Proceedings of Joint 45th International Symposium on Robotics (ISR) and 8th German Conference on Robotics (ROBOTIK), Munich, June 2014.

Hierarchical Planning with 3D Local Multiresolution Obstacle Avoidance for Micro Aerial Vehicles

Matthias Nieuwenhuisen and Sven Behnke
Autonomous Intelligent Systems Group
University of Bonn, Germany

Abstract

Micro aerial vehicles (MAVs), such as multicopters, are particular well suited for the inspection of human-built structures, e. g., for maintenance or disaster management. Today, the operation of MAVs in the close vicinity of these structures requires a human operator to remotely control the vehicle. For fully autonomous operation, a detailed model of the environment is essential.
Building such a model by means of autonomous exploration is time consuming and delays the execution of the main mission. In many real-world applications, a coarse model of the environment already exists and can be used for high-level planning. Nevertheless, detailed obstacle maps, needed for safe navigation, are often not available. We employ the coarse information for global mission and path planning and refine the path on the fly, whenever the vehicle can acquire information with its onboard sensors. To allow for fast replanning during the flight, we present a 3D local multiresolution path planning approach making online grid-based planning for our MAV platform tractable.

1 Introduction

In the last years, micro aerial vehicles (MAV) have become widely available. Due to their flexibility, they are used today for inspection and surveillance missions. In most cases, a human operator pilots the MAV remotely to fulfill a specific task or the MAV is following a predefined path of GPS waypoints in an obstacle-free altitude.

We developed a micro aerial vehicle with an omnidirectional sensor setup [20] to autonomously build mission specific semantic maps on demand. Coarse knowledge about the environment to be mapped in detail is often available prior to the mission, e.g., 3D city models acquired by land surveying authorities.

We incorporate such city models as acquired by land surveying authorities, i.e., Level-of-Detail 2 (LoD2) models containing footprint, height, and roof-shape of buildings [9]. These models do not include other structures. Thus, plans based on these representations need to be adjusted on the fly, whenever more information becomes available during a flight. Thus, we use a multi-layer approach to navigation from slower global planning to fast, reactive, local obstacle avoidance, as illustrated in **Fig. 4**. Whenever new information is acquired by the MAV's onboard sensors, the faster low-level planning and collision avoidance layers incorporate these measurements to adjust the flight plan locally (see **Fig. 1**) or trigger a higher-level replanning with the new information, if necessary. We assume, that the most obstructing structures, i.e., buildings, are known prior to a mission and most unknown obstacles are small enough to be avoided locally.

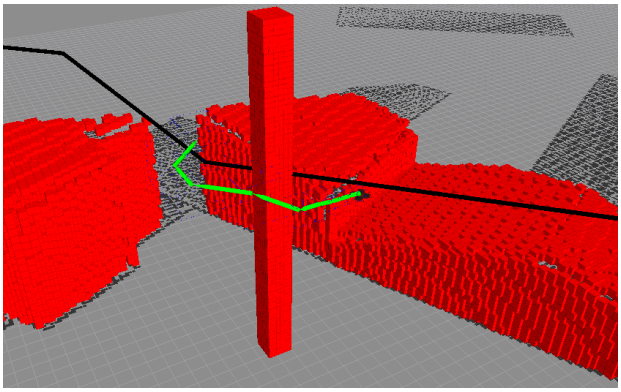


Figure 1: While following a path planned by a global planner (black lines) employing a city model, a pole not in the global model is locally avoided using local multiresolution path planning (green lines).

Our MAV is equipped with a rotating 3D laser scanner, stereo cameras, and ultrasonic sensors covering the volume around the MAV up to $30\,\mathrm{m}$ range. These sensors have only local precision. Thus, the local planner operating on maps constructed from these measurements should take account of these sensor characteristics. Furthermore, plans are more likely to become invalid in the future. We are convinced that replanning at a high frequency is more beneficial than planning fine grained for the future with local path planning. We employ 3D local multiresolution path planning, extending ideas from our prior work [1].

In this paper, we present our multi-layered approach to navigation: global mission and path planning with city models, local multiresolution path planning, and reactive

collision avoidance.

After a discussion of related work in the next section, we will briefly describe our MAV in Sec. 3. Our hierarchical control architecture from global path planning to low-level control is outlined in Sec. 4. In Sec. 5, we detail our approach to 3D local multiresolution path planning and present evaluation results in Sec. 6.

2 Related Work

The application of MAVs in recent robotics research varies especially in the level of autonomy—ranging from basic hovering and position holding [3] over trajectory tracking and waypoint navigation [22] to fully autonomous navigation [10]. A summary on autonomy levels is given in [17].

Particularly important for fully autonomous operation is the ability to perceive obstacles and avoid collisions. Obstacle avoidance is often neglected, e.g., by flying in a sufficient height when autonomously flying between waypoints. Most approaches to obstacle avoidance for MAVs are camera-based due to the limited payload [19, 24, 25]. Hence, collision avoidance is restricted to the narrow field of view (FoV) of the cameras.

Other groups use 2D laser range finders (LRF) to localize the aerial vehicle and to avoid obstacles [10], limiting obstacle avoidance to the measurement plane of the LRF, or combine LRFs and visual obstacle detection [26, 16]. 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. [4].

A two-level approach to collision-free navigation using artificial potential fields on the lower layer is proposed in [21]. Similar to our work, completeness of the path planner is guaranteed by a layer on top of local collision avoidance. Another reactive approach modifies trajectories locally by finding via points that move the trajectory away from obstacles [14].

Some reactive collision avoidance algorithms for MAVs are based on optical flow [8] or a combination of flow and stereo vision [15]. 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 [18, 5]. A good survey on approaches to motion planning for MAVs is given in [7]. These methods assume complete knowledge of the scene geometry—an assumption that we do not make here.

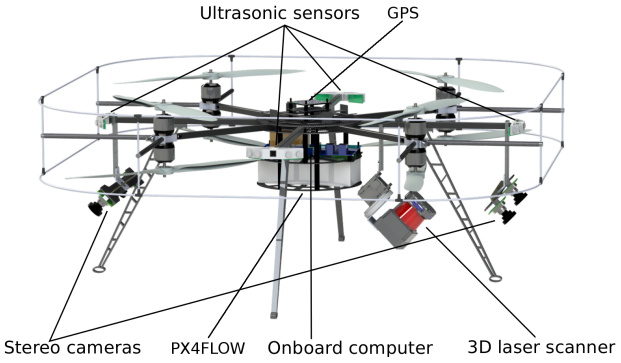


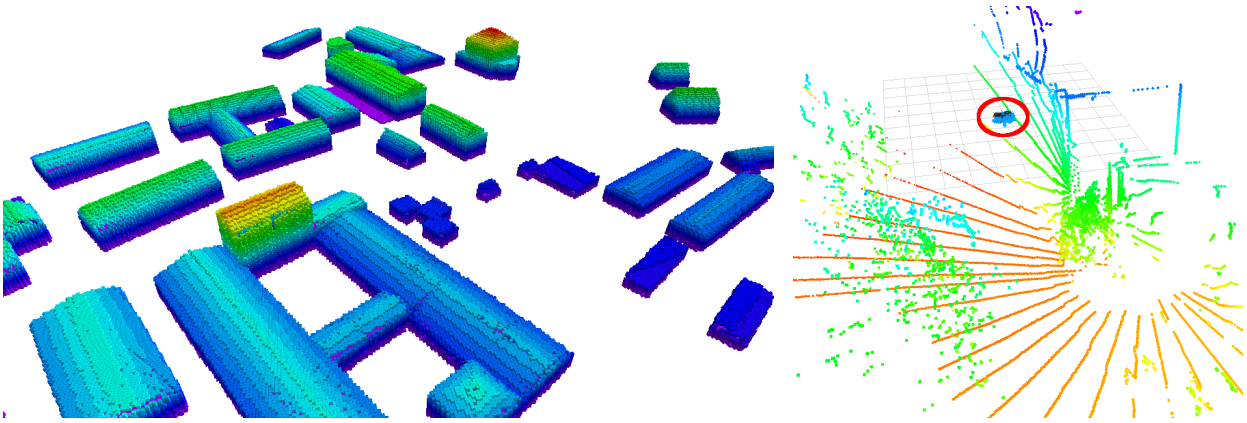
Figure 2: Our MAV includes a continuously rotating 3D laser range finder as main sensor, two stereo camera pairs, and a ring of ultrasonic distance sensors to detect small obstacles in the proximity. Position and velocities are determined by means of a precise GPS system and an optical flow camera. Eight co-axial mounted rotors provide enough thrust to lift the 5 kg vehicle.

3 Micro Aerial Vehicle

Our sensor setup aims at perceiving as much of the MAV's surroundings as possible in order to obtain almost omnidirectional obstacle perception. We make use of two stereo camera pairs (one pointing forward, one pointing backwards) and a tilted continuously rotating 3D laser scanner for perceiving the environment in all directions. Depending on the direction, the measurement density of the 3D laser scanner varies and has its maximum in a forward-facing cone. Only a small portion above the MAV's back is shadowed. In addition, eight ultrasonic sensors are mounted in a ring around the MAV. Although both range and accuracy of ultrasonic sensors are very limited, they are very well suited for perceiving even small obstacles in the vicinity such as tree branches, overhead power cables, and transmission lines. For localization and state estimation, we use an optical flow camera [12] in addition to the two stereo camera pairs and the 3D laser scanner. It is pointing downwards to the ground and can—depending on the lighting conditions—measure velocities relative to the ground-plane with more than 100 Hz. For a detailed description of our sensor setup and the processing pipeline see [20]. Our platform is based on the open source *MikroKopter* octocopter kit, with a co-axial arrangement of rotors (see **Fig. 2**). The onboard computer (Intel Core i7-3820QM, 2.7 GHz) has ample computing power for tasks of advanced complexity and the variety of sensors. As middleware, we employ the Robot Operating System ROS [23].

4 Control Architecture

We designed a hierarchical control architecture for our MAV, with high-frequency controllers on the lower layers



(a) OctoMap derived from the 3D city model.

(b) Local laser scan from the environment, including a building (right) and vegetation (left). The MAV's self measurements are in the red circle.

Figure 3: For many MAV missions, coarse knowledge about the environment is available. We employ a 3D city model as provided by land surveying authorities (left) for global path planning and replan continuously during the MAV's flight incorporating onboard sensor measurements, e.g., from a 3D laser scanner (right).

and slower planners on the upper layers, that solve more complex path and mission planning problems at a lower frequency (see Fig. 4).

The topmost layers in our control concept are a mission planner and a global path planner. Both use a static representation of the mission area known in advance. These layers specify a flight plan composed of a list of waypoints the MAV should pass approximately or reach exactly, depending on the mission's objectives. The global environment is represented efficiently in an OctoMap [13]. As the environment model (at this point) does not change during a flight, we trigger the global planner at the beginning of the mission. Replanning is necessary only, if unforeseen obstacles block mission goals and local planning is not sufficient to find a feasible detour. In that case, the global map can easily be updated with local sensor measurements and replanning can be triggered. The planned global path is fed as input to the next layer, the local path planner. This layer will be described in the next section.

On the lowest layer, we employ a high-frequency potential field-based local obstacle avoidance as a safety layer while following paths. This layer helps to avoid small obstacles and to prevent collisions with the perception frequency. For this purpose, we extend standard artificial potential fields as these can be evaluated at the same frequency as obstacle perceptions arrive. In general, the robot is modeled as a particle passively moving through a field induced by attractive (towards an intermediate goal) and repulsive forces (induced by obstacles). The resulting force that determines the motion direction is now calculated from these forces. In contrast to relatively slow moving ground robots, MAVs cannot stop immediately. They are able to change their dynamic state completely within a tightly bound time horizon, though. We use this property and predict the trajectory of the MAV for the near future, i.e., the time the

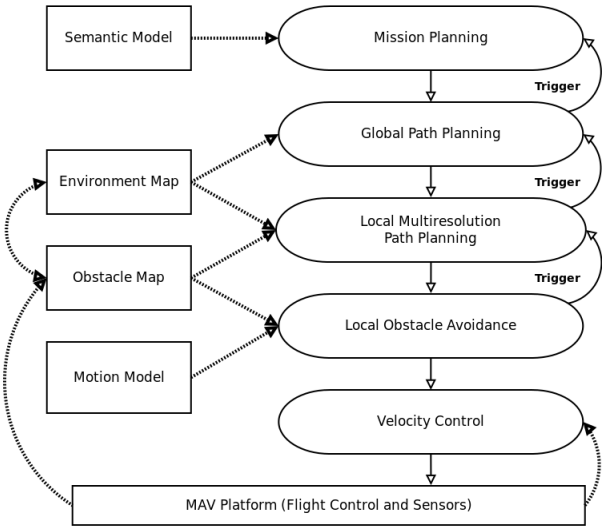


Figure 4: The control concept of our MAV is a hierarchical control architecture with slower planning layers on the top and faster control layers on the bottom. A global path planner ensures planning completeness, a local path planner incorporation of local sensor data, and a fast obstacle avoidance layer selects appropriate velocity commands. These are fed to low-level MAV controllers. Commands are depicted by solid lines, data flow is depicted by dotted lines.

MAV needs to stop, given the artificial potential field and the current motion estimate.

If the vehicle will approach obstacles too close, this safety layer can reduce the MAV's velocity or stop it completely. The local obstacle avoidance commands ego-centric linear velocities to the low-level control layer. The attitude is controlled by the MAVs onboard microcontroller. For staying at a position, we employ a hover controller [2].

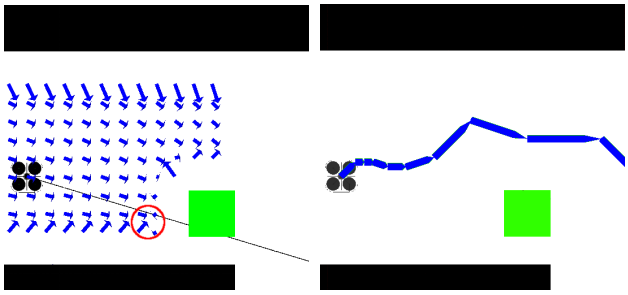


Figure 5: Example situation with a given map and an apriori unknown obstacle perceived with onboard sensors (green). Whereas our reactive collision avoidance (left) is able to avoid the obstacle, without global replanning the MAV will remain in the local minimum (red circle). Our local path planner is capable to deviate from the global path and to proceed without global replanning (right).

Due to the reactive nature of this layer, it is prone to get stuck in local minima in more complex situations. **Fig. 5** shows a situation where the reactive approach would require global replanning with the newly acquired information. In this paper, we extend our prior work [20] with a robot-centric local path planning layer refining the globally consistent path by using an excerpt of the global map and a robot-centered local obstacle map containing measurements from onboard sensors to reduce the necessity for global replanning. An example of a laser scan is depicted in **Fig. 3b**. We will detail our approach in the next section.

5 3D Local Multiresolution Path Planning

Between global path planning and reactive obstacle avoidance, we employ local multiresolution path planning. This layer plans to intermediate goals derived from the globally planned path, incorporating an excerpt from the global environment map and the local obstacle map acquired with onboard sensors. Replanning is performed at a rate similar to the frequency of full updates of the local obstacle map, which is 2 Hz with our setup.

In contrast to the global environment map, the accurateness of the robot-centered local obstacle map decreases with increasing distance to the MAV. The onboard sensors measure distant obstacles more coarsely than closer ones and small errors in the measurement of the MAVs attitude cause larger deviations in the distance. Furthermore, the uncertainty in the MAV's motion makes exact long-term plans infeasible. Hence, we have chosen a grid-based planning representation that models the volume in the vicinity of the MAV at a fine resolution and decreases the resolution with increasing distance to it.

Our representation consists of multiple robot-centered 3D grids of size M×M×M. The innermost grid has grid cells with edge length s. Recursively, these grids are embedded into the next coarser grid with cells with a doubled edge length of $2s$. The inner grid covers the part $\left[\frac{M}{4} : \frac{3M}{4}\right]^3$ of the grid at the next coarser level. A cut through the resulting multiresolution grid is depicted in **Fig. 6**. In order to cover the same volume as a uniform N×N×N grid, a multiresolution grid with M^3 cells per grid contains $(\log_2(N/M) + 1)\, M^3$ cells in total.

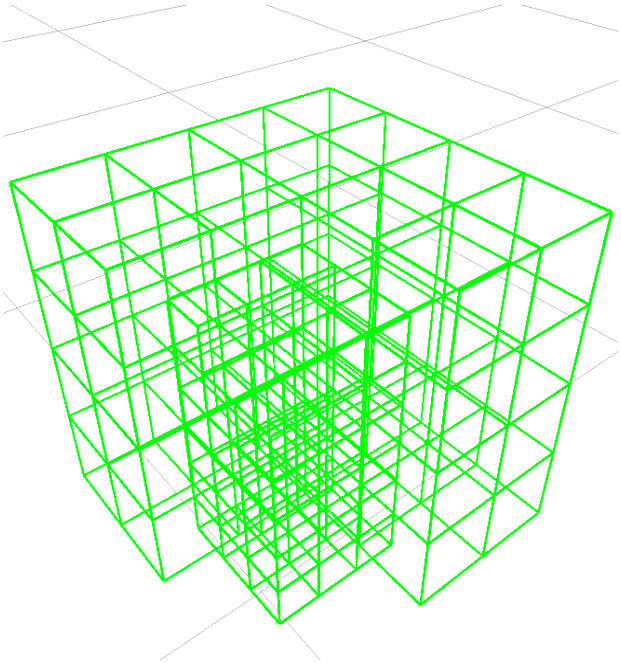


Figure 6: Cut through a local multiresolution grid map. Multiple grids of different resolutions are embedded into each other to represent the environment close to the current MAV's position fine—and coarser with increasing distance.

For planning within this grid, we embed an undirected graph. Grid cells are connected to all surrounding neighbors, but in contrast to uniform grids, the number of neighbors differs between 14 and 33. In the robot-centered grid, the search does not start within an actual grid cell, but in an extra start node connected to the eight innermost cells. We use the A* graph-search algorithm for path planning [11]. The costs to traverse an edge are given by the obstacle costs of the cells it is connecting and its length given by the Euclidean distance between the cell centers. The obstacle costs are multiplied by the fraction of the edge length within the respective cells. Obstacle costs are calculated according to the obstacle model depicted in **Fig. 7**: a core of the perceived obstacle enlarged by the approximate robot's 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 employed heuristic is the Euclidean distance to the center of the goal cell.

The advantages of this representation are the low memory requirements and the inherent representation of uncertainties in sensing and motion of the MAV.

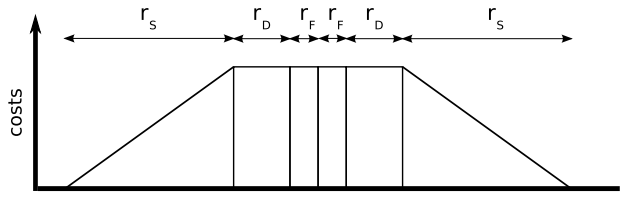


Figure 7: Obstacles in the local multiresolution grid are modeled as a fixed core (enlarged by the robot's radius) r_F and a distance dependent part r_D with maximum costs and a safety margin r_S with decreasing costs.

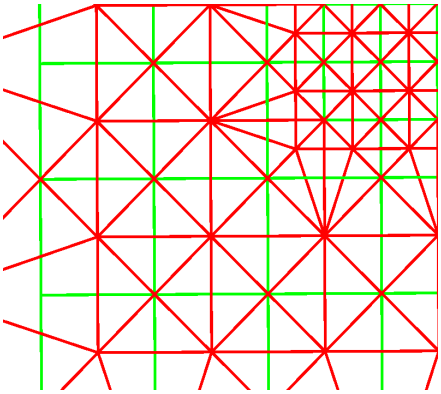


Figure 8: Planning is performed in a robot-centered multiresolution grid (green). The red lines depict the graph used for path planning (shown only in 2D).

5.1 Efficient Map Maintenance

To build a local obstacle map, we aggregate distance measurements into a hybrid representation, storing occupancy information and corresponding 3D points in each cell [6]. Point measurements of consecutive 3D measurements are stored in fixed-size circular buffers, allowing for point-based data processing, without keeping a long history of point clouds in memory. Old measurements are discarded when necessary. To maintain the property of a robot-centered map, the grid has to be translated regularly. To perform these translations efficiently in constant time, the cells are stored in circular buffers. As always the most recent points are kept, the map remains locally consistent.

6 Evaluation

We evaluate the computing time and the resulting flight trajectories using our 3D local multiresolution path planner in simulation. For this purpose, we plan a global path using our 3D city model in a uniform grid and add obstacles that are only perceived within the MAV's sensor range. The local multiresolution grid used for local planning has a minimal cell size of $(0.25\,\mathrm{m})^3$ and 8^3 cells per grid. It covers a total volume of $(64\,\mathrm{m})^3$ with 6 resolution levels and a total of 3,072 cells. We compare the local multiresolution approach with two uniform grids covering $(40\,\mathrm{m})^3$, one with a cell size of $(0.25\,\mathrm{m})^3$ and one with a cell size of $(1\,\mathrm{m})^3$. This results in $4,096,000$ and $64,000$ cells, respectively. All timings are measured on the MAV's onboard computer. The three evaluated grid representations perform equally well if the globally planned path can be followed. Obviously, the exploration of the grid is perfectly directed by the heuristic leading to a neglectable linear increase in the planning time with increasing grid resolution. In the case unknown obstacles are perceived on this path, the MAV must deviate from it to surround the obstacles. Hence, the exploration of the grids becomes broader while approaching the obstacle. **Fig. 9** shows the effect on the planning time per planner iteration for a uniform and the multiresolution grid. In the experiments, the planners are running at a target frequency of $4\,\mathrm{Hz}$. The resulting time window of $0.25\,\mathrm{s}$ for local planning is substantially exceeded when using the uniform grid. The planning time peak is at approximately $3.4\,\mathrm{s}$ for a single iteration, rendering this planner unfeasible for continuous replanning with this grid resolution. By reducing the resolution by a factor of 4^3, the uniform planner is able to replan as fast as the local multiresolution planner. The minimal and maximal planning times per iteration are summarized in **Tab. 1**.

Tab. 1 also shows a comparison of the resulting path lengths in the vicinity of an unknown obstacle. We normalize all path lengths to the paths planned by the high resolution uniform planner. Using the planner with the coarse uniform grid results in 9% longer paths. Using our proposed local multiresolution grid results in 3% longer paths, as the close vicinity of the MAV is always represented as high resolution grid.

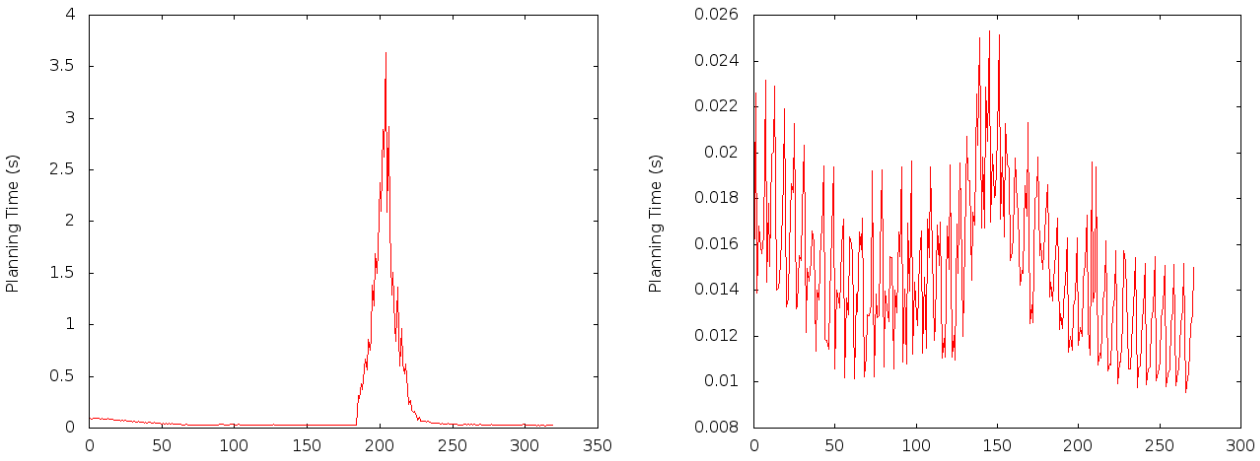
Table 1: Planning time (in milliseconds) and normalized lengths of resulting trajectories.

grid representation	cell size (in m)	planning time min.	planning time max.	length
multiresolution	0.25	12	35	1.03
uniform	0.25	26	3395	1.00
uniform	1.00	4	20	1.09

7 Conclusion and Future Work

To operate MAVs safely in the vicinity of structures fast reactions on new obstacle perceptions are required. We approach this challenge by employing local multiresolution techniques that facilitate frequent replanning to refine the path locally based on onboard sensing, supported by a reactive collision avoidance layer.

For planning paths for MAV missions, a coarse model of the buildings in the environment can help to plan a global path. The remaining obstacles are often trees and smaller human-built structures like power cables. These obstacles can be avoided using a plan refinement based on a local map acquired with onboard sensing. With local multireso-



(a) Uniform Grid (0.25 m)

(b) Multiresolution Grid

Figure 9: Plot of planning times per iteration. The MAV is approaching an obstacle where it has to deviate from the globally planned path. Top: If the uniform grid with $(0.25\,m)^3$ cell size is used the planning time is exceeded substantially. Bottom: The local multiresolution planner is able to replan without a strong increase in planning time.

lution grids, continuous replanning is feasible. Due to the fine map resolution in the vicinity of the MAV, the planned paths are only slightly longer than paths planned in a uniform grid.
In future work, we will use multiresolution on the global path planning layer and extend all planning layers to reuse plans if possible or repair them otherwise. This will further reduce the planning time and still allow to react quickly on newly perceived obstacles.

Acknowledgment

This work has been supported partially by grant BE 2556/8-1 of the German Research Foundation (DFG).

References

[1] S. Behnke. Local multiresolution path planning. In *RoboCup 2003: Robot Soccer World Cup VII*, volume 3020 of *Lecture Notes in Computer Science*, pages 332–343. 2004.

[2] M. Beul, R. Worst, and S. Behnke. Nonlinear model-based 2D-position control for quadrotor UAVs. In *Proceedings of the Joint Int. Symposium on Robotics (ISR) and the German Conference on Robotics (ROBOTIK)*, 2014.

[3] S. Bouabdallah, P. Murrieri, and R. Siegwart. Design and control of an indoor micro quadrotor. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2004.

[4] A. Chambers, S. Achar, S. Nuske, J. Rehder, B. Kitt, L. Chamberlain, J. Haines, S. Scherer, , and S. Singh. Perception for a river mapping robot. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[5] H. Cover, S. Choudhury, S. Scherer, and S. Singh. Sparse tangential network (spartan): Motion planning for micro aerial vehicles. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[6] D. Droeschel, J. Stückler, and S. Behnke. Local multi-resolution representation for 6D motion estimation and mapping with a continuously rotating 3D laser scanner. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

[7] C. Goerzen, Z. Kong, and B. Mettler. A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *Journal of Intelligent and Robotic Systems*, 57(1-4):65–100, 2010.

[8] W. Green and P. Oh. Optic-flow-based collision avoidance. *IEEE Robotics and Automation Magazine*, 15(1):96–103, 2008.

[9] G. Gröger, T. H. Kolbe, A. Czerwinski, and C. Nagel. Opengis city geography markup language (citygml) encoding standard. *Open Geospatial Consortium Inc. Reference number of this OGC® project document: OGC*, 2008.

[10] S. Grzonka, G. Grisetti, and W. Burgard. A fully autonomous indoor quadrotor. *IEEE Trans. on Robotics*, 28(1):90–100, 2012.

[11] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[12] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys. 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 (ICRA)*, 2013.

[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013.

[14] S. Hrabar. Reactive obstacle avoidance for rotorcraft UAVs. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[15] S. Hrabar, G. Sukhatme, P. Corke, K. Usher, and J. Roberts. 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 (IROS)*, 2005.

[16] S. Huh, D. Shim, and J. Kim. 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 (IROS)*, pages 3158–3163, 2013.

[17] F. Kendoul. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 22(2):315–378, 2012.

[18] B. MacAllister, J. Butzke, A. Kushleyev, H. Pandey, and M. Likhachev. Path planning for non-circular micro aerial vehicles in constrained environments. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[19] T. Mori and S. Scherer. 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 (ICRA)*, 2013.

[20] M. Nieuwenhuisen, D. Droeschel, J. Schneider, D. Holz, T. Läbe, and S. Behnke. Multimodal obstacle detection and collision avoidance for micro aerial vehicles. In *Proc. of European Conference on Mobile Robots (ECMR)*, 2013.

[21] K. Ok, S. Ansari, B. Gallagher, W. Sica, F. Dellaert, and M. Stilman. Path planning with uncertainty: Voronoi uncertainty fields. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[22] T. Puls, M. Kemper, R. Kuke, and A. Hein. GPS-based position control and waypoint navigation system for quadrocopters. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[23] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[24] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert. Learning monocular reactive uav control in cluttered natural environments. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[25] K. Schmid, T. Tomic, F. Ruess, H. Hirschmüller, and M. Suppa. Stereo vision based indoor/outdoor navigation for flying robots. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3955–3962, 2013.

[26] T. Tomić, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka. Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. *IEEE Robotics and Automation Magazine*, 19(3):46–56, 2012.