Localization and Navigation of a Mobile Robot in an Office-like Environment

Paulo Alves*, Hugo Costelha*†, Carlos Neves*‡
*Escola Superior de Tecnologia e Gestão de Leiria, Instituto Politécnico de Leiria, Leiria, Portugal
†Robis, INESCTEC, Porto, Portugal
‡INESCC, Coimbra, Portugal
PauloFernando.Alves@edp.pt, {hugo.costelha, cneves}@ipleiria.pt

***Abstract* - This article focuses on the localization and navigation of a mobile differential robot in an indoor office-like environment. These are fundamental issues to service robotics, which is a branch with a strong market growth. The work implements a vision tracking system, environment mapping, route planning and navigation for an autonomous robot application inside services buildings. One goal of the methodology is its application with low cost equipment. The test bed chosen was a Pioneer P3-DX robot [16] in a service building, with an attached USB webcam, pointed at the ceiling to take advantage of the position of the light fixtures as natural landmarks. The robot location is estimated through two distinct probabilistic methods: a particle filter, when there is no information about the starting location of the robot, and the Kalman filter, given the convergence of the particle filter. Both methods use the detection of light fixtures together with the robot kinematics as information to estimate the pose. The mapping of the environment and its obstacles is obtained from the localization estimates and the information gathered by ultrasound sensors, representing the entire navigation space discretized in the form of an occupation grid. Planning the navigation path is determined by a simple search algorithm, namely the Wavefront algorithm, based on the information contained in the occupancy grid. For a given path, navigation is performed with obstacle avoidance using the virtual forces method. Replanning is used to recover from local minima situations.**

Keywords: Mobile Robots, Localization, Navigation, Mapping, Office Environment.

I. INTRODUCTION

Mobile robotics is currently considered a major area of interest, where there is a great willingness by the scientific community to develop its state of the art. Moreover, its interdisciplinary nature and the complexity of the associated problems make this advance depend on the development of mathematical tools, sensors, actuators and the materials themselves, which limit their progress. Whichever vehicle you want to use an autonomous behavior in, whether in health care, assistance at home, industry or even in an office environment, the basis of their performance includes localization and navigation tasks. In the particular case of offices, despite the circulation of a significant number of people, the environment is highly structured, with unique features which facilitate localization and navigation.

Mobile robots have unique capabilities to perform tasks requiring mobility that can be automated, freeing humans to other more creative activities [11]. Today there are several tasks already automated in different areas, such as the mapping and monitoring of land surface [1], large-scale agriculture [3], environmental monitoring [20] and industry [2]. On the other hand, there are high risk tasks or ones which are even impossible to accomplish by humans, such as the verification of explosive devices [5], the exploration and mapping of areas contaminated by biological or nuclear waste to assess the damage [15], or even space exploration of distant celestial bodies [13]. There are other emerging application areas, such as personal assistance [21], rehabilitation and entertainment. For this type of applications to be possible with some degree of autonomy, robots must be equipped with various types of sensors (force sensors, inertial sensors, GPS, odometry, distance sensors or vision), to feedback the work environment perception. The robot movement can be made either by wheels or tracks, on land robots, propellers or wings, for aerial robots, and control surfaces or propellers for aquatic robots. Whether on land, air or water, two fundamental problems coexist: localization and navigation. These two problems are interdependent and crucial for autonomous mobile robotic systems to interact with the physical world correctly by extended periods of time.

In [18] a filter of Simultaneous Localization and Map Building SLAM [12] is presented, which successfully exploited a low cost vision system, with means to improve dead reckoning pose estimate and maintain a correct estimate of the pose of a mobile robot for constructing a map of the environment. As in the present work, limited resources of hardware and low cost vision where used. These limitations impose the use of image processing algorithms involving low processing requirements. To avoid the computational complexity, the paper proposes a fast algorithm for feature extraction which includes the lens distortion model from the SLAM filter. In this methodology, as well as in the present work, the light fixtures positions were used as natural landmarks. Unlike the present work, in [18] the light fixture poses are not known a priori and will be mapped during navigation. Present work provides a better accuracy in pose estimate, since light fixture poses are known instead of calculated based in detection and pose estimation.

This paper proposes the application of various methods to solve the problem of localization and navigation of a Pioneer P3-DX mobile differential robot in an indoor environment (the robotics laboratory and adjoining hallways of *Escola Superior de Tecnologia e Gestão*, ESTG, from the Polytechnic Institute of Leiria). It describes an autonomous navigation algorithm, requiring only the indication of the destination point. The algorithm takes advantage of the environment structure elements using low cost sensors, while minimizing the use of techniques which are computationally expensive, so as to be used in systems with limited computational resources. This work aims to be the base for a guide robot or an autonomous system at ESTG.

The problem of localization was divided into two distinct situations: when there is confidence in the estimate of the real pose and when that confidence is insufficient. In the situation where there is a good confidence in the pose estimate, a Kalman filter [10][14] is used, which is an unimodal estimator, allowing to correct the accumulation of odometry errors during navigation. When there is insufficient certainty in the pose estimation, a Particle Filter [10] is used, which is a multimodal estimator that assumes that the robot can be in a number of possible poses. To use these probabilistic methods one needs information about the environment. As one of the goals is to save computational and economic resources, an off the shelf artificial vision system was used, with a video webcam mounted on top of the robot, pointing at the ceiling in order to recognize the light fixtures as natural landmarks.

For a planned navigation one needs to determine waypoints and, therefore, the occupation of the workspace must be integrated in the model. With this goal in mind, a representation of space was implemented in the form of an occupancy grid, where every cell represents the probability of occupation of the corresponding space. This grid is updated based on the distance information provided by the ultrasound sensors and actual pose estimate, as long as the localization estimate confidence is acceptable. Determining the path to follow is implemented with the Wavefront algorithm, based on the information in the occupancy grid. Finally, the virtual forces method is used to execute the planned path, taking into account the distance information from ultrasound sensors to avoid unexpected obstacles.

Aiming to give the work a strong portability and reuse, and given current trends in the field of robotics, the implementation of this system was based on the ROS environment [19]. It is a development environment which provides a set of libraries and tools that facilitate the implementation of mobile robotics applications. This includes a strong abstraction layer of the hardware, peripheral drivers, data viewers, messaging, package management, and makes easier the reuse and publication of the developed work. This software is available under various open source licenses, mainly the BSD license.

The remainder of this paper is organized as follows. Section II addresses the issues of artificial vision, recognition of objects in images from the webcam and the representation of objects as natural landmarks in the robot coordinate frame. Section III deals with localization, divided in global localization and accumulation of errors correction. Section IV details the mapping of the environment in the form of an occupancy grid. Section V describes the planning algorithms and path execution with unexpected objects avoidance, while Section VI describes the use of the ROS environment. Conclusions and future works are presented in Section VII.

II. Artificial Vision

One of the goals of this work was the use of low-cost equipment and algorithms that do not require high computational resources. For this purpose one used the recognition of elements of the ceiling, namely the pose of light fixtures, as illustrated in Fig. 1. Generally there are no obstacles between the robot and these, allowing for an easier and consistent detection. Moreover the fact that this type of structure typically exists in any office-type environment, allows the developed algorithms to be used generally for indoor service robots.

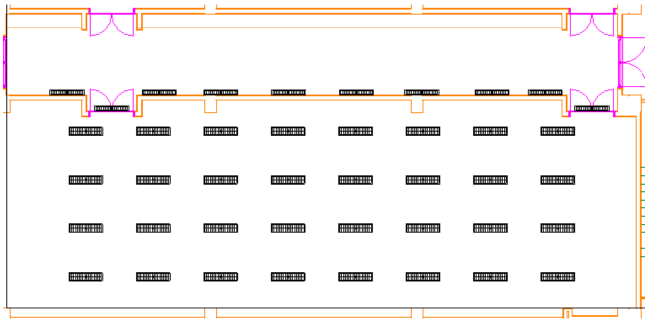

Fig. 1. Workspace plan with the light fixtures location.

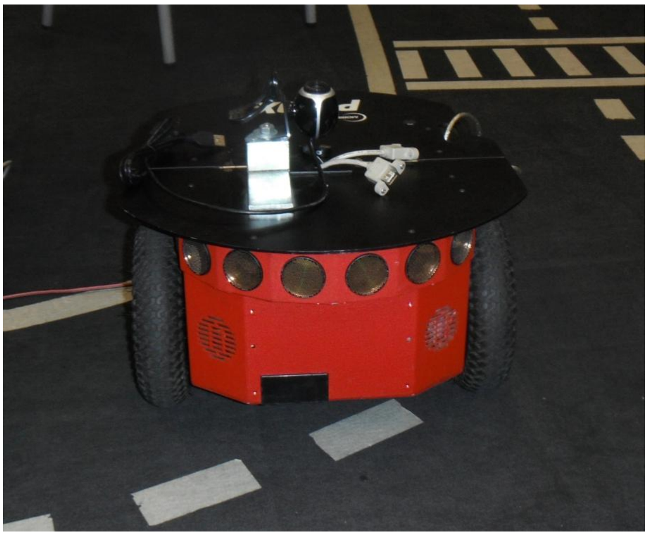

Fig. 2. Webcam mounted on top of the robot.

The vision system used in this work is based upon a low cost webcam, mounted on top of a mobile robot, pointed at the ceiling, as illustrated in Fig. 2. The camera transmits the data to the robot PC via an USB connection. The images are processed for the detection of the light fixtures position and orientation on the robot local reference frame, according to an algorithm divided into three steps: feature extraction, characteristics validation and calculation of the pose in local coordinates.

In order to reduce the computation resources needed, the image is processed in grayscale. To better adjust the image to the lightning conditions the sensor exposure time is controlled to keep the average pixels value in a given range. The algorithm performs feature extraction using information from the luminance level of the pixels of the digitized image. Assuming that a light fixture on, has a high value of luminance, a threshold operation is performed with a high value to distinguish pixels corresponding to a lighted light fixture. Frequently the level of natural illumination is adequate and the lamps are off, or partially off. In this case, the pixels corresponding to an unlighted light fixture has a luminance lower than the ceiling. For detection of these light fixtures another threshold operation is performed, this time with a reduced value and lower than operation. These two operations are combined and the resulting binary image is subject to erosion followed by dilation operations in order to smooth the contours and eliminate small areas. These areas correspond to reflections or small elements too dark or too bright which might be rejected in the validation process.

After these operations a contour detection is done using the Canny algorithm [8] and the calculation of the respective areas. Only contours with sufficient area and length, corresponding to light fixtures dimensions in image units, will be considered. Due to the existence of light fixtures with two lamps with only one lighted up, and light fixtures in corridors containing only one lamp, the width value considered valid lies between the height of the light fixture with two lamps and the height of a light fixture with only one lamp. For these values a tolerance is allowed for the validation operation to be more immune to variations in ambient lighting. This procedure discards objects which do not match light fixtures dimensions, which would have a negative impact in the localization process.

Using OpenCV library functions [7], the contours moments are determined, and then the mass center and the corners of the smallest rectangle that surrounds the entire contour, which provides the orientation as shown in Fig. 3.

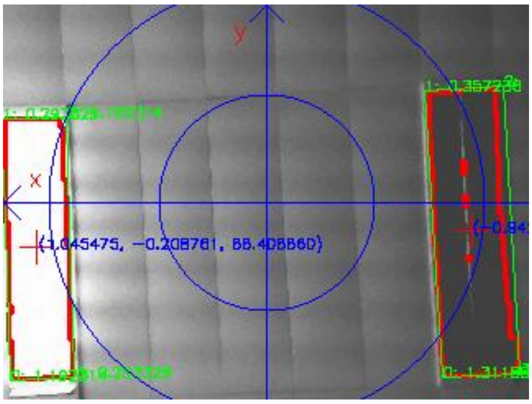

Fig. 3. Results of the light fixtures detection algorithm.

This illustrates the situation in which one light fixture is found on and the other off. The preview of the detected contours are red, the minimum rectangle that surrounds the contours in green, and the validated information of the light fixture pose in the robot reference frame are shown in blue.

III. LOCALIZATION

When there is no initial information about the pose of the robot, or the pose estimation confidence is low, one uses a particle filter to solve the global localization problem, which initially generates hypothesis randomly distributed in the available space [4]. Having solved the global localization problem with high enough confidence, the particle filter algorithm is abandoned and the Kalman filter, which is an unimodal estimator, is used instead.

A. Global localization

The following describes the application of the particle filter to estimate the robot pose, based on the robot odometry and the pose of the light fixtures detected by the previous method, following the standard three-step approach: prediction, update and resampling.

In the prediction step each particle, which represents a pose hypothesis, is updated based on the information of the robot linear displacement and rotation (odometry), including Gaussian white noise. This particle update is limited to the free map space, since the robot cannot be inside a wall or outside the map.

In the update step, based on the observations of the light fixture, the weight of each particle is calculated. The weight is proportional to the proximity between the expected values for the observations and the values obtained. There is a particular problem that my cause ambiguities, since there is no information about witch light fixture was observed; it's assumed that the observed light fixture is the closest from expected position. Two factors are included which help in disambiguation: the orientation of the light fixtures relative to the robot and the space occupation probability for the position of each particle. A noise factor is added corresponding to the observation error model, even when there is no motion. These calculations are represented in equation (Eq.1):

$$w_j = \frac{1}{p(k)} \sum_{i=1}^{p(k)} \frac{1}{1 + k_d \sqrt{\left(x_i - \hat{x}_{j_i}\right)^2 + \left(y_i - \hat{y}_{j_i}\right)^2} + k_\theta \operatorname{abs}\left(\theta_i - \hat{\theta}_{j_i}\right)} \times P\left(r_j, c_j\right) + \operatorname{norm}(\sigma) \tag{1}$$

where x_i, y_i and θ_i correspond to the pose in world coordinates of the landmark i, $\hat{x}_{j_i}, \hat{y}_{j_i}$ and $\hat{\theta}_{j_i}$ correspond to the landmark i pose in global coordinates based on the pose of particle j, k_d and k_θ are adjustable gains enabling to vary the weight sensitivity to errors in position and orientation, $P(r_j, c_j)$ is the probability of the space corresponding to particle j position to be free, where (r_j, c_j) is the pair row / column corresponding to the position of the particle j in the occupancy grid, and $\operatorname{norm}(\sigma)$ is an independent factor representing the observation noise. After this procedure the weight of all the particles is then normalized so that the sum of all weights is one.

The resampling step consists in replicating the best 10% of particles over the worst 10%. It was further introduced a random redistribution of 5% of the worst particles, to enable recovering from situations where no particles where placed near the robot true pose.

To estimate the pose, one considered clusters of particles with a circular boundary with a radius smaller than the minimum distance between light fixtures, to distinguish the different groups associated with each light fixture possible sighting (see Fig. 4 for an example). The first cluster starts with the particle with the highest weight. Then, for each particle in order of importance, one checks if it lies within the boundaries of an existent cluster. If so, it is considered to belong to that cluster and contribute to its accumulated weight. If the location of the particle is far from existing clusters, it starts a new cluster with the same procedure. At the end of this process all particles will be part of a given cluster, even if it consists of only one particle. The weight of each cluster will be the sum of all weights of the particles that are associated with it. The estimated pose is the average pose of all particles poses of the cluster with higher weight. Generally available clustering techniques, such as K-means [9] or Nearest Neighbor are used with good results, but these are computationally intensive, which is contrary to our goal of having an algorithm with a low computational demand.

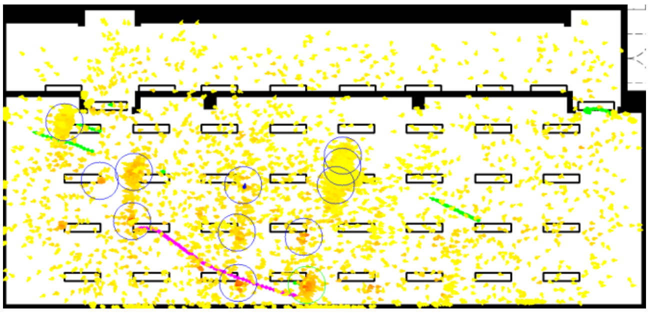

Fig. 4. Particle distribution.

B. Localization update after global localization

Having solved the global localization problem, the particle filter is abandoned and the Kalman filter started. The main advantage is the processing time, much lower in the case of the Kalman filter (in our case in the order of tens of ms for each iteration, while the particulate filter takes between 200 and 300 ms by iteration, meaning a factor between 20x and 30x). This happens because in the Kalman filter the calculations are made only for one pose, whereas in the particulate filter calculations are done for all particles. The computational resources are then more available for mapping the environment and navigation (recall that mapping is not done while the robot is not confident enough in its pose estimate). Note that generally the robot only has to perform the global localization when starting, meaning that most of the time it is using the Kalman filter and not the particle filter. Given that the system is non-linear, the Extended Kalman Filter is used, as described in [10]. Here the state of the system to estimate is the robot pose, the control signal are linear and angular velocity from odometry and the external information is the observed pose of the light fixtures, in world coordinates.

Fig. 5 shows loop closure navigation through the robotics laboratory at ESTG of the robot in teleoperation mode. Localization process started with the particle filter (green) and switched to Kalman filter (blue) after convergence. The odometry data is represented in magenta.

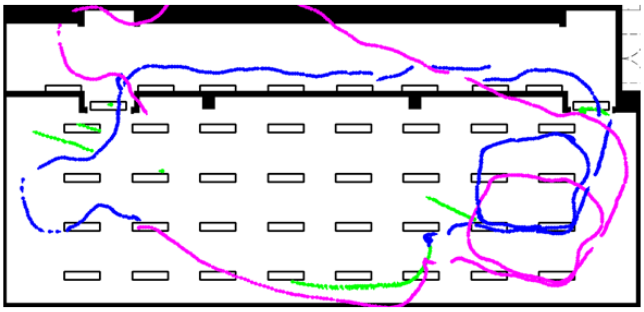

Fig. 5. Loop closure navigation.

IV. OCCUPANCY GRID

Space is represented as an occupancy grid based on the estimated pose provided by the localization algorithm and only when there is an acceptable degree of confidence in the estimate. Otherwise the occupancy grid is not updated.

For each distance measurement using the ultrasound sensors one assumes that the volume of the sonar beam, until the measured distance, is likely free space. If the measurement is smaller than the maximum range, at this distance, space is likely to be occupied. For the sake of reducing the computational complexity, one chose to consider only a line segment starting in the sensor with its orientation and ending after the measured distance.

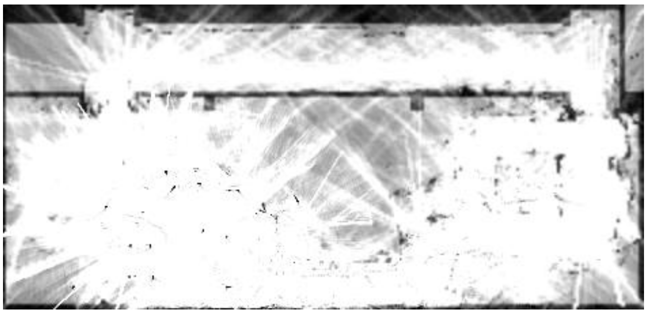

Fig. 6. Occupancy grid.

Regarding the resolution of the sonar sensor and the ultrasonic dispersion cone, a 10 pixel/m resolution to the occupation grid was chosen. This satisfies the compromise between a good representation of space occupation and processing resources. An example of the occupancy grid is shown on Fig. 6.

V. NAVIGATION

After solving the localization problem, one needs to compute the movements required for the robot to reach the desired pose. This question can be divided in two: deciding which way to go and determining the execution speed.

The Wavefront algorithm [10] was chosen for path planning, since it is one of the simplest solutions that guarantees finding a path if it exists, with immunity to local minima situations. Furthermore its use is possible because the navigation environment is discretized as an occupancy grid. The operating principle of the algorithm is essentially as follows: initially each cell considered as occupied is labeled with the value 1, and each free cell with the value 0. The cell corresponding to the target is labeled with the value 2. In the first step, all cells surrounding the target, using eight-connectivity, are labeled with the value 3. In the next step all cells with the value 0 which involve the value 3 are labeled with the value 4. These causes a "wavefront" to grow from the target cell where, in each iteration, all the cells in the "wavefront" have the same path length, in pixels, from the target cell. This procedure terminates when the "wavefront" reaches the starting point.

To account for the size of the robot, the occupancy grid must be expanded to configuration space, which consists in expanding the occupied space for the robot action radius, as showed in Fig. 7. In order to smooth the navigation trajectory, a temporary objective for navigation is defined, advanced relative to the current position on the path. To do so, the current position on the path was considered as the closest cell to the robot's position in the path set. The objective will be a number of cells along the path, determined by the compromise between a smoother trajectory, and the correct following of the planned path, given existing obstacles. Fig. 9 shows an example of a full planned path.

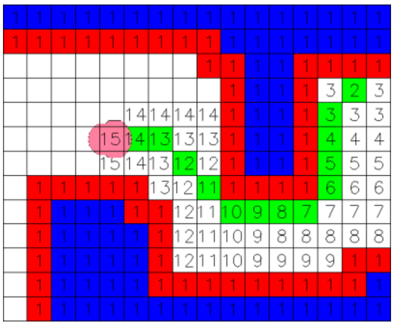

Fig. 7. Path in configuration space

While navigating the path defined above, unexpected obstacles can be founded. To account for this situation, one uses the information from the ultrasound sensors through the method of virtual forces [6] [17]. When an obstacle is detected on the path ahead, or close to it, its position is associated with a virtual repulsive force and the temporary navigation goal is associated with a virtual attractive force. The orientation of the robot is obtained from the resultant of these forces.

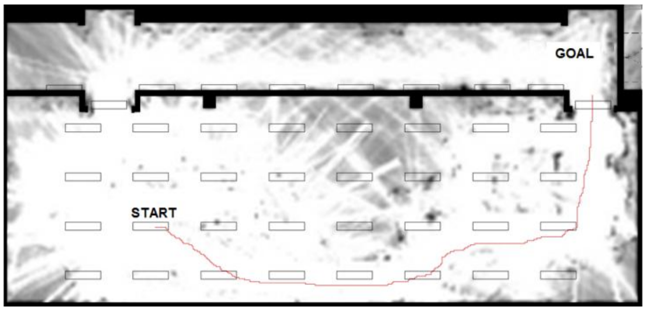

Fig. 9. Path determined by the Wavefront algorithm.

VI. ROS ENVIRONMENT

To carry out the implementation of these methodologies, we used the ROS [19] development environment from Willow Garage. As stated previously, it consists of a development environment which provides a set of libraries and tools that facilitate robotic applications, especially in mobile robotics. It includes a nearly total abstraction of the hardware, peripheral drivers, data viewers, messaging and package management. Its architecture contemplates the existence of nodes, consisting in applications with the ability to publish and subscribe to topics, which are a type of data bus for transmitting information via messages with predefined data structures.

For the communication with the hardware of the Pioneer P3-DX robot one used the *p2os* node [22], while the *uvc_camera* node [23] was used to access the webcam, both nodes were provided by the ROS community. The remaining software and algorithms were implemented in the *main_node* node. The representation of the full system structure is illustrated in Fig. 8, where the used nodes and topics are marked in red. This implementation allows using the onboard computer to perform all the computations, but also allows the use of external, or even multiple, computers in the future if needed, with only minimal changes in the configuration

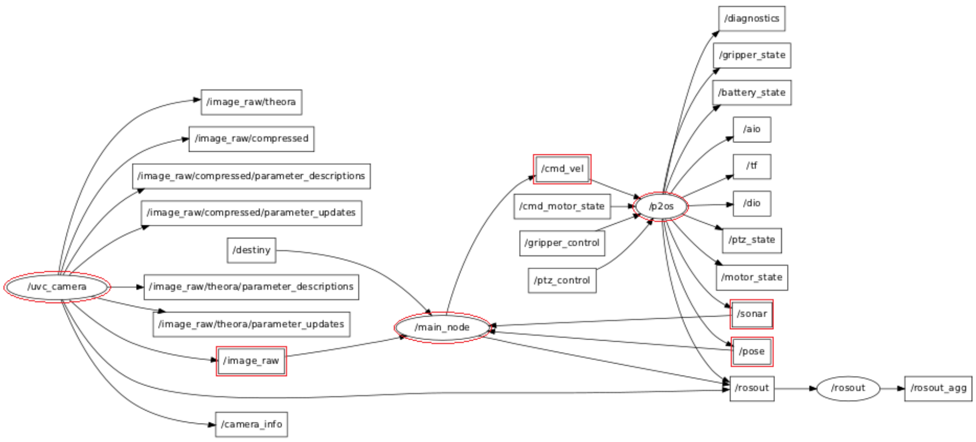

Fig. 8. Full ROS system structure.

VII. Conclusions and Future Work

A localization and navigation system was implemented to perform general tasks in office-like environments. This system maintains a correct estimate of the robot´s pose in real-time, being able to correct the error accumulation of intrinsic sensors and estimate its own pose if this information is not provided in advance.

Artificial vision is used with a common video camera for recognizing ceiling light fixture fixtures as external landmarks, whose poses are known a priori. However, given the similarity between the various fixtures and the symmetry of those poses, the determination between the real and the corresponding fixture detected is strongly dependent on the robot's estimated pose. This may generate erroneous matches, leading to ambiguity and getting reasonable confidence in a pose that may not be real. Also, the ambient lighting conditions can lead to false detections that can compromise severely the location and hence the mapping of the environment. Nevertheless, the continuous movement of the robot combined with the particle filter used for global localization and the Kalman filter localization update, given the localization estimate confidence, allows the robot to recover when lost, and resume its normal operation.

For determining a safe and short path an occupancy grid was kept and updated with information of the space occupation likelihood. This map is updated based on the distance information of ultrasound sensors and pose estimate when this estimate has sufficiently high confidence. Based on the information of this occupancy grid the path to follow is determined with the Wavefront algorithm.

Finally, while performing the determined path, this system reacts to unexpected obstacles such as people or objects left in previously free space. To avoid these obstacles one used the virtual forces method for navigation, resuming the path whenever possible.

To speed up localization it would be interesting to explore matching techniques directly from the camera images or contours with a representation of the light fixtures and other ceiling elements, as air vents and fire detectors. This avoids significant movement without detection of landmarks, resulting in faster convergence for the particle filter and more accuracy for the Kalman filter and mapping. Furthermore, it is also expected to be more immune to erroneous detections caused by reflections, foreign objects or anomalous lighting conditions. This is difficult to combine with a high number of particles because it has to be done for each particle, which leads to a large consumption of processing resources, unavailable on the Pioneer P3-DX robot standard platform.

As a follow up of the present work, one plans to use the developed work as a base for specific tasks, such as an autonomous guide or carrier robot.

References

[1] Ahmad, A., & Samad, A. (2010). Aerial mapping using high resolution digital camera and unmanned aerial vehicle for Geographical Information System. Signal Processing and Its Applications (CSPA), 2010 6th International Colloquium on .

[2] Angerer, A., Hoffmann, A., Schierl, A., Vistein, M., & Reif, W. (2010). The Robotics API: An object-oriented framework for modeling industrial robotics applications. International Conference on Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ.

[3] Arguenon, V., Bergues-Lagarde, A., Rosenberger, C., Bro, P., & Smari, W. (2006). Multi-Agent Based Prototyping of Agriculture Robots. International Symposium on Collaborative Technologies and Systems.

[4] Arulampalam, M. S., Maskell, S., & Gordon, N. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing.

[5] Baumann, M., Hunter, J., Hannum, D., & Horton, R. (2004). Robohound™ remote explosives detection. Conference on Security Technology. 38th Annual 2004 International Carnahan.

[6] Borenstein, J. (1991). The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots. IEEE Transactions on Robotics and Automation, Vol. 7, No.3.

[7] Bradsky, G. R.; Pisarevsky, V.; Bouguet, J. Learning OpenCV: Computer Vision with the OpenCV Library. Springer, 2006.

[8] Canny, J. (1986). A Computational Approach to Edge Detection. IEEE transactions on pattern analysis and machine intelligence, vol. Pami-8, no. 6.

[9] S. P. Lloyd, ªLeast Squares Quantization in PCM,º IEEE Trans. Information Theory, vol. 28, 129-137, 1982.

[10] Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraky, L. E., et al. (2004). Principles of Robot Motion – Theory, Algorithms and Implementations. MIT Press.

[11] Couto, M. S. (2010). Localização e Navegação entre Robôs Móveis. Dissertação para a obtenção do Grau de Mestre em Engenharia Mecânica - Instituto Superior Técnico.

[12] Dissanayake, M. W., Newman, P., Clark, S., Whyte, H., & Csorba, M. (2001). A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. IEEE Transactions on robotics and automation, Vol. 17, NO. 3, June 2001.

[13] Grotzinger, J. P., Ba, J. C., Blake, D. F., & all, e. (2011). Mars Science Laboratory Mission and Science. Space Sci Rev DOI 10.1007/s11214-012-9892-2.

[14] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Journal of Basic Engineering 82 (1), 35–45.

[15] Manocha, K., Pernalete, N., & Dubey, R. (2001). Variable position mapping based assistance in teleoperation for nuclear cleanup . IEEE International Conference on Robotics and Automation. Proceedings 2001 ICRA.

[16] Mobile Robots, A. (2010). Pioneer 3 Operations Manual.

[17] Neves, C. C. (1998). A Generalized Framework for the Analysis of System Architectures in Automonomous Robots. Telford Research Institute Department of Electronic and Electric Engineering University of Salford, UK.

[18] Panzieri, S., Pascuci, F., & Ulivi, G. (2003). Vision based navigation using Kalman approach for SLAM. Proc. of the 11th International Conference on Advanced Robotics.

[19] Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., et al. (2009). ROS: an open-source Robot Operating System. ICRA workshop.

[20] Rahimi, M., Pon, R., Kaiser, W., Sukhatme, G., Estrin, D., & Srivastava, M. (2004). Adaptive sampling for environmental robotics. IEEE International Conference on Robotics and Automation.

[21] Srinivasa, S. S., Ferguson, D., Helfrich, C. J., Berenson, D., Collet, A., Diankov, R., et al. (2009). HERB: a home exploring robotic butler. Springer Science+Business Media, LLC 2009.

[22] http://www.ros.org/wiki/p2os_driver

[23] http://www.ros.org/wiki/uvc_camera