Fast Autonomous Flight in Warehouses for Inventory Applications

Marius Beul, David Droeschel, Matthias Nieuwenhuisen, Jan Quenzel, Sebastian Houben, and Sven Behnke

Abstract—The past years have shown a remarkable growth in use-cases for micro aerial vehicles (MAVs). Conceivable indoor applications require highly robust environment perception, fast reaction to changing situations, and stable navigation, but reliable sources of absolute positioning like GNSS or compass measurements are unavailable during indoor flights.

We present a high-performance autonomous inventory MAV for operation inside warehouses. The MAV navigates along warehouse aisles and detects the placed stock in the shelves alongside its path with a multimodal sensor setup containing an RFID reader and two high-resolution cameras. We describe in detail the SLAM pipeline based on a 3D lidar, the setup for stock recognition, the mission planning and trajectory generation, as well as a low-level routine for avoidance of dynamical or previously unobserved obstacles. Experiments were performed in an operative warehouse of a logistics provider, in which an external warehouse management system provided the MAV with high-level inspection missions that are executed fully autonomously.

Index Terms—Aerial Systems: Applications; Aerial Systems: Perception and Autonomy; Motion and Path Planning;

I. INTRODUCTION

IN the last years, many novel applications for flying robots emerged, enabled by two main factors: i) manufacturers developed affordable and capable micro aerial vehicles (MAVs) for hobby, recreation and professional usage that do not require extensive flight training; ii) recent advances in robotic research led to efficient methods for environment perception and safe navigation, enabling various applications that can only be performed autonomously. This includes operations at high velocities and close to structures. Both conditions are prohibitive for safe operation by a human pilot. One driver for developing such systems is also the DARPA-formulated goal of flying fast and autonomously in cluttered environments without GPS and external sensing or control in their Fast Lightweight Autonomy Program (FLA) [1].

While in most current applications, MAVs maintain a safe distance from the object to inspect or follow; many future applications require the MAV to operate close to obstacles or even in restricted indoor spaces. As an example, in this paper, we consider the use case of automatic inventory in a warehouse. It requires the MAV to quickly detect, identify,
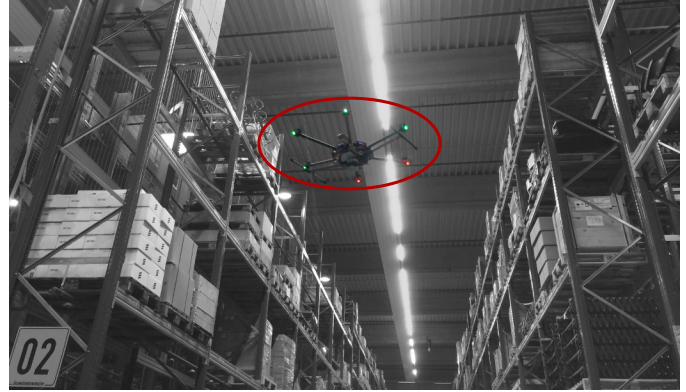
Fig. 1. Our inventory system performs a fully autonomous inspection of a warehouse. The main challenges are the fast navigation in narrow passages close to structures and the localization in a large self-similar indoor environment with distant walls.

and map the stored items. In this way, it is possible to keep an always-up-to-date inventory record of the contents within the warehouse. Current commercial systems [2], [3] for this task merely deploy a scanner on the platform and perform a piloted flight in order to read tags on the goods.

Autonomous maneuvering inside such a building is highly challenging as most of the space is occupied with high shelves filled with stocked goods as shown in Fig. 1. This leaves only small aisles for navigation which might also be obstructed by other objects like forklifts. Additionally, the shelf rows lack distinctive geometric features and are highly self-similar which makes precise self-localization difficult. On the other hand, these narrow structures are embedded in large halls with stable, but far-away localization aids like walls. This requires real-time localization with long-distance sensors in large maps with many structures.

We present our self-localization and mapping approach based on a 3D lidar, which is able to handle these challenging situations robustly. The lidar is also the basis of a low-level obstacle avoidance mechanism. In addition, the robot carries a sensor setup to identify the stocked material by means of fiducial markers and RFID tags. The flight mission is provided by a warehouse management system (WMS) as a sequence of storage panels that have to be inspected. The mission is planned in a semantic, yet metric, map of the warehouse that contains the approximate placing of all the shelf rows and the number and relative position of the storage panels within. The laser-based map is aligned with this representation in order to define the inspection poses that the robot consecutively visits during its flight.

Experiments are performed in a warehouse of a logistics provider containing narrow aisles between shelves and larger open areas. We mapped several shelf rows and performed
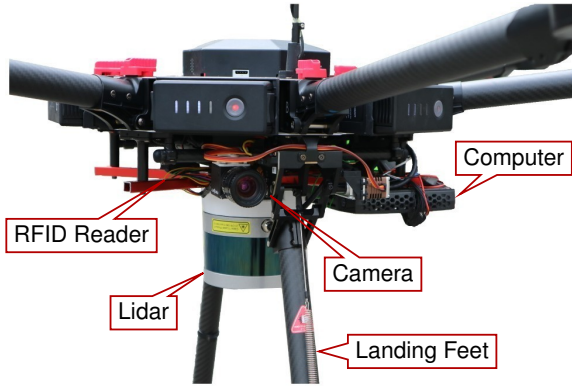
Fig. 2. Design of our MAV equipped with a Velodyne Puck LITE, fast onboard computer, two synchronized global shutter color cameras, and an RFID reader. The landing feet are retractable to allow for true 360° perception.

autonomous inventory missions including the transition between rows and the avoidance of static obstacles. Furthermore, we demonstrated the reactive avoidance of dynamic obstacles approaching the MAV. We discuss guidelines for the development of future systems for autonomous indoor operation and draw prospects for the future of autonomous inventory robots. To demonstrate the robustness of our localization and control at high velocities, not reachable in our indoor environments due to the required acceleration distance, we further evaluate our system outdoors with flights reaching velocities over $28\,\mathrm{km/h}$ without GNSS feedback.

In our integrated system, we employ and extend methods based on our own previous work: The SLAM system is detailed in [4] and [5]. Our obstacle avoidance extends [6] and the mechanics of our model predictive controller (MPC) are described in [7].

Our main contributions are

- robust self-localization solely based on an onboard lidar at high velocities up to $7.8\,\mathrm{m/s}$ (Sec. IV),
- fast fully autonomous navigation and control, including avoidance of static and dynamic obstacles in indoor and outdoor environments (Sec. V),
- an integrated autonomous robot system for aerial stocktaking with multimodal tag detection, evaluated in an operative warehouse (Sec. VI).

II. RELATED WORK

Today, fast MAV flight without external sensing is mostly vision-based. Recently, Falanga et al. [8] presented an MAV flying with $3\,\mathrm{m/s}$ through narrow gaps. This requires precise relative localization and navigation. We focus on fast navigation in allocentric maps with reliable obstacle avoidance, which currently is not achievable by using cameras alone.

Shen et al. [9] present an MAV that is capable of autonomous vision-based flight with up to $4\,\mathrm{m/s}$ on a straight line, $2\,\mathrm{m/s}$ on a figure eight, and $1.5\,\mathrm{m/s}$ in an indoor environment. Although the system is relatively fast, the authors report significant drift, induced by solely relying on cameras for state estimation.

Another vision-based, lightweight MAV system has been presented by Burri et al. [10]. Their work focuses on industrial boiler inspection with agile flight in industrial environments.
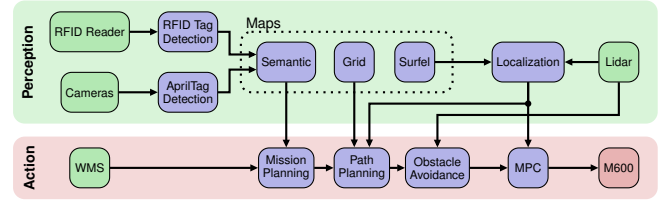

Fig. 3. System overview. Inputs are depicted in green and software components in blue. An external warehouse management system (WMS) provides an unordered list of waypoints of to be inspected goods to the mission planning. Control commands are sent to the SDK of the DJI Matrice 600 (red).

Florence et al. [11] use a combination of vision and a 2D laser scanner to avoid obstacles at high velocities. Their system flies in cluttered unknown environments with large state uncertainties. For our application, we rely on precise, but still fast, allocentric localization and assume that an allocentric map contains the major, complex obstacles.

Our targeted scenario is particularly adverse for the use of visual perception. On one hand, panels and shelf rows carry highly similar and repetitive visual cues, which preclude any form of place recognition. On the other hand, local geometry is also highly self-similar and symmetric. Hence, we solely rely on high-frequency 3D laser scans for obstacle perception and state estimation. Its large field-of-view and long measurement range allow for resolving local similarities by using large-scale structures for localization.

Ma et al. [12] addressed automatic inventory with a lightweight Parrot drone. Their RFly system relays the RFID signal to a reader and is able to triangulate the location of the tag with a reported accuracy of below $20\,\mathrm{cm}$. However, in order to self-localize the robot, they rely on an external motion capturing setup, which limits the practical feasibility.

Similar to our system, Ortiz et al. [13] perform inspection in narrow spaces. They developed a quadrotor MAV for autonomous vessel inspection. A combination of laser localization and visual odometry yields a 2D localization approach decoupled from the height measurements. Our system performs SLAM with 6D pose estimation based on a high-performance 3D lidar.

An early version of our inventory MAV [14] relied on a combination of two rotating 2D lidars for a low frequency localization and mapping with visual odometry performed with three pairs of wide-angle stereo cameras. Although the system proved itself robust, the setup proposed in this paper only relies on a single 3D lidar as primary sensor and, hence, significantly reduces the overall system complexity. Furthermore, the increased frequency of 360° scans obviates the requirement for additional visual odometry. To account for the narrower vertical field of view in the proposed system, our path planning optionally limits the ascension and descension angle.

III. SYSTEM SETUP

Our MAV, shown in Fig. 2, is based on the DJI Matrice 600 platform with a diameter of approximately $170\,\mathrm{cm}$. It is equipped with a lightweight, yet powerful, Intel NUC6i7KYK onboard PC with an Intel Core i7-6770HQ quadcore CPU running at $2.6/3.5\,\mathrm{GHz}$ and $32\,\mathrm{GB}$ of RAM. As primary
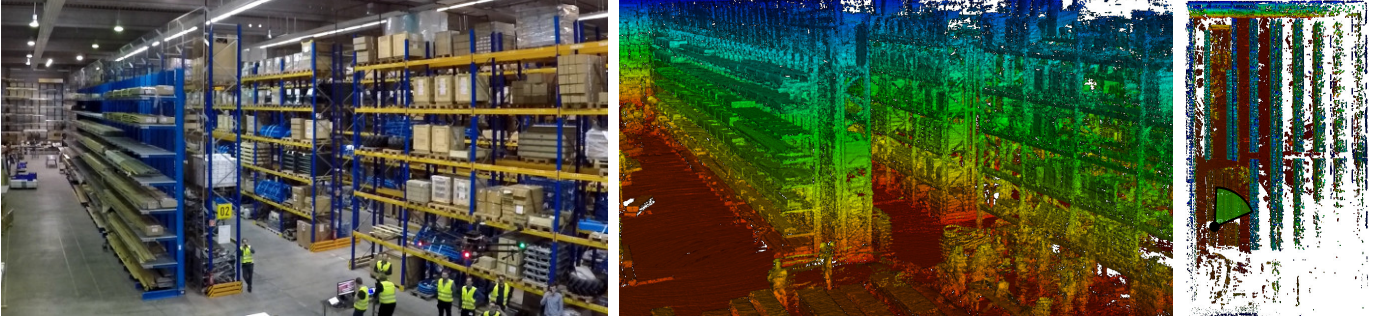
Fig. 4. 3D map from the initial manual flight. The top-down view (right) shows the dimensions of the acquired map of the $100 \times 60\,\text{m}$ warehouse. The camera perspective is highlighted in green. The warehouse contains tight, self-repetitive, and cluttered structures like shelves and stock, and larger, far-away structures like walls. For robust localization, the MAV has to employ a map of the structure of the large building.

environment perception sensor, a Velodyne Puck LITE lidar is deployed. It features a low weight of $590\,\text{g}$ and yields 300,000 range measurements per second in 16 horizontal scan lines at a vertical angle of $30°$. Its maximum range is $100\,\text{m}$.

In order to perceive visual tags in nearby shelf panels during flight, the MAV is equipped with two synchronized global shutter Point Grey Blackfly-S U3-51S5C-C color cameras with $5.0\,\text{MP}$. The Computar M0814MP2 lens features an apex angle of $56.3 \times 43.7°$. Each camera captures 3 frames per second. For detection of RFID tags, the MAV is also equipped with a ThingMagic M6e RFID reader with a SkyeTek SP-AN-04-UF-BB6LP antenna.

The low weight of the components ($11.2\,\text{kg}$ take-off weight) and a battery capacity of $600\,\text{Wh}$, yields a flight time of approximately $20\,\text{min}$ which allows to capture $1\,\text{km}$ of shelf. Since the batteries are hot-swappable, continuous operation can be performed with only minimal interruptions.

The system uses the robot operating system (ROS) as middleware on both the MAV and an additional ground control station. We show an overview of the system in Fig. 3.

IV. ENVIRONMENT PERCEPTION

A. 3D Mapping

To localize the MAV within the environment, we build an allocentric map of the warehouse from measurements of the lidar. Fig. 4 shows parts of the warehouse and the initial map. We incorporate measurements of the IMU to account for motion of the sensor during acquisition. Using an extended version of our lidar-based SLAM method described in [4], we first aggregate 3D scans in a local multiresolution grid map. Local multiresolution maps correspond to the sensor measurement characteristics by having a high resolution close to the sensor and a coarser resolution farther away. For each grid cell, a local surface element (surfel) is estimated which summarizes the aggregated measurements in the cell's volume and captures the statistics of the points. We recover the transformation between a newly acquired scan and the local map by matching surfels [5]. Compared to point-based registration, considerably less elements are taken into account for registration, allowing for efficient registration of the extensive amount of measurements from the sensor.

Registered 3D scans are added to the local map, replacing older measurements. Local mapping allows to track the robot

in a local frame and provides a dense aggregation of measurements in the robot's vicinity.

We construct an allocentric pose graph by aligning local multiresolution maps from different view poses which allows the robot to localize itself in an allocentric frame. Hence, local multiresolution maps from different view poses model nodes in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that are connected by edges. Edges model spatial constraints between nodes and result from aligning two local multiresolution maps by surfel-based registration. The registration result x_i^j between a new node v_i and the previous node v_j constitutes an edge $e_{ij} \in \mathcal{E}$.

Additionally, the current local map is registered towards a reference node in order to connect the current pose to the global pose graph and enable a straightforward optimization. The reference node is the local map that is closest to the current MAV pose. If the robot moved sufficiently far, we extend the pose graph by the current local map.

Furthermore, we include edges between the newly added local map and close-by local maps to obtain loop closure if the robot revisits previously mapped areas. Hence, we check for one new edge between the current reference v_ref and other nodes v_cmp. We determine a probability

$$p_\text{chk}(v_\text{cmp}) = \mathcal{N}\left(d(x_\text{ref}, x_\text{cmp}); 0, \sigma_d^2\right)$$

that depends on the linear distance $d(x_\text{ref}, x_\text{cmp})$ between the view poses x_ref and x_cmp. According to $p_\text{chk}(v)$, we draw a node v from the graph and determine a spatial constraint between the nodes using our surfel registration method.

From the spatial constraints, we infer the probability of the trajectory estimate given all relative pose observations

$$p(\mathcal{V} \mid \mathcal{E}) \propto \prod_{e_{ij} \in \mathcal{E}} p(x_i^j \mid x_i, x_j).$$

Each spatial constraint is a normally distributed estimate with mean and covariance determined by our probabilistic registration method. This pose graph optimization is efficiently solved using g^{2}o [15], yielding maximum likelihood estimates of the view poses x_i.

We extend our mapping approach presented in [4] to allow for efficient processing of Velodyne scans. In contrast to the approach presented in our previous work, we do not aggregate multiple 3D scans using odometry information but register single 3D scans from the lidar sensor to the local multiresolution map—only using orientation information from the IMU and barometric height as prior for registration.
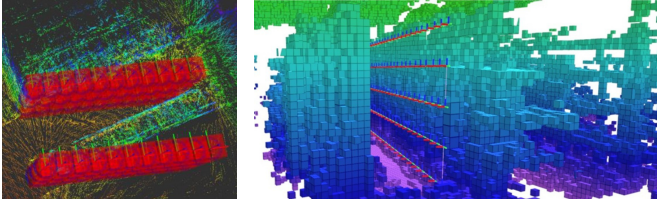
Fig. 5. Left: Registration of a semantic map (coordinates of storage units, geometric shelf model depicted in red) with a 3D laser map. Color encodes height. Right: Generated inventory mission (depicted by the coordinate axes) in the obstacle grid map.

B. Lidar-based Localization

Prior to autonomous operation, we acquire an initial map from a manual flight. We extend our SLAM system to serialize the graph-based structure of the allocentric map to gain persistent storage of the so-far acquired pose graph.

For autonomous operation during mission, the mapping system is initialized with the pose graph from the initial flight. By aligning the current local map to the pose graph, we gain a localization pose with respect to the initial map and the warehouse model. Although the pose graph (and the associated map) can be extended if the MAV traverses parts of the environment that where not covered by the initial flight, we choose the coverage volume of the initial flight to be larger than the MAV's workspace in the experiments since this is the envisaged operating mode during standard inventory missions.

While executing the mission, we localize towards the closest local map in the graph by registering the current local map with it. Our approach allows to process the lidar scans in real-time.

C. Tag Detection

We perceive the position of stock in the warehouse by means of visual fiducial markers (AprilTags) and RFID tags attached to storage boxes. Perceived RFID tags, the current MAV position, signal strength, and direction of the detecting antenna are transmitted to the WMS for further processing, e.g., assigning stock to storage units. Regarding the fiducial markers, we use the implementation by Olson et al. [16] and transform the camera-based relative tag pose into an allocentric frame via the known camera extrinsics and the estimated pose of the MAV. Likewise, these allocentric positions and the corresponding tag IDs are sent to the WMS for incorporation into the warehouse model. We use the tag family `36h11` as we experienced it to be very reliable.

V. NAVIGATION AND CONTROL

Autonomous navigation is a key capability for automated stocktaking. Operator assistance functions—or optionally fully autonomous operation—opens up the applicability of the system to a large group of end users who are not trained MAV pilots. Autonomy generates a direct interface between logistics personnel and the stocktaking system without the indirection of a professional pilot. We implement a hierarchical navigation and control system that makes use of time scale separation between the layers. On the top layer of our navigation stack, global mission planning is executed once per mission. The
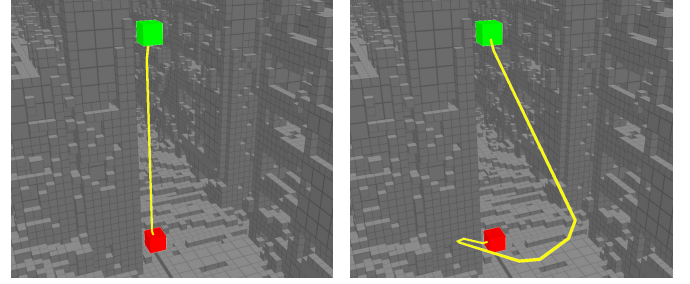

Fig. 6. Planning under visibility constraints. Left: Without visibility constraints the shortest path (yellow) from a start (green) to a target position (red) below solely descents in place. Right: With visibility constraints, the MAV has to move within the field of view of the lidar and consequently follows a longer descent path with an angle of 15°.

next layer (allocentric path planning) is run in the order of seconds, while the lowest layer (model predictive trajectory planning) is executed every $20\,\mathrm{ms}$.

A. Mission Planning

For the connection to a WMS, we developed a tool that augments the laser-based maps described in Sec. IV with semantic information. Fig. 5 shows the registration of the semantic warehouse model with the laser-based map. After a coarse manual alignment, we use the Iterative Closest Point Algorithm (ICP) to automatically register both maps. This enables us to semantically describe an inventory mission and automatically derive shelf numbers and indices of storage places. The WMS can specify missions covering whole shelves—with a coverage pattern shown in Fig. 5—or single storage units to inspect. Here, all common strategies for manual inventory like e.g., sampling inventory with sequential probability ratio test (SPRT) can be utilized. An ordered list of view poses is then sent to the MAV onboard computer for execution. Before execution, this list is simplified to merge collinear path segments, e.g., a number of storage units on the same height, to achieve a smooth sweeping motion along the shelves.

B. Path Planning

The result of the mission planning is an ordered list of 4D-poses (x,y,z,θ) in a discrete allocentric grid. We connect these poses with an instance of A* planning and use the Ramer-Douglas-Peucker algorithm to cull superfluous nodes. This is necessary to allow for the generation of longer and more continuous trajectories by our controller, described below. During mission execution, the path is frequently replanned to compensate for path deviations of the MAV, either by inaccurate command execution, external disturbances or avoidance of obstacles. Replanning takes place whenever a target pose is reached and the next pose from the mission plan is processed or at least every $10\,\mathrm{s}$ to correct deviations from the path. Grid-based planning resembles the orthogonal structure of warehouses if the aisles are parallel to the planning grid axes. The planning grid and the model are, therefore, aligned after the exploration flight. Our approach, in contrast to sampling-based planners, has the advantage to follow the shelf-fronts well, without much postprocessing and trajectory smoothing.
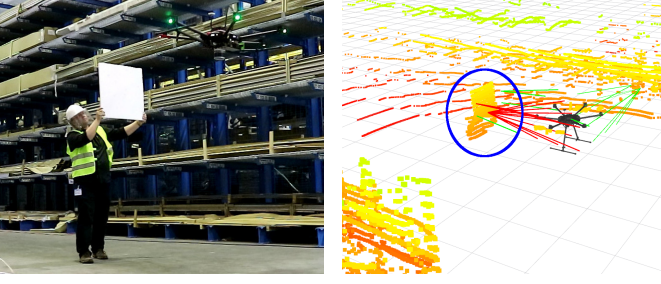
Fig. 7. Reactive obstacle avoidance with artificial potential fields. A person (circled blue in the laser map) approaches the MAV. The MAV is repelled by the artificial forces (red lines) and dodges the obstacle. Green lines depict the influence of obstacles in the passive avoidance distance.

In more generalized settings, the approach could benefit from any-angle planning, e.g., Theta* [17], which we can omit here.

The onboard lidar does not cover a spherical field of view. To nevertheless allow for safe navigation in cluttered environments or in the presence of dynamic obstacles, we extended our planner with visibility-constrained planning. With this extension, the planned MAV movements are restricted to directions in the field of view of the lidar, i.e., $15°$ below and above the current horizontal plane. To this end, we employ a grid with anisotropic voxels to reduce the ascent and descent angles from $45°$ in a grid with isotropic voxels to the opening angle of the sensor. The resulting voxels have a height of $\tan(15°) \approx \frac{1}{4}$ of the horizontal voxel size. Furthermore, we remove edges connecting cells directly on top of each other, disallowing ascents and descents in place. The direction of flight—discretized to the eight possible transitions in the plane—is introduced as a new planning dimension to penalize changes in the flight direction. Angles of up to $45°$ are not penalized. Without this penalty, a zigzag motion to ascent or descent would be equal to larger straight glide paths in path costs, but would significantly slow down the MAV due to numerous stops to change direction. Fig. 6 illustrates the resulting plans with and without visibility constraints.

C. Reactive Obstacle Avoidance

We use reactive obstacle avoidance as a low-level safety layer complementing the deliberative path planning. For our application, reactive obstacle avoidance has two important properties—compared to fast local planning [18], or optimization-based approaches [19]. First, it has the ability to elude approaching dynamic obstacles, depicted in Fig. 7. This might include leaving a hover position or even moving into the opposite direction of the commanded flight path. Second, a hazard minimizing solution will always be found even if the distance constraints are violated. Furthermore, reactive obstacle avoidance is computationally cheap and, consequently, can be executed with the lidar frequency of $10\,\text{Hz}$. Our obstacle avoidance is based on [6] but directly modifies the allocentric target waypoints from the global path planner instead of velocity commands to adapt to the new low-level trajectory controller.

We modified the basic algorithm to facilitate smoother flight in narrow spaces by adding two spheres of influence around the MAV, depicted in Fig. 8. Obstacles in the passive avoidance
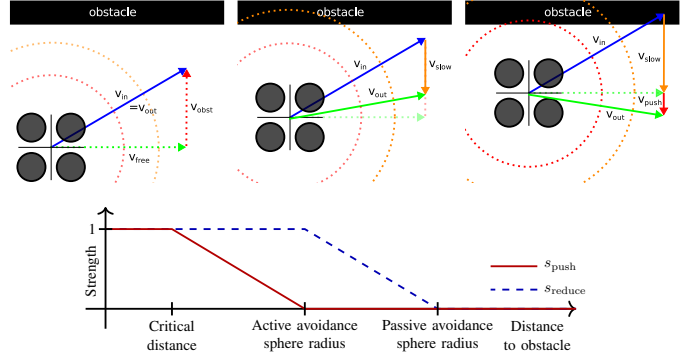

Fig. 8. Reactive obstacle avoidance. Top-Left: The MAV velocity setpoint vector v_{in} is split into the projection towards an obstacle v_{obst} and the remainder v_{free}. If the MAV is not close to obstacles, the output velocity v_{out} is equal to the setpoint. Top-Middle: When an obstacle is in the passive avoidance sphere (dotted orange), v_{in} is reduced by $v_{slow} = -s_{reduce} v_{obst}$. Top-Right: Obstacles in the active avoidance sphere (dotted red) induce an additional repulsive force resulting in the pushing velocity v_{push} directing the MAV into free-space. For simplicity, we depict velocity vectors, the pose modification vectors c_o and f_o follow straightforward. Bottom: Scaling factors in relation to the obstacle distance.

sphere with radius d_p, cause a reduction of the MAV motion into the direction of the obstacles. In the active avoidance sphere with radius d_a, obstacles exert artificial repulsive forces, increasing with proximity, that push the MAV away. By dividing the obstacle avoidance into these two phases, we achieve a stable equilibrium distance between obstacles and MAV regardless of the MAV control inputs without influencing the motion into orthogonal directions in the passive sphere— e.g., the MAV can follow an exploration pattern along a shelf even if the commanded pattern is too close to the shelf due to protruding goods. In the warehouse, we set d_a and d_p to MAV radius plus $1\,\text{m}$ and $2\,\text{m}$, respectively.

For simplicity of notation, all further calculations are depicted in an egocentric MAV frame to omit the localization transform matrices. If both spheres are obstacle-free, we execute the commands from the planning layer unaltered. Egocentric targets farther away than $1\,\text{m}$ are first normalized; shorter vectors are processed without prior normalization to avoid a speed up of the MAV while approaching an obstacle. The new egocentric target position t_{new} is calculated as

$$t_{\text{new}} = t_{\text{orig}} - c_o s_{\text{push}} + f_o s_{\text{reduce}}.$$

Here, c_o is the projection of the current target t_{orig} onto the direction of the obstacle, thus, the part of the command that steers the MAV closer to the obstacle. The artificial force direction f_o is a normalized vector pointing away from the obstacle. The magnitudes of the slow down strength s_{push} and the push back strength s_{reduce}—depicted in Fig. 8—are calculated as

$$s_{\text{push}} = \frac{d_p + d_a - d}{d_p - d_a}, \quad s_{\text{reduce}} = \frac{d_a - d}{d_a}$$

with distance d to the obstacle. Both results are clipped to the interval $[0, 1]$ afterwards.

D. Model Predictive Control

Since higher layers assume a straight connection between waypoints (due to Ramer-Douglas-Peucker culling), flying on
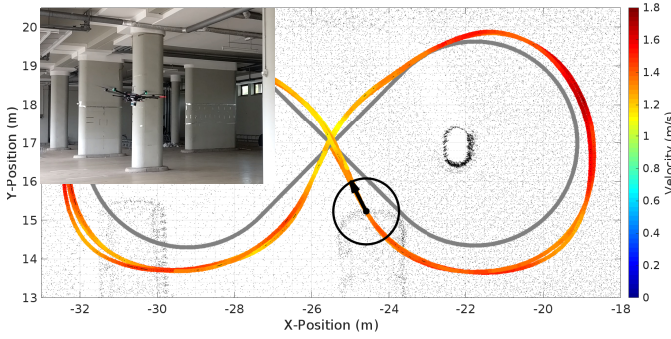
Fig. 9. The MAV continuously flies in a figure eight around pillars in a parking garage. All perception and computation is done onboard. Velocities exceeding $1.7\,\mathrm{m/s}$ in the vicinity of obstacles require robust methods for state estimation and control. The size of the ring represents the actual MAV size. The arrow depicts the flight direction.
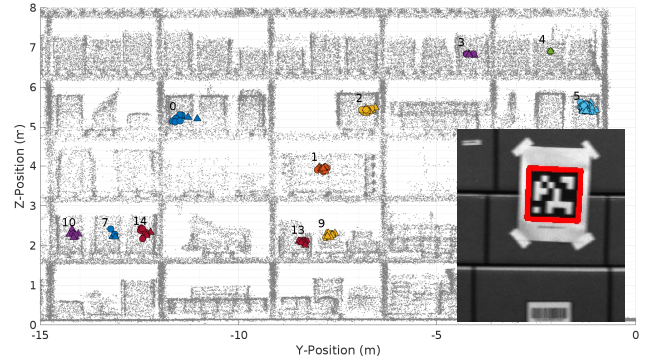

Fig. 10. AprilTag detection. With two cameras directed to each side of the aisle we detect AprilTags attached to the stock. The clusters of colored markers show the estimated positions of the detected tags in a subsection of an aisle during two consecutive flights. See Fig. 11 for the corresponding trajectories. Detections from the first flight are marked with circles; detections from the second flight are marked with triangles. Different colors correspond to different tag IDs. In the bottom right corner, a detection of ID 14 is shown. One can see the motion-blur induced by the relative motion between AprilTag and MAV. We report statistics in Tab. I.

a straight trajectory is mandatory and overshoot is not permissible despite large turbulences caused by nearby obstacles. Also inventory of large warehouses with multiple kilometers of shelf requires fast flight to reduce the impact on the regular logistic processes. We tackle this problem by employing time-optimal trajectory generation and online replanning with $50\,\mathrm{Hz}$ for low-level control. We use an extended version of the method described in [7]. Planning is based on a simple dynamic model of the MAV with three-dimensional jerk j as only input. The method plans smooth, time-optimal trajectories from the current 9-dimensional allocentric MAV state

$$\boldsymbol{x} = \begin{pmatrix} p_x & p_y & p_z \\ v_x & v_y & v_z \\ a_x & a_y & a_z \end{pmatrix}$$

to the corresponding 9-dimensional target state by analytically solving a system of 21 differential equations

$$p_n = p_{n-1} + \int_{t_{n-1}}^{t_n} v_n\,\mathrm{d}t,$$

$$v_n = v_{n-1} + \int_{t_{n-1}}^{t_n} a_n\,\mathrm{d}t, \qquad n = \{1; \dots; 7\}$$

$$a_n = a_{n-1} + \int_{t_{n-1}}^{t_n} j_n\,\mathrm{d}t$$

per axis (x,y,z). Generated trajectories consist of up to $n = 7$ phases of constant jerk input, resulting in a bang-singular-bang trajectory. Individual axes are coupled by synchronizing the total time of the entire trajectory. The trajectories respect per-axis constraints on minimum and maximum velocity, acceleration and jerk.

With the ability to predict the target state, trajectories end in an optimal interception point when the waypoint is non-stationary like shown in Fig. 9. Since our method is very fast, we use it in closed loop and send smooth pitch θ, roll ϕ and climb rates v_z to the DJI flight control.

We assume the yaw to be decoupled from the translatory axes and use simple proportional control for the yaw. The yaw rate setpoint $\dot{\Psi}_{setp} = K_p \cdot (\Psi_{setp} - \Psi)$ with proportional gain K_p is sent to the MAV.

In comparison to approaches that utilize a complex motion model like Kamel et al. [20], our approach is very fast and the

model does not need (often abstract) parameters. In contrast to complex models, approaches like Mueller et al. [21] use a simple motion model and are comparably fast. The generated trajectories however are not time-optimal. Simple PID-control is also not suitable, since overshoot is not permissible in the close corridors. Thus, the controller would have to be parameterized very conservative which would result in slow MAV movement.

Please note that we use the same parameters for the controller as in [22] in which we employed the method on a DJI Matrice 100 that weighs only a quarter of the MAV used here with a corresponding bounding box volume ratio of 1:12. This shows the independence of our approach regarding model parameters.

VI. EVALUATION

We evaluate our system in indoor and outdoor scenarios, including an inventory mission in an active warehouse. A video showing autonomous mission execution and reactive obstacle avoidance can be found on our website[1]. Here, we also publish recorded datasets, tools, and parts of our pipeline.

First, we test the robustness of the localization and control pipeline with an experiment that involves fast flight between alternating waypoints in an obstacle free courtyard over a distance of $25\,\mathrm{m}$. The localization in an allocentric map of the courtyard and state estimation of the MAV was solely based on the onboard 3D lidar and the IMU; no GNSS feedback was used. Between the acceleration and deceleration phases of the flight, the MAV reached a maximum velocity of $7.8\,\mathrm{m/s}$, measured by the onboard DJI GPS – considered as ground truth. The laser localization was running at $20\,\mathrm{Hz}$ to account for the large velocities. It was able to robustly track the MAV pose during the whole flight. Despite strong wind, the maximum deviation from the straight line connection between both waypoints was only $49\,\mathrm{cm}$ during all 11 alternations.

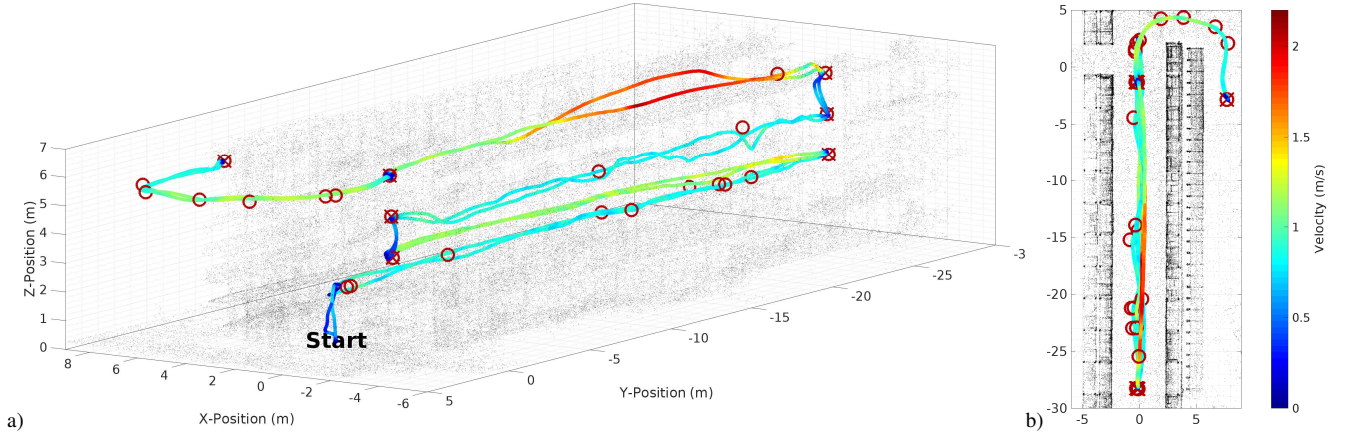[1]http://www.ais.uni-bonn.de/videos/IROS_2018_InventAIRy

Fig. 11. Visualization of two consecutive flights in a warehouse. a) Side view, b) Top view. Despite flying in close proximity to obstacles, the MAV reaches velocities up to $2.1\,\mathrm{m/s}$. Waypoints are precisely reached without overshoot. It can be seen that the flight behavior is repeatable and that aisle changes are possible. View poses are marked with a red crossed ring. Via points that are inserted by the A* planner are marked with a red ring. Except for the manual start, the whole flight was fully autonomous.

The maximum overshoot recorded during the experiment was $1.2\,\mathrm{m}$.

In a second experiment, the MAV flew a figure eight around two pillars in a garage to additionally test the influence of turbulences close to structures and the ground. Due to the high accelerations of approximately $0.85\,\mathrm{m/s^2}$ in the curved segments of the trajectory, the maximum velocity in these runs was reduced to yield a feasible, collision-free trajectory. Still, the MAV reached velocities up to $1.75\,\mathrm{m/s}$ in this indoor environment. The laser localization tracked the MAV pose with $10\,\mathrm{Hz}$ and was able to keep the MAV localized in the map of the garage at all times. Fig. 9 shows the resulting trajectory in the map of the garage. It can be seen that our method yields robust repeatability in four consecutive flights despite turbulences. Nevertheless, it can be seen that the MAV spirals out of the curved segments as it cannot accurately track the moving waypoint.

In a third experiment, our integrated system, including laser-based localization, planned navigation, obstacle avoidance, and acquisition of information about stock positions, was demonstrated in a warehouse with a building area of $100 \times 60\,\mathrm{m}$ with $1.3\,\mathrm{km}$ shelf (approximately $12\,000\,\mathrm{m^2}$ storage front). As described in Sec. IV-A, we built an initial laser-based SLAM map of the environment with a manual flight, shown in Fig. 4. This map is aligned with the semantic map containing storage units from the WMS. For the demonstration of autonomous inventory, a mission containing the complete inventory of one shelf row and the inspection of a single storage unit in another row was specified in the WMS. The MAV executed this mission autonomously multiple times while avoiding static obstacles, e.g., the shelves and stock protuding from the shelves. In Fig. 11 we visualize the trajectory of two consecutive flights in the warehouse. The MAV reaches velocities up to $2.1\,\mathrm{m/s}$. Although faster flight is possible (as shown in the previous experiments), we used the ability of our MPC to limit the maximum velocity a) to account for the acceleration/deceleration distance needed by the MAV and b) to reduce motion blur in the cameras (see Fig. 10):

The closed loop dynamics of MAV and MPC dictate the dy-

namic behavior of the system. Even under the assumption that the MAV is able to perfectly track the trajectories generated by the MPC and without any perception- or communication delay, an acceleration/deceleration distance of $12.1\,\mathrm{m}$ is necessary with a maximum velocity of $7.8\,\mathrm{m/s}$ (with parameters $a_{max} = 3.5\,\mathrm{m/s^2}$, $j_{max} = 4.0\,\mathrm{m/s^3}$).

Furthermore, due to the artificial lighting in the warehouse, the camera exposure time had to be set to at least $4\,\mathrm{ms}$ for acceptable image quality. The used AprilTags have an edge length of $16\,\mathrm{cm}$ that results in a patch size of $2 \times 2\,\mathrm{cm}$. Thus, the Nyquist frequency limits the relative velocity to $10\,\mathrm{m/s}$ under ideal conditions. This velocity, however, would require special signal reconstruction techniques to preprocess the image for the AprilTag detector. Also roll, pitch, and yaw motion superimpose the linear MAV velocity and generate relative motion between tag and MAV. High-frequency vibrations generated by the propellers provoke additional blur. Therefore, we conservatively constrained the linear velocity in favor of robust detections in the warehouse experiment.

In contrast to the visual detection pipeline, the RFID reader did not limit the inventory speed since it is able to read up to 750 tags per second. We throttled the speed to 20 reads/second which was enough for our experiments and allowed for a higher detection range.

Every view pose is reached with a mean deviation of only $9.65\,\mathrm{cm}$ respectively $5.78\,\mathrm{cm}$ in both flights. As no dynamic obstacles above the MAV were to be expected in this demonstration, we neglected the planning with visibility constraints in favor of faster mission execution.

During both flights, AprilTags on the sides of the aisle and RFID tags of the specified shelf row were captured and sent to the WMS. Fig. 10 and Tab. I show the result of the two flights. It can be seen that except for Tag 6, and 11, all tags are reliably detected (Tags 8 and 12 were not used). Our method was unable to detect Tag 11 due to a shadow that partially covered the tag on a disadvantageously positioned stock. Tag 6 was not attached properly and was flipped by turbulent air from the MAV. Not a single false positive detection happened during the experiment. It can be seen that only minimal scattering

TABLE I
STATISTICS OF APRILTAG DETECTIONS FOR TWO FLIGHTS.

Tag ID	0	1	2	3	4	5	7	9	10	13	14
n_1	3	7	6	3	1	64	2	10	6	4	6
n_2	7	7	6	3	1	41	3	5	3	4	5
σ_1	10.4	3.2	4.7	2.7	-	4.7	-	3.3	4.2	3.3	3.9
σ_2	3.9	4.3	5.0	3.4	-	3.8	3.3	3.5	2.1	2.8	4.8
$|\mu_{1-2}|$	28.6	2.9	8.9	5.5	3.3	2.1	1.2	3.2	1.0	2.1	10.7

n_i is the number of detections per flight, σ_i the deviation of the detections in cm. $|\mu_{1-2}|$ is the distance of the means μ_i in cm.

occurs and thus the relative detection error is small.

After the executed inventory mission, the MAV hovered at a height of $2\,\mathrm{m}$ above the ground. A person approached the MAV, which avoided the dynamic obstacle by means of our reactive obstacle avoidance, shown in Fig. 7. Furthermore, a person stepped into the way of the MAV while it approached a waypoint. The MAV stopped at a safe distance in all cases.

As shown in the experiments, the limiting factor for faster inventory is motion blur in the cameras caused by the large exposure time due to bad lighting conditions. In future work, we want to oppose this bottleneck either by illuminating the scene ourselves (by using a flash on the MAV) or by using special equipment like e.g., event based cameras.

In the current setup, the MAV continuously records images with $3\,\mathrm{Hz}$. The AprilTags however only cover less than $3\,\%$ of the area ($0.0256\,\mathrm{m}^2$ tag size vs. $0.96\,\mathrm{m}^2$ storage unit front). A more targeted strategy would reduce the generated data. Furthermore, we also plan to extend our vision pipeline to not only detect AprilTags, but also other visual indicators like, e.g., barcodes, QR codes, and human readable text, commonly found on stock. This would further enhance the versatility of the system. We also plan to integrate multiple MAVs into the mission planner for simultaneous inventory to speed up the process even more.

One might also think of eliminating the first manual flight in favor of an automated SLAM process, but that a manual flight is more robust in this crucial map building phase. Furthermore, in comparison to the service life of such a system, the map building phase only causes a small fixed effort, since the map is reusable. After the manual flight, the operators can check the map for possible artifacts and misregistrations.

VII. CONCLUSION

In this paper, we presented an MAV that is capable of fast autonomous indoor and outdoor flight without the aid of external infrastructure, solely relying on an omnidirectional laser scanner for localization. We approached this challenge by employing fast 6D lidar based localization in 3D maps in combination with time-optimal model predictive control. Due to the fast runtime of our methods, the MAV motion can be tracked and controlled even under high velocities and accelerations. Our ROS-based mapping and navigation pipeline allows for fully autonomous flight even in GNSS-denied environments.

Ample onboard processing power in combination with a high bandwidth ground connection and long battery life leads to a system that is suitable to be deployed, e.g., in warehouses for extensive stocktaking applications. We demonstrated the system robustness in multiple experiments where the only manual interactions were the starting and landing phases.

REFERENCES

[1] DARPA, "Fast Lightweight Autonomy (FLA)," 2015, solicitation number DARPA-BAA-15-16.
[2] J. Pons. (2017) DroneScan. [Online]. Available: www.dronescan.co
[3] Hardis Group. (2017) EyeSee. [Online]. Available: eyesee-drone.com
[4] D. Droeschel, M. Schwarz, and S. Behnke, "Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner," *Robot. Auton. Syst.*, vol. 88, pp. 104 – 115, 2017.
[5] D. Droeschel, J. Stückler, and S. Behnke, "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 (ICRA)*, 2014.
[6] M. Nieuwenhuisen, M. Schadler, and S. Behnke, "Predictive potential field-based collision avoidance for multicopters," in *International Arch. Photogramm. Remote Sens. Spatial Inf. Sci. (ISPRS)*, vol. XL-1/W2, 2013, pp. 293–298.
[7] M. Beul and S. Behnke, "Fast full state trajectory generation for multirotors," in *Proc. of Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, 2017.
[8] D. Falanga, E. Mueggler, M. Faessler, and D. Scaramuzza, "Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017.
[9] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor," in *Proc. of Robotics: Science and Systems (RSS)*, 2013.
[10] M. Burri, J. Nikolic, C. Hürzeler, G. Caprari, and R. Siegwart, "Aerial service robots for visual inspection of thermal power plant boiler systems," in *Proc. of Int. Conf. on Applied Robotics for the Power Industry (CARPI)*, 2012.
[11] P. Florence, J. Carter, , and R. Tedrake, "Integrated perception and control at high speed: Evaluating collision avoidance maneuvers without maps," in *Proc. of Int. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2016.
[12] Y. Ma, N. Selby, and F. Adib, "Drone relays for battery-free networks," in *Proc. of Annual Conf. of the ACM Special Interest Group on Data Communication (SIGCOMM)*, 2017.
[13] A. Ortiz, F. Bonnin-Pascual, and E. Garcia-Fidalgo, "Vessel inspection: A micro-aerial vehicle-based approach," *J. Intell. Robot. Syst.*, vol. 76, no. 1, pp. 151–167, 2014.
[14] M. Beul, N. Krombach, M. Nieuwenhuisen, D. Droeschel, and S. Behnke, "Autonomous navigation in a warehouse with a cognitive micro aerial vehicle," in *Robot Operating System (ROS): The Complete Reference (Volume 2)*, A. Koubaa, Ed. Springer, 2017, pp. 487–524.
[15] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.
[16] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.
[17] A. Nash, S. Koenig, and C. Tovey, "Lazy Theta*: Any-angle path planning and path length analysis in 3D," in *Proc. of Nat. Conf. on Artificial Intelligence (AAAI)*, 2010.
[18] S. Vanneste, B. Bellekens, and M. Weyn, "3DVFH+: Real-time three-dimensional obstacle avoidance using an Octomap," in *Int. Workshop on Model-Driven Robot Software Engineering*, 2014.
[19] J. Israelsen, M. Beall, D. Bareiss, D. Stuart, E. Keeney, and J. van den Berg, "Automatic collision avoidance for manually tele-operated unmanned aerial vehicles," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.
[20] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, "Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system," in *Robot Operating System (ROS): The complete reference (Volume 2)*, A. Koubaa, Ed. Springer, 2017.
[21] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient algorithm for state-to-state quadrocopter trajectory generation and feasability verification," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.
[22] M. Beul, S. Houben, M. Nieuwenhuisen, and S. Behnke, "Fast autonomous landing on a moving target at MBZIRC," in *Proc. of European Conf. on Mobile Robots (ECMR)*, 2017.