Omnidirectional humanoid navigation in cluttered environments based on optical flow information

Marco Ferro, Antonio Paolillo, Andrea Cherubini, Marilena Vendittelli

Omnidirectional humanoid navigation in cluttered environments based on optical flow information

Marco Ferro[1] Antonio Paolillo[2] Andrea Cherubini[2] Marilena Vendittelli[1]

Abstract— In this paper, we address the problem of humanoid navigation in a priori unknown environments, cluttered by obstacles. The robot task is to move within the environment without colliding with obstacles and using only ordinary on-board sensors, like monocular cameras and encoders. The proposed approach relies on: (i) optical flow information, to construct a local representation of the environment obstacles and free space; (ii) visual servoing techniques, to achieve safe motion within the environment while regulating appropriate visual features and the robot internal configuration. In case of navigation in a straight corridor, it can be formally proved that the robot converges to the corridor bisector. With respect to previous works, the algorithm proposed here does not make use of any information about the environment, and exploits the humanoid omnidirectional walking capability to achieve safe navigation in narrow passages. The approach is validated through simulations and experiments with NAO.

I. INTRODUCTION

In recent years, the interest of the robotics community has turned to humanoids, with the ultimate goal of deploying them in partially unknown environments where they must accomplish tasks while avoiding obstacles.

Sensor-based humanoid navigation in unknown environments is, however, still at its dawn. Among the not so rich literature on the subject it is worth citing sensor-based localization and navigation techniques, historically developed for wheeled robots, that have been transferred to humanoids. For instance, in [1], humanoid localization and path planning are realized, for a tour guide robot, by relying on potential fields. The authors of [2] present a learning-based approach, achieving navigation among obstacles through a pair of cameras and a laser range finder.

The specificities of humanoids have required in some cases adaptations of the original methods. For example, in [3] the robot footstep generator is based on visual feedback to obtain a dynamically stable walking. Furthemore cameras are often the most popular choice in humanoids rather than lidars or other range sensors used in wheeled mobile robots. Visual SLAM algorithm relying on depth information provided by a kinect sensor is used in [4], while a monocular camera is sufficient in [5] to achieve humanoid navigation using a view-based approach [6]. The drawback of view-based approaches is that they require the environment to be visited beforehand by the robot, to record a sequence of reference images. Instead, in [7], a humanoid navigates in networks of unknown corridors, connected through curves and junctions, using a visual servoing technique and images from a monocular camera. Similarly, in [8], autonomous navigation in a featureless maze is achieved through classification of image regions, and subsequent topological map construction. This approach does not allow a real-time performance as the robot must regularly stop walking to process the images and distinguish walls and floors.

In most navigation works, humanoid motion is modelled as that of a unicycle, with the linear velocity constrained to lay on the sagittal plane. While this approach proves suitable for modeling human motion [9] and for long distance navigation in uncluttered environments, it does not take advantage of the omnidirectional walk capability of humanoids.

In this paper we do not take any hypothesis on the environment geometry, nor assume any a priori information about the environment, except that the robot moves on a flat floor, and allow for ominidirectional walk when this is needed to cross narrow passages among obstacles. In addition, differently from [7], there is no need to introduce a logic when navigating in maze-like environments.

The paper is organized as follows. Sect. II presents the problem formulation. Our perception and control methods are outlined respectively in Sect. III and IV. Simulation and experimental results are presented in Sect. V, and we conclude in Sect. VI.

II. PROBLEM DESCRIPTION AND PROPOSED APPROACH

The problem approached in this paper is the avoidance of workspace obstacles during navigation of a humanoid robot, without any a priori knowledge of the environment geometry and relying on encoders and visual information from a monocular camera.

Given these working conditions, the most direct approach would be to steer the robot toward regions of maximum clearance from image obstacles. This, however, does not guarantee avoidance of obstacles in the workspace unless the robot footprint is somehow taken into account. In some situations, for example, crossing of narrow passages by a humanoid is only possible if the robot body is oriented so as to reduce the lateral encumbrance.

We propose in this work an approach that allows the navigation of humanoids in a priori unknown environments, populated by obstacles of any shape, possibly with narrow passages, that require appropriate robot body reorientation.

The main ingredients of our approach are: (i) processing of visual information to distinguish the plane where the robot can safely walk from the obstacles' images; (ii) visual servoing techniques to regulate a suitable visual feature to a desired value which maximizes the averaged distance of

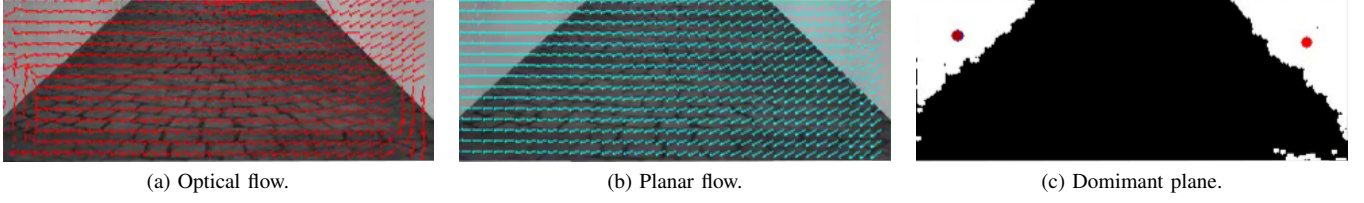(a) Optical flow.   (b) Planar flow.   (c) Dominant plane.

Fig. 1.   Main steps of the image processing algorithm.

obstacles images to the image plane ordinate axis; (iii) omnidirectional walking to cross narrow passages.

To allow proper orientation of the robot body in the workspace while performing the visual task we control the camera orientation with respect to the robot body and enforce a nonholonomic constraint on the camera motion that aligns the direction of motion with the robot gaze. In other words, the robot always moves toward the direction it is looking at.

Note that maximizing the distance of obstacles to the image center does not imply that the robot is moving with the maximum clearance to all the workspace obstacles. However, in case of navigation along a corridor, convergence to the corridor bisector can be formally proven, similarly to [7]. With respect to [7], the approach proposed here does not rely on geometric assumptions about the environment and it allows for the presence of obstacles of any shape and uses omnidirectional walk to cross narrow passages.

III. PERCEPTION

The perception algorithm extracts from the camera frames the information necessary to compute the visual feature used to control the robot navigation, as described in Sec. IV. In particular, pixels belonging to a dominant plane are discriminated from those that are outside this plane. Under the assumption that the ground plane occupies at least 50% of the image, all pixels outside this dominant plane are considered as obstacles.

The main steps of the adopted algorithm, fully described in [10], are shown in Fig. 1 and summarized here. At each instant of time t, compute (i) the optical flow $\boldsymbol{f}_o(u,v,t)$, and (ii) the planar flow $\boldsymbol{f}_p(u,v,t)$ for each pixel with coordinates $(u,v)^T$ belonging to a ROI of the original image. The optical flow (red arrows in Fig. 1a) is the apparent motion of the features. Instead, the planar flow, shown with cyan arrows in Fig. 1b, and computed using a RANSAC-based method, is the motion of the features obtained by assuming that they belong to the same dominant plane (the ground).

The thresholded difference between $\boldsymbol{f}_o$ and $\boldsymbol{f}_p$

$$\boldsymbol{d}(u,v,t) = \begin{cases} 0, & \text{if } \|\boldsymbol{f}_o(u,v,t) - \boldsymbol{f}_p(u,v,t)\| < \epsilon, \\ 1, & \text{otherwise,} \end{cases}$$

$$(1)$$

determines the dominant plane, and is represented as a binary image (black pixels correspond to the ground). The features satisfying the first condition in (1) are considered inliers for the next iteration of the RANSAC-based method, to robustly estimate the homography underlying the computation of the planar flow. Erosion and dilation operators are further applied to $\boldsymbol{d}$, to remove salt-and-pepper noise and obtain the final image in Fig. 1c.

Pixels outside the dominant plane belonging to a same connected region form image obstacles $\mathcal{O}_i$. For each of these obstacles we compute the centroid abscissa x_{i_c} in normalized image coordinates. Then, we take the average of the centroids abscissae on the left, and on the right, respectively as:

$$x_{l_c} = \frac{1}{n_l} \sum_{i=1}^{n_l} x_{i_c}, \; x_{i_c} < 0$$
$$x_{r_c} = \frac{1}{n_r} \sum_{i=1}^{n_r} x_{i_c}, \; x_{i_c} > 0, \tag{2}$$

where n_l (res. n_r) is the number of obstacle centroids with negative (res. positive) abscissa. If no obstacle is detected or present on the image plane, the value of x_{l_c} (x_{r_c}) is set equal to $-w_i/2$ ($w_i/2$), where w_i is the width of the image plane. A low-pass frequency filter eliminates noise and sway motion effect on x_{l_c} and x_{r_c} (shown as red dots in Fig. 1c).

Finally, the visual feature used to control the robot motion is chosen as the middle point between x_{l_c} and x_{r_c}:

$$x_m = \frac{1}{2}(x_{l_c} + x_{r_c}). \tag{3}$$

Figure 2 illustrates the position of the obstacle centroids and of x_m in the example case of a corridor with an obstacle.

IV. VISION-BASED CONTROL

Consider a humanoid robot, like NAO, as depicted in Fig. 3. The frames of interest for our work are: (i) $\mathcal{F}_w$, a world inertial frame; (ii) the robot frame $\mathcal{F}_r$, placed in the middle of the robot feet; (iii) the camera frame $\mathcal{F}_c$, with origin in the optical center of the camera and the z-axis aligned with the focal axis. The camera pan angle is q_p (positive clockwise, see Fig. 3), and the orientation of the robot frame with respect to the world frame is denoted with θ. Hence, the projection of the camera pose on the ground is defined by $(x, y, \theta_c)^T$, with x and y the optical center world frame coordinates, and $\theta_c = \theta + q_p$ the orientation of the optical axis with respect to the world frame axis y_w.

The cartesian task consists of making the humanoid safely walk among obstacles (for example, the walls of a corridor). This may require that the robot body properly aligns with the direction of maximum clearance to the obstacles to allow crossing of narrow passages without collisions, while the robot gaze and the linear velocity direction are kept aligned.

To achieve this task, we first determine in Sect. IV-A the relationship between the velocity of the visual feature x_m and the robot velocity input available for control. Then, we derive a controller in Sect, IV-B that, using this visual information, steers the robot among the environment obstacles. In particular, the defined control law guarantees exponential convergence of the camera frame to the center of a corridor with the optical axis aligned with the corridor bisector. Based
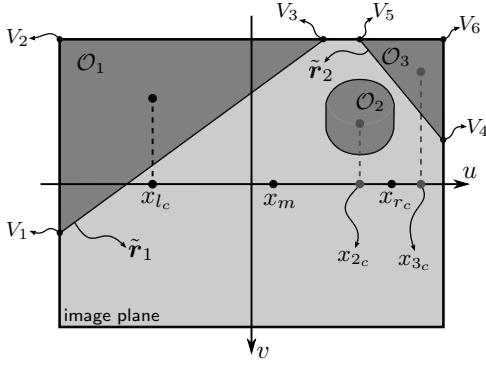
Fig. 2. Example of image plane obstacles during robot navigation.

(a) Top view   (b) Side view

Fig. 3. NAO frames and variables of interest to the navigation algorithm.

on the size of the traversable space, the robot body is oriented so as to reduce its transversal encumbrance. This behaviour is obtained by exploiting the humanoid omnidirectional walk allowing robot velocity directions outside its sagittal plane.

A. Dynamics of the centroid abscissa

The dynamics of the middle point x_m can be related to $\boldsymbol{v}_c$, the 6-D velocity of the camera, expressed in the camera frame, by the 1×6 row interaction matrix $\boldsymbol{L}_m$ [11]:

$$\dot{x}_m = \boldsymbol{L}_m \boldsymbol{v}_c. \tag{4}$$

To derive $\boldsymbol{L}_m$, we must first compute the interaction matrix $\boldsymbol{L}_{i_s}$ of a generic obstacle centroid abscissa x_{i_s} ($s = \{l, r\}$, depending on the side of the image that is considered). This can be obtained using image moments [12]:

$$\boldsymbol{L}_{i_s} = \frac{1}{N_{i,s}} \sum_{j=1}^{N_{i,s}} \left(-\frac{1}{z_{c,j}} \quad 0 \quad \frac{x_j}{z_{c,j}} \quad x_j y_j \quad -1-x_j^2 \quad y_j \right), \tag{5}$$

with $N_{i,s}$ the obstacle area in pixels, (x_j, y_j) the normalized image coordinates of the obstacle points and $z_{c,j}$ the depth of the j-th point in the camera frame. Ideally, to obtain a good approximation of $\boldsymbol{L}_m$, all the $z_{c,j}$ should be precisely estimated. However, this requires a 3D reconstruction algorithm that is out of scope in this work. Instead, here, we have decided to overestimate the pixel depths, by projecting them on the ground plane. To this end, the ground plane position in the camera frame is derived from the robot proprioception. In the experimental section, we will show that this approximation provides satisfactory results. A second approximation underlying (5), is that the obstacle is always entirely visible. In fact, when this is not the case, the visible centroid will not always correspond to the 3D centroid of the same object. However, when the dynamics of both camera and environment are slow with respect to the control rate (as in this case), this approximation will not affect the system performance.

To get $\boldsymbol{L}_m$ from the $\boldsymbol{L}_{i_s}$, we plug (2) into (3) and derive:

$$\boldsymbol{L}_m = \frac{1}{2} \left(\frac{1}{n_l} \sum_{i_l=1}^{n_l} \boldsymbol{L}_{i_l} + \frac{1}{n_r} \sum_{i_r=1}^{n_r} \boldsymbol{L}_{i_r} \right), \tag{6}$$

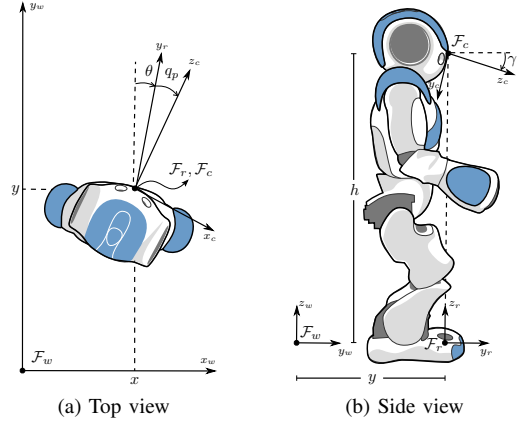with indices i_l and i_r accounting respectively for left and right centroids.

The dynamics of the visual feature (4) can now be related to the robot 6-D velocity $\boldsymbol{v}_r = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^T$ via

$$\dot{x}_m = \boldsymbol{L}_m \, {}^c\boldsymbol{W}_r \, \boldsymbol{v}_r = \boldsymbol{J} \boldsymbol{v}_r, \tag{7}$$

with ${}^c\boldsymbol{W}_r$ the twist velocity transformation matrix:

$$ {}^c\boldsymbol{W}_r = \begin{pmatrix} {}^c\boldsymbol{R}_r & {}^c\boldsymbol{S}_r \, {}^c\boldsymbol{R}_r \\ 0 & {}^c\boldsymbol{R}_r \end{pmatrix}, \tag{8}$$

${}^c\boldsymbol{R}_r$ the rotation matrix expressing the robot frame with respect to the camera frame, and ${}^c\boldsymbol{S}_r$ the skew matrix associated to the translation of the robot frame with respect to the camera frame. Both ${}^c\boldsymbol{R}_r$ and ${}^c\boldsymbol{S}_r$ can be deduced from the robot proprioception.

Assuming a planar motion of robot and camera and filtering out the lateral motion due to walking (sway motion)[1], we can consider v_x and v_y as the the component of the averaged velocity of the robot frame $\mathcal{F}_r$ in the plane. Together with ω_z this is a velocity input usually available for control in most humanoid robots. In addition, to allow independent alignment of optical axis and robot sagittal plane, we consider as control input also the neck joint yaw velocity $\dot{q}_p$ which controls the pan angle.

In summary, the robot velocity inputs available for control affect the middle point dynamics as follows:

$$\dot{x}_m = J_{v_x}^x v_x + J_{v_y}^x v_y + J_{\omega_z}^x \omega_z + J_p \dot{q}_p \tag{9}$$

with $J_{v_x}^x$, $J_{v_y}^x$, and $J_{\omega_z}^x$ the first, second and sixth entry of the row matrix $\boldsymbol{J}$. Note that the effect of $\dot{q}_p$ on the feature velocity is identical to that of ω_z under the assumption that camera pan and robot rotation axes are aligned. Taking this assumption, we consider $J_p = J_{\omega_z}^x$ in the derivations that follow.

B. Controller

1) Definition of the error: The objective of this section is to define a controller that orients the robot body along the direction of maximum clearance to the environment obstacles, while keeping the robot velocity direction aligned with its gaze, i.e., moving the robot toward the traversable space perceived through visual information.

---

[1]The effect of camera motion due to walking on the visual feature dynamics is filtered at the image processing level, as mentioned in Sect. III.

This behavior is obtained by regulating q_p, the relative angle between the head and the robot sagittal plane, to a desired value q_p^* and a properly defined visual error to zero. In defining the visual error a reasonable choice would be to keep the obstacles symmetrical with respect to the vertical axis of the image plane. The error vector is then defined as $\boldsymbol{e} = (x_m, \ q_p - q_p^*)^T$, where x_m is defined by (3).

The following proposition states the relationship between the visual error regulation and the robot workspace paths with reference to a corridor navigation scenario.

Proposition 1: Assuming navigation along a straight corridor without obstacles, regulation of the visual feature x_m to zero implies convergence of the camera position and orientation along the corridor bisector.

Proof: Placing the origin of the world reference frame $\mathcal{F}_w$ at the center of the corridor, the proposition claim is formally expressed through the following implication

$$x_m \to 0 \Rightarrow (x, \theta_c)^T \to (0,0)^T, \qquad (10)$$

where $\theta_c = \theta + q_p$ is the orientation of the optical axis of the camera with respect to the corridor bisector.

The corridor guidelines, i.e., the lines at the intersection between the floor and the walls, are represented through the world frame homogeneous coordinates of two points belonging to each line:

$$\tilde{\boldsymbol{r}}_1 = (-1, 0, -d)^T \ \text{and} \ \tilde{\boldsymbol{r}}_2 = (-1, 0, d)^T$$

where d is the semidistance between the two guidelines. These lines project on the image plane as:

$$\hat{\boldsymbol{r}}_1 = \boldsymbol{P}^{-T}\tilde{\boldsymbol{r}}_1 = (a_1, b_1, c_1)^T, \quad \hat{\boldsymbol{r}}_2 = \boldsymbol{P}^{-T}\tilde{\boldsymbol{r}}_1 = (a_2, b_2, c_2)^T$$

where $\boldsymbol{P}$ is the projection matrix relating 3-D points of the cartesian space with 2-D points of the image plane [13]:

$$\boldsymbol{P} = \begin{pmatrix} S_u c_{\theta_c} & -S_u s_{\theta_c} & -S_u x c_{\theta_c} \\ -S_v s_\gamma s_{\theta_c} & -S_v c_{\theta_c} s_\gamma & S_v(hc_\gamma + x s_\gamma s_{\theta_c}) \\ c_\gamma s_{\theta_c} & c_\gamma c_{\theta_c} & h s_\gamma - x c_\gamma s_{\theta_c} \end{pmatrix}. \qquad (11)$$

with γ the camera tilt angle around x_c (for their sign, refer to Fig. 3) and S_u, S_v the focal lengths in pixel along the image plane axes. For the sake of brevity, in (11) we have used $c_{(\cdot)}$ and $s_{(\cdot)}$ as abbreviations of $\cos(\cdot)$ and $\sin(\cdot)$, respectively. The ordinate y of camera position has been omitted since it is irrelevant for the considered control problem (i.e., only the distance to the bisector is controlled and not the position along the corridor as the robot goal is to move forward). For details on the computation of $\boldsymbol{P}$, refer to [7].

Under the considered working conditions, the corridor walls appear on the image plane as two triangles (see in Fig. 2), with vertexes $\boldsymbol{V}_i$, $i = 1, \ldots, 3$ and $\boldsymbol{V}_j$, $j = 4, \ldots, 6$. Computed x_{l_c} and x_{r_c} respectively as the averaged abscissa of each of the two triangles, using the equation of the projected lines, we have:

$$x_m = \frac{1}{2}(x_{l_c} + x_{r_c}) = k_1 \frac{x}{c_{\theta_c}} + k_2 \tan\theta_c, \qquad (12)$$

with $k_1 = S_u/6h(h_i c_\gamma/S_v - 2s_\gamma)$, $k_2 = -S_u/6h(h_i s_\gamma/S_v + 2c_\gamma)$ constants depending on the camera intrinsic and extrinsic parameters, and on the height of the image plane h_i.
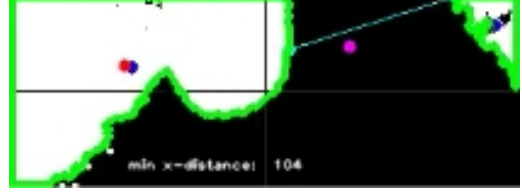

Fig. 4. Distance computation between left and right obstacles. Green curves highlight the contours of the detected obstacles. Blue dots are the corresponding centroids, red dots are the filtered centroids with abscissae x_{l_c} and x_{r_c}. The purple dot is the middle point with abscissa x_m. The horizontal component $d_{x,min}$ of the minimum distance (cyan line) between left and right obstacles is computed on each frame to automatically generate the reference value q_p^*.

From (12) it follows that $x_m = 0$ implies $x = -k_2/k_1 s_{\theta_c}$. Of the locus of points described by this last equation only $(x, \theta_c) = (0, 0)$ is a stable equilibrium point. In fact, the visual task $x_m = 0$ is satisfied at all times only if the camera is aligned with the corridor bisector due to the enforced mobility model of the camera that keeps the optical axis always aligned with the direction of motion (see eq. (15) in next Sect. IV-B.2). In other words, if the robot does not move along the bisector the camera will move toward one of the two corridor walls. This will perturb the symmetric position of the obstacles centroids in the image plane provided that

$$k_2/k_1 > 0 \Rightarrow \tan\gamma > \frac{h_i}{2S_v}, \qquad (13)$$

i.e., the robot camera is sufficiently tilted toward the floor, a condition satisfied by our operational setup. Equation (13) is derived from geometric reasoning, omitted here for brevity. $\square$

2) Modeling the error dynamics: Using eq. (9), the error dynamics can be written as

$$\begin{cases} \dot{x}_m = J_{v_x}^x v_x + J_{v_y}^x v_y + J_{\omega_z}^x \omega_z + J_p^x u_p, \\ \dot{q}_p = u_p. \end{cases} \qquad (14)$$

Aligning the robot linear velocity with the robot gaze:

$$\begin{aligned} v_x &= v \sin q_p \\ v_y &= v \cos q_p \end{aligned} \qquad (15)$$

and assigning to the robot speed $v = \sqrt{v_x^2 + v_y^2}$ a positive constant value, the remaining inputs available for the visual task and the regulation of the camera orientation with respect to the robot are ω_z and u_p.

A simple control guaranteeing exponential convergence of the error to zero is:

$$\begin{aligned} \omega_z &= (J_{\omega_z}^x)^{-1}(-k_m x_m - J_{v_x}^x v_x - J_{v_y}^x v_y) - u_p \\ u_p &= -k_p(q_p - q_p^*), \end{aligned} \qquad (16)$$

where we have set $J_p = J_{\omega_z}^x$ and used k_m and k_p to denote positive control gains. The analisys of the singularities of the above control law is quite complicated and is not presented in this work. However, during validation the algorithm never incurred in a singularity. Future work includes this analysis.

Note that eq. (15) represents the robot velocity in the robot frame and is aligned with the camera optical axis. With respect to the world reference frame we have that the velocity
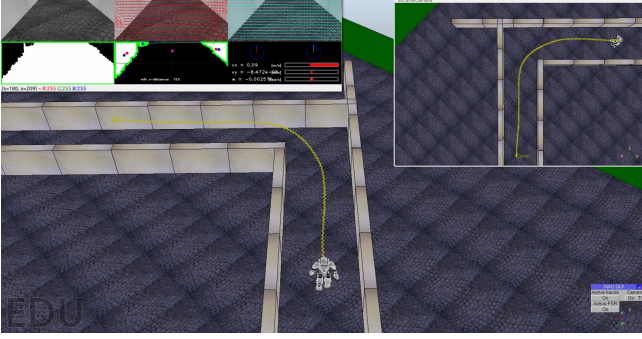
Fig. 5. Snapshot from the simulation of a curve negotiation.

vector is oriented as $\theta_c = \theta + q_p$ resulting in the following (averaged) planar motion of the camera

$$\begin{aligned} \dot{x} &= v \sin \theta_c \\ \dot{y} &= v \cos \theta_c \end{aligned} \qquad (17)$$

according to which the camera orientation is always tangent to its cartesian path (with sway motion filtered out). This is the argument used in the proof of Proposition 1.

V. SETUP AND VALIDATION

For the validation of the proposed navigation algorithm, we used the small humanoid NAO. This robot is equipped with two cameras, mounted on the top and bottom of the forehead. Since the robot shoulder could occlude the field of view of the bottom camera when NAO turns the head, we chose to use the top camera. The tilt angle was set to the constant $\gamma = 0.22$ rad. This angle allows a proper view on the scene, and also free motion of the pan angle q_p without risking collisions between the head and the shoulders. The images were acquired with a resolution of 320×240 pixels. The image processing described in Sect. III were performed on a ROI of 320×120 pixels placed at the bottom of the image. The optical flow has been computed through the Lucas-Kanade implementation available in the OpenCV open-source library. NAO is also equipped with a built-in walking engine by which it is possible to send to the robot the linear and angular velocity command. The desired pan angle of the head q_p^* has been set according to horizontal component of the minimum distance $d_{x,min}$ (see Fig. 4) between the images of left and right obstacles:

$$q_p^* = \begin{cases} q_{p,max}, & \text{if } d_{x,min} < t_l \\ 0, & \text{if } d_{x,min} > t_h \\ q_{p,max} - \left(\frac{d_{x,min}-t_l}{t_h-t_l}\right) * q_{p,max}, & \text{if } t_l < d_{x,min} < t_h \end{cases} \qquad (18)$$

where $t_l = \left\{ t_l^-, t_l^+ \right\}$ and $t_h = \left\{ t_h^-, t_h^+ \right\}$, experimentally found, are evaluated through an hysteresis thresholding, to get a measurement of $d_{x,min}$ robust to the image noise, and such that $t_l < t_h$.

In order to validate the method, we performed both simulations, using NAO model in the V-REP environment, and experiments with the real robot.

A. Simulations

A maze-like environment has been built tailored to the size of NAO. In all the presented simulations, the robot speed and
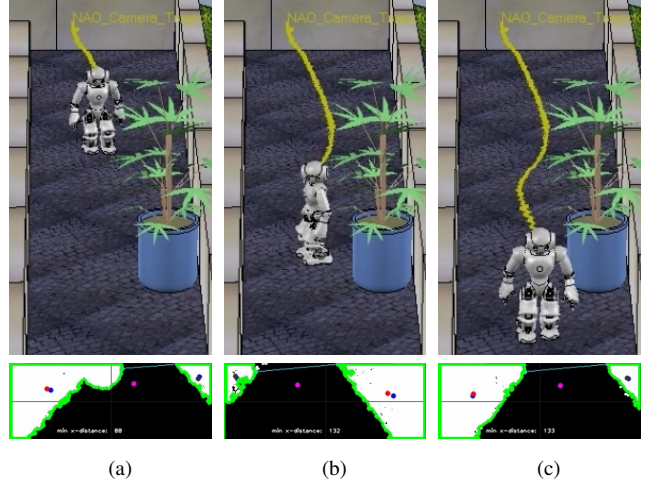

(a)   (b)   (c)

Fig. 6. NAO navigates along a straight corridor and avoids an obstacle by walking sideways.

the controller gains values are respectively $v = 0.0952$ m/s, $k_m = 0.4$, $k_p = 0.05$, while in the estimation of the dominant plane it was set $\epsilon = 0.15$.

With a first set of simulations, we validated the effectiveness of the approach to make NAO navigate at the center of (not necessarily straight) corridors, negotiate a curve and turning a T-junction. In these cases, no obstacles other than the walls are considered, and the navigation algorithm is run with the camera aligned with the torso ($q_p^* = 0$).

Of the four simulations documented in the video clip accompanying the paper, we report here snapshots of two significant cases: curve negotiation and avoidance of obstacles in a corridor.

Figure 5 reports a snapshot from the simulation environment. In this snapshot, the robot is at the end of a curve negotiation task execution. The path followed by the robot is reported in yellow. In the upper left corner the output of the image processing and critical control signals for online monitoring. In the upper right corner a top view of the scene. As can be appreciated from the video clip accompanying the paper, NAO first detects only the corridor walls as obstacles and converges to the corridor center. When it is close to the curve one of the two walls disappears from the image, the front wall enters in the camera view and the resulting single obstacle has a centroid shifted on the right, causing the robot to turn in the right direction.

The accompanying video also shows the robot turning at a T-junction. Note that, differently from [7], no logic is used to classify the visited places (i.e., straight corridor, curve, T-junction,...) since the robot motion only depends on the position of the image obstacles. At the T-junction a direction of motion could of course be commanded if a specific task would require the robot to move in a preferred direction.

In the second simulation, we tested the effectiveness of the navigation algorithm when obstacles other than a corridor walls are present in the environment. In particular, we put a plant in the middle of a corridor with non-parallel walls, to create a narrow passage that the robot has to cross by maximizing the clearance to the obstacles. The reference pan angle q_p^* is computed in accordance to eq. (18), where we
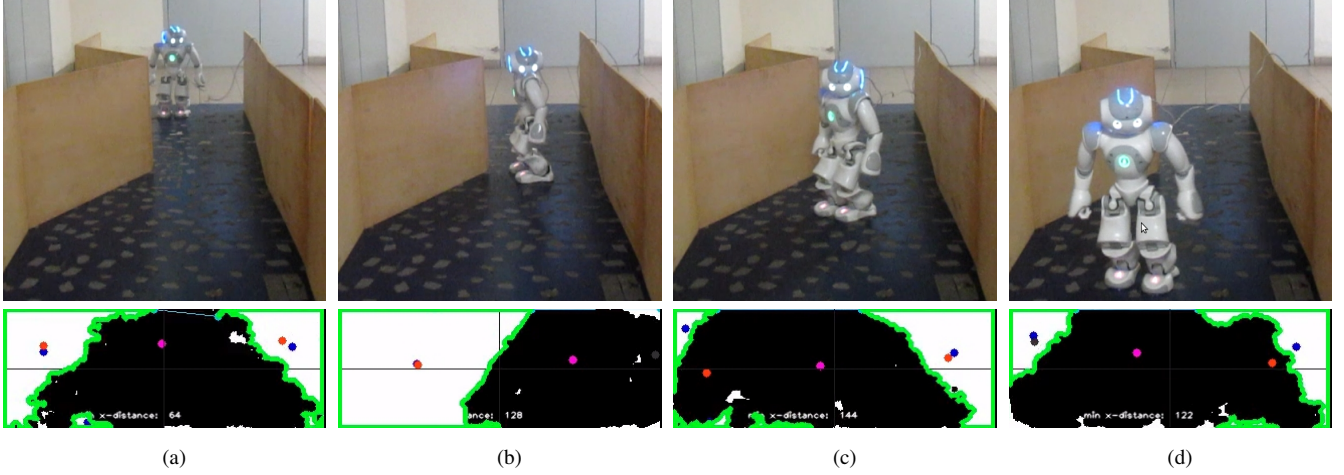
Fig. 7. Experiment with a real NAO. Top: snapshots of the robot moving in the corridor. Bottom: corresponding image procesisng. In the first snapshot a narrow passage is detected and used to set the desired value of q_p^* for camera rotation. When the camera orientation is equal to the desired value, the robot reaches the passage, as shown in the second snapshot. The third and the fourth snapshots show the robot re-aligning the camera with the sagittal plane, in correspondence of a wider free space.

set $t_l = \{90, 130\}$, $t_r = \{110, 140\}$. Snapshots of this last simulation are shown in Fig. 6. The robot starts walking off the corridor center, but it quickly converges to its bisector. Next, when the plant becomes visible in the image plane, the minimum distance $d_{x,min}$ between left and right obstacles decreases and drops below t_l.

The camera changes orientation with respect to the torso of the robot, that rotates to reduce the visual error, according the control law (16). This induces the robot to walk sideways reducing the lateral encumbrance while traversing the narrow passage. When the obstacle disappears from the image, the minimum distance $d_{x,min}$ exceeds t_h and the robot aligns the camera again with the torso, restoring the standard walking.

B. Experiments

Our approach has been validated on a real NAO in an environment similar to the virtual scenarios and adapted to its size. The parameter values used in the experiment are $v = 0.07$ m/s, $k_w = 0.5$, $k_p = 0.05$, $t_l = \{110, 120\}$, $t_h = \{125, 130\}$. Snapshots of the experiment and the corresponding processed images are shown in Fig. 7. A narrow passage is created at the beginning of the corridor, so the camera rotates to the desired value $q_p^* = 90°$ when the robot begins to move. The direction of the gaze determines the direction of the driving velocity, so the robot moves sideways when the walls are closer. When the space between walls becomes wider the camera rotates to reach the desired value of $q_p^* = 0°$ and realigns with the robot sagittal plane.

VI. Conclusions

In this paper we addressed the problem of humanoid navigation in environments populated by obstacles. The proposed approach is effective for humanoid navigation in corridors with curves and T-junctions, and obstacles placed along the corridors. The method uses optical flow information to reconstruct a local representation of the obstacles. A visual controller provides the robot velocity command for safe navigation, and the angle command to turn the body for passing through narrow passages. To validate the approach,

we performed both simulation and experiments with the humanoid robot NAO. Future work will aim at: improving the image processing to make it more robust to environmental conditions, analysing the robustness of the approach with respect to actuation inaccuracies and unmodeled phenomena induced by impact dynamics, extension to non-flat terrain and dynamic obstacles.

References

[1] F. Faber, M. Bennewitz, C. Eppner, A. Görög, C. Gonsior, D. Joho, M. Schreiber, and S. Behnke, "The humanoid museum tour guide robotinho," in *The 18th IEEE Int. Symp. on Robot and Human Interactive Communication, RO-MAN 2009*.

[2] D. Maier, C. Stachniss, and M. Bennewitz, "Vision-based humanoid navigation using self-supervised obstacle detection," *Int. Journal of Humanoid Robotics*, vol. 10, no. 2, 2013.

[3] M. Garcia, O. Stasse, J.-B. Hayet, C. Dune, C. Esteves, and J.-P. Laumond, "Vision-guided motion primitives for humanoid reactive walking: decoupled vs. coupled approaches," *Int. Journal of Robotics Research*, vol. 34, no. 4-5, pp. 402–419, 2014.

[4] D. Petit, P. Gergondet, A. Cherubini, and A. Kheddar, "An integrated framework for humanoid embodiment with a BCI," in *IEEE Int. Conf. on Robotics and Automation, ICRA 2015*.

[5] J. Ido, Y. Shimizu, Y. Matsumoto, and T. Ogasawara, "Indoor navigation for a humanoid robot using a view sequence," *The Int. Journal of Robotics Research*, vol. 28, no. 2, pp. 315–325, 2009.

[6] Y. Matsumoto, M. Inaba, and H. Inoue, "View-based navigation using an omniview sequence in a corridor environment," *Machine Vision and Applications*, vol. 14, no. 2, pp. 121–128, 2003.

[7] A. Paolillo, A. Faragasso, G. Oriolo, and M. Vendittelli, "Vision-based maze navigation for humanoid robots," *Autonomous Robots*, 2016.

[8] J. Delgado-Galvan, A. Navarro-Ramirez, J. Nunez-Varela, C. Puente-Montejano, and F. Martinez-Perez, "Vision-based humanoid robot navigation in a featureless environment," in *Pattern Recognition*. Springer, 2015, pp. 169–178.

[9] G. Arechavaleta, J. P. Laumond, H. Hicheur, and A. Berthoz, "The nonholonomic nature of human locomotion: a modeling study," in *The First IEEE/RAS-EMBS Int. Conf. on Biomedical Robotics and Biomechatronics, BioRob 2006.*, 2006, pp. 158–163.

[10] N. Ohnishi and A. Imiya, "Appearance-based navigation and homing for autonomous mobile robot," *Image and Vision Computing*, vol. 31, no. 6, pp. 511–532, 2013.

[11] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Robot. Aut. Mag.*, vol. 13, no. 4, pp. 82–90, 2006.

[12] O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects," *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1116–1127, 2005.

[13] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2003.