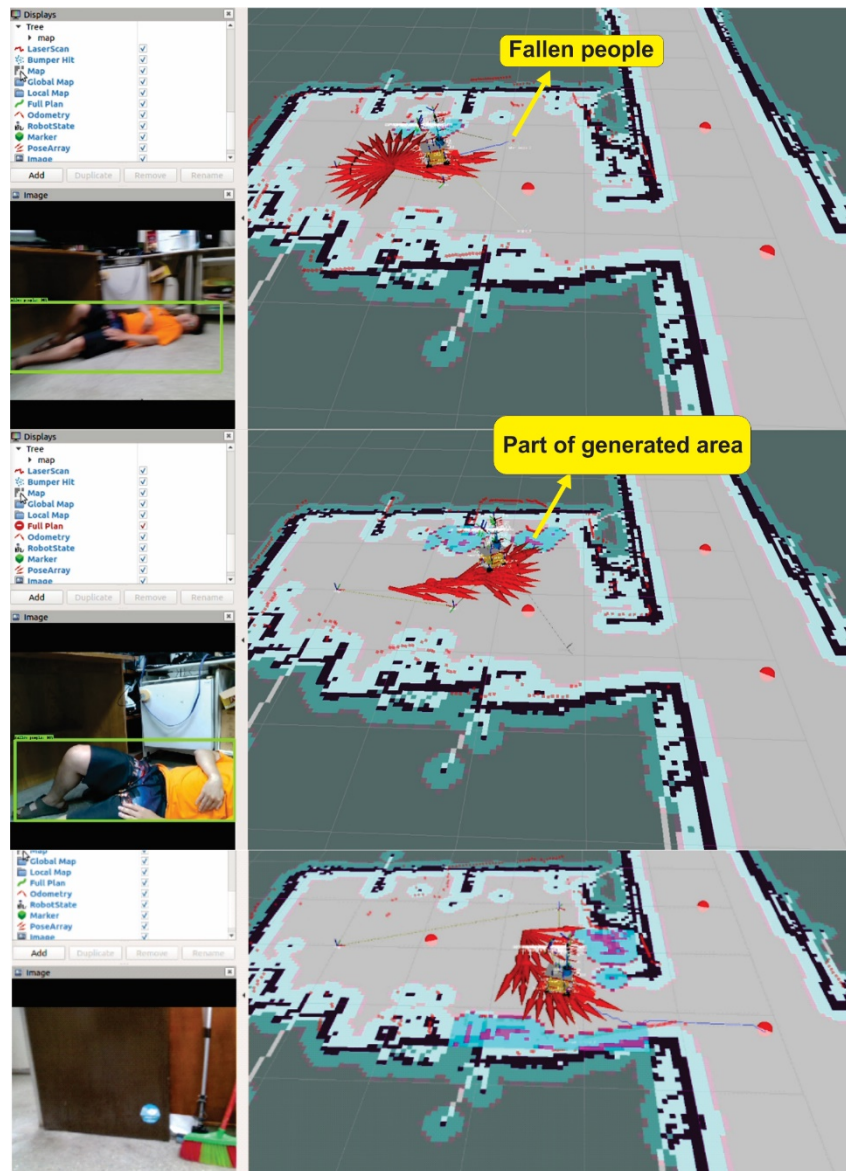


Fig. 18 Path planning for reaching fallen people and proceeding to various points sequentially

Upon detection of a human mouth, the depth camera converts its coordinate point to a 3D one, which the robotic arm can then use to determine the rotation angle for each of its joints via ROS MoveIt's reverse kinematics. The relevant information will then be sent to the Mbed board driving the robotic arm, to enable completion of the supply mission. Mouth-position data will gradually

disappear over approximately 10 seconds, after which the robotic arm will return to its original posture and prepare for its next supply mission, as shown in Figure 20.

Having been created merely for research purposes, the study's robotic arm should not be expected to operate with the same degree of precision as its industrial counterparts. Likewise, the experimental results indicate that there may have been discrepancies in both the installation angle and height of the depth camera, as compared with the ROS's 3D installation model. The robotic arm's actuator also had some difficulty reaching the center of the target object, due to discrepancies between the depth values output by the camera and actual distances, as set forth in detail in Figure 19. The error average in the x, y and z-axis is 1.9 cm, 1.3 cm, and 2.2 cm respectively. The z-axis, which can be described as approaching the target object, had the largest error. Its presumably because the study model's inference image was converted to global coordinates via plane transformation, resulting in rotation errors when the tested object (i.e., a human face) was not pointed directly at the camera lens.

Other constraints on the robot's ability to deliver supplies included the height of the tested object above the floor, due to the shortness of its arm. Also, robotic-arm supply attempts failed when fallen people's mouths were pointing straight upwards, beyond the detection field of the depth camera, which cannot rotate automatically. These mechanism problems will be addressed in future studies.

The robot in this study successfully leveraged the navigation system described above to achieve both automatic obstacle avoidance and targeting of its robotic arm to people's faces. Upon detecting a fallen person's mouth, the robot conducted path planning and proceeded to that point, avoiding obstacles in its path automatically; conducted path planning for its robotic arm; and delivered supplies with that arm, maintaining a certain distance from the fallen person after arrival, since she/he is also a kind of obstacle. The robot also labeled the person's mouth's 3D coordinates in the Rviz monitoring interface. When the mouth is within the arm path plan's reachable scope, the Jetson TX2 host computer released the appropriate joint angles to the Mbed Slave computer, thereby driving the robotic arm to complete its supply mission. The arm in the real world and the one in the simulated ROS environment proceeded to the target point in synchronization, as shown in Figure 21.

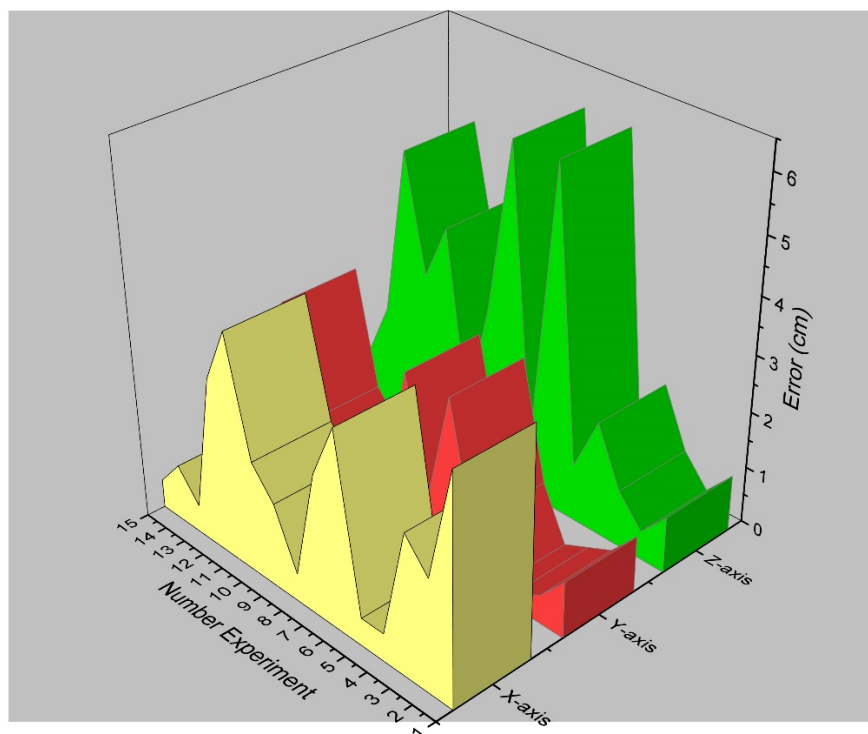


Fig. 19 Absolute value of discrepancies between terminal point of robotic arm and central point of mouth

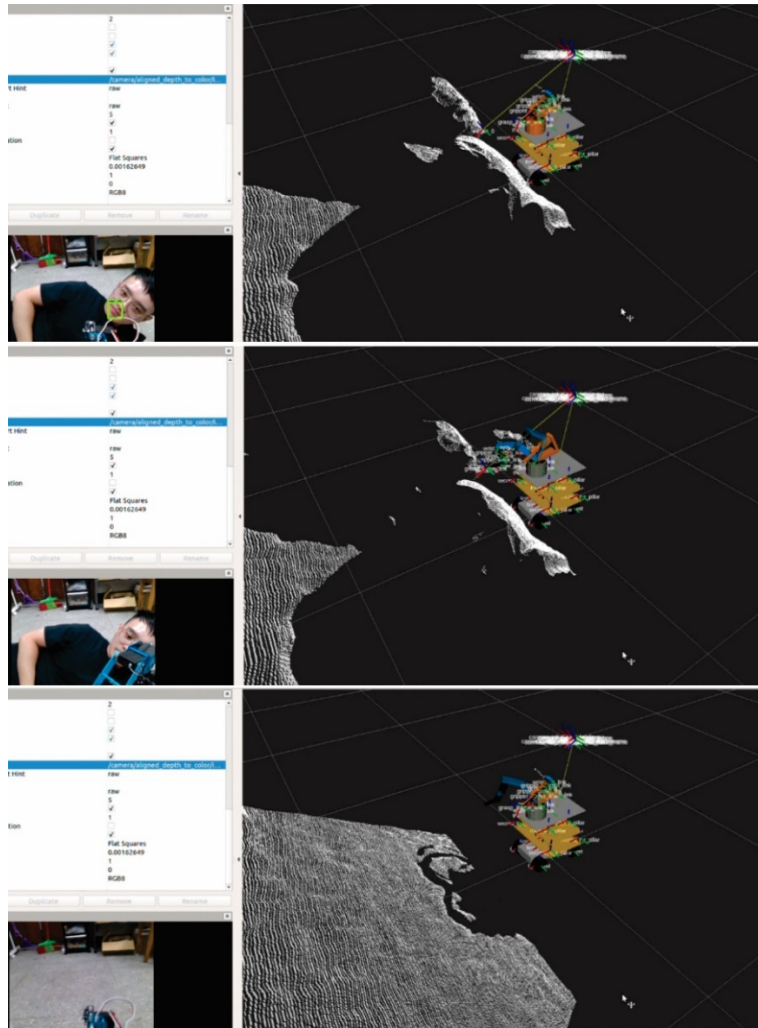


Fig. 20 Robot returning to its original posture after completing a supply mission

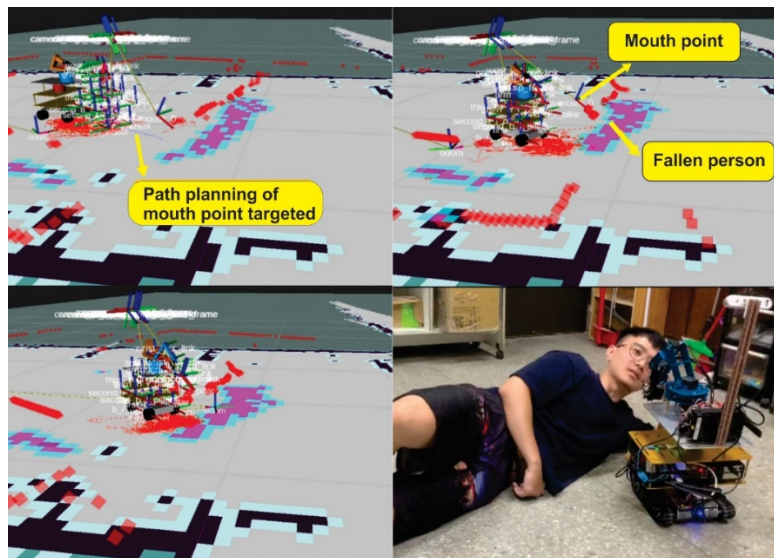


Fig. 21 Test of the robot's automatic navigation and delivery of supplies

5. Conclusion and Future Work

In line with prior applications of robots for search and relief in recent years, this solution took the form of a smart robot integrating Internet of Things (IoT) and artificial intelligence (AI) technologies. Specifically, the proposed first-generation robot includes embedded Arduino and Mbed microcontrollers for driving its chassis and robotic arm, respectively, and a host-computer ROS enabling navigation, automatic obstacle avoidance, and real-time image identification. These features enable it to issue a timely warning when an earthquake hits, conduct fixed-point cruising to search for fallen people, track their mouth features, and deliver relief supplies directly to their mouths with its robotic arm, guided by reverse kinematics. This robot's strong test performance suggests that, with a few modifications, it could be used in various other types of disasters, such as fires in highway tunnels, or – if coupled with ships' automatic identification systems – maritime accidents.

However, further experiments and training will be needed to improve the system's ability to distinguish fallen people from other types of objects. Specifically, subsequent augmentations to the system are expected to include the integration into the ROS of millimeter-wave sensing of the human heartbeat, to reduce mis-identification of non-human objects as fallen people. Also, the existing robotic arm is a simplified type for research purposes, and insufficient for real-world applications in terms of both its motor precision and its length. Nevertheless, it is hoped that this study's robot can serve as groundwork for the development of robots with improved functions for use in a variety of disaster-relief and other applications.

Acknowledgments

The authors would like to thank the Ministry of Science and Technology of the Republic of China, Taiwan, for financially supporting this research (ID:109-2221-E-008-003-MY3).

Data availability statement All relevant data are included in the paper or its supplementary information.

Declarations

Conflict of interest None.

References

- Ang KH, Chong G, Li Y (2005) PID control system analysis, design, and technology. *IEEE Transactions on Control System Technology* 13:1043-1049
- Antonello M, Carraro M, Pierobon M et al (2017) Fast and robust detection of fallen people from a mobile robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, BC, Canada, pp 4159-4166
- Balaguer C, Gimenez A, Jardon A et al (2005) Live experimentation of the service robot applications for elderly people care in home environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, AB, Canada, pp 2345-2350
- Chen KP, Chang WY (2017) Estimating seismic hazards from the catalog of Taiwan earthquakes from 1900 to 2014 in terms of maximum magnitude. *Pure Appl Geophys* 174:1615-1627
- Ciabattini L, Ferracuti F, Foresi G et al (2017) Real-time fall detection system by using mobile robots in smart homes. In *IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Berlin, Germany, pp 15-16.
- Ciabattini L, Foresi G, Monteriù A et al (2018) Fall Detection System by Using Ambient Intelligence and Mobile Robots. In *Zooming Innovation in Consumer Technologies Conference*, Novi Sad, Serbia, pp 130-131
- Cubber GD, Doroftei D, Rudin K et al (2017) Introduction to the use of robotic tools for Search and rescue. From Theory to Practice, IntechOpen. <https://www.intechopen.com/books/search-and-rescue-robotics-from-theory-to-practice/introduction-to-the-use-of-robotic-tools-for-search-and-rescue> . Accessed 10 April 2021
- Fang Q, Kyrarini M, Ristić-Durrant D (2018) RGB-D camera based 3D human mouth detection and tracking towards robotic feeding assistance. In *Proceedings of the 11th Pervasive Technologies Related to Assistive Environments Conference*, Corfu, Greece, pp 391-396
- Kruijff GJM, Pirri F, Gianni M et al (2012) Rescue robots at earthquake-hit Mirandola, Italy: A field report. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, College Station, TX, USA, pp 1-8
- Maldonado-Bascón S, Iglesias-Iglesias C, Martín-Martín P et al (2019) Fallen People Detection Capabilities Using Assistive Robot. *Electronics* 8:915-935
- Matsuno F, Sato N, Kon K et al (2014) Utilization of robot systems in disaster sites of the Great Eastern Japan earthquake. In: Yoshida K., Tadokoro S. (eds) *Field and Service Robotics*. Springer Tracts in Advanced Robotics, vol 92. Springer, Berlin, Heidelberg, pp 1-17
- Mullane J, Vo B, Adams MD et al (2008) A random set formulation for Bayesian SLAM. In *IEEE International Conference on Intelligent Robots and Systems*, Nice, France, pp 1043-1049
- Murphy RR, Steimle E, Griffin C et al (2008) Cooperative use of unmanned sea surface and micro aerial vehicle at Hurricane

Wilma. J Field Robotics 25:164-180

Reiser U, Connette C, Fischer J et al (2009) Care-o-bot® 3-creating a product vision for service robot applications by integrating design and technology. In IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, pp 1992-1998

Solbach MD, Tsotsos JK (2017) Vision-based fallen person detection for the elderly. In IEEE International Conference on Computer Vision Workshops, Venice, Italy, pp 1433-1442

Volkhardt M, Schneemann F, Gross H (2013) Fallen person detection for mobile robots using 3D depth data. In IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, pp 3573-3578

Wu YM, Chung JK, Hsiao NC et al (1999) Development of an integrated earthquake early warning system in Taiwan-case for the Hualien area earthquakes. Terrestrial Atmospheric and Oceanic Science J 10:719-736

Wu YM, Hsiao NC, Lee WHK et al (2007) State of the Art and progress in the earthquake early warning system in Taiwan. In: Gasparini P., Manfredi G., Zschau J. (eds) Earthquake Early Warning Systems. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-72241-0_14

Wu YM, Hsiao NC, Chin TL et al (2013) Earthquake early warning system in Taiwan. Encyclopedia of Earthquake Engineering. https://link.springer.com/content/pdf/10.1007%2F978-3-642-36197-5_99-1.pdf. Accessed 10 April 2021

Yao J, Lin C, Xie X et al (2010) Path planning for virtual human motion using improved A* star algorithm. In Seventh International Conference on Information Technology: New Generations, Las Vegas, NV, USA, pp 1154-1158

Figures

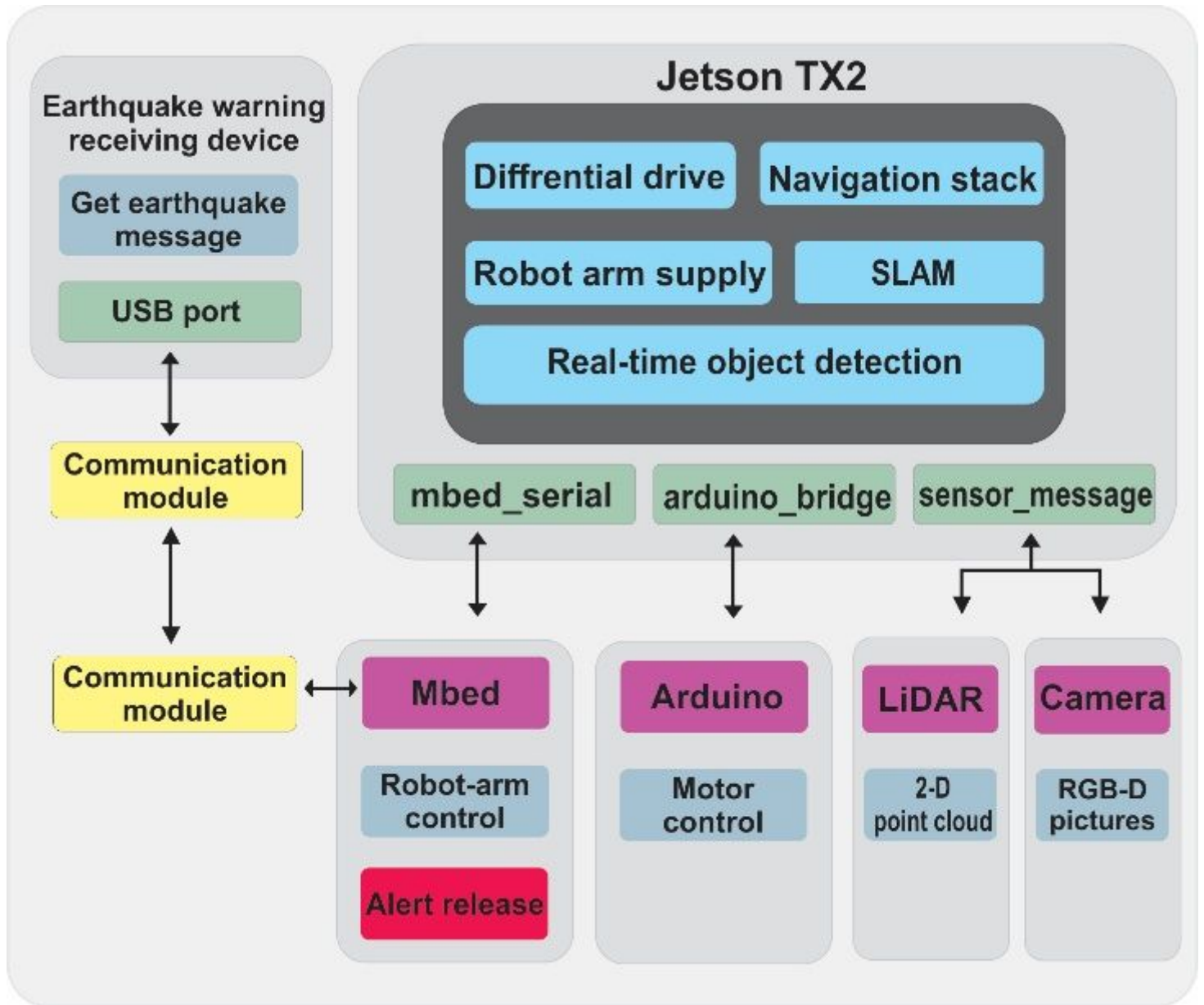


Figure 1

Architecture of the proposed robot's functions

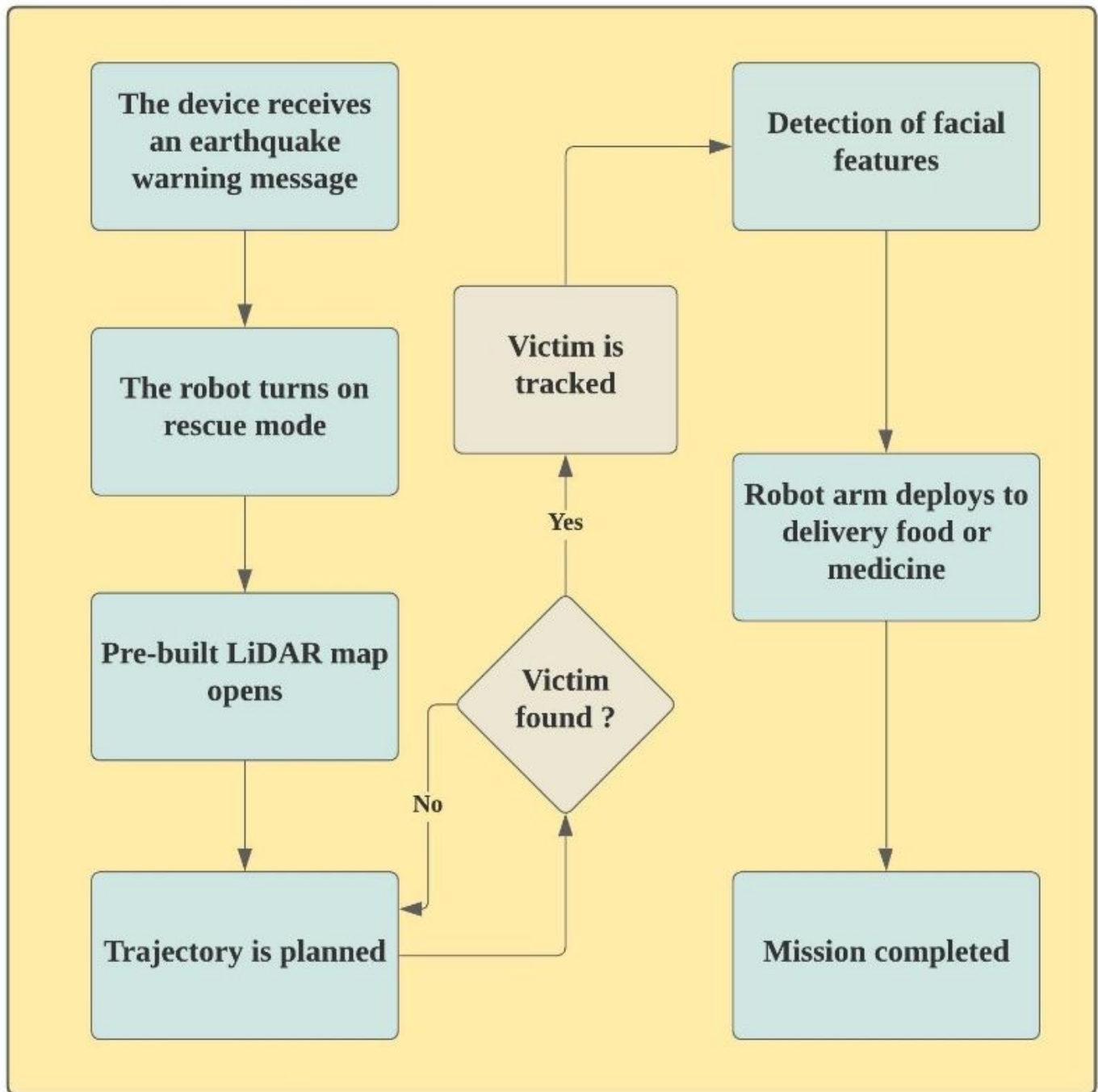


Figure 2

Operating flow of the proposed system

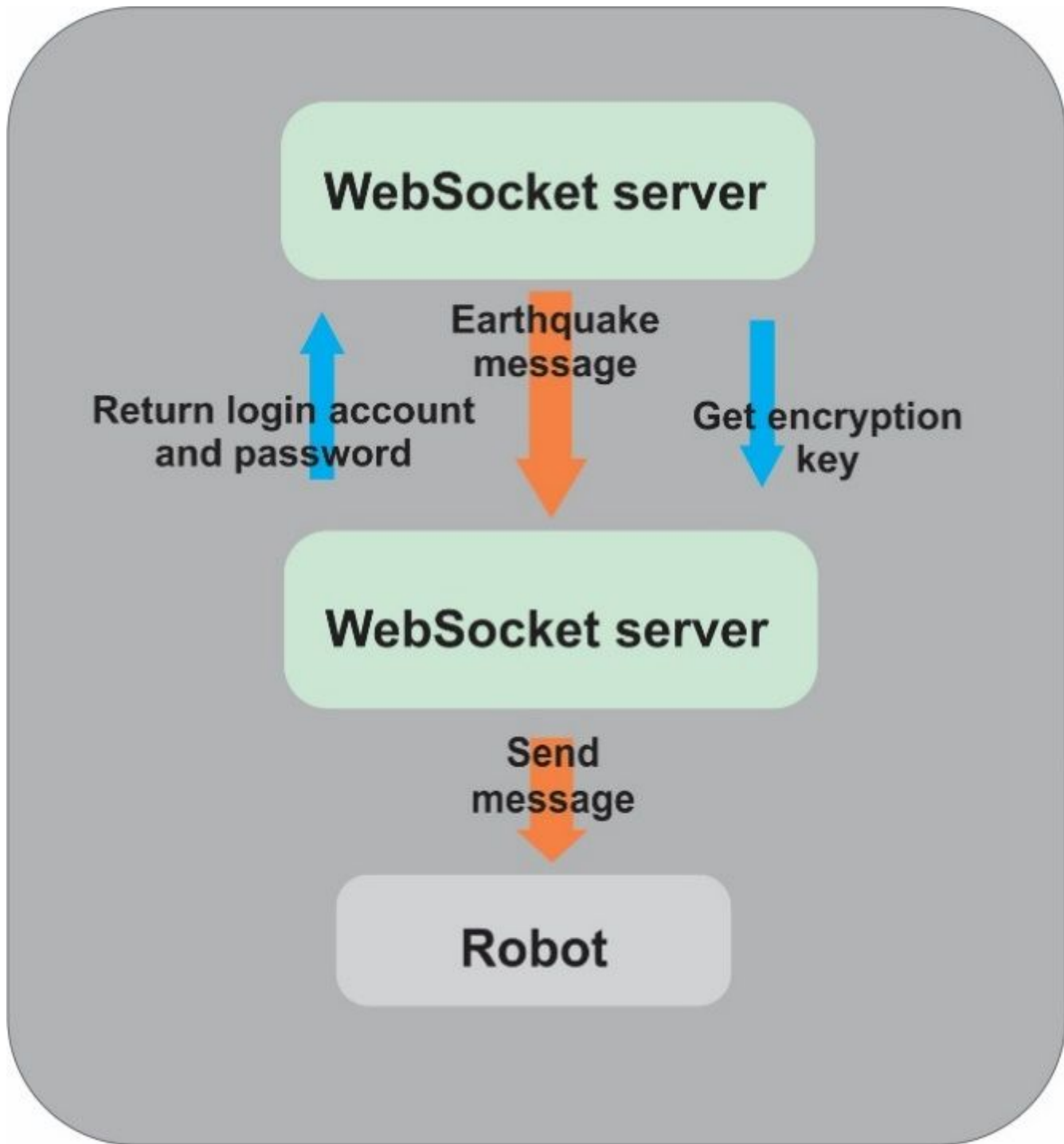


Figure 3

Architecture of the proposed system's earthquake interface

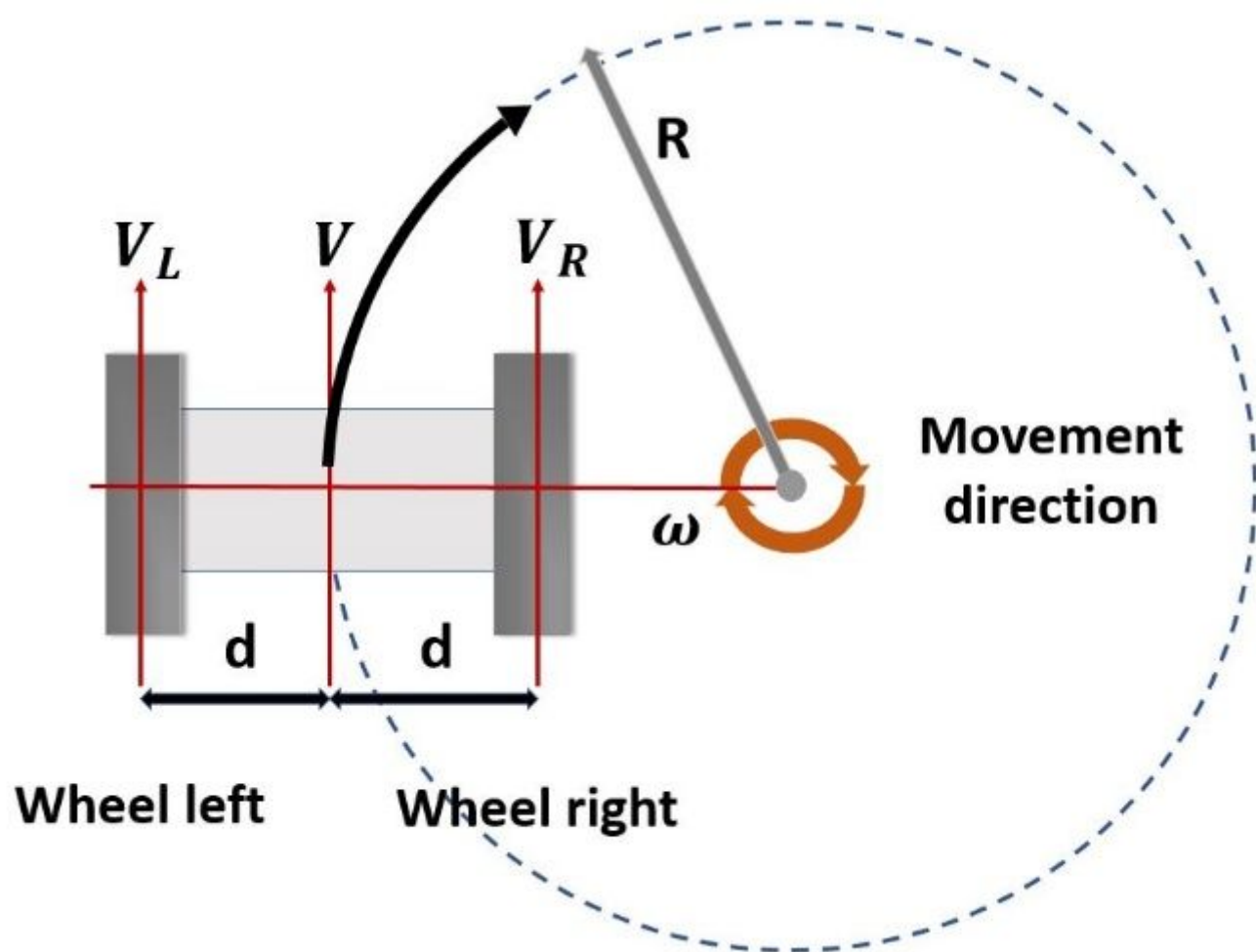


Figure 4

Motions of the differential-wheel chassis

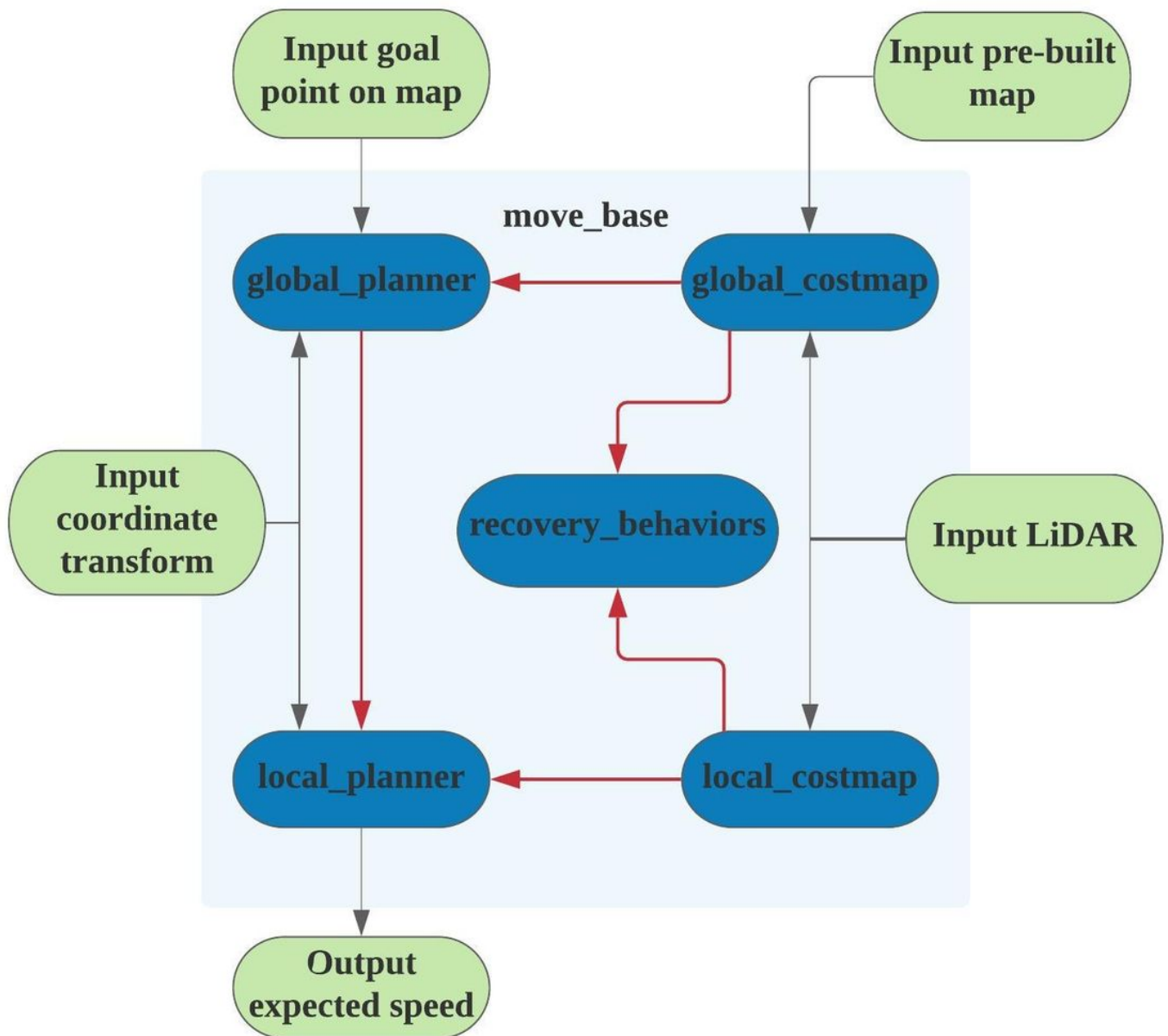


Figure 5

System framework of move base

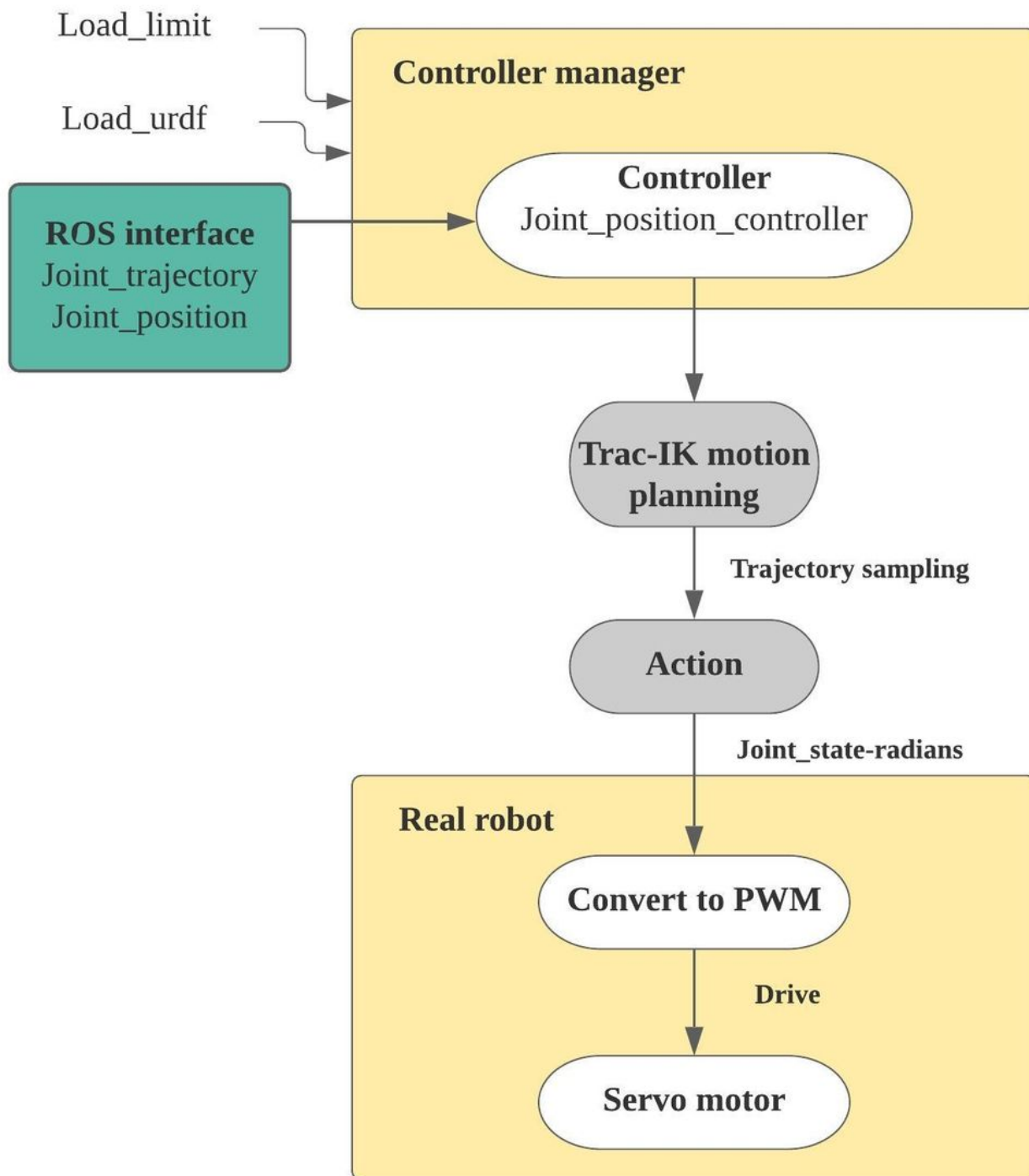


Figure 6

Control flow for the robotic arm

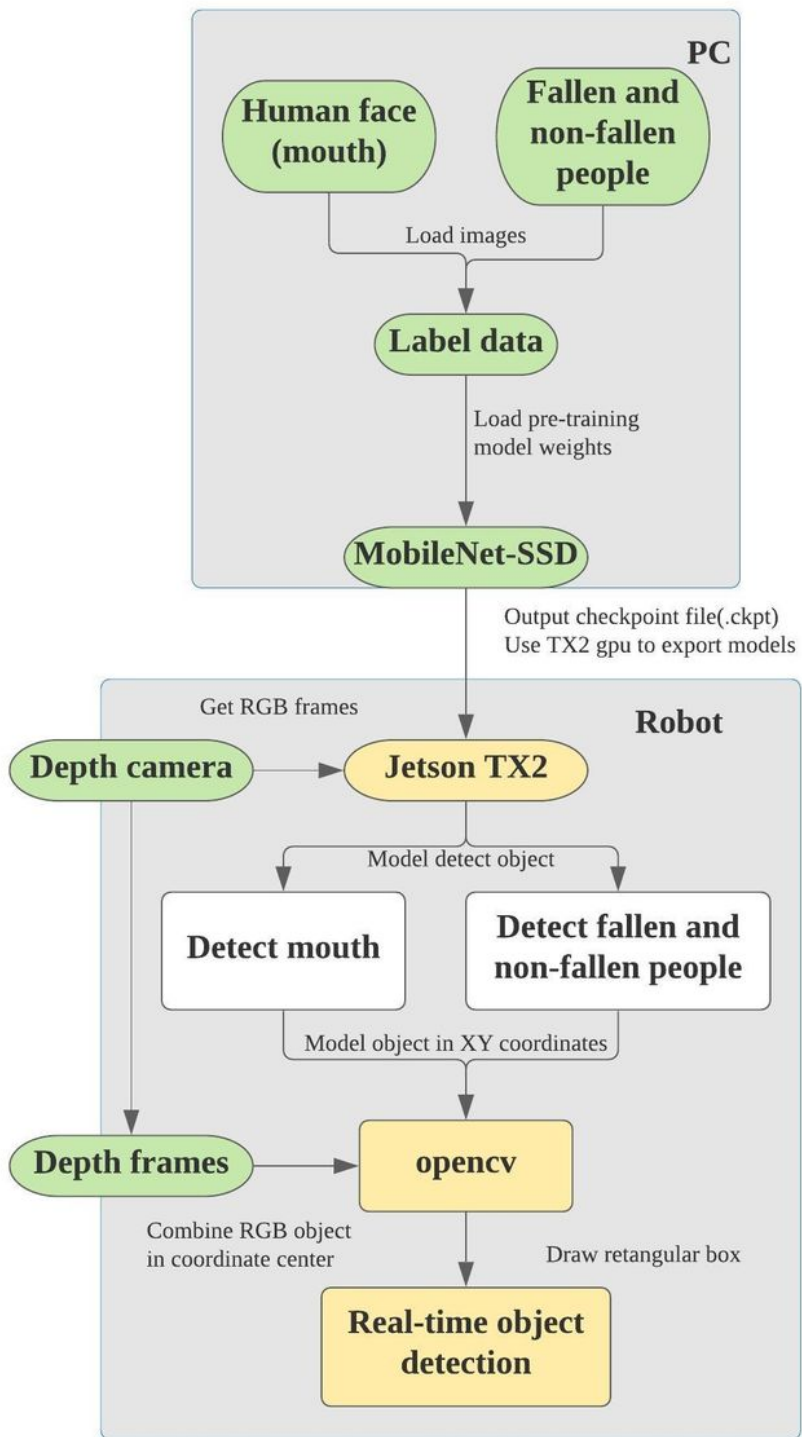


Figure 7

Flow chart of real-time image identification

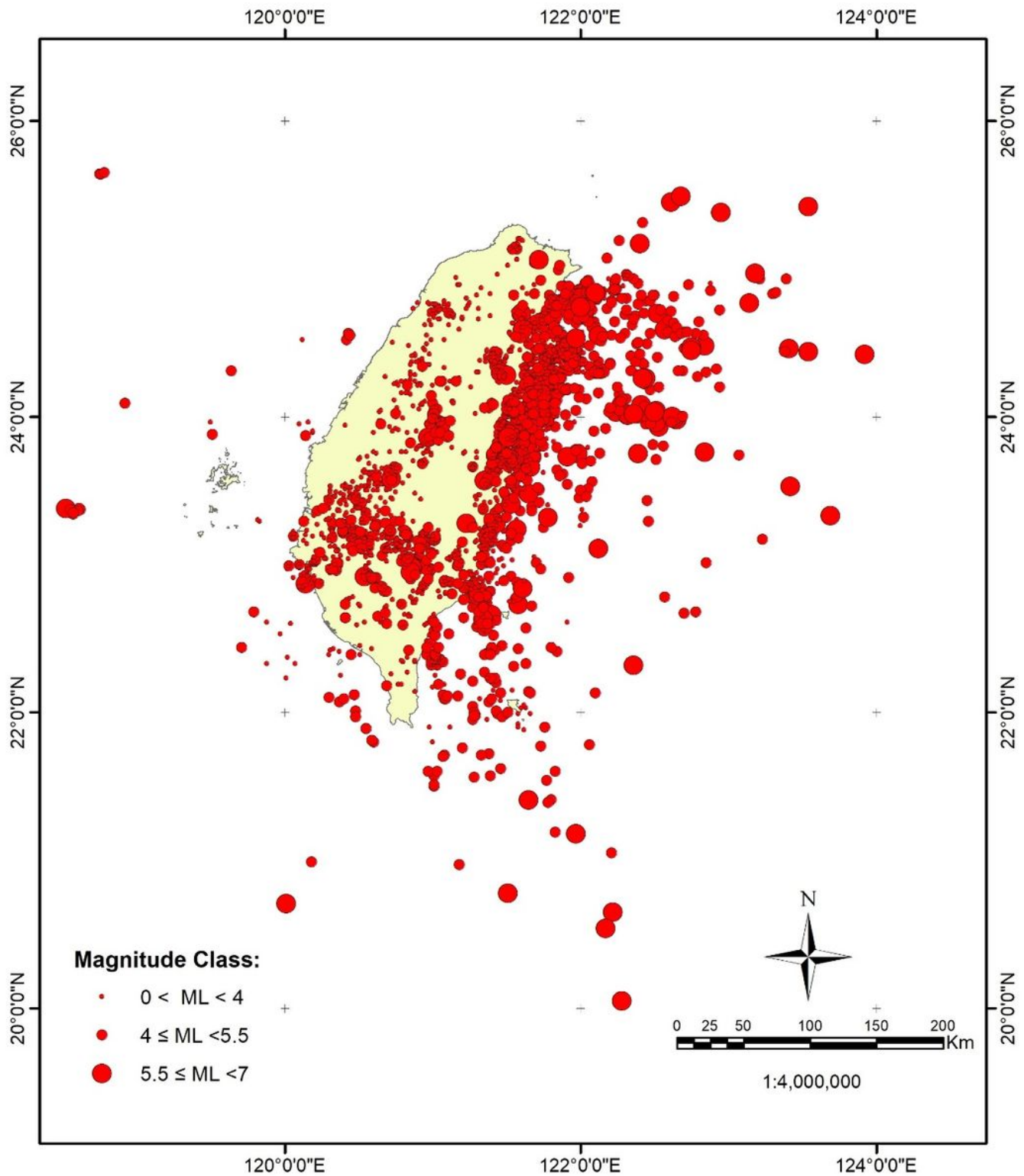


Figure 8

Epicentre distribution of the earthquakes during 2013 to 2020 around Taiwan

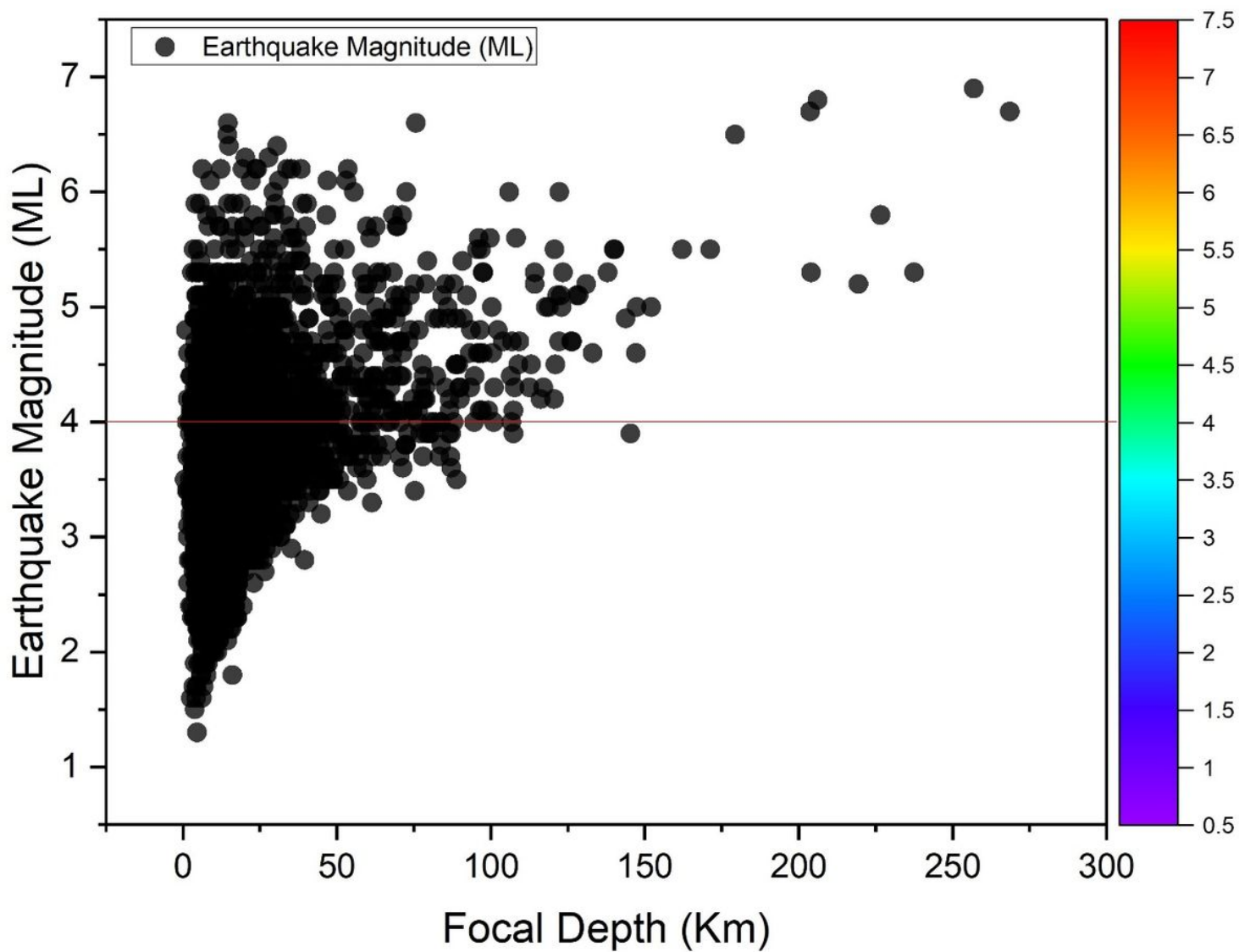


Figure 9

Focal depth distribution of earthquake magnitude during 2013 to 2020

Figure 11

Early earthquake warning report interface and robot's alarm device

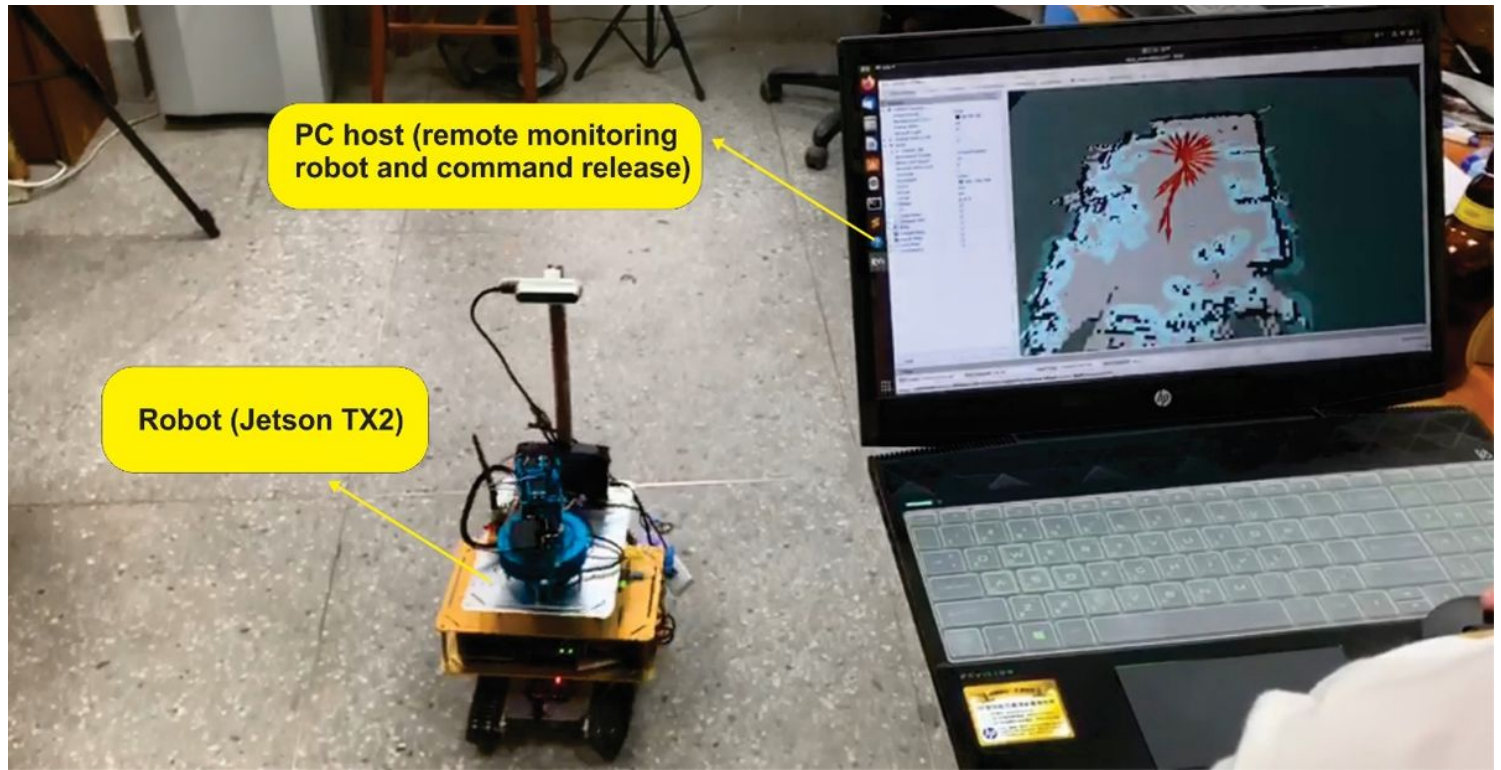


Figure 12

Remote keyboard control of the robot's mapping actions

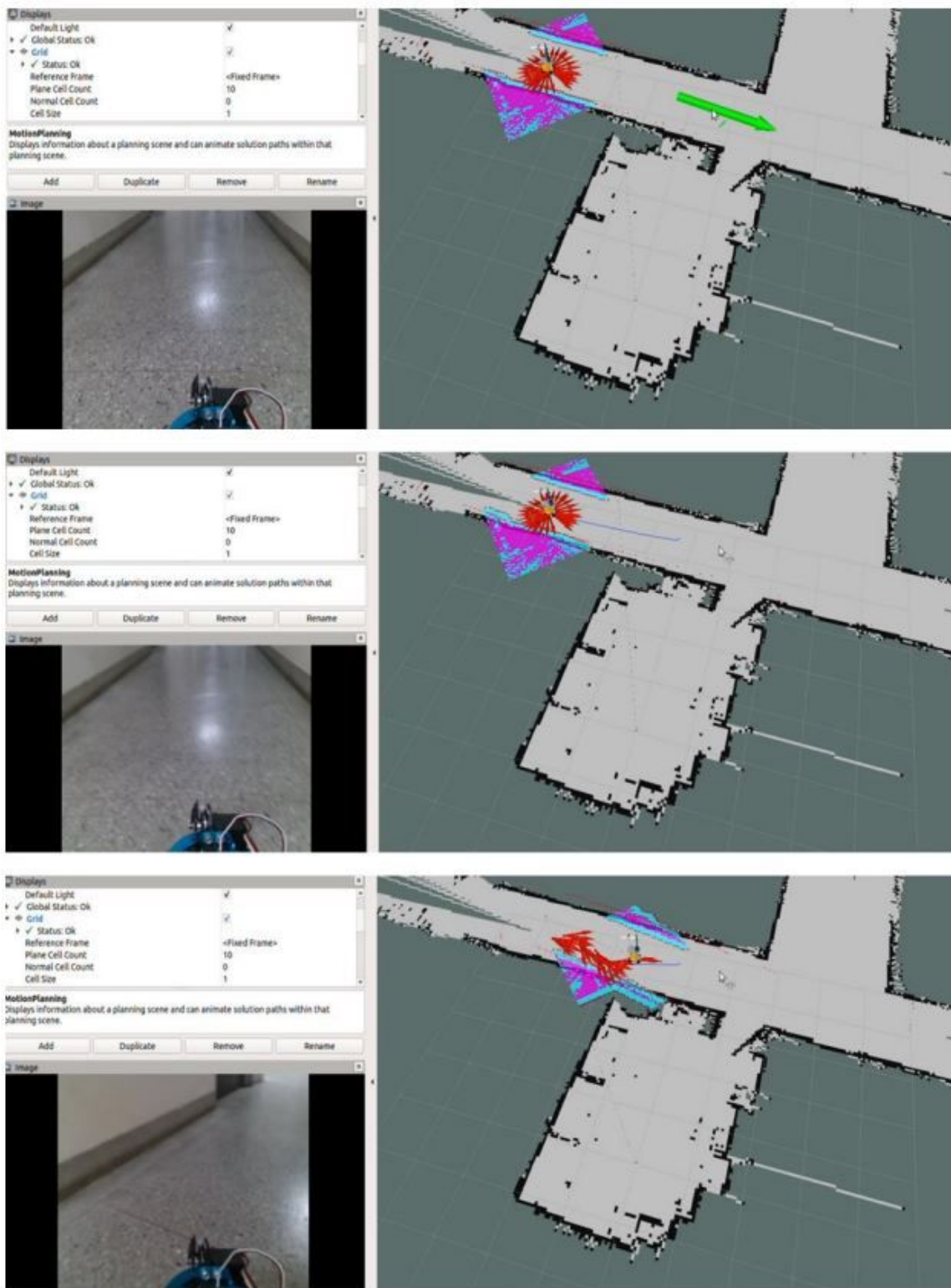


Figure 13

Remotely released coordinates approach to control of the robot's mapping actions

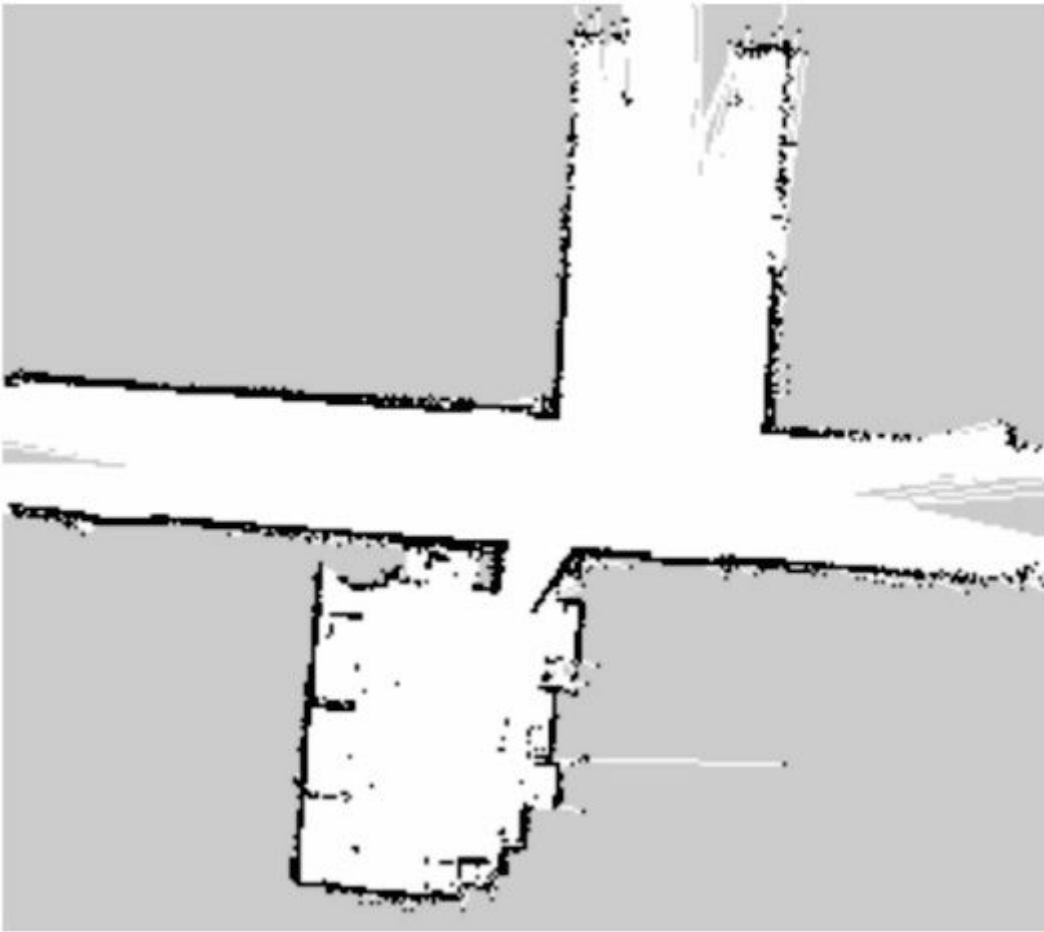


Figure 14

Complete mapping of the experiment site

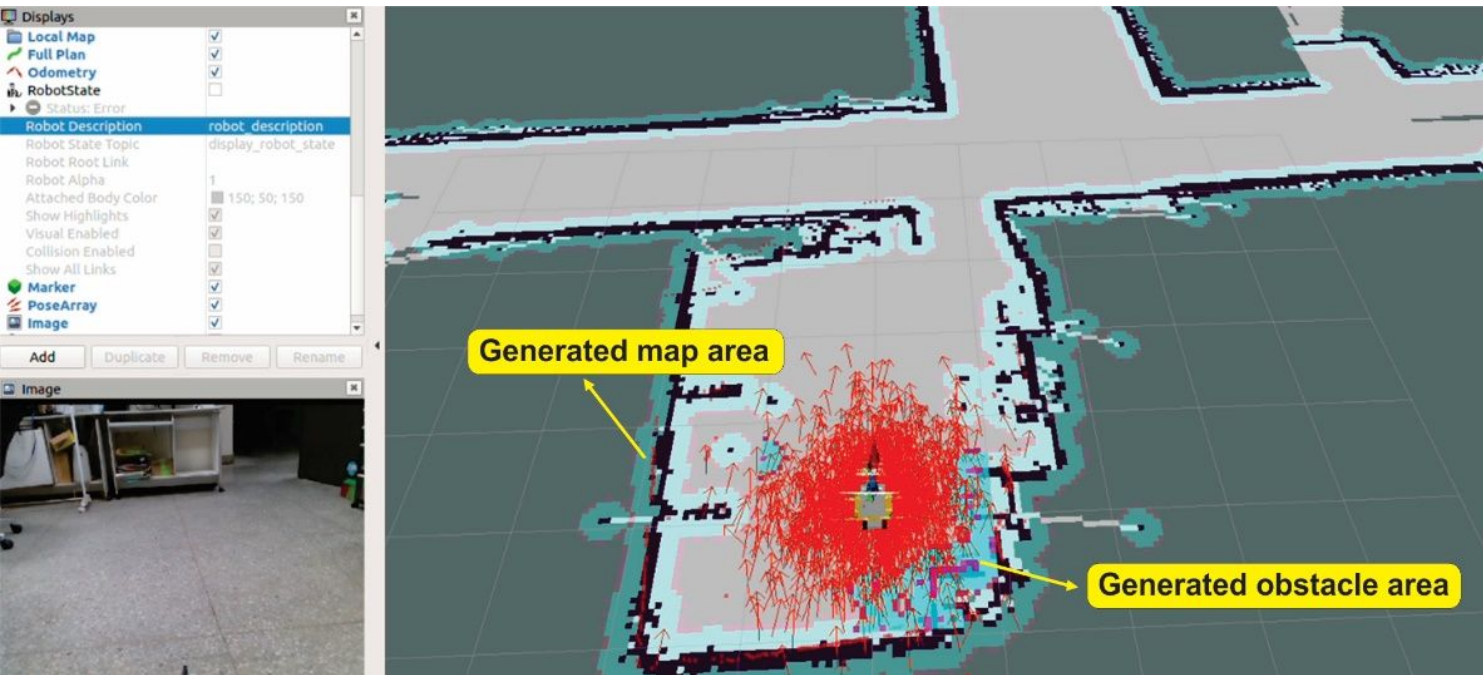


Figure 15

AMCL positioning

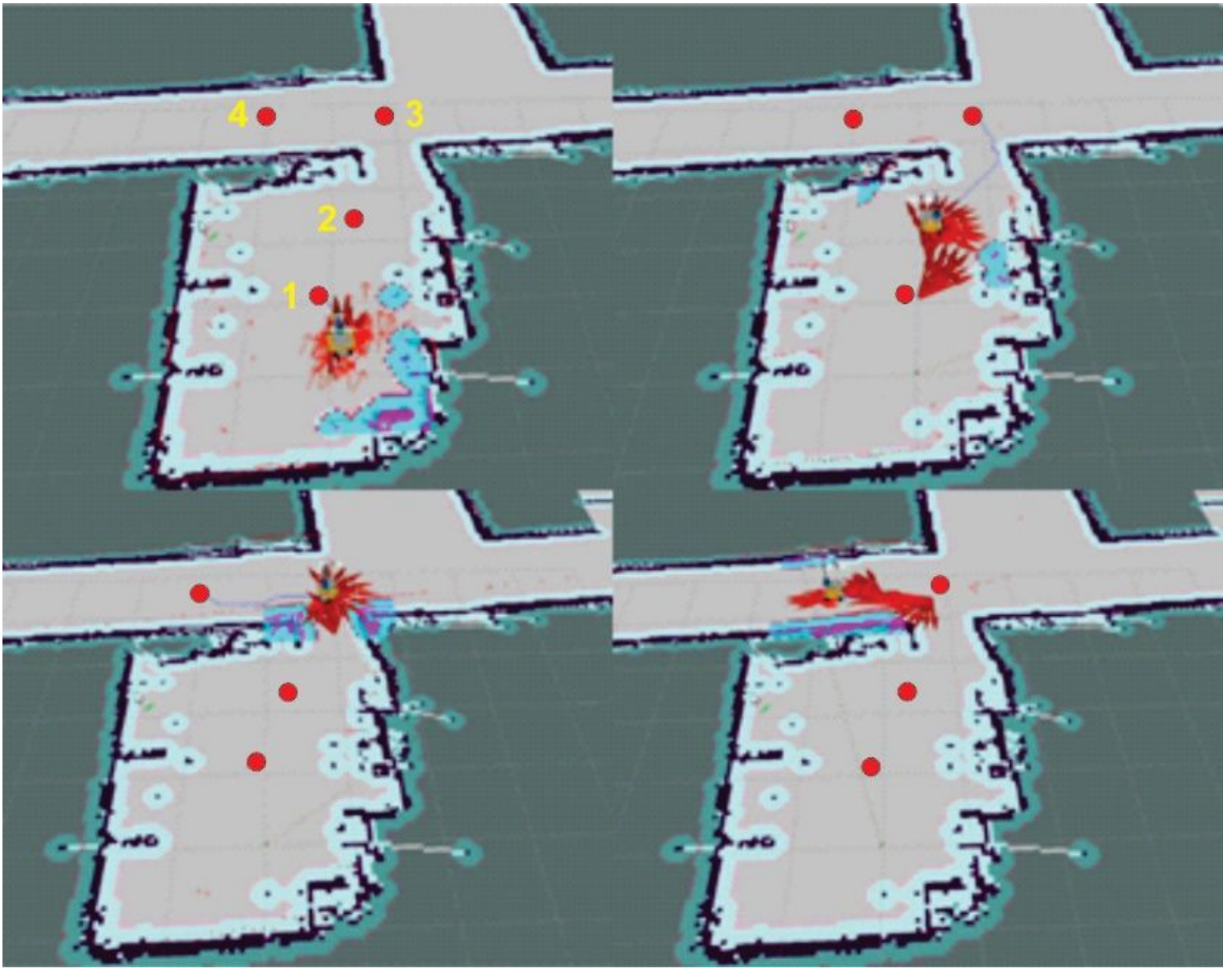


Figure 16

Robot proceeds to various spots



Figure 17

Identification results, fallen and non-fallen people

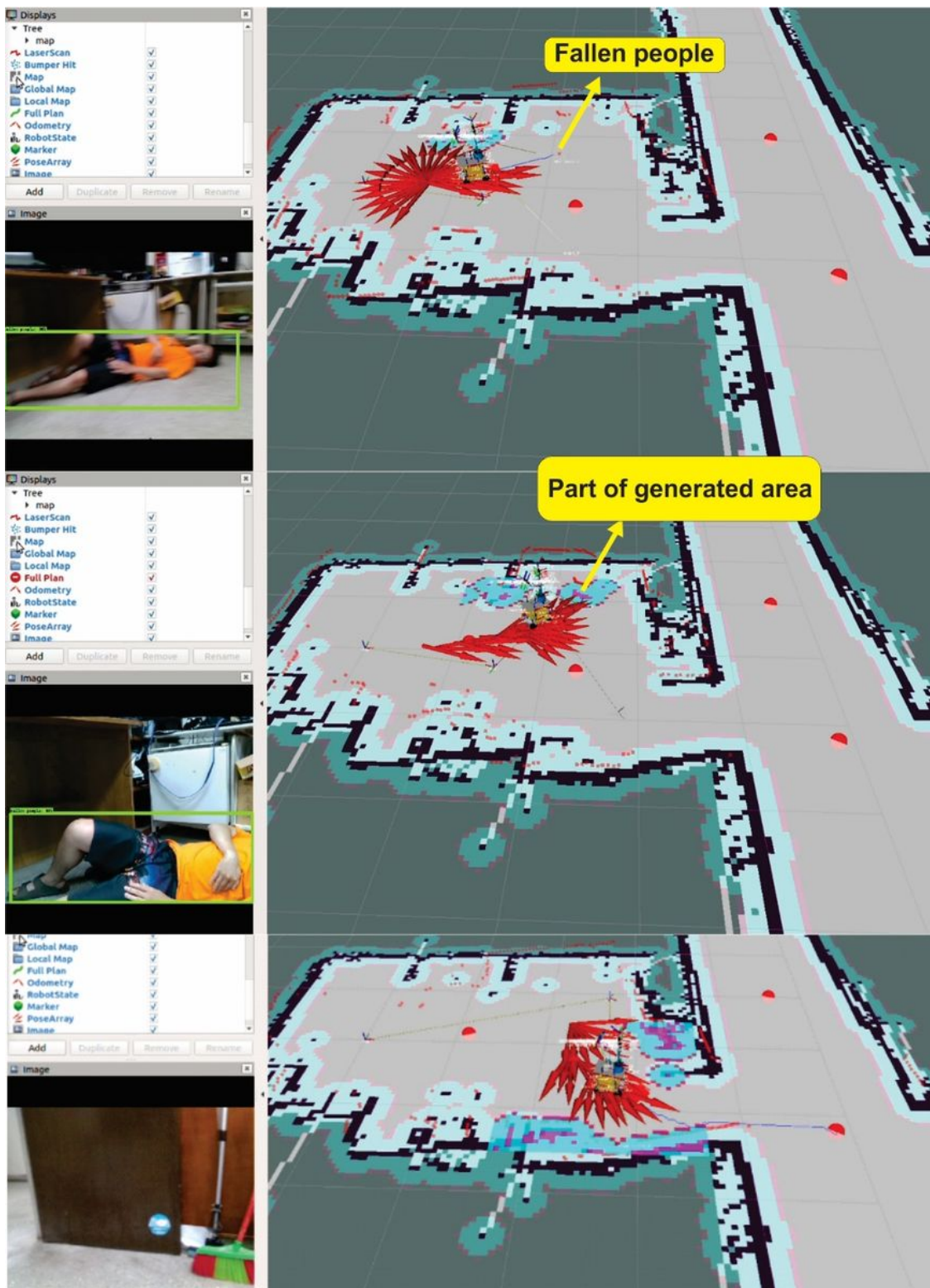


Figure 18

Path planning for reaching fallen people and proceeding to various points sequentially

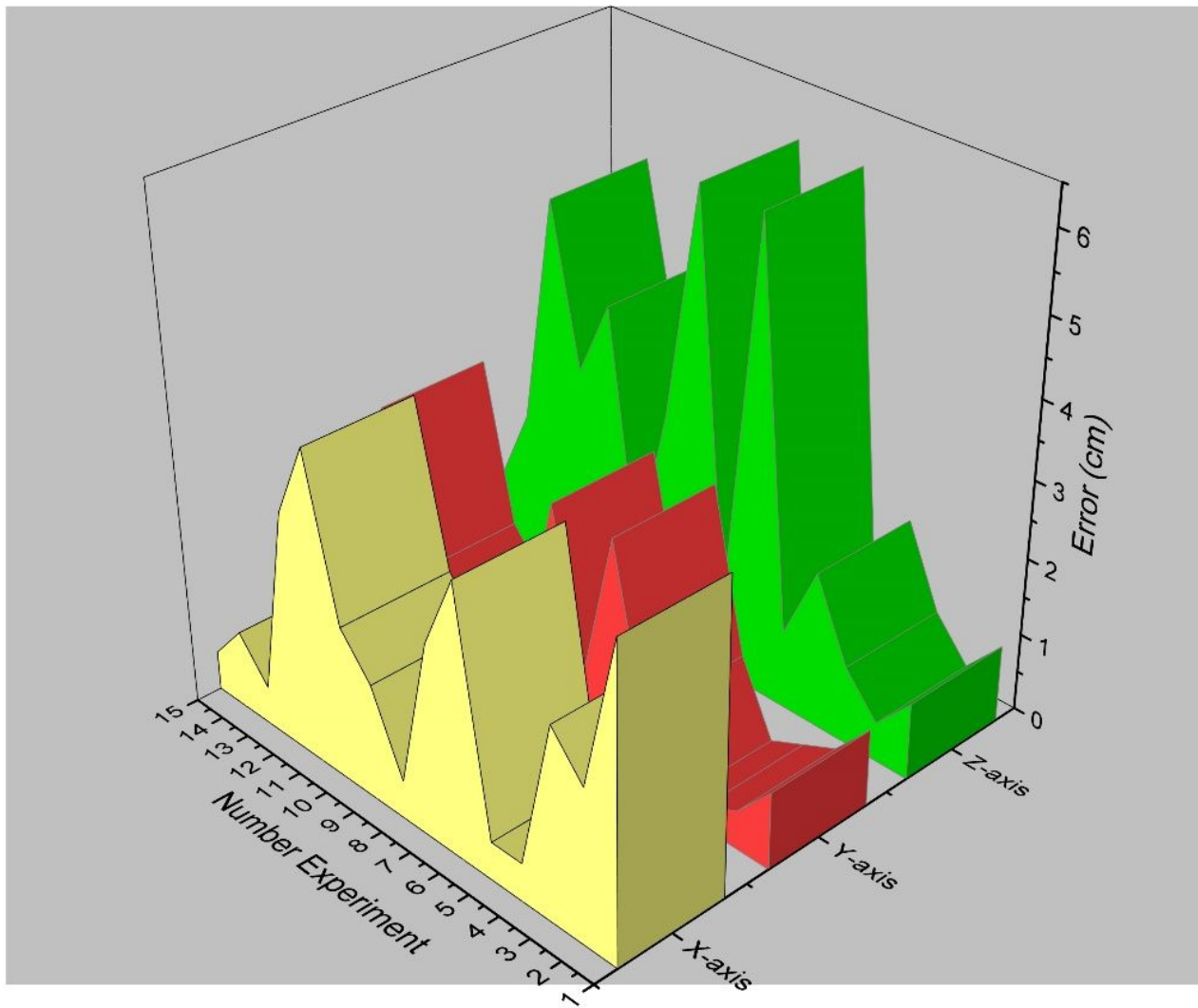


Figure 19

Absolute value of discrepancies between terminal point of robotic arm and central point of mouth

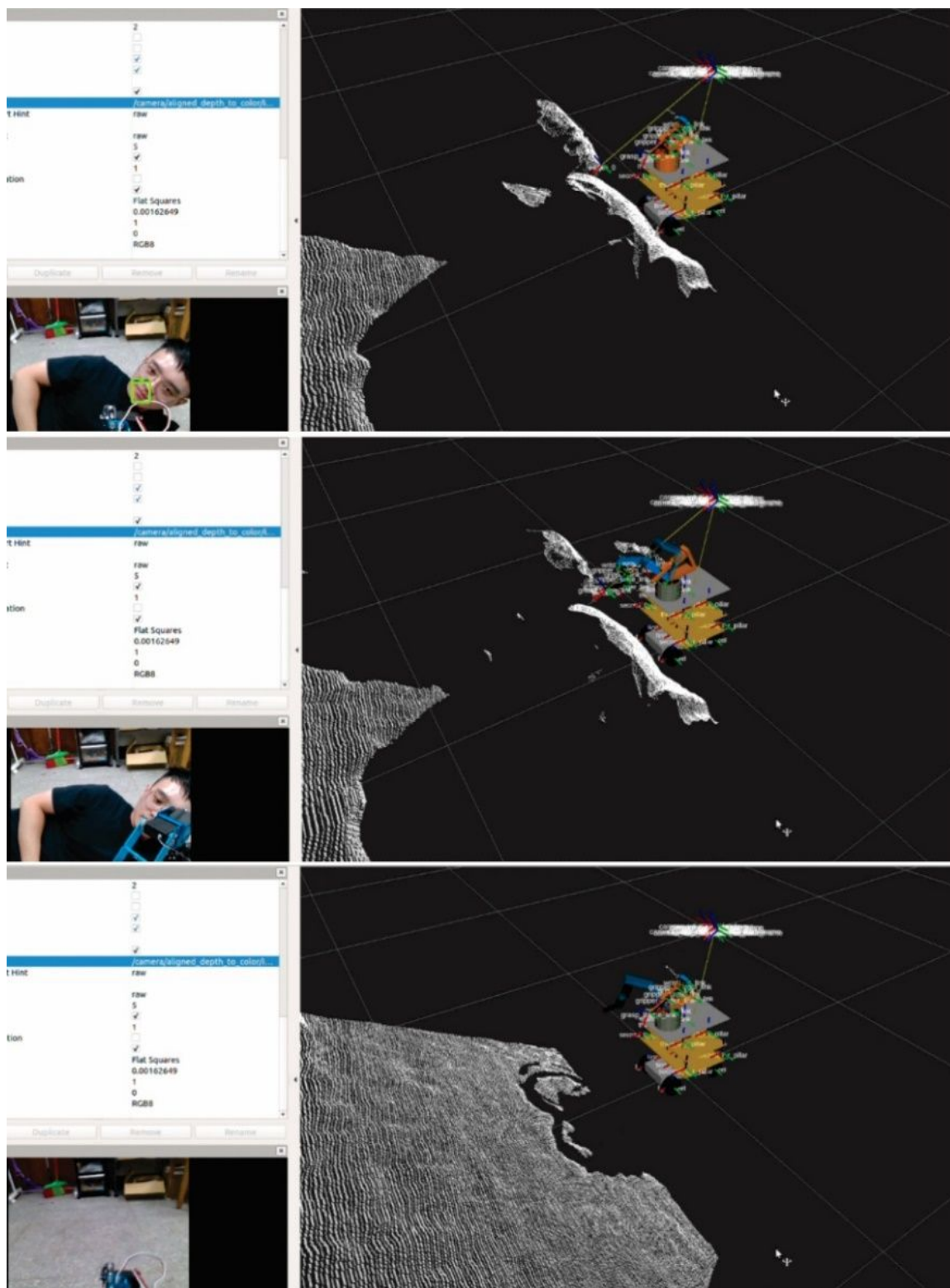


Figure 20

Robot returning to its original posture after completing a supply mission

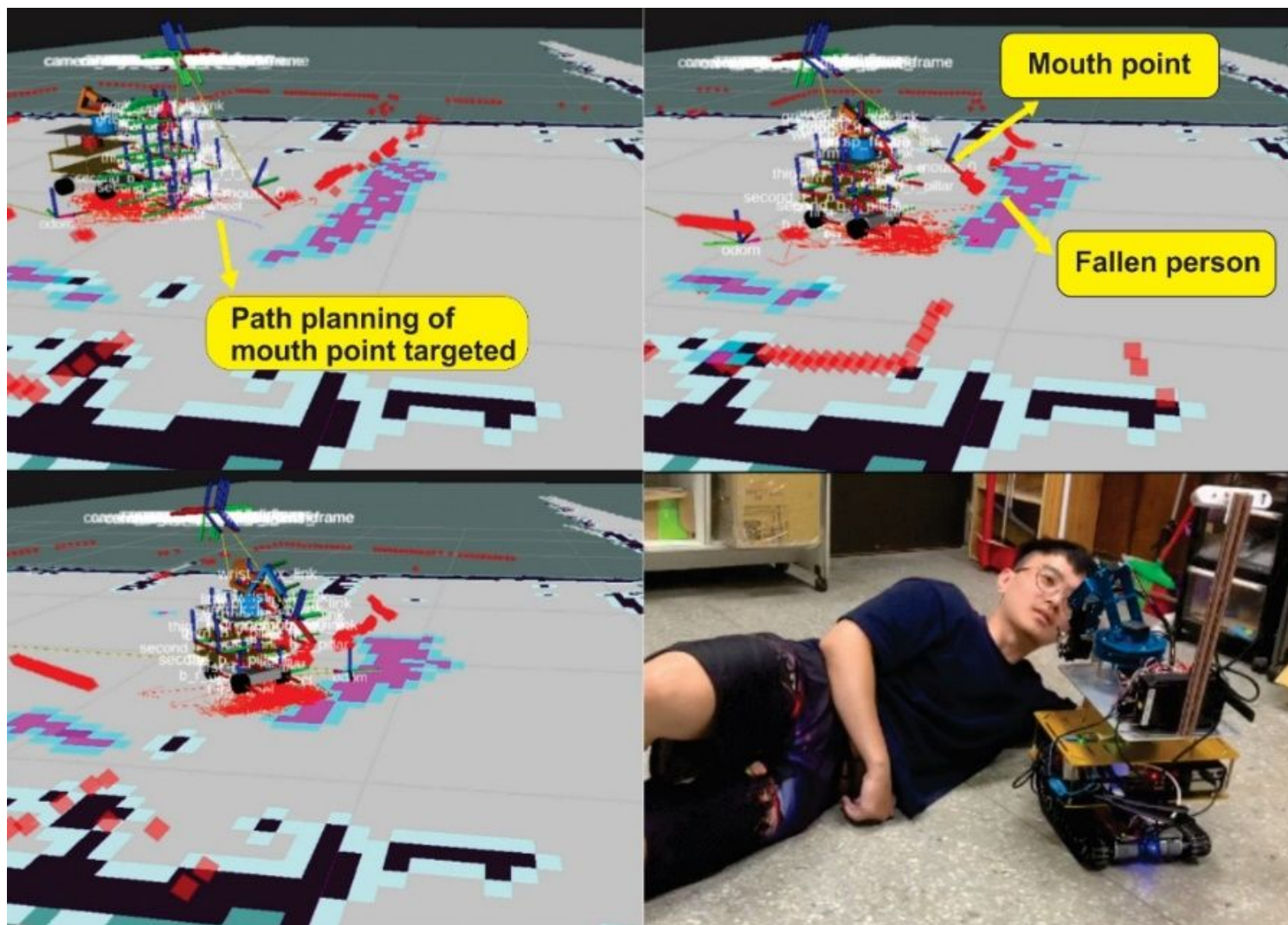


Figure 21

Test of the robot's automatic navigation and delivery of supplies