

Layered Mission and Path Planning for MAV Navigation with Partial Environment Knowledge

Matthias Nieuwenhuisen and Sven Behnke

Autonomous Intelligent Systems Group
Computer Science Institute VI
University of Bonn, 53113 Bonn, Germany
`nieuwenh@ais.uni-bonn.de`, `behnke@cs.uni-bonn.de`

Abstract. Successful operation of micro aerial vehicles in partially known environments requires globally consistent plans based on incomplete environment models and quick reactions to unknown obstacles by means of real-time planning of collision-free trajectories.

In this paper, we propose a complete layered mission and navigation planning system based on coarse prior knowledge and local maps from omnidirectional onboard obstacle perception. We generate trajectories in a multi-layered approach: from mission planning to global and local trajectory planning to motion control.

Keywords: path planning, obstacle avoidance, MAV

1 Introduction

Today, micro aerial vehicles (MAV) become relevant for all types of applications, from inspection and surveillance mission to delivery tasks, due to their low cost and flexibility. Up to now, MAV missions are mostly specified by a path of GPS waypoints that are reached in an obstacle-free altitude. More complex operations have to be performed by a human operator.

We aim for a fully autonomous creation of mission-specific semantic maps on demand [20]. Special focus lies on the inspection of a building's facade [21]. Hence, our MAV (Fig. 1) has to operate in the vicinity of buildings and other structures, e.g., trees and power cables. To achieve these objectives, we cannot solely rely on predefined GPS waypoint following—especially as the accuracy drops in the vicinity of larger obstacles—but need more elaborated means of navigation. To react quickly on obstacles and to allow for globally consistent planning, we follow a multi-layer approach: from slower deliberative to fast reactive layers, including mission planning, global and local path planning, fast local obstacle avoidance, and robust motion controllers.

In our application domain, we assume—in contrast to fully autonomous exploration—to have partial environment knowledge in advance. For initial mission and path planning we employ digital elevation models (DEM) and 3D city models acquired by land surveying authorities. These models—specified as Level-of-Detail 2 in CityGML [12]—contain buildings' footprints, heights,

and roof shapes. Other structures, e.g., street lighting and vegetation, are not included in these models. Nevertheless, these smaller structures constitute a collision hazard for the MAV. We address this problem by adjusting the global plans locally based on onboard sensors. In many cases these smaller obstacles can be avoided by means of our local obstacle map without global replanning.

Due to their size and weight constraints, MAVs often are equipped with few sensors and perception algorithms are limited by computing power. In order to enable autonomous navigation in difficult 3D environments, we designed a small and lightweight continuously rotating 3D laser scanner that measures distances of up to 30 m in almost all directions. Additionally, our MAV is equipped with two stereo camera pairs, and ultrasonic sensors covering the volume around the MAV up to 30 m range [15]. All these sensors have only local precision. This is reflected in the local multiresolution property of our MAV-centric obstacle map. The local navigation planner operates directly on this representation.

After a discussion of related work in the next section, we will briefly describe our MAV in Sec. 3. Our hierarchical control architecture from mission planning to low-level control is detailed in Sec. 4. We present evaluation results in Sec. 5.

2 Related Work

The application of MAVs varies especially in the level of autonomy—ranging from basic hovering and position holding [4] over trajectory tracking and way-point navigation [27] to fully autonomous navigation [13].

Particularly important for fully autonomous operation is the ability to perceive obstacles and to avoid collisions. Obstacle avoidance is often neglected, e.g., by flying in a sufficient height when autonomously flying between way-points. Most approaches to obstacle avoidance for MAVs are camera-based, due to the limited payload [23, 29, 30]. Hence, collision avoidance is restricted to the field of view (FoV) of the cameras.

Other groups use 2D laser range finders (LRF) to localize the UAV and to avoid obstacles [13], limiting obstacle avoidance to the measurement plane of the LRF, or combine LRFs and visual obstacle detection [31, 19]. Still, their perceptual field is limited to the apex angle of the stereo camera pair (facing forwards), and the 2D measurement plane of the scanner when flying sideways. They do not perceive obstacles outside of this region or behind the vehicle.

We allow omnidirectional 4D movements (3D translation + yaw rotation) of our MAV, thus we have to take obstacles in all directions into account. Another MAV with a sensor setup that allows omnidirectional obstacle perception has been described by Chambers et al. [5].

A two-level approach to collision-free navigation using artificial potential fields on the lower layer is proposed in [26]. Similar to our work, completeness of the path planner is guaranteed by an allocentric layer on top of local collision avoidance.

Some reactive collision avoidance methods for MAVs are based on optical flow [11] or a combination of flow and stereo vision [18]. However, solely optical

flow-based solutions cannot cope well with frontal obstacles and these methods are not well suited for omnidirectional obstacle avoidance as needed for our scenario.

Recent search-based methods for obstacle-free navigation include work of MacAllister et al. [22] and Cover et al. [6]. A good survey on approaches to motion planning for MAVs is given in [10]. These methods assume complete knowledge of the scene geometry—an assumption that we do not make here.

3 System Setup

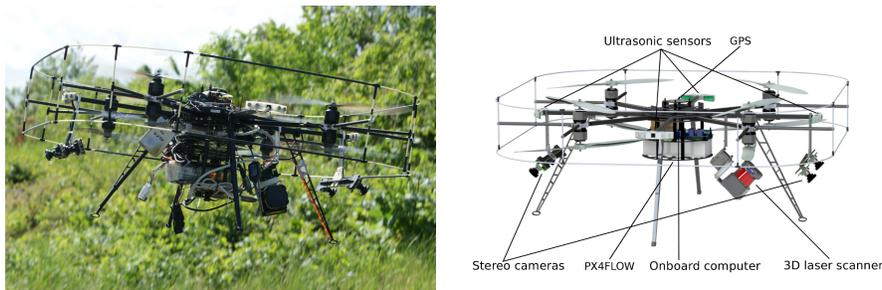


Fig. 1: Our MAV is equipped with eight co-axial rotors. For localization and obstacle perception it employs a multimodal sensor setup, including a continuously rotating 3D laser scanner, two stereo cameras, and a precise GPS/INS system.

Our robot, shown in Fig. 1, is an octocopter MAV equipped with a multimodal sensor setup to obtain almost omnidirectional obstacle perceptions. The main sensor for this purpose is a continuously rotating 3D laser scanner based on a lightweight Hokuyo UTM-30LX-EW 2D LRF. This laser covers the space around the MAV in almost all directions—only a small cone towards the upper rear of the MAV is shadowed—at a rate of 2 Hz. Other sensors include two stereo camera pairs with fisheye lenses and eight ultrasonic sensors to detect transparent obstacles in the close vicinity of the MAV that are hard to perceive with the other sensors.

We use the stereo cameras, laser scanner, an optical flow camera [16], and a precise differential GPS [9], for localization and state estimation in indoor and outdoor environments. For details on our sensor setup and processing see [15, 7]. To process the sensor information and to solve planning problems independent from the connection to a ground control station our MAV is equipped with an onboard computer providing sufficient computing power (Intel Core i7 3820QM 2.7 GHz). As middleware, we employ the Robot Operating System ROS [28].

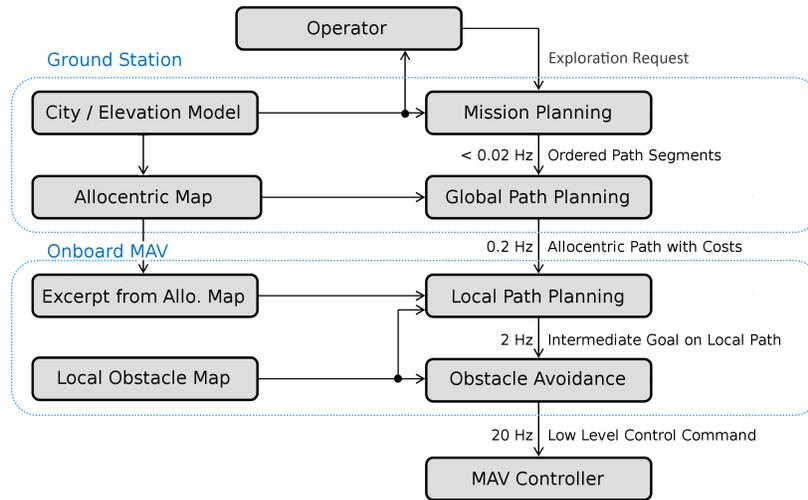


Fig. 2: Planning hierarchy. From top to bottom the execution frequency increases while the environment representations become more local and the planned action sequences cover shorter durations and become more concrete.

4 Planning and Navigation Hierarchy

For successful fulfillment of and safe navigation during a specified exploration and mapping mission, multiple planning and navigation tasks have to be executed: from mission planning to low-level motion control. These tasks require different abstractions of the environment, as illustrated in Fig. 2.

To plan a mapping mission, we need a coarse (semantic) model of the environment; to plan collision-free paths, we need a finer and up-to-date consistent geometric model; and to avoid collisions, we need a non-aggregated local representation of the close vicinity of the MAV. The planned actions also have different granularity, which is represented by the planning frequency, from once per mission to multiple times per second. The higher-layer planners set goals for the lower-level planners which produce more concrete action sequences based on more local and up-to date environment representations.

4.1 Mission Planning

The topmost layer of our hierarchy is a mission planner. A set of mission-relevant 4D view and auxiliary poses for the MAV is specified by a human operator or an external view pose planning module. The mission planner determines the best order of the mission poses in terms of total flight path costs. To determine the path costs, the mission planner employs a digital elevation model (DEM) of the mission area and a 3D city model (LoD 2). These relative coarse environment representations are available from land surveying authorities. Fig. 3 shows the input data that is stored efficiently in an OctoMap [17].

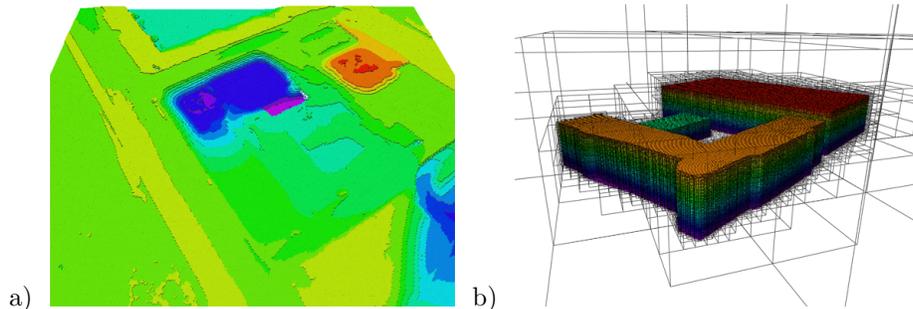


Fig. 3: The static environment representation a priori known, consists of (a) a digital elevation model and (b) a 3D city model. The color corresponds to height. For planning, geometric representations of both are stored in an OctoMap.

The estimated flight costs between every pair of mission poses is determined by means of the global path planner described in the next section, operating on the static environment model (see Fig. 4a). For efficiency, we reuse information in the grid-based path planning representation in consecutive cycles, i.e., the obstacle induced costs and—if the start node remains the same—the costs to reach grid cells. The evaluation of the minimal-cost order of mission poses is an instance of the traveling salesman problem. As our problem instance is comparably small (tens of poses), it is feasible to solve the problem exactly. We employ the freely available Concorde TSP solver [1] with our previously calculated distance matrix. Together with the costs of pairs of mission poses, the corresponding planned paths are stored. These paths can serve as initial guess to speed up the global path planning on the next lower layer.

4.2 Global Path Planning

The next layer in the planning hierarchy is a global path planner. This layer plans globally consistent plans based on the static environment model discretized to grid cells with 1 m edge length, the current pose estimate of the MAV, and a robot-centric local grid map representing the vicinity of the MAV.

Planning frequency is 0.2 Hz and we use the A* algorithm to find cost-optimal paths. We assume that in our application domain most obstacles not known in advance can be surrounded locally without the need for global replanning. Hence, it is sufficient to replan globally on a more long-term time scale to keep the local planner’s deviations synchronized to the global plan and to avoid the MAV to get stuck in a local minimum that the local planner cannot solve due to its restricted view on the environment. As via points that are not mission critical can be blocked by locally perceived obstacles, it is not sufficient to send the next waypoint of the global path to the local planning layers. Instead, the input to the local planner is the complete global plan. The global path is optimal with respect to the allocentric map. Hence, the path costs of the global path are a

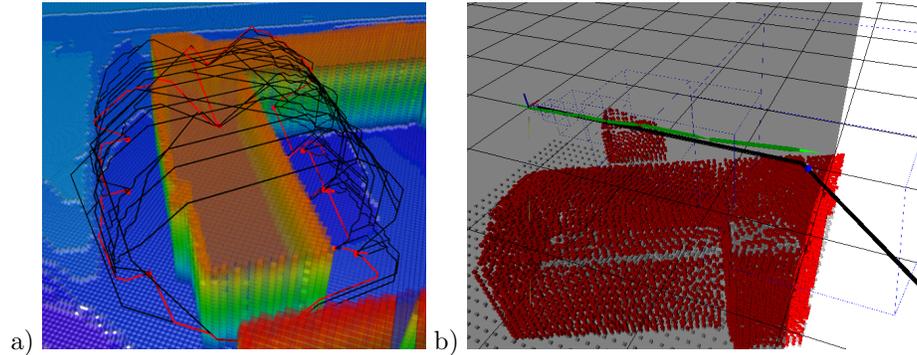


Fig. 4: Two layers of planning. (a) On the top layer, a mission planner evaluates the best execution order of mission poses (red line connecting red squares). The black lines depict all cost-optimal paths between each pair of mission poses. (b) A local planner refines global plans employing a robot-centric local map augmented with sensor measurements (red). The local planner’s goal is the intersection (blue dot) of a globally planned path (black lines) with the view frustum (gray box) of the MAV resulting in a locally refined path (green arrows).

lower bound to path costs for refined plans, based on newly acquired sensor information—mostly dynamic and static previously unknown obstacles—and a local path deviating from the global plan cannot be shorter in terms of path costs. Locally shorter plans on lower layers with a local view on the map may yield globally suboptimal paths, eventually. Thus, we add the estimated path costs between waypoints of the global plan to its edges to facilitate efficient exploration of the search space on the next lower layer. So, the optimal costs between every two waypoints of the global path can be determined by the local planner and there exists no shorter way between these. Also, mission goals are marked as the local planner has to reach these exactly. If this is not possible, the mission planning has to resolve this failure condition.

4.3 Local Path Planning

To quickly react on obstacles not represented in the allocentric representation of the environment, the globally planned path is further refined on a local path planning layer. Fig. 5a shows a local deviation from a global plan without global replanning. This planner runs at a frequency of 2 Hz—equal to the rate at which the MAV’s rotating laser scanner acquires a new 3D point cloud of the close environment. The spatial planning horizon is restricted to the measurement range of the laser sensor. Scans are aggregated into a grid-based local multiresolution obstacle map (cf. [8]) with edge length 40 m. To avoid obstacles not locally perceived but represented in the allocentric map, we add the occupied parts of the allocentric map inside of the local planning volume to the egocentric map.

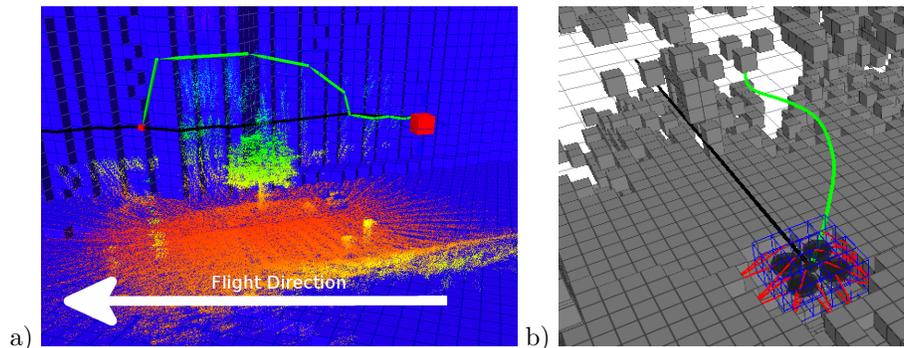


Fig. 5: Local planning and obstacle avoidance. (a) While following a planned path (black) based on the allocentric environment model (blue) a tree is perceived with local sensing (colored measurements). The local multiresolution path planner finds a local plan around the obstacle without global replanning (green). (b) The green line depicts the predicted trajectory of the MAV for the potential field-based reactive collision avoidance. The black line is the direct connection to an intermediate egocentric goal attracting the robot. The artificial forces applied to the current MAV’s position are illustrated by red lines.

We employ 3D local multiresolution path planning, extending ideas from our prior work [2]. This efficient planning technique allows for frequent replanning, which makes 3D navigation in dynamic, unpredictable environments possible.

To resemble the onboard sensors’ characteristics with distance dependent precision, and the uncertainty in the motion execution of the MAV by external influences, e.g., gusts of wind, the planner operates on a grid-based robot-centric obstacle maps with higher resolution in the center and decreasing resolution in the distance. For this purpose, we embed grids with M cells in each dimension into each other (see Fig. 6a) with a cell size s_l of $2^l s_{min}$, where l is the multiresolution level. The inner grids cover an eighth of the volume of the outer grids, i.e., the cells with the indices $[\frac{M}{4} : \frac{3M}{4}]$. The resulting multiresolution representation covers the same volume as a uniform $N \times N \times N$ grid with LM^3 cells. Here, $L = \log_2(N/M) + 1$ is the number of necessary resolution levels.

Fig. 6b shows our obstacle model: a core of the perceived obstacle enlarged by the approximate robot radius r_F and a distance-dependent part r_D that models the uncertainty of farther away perceptions and motions with high costs. Added is a part with linear decreasing costs with increasing distance to the obstacle r_S that the MAV shall avoid if possible. The integral of the obstacle stays constant by reducing its maximum costs h_{max} with increasing radius.

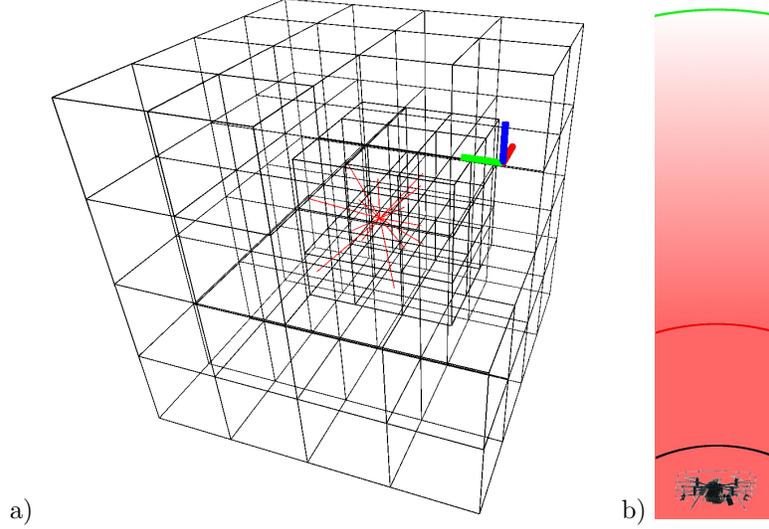


Fig. 6: Local multiresolution path planning. (a) Cut through the robot-centered multiresolution planning grid. The robot’s position is depicted by the axes in the origin. Red lines depict edges from one example cell to its neighbors. (b) We model obstacles (depicted by the MAV) in the local multiresolution grid as a fixed core (within black circle), a safety area with maximum costs dependent on the distance between obstacle and MAV (between black and red circle), and an avoidance zone with linear decreasing costs (between red and green circle).

For a distance d between a grid cell center c and the obstacle center o the obstacle costs h_o are given by

$$h_o(d) = \begin{cases} h_{max} & \text{if } d \leq (r_F + r_D) \\ h_{max} \frac{1-d-(r_F+r_D)}{2*(r_F+r_D)} & \text{if } (r_F + r_D) < d < 3 * (r_F + r_D) . \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The distance dependent part r_D grows linear with the distance of the obstacle o to the map center:

$$r_D = r_F + 0.1\|o\|. \quad (2)$$

In every planning cycle, the local obstacle map obtained from the onboard sensors is merged with a local excerpt of the allocentric world model given the current MAV’s pose estimate. Distances to all obstacles are calculated for estimating their radius in later processing steps. We use lazy evaluation of the obstacle costs in the planning grid during the node expansions of the planner.

The local planner is initialized with a local goal on the global path. This goal is the intersection of the global plan with the local planning volume (Fig. 4b). If the global plan contains mission-relevant waypoints within the local planning volume, the closest of these is taken as the local goal.

To bind the local plan to the globally consistent path we add a proximity term to the local planner’s cost function. The cost function for a cell with center c incorporates the sum of all obstacle costs h_o according to Eq. 1 and the distance of the cell’s center to the closest path segment s of the global plan G :

$$\text{cost}(c) = \sum_{o \in \text{Obst.}} h_o(\text{dist}(c, o)) + \gamma \underset{s \in G}{\text{argmin}} \text{dist}(c, s). \quad (3)$$

The parameter γ controls how tightly the local plan is bound to the global plan.

For planning we embed an undirected graph into this grid and perform A* graph search [14] from the center of the MAV-centered grid to the goal. The costs of expanding an edge e from a grid cell c_{from} to a cell c_{to} are the cells’ costs multiplied by the lengths of the edge within the respective cells:

$$\text{cost}(e) = \|e_{\text{from}}\| \text{cost}(c_{\text{from}}) + \|e_{\text{to}}\| \text{cost}(c_{\text{to}}). \quad (4)$$

The edge length $\|e_c\|$ within a cell is the Euclidean distance from the cell’s center to the intersection point between the cells.

4.4 Reactive Collision Avoidance

In order to incorporate sensor measurements arriving at a rate of more than 2 Hz into collision avoidance and to react on quick movements of the MAV, we employ a reactive collision avoidance layer between planning and control. For resource efficiency and to mitigate latencies between measurements and control commands, we developed a potential field-based obstacle avoidance. Our approach predicts the effects of control commands to the MAV’s trajectory for a short time horizon and reduces the velocity if the MAV will approach obstacles too closely in the prediction horizon [25]. Fig. 5b shows an example situation in a garage indoor environment.

The MAV position is modified according to the accumulated obstacle-repelling potentials of its parts. Furthermore, the effects of the potential field on the robot’s orientation are evaluated according to Eq. 5. As the robot is represented as a discretized 3D model, we can calculate the angular momentum on the MAV’s center by all artificial forces \mathbf{F}_i applied to the individual robot cells i and their respective relative positions \mathbf{p}_i :

$$\mathbf{M} = \sum_i \mathbf{p}_i \times \mathbf{F}_i. \quad (5)$$

From the three rotational velocities, only the yaw velocity can be chosen independently from the linear velocities of the MAV. Hence, we project all \mathbf{p}_i and \mathbf{F}_i to a plane parallel to the ground. Thus, we get an acceleration around the z-axis resulting in an angular velocity.

The result of the reactive collision avoidance layer is a 4D velocity vector $\mathbf{v} = (v_x, v_y, v_z, v_\theta)$. It runs at a rate of 20 Hz in our navigation hierarchy.

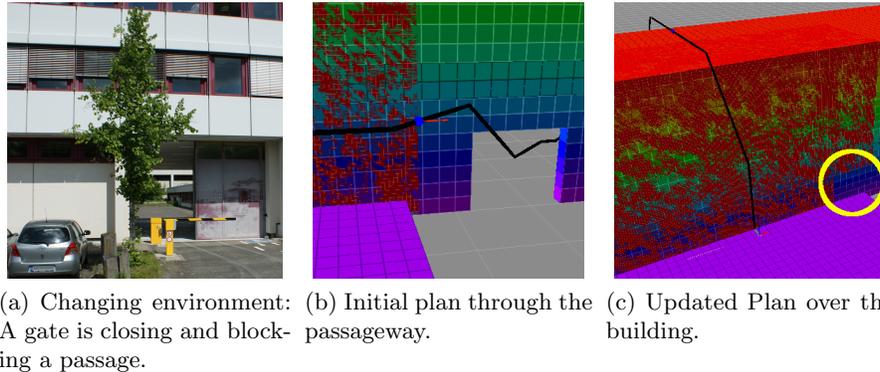


Fig. 7: Frequent replanning on the allocentric planning layer incorporates new sensor information to update the global plan. a) This example shows a passageway with a closing gate. b) In the global model the gate is open, the initial plan goes through the passageway. The red colored surface depicts the parts of the buildings observed by local sensors. c) While approaching the closed door (circled yellow) the MAV perceives it and updates the model. An alternative path over the building is planned.

4.5 Low-level MAV Flight Controllers

The lowest layer of our control hierarchy consists of a velocity controller that generates attitude commands for the attitude controller running on the MAV microcontroller board. A downward viewing camera is used in combination with an ultrasonic ground distance sensor and the inertial measurement unit to estimate the vehicle speed at high rate. It is also possible to switch to a pose controller, e.g., to hover at a view pose. We detail our controllers in [3].

5 Evaluation

We evaluated the mission, global, and local path planning in simulation. While following globally planned paths based on a city model, both planners were able to achieve the desired replanning frequencies. The allocentric model was represented in a $(100\text{ m})^3$ grid with 1 m cell size.

We evaluated the allocentric planning layers qualitatively. Fig. 7 shows a situation in which the global map needs to be updated with local sensor information—a door is closed and blocks a passageway—because local planning cannot find a feasible path around the obstacle. By frequent replanning, a new globally consistent path is found shortly after the blockage is incorporated into the allocentric map. We show resulting plans towards a goal throughout the MAV’s flight in Fig. 8.

We compared our path planner with local multiresolution grid to two with a uniform grid. Our multiresolution grid has an inner cell size of 0.25 m and

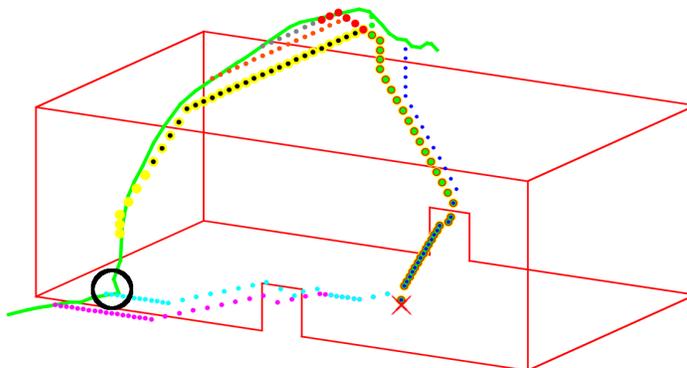


Fig. 8: Global plans are frequently updated while the MAV’s onboard sensors perceive the environment. The green line depicts the MAV’s trajectory, starting on the left side. The dotted lines depict plans from the respective MAV position towards the goal (red cross). A closed passage is perceived and the global plan is updated (black circle).

the duration of one planning cycle is in average 12 ms. The maximum is 35 ms. With a uniform grid with cells of 0.25 m the average planning time of 26 ms is still acceptable, but the maximum time of 3.4s exceeds the time window significantly. Increasing the cell size to 1 m reduces the planning times to 4 ms in average and 20 ms in maximum. Our multiresolution approach results in 3% longer trajectories than the higher resolution uniform approach. Using the lower resolution uniform grid results in 9% longer trajectories.

An detailed evaluation of the reactive collision avoidance in simulation and on our MAV is presented in [24]. Summarizing the evaluation, the prediction of trajectories leads to smoother trajectories. Experiments with the real MAV show that obstacles can be perceived and avoided. The achievable planning rate is an order of magnitude higher than needed in our system.

6 Conclusions and Future Work

By employing a global-to-local approach in our navigation planning pipeline, we achieve replanning frequencies that match the rate of expected changes in the environment model on different layers without losing the property of globally consistent plans. The plans on all layers are updated with appropriate frequencies and newly acquired information can be incorporated into globally optimal plans after a short time period.

In future work, we will further investigate the conditions in which plans on a layer need to be adjusted, i.e., global plans are more likely to fail in the vicinity of the MAV and local plans at the border of the planning volume. This will allow for further improvements of the interface between the individual planning layers.

Acknowledgments

This work has been supported by grant BE 2556/8 of German Research Foundation (DFG).

References

1. Applegate, D., Bixby, R., Chvatal, V., Cook, W.: Concorde TSP solver (2006)
2. Behnke, S.: Local multiresolution path planning. *Robocup 2003: Robot Soccer World Cup VII* pp. 332–343 (2004)
3. Beul, M., Worst, R., Behnke, S.: Nonlinear model-based position control for quadrotor UAVs. In: *Proc. of Joint Int. Symp. on Robotics and German Conf. on Robotics* (2014)
4. Bouabdallah, S., Murrieri, P., Siegwart, R.: Design and control of an indoor micro quadrotor. In: *Proc. of IEEE Int. Conf. on Robotics and Automation* (2004)
5. Chambers, A., Achar, S., Nuske, S., Rehder, J., Kitt, B., Chamberlain, L., Haines, J., Scherer, S., Singh, S.: Perception for a river mapping robot. In: *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2011)
6. Cover, H., Choudhury, S., Scherer, S., Singh, S.: Sparse tangential network (SPARTAN): Motion planning for micro aerial vehicles. In: *Proc. of IEEE Int. Conf. on Robotics and Automation* (2013)
7. Droeschel, D., Schreiber, M., Behnke, S.: Omnidirectional perception for lightweight UAVs using a continuous rotating laser scanner. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. vol. XL-1/W2*, pp. 107–112 (2013)
8. Droeschel, D., Stückler, J., Behnke, S.: Local multiresolution representation for 6D motion estimation and mapping with a continuously rotating 3D laser scanner. In: *Proc. of IEEE Int. Conf. on Robotics and Automation* (2014)
9. Eling, C., Klingbeil, L., Wieland, M., Kuhlmann, H.: A precise position and attitude determination system for lightweight unmanned aerial vehicles. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* (2013)
10. Goerzen, C., Kong, Z., Mettler, B.: A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *J. of Intelligent and Robotic Systems* 57(1-4), 65–100 (2010)
11. Green, W., Oh, P.: Optic-flow-based collision avoidance. *Robotics Automation Magazine, IEEE* 15(1), 96–103 (2008)
12. Gröger, G., Kolbe, T.H., Czerwinski, A., Nagel, C.: OpenGIS city geography markup language (citygml) encoding standard. Open Geospatial Consortium Inc. Reference number of this OGC® project document: OGC (2008)
13. Grzonka, S., Grisetti, G., Burgard, W.: A fully autonomous indoor quadrotor. *IEEE Trans. on Robotics* 28(1), 90–100 (2012)
14. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. on Systems Science and Cybernetics* 4(2), 100–107 (1968)
15. Holz, D., Nieuwenhuisen, M., Droeschel, D., Schreiber, M., Behnke, S.: Towards multimodal omnidirectional obstacle detection for autonomous unmanned aerial vehicles. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. vol. XL-1/W2*, pp. 201–206 (2013)
16. Honegger, D., Meier, L., Tanskanen, P., Pollefeys, M.: An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications. In: *Proc. of IEEE Int. Conf. on Robotics and Automation* (2013)

17. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots* (2013)
18. Hrabar, S., Sukhatme, G., Corke, P., Usher, K., Roberts, J.: Combined optic-flow and stereo-based navigation of urban canyons for a UAV. In: *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2005)
19. Huh, S., Shim, D., Kim, J.: Integrated navigation system using camera and gimbaled laser scanner for indoor and outdoor autonomous flight of UAVs. In: *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. pp. 3158–3163 (2013)
20. Klingbeil, L., Nieuwenhuisen, M., Schneider, J., Eling, C., Droeschel, D., Holz, D., Läbe, T., Förstner, W., Behnke, S., Kuhlmann, H.: Towards autonomous navigation of an UAV-based mobile mapping system. In: *Proc. of Int. Conf. on Machine Control & Guidance (MCG)* (2014)
21. Loch-Dehbi, S., Dehbi, Y., Plümer, L.: Stochastic reasoning for UAV supported reconstruction of 3d building models. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* vol. XL-1/W2, pp. 257–261 (2013)
22. MacAllister, B., Butzke, J., Kushleyev, A., Pandey, H., Likhachev, M.: Path planning for non-circular micro aerial vehicles in constrained environments. In: *Proc. of IEEE Int. Conf. on Robotics and Automation* (2013)
23. Mori, T., Scherer, S.: First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles. In: *Proc. of IEEE Int. Conf. on Robotics and Automation* (2013)
24. Nieuwenhuisen, M., Droeschel, D., Schneider, J., Holz, D., Läbe, T., Behnke, S.: Multimodal obstacle detection and collision avoidance for micro aerial vehicles. In: *Proc. of European Conference on Mobile Robots* (2013)
25. Nieuwenhuisen, M., Schadler, M., Behnke, S.: Predictive potential field-based collision avoidance for multicopters. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* (2013)
26. Ok, K., Ansari, S., Gallagher, B., Sica, W., Dellaert, F., Stilman, M.: Path planning with uncertainty: Voronoi uncertainty fields. In: *Proc. of IEEE Int. Conf. on Robotics and Automation* (2013)
27. Puls, T., Kemper, M., Kuke, R., Hein, A.: GPS-based position control and waypoint navigation system for quadcopters. In: *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2009)
28. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: *ICRA Workshop on Open Source Software* (2009)
29. Ross, S., Melik-Barkhudarov, N., Shankar, K.S., Wendel, A., Dey, D., Bagnell, J.A., Hebert, M.: Learning monocular reactive UAV control in cluttered natural environments. In: *Proc. of IEEE Int. Conf. on Robotics and Automation* (2013)
30. Schmid, K., Tomic, T., Ruess, F., Hirschmüller, H., Suppa, M.: Stereo vision based indoor/outdoor navigation for flying robots. In: *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. pp. 3955–3962 (2013)
31. Tomić, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixia, I., Ruess, F., Suppa, M., Burschka, D.: Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. *Robotics Automation Magazine, IEEE* 19(3), 46–56 (2012)