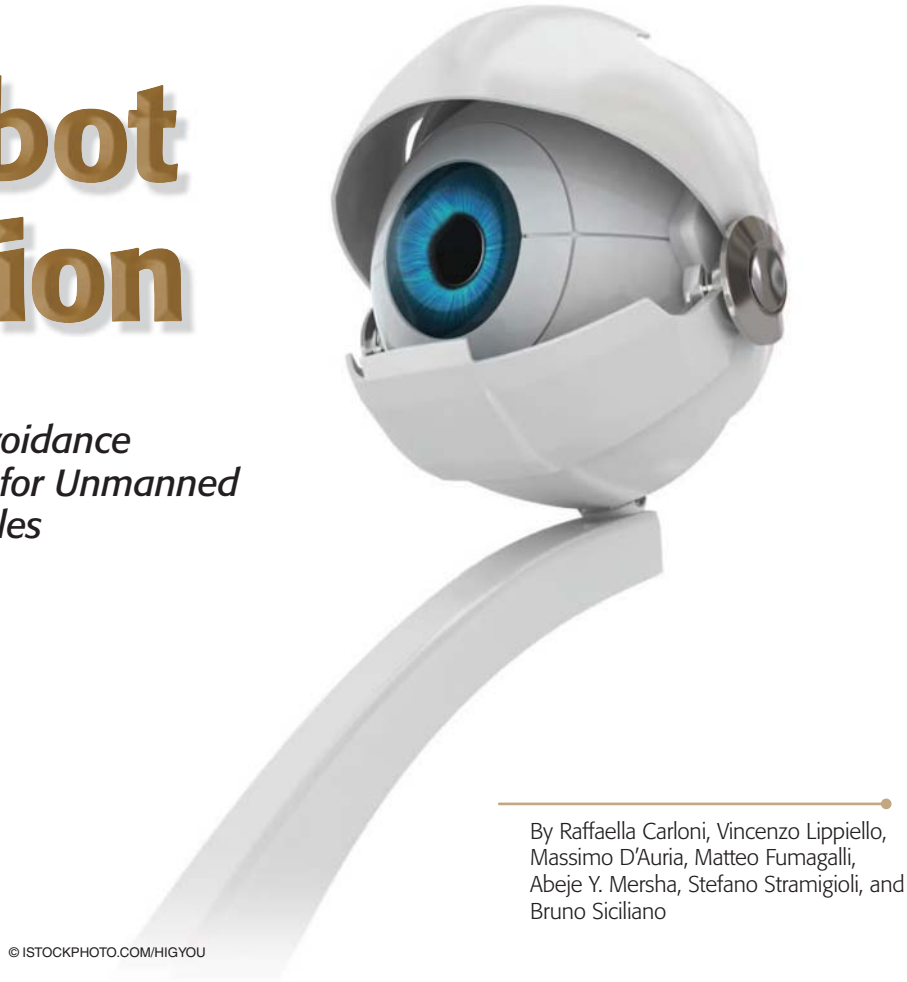
Robot Vision

Obstacle-Avoidance Techniques for Unmanned Aerial Vehicles



By Raffaella Carloni, Vincenzo Lippiello, Massimo D'Auria, Matteo Fumagalli, Abeje Y. Mersha, Stefano Stramigioli, and Bruno Siciliano

In this article, a vision-based technique for obstacle avoidance and target identification is combined with haptic feedback to develop a new teleoperated navigation system for underactuated aerial vehicles in unknown environments. A three-dimensional (3-D) map of the surrounding environment is built by matching the keypoints among several images, which are acquired by an onboard camera and stored in a buffer together with the corresponding estimated odometry. Hence, based on the 3-D map, a visual identification algorithm is employed to localize both obstacles and the desired target to build a virtual field accordingly. A bilateral control system has been developed such that an operator can safely navigate in an unknown environment and perceive it by means of a vision-based haptic force-feedback device. Experimental tests in an indoor environment verify the effectiveness of the proposed teleoperated control.

Digital Object Identifier 10.1109/MRA.2013.2283632
Date of publication: 31 October 2013

Vision-Based Obstacle-Avoidance Techniques

Interest in unmanned aerial vehicles (UAVs) has increased due to the wide range of their application fields, which include surveillance, rescue, and inspection. So far, UAVs have mainly been used outdoors with the support of a global positioning system (GPS) for navigation purposes. However, when UAVs are flying indoors in an unknown and unstructured environment, GPS information will not be available. Therefore, in such situations, different navigation and obstacle-avoidance techniques have been investigated using onboard sensors such as lasers, sonars, cameras, radars, and inertial measurement units (IMUs) that give a perception of the environment. Obstacle avoidance is a core issue since any autonomous navigation system must preserve the safety of both the UAV and the surrounding environment.

Several approaches can be found to address this problem. In [1] and [2], radar-based navigation and obstacle avoidance are implemented, while a laser range finder for obstacle detection is employed in [3]. The main drawbacks are the high power consumption and weight of these sensors.

1070-9932/13/$31.00©2013IEEE

A camera is a lightweight sensor with low power consumption. Different vision-based obstacle-avoidance techniques are implemented in this article, most of which are based on the biologically inspired concept of optical flow. In [4], fruit flies avoid obstacles when turning away from a region with high optical flow, while in [5], honeybees flying through a tunnel try to balance the amount of lateral optical flow to keep the same distance from the walls. By exploiting this concept, a fisheye camera pointing downwards is employed in [6]–[8] to get image frames and to compute the lateral optical flow: by using this information, a depth map of a corridor is built to control the UAV along a trajectory centered in the middle of the observed obstacles. In [9], two optical flow controllers for a fully actuated UAV that flies in confined indoor and outdoor environments are used: the first controller regulates the UAV's forward thrust, and the second one controls the UAV's side-slip thrust.

Two different approaches with and without optical flow are proposed in [10], and are based on the concept of the time-to-contact needed before the vehicle collides with an obstacle while the UAV is moving with a specific translational velocity. A triangulation-based method for the computation of the distances of detected objects is shown in [11], where a stereo camera system is used. Although stereo images benefit in terms of accuracy of the estimation and odometry of the obstacles position, a high payload and computational power is required from the UAV. On the other hand, the main drawback in the use of a monocamera system is the need of a translational movement of the UAV to detect the surrounding obstacles. In [12], a single-camera front-obstacles collision-avoidance approach for a microflyer is presented based on the divergence of the optical flow on the left and right side of the direction of travel.

In this article, two main problems are addressed: 1) a monocamera vision-based obstacle avoidance for a safe navigation of a bilateral teleoperated UAV by means of a haptic device that endows the human operator with a vision-based haptic feedback of the environment and 2) an image target identification system to generate a force field that leads the human operator to safely reach the target. This new approach makes use of a single onboard calibrated camera together with a monocamera odometry estimation system based on Parallel Tracking and Mapping (PTAM) [13]. The images, grabbed by the onboard camera, are stored in a buffer together with the corresponding UAV position estimation, i.e., the UAV odometry. Then, robust image keypoints are extracted and matched between image pairs by means of a Pyramidal Lukas-Kanade algorithm [14], [15]. Moreover, a Random Sample Consensus (RANSAC)-based algorithm [16] is adopted for outlier rejection [17]. A new statistically robust triangulation-based approach is then proposed to estimate the corresponding 3-D points, which are used to build a 3-D occupancy grid map of the obstacles in the surrounding environment.

The map is updated by adding the data provided by a target identification algorithm, based on an elaboration flow similar to the obstacle localization method. Based on this map, a system of repulsive forces from obstacles and attractive forces to the target is built. This force is then applied to generate a feedback force on the haptic device, provide the human operator with a perception of the environment, and generate an additional position reference for the UAV controller. With this new navigation support system, the human operator is able to steer the UAV to the identified target while avoiding obstacles along its trajectory. In the experimental validation, the UAV pose (position and orientation) is stabilized by means of an external motion capture system to achieve higher robustness.

System Overview

In this section, the main functionalities of the developed teleoperation system are described, as depicted in Figure 1.

The Haptic Interface

The human operator interacts with a haptic device to give a reference command to the UAV. At the same time, the operator can feel a force feedback, which provides a haptic feeling of the identified environment, i.e., how close the UAV is to the localized obstacles and target.

The Ground Station

In the proposed framework, the ground station has to: 1) send the desired reference to the onboard controller, 2) generate the force feedback to be displayed by the haptic device, 3) estimate the UAV's odometry by using the images acquired by the onboard camera, and 4) process the images to build a local map of the environment. Based on the map and the estimated state of the UAV, a virtual interaction force applied to the UAV is built when obstacles and/or the target are detected. This force arises when either an obstacle enters in a

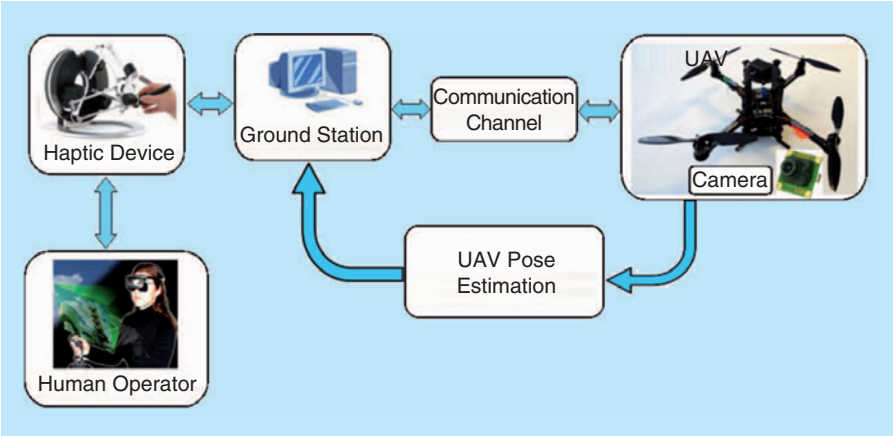


Figure 1. A block diagram of the overall teleoperated control system.

safety volume around the vehicle or when the target is in the field of view of the vehicle. In particular, this force depends on the distance between the UAV and the obstacles/target.

Pose Estimation

The proposed approach requires the pose estimation of the UAV flying in a cluttered unknown environment. Several approaches have been developed to estimate the UAV pose using one onboard camera. In [18] and [19], the pose is estimated by fusing data extracted from a camera flow and IMU measurements. A square root unscented Kalman filter is used in [20] to solve the simultaneous localization and mapping (SLAM) problem for a single camera. In [21], a real-time monocular SLAM is implemented that uses an extended Kalman filter and a straight-line detector.

In the proposed scheme, the odometry of the UAV is evaluated on the basis of an onboard monocamera video streaming, which is the input to a module based on PTAM [13], originally developed as a camera tracking system for augmented reality. It requires no markers, premade maps, known templates, or inertial sensors. The low-level egomotion estimation provided by PTAM and the available IMU measurements have been suitably combined with an extended Kalman filter to develop an effective odometry estimation module. Figure 2 illustrates the odometry of a test path, where the initial and the final pose of the vehicle are coincident. The total length of the considered path is 11.3 m, while the norm of the residual positional error is 2.8 cm.

In the indoor experiments, as described in the "Experiments" section, an external tracker has been used to enhance the stabilization of the pose of the UAV.

3-D Environment Map

The visual measurements together with the vehicle odometry are employed for the online estimation of a 3-D occupancy map of the surrounding environment. Simultaneously, an image-target identification algorithm runs to localize the target, which could be present in the observed scene, in the map. In the following subsections, the employed camera model and obstacle localization algorithm, as well as the 3-D occupancy map construction procedure, are presented.

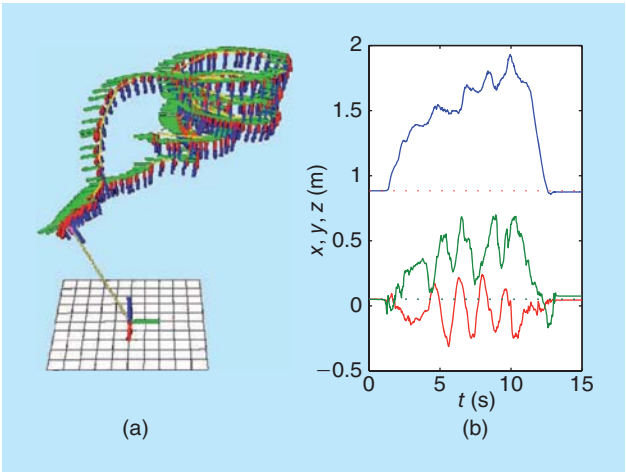


Figure 2. The vehicle trajectory estimated by the proposed PTAM-based module of a spiral-like path of 11.3-m length (the initial pose is coincident with the final pose). (a) A 3-D representation of the estimated vehicle odometry. (b) The estimated positional trajectory (*x*-component in red, *y*-component in green, and *z*-component in blue) with respect to the camera frame, which results in a norm of the final residual positional error of 2.8 cm.

Camera Model

In this article, the well-known pinhole camera model is employed [22]. The perspective relation between the 3-D coordinate of a point $\boldsymbol{p} = [p_x \;\; p_y \;\; p_z]^T$ expressed in the world reference frame and its 2-D projection on the image plane of the camera, expressed in pixel coordinate, is given by

$$\lambda \begin{bmatrix} X_f \\ Y_f \\ 1 \end{bmatrix} = \boldsymbol{\Omega} \, [\boldsymbol{I}_3 \;\; \boldsymbol{0}] \, \boldsymbol{T}_b^c \, \tilde{\boldsymbol{p}},$$

where $\lambda > 0$ is a projection scale factor, $\boldsymbol{I}$ is the identity matrix, $\tilde{\boldsymbol{p}} = [\boldsymbol{p}^T \;\; 1]^T$ is the homogeneous representation of $\boldsymbol{p}$, and $\boldsymbol{\Omega}$ is the camera calibration matrix, i.e., the camera intrinsic parameters, given by

$$\boldsymbol{\Omega} = \begin{bmatrix} f\alpha_x & 0 & X_0 \\ 0 & f\alpha_y & Y_0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $f\alpha_x$ and $f\alpha_y$ are the row and column focal length, respectively, and $[X_0 \;\; Y_0]^T$ is the position of the image center expressed in pixels. The matrix $\boldsymbol{T}_b^c$ represents the homogeneous transformation between the camera frame and the base frame $\boldsymbol{T}_b^c = \boldsymbol{T}_v^c (\boldsymbol{T}_v^b)^{-1}$, where $\boldsymbol{T}_v^c$ is the constant homogeneous transformation matrix between the camera frame and the UAV body frame, i.e., the camera extrinsic parameters, with the origin in the center of mass of the vehicle, and $\boldsymbol{T}_v^b$ is the current homogeneous transformation matrix between the body frame and the base frame, provided by the odometry estimation algorithm. More details can be found in [22].

Obstacle Localization

In the proposed approach, the estimation of the positions of the obstacles present in the observed scene is evaluated using information extracted by a single camera and the odometry. In detail, since only one camera is available, while at least two different views of an image feature are required, the image flow is continuously stored in a buffer. First, the grabbed images are synchronized temporally with the pose estimation of the vehicle, provided by the odometry estimation system. Then, only the most robust keypoints of the image are extracted and linked to the corresponding image-odometry pair. For this purpose, the well-known Shi-Tomasi corner detector [23] is employed, even if other keypoints extractors could also be used as well. Finally, this set of information, i.e., the *key-frames*, is stored in a finite-length buffer.

Hence, an image elaboration algorithm selects a pair of key-frames, grabbed from positions a sufficient distance from each

other so as to potentially guarantee a good spatial triangulation. Then, the second elaboration step consists of a matching process between the two sets of keypoints, performed using a Pyramidal Lucas-Kanade algorithm [14]. As often happens with image matching algorithms, some outlier might be present. To reduce this, a RANSAC-based outlier rejection algorithm on the epipolar constraint is performed [17]. Finally, the normalized image coordinates, corresponding to the matched keypoints, are computed. An example of this process on real images is shown in Figure 3.

By using the extracted pairs of normalized image points (the projections of the 3-D point $\boldsymbol{p}$, expressed in the camera frame, on the image planes of the cameras $\boldsymbol{P}_1 = [X_1 \;\; Y_1]^T$ and $\boldsymbol{P}_2 = [X_2 \;\; Y_2]^T$), a spatial triangulation problem is solved, thus estimating the position of the corresponding 3-D points $\boldsymbol{p}$ (see Figure 4). A robust least-squares solution can be found in [22], where the poses of the camera frames corresponding to the grabbed images are required.

The accuracy of the triangulation process increases if a large baseline between the two camera frames is employed, i.e., the normal distance between the two camera positions corresponds to the employed images. On the other hand, the accuracy of the vehicle odometry between camera poses degrades with the distance traveled by the UAV. Moreover, to succeed in the keypoints matching process, it is preferable to use images that have been grabbed from the closest possible pose. These demands are evidently in contrast and, therefore, a tradeoff should be implemented. On the basis of the performed experimental tests, by selecting images grabbed at distances in a range from 15–20 cm, a well-conditioned triangulation process can be achieved with a good matching rate and with an odometry drift that does not affect the accuracy of the overall process.

The criterion adopted for the image-pair selection from the buffer is as follows: the last image stored in the buffer is considered as the first key-frame of the pair; the second one is chosen in the key-frames buffer by considering the latest one grabbed at a distance greater than a suitable threshold from the first one. If no matches are found, a finite number of next images are considered until a right pair is found. If this process fails, the map is not updated during this step. Finally, the 3-D coordinates of points corresponding to the matched keypoints are computed. These points are considered as obstacles candidates, because they are not yet statistically robust. In the following section, "3-D Occupancy Map," the use of these candidates for the construction/update of the occupancy map is presented.

Figure 3. Examples of keypoints matching between images taken from different positions and stored into the buffer.

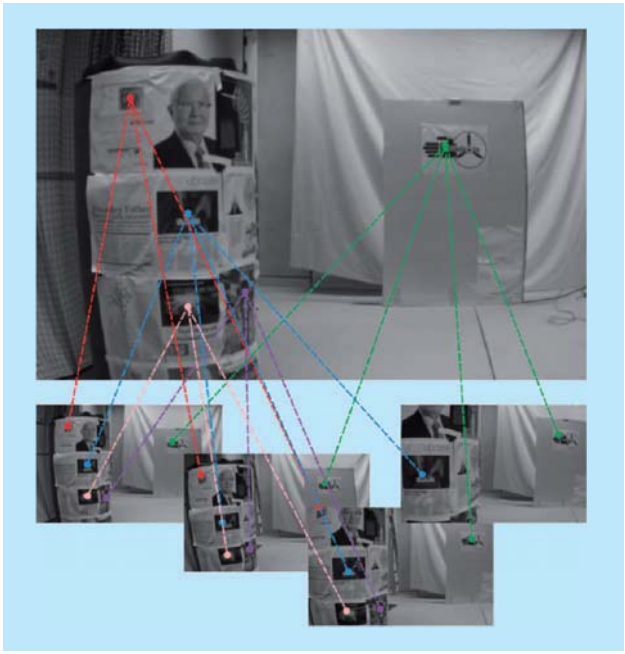


Figure 4. The triangulation process between keypoint optical rays evaluated from images extracted in different positions and stored into the buffer.

3-D Occupancy Map

A 3-D discrete occupancy map that is obtained by discretizing the vehicle's workspace with elementary cubes of equal size is considered. From a general point of view, the size of the cubes should be less than half the size of the vehicle along any direction to ensure the safety of the system. Obviously, smaller cubes increase the capability of the system to describe the environment, but the required memory increases as well, often without a real benefit for the navigation safety. Hence, a suitable tradeoff must be considered. For example, with respect to the environment of Figure 5 and a vehicle of $(50 \times 50 \times 20)$ cm size, cubes with 10-cm edge length have

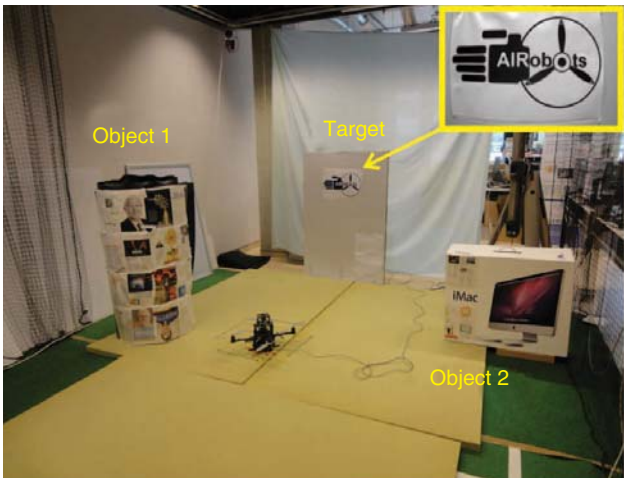


Figure 5. The workspace of the experimental tests.

been chosen, which corresponds to the smallest half-dimension of the vehicle.

Several information data are associated to each cube: the number of inliers (3-D triangulated points) had fallen into the cube volume, the last camera position from which an inlier has been found, and the state of the cube. In detail, the number of inliers represents the number of different points of view from which the same obstacle has been detected; the last camera position is required in case of hovering to prevent the same image features from generating several wrong inliers, while it is possible that the same outlier is achieved from different points of view; finally, the state of the cube is employed to generate the force field, as will be described in the next section, and it can be one of the following values: "FREE," "OCCUPIED," "OBSTACLE," "IGNORED," or "TARGET."

When a 3-D triangulated point is detected in the cube, then the value of the cube inliers is incremented and the state is set to "OCCUPIED."

Initially, each cube is set to "FREE." When a 3-D triangulated point is detected in the cube, then the value of the cube inliers is incremented and the state is set to "OCCUPIED." When the number of inliers reaches the assigned threshold, the state is set to "OBSTACLE." On the other hand, when a target is identified, the relative cube state is set to "TARGET." Moreover, since the camera can observe the target from each position that had generated a valid target viewpoint, all the cubes laying along these optical rays are set to "IGNORED." In this way, the map also implicitly provides a free way to reach the target. Figure 6 shows the reconstructed occupancy map corresponding to the environment of Figure 5.

For wide environments, a sparse representation of the occupancy grid map is considered together with a spatial/temporal vanishing criterion. This determines whether an occupancy cube is reliable or if it has to be discarded, based on the distance traveled by the vehicle or on the time interval spent after its last update. In fact, due to the drift of the vehicle pose estimation, obstacles that have been observed far from the current position or a long time cannot be considered reliable in the current map representation, and they must be refreshed. In this way, the reliability and scalability of the map representation can be tuned with respect to any environment.

In Figure 7, the environment map of an outdoor unknown space has been reconstructed by using only the onboard visual odometry estimation system and a sparse representation of the grid. In a particular point, three texturized objects have been positioned in the environment: the UAV has started from a position close to object number three and then it has been moved away by traveling from object number one to two. 3-D points matched at a distance more than 5 m from the current UAV position have been discarded during map construction to avoid noise due to the low resolution of the vision system at greater distance. Analogously, all the points that are close to the floor have been discarded to avoid unnecessary information.

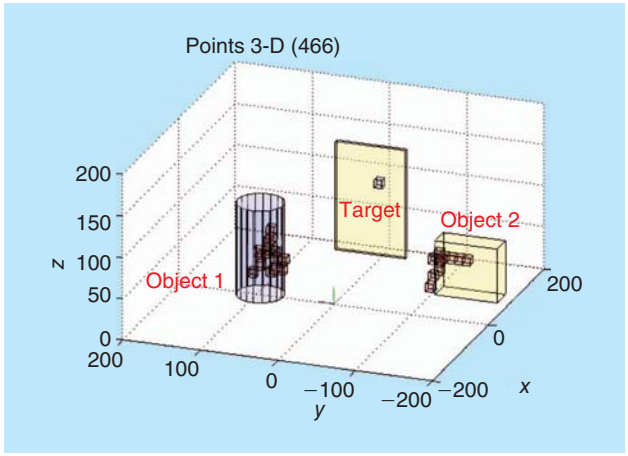


Figure 6. An example of a reconstructed map of the environment shown in Figure 5.

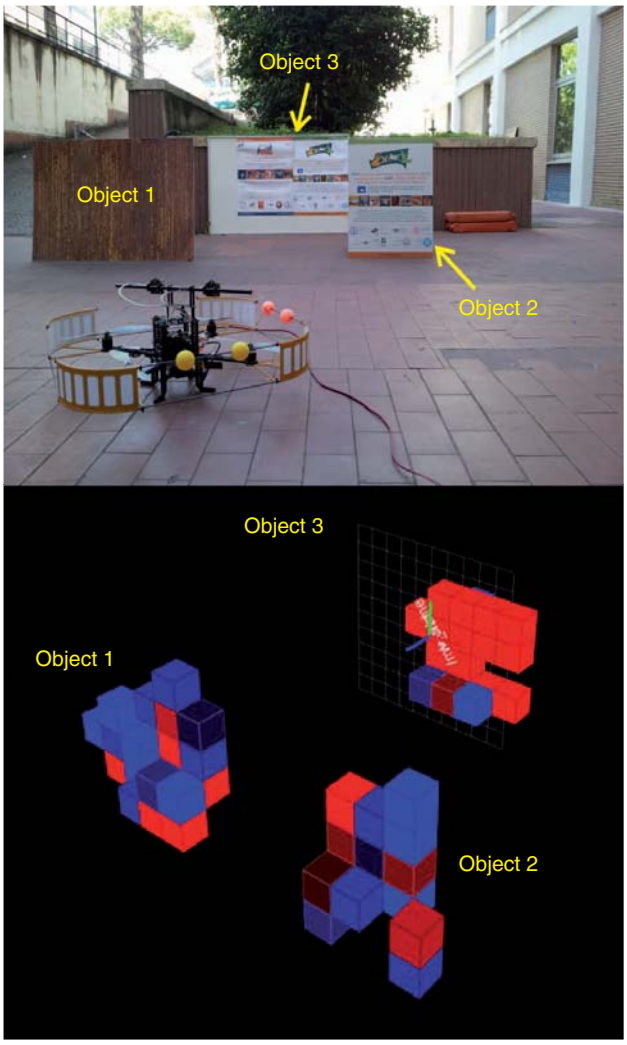


Figure 7. An example of an occupancy grid map of an outdoor environment that has been achieved by using only the onboard vision-based odometry.

The same data used for the occupancy map construction can also be employed for the building of an object-based map useful for high-level task-planner algorithms [24].

Image-Target Localization

The proposed teleoperated control framework is also endowed with the capability to autonomously identify a target in the environment for a given desired target image. A collection of images of the desired target is stored in the memory (e.g., the AIRobots [25] project logo of Figure 5 has been chosen for an experimental test). Through an off-line operation, the keypoints are found on the assigned target images, as well as the corresponding image descriptors, e.g., SURF citesurf or BRIEF [27]. During the operative phase, the keypoints extracted from the current image (i.e., the same keypoints already extracted for the obstacle detection algorithm) are employed to extract the corresponding image descriptors. Hence, the matching of the target keypoints with the current image is performed with a RANSAC-based algorithm. Note that the main difference with respect to the approach used for the obstacle detection is the use of a SURF/BRIEF descriptor instead of the Pyramidal Lukas-Kanade algorithm. Moreover, to improve the robustness of the detection algorithm, the target is considered to be statistically identified whether the number of the target inliers, extracted from the current image, is greater than a suitable threshold. This threshold value depends on the dimension and on the texture appearance of the target (i.e., on the number of features extracted off-line from the target image), as well as on the degree of robustness required for the identification. For example, for the experimental test with the image target of Figure 5, the threshold value has been chosen to be 20, while the total number of extracted features is about 30.

The localization of the target into the current environment map is required to set up an effective support navigation system. Hence, when a target is identified, the center of mass of the matching keypoints is evaluated in the image plane and the corresponding observation ray is stored in a dedicated buffer. This process is executed for each new grabbed image in which the target has been identified. When a number of observation rays with a suitable spatial distribution is reached, a least-squares triangulation process is employed to estimate the 3-D point that minimizes the distance from the observation rays, i.e., a sort of intersection point (in the ideal case this point lays on each observation ray, but in a real case there is no intersection point). To increase the statistical robustness of this process, some observation rays may be discarded if they are far from the current estimation with respect to the measurements' standard deviation. The estimation process is repeated until a discard occurs. After the target identification, i.e., the target is allocated in the map, a refinement process continuously updates its position estimation by adding newly available observation rays during the UAV motion.

Virtual Force Field

The information stored in the map is used to build a 3-D virtual force field that is displayed on the haptic device. Let $\boldsymbol{p}_r$ and $\boldsymbol{p}_o$ be the positions of the vehicle and of the considered obstacle (i.e., the corresponding cube), respectively. Hence, for each cube with a state equal to "OBSTACLE," a virtual repulsive force $\boldsymbol{f}_o(\boldsymbol{p}_r, \boldsymbol{p}_o)$ is generated as a sigmoid

$$\boldsymbol{f}_o(\boldsymbol{p}_r, \boldsymbol{p}_o) = \begin{cases} \dfrac{k_r \hat{\boldsymbol{r}}_{ro}}{1 + a_1 \exp(a_2 \cdot \| \boldsymbol{r}_{ro} \| - a_3)} & \text{if } \| \boldsymbol{r}_{ro} \| < d_t \\ 0 & \text{otherwise,} \end{cases}$$

where $\boldsymbol{r}_{ro} = \boldsymbol{p}_r - \boldsymbol{p}_o$, $\hat{\boldsymbol{r}}_{ro} = (\boldsymbol{r}_{ro}) / \| \boldsymbol{r}_{ro} \|$, k_r is a gain selected to scale the force into a desired range, d_t is the threshold distance under which the repulsive force starts to act on the UAV, and the parameters a define the shape of the sigmoid.

On the other hand, for the cube with a state set to "TARGET," an attractive force $\boldsymbol{f}_T(\boldsymbol{p}_r, \boldsymbol{p}_T)$, where $\boldsymbol{p}_T$ represents the position of the target, is generated as a sigmoid

$$\boldsymbol{f}_T(\boldsymbol{p}_r, \boldsymbol{p}_T) = \begin{cases} \dfrac{-k_r \hat{\boldsymbol{r}}_{rT}}{1 + b_1 \cdot \exp(b_2 \cdot \| \boldsymbol{r}_{rT} \| - b_3)} & \text{if } t_s < \| \boldsymbol{r}_{rT} \| < t_z \\ \dfrac{-k_r \hat{\boldsymbol{r}}_{rT}}{1 + c_1 \cdot \exp(c_2 \cdot \| \boldsymbol{r}_{rT} \| - c_3)} & \text{if } \| \boldsymbol{r}_{rT} \| \geq t_z \\ 0 & \text{otherwise,} \end{cases}$$

where $\boldsymbol{r}_{rT} = \boldsymbol{p}_r - \boldsymbol{p}_T$, $\hat{\boldsymbol{r}}_{rT} = (\boldsymbol{r}_{rT}) / \| \boldsymbol{r}_{rT} \|$, t_s represents the safety distance that has to be kept from the target, t_z delimits the area in which the vehicle is attracted to the target, and the parameters b and c describe the shape of the sigmoid.

To compute the total virtual force $\boldsymbol{F}_V$ exerted by the environment on the vehicle, the mean of all the forces $\boldsymbol{f}_{o,i}$ produced by each detected obstacle $i = 1, \cdots, n$ is taken together with the attractive force $\boldsymbol{f}_T$, i.e.,

$$\boldsymbol{F}_V = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{f}_{o,i}(\boldsymbol{p}_r, \boldsymbol{p}_{o,i}) + \boldsymbol{f}_T, \tag{1}$$

which is obtained with respect to the base reference frame. In Figure 8, an example of the force field, generated from the map of Figure 6 during a trajectory motion, is shown.

Obstacle avoidance is a core issue since any autonomous navigation system must preserve the safety of both the UAV and the surrounding environment.

Control Design

Figure 8 shows the proposed control architecture, composed of a high- and low-level controller [28]. Through this control structure, the aerial vehicle is endowed with the capability to autonomously avoid obstacles, to reach a desired target, and to support the human operator in the navigation maneuvers.

High-Level Control

The high-level teleoperation controller consists of two parts: the master system and the virtual slave system, which are coupled by means of the communication channel. The high-level controller is responsible for the high-level commands coming from the human operator, for the generation of the reference control inputs for the low-level controller and for the generation of the force-feedback.

The Master System

The human operator uses a master haptic device, characterized by a finite stroke $\boldsymbol{p}_m \in W_H$, where W_H is the finite haptic-device workspace, and a limited force $\boldsymbol{f}_f$, with $\|\boldsymbol{f}_f\| \in [\underline{f}_f, \overline{f}_f]$.

In steady state, a constant value of the position of the tip of the master is mapped to a constant reference speed for the aerial vehicle. This choice is due to the fact that the master has a finite configuration space, i.e., a finite range of motion, while the aircraft has an infinite configuration space [28].

The virtual slave is the real-time simulation of a fully actuated aerial vehicle, for which it is possible to measure all the information about its state.

The master controller is responsible of two main tasks: to generate a desired velocity set-point, $\boldsymbol{v}_v^*$, for the virtual slave and a feedback signal, $\boldsymbol{f}_f$, for the haptic device. The desired velocity, $\boldsymbol{v}_v^*$, is obtained as follows:

$$\boldsymbol{v}_v^* = \frac{\overline{v}_v}{\overline{f}_f}\,\boldsymbol{p}_m =: \alpha\,\boldsymbol{p}_m,$$

where $\overline{v}_v$ is the maximum velocity imposed by the user.

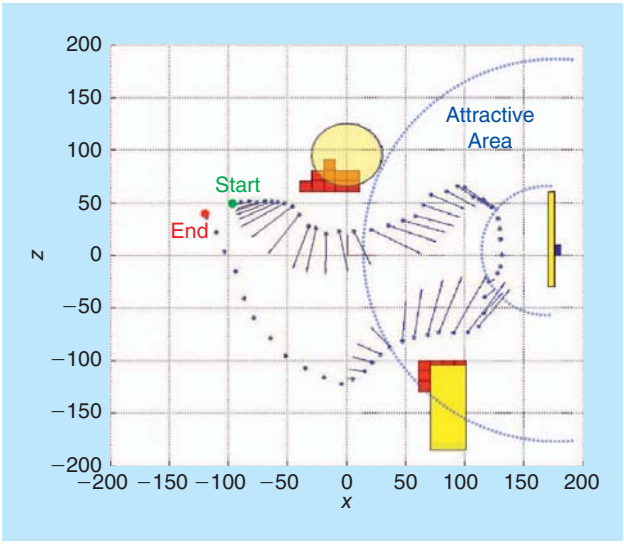


Figure 8. A force field generated from the map of Figure 6 during a trajectory motion.

The feedback force applied to the haptic device is proportional to the error between the position of the master $\boldsymbol{p}_m$ and the velocity of the virtual slave $\boldsymbol{v}_v$ as it is mapped to the master position, i.e., $\boldsymbol{p}_m^* = (1/\alpha)\,\boldsymbol{v}_v$. This implies that

$$\boldsymbol{f}_f = k_f\,(\boldsymbol{p}_m^* - \boldsymbol{p}_m),$$

where k_f is a proportional gain in units of a spring.

The Virtual Slave

The virtual slave is made of two main blocks: the virtual slave controller and the virtual slave [28], [29].

The virtual slave is the real-time simulation of a fully actuated aerial vehicle, for which it is possible to measure all the information about its state, i.e., its velocity $\boldsymbol{v}_v$. The virtual slave is moving in a gravityless and frictionless space and its dynamics are influenced by a viscoelastic coupling $\boldsymbol{F}_c$ to the real vehicle, the virtual vision force $\boldsymbol{F}_V$ in (1), and the force $\boldsymbol{f}_v^m$ impressed by the master's movement and computed in the virtual slave controller as

$$\boldsymbol{f}_v^m = b_v(\boldsymbol{v}_v^* - \boldsymbol{v}_v), \tag{2}$$

where b_v is a proportional gain in units of a viscous damper. For simplicity, the vehicle is modeled as point mass dynamics

$$m_v\dot{\boldsymbol{v}}_v = \boldsymbol{F}_c + \boldsymbol{F}_V + \boldsymbol{f}_v^m =: \boldsymbol{F}_v,$$

where m_v is the mass of the virtual slave, $\dot{\boldsymbol{v}}_v \in \mathbb{R}^3$ is the virtual slave acceleration. Note that, according to the scheme in Figure 9, the position of the virtual slave is assumed as the desired position for the real vehicle.

Low-Level Control

The low-level control is the controller of the real vehicle. Its main goal is to effectively track the desired state of the virtual slave by generating the proper actuation input for the real vehicle to solve the underactuation of the real slave. The real slave controller takes into account the possible intervention of the vision-based controller that modifies the desired reference to avoid a detected obstacle. Moreover, this controller yields the coupling force $\boldsymbol{F}_c$ to the high-level controller to provide information on the state of the real vehicle and the environment.

Vision-Based Control

The vision-based controller is responsible for the generation of the vision-based force, built as explained before. This force counteracts the force provided by the master through the virtual slave. Hence, the resultant force is applied to the real vehicle. When the distance between the vehicle and the obstacle is less than a certain threshold, the vision-based controller is applying a force, as defined in (1).

Passivity

To guarantee the stability of the overall system, a passivity-based approach is adopted [30]. More precisely, by assuring

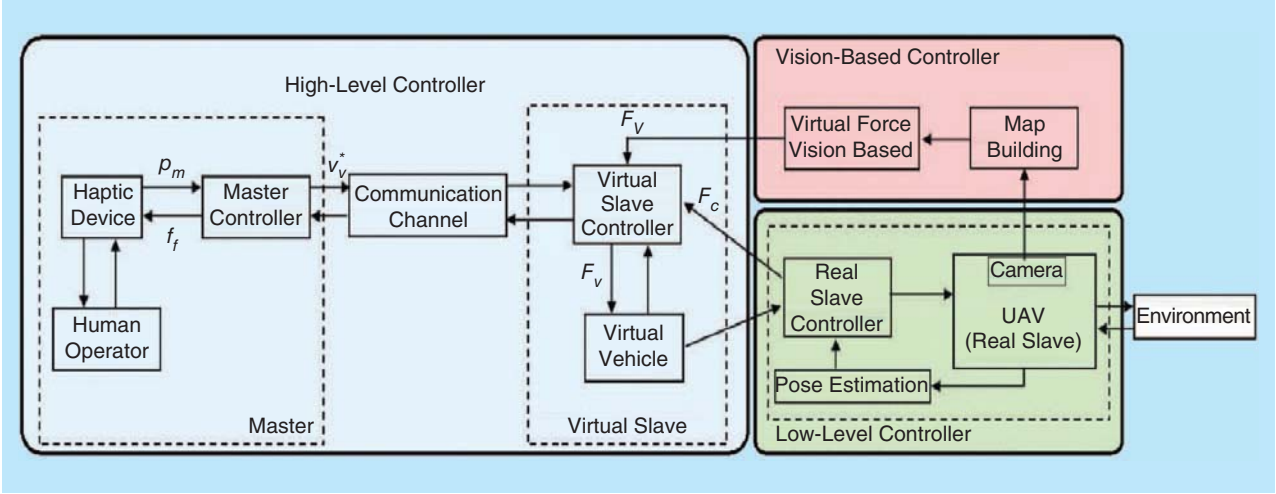


Figure 9. An overview of the vision-based teleoperation control architecture.

that every block of the teleoperation control scheme is passive (i.e., its energy is kept bounded), the complete system is guaranteed to be passive and, as a consequence, stable. With this respect, the human operator is assumed to be passive and the passivity of the master system is guaranteed by its local controller. The virtual slave is passive thanks to the passivity-enforcing supervisor, as proposed in [28]. Finally, the passivity of the low-level controller can be ensured separately by including an additional protection layer in it.

Experiments

The Omega.6 (Force Dimension, Switzerland), a pan-shaped force-feedback device, has been used as a haptic interface. It is intuitive and accurate due to the full gravity compensation and the driftless calibration. The device has six degrees of freedom, three of which are translational actuated axes, while the other three are rotational nonactuated axes. The local controller runs on Linux at a frequency of about 2.5 KHz.

To stabilize the pose of the vehicle through a position controller, the external tracker VisualEyez VZ 4,000 (Phoenix Technologies Inc., Canada) has been used. This tracker provides a motion capture of the vehicle, on which a set of Wi-Fi marker sensors are attached. The system has an accuracy of 0.5 mm, a 90° operation angle, and a capture area of 7×9 m^2.

The vehicle used for the experiments is the AscTec Pelican (Ascending Technologies GmbH, Germany), a quadrotor helicopter with dimensions $50 \times 50 \times 20$ cm, weight 750 g, and a maximum payload of 500 g. The operational flying time may vary in a range between 12 min with full payload and 25 min without payload with a standard battery. The UAV is equipped with an IMU that provides information about the pitch, roll, and yaw angles of the vehicle. Furthermore, there is an already built-in low-level controller, i.e., the autopilot, that allows for regulating the attitude, the angular speed of the yaw angle and the overall thrust. This controller runs on an ATOM processor of 1.6 GHz and 1 GB of random-access memory (RAM) mounted on board the vehicle.

The UAV is equipped with an Imaging Development Systems (IDS) μEye monochrome camera, mounted forward, with a resolution of 752×480 pixels and a maximal possible frame rate of 87 frames/s. The lens allows a $71° \times 49°$ (horizontal $\times$ vertical) field of view.

The goal of the performed experiment is to show that, during the inspection of the environment, the system is able to identify the obstacles and a target and, hence, to support the teleoperated navigation by providing a force feedback to the human operator to achieve a safer navigation and to reach the target more easily. The working scene is shown in Figure 5, where two different obstacles are present. As said before, the AIRobots project logo attached to a wall is employed as a target image.

In Figures 10–12, the position and velocity of the virtual slave and of the real vehicle are shown. The red lines in the top (middle) plots represent the position (velocity) of the virtual slave adopted as reference position (velocity) for the real vehicle, while the blue lines represent the position (velocity) of the real vehicle. In the bottom plots, the attractive (in green) and repulsive (in purple) forces applied to the virtual slave are represented. The blue lines represent the force feedback displayed on the haptic device as feedback for the human operator.

Note that the time scale is sliced into four periods. In the first period, between 4 and 29.5 s, the vehicle is maneuvered in a shared way, inspecting the environment to detect obstacles and target. In particular, when the haptic control is on, the operator gradually feels the force generated by the presence of the obstacles, which are identified in real time. In the second period, between 29.5 and 33 s, the vehicle is hovering (the haptic control is off) but the environment force is continuously generated to ensure that the vehicle keeps a safe distance from the obstacles. In the third period, between 33 and 40.5 s, the vehicle is piloted in such a way that the cylindrical obstacle is close to the vehicle (i.e., at a distance less than the safe threshold) and placed inside the attractive area of the localized target. In the last period, between 40.5 and 50 s, the vehicle is steered to the target under the only action of the

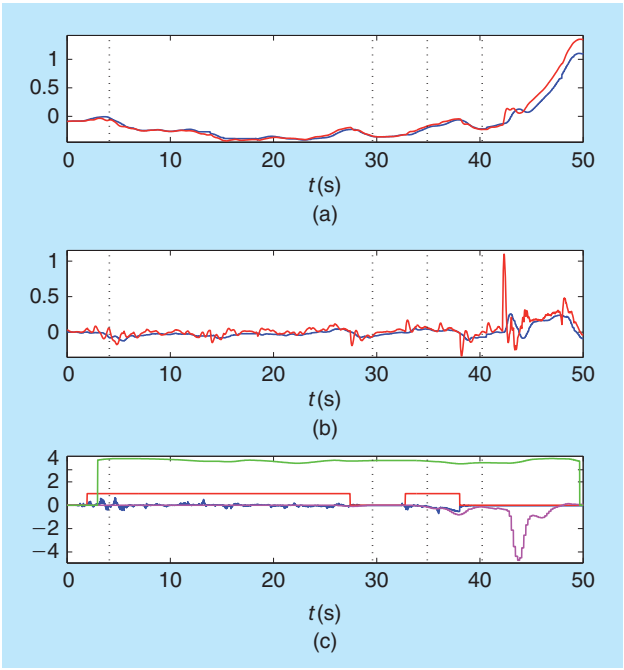


Figure 10. The *x*-component of the (a) position and (b) velocity of the virtual slave (red line) and real vehicle (blue line). (c) The corresponding haptic control state (red: OFF is 0, ON is 1) and of the attractive (green line), repulsive (purple line), and feedback force (blue line).

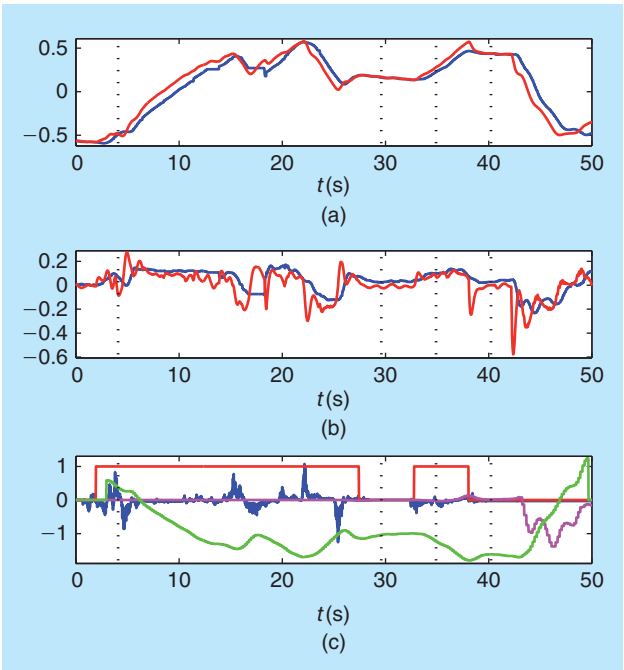


Figure 11. The *y*-component of the (a) position and (b) velocity of the virtual slave (red line) and real vehicle (blue line). (c) The corresponding haptic control state (red: OFF is 0, ON is 1) and of the attractive (green line), repulsive (purple line), and feedback force (blue line).

attractive and repulsive force provided by the vision algorithms, without any human intervention. It is easy to note that, in correspondence of about 43 s, the growth of the *x*–position component stops and the *y*–component increases. This means that the repulsive force generated by the obstacles

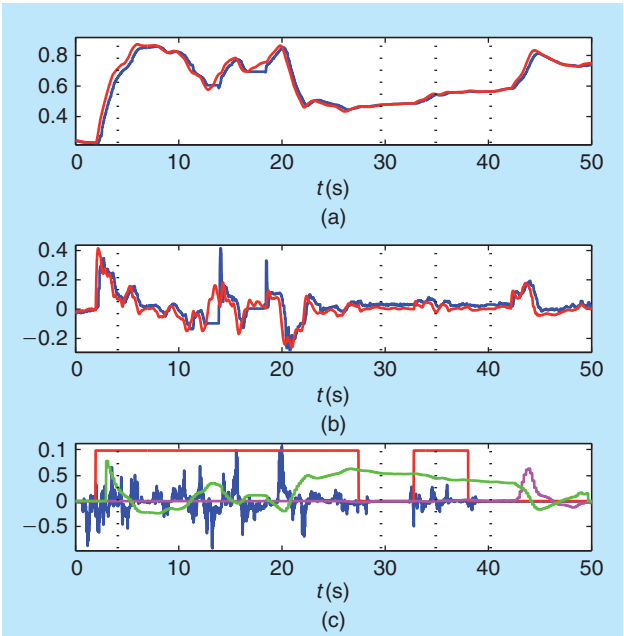


Figure 12. The *z*-component of the (a) position and (b) velocity of the virtual slave (red line) and real vehicle (blue line). (c) The corresponding haptic control state (red: OFF is 0, ON is 1) and of the attractive (green line), repulsive (purple line), and feedback force (blue line).

changes the trajectory of the vehicle to avoid the impact. Simultaneously, the attractive force toward the target is almost constant and it becomes zero when the vehicle reaches a commanded distance from the target, fixed to 50 cm.

Conclusions

In this article, a vision-based obstacle-avoidance/target identification technique has been combined with a haptic feedback device, developing a new teleoperated navigation system for underactuated UAVs in unknown indoor environments. A 3-D map of the surrounding environment has been estimated by matching keypoints between several images provided by an onboard camera and stored in a buffer with the corresponding estimated odometry provided by a PTAM-based module. Moreover, it has been shown by employing a visual identification algorithm that a desired target in the environment can be localized and reached by generating a virtual force field that the operator feels through a haptic force-feedback device. Finally, the overall method has been successfully tested on a UAV platform in a small indoor environment with obstacles and a target.

References

[1] Y. Kwag and J. Kang, "Obstacle awareness and collision avoidance radar sensor system for low-altitude flying smart UAV," in *Proc. Digital Avionics Systems Conf.*, 2004, pp. 12.D.2-1–12.D.2-10.

[2] B. A. Kumar and D. Ghose, "Radar-assisted collision avoidance, guidance strategy for planar flight," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 1, pp. 77–90, 2001.

[3] J. B. Saunders, O. Call, A. Curtis, A. W. Beard, and T. W. Mclain, "Static and dynamic obstacle avoidance in miniature air vehicles," in *Proc. Infotech Aerospace Conf.*, 2005, pp. 2005–6950.

[4] L. F. Tammero, V. L. S. Building, and D. Of, "The influence of visual landscape on the free flight behavior of the fruit fly drosophila melanogaster," *J. Exp. Biol.*, vol. 205, no. 3, pp. 327–343, 2002.

[5] M. V. Srinivasan, S. W. Zhang, M. Lehrer, and T. S. Collett, "Honeybee navigation en route to the goal: Visual flight control and odometry," *J. Exp. Biol.*, vol. 199, pt. 1, pp. 237–244, 1996.

[6] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart, "MAV navigation through indoor corridors using optical flow," in *Proc. IEEE Int. Conf. Robotics Automation*, 2010, pp. 3361–3368.

[7] V. Lippiello, G. Loianno, and B. Siciliano, "MAV indoor navigation based on a closed-form solution for absolute scale velocity estimation using optical flow and inertial data," in *Proc. IEEE Int. Conf. Decision Control European Control Conf.*, 2011, pp. 3566–3571.

[8] V. Lippiello and B. Siciliano, "Wall inspection control of a VTOL unmanned aerial vehicle based on a stereo optical flow," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2012, pp. 4296–4302.

[9] J. Serres, F. Ruffier, and N. Franceschini, "Two optic flow regulators for speed control and obstacle avoidance," in *Proc. IEEE/RAS-EMBS Int. Conf. Biomedical Robotics Biomechatronics*, 2006, pp. 750–757.

[10] A. H. P. Selvatici and A. H. R. Costa, "Obstacle avoidance using time-to-contact information," *Intell. Auton. Veh.*, vol. 1, pp. 509–514, July, 2004.

[11] S. E. Hrabar, "Vision-based 3-D navigation for an autonomous helicopter," Ph.D. dissertation, Dept. Comput. Sci., Univ. Southern California, Los Angeles, CA, 2006.

[12] J. C. Zufferey, *Bio-Inspired Flying Robots: Experimental Synthesis of Autonomous Indoor Flyers*. Lausanne: EPFL/CRC Press, 2008.

[13] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. IEEE/ACM Int. Symp. Mixed Augmented Reality*, 2007, pp. 225–234.

[14] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. DARPA Image Understanding Workshop*, 1981, pp. 674–679.

[15] J. Y. Bouguet. (2000). Pyramidal implementation of the Lucas-Kanade feature tracker: Description of the algorithm. *Intel Corp. Microprocess. Res. Labs* [Online]. Available: http://robots.stanford.edu/cs223b04/algo tracking.pdf

[16] M. A. Fischler and R. C. Bolles, "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.

[17] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme," in *Proc. IEEE Intelligent Vehicles Symp.*, 2010, pp. 486–492.

[18] L. Kneip, M. Chli, and R. Siegwart, "Robust real-time visual odometry with a single camera and an IMU," in *Proc. British Machine Vision Conf.*, 2011, pp. 16.1–16.11.

[19] V. Lippiello and R. Mebarki, "Closed-form solution for absolute scale velocity estimation using visual and inertial data with a sliding least-squares estimation," in *Proc. Mediterranean Conf. Control Automation*, 2013, pp. 1261–1266.

[20] S. Holmes, G. Klein, and D. W. Murray, "A square root unscented Kalman filter for visual monoSLAM," in *Proc. IEEE Int. Conf. Robotics Automation*, 2008, pp. 3710–3716.

[21] P. Smith, I. Reid, and A. Davison, "Real-time monocular SLAM with straight lines," in *Proc. British Machine Vision Conf.*, 2006, pp. 17–26.

[22] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics—Modelling, Planning and Control*. New York: Springer-Verlag, 2009.

[23] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 1994, pp. 593–600.

[24] F. Donnarumma, V. Lippiello, and M. Saveriano, "Fast incremental clustering and representation of a 3-D point cloud sequence with planar regions," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2012, pp. 3475–3480.

[25] L. Marconi, L. Basile, G. Caprari, R. Carloni, P. Chiacchio, C. Huerzeler, V. Lippiello, R. Naldi, N. Janosch, B. Siciliano, S. Stramigioli, and E. Zwicker, "Aerial service robotics: The AIRobots perspective," in *Proc. Int. Conf. Applied Robotics Power Industry*, 2012, pp. 64–69.

[26] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[27] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Proc. European Conf. Computer Vision: Part IV*, 2010, pp. 778–792.

[28] A. Y. Mersha, S. Stramigioli, and R. Carloni, "Bilateral teleoperation of underactuated aerial vehicles: The virtual slave concept," in *Proc. IEEE Int. Conf. Robotics Automation*, 2012, pp. 4614–4620.

[29] A. Y. Mersha, X. Hou, R. Mohony, S. Stramigioli, P. Corke, and R. Carloni, "Intercontinental haptic teleoperation of a flying vehicle: A step towards real-time applications," in *Proc. IEEE Int. Conf. Robotics Automation*, 2013.

[30] A. Isidori, *Nonlinear Control Systems II, Communication and Control Engineering Series*. London: Springer-Verlag, 1998.

Raffaella Carloni, Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Twente, The Netherlands. E-mail: r.carloni@utwente.nl.

Vincenzo Lippiello, PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy. E-mail: vincenzo.lippiello@unina.it.

Massimo D'Auria, PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy. E-mail: massimodauria6@hotmail.com.

Matteo Fumagalli, Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Twente, The Netherlands. E-mail: m.fumagalli@utwente.nl.

Abeje Y. Mersha, Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Twente, The Netherlands. E-mail: a.y.mersha@utwente.nl.

Stefano Stramigioli, Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Twente, The Netherlands. E-mail: s.stramigioli@utwente.nl.

Bruno Siciliano, PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy. E-mail: bruno.siciliano@unina.it.