2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

MAV Indoor Navigation Based on a Closed-Form Solution for Absolute Scale Velocity Estimation Using Optical Flow and Inertial Data

Vincenzo Lippiello, Giuseppe Loianno and Bruno Siciliano

Abstract— **A new vision-based obstacle avoidance technique for indoor navigation of Micro Aerial Vehicles (MAVs) is presented in this paper. The vehicle trajectory is modified according to the obstacles detected through the Depth Map of the surrounding environment, which is computed online using the Optical Flow provided by a single onboard omnidirectional camera. An existing closed-form solution for the absolute-scale velocity estimation based on visual correspondences and inertial measurements is generalized and here employed for the Depth Map estimation. Moreover, a dynamic region-of-interest for image features extraction and a self-limitation control for the navigation velocity are proposed to improve safety in view of the estimated vehicle velocity. The proposed solutions are validated by means of simulations.**

I. INTRODUCTION

The recent years have seen a growing interest on MAVs applications in several environments, e.g. surveillance and human dangerous scenarios. For indoor autonomous navigation the obstacle avoidance is one of the most relevant drawback, also because the GPS signal and a detailed environment map are often unavailable. A number of control strategies have been developed based on other on-board sensors like cameras, radar, lasers, sonars and IMUs (Inertial Measurement Units). However, the most promising approaches make use of visual sensors.

Several methods based on visual collision avoidance have been proposed. When a stereo camera system is available, an image couple can be employed as in [1] to compute distances towards detected objects based on triangulation. However, stereo systems require a high payload and onboard computational capacity. Several biologically inspired approach are also been presented. In [2] it is shown that fruit flies avoid obstacles when they turn away from the region with a high level of Optical Flow (OF). On the other hand, in [3] it is found out that honeybees try balancing the amount of lateral OF in order to stay equidistant from the flanking walls.

Different studies in the last years have concerned with the use of Optical Flow for obstacle avoidance. In some approaches the average intensity of the left and right OF vectors is balanced, according to the fact that if the left optical flow is larger than the right one, it means that the object is closer to the left side than the right one, and viceversa. A nonlinear control strategy for obstacle avoidance based on the OF is presented in [4], while autopilots for lateral obstacle avoidance of an hovercraft using two one-dimensional sensors pointing at $\pm 90^\circ$ have been developed in [3] and [5]. A single-camera frontal collision-avoidance strategy computing the divergence of the OF is proposed in [6], where an increase of the OF divergence indicate the presence of a frontal obstacle.

The optical flow has also been used for implementing altitude control for MAVs, e.g regulating the altitude of a helicopter using two downward optical flow sensors as in [7]. In this last, constant speed obtained by a constant pitch angle implies that the amount of OF is constant so that the vehicle stays at a constant height above ground [6], [8].

In [9] two different strategies, with and without the adoption of the OF, based on the *Time to Contact* –time needed to obtain a collision between the obstacle and the vehicle, while it is moving with a translational speed– have been proposed.

The *Depth Map* (DM) of the environment can be computed using the OF and GPS measurements. In [10], [11] an intuitive 3D map providing obstacle locations is provided using only OF and GPS data. A lateral obstacle avoidance algorithm for a wheeled robot has been proposed in [12], where a depth map obtained from the OF evaluated with an omnidirectional camera has been used. In [13] a real-time algorithm to compute the *Relative Depth Map* (RDM) from the OF independently of the performed motion, while in [14] the RDP is employed for the navigation through indoor corridors in the case of linear motion.

In this paper a new vision-based obstacle avoidance technique for indoor navigation is presented for MAVs applications. The vehicle trajectory is modified according to a repulsive force field generating from the DM of the surrounding environment computed online using the OF. A single onboard omnidirectional camera is assumed to be available. In particular, a new formulation for a closed-form solution for the absolute-scale velocity estimation problem, which are required for the DM estimation, is presented. Starting from the solution proposed in [15], where in addition to inertial measurements the correspondences of an image feature between three image frames (here referred as *visual station*) are required, a new compact formulation is adopted also generalizing to the case of multiple visual station and image features. Finally, a dynamic region-of-interest for image feature extraction and a navigation velocity self-limitation control are considered to improve safety during

The research leading to these results has been supported by the AIRobots collaborative project, which has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-248669. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

The authors are with PRISMA Lab, Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, via Claudio 21, 80125, Naples, Italy vincenzo.lippiello@unina.it, giuseppe.loianno@unina.it, bruno.siciliano@unina.it

navigation in view of the estimated vehicle velocity.

II. Depth map construction

The Optical Flow can be defined as the apparent motion of a image features (objects, surfaces, etc.) between two consecutive camera frames caused by the relative motion between the camera and the scene. It is known that the motion of obstacles observed in an image sequence depends on the distance of the object with respect to the camera, and thus the OF can be profitably exploited estimating the distances of surrounding obstacles. For this reason, OF is often employed in non-stereo visual based obstacle avoidance. However, the estimation of the absolute distance of an obstacle requires the knowledge of the vehicle translational velocity, which is here evaluated with a new closed-form solution based on image correspondences and IMU measurements.

A. Depth map construction with Optical Flow

In the case of a purely translational motion of the vehicle, assuming that all the objects in the scene are stationary, the translational Optical Flow ω_{T} of an image feature of an observed object depends on the relative velocity between the camera and the object itself $\boldsymbol{v}$ and on the angle between the direction of motion and the observed feature α, as shown in Fig. 1, with the following rule:

$$d = \frac{\|\boldsymbol{v}\|}{\omega_{\mathrm{T}}} \sin(\alpha), \tag{1}$$

where d is the distance between the object feature and the camera. If the velocity is available, the distance and so the position of the observed obstacle can be estimated. However, in a general case, the motion of the vehicle is composed of a translational part and of a rotational part, namely ω_{T} and ω_{R}, each of which produces a rate of the OF.

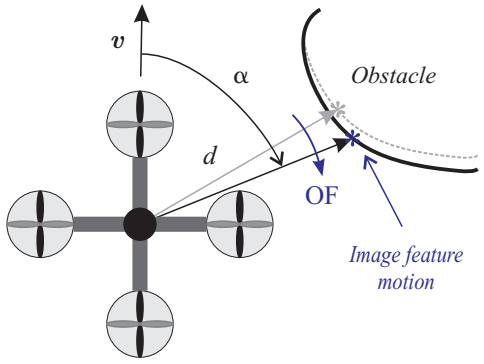

Fig. 1. Optical flow during a translational motion.

The computation of the ω_{T} component can be performed applying a compensation of the rotational effect as described in [14]. With reference to Fig. 2, the inertial and the camera reference frames are denoted with $I - x_I y_I z_I$ and $O - xyz$, respectively. Without loss of generality, it is supposed that the camera and the vehicle frames are coincident. The camera velocity $\boldsymbol{v}$ and acceleration $\boldsymbol{a}$, this last provided by the onboard IMU system with a period T, are expressed in camera frame. The orientation of the camera frame, also extracted using the IMU measurements, is referred to the inertial frame and expressed using the well-known Tait-Bryan (Euler) angles roll, pitch, and yaw $\boldsymbol{\phi} = (\varphi, \theta, \psi)$.

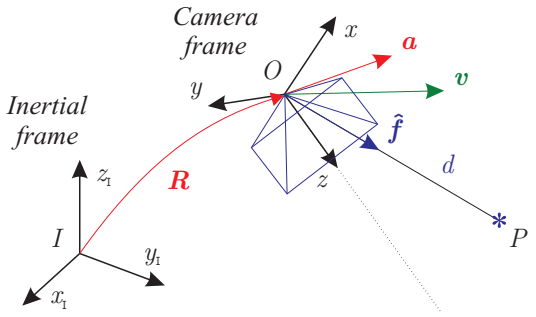

Fig. 2. Inertial and camera reference frames.

Adopting a classical pin-hole camera model (other models can be considered in view of the available hardware, e.g. see [16] for the case of fisheye lens) and assuming known the camera calibration parameters, the image feature vector $\boldsymbol{f} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{T}}$, i.e. the position of the observed feature with respect to the camera, can be expressed using the normalized image coordinates X and Y as follows

$$\boldsymbol{f} = z \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = d \cdot \hat{\boldsymbol{f}}, \tag{2}$$

where $d = \|\boldsymbol{f}\|$ is the distance of the feature and $\hat{\boldsymbol{f}}$ is the unit feature vector depending only on visual measurements X and Y.

The image features considered in this paper are corners, while the Pyramidal Lucas-Kanade algorithm [17], [18] has been employed to find matches. Denote with $\hat{\boldsymbol{f}}_1^1$ and $\hat{\boldsymbol{f}}_2^2$ the unit feature vectors of a correspondence between two consecutive images, both represented in the respective reference frames –conventionally, for vectors and matrices the reference frame is indicated as superscript– and with $\boldsymbol{\phi}_{12}$ the corresponding angular changes for the camera orientation. Then, the unit feature vector $\hat{\boldsymbol{f}}_2^1$ representing the position of the image feature measured in frame 2 reported in frame 1 can be evaluated as follows

$$\hat{\boldsymbol{f}}_2^1 = \boldsymbol{R}_2^1 \hat{\boldsymbol{f}}_2^2, \tag{3}$$

where $\boldsymbol{R}_2^1 = \boldsymbol{R}(\boldsymbol{\phi}_{12})$ is the rotational matrix representing the rotation performed by the camera in the form

$$\boldsymbol{R}(\boldsymbol{\phi}) = \begin{bmatrix} c_\varphi c_\theta & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\varphi s_\theta c_\psi + s_\phi s_\psi \\ s_\varphi c_\theta & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\varphi s_\theta c_\psi - c_\phi c_\psi \\ -s_\theta & -c_\theta s_\psi & c_\theta c_\psi \end{bmatrix}.$$

The corresponding ω_{T} can be estimated as the angular velocity of the feature vector evaluated in the interval Δt_{12}, between the image frames 1 and 2, given by

$$\omega_{\mathrm{T}} = \frac{\cos^{-1}\left(\hat{\boldsymbol{f}}_1^1 \cdot \hat{\boldsymbol{f}}_2^1\right)}{\Delta t_{12}}. \tag{4}$$

Figure 3 shows the ω_{T} computed in a real indoor scene with an omnidirectional fisheye camera.

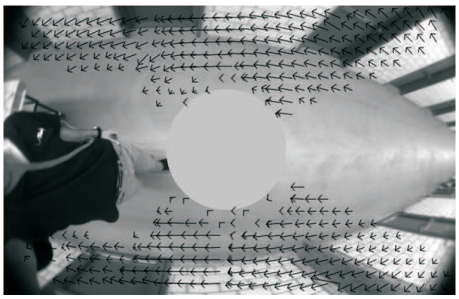

Fig. 3. Optical Flow estimated in a real scene.

For a given vehicle translational velocity $\boldsymbol{v}$, substituting (4) in (1) and the result in (2), the set of all feature vectors $\boldsymbol{f}$ of the available image correspondences can be evaluated, constituting the instant *Depth Map* of the surrounding environment at the time of the image frame acquisition.

B. Velocity estimation

In this section a generalization of the method proposed in [15] is presented with a more compact analytical formulation, where the extension to a multi-frame multi-feature correspondence is explicitly considered. Without loss of generality, it is assumed that the period of the visual system is N times the period of the IMU system T. This means that between two consecutive images there are N available measures provided by the IMU. Moreover, it is assumed that the IMU and the camera reference frames are coincident –if both are calibrated it is easy to refer IMU data to the camera frame– and that the IMU is ideal, i.e. it provides gravity and bias-free acceleration and gyroscopic measurements. Therefore, only the camera frame will be considered in the rest of the section. Finally, the acceleration $\boldsymbol{a}$ is always expressed in the current camera frame (e.g. $\boldsymbol{a}_j = \boldsymbol{a}_j^j$, where j refers to the camera frame at the time instant t_j).

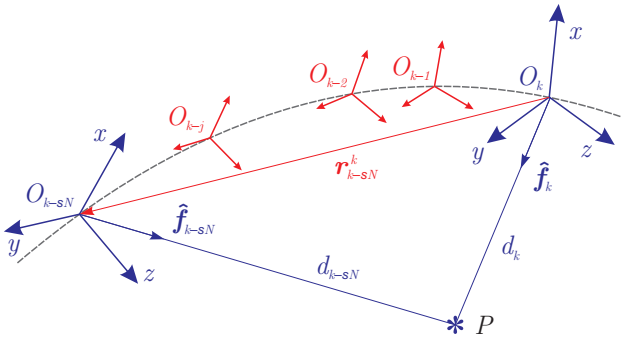

Fig. 4. Camera (blu) and IMU measurement reference frames.

Considering a camera motion as shown in Fig. 4 and assuming that t_k is the last sample time with available visual data, the previous available visual measurements are referred to the sample times t_{k-sN}, with $s \in \mathbb{N}$ (s identifies each visual station). By denoting with $\boldsymbol{r}_i^j$ the relative displacement of the frame i with respect and referred to the frame j and considering a single image feature match between frames k and $k_s = k - sN$, the following relation can be written

$$d_{k_s}\hat{\boldsymbol{f}}^{k_s}_{k_s} = \left(\boldsymbol{R}^k_{k_s}\right)^{\mathrm{T}}\left(d_k\hat{\boldsymbol{f}}^k_k - \boldsymbol{r}^k_{k_s}\right), \tag{5}$$

where $\boldsymbol{R}^k_{k_s} = \left[\begin{array}{ccc}\boldsymbol{r}_x & \boldsymbol{r}_y & \boldsymbol{r}_z\end{array}\right]^k_{k_s}$ is the rotational matrix representing the orientation of frame k_s with respect to frame k, and $\boldsymbol{r}_x$, $\boldsymbol{r}_y$, and $\boldsymbol{r}_z$ are the its column vectors. This relative displacement can be expressed in terms of the current velocity, with respect to the current camera frame k, and the integration of acceleration samples between t_{k-sN} and t_k. Let us consider the relative displacement and velocity between two consecutive frames:

$$\boldsymbol{r}_j^{j-1} = \boldsymbol{v}_{j-1}^{j-1}T + \frac{1}{2}\boldsymbol{a}_{j-1}T^2 \tag{6}$$

$$\boldsymbol{v}_j^{j-1} = \boldsymbol{v}_{j-1}^{j-1} + \boldsymbol{a}_{j-1}T \tag{7}$$

$$\boldsymbol{r}_{j-1}^{j} = -\boldsymbol{R}_{j-1}^{j}\boldsymbol{r}_j^{j-1} = -\boldsymbol{v}_{j-1}T - \frac{1}{2}\boldsymbol{R}_{j-1}^{j}\boldsymbol{a}_{j-1}T^2 \tag{8}$$

$$\boldsymbol{v}_j = \boldsymbol{R}_{j-1}^{j}\boldsymbol{v}_j^{j-1} = \boldsymbol{v}_{j-1} + \boldsymbol{R}_{j-1}^{j}\boldsymbol{a}_{j-1}T. \tag{9}$$

Replacing (9) in (8) yelds

$$\boldsymbol{r}_{j-1}^{j} = -\boldsymbol{v}_j T + \frac{1}{2}\boldsymbol{R}_{j-1}^{j}\boldsymbol{a}_{j-1}T^2. \tag{10}$$

The whole displacement between two consecutive visual frames can be achieved adding all the displacements corresponding to the intermediate time intervals where only IMU data are available, obtaining

$$\boldsymbol{r}^k_{k_s} = -sNT\boldsymbol{v}_k + \frac{1}{2}\bar{\boldsymbol{a}}^k_{k_s}T^2, \tag{11}$$

with

$$\bar{\boldsymbol{a}}^k_{k_s} = \sum_{j=1}^{sN}(2(sN-j)+1)\boldsymbol{R}^k_{k-j}\boldsymbol{a}_{k-j}, \tag{12}$$

which can also be expressed in a recursive formulation, here omitted for brevity.

By plugging (11) in (5) and considering (2), the following system of equations for a one-point image correspondence between frames k and k_s is derived

$$X_{k_s} = \frac{\left(\boldsymbol{r}^k_{x,k_s}\right)^{\mathrm{T}}\left(d_k\hat{\boldsymbol{f}}^k_k + sNT\boldsymbol{v}_k - \frac{1}{2}\bar{\boldsymbol{a}}^k_{k_s}T^2\right)}{\left(\boldsymbol{r}^k_{z,k_s}\right)^{\mathrm{T}}\left(d_k\hat{\boldsymbol{f}}^k_k + sNT\boldsymbol{v}_k - \frac{1}{2}\bar{\boldsymbol{a}}^k_{k_s}T^2\right)} \tag{13}$$

$$Y_{k_s} = \frac{\left(\boldsymbol{r}^k_{y,k_s}\right)^{\mathrm{T}}\left(d_k\hat{\boldsymbol{f}}^k_k + sNT\boldsymbol{v}_k - \frac{1}{2}\bar{\boldsymbol{a}}^k_{k_s}T^2\right)}{\left(\boldsymbol{r}^k_{z,k_s}\right)^{\mathrm{T}}\left(d_k\hat{\boldsymbol{f}}^k_k + sNT\boldsymbol{v}_k - \frac{1}{2}\bar{\boldsymbol{a}}^k_{k_s}T^2\right)}. \tag{14}$$

In the general case, by considering $n_s \geq 2$ visual stations and n_f image features, a system of $2n_sn_f$ equation with $3 + n_f$ unknowns $\boldsymbol{v}_k$ and $\boldsymbol{d}_k$, where $\boldsymbol{d}_k$ is the n_f vector of distances of each image feature, is achieved. This linear system can be easily arranged in the classical form

$$\boldsymbol{A}\left[\begin{array}{c}\boldsymbol{v}_k\\ \boldsymbol{d}_k\end{array}\right] = \boldsymbol{b}, \tag{15}$$

that for $n_s = 2$ and $n_f = 1$ becomes a square system of 4 equations in 4 unknowns. However, by increasing n_s and/or n_f, a least-squares solution can be achieved, which is

robust to noise, but with some limitations. If n_s is increased, the number of unknowns do not change, i.e. the complexity of the system solution remains the same, and the baseline employed for the triangulation considered in the equation system is enlarged. resulting in a well numerical conditioned problem. However, in this case more IMU samples will be integrated, resulting in a bad solution is the quality of the IMU system is poor, as the typical case of MAVs. On the other hand, increasing n_f the same number of IMU data is employed but the number of unknowns increases linearly: the matrix $\boldsymbol{A}$ assumes a sparse conformation and the solution of the system becomes quickly inefficient; the complexity of the image feature matching algorithm increase and becomes less robust (increase the probability of outliers).

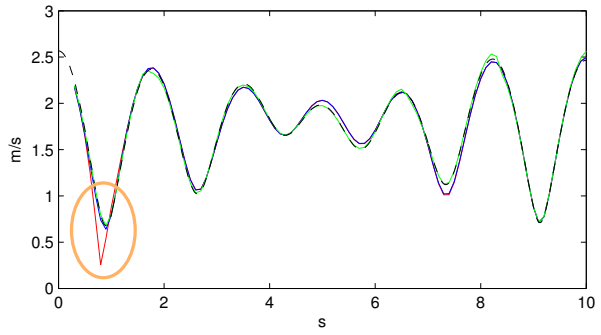

Fig. 5. A comparison of several cases for the absolute-scale velocity estimation: true value (dark dashed line), case with $n_s = 2$ and $n_f = 1$ (red line), case with $n_s = 2$ and $n_f = 2$ (green line), and case with $n_s = 3$ and $n_f = 1$ (blue line).

Taking into account these considerations, a tradeoff is required (e.g. $n_s = 3$ or 4 is a good IMU system is available, $n_f \leq 3$). A comparison between several cases is showed in Fig. 5, where the ideal case with $T = 10$ ms, $N = 10$ is considered. Obviously, best results are achieved when the number of image features are increased, while at the beginning of the trajectory it is noticeable a bad numerical solution for the minimum system case. This last condition happens with a significant frequency for a number of tested trajectories, then this choice it is inadvisable for a real case.

Notice that the proposed solution becomes singular in two cases: 1) when the velocity of the camera is constant, i.e. when the value of the integral of the acceleration over two camera observation points is very small, and hence the motion remain unobservable. However, this case can be easily detected at runtime monitoring the result of the IMU integration. 2) when the selected image features are aligned along the motion direction. In this case it is sufficient a selection of a new candidate feature set.

III. Navigation control

Once estimated the vehicle velocity, the distance of each feature observed in the scene and associated to an OF element can be evaluated and collected together with the corresponding optical rays. The result is a temporary environmental map, namely Depth Map, which can be fully exploited for lateral obstacle avoidance during the navigation.

A. Dynamic region-of-interest

The OF computation requires, as explained before, an image feature extraction algorithm and a matching algorithm, that can be computational expensive for the typical processor units available on a MAV. In the case of an omnidirectional camera, the adoption of region-of-interest (RoI) for the image elaboration processes may provide a large benefit in terms of computational requirement, while the main drawback is that the systems becomes "blind" outside the RoI. However, the adoption od a dynamic RoI that is smartly adapted online to the real environmental and navigation conditions may reduce the risk of an unpredicted impact. Observing that, due to the inertial of the system, an obstacle can be avoided only if it is detected as early as possible with respect to the vehicle velocity, the solution proposed is to adopt a RoI that "looks" more forward as the vehicle is moving quickly.

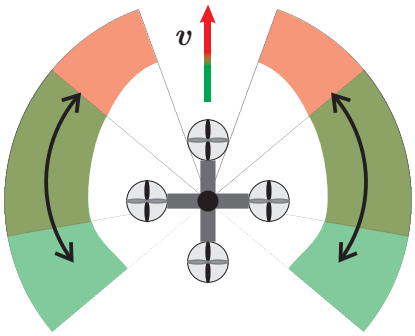

Fig. 6. Dynamic region of interest.

In this paper the RoI is composed of two regions, namely left and right RoI, which are symmetric with respect to the direction of motion. Both regions have a fixed total extension around the vertical axis, but they are rotated in view of an angular offset θ_{of} with respect to the navigation velocity (see Fig. 6). Notice that the forward region in the direction of motion is discarded due to numerical inconsistency of the OF along this direction. By denoting with θ_M the maximum offset angle for the RoI, an exponential adaptation law is considered for an offset anglewith respect to the motion direction as follows

$$\theta_{of} = \begin{cases} \theta_M \left(1 - e^{-4\frac{\|\boldsymbol{v}\| - v_m}{v_M - v_m}}\right) & \text{if } \|\boldsymbol{v}\| > v_m \\ 0 & \text{if } \|\boldsymbol{v}\| \leq v_m, \end{cases} \tag{16}$$

where v_m and v_M are the minimum and maximum values which can be assumed from the cruise velocity.

Also the vertical extension of the RoI is shaped in view of the offset, symmetrically reducing its range with the increase of θ_{of}. This behavior is required for omnidirectional cameras, that compresses objects extension in the image as far as they are along the direction of motion.

B. Lateral obstacle avoidance control

The safety of the vehicle during navigation within an indoor environment depends on its capability to avoid unplanned lateral obstacles.

With respect to the dynamic left and right RoI presented above and for each available DM, the distances of the vehicle with respect to the left and right side of the surrounding environment are computed with the following procedure. By denoting with $\hat{\boldsymbol{v}} = \boldsymbol{v}/\|\boldsymbol{v}\|$ the unit vector pointing along the motion direction, the distances of each detected feature, which is characterized by its feature estimated vector $\boldsymbol{f}$, along the motion direction $s_{\hat{\boldsymbol{v}}}(\boldsymbol{f}) = \boldsymbol{f}^{\mathrm{T}} \cdot \hat{\boldsymbol{v}}$ and with respect to the forward axis $d_{\hat{\boldsymbol{v}}}(\boldsymbol{f}) = \|x_{\hat{\boldsymbol{v}}}(\boldsymbol{f})\hat{\boldsymbol{v}} - \boldsymbol{f}\|$ are computed. Then, the vectors of distances from the left $\boldsymbol{d}^L_{\hat{\boldsymbol{v}}}$ and the right $\boldsymbol{d}^R_{\hat{\boldsymbol{v}}}$ sides of the navigation direction are composed using increasing values of $s_{\hat{\boldsymbol{v}}}$ as a sort criteria. Finally the minimum of each distance vector is found and a local spacial average is applied resulting in the minimum mean distances $\bar{d}^L_{\hat{\boldsymbol{v}}}$ and $\bar{d}^R_{\hat{\boldsymbol{v}}}$. Depending on the application, a LP-filter can be considered to reduce discontinuities due to the changing of the observed features.

Assuming d_{ls} as a safety lateral distance, a course correction is obtained through a PD controller acting on the following error

$$e_l = \begin{cases} \frac{\bar{d}^L_{\hat{\boldsymbol{v}}} - \bar{d}^R_{\hat{\boldsymbol{v}}}}{d_{ls}} & \text{if } \bar{d}^L_{\hat{\boldsymbol{v}}} + \bar{d}^R_{\hat{\boldsymbol{v}}} < 2d_{ls} \\ 1 - \frac{\bar{d}^R_{\hat{\boldsymbol{v}}}}{d_{ls}} & \text{if } \bar{d}^L_{\hat{\boldsymbol{v}}} \geq d_{ls}, \bar{d}^R_{\hat{\boldsymbol{v}}} < d_{ls} \\ \frac{\bar{d}^L_{\hat{\boldsymbol{v}}}}{d_{ls}} - 1 & \text{if } \bar{d}^L_{\hat{\boldsymbol{v}}} < d_{ls}, \bar{d}^R_{\hat{\boldsymbol{v}}} \geq d_{ls} \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

Notice that $\bar{d}^L_{\hat{\boldsymbol{v}}} + \bar{d}^R_{\hat{\boldsymbol{v}}} < 2d_{ls}$ means that the vehicle is navigating in a narrow environment, e.g. a corridor, and in this case the previous control tries keeping the vehicle in the middle of the free space, while the following cruise control reduces the vehicle velocity.

C. Cruise control

The proposed navigation control considers a *cruise velocity* of the vehicle v_c along the direction of motion in the case of free space. However, for the safety of the vehicle, when an obstacle is detected or when the dimension of the space that is free for the motion is reduced, i.e. the minimum distance with respect to the environment d becomes less than a safety distance d_s, a reduction of the navigation velocity is commanded. The module of the navigation velocity is generated applying a virtual control force f_v in the desired direction of motion, which is generated with an exponential law as follows

$$f_v = f_p \left(1 - e^{-4\frac{\|\boldsymbol{v}\|}{v_c}}\right) - \tag{18}$$

$$f_{sM} \left(1 - e^{-4\frac{d_s - d}{\gamma_v d_s}}\right) \left(1 - e^{-4\frac{\|\boldsymbol{v}\| - v_m}{v_c - v_m}}\right), \tag{19}$$

with

$$f_{sM} = \begin{cases} F_s & \text{if } \|\boldsymbol{v}\| > v_m, d < d_s \\ 0 & \text{otherwise,} \end{cases}$$

where $\gamma_v \in (0, 1)$ determines the rate of reduction of the velocity when the distance d becomes less than d_s, v_m is the minimum cruise velocity that has to be assured, and F_s is the maximum braking force.

IV. Simulation results

The performance of the proposed DM construction algorithm and of the navigation control has been tested with simulations using the MATLAB/Simulink environment.

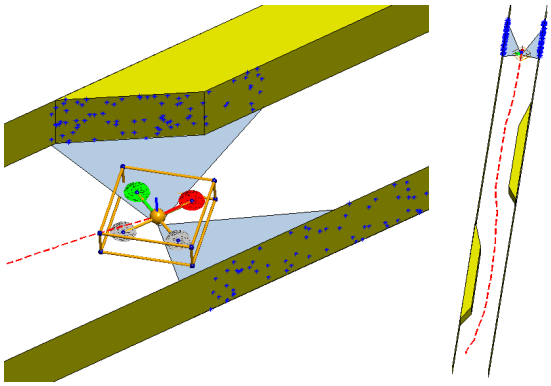

Fig. 7. Simulated indoor environment.

In Fig. 7 a sketch of the employed simulator is showed. The considered indoor environment is similar to a corridor of a total length of 25 m and with a longitudinal shape that changes along the path. In particular the width of the free navigable space varies several times from 2 to 1 m, and vice versa, also changing in its middle line position.

A random occurrence of image features has been considered on both sides of the environment without outliers. Gaussian white noise has been added on image and IMU measurements. For the velocity estimation, the case $n_s = 2$ and $n_f = 2$ has been considered with $T = 0.01$ s and $N = 10$.

The adopted dynamic model of the vehicle can be found in [19]. The control inputs are the two tilt angles, the angular velocity around the vertical axis and the thrust, while the outputs are the position and the yaw angle. In particular, the vehicle is modeled in the inertial frame as a simple point-mass model using the second Newton's law. The forces acting on the system are the controlled thrust τ and the gravity g, as shown in the following system:

$$\ddot{\boldsymbol{a}} = \frac{1}{m}\boldsymbol{R}^I(\varphi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ -\tau \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \tag{20}$$

where $m = 0.5$ kg is the vehicle mass and $\boldsymbol{R}^I(\varphi, \theta, \psi)$ is the rotation matrix of the vehicle frame with respect to the inertial frame, which depends on the roll, pitch and yaw angles. The delay acting on the control angles due to the internal controller action can be modeled as a second order system:

$$L(s) = \frac{\omega^2}{s^2 + 2 \cdot d \cdot \omega \cdot s + \omega^2}, \tag{21}$$

where $\omega = 15.92$ rad/s and $d = 1.22$. Supposing that the controller is fast enough and smooth, it is possible to consider the delay acting on forces and not on the angles, so to obtain four linear and decoupled systems respect to the forces. With respect to these parameters, the PD controller of

the lateral obstacle avoidance control has been designed in the frequency domain with the following transfer function:

$$C(s) = \frac{0.008(100s+1)}{0.001s+1}. \quad (22)$$

Some of the most significant adopted parameters are as follows: $\theta_M = 30°$ for a total lateral angle of view of $80°$, $v_c = 2.44\ m/s$, $v_m = v_c/4$, $d_{ls} = d_s = 1.0$ m, $\gamma_v = \gamma_l = 0.25$.

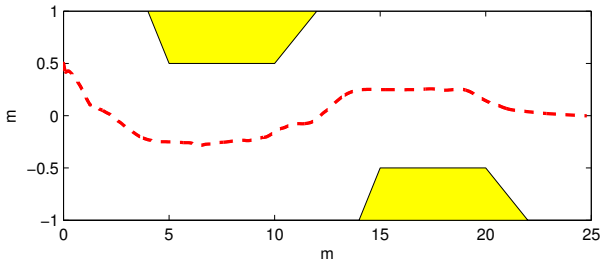

Fig. 8. Course correction during navigation in view of the detected obstacles.

The course correction achieved during the navigation is shown in Fig. 8, where also the shape of the environment has been reported. The vehicle starts from the home position that is near to the left side of the environment. The path followed by the vehicle is almost centered in the middle of the available free space left to the vehicle as desired.

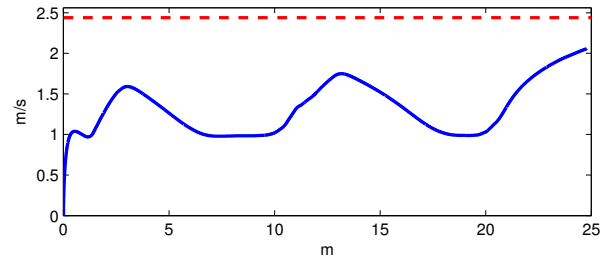

Fig. 9. Navigation velocity modified in view of the detected obstacles and of the current free space (blue line) and adopted v_c (red dashed line).

In Fig. 9 the navigation velocity modified in view of the detected obstacles and of the current free space is shown. As expected, the velocity is reduced when the vehicle is near to obstacles or in a restricted area. The cruise velocity in the narrow part of the environment is decreased, in view of the adopted parameters, to about 1 m/s, while when the available space increases also the velocity increases tending to v_c.

V. Conclusion

A new vision-based obstacle avoidance technique for indoor navigation of Micro Aerial Vehicles has been presented. The Depth Map of the surrounding environment has been constructed using only visual and inertial measurements. In particular, an existing closed-form solution for the absolute-scale velocity estimation based on visual correspondences and inertial measurements has been generalized and employed for the velocity estimation. This last has been used for the evaluation of the absolute-scaled Optical Flow, which allows the construction of the desired Depth Map. Based on this map a safe navigation control has been proposed, which able to avoid lateral obstacles, to self-limit the cruise velocity in view of the available free space, and to dynamically set the regions of interest for image features extraction. Simulations have been carried out to prove the effectiveness of the proposed solution.

References

[1] S.E. Hrabar, *Vision-based 3D Navigation for an Autonomous Helicopter*, Ph.D. thesis, USC, 2006.

[2] L.F. Tammero, V.L.S. Building, D.Of, "The influence of visual landscape on the free flight behavior of the fruit fly drosophila melanogaster", *Journal of Experimental Biology*, vol. 205, pp. 327–343, 2002.

[3] M.V. Srinivasan, S.W. Zhang, M. Lehrer, T.S. Collett, "Honeybee navigation en route to the goal: Visual flight control and odometry", *Journal of Experimental Biology*, vol. 199, pp. 237–244, 1996.

[4] F. Kendoul, I. Fantoni, K. Nonami, "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles", *Journal of Robotics and Autonomous Systems*, vol. 57, pp. 591–602, 2009.

[5] J. Serres, D. Dray, F. Ruffier, N. Franceschini, "A vision-based autopilot for a miniature air vehicle: Joint speed control and lateral obstacle avoidance", *Autonomous Robots*, vol. 25, pp. 103–122, 2008.

[6] J.C. Zufferey, A. Klaptocz, A. Beyeler, J.D. Nicoud, D. Floreano, "A 10-gram vision-based flying robot", *Advanced Robotics*, vol. 21, pp. 1671–1684, 2007.

[7] F. Ruffier, N.H. Franceschini, "Aerial robot piloted in steep relief by optic flow sensors", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 2008.

[8] J. Serres, D. Dray, F. Ruffier, N. Franceschini, "Autonomous landing for indoor flying robots using optic flow", *ASME International Mechanical Engineering Congress*, Washington, DC, 2003.

[9] H.A.P. Selvatici, A.H.R. Costa, "Obstacle avoidance using time-to-contact information", *IEEE International Conference on Robotics and Automation*, Rome, Italy, 2007.

[10] B.R. Call, *Obstacle Avoidance for Unmanned Air Vehicles*, M.S. thesis, Brigham Young University, 2006.

[11] B.R. Call, R. Beard, C. Taylor, B. Barber, "Obstacle avoidance for unmanned air vehicles using image feature tracking", *AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, 2006.

[12] M. Zamponi, *Optical Flow Based Navigation*, M.S. thesis, ETH Zurich, 2009.

[13] C. McCarthy, N. Barnes, M. Srinivasan, "Real time biologically-inspired depth maps from spherical flow", *International Journal of Computer Vision*, vol. 12, pp. 43–77, 1994.

[14] S. Zingg, D. Scaramuzza, S. Weiss, R. Siegwart, "MAV navigation through indoor corridors using optical flow", *IEEE International Conference on Robotics and Automation*, Anchorage, AK, 2010.

[15] L. Kneip, S. Weiss, A. Martinelli, D. Scaramuzza, R. Siegwart, "A closed-form solution for absolute scale velocity determination combining inertial measurements and a single feature correspondence", *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[16] D. Scaramuzza, A. Martinelli, R. Siegwart, "Toolbox for easily calibrating omnidirectional cameras", *"IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.

[17] B.D. Lucas, T. Kanade, "An iterative image registration technique with an application to stereo vision", *International Joint Conference on Artificial Intelligence*, Vancouver, Canada, 1981.

[18] J.Y. Bouguet, *Piramidal Implementation of the Lucas-Kanade Feature Tracker: Description of the Algorithm*, Intel Corporation Microprocessor Research Labs, 2002.

[19] M. Blösch, S. Weiss, D. Scaramuzza, R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments", *IEEE International Conference on Robotics and Automation*, Anchorage, AK, 2010.