Autonomous camp surveillance with the ROBUDEM robot: challenges and results

Geert De Cubber, Daniela Doroftei,
Kristel Verbiest, Sid Ahmed Berrabah

Royal Military Academy of Belgium
geert.de.cubber@rma.ac.be

ABSTRACT

Autonomous robotic systems can help for risky interventions to reduce the risk to human lives. An example of such a risky intervention is a camp surveillance scenario, where an environment needs to be patrolled and intruders need to be detected and intercepted. This paper describes the development of a mobile outdoor robot which is capable of performing such a camp surveillance task. The key research issues tackled are the robot design, geo-referenced localization and path planning, traversability estimation, the optimization of the terrain coverage strategy and the development of an intuitive human-robot interface.

1 Introduction

In military operations, it is often the case that large areas need to be patrolled in order to prevent the illicit entry of people / enemies. This task is a dull and cumbersome process, as it takes up valuable amounts of time and manpower which cannot be attributed to more "interesting" activities. Moreover, it is also a dangerous task as confrontation with enemy forces can bring the patrolling humans at risk. Autonomous robotic agents can therefore be a valuable alternative, provided that they can offer the same level of reliability as human patrol agents.

However, the design of a robotic agent able to operate in such difficult and varying conditions requires the careful consideration of multiple research aspects:

- The robot requires a geo-referenced localization and path planning capability. In this paper, we propose a methodology which fuses data from an on-board differential GPS system with an inertial measurement unit in an extended Kalman filter.
- The robot needs to estimate the degree of traversability of the terrain, which isn't an easy problem as the terrain may be very rough. In this paper, we propose a novel approach based on 3D-interpretation of the depth as seen by an on-board time-of flight camera.
- The robot needs to negotiate an optimal terrain coverage strategy to maximize the possibility of finding and intercepting intruders. In this paper, we propose a novel approach based on coverage maximization and intervention time minimization.
- For remote human operators, the robot must be easy to use. Therefore, an intuitive human-robot interface was developed which allows high-level communication with the robot without violating the real-time constraints of the robot controller.

2 Vehicle Description

The base vehicle used as a platform in the context of this paper is a Robucar TT model, produced by RoboSoft. A particularity with this kind of platform, called ROBUDEM, is that – as the front and rear axle can be steered independently – it is possible to adopt different movement modes (e.g. crab-like motion). However, in this setup, this possibility was not used in order not to over-complicate the design of the kinematic controllers. There were two main reasons for choosing this platform. First, as a 4-wheel-drive all terrain mobile platform it is well adapted to a rough outdoor environment. Second, its payload of 300 kilograms makes it possible to pack multiple sensors and on-board processing equipment.

Figure 1: The ROBUDEM vehicle used throughout this paper

3 Autonomous Operations

3.1 Processing

3.1.1 Computing Systems

The ROBUDEM vehicle features 3 embedded PCs, interconnected over an on-board LAN: a LINUX system for low-level motor control, a WINDOWS XP system (Intel Core) for visual and depth perception and a WINDOWS XP system (Intel Core) for mapping and navigation.

Conceptually, the computing architecture considers two levels of control: a low-level motor control layer which is monitored by the Real-Time Linux system provided by RoboSoft and a high level control layer featuring two embedded PCs. The high-level control was split over two computing systems to spread the computational burden and achieve (near) real-time functionality.

3.1.2 Processing Architecture & Development Process

Distributing the control architecture over the different computing systems was possible through the use of a distributed control architecture, entitled CoRoBa [1]. CoRoBa is a robot control architecture based upon the CORBA [2] architecture, but specially adapted to the needs of mobile robotics. Following this architecture, different components / modules were developed (in C++), each realizing part of the robot task. Following the CoRoBa development process, the development process started by drawing a UML diagram of the architecture, defining interfaces for all modules. A functional block diagram of the processing architecture is shown on Figure 2.

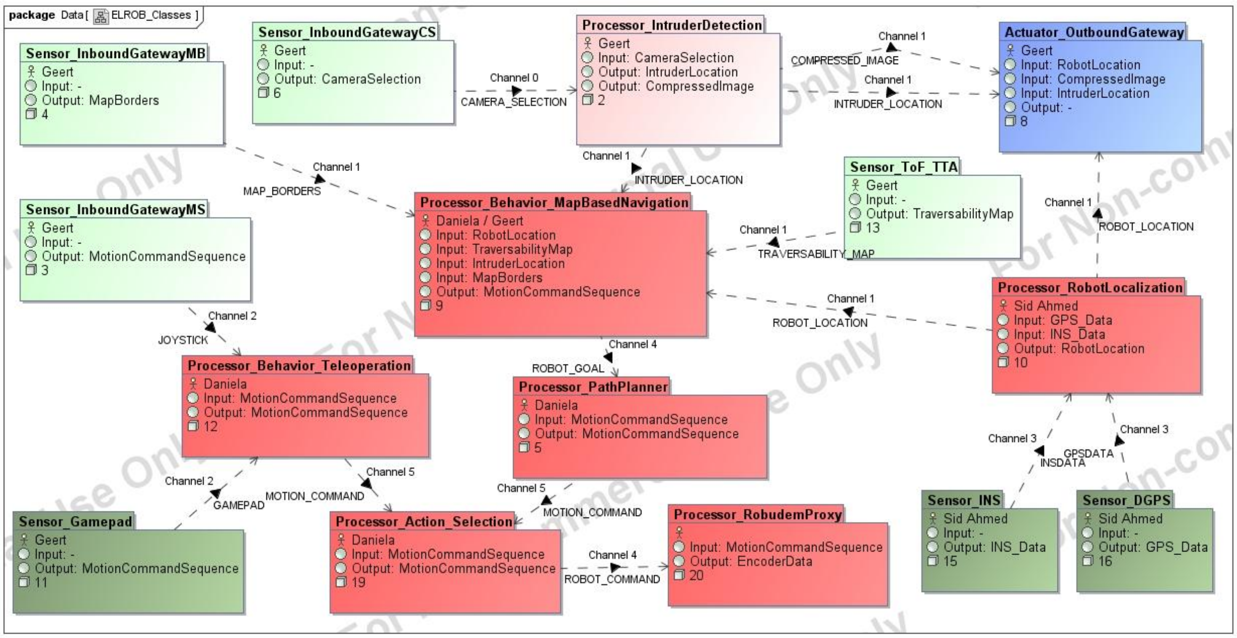

Figure 2: Functional block diagram of the processing architecture (green: sensors, red: processors, blue: actuators; dark colour: on PC1, light colour: on PC2)

Following Figure 2, it can be seen how information flows from the sensor elements (green), towards the different processors. In the end, the command control is effectuated by the RobudemProxy component indicated below.

Inter-component communication happens over TCP/IP. In order to reduce delays in this communication process, we set up a Gigabit Ethernet Hub on-board of the robot. The on-board Ethernet hub doubles as a Wireless Router, which enables remote access to the robot. For the connection to the remote base station, inbound and outbound gateways were foreseen. The reason for this is that all data which is sent over the wireless connection is sent using the UDP protocol instead of the TCP protocol, in order not to have problems with the unreliability of the wireless link. This leads to a network architecture as depicted on Figure 3.

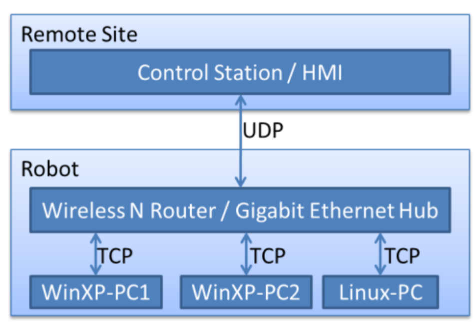

Figure 3: Network architecture

3.2 Visual Sensing

3.2.1 Sensor Setup

A multi-camera system was used for visual sensing. The system consists of a Point Grey Bumblebee stereo vision camera [3] and two Logitech HD Webcams, as shown by Figure 4.

To maximize the field of view, the webcams are placed at an angle of about 40° with respect to the robot axis. This results in a total field of view of about 160°.

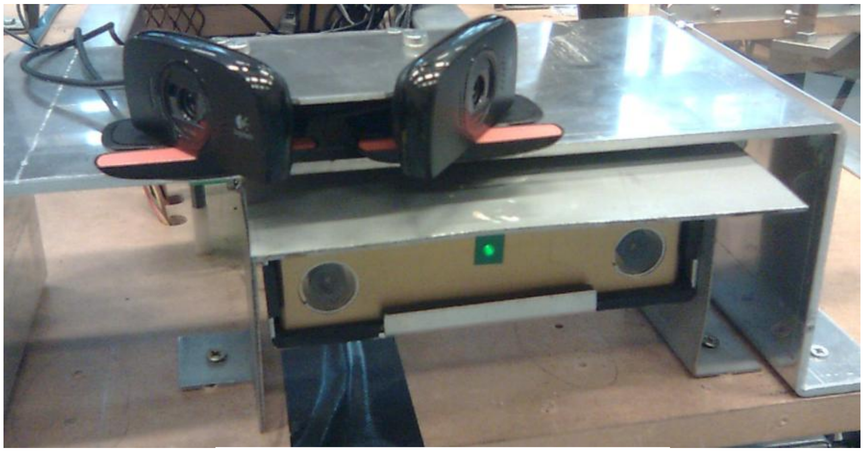

Figure 4: Visual Sensing System

3.2.2 Intruder Detection

Visual sensing is primarily intended to detect intruders. As the intruders are wearing a specific kind of clothing, the detection strategy is to search for a certain colour pattern in the visual images. To maximize the field of view, multiple cameras are used at the same time. Thus, an important issue in the development and design of the intruder detection mechanism is to balance the computational complexity and the associated processing requirements with the number of cameras that can be processed at once.

The intruder detection algorithm is implemented in OpenCV [4] and follows a 3-step approach:

1. In order to remove image noise, the images are pre-filtered and smoothed
2. Colour thresholding is applied in the HSV colour space to detect the clothing of the intruders independent of the illumination conditions
3. Standard morphology filtering operations like multiple image erosions and dilations are applied to increase the reliability of the detection result

At the end of this processing pipeline, the position *(x,y)* of the intruder (if detected) in the image plane is returned. This intruder detection is performed in 3 cameras (stereo camera + 2 webcams) at the same time and all detection results are fused. Following this fusion, it is possible to deduce the estimated position *(distance,* θ*)* of the intruder. The angle θ towards the intruder is straightforward to estimate from the known position in the image plane and the orientation of the camera. The distance of the intruder to the camera is (roughly) estimated by depth scaling: the detected size in pixels of the intruder is compared to the (known) size of the intruders. Using this approach, it is possible to detect intruders up to a distance of about 10m.

3.3 Depth Sensing

3.3.1 Sensor Setup

A PMDTec Camcube [5] Time-Of-Flight (TOF) camera was used for real-time depth sensing. The reason for using this sensor is that it is the first depth camera in the world which is able to operate outdoors under difficult illumination conditions. To optimize the field of view the sensor was mounted on top of the robot with an angle tilted towards the ground plane. The sensor has a modulation distance of 7.5m, so the tilting angle was specifically calculated in order for the sensor not to surpass that distance. This resulted in a field of view between 1m and 7.5m in front of the robot, as shown on Figure 5.

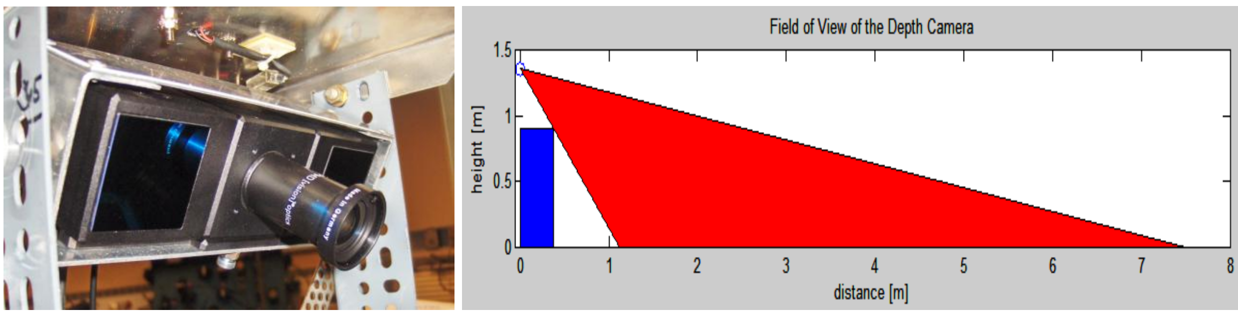

Figure 5: PMDTec Camcube 3D Camera and Optimization of its field of view

3.3.2 Traversability Estimation

Depth sensing is used for traversability estimation. Autonomous robotic systems operating in unstructured outdoor environments need to estimate the traversability of the terrain in order to navigate safely. Traversability estimation is a challenging problem, as the traversability is a complex function of both the terrain characteristics, such as slopes, vegetation, rocks, etc and the robot mobility characteristics, i.e. locomotion method, wheels, etc. It is thus required to analyse in real-time the 3D characteristics of the terrain and pair this data to the robot capabilities.

The methodology towards time-of-flight-based terrain traversability analysis extends our previous work on TOF-based [6] and stereo-based [7] terrain classification approaches. Following this strategy, the RGB data stream is segmen ted to group pixels belonging to the same physical objects. From the *Depth* data stream, the *v-disparity* [8] is calculated to estimate the ground plane, which leads to a first estimation of the terrain traversability. From this estimation, a number of pixels are selected which have a high probability of belonging to the ground plane (low distance to the estimated ground plane). The mean *a* and *b* colour values in the *Lab* colour space of these pixels are recorded as *c*.

The presented methodology then classifies all image pixels as traversable or not by estimating for each pixel a traversability score which is based upon the analysis of the segmented colour image and the *v-disparity* depth image. For each pixel *i* in the image, the colour difference and the obstacle density in the region where the pixel belongs to are calculated. The obstacle density is here defined as: $\delta_i = \langle o \in A_i \rangle / \langle A_i \rangle$, where *o* denotes the pixels marked as obstacles (high distance to the estimated ground plane) and A_i denotes the segment where pixel *i* belongs to. This allows us to define a traversability score as $\tau_i = \delta_i \left\| \mathbf{c}_i - \mathbf{c} \right\|$, which is used for classification. This is done by setting up a dynamic threshold, as a function of the distance measured.

An important issue when dealing with data from a TOF sensor is the correct assessment of erroneous input data and noise. Therefore, the algorithm automatically detects regions with low intensities and large variances in distance measurements and marks these as "suspicious".

Figure 6 shows an example of the terrain classification result. Obstacles are red, well traversable terrain is green and "suspicious" areas (not enough data) are blue. It can be noticed that the classification is correct, as the obstacle (the tree) is well-detected. In the upper left corner, there are some problems with foliage giving erroneous reflections (blue area), which is due to the sensor. As the TOF camera orientation is fixed, the traversability estimation of Figure 6 can be projected on the ground plane to retrieve a local traversability model, which is used for navigation.

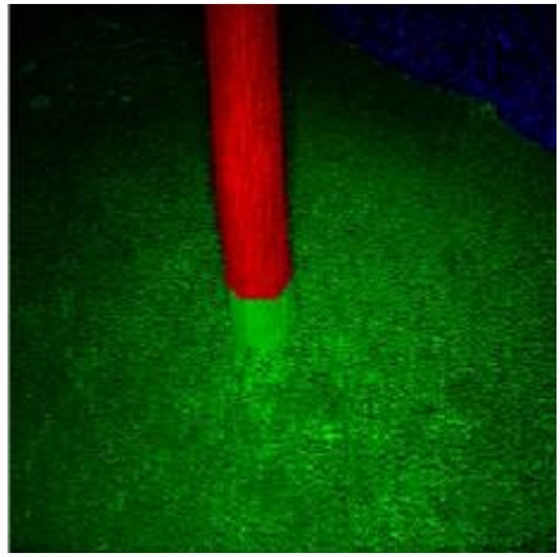

Figure 6: Traversability Estimation

3.4 Localization

To be able to operate and act successfully, the robot needs to know at any time where it is. This means the robot has to find out its location relative to the environment. In this application, we used an approach based Extended Kalman Filter (EKF) to improve the Global Positioning System (GPS) localization based on data from an Inertial Navigation System (INS) and wheels' encoders. In the following we will describe the used sensors and the proposed integration approach (for more details see [11-12]).

3.4.1 Global Positioning System

For outdoor applications, a GPS system could be used for robot localization in a geo-referenced map of the environment. However, GPS systems are subject to several sources of errors, among them, ionosphere and troposphere delays, signal multi-path, number of visible satellites, etc… A typical GPS receiver for civil applications provides 6-12 meters accuracy, depending on the number of available satellites. Another limitation for the use of GPS systems, is the necessity to operate in open aeria where the GPS receiver has permanently access to satellites. This is not always possible, especially in urban environments.
The positioning equations for n_s satellites in sight at time instant t can be defined as [13]:

$$r_i^t = \sqrt{\left(X_i^t - x^t\right)^2 + \left(Y_i^t - y^t\right)^2 + (Z_i^t)^2} + b^t + w_i^t \quad ; \qquad i = 1,\dots, n_s \tag{1}$$

where r_i^t is the pseudo-range between the GPS receiver and the i^{th} satellite, $[X_i^t, Y_i^t, Z_i^t\,]^T$ is the position of the i^{th} satellite, b^t is the GPS receiver clock offset in meters, w_i^t is the measurement error and $[y_1,y_2]$ is the vehicle position to be estimated (the vehicle altitude is $y_3=0$).

3.4.2 Inertial Navigation System (INS)

The inertial navigation system (INS) is a self-contained navigation technique in which measurements provided by accelerometers and gyroscopes are used to track the position and orientation of a robot on which the INS device is mounted.
Inertial navigation systems usually can only provide an accurate solution for a short period of time. As the acceleration is integrated twice to obtain the position, any error in the acceleration measurement will also be integrated and causes a bias on the estimated velocity and a continuous drift on the position estimate by the INS. Additionally, the INS software must use an estimate of the angular position of the accelerometers when conducting this integration. Typically, the angular position is tracked through an integration of the angular rate from the gyro sensors. These also produce unknown biases that affect the integration to get the position of the unit.
The differential equations relating the measured quantities to the dynamics are defined by [13]:

$$\dot{v}_{en} = R_{pn} f_p + g_n - (\Omega_{en} + 2\Omega_{ie}) v_e - \Omega_{ie}^2 p_n \tag{2}$$

Where f_p is the non gravitational acceleration, R_{ab} is the rotation matrix from frame *a* to frame *b*, p_n is the location of the vehicle in the frame *b*, Ω_{ab} is the rotation rate from frame *a* to frame *b*, v_a is the velocity relative to frame *a*, and g_n is the gravitational acceleration. The subscripts refer to the different coordinate frames, i.e., *i*: inertial frame, *e*: earth centred earth fixed frame, *n*: local geographic frame, *p*: platform frame.

$$\dot{p}_n = \begin{pmatrix} \dot{\lambda} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \frac{1}{R_\lambda} & 0 \\ 0 & \frac{1}{R_\phi \cos\lambda} \end{pmatrix} \tag{3}$$

Where λ and ϕ are the latitude and longitude of the vehicle, R_λ is the earth radius of curvature in the meridian, and R_ϕ is the transverse radius. These equations are integrated to obtain the vehicle position and velocity. This integration will entail a drift in stand-alone INS estimates due to a bias affecting the INS measurements.

3.4.3 Sensor Fusion

Kalman Filter is a statistical tool for the analysis of time-varying physical systems in the presence of noise. Its main goal is the estimation of the current state of a dynamic system by using data provided by the sensor measurements.
The state vector for GPS/INS/Encoders EKF integration filter is composed of the vehicle position, angle of direction, linear and angular velocities, and the GPS bias term.
The prediction in the EKF filter is based on the equation (2). The mean speed V and the yaw rate (angular velocity) $\dot{\gamma}$ of the vehicle at time *t* are computed based on the wheels encoders as follows:

$$V = \frac{\theta_r^t R_r + \theta_l^t R_l}{2} \qquad \dot{\gamma} = \frac{\theta_r^t R_r + \theta_l^t R_l}{L} \tag{4}$$

Where θ_r^t and θ_l^t are the angular velocities of the right and left rear wheels, respectively, and R_r and R_l their corresponding radius. *L* is the distance between rear wheels.
The update step is based on the measurement given by equation (1).

3.5 Coverage and Terrain Mapping

The robot builds two kinds of models of the environment: a coverage map and a terrain traversability map. Both maps are global geo-referenced maps, with a metric representation of the environment. The resolution of the maps is 20cm.

The coverage map is updated with each scan of the intruder detection sensing system. The robot stores the returned sensor data in a local coverage model, by storing the value ”1” in all cells which have been “viewed” by the robot sensors and where no intruder was detected. The visibility model employed here also takes into consideration possible occlusions due to obstacles. At each iteration, the information in the coverage model is “aged” by multiplying all entries of the local coverage map with a value between 0 and 1 (in practice: 0.99). The purpose of this approach is to represent the unreliability of “old” data. After some time, this leads to a coverage map as indicated by Figure 7.

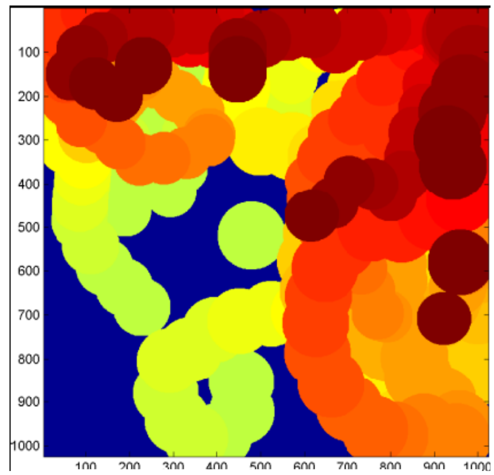

Figure 7: Coverage Map

The traversability map is updated with each scan of the traversability (depth) sensing system, by incorporating the local traversability data (the traversability model of Figure 6 projected on the 3D ground plane) in the global reference frame.

3.6 Vehicle Control

3.6.1 Information Maximization Approach

The approach towards vehicle control builds upon earlier work in [9] and [10]. An important aspect of the vehicle control behavior is that enemy forces should not be entirely able to predict the movements made by the robot, so there must be a certain randomization in the robot control. This randomization is achieved by selecting a number of discrete positions where the robot could go next. These random discrete points are extracted from a Gaussian distribution, taking into account the current robot position and orientation.

The candidate points are subjected to an optimization strategy. The objective of this optimization scheme is to maximize the global coverage and the ease of traversability. This means that for each of the candidate points, the trajectory towards the point is estimated. Then, for each point along that trajectory, the traversability and the coverage is assessed from the respective maps. Trajectories which pass untraversable areas are directly discarded. For selecting the best one of the remaining trajectories, an optimization strategy is followed, following the information maximization approach described in [10]. This means that trajectories which pass by difficultly traversable areas are penalized and that trajectories which pass by areas which were previously not visible are favoured. This is done by analysing the traversability and coverage map and calculating for each of the candidate points, representing a move to be made and leading to a new position *x*, the degree of optimality of the trajectory, taken into account the visibility model at location *x.* The point *x* which maximizes the information gain wins the optimization process and is thus selected as the place to go to detect intruders.

3.6.2 Path Planning

The used navigation strategy consists in robot tracking of a list of local goals defining the requested global path. For local goals tracking the robot uses a fuzzy logic controller to compute the change of orientation. The Controller is of a “Sugeno” type implementing a simple reasoning: If the goal is on the right and the distance is big then make a small turn to the right with big velocity. If the goal is on the right and the distance is small then make a big turn to the right with small velocity. More details on the implemented navigation strategy are given in [14].

3.6.3 Human-Machine Interface

To monitor mission progress and robot status a HMI was developed, which also provides the possibility for human intervention when deemed necessary. Figure 8 shows a screenshot of the HMI consisting of 4 main parts: camera output, map, info and dashboard.

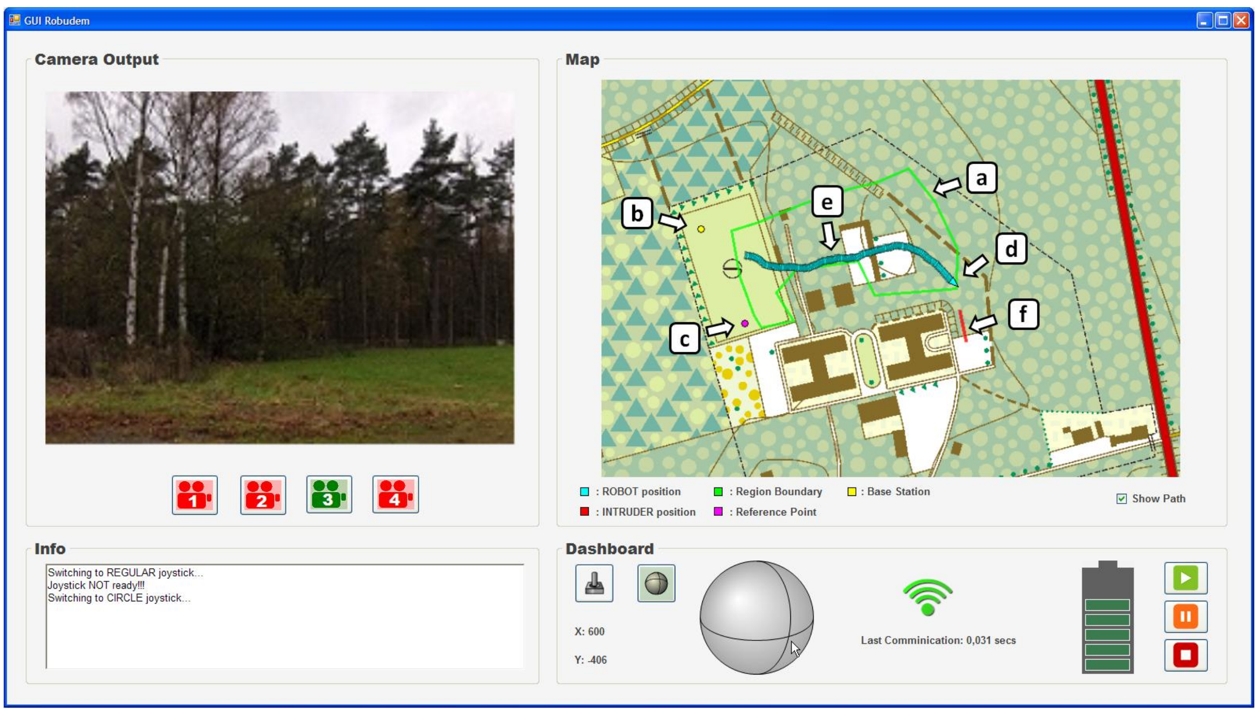

Figure 8: Screenshot of the HMI. Mission progress can be monitored by means of camera output, current robot position on the map, intruder position, communication status... The robot can optionally be controlled by a joystick.

The camera output is shown in the upper left corner. The user can select to see the images from one of the cameras mounted on the robot.

A map of the terrain is shown in the upper right corner. During the mission the robot must remain within certain boundaries of the terrain. A series of UTM coordinates are given at the beginning of the mission that delimit the permitted area for the robot to navigate in, the resulting region is visualized on the map (a). This region is also communicated with the robot. Other entities depicted on the map are the position of the base station (b) and the reference point (c) for the locally defined ENU coordinate frame used by the robot. The current robot position is shown as a triangle indicating its current position and orientation (d). The traversed path can be visualized as a trail if desired (e). Detected intruders are shown in the map with respect to the current robot position as a line indicating direction and approximate distance from the robot (f). As different coordinates representations are used in the region boundary coordinates (UTM), the map (Lambert72), the GPS (WGS84) and the robot (local ENU), all the necessary conversions are applied for correct information processing.

An information window is located in the bottom left corner for communicating miscellaneous information with the user relating to software, hardware, robot etc.

The dashboard is located in the bottom right corner. The robot can be controlled by the use of either a joystick or by dragging the cross on the circular joystick with a mouse. A communication indicator displays the time of last communication with the robot. When this time exceeds a predefined time, it will start blinking indicating communication has likely been lost. A battery indicator shows current battery level. Stop/Play/Pause buttons can be used to control the robot accordingly.

The HMI and the robot continuously exchange information via a wireless network; this communication is implemented using the open-source framework ACE (Adaptive Communication Environment) [15].

4 System Tests

4.1.1 Testing Strategy

System testing was performed at the RMA campus in the centre of Brussels. In order to conduct these experiments, a map of the campus was generated and an area to be patrolled was indicated on the human-machine interface.

4.1.2 Results

Major problems discovered during testing were:

- Lack of good GPS reception in the urban area surrounded by high buildings
- Traversability estimation testing is problematic as the area is mainly asphalted
- Reliability of the intruder detection under varying illumination conditions
- Reliability of the Traversability estimation under heavy sunlight conditions

5 Conclusions

In this paper, we have described the development process of an outdoor mobile robot, which is prepared for a camp surveillance mission. The robot is capable of detecting intruders using a visual sensing system. The robot is also able to navigate fully autonomously thanks to a novel traversability estimation methodology using a depth-sensing Time-Of-Flight camera and a navigational controller which incorporates a traversability and coverage model of the environment to select the optimal terrain covering strategy. Next to this autonomous functionality, a human-friendly user interface was developed to give also to non-experts the possibility to control the robot. A robotic system as presented here could be a valuable asset, e.g. in military operations where a large area needs to be patrolled autonomously by a robot or a team of robots.

References

[1] E. Colon, CoRoBa, a multi mobile robot control and simulation framework, Ph.D. Thesis, 2006, Vrije Universiteit Brussel - Koninklijke Militaire School

[2] D. Slama, J. Garbis, P. Russell. Enterprise CORBA. Prentice Hall

[3] Bumblebee: http://www.ptgrey.com/products/bumblebee2/bumblebee2_xb3_datasheet.pdf

[4] OpenCV: http://opencv.willowgarage.com/wiki/

[5] PMD CamCube http://www.pmdtec.com/products-services/pmdvisionr-cameras/pmdvisionr-camcube-30/

[6] G. De Cubber, D. Doroftei, H. Sahli and Y. Baudoin, "Outdoor Terrain Traversability Analysis for Robot Navigation using a Time-Of-Flight Camera", in Proc. RGB-D Workshop on 3D Perception in Robotics, Vasteras, Sweden, 2011

[7] G. De Cubber and D. Doroftei, "Multimodal terrain analysis for an all-terrain crisis Management Robot," in Proc. IARP HUDEM 2011, Sibenik, Croatia, 2011

[8] R. Labayrade and D. Aubert, "In - vehicle obstacles detection and characterization by stereovision," in Int. Workshop on In-Vehicle Cognitive Comp. Vision Systems, 2003

[9] D. Doroftei and E. Colon, "Decentralized multi-robot coordination for a risky surveillance application". in Proc. IARP HUDEM 2011, Sibenik, Croatia, 2011

[10] D. Doroftei and E. Colon, "Decentralized Multi-Robot Coordination for Risky Interventions", Fourth International Workshop on Robotics for risky interventions and Environmental Surveillance-Maintenance, Sheffield, UK, 2010

[11] S.A. Berrabah, Y. Baudoin, GPS data correction using encoders and INS sensors, Third International Workshop on Robotics for risky interventions and Environmental Surveillance-Maintenance, RISE'2009- Brussels, Belgium, January 2009.

[12] S.A. Berrabah, Y.Baudoin, Data Association for Robot Localization in Satellite Images, International Journal of Advanced Robotics, Vol 6, N°1, March 2010.

[13] J.A. Farrel, M. Barth, The global positioning system and inertial navigation, New York: McGraw-Hill, 1999.

[14] S.A. Berrabah, C. G. Rozmarin, Robot Navigation Based on Adaptive Fuzzy Controller, Third International Workshop on Robotics for risky interventions and Environmental Surveillance-Maintenance, RISE'2009- Brussels, Belgium, January 2009.

[15] ACE. Website: http://www1.cse.wustl.edu/~schmidt/ACE.html