Lessons Learned in a Ball Fetch-And-Carry Robotic Competition

Submitted: 10th January 2013; accepted: 10th February 2013

Marco Cigolini, Alessandro Costalunga, Federico Parisi, Marco Patander, Isabella Salsi, Andrea Signifredi, Davide Valeriani, Dario Lodi Rizzini, Stefano Caselli

DOI: 10.1431/JAMRIS_1-2014/11

Abstract:
Robot competitions are effective means to learn the issues of autonomous systems on the field, by solving a complex problem end-to-end. In this paper, we illustrate Red Beard Button, the robotic system that we developed for the Sick Robot Day 2012 competition, and we highlight notions about design and implementation of robotic systems acquired through this experience. The aim of the contest was to detect, fetch and carry balls with an assigned color to a dropping area, similarly to a foraging navigation task. The developed robotic system was required to perceive colored balls, to grasp and transport balls, and to localize itself and navigate to assigned areas. Through extensive experiments the team developed an initial prototype, discovered pitfalls, revised the initial assumptions and design decisions, and took advantage of the iteration process to perform successfully at the competition.

Keywords: *Robotic Competition*

1. Introduction

Robot competitions constitute an effective mean in robotic education [4, 10]. Through the contest students can learn to address robotic problems and tasks, to work as a group, to design complex systems including mechanical structure, electronic components and software architecture, and to check the initial assumptions with the results on the field. In common robotic practice as well as in student projects, researchers and students tend to concentrate on specific aspects of robotics such as perception with a specific sensor, localization or navigation. Thus, the main result is a single component or an algorithm, whose experimental assessment is usually accurate but aims at achieving proof-of-concept and sometimes artificial demonstrations. On the other hand, solutions developed for a robotic competition must be effective and take into account the interaction of each component with the whole robotic architecture. A method that works correctly in laboratory experiments may not achieve the same results when used in different setups like those involved in a competition. Thus, students can learn through competitions that "the whole is greater than the sum of its parts" as well as appreciate the importance of tests on the field.

Sick AG, a leading manufacturer in sensor technologies and laser scanners, organizes *Sick Robot Day*, a competition open to student teams from universities and other educational institutions aimed at promoting mobile robotics and automation technologies in education. In 2012 Sick Robot Day reached its fourth edition. While previous editions involved perception and navigation capabilities, in the latest challenge the robots were required to detect, fetch and carry balls with an assigned color to a designated area called pen. The proposed problem falls in the well-studied category of the foraging tasks [2]. The contestants had to address several problems including which sensors to use for detecting balls, obstacles and pen, how to carry the balls, how to find the pen, and which tasks to execute. The robot systems were developed by participating teams under imperfect knowledge of the final competition environment, shown in Figure 1.

Fig. 1. The arena of Sick Robot Day 2012 delimited by a fence and three pens. A pen is shown in the bottom-left.

In this paper, we illustrate the robotic system implemented for Sick Robot Day 2012 by a team of students of the University of Parma and the lessons learned during its development. The implementation of the control architecture required the team to make design decisions and to verify the obtained results on the field. Experiments have proven fundamental for discovering pitfalls and for developing more robust and effective solutions. The robotic competition has proven a valuable experience to check initial assumptions and to learn how to implement components that can perform the required tasks in practice. The final autonomous system has proven quite effective and our robot, *Red Beard Button*, achieved first place at the competition.

The paper is organized as follows. Section 2 summarizes the competition rules. Section 3 illustrates the architecture of Red Beard Button and shortly describes the development history. Section 4 illustrates the experiments performed before and at the competition as well as the problems met. Section 5 discusses the lessons learned through this experience, while sec-

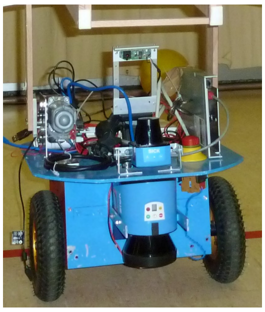

Fig. 2. The robot equipped with Sick LMS100 and TiM300 laser scanners, Logitech C270 camera, and the motorized fork lift

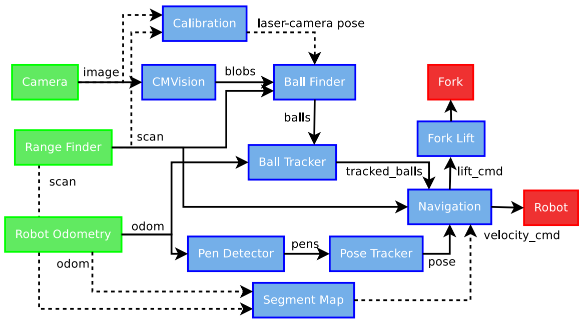

Fig. 3. The robotic architecture of the system composed of ROS framework nodes

tion 6 provides the concluding remarks.

2. Competition Rules

This section summarizes the rules of Sick Robot Day 2012 in order to clarify the design decisions to the reader. The contest takes place in an indoor polygonal arena, whose diameter size is about $10 \div 20\ m$. The arena contains balls of three different colors with $20 \div 25\ cm$ diameter. The ring fence of the arena gaps in three zones where three pens are placed. Each pen is distinguished by one of the three colors and is used as a starting position for one of the robots and as the ball dropping area.

The aim of challenge is to detect, fetch and carry to the pen as many balls of the assigned color as possible. The contest consists of several 10 minutes rounds (also called runs) and three robots compete at the same round, each looking for balls of a given color. Each robot participates to two rounds and a different color is assigned in the two rounds. The score of each round is equal to the number of balls of the assigned color, except for penalties. The balls of a wrong color reaching the pen are subtracted from the score of the round. Furthermore, every contact of the robot with the fence is sanctioned with a half point and collision with another robot leads to instant disqualification from the current round. Contact with balls is allowed irrespective of their color. Thus, the position of the balls is likely to change during a run since robots may carry or push them. The final placement of the teams depends on their best performance in either of the two rounds. Several details, like ball colors, exact dimensions of the balls and of the pen, or number of balls placed inside the arena, were not defined by the rules of procedure and have been discovered by teams with short notice on the very day of the competition.

3. Robot Architecture

In this section, we present the final architecture of the Red Beard Button robot implemented for Sick Robot Day 2012. We also briefly discuss the variants implemented before reaching the final one and the motivation for the design decisions. The system has been decomposed into parts to address the three main challenges posed by the competition: ball detection, ball picking and transportation, and robot localization for returning to the pen. These three tasks are coordinated by the robot navigation system.

The robotic platform used in Red Beard Button is a MobileRobots Pioneer 3DX equipped with two laser scanners, Sick LMS100 and Sick TiM300, and a Logitech C270 camera (Figure 2). The scan plane of the LMS100 laser scanner is approximately parallel and $10\ cm$ above the ground plane. The TiM300 laser scanner has been included in the architecture to overcome ball occlusion problems. However, it has hot been used in the final robot setup due to design decisions discussed later in the paper.

The perception component detects the balls of the required color by performing sensor fusion. The device adopted for carrying balls is relevant for the navigation strategy. Two ball picking structures have been implemented: a simple static fork, that requires specific navigation policies to avoid loosing the ball, and a motorized fork, that lifts and cages the ball thereby avoiding any occlusion in front of the robot. A localization and mapping algorithm is required to estimate the robot position w.r.t. the pen area where the ball must be dropped. Since the map of the environment is unknown, the robot must extract landmarks to find its position. The only stable elements in the given competition arena are the fence and the pens. Finally, the navigation component handles the robot task state and coordinates perception and action using the information provided by the other components. The different tasks have been implemented as ROS[1] nodes and are illustrated in Figure 3. In the following the details of the main components are described.

3.1. Navigation

The navigation component is responsible for the execution of robot motion and for the management of the state of competition. The navigation task coordinates all the other tasks, since it receives and uses their outputs to carry out the robot main task. In the arena, the robot interacts with different kinds of objects:

- *static objects* like arena fence, that must be avoided in order not to incur into penalties;
- *semi-static objects* like balls, that may be moved or avoided depending on the adopted policy;
- *dynamic objects* like the other robots, that may lead to disqualification if a collision occurs.

The presence of several dynamic and semi-static objects in the arena makes path planning an ineffective solution, since a plan may quickly become outdated due to the change in obstacle configuration. Thus, a reactive approach has been preferred for robot navigation. The development of navigation components has been simplified by the choice of the motorized fork lift that is discussed in section 3.3. The navigation task is divided into several subtasks, each corresponding to a robotic behavior with a specific goal:

- *exploration*: the robot moves and searches target balls;
- *ball approaching*: when a target ball has been detected, the robot approaches it;
- *ball grasping*: the robot reaches the ball and raises the fork;
- *transportation*: the robot returns to the pen to drop the ball;
- *ball release*: the ball is released into the pen.

Figure 4 illustrates the flowchart of navigation decomposed into subtasks.

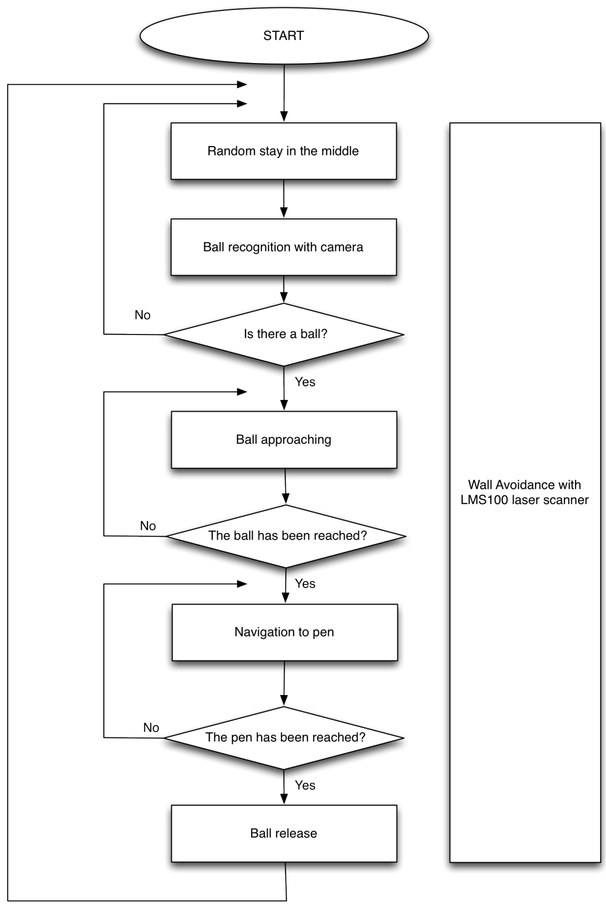

Fig. 4. Flowchart of navigation decomposed into subtasks

Safe navigation is guaranteed by a collision avoidance behavior, which interrupts the execution of current subtasks when the distance from the closest obstacle is less than a given threshold ($0.55\ m$). When collision avoidance is active, the robot steers in the opposite direction w.r.t. the obstacle until free space is observed in front of the robot. Such behavior is disabled only during the approach to or the release of a target ball.

The exploration task has been developed using a hybrid approach: the main behaviour is a standard *stay-in-the-middle* behavior [1] that allows the robot to move in the environment keeping about the same distance from the nearest obstacles on its left and on its right. In order to move to all the directions and explore the environment, every 12 seconds the robot randomly steers. During exploration, the robot speed may reach $0.45\ m/s$ and the fork lift is held raised in order not to occlude the laser scanner.

When the ball detector component observes a target ball, the ball approaching behaviour is activated. Then, the mobile robot rotates towards the centroid of the ball and moves with a speed proportional to the ball distance. If the ball is lost, e.g. the collision avoidance switches on, the exploration task is reactivated to search and reach other interesting balls. However, the ball tracking module described in the following avoids intermittent observations of the goal and prevents unnecessary transitions between ball approaching and exploration.

When the distance to the ball is less than a given threshold (about $0.70\ m$), the fork is lowered and ball grasping task is performed. During ball grasping, perception of the target balls and obstacles is handled by a specific procedure due to the limited field of view of the camera, which prevents the observation of balls, and the occlusion of the laser scanner caused by the lowered fork. The robot moves towards the ball until it correctly grabs the ball or fails. The outcome of such operation is monitored by a selected subset of frontal range finder beams that are not occluded. When the ball is caught, the robot raises the fork and starts to navigate towards the pen. Otherwise, after having lifted the fork, the robot resumes exploring the environment. Since the ball is caged by the fork, the ball never falls down during the lift.

The navigation back to the pen is driven by the information provided by the localization module. This subtask directs the mobile robot towards a goal point placed in the middle of the pen, setting the orientation properly to approach the pen frontally. In order to prevent collisions, the *collision avoidance* behavior runs in background with higher priority. Moreover, when the robot is near to the pen ($1.2\,m$) the linear velocity is reduced to $0.2\ m/s$ to perform a more accurate motion.

When the final position is reached with the right orientation, the *ball releasing* task is activated. After lowering the fork, the robot pushes the ball in the pen moving forward and suddenly backward. If the ball is correctly released, the robot rotates around its axis about $180°$ and restarts the exploration of the arena to search another ball of the assigned color.

3.2. Ball Detection

The main task of the detection module is to distinguish the target balls from all the other objects placed in the arena. Therefore, during exploration the robot must be able to segment its sensor measurements and extract those segments that meet the requirements of goal objects like shape, aspect ratio, size, colour and a position consistent with physical constraints (e.g. balls lie *on* the ground). Since two different types of sensors, namely a RGB camera and a laser scanner, are available, recognition of candidate target balls is separately performed in the two sensor domains (laser scans and images) and the results are associated only in a second phase. In this way, the algorithm takes advantage of both devices and, at the same time, processing can be performed by two separate components. The laser scanner provides an accurate estimation of ball position, while the camera is able to assess the color and the aspect ratio of the region-of-interest (ROI) corresponding to balls.

The robot control application, developed for the ROS framework, consists of four nodes. The first node is the *CMVision* package (Color Machine Vision project) [5] that extracts blobs of a given color from the frames acquired by the camera. Since the segmentation of images is independent from the laser scanner, it has been easy to integrate this library package into our system. The second node is dedicated to the calibration procedure, which is performed only offline before using the detector. The third node is the ball detection core component, which processes laser scans and associates laser segments to the color blobs extracted by CMVision. The fourth node is a *ball tracking* node that addresses the intermittent detection caused by laser scan and image segmentation failures or by missing associations between the two sensor domains.

The purpose of the calibration node is the estimation of the transformation matrix between a point P_{laser} in the laser reference frame and the corresponding point P_{img} in the image plane and viceversa as expressed by equation

$$P_{img} = KK \cdot {}^{C}_{L}T \cdot P_{laser}$$

where KK is the intrinsic parameters matrix of the camera and ${}^{C}_{L}T$ the transformation matrix from laser frame to camera frame. While there are several packages for estimating KK, the few libraries for assessing ${}^{C}_{L}T$ strongly depend on the setup and the calibration object. The calibration object must be chosen so that it is possible to detect and match a pair of homologous points in the two sensor domains. We have investigated the algorithm proposed in [12] that jointly calibrates a laser scanner and a camera by matching slices of a planar checkerboard with the plane of the same checkerboard. Unfortunately, we have not achieved satisfactory results, possibly due to the noisy perception of the checkerboard or to numerical stability problems of the proposed method.

Thus, we have implemented an iterative procedure based on the manual association of the measurements of a ball acquired with the laser scanner and the camera. Although not automatic, this method allows quick and reliable estimation and has the advantage of using the object to be detected (the ball) as a calibration target. This method exploits the same segmentation procedures of the image and of the laser scan used during detection. However, since the algorithm starts from an initial guess of the transformation ${}^{C}_{L}T$ to be estimated, the blobs returned by CMVision are filtered according to strict criteria on the area and aspect ratio of the balls. Then, the centroids of the laser segments are projected into the image plane according to the current value of ${}^{C}_{L}T$ and roughly associated with the blobs. The user can iteratively change the values of translation and rotation parameters of ${}^{C}_{L}T$ until the projected laser points overlap with the centroids of blobs.

After the initialization of parameters, the detection cycle consists of four steps:

- segmentation of laser scan using a discontinuity threshold and selection of intervals checking their diameter;
- projection of these valid segments in the image frame;
- if a segment falls into a bounding box, it takes on its colour and it is classified as belonging to a ball;
- publication of the recognized balls list, including useful information for navigation and collection, such as colour or position in the laser reference frame.

The tracking node has been designed to address intermittent detection of balls due to temporary failure of the ball detector illustrated before. The node keeps an estimation of the observed balls by updating their position w.r.t. the robot according to robot odometry and the sensor observations. The tracking algorithm implements Kalman filter equations. Objects that have not been observed for a given time interval, are removed from the state.

Tests in the laboratory, with controlled light, have shown that the algorithm is able to identify and locate with satisfactory accuracy all the balls. The association is correct, even though the calibration is performed with the manual algorithm. However, larger environments with reflections and abrupt light changes strongly affect the performance of the CMVision component. The problems of this component are further discussed in section 4.

3.3. Ball Grasping Device

An important requirement to succeed in the competition was to provide the robot with a device to move the balls that are inside the arena. Among several possible solutions, we have built a static fork and a motorized fork lift. The first device consists of two plain wooden bars that can be used to push the target ball as shown in Figure 5(a). This device requires the availability of an additional laser scanner at a different height (in our case the TiM300) since the LMS100 is occluded during ball transportation. The second device is a motorized fork lift, shown in Figure 5(b), that can raise the ball when it has been caged among the fork bars. Since the fork is raised during exploration and ball transportation, the laser scanner is occluded only during ball grasping and release.

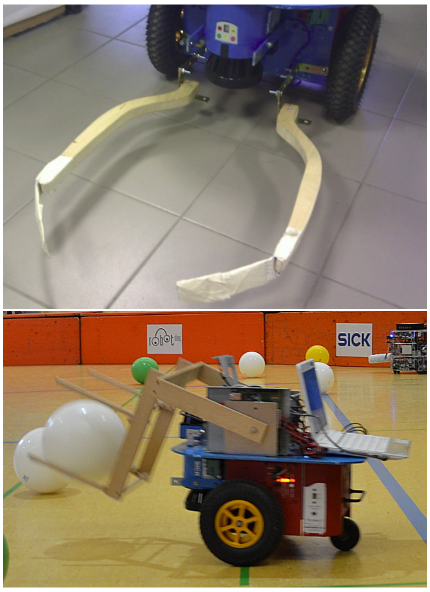

***Fig. 5.** The static fork (a) and the motorized fork lift (b) built to cage and carry balls*

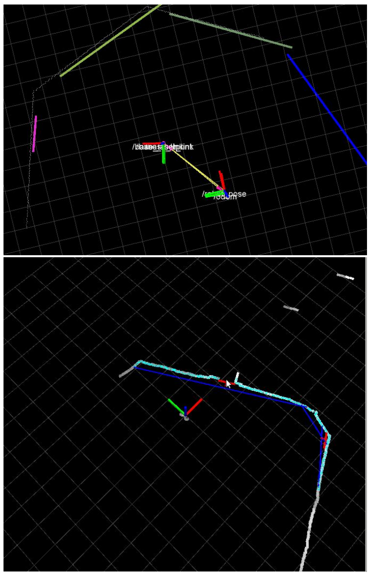

***Fig. 6.** Ouputs of the two localization and mapping nodes: the segment landmark graphical map (a) and the pen landmark localizer (b)*

We experimented with both solutions until few weeks before competition. The static fork was appealing for its simplicity in construction and reliability, but ball transportation proved difficult since the ball was not caged. The fork lift required some iterations in mechanical and electronic design and was eventually preferred for the competition. Indeed, with the fork lift the robot does not lose the ball while moving because the ball is well caged without occluding the sensor.

The construction of the motorized fork lift requires a mechanical structure, an electric motor, the electronic components for its control, and a software interface with the laptop computer. The system consists of the following components:

- a DC geared motor with a high reduction ratio, so as to decrease the maximum speed and increase the torque output;
- a *Microchip Technology Inc PICDem2* board, which consists of a microcontroller, the output interface with the powerboard, an Ethernet port and other elements not used in this project;
- a power board, built in the university laboratory, which controls the power supply of the motor according to the logic signals output from the PICDem2 board;
- two limit switches, which signal when the fork is completely raised or lowered.

The limit switches are the only devices available to monitor the fork state. No other information is available while the fork is in an intermediate position.

A ROS node is responsible for the communication between the laptop computer and the control board through a custom protocol on TCP/IP port. The microcontroller waits for commands from the computer and sends control signals to the motor when it receives a command. To control the motor, the board generates a PWM modulation: a pair of square waves, one opposite the other, are generated and overlapped into a single signal to the motor. The amplitude of the signal is $12\ V$. The final performance of the system is satisfactory, since the fork reliably raises and releases balls.

3.4. Localization and Mapping

Localization is a crucial task for the successful accomplishment of the proposed challenge. When a ball is fetched using the fork lift, Red Beard Button must reach its pen and drop the ball there. Without knowing its pose, the robot cannot plan its path or even guess the direction toward the pen. The information provided by odometry is unreliable, since odometry is sensitive to steering and its error increases with the travelled path length. In order to estimate its own position and orientation, the robot requires a map containing the landmarks or implicit references that can be easily detected in the environment. When such map is not available, the system must be able to build a map from the acquired measurements. This problem has been investigated by robotic research for decades and is known as *simultaneous localization and mapping* (SLAM) [11].

In the scenario of the Sick Robot Day 2012 competition, a major complication is represented by the lack of stable and continuously observable landmarks. The arena shown in Figure 1 chiefly consists of balls, whose position rapidly changes and which occlude the border of the arena. The fence and the pens, which appear as gaps in the fence, are the only invariants in the scene. Both types of candidate landmarks are distinguishable in laser scans by detecting aligned points. Two different approaches have been developed for map construction and localization, each using one of the two landmarks. Figure 6 illustrates the output of the two methods.

The first method builds a map of segment landmarks to represent the boundaries of the arena. These boundaries do not change, but they may be occluded by other dynamic or semi-static elements of the environment like balls and other robots. The scan plane of laser scanner Sick TiM300 does not intersect the balls. Thus, this range finder can be used to extract boundary segments, although its maximum range is limited to $4\ m$. More in detail, the algorithm performs four main operations. First, the scans acquired by the laser scanner are segmented into intervals and are split according to endpoints [8]. In the second step, the parametric model of the segment and its uncertainty are computed through least square estimation within the geometric limits represented by the two segment endpoints [3]. The association between the segments and the landmarks already stored in the map is performed using Hausdorff and Mahalanobis distances. Finally, a Graph SLAM algorithm takes the odometric data, the previous landmarks, and the landmark measurements given by the associations to estimate the pose of the robot. The sensor model uses the *SP Map* representation [6] applied to segments. Instead of using Bayesian filtering, the map has been represented by a graphical model that encodes the constraints between the variables of the problem. The estimation has been performed using the *G2O* library [9] for the optimization of constraint networks. Unfortunately, this promising and general approach has proven unreliable in this case due to the limited visibility of the fence, as well as prone to numerical instability.

The second localization method, developed to address the limitation of the first solution, focuses on the detection of the pens. Although there are only three pens in the arena (one for each robot that concurrently takes part to a round) and only the initial pen is frequently observed, the detection of a gap in the fence is rather robust. Furthermore, the range finder view of the pen is seldom occluded by balls, since the robot starts with the closest balls right in front of the dropping area and progressively cleans the space. The developed method exploits the odometry to predict the robot pose and then corrects the estimation by using the landmark when available. After taking the ball, the robot tries to reach the pen assuming that it is located in the origin of the reference frame, located in the initial pose. Moreover, it activates the pen detection routine. A pen has been modelled with two segments lying on almost parallel lines with a gap in the between. The laser scanner data are used to build this model using an algorithm based on the *Hough Spectrum* and *Hough Transform* [7]. Whenever a pen is detected, the system checks whether the pen is the one assigned to the robot for the current round by computing the Euclidean distance between the pen and the map reference frame origin. If this is the case, the current estimation of the robot pose, which is updated using odometry at each iteration, is corrected according to the observation.

During the competition the second approach has been used. This approach has the advantages of being simpler, more goal-oriented and it better fits the problem. The first approach would have been more general and the provided correction potentially more frequent. However, it suffers from the inaccuracy of the fence detection with several occluding balls, from the numerical instability of segment landmarks and from the ambiguity of landmark association criteria, either based on the segment endpoint position or on the support line parameters. Moreover, the environment of the competition had a lot of balls that occluded the LMS100 laser whereas the arena was too large to rely on the TiM300.

(a)

(b)

(c)

***Fig. 7.** Environments where Red Beard Button has been tested: the RIMLab Robotics laboratory (`lab`) (a), the gym of the University of Parma (`gym`) (b), and the Sick Robot Day arena (`arena`) (c)*

4. Experiments

The development of the robotic architecture illustrated in the previous section has been supported by experiments in the Robotics Laboratory of the Department of Information Engineering (`lab`) and in the gym of the University of Parma (`gym`). The second environment has been chosen for its presumed similarity with the Sick Robot Day arena (`arena`). The three environments are illustrated in Figure 7. In this section, we present the experimental assessment, the correction proposed to the observed pitfalls, and the final results achieved in the competition.

4.1. Training Tests

The initial tests in `lab` allowed the development and fast testing of some components of the robotic architecture. In particular, the implementation of the ball detection algorithm, the fork lift and the robot navigation core have taken advantage of the laboratory test. However, only the next set of tests in gym allowed the full assessment and the identification of the system pitfalls. There are two main differences between lab and gym: the scale and the lighting conditions. The hallway of the department can be approximately divided into two narrow trunks, each with size about $10 \times 2.5\ m$. On the other hand, the region of gym used in the experiments has $18\ m$ diameter and is more similar to the competition field. Such large field does not constrain the robot motion and allows the tuning of parameters like maximum linear and angular speeds, segmentation thresholds, and pen size.

During such extensive tests, which have taken place for about a month, new problems and limitations have been detected and addressed. First, the ball detection algorithm failed when the light conditions were difficult as shown in Figure 7(b). Abrupt changes in light intensity, reflections on the ground, etc. make the color segmentation of the acquired frames unreliable. The three colors of the balls (green, yellow and white) have been announced about 2 months before the competition, when the detection algorithm had already been implemented (and team members were busy with exams and other academic duties). In order to lessen this problem, some solutions have been developed. For example, the ball tracking module described in section 3.2 has been applied to keep the previously detected position of balls in case of intermittent detection. The extended components worked well in the case of green and yellow balls. However, the detection of white patches in the image is unreliable when the light conditions are not fully controlled like in lab. This perception pitfall remained unsolved in the final competition field, since a radical change of approach and new design of the ball detection component would have been required to address it. In fact, color segmentation using an off-the-shelf component like *CMvision* has proven unreliable outside the laboratory. A customized, laser-driven approach could have been more effective.

An unforseen deadlock condition has been identified in the fork control module. In a trial, while the robot approached the ball, the fork has been lowered too early causing the block of the fork on the ball. Since the robot waits for completion of fork lowering, the system stays indefinitely in such state. A trivial solution to address such sporadic condition has been implemented by setting a timeout on the lowering action. If this action is not completed before the deadline, the fork lift is raised.

In the gym, the localization component has proven to be crucial for reliable robot operation in large environments. Estimation of robot pose w.r.t. the pen can be performed using only the odometry only if the size of the environment and the travelled path are limited. However, if the robot moves for 10 minutes at high speed and frequently steers, the odometric error of Pioneer 3DX largely increases and the localization of the robot becomes unreliable. In early odometry-based trials the robot missed the pen with an error up to $5\ m$. We then developed the two methods discussed in section 3.4: localization and mapping using segment landmarks and localization using pens as landmarks. Experiments on the two methods had to cope with the limited availability of the gym as well as with the time pressure of the incoming competition. After some experiments in the gym, we adopted the approach based on pen detection, which was simpler, more robust and effective. Although only the starting pen is usually observed due to the travelled path and occlusions, Red Beard Button has always been able to reach its target configuration.

4.2. Competion Results

Sick Robot Day 2012 took place on October 6th in the Stadthalle in Waldkirch (Germany). Although the rule of procedure describes the general geometrical features of the competition field, the arena (Figure 7(c)) was seen for the first time by the 14 teams from Germany, Czech Republic and Italy only few hours before the beginning of the competition. The diameter of the real arena was about $15\ m$ and the arena contained 29 balls for each of the three colors. The morning was devoted to setup of the mobile robot, to parameter tuning and system configuration testing whenever the field was available. Assignment of ball colors and of the rounds have been announced to the teams just before the morning trials. The competition started at $2\ pm$ by alternating 10 rounds of 10 minutes each.

In its first round, Red Beard Button had to collect green balls. The detection algorithm has always been able to correctly identify the items with this color both during the morning tests and in the competition. In fact, during the competition the robot has collected 7 green balls in the assigned time. However, Red Beard Button hit the arena fence four times due to too low safety distance in the ball dropping phase. Hence, the final awarded score was 5, accounting for 2 point penalty assigned.

In the second round, Red Beard Button was required to collect white balls. As mentioned above, correct white ball detection was an unsolved problem. Due to the non-uniform lighting and too strong false positive control, Red Beard Button was unable to fully identify white balls in the arena. Thus, the ball detection method never estimated false positives, whereas other teams incurred in significant penalties due to the collection of balls with the wrong color.

The 5 points score achieved in the first round eventually won our team the first place in the competition, with the second and third teams obtaining 3 points and 1 point respectively. The whole system implemented in Red Beard Button has worked properly, except for the arena edge hits in the first round and the white ball detection problem in the second one.

5. Discussion

Experiments and the competition itself have allowed the team member to learn some lessons about the design and implementation of autonomous robotic systems. In the following, we propose a list of suggestions that summarize our experience.

- Perception is the most important reason for the success or failure in accomplishing a given robotic task. The correct detection of green balls has allowed the successful execution of the foraging task, while the uncertain identification of white balls within cautious acceptance policies has led to an opposite result. The interpretation of sensor measurement is critical when the decisions of the autonomous robot depend on the outcome of a classifier.
- The robotic system becomes more efficient and less prone to error when the sensor measurements are collected and organized in a coherent representation. The importance of the environment representation increases with the complexity of the task and the scale of the environment where the robot operates. This lesson has been proven both by the ball tracking module and by the robot global localizer. The former method is an example of short-term memory suitable to track dynamic and ephemeral objects like balls. The success of localization depends on the presence of invariant elements of the environment that can be used as landmarks.
- The complexity of the solution should be proportional to the complexity of the problem. The color segmentation used to detect balls in images has proven unsatisfactory in many cases. Such naive approach has not worked well for white balls outside the robotic laboratory, whenever the color is not an invariant property of the target objects. On the other hand, solutions like the general segment-based graphical map algorithm have proven too complex for the problem.
- Robot system development should be guided by experiments on the complete system. Each robot component has been tested in depth in the lab before the integration tests in the gym, but the problems arose only with the complete system. Unpredicted conditions may depend on the interaction between robot components and the environment: perception deficiencies may appear only when the robot (and the sensor) moves, the motion of the robot and the actuated components may be affected by objects (e.g. the fork blocked by a ball), etc. Furthermore, the experimental setup should be as similar as possible w.r.t. light conditions, dimension, etc. to the environment where the task must be performed. Of course, experiments are time consuming and the complete system is not available until the development reaches an advanced state.
- Robot developers often design and implement the system under uncertain information and cannot control all the possible conditions. For example, the color of the balls was not initially known and the ball detector has been designed without exploiting such information. Moreover, the high density of balls in the competition arena, which could be critical for a planner, was apparent only the day of the competition. Several critical conditions arose only during the last extensive experiments. Thus, the only possible countermeasure is to arrange multiple solutions to address the same task and to anticipate the criticalities by performing experiments in difficult environments. Indeed, we developed two ball carrying tools and two localization methods, and for each feature the most effective approach has been selected.

6. Conclusion

In this paper, we have presented Red Beard Button, a robotic system designed for the Sick Robot Day 2012 competition, and the lessons learned during its development. The aim of the contest was to detect, fetch and carry balls with an assigned color to a dropping area, similarly to a foraging navigation task. The developed robot system consists of several software and electromechanical components to perceive colored balls, to grasp and transport balls, and to localize the robot and navigate to assigned areas. Some subtasks like ball grasping and localization have been addressed by multiple solutions and experiments have proven fundamental for selecting the most effective one. Through extensive tests in the field, the team discovered pitfalls, revised the initial assumptions and design decisions, and took advantage of the iteration process to perform successfully at the competition.

Acknowledgements

We thank Sick Italia Spa for sponsoring the logistic of the competition to the *Red Beard Button* team and Sick AG for promoting and organizing the competition.

Notes

[1] ROS (Robot Operating System - http://www.ros.org) is an open-source meta-operating system for robots.

AUTHORS

Marco Cigolini – Dipartimento di Ingegneria dell'Informazione, University of Parma, viale Parco Area delle Scienze, 181A, 43124 Parma, Italy, e-mail: cigolini@ce.unipr.it, www: http://rimlab.ce.unipr.it/.

Alessandro Costalunga – Dipartimento di Ingegneria dell'Informazione, University of Parma, viale Parco Area delle Scienze, 181A, 43124 Parma, Italy, e-mail: costa@ce.unipr.it, www: http://rimlab.ce.unipr.it/.

Federico Parisi – Dipartimento di Ingegneria dell'Informazione, University of Parma, viale Parco Area delle Scienze, 181A, 43124 Parma, Italy, e-mail: parisi@ce.unipr.it, www: http://rimlab.ce.unipr.it/.

Marco Patander – Dipartimento di Ingegneria dell'Informazione, University of Parma, viale Parco Area delle Scienze, 181A, 43124 Parma, Italy, e-mail: patander@ce.unipr.it, www: http://rimlab.ce.unipr.it/.

Isabella Salsi – Dipartimento di Ingegneria dell'Informazione, University of Parma, viale Parco Area delle Scienze, 181A, 43124 Parma, Italy, e-mail: lsalsi@ce.unipr.it, www: http://rimlab.ce.unipr.it/.

Andrea Signifredi – Dipartimento di Ingegneria dell'Informazione, University of Parma, viale Parco Area delle Scienze, 181A, 43124 Parma, Italy, e-mail: a.signi@ce.unipr.it, www: http://rimlab.ce.unipr.it/.
Davide Valeriani – Dipartimento di Ingegneria dell'Informazione, University of Parma, viale Parco Area delle Scienze, 181A, 43124 Parma, Italy, e-mail: davide.v@ce.unipr.it, www: http://rimlab.ce.unipr.it/.
Dario Lodi Rizzini* – Dipartimento di Ingegneria dell'Informazione, University of Parma, viale Parco Area delle Scienze, 181A, 43124 Parma, Italy, e-mail: dlr@ce.unipr.it, www: http://rimlab.ce.unipr.it/.
Stefano Caselli – Dipartimento di Ingegneria dell'Informazione, University of Parma, viale Parco Area delle Scienze, 181A, 43124 Parma, Italy, e-mail: caselli@ce.unipr.it, www: http://rimlab.ce.unipr.it/.

*Corresponding author

REFERENCES

[1] P. Althaus and H. I. Christensen, "Behaviour coordination in structured environments", *Annals of Mathematical Statistics*, vol. 17, no. 7, 2003, pp. 657–674, DOI 10.1109/IRDS.2002.1041610.

[2] R. Arkin, *Behavior-Based Robotics*, MIT Press, 1998, ISBN 978-0262011655.

[3] K. O. Arras. *Feature-Based Robot Navigation in Known and Unknown Environments*. PhD thesis, Swiss Federal Institute of Technology Lausanne (EPFL), Thèse No. 2765, June 2003. DOI 10.5075/epfl-thesis-2765.

[4] T. Braeunl, "Research Relevance of Mobile Robot Competitions", *IEEE Robotics & Automation Magazine*, vol. 6, no. 4, 1999, pp. 32–37, DOI 10.1109/100.813825.

[5] J. Bruce, T. Balch, and M. Veloso, "Fast and Inexpensive Color Image Segmentation for Interactive Robots". In: *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2000, pp. 2061–2066, DOI 10.1109/IROS.2000.895274.

[6] J. Castellanos, J. Montiel, J. Neira, and J. Tardos, "The SPmap: A Probabilistic Framework for Simultaneous Localization and Map Building", *IEEE Transactions on Robotics*, vol. 15, no. 5, 1999, pp. 948–952, DOI 10.1109/70.795798.

[7] A. Censi, L. Iocchi, and G. Grisetti, "Scan matching in the Hough domain". In: *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005, pp. 2739–2744, DOI 10.1109/ROBOT.2005.1570528.

[8] D. Douglas and T. Peuker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", *Cartogr. Int. J. Geogr. Inf. Geovis.*, vol. 10, no. 2, 1973, pp. 112–122, DOI 10.3138/FM57-6770-U75U-7727.

[9] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization". In: *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011, pp. 3607–3613, DOI 10.1109/ICRA.2011.5979949.

[10] P. Lima and R. Cortesao, "IROS 2012 Competitions", *IEEE Robotics & Automation Magazine*, vol. 20, no. 1, 2013, pp. 10–12, DOI 10.1109/MRA.2012.2236232.

[11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press: Cambridge, MA, 2005, DOI 10.1017/S0269888906210993.

[12] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)". In: *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2004, pp. 2301–2306, DOI 10.1109/IROS.2004.1389752.