2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Perception-aware Full Body Trajectory Planning for Autonomous Systems using Motion Primitives

Moritz Kuhne[1], Riccardo Giubilato[1], Martin J. Schuster[1], Maximo A. Roa[1]

***Abstract*— Many robotic systems rely on visual sensing to accomplish simultaneously the tasks of state estimation, mapping, and path planning. One one hand, the usage of camera sensors represents a power-efficient and lightweight option for solving this problem. On the other hand, these tasks pose requirements on the quality of the visual input (e.g. number of tracked features for Visual Odometry) that are often in contrast to the optimal viewpoint planning for local mapping and obstacle avoidance. Dealing with this constraint is actively researched in the field of perception-aware planning. The approaches delivered by this field mostly concern Micro air vehicles (MAVs), but could be applied to a larger group of robotic systems. We propose a perception-aware trajectory planner for a class of robotic systems that can orient their cameras independently from their direction of travel. By using motion primitives, our planner does not require differentiable models for motion and perception objectives. We evaluate our method in simulation, showing increased capabilities in localization-aware motions around obstacles, and demonstrate its run-time capability on a real planetary rover. The code is released publicly under github.com/DLR-RM/palp.**

I. Introduction

In order to perform tasks using full or partial autonomy, robotic systems need to plan paths to reach target locations, as well as estimate their state, e.g. their pose with respect to an arbitrary frame. Vision-based state estimation, by means of Visual (Inertial) Odometry or Visual SLAM algorithms, often require images densely and evenly populated with features. On the other hand, towards effective avoidance of obstacles and efficient traversal, path planning algorithms require coverage of the environment ahead of the system, in the direction of motion. This problem is addressed by perception-aware planning algorithms, which maximize perception- and motion-related objectives towards enhancing reliability for autonomous systems in perceptually challenging environments (see Fig. 1)

Most of these localization-aware algorithms have been developed for micro air vehicles (MAVs), especially quadcopters, with a camera fixed to their body frame. These systems offer one degree of freedom between their body orientation and their body's translational motion. More precisely, they can control their body's yaw angle, and therefore camera orientation along yaw, independently of their linear motion. Many perception-aware algorithms for MAVs plan trajectories for the flat outputs [1] of translational motion and yaw orientation [2], [3], [4], [5], [6], [7], [8], [9]. By

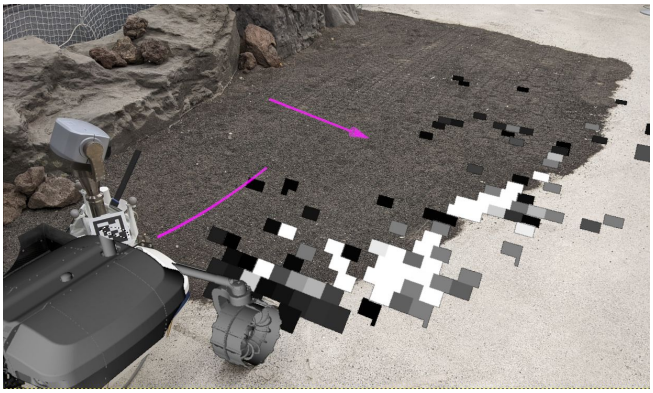

Fig. 1: A planetary rover traversing low-contrast gravel that provides few features for visual navigation. Features (black to white boxes for low to high feature density) are mostly aggregated along the high-contrast ridge between floor and gravel. A perception-aware trajectory is planned over the combined motions for body (pink line) and actuated camera orientation (pink arrow indicates viewing direction at the end of the planned body motion).

leveraging the capability of free camera orientation, these robots can fully utilize their sensing and motion abilities in the context of perception-aware navigation. This differs from planning solely for translational motions, where the camera's yaw orientation remains fixed to the direction of travel [10]. More specifically, the additional degree of freedom in planning offers increased abilities in negotiating perception and motion objectives, e.g. tracking features used for localization while circumventing obstacles. For the remainder of this paper, we use *full body* motions when referring to planning for the combined states of body motion and camera motion. In contrast, when fixing the camera orientation to the robot's motion, we refer to *body only* motion.

For autonomous navigation in unknown environments, few works consider the application of full body perception-aware planning on other robotic platforms that offer full body motions, such as planetary rovers [11]. The generalized problem of full body perception-aware trajectory generation thus involves search over the combined degrees of freedom in body and camera motions. Solutions to this problem consider both motion objectives based on visual information gathered along the path.

We propose to evaluate motion primitives of the combined camera and body dynamics as a full body perception-aware trajectory search problem. Our objective is to formulate a

* This work was supported by projects ARCHES (contract number ZT-0033) and iFOODis (contract number KA2-HSC-06i_FOODis).

[1]The authors are with Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Muenchener Str. 20, 82234 Wessling, Germany firstname.lastname@dlr.de

trajectory planning algorithm for perception-aware full body motion applicable to various mobile systems. Using full body motion primitives for camera orientation and body motion allows translating this algorithm to a large group of autonomous robots and tasks, as dynamics and cost functions can be modeled without the need for differentiability. We apply our solution in simulation on an MAV and in real-world experiments on a planetary rover. To the best of our knowledge, this displays the first time that perception-aware trajectory search based on motion primitives is done for the combined states of robot body and camera orientation.

At the core, our contributions are

- the design of a perception-aware trajectory planning algorithm that can be deployed on a broad range of autonomous robots
- the design of a motion planner for camera orientation and body motion to negotiate the needs for perception and motion in navigation on mobile systems
- evaluating our approach in simulation, showing the benefits for the accuracy of state estimation, and demonstrate its deployment on a real robot.

The rest of this paper is structured in the following manner. Section II reviews related work. In Section III we describe our algorithm of perception-aware full body trajectory planning for a generalized robotic system. The algorithm is demonstrated with experiments in simulation on an MAV and on a real-life planetary rover in Section IV. Finally, we draw conclusions and provide an outlook on future works in Section V.

II. Related Work

Receding horizon planning facilitates real-time responsiveness for mobile robots navigating unknown environments by continuously updating trajectories based on current sensor data, ensuring safe and efficient navigation amidst dynamically changing surroundings.

Perception-aware trajectory planning for MAVs is executed in a receding horizon fashion while evaluating localization objectives [10]. To achieve real-time responsiveness in navigating an unknown environment, a local solution is recomputed at a predefined rate for a limited time horizon. Additionally, feature quality for localization is considered in terms of motion-blur [2] and semantics [3]. Perception-aware trajectory planning is employed to discover unknown obstacles along the path to allow for fast flight in [4], and a known set of visual features is kept in the camera view while following a path in [5]. In [6], an MPC framework is used to keep a point of interest in the camera centre. Active tracking of dynamic obstacles is done in [7]. Perception-aware trajectory planning is used as sub-component for fast [12] and uncertainty-aware [8] exploration and surface reconstruction [9].

The approaches [3], [4], [5], [6], [7], [8], [9] plan over full body trajectories, i.e. decoupled direction of travel and camera orientation. In contrast [10], [12] align the camera orientation with the direction of travel and [2] orients the camera to face the goal location. These approaches with fixed camera orientations limit the capabilities to negotiate localization and motion objectives.

Only very few perception-aware trajectory planning algorithms are developed for platforms other than MAVs. In [11], a plan for the camera mast orientation of a planetary rover is created. This approach takes a given body trajectory as input and, therefore, does not plan for the combined system of robot body and camera. To the best of our knowledge, there are no perception-aware full body trajectory planners for other platforms than MAVs.

Application on multiple robotic platforms is considered by [13] by designing a risk-aware planner to work across multiple ground-based systems. [11] reasons that their camera orientation planner can also be adapted to other platforms beyond planetary rovers, that can move body and camera orientation independently. The software stack design significantly influences cross-platform portability, with a focus on mitigating platform specificity through the incorporation of modular design principles [14]. In line with this reasoning, end-to-end approaches [15] do not translate well to our design goal of deployment on a large group of systems.

Model based approaches solve the trajectory planning problem either by optimization or sampling. The authors of [3], [4], [5], [6], [7] chose to solve the problem by using an optimization approach. This requires robot dynamics, obstacle avoidance, path following, and perception objectives to be expressed as differentiable functions. Sampling-based approaches do not have this requirement and thus provide more flexibility in formulating the trajectory planning problem, therefore making it adaptable to various robotic systems and cost functions. Though run-time performance of sampling decreases with a growing number of degrees of freedom to explore, their computational demand is well controlled by the number of evaluated trajectory candidates. This makes them particularly well suited for our application in online trajectory search for autonomous systems navigating through unknown environments. Therefore, the results of sampling-based solutions [10], [2], [12], [8], [9], [11] are of special interest to our work.

III. Perception-aware Full Body Trajectory Planning

This section introduces our algorithm to tackle the challenge of perception-aware trajectory planning for autonomous systems with degrees of freedom in both body motion and camera orientation. Our approach is based on motion primitives. The underlying trajectory generators are system-specific and explained in more detail in Section IV. The evaluation of candidate primitives is generalized and explained in Section III-C. For our algorithm, we assume that a feature-based visual odometry (VO) is used for state estimation. The features tracked by the VO are integrated into a feature map. In the experiments in Section IV we chose to represent the feature map as a rolling OctoMap [16] that stores the amount of its integrated features per voxel. This representation removes multiple observations of the same features, by low-pass filtering the amount of features per

voxel on every integration of a new camera view. For the remainder of this paper, a feature refers to a voxelized feature in the feature map. The size of this robot-centric map can be limited to the expected distances of features used for state estimation.

A. Notation

The robot's full body state $\mathbf{x}(t)_{fb}$ is composed of the independently controllable states for body pose $\mathbf{x}_b(t)$ and camera orientation $\mathbf{x}_c(t)$, i.e. $\mathbf{x}(t)_{fb} = [\ \mathbf{x}_b(t)\ \mathbf{x}_c(t)\ ]^T$. A trajectory ρ over discretized planning time steps t_i from initial time t_0 to final time t_f is expressed as

$$\rho(t_i) = [\ \mathbf{x}_{fb}(t_i)\ \dot{\mathbf{x}}_{fb}(t_i)]^T\ \forall t_i \in [t_0, \ldots, t_f]. \tag{1}$$

For simplicity, when a variable or function depends on the robot state at a given time step, we express this with a dependency on that time step only.

B. Full Body Trajectory Generation

Full-body trajectory planning is performed in a two-stage approach. First, a set of body trajectories is generated and evaluated on motion objectives given in Section III-D. Second, a set of camera trajectories is added to each collision-free body trajectory and scored on perception objectives, as described in Section III-E. The resulting perception-aware trajectories for the combined system of robot body and camera are shown in Fig. 2. The underlying trajectory generators are system-specific and explained in more detail in Section IV.

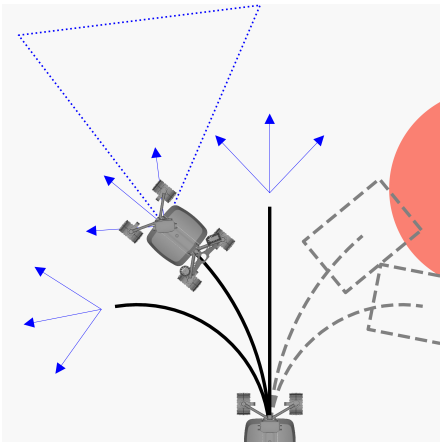

Fig. 2: Full-body trajectories are generated by augmenting collision-free body primitives (solid line trajectories are collision free, dashed lines indicate trajectories where the robot's bounding box is in collision, red area is an obstacle) with a set of camera motions (arrows indicate camera orientation, and an example of camera frustrum is shown by a dotted blue triangle).

C. Full-Body Trajectory Evaluation

Full-body trajectory candidates are evaluated by the sum of multiple costs for body motion and perception objectives. In total, we consider the costs for progression towards the goal (c_{goal}), obstacle distance (c_{obs}), co-visibility of features across initial and final robot states (c_{cv}), and camera alignment towards the direction of travel (c_{algn}). Thus, the total cost c_{total} of a trajectory ρ is given by

$$c_{total} = \sum_{\rho} \left(c_{goal} + c_{obs} + c_{cv} + c_{algn}\right). \tag{2}$$

This set of cost functions, inspired by [10], covers the minimal functionality for a goal-reaching task on autonomous system using perception. The additional cost c_{algn} for aligning robot motion and camera is required due to the added degrees of freedom between robot body and camera orientation. Other costs, like tracking error to a global path or control effort, could be added as additional terms if required. All costs are formulated as barrier functions, whereby each cost incrementally adds a marginal penalty within a defined range of inputs. Beyond this range, the barrier functions grow rapidly. By adopting this approach, we can identify a parameter set that effectively balances the competing demands of individual cost functions. For instance, in feature-rich environments, the localization cost should not dictate the robot's trajectory concerning obstacle avoidance, or goal progress should influence the robot's behavior less if it is far away from the goal.

D. Cost of body motion

Goal progress: In a long range navigation task, the cost for goal progress should not dictate the robot's immediate behavior ignoring localization objectives. Thus, we adopt the goal progress from [10] and reformulate it as a cost, computed as

$$c_{goal} = b_{goal}\, d(t_f) \left(\frac{d_{goal}}{d(t_0)}\right)^{k_{goal}} \tag{3}$$

where $d(t_0)$ and $d(t_f)$ are the goal distances between the current position and the planned trajectory's end position, respectively. The constants b_{goal}, d_{goal}, and k_{goal} form the cost function and are required to be > 0. Fig. 3 shows the goal progress cost from eq. 3. This cost function contributes more to the robot's behavior when the the robot is closer to the goal.

Obstacle distance: The cost for obstacle distance associates the summed cost over vertices $\mathcal{V}_j$ in the robot's bounding box $\mathcal{V}$ (see Fig. 2) along the trajectory

$$c_{obs} = b_{obs} \sum_{\rho} \sum_{\mathcal{V}} \left(\frac{d_{obs}}{d(\mathcal{V}_j(t_i))}\right)^{k_{obs}} \tag{4}$$

where $d(\mathcal{V}_j(t_i))$ is the L^2 distance of vertex $\mathcal{V}_j$ to an obstacle at sampled pose ρ_i. The barrier function c_{obs} is shaped by constants b_{obs}, d_{obs}, and k_{obs}, all being > 0. Here, the barrier parameter d_{obs} signals an obstacle distance that should be maintained. Fig. 3 displays the obstacle cost for various values of d_{obs}.

E. Perception objectives

Co-visibility: The authors of [17] propose co-visibility of features along a trajectory as a localization objective, i.e. features remain consistently visible within the camera's field of view across consecutive planned views, thereby enhancing the accuracy of state estimation. We compute the visible

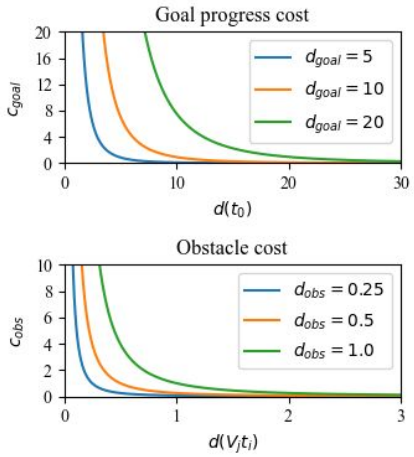

Fig. 3: Cost of body motion. Top: Goal progress cost function over $d(t_0)$, with $d(t_f) = d(t_0)$, $b_{goal} = d_{goal}^{-1}$, and $k_{goal} = 4$. Bottom: Obstacle cost over distance between a corner in the robot's bounding box to obstacle $d(\mathcal{V}_j(t_i))$.

voxelized features $\mathcal{F}(t_0)$ at the current state and at the trajectory's final view $\mathcal{F}(t_f)$. The amount $n(t_f)$ of co-visible features is given by

$$n(t_f) = \sum n_{f_k} \forall f_k \in F(t_0) \cap F(t_f), \tag{5}$$

where n_{f_k} is the number of features integrated in the co-visible voxel f_k. The co-visibility cost is then computed as

$$c_{cv} = b_{cv} \left(\frac{n_{cv}}{n(t_f)} \right)^{k_{cv}}. \tag{6}$$

The constants b_{cv}, n_{cv}, and k_{cv} are all > 0. Rationale is, n_{cv} is an amount of features that is required to be tracked along the trajectory for successful state estimation. If the number of features $n(t_f)$ is lower than n_{cv}, the cost grows quickly.

We assume overlap between the initial and final field of view, and we check co-visibility between the current state and the final state of a candidate only. This in turn allows to increase the amount of explored full-body primitives. For systems with more aggressive motions or longer trajectories, additional poses along the trajectory need to be checked for co-visibility.

Camera alignment: We assign a cost c_{algn} for aligning the camera orientation $\hat{v}(t_i)$ with the direction of body velocity $\dot{\mathbf{x}}_b$

$$c_{algn} = b_{algn} \sum_{\rho} \left(\frac{1}{\phi_{algn}} \cdot \arccos \frac{\hat{v}(t_i) \cdot \dot{\mathbf{x}}_b(t_i)}{\|\dot{\mathbf{x}}_b(t_i)\|} \right)^{k_{algn}}, \tag{7}$$

where $\hat{v}(t_i) = f(\mathbf{x}_{fb})$ has unit length and is a function of the full body pose. Constants c_{algn}, ϕ_{algn} and k_{algn} are > 0. The parameter ϕ_{al} is determined by the camera's field of view. If the body moves within the field of view, this barrier function contributes a small cost, and a large cost otherwise.

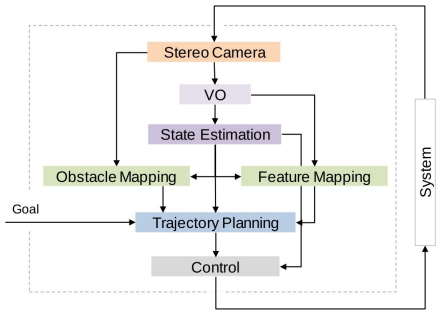

Fig. 4: Overview of the navigation stack.

IV. Experiments

The portability of the proposed method across autonomous systems is a key focus of this research. In the following experiments, we evaluate it using our robot simulator [18] and demonstrate its implementation on a real planetary rover [19]. The systems operate in an unknown environment, and trajectory planning is performed in a receding horizon manner. All experiments in simulation run the same modular navigation stack, shown in Fig. 4. The state-estimation differs from our typical setup by purely using VO instead of visual-inertial odometry (VIO). We do so to evaluate purely the effects of visual input, i.e., poor perception quality directly affects the state-estimation as no IMU data is integrated to stabilize the estimation. However, all experiments could have been performed with a VIO setup. For demonstration on the real rover, state-estimation is based on VIO and wheel odometry.

As a starting point for our perception-aware implementation, we employ the perception-agnostic trajectory planner of the ROS *navigation* package [20] for the robot's body motion. We chose this planner as it is designed to be executed in a receding horizon environment and utilizes ROS as middleware [21] as we do in our systems. Additionally, we use ROS's rviz for visualization.

For the simulation experiments, we compare the proposed full-body planner based on motion primitives to its counterpart from [10], which plans for body motion only and aligns the camera orientation to the direction of travel. Comparing to other active methods is hard due to the lack of approaches that assess active vision systems [22], and very scarce open source code availability.

A. Execution on a Simulated Planetary Rover

In this experiment, we test our approach in a simulation of the lightweight rover unit [19] to navigate along a rocky outcrop in a feature-light desert. Sandy rubble accumulated along the outcrops foot and forms an obstacle. In this experiment, a perception-aware planner must manage the distance to the obstacle at the base of the rock face while directing its focus towards the features on the rock surface. The experiment setup is shown in Fig. 5.

We perform trajectory planning of the robot's body in a 3-dimensional subspace and denote the body pose as $\mathbf{x}_b = [\, x\ y\ \theta\,]^T$. The orientation of the actuated mast camera unit

is given by $\mathbf{x}_c = [\ \phi\ ]$. We construct the full-body pose as $\mathbf{x}_{fb} = [\ \mathbf{x}_b\ \mathbf{x}_c\ ]^T$. A set of collision-free body trajectories is generated by sampling over accelerations in x and θ. Next, we sample over accelerations in the camera's ϕ orientation to dynamically generate a set of feasible camera motions.

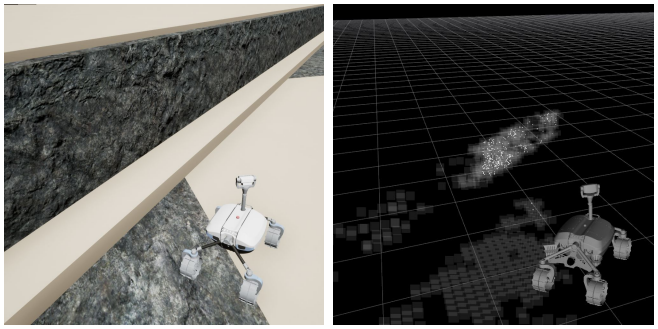

Fig. 5: Simulation of a planetary rover traversing a feature-deprived desert. Left: The rover strides along a rock face with sandy talus. Right: Visualization of feature density and mapping. Visual features only exists on rock. White dots show currently visible features, and grey boxes visualize the integrated feature map.

Fig. 6 shows the resulting robot paths for varied initial conditions. A total of nine trajectories were generated by executing full-body movements with our approach (blue lines in Fig. 6); eight of them successfully navigate along the rock face. Conversely, nine trials were generated with the camera oriented along the direction of travel (red lines in Fig. 6); all fail in this experiment, see Table I. While the full-body planner accomplishes this by adjusting the camera orientation, the body-only planner must maneuver closely to the obstacle to maintain visibility of features at the edges of its field of view. Two error scenarios arise for the body-only planner. Firstly, it may fail to find a new collision-free trajectory. Secondly, the state estimation diverges due to the absence of visual features, resulting in the termination of trajectory planning when the estimated initial planning position is within an obstacle.

B. Execution on a Simulated Hexacopter

We examine the suitability of our trajectory planner for implementation on a a MAV, testing it in a simulation of the ARDEA hexacopter [23]. Within the MAV's operating environment, features predominantly exist on the rocky terrain and obstacles, as illustrated in Fig. 7.

Trajectory planning for the MAV body is done within a 2D subspace, maintaining a fixed height. The robot pose is denoted as $\mathbf{x}_b = [\ x\ y\ ]^T$. The orientation of the camera unit is determined by the MAV's yaw rotation, represented as $\mathbf{x}_c = [\ \psi\ ]$. We synthesize the full-body pose within the local subspace as $\mathbf{x}_{fb} = [\ \mathbf{x}_b\ \mathbf{x}_c\ ]^T$. Since our motions are conservative, we neglect the camera's motions in roll and pitch that are introduced due to the MAV's kinodynamic constraints for perception-aware planning.

Trajectories $\rho(t_i) = [\ \mathbf{x}_{fb}(t_i)\ \dot{\mathbf{x}}_{fb}(t_i)\ ]^T$ are generated for the MAV by sampling over accelerations $\ddot{\mathbf{x}}_{fb}(t_i)$. Propagating these accelerations yields the final states $\rho(t_f)$ of the planned trajectory. The interpolation between the initial and final states is computed by generating intermediate states along the trajectory candidate $\rho(t_i)$, following the approach in [24]. To compare the full-body planner, we create a body-only planner for which the camera's orientation is aligned with the body motion.

For this experiment, our focus lies in investigating how the perception-aware planners navigate the trade-off between obstacle distance and visual objectives across a diverse range

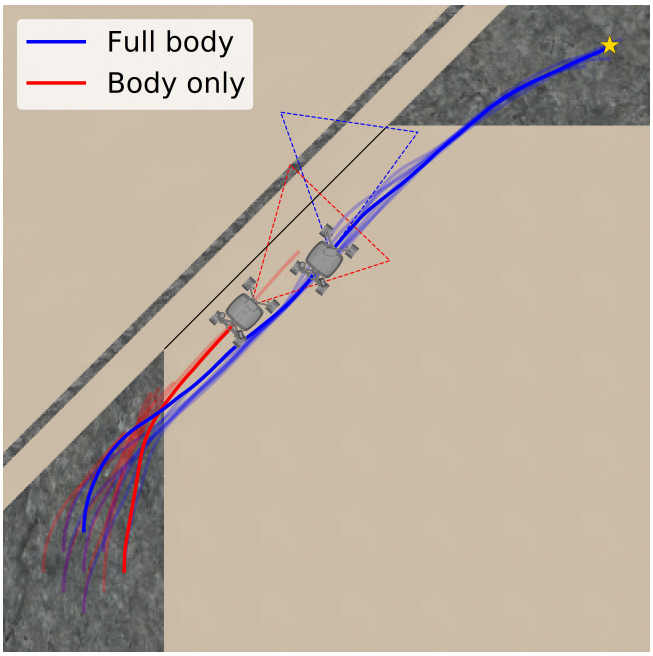

Fig. 6: Robot paths for full-body (blue lines) and body-only (red lines) planning along a rock face towards a goal region (star). Triangles show the robot's view at a sample along the path. The body-only trajectories stay close to the obstacle to keep its features in the field of view. The full-body planner tracks features by by orienting its camera towards the rock face, allowing it to keep a larger distance to the obstacle.

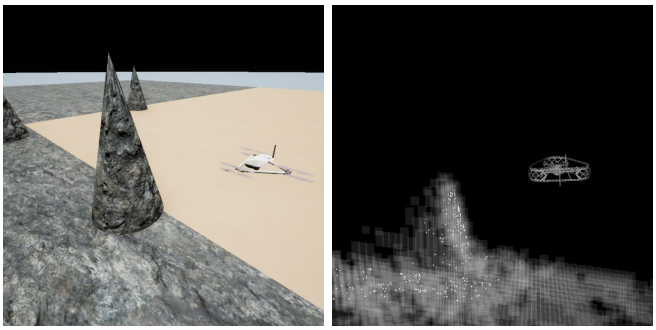

Fig. 7: Execution on a simulated hexacopter: (a) Test environment for navigation with an MAV, with feature-rich obstacles. (b) Visualization of feature mapping. Features do not exist on sand. Grey boxes represent the integrated feature visible on the rock surface.

of initial and final conditions. The paths resulting for varied initial and end positions are shown in Fig. 8. The blue and pink runs are especially demanding, since the goal is on the back of an obstacle in the feature-deprived sandy environment.

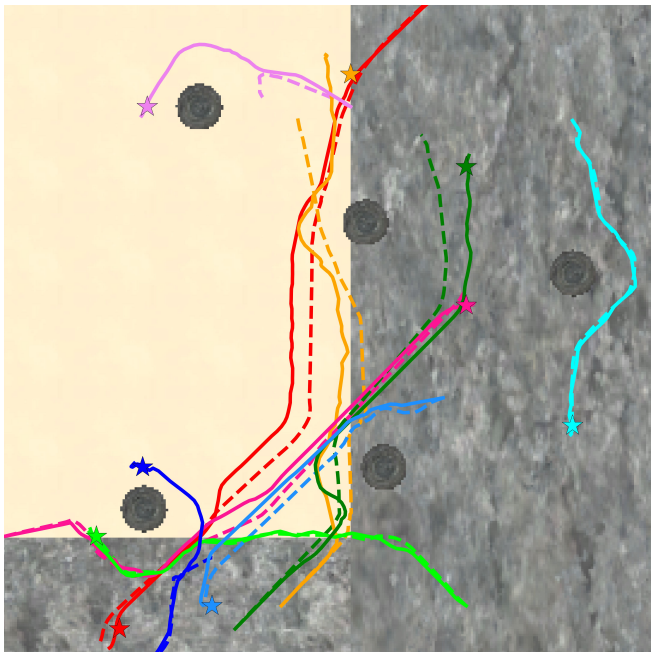

Fig. 8: The trajectories of the robot for both full-body (solid lines) and body-only (dashed lines) planning are illustrated as they navigate a feature-light environment towards designated goals (stars). Lines of the same color are associated to the same start and goal conditions.

For pairs of start and goal locations, we compute obstacle distance and state-estimation error as metrics for the motion and perception objectives, respectively. The challenging blue and pink paths are excluded from evaluation, since the body-only planner did not approach the goal, whereas the full-body planner did, see Table I. The body-only trails were stopped once no features present on the obstacles were co-visible in the current planning cycle. This criterion is based on the incapability to recover these features with a local trajectory planner. For completeness, we also add the statistics for the full-body planner over all trials. In total, the full-body planner approaches the goal in all nine trials, while the body-only planner fails in two out of nine runs.

TABLE I: Successful runs out of nine trials in simulation experiments.

Experiment	Full-Body	Body-Only
Simulated Rover	**7**	0
Simulated Hexacopter	**9**	7

The minimum obstacle distance d_{min} is given as the smallest distance of any vertex in the robot's bounding box during a single trial, with statistical values for average $\overline{d_{min}}$ and standard deviation $\sigma(d_{min})$. The results for d_{min} are

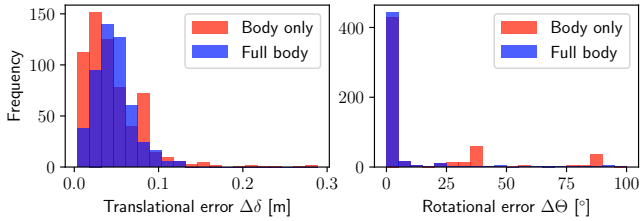

Fig. 9: Frequencies for translational error $\Delta\delta$ and rotational error $\Delta\theta$ over segments of 1 m.

shown in Table II. For comparison, the barrier experiments parameter in Eq. 4 is $d_{obs} = 0.5\,\mathrm{m}$. We can observe an increase in $\min(d_{min})$ and $\overline{d_{min}}$ for the full-body planner compared to its counterpart with body-only motions. This increase is mostly correlated to the green, yellow, and red trials. In these tests the body-only planner has to choose a trajectory close to an obstacle to maintain the features present on the obstacle at the edges of the field of view. The full-body planner navigates around these obstacles at a larger distance by orienting its camera towards the feature-rich obstacles. For the cyan and lime green trials, the two planners produce highly similar paths. This effect is due to the abundance of features along the paths. In these cases, the co-visibility cost function returns low values and therefore does not dictate the robot's movement.

TABLE II: Statistics for minimum obstacle distance d_{min}

Test	$\min(d_{min})$ [m]	$\overline{d_{min}}$ [m]	$\sigma(d_{min})$ [m]
Full-body	**0.65**	**0.70**	**0.03**
Body-only	0.21	0.53	0.17
Full-body (all)	0.34	0.63	0.13

As metric for state estimation we compute the translational error $\Delta\delta$ and rotational error $\Delta\theta$ over segments of 1 m along the trajectory. We compare the state estimation proficiency of the full-body and body-only planners based on the average, mean, and standard deviation of $\Delta\delta$ and $\Delta\theta$.

The errors in translation $\Delta\delta$ and error distribution are highly similar for the full-body and body-only trials, see Table III. The errors in rotation $\Delta\theta$ show lower values in $\sigma(\Delta\theta)$ for the full-body planner, see Table IV. This is attributed to higher frequencies in large errors in $\Delta\theta$ for the body-only planner, which have only small effects on $\tilde{\Delta\theta}$, see the histogram in Fig. 9.

The results on the motion metric $\min(d_{min})$ and perception metric $\Delta\theta$ and $\Delta\delta$ suggest that the full-body primitive-based trajectory planner performs better in negotiating the competitive demands in challenging trials.

C. Experiment on a real Planetary Rover

We deploy the developed algorithm on a planetary rover [19]. The robot used in this experiment is equipped with an `Intel(R) Xeon(R) W-11865MLE` processor and 62 GB of RAM. The trajectory planning algorithm is executed at 10 Hz. For this experiment, we integrate VIO data to provide

TABLE III: Statistics on translational errors $\Delta\delta$

Test	$\overline{\Delta\delta}$ [m]	$\tilde{\Delta\delta}$ [m]	$\sigma(\Delta\delta)$ [m]
Full body	0.04	0.04	**0.02**
Body only	0.04	**0.03**	0.03
Full body (all)	0.04	0.04	**0.02**

TABLE IV: Statistics on rotational errors $\Delta\theta$

Test	$\overline{\Delta\theta}$ [°]	$\tilde{\Delta\theta}$ [°]	$\sigma(\Delta\theta)$ [°]
Full body	**0.76**	**0.89**	19.91
Body only	17.56	1.12	34.20
Full body (all)	7.18	0.98	**19.80**

a high-frequency, smooth, and robust state-estimation on the real system.

In this lab environment, shown in Fig. 1, features are mostly accumulated at the junction of floor and gravel. The navigation goal is a point ahead of the rover, across the feature-deprived field of gravel. The resulting path after executing the full-body trajectory planner in a receding horizon fashion is shown in Fig. 10. The robot progresses towards the goal while circumventing an obstacle. Comparable to the outcomes observed in our simulation experiments, the planner orients its camera towards the feature-rich region at the border of the gravel.

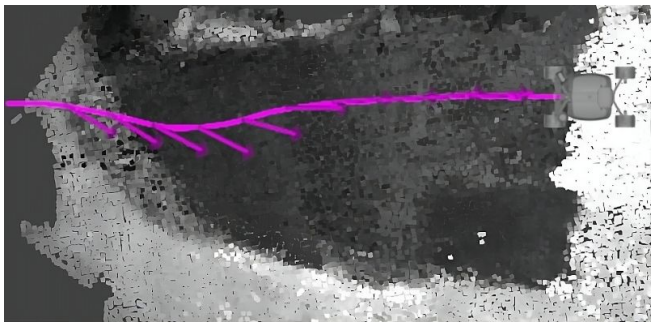

Fig. 10: Top down visualization of the environment shown in Fig. 1, where features are mostly detected at the edge between gravel (dark area) and floor (light area). The resulting path of a goal reaching task using a full body perception-aware planner, consisting of the body path (pink line) and camera orientations (pink arrows).

V. Conclusion

In this paper, we proposed a perception-aware trajectory planner that searches over the combined body and camera states. To make the algorithm applicable to a large group of robots, we design it to use motion primitives as candidates for the trajectory generation. The autonomous systems considered for our method employ a modular navigation stack that integrates feature-based V(I)O measurements for state estimation. We demonstrate the developed algorithm's effectiveness and portability by deploying it on an MAV and planetary rover, both in simulation and in the real world.

Our experiments show improved perception-aware motion capabilities for the full-body planner when compared to a baseline planner with a camera fixed to the body motion. This effect is especially pronounced in difficult situations, where obstacle avoidance and feature tracking demands competitive actions for the body-only planner. The outcome is a higher success rate for our proposed full-body planner. For navigation in feature-rich environments, where the full-body planning capability is not required, our proposed algorithm yields highly similar paths to its body-only counterpart. This is credited to the barrier-like formulation for all costs, enabling negotiation and tuning of competing behaviors introduced by individual cost functions.

The proposed algorithm perceives its environment towards the goal by posing an alignment cost between body velocity and camera orientation. This choice, however, may lead to situations where the robot either moves towards unmapped terrain or a sufficient amount of features is not visible during motion. These risky behaviors can be reduced by increasing the gain for camera alignment. Nevertheless, this mitigation approach has its limitations, and aligning the camera restricts the robot's full-body motion capabilities. Future work could consider replacing the alignment cost c_{algn} with a behavior that actively and safely explores the environment.

The quantitative results presented in this paper are based on comparison on a limited number of scenarios. Especially the interaction between the competing cost functions for tracking known features and exploration could be studied by more variation in tested environments. Furthermore, the benefits of increased state-estimation accuracy, when compared to the base line body only perception-aware planner, could be proved in more experiments on a real robot.

ACKNOWLEDGMENT

The authors want to thank all the colleagues of the mobile robotics teams at DLR Institute for Robotics and Mechatronics who have helped with the development and experimentation with the LRU (rover) and ARDEA (hexacopter) systems.

References

[1] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*. Shanghai, China: IEEE, May 2011, pp. 2520–2525.

[2] Wu, "Perception-Aware Receding Horizon Trajectory Planning," vol. 10, pp. 87 911–87 922, 2022.

[3] L. Bartolomei, L. Teixeira, and M. Chli, "Perception-aware Path Planning for UAVs using Semantic Segmentation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 5808–5815.

[4] B. Zhou, J. Pan, F. Gao, and S. Shen, "RAPTOR: Robust and Perception-Aware Trajectory Replanning for Quadrotor Fast Flight," *IEEE Transactions on Robotics*, pp. 1–18, 2021.

[5] I. Spasojevic, V. Murali, and S. Karaman, "Perception-aware time optimal path parameterization for quadrotors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. Paris, France: IEEE, May 2020, pp. 3213–3219.

[6] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "PAMPC: Perception-Aware Model Predictive Control for Quadrotors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid: IEEE, Oct. 2018, pp. 1–8.

[7] J. Tordesillas and J. P. How, "PANTHER: Perception-Aware Trajectory Planner in Dynamic Environments," *IEEE Access*, vol. 10, pp. 22 662–22 677, 2022.
[8] C. Papachristos, S. Khattak, and K. Alexis, "Uncertainty-aware receding horizon exploration and mapping using aerial robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. Singapore, Singapore: IEEE, May 2017, pp. 4568–4575.
[9] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An Efficient Sampling-Based Method for Online Informative Path Planning in Unknown Environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, Apr. 2020, number: 2.
[10] Z. Zhang and D. Scaramuzza, "Perception-aware Receding Horizon Navigation for MAVs," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD: IEEE, May 2018, pp. 2534–2541.
[11] J. Strader, K. Otsu, and A.-a. Agha-mohammadi, "Perception-aware Autonomous Mast Motion Planning for Planetary Exploration Rovers," *Journal of Field Robotics*, vol. 37, no. 5, pp. 812–829, Aug. 2020, arXiv: 1912.06898.
[12] M. Dharmadhikari, T. Dang, L. Solanka, J. Loje, H. Nguyen, N. Khedekar, and K. Alexis, "Motion Primitives-based Path Planning for Fast and Agile Exploration using Aerial Robots," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. Paris, France: IEEE, May 2020, pp. 179–185.
[13] D. D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A.-A. Agha-Mohammadi, "STEP: Stochastic Traversability Evaluation and Planning for Risk-Aware Off-road Navigation," *arXiv:2103.02828 [cs, eess]*, Jun. 2021, arXiv: 2103.02828.
[14] M. W. Mueller, S. J. Lee, and R. D'Andrea, "Design and Control of Drones," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 161–177, May 2022.
[15] Y. Song, K. Shi, R. Penicka, and D. Scaramuzza, "Learning Perception-Aware Agile Flight in Cluttered Environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. London, United Kingdom: IEEE, May 2023, pp. 1989–1995.
[16] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: an efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013, number: 3.
[17] V. Murali, I. Spasojevic, W. Guerra, and S. Karaman, "Perception-aware trajectory generation for aggressive quadrotor flight using differential flatness," in *2019 American Control Conference (ACC)*. Philadelphia, PA, USA: IEEE, Jul. 2019, pp. 3936–3943.
[18] M. Sewtz, H. Lehner, Y. Fanger, J. Eberle, M. Wudenka, M. G. Muller, T. Bodenmuller, and M. J. Schuster, "URSim - A Versatile Robot Simulator for Extra-Terrestrial Exploration," in *2022 IEEE Aerospace Conference (AERO)*. Big Sky, MT, USA: IEEE, Mar. 2022, pp. 1–14.
[19] M. J. Schuster, S. G. Brunner, K. Bussmann, S. Büttner, A. Dömel, M. Hellerer, H. Lehner, P. Lehner, O. Porges, J. Reill, S. Riedel, M. Vayugundla, B. Vodermayer, T. Bodenmüller, C. Brand, W. Friedl, I. Grixa, H. Hirschmüller, M. Kaßecker, Z.-C. Márton, C. Nissler, F. Ruess, M. Suppa, and A. Wedler, "Towards Autonomous Planetary Exploration: The Lightweight Rover Unit (LRU), its Success in the SpaceBotCamp Challenge, and Beyond," *Journal of Intelligent & Robotic Systems*, vol. 93, no. 3-4, pp. 461–494, Mar. 2019, number: 3-4.
[20] "Robot Operating System (ROS)," in *Robot Operating System (ROS)*, ser. Studies in Computational Intelligence, A. Koubaa, Ed. Cham: Springer International Publishing, 2017, vol. 1, pp. 112–156.
[21] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," vol. 3, 2009, p. 5.
[22] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, Feb. 2018.
[23] P. Lutz, M. G. Müller, M. Maier, S. Stoneman, T. Tomić, I. Bargen, M. J. Schuster, F. Steidle, A. Wedler, W. Stürzl, and R. Triebel, "ARDEA—An MAV with skills for future planetary missions," *Journal of Field Robotics*, vol. 37, no. 4, pp. 515–551, Jun. 2020, number: 4.
[24] M. W. Mueller, M. Hehn, and R. D'Andrea, "A Computationally Efficient Motion Primitive for Quadrocopter Trajectory Generation," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1294–1310, Dec. 2015, number: 6.