SLAM-Based Return to Take-Off Point for UAS

Daniel Bender[1(✉)], Wolfgang Koch[1], and Daniel Cremers[2]

[1] Department Sensor Data and Information Fusion,
Fraunhofer Institute for Communication,
Information Processing and Ergonomics FKIE, Wachtberg, Germany
{daniel.bender,wolfgang.koch}@fkie.fraunhofer.de
[2] Computer Vision Group, Technical University of Munich (TUM),
Garching, Germany
cremers@in.tum.de

Abstract. Up to the present day, GPS signals are the key component in almost all outdoor navigation tasks of robotic platforms. To obtain the platform pose, comprising the position as well as the orientation, and receive information at a higher frequency, the GPS signals are commonly used in a GPS-corrected inertial navigation system (INS). However, the GPS is a critical single point of failure for unmanned aircraft systems (UAS). We propose an approach which creates a metric map of the overflown area by fusing camera images with inertial and GPS data during normal UAS operation and use this map to steer the system efficiently to its home position in the case of an GPS outage. A naive approach would follow the previously traveled path and get accurate pose estimates by comparing the current camera image with the previously created map. The presented procedure allows the usage of shortcuts through unexplored areas to minimize the travel distance. Thereby, we ensure to reach the starting point by taking into consideration the maximal positional drift while performing pure visual navigation in unknown areas. We achieved close to optimal results in intensive numerical studies and demonstrate the usage of the algorithm in a realistic simulation environment and the real-world.

Keywords: UAS · Drone · SLAM · Path planning · Navigation

1 Introduction

Two critical aspect of unmanned aircraft systems (UAS) are the flight control and the navigation in large outdoor scenarios. For a long time these tasks were almost exclusively based on readings from inertial sensors corrected by GPS measurements in a so called inertial navigation system (INS). In contrast to the manned aviation, the navigation systems in most UAS are not redundant. A long-lasting GPS outage is for nearly all currently available UAS a major problem. Without a pilot takeover, most systems initiate an emergency landing

S. Lee et al. (Eds.): MFI 2017, LNEE 501, pp. 168–185, 2018.
https://doi.org/10.1007/978-3-319-90509-9_10

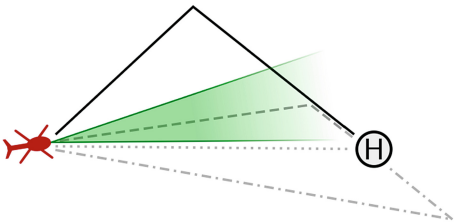


Fig. 1. The UAS started at the heliport and flew along the solid path. At its current position the GPS fails and the platform starts an autonomous homing. The safest approach would follow the exact path the UAS was using to reach its current location. But this path can be quite long compared to the direct connection (dotted). However, a positional drift may occur, resulting in missing the heliport (dotted/dashed). A safer approach aims to intersect the previous path close to the goal position (dashed). This is the main concept of the proposed path planning approach.

in this situation. The critical aspect is that the maneuver is most likely initiated at the current UAS position, regardless of the suitability of the area for a safe landing procedure.

Nearly all UAS are equipped with a camera for the real-time visualization of the observed area. The proposed approach uses these images in combination with the INS measurements for an efficient homing strategy. During normal operation the data from all sensors is fused to create a metric map of the area with a simultaneous localization and mapping (SLAM) approach [3]. In the case of a GPS outage, only the camera and the previously generated map are used to return to the take-off position. The main contribution of this work is a path planning procedure that uses shortcuts through unobserved areas to minimizes the travel distance. In doing so, it is ensured to reach the goal by taking into consideration the maximal drift of the pure visual navigation (Fig. 1). This is an extended and revised version of our paper presented at the MFI 2017 conference [2].

2 Related Work

Visual homing of an autonomous systems describes the process of the system guiding itself to a previous location on the basis of visual sensor inputs.

A group of approaches performs a direct association of visual patterns and steering commands without a world model. Examples are the road following by Pomerleau [13] and the navigation along forest trails by Giusti *et al.* [6]. Both approaches were realized with a neural net. A homing based on scene familiarity has been proposed by Nelson. The procedure looks for the best match of the current view to a set of previous collected images, saved with associated directions of movement [11].

Other approaches are based on maps that store the position of objects and locations in a common reference frame. Errors are incorporated in the map by noisy sensor measurements and moving objects. The discrepancies between the map and the

actual environment may be a problem for the path planning [9]. These uncertainties have nicely been covered in the work of Valencia *et al.* [17] by using the Pose SLAM graph directly as belief roadmap to perform a collision free path planning along the route with the lowest accumulated robot pose uncertainty. A method using a graph of poses generated with a bundle adjustment as basis for the path planning has been proposed in [15]. All these approaches consider only the already traversed trajectory as feasible and obstacle free, which is a valid and useful assumption, especially for the ground based navigation.

In contrast, the in the following processed scenario considers an UAS with a downward looking camera at a fixed altitude without obstacles. As a result, the usage of shortcuts through previously not visited areas is possible. This navigational task has not been covered before in literature. The proposed algorithm works analog to the navigational abilities of dogs [16]. Chapuis validated in experiments, that dogs have a metric representation from previous incomplete explorations. They use shortcuts between known areas, whereby they perform a safety strategy to make a correction in the case of bad direction estimate [4]. The main idea of this concept, already transfered to the UAS context, is visualized in Fig. 1. One major prerequisite for safe shortcuts by navigating an UAS through previously unexplored areas is the knowledge of the drift for pose estimation from pure visual odometry. A comprehensive theoretical analysis has been performed by Liu *et al.* [8]. They state that the drift is a random process that will not increase linearly and in some situations even may decrease. However, they declare that the end-point drift of visual odometry algorithms is generally between 1° and 5° of the traveled distance.

3 Problem Description

In this section, the fundamental concept of the proposed UAS homing approach is explained. During normal operation, the INS-based LSD-SLAM is used to build a map in the form of a metric and georeferenced 3D point cloud of the observed environment and perform a self-localization at the same time [3]. The integration of measurements from an INS in the LSD-SLAM algorithm eliminates the drift and generates metric depth map estimations for the processed images. As a consequence, the generated point clouds of the observed areas are also metric and furthermore, due to the utilized GPS measurements, georeferenced. On the basis of real time kinematic (RTK) corrections, the created maps are accurate in the centimeter range. In contrast to other computational intensive camera-based algorithms, the results of the approach are generated in real-time and thus suitable for the UAS navigation. The approach generates a factor graph that consists of keyframes and image constraints between them (Fig. 2).

In the case of an GPS outage, the platform returns to its start position by only using camera images and the previously generated outputs of the metric SLAM. The most secure approach would follow the previously traveled path in reverse. However, this may imply a really large detour, compared with the direct connection to the starting point. By leaving previously exploited areas,

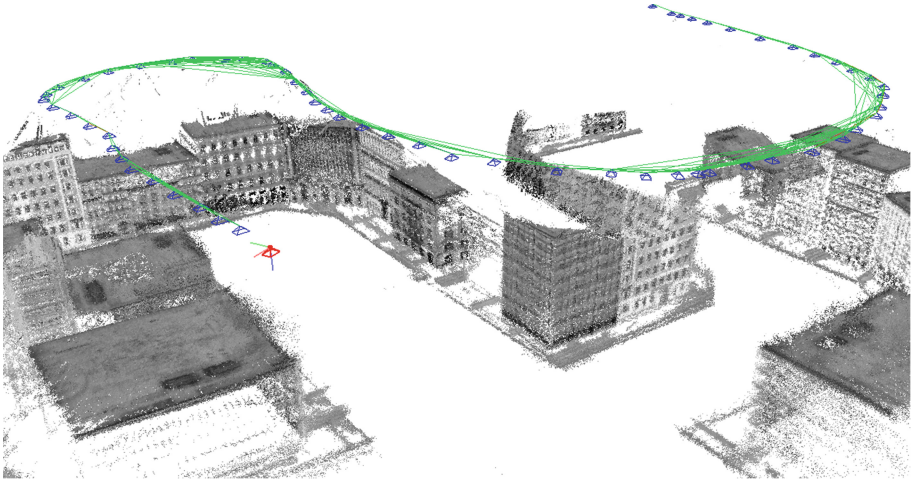


Fig. 2. Visualization of the LSD-SLAM output. The flight path of the UAS is depicted by the blue keyframes and the image constraints between them. The current camera pose is depicted in red with its coordinate system axes

the own position is estimated based on visual odometry [5]. Thereby, the relative estimates between the collected images lead to a drift in the self-localization, which in the given scenario can only be corrected by reentering an area observed in the previous map building phase.

The problem considered is the path planning for homing an UAS along a fast and at the same time safe path by only using the current camera images and the previously generated output of a metric SLAM.

4 Path Planning Using Shortcuts

By using the output of the INS-based LSD-SLAM as basis for planning, a path back to the starting position is determined under consideration of safe shortcuts. In the following the problem is defined in a more abstract representation and the algorithm to perform the path planning is formulated.

4.1 Problem Definition

In contrast to most approaches, especially the ones for ground based robots, the area contains no obstacles. Further, the planning is performed in the xy-plane. Dropping the z-coordinate is straightforward for missions which are completely performed at a fixed altitude.

The factor graph, generated by the INS-based LSD-SLAM during normal operation of the UAS, is rephrased as a 2D graph $G = (V, E)$. Thereby, the Cartesian coordinates of the keyframes describe the vertices V and each, except the first, keyframe is connected with its direct predecessor by an edge added to E.

If the current position does not coincide with the last keyframe, it is added as new vertex $\mathbf{v}_n$ to V and an edge which connects $\mathbf{v}_n$ with the vertex of the last keyframe to E. By traveling in previously not visited areas visual odometry is used to perform a self-localization. This leads to an integration drift that is bounded by a known factor in relation to the traveled distance. The latter can be transformed to the maximal angular drift α. The graph describes the path already observed during normal operation and any intersection with the edges E allows us to perform an accurate localization of the UAS by comparing the current camera image with the previously generated map.

By defining the current position as start $\mathbf{v}_s \in V$ and the first keyframe as goal $\mathbf{v}_g \in V$, the problem of UAS homing is reformulated to the search of a path between these two vertices. Thereby, the path is not exclusively bound to the edges E already in the graph, but using a shortcut needs to consider the maximal angular drift α to guarantee an intersection with one of the graph edges E. A solution needs to converge in any scenario and the distance of the traveled path should be close to the distance of the direct connection between the start and goal vertex.

4.2 Algorithm Description

Each vertex is described by its coordinates $\mathbf{v}_i = [x_i, y_i]^\top$. The shortest path between the start vertex $\mathbf{v}_s$ and the goal vertex $\mathbf{v}_g$ is the direct connection, which is part of the following ray:

$$\mathbf{r} = \mathbf{v}_s + \lambda[\mathbf{v}_g - \mathbf{v}_s], \tag{1}$$

with $\lambda \in \mathbb{R}_+$. The maximal angular drift α allows us to define a left and a right ray enclosing the drift area:

$$\mathbf{r}_l = \mathbf{v}_s + \lambda\mathbf{R}(\alpha)[\mathbf{v}_g - \mathbf{v}_s], \tag{2}$$

$$\mathbf{r}_r = \mathbf{v}_s + \lambda\mathbf{R}(-\alpha)[\mathbf{v}_g - \mathbf{v}_s], \tag{3}$$

with the rotation matrix $\mathbf{R} \in SO(2)$. Further, the Euclidean norm is used to describe the distance between two vertices by using the scalar product as follows:

$$\Phi : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_+, \tag{4}$$

$$\Phi(\mathbf{v}_i, \mathbf{v}_j) = \sqrt{[\mathbf{v}_i - \mathbf{v}_j] \cdot [\mathbf{v}_i - \mathbf{v}_j]}. \tag{5}$$

The path planning is performed shortly after the UAS looses the GPS signal. Therefore, the starting pose can be treated as accurate and the uncertainties from small errors in the position or heading may be covered by increasing the maximal angular drift slightly.

A temporary graph $G_t = (V_t, E_t)$ is created to efficiently work with the graph entities within the drift area. This graph is created from G as follows:

1. Add temporary vertices at the intersections between the maximal drift rays $\mathbf{r}_l$ and $\mathbf{r}_r$ with the edges in E.

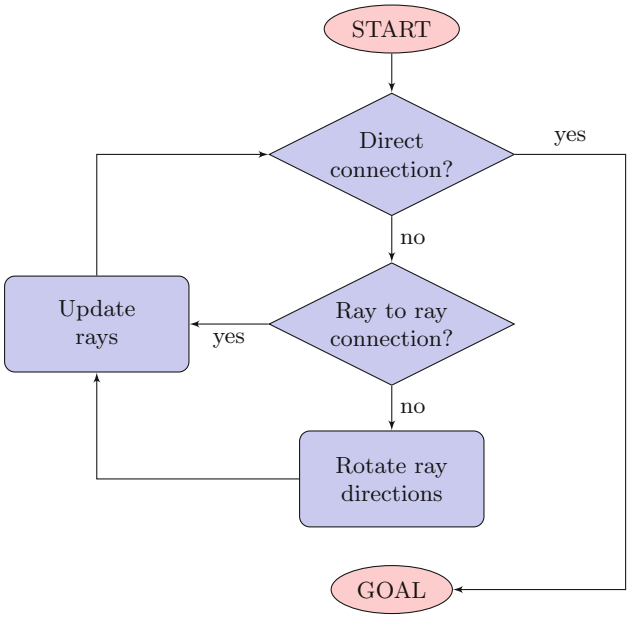
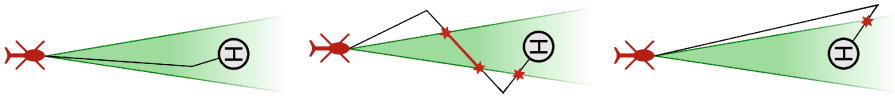


Fig. 3. Flowchart visualizing the workflow of the proposed algorithm to navigate from a start to a goal position.



(a) Direct connection (b) Ray to ray connection (c) Rotate ray directions

Fig. 4. The three cases of the proposed approach. The algorithm investigates the intersections (red stars) between the graph (black lines) and the maximal drift rays (outer green lines, spanning the green area).

2. For each new vertex, add two temporary edges connecting the vertex with the source and target vertex of the original edge it intersects.
3. Remove all vertices and edges outside the drift area. (This can be performed very efficiently by temporarily disabling the entities in a so called graph filtering.)

For each change of the target ray $\mathbf{r}$ the temporary graph G_t is updated. As a first step, the original graph G is recreated by removing the temporary vertices and edges as well as the graph filter. Then the temporary graph G_t is created as described above.

The proposed approach follows the workflow depicted in Fig. 3. The contained cases are visualized in Fig. 4 and defined as follows:

1. *Direct connection*: Within the drift area exists a path between $\mathbf{v}_s$ and $\mathbf{v}_g$ which only contains edges in E_t. By following this path the goal position $\mathbf{v}_g$ will be reached (Fig. 4a).
2. *Ray to ray connection*: There exists a valid sequence of vertices $\mathbf{s} = (\mathbf{v}_l, ..., \mathbf{v}_r)$ connecting a vertex placed on the left ray with a vertex located on the right ray only by edges in E_t. A sequence is considered as valid, if it is either close or connected to the goal vertex. Regarding the first option, a sequence is considered as close, if for all vertices $\mathbf{v}_i$ the distance to the goal vertex $\mathbf{v}_g$ is less than the distance between the start and the goal vertex (Fig. 4b):

$$\forall \mathbf{v}_i \in \mathbf{s} : \Phi(\mathbf{v}_i, \mathbf{v}_g) < \Phi(\mathbf{v}_s, \mathbf{v}_g). \tag{6}$$

   Alternatively, a valid sequence contains or is connected to the goal vertex $\mathbf{v}_g$ by edges in E_t. If a valid sequence has been found it is safe to travel and the random drift determines where the graph is intersected. The intersection becomes the new $\mathbf{v}_s$ and as a consequence the rays defined in equation (1), (2) and (3) have to be updated.
3. *Rotate ray directions*: If neither a direct nor a valid ray to ray connection exists in the temporary graph G_t (Fig. 4c), the directions of the target ray $\mathbf{r}$ as well as of the rays $\mathbf{r}_l$ and $\mathbf{r}_r$ enclosing the drift area are adapted. By entering this case for the first time with the current start vertex $\mathbf{v}_s$, all rotation angles that would rotate either the left ray $\mathbf{r}_l$ or the right ray $\mathbf{r}_l$ on a vertex in V are determined. Note that this evaluates the vertices in the original graph G. By discarding all angles, with an absolute value larger than the maximal angular drift α, it is guaranteed that the goal vertex is placed in the new drift area after the rotation is applied. The angles are sorted according to their absolute values. Now and in all following iterations of this case with the current start vertex $\mathbf{v}_s$, one angle is removed from the sorted list and the initial directions of the three rays $\mathbf{r}$, $\mathbf{r}_l$ and $\mathbf{r}_r$ are rotated accordingly.

An example of a complete path planning sequence visualizing the iterated cases is given in Fig. 5.

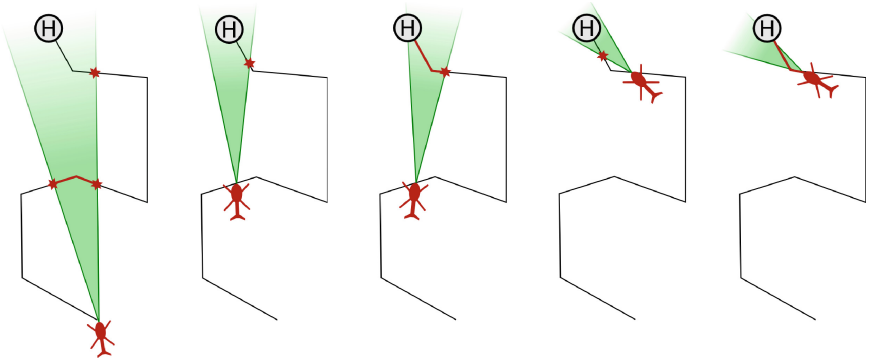
4.3 Algorithm Convergence

In the following, the proof for the convergence of the proposed algorithm is stated. For reasons of clarity, two Lemmas are formulated and proofed first.

Lemma 1. *After a finite number of ray to ray connections a direct connection will be found.*

Proof. In the previous section two options for a valid *ray to ray connection* were defined:

1. *Close*: The intersection from traveling, which is used as new start vertex in the next iteration, is always closer to $\mathbf{v}_g$ than $\mathbf{v}_s$ to $\mathbf{v}_g$.



(a) Ray to ray connection (b) Rotate ray directions (c) Ray to ray connection (d) Rotate ray directions (e) Direct connection

Fig. 5. Sequence of steps performed to reach the goal in the presented scenario. For each new start position, the path considered first aims at the goal position and investigates the intersections (red stars) between the graph (black lines) and the maximal drift rays (outer green lines, spanning the green area).

2. *Connected to* $\mathbf{v}_g$: It is possible to intersect the known path behind the goal vertex. This case has to be investigated further if the new start vertex has a larger distance to $\mathbf{v}_g$ than the distance between the current start vertex $\mathbf{v}_s$ and $\mathbf{v}_g$. The latter will be referred as $\Phi_1 = \Phi(\mathbf{v}_s, \mathbf{v}_g)$ in the following. If this happens only once, we will constantly decrease the distance to $\mathbf{v}_g$ in the following iterations. After another detour according to the same principle, the distance from our new start vertex to $\mathbf{v}_g$ will be smaller than Φ_1. This is based on the fact, that the connection to $\mathbf{v}_g$ was enclosed by the drift area before our first detour. Therefore, the distance to $\mathbf{v}_g$ will also be decreased in this situation.

By constantly decreasing the distance to $\mathbf{v}_g$ a direct connection will be found, at the latest when the new $\mathbf{v}_s$ is located on the edge directly connected to $\mathbf{v}_g$. □

Lemma 2. *For each start vertex* $\mathbf{v}_s$ *at least one rotation angle in the rotate ray directions case will result in a direct connection or ray to ray connection.*

Proof. By adding an edge for each new keyframe to its predecessor, there exist exactly one path between the start vertex $\mathbf{v}_s$ and the goal vertex $\mathbf{v}_g$. Both vertices are located in the drift area enclosed by the two rays $\mathbf{r}_l$ and $\mathbf{r}_r$.

There exists one rotation that places $\mathbf{r}_l$ on $\mathbf{v}_g$ and one which does the same for $\mathbf{r}_r$. In one of these two instances, the path directly connected to $\mathbf{v}_g$ will proceed in the drift area. Based on the fact that there exists a connection between $\mathbf{v}_s$ and $\mathbf{v}_g$, an intersection $\mathbf{v}_i$ with $\mathbf{r}_l$ or $\mathbf{r}_r$ will be found by traversing the edges starting from $\mathbf{v}_g$:

1. Intersections with both rays: corresponds to an intersection with $\mathbf{v}_s$ which results in a *direct connection*

2. Intersection with the other ray: *ray to ray connection*
3. Intersection with the same ray: *rotate ray directions*.

If the new intersection $\mathbf{v}_i$ occurs with the same ray $\mathbf{v}_g$ is placed on, the path between $\mathbf{v}_g$ and $\mathbf{v}_i$ is traversed and the vertex closest to the other ray is determined. The drift area is rotated in a way that the other ray intersect this vertex, which becomes the new $\mathbf{v}_i$. This results in a connection between $\mathbf{v}_g$ and $\mathbf{v}_i$ within the drift area. Traversing from $\mathbf{v}_i$ away from $\mathbf{v}_g$ will continue in the drift area. A new intersection $\mathbf{v}_i$ with $\mathbf{r}_l$ or $\mathbf{r}_r$ will be found and investigated as described above. Because there exists a path between $\mathbf{v}_s$ and $\mathbf{v}_g$, iterating this procedure will converge in either a *direct connection* or a *ray to ray connection*. □

Theorem 1. *The proposed algorithm will always converge in the goal position.*

Proof. As depicted in Fig. 3 and described in the previous section, the algorithm iterates through three cases:

1. *Direct connection*: Follow the connection and reach the goal position.
2. *Ray to ray connection*: Lemma 1 states that after a finite number of *ray to ray connections*, a *direct connection* will be found.
3. *Rotate ray directions*: According to Lemma 2, there will always a rotation be found which results in a *direct connection* or a *ray to ray connection*.

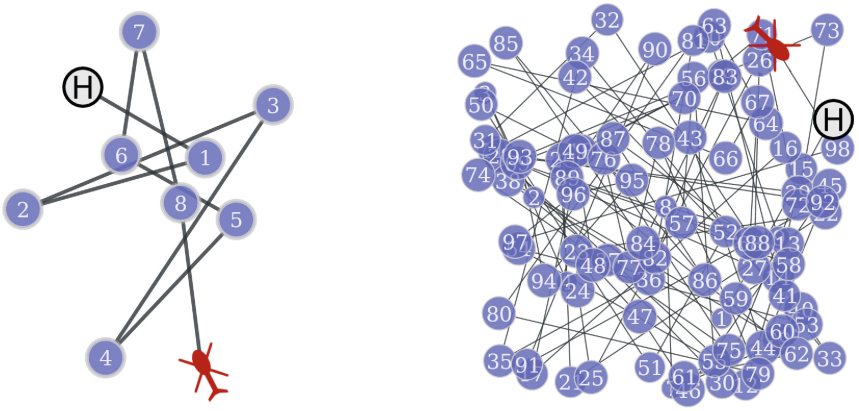
According to these observations a direct connection that leads to the goal position will always be found. □

5 Evaluation

In the following the proposed approach is evaluated in extensive numerical studies. Afterwards it is shown how it performed in a realistic simulation environment and a proof of concept in a real-world scenario is given.

5.1 Numerical Studies

As a first evaluation of the presented approach, Monte Carlo (MC) simulations on random graphs were performed. The latter represent flight courses from the take-off positions to the locations of the GPS loss. To generate a graph, n vertices were sampled at random coordinates in a square area with a side length of x meter. For each new vertex an edge connecting it with its predecessor was added to the graph. This resulted in a random graphs with n vertices and $n-1$ edges. The first vertex was used as take-off position and the last vertex as the start coordinate for the UAS homing. The created graphs look quite chaotic and in most cases are no flight maneuvers an operator would plan for an UAS (Fig. 6). Nevertheless, the random graphs allow us to evaluate the approach in a comprehensive number of experiments, which with a high probability detect any problems and weaknesses of the procedure.



(a) 10 vertices (b) 100 vertices

Fig. 6. The random graphs allow an intensive evaluation of the presented approach. The first vertex is considered as take-off position we want return to and the last vertex as coordinate of the UAS at the time of the GPS outage.

Three series of MC runs were performed, each with a different combination of the graph configuration parameters n and x. Further, a maximal drift ratio of 5% in relation to the traveled distance was defined. For each configuration 1000 random graphs were generated and the path planning for each graph was realized in 100 MC runs.

The mean distance used to reach the goal was calculated for each set of 100 MC runs. This value was divided by the optimal solution, characterized by the Euclidean distance between the start and the goal vertex, to form the travel distance ratio:

$$r_d = \frac{\frac{1}{100}\sum_{i=1}^{100} d_i}{\Phi(\mathbf{v}_s, \mathbf{v}_g)}, \tag{7}$$

with d_i stating the travel distance used to reach the goal in the i-th MC run. Thus, a r_d of two states that on average the distance is twice the optimal solution. Further would a value of one point out that the algorithm produced for all runs the optimal solution. The latter is not possible because of the positional drift which has to be considered, but the closer the results are to one, the better.

Table 1. Travel distance ratio (achieved mean/optimal). Quantiles of 1000 random graphs for each data set, with 100 MC runs per graph.

Data set	Vertices	Square side length	50%	95%	99%
1	10	500	1.010	1.160	1.479
2	10	5000	1.011	1.139	1.449
3	100	500	1.002	1.021	1.093

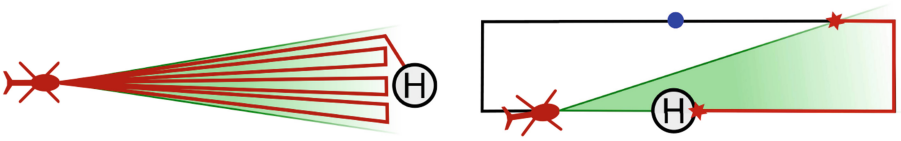


Fig. 7. Suboptimal planning. **Left:** A connection inside the drift area of the visual odometry (green) may lead to large detours. **Right:** In the visualized scenario, the planning will result in a relocalization at one point on the marked edges (red). An optimal solution would plan a detour using an interim goal (blue).

For each of the three configurations the quantiles of the mean traveled distance ratio from the 1000 random graphs were calculated (Table 1). The absolute time and distance savings are a lot bigger for data set 2 compared to data set 1 due to the larger size of the area. However, the evaluation based on the ratio eliminates this difference as expected. The increase in the number of vertices for data set 3 leads to better results. This is based on the higher number of edges resulting in more ray to ray connections while targeting the goal directly. Following this optimal direction more often, avoids detours and decreases the used travel distance.

By investigating the outliers of the numerical studies two special cases were identified. They are depicted in Fig. 7. The left image shows that a direct connection between the start and goal vertex can be quite long. The visualized instance shows an extreme situation that is very unlikely to happen. Nevertheless, also simpler detours following the same scheme may occur and in all these cases the presented algorithm will traverse the direct connection. Most of these detours could be prevented by traversing the direct connection, but performing a new planing according to the presented approach after a distance threshold is exceeded. This small adaption will very likely lead to new shortcuts. The second type of detours the algorithm will produce is also visualized in the right image of Fig. 7. By using the heuristic that only movements which contain the goal in the visual odometry drift area are valid, the depicted shortcut using an interim goal will be missed.

5.2 Simulations

The robot simulation Gazebo [7] was used to test the approach in a realistic scenario and gather valuable information for real-world experiments. The modeling, control and simulation of a quadcopter UAS within Gazebo were developed by Meyer *et al.* [10]. The pose information of the quadcopter are on the one hand used as ground truth in the evaluation and on the other hand the basis for the generation of noisy INS data. For the latter, white Gaussian noise was added to the poses using a standard deviation of $\sigma_{\mathrm{t}} = (0.02\,\mathrm{m}, 0.02\,\mathrm{m}, 0.04\,\mathrm{m})$ for the positional components and $\sigma_{\mathrm{r}} = (0.1^\circ, 0.1^\circ, 0.2^\circ)$ for the rotational components represented by Euler angles. Thereby the altitude as well as the rotation around the z-axis are chosen twice as big as the other components to model the standard error behavior of an INS. The modeled values represent the accuracies of a small INS with GPS measurements corrected by the RTK technique [14]. Furthermore,

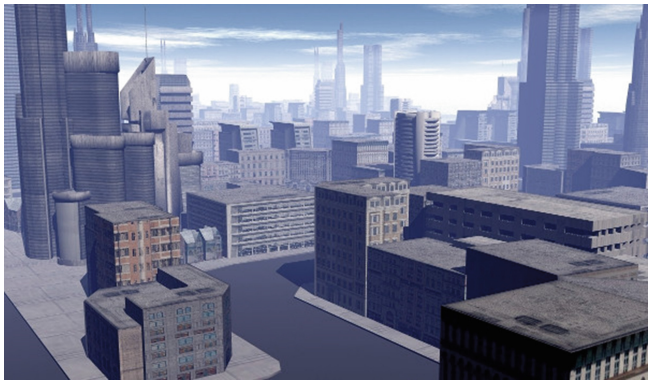


Fig. 8. Rendered image of the 3D model 'The City', which was used as environment in Gazebo.

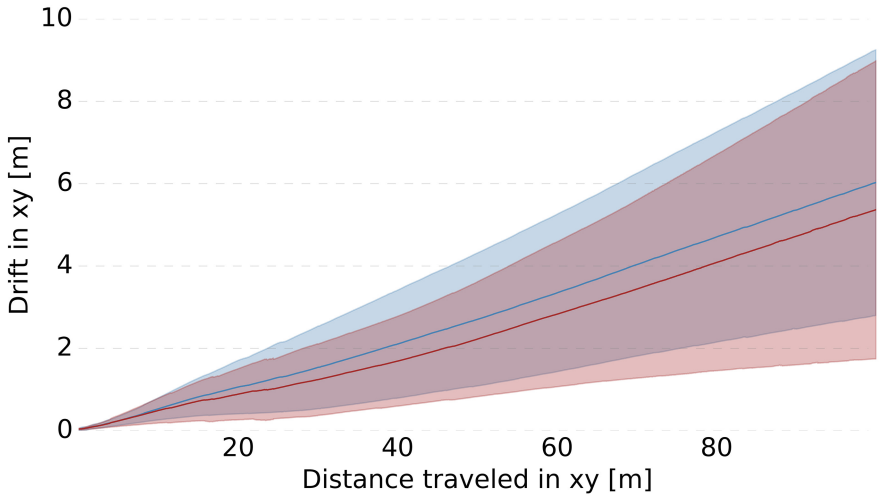


Fig. 9. Mean and standard deviation of the camera only LSD-SLAM drift from 100 MC runs for two straight level flights courses in Gazebo.

the camera simulation provided by Gazebo was used to generate images from a downward facing camera with 640 × 480 pixels at a frame rate of 30 Hz. The chosen aperture angle of the camera of 100° corresponds to a wide-angle lens. The mounting offsets between the camera and the INS are known in the simulation. As environment the 3D model 'The City' created by Herminio Nieves was used (Fig. 8). He published this [12] and other models free for commercial and non commercial use.

The drift of the LSD-SLAM was analyzed by performing MC runs for two different directions at a straight flight. The error behavior shows a linear increase and is quite similar for both directions (Fig. 9). At a travel distance of 100 m the

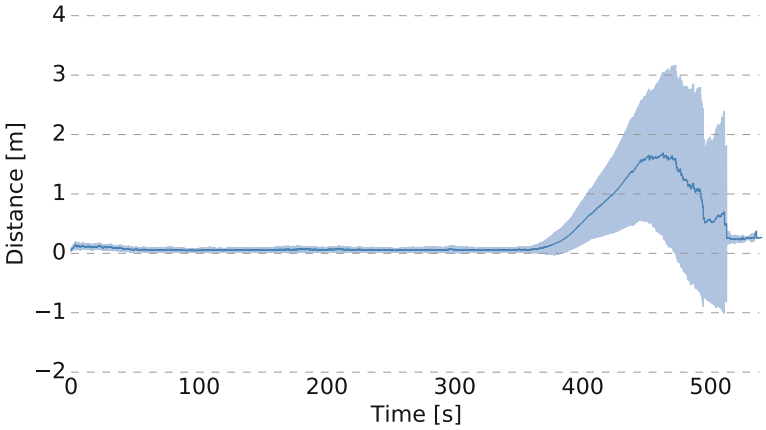


Fig. 10. Mean and standard deviation of the Euclidean distance between the pose estimation and the ground truth from 100 MC runs performing an UAS homing in Gazebo. INS-based LSD-SLAM was used before the GPS outage around 350 s and the original LSD-SLAM afterwards.

drift is characterized by a mean value of $\mu = 6.03$ m and a standard deviation of $\sigma = 3.23$ m for one direction as well as $\mu = 5.37$ m and $\sigma = 3.62$ m for the other. These values are quite high compared to analyses stated in literature [8] and may be a consequence of the 3D world that is partly without texture (Fig. 8). However, the large drift can be handled by increasing the upper bound of the angular error to 10° This value corresponds to an error of 17.63% in the travel distance, which covers for both evaluated directions more than three times the standard deviation added to the mean. The high value will result in some detours during the planning but guarantees convergence.

The evaluation of the proposed approach was realized with MC runs in Gazebo as follows. The initial path was traveled according to a list of waypoints which describe the path depicted in Fig. 5. This path as well as the map were created for each run from scratch and differ slightly due to the non-deterministic design of the LSD-SLAM. At the last waypoint, the GPS outage is simulated and the UAS uses the proposed algorithm to return autonomously to its take-off position. Thereby the direction of the UAS are updated every second based on the current pose estimate of the LSD-SLAM.

The Euclidean distance between the ground truth and the pose estimations shows the expected increase in uncertainty while traveling in unexplored areas and drops again when the LSD-SLAM creates loop closures by adding constraints between the latest and previously created keyframes (Fig. 10). In the end of the investigated scenario a mean distance of about 0.25 m is obtained, which is roughly three times higher than the accuracy achieved with the INS-based LSD-SLAM used before the GPS outage. This value depends on the camera resolution as well as the ground sample distance and will differ in other setups. The time at which the runs successfully performed the loop closure differs slightly, which is apparent

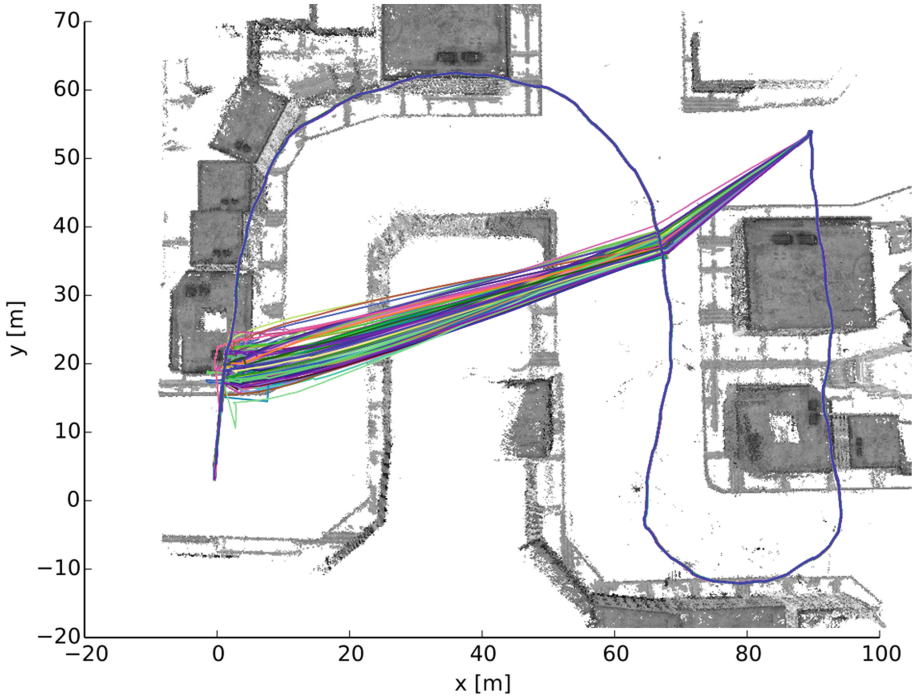


Fig. 11. The path information of 100 MC runs for returning to the start position at $(0, 0)$ after a simulated GPS outage in Gazebo. The path information is plotted on top of the map created by a single run before the GPS outage.

by the slowly decreasing mean value. The small standard deviations in the last seconds state that the relocalization worked in all runs.

The path information of the MC runs is visualized in Fig. 11. It shows the change in the course direction at crossing the previous path the first time and how the drift in the visual odometry influences the flight course. After successfully performing a loop closure to relocalize, the last meters are traversed for each run along the previous path to reach the start position.

Traveling according to waypoints till the GPS outage occurs covered a mean travel distance of 251.85 m. The direct connection back to the take-off position had a mean distance of 100.5 m and the path actually traveled by steering commands from the planning algorithm based on LSD-SLAM pose estimations lead to a mean distance of 108.53 m. This results in a travel distance ratio, according to equation (7), of 1.08 which is good, especially under consideration of the high maximal angular error of 10%.

5.3 Real-World Experiments

The utilized platform is a small hexacopter equipped with a payload containing an INS based on microelectromechanical systems (MEMS) and a camera.

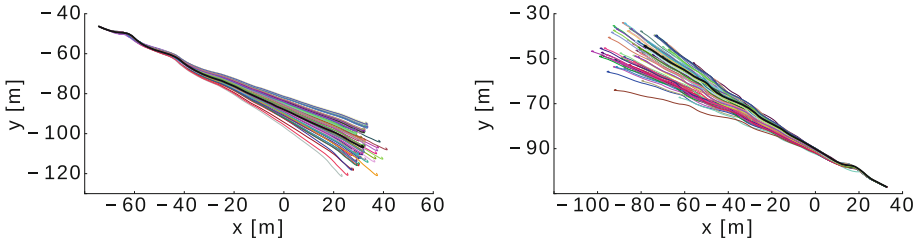


Fig. 12. Path visualization of the camera-only LSD-SLAM drift in the real-world. The two plots show straight flights with the INS path (black) and MC runs (colors).

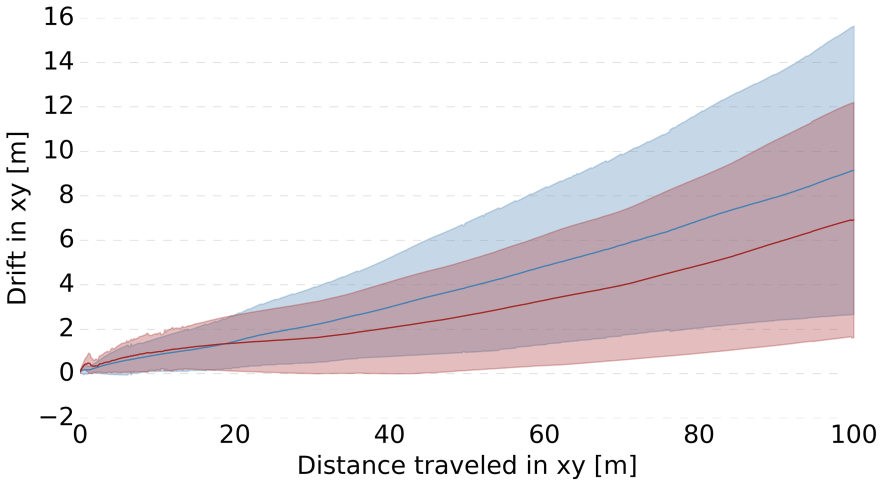


Fig. 13. Mean and standard deviation of the camera only LSD-SLAM drift from MC runs for two straight level flights courses in the real-world. The INS path, with an error in the range of a few centimeters, is used as ground truth.

As INS an Ellipse-D from SBG Systems with standard deviations of $\sigma_\psi = 0.2°$, $\sigma_\theta = 0.1°$ and $\sigma_\varphi = 0.1°$, stated in the technical data sheet [14], is used. The camera from XIMEA captures images with a resolution of 2048×1088 pixels and is attached to a wide-angle lens with a focal length of 4.8 mm. A pixel binning downscales the images by a factor of four to 512×272 pixels to achieve real-time performance on a standard Laptop CPU. The sensors are rigidly mounted and connected with a synchronization cable for hardware trigger signals. The mounting offsets between the sensors were estimated in a system calibration of the sensor setup with data collected beforehand [1].

To estimate the upper bound of the positional drift, MC runs on two data sets recorded while performing straight flights were performed. The nondeterministic LSD-SLAM leads for multiple runs to differing pose estimations (Fig. 12). The evaluation results in a mean value of $\mu = 6.03$ m and a standard deviation of $\sigma = 3.23$ m for one direction as well as $\mu = 5.37$ m and $\sigma = 3.62$ m for the other (Fig. 13).

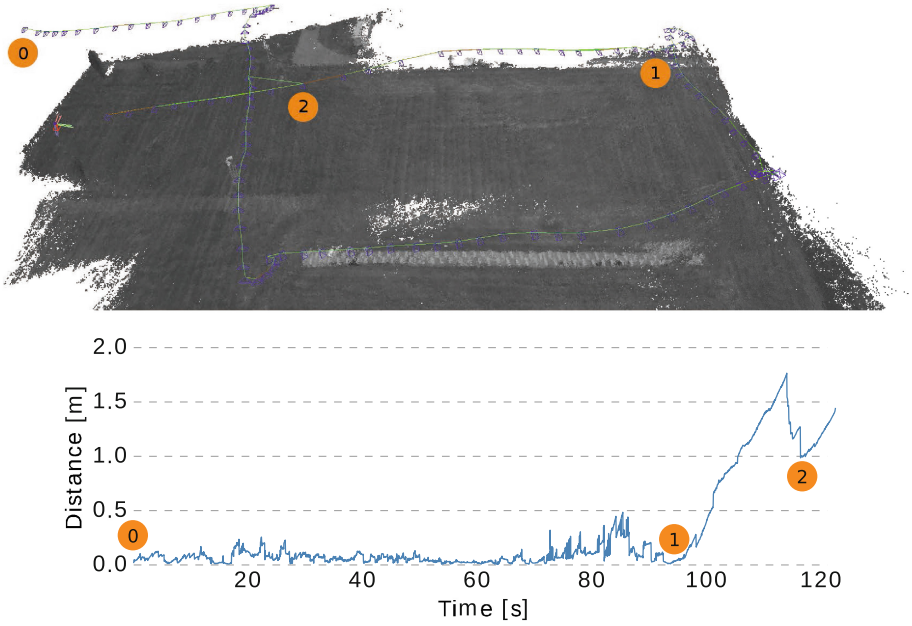


Fig. 14. Real-world scenario. The copter starts at position "0" with the INS-based LSD-SLAM and switches to the original LSD-SLAM at "1". A loop closure to a previously generated keyframe is performed at position "2". **Top:** Visualization of the LSD-SLAM results showing the created map, keyframes and image constraints. **Bottom:** Distance between the pose estimation of the LSD-SLAM and the INS measurements.

Although nearly the whole area observed by the camera contains gradients large enough to perform pixel-wise stereo matching by the LSD-SLAM, the drift is larger compared to the Gazebo simulation performed in the previous section. This may be the result of repetitive structures in the form of a lot of grass covering the area. Nevertheless, the estimation errors can be handled by setting the maximal drift accordingly and the only drawback is that the shortcut algorithm will plan larger detours to ensure reentering previously visited areas. The bigger problem is the number of false loop-closures, which happen because of the repetitive structures in the area. By increasing the strictness for loop closures most of the false loop closure, but at the same time a lot of real loop closures, are eliminated. The detection of previously visited areas did not work reliable in the outdoor area covered by the available flight permission. The flight visualized in Fig. 14 shows that the detection of only one connection to the previous path will already reduce the error of the pose estimation. More connection would add additional constraints to the mapping part of the LSD-SLAM and decrease the error of the pose estimation even more. This is presented as proof of concept, although it was not possible to perform a successful homing with the presented shortcut algorithm in the test area.

6 Summary

An approach which performs a path planning for UAS using a previous build map and actual camera images was presented. The travel distance is minimized by exploiting safe shortcut through unexplored areas. Therefore, a very fast heuristic to perform a local path optimization which produces almost optimal results in nearly all situations is used.

After the proof of convergence, the approach was evaluated in extensive numerical studies. A realistic scenario was realized in the simulation framework Gazebo. The first part of the flights used the INS-based LSD-SLAM and a switch to the original LSD-SLAM was performed after a simulated GPS outtake. To return to the take-off position, the path planning was realized with the presented algorithm and resulted in a relocalization in the previously visited area close to the take-off position. Based on the simulations in Gazebo, real-world experiments were performed. Thereby, the concept was proofed by showing how the loop-closure to previous keyframes reduces the localization error.

References

1. Bender, D., Cremers, D., Koch, W.: A position free boresight calibration for INS-camera systems. In: 2016 International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp. 52–57 (2016)
2. Bender, D., Cremers, D., Koch, W.: Map-based drone homing using shortcuts. In: 2017 International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp. 505–511 (2017)
3. Bender, D., Rouatbi, F., Schikora, M., Cremers, D., Koch, W.: Scaling the world of monocular SLAM with INS-measurements for UAS navigation. In: 2016 19th International Conference on Information Fusion (FUSION), pp. 1493–1500 (2016)
4. Chapuis, N.: Les opérations structurantes dans la connaissance de l'espace chez les mammifères: détour, raccourci et retour. Ph.D. thesis, Université Aix-Marseille 2 (1988)
5. Engel, J., Sturm, J., Cremers, D.: Semi-dense visual odometry for a monocular camera. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 1449–1456 (2013)
6. Giusti, A., Guzzi, J., Cireşan, D.C., He, F.L., Rodriguez, J.P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Di Caro, G., Scaramuzza, D., Gambardella, L.M.: A machine learning approach to visual perception of forest trails for mobile robots. IEEE Robot. Autom. Lett. **1**(2), 661–667 (2016)
7. Koenig, N., Howard, A.: Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 3, pp. 2149–2154 (2004)
8. Liu, H., Jiang, R., Hu, W., Wang, S.: Navigational drift analysis for visual odometry. Comput. Inform. **33**(3), 685–706 (2014)
9. Meyer, J.A., Filliat, D.: Map-based navigation in mobile robots: II. A review of map-learning and path-planning strategies. Cogn. Syst. Res. **4**(4), 283–317 (2003)
10. Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., von Stryk, O.: Comprehensive simulation of quadrotor UAVs using ROS and Gazebo. In: Simulation, Modeling, and Programming for Autonomous Robots, pp. 400–411. Springer, Heidelberg (2012)

11. Nelson, R.C.: Visual homing using an associative memory. Bio. Cybern. **65**(4), 281–291 (1991)
12. Nieves, H.: The City: 3D Model. http://sharecg.com/v/79711/gallery/5/3D-Model/The-City (2015). Accessed 20 Feb 2018
13. Pomerleau, D.A.: Neural network based autonomous navigation. In: Vision and Navigation, pp. 83–93. Springer, Boston (1990)
14. SBG Systems: Ellipse Series: Miniature High Performance Inertial Sensors: technical data sheet. https://www.sbg-systems.com/docs/Ellipse_Series_Leaflet.pdf (2015). Accessed 3 Oct 2017
15. Sibley, G., Mei, C., Reid, I., Newman, P.: Vast-scale outdoor navigation using adaptive relative bundle adjustment. Int. J. Robot. Res. **29**(8), 958–980 (2010)
16. Trullier, O., Wiener, S.I., Berthoz, A., Meyer, J.A.: Biologically based artificial navigation systems: review and prospects. Prog. Neurobiol. **51**(5), 483–544 (1997)
17. Valencia, R., Andrade-Cetto, J., Porta, J.M.: Path planning in belief space with pose SLAM. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 78–83 (2011)