An optimization technique for positioning multiple maps for self-driving car's autonomous navigation

Dominguez S.[1], Khomutenko B.[1], Garcia G.[1], and Martinet P.[1]

Abstract—**Self-driving car's navigation requires a very precise localization covering wide areas and long distances. Moreover, they have to do it at faster speeds than conventional mobile robots. This paper reports on an efficient technique to optimize the position of a sequence of maps along a journey. We take advantage of the short-term precision and reduced space on disk of the localization using 2D occupancy grid maps, from now on called *sub-maps*, as well as, the long-term global consistency of a Kalman filter that fuses odometry and GPS measurements. In our approach, horizontal planar LiDARs and odometry measurements are used to perform 2D-SLAM generating the sub-maps, and the EKF to generate the trajectory followed by the car in global coordinates. During the trip, after finishing each sub-map, a relaxation process is applied to a set of the last sub-maps to position them globally using both, global and map's local path. The importance of this method lies on its performance, expending low computing resources, so it can work in real time on a computer with conventional characteristics and on its robustness which makes it suitable for being used on a self-driving car as it doesn't depend excessively on the availability of GPS signal or the eventual appearance of moving objects around the car. Extensive testing has been performed in the suburbs and in the down-town of Nantes (France) covering a distance of 25 kilometers with different traffic conditions obtaining satisfactory results for autonomous driving.**

I. Introduction

Most daily journeys people do are repetitive, like for instance, going from home to work and back, wasting everyday a lot of time and energy. This kind of trips can be pre-recorded by driving the car once manually and after using the data recorded to localize itself with high precision and robustness during autonomous navigation mode.

Our research is focused in self-driving cars covering long distances mostly along urban environment. In this kind of environments, the precision obtained with just sensor fusion of GPS, and odometry, has demonstrated to be insufficient for autonomous navigation, specially in those places where the GPS signal is not available or suffers from *multi-path effect* due to reflection on the surrounding buildings. The evolution model of the fusion provides a dead-reckoned estimate, and the drifting can be corrected using a model of the environment, like for example a map. Moreover, we expect to deal with static and moving objects (e.g. cars, people around the vehicle, etc.). Obstacles can be detected by using sensors like LiDARs, cameras, ultrasonic sensors, radars, etc., and be classified as

static or moving. The amount of information to be processed along the journey can be so huge that, if the data coming from the sensors are not well managed, it can be a drawback that slows down the fluidity of the processing. To this end, we have developed a localization system that takes advantage of the good short-term precision, quickness and little space on disk occupied using 2D-SLAM in a local area that we call sub-map, as well as, the consistency of global positioning that corrects the error produced by the integration of local positioning along the time.

The global pose (latitude, longitude and heading) is produced by an EKF, which fuses odometry, and GPS measurements providing also the covariances of the errors. On the other hand, we use a chain of occupancy grid sub-maps instead of a big map covering the working area. The sub-maps are supposed to be rigid and connected one after another through the so called *connection points*.

During the map building phase, for each sub-map i we register simultaneously at pre-defined intervals of distance $L_{threshold}$ the trajectory of the car in the map $\mathcal{P}_m^i = \{P_{j_m}^i\}_{\{j=0...N_i-1\}}$, and the corresponding trajectory and the position error's covariance in global coordinates $\mathcal{P}_g^i = \{P_{j_g}^i, \sigma_{j_g}^2\}_{\{j=0...N_i-1\}}$ being N_i the number of points of the local trajectory. In this way we obtain two separate paths that we call from now on *map path* and *global path* for the SLAM-2D and EKF positions respectively (See Figure 1).
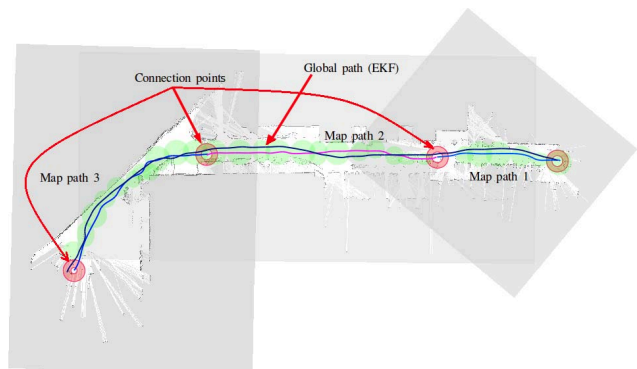

Fig. 1. *global path* and *map paths* represented with different colors along a journey composed by chain of sub-maps connected through the *connection points*. The green circles represent the uncertainty on the global's position.

All points in both paths are referenced to the same common frame called *global frame*(See Figure 3). Successive sub-maps's are connected through the so called (*connection points*) to keep continuity between them. We apply an iterative process of relaxation that tries to match the map path with the global path, based on the idea that points with more precise global

[1]IRCCyN, Ecole Centrale de Nantes, 1 Rue de la noe, 44321 Nantes, France
Salvador.Dominguez@irccyn.ec-nantes.fr
khomutenko.bogdan@irccyn.ec-nantes.fr
Gaetan.Garcia@ec-nantes.fr
Philippe.Martinet@irccyn.ec-nantes.fr

coordinates must have more influence in the *matching* process. The idea is to treat the global path points as static and linked to their respective map path points through an elastic force that is proportional to the distance between both points. The proportional constant is set to the inverse of the covariance of the error of the global path point. In such situation, the chain of sub-map paths tries to find a balancing point where the energy accumulated by the virtual attraction forces is minimal. After the relaxation process, the result is a sequence of sub-maps with their map paths positions more consistent with the global position. Note that areas with low quality GPS signal (e.g. urban canyon) or non-existent (tunnels) leads to high covariance and therefore low or null attraction forces allowing neighbour sub-maps with good GPS position to put them in place indirectly.

In the other hand, during the localization phase, a probabilistic particle filter is used to localize locally on the sub-map that corresponds to the position of the car. The wheel odometry is used for predicting the evolution of the particles, and the laser scan for selecting the particle that better matches the current sub-map. A re-sampling of the particles is performed periodically by distance travelled, angle rotated and also in time. The surviving particles define the evolution of the car's position. Note that, as the sub-maps are globally positioned, it is straightforward to convert the local pose into global coordinates. A map manager loads the next sub-map whenever the car reaches the end of the sub-map in use. When that happens, the particle filter needs to be initialized in the new sub-map with an initial dispersion. Although, such initialization is in most cases successful, in some cases can cause uncertainty during the first meters until the particle filter converges again. In order to accelerate the convergence we increase the re-sampling rate during a certain distance after loading the sub-map.

The method has proved to be robust and reliable along intensive tests in city center and suburbs of the french city of Nantes. We use a car Renault Zoe ZE in which we have installed strategically three LiDARs LMS151 covering 360^0 around the car (See Figure 2) and the speed is obtained from the OBD-II interface of the car. Regarding the resources consumed by the map-building process, the space occupied on disk of each sub-map is limited to 750Kb, which gives us an experimental average of about 5 Mb/km travelled using a map grid cell size of 20x20cm. The CPU consumption in this phase is less than 50% of a core in an Intel(R) Core(TM) i5-3610ME CPU @ 2.70GHz travelling at 50Km/h. During the localization phase the CPU consumption is about 25 % of one core, and the precision of the localization relative to the local environment is less than 10cm in average.

LiDARs installed on the front

LiDARs on the rear side

Fig. 2. The LiDARs Sick LMS151 are installed in strategical positions to cover the surrounding area of the vehicle.

II. RELATED WORK

In our approach we define the route to follow as a sequence of maps built sequentially using 2D-SLAM. In the bibliography there exist multiple approaches to handle the SLAM problem.

a) Visual SLAM: The vSLAM techniques use different data, extracted out of images. It can be sparse set of key points or a dense or semi-dense depth map. Multiple successful results for vSLAM using sparse key point cloud have been reported [1]. Among the most recent developments, in [2] they propose a semi-dense vSLAM, which is reported to work in real time on a consumer PC. Another interesting point for us, is the map representation method. It is represented, not as a huge point cloud, but as a pose graph. There is a key frame attached to each node of the graph, and for each pose on it, the image itself and the depth map are kept. Another interesting approach is given in [3], this time using a small base-line stereo camera and performing multi-map localization by formulating the SLAM problem as a non-linear batch optimization (tested only in indoors).

Visual SLAM techniques are quite elaborated for the case, when the environment is static. They are robust to environment changes to some extend but don't treat environmental dynamics explicitly. There are some approaches to tackle the dynamic environment using vision. In paper [4] a notion of visual experiences is introduced. This learning-like technique requires multiple passes by streets with high environmental dynamics to attain stable localization. In [5] a technique to integrate features from different seasons into the map is developed. It makes the localization more robust with respect to the environmental changes during the year. While this technique gives good results, the computations are carried out off-line using brute-force approach for loop closure.

One of the challenges of vSLAM is its dependence on lighting conditions to extract reliable features for being identified in subsequent passes through that area regardless of the lighting conditions. Probably that is the reason why one of the most widely used technologies today is the measurement with LiDARs. LiDARs are convenient tools to perform mapping because they provide directly metric information. As already mentioned in the introduction we have installed three LiDARs to cover a 360^0 view around the car.

b) LiDARs SLAM: Many works are concentrated on an indoor environment, what implies a limited space and low speeds. There are techniques that are able to handle outdoor dynamic environment. In [6] a technique to represent a map for a dynamic environment is proposed. The map is represented as a pose graph in that approach. Sampled measurements for different time scales are kept. There are works dedicated to outdoors. In [7] they propose a technique to explicitly track moving objects and discard them from the localization process.

In most of Bayesian probabilistic techniques, even though moving objects modify temporally the view of the environment, as they do not stay in the same place for long time, statistically they are not considered to form part of the map. In our approach, we have performed several test of localization in the center of the city of Nantes, using maps built months ago using LiDARs and still getting good localization in such

conditions without removing the moving objects from the laser scans.

c) Multi-map SLAM: In [8] they propose building a sequence of consecutive uncorrelated local maps and joining them together in a *a priori* known map using combined constraint data association (CCDA) in an EFK SLAM approach. More recent works like in [9] focus their study on computer resources optimization techniques based on hierarchical decomposition in small sub-maps with limited size, so that it can be run in real time without using too much memory. In this paper we propose a different approach but basically with the same idea in mind. The algorithm must be able to be executed in real time on a PC with conventional characteristics producing a set of local maps globally positioned ready to be used for vehicle localization.

III. METODOLOGY

In this section we describe the process of estimating the sub-maps positions by relaxation of forces between map path and global path.

A. Building paths and forces

The positioning obtained when performing 2D-SLAM is local to the respective sub-map's origin, but we need it to be converted into global coordinates. The origin of the first sub-map is initially set to its corresponding global pose and the origin of the rest sub-maps are connected to the last pose of the previous sub-map through the respective connection point to keep continuity between map paths (See Figure 3). For each
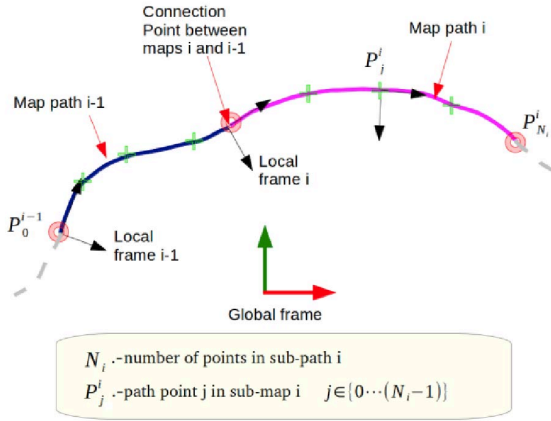

Fig. 3. global and local submap frames used in the optimization process.

sub-map, we record at regular intervals of distance L_{thresh} the position in submap's local coordinates, the global position provided by the EKF and its estimated error covariance. Thus we generate for each sub-map two paths: local map path and global path (position and covariance). The global path is used as reference in the relaxation process. As explained in the introduction, we consider virtual elastic forces between the map path points and the corresponding global path points (See Figure 5). We have chosen this model because we can define on it the weight of the force for each pair of points through the elasticity constant, which in our case will be a function of the

global position's error. A spring network always tries to reach the minimum global energy that in our case correspond to the balancing state where the global positioning error of the whole path is minimal. The global path's points that are more accurate attract stronger the corresponding map path's points than those with lower precision. That behaviour is modelled through the *spring* coefficient K_j^i for each pair of points $P_{j_g}^i, P_{j_m}^i$ for global path and map path respectively.

B. Sub-map paths relaxation

When the sub-map size under construction reaches to a maximum threshold, a relaxation process begins. The combined action of all forces along the whole trajectory will put all sub-maps in the position that minimizes the sum of the individual positioning errors, in other words, the position that minimizes the total energy of the elastic interconnections between global and map paths. The process of relaxation computes iteratively the increment of angle that every sub-map must rotate in order to get closer to the balance position, taking into account that the sub-map are chain-joined through the corresponding *connection points*.

Figure 4 shows the position of a generic sub-map path point P_j^i. The absolute orientation of each sub-map is represented by θ^i and the angular position of the jth point of sub-map i by θ_j^i. The values of those angles before the relaxation process are known but we are interested in the values that minimize the total energy. To simplify we can represent the balanced position of any map path's point $P_{j_m}^i$ as:

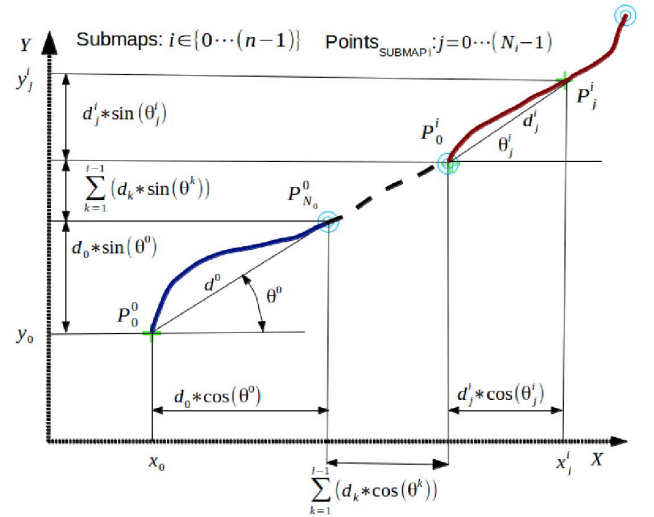$$\vec{l}_{j_m}^i = \{x_j^i, y_j^i\} = f_j^i(\theta_0, \ldots, \theta_i) \quad (1)$$

Fig. 4. Trigonometric representation of the position of a generic point *j* in sub-map *i* P_j^i

Figure 5 shows the variables involved in balancing a generic sub-map *i*. Each map path point $P_{j_m}^i$ is attracted by a global path point $P_{j_g}^i$ with a force

$$\vec{F}_j^i = K_j^i * (\vec{l}_{j_g}^i - \vec{l}_{j_m}^i) \quad (2)$$

simulating a virtual spring between them, where $K_j^i = \frac{1}{cov_j^i}$ is the elasticity coefficient and $\vec{l}_{j_g}^i$ is the vector position of the global path point. Both, cov_j^i and $\vec{l}_{j_g}^i$ are provided by the EKF and according to the expression (1) $\vec{l}_{j_m}^i$ only depends on the angles $\{\theta^0, \ldots, \theta^i\}$, so we can consider that the forces in (2) only depend on such angles.
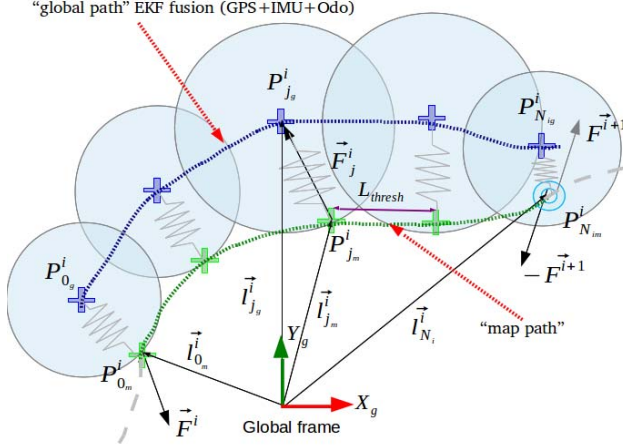

Fig. 5. Each *map path* point $P_{j_p}^i$ is attracted to its corresponding *global path* point $P_{j_g}^i$ with a force $\vec{F}_j^i$ proportional to the inverse of the covariance of the error cov_j^i and the relative position between them $\vec{l}_j^i$. Note that in the connection point $P_{N_m}^i$ the force between the connected sub-maps have same value but contrary direction by the principle of action and reaction

For each sub-map i we apply balancing of forces and momentums:

$$\sum_{j=0}^{N_i-1} \vec{F}_j^i + \vec{F}^i - \vec{F}^{i+1} = \vec{0} \tag{3}$$

$$\sum_{j=0}^{N_i-1} \vec{F}_j^i \times \vec{l}_{j_m}^i + \vec{F}^i \times \vec{l}_{0_m}^i - \vec{F}^{i+1} \times \vec{l}_{N_{i_m}}^i = \vec{0} \tag{4}$$

where $\vec{F}^i$ and $\vec{F}^{i+1}$ are the unknown reaction forces of sub-map i with sub-maps *i-1* and *i+1* respectively.

Thus, we obtain for the reference frame's origin of sub-map i ($\bar{X}^i$) the vector of unknowns

$$\bar{X}^i = \theta^0, \ldots, F_{x_j}^i, F_{y_j}^i, \theta^i, F^{i+1}i_{x_j}, F^{i+1}i_{y_j} \tag{5}$$

If we consider that the total number of sub-maps to balance is n ($n > i$), the full set of unknowns in the system is:

$$\bar{X} = \{F_x^0, F_y^0, \theta^0, \ldots, F_x^{n-1}, F_y^{n-1}, \theta^{n-1}\} \tag{6}$$

Taking (1) and (2) into account, we can express (3) and (4) as three scalar equations:

- force balancing x axis: $g_{fx}^i(\bar{X}) = 0$
- force balancing y axis: $g_{fy}^i(\bar{X}) = 0$
- momentum balancing: $g_M^i(\bar{X}) = 0$

that we can express in form of array as:

$$\bar{g}^i = \begin{bmatrix} g_{fx}^i(\bar{X}) \\ g_{fy}^i(\bar{X}) \\ g_M^i(\bar{X}) \end{bmatrix} \quad i \in \{0 \ldots (n-1)\} \tag{7}$$

we can linearise (7) by building the *3 x (3n)* partial Jacobian for sub-map i:

$$J^i = \begin{bmatrix} \frac{\partial g_{fx}^i}{\partial F_x^0} & \frac{\partial g_{fx}^i}{\partial F_y^0} & \frac{\partial g_{fx}^i}{\partial \theta^0} & \cdots & \frac{\partial g_{fx}^i}{\partial F_x^{n-1}} & \frac{\partial g_{fx}^i}{\partial F_y^{n-1}} & \frac{\partial g_{fx}^i}{\partial \theta^{n-1}} \\ \frac{\partial g_{fy}^i}{\partial F_x^0} & \frac{\partial g_{fy}^i}{\partial F_y^0} & \frac{\partial g_{fy}^i}{\partial \theta^0} & \cdots & \frac{\partial g_{fy}^i}{\partial F_x^{n-1}} & \frac{\partial g_{fy}^i}{\partial F_y^{n-1}} & \frac{\partial g_{fy}^i}{\partial \theta^{n-1}} \\ \frac{\partial g_M^i}{\partial F_x^0} & \frac{\partial g_M^i}{\partial F_y^0} & \frac{\partial g_M^i}{\partial \theta^0} & \cdots & \frac{\partial g_M^i}{\partial F_x^{n-1}} & \frac{\partial g_M^i}{\partial F_y^{n-1}} & \frac{\partial g_M^i}{\partial \theta^{n-1}} \end{bmatrix} \tag{8}$$

The total Jacobian is a composition of the n partial J^i ($i \in \{0 \ldots (n-1)\}$):

$$J = \begin{bmatrix} J^0 \\ \ldots \\ J^{n-1} \end{bmatrix} \{(3*n) \times (3*n)\} \tag{9}$$

The balancing status can be obtained by approximation applying the method of Newton-Raphson. At iteration $k+1$ we get closer to the solution solving the system

$$J * (\bar{X}_{k+1} - \bar{X}_k) = J * \Delta \bar{X} = -g(\bar{X}_k) \tag{10}$$

as J is a regular matrix $\{(3*n) \times (3*n)\}$ can be inverted obtaining directly the solution

$$\Delta \bar{X} = -J^{-1} * g(\bar{X}_k) \tag{11}$$

and finally

$$\bar{X}_{k+1} = \bar{X}_k + \Delta \bar{X} \tag{12}$$

the iteration process stops when the number of iteration reaches to a maximum or $\|\Delta \bar{X}\| \leqslant \epsilon$.

The final pose of the sub-maps can be obtained by applying the expression (1) that corresponds to Figure 4.

IV. EXPERIMENTAL RESULTS

To test the quality of the map positioning we have performed an experiment that covers about 25.13 Km in the suburbs and down-town Nantes (France). We present the experiment's set-up and the path obtained after the optimization compared with the ground truth in the points where it is available, meaning that the precision is enough to perform the comparison. We generate the ground truth using the position provided by the RTK GPS.

The raw sensor data has been captured using the software "Effibox" which ensures a right time-stamping. The dataset includes:

- Ground truth global position using a PROFLEX 800 RTK-GPS receiver. With a precision of up to $\pm 1cm$ when it is in "fix mode".

- Global position using a 'low-cost' UBLOX6 GPS receiver. Used as input in the EKF solution.

- Linear speed of each of the four wheels and angle of the steering wheel read from the OBD-II port of the car.

- Three-axis angular speeds and accelerations provided by an IMU Sparkfun Razor 9DOF.

- Laser scans from the three LiDARs LMS151 installed in the car. Used in 2D-SLAM for map building and in the particle filter for localization on sub-map.

Both the car's speed obtained from the OBD-II port and IMU measurements are used to generate the odometry of the car. A previous process for the calibration of the odometry has been performed to ensure good precision all way through each sub-map. During the SLAM-2D map-building, a procedure of sub-map relaxation is done after a new sub-map is built using a buffer of the last four sub-maps. We save the maps and paths on disk for further analysis of the error. To obtain the errors we compare those map paths with respect to the ground truth.

For each ground truth position we interpolate the map path, using a *B-spline* that fits a neighbour set of points of the sub-map positions computing the orthogonal projection Q_s, of the ground truth position onto the spline. That point is the closest point of the spline to the GT position in which the dot product of the vector $\vec{v}_1$ and the spline's tangent $\vec{v}_2$ is null. This smooths the path, making it more continuous instead of using a chained list of rectilinear segments. The error d_{error} is the distance between the ground truth position and Q_s, and the angular error is the difference between the orientation of the ground truth's position Θ^{rtk} and the angle of the spline's tangent in Q_s, Θ^{map} (See Figure 6). Figure 7
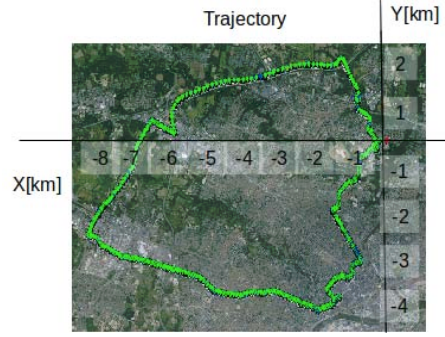

Fig. 7. Trajectory of the experiment 25Km long. Starting and finish points are the same position. The route was performed in counter-clockwise
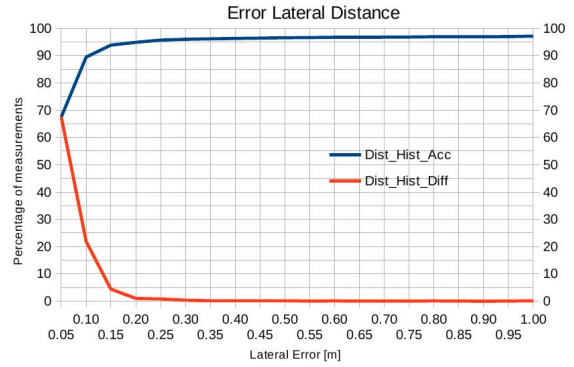

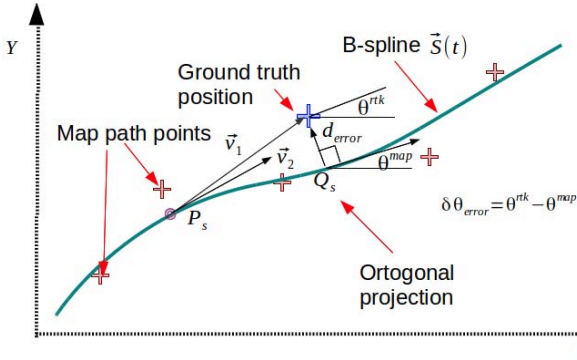Fig. 8. Accumulated and differential histograms of the lateral error

Fig. 6. Obtention of the path errors. Θ^{rtk} and Θ^{map} are the heading of ground truth and computed solution respectively. P_s is a generic point on the spline interpolated path and Q_s is the point of the spline in which the spline's tangent is orthogonal to the vector between the spline and the corresponding ground truth position.

represents the route of about 25 kilometers held in the city of Nantes (France).

The analysis of the error have been made with the points where the GPS-RTK is within an error of less than 3 cm. The number of points compared in this dataset is 2007 along 25.13 Km. Figures 8 and 9 show the lateral distance and orientation errors respectively in form of accumulated and differential histogram. As we can see, about 95% of the measurements lie under 15 centimeters and 5.0 degrees. Figure 10 shows the lateral error and linear speed vs. distance travelled. As

we can see, there are several outliers with lateral errors of 3, 1.3, 3.6 and 3.1 meters respectively. Those errors are normally located in parts of the trajectory with sinuous geometry and/or low quality of the GPS signal. Those are parts that produce high stress in the connection points between sub-maps due to a combination effect of odometry error and weak global position attraction forces. Figure 11 shows the orientation error and angular speed vs. distance travelled. The orientation of the ground truth is estimated by increments of the position, that could introduce some error. We also can observe that the parts of the trip with higher angular speeds produce more orientation errors, after some analysis we have observed that those errors in orientation are specially located near the connection points,
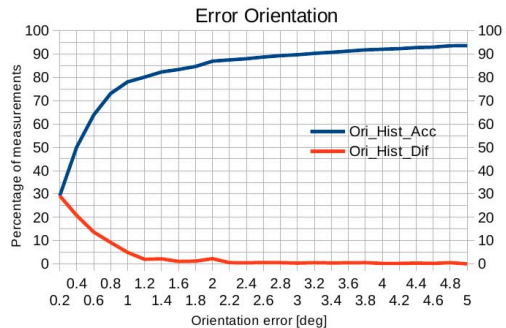

Fig. 9. Accumulated and differential histograms of the orientation error

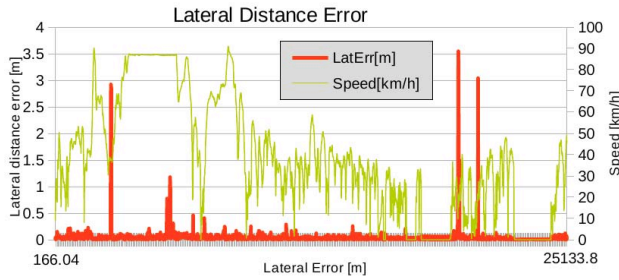reinforcing the idea that the stress in between the sub-maps should be relaxed.

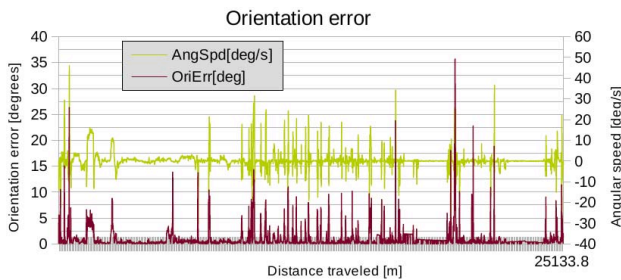Fig. 10.   Lateral error and linear speed during map-building along the trip

Fig. 11.   Orientation error and angular speed during map-building along the trip

Regardless of the outliers, the average of the lateral error along the full path is 4.3 cm. and the average of the error in orientation 1.74 degrees in 2007 measurements distributed non-uniformly in 238 sub-maps. As mentioned above we only compare points in the maps that have good RTK-GPS position measurement.

V.   CONCLUSIONS

In this paper we have presented a multi-map positioning optimization based on a relaxation technique that uses virtual forces between a global positions generated by a Kalman filter (EKF) fusioning low-cost GPS and odometry data, and a path of points generated by 2D-SLAM while building the maps. The force strength coefficient is modelled through the covariance of the EKF solution.

With this approach we obtain several advantages:

- Efficiency. The method for building the maps is fast, as it uses 2D occupancy grid local maps which does not contain too much information.

- Scalability. The data processed at any of both phases, map building and localization, is limited independently of the length of the trip as only one sub-map is loaded at a time.

- Precision. As explained in the experimental results, the optimization performed on the sub-maps positions minimizes the error obtained between the map path and the global path, providing a set of maps ready to be used for localization of the vehicle at centimetric

level, resting to solve the problem of the outliers on the *connection points*.

The method has been put under test along a trip of about 25.13 km without interruption in the town of Nantes (France) providing enough information to evaluate the quality of the map positioning with satisfactory results.

VI.   FUTURE WORK

As explained in paragraph A, Section III, forces between maps appear due to inexactitude of the map building process. A future improvement of this technique will be trying to remove the stress between consecutive sub-maps absorbing those errors by re-computing the local position of the connexion points. This could improve the positioning of the maps, as well as, making the map switching smoother during the localization phase. Another improvement of the method could be to update the global path and map path each time we re-visit a particular area and the precision on the global EKF position is better than the previous update. This improvement may be due to a more appropriate satellites constellation positioning. In that case, a relaxation process to re-position the sub-maps would be applied improving the precision of the localization in that area.

REFERENCES

[1]   N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian and M. E. Munich, The vSLAM Algorithm for Robust Localization and Mapping. IEEE Int. Conf. on Robotics and Automation (ICRA), 2005, p. 24-29.

[2]   J. Engel, T. Schps and D. Cremers, LSD-SLAM: Large-Scale Direct Monocular SLAM. In proceedings of European Conference on Computer Vision (ECCV), 2014.

[3]   D.C. Herath, S. Kodagoda, and G. Dissanayake. New Framework for Simultaneous Localization and Mapping: Multi Map SLAM. IEEE Int. Conf. on Robotics and Automation (ICRA), 2008. p. 1892-1897.

[4]   W. Churchill and P. Newman, Continually Improving Large Scale Long Term Visual Navigation of a Vehicle in Dynamic Urban Environments. IEEE Intelligent Transportation Systems Conference(ITSC), 2012. p. 1371-1376.

[5]   C. Beall, F. Dellaert, Appearance-based Localization across Seasons in a Metric Map. 6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV), 2014.

[6]   P. Biber and T. Duckett. Dynamic Maps for Long-Term Operation of Mobile Service Robots. In Proc. of Robotics: Science and Systems (RSS), 2005.

[7]   H. Zhao, M. Chiba, R. Shibasaki, X. Shao, J. Cui, H. Zha, SLAM in a Dynamic Large Outdoor Environment using a Laser Scanner. IEEE Int. Conf. on Robotics and Automation (ICRA), 2008. p. 1455-1462.

[8]   F. Chanier, P. Checchin, and C. Blanc, L. Trassoudaine. Map fusion based on a multi-map SLAM framework. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), 2008. 2008. p. 533-538.

[9]   Suger, B. Diego Tipaldi, G., Spinello, L., and Burgard, W. An approach to solving large-scale SLAM problems with a small memory footprint. IEEE International Conference on Robotics and Automation (ICRA), 2014. p. 3632-3637.