NOVEL 3D MATCHING SELF-LOCALISATION ALGORITHM

Miguel Pinto, A. Paulo Moreira, Aníbal Matos, Héber Sobreira
INESC Porto - Institute for Systems and Computer Engineering of Porto,
Department of Electrical and Computer Engineering,
Faculty of Engineering, University of Porto, Porto, Portugal

ABSTRACT

A new and fast methodology is discussed as a solution to pinpointing the location of a robot called RobVigil, in a robust way, without environment preparation, even in dynamic scenarios. This solution does not require a high computational power. Firstly, the EKF-SLAM is used to find the location of the vehicle, allowing to map the surrounding area in the 3D space. Afterwards, the constructed map is used in a 3D matching algorithm during the normal operation of the vehicle to pinpoint its location. The 3D matching algorithm uses data from a tilting Laser Range Finder. Experimental results on the performance and accuracy of the proposed method are presented in this paper.

KEYWORDS: *Mobile Robot, Service Robots,3D Matching, Simultaneous Localisation and Mapping, Laser Range Finder*

I. INTRODUCTION

To be truly autonomous, a robot must be able to pinpoint its location inside dynamic environments, moving in an unlimited area, without preparation needs.

The application of localisation systems, which need the preparation of the indoor building and a certain setup time, in some situations, became impractical both in terms of aesthetic and functional. Furthermore, the associated costs cannot be considered negligible.

In order to fulfil this definition of autonomy, the fundamental motivation and opportunity of this work, is the implementation of a robust strategy of localisation that runs in a short execution time.

The developed approach is a *three-dimensional map based* localisation method, with the objective of solve the problem of accumulative error when the odometry is used, using the environment infrastructure, without constraints in terms of the navigation area and with no need of prepare the environment with artificial landmarks or beacons.

It is applied to the RobVigil robot, shown in Fig. 1. The main application of the RobVigil is the surveillance of public facilities, i.e. dynamic environments, as shopping malls, hospitals or others service scenarios. The use of a robot of this type, in these scenarios, allows systematic and uninterruptable inspection routines, with minimum human intervention.

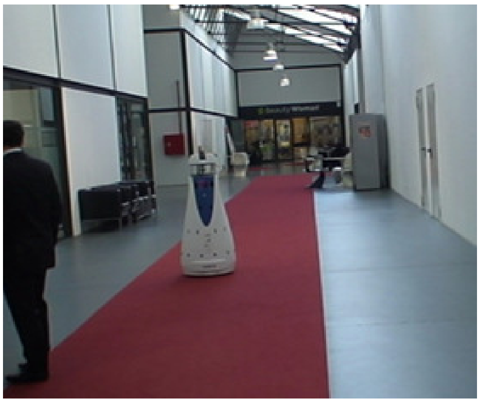

Fig. 1. The RobVigil performing a surveillance routine in a Shopping Mall.

This is a differential wheel traction vehicle, equipped with odometers and a tilting Laser Range Finder, which acquires three-dimensional data about the surrounding environment and is used for mapping and localisation.

To perform the surveillance task, the RobVigil is equipped with sensors to detect dangerous situations, like fires, floods or gas leaks. It is equipped as well with three surveillance cameras, including an omnidirectional, an High Resolution, and finally, a thermal camera.7

II. MANUSCRIPT ORGANIZATION

The paper organization can be described as the following: in the second section a literature review is done, while in third section, the strategy adopted for this localisation methodology is described. The localisation method is described in Section V. The experimental setup is described in the Section IX. The experimental results about the accuracy and performance of the localisation algorithm are shown in Section X. Two possible research topics referred as possible future works are presented at the Section XI. Finally, conclusions about the improvements and contributes made in this work are presented in the Section XII.

III. LITERATURE REVIEW

Different sensors and techniques for the localisation of vehicles are described in [1]. These sensors and techniques are divided into absolute and relative localisation.

The dead-reckoning are sensors of relative localisation leading to an increase of the error over time. Examples of dead-reckoning sensors are: odometry (the most commonly used), accelerometers, gyroscopes, inertial navigation sensors (INS) or inertial measurement units (IMU) and Doppler-effect sensors (DVL).

Due to their high frequency rate, the dead-reckoning sensors are commonly used to fuse with more complex localisation techniques or sensors, through probabilistic methods as is example the Kalman Filters and the Particle Filters, [2]. Examples include infrared sensors, ultrasound sonars, laser range finders, artificial vision and techniques based on passive or active beacons.

The sensors and techniques of absolute localisation give information about the robot's position in the world frame. Examples are the attitude sensors, digital compasses, GPS and passive or active beacons, as is example the acoustic beacons [1].

The two essential localisation techniques based on active or passive beacons are the: triangulation and trilateration, [3]. Unfortunately, this methods require environment preparation.

The algorithms concerning the localisation of mobile robots can be divided in two large areas: the matching algorithms and the Simultaneous Localisation and Mapping algorithms (SLAM).

There are matching algorithms that need a prior knowledge about the navigation area, as is example the works [4] and [5]. Another example is the Perfect Match described by M.Lauren *et al.* [6], used in the Robotic Soccer Middle Size League (MSL) at RoboCup. The Perfect Match is a time saver algorithm.

There are other type of matching algorithms, which compute the overlapping zone between consecutive observations, to obtain the vehicle displacement. Examples are the family of Iterative Closest Point algorithms (ICP) [7].

In addition to the computational time spent, the ICP approach has another problem, sometimes there is no sufficient overlapping between two consecutive laser scans and it is hard to find a correct solution.

The most common solutions for the SLAM problem are: the Extended Kalman Filter applied (EKF-SLAM) as are examples the works [8] to [10], and the FastSlam or the Rao-Blackwellized Particle Filters, as is example the works [11] to [13].

The EKF-SLAM is a variant of the Extended Kalman Filter and uses only one state matrix representing the vehicle state and the landmarks of the feature map. This state matrix is increased ever that a new feature is found. In contrary, the FastSlam solution can be seen as a robot and a collection of N landmarks estimation problems. Each particle has its pose estimative and tiny state matrices representing each landmarks of the feature map.

The EKF-SLAM computationally complexity is$O(N^2)$, while the FastSlam has a lower computational complexity,$O(M \log N)$, with M particles and where N landmarks.

IV. Strategy of Localisation

The localistation methodology is divided in the following steps:

1) *Pre-Localisation* and *mapping*: locating the vehicle, using a two-dimensional SLAM algorithm. Once the location is obtained, it is possible to build and store the 3D map of the environment. This procedure is performed in a controlled scenario, without dynamic objects or people moving in the navigation area. It is a preparation task, performed offline and executed only once. The aim is to obtain a 3D map of the facility.

Regarding the *pre-Localisation* and *mapping* step, once the SLAM has been applied, the 2D feature map with segments and points is obtained. The same map is after used to determine the vehicle's location (*pre-localisation* procedure). The SLAM solution is based in the state of art EKF-SLAM algorithms, as the described in [2].

Finally, still offline and with the vehicle's position obtained during the *pre-localisation*, the three-dimensional map can be built and the respective distance and gradient matrices can be created and stored (*mapping* procedure)

The stored distance and gradient matrices are used as look-up tables for the 3D Matching *localisation* procedure, in the normal vehicle operation, as described in the next section.

To create the distance matrix, the distance transform is applied in the 3D space, on the occupancy grid of the building. Furthermore, the Sobel filter, again in the 3D space, is applied to obtain the gradient matrices, in both the directions x and y.

2) *Localisation* (section V): the stored 3D map makes it possible to pinpoint the location of the vehicle by comparing the 3D map and the observation module's readings, during the normal vehicle operation (3D Matching).

The used data, corresponds to 3D points acquired by the observation module (tilting LRF) on the upper side of the indoor environment, which can be considered almost static (without dynamic objects moving). The upper side/headroom of a building remains static during large periods of time, and doesn't suffer perturbations of people and dynamic objects crossing the navigation area.

As the localisation algorithm is applied in the RobVigil, which navigates in public facilities, with people an dynamic objects crossing the navigation area, only data and the map about the headroom of the building is used, aiming to improve the methodology accuracy and robustness.

Videos and downloads about this localisation methodology can be found at [14].

V. 3D Matching Localisation Algorithm

The light computational Perfect Match algorithm, described in [6] by M.Lauren *et al.*, was adapted from 2D to the 3D space, using Laser Range Finder data, maintaining the low computational requirements.

The 3D matching localisation procedure uses the result of the 3D Perfect Match in a position tracking algorithm, which has three fundamental stages: 1) the Kalman Filter Prediction, 2) the 3D Perfect Match procedure, and 3) the Kalman Filter Update. The Kalman Filter equations can be seen in [2].

VI. Kalman Filter Prediction

The Kalman Filter Prediction stage takes the previously estimated vehicle state $(X_v(k))$ and, using odometry, it estimates the next vehicle state$(X_{Odo}(k+1))$. The previous vehicle state $X_v(k))$ is equal to the following vector:

$$X_v(k) = [x_v(k) \quad y_v(k) \quad \theta_v(k)]^T \tag{1}$$

where the variables x_v, y_v and θ_v (2D coordinates and orientation relatively to the x direction).
Aiming to model the vehicle's kinematic movement, the following transition function $f_v(X_v(k), Odo, q_v)$ was used:

$$f_v(X_v, d, \Delta\theta, q_v) = \begin{bmatrix} x_v(k) + d \cdot cos\left(\theta_v(k) + \frac{\Delta\theta}{2}\right) + \varepsilon_{qx} \\ y_v(k) + d \cdot sin\left(\theta_v(k) + \frac{\Delta\theta}{2}\right) + \varepsilon_{qy} \\ \theta_v(k) + \Delta\theta + \varepsilon_{q\theta} \end{bmatrix} \tag{2}$$

where d and $\Delta\theta$ represent the linear and angular displacement respectively between two consecutive time steps $(k+1)$ and (k). The kinematic model's error q_v was modelled as additive Gaussian noise with zero mean and co-variance Q_v. The vector q_v is represented as the following :

$$q_v = [\varepsilon_{qx} \quad \varepsilon_{qy} \quad \varepsilon_{q\theta}]^T \tag{3}$$

Therefore, the new vehicle state appears equal to the following equation:

$$X_{Odo}(k+1) = f_v(X_v(k), d, \Delta\theta, \hat{q}_v) \tag{4}$$

where $\hat{q}_v = 0$, since q_v is modelled as Gaussian noise with zero mean. Therefore, the new estimated state is given by:

$$X_{Odo}(k+1) = \begin{bmatrix} x_v(k) + d \cdot cos\left(\theta_v(k) + \frac{\Delta\theta}{2}\right) \\ y_v(k) + d \cdot sin\left(\theta_v(k) + \frac{\Delta\theta}{2}\right) \\ \theta_v(k) + \Delta\theta \end{bmatrix} \tag{5}$$

The Kalman prediction stage also uses the previous estimated covariance matrix $(P_v(k))$ and computes the next $(P_{Odo}(k+1))$. The estimated covariance after the prediction stage is given by the equation:

$$P_{Odo}(k+1) = \frac{\partial f_v}{\partial X_v} \cdot P_v(k) \cdot \frac{\partial f_v}{\partial X_v}^T + \frac{\partial f_v}{\partial q_v} \cdot Q_v \cdot \frac{\partial f_v}{\partial q_v}^T \tag{6}$$

where, the gradient of the model transition function $f_v(X_v, Odo, q_v)$, represented by $\frac{\partial f_v}{\partial X_v}$, is equal to:

$$\frac{\partial f_v}{\partial X_v} = \begin{bmatrix} 1 & 0 & -d \cdot sin\left(\theta_v(k) + \frac{\Delta\theta}{2}\right) \\ 0 & 1 & d \cdot cos\left(\theta_v(k) + \frac{\Delta\theta}{2}\right) \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

The gradient of the transition function f_v in order to the noise q_v, equal to $\frac{\partial f_v}{\partial q_v}$, is the identity matrix. The covariance Q_v depends (increase) with the vehicle's translation and rotation (d and $\Delta\theta$):

$$Q_v = \begin{bmatrix} \left[d \cdot cos\left(\theta_v(k) + \frac{\Delta\theta}{2}\right) \cdot \sigma_d\right]^2 & 0 & 0 \\ 0 & \left[d \cdot sin\left(\theta_v(k) + \frac{\Delta\theta}{2}\right) \cdot \sigma_d\right]^2 & 0 \\ 0 & 0 & (d \cdot \sigma_{d\theta})^2 + (\Delta\theta \cdot \sigma_\theta)^2 \end{bmatrix} \tag{8}$$

When the vehicle moved 1 meter in front, the odometry, accumulate a translational and rotational error equal to σ_d and $\sigma_{d\theta}$, respectively. When the vehicle rotates 1 radian, the odometry error of the vehicle's rotation is equal to σ_θ. The parameters were obtained measuring the difference between the real and estimated, with odometry, displacement and rotation of the vehicle, during 40 samples. The obtained values were: σ_d =0.18264 meters/meters, $\sigma_{d\theta}$ =0.08961radians/meters and σ_θ =0.02819 radians/radians.

VII. PERFECT MATCH

Consider the list of points that constitute the scan of the laser range finder, in the laser frame ($PntList^L$). Each point $PntList(i)^L$ has coordinates in the laser referential equal to $\left(x_i^L, y_i^L, z_i^L\right)^T$. The correspondence of this point in the world frame, results in the point PntList(i), with coordinates $(x_i, y_i, z_i)^T$. This point can be computed with the following expression:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} x_v \\ y_v \\ \theta_v \end{bmatrix} + R_L^W \cdot \begin{bmatrix} x_i^L \\ y_i^L \\ z_i^L \end{bmatrix}, R_L^W = \begin{bmatrix} \cos\theta_v & -\sin\theta_v & 0 \\ \sin\theta_v & \cos\theta_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

where the R_L^W matrix is the rotation of the vehicle in relation to the world referential:

The developed 3D Perfect Match takes the vehicle's state obtained in the Kalman Filter Prediction step ($X_{Odo}(k+1)$) and runs the following steps: 1) matching error; 2) optimisation routine Resilient Back-Propagation (RPROP); and 3) second derivative.

The first two steps are continuously iterated until the maximum number of iterations is reached. The last and third step is performed only once after obtained the RPROP's result.

The distance matrix, stored in memory, is used to compute the matching error. The matching error is computed through the cost value of the list of points of the Laser Range Finder scan ($PntList$):

$$E = \sum_{i=1}^{N} E_i, \qquad E_i = 1 - \frac{L_c^2}{L_c^2 + d_i^2} \tag{10}$$

where d_i and E_i are representative of the distance matrix and cost function values, which correspond to the point $PntList(i)$, with coordinates $(x_i, y_i, z_i)^T$. N is the number of points in the list of three-dimensional points $PntList$. The parameter L_c is an adjustable parameter. Aiming to have a cost function obeying to the condition $0.5 \le E_i \le 1$, when the distance of the point $PntList(i)$ is equal or higher than 1 meters, $d_i \ge 1\ meter$, the value of L_c, used in the work, was 1 meter.

The cost function E_i was designed to achieve a similar behaviour with the quadratic error function for the case of small values of d_i , neglecting points with large values of d_i.

After computing the matching error, it is possible to apply the RPROP algorithm to each vehicle state variable. The algorithm takes the previous state of the vehicle $X_{Match}(n-1)$ and estimates a new position for the vehicle $X_{Match}(n)$, which is used in the next RPROP iteration.

The initial state of the vehicle to be used in the RPROP algorithm is given by the Kalman Filter Prediction stage, $X_{Match}(0) = X_{Odo}(k+1)$.

The distance and gradients matrices (∇x and the ∇y), stored in memory, are used to compute the cost function derivatives in order to the vehicle state, used in the RPROP algorithm.

The RPROP routine can be described as the following: during a number limit of iterations, the next steps are performed to each variable that is intended to estimate, x_v, y_v and θ_v.

1) If the actual derivative $\frac{\partial E}{\partial x_v}(t)$, $\frac{\partial E}{\partial y_v}(t)$ and $\frac{\partial E}{\partial \theta_v}(t)$, depending on the variable, is different of zero, they are compared with the previous derivatives, $\frac{\partial E}{\partial x_v}(t-1)$, $\frac{\partial E}{\partial y_v}(t-1)$ and $\frac{\partial E}{\partial \theta_v}(t-1)$.

2) If the product $\frac{\partial E}{\partial X_v}(t) \cdot \frac{\partial E}{\partial X_v}(t-1)$ is lower than zero, it means that the algorithm already passes a local minima, and then the direction of the convergence need to be inverted.

3) If the product $\frac{\partial E}{\partial X_v}(t) \cdot \frac{\partial E}{\partial X_v}(t-1)$ is higher than zero, it means that the algorithm continues to converge to the local minima, and then the direction of the convergence should be maintained with the same value.

The RPROP algorithm can be seen with more detail in the following pseudo-code.

```
                    $\boldsymbol{RPROP\left(\frac{\partial E}{\partial X_v}(n), \frac{\partial E}{\partial X_v}(n-1), X_{Match}(n-1)\right)}$
$\boldsymbol{for}$ $each\ state\ varible\ x_v, y_v\ and\ \theta_v$ $\boldsymbol{do}$
    $\boldsymbol{if}\left(\frac{\partial E}{\partial X_v}(n) \neq 0\right)$ $\boldsymbol{then}$
        $\boldsymbol{if}\left(\frac{\partial E}{\partial X_v}(n) \cdot \frac{\partial E}{\partial X_v}(n-1) > 0\right)$ $\boldsymbol{then}$
            $\lambda(n) \leftarrow \lambda(n-1) \cdot \eta^+$
        $\boldsymbol{end\ if}$
        $\boldsymbol{if}\left(\frac{\partial E}{\partial X_v}(n) \cdot \frac{\partial E}{\partial X_v}(n-1) < 0\right)$ $\boldsymbol{then}$
            $\lambda(n) \leftarrow \lambda(n-1) \cdot \eta^-$
        $\boldsymbol{end\ if}$
        $\boldsymbol{if}\left(\frac{\partial E}{\partial X_v}(n) > 0\right)$ $\boldsymbol{then}$
            $X_{Math}(n) \leftarrow X_{Math}(n-1) - \lambda(n)$
        $\boldsymbol{end\ if}$
        $\boldsymbol{if}\left(\frac{\partial E}{\partial X_v}(n) < 0\right)$ $\boldsymbol{then}$
            $X_{Math}(n) \leftarrow X_{Math}(n-1) + \lambda(n)$
        $\boldsymbol{end\ if}$
    $\boldsymbol{end\ if}$
$\boldsymbol{end\ for}$
```

The values $\eta_{x_v}^+$, $\eta_{y_v}^+$, $\eta_{\theta_v}^+$ and $\eta_{x_v}^-$, $\eta_{y_v}^-$, $\eta_{\theta_v}^-$are empirical and adjustable values. They are tested in the intervals $\eta_{x_v}^+$, $\eta_{y_v}^+$, $\eta_{\theta_v}^+ \ \epsilon[1,2]$ and $\eta_{x_v}^-$, $\eta_{y_v}^-$, $\eta_{\theta_v}^- \ \epsilon]0,1[$. The best performance was achieved to $\eta_{x_v}^+ = \eta_{y_v}^+ = \eta_{\theta_v}^+ = \eta^+ = 1.2$, and $\eta_{x_v}^- = \eta_{y_v}^- = \eta_{\theta_v}^- = \eta^- = 0.5$.

The initial value of the variables steps $\lambda_{x_v}(0), \lambda_{y_v}(0), \lambda_{\theta_v}(0)$ are also empirical and adjustable parameters. The best performance was achieved to $\lambda_{x_v}(0) = \lambda_{y_v}(0) = 0.01$ meters and $\lambda_{\theta_v}(0) = 0.05$ radians.

The limitation of the number of iterations in the RPROP routine makes it possible to guarantee a maximum time of execution for the algorithm. Therefore, as it is intended to operate online, it is necessary to ensure that this maximum time is lower than the observation module sample rate (100 milliseconds).

The following experiment was conducted aiming to obtain the maximum number of iterations: the 3D Perfect Match algorithm was applied with the knowledge of the true vehicle's position. For the large majority of the cases, the estimated position reached the true position and was achieved in less than 8 iterations. On the contrary, in the few cases where the solution not reached the real position, it was close enough to be achieved in the following cycle (next observation module time step). That way, the number of maximum iterations used in the RPROP routine was ten iterations.

The gradient $\frac{\partial E}{\partial X_v}$ in order to the state X_v is given by the following expression:

$$\frac{\partial E}{\partial X_v} = \sum_{i=1}^{PntList.Count} \frac{\partial E_i}{\partial X_v}, \qquad \frac{\partial E_i}{\partial X_v} = \frac{2L_c^2 \cdot d_i}{\left(L_c^2 + d_i^2\right)^2} \cdot \frac{\partial d_i}{\partial X_v} \tag{11}$$

where $\frac{\partial E_i}{\partial X_v}$ is the gradient of the cost function, of each point i, in order to the vehicle stat. The partial derivatives, $\frac{\partial d_i}{\partial X_v} = \left(\frac{\partial d_i}{\partial x_v}, \frac{\partial d_i}{\partial y_v}, \frac{\partial d_i}{\partial \theta_v}\right)$, are given by the following vector:

$$\frac{\partial d_i}{X_v} = \left(\frac{\partial d_i}{\partial x_i} \cdot \frac{\partial x_i}{\partial x_v}; \frac{\partial d_i}{\partial x_i} \cdot \frac{\partial y_i}{\partial y_v}; \frac{\partial d_i}{\partial x_i} \cdot \frac{\partial x_i}{\partial \theta_v} + \frac{\partial d_i}{\partial y_i} \cdot \frac{\partial y_i}{\partial \theta_v}\right) \tag{12}$$

Using the equations presented at (9), the vector (12) can be re-written as the following expressions:

$$\frac{\partial d_i}{X_v} = \left(\nabla x(x_i, y_i, z_i\ ); \nabla y(x_i, y_i, z_i); \begin{bmatrix}\nabla x(x_i, y_i, z_i\ ) \\ \nabla y(x_i, y_i, z_i)\end{bmatrix}^T \begin{bmatrix}-\sin\theta_v & -\cos\theta_v \\ \cos\theta_v & -\sin\theta_v\end{bmatrix}\begin{bmatrix}x_i \\ y_i\end{bmatrix}\right) \tag{13}$$

where $\nabla x(x_i, y_i, z_i\ )$ and $\nabla y(x_i, y_i, z_i)$ are the gradient values at the position (x_i, y_i, z_i), of the pre-computed gradient matrices, stored in memory, in x and y direction, respectively.

To completely perform the 3D Perfect Match stage, it is necessary to calculate the second derivative. The analysis of the second derivative allows to find an error classification for the 3D Perfect Match solution.

For an actual set of 3D LRF data, if the cost function E has a perfect distinctive minimum, the second derivative, $\frac{\partial^2 \mathrm{E}}{\partial \mathrm{X}_\mathrm{v}^2}$, is high. In contrary, when for those points the cost function E is flat, and there are no a distinctive minimum, the second derivative is low. Therefore, a higher second derivative represents situations where the confidence in the solution obtained by the 3D Perfect Match algorithm is higher. In the other hand, a lower second derivative functions represent cases where the confidence is lower.

The 3D Perfect Match's covariance matrix represent the error, i.e. the confidence, that there are in the solution obtained by algorithm:

$$P_{Match} = diag\left(K_{XY}/\frac{\partial^2 E}{\partial x_v^2}, K_{XY}/\frac{\partial^2 E}{\partial y_v^2}, K_\theta/\frac{\partial^2 E}{\partial \theta_v^2}\right) \tag{14}$$

where diag(.,.,.) is the diagonal matrix 3x3.

The parameters $\mathrm{K_{XY}}$ and $\mathrm{K_\theta}$, are normalized values. The algorithm was tested with values of $\mathrm{K_{XY}}$ and $\mathrm{K_\theta}$ in the gap between $[10^{-6},\ 10^{-1}]$. The best performance achieved was $\mathrm{K_{XY}} = 10^{-3}$ and $\mathrm{K_\theta} = 10^{-3}$.

To compute the second derivative, the previously defined cost function equation (10) is replaced by the quadratic cost function (15). This occurs in order to ascertain which cost function is positively definite for all laser scan points. The cost function is given by:

$$E = \sum_{i=1}^{N} E_i, \qquad E_i = \frac{1}{2}\cdot\left(\frac{d_i}{L_c}\right)^2 \tag{15}$$

The second derivative of the total cost function are given by:

$$\frac{\partial^2 E}{\partial X_v^2} = \sum_{i=1}^{N}\frac{\partial^2 E_i}{\partial X_v^2}, \qquad \frac{\partial^2 E_i}{\partial X_v^2} = \frac{1}{L_c^2}\cdot\frac{\partial^2 d_i}{\partial X_v^2} \tag{16}$$

where $\frac{\partial^2 d_i}{\partial X_v^2} = \left(\frac{\partial^2 d_i}{\partial x_v^2}, \frac{\partial^2 d_i}{\partial y_v^2}, \frac{\partial^2 d_i}{\partial \theta_v^2}\right)$ is equal to the following vector:

$$\frac{\partial^2 d_i}{\partial X_v^2} = \left(\nabla x^2(x_i, y_i, z_i\ ); \nabla y^2(x_i, y_i, z_i\ ); \left[\left(\frac{\partial d_i}{\partial \theta_v}\right)^2 + d_i\cdot\frac{\partial^2 d_i}{\partial \theta_v^2}\right]\right) \tag{17}$$

$$\frac{\partial^2 d_i}{\partial \theta_v} = \begin{bmatrix}\nabla x(x_i, y_i, z_i\ ) \\ \nabla y(x_i, y_i, z_i\ )\end{bmatrix}^T \begin{bmatrix}-\cos\theta_v & \sin\theta_v \\ -\sin\theta_v & -\cos\theta_v\end{bmatrix}\begin{bmatrix}x_i \\ y_i\end{bmatrix} \tag{18}$$

$$\begin{bmatrix}\nabla x(x_i, y_i, z_i\ ) \\ \nabla y(x_i, y_i, z_i)\end{bmatrix}^T \begin{bmatrix}-\sin\theta_v & -\cos\theta_v \\ \cos\theta_v & -\sin\theta_v\end{bmatrix}\begin{bmatrix}x_i \\ y_i\end{bmatrix} \tag{19}$$

VIII. KALMAN FILTER UPDATE

The Kalman filter update stage combines the estimated state using the odometry $X_{Odo}(k+1)$, and the Perfect Match procedure $X_{Match}(k+1)$. The Kalman Filter equations can be seen in [2]

The observation model $h_v\ (X_v, r)$ in the update stage is equal to the vehicle state X_v:

$$h_v\ (X_v, r) = \begin{bmatrix} x_v + e_{rx} \\ y_v + e_{ry} \\ \theta_v + \varepsilon_{r\theta} \end{bmatrix} \tag{20}$$

where r is modelled as white noise, with a Gaussian distribution with zero mean ($\hat{r} = 0$) and covariance matrix R.

Therefore, in the update stage the observation is equal to the state obtained after the application of the 3D Perfect Match:

$$Z_v = X_{Match}(k+1) \tag{21}$$

The estimated observation is equal to the present estimative of the vehicle state, propagated during the Kalman Filter Prediction stage:

$$\hat{h}_v\ (X_v, \hat{r}) = X_{Odo}(k+1) \tag{22}$$

In that way, the innovation of the Kalman filter ($V(k+1)$) is equal to:

$$V(k+1) = Z_v - \hat{h}_v\ (X_v, \hat{r}) \tag{23}$$

In this stage, the propagated covariance matrix, using odometry ($P_{Odo}(k+1)$), and the covariance matrix computed in the 3D Perfect Match procedure ($P_{Match}(k+1)$), are used to determine the Kalman Filter gain ($W(k+1)$):

$$W(k+1) = P_{Odo}(k+1)\frac{\partial h_v}{\partial X_v}^T \left[\frac{\partial h_v}{\partial X_v} P_{Odo}(k+1)\frac{\partial h_v}{\partial X_v}^T + \frac{\partial h_v}{\partial r} P_{Match}(k+1)\frac{\partial h_v}{\partial r}^T\right]^{-1} \tag{24}$$

The gradient of the observation module, in order to the vehicle state and the observation noise, $\frac{\partial h_v}{\partial X_v}$ and $\frac{\partial h_v}{\partial r}$ respectively, are identity matrices. Therefore, the previous equation can be re-written as the following:

$$W(k+1) = P_{Odo}(k+1)[P_{Odo}(k+1) + P_{Match}(k+1)]^{-1} \tag{25}$$

Therefore, after the update stage the new estimated state ($X_v(k+1)$), is given by the expression:

$$X_v(k+1) = X_{Odo}(k+1) + W(k+1) \cdot V(k+1), \tag{26}$$

The new covariance matrix, decreases with the following equation:

$$P_v(k+1) = [I^{3x3} - W(k+1)] \cdot P_{Odo}(k+1) \tag{27}$$

where I^{3x3} is the square identity matrix with the dimension 3x3.

IX. EXPERIMENTAL SETUP

The RobVigil is a differential wheel traction vehicle, equipped with odometers and a tilting Laser Range Finder.

The Laser Range Finder (LRF) – Hokuyo URG-04LX-UG01 – was used to perceive the environment. To obtain a three-dimensional sensor, a tilting platform was created based on the dc servo motor, the AX-12 Dynamixel Bioloid. The complete LRF solution is shown in Fig. 1 (right image).

The tilting Laser Range Finder is used as a sensor for the environment mapping and self-localisation. The AX-12 motor allows the LRF to rotate between the angles of 35º and 90º (angles between LRF's plan and the horizontal), with a resolution of 0.29°. The LRF's has a range angle of 240°, with an angular resolution of 0.35°. The LRF also has a distance range of 5 meters. A scan of 769 points is obtained at every 100 milliseconds. In this experiment the tilting LRF rotates at a speed of 10 rpm.
The RobVigil moves in this experience at an average speed of 0.4 m/s. The robot is shown at Fig. 2 on the right, while the developed observation module solution.

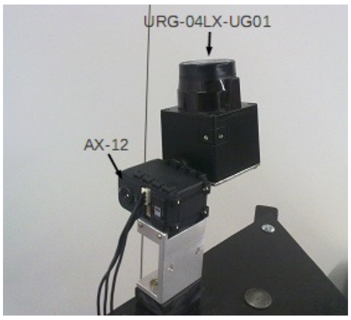

Fig. 2. Image on the left: the observation module developed. Image on the right: the RobVigil robot equipped with the observation module.

X. Experimental Results

The Fig. shows a large indoor environment. The mapping of this environment was obtained with success applying the strategy described for *pre-localisation* and *mapping* at Section IV.
In the figures of Fig. , the red points represent the upper side of the building (above 1.8 meters of height), almost static, used to perform the 3D Matching.

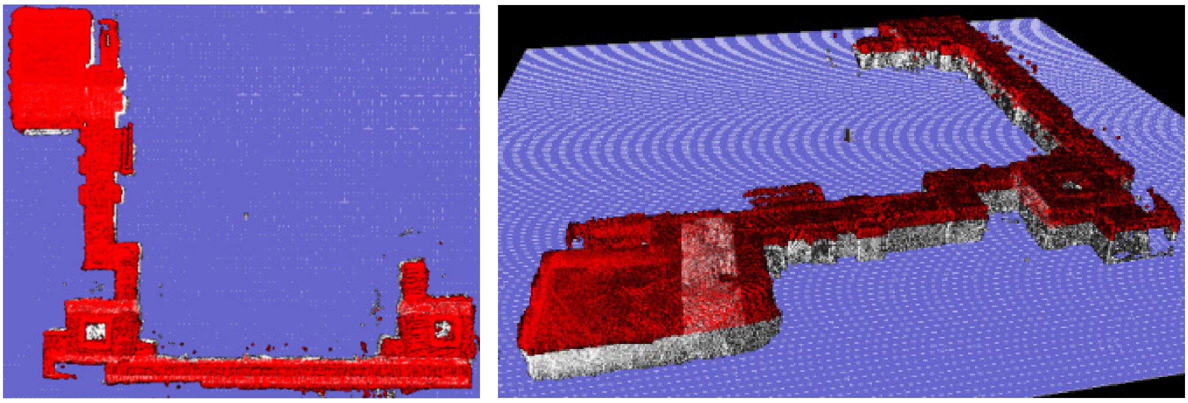

Fig. 3 Occupancy grid of the a scenario, with a square shape of 60 x 60 meters.

Aiming to evaluate the accuracy of the 3D matching localisation algorithm presented in this paper, at Section V, experiments were made with the high precision Ground Truth (GND) Positioning system, the SICK NAV 350, a commercial solution for autonomous guided vehicles (AGVs).
The SICK Nav350 uses reflectors to output its self location, with a sample rate of 125 milliseconds and an accuracy of 4 millimetres to 25 millimetres.
The Fig. shows the trajectory in a corridor of the mapped building Fig. . Only in this part of the building it is available the GND system. Therefore, only in corridor are shown accuracy results.

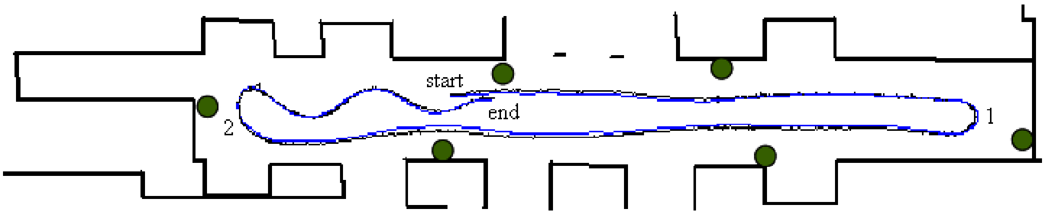

Fig. 4 The black vehicle position is the NAV350's estimated location. The blue vehicle trajectory 3D Matching estimative. The green circles are the reflectors placed in the scenario. The corridor has 25 x 8 meters.

In the figure, the circles are the reflectors used by the Nav350, the black line is the true robot trajectory, while the blue line is the trajectory estimated by the localisation algorithm (3D matching).

The Euclidian distance average between the 3D matching and the Nav350 position is 0.08 meters, with a standard deviation of 0.086 meters. The absolute average orientation difference is 1.18º, with a standard deviation of 1.72º. The reached accuracy is the sufficient and the desired, to perform the RobVigil application (surveillance of public facilities).

As the number of points acquired by the LRF is limited to 769 points, the maximum time spent in the localisation algorithm is also limited. The maximum time spent is 20 milliseconds and is lower than the sample rate imposed by the observation module (100 milliseconds), which allows the algorithm to be used online, with three-dimensional data, in the Mini ITX, EPIA M10000G, processor of 1.0GHz.

In this three-dimensional map based approach the time saver algorithm, Perfect Match described by M. Lauer *et al.* [6], was adapted to operate in the three-dimensional space, with Laser Range Finder data, instead the use of artificial vision. The computational complexity and needs of the Perfect Match was maintained, even using three-dimensional.

Since the computation time spent is not compromised, the experiment conducted by *M. Lauer et al* [6], which elects the Perfect Match as a faster algorithm when compared with the Particle Filter algorithm, remains valid. In this experiment, while the Perfect Match has a spent time of 4.2 milliseconds, the Particle Filter, using 200 and 500 particles, spends 17.9 milliseconds (four times higher) and 48.3 milliseconds (ten times higher), respectively.

Furthermore, comparing with the localisation method described by *M. Lauer et al* [6], the *localisation* procedure of this three-dimensional map based approach, was improved. It was applied the Extended Kalman Filter as multi fusion sensor system, aiming to join the odometry information and the three-dimensional Perfect Match result.

Comparing the localisation algorithm proposed and described in this paper with the ICP algorithms, it is faster and can be applied online in smaller cycle times, even when using a bigger quantity of Laser Range Finder points.

The MbICP algorithm described in [7], which already shows improvements to the standard ICP, takes an average time of 76 milliseconds to find the optimal solution, using only 361 Laser Range Finder points, in a sample rate of 200 milliseconds, in a Pentium IV 1.8GHz.

Furthermore, the 3D matching algorithm, proposed in this work, has a limited time of execution, depending on the number of the used points. In the ICP algorithms, the point to point match step is necessary to obtain the correspondence with the previous scan. Such step is not limited in the execution time and is widely dependent on the overlapping zone between consecutive scans. This overlapping zone influences as well the quality of the reached solution.

XI. Future Work

The 3D matching algorithm needs an initialization of the vehicle's initial pose. This initialization needs to be close enough to the true vehicle location, allowing after, the correct operation of the 3D matching algorithm.

At this moment, this initialization is passed as parameter of the 3D Matching algorithm, and therefore, it is not executed in an autonomous way.

As future work it is intended to implement a Initial Position Computation algorithm to initialize the 3D matching in an autonomous way.

The tilting Laser Range Finder has a set of parameters that are used to transform the 2D points of the LRF in 3D points about the surrounding environment. These parameters are about the translation and rotation of the tilting LRF, relatively to the vehicle frame. They are, at the moment, carefully measured. In the future, it is intended to develop a methodology of calibration capable to be executed autonomously.

XII. Conclusions

The three-dimensional map based localisation algorithm presented here, improves computational requirement comparatively to 2D and 3D SLAM algorithms. Furthermore, the time necessary to locate the robot is also reduced comparatively to Particle Filters and ICP algorithms. The approach described in this paper allows the localisation algorithm to be executed online.

The contributes made in this work were: i) adaptation of a light computational matching algorithm, the Perfect Match, to be used in the 3D space instead 2D, using Laser Range Finder data, maintaining the low computational requirements. ii) Improvement of the fusion system between the matching algorithm described in [6] by M.Lauren et al. and odometry data, using an EKF. iii) Only 3D data about the upper side of a building (almost a static scenario) is used, becoming the localisation more robust and reliable, since in dynamic scenarios. The use of three-dimensional data about the upper side of a building, increases the quantity and quality of the information, especially because it is almost static. iv) The localisation methodology can be used with any observation module, which acquires 3D data: Kinect, 3D camera (MESA), stereo vision or commercial 3D LRF. v) Development of a localisation methodology that becomes the robot RobVigil an economically practicable robotic platform.

Acknowledgements

This work is funded (or part-funded) by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project «FCOMP - 01-0124-FEDER-022701».

Miguel Pinto acknowledges FCT for his PhD grant (SFRH/BD/60630/2009).

References

[1]. J. Borenstein, H. R. Everett, L.Feng and D. Wehe, "Mobile Robot Positioning and Sensors and Techniques", Journal of Robotic Systems, Special Issue on Mobile Robots, Vol. 14 No. 4, pp. 231-249, April 1997.

[2]. S. Thrun, S. , & Burgard, W., & Fox, D. (2005). Probabilistic Robotics. Cambridge, Massachusetts: The MIT Press.

[3]. Héber Sobreira, A. Paulo Moreira and João Sena Esteves, "Characterization of Position and Orientation Measurement Uncertainties in a Low-Cost Mobile Platform", 9th Portuguese Conference on Automatic Control, Controlo 2010, Coimbra, Portugal, pp. 635-640, 8-10 September, 2010.

[4]. A. Sousa, P. Moreira and P. Costa, "Multi Hypotheses Navigation for Indoor Cleaning Robots", 3rd International Workshop on Intelligent Robotics (IRobot 2008), pp. 71-82, Lisbon, Portugal, October, 2008.

[5]. M. Pinto, A. P. Moreira and A. Matos, "Localization of Mobile Robots Using an Extended Kalman Filter in a LEGO NXT", IEEE Transactions On Education, Vol 55, No 1, pp. 135-144, February 2012.

[6]. M. Lauer, S. Lange and M. Riedmiller, "Calculating the perfect match: an efficient and accurate approach for robot self-localization", RoboCup Symposium, pp. 142-53, Osaka, Japan, 13-19 July, 2005.

[7]. J. Minguez, F. Lamiraux, and L. Montesano, "Metric-based scan matching algorithms for mobile robot displacement estimation", IEEE Transactions On Robotics, Vol 22, No 5, pp. 1047-1054, October 2006.

[8]. CSorba, M., (1997), Simultaneous Localisation and Map Building, Thesis, (PhD), Robotic Research Group Department of Engineering of Oxford, Oxford, England.

[9]. Andrea Garulli, Antonio Giannitrapani, Andrea Rossi, Antonio Vicino, "Mobile robot SLAM for line-based environment representation", 44th IEEE Conference on Decision and Control Decision and Control, 2005 and 2005 European Control Conference, (CDC-ECC '05), pp. 2041 - 2046, Seville, Spain, 12-15 Dec. 2005.

[10]. L. Teslić, I. Škrjanc and G. Klančar, "Using a LRF sensor in the Kalman-filtering-based localization of a mobile robot", ISA Transactions (Elsevier), Vol. 49, No. 1, pp. 145-153, January 2010.

[11]. Thrun and D. Fox, "A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping". Best Conference Paper Award, IEEE International Conference on Robotics and Automation. San Francisco, Vol. 1, pp 321-328, April 2000.

[12]. D. Hähnel, W. Burgard, and S. Thrun. "Learning compact 3D models of indoor and outdoor environments with a mobile robot". Robotics and Autonomous Systems (Elsevier), Vol. 44, No. 1, pp. 15-27, July 2003.

[13]. G. Grisetti , C. Stachniss and W. Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters", IEEE Transactions on Robotics, Vol. 23, No. 1, pp. 34-46, February 2007.

[14]. Webpage of "SLAM for 3D Map building to be used in a Matching 3D algorithm", available at: "www.fe.up.pt/~dee09013", June, 2012.

AUTHORS

Miguel Pinto graduated with a M.Sc. degree in Electrical Engineering from the University of Porto, Portugal in 2009. Since 2009, he has been a Ph.D. student at this department, developing his research within the Robotic and Intelligent Systems Unit of INESC Porto (the Institute for Systems and Computer Engineering of Porto). His main research areas are in Process Control and Robotics, navigation and localisation of autonomous vehicles.

A. Paulo Moreira graduated with a degree in Electrical Engineering from the University of Porto in 1986. He then pursued graduate studies at the University of Porto, completing a M.Sc. degree in Electrical Engineering - Systems in 1991 and a Ph.D. degree in Electrical Engineering in 1998. From1986 to 1998 he also worked as an assistant lecturer in the Electrical Engineering Department of the University of Porto. He is currently a lecturer in Electrical Engineering, developing his research within the Robotic and Intelligent Systems Unit of INESC Porto. His main research areas are Process Control and Robotics.

Aníbal Matos completed a B.Sc., an M.Sc. and a Ph.D. degree in Electrical and Computer Engineering at the University Porto in 1991, 1994, and 2001, respectively. He is currently working as an assistant lecturer at the Electrical and Computer Engineering Department of the University of Porto and he is also a researcher at the Robotics and Intelligent Systems Unit at INESC Porto. His research areas include modelling, navigation and control of autonomous vehicles, nonlinear control systems, and marine robotics.

Heber Sobreira graduated with a M.Sc. degree in Electrical Engineering from the University of Porto in 2009. Since then, he has been a Ph.D. student at Electrical and Computer Engineering Department of the University of Porto, developing his research within the Robotic and Intelligent Systems Unit of INESC-Porto (the Institute for Systems and Computer Engineering of Porto). His main research area is navigation and control of indoor autonomous vehicles.