NAPS: a Nomadic and Accurate Positioning System

Virgile Daugé, Sylvain Contassot-Vivier, Laurent Ciarletta

NAPS: a Nomadic and Accurate Positioning System

Virgile Daugé[1], Sylvain Contassot-Vivier[1], and Laurent Ciarletta[1]

Abstract— **Providing accurate pose estimation over time in various and unknown environments is a key component to the realization of autonomous missions where an absolute positioning system (like GPS) is not available. A lot of work has been done in that field where Simultaneous Localization And Mapping systems (SLAM) are still highly environment-dependent and computationally expensive. However, a major issue in SLAM solutions is the drift that appears in their odometry, due to error accumulation in the successive pose estimates. This may imply significant errors in the environment reconstruction or mission success.**

In this paper, a Nomadic and Accurate Positioning System (NAPS), based on the collaboration of mobile robots and their mutual sensing, is proposed. The originality and advantage of our solution is to provide a motion capture system with the nomadic ability. The result is environment-independent, computationally efficient, accurate and mobile. The system follows one or several mobile robots (explorers) and provides accurate positioning all along their mission.

The NAPS is described as well as its moving process, and its accuracy is confirmed through a series of experiments, validating the whole approach.

I. Introduction

One of the major issues in UAVs (Unmanned Autonomous Vehicles) is to obtain and maintain very accurate localization and orientation, especially for indoor contexts. Such information is essential in many tasks, either to ensure the success of the task (actions), the validity of the results (measurements), or to avoid costly post-processing phases of acquired data.

In common approaches of SLAM (Simultaneous Localization And Mapping), both a map/representation of the environment along with the relative poses of one or several UAVs in that environment are regularly updated, in real-time when possible. However, those approaches suffer from two major issues: the pose estimates accuracy and the drift in the estimates along the path followed by the UAV.

Errors in pose estimation are directly related to the sensors that are embedded in the UAV to be tracked. In most cases, there is an IMU (Inertial Measurement Unit) that provides gravity and acceleration information. Also, cameras are often used to complete the positioning system. However, IMUs can only provide relative information and are prone to errors, and the cameras require expensive image processing to find elements (landmarks) that allow a better position estimate. Moreover, cameras are environment dependent as they require luminosity and clear atmosphere.

The way to integrate measurements and uncertainties often use Bayesian approaches to maintain correct estimates of the robot pose and environment map. However, such solutions are not sufficient to avoid the drift and its accumulation that occur along the path of the tracked UAV.

Université de Lorraine, CNRS, LORIA, F-54000 Nancy, France. virgile.dauge@loria.fr

In fact, the positioning problem in classical SLAM solutions mainly comes from the fact that the exploring UAV tries to evaluate its own position from its embedded sensors. Yet, the best technique to obtain very accurate position measurements is the one used by the land surveyors, with external and still sensors. In our case, the principle is to use a small set of additional UAVs (beacons) with specific sensors allowing them to track each others. When tracking the UAVs of interest (mission UAVs), the beacons stay immobile. Otherwise, they move to the next place of interest.

In this paper, we propose to take advantage of such a solution by designing and implementing an external tracking system that has the ability to be nomadic. This solution is called **NAPS**: ***N****omadic and* ***A****ccurate* ***P****ositioning* ***S****ystem.*

In the next section, a brief overview of works related to pose estimation is provided. Then, in Section III is fully described the NAPS concept as well as its implementation aspects. Experimental validation is given in Section IV, demonstrating the interest of NAPS.

II. Related works

To the best of our knowledge, the originality of NAPS leads to a lack of similar studies in the literature. However, the proposed objectives and use cases are similar to the Simultaneous Localization and Mapping (SLAM) problem. According to [1], SLAM is not solved yet, despite the extensive research in the field, as long as the robustness and the environment dependence problems (lightning, structure,...) are not addressed. Modern formulations of SLAM are often broken down into two parts : *front-end* and *back-end*. The front-end produces estimations from sensors data while the back-end merges and refines those estimations, often a posteriori. In this paper, we consider our contribution and the other odometry solutions in the front-end category, although boundaries between front and back-end are sometimes fuzzy.

A. Back-end

The way to deal with uncertainty in measurement and sensor combination in robotics has received much attention, and Bayesian solutions like Kalman filtering [2] are widely used as part of current SLAM solutions. Furthermore, some state of the art libraries are available, providing optimal solutions formalized as factor graphs computing *maximum a-posteriori estimations* [3], [4].

The drift and poor estimates can be partially corrected either by loop closure (or visual place recognition [5]) or by offline optimization tools/frameworks [6] at the expense of high computation cost.

B. Front-end

In robotics, multiple and heterogeneous sensors are commonly used. By providing very dense geometric data, namely point-clouds, the Lidar allowed some valuable results [7]. Nevertheless, as stated in [8], despite the extensive use of Iterative Closest Point (ICP) and point-cloud registration in general, a consequent work is dedicated to choose, organize and tune different parts of existing software to make it work in a particular context. Research works based on Lidars have lost popularity in recent years, probably with an exception in the field of autonomous vehicles that often operate in a different context, i.e., in more controlled environments.

The trend is currently on the side of cameras. Indeed, they are quite cheap and easy to embed, even in small UAVs. Some work has been done with RGB-Depth cameras [9], but most of the research on UAVs odometry is now dedicated to visual-inertial odometry. By merging camera and IMU data, those approaches yield promising results [10], [11]. However, these algorithms are still computationally expensive and environmentally dependent as they require a significant and constant illumination, as well as sufficiently discriminating textures in the environment. Also, the tracking frequency of those techniques ($\approx$20Hz) does not always allow to get an accurate tracking of rapid moves that often occur with UAVs, especially the flying ones.

C. Performance Evaluation

Performance evaluation and benchmarking is complex and not often done in SLAM field due to closed-source implementations of many algorithms as well as a lack of common datasets. Fortunately, this is not the case for visual-inertial odometry, where several datasets [12], [13] and even benchmarking tools [14], [15] are available. Also, some works take into account computational cost to strengthen quantitative comparisons between algorithms [11]. In Section IV, these tools are extensively used to get a fair evaluation of NAPS.

III. NAPS POSITIONING SYSTEM

As mentioned in the introduction, there are two major problems with mobile positioning systems used in SLAM. The former is the tracking accuracy that is generally limited and the latter is the drift that occurs during the moves of the explorer. Moreover, such systems often require a large amount of computation to update the explorer position (often based on acquired environment data).

In this context, we propose an original system that combines the accuracy of static positioning systems (used by land surveyors or for motion-capture) with the mobility of collaborative UAVs.

The interest of the NAPS approach is to preserve a very accurate positioning of the explorer with very few computations that can be made on-board. Moreover, NAPS is much less environment dependent than existing systems as it does not rely on any specific hypothesis over its features (structure, textures properties, light,...).

A. Concept

As mentioned above, NAPS is inspired by the techniques used by the land surveyors who manually place artificial beacons in the environment.

The main concept of NAPS is to use a small set of additional UAVs that forms a positioning system for the explorer. In this system, those UAVs are used as reference beacons that are themselves accurately localized in the initial map. Then, the relative position of the explorer with respect to the beacons is used to accurately deduce its global map position in real-time. At each time, one element is used as the reference position and all other elements are localized relatively to this reference.

The only constraint to build the NAPS is to have some sensors that allow to evaluate relative positions between elements. As the hardware implementation of our prototype is described in the next subsection, we assume here that each element can get its relative position with respect to any other element in the system.

Let's describe how the system operates. At the initialization of the system, the beacons are disposed to form an area of a few meters diameter (depending on the sensors range). Then, one of them is set as reference and its position and orientation in the global map is computed. All other elements are localized with respect to that reference thanks to the sensors. According to those local positions, they can be accurately localized in the map by the following computation: let's denote ${}^{M}T_R$ the position/orientation (pose) matrix (4×4 homogeneous transform matrix) of the reference in the map, ${}^{L}T_R$ the measured relative pose matrix of the reference in the local coordinate system, and ${}^{L}T_D$ the measured relative pose matrix of any other device in the local system. Then, the global pose matrix ${}^{M}T_D$ of that device in the map can be deduced by:

$$ {}^{M}T_D = {}^{M}T_R . {}^{L}T_R^{-1} . {}^{L}T_D \qquad (1)$$

as depicted in Figure 1.

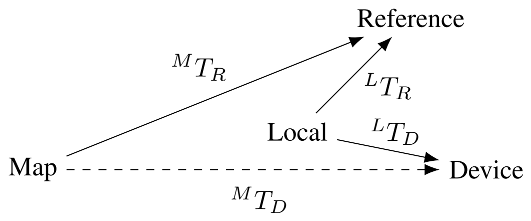

Fig. 1: Global pose deduction from local and reference poses.

So, at this point, the explorer can move inside the geometric volume described by the beacons. Its relative position can be accurately updated in real-time by Equation 1. This is the *phase 1* of the NAPS process. It is worth noticing that when the explorer moves, it is located by a static system, so no additional error is induced. Moreover, it would be possible to use and track several explorers within the beacons area.

Once the explorer has scanned the area, it stops (saving its energy) and one (or several) of the UAVs is set as

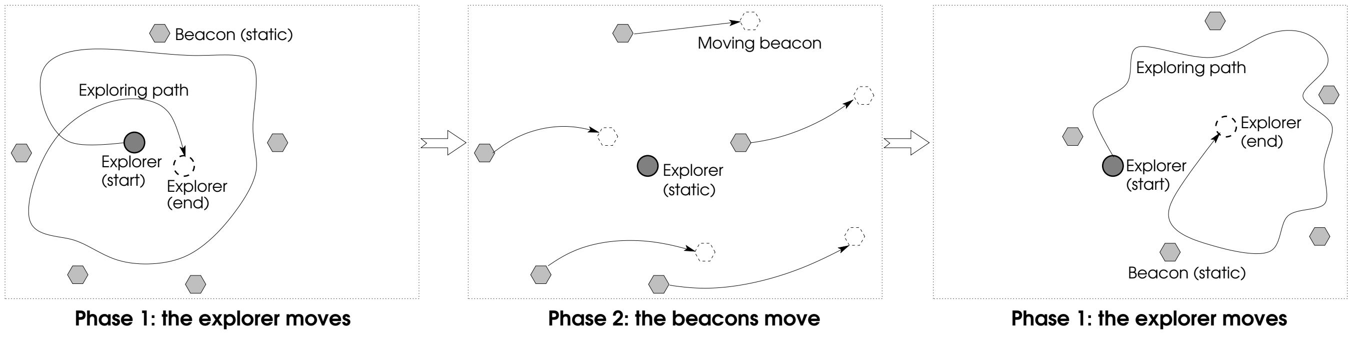

Fig. 2: NAPS process: explorer in action tracked by the beacons (left and right), beacons moving to another area (center). The dashed rectangle (not used in the process) represents the same part of environment. It is drawn to help the understanding.

the new reference (it can be an explorer). Then, all other elements can move to other positions in order to change the exploration area. Once they have moved, their relative and global poses are computed thanks to the current reference. This is the *phase 2* of NAPS. Once this is done, the explorer can begin to scan the new area (phase 1). This process, depicted in Figure 2, can be repeated as many times as needed to cover the zone to be explored. Here again, it is worth mentioning that positioning error evolves only when the set of beacons moves. Hence, the position accuracy of the explorer stays stable during its scans. Moreover, the accuracy loss that occurs when moving the beacons may be reduced by limiting the number of beacons that move at the same time. In fact, if one beacon is moved at a time, the error made in the evaluation of its new position should be smaller than when using only one reference beacon. The choice of the number of moving beacons at each step of the process directly depends on the accuracy and range of the sensors used to measure the relative positions. Thus, with very accurate and wide-range sensors, only one beacon may be used as reference while the others move, as in Figure 2. Also, whatever the number of beacons are moved, the accuracy loss can be restrained by a temporal integration of real-time measures. The idea is to compute the average of a set of measures obtained within a temporal window (typically 1s). Such integration is pertinent for the beacons at the end of phase 2 since they do not move during phase 1.

As can be seen, the major difference with classical land surveyor systems is that our beacons are mobile. Also, compared to SLAM techniques, our beacons play the same role as detected features (landmarks) in the environment, both drastically reducing measurement errors and eliminating the possibility of errors in data associations that are required in SLAM, but not in NAPS.

The counterpart of those advantages is that NAPS makes use of several beacons (at least two to provide accuracy and nomadism), which means additional UAVs (at least one if the explorer plays also the role of a beacon). However, those UAVs need only to carry the positioning sensors. So, they can be relatively small and simple, contrary to the explorer that may embed many sensors of different kinds, or even actuators, to fulfill its mission.

B. Implementation

In order to demonstrate the validity and interest of the NAPS approach, we have implemented a prototype system.

As NAPS requires additional UAVs as beacons (at least one), their price may condition the choices of UAV models and sensors to be used. This may affect the final accuracy of the system as well as its autonomy. Indeed, this is just a trade-off between cost and performance. However, a more fundamental constraint that must be taken into account is the embedding ability of the sensors. As our prototype is a proof of concept, that aspect was not a priority, although the hardware we have used can be technically embedded on UAVs with electric power adaptations. A last constraint that is mandatory for building the NAPS is the ability of the hardware to get unique identifiers for all the devices, in order to distinguish the UAVs.

According to all those requirements, we have chosen the Vive Pro tracking system [16] as it provides accurate measures (an evaluation is provided in Section IV) and the cost of the elements (trackers and lighthouses) is quite limited. This system is based on one or two *lighthouses* that emit lasers, and on *trackers* that receive the lasers on different small sensors, allowing for the computation of their relative pose according to a local system. Indeed, the system provides a transforms tree of all connected objects in real-time.

In our context, the major issue encountered with this system is that it is designed for static use-cases (virtual reality experience in a static area), i.e., the lighthouses are used as static references, and only the trackers are meant to move. So, we developed additional software in order to add the ability to move the lighthouses. In fact, the Vive API does not include any lighthouses recalibration on the fly, so the measures are no longer valid when moving one of them. As a consequence, we implemented a software environment that allowed us to recalibrate the system after each move of the lighthouses beacons. Obviously, for release versions of the system, such problem would be considered at the design level so that moving any beacon would not require such additional recalibration step.

Finally, we obtain a prototype that allows us to experimentally validate the NAPS approach.

IV. Experiments

The particularity of NAPS, namely the use of other sensors than cameras and IMUs, prevented the use of datasets commonly referenced by the community. Therefore, some datasets have been specifically built, keeping in mind that accurate ground-truth acquisition is a key determinant for the validation of the approach. In this context, we have used a motion capture system composed of 8 *Optitrack Prime 17w* infrared cameras providing a 6DOF tracking at 250 Hz in a 5×4×3 meters capturing volume with a sub-millimeter accuracy on pose estimation (calibration feedback). Another constraint has been the limited number of devices at our disposal (2 lighthouses and 3 trackers). It is worth mentioning that the motion capture system illuminates the whole area with infrared light. This can disturb the *VIVE* tracking system which is based also on infrared communication between devices. However, although this setup may disadvantage NAPS a little by reducing its accuracy, preliminary tests have shown that such perturbations were not significant enough to invalidate the experiments.

A. Protocols

To demonstrate the validity of our approach, two distinct experimental protocols have been designed. Both of them use our moving process in two phases, described in Section III, alternating between explorer(s) tracking and beacons repositioning. Also, they are both designed according to the limited ground-truth (Optitrack) acquiring area at our disposal (covering $\approx 20m^2$ while one lighthouse covers $\approx 30m^2$). Due to that limitation, almost all beacons where located out of the ground-truth area, preventing us from getting that information for all devices. This explains why we only evaluate one trajectory (that could be the explorer's one). Indeed, comparing only the explorer position estimates is sufficient to validate our system as any error on the beacons positions would imply an error on the explorer.

As our implementation of NAPS uses the Robot Operating System (ROS), pose estimates are computed and published on topics for all beacons and explorers at a 150 Hz rate. In addition, a ROS node writes the ground-truth and pose estimates immediately upon their publication. Also, after each move of the beacons, their positions are updated with a temporal integration of 1s, i.e., around 150 measurements to obtain one estimate.

a) Random move: The objective of the first protocol is to cover the transition phases of an exploration process, namely the local repositioning (translations and rotations) of the system. In the first phase, the explorer moves freely in the area formed by the NAPS beacons in order to fulfill its mission, then it lands anywhere in that area. In the second phase, all the beacons are randomly re-arranged in the same area. It is important to notice that all beacons are moved, effectively adding estimations errors to the system.

In the context of this experiment, the explorer is used as the reference and all the beacons move to form a new tracking area. However, the use of the explorer as reference is only a practical choice but it does not induce any loss of generality of the approach since all the UAVs in the system are tracked with the same accuracy.

The results of that experiment are labeled **RMi**. For each experiment, the raw concatenation (no post-treatment) of the consecutive steps (phases 1 and 2) provides time-stamped trajectories of one UAV (the explorer) for both ground-truth and NAPS pose estimates.

b) Guided move: This second protocol is designed to reproduce a complete guided exploration process, i.e. moving the system in a specific direction. The mechanism is quite the same, except that instead of moving randomly, the beacons are moved to a neighboring area, in a given direction. In order to preserve the accuracy of the system when the beacons move to another area, the explorer must land near the border of the current area that is common to the aimed area. Then, the second phase can begin.

This iterative process is repeated several times in order to evaluate the error accumulation along a realistic path length (several tens of meters) requiring several tracking areas. Those experiments are labeled **GMi** in the following.

B. Results

In order to compare our results with state of the art (visual-inertial) odometry, we need to use the same metrics. What we need to quantify is the error between the ground-truth trajectory and the trajectory estimated by NAPS. In Table I are gathered the general features of the datasets collected during our experiments: 4 random moves (RM) and 4 guided moves (GM). Also, a high-speed test (HST) is added, in which the robustness of NAPS to fast moves of the explorer (above classical use) is tested. The differences between the number of ground-truth poses and the number of NAPS poses come from different acquisition frequencies.

	Length (m)	NAPS moves	Ground-truth poses	NAPS poses
RM1	40	2	45k	22k
RM2	50	3	46k	28k
RM3	53	2	27k	17k
RM4	52	2	27k	17k
GM1	51	1	26k	16k
GM2	74	4	65k	32k
GM3	40	2	33k	8k
GM4	176	9	125k	50k
HST	117	2	22k	10k

TABLE I: Datasets collected during the experiments: total trajectory length, number of NAPS moves and number of recorded poses for both ground-truth and NAPS.

The open Toolbox for trajectory evaluation [15] provided by the *University of Zurich and ETH Zurich Robotics and Perception Group* has been used to compute translation and orientation root mean square errors (RMSE) along the complete trajectories, as well as to produce KITTI style analysis [14], giving the errors for different segment lengths along the trajectories.

a) Raw data: As a first accuracy overview, the overlapped trajectories measured by Optitrack and NAPS are given in Figure 3 and 4, respectively for a random move experiment and for a guided move one. As the trajectories are initially expressed in different coordinate systems (Optitrack and Vive), the alignment toolbox (sim3 type) has been used to obtain the overlaps. As can be seen, Optitrack and NAPS trajectories are pretty close to each other in both cases.

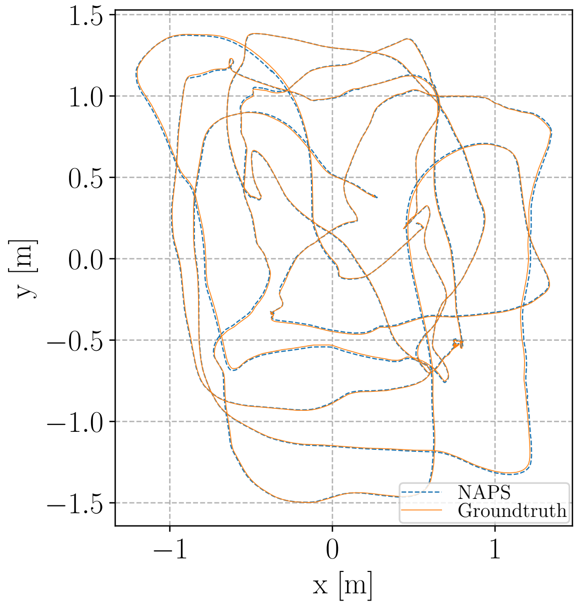

Fig. 3: Trajectory from the RM2 experiment (top view)

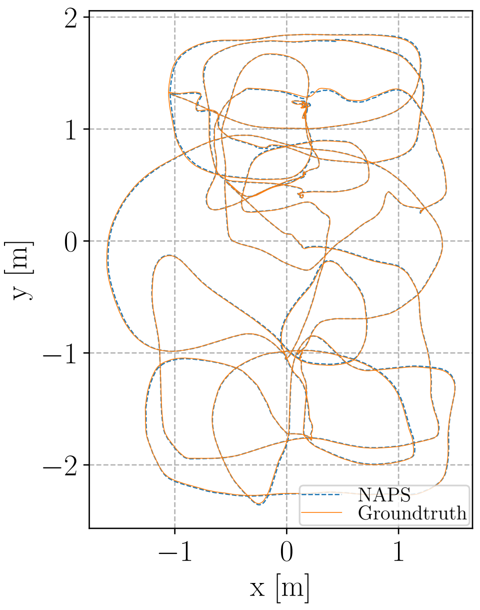

Fig. 4: Trajectory from the GM2 experiment (top view)

In order to get a closer look to the accuracy of our system, translation and orientation errors of the NAPS pose estimations in the GM2 experiment are plotted in Figure 5, for all dimensions or axes. Other experiments obtain similar plots, but in GM2, there is a translation error peak is visible in step 0 for one x value. Indeed, this corresponds to a temporary loss of sight between explorer and beacons during manual testing. However, such problem can be avoided by moving algorithms ensuring line of sight. All other measures are coherent and stay in an error interval around [-40:30] mm.

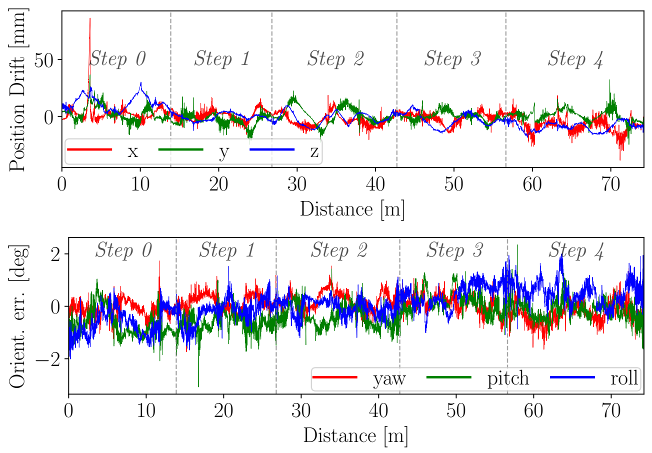

Fig. 5: Pose estimation errors over the whole GM2 experiment. Each vertical separation corresponds to a transition phase in which all beacons are moved (phase 2 of NAPS).

b) Absolute error: The Root Mean Square Error metric is often used for trajectory error analysis. It offers the benefit to summarize the errors on the whole trajectory with only one value. In Table II are presented the RMSE obtained in our experiments. As already mentioned, a direct comparison of our results with those obtained by other algorithms is quite difficult as the datasets are different. However, the results presented in several odometry benchmarks [13], [11] suggest a higher order of the RMSE metrics. For instance, the best results reported in [11] are 0.03 m for a specific dataset whereas it is the maximal RMSE value observed for NAPS, with the longest trajectory (176 m).

	RM1	RM2	RM3	RM4	GM1	GM2	GM3	GM4	EST
Trans. (m)	0.011	0.010	0.016	0.015	0.011	0.011	0.010	0.031	0.014
Ori. (deg)	1.288	0.936	1.090	0.894	0.973	0.939	1.492	1.669	1.824

TABLE II: Translation and orientation RMSEs for NAPS.

c) Relative Error: Another common approach for trajectory analysis is the KITTI style analysis [15]. Its principle is to split the whole trajectory into sections of given lengths, to align the couples of ground-truth and estimate values for each section, and finally to compute the relative error for each section. This technique has been used to produce Figure 6, in which the relative error distributions in translation and yaw are depicted for several section lengths along every trajectories obtained in our entire set of experiments. It can be seen that NAPS median errors are 29 mm and 0.5 ° for a 35 m section length. As a comparison, the best results reported in the literature [11] are obtained by the *vinsmonolc* method [10], with a median translation error of 150 mm the same path length. Although the datasets are not the same, the diversity, complexity and generality of our experiments make them representative of general use-cases.

So, we can conclude that according to the limited environment dependency of our NAPS prototype, its high quality sensors for pose measurements and its update rate of 150 Hz allowing time integration, similar results should be obtained in most real-case contexts. One of our future works will consist in testing our prototype in the real context of a mine.

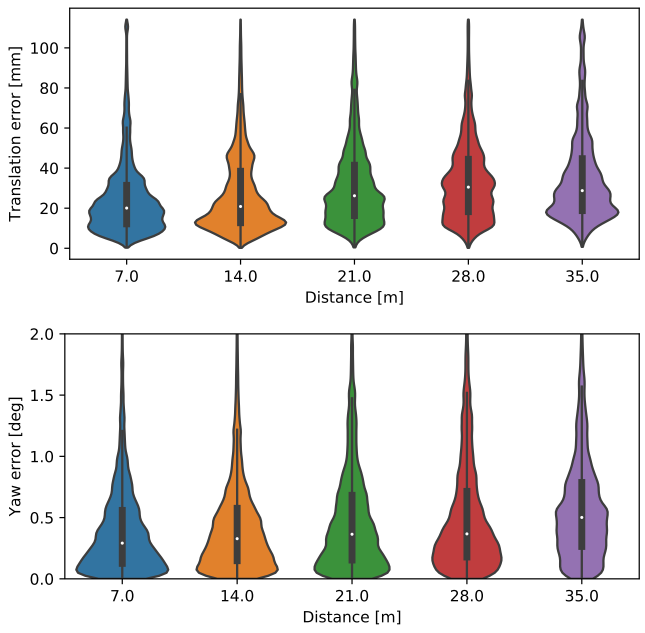

Fig. 6: Translation (top) and yaw (bottom) errors over the 9 datasets.

V. Conclusion

A nomadic and accurate positioning system (NAPS) has been presented. Its originality is to combine advantages of static localization systems with mobility provided by UAVs. Indeed, the system is composed of two sets of UAVs. One set contains positioning beacons and the other set contains one or more explorers.

The interest and validity of the system has been practically demonstrated by a series of experiments, showing the very good accuracy of pose estimates of any of the elements (beacons and explorers). The NAPS positioning errors stay in the order of the centimeter along quite long paths (<45 mm for 175 m), which is much smaller that existing solutions.

Although the NAPS results are very promising, several tracks for further works can be identified. In particular, our experiments have pointed out that the way the beacons are moved may have an impact over the overall accuracy. So, we plan to study different strategies for moving the beacons in order to preserve accuracy while optimizing the energy consumption of all elements. Also, the number of beacons is an important parameter. A minimum of two beacons is necessary. In an extreme case, only one beacon may be required if the explorer plays also the role of a beacon. However, such reduction of the number of beacons would induce significant constraints over the consecutive moves of the system, probably implying more steps in the moving process. This issue deserves a specific study in itself. Another work of interest will be to test the behavior of NAPS in different environmental conditions (smoky/dusty air,...). Finally, we would like to enhance our current hardware prototype by embedding the lighthouses on UAVs. This will make our system fully autonomous and will allow us to conduct real-case explorations with additional sensors (Lidar,...) for full 3D reconstructions.

Acknowledgment

This work was supported by the Interreg V - Grone - Project 024-4-09-076 of the European Union and the Region GrandEst. The experiments presented in the paper have been done in the Creativ'Lab at Loria.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, dec 2016.

[2] S. Y. Chen, “Kalman filter for robot vision: A survey,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, 2012.

[3] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G2o: A general framework for graph optimization,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.

[4] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “ISAM2: Incremental smoothing and mapping using the Bayes tree,” *International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[5] S. Lowry, N. Sunderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, “Visual Place Recognition: A Survey,” *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, feb 2016.

[6] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, “maplab: An Open Framework for Research in Visual-inertial Mapping and Localization,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1418–1425, 2017.

[7] J. Zhang and S. Singh, “LOAM: Lidar Odometry and Mapping in Real-time,” *Robotics: Science and Systems X*, 2014.

[8] F. Pomerleau, F. Colas, and R. Siegwart, “A Review of Point Cloud Registration Algorithms for Mobile Robotics,” *Foundations and Trends in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.

[9] C. Kerl, J. Sturm, and D. Cremers, “Robust odometry estimation for RGB-D cameras,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013, pp. 3748–3754.

[10] T. Qin, P. Li, and S. Shen, “VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1–14, 2018.

[11] J. Delmerico and D. Scaramuzza, “A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2018, pp. 2502–2509.

[12] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The EuRoC micro aerial vehicle datasets,” *International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[13] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stuckler, and D. Cremers, “The TUM VI Benchmark for Evaluating Visual-Inertial Odometry,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 1680–1687, 2018.

[14] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the KITTI vision benchmark suite,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

[15] Z. Zhang and D. Scaramuzza, “A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2018, pp. 7244–7251.

[16] D. C. Niehorster, L. Li, and M. Lappe, “The accuracy and precision of position and orientation tracking in the HTC vive virtual reality system for scientific research,” *i-Perception*, vol. 8, no. 3, pp. 1–23, jun 2017.