SLAM for Robotic Assistance to fire-Fighting services

Sid Ahmed Berrabah[*,**], Yvan Baudoin[*], Hichem Sahli[**]
** Royal Military Academy of Belgium (RMA), Av. de la Renaissance 30, B1000 Brussels, Belgium*
Corresponding author E-mail: sidahmed.berrabah@rma.ac.be
*** Vrije Universiteit Brussel (VUB), Pleilaan 2, B-1050 Brussels – Belgium*

***Abstract* - In the event of an emergency, due to a fire or other crisis, a necessary but time consuming pre-requisite, that could delay the real rescue operation, is to establish whether the ground can be entered safely by human emergency workers. The objective of the VIEW-FINDER project is to develop robots which have the primary task of gathering data. The robots are equipped with sensors that detect the presence of chemicals and, in parallel, image data is collected and forwarded to an advanced base station**

One of the problems tackled in this project is the robot navigation. The used robot for the outdoor scenario is equipped with a set of sensors: camera, GPS, inertial navigation system (INS), wheel encoders, and ultrasounds sensors. The robot uses a Simultaneous Localization and Mapping (SLAM) approach to combine data from different sensors for an accurate positioning. The paper gives an overview on the proposed algorithm.

***Index Terms* - View-Finder Project, Risky Intervention, Mobile Robots, Visual Simultaneous Localization and Mapping**

I. INTRODUCTION

The objective of the View-Finder project is to develop and use advance robotics systems, equipped with a wide array of sensors and cameras, to explore a crisis ground in order to understand and reconstruct the investigated scene and thus to improve decision making.

Using robotics in this type of scene needs to be with high precision. This contribution introduces the increase of mobile robot positioning accuracy using a SLAM approach. The SLAM algorithm uses data from a single monocular camera together with data from other sensors (Global Positioning System (GPS), Inertial Navigation System (INS) and wheel encoders) for robot localization in large-scale environments.

The SLAM problem is tackled as a stochastic problem and it has been addressed with approaches based on Bayesian filtering [1-5]. The main problem of those approaches is that the computational complexity growth with the size of the mapped space, which limits their applicability in large-scale areas. In the case of vision based SLAM approaches, other challenges have to be tackled, as the high rate of the input data, the inherent 3D quality of visual data, the lack of direct depth measurement and the difficulty in extracting long-term features to map.

In this project we are concerned with robot navigation in large outdoor environments, for that we propose to build several size limited local maps and combine them into a global map using an 'history memory' which accumulates sensory evidence over time to identify places with a stochastic model of the correlation between map features. In our implementation, the dynamic model of the camera takes into account that the camera is on the top of a mobile robot which moves on a ground-plane. The SIFT algorithm [6] is used for features detection.

The data from GPS, if available, are used to help localizing the robot and features in satellite images. While the data from the inertial sensor and the wheel encoders are introduced in the vehicle modeling.

II. SYSTEM MODELING AND FEATURE EXTRACTION

In our application, a camera is fixed on the top of a mobile car-like robot "ROBUDEM" (figure 1). The vehicle travels through the environment using its sensors to observe features around it. A world coordinate frame W is defined such that its X and Z axes lie in the ground plane, and its Y axis point vertically upwards.

Fig 1: The used robot in the VIEW-FINDER project.

The system state vector of the stereo-camera $\mathbf{y}_R$ in this case is defined with the 3D position vector $r = (y_1, y_2, y_3)$ of the head center in the world frame coordinates and the robot's orientations roll, pitch and yaw about the Z, X, and Y axes, respectively $(\gamma, \theta, \varphi)$:

$$\mathbf{y}_R = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \gamma \\ \theta \\ \varphi \end{bmatrix}$$

The dynamic model or motion model is the relationship between the robot's previous state, $\mathbf{y}_R^{t-1}$, and its current state, $\mathbf{y}_R^t$, given a control input u^t

$$\boldsymbol{y}_R^t = \boldsymbol{f}(\boldsymbol{y}_R^{t-1}, u^t, \boldsymbol{v}^t) \qquad (1)$$

where $\mathbf{f}$ is a function representing the mobility, kinematics and dynamics of the robot (transition function) and $\mathbf{v}$ is a random vector describing the unmodelled aspects of the vehicle (process noise such as wheel sleep or odometry error).

The system dynamic model in our case, considering the control u as identity, is given by:

$$\mathbf{y}_R^t = \begin{bmatrix} y_1^t \\ y_2^t \\ y_3^t \\ \gamma^t \\ \theta^t \\ \varphi^t \end{bmatrix} = \begin{bmatrix} y_1^{t-1} + (\mathbf{v}^{t-1} + \mathbf{V})\cos(\gamma^{t-1})\Delta t \\ y_2^{t-1} + (\mathbf{v}^{t-1} + \mathbf{V})\sin(\gamma^{t-1})\Delta t \\ y_3^{t-1} \\ \gamma^{t-1} + (\omega^{t-1} + \mathbf{\Omega})\Delta t \\ \theta^{t-1} \\ \varphi^{t-1} \end{bmatrix} \quad (2)$$

$\mathbf{v}$ (measured by the wheels' encoders) and ω (measured by the inertial sensor) are the linear and the angular velocities, respectively. $\mathbf{V}$ and $\mathbf{\Omega}$ are the Gaussian distributed perturbations to the camera's linear and angular velocity, respectively.

Usually the features used in vision-based localization algorithms are salient and distinctive objects detected from images. Typical features might include regions, edges, object contours, corners etc. In our work, the map features are obtained using the SIFT feature detector [6], which maps an image data into scale-invariant coordinates relative to local features (e.g for detected SIFT features in figure 2) . These features were contemplated to be highly distinctive and invariant to image scale and rotation. The work of Mikolajczyk and Schmid [7] proved that SIFT features remain stable to affine distortions, change of viewpoint, noise and change in illumination.

Fig 2: Features detected using the SIFT algorithm

To deal with the problem of SLAM in dynamic scenes with moving object we use an algorithm for motion segmentation [8] to remove the outliers features which are associated with moving objects. In other words, the detected features which correspond to the moving parts in the scene are not considered in the built map. For more security we use a bounding box around the moving objects (figure 3). Another margin of security is used; the newly detected features are not added directly to the map but they should be detected and matched in at least n consecutive frames (in our application n=5).

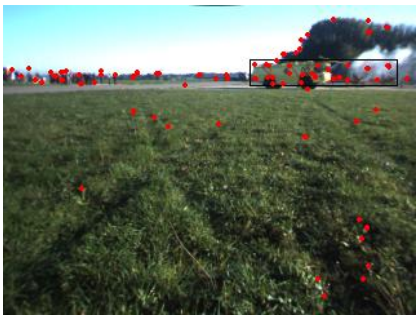

Fig 3: Features detected in a scene with moving objects

Features are represented in the system state vector by their 3D location in the world coordinate system W:

$$\mathbf{x}_i = (x_{1,i}, x_{2,i}, x_{3,i})^T$$

The observation model describes the physics and the error model of the robot's sensor.The observations are related to the system state according to:

$$\mathbf{z}^t = \mathbf{h}(\mathbf{x}^t) + \mathbf{w}^t \quad (3)$$

where $\mathbf{z}^t$ is the observation vector at time t and $\mathbf{h}$ is the observation model. The vector $\mathbf{z}_i^t$ is an observation at instant t of the i'th landmark location $\mathbf{x}_i^t$ relative to the robot's location $\mathbf{y}_R^t$.

Making a measurement of a feature i consists of determining its position in the camera image. Using a perspective projection, the observation model in the robot coordinate system is obtained as follows:

$$\mathbf{z}_\mathbf{i}^\mathbf{t} = \mathbf{h}(\mathbf{x}_\mathbf{i}^\mathbf{t}) = \begin{bmatrix} \mathrm{x}_0 + f\frac{{}^R x_{1,i}}{{}^R x_{3,i}} \\ \mathrm{y}_0 + f\frac{{}^R x_{2,i}}{{}^R x_{3,i}} \end{bmatrix} \quad (4)$$

where x_0 and y_0 are the image center coordinates and f is the focal length of the camera.

${}^R\mathbf{x}_i = ({}^R x_{1,i}, {}^R x_{2,i}, {}^R x_{3,i})^T$ are the coordinates of the feature i in the robot coordinate frame R. They are related to $\mathbf{x}_i$ by:

$${}^R\mathbf{x}_i = \begin{pmatrix} cos(\gamma) & 0 & -sin(\gamma) \\ 0 & 1 & 0 \\ sin(\gamma) & 0 & cos(\gamma) \end{pmatrix} \begin{pmatrix} x_{1,i}^t - y_1 \\ x_{2,i}^t - h \\ x_{3,i}^t - y_2 \end{pmatrix} \quad (5)$$

h is the high of the camera.

The state of the system at time t can therefore be represented by the augmented state vector, $\mathbf{x}^t$, consisting of the n_R states representing the robot, $\mathbf{y}_R^t$, and the n states describing the observed landmarks, $\mathbf{x}_i^t$, $i = 1, \ldots, n$.

The robot position and therefor the features position are measured in the universal GPS coordinate system (west-east, south-north) based on the GPS measurement, if existing.

III. EXTENDED KALMAN FILTER FOR SLAM

Given a model for the motion and observation, the SLAM process consists of generating the best estimate for the system states given the information available to the system. This can be accomplished using a recursive, three stage procedure comprising prediction, observation and update of the posterior. This recursive update rule, known as filtering for SLAM, is the basis for the majority of SLAM algorithms.

Extended Kalman Filter (EKF) is the most well-known Gaussian filter for treating the SLAM problem, where the belief is represented by a Gaussian distribution. The Kalman Filter is a general statistical tool for the analysis of time-varying physical systems in the presence of noise. Its main goal is the estimation of the current state of a dynamic

system by using data provided by the sensor measurements. Whenever a landmark is observed by the on-board sensors of the robot, the system determines whether it has been already registered and updates the filter. In addition, when a part of the scene is revisited, all the gathered information from past observations is used by the system to reduce uncertainty in the whole mapping, strategy known as closing-the-loop.

In EKF-based SLAM approaches, the environment is represented by a stochastic map $\mathcal{M} = (\hat{\mathbf{x}}, \mathbf{P})$, where $\hat{\mathbf{x}}$ is the estimated state vector (mean), containing the location of the vehicle and the n environment features, and $\mathbf{P}$ is the estimated error covariance matrix, where all the correlations between the elements of the state vector are defined. All data is represented in the same reference system. The map $\mathcal{M}$ is built incrementally, using the set of measurements $\mathbf{z_k}$ obtained by the camera. For each new acquisition, data association process is carried out with the aim of detecting correspondences between the new acquired features and the previously perceived ones.

$$\hat{\boldsymbol{x}}^t = E[\boldsymbol{x}^t] = \begin{bmatrix} \hat{\boldsymbol{y}}_R^t \\ \hat{\boldsymbol{x}}_1^t \\ \vdots \\ \hat{\boldsymbol{x}}_n^t \end{bmatrix}$$

$$P^t = E[(\mathbf{x}^t - \hat{\mathbf{x}}^t)(\mathbf{x}^t - \hat{\mathbf{x}}^t)^T] = \begin{bmatrix} \mathbf{P}_{RR}^t & \mathbf{P}_{R1}^t & \cdots & \mathbf{P}_{Rn}^t \\ \mathbf{P}_{1R}^t & \mathbf{P}_{11}^t & \cdots & \mathbf{P}_{1n}^t \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{nR}^t & \mathbf{P}_{n1}^t & \cdots & \mathbf{P}_{nn}^t \end{bmatrix}$$

The sub-matrices, $\mathbf{P}_{RR}^t$, $\mathbf{P}_{Ri}^t$ and $\mathbf{P}_{ii}^t$ are, respectively, the robot to robot, robot to feature and feature to feature covariances. The sub-matrices $\mathbf{P}_{ij}^t$ are the feature to feature cross-correlations. $\mathbf{x}$ and $\mathbf{P}$ will change in dimension as features are added or delated from the map.

The Extended Kalm Filter consists in two steps:

a) prediction step, which estimates the system state according to the state transition function f and the covariance matrix $\mathbf{P}$ to reflect the increase in uncertainty in the state du to noise Q (unmodelled aspects of the system) :

$$\mathbf{x}^{t|t-1} = \begin{bmatrix} f\left(\mathbf{y}_R^{t-1|t-1}, u = 0\right) \\ \mathbf{x}_1^{t-1|t-1} \\ \vdots \end{bmatrix}$$

$$\mathbf{P}^{t|t-1} = \mathbf{F}\mathbf{P}^{t-1|t-1}\mathbf{F}^T + \mathbf{Q}^{t-1} \qquad (5)$$

where $\mathbf{F} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^{t-1|t-1}} = diag\left(\left.\frac{\partial \mathbf{f}}{\partial \mathbf{y}_R}\right|_{\mathbf{y}_R^{t-1|t-1}}, \mathrm{I}\right)$

is the Jacobian of $\mathbf{f}$ with respect to the state vector $\mathbf{x}$ and $\mathbf{Q}$ is the process noise covariance.

Considering a constant velocity model for the smooth camera motion:

$$\left.\frac{\partial \mathbf{f}}{\partial \mathbf{y}_R}\right|_{\mathbf{y}_R^{t-1|t-1}} = \begin{bmatrix} 1 & 0 & -\sin(\gamma^{t-1})(\mathbf{v}^{t-1} + \mathbf{V})\Delta t \\ 0 & 1 & \cos(\gamma^{t-1})(\mathbf{v}^{t-1} + \mathbf{V})\Delta t \\ 0 & 0 & 1 \end{bmatrix} \qquad (8)$$

b) The Update step uses the current measurement to improve the estimated state, and therefor the uncertainty represented by P is reduced.

$$\mathbf{x}^{t|t} = \mathbf{x}^{t|t-1} + \mathbf{W}^t \varepsilon^t \qquad (9)$$

$$\mathbf{P}^{t|t} = \mathbf{P}^{t|t-1} - \mathbf{W}^t \mathbf{S}^t \mathbf{W}^{t^T} \qquad (10)$$

Where $$\mathbf{W}^t = \mathbf{P}^{t|t-1}\mathbf{H}^T(\mathbf{S}^t)^{-1} \qquad (11)$$

$$\mathbf{S}^t = \mathbf{H}\mathbf{P}^{t|t-1}\mathbf{H} + \mathbf{R}^t \qquad (12)$$

$$\varepsilon = \mathbf{z}^t - \mathbf{h}(\mathbf{x}^{t|t-1}) \qquad (13)$$

$\mathbf{Q}$ and $\mathbf{R}$ are block-diagonal matrices (obtained empirically) defining the error covariance matrices characterizing the noise in the model and the observations, respectively.

$\mathbf{H}$ is the Jacobian of the measurement model $\mathbf{h}$ with respect to the state vector. A measurement of feature $\mathbf{x}_i$ is not related to the measurement of any other feature so

$$\frac{\partial \mathbf{h}_i}{\partial \mathbf{x}} = [\frac{\partial \mathbf{h}_i}{\partial \mathbf{y}_R} \quad 0 \quad 0 \cdots \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_i} \quad 0 \quad \cdots] \qquad (14)$$

where $\mathbf{h}_i$ is the measurement model for the i'th feature.

IV. FEATURE INITIALIZATION

When a feature is first detected, measurement from a single camera position provides good information on its direction relative to the camera, but its depth is initially unknown.

Since depth information is not provided, EKF can not be directly initialized, leading to a new challenge known as Bearing-Only SLAM. An early approach was proposed by Deans [13], who combined Kalman filter and bundle adjustment in filter initialization, obtaining accurate results at the expense of increasing filter complexity. In [5], Davison uses for initialization an A4 piece of paper as a landmark to recover metric information of the scene. Then, whenever a scene feature is observed a set of depth hypotheses are made along its direction. In subsequent steps, the same feature is seen from different positions reducing the number of hypotheses and leading to an accurate landmark pose estimation. Besides, Solà et al. [14] proposed a 3D Bearing-Only SLAM algorithm based on EKF filters, in which each feature is represented by a sum of Gaussians.

In our application, to estimate the 3D position of the detected features, we use an approach based on *epipolar geometry*. This geometry represents the geometric relationship between multiple viewpoints of a rigid body and it depends on the internal parameters and relative positions of the camera. The essence of the epipolar geometry is illustrated in figure 4

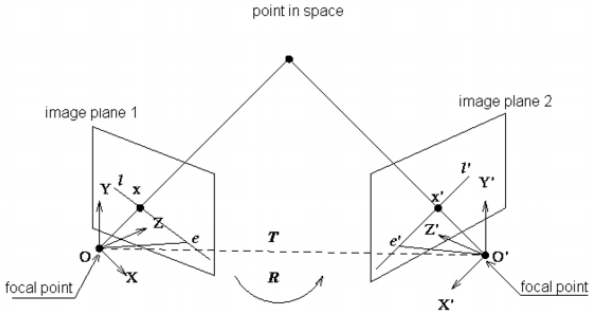

Fig 4: Illustration of the epipolar geometry

The fundamental matrix $\mathbb{F}$ (a 3×3 matrix of rank 2) encapsulates this intrinsic geometry. It describes the relationship between matching points: if a point $\tilde{\mathbf{X}}$ is imaged as $\mathbf{x}$ in the first view, and $\mathbf{x}'$ in the second, then the image points must satisfy the relation $\mathbf{x}^T \mathbb{F} \mathbf{x}' = 0$. The fundamental matrix is independent of scene structure. However, it can be computed from correspondences of imaged scene points alone, without requiring knowledge of the cameras' internal parameters or relative pose. Given a set of n pairs of image correspondences $(\mathbf{x}_j, \mathbf{x}'_j), j = 1..n$, we compute $\mathbf{R}$ and $\mathbf{t}$ such the epipolar error is minimized

$$\min_{\mathbb{F}} \sum_{j=1}^{n} \mathbf{x}'_j \mathbb{F} \mathbf{x}_j \tag{15}$$

For the minimization, we use the Random Sample Consensus (RANSAC) algorithm.

The camera coordinate systems corresponding to tow views are related by a rotation matrix, $\mathbf{R}$, and a translation vector, $\mathbf{t}$:

$$\mathbf{x}' = \mathbf{R}\mathbf{x} + \mathbf{t} \tag{16}$$

Taking the vector product with $\mathbf{t}$, followed by the scalar product with $\mathbf{x}'$, we obtain:

$$\mathbf{x}'.(\mathbf{t} \wedge \mathbf{R}\mathbf{x}) = \mathbf{0} \tag{17}$$

This can also be written as

$$\mathbf{x}'\mathbf{E}\mathbf{x} = \mathbf{0} \tag{18}$$

where

$$\mathbf{E} = \mathbf{t}_{\times}\mathbf{R} \tag{19}$$

is the essential matrix, and $\mathbf{t}_{\times}$ denotes skew symmetric cross product matrix for $\mathbf{t}$

$$\mathbf{t}_{\times} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

The rotation $\mathbf{R}$ and translation $\mathbf{t}$ between the two camera frames are then calculated by singular value decomposition SVD of $\mathbf{E}$.

Suppose that the SVD decomposition of $\mathbf{E}$ is $\mathbf{U}\text{diag}(1,1,0)\mathbf{V}^{\mathrm{T}}$. The factorization $\mathbf{E} = \mathbf{t}_{\times}\mathbf{R}$ corresponds to:

$$\mathbf{t}_{\times} = \mathbf{U}\mathbf{Z}\mathbf{U}^{\mathrm{T}} \qquad \mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^{\mathrm{T}}$$

Where

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{Z} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

And then the camera projection are given by:

- camera projection matrix for the first view $\mathbf{P} = [I|0]$
- camera projection matrix at the second view $\mathbf{P}' = [\mathbf{R}|\mathbf{t}]$

Knowing the camera calibration matrix $\mathbf{K}$, we can calculate the essential matrix $\mathbf{E}$ as follows:

$$\mathbf{E} = \mathbf{K}^T \mathbb{F} \mathbf{K} \tag{20}$$

The camera calibration matrix $\mathbf{K}$ encodes the transformation from image coordinates to pixel coordinates in the image plane. It depends on the so-called intrinsic parameters: focal distance f (in mm), principal point (or image centre) coordinates o_x,o_y (in pixel), width (s_x) and height (s_y) of the pixel footprint on the camera photosensor (in mm), and angle θ between the axes (usually $\pi/2$). The ratio s_y/s_x is the aspect ratio (usually close to 1).

$$\mathbf{K} = \begin{bmatrix} f/s_x & f/s_x \cos\theta & o_x \\ 0 & f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Let $\mathbf{x}$ and $\mathbf{x}'$ be the two corresponding points satisfaying the epipolar constraint $\mathbf{x}^T \mathbb{F} \mathbf{x}' = 0$. Given the camera matrices$\mathbf{P}$ and $\mathbf{P}'$, the depth of the 3D point corresponding to $\mathbf{x}$ and $\mathbf{x}'$ can be calculated by:

$$\boldsymbol{z} = \frac{(\boldsymbol{e} \times \mathbf{x})(\mathbf{x} \times \mathbf{x}')}{\|\mathbf{x} \times \mathbf{x}'\|^2} \tag{21}$$

where $\boldsymbol{e}$ is the epipole at the first view.

V. FEATURE MATCHING

At step *t*, the onboard sensor obtains a set of measurements $\mathbf{z}_i^t$ (*i = 1, ... ,m*) of *m* environment features. Feature matching corresponds to data association, also known as the correspondence problem, which consists in determining the origin of each measurement, in terms of the map features $\mathbf{x}_j$, *j = 1, . . . , n*. The measurement z_i^t can be considered corresponding to the feature *j* if the Mahalanobis distance $D_{ij}^{2\,t}$ satisfies:

$$D_{ij}^2 = \varepsilon^T \mathbf{S}^{-1} \varepsilon < th \tag{22}$$

(20)

where the covariance $\mathbf{S}^t$ and the innovation ε^t are given by equations (12) and (13), respectively.

In our application, as we are using SIFT features, the matching between feature is checked using a product of the Mahalanobis distance between measurements and their predictions and the Euclidean distance between the descriptor vectors of the features. This will allow using the advantage of looking for feature matching based on the prediction of their position based on the system model and the advantage of the space-scale invariance parameters.

$$D^2 = D_{ij}^2 + D_{desc}^2 < th \tag{23}$$

where

$$D^2_{desc} = \|desc_1 - desc_2\|$$

is the Euclidean distance between the discriptor vectors of the features.

Additionally, corresponding features should satisfy the epipolar constraint, hence an image point $\mathbf{x}_i^t$ that corresponds to $\mathbf{x}_i^{t-1}$ is located on or near the epipolar line that is induced by $\mathbf{x}_i^{t-1}$. The distance of the image point $\mathbf{x}_i^t$ from that epipolar line is computed as follows:

$\mathbb{F}\mathbf{x}_i^{t-1}|_j$ is the j component of the vector $\mathbb{F}\mathbf{x}_i^{t-1}$. $\mathbb{F}$ is the fundamental matrix which is computed based on the estimations from the Extended Kalman Filter.

Therefore, our cost function for features matching is the sum of D^2 and D^2_{epi}:

$$D^2_{match} = D^2 + D^2_{epi} \quad (24)$$

VI. SLAM IN LARGE-SCALE AREAS

The main open problem of the current state of the art SLAM approaches and particularly vision based approaches is mapping large-scale areas. Relevant shortcomings of this problem are, on the one hand, the computational burden, which limits the applicability of the EKF-based SLAM in large-scale real time applications and, on the other hand, the use of linearized solutions which compromises the consistency of the estimation process. The computational complexity of the EKF stems from the fact that covariance matrix **P** represents every pairwise correlation between the state variables. Incorporating an observation of a single landmark will necessarily have an effect on every other state variable. This make the EKF computationally infeasible for SLAM in large environment.

Methods like Network Coupled Feature Maps [9], Sequential Map Joining [10], and the Constrained Local Submap Filter (CRSF) [11], have been proposed to solve the problem of SLAM in large spaces by breaking the global map into submaps. This leads to a more sparse description of the correlations between map elements. When the robot moves out of one submap, it either creates a new submap or relocates itself in a previously defined submap. By limiting the size of the local map, this operation is constant time per step. Local maps are joined periodically into a global absolute map, in an $O(N^2)$ step. Each approach reduces the computational requirement of incorporating an observation to constant time. However, these computational gains come at the cost of slowing down the overall rate of convergence.

The Constrained Relative Submap Filter [11] proposes to maintain the local map structure. Each map contains links to other neighboring maps, forming a tree structure (where loops cannot be represented). The method converges by revisiting the local maps and updating the links through correlations. Whereas in the hierarchical SLAM [12], links between local maps form an adjacency graph. This method allows to reduce the computational time and memory requirements and to obtain accurate metric maps of large environments in real time.

To solve the problem of SLAM in large spaces, in our study, we propose a procedure to break the global map into submaps by building a global representation of the environment based on several size limited local maps built using the previously described approach. The global map is a set of robot positions where new local maps started (i.e. the base references of the local maps). The base frame for the global map is the robot position at instant t_0.

Each local map is built as follows: at a given instant t_k, a new map is initialized using the current vehicle location, $\mathbf{y}_R^{tk}$, as base reference $B_k = \mathbf{y}_R^{tk}$, k=1, 2,... being the local map order. Then, the vehicle performs a limited motion acquiring sensor information about the L_i neighboring environment features.

The 'k'th local map is defined by:

$$\mathfrak{M}_k = (\mathbf{X}_k, \mathbf{P}_k)$$

where $\mathbf{X}_k$ is the state vector in the base reference B_k of the L_k detected features and $\mathbf{P}_k$ is their covariance matrix estimated in B_k.

The decision to start building a new local map at an instant t_k is based on two criteria: the number of features in the current local map and the scene cut detection result. The instant t_k is called a key-instant. In our application we defined two thresholds for the number of features in the local maps: a lower Th^- and a higher Th^+ thresholds. A key-instant is selected if the number of features n_l^k in the current local map k is bigger then the lower threshold and a scene cut has been detected or the number of features has reached the higher threshold. This allows kipping reasonable dimensions of the local maps and avoids building too small maps.

The global map is:

$$\mathfrak{M}_G^B = (\bar{\mathbf{y}}_R^0, \bar{\mathbf{y}}_R^1, \bar{\mathbf{y}}_R^2, \dots) \quad (25)$$

where $\bar{\mathbf{y}}_R^k$ are the robot coordinates in B_0, where it decides to build the local map $\mathfrak{M}_k$ at instant t_k.

$$\begin{pmatrix} \bar{\mathbf{y}}_R^k \\ 1 \end{pmatrix} = \mathcal{T}_{k\to 0} . \begin{pmatrix} \mathbf{y}_R^{t_k} \\ 1 \end{pmatrix}$$

$t_0 = 0$ and $\bar{\mathbf{y}}_R^0 = \mathbf{y}_R^{t_0} = (0,0,0)$.

The transformation matrix $\mathcal{T}_{k\to 0}$ is obtained by successive transformations:

$$\mathcal{T}_{k\to 0} = \mathcal{T}_{1\to 0} . \mathcal{T}_{2\to 1} \dots \mathcal{T}_{(k-1)\to(k-2)}$$

where $\mathcal{T}_{i\to i-1} = (\mathcal{R}|\mathbf{t})$ is the transformation matrix corresponding to rotation $\mathcal{R}$ and translation $\mathbf{t}$ of frame B_i regarding to frame B_{i-1}:

$$\mathcal{T}_{i\to i-1} = \begin{pmatrix} \cos(\gamma^{t_i}) & 0 & -\sin(\gamma^{t_i}) & y_1^{t_i} \\ 0 & 1 & 0 & 0 \\ \sin(\gamma^{t_i}) & 0 & \cos(\gamma^{t_i}) & y_2^{t_i} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In this case, for feature matching at instant t, the robot uses the local map with the closest base frame to its current location:

$$\underset{i}{\operatorname{argmin}}(\mathrm{P}\bar{\mathbf{y}}_R^k - \bar{\mathbf{y}}_R^t \mathrm{P})$$

where $\bar{\mathbf{y}}_R^t$ is the robot position at instant t in B_0.

Fig5 and fig6 show respectively an example of the proposed SLAM process and the results of the ROBUDEM localization in a real environment. Black squares in figure 5, describes the positions where the algorithm started a new local map. The ellipses around the features on the original frame (fig 5) represent the estimated covariance, cyan for non matched feature and red for matched features

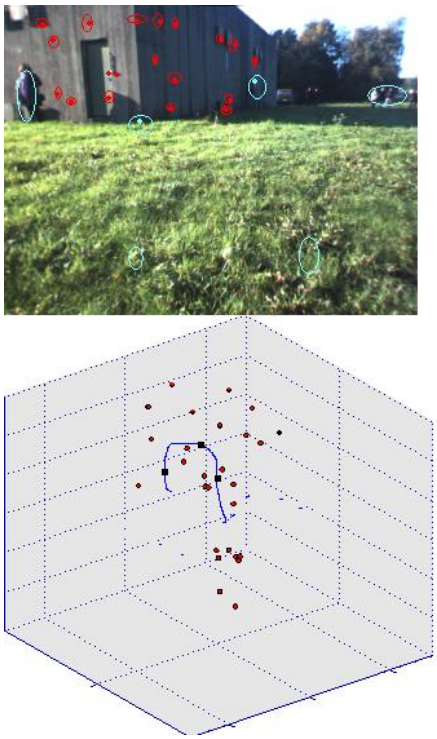

Fig 5: example of SLAM process

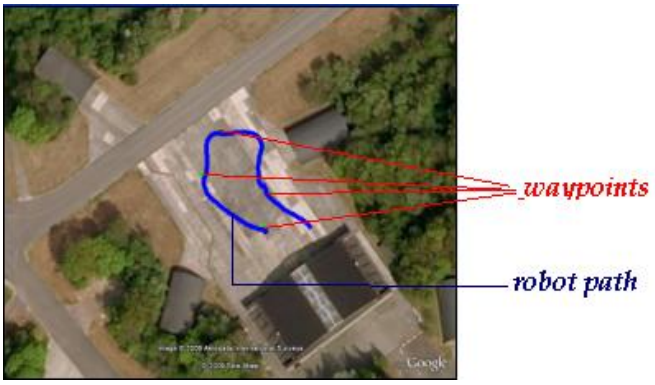

Fig 6**:** ROBUDEM localization in a real environment.

Fig7 shows the error on the robot position. The results shows how precise is the proposed algorithm where the error is kept less than 1m.

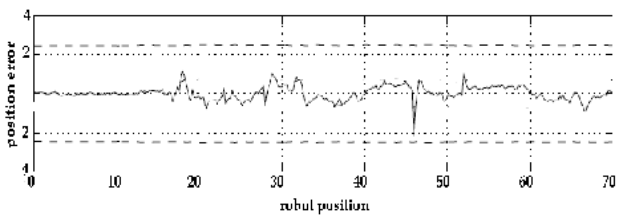

Fig 7**:** Estimated error on robot position

VII. CONCLUSION

In this paper, we presented an algorithm for robot localization using a SALM Approach combining data from different sensors: a monocular camera, GPS, INS, and wheels encoders. The proposed approach has been applied successfully for a real robot localization in the framework of an European project VIEW-FINDER

ACKNOWLEDGMENT

This research is founded by the European View-Finder FP6 IST 045541 project.

REFERENCES

[1] J. Folkesson, P. Jensfelt, H. Christensen, *Graphical SLAM using vision and the measurement subspace*, In IEEE/JRS -Intl Conf. on Intelligent Robotics and Systems (IROS), Edmundton-Canada, August, 2005.

[2] D. Wolf, G.S. Sukhatme, *Online Simultaneous Localization and Mapping in Dynamic Environments*, Proceedings of the Intl. Conf. on Robotics and Automation ICRA New Orleans, Louisiana, April, 2004.

[3] F. Andrade-Cetto, A. Sanfelin, *Concurrent Map Building and Localization with landmark validation*, 16th Intenational Conference on Pattern Recognition ICPR'02, 2002, vol.2.

[4] J. W. Fenwick, P. M. Newman, J. J. Leonard, *Cooperative Concurrent Mapping and Localization*, IEEE International Conference on Robotics and Automation, May 2002, Washington, USA, pp.1810-1817.

[5] J. Davison, I. D. Reid, N. D. Molton, O. Stasse, *MonoSLAM: Real-Time Single Camera SLAM*, IEEE Transaction on Pattern Analysis and Machine Intelligence, JUNE 2007, Vol.29, N.6.

[6] D. G. Lowe, *Distinctive image features from scale-invariant keypoints***,** International Journal of Computer Vision, 60, (2) 2004, pp. 91-110.

[7] K. Mikolajczyk, C. Schmid. *A performance evaluation of local descriptors*, Proceedings of Computer Vision and Pattern Recognition, 2003

[8] S. A. Berrabah, G. De Cubber, V. Enescu, H. Sahli, *MRF-based foreground detection in image sequences from a moving camera*, Proceedings of the International Conference on Image Processing, ICIP2006, Atlanta, USA, October 8-11, 2006.

[9] I. Bailey, *Mobile robot localisation and mapping in extensive outdoor environments*. PhD thesis, Australian Centre for Field Robotics, University of Sydney, Australia, August 2002.

[10] J.D. Trados, J. Neira, P. Newman, J. Leonard, *Robust mapping and localization in indoor environments using sonar data*, International Journal of Robotics Research, 2002, N. 21, pp.311-330.

[11] S. B. Williams, *Efficient Solutions to Autonomous Mapping and Navigation Problems*, PhD thesis, Australian Centre for Field Robotics, University of Sydney, Australia, September 2001.

[12] C. Estrada, J. Neira, J. D. Tardos, *Hierarchical SLAM: real-time accurate mapping of large environments*, IEEE Transactions on Robotics, 2005, Vol.21, N.4, pp.588-596.

[13] M Deans, M. Hebert, *Experimental comparison of techniques for localization and mapping using a bearing-only sensor*, In Seventh International Symposium on Experimental Robotics, Honolulu, Hawaii, December 2000, volume 271, pp. 395- 404.

[14] J. Solà, T. Lemaire, M. Devy, S. Lacroix, A. Monin, *Delayed vs Undelayed Landmark Initialization for Bearing Only SLAM*, In Proceedings of the the IEEE Int. Conf. on Robotics and Automation workshop on SLAM, Barcelona, Spain, April 2005.