

# Low-Level Active Visual Navigation: Increasing Robustness of Vision-Based Localization Using Potential Fields

Rômulo T. Rodrigues<sup>1b</sup>, Meysam Basiri<sup>1b</sup>, A. Pedro Aguiar<sup>1b</sup>, and Pedro Miraldo<sup>1b</sup>

**Abstract**—This letter proposes a low-level visual navigation algorithm to improve visual localization of a mobile robot. The algorithm, based on artificial potential fields, associates each feature in the current image frame with an attractive or neutral potential energy, with the objective of generating a control action that drives the vehicle towards the goal, while still favoring feature rich areas within a local scope, thus improving the localization performance. One key property of the proposed method is that it does not rely on mapping, and therefore it is a lightweight solution that can be deployed on miniaturized aerial robots, in which memory and computational power are major constraints. Simulations and real experimental results using a mini quadrotor equipped with a downward looking camera demonstrate that the proposed method can effectively drive the vehicle to a designated goal through a path that prevents localization failure.

**Index Terms**—Aerial systems, perception and autonomy, visual-based navigation, sensor-based control.

## I. INTRODUCTION

THE problem of estimating the pose of a robot with respect to its environment, usually denoted as the localization problem, is one of the key challenges imposed on the deployment of autonomous Micro Aerial Vehicles (MAVs) in GPS-denied environments. Also addressed as state estimation, localization is of utmost importance, as the performance of other primary tasks such as autonomous navigation and obstacle avoidance relies on its accuracy. Beside compromising the mission, localization failure is potentially dangerous for humans and buildings in the vicinity, and the robot itself.

Manuscript received September 10, 2017; accepted February 6, 2018. Date of publication February 27, 2018; date of current version March 23, 2018. This letter was recommended for publication by Associate Editor A. Elfes and Editor J. Roberts upon evaluation of the reviewers' comments. This work was supported by FCT Project [UID/EEA/50009/2013] and Grant [SFRH/BPD/111495/2015]. (Corresponding author: Rômulo T. Rodrigues.)

R. T. Rodrigues, M. Basiri, and P. Miraldo are with the Institute for Systems and Robotics, Instituto Superior Técnico, University of Lisboa, Lisboa 1649-004, Portugal (e-mail: romulo.rodrigues@tecnico.ulisboa.pt; meysam.basiri@tecnico.ulisboa.pt; pedro.miraldo@tecnico.ulisboa.pt).

A. P. Aguiar is with the Faculty of Engineering, University of Porto, Porto 4099-002, Portugal (e-mail: pedro.aguiar@fe.up.pt).

This letter has supplemental downloadable multimedia material available at <http://ieeexplore.ieee.org>, provided by the authors. The Supplementary Materials contain a video briefly explaining the proposed feature-based active navigation method. In addition, it shows footage of experiments conducted with a mini-quadrotor in a real world scenario, comparing the performance of visual SLAM for both passive and active navigation methods. This material is 7.63 MB in size.

Digital Object Identifier 10.1109/LRA.2018.2809628

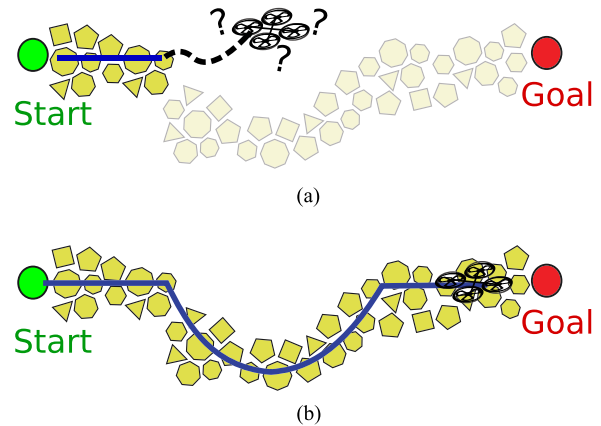


Fig. 1. Fig. (a) illustrates the problem of navigation without taking into consideration features in the environment, which can lead to localization failure. Fig. (b) presents the proposed solution that drives the vehicle towards a feature-rich area, enhancing localization robustness.

MAV localization in GPS-denied environments can be achieved with the aid of external systems such as motion tracking cameras. However, since such systems are not always available or practical, much research effort has been devoted to the development of alternative systems relying on the use of lightweight onboard sensors. The use of vision-based approaches for MAV localization has been the subject of many research works lately, e.g., [1]–[3]. Effective solutions combine visual estimation algorithms and inertial data within a filtering framework [4], [6]–[8]. In particular, recent solutions such as ORB-SLAM [7] and SVO [6] are accurate in cluttered, feature rich environments and high texture scenarios. However, when visual cues are not available, the vision pipeline fails and the algorithm must be reinitialized. Moreover, during the re-initialization step state estimation relies on model propagation and inertial data, compromising localization and map consistency. If visual cues are not detected within few seconds, state estimation error and uncertainty grow fast and the vehicle may get lost, as is illustrated in Fig. 1(a).

This letter proposes a low-level strategy for MAVs to improve Visual Odometry (VO) performance and map consistency. The solution aims at guiding the vehicle through paths with high quality image features. The method builds upon an Artificial Potential Field (APF) framework, in which each image feature is associated with a corresponding potential level. This yields

an additional reference velocity to the vehicle's control loop aimed at biasing its motion towards a feature-rich region in the current image frame. Spatial goal and APF-generated velocities are combined such that the vehicle drives toward the goal, while avoiding texture-less and non-static features regions, as illustrated in Fig. 1(b). This behaviour has the potential to reduce the growth of state estimation uncertainty and improve the localization accuracy. In addition, since this method does not rely on a map, it will not experience the same type of errors as in active SLAM solutions and, most important, it can be integrated in a complementary fashion. The proposed solution is specially appealing for small scale vehicles with low memory and limited processing power constraints that prevent large map storage, loop closure detection, and global pose optimization. For these systems, visual SLAM is constrained to a low number of keyframes. Once the allocated memory is full, previous keyframes are discarded as new frames arrive [2]. Yet, local map consistency is required for proper 3D world point initialization.

The paper is organized as follows: Section II summarizes related work regarding passive localization, active localization, and artificial potential fields. Section III introduces some basic notation and definitions. In Section IV describes the new method proposed. Section V presents simulation and experimental results. Finally, Section VI contains the final remarks. This letter builds upon and extends previous results that were presented in [9] by the authors. In particular, simulations validate the proposed algorithm in different scenarios, the effect of some parameters on the proposed is evaluated, the benefits of an auto-tuning technique are demonstrated, and extensive experiments on a real world scenario demonstrate the significant improvement on VO and SLAM performance for specific cases using the proposed method. The supplemental multimedia material that accompanies this manuscript contains footage of the real world experiments.

## II. LITERATURE REVIEW

This section provides a general overview of the foundations of the problem tackled in this work. Relevant solution regarding visual odometry, active localization, and artificial potential field are discussed.

### A. Visual Odometry

VO estimates the camera motion by taking into consideration the motion of tracked features in two consecutive frames. Thus, it is a dead-reckoning method, i.e., estimates relative, rather than absolute pose. In passive VO [5], interest image regions are initialized and tracked as a consequence of navigation. Recent solutions, e.g., [6]–[8], have achieved impressive results. The keypoint method developed in [7] extracts features from salient image regions, to recover camera pose using epipolar geometry. The hybrid method proposed in [6] combines both features and photometric information at high frame rates for a robust pose estimation. The dense strategy in [8] exploits photometric error to recover camera pose, integrating lens and camera parameters. The two last methods operate directly over the pixel intensity, being therefore more robust and accurate in

sparsely textured environment. However, as suggested in [8], the performance depends on the camera model, frame rate, and environment brightness. Nonetheless, we are not aware of any VO that does not require a quasi-static scenario and texture for working properly. The method addressed in this letter assures these two underlying assumptions hold during flight.

### B. Active Localization

Active localization adds visual information in the control loop, with the goal of minimizing pose uncertainty. The problem is addressed in the literature under different formulations, with significant overlapping: active-SLAM [10]–[12], planning under uncertainty [13], [14], or next-best-view [15], [16].

Davison and Murray [10] define the main goal in active feature selection as building a map of features, which helps localizing the robot rather than an end result in itself. Their solution is not active in the sense that it does not actuate on the camera path. Instead, it chooses the best features to fixate the camera's view. Vidal-Calleja *et al.* [11] extends the former solution proposing a control law that drives the camera to the location that maximizes the expected information gain. Based on information theory, features corresponding to the hardest prediction measurement are chosen. A formal approach that relates SLAM observability and vehicle motion is addressed by Bryson and Sukkarieh [12]. The proposed on-line path planner decides whether explore new regions or revisit known features to improve localization. Trajectories that excite locally unobservable modes are preferred.

Considering a known map (given *a priori*), the navigation task can be improved by planning routes that favor texture-rich areas. Achtelik *et al.* [13] addressed a Rapid-exploring Random Belief Tree (RRBT) framework that incorporate MAV dynamics and pose uncertainty. The method fails in a non-static scenario, or under illumination changes. In [14], Rapid-exploring Random Trees (RRT\*) is extended, to take into account the pose uncertainty. Most informative trajectories are selected using Fischer informative matrix. Previous non-mapped or non-static regions have impact on local edges and vertexes affected by new information. While local changes in the nodes handle small discrepancies between map and environment, these methods are not scalable since an offline path planner requires an accurate map.

Sadat *et al.* [15] propose a scoring function that takes into account the expected number of features for a given camera's viewpoint, using a mesh of triangles. The cost function of the optimized path planner includes path length and expect number of features. Mostegel *et al.* [16] propose a set of measurements, including geometric point quality and recognition probability, to analyze the impact of possible camera motions, and avoid localization loss. The solution relies on SLAM key-frames data. A local planner assures the vehicle reaches the destination in unexplored maps.

### C. Artificial Potential Fields

Early work on artificial potential fields was done by Khatib [17] for real-time obstacle avoidance. In this set-up, the spatial goal was associated to an attractive force, while obstacles, detected on-the-fly, were associated to a repulsive force. In contrast

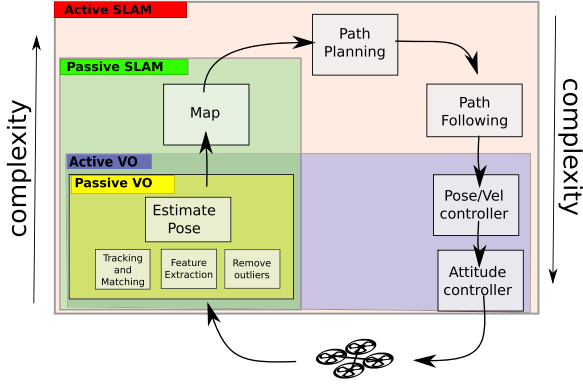


Fig. 2. Comparison between the method proposed in this letter (here denoted as Active VO) against most of the state-of-the-art alternatives (identified as Active SLAM).

to prior works that tackled obstacle avoidance as a high-level planning problem, APF proposed a low level real-time local solution. Furthermore, the author suggested integrating both high-level and low-level for better performance. The popularity of the potential fields approach has grown steadily and, lately, it has found new applications besides obstacle avoidance, such as in navigation [18], loop-closure detection [19] and mobile robot exploration [20].

Fig. 2, inspired in [21], illustrates both active and passive localization frameworks. Most active localization solutions addressed in the literature are built on top of SLAM, and are therefore relatively complex and map dependent. In contrast, our method can be classified as active VO: it does not require a map, but only features selected as inliers in the current frame.

### III. NOTATIONS AND DEFINITIONS

Let  $\{B\}$  and  $\{I\}$  denote the 3D body fixed frame attached to the vehicle and the 2D image plane frame attached to image plane, respectively. The origin of  $\{B\}$  coincides with the center of gravity of the vehicle and the origin of  $\{I\}$  corresponds to the top-left image pixel. Vectors are described in lower case bold and a leading superscript indicates its coordinate frame. The homogeneous coordinate of vector  $\mathbf{v}$  is denoted as  $\bar{\mathbf{v}}$ . When a vector is described in  $\{I\}$ , the leading superscript is omitted. Matrices are written in upper case and sets in calligraphic letter. The transformation from body to image frame  $T = K[R|\mathbf{t}] \in \mathbb{R}^{3 \times 4}$  is known, where  $R$  and  $\mathbf{t}$  denote the rotation and translation from body frame to the downward looking camera, respectively, and  $K$  is the intrinsic parameter matrix of the camera. For the sake of simplicity, the camera frame is omitted, as usually a static or a known transformation relates camera and body frames.

A high level positioning controller computes  ${}^B\mathbf{v}_g = (v_x, v_y, 0) \in \mathbb{R}^3$ , a velocity reference that drives the vehicle to the spatial goal. Notice that only planar motion is considered, that is, the vehicle keeps a constant height. Also, let  $\mathbf{p} = (u, v) \in \mathbb{R}^2$  be the undistorted coordinates of an image feature [22]. In particular,  $\mathcal{F}$  is the set of features tracked and selected as inliers in the current image frame.

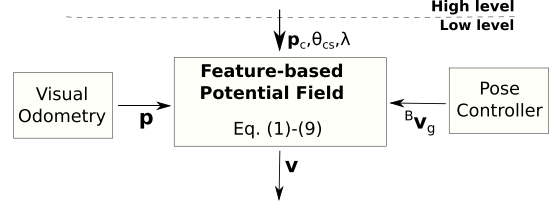


Fig. 3. Block diagram representing the proposed APF framework, and its communication with the other modules.

### IV. LOW-LEVEL ACTIVE VISUAL NAVIGATION

This section describes the method proposed to compute the velocity reference that drives the vehicle towards a spatial goal, while avoiding low feature areas. This is achieved by adding a component to the goal velocity vector  ${}^B\mathbf{v}_g$  so as to favor rich regions regarding features. A block diagram shown in Fig. 3 allows visualizing the information flow addressed in the remainder of this section. The proposed method is fed by a visual odometry algorithm, which provides inlier features, and a pose controller, e.g., path-following algorithm, that computes the goal velocity reference. The desired vehicle velocity is computed by the proposed low level active visual navigation method. A higher level system, such as a mission controller, may dynamically tune the parameters of the method proposed, which will be presented and discussed next.

#### A. Features to Charge

To each feature, an attractive or neutral potential energy is associated. Associating similar potential energy to every feature in the image frame is not adequate, since the vehicle could be easily trapped at a local minimum or subject to sudden changes when new features are extracted. Instead, taking advantage of the fact that the camera provides bearing information, the proposed method considers the orientation of each feature w.r.t. the direction of  $\mathbf{v}_g$ , i.e., the projection of the goal velocity in the image frame. As expected in a potential field framework, the final goal itself plays a role in the local decision making process. Let  $\mathbf{p}_o = (c_x, c_y)$  be the camera optical center obtained in the camera calibration step. For each feature  $\mathbf{p}_i \in \mathcal{F}$ , compute

$$\hat{\mathbf{p}}_i = \mathbf{p}_i - \mathbf{p}_o \quad (1)$$

$$\theta_i = \arccos \left( \frac{\langle \hat{\mathbf{p}}_i, \mathbf{v}_g \rangle}{\|\hat{\mathbf{p}}_i\| \|\mathbf{v}_g\|} \right), \quad (2)$$

where  $\theta_i \in [0, \pi]$  and  $\mathbf{v}_g \sim T^B \bar{\mathbf{v}}_g$ . Notice that only the direction of  ${}^B\mathbf{v}_g$  is taken into account for computing the feature-based velocity vector.<sup>1</sup> Without loss of generality, features in the image plane can be confined within the boundary of a circle centered at  $\mathbf{p}_o$ , as shown in Fig. 4. Let the central angle  $\hat{\theta}_{cs}$  define a circular segment in the circle and  $\theta_{cs} = \pi - \hat{\theta}_{cs}/2$ . A charge is represented as the tuple  $Q_i = (\hat{\mathbf{p}}_i, q_i, \theta_{cs})$ , where  $q_i \in [0, 1]$  is its corresponding potential energy. The charging

<sup>1</sup>The last coordinate of its homogeneous form must be set to 0.

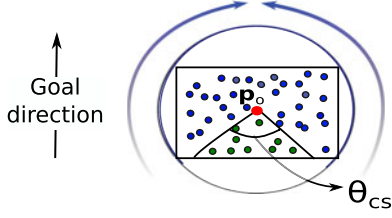


Fig. 4. Attractive potential energy increase in the direction of the blue arrow. Charges are neutral within the circle segment defined by  $\theta_{cs}$ . Attractive and neutral charges are represented by blue and green dots, respectively.

policy is defined below:

$$q_i = \begin{cases} 1 - \frac{\theta_i}{\theta_{cs}}, & \text{if } \theta_i \leq \theta_{cs} \\ 0, & \text{if } \theta_i > \theta_{cs} \end{cases} \quad (3)$$

Based on the current frame information, the considered charging policy associates high attractive potential energy to features localized in the goal direction. Features localized away from the goal direction, on the region defined by the circle segment, have a neutral charge.

### B. Vector Field

Each charge  $Q_i \in \mathcal{Q}$  exerts a force  $\mathbf{f}_i$  at a point  $\mathbf{p}_c$  that belongs to the image frame, given by

$$\mathbf{f}_i = \begin{cases} (0, 0)^T, & \text{if } d_i < r \\ \frac{(d_i - r)}{s} q_i \begin{pmatrix} \cos(\phi_i) \\ \sin(\phi_i) \end{pmatrix}^T, & \text{if } r \leq d_i \leq s + r \\ q_i \begin{pmatrix} \cos(\phi_i) \\ \sin(\phi_i) \end{pmatrix}^T, & \text{if } d_i > s + r \end{cases} \quad (4)$$

where  $r$  is the distance in pixels that a charge must be from the evaluated point to exert any force on it;  $s$  is the spread in pixels of the potential field, and  $d_i$  and  $\phi_i$  are computed as follow

$$d_i = \|\mathbf{p}_i - \mathbf{p}_c\| \quad (5)$$

$$\phi_i = \angle(\mathbf{p}_i, \mathbf{p}_c). \quad (6)$$

The total force  $\mathbf{f}$  on the point  $\mathbf{p}_c$  can be computed as  $\mathbf{f} = \sum_i \mathbf{f}_i$ , which can be normalized and transformed into a feature based velocity command  $\bar{\mathbf{v}}_f$ . Its direction in homogeneous coordinates is given as

$$\bar{\mathbf{v}}_f = \frac{1}{\|\mathbf{f}\|} \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}. \quad (7)$$

Finally, the proposed reference velocity takes the form

$$\bar{\mathbf{v}} = \lambda \bar{\mathbf{v}}_g + (1 - \lambda) \bar{\mathbf{v}}_f, \quad (8)$$

where  $\lambda$  is a weight factor and  $\bar{\mathbf{v}}_g$  is a normalized velocity. The velocity  $\bar{\mathbf{v}}$  can be transformed from the image frame to the body frame and scaled accordingly.

### C. Discussion

Artificial potential fields frameworks usually take into consideration the robot position when computing forces - obstacles exert repulsive force and the goal an attractive force. In the proposed framework, features can be either attractive or neutral

accordingly to their position w.r.t. the point  $\mathbf{p}_c$  being considered and the goal direction.

A case study is illustrated in Fig. 5. In particular, Fig. 5(a) shows a frame and extracted features classified as inliers. The goal velocity  $\mathbf{v}_g$  is directed towards the top-right pixel in the image frame - a poor zone regarding the number of features. Fig. 5(b) depicts the potential energy associate with each feature when evaluating the action induced at the central pixel of the camera,  $\mathbf{p}_o$ . The potential field map [see Fig. 5(c)] shows the corresponding field for different values of  $\mathbf{p}_c$ . For each point in the map, (1)–(8) must be computed. However, for visualization,  $\mathbf{v}_f$  is not normalized. Observe that the map can be classified in two different zones, which are labelled as *goal-friendly* and *feature-friendly* actuation zones. Both regions are shown in the potential field map in the yellow (goal-friendly) and the red (feature-friendly) background. Suppose  $\lambda = 0$ ; then, according to (8) the vehicle follows  $\mathbf{v}_f$ . If  $\mathbf{p}_c$  is within a feature-friendly region, the vehicle favors more the features than the goal. As a matter of fact, the vehicle will move away from the goal. On the contrary, if  $\mathbf{p}_c$  is within a goal-friendly region, the vehicle will move towards the goal. The absolute value of the angle between the goal velocity and the feature based velocity determines these regions – an acute angle indicates a goal-friendly zone and an obtuse angle is related to a feature-friendly zone. The radius of the charge  $r$  determines whether features close to the point  $\mathbf{p}_c$  affect the solution or not. Meanwhile, the spread  $s$  determines the strength of each charge. The larger the spread is, the more influence the charges away from the point evaluated will have on the feature driven action. Thus, it limits the prediction horizon based on the local frame, that is, the belief that a feature on an edge indicates that there will be more features on that direction.

### D. Auto-Tuning

The value of  $\lambda$  determines whether the vehicle shall follow the feature-based velocity or the goal velocity. The main motivation to follow the latter rather than the former is the possibility of localization failure. Therefore, the value of  $\lambda$  shall be selected considering a given performance index of the localization algorithm being employed. In this letter, the number of inlier features tracked across consecutive frames ( $N_f$ ) is considered. If the number of tracked features is above an upper threshold ( $T_{fMAX}$ ),  $\lambda$  is set to  $\lambda_{MAX}$ . Conversely, if the number of tracked features is below a lower threshold ( $T_{fMIN}$ ),  $\lambda$  is set to  $\lambda_{MIN}$ . If the number of tracked features is in between these two thresholds, then  $\lambda$  is proportional to the number of features. The auto-tuning method is summarized as

$$\lambda(N_f) = \begin{cases} \lambda_{MIN}, & \text{if } N_f < T_{fMIN} \\ \alpha N_f + \beta, & \text{if } T_{fMIN} \leq N_f \leq T_{fMAX} \\ \lambda_{MAX}, & \text{if } N_f > T_{fMAX} \end{cases} \quad (9)$$

where  $\alpha$  and  $\beta$  are constants such that  $\lambda(N_f)$  is continuous.

## V. RESULTS

This section presents simulation results and real world experiments for multiple flight tests. For both simulation and real world experiments, images and commands are published and received within the ROS (Robot Operating System) environ-



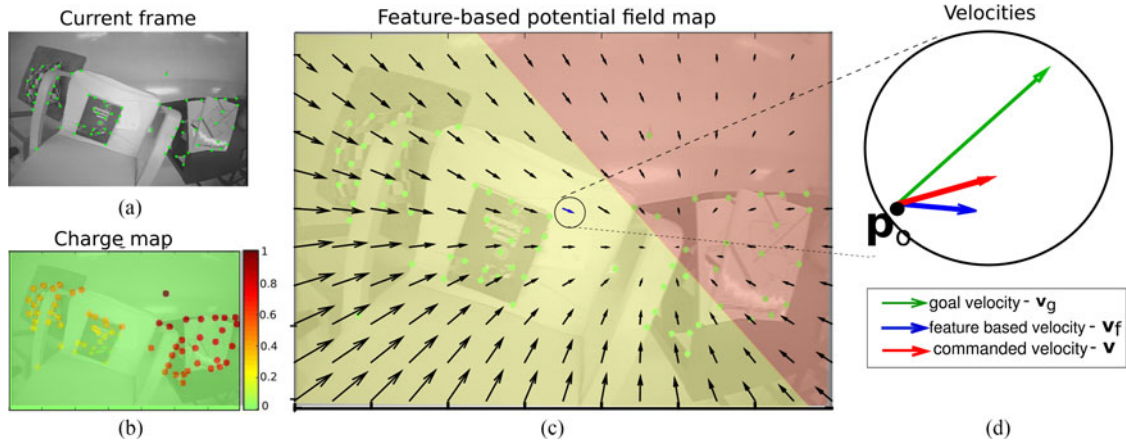


Fig. 5. Case study for the feature based potential field. Current image frame and inlier features (a). Considering  $\mathbf{p}_c$  as the optical center, the charge map is built and represented as a heat-map. (b). The potential field map shows the action derived when evaluating at different  $\mathbf{p}_c$  (c). Goal-friendly region is shown in yellow and feature-friendly in red (more detail about these regions is shown in Section IV-C). (d) shows commanded reference velocity as the combination of goal and feature velocity. ( $\mathbf{p}_c = \mathbf{p}_o$ ,  $\lambda = 0.1$ ,  $\theta_{CS} = 10^\circ$ ,  $d = 50$  pixel,  $s = 150$  pixel).

ment [23]. The feature-based potential field runs online as the decision making process, sending desired velocity commands to the velocity controller. A first order low pass filter, with a cutoff frequency at 20 Hz, provides smooth control reference for the feature driven action. When a new image frame is received, Shi-Tomasi features [24] are extracted. In addition to a minimum quality threshold, only 100 features with the highest response are selected. Then, features are tracked across two consecutive frames using the Lucas-Kanade Tracker (LKT) [25]. By resorting to a RANSAC [26] framework, the 8-point algorithm classifies features as inliers or outliers. The solution is robust in the presence of few false inliers, such that a matching step is unnecessary. Both Shi-Tomasi and LKT are implemented in the OpenCV library [27].

#### A. Simulation

The proposed solution was validated in different scenarios using the simulator V-REP by Coppelia Robotics. Fig. 6(a)–(c) illustrate scenarios for which flying in a straight path to the goal is potentially risky for localization and map consistency. Therefore, in such scenarios, the proposed method is an asset, as it drives the vehicle through feature rich zones. Fig. 6(d) considers a well textured scene. Although the potential-field framework does not benefit the navigation, it does not compromise reaching the goal. For 10 trials, the trajectory was  $0.9 \pm 0.2$  m longer than flying in a straight line, which corresponds to 9% of the 10 m straight path length. Fig. 6(e)–(f) correspond to local-minima scenarios. The proposed method struggles to overcome a symmetric bifurcation [see Fig. 6(e)]. Such scenario with equally distributed features in opposite direction is not likely to happen in a real world environment. Finally, Fig. 6(f) presents a scenario for which the method may eventually fail due to the local nature of potential field based solutions. Analyzing the simulations, if the goal velocity and the feature based velocity are pointing in opposite directions, the vehicle has reached a local minimum. This problem could be addressed by a high-level SLAM, which memorizes paths already travelled and selects an alternative path to the goal. If SLAM is not available – due

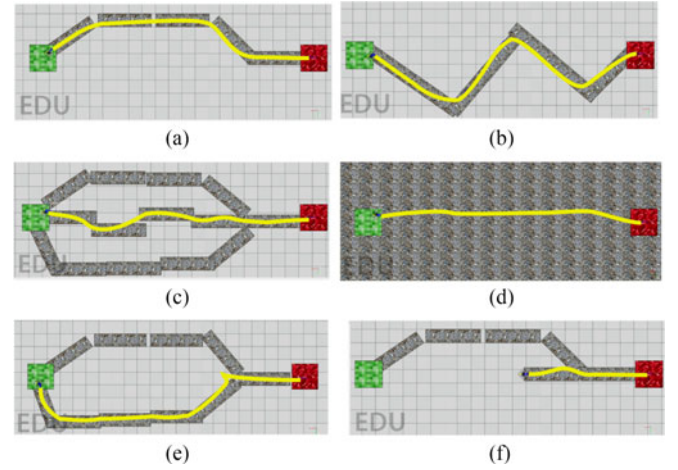


Fig. 6. Simulations for evaluation of the proposed algorithm. The task consists in flying a 10 m length path from the red square (right) to the green square (left). (a)–(e) illustrate scenarios that have one or more feature-rich paths to the goal and (f) depicts a scenario where the proposed algorithm fails due to a local minimum. Parameters were kept constant:  $\mathbf{p}_c = \mathbf{p}_o$ ,  $\lambda = 0.4$ ,  $\theta_{CS} = 60^\circ$ ,  $d = 50$  pixel,  $s = 150$  pixel.

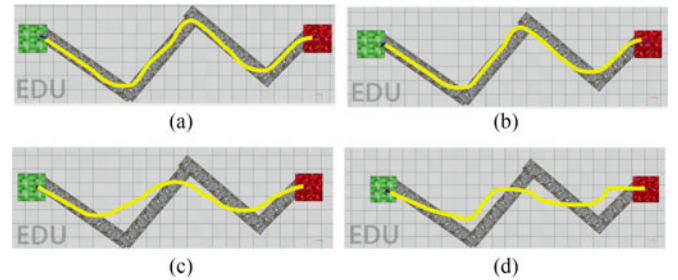


Fig. 7. Effect of parameter  $\lambda$ : (a)  $\lambda = 0.3$ , (b)  $\lambda = 0.5$ , (c)  $\lambda = 0.6$  and (d) auto-tuning, on the experimental setup of Fig. 6(b).

memory or computational power constraints – increasing the value of  $\lambda$  would enable to overcome the local minimum.

Fig. 7 addresses the effect of varying  $\lambda$ . If  $\lambda$  is low [see Fig. 7(a)] the vehicle takes a path more conservative with respect to features. In contrast, as  $\lambda$  increases, the vehicle follows a trajectory closer to the straight line that links the start and

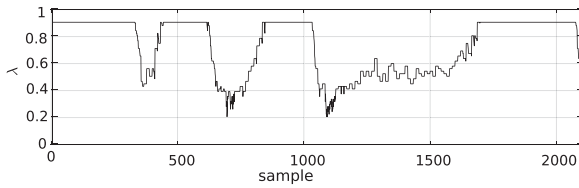


Fig. 8. Value of parameter  $\lambda$  using auto-tuning technique. Path travelled is as shown in Fig. 7(d).

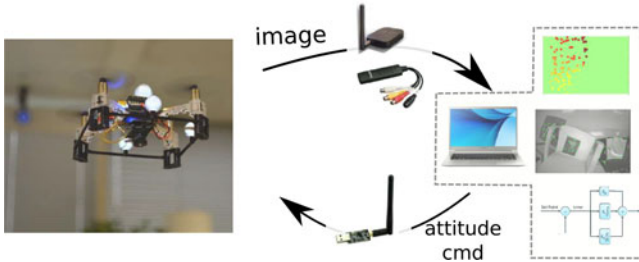


Fig. 9. Experimental setup. Clockwise, starting from the top: the crazyflie equipped camera module FX798T; image receiver, and capture card; a notebook that runs the code; and an antenna for the communication with the vehicle.

goal positions [see Fig. 7(c)]. An evaluation of the auto-tuning technique described in Section IV-D is shown in Fig. 7(d).  $T_{f_{MAX}}$  and  $T_{f_{MIN}}$  were set to 90% and 25% of the maximum number of features extracted per frame, respectively. The value of  $\lambda$  throughout the simulation of Fig. 7(d) can be seen in Fig. 8. Online tuning had a positive effect on the overall performance of the task. In Fig. 7(a) the vehicle travelled 12.05 m to the goal, while using the auto-tuning technique, the path travelled was 11.1 m length.

A total of 30 simulations using the scenario depicted in Fig. 6 were performed. For half of these simulations, the potential field framework was turned off. For the other half, it was turned on. In the first situation, vision-based localization failed after flying a fourth of the the straight line path. Afterwards, within an EKF framework, state estimation was purely based on inertial measurements propagation. IMU covariance error was extracted from the bagfile described in [6]. Meanwhile, the proposed method ensured visual localization worked properly during the entire flight. For both methodologies, based on its state estimation, the vehicle eventually reached the estimated goal position. For each trial, the distance between the final position vehicle and the real goal position was measured. While when using the proposed active pipeline our method reached a distance of  $0.85 \pm 0.48$  m, the passive technique got  $2.33 \pm 1.02$  m, showing that the proposed technique improves the overall location estimation.

### B. Experimental Setup With Real Data

Fig. 9 presents the main components employed in the experimental validation of the proposed method. The algorithm was tested in the mini-quadrotor Crazyflie 2.0,<sup>2</sup> manufactured by BitCraze, equipped with accelerometers, gyroscopes, magnetometer, pressure sensor, bluetooth, and a low-latency/long-

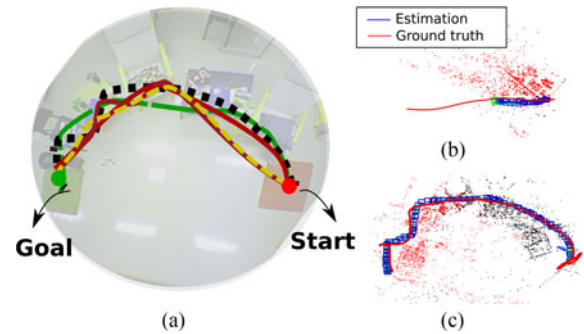


Fig. 10. Fig. (a) shows the first real scenario for evaluation of proposed algorithm. Fig. (b) shows that the localization fails in the absence of features (passive technique). In Fig. (c), we show results with our method, i.e., a feature based velocity vector drives the vehicle to the goal through a region rich in features.

range radio that allows for the robot to be operated (and log data) from a remote PC. The manufacturer's radio - Crazyradio - interfaces communication with a notebook, which computes the proposed action - collective thrust and attitude commands [28]. We attached to the vehicle a 4.7 g mini transmitter camera module FX798T equipped with  $120^\circ$  field of view lens - the camera faces downward. The frame-rate is 30 Hz for a  $720 \times 480$  image resolution. The proposed algorithm runs off-board. The images are transmitted via radio. Although variable, the measured delay upon image reception on the remote computer is about 150 ms. The flying arena is equipped with an Optitrack<sup>3</sup> motion capture system, which estimates the pose of the vehicle with millimetric precision. The motion capture system provides feedback for the control loop, while the feature-based potential field works online as the decision making process, sending desired velocity commands to the controller.

### C. Real Data Experiments Evaluation

Multiple flights were performed in a  $10 \text{ m} \times 6 \text{ m}$  home alike arena employed in European Robotics League (ERL).<sup>4</sup> In the first scenario, shown in Fig. 10(a), the robot was autonomously controlled to fly from the starting position to the goal position as indicated by the two carpets. The straight path that links both carpets is a regular floor that does not contain good visual cues. The localization task for a trial is said to *succeed* if ORB-SLAM [7] manages to keep track of the pose of the vehicle and, therefore, a consistent map is built. A similar qualitative metric is employed in [15]. We stress again that one of the advantages of the proposed technique is that it can work under a SLAM framework, rather than on top (as most previous active perception strategies), ensuring a better localization and mapping result. Notice that we run ORB-SLAM offline, just for evaluation purposes, not for online localization. Multiple trials showed that taking the straight path towards the goal always results in localization failure due to the lack of visual features. Fig. 10(b) shows one of the trials where the localization estimation fails

<sup>3</sup><http://optitrack.com/>

<sup>4</sup>[https://www.eu-robotics.net/robotics\\_league/erl-service/certified-test-beds/certified-test-beds.html](https://www.eu-robotics.net/robotics_league/erl-service/certified-test-beds/certified-test-beds.html)

<sup>2</sup><https://www.bitcraze.io/crazyflie-2/>

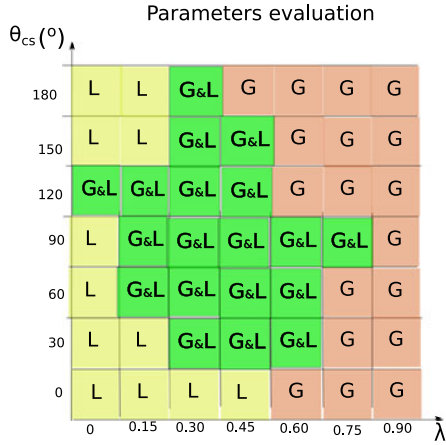


Fig. 11. Algorithm evaluation for different values of  $\lambda$  and  $\hat{\theta}_{cs}$ . It was considered the experimental results presented in Fig. 10(a). *L* means localization succeeded, *G* means goal task succeeded, and *G&L* means both task succeed.

when a straight path is taken. Alternatively, using the proposed active method, the robot was always capable of reaching the goal while maintaining localization. Fig. 10(c) shows the path the robot takes, as well as the estimated state throughout the time in one of the experiments. Fig. 10(c) also shows the trajectory travelled for different trials with the proposed method where the robot reaches the goal while keeping track of its location.

For the same scenario, the method was also evaluated for different values of  $\lambda$  and  $\hat{\theta}_{cs}$ , while keeping  $\mathbf{p}_c = \mathbf{p}_o$  constant. The goal task is said to *succeed* if the vehicle reaches the goal within less than 0.15 m. Localization task success is defined as before. Results are shown in Fig. 11, where *L* means localization task succeed, *G* means the goal task succeed and *G&L* means both task succeed. The localization task fails when the weight of the goal velocity vector is large, i.e.,  $\lambda$  is large, such that the vehicle takes a path similar to the straight one. Meanwhile, the goal task fails when the feature based potential field falls in a local minimum. Analyzing the image sequence, it is possible to observe that the vehicle hovers over a region that new features are not initialized and features contrary to the goal direction exert some attractive force. Consequently, the potential field force is not able to push the vehicle in the direction of the goal. As the value of  $\hat{\theta}_{cs}$  increases, the attractive potential energy associated to features opposing the goal direction decreases. Therefore, the vehicle is no longer pulled backwards w.r.t. to the goal and reaches the final destination.

To further evaluate the method in more realistic environments, we performed several long flights using all rooms of the home alike arena (Fig. 12). The vehicle must visit waypoints (WP) selected by a human operator to explore different rooms of the apartment. As shown, the proposed method deviates the vehicle from straight paths that link consecutive waypoints. The VO performance is depicted in Fig. 13 with (active mode) or without (passive mode) the proposed low level algorithm. The VO is obtained from ORB-SLAM, which requires proper 3D point initialization to propagate the relative scale. As long as enough visual cues are within the field of view of the camera, the error is small for both active and passive method. However,

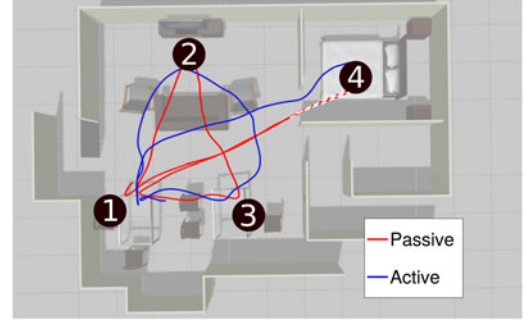


Fig. 12. Passive and active flight on a home alike arena as employed in ERL. The vehicle visits waypoints in the order: 1, 2, 3, 1, and 4.

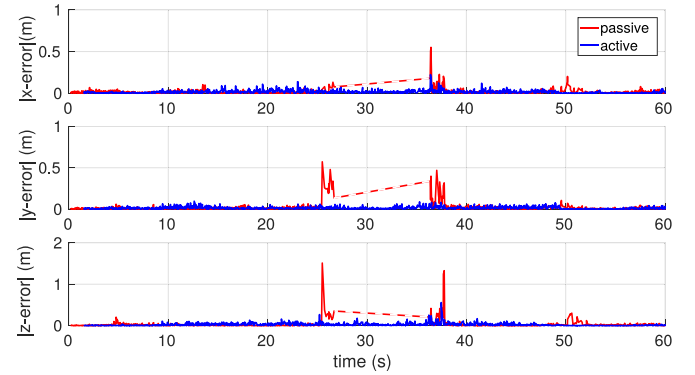


Fig. 13. Comparison of the visual odometry error (consecutive frames) for both the passive and active techniques using ORB-SLAM, for the experimental setup presented in Fig. 12. The dash red line indicates that the robot is lost.

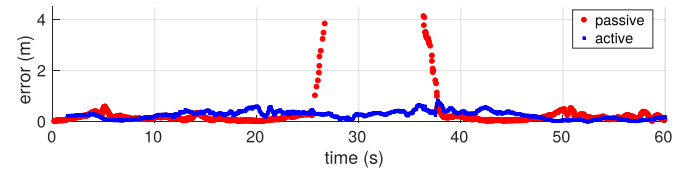


Fig. 14. Curves representing the SLAM estimated position error, for both passive and active techniques while doing the path presented in Fig. 12. Points in the graph represent the interval of estimation.

flying straight to the goal leads to poor feature regions closed to the bedroom entrance at  $t = 25$  s, where the passive estimator gradually fails to estimate the position. Before completely failing (dashed segment) visual odometry degrades quickly. The SLAM localization error (Fig. 14) presents a similar behaviour - error grows large before failing, but it is kept small when the map is concise. On the way out of the bedroom, the vehicle revisits some poorly initialized features leading to large errors during the recovering phase. If feedback was not provided by the motion capture system, such pose estimation failure would require to abort the mission and land in an open loop manner. In particular, for the active method flight tests, the value of  $\lambda$  was computed using the auto-tuning method and it is shown in Fig. 15. Overall, we performed about 150 meters of flight per method, corresponding to more than 8000 frames. With the proposed strategy, SLAM manages to estimate the pose for 99.1%



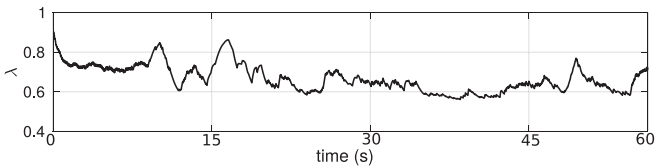


Fig. 15. Representation of the  $\lambda$ 's auto-tuning of the method presented in this letter, while making the path shown in Fig. 12 (blue curve).

of the frames whereas only for 90.2% was estimated using passive navigation. Therefore, thanks to the feature based strategy, VO and SLAM error are more likely to be kept small throughout the entire flight.

## VI. CONCLUSION

This letter proposes a simple, fast, low complex and map-independent solution for the active visual localization problem. The method, based on artificial potential fields, associates a potential energy to features that are classified as inliers in the current image frame. The goal direction is used to determine the corresponding charge intensity of these image features. Simulations and experimental results using a micro aerial vehicle, equipped with a downward looking camera, showed that the method can effectively drive the vehicle towards the goal, while avoiding no or poor featured regions. This active behaviour can greatly improve the localization performance and prevent common localization failures that are caused by low quality image features. The proposed active solution does not rely on a map and hence can be integrated within a SLAM framework to improve the accuracy and robustness of localization and mapping. Finally, the proposed method could benefit from the many techniques developed for potential fields, such random perturbations for escaping a local minima.

## REFERENCES

- [1] G. Loianno, M. Watterson, and V. Kumar, "Visual inertial odometry for quadrotors on SE(3)," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1544–1551.
- [2] S. Weiss *et al.*, "Monocular vision for long-term micro aerial vehicle state estimation: A compendium," *J. Field Robot.*, vol. 30, no. 5, pp. 803–831, 2013.
- [3] J. Engel, J. Sturm, and D. Cremers, "Accurate figure flying with a quadcopter using onboard visual and inertial sensing," in *Proc. Workshop Vis. Control Mobile Robots*, 2012.
- [4] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. IEEE/ACM Int. Symp. Mixed Augmented Real.*, 2007, pp. 225–234.
- [5] D. Scaramuzza and F. Fraundorfer, "Visual odometry [Tutorial]," *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011.
- [6] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 15–22.
- [7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardas, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
- [8] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," in *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.
- [9] R. T. Rodrigues, M. Basiri, A. P. Aguiar, and P. Miraldo, "Feature based potential field for low-level active visual navigation," *Robot'17: 3rd Iberian Robot. Conf., Adv. Robot.*, 2017, pp. 791–800.
- [10] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 865–880, Jul. 2002.
- [11] T. A. Vidal-Calleja, A. Sanfeliu, and J. Andrade-Cetto, "Action selection for single-camera SLAM," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 40, no. 6, pp. 1567–1581, Dec. 2010.
- [12] M. Bryson and S. Sukkarieh, "Observability analysis and active control for airborne SLAM," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 1, pp. 261–280, Jan. 2008.
- [13] M. W. Achtelik, S. Lynen, S. Weiss, M. Chli, and R. Siegwart, "Motion- and uncertainty-aware path planning for micro aerial vehicles," *J. Field Robot.*, vol. 31, no. 4, pp. 676–698, 2014.
- [14] G. Costante, J. Delmerico, M. Werlberger, P. Valigi, and D. Scaramuzza, "Exploiting photometric information for planning under uncertainty," in *Proc. Adv. Robot., Robotics Research*, Springer, vol. 2., pp. 107–124, 20178.
- [15] S. A. Sadat, K. Chutskoff, D. Jungic, J. Wawerla, and R. Vaughan, "Feature-rich path planning for robust navigation of MAVs with mono-SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 3870–3875.
- [16] C. Mostegel, A. Wendel, and H. Bischof, "Active monocular localization: Towards autonomous monocular exploration for multirotor MAVs," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 3848–3855.
- [17] O. Khatib, "The potential field approach and operational space formulation in robot control," *Adapt. Learn. Syst., Theory Appl.*, pp. 367–377, 1986.
- [18] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.*, vol. 8, no. 5, pp. 501–518, Oct. 1992.
- [19] J. Vallv and J. Andrade-Cetto, "Potential information fields for mobile robot exploration," *Robot. Auton. Syst.*, vol. 15, pp. 68–79, 2015.
- [20] V. A. M. Jorge *et al.*, "Ouroboros: Using potential field in unexplored regions to close loops," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 2125–2131.
- [21] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, Bradford Co., Scituate, MA, USA, 2004.
- [22] R. I. Hartley and A. Zisserman, "Multiple view geometry in computer vision," Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [23] M. Quigley *et al.*, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, 2009.
- [24] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 593–600.
- [25] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. 7th Int. Joint Conf. Artif. Intell.*, vol. 2, pp. 674–679, 1981.
- [26] M. A. Fischler and R. C. Bolle, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [27] G. Bradski and A. Kaehler, "Learning OpenCV: Computer vision with the OpenCV library," Sebastopol, CA, USA: O'Reilly Media, 2008.
- [28] W. Hoenig, C. Milanes, L. Scaria, T. Phan, M. Bolas, and N. Ayanian, "Mixed reality for robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 5382–5387.