CAPÍTULO 10

A UGV NAVIGATION SYSTEM FOR LARGE OUTDOOR ENVIRONMENTS INCLUDING VIRTUAL OBSTACLES FOR NO-GO ZONES

M. GARZON, E. FOTIADIS, P. PUENTE, A. ALACID y A. BARRIENTOS

Centro de Automática y Robótica UPM-CSIC, Calle José Gutiérrez Abascal, 2. 28006 Madrid, Spain – ma.garzon@upm.es.

This work presents a navigation system for UGVs in large outdoor environments; virtual obstacles are added to the system in order to avoid zones that may present risks to the UGV or the elements in its surroundings. The platform, software architecture and the modifications necessary to handle the virtual obstacles are explained in detail. Several tests have been performed and their results show that the system proposed is capable of performing safe navigation in complex environments.

1 Introduction

The autonomous navigation for Unmanned Ground Robots (UGV) is a very useful tool for many robotic applications; one of the areas where it can be highly exploited is the autonomous surveillance of large infrastructures, in which this work is framed. The autonomous navigation can be defined as the problem of finding and following a collision-free motion from a start to a goal position. This paper presents a scheme to perform the autonomous navigation, and to improve the safety of the system by adding a series of virtual obstacles in order to avoid zones where a potential risk, which may not be detected by the sensors on board the UGV, has been identified.

One of the main concerns regarding autonomous navigation is the safety of both the UGV performing the navigation and the elements that may be in the surroundings. In order to avoid collisions, most systems rely on the information provided by the sensors onboard the UGV – mainly on laser rangefinders – but when the environments are more complex those sensors does not provide all the information necessary, making it necessary to use more complex sensors and therefore increasing the complexity of the system and still not assuring the detection of all the obstacles. For the proposed surveillance application, most of the obstacles that cannot be detected by the laser rangefinders, such as: sidewalks, fences, benches, negative obstacles or small bushes; are placed in fixed locations, and therefore can be known *a priori*.

This work proposes a very simple solution to that problem, and it is to create virtual obstacles for those elements denoted as No-Go Zones (NGZ), and add them to the map created by the SLAM algorithm. This modified map is sent to the path planning algorithm, and will allow the navigation system to avoid such obstacles; also some modifications to the basic navigation architecture are made to prevent misbehaviors of any modules of the navigation system.

The outline of this paper is as follows: after this brief introduction, Section 2 reviews some related works. Section 3 describes the hardware components used. The software architecture for the navigation is described in Section 4. In Section 5, the Virtual Obstacles Approach and the changes to adapt the architecture to this approach are detailed. Section 6 describes the experiments that have been carried on and their respective results. Finally, Section 7 presents the conclusions and future lines of work.

2 Related Work

The Autonomous Navigation Robot world is very large, so in review of the related work, the content will be enclosed to the navigation of Unmanned Ground Vehicles (UGVs) dedicated to outdoor surveillance tasks. This is due to the huge amount of possibilities of surveillance, whether indoor or outdoor, that are carried through. For example, from using fixed cameras which help the UGVs in its task to Unmanned Aerial Vehicles (UAVs) that work cooperatively.

The main objective of any navigation system is to move safely from a start position to a goal position, both of them determined previously. That means avoiding obstacles and making the path in the lowest possible time. For example, in (Kunwar, et al., 2006), they develop a moving target interceptor in environments where obstacles may move or not. The navigation concept in here is determined by the position of the target, in which velocity and position are taken in order to intercept the target with an algorithm called rendezvous-guidance (RG). They just base their algorithm in a vision module. Other researchers have been including fuzzy logic in the navigation algorithm through the time. These method can be seen in (Zhu, et al., 2007), where they develop a fuzzy logic general navigation system which produces smooth trajectories. Here, the dead cycle problem is solved and static and moving obstacles are also avoided.

Quite a lot of sensors configuration can be arranged to perform autonomous navigation. Despite the most utilization of lasers combined with other sensors on these platforms, in another configuration is used. They develop a mobile robot whose navigation is based on GPS and Sonars. They introduce a radial basis function network (RBFN) used to locate the GPS into the robot coordinates, giving a smooth turning of the platform when avoiding obstacles. Other work bases its navigation just in vision module, as (Kunwar, et al., 2006) or (De Cubber, et al., 2009). This last one focuses its purpose on determining the terrain traversability with stereo images.

Returning to the most used method, the Laser Range Finders (LRF), in (Aliakbarpur, et al., 2009), they suggest a navigation system based on a 3D laser, a stereo camera and an Inertial Measurement Unit (IMU). In its work they develop an algorithm to calibrate these devices with the objective of being useful for surveillance and show its utility. Another LRF and camera based system is the one presented in (Kumar, et al., 2009). Here, besides the navigation system, human faces can also be detected, but is not optimal in outdoors. Other platform used in outdoor surveillance is (Yang, et al., 2010). In here, fast obstacle detection is developed based on two color cameras and helped with a 1D LRF. Regarding the navigation system, it is based on an IMU and a GPS. The innovation here is the method used to get the main ground disparity (MGD) from the V-disparity images in order to adapt the platform to intricate terrains.

3 Hardware Components Description

This section describes the hardware components used for this work. It includes a base platform, an on-board computer and series of on-board sensors that, when used together, allow the system to perform safe navigation.

3.1. Base Platform

The mobile platform is based on the Summit XL. It has skid-steering kinematics based on four high efficiency motors. The odometry is provided by an encoder on each wheel and a high precision angular sensor assembled inside the chassis. The robot can move autonomously or it can be teleoperated using images from a camera on-board. The Summit XL uses an open and modular control architecture based on ROS. An image of the robot before all the equipment was mounted is shown in *Fig. 7*

Fig. 7. Summit XL Mobile Platform

3.2. On-Board Components

On-Board Computer:

The robot has an embedded Linux PC. It is an Intel DN2800MT with mini ITX form factor located in the middle of the robot, The computer is completed with 4GB of RAM and a 2.5" SATA HDD. This computer allows having all the data processing and navigation algorithms in a fully autonomous manner.

Laser Rangefinders:

Two Hokuyo laser rangefinders are mounted on the platform: the first one is an UTM-30LX-EW, it can scan a 270º semicircular field, with a guaranteed range that goes from 0.1 to 30 meters and a maximum output frequency of 40Hz. The second laser rangefinder is an URG-04LX, it has a semicircular scanning area of 240º; the guaranteed range for this device goes from 0.06 to 4.09 meters and it can operate at a maximum frequency of 10 Hz.

Both devices are shown in *Fig. 8*. They are placed at different heights, with the 4 meters rangefinder at 10 centimeters from the floor in the front of the robot, and the 30 meters laser at 60 centimeters over the floor and in the center part of the robot.

UTM-30LX-EW

URG-04LX

Fig. 8. Laser Rangefinders Mounted On-Board

Global Position System:

A Novatel OEM-4 GPS engine is used; it can offer centimeter level positioning accuracy with a frequency of 2Hz. RS232 serial communication is used to read the incoming data and send correction commands. The engine is complemented with an ANT-A72GOLA-TW GPS antenna.

Video Cameras.

There are also two cameras on-board the robot: the first one is a Logitech Sphere Camera, with motorized tracking (189° horizontal and 102° vertical) and Autofocus lens system. It provides a frame rate of up to 30 fps and a resolution of 1600 by 1200 pixels (HD quality); the second camera is a Firefly MV FMVU-03MTM, it has a frame rate of 60 fps and a resolution of 752 x 480 pixels.

The sphere camera is used mainly for teleoperation, due to its Pan-Tilt capability, while the Firefly camera is used for high level video processing because it is faster and has better optics.

Inertial Measurement Unit.

A MicroStrain 3DM-GX3 25 which is a high-performance, miniature Attitude Heading Reference System (AHRS) is mounted inside the robot. It combines accelerometers, gyroscopes and magnetometers in the three axes, with temperature sensors and an embedded processor to provide static and dynamic orientation, as well as inertial measurements.

Fig. 9. Platform with all sensors and components.

The *Fig. 9* shows a picture of the complete system, the on-board computer and IMU are inside the chassis. The 30m rangefinder and one camera are mounted on a pedestal to have a better detection of the external objects, and the 4m laser rangefinder is below the robot and allows avoiding lower obstacles.

4 Software Architecture

This section describes the software architecture used for this work; it controls all the components of the UGV as well as the high level algorithms for mapping and navigation, including the proposed schema to avoid NGZ. The entirely software is supported by the software framework ROS (Robot Operating System), which comprises libraries and tools that facilitate the development of new robot applications.

First the basic components necessary to control de the robot will be mentioned. After that, the main algorithms used for the navigation will be described.

4.1 Basic Components

4.1.1 tf

To set the sensor locations and maintain the relationship between coordinate frames in a tree structure, *tf* (transform functions) package is executed. It allows keeping track of multiple coordinate frames and their corresponding transformations over time. A clear example is the sensor locations with respect to the robot body or the global position of the UGV in the reference frame. It is a very important component, because a simple deviation of a laser measurement could result in the robot hitting an obstacle.

4.1.1 summit_xl_ctrl

The *summit_xl_ctrl* package performs the low level control of the robot. This means that it receives the linear and angular velocity commands from either the teleoperation or the autonomous navigation modules. Those commands are processed using the inverse kinematics of the UGV in order to obtain the required speed for each wheel. The resulting control signals are sent directly to the control of each wheel-motor in order to obtain the desired speed. This package also receives the readings from the encoders on each wheel and from the angular sensor (gyroscope). These data are used to calculate the odometry which then is sent back to the other nodes in the system.

4.1.2 summit_xl_pad

The *summit_xl_pad* node is used to teleoperate the UGV. It uses data incoming from any joystick or control pad, and converts those data to velocity commands that are sent directly to the *summit_xl_ctrl* node, thus performing a direct control over the movements of the UGV.

4.1.3 EKF pose

The *EKF (Extended Kalman Filter) pose* package is used to estimate the three-dimensional pose (position and orientation) of the robot. Once the robot odometry, GPS and IMU readings are received and their corresponding transformations are set. It is possible to establish a measurement model that links the measurements of each sensor with the global position of the mobile robot. Those models are first combined using probability fusion techniques and then expressed as another probability density function. A detailed explanation of this process can be found in (Garzón, et al., 2013).

The models and the Kalman Filter itself are implemented using a ROS wrapper of the Bayesian Filter Library. The combined pose is sent back to the system as a ROS topic and also by setting the corresponding transformations in the *tf* topic. The estimated pose can be obtained with the data available at each that time, even if one sensor stops sending information. Also, if the information is received after a time-out, it is disregarded.

4.2 Main Navigation Algorithms

The main navigation algorithms are a series of modules that, when combined, provide the UGV with the ability of performing autonomous navigation. Here each one of the modules will be briefly described: the first module is used for creating maps using readings from the laser and the odometer (SLAM), another module is used for handling and publishing the maps, a third one is in charge of the localization within a given map, and finally, the main component performs path planning and following.

4.2.1 SLAM (gmapping)

The navigation scheme proposed is based on navigating on a known map, thus building this map is a main part of the overall process. In this work a ROS wrapper for *GMapping* is used. *GMapping* takes raw laser range data, in this case the combined odometry from the EKF, and uses it to build grid maps using a highly efficient Rao-Blackwellized particle filer (Grisetti, et al., 2007).

4.2.2 Map Server

The map server provides the map data to all other elements on the system. The maps are based on grayscale images, where each pixel represents the occupancy state of each cell of the world; white pixels represent free space, black ones represent occupied cells and gray ones represent unknown spaces. The values are loaded from the images and published in one or more occupancy grids.

4.2.3 Localization (AMCL)

This package implements the Adaptive Monte-Carlo Localization (AMCL). It works as a probabilistic localization system, using a particle filter to track the pose of the UGV in a known map. It relays on the information provided by the scans of laser rangefinder and the *tf* messages. The AMCL publishes the estimated pose of the robot in the map with its corresponding covariance, and the set of pose estimates being maintained by the filter.

4.2.4 Path Planning and Following (Move Base)

This package includes some major components for the two-dimensional navigation. It takes a goal in the world, and attempts to reach it with a mobile base. The planning is separated in two phases: global and local; each one of them uses its own cost map.

The global planner finds a minimum cost plan, from a start to an end point, using the Dijkstra's algorithm in a given global cost map. It works under the assumption that the UGV is a circular robot and provides only the list of points necessary to reach the goal, it does not provides orientations, times nor velocity commands.

The local planner provides a controller that helps the robot to follow the global path created by the global planner. It uses a smaller local occupancy map that is constantly redefined around the current pose of the robot. The local planner creates a kinematic trajectory for the robot to get from a start to a local goal pose, providing the linear and angular velocities that will allow the platform to safely traverse the local occupancy grid. The local goal pose is selected as the farthest point of the global path that is covered by the local map. It takes into account static and dynamic obstacles that may be added by the sensors on-board the robot. The velocity commands are sent directly to the robot's controller.

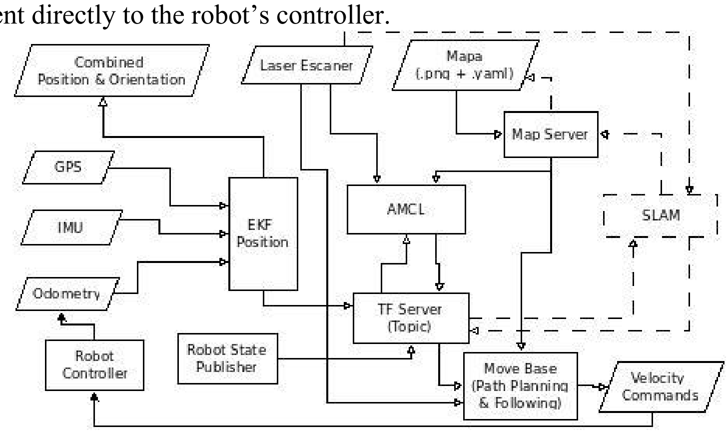


Fig. 10. Software Architecture.

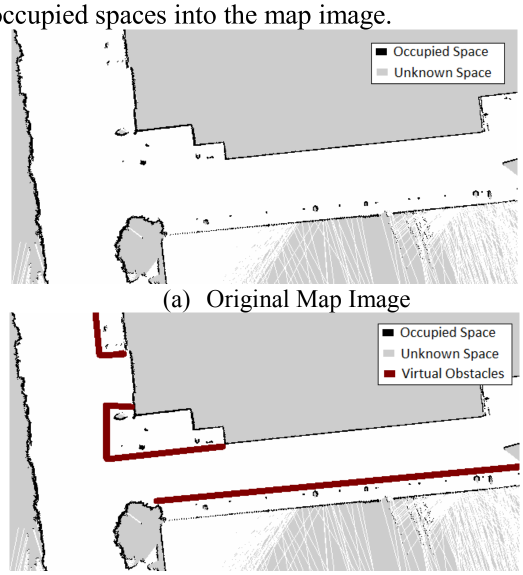
The Fig. 10 shows the complete software architecture, all important modules, inputs and outputs, as well as their respective connections are shown; the dashed lines represent the modules and connections used only for the SLAM algorithm.

5 Virtual Obstacles Approach

This section describes the changes and additions made to the software architecture described in Section 4, to enhance it with the ability to avoid NGZ. The basic idea is to modify the occupancy map, in order to mark the forbidden zones as if they had obstacles in them. Making changes in the map has two main drawbacks the performance of the system that need to be solved: first the localization system will not work correctly since it will no longer be able to match correctly the readings of the laser rangefinder against the given map; the second consequence is that since the laser rangefinder will not find any obstacles, it will clear the NGZ even if they were previously marked as occupied. In the next subsections, the methodology to add the obstacles will be explained, also the solutions the localization and clearing zones problem will be detailed.

5.1 Changing the Map to Add No-Go Zones

As was described in Section 4.2.2, the map is represented as an image where each pixel represents a cell and its occupancy is given by the color of the pixel. Since the location and shape of the NGZ is assumed to be known *a priori*, virtual obstacles, which represent those zones, can be added *offline* as occupied spaces into the map image.



(a) Original Map Image

(b) Inclusion of Virtual Obstacles

Fig. 11 Virtual Obstacle Addition.

The Fig. 11 shows a portion of the map before and after the virtual obstacles were added. In this example, the virtual obstacles are used to represent zones where there is a sidewalk with a height that cannot be detected by the laser rangefinder.

5.2 Solving the Issues of the Modified Map

As was mentioned before, one of the main issues of modifying the occupancy grid is that the laser scanner will no longer match the information contained in the map. This is the most problematic issue, because it affects the localization algorithms which are basic for the navigation. To solve this problem two separated map server instances were created: the first instance uses the map without any modification, thus allowing the localization algorithm to have a correct representation of the world around the UGV. The second instance uses the map with the virtual obstacles and is sent to the path planning and following module. This module reads the map and performs the planning avoiding the NGZ. In the Fig. 4 it can be seen that there is only one instance of the map server, that loads the map and sends it to both the localization and the path planning and following modules. This will be replaced by two separated modules, the first including the virtual obstacles and only communicated with the move base module; a second one without modifications that will be connected only to the AMCL module.

The second issue is that the laser scanner may clear the location where the cost map has been modified. This may cause that the planning algorithm to generate paths that may traverse the NGZ, to solve this problem a service provided by the cost map handler is used, this service reloads the original cost map from the stored image, so before the global planning is performed, the cost map is reloaded from the image. This gives as result on a path that will always avoid the NGZ.

6 Experiments and Results

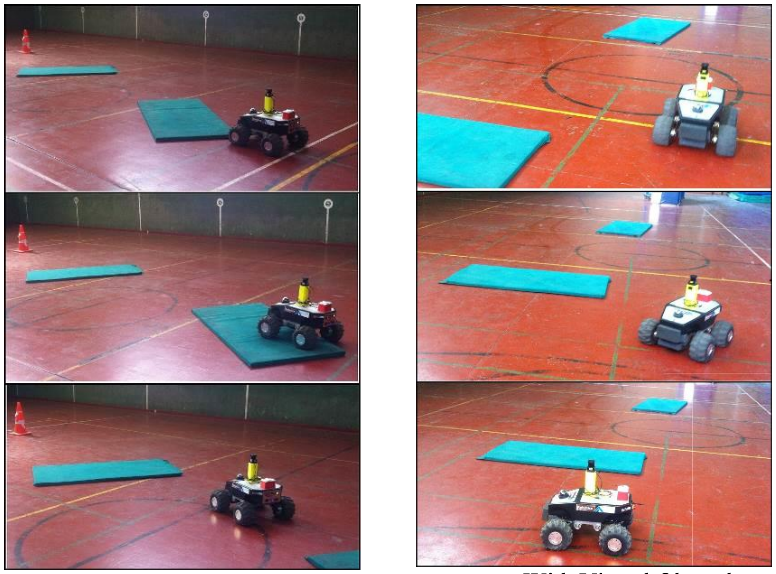
This section describes the scenarios and experiments that were carried on in order to evaluate the performance of the proposed technique. The Hokuyo UTM-30-LX-EWTM 30m laser rangefinder described in Section 3.2 was used for robot localization and obstacle avoiding. It is placed in a height of about 60cm from ground level. It is evident that in real world environments, many potential obstacles lie below this limit.

The experiments where held in two different settings. The first one took place in the campus gymnasium, where normally there are no obstacles below the laser height. This relatively safe zone allows the system to perform qualitative comparisons between using or not NGZ. First the map is constructed using the SLAM technique explained in Section 4.2. In this step the robot is being moved manually around the gymnasium. Next, the virtual obstacles for NGZs are superimposed on the map such as if they were real objects. As was explained in Section 5, placing virtual obstacles in a map renders it inappropriate for navigation, since these obstacles cannot actually be detected by the robot's laser sensor. To address this, two separated instances of the map were created: one used for navigation and another for the obstacle avoidance.

In order to have real NGZs in a controlled way, training mats were placed on the floor of the gymnasium. This facilitates the visual assessment of the methodology proposed. Two different tests were made in this scenario: the first one was done using the standard configuration of the navigation system; for the second one, the virtual obstacles for NGZs proposed were included.

Fig. 12. Gymnasium Test Scenario.

Fig. 12 shows this test scenario. The UGV is in its starting position and the goal is represented by the orange cone, also the green mats on the floor show the relative placement of NGZs. Using this setup, the performance of the navigation system, regarding its ability to handle the presence or absence of No Go Zones, was compared. During the first test, when only the navigation map was used, the robot was unable to avoid the obstacles because the laser rangefinder could not detect them. On the other hand, when a second layer containing the virtual obstacles was used, the path performed by the UGV circumvented the NGZs by passing around them. The Fig. 13 shows images of the platform while executing the two tests.

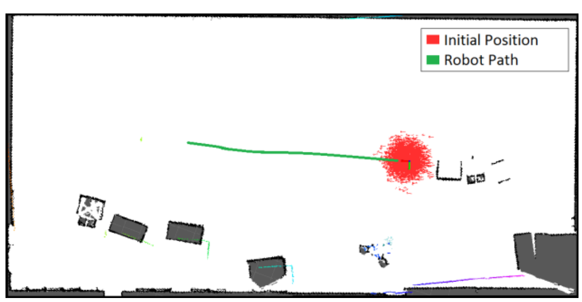


(a) Without Virtual Obstacles (b) With Virtual Obstacles

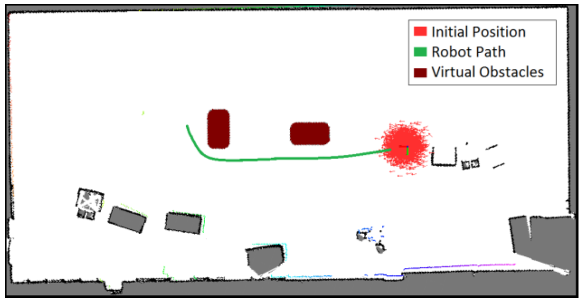
Fig. 13. Test with (b) and without (a) Virtual Obstacles for NGZ

It can be seen in Fig. 13(a) that the robot, in the first test, passed over the mats without detecting them, while in the second test, shown in Fig. 13(b), the robot makes a route around the mats and successfully avoiding the obstacles.

The result of the two first tests is more evident by comparing the path when NGZs where ignored and the path when they were taken in account. As it can be seen in Fig. 14(a), when the NGZ are ignored the path planning creates an almost straight line from the starting position to the goal. On the other hand, when the virtual obstacles are present, the path is deviated in order to avoid the mats as observed in Fig. 14(b).



(a) Ignoring NGZ



(b) Using Virtual Obstacles for NGZ

Fig. 14. Test paths with (b) and without (a) NGZ.

The second experiment was held outdoors in the university campus. For this setting, the map used was constructed in a previous experiment. In this more complex environment, a lot of objects are found below the laser height, examples of some of those elements are: sidewalks, pavements, benches, flower beds, or dustbins. Furthermore, there are other elements, such as fences, nets or sparse plants, which regardless of their height, cannot be detected correctly by the sensor. All these obstacles, which may present high risk for the robot, were marked as No-Go Zones. Therefore, corresponding virtual obstacles were manually added to the existing obstacle map. According with the process described, the original map was used for localization and the one with the modifications for the path planning and following. In order to avoid potential damages for the equipment, only the test with the NGZ map was executed.

Fig. 15. Outdoor Scenario.

The scenario where the second experiment was held can be observed in Fig. 15. There is a pavement and some plants that the laser cannot detect, and a fence on the other side of the road. If the original map were used, then collision would be very probable. By using the virtual obstacles in the NGZ, the robot can effectively avoid these collisions and it drives safely towards the desired goal position.

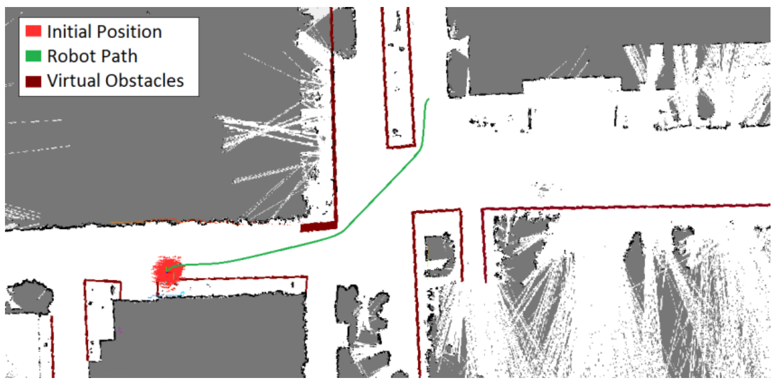


Fig. 16. Outdoor Test Using NGZ

The path that the UGV follow during this experiment is shown in Fig. 16.**Fig. 16. Outdoor Test Using NGZ** The original map and the virtual obstacles are also depicted. It is clear how the path is deformed in order to avoid the virtual obstacles as well as the normal ones, thus proving the ability of the proposed scheme to handle this complex scenario.

7 Conclusions

A system capable of navigating within simple and more complex real world environments was presented. Safe navigation in complex environments is assured by using a simple technique that allows the UGV to avoid zones where there may be risk of collisions or damage to the robot, consisting on adding virtual obstacles to the map created by the SLAM algorithm.

The main concerns of modifying a given SLAM map are studied and solved by modifying the navigation architecture, creating two instances of the map server and assuring that the virtual obstacles are taken into account in the path planning process.

Several tests were performed, first in indoor controlled scenarios and then in large and complex outdoor scenarios, all the tests show that the inclusion of the virtual obstacles enhances the navigation system, without having negative influences on the behavior of other modules of the system.

Future lines of work include the automatic inclusion of the virtual obstacles, by using an external sensor on a fixed location or onboard an mini UAV hovering over the UGV, and studying techniques to modify directly the cost map without using two instances of the map server.

Acknowledgements

This work was supported by the Robotics and Cybernetics Group at *Universidad Politécnica de Madrid* (Spain), and funded under the projects: ROTOS - Multi-robot system for outdoor infrastructures protection, sponsored by Spanish Ministry of Education and Science (DPI2010-17998), and ROBOCITY 2030, sponsored by the Community of Madrid (S-0505/DPI/000235).

References

Aliakbarpour, H., Núñez, P., Prado, J., Khoshhal, K., & Dias, J. 2009. An Efficient Algorithm for Extrinsic Calibration between a 3D Laser Range Finder and a Stereo Camera for Surveillance. *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pp. 1 - 6.

De Cubber, G., Doroftei, D., Nalpantidis, L., Sirakoulis, G. C., & Gasteratos, A. 2009. Stereo-based Terrain Traversability Analysis for Robot Navigation. *IARP/EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance.*

Garzón, M., Valente, J., Zapata, D., & Barrientos, A. 2013. An Aerial-Ground Robotic System for Navigation and Obstacle Mapping in Large Outdoor Areas. *Sensors, 13*(1): 1247-1267.

Kumar, S., & Awasthi, P. 2009. Navigation Architecture for Autonomous Surveillance Rover. *International Journal of Computer Theory and Engineering*, 1(3): 231 - 235.

Kunwar, F., & Benhabib, B. 2006. Rendezvous-Guidance Trajectory Planning for Robotic Dynamic Obstacle Avoidance and Interception.

Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 36(6): 1432 - 1441.

Ray, A. K., Behera, L., & Jamshidi, M. 2009. GPS and Sonar Based Area Mapping and Navigation by Mobile Robots. *Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference on*, pp. 801 - 806.

Yang, C., Jun-Jian, P., Jing, S., Lin-Lin, Z., & Yan-Dong, T. 2010. V-disparity Based UGV Obstacle Detection in Rough Outdoor Terrain. *A Acta Automatica Sinica*, 36(5): 667 -673.

Zhu, A., & Yang, S. X. (2007). Neurofuzzy-Based Approach to Mobile Robot Navigation in Unknown Environments. *Systems, Man, and Cybernetics, Part C: Applications and Reviews,* 37(4): 610 - 621.