2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Mapping and Path Planning in Complex Environments: An Obstacle Avoidance Approach for an Unmanned Helicopter

Franz Andert, Florian Adolf, Lukas Goormann, and Jörg Dittrich
German Aerospace Center (DLR), Institute of Flight Systems
38108 Braunschweig, Germany
{*franz.andert, florian.adolf, lukas.goormann, joerg.dittrich*}*@dlr.de*

Abstract— **This paper presents an obstacle avoidance method that is performed with an unmanned helicopter. The approach begins with a mapping step where information from sensor data about previously unknown dangers is extracted into an occupancy grid and eventually converted into a polygonal 3D world model. This continuously updating map is used by a path planner that generates and updates a 3D trajectory guiding the vehicle through safe passages around the detected objects. The algorithms are generic but optimized for unmanned aircraft and a stereo camera as the environmental sensor. Computation is fully executed on board so that a ground control station is only needed for supervision. With successful obstacle detection and avoidance flight tests, the paper shows the qualification of the presented method under real operational conditions.**

I. Introduction

Applications at low altitudes and in unknown or only partially known areas are of high importance for unmanned aerial vehicles. Typical missions beyond line of sight allow only limited manual control due to interrupted and delayed communication with the ground control station. These conditions necessitate safe vehicle operation even without direct command inputs. When autonomous flights in the vicinity of unknown dangers are to be performed, a collision-free path towards a desired target position must be maintained. This refers to obstacle detection and avoidance for urban scenarios, which has to be computed onboard the vehicle.

Many research activities focus on reactive behaviors, especially when small aircraft with limited sensor payload and onboard processing capabilities are used. Inspired by insects, optical flow is an easy method to gather information about near objects with lightweight passive camera sensors, which makes it very popular to control vehicles through canyons or to keep the ground distance constant over rough terrain. For example, mature flying applications are presented in [1] and [2]. Other research activities improve the obstacle detection capabilities with stereo vision [3] or laser range finders [4].

Larger aerial vehicles allow more complex control concepts due to increased payload capabilities for avionics, sensors, and computation. For example, trajectories achieved by model-predictive control [5] or collision cone approaches [6] tend to be smoother and more useable for fragile payload than the insect-like ones from optical flow.

Furthermore, path planning methods are developed for autonomous flights. The goal is to generate an optimal and collision-free path towards a desired target, considering known [7] or a-priori unknown obstacles [8] as well as flight restrictions and an overall integration to a complex mission plan [9]. One well-engineered approach for an unmanned helicopter is presented in [10] where a local and a global path planner are combined with the detection and mapping of dangerous objects. According to these research investigations, this paper focuses on mapping and path planning as a concept for autonomous obstacle avoidance. Details of the main contribution are described in section III.

II. Research Platform

The DLR Institute of Flight Systems operates several types of unmanned aircraft for aeronautic and flight robotics research. Here, the test vehicle is an auto-piloted model helicopter (fig. 1) with all computational power on board. The avionics payload comprises GPS, IMU and magnetometer for navigation, sonar altimeter for landing, wireless data links, manual R/C and video data links, a telemetry module and a flight control computer. See [11] for further details. Environmental sensing is performed with a stereo camera (Videre Design STOC, 30 cm baseline, 640×480 pixels, max. 30 fps, lens: $50^\circ \times 40^\circ$ field of view). For fast image processing and mapping, the helicopter is equipped with a separate computer (Intel Core 2, 1.5 GHz, PC/104 board).

Fig. 1. The research UAV ARTIS (Autonomous Rotorcraft Testbed for Intelligent Systems) in an urban area. It is a combustion-engine-powered autonomous helicopter with 1.9 m rotor diameter and 12 kg gross weight.

Basis for robotic applications are a navigation filter providing the flight state, and a flight controller that allows precise vehicle guidance and the input of high-level commands from external applications. Note that the problem of GPS signal dropouts is not tackled within this paper. Obstacle avoidance flights do not require controller modifications. Here, the algorithms presented in this paper act as the application that triggers waypoint position instructions to the autopilot.

III. THE OBSTACLE AVOIDANCE CONCEPT

Autonomous flights are planned as illustrated in figure 2. The flight is initialized by the operator that specifies a set of waypoints to be passed by the vehicle. The path planning system automatically generates an initial collision-free path before the flight, where a-priori knowledge about dangers is considered. While the vehicle is flying along this initial path, it is gathering new information about obstacles. A map is incrementally extended with each new set of sensor data. Objects and probable collisions can be detected and in that case, the flight path has to be updated. Guiding the vehicle along the new path avoids the obstacle and ensures that the desired targets are reached if possible. Multiple path updates which are most likely in complex environments are handled.

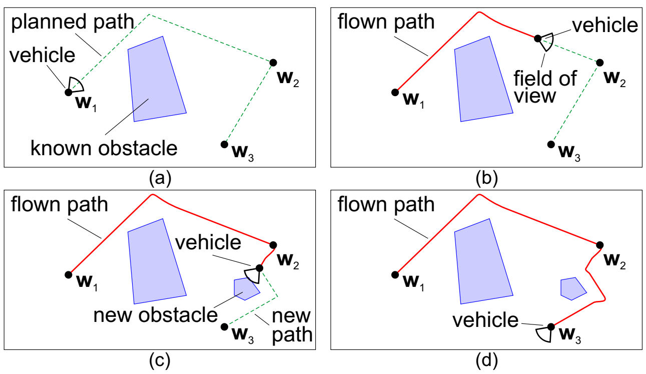

Fig. 2. Concept for autonomous flights (2D simplification). Initial path (a), vehicle guidance if no additional objects are detected (b), updated path around the detected obstacle (c), and mission ending at the last waypoint (d).

With that, the challenge of avoiding obstacles in potentially unknown or outdated areas leads to two problems that are tackled in the next two sections of this paper:

- *Mapping:* Interpret sensor data to update geodetic environmental information.
- *Path planning:* Plan a path based on this map. When the map is updated, check for collisions on the actual planned path and if necessary, repair the path plan efficiently.

Figure 3 illustrates details of the process that consists of a mapping step that identifies stationary objects, and a path planning step that uses the objects that have been identified or are initially given. Different types of environment representations optimized for specific tasks are introduced throughout this paper. The world model itself (layer 3 in the figure) acts as an interface between the two main steps mapping and path planning. In the presented testing setup, these two main steps are executed on different computers and hence, world model updates must be compact enough to be sent over networks with limited bandwidth. As an additional benefit, this property would also allow map update exchange for multiple vehicles in the future. Eventually, the output of the path planner is a regularly updated and smoothed path. Flights along path segments are handled by the flight control system. The functionality of mapping, path planning, and their integration into complete obstacle avoidance is tested in flight, described in section VI.

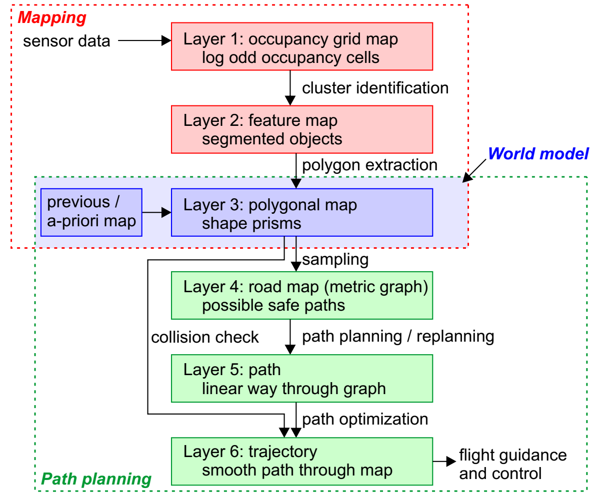

Fig. 3. The general sensor-based path planning process divided into a mapping and a path planning step and different layers of environmental representation.

IV. STEREO-BASED ONBOARD MAPPING

A. Combining Different Map Types

In general, the mapping procedure converts sensor data into a compact three-dimensional representation of potential dangers. This basically requires to handle noisy and erroneous input data and to calculate updates in real-time because dangers have to be avoided immediately after detection. To fulfil requirements from both, sensor fusion and application purposes, it was chosen to combine different map types being advantageous for different sub-tasks as described in a very large amount of robotic research publications, see e.g. [12] for an overview. The idea is to use a 3D occupancy grid for sensor fusion and to convert its incremental updates into a compact polygonal representation of the occupied area. It should be noted that other world model formats (e.g. octrees) may be suitable for this obstacle avoidance concept as well.

Here, the mapping procedure works as follows:

1) Initialize an occupancy grid with "unknown" occupancy. If possible, insert a-priori information.
2) Insert the actual sensor data information into the grid.
3) Find clusters of occupied grid cells and mark them as single obstacle features.
4) Find out which obstacle features are new and which are updates of objects that have already been identified in the previous loop cycle. Preexisting objects may also be removed.
5) Calculate the shape of each new or updated feature and send updates to the path planner.
6) To insert the next sensor data, go back to step 2.

This approach uses different map types as already illustrated in figure 3. The occupancy grid map for sensor inputs (layer 1) is segmented (layer 2) and finally used to generate geodetic obstacle shape information (layer 3). Grid resolution is user-defined and not changed over time when processing image series. For further details and a description how to handle large area sizes, see [13]. Figure 4 shows with an example how sensor data is processed into a polygonal world model.

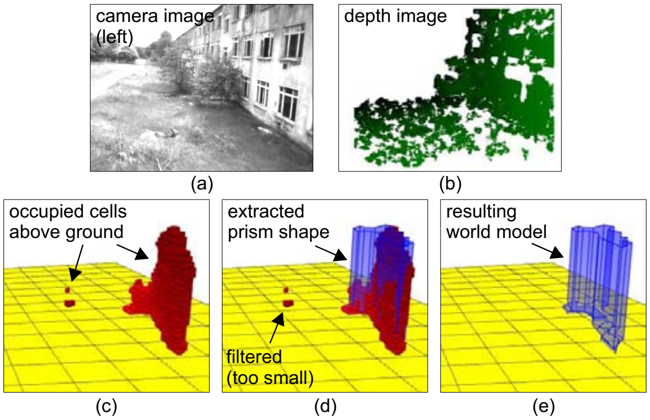

Fig. 4. Example of an intensity (a) and depth image (b) taken during a helicopter flight. In the depth image, darker areas refer to far distances, missing or filtered depth values are white. Occupancy grid mapping results (c), overlain prism shapes (d) and final map output without grid (e).

B. Grid Mapping from Sensor Data

A geodetic 3D grid map with log odd occupancy values is created incrementally using stereo depth images, the manually measured camera alignment on the vehicle, and the vehicle orientation from the navigation filter. With the given camera configuration, distances up to 40 m are taken into account. For fast mapping and to minimize the chance of errors in outdoor environments without significant structures close to the vehicle, indoor-qualified map data registration (i.e. SLAM) is not used. Instead of that, other research activities focus on improving flight state and camera misalignment estimation with visual methods independently.

The obstacle identification is a search for clusters of connected occupied cells. To mark occupied cells, the grid is segmented using a threshold for the minimal occupancy probability value. After that, a cluster can be found with a flood fill that is seeded at an occupied cell. This is done until every occupied cell belongs to a cluster. Each cluster is stored with its bounding box and its binary cell shape.

C. World Model Extraction

Now, the shape is calculated for each obstacle feature identified in the grid map that passes a minimum size threshold. For a very fast calculation of polygonal object shapes, the presented approach models the shape of an object with bounding prisms with horizontal bases. Since only one bounding prism may be too rough to represent a complex shape, the cell-based shape is initially partitioned into horizontal slices with the height of one cell as illustrated in figure 5. Such a slice can be represented by a 2D cell array. Similar superposed slices are merged using a logical disjunction of occupied values in different heights so that the resulting multi-slices can be represented by a 2D cell array as well. Shape prisms are calculated for each multi-slice which yields a generation of one or multiple prisms for each object.

Prism shape calculation is based on binary 2D-image algorithms as illustrated in figure 6. First, the contour cells are calculated for each multi-slice using the tracing algorithm presented in [14]. This algorithm generates a list of an object's contour pixels which can be interpreted

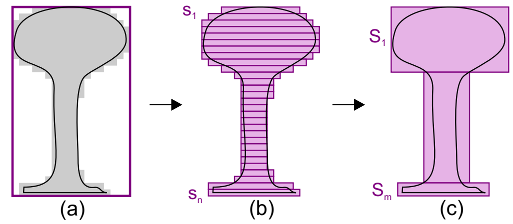

Fig. 5. Grid shape and bounding box of feature (a), single slices (b) and merged multi-slices (c). Side view of a tree as an example.

as a polygon vertex by using the corresponding grid cell center coordinates. To compress the object representation, the polygon is compressed with the algorithm of [15]. This step removes dispensable vertices with 180° interior angle and approximates the contour with straighter edges that need a lower number of vertices. The result is shown in figure 6d.

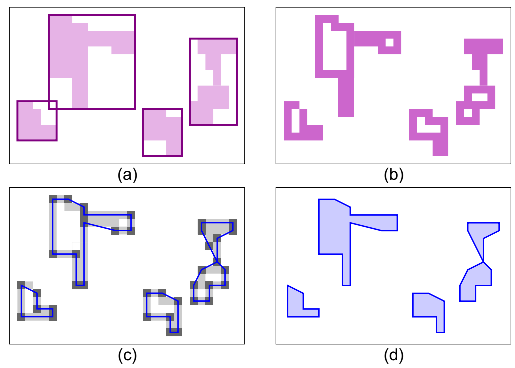

Fig. 6. Shape Extraction, top view. Cell-based shapes (a), contours (b), approximated polygons with its vertices (c) and final ground shapes (d).

This world model with prism shapes is now sent to the path planner that will generate collision-free paths. Since it is generated incrementally and in real-time onboard the vehicle, updates are generated with every new set of sensor data, i.e. with every processed depth image. Beside the generation of prisms, the ground plane is extracted in a different way which is described in [13].

V. Adaptive Any-Time Path Planning

The presented approach avoids common local planning and avoidance strategies to prevent the system from deadlocks e.g. caused by local minima, and to ensure that desired targets are reached if possible. Here, an initial path based on the a-priori environmental knowledge is generated before flight. During the flight, an updated collision-free path between the current vehicle position and the next target waypoint is calculated if it is required due to new obstacles inserted into the world model. At this stage, planned and updated paths consist of linear segments between nodes where the vehicle has to stop and turn while hovering.

A. Road Map Sampling and Updating

In this context, a road map is a graph representation of the environment providing safe positions by its vertices and collision-free passages by its edges. Road maps are sampled automatically from a polygonal world model and used as a first step to search for suitable paths using graph theory. Since the quality of a resulting path highly depends on the quality of the road map, this graph must be a well-qualified representation of free space. It must connect all reachable areas, cover narrow passages as well as large free space areas and allow efficient paths through any kind of environment. Furthermore, an efficient path search must be possible when the environmental knowledge is updated.

There are many sampling methods using a lattice-based or a probabilistic distribution of graph vertices. See e.g. [16] for a comparison of some common algorithms. Tests have shown that a quasi-random road map (QRM) sampling strategy, see [17], provides best results for the desired purposes and is used here to generate a graph from a given polygonal world model. To generate paths from and to arbitrary coordinates, the actual vehicle position and previously given target waypoints are inserted into the road map including their connection to neighboring vertices. A fast method to consider map changes due to new detected obstacles is a removal of vertices and edges that collide with them. This must be followed by a graph repair method if connectivities are lost. Figure 7 illustrates this principle of graph sampling and updating.

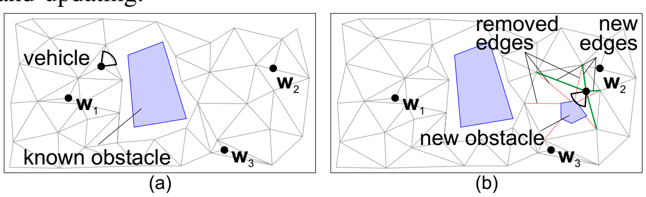

Fig. 7. Initial road map considering waypoint and vehicle positions (a), updated road map after obstacle detection (b) with removed edges, and inserted edges from the actual vehicle position. 2D simplification.

B. Graph Search for Path Planning

For the initial path search, the A* algorithm [18] is applied to the initially sampled road map. Result is a path along the given vertices that begins at the actual vehicle position and passes all waypoints in the desired order. Replanning in the same graph, e.g. after a manual insertion of waypoints, is done with the anytime-capable AD* algorithm [19], which provides a faster solution by re-using previous solutions and by improving the result over time if additional calculation time is available. This solution provides more remaining calculation time useable for a complete new replanning between two succeeding waypoints that is done when the graph is changed by the detection or removal of obstacles.

Since a path within a road map is limited to the given edges, path optimization methods become useful. Road map paths may include circuitous routes with unnecessary corners and hardly achievable vehicle trajectories due to the piecewise linearity of the path that results in instant changes of the vehicle's heading.

The first problem is solved with a search for short cuts in the output of the graph search, see figure 8. This simplification removes a given node if its predecessor and successor are connected by a collision-free path segment. The collision checks must refer to the polygonal map again since the new segments are not part of the initial road map which contains no obstacle information. The second problem is easy to solve when using a helicopter by simply stopping the vehicle on each corner and setting the correct heading during the hovering phase. With that, any kind of path that consists of linear segments can be flown.

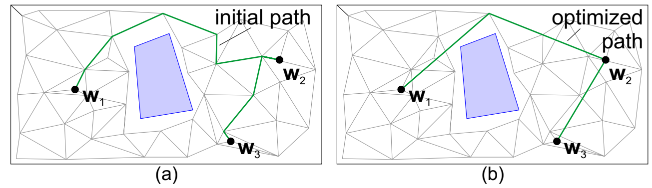

Fig. 8. Path along the road map, generated with the A* or AD* algorithm (a), simplified path after removing unnecessary corners (b).

VI. Flight Test and Evaluation

While a detailed but separated evaluation of the two components mapping and path planning has already been published [20], the focus of this paper is on the tests showing their integration to avoid obstacles in the real world. Nevertheless, mapping is presented here for a better understanding.

A. Mapping Test

As a preliminary evaluation, the mapping approach is tested in outdoor flights under real operational conditions. In the presented example, the 3D grid resolution is set to 0.5 m which provides mapping between 15 and 20 Hz.

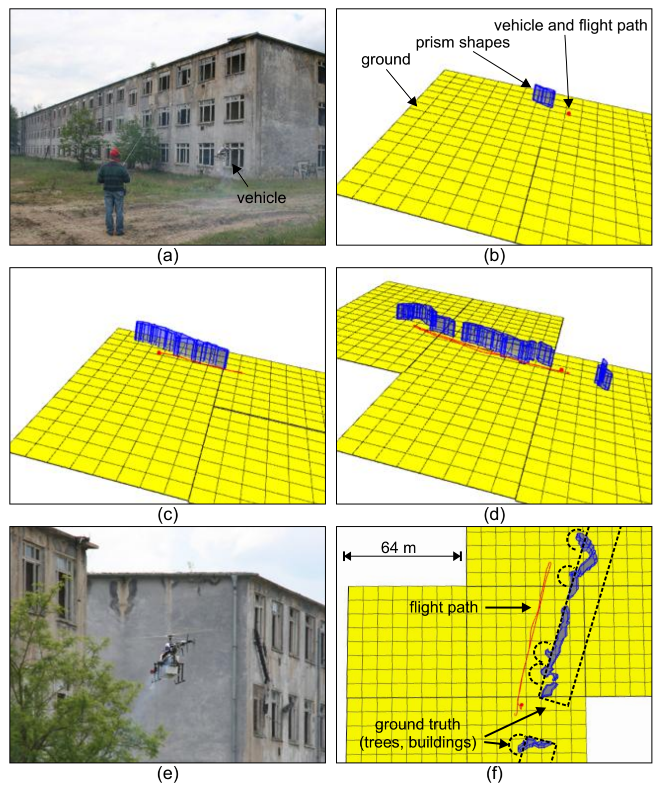

Fig. 9. Incremental map building during a helicopter flight in an urban area. Helicopter and the safety pilot for supervision in the beginning of the mapping flight (a), snapshots of the mapping process (b-d), photo of the helicopter at the ending position (e), top view in comparison with ground truth (f). Buildings are shown with original size, trees are marked as circles.

Figure 9 illustrates the incremental creation of the world model with prism shapes and ground planes. Subfigure (a) shows the helicopter in an urban terrain at the beginning of the flight. Mapping starts with onboard image acquisition as seen in (b) where confident obstacles are already detected.

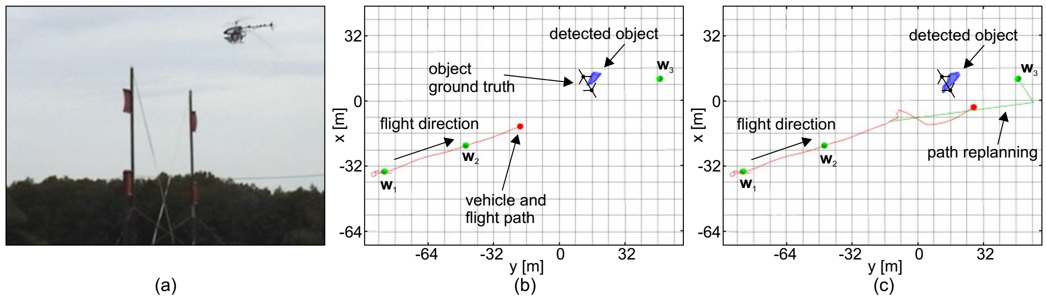

Fig. 10.1: Autonomous flight at a test field. Image of the helicopter during flight (a), map and flight path at the time the object appears in the map, top view (b), map and flight path during the obstacle avoidance maneuver, top view (c).

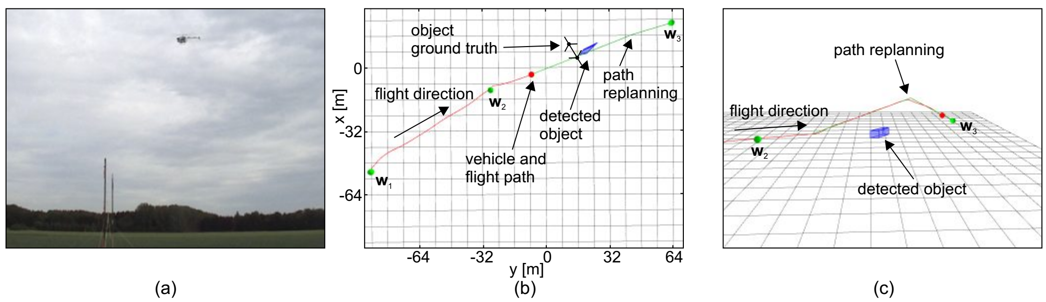

Fig. 10.2: Autonomous flight at a test field. Image of the helicopter during flight (a), map and flight path at the time the object appears in the map, top view (b), map and flight path during the obstacle avoidance maneuver, side view.

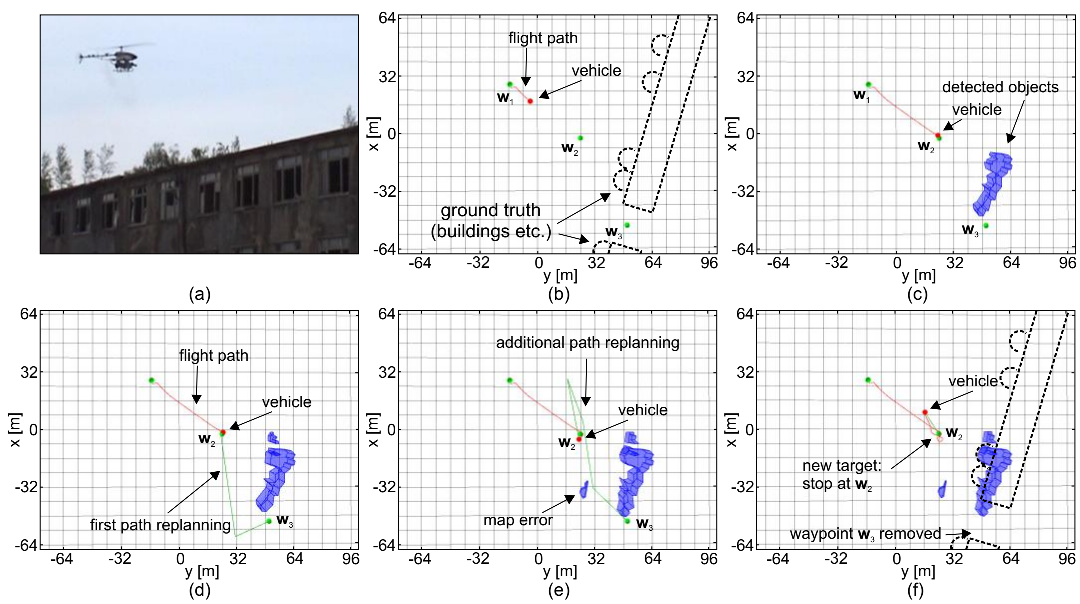

Fig. 10.3: Autonomous flight in an urban area. Image while approaching the buildings (a), initial empty map while approaching (b), map status shortly before reaching waypoint $\mathbf{w}_2$, (c), path replannings on the way to $\mathbf{w}_3$ (d,e), removing the last waypoint $\mathbf{w}_3$ due to its dangerousness (f). Top views.

Fig. 10. Results of three autonomous flights.

While the helicopter is directed to fly a path along the houses and back with a speed of up to 4.5 m/s (c,d), additional objects and ground planes of other zones are added to the map until the helicopter reaches its final position (e). By comparing the resulting map with ground truth data derived from aerial images, it is shown in (f) that visible walls, trees, ground and the gap between the buildings are identified and represented in the world model. The tests also show that the quality of vision and inertial navigation data on a vibrating vehicle is good enough to perform robotic mapping approaches on a helicopter system.

B. Obstacle Avoidance Flights

Adaptive collision-free path planning around objects is tested with autonomous flights through user-defined waypoints. There are previously unknown objects between the waypoints, and the vehicle has to detect and to avoid them automatically. This paper presents results of three flights shown in figure 10. In these cases, the waypoints $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$ shall be passed in the given order, and the autonomous flight mode is switched on when the vehicle is close to the first waypoint. In all cases, the chosen setup requires to avoid the obstacle between $\mathbf{w}_2$ and $\mathbf{w}_3$. Velocity is set to 2 m/s.

Figure 10.1 shows a flight on a model aircraft field where an obstacle with a width of 6 m is placed. For safety reasons, the initial waypoints are set to 7 m above ground. The object is first mapped from 35 m distance, and a path beside the object is generated immediately. Then, the vehicle stops with an unfortunate deviation to the left, but it eventually flies around the object. The results presented in 10.2 refer to the same test setup. Here, dangers appear in the map when the vehicle is 25 m away from the object, and a generated path above the object leads to an avoided collision. The difference to the first flight is due to a slightly different representation of the obstacle in the map that yields another optimal path generated by the planner. In both cases, the last waypoint $\mathbf{w}_3$ is reached.

The graphics in figure 10.3 illustrate an obstacle avoidance test in the urban environment where the preliminary mapping test took place. Here, the flight altitude is set to 10 m above ground. It is limited so that the vehicle must not fly over the buildings. Walls can be detected and mapped while flying towards $\mathbf{w}_2$, and the path to $\mathbf{w}_3$ is changed multiple times. As visible in the plots, the map includes one false-positive obstacle representation. It becomes also visible that the planned path, which is limited to the road map, is not optimal due to its detours. The last waypoint is removed since it comes very close to the building, which yields a stop command at the second waypoint.

As a conclusion, all dangers can be avoided and the helicopter stops automatically at one of the user-specified waypoints. The tests have shown that mapping is real-time capable at a resolution of 0.5 m. Objects in the map sent to the path planner immediately result in a new path if the old path has a high risk of collision. Instructions to avoid collisions are given to the flight controller. The tests show also that both flight controller and the vehicle can handle these path changes during the flight. The final results are successful autonomous helicopter flights without collision.

VII. Conclusion and Future Work

This work presents a method to navigate a flying vehicle through unknown 3D environments and demonstrates its functionality in flight. The presented method consists of a world modeling step to gather environmental information and a global path planning step, based on the continuously updated world model. All algorithms are optimized to run on standard computer hardware onboard a small aerial vehicle and provide key enabling techniques for its autonomous navigation in complex environments.

To demonstrate the functionality of this approach to realistic obstacle avoidance applications, the algorithms are tested in flight. The combined mapping and path planning algorithms are executed under real conditions on board an autopiloted unmanned helicopter where a ground control station is only necessary for supervision. Results are autonomous flights with a successful avoidance of a single obstacle where the desired target behind the obstacle is reached, and the detection and avoidance of buildings in an urban area, where the vehicle was instructed to stop in front of the building.

In future applications, mapping will especially benefit from multiple cameras or laser scanners providing a larger field of view and measuring distance. Without the tunnel vision of the presented system, large improvements of possible reactions to detected objects are expected. This could provide a generation of foresighted alternative paths for example when large and wide objects appear in the map or when the vehicle flies through canyons with dangers on both sides.

Upcoming research will also focus on path planner improvements so that the vehicle will be able to fly smooth paths without stopping at each node and waypoint. To extend the flight performance, research is underway to relax the holonomic path planning constraint. This requires the generation of spline paths through a given environment and improved flight guidance and trajectory tracking algorithms to keep the vehicle on such splines. Moreover, an online task scheduling is desirable to avoid situations where the overall mission cost (e.g. total duration time) is affected due to path replanning. For example, if the initial plan was to fly from task A to C over B, it is possible that due to new obstacles it could be more suitable to fly to C first and then to task B.

References

[1] W. E. Green, P. Y. Oh, and G. Barrows, "Flying insect inspired vision for autonomous aerial robot maneuvers in near-earth environments," in *IEEE ICRA*, 2004, pp. 2347–2352.

[2] A. Beyeler, J.-C. Zufferey, and D. Floreano, "Vision-based control of near-obstacle flight," *Auton Robots*, vol. 27, no. 3, pp. 201–219, 2009.

[3] S. Hrabar et al., "Combined optic-flow and stereo-based navigation of urban canyons for a UAV," in *IEEE/RSJ IROS*, 2005, pp. 302–309.

[4] S. Griffiths et al, "Obstacle and terrain avoidance for miniature aerial vehicles," *Robotics and Autom. Mag.*, vol. 13, no. 3, pp. 34–43, 2006.

[5] D. H. Shim et al, "Conflict-free navigation in unknown urban environments," *Robotics and Autom. Mag.*, vol. 13, no. 3, pp. 27–33, 2006.

[6] Y. Watanabe, A. J. Calisey, and E. N. Johnson, "Vision-based obstacle avoidance for UAVs," in *AIAA Guidance, Nav. & Control Conf.*, 2007.

[7] P. O. Pettersson et al., "Probabilistic roadmap based path planning for an autonomous unmanned aerial vehicle," in *AAAI ICAPS*, 2004.

[8] S. Hrabar, "3d path planning and stereo-based obstacle avoidance for rotorcraft UAVs," in *IEEE/RSJ IROS*, 2008, pp. 807–814.

[9] F. Adolf, F. Andert, and J. F. Rocha, "Rapid online path planning onboard a VTOL UAV," in *AIAA Infotech@Aerospace Conf.*, 2010.

[10] S. Scherer et al., "Flying fast and low among obstacles: Methodology and experiments," *Int. J. of Rob. Res.*, vol. 27, no. 5, pp. 549–574, 2008.

[11] F. Adolf et al., "An unmanned helicopter for autonomous flights in urban terrain," in *Advances in Robotics Research*, T. Kröger and F. M. Wahl, Eds. Berlin: Springer, 2009, pp. 275–285.

[12] S. Thrun, "Robotic mapping: A survey," in *Exploring AI in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufm., 2002.

[13] F. Andert and L. Goormann, "A fast and small 3-d obstacle model for autonomous applications," in *IEEE/RSJ IROS*, 2008, pp. 2883–2889.

[14] M. Ren, J. Yang, and H. Sun, "Tracing boundary contours in a binary image," *Image and Vision Computing*, vol. 20, pp. 125–131, 2002.

[15] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Computer Graphics and Image Processing*, vol. 1, no. 3, pp. 244–256, 1972.

[16] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *Int. J. of Robotics Research*, vol. 23, no. 7-8, pp. 673–692, 2004.

[17] M. S. Branicky, S. M. LaValle, K. Olson, and L. Yang, "Quasi-randomized path planning," in *IEEE ICRA*, 2001, pp. 1481–1487.

[18] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[19] M. Likhachev et al., "Anytime dynamic A*: An anytime, replanning algorithm," in *AAAI Int. Conf. on Autom. Planning and Sched.*, 2005.

[20] F. Andert and F. Adolf, "Online world modeling and path planning for an unmanned helicopter," *Auton Robots*, vol. 27, no. 3, pp. 147–164, 2009.