HAL
open science

A visual servoing approach for autonomous corridor following and doorway passing in a wheelchair

François Pasteau, Vishnu Karakkat Narayanan, Marie Babel, François Chaumette

To cite this version:

François Pasteau, Vishnu Karakkat Narayanan, Marie Babel, François Chaumette. A visual servoing approach for autonomous corridor following and doorway passing in a wheelchair. Robotics and Autonomous Systems, 2016, 75, part A, pp.28-40. 10.1016/j.robot.2014.10.017 . hal-01068163v2

HAL Id: hal-01068163

https://inria.hal.science/hal-01068163v2

Submitted on 27 Nov 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A visual servoing approach for autonomous corridor following and doorway passing in a wheelchair

François Pasteau[a], Vishnu K.Narayanan[b], Marie Babel[a], François Chaumette[b]

[a]*INSA Rennes,Irisa and Inria Rennes, France*
[b]*Inria, Irisa and Inria Rennes, France*

Abstract

Navigating within an unknown indoor environment using an electric wheelchair is a challenging task, especially if the user suffers from severe disabilities. In order to reduce fatigability and increase autonomy, control architectures have to be designed that would assist users in wheelchair navigation. We present a framework for vision-based autonomous indoor navigation in an electric wheelchair capable of following corridors, and passing through open doorways using a single doorpost. Visual features extracted from cameras on board the wheelchair are used as inputs for image based controllers built-in the wheelchair. It has to be noted that no a-priori information is utilised except for the assumption that the wheelchair moves in a typical indoor environment while the system is coarsely calibrated. The designed control schemes have been implemented onto a robotized wheelchair and experimental results show the robust behaviour of the designed system.

Keywords: Visual servoing, Wheelchair navigation, Assistive robotics, Vision-based robotics

1. Introduction

Independent mobility for wheelchair users is a necessary part of the social and physical well being. While mobility increases social opportunity and reduces dependence, numerous studies have shown that a decrease in mobility can lead to social disorders like depression, anxiety and a fear of abandonment [1] [2]. However utilizing an electric wheelchair, especially in constrained environments, can be a challenging task for disabled individuals. In the case of severe disability, it is almost impossible for a user to effectively navigate without the help of another person. The issue of security is also an important one in these cases where for example, a person with neuromuscular disabilities finds it difficult to steer a wheelchair, particularly in limited space environments, without the risk of collision [3]. In addition, it was reported that *accessibility* was the main problem encountered by wheelchair users [4] when more than four fifths of mobility device users in the United States identified themselves as living in a house with all the rooms on the same level.

Recent advances in robotics have paved way for a variety of opportunities facilitating the design of smart assistive technologies in order to help improve the lives of disabled people [5]. Keeping pace these advances in robotics, an ordinary powered wheelchair can be enhanced into an "Intelligent wheelchair" using sensors that perceive the surroundings and control architectures that affect the motion of the wheelchair.

Extensive works have previously been reported in this area, mainly focussing on the issue of increasing the safety levels of the wheelchair (and thereby decreasing the amount of human intervention). In all these systems, the human controls the higher level intelligence whereas the wheelchair is responsible for the lower level tasks. For example one can mention the TAO Project [6], NavChair [7], European FP7 Radhar project [8] and the recent SYSIASS project [9] all of which are designed to provide assistance in tasks like navigating through indoor environments like corridors, basic obstacle detection/avoidance and also, in some cases, autonomous passage through narrow doorways.

We observe that such behaviour-based design approaches have been proven to lead to robust systems. However, it has to be noted that the majority of the previous works in this area use a multi-sensor system with a distributed

Email address: `Marie.Babel@irisa.fr` (Marie Babel)

architecture. The motivation of our work comes partly from the fact that low cost sensors such as vision sensors could be employed in designing such systems. This can ensure widespread usage taking into account the fact that reimbursement from government/healthcare companies typically only covers the wheelchair, and not its technological adaptations. But relevant control systems have to be designed around a vision-based paradigm which still remains a highly challenging task. Thus, in this work, we aim to develop a solution for autonomous wheelchair navigation in an indoor environment using a set of monocular cameras with each one being devoted to a particular task.

That being said, the aim of the work is to serve as a first step in designing assistive systems (particularly low-cost vision-based systems) that may have a human in the control loop. Such semi-autonomous wheelchair systems would be helpful in conditions where motor impairments may hinder secure and effective navigation, by providing safe assistance in performing fundamental tasks while giving high level control to the user. As such, we have tackled two important low-level tasks related to indoor navigation: autonomously keeping a stable position along corridors in a secure way and detecting and passing through doorways. We will see that this second task can also be used directly for turning from a corridor to another one. These tasks represent the fundamental capability of a wheelchair performing indoor navigation.

While specifically considering corridor following using a wheeled robot, in [10] a vision-based approach was designed using an omni-directional camera along with a visual memory framework. Whereas in [11] a stable control system was designed for corridor following and wall-following for sonar-based mobile robots (which may be extended to a wheelchair platform). Furthermore, in [12] two vision-based control algorithms for corridor navigation were presented that exploited the geometry of a typical corridor. The first one uses the optical flow measured from the corridor's lateral walls to generate an angular velocity command for the mobile robot. The second scheme finds the perspective lines of the walls meeting the floor to generate the angular velocity command for the robot. Therefore, the geometry of a corridor in an image is a viable solution to employ while designing vision-based algorithms around it for realising the task of corridor following.

Several results have also been reported in the area of doorway passing with mobile/wheeled robots. A multi-sensor based algorithm for guiding a non-holonomic platform through a doorway was presented in [13]. The controller uses information from a camera system and a laser range finder for navigation. In the case of the previously mentioned SYSIASS project, a laser range finder with a PID controller is employed for doorway passing with a mobile robot. Another solution for secure navigation of a mobile robot in particular for doorway passing was proposed by [14]. The framework exploited a laser-based hybrid control scheme with dynamic obstacle avoidance and path planning. It has to be noted that a purely vision-based solution for doorway passing has not been proposed in literature and is therefore one of our major contributions.

Another important constraint arises when considering that the wheelchair must be able to perform in an indoor environment without any a-priori knowledge of the environment. Consequently, a solution where the geometry of the environment and the metrical data are not known is expected. Therefore we apply an Image-based Visual Servoing (IBVS) scheme which controls the relevant degrees of freedom (dof) of the wheelchair based on visual features directly extracted from a camera on board the wheelchair [15].

The initial objective in any visual servoing scheme is to select features that represent visual data which can be effectively exploited to perform the task at hand. With respect to the task of autonomously following a corridor, features like the vanishing point and the straight lines that represent the edges of the corridors are relevant [16] [17]. There are several studies which indicate the robustness of the aforementioned features [18]. Consequently to design a stable control law that achieves the objective robustly, we have developed a system that employs the measure of the vanishing point and the position of the vanishing line that corresponds to the median line of the corridor.

When dealing with the task of passing through doorways, the features available are mostly line features that represent the position of the doorposts in the image. This is a highly challenging task while employing a monocular vision sensor especially in the case where a single doorpost is visible. Thus, we have developed a novel and robust control scheme which only uses the position of the line feature corresponding to the nearest doorpost in the image to achieve the goal. This approach is utilized since global asymptotic stability [19] of the system can be demonstrated as will be shown in Section 4.3.2. Moreover, higher level constraints like the status of the door (open/closed), the decision making process of passing through the door (i.e. which door to choose from), etc, make the task non-trivial. We here focus on low level control without the help of a global planning framework. The high level attributes are left for the user/home automation system to set as they are not the objective of this work.

The developed schemes do not require any initialization step owing to the fact that a dedicated line detection

framework was designed to detect and track the relevant line features in real time. Thus, bottleneck stage that still remains in most of the servoing schemes is avoided. In addition, as mentioned before, no a-priori environment data is required or utilized in the servoing process.

The paper is organized as follows. Section 2 deals with the geometric modelling of the robotized wheelchair. Section 3 details the task of corridor following with the definition of the visual features used and the control law formulation. Section 4 explains the task of doorway passing and proposes a new Lyapunov-based controller. Finally Section 5 exposes the experiments done and the results obtained while implementing the designed framework on a robotized wheelchair.

2. Modelling

The wheelchair is modelled as a six wheeled robot moving on a horizontal/inclined plane. Two differentially actuated wheels located in the middle of the wheelchair provide motion along with four passive caster wheels with two each located in the front and back. The wheelchair can then be though of as a simple unicycle robot [20], thus matching non-holonomous constraints.

Owing to the actuated wheels, two variables can be controlled, namely the translational velocity v and the rotational velocity ω. In this study, the control variable is restricted to the rotational velocity ω while a constant forward velocity v^* is maintained for motion exigency. Three cameras located at different positions on the wheelchair are employed in order to realise the two different tasks. A front facing camera denoted as *camera 1* which is located on the left handle at a height h_1 from the floor is employed for the task of corridor following. The second and third cameras denoted as *camera 2* and *camera 3* are respectively located on the right and left handle at heights h_2 and h_3 from the floor. These cameras are used for realising doorway passing: the right one to pass doors located on the right wall and the left one for doors on the left wall. These cameras are oriented in order to ensure that the doorpost stays in the camera field of view during the servoing process. In this paper, we refrain the study to the camera located on the right side of the wheelchair as the solution for the left camera is straightforward and symmetric.

Now, according to Figure 1, the robot frame is defined as $\mathcal{F}_r(P_O, x_r, y_r, z_r)$. The origin of the frame $\mathcal{F}_r$ is chosen as the mid-point of the line segment joining the two centres of the differential wheels. The camera frames are defined namely as, $\mathcal{F}_{c_i}(C_i, x_{c_i}, y_{c_i}, z_{c_i}), i = 1, 3$ where C_i denotes the optical center of each camera. We finally denote the orientation of cameras i with respect to frame $\mathcal{F}_r$ as θ_i.

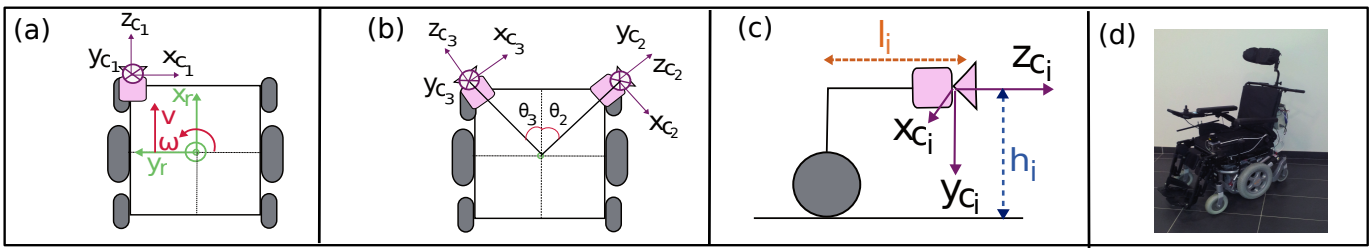


Figure 1: Robot and camera frame definitions (a) Top view with relative position of robot and camera 1 frames. (b) Top view with camera 2 and camera 3 frames. (c) Simplified side view of the camera frames. (d) Robotic platform.

Consequently, the rotation matrices relative to the robot frame $\mathcal{F}_r$ and the camera frames $\mathcal{F}_{c_i}$ are given by

$$^{c_i}\mathbf{R}_r = \begin{bmatrix} \sin\theta_i & -\cos\theta_i & 0 \\ 0 & 0 & -1 \\ \cos\theta_i & \sin\theta_i & 0 \end{bmatrix}. \tag{1}$$

By denoting $^f\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega}) = \left(^f v_x, ^f v_y, ^f v_z, ^f \omega_x, ^f \omega_y, ^f \omega_z\right)$ the velocity of a frame $\mathcal{F}_f$ expressed in $\mathcal{F}_f$, where the first three components $\boldsymbol{v}$ represent the translational velocities and the last three components $\boldsymbol{\omega}$ the rotational velocities, we have

$$^r\mathbf{v} = (v, 0, 0, 0, 0, \omega)\,. \tag{2}$$

The translation ${}^{\mathbf{r}}\mathbf{t}_{\mathbf{c_i}}$ between the robot frame and camera frames is given by $(l_i, w_i, 0)$. Note that for *camera 1* on the left, we have $\theta_1 = 0$, $l_1 > 0$ and $w_1 > 0$ while for *camera 2* on the right, $\theta_2 < 0$, $l_2 > 0$ and $w_2 < 0$.

By applying the well known formula

$$ {}^{c_i}\mathbf{v} = \begin{bmatrix} {}^{c_i}\mathbf{R}_r & [{}^{c_i}\mathbf{t}_r]_\times \, {}^{c_i}\mathbf{R}_r \\ 0 & {}^{c_i}\mathbf{R}_r \end{bmatrix} . {}^{r}\mathbf{v} \tag{3} $$

we obtain using (1)

$$ {}^{c_i}\mathbf{v} = \left({}^{c_i}v_x, 0, {}^{c_i}v_z, 0, {}^{c_i}\omega_y, 0 \right) \tag{4} $$

where

$$ \begin{cases} {}^{c_i}v_x = v \sin\theta_i - \omega(l_i \cos\theta_i + w_i \sin\theta_i) \\ {}^{c_i}v_z = v \cos\theta_i + \omega(l_i \sin\theta_i - w_i \cos\theta_i) \\ {}^{c_i}\omega_y = -\omega \end{cases} \tag{5} $$

3. Autonomous Corridor Following

As stated earlier, the task of autonomously following a corridor is modelled as an IBVS problem. The objective is to devise a control law which minimises the error between a set of observed visual features and the set of their desired values. A discriminative representation of the relevant features of a corridor is given in Figure 2 from which some of the visual features can be observed which could be exploited for realising the task. The desired system should perform the task of following a corridor without collision with the walls, that is to say the system should position the optical axis of camera parallel to the wall and at the middle of the corridor. To design such a task, two visual features are selected.

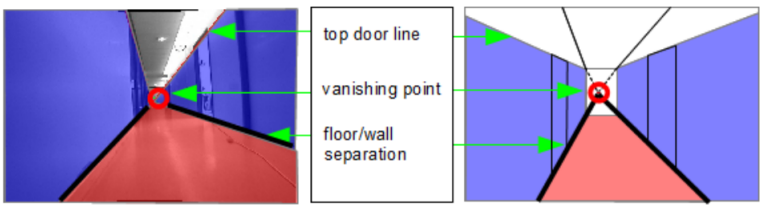


Figure 2: Corridor geometrical structure with floor and wall planes

3.1. Visual Features - Definition and Extraction

The x-coordinate x_f of the vanishing point $f_t = (x_f, y_f)$ at instant t (shown in Figure 3) is chosen as the first feature. When the robot is positioned parallel to the wall, looking forward, this feature goes to zero. As the second feature, the angle θ_m made by the z-axis of the camera with the median line of the corridor is chosen (see Figure 3). When the position of the camera is in the middle of the corridor, this feature is equal to zero.

Since the median line is not visible in the image, this feature is computed from the orientations θ_l and θ_r of the straight lines in the image related to the intersection of the walls and the floor of the corridor. The median of the corridor is projected onto the image as a straight line parametrized by (ρ_m, θ_m) where

$$ \rho_m = x_f \cos\theta_m + y_f \sin\theta_m. \tag{6} $$

Geometrically, in the 3D world, the lines related to the wall and the median of the corridor are parallel and coplanar. As a consequence, the corresponding lines in the image intersect at the vanishing point while respecting

$$ \theta_m = \arctan\left(\frac{\tan\theta_l + \tan\theta_r}{2} \right). \tag{7} $$

Also, the lateral distance between the camera and the median of the corridor is given by $y = h_1 \tan(\theta_m)$, where h_1 is the distance between the camera 1 optical centre and the floor (see Figure 1). Therefore, we get $y = 0$ when $\theta_m = 0$.

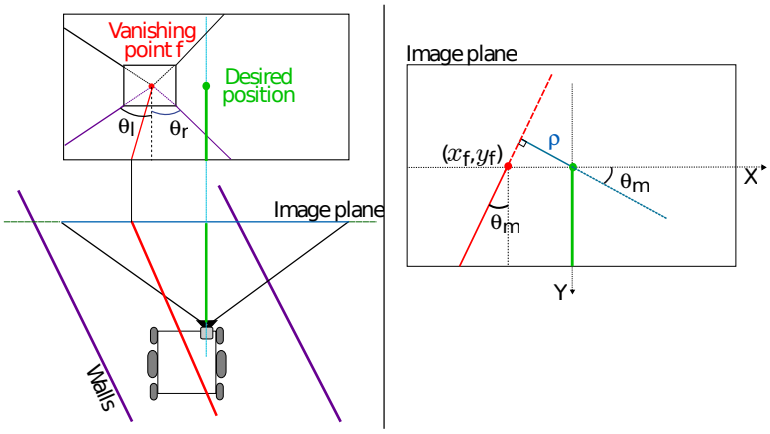


Figure 3: x_f and θ_m visual features

3.1.1. Estimation of x_f

Estimating the vanishing point robustly and in real time is still an open area of research in computer vision. In the case of navigating through a corridor, the vanishing point corresponds to the point where a significant number of straight lines may intersect. We use the classical Gaussian sphere projection framework [21] [22] to estimate the vanishing point. The idea is to project onto this sphere the set of detected non-vertical lines in the image. Then the vanishing point is chosen as the point which has accumulated the maximum number of votes.

For extracting the straight lines in the image, the Line Segment Detection (LSD) algorithm is used [23]. It is based on local gradient orientations in the image, from which major segments are detected. The detected segments are then classified into vertical lines and non-vertical lines. The latter is used for estimating the vanishing point.

However, the main issue of the LSD algorithm is the cutting off of one line when the direction of the observed gradient changes. Thus, to increase the robustness, a dedicated merging process is applied to a group of segments that can be considered as a single straight line. For two segments, this process is done by taking into account the slope and extremities, and if they are close enough, they are merged to form a unique line. More details are given in [24].

Since the vanishing point f_t is estimated at each frame during a sequence, a factor $\alpha \in]0, 1[$ is introduced for temporal filtering as

$$f_t = \alpha f_{t-1} + (1 - \alpha) f_t$$

in order to ensure a smooth variation of the resulting estimated vanishing point in the current frame. The value of α is empirically tuned and has been chosen for the experiments at $\alpha = 0.1$.

3.1.2. Estimation of θ_m

The angles θ_l and θ_r can be easily calculated if we have an accurate estimate of the wall/floor boundaries. A variety of techniques have been proposed in the area of wall/floor boundary detection. For example, wall/floor features are defined by the corners corresponding to the intersection of a vertical line and the floor plane in [25]. Whereas in [26], floor boundary is estimated by a dynamical Bayesian network model which is applied on each column of the image.

In our proposed scheme, a set of non-vertical lines are searched for along the image that correspond to the wall/floor boundary. The search is conducted based on two criteria: the first one being that they contribute to the vanishing point and, the second one being that they cross the bottom extremities of most number of vertical lines. In order to minimize false positives, a maximal distance between the vertical line extremity and the vanishing line is defined.

Then, from the angles θ_l and θ_r, the value of θ_m is directly obtained using equation 7.

3.2. Image-based visual servoing scheme

Visual servoing is a classical approach to control robot motion from visual data [15]. The design of the kinematics controller is based on modelling the variation $\dot{\mathbf{s}}$ of the visual features $\mathbf{s}$ with respect to the camera velocity $\mathbf{v}_c$. More

precisely, when the observed environment is static, we have

$$\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{v}_c \tag{8}$$

where $\mathbf{L_s}$ is named the interaction matrix. In our case when $\mathbf{s} = (x_f, \theta_m)$, the analytical form of $\mathbf{L_s}$ is given in [27] and [28] as

$$\mathbf{L_s} = \begin{bmatrix} 0 & 0 & 0 & x_f y_f & -1 - x_f^2 & y_f \\ \lambda_{\theta_m} & \lambda_{\theta_m} \sin\theta_m & -\lambda_{\theta_m}\rho_m & -\rho_m \cos\theta_m & -\rho_m \sin\theta_m & -1 \end{bmatrix}, \tag{9}$$

where $\lambda_{\theta_m} = (A \sin\theta_m - B\cos\theta_m)/D$ wherein $AX + BY + CZ + D = 0$ defines a plane to which the median line belongs. In the present case, this line belongs to the floor plane which is defined by $Y = h_1$. Therefore $A = C = 0$ and we obtain

$$\lambda_{\theta_m} = \cos\theta_m / h_1. \tag{10}$$

Using 4 in 8, we obtain the variation of the task function $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ (where $\mathbf{s}^* = (\mathbf{0}, \mathbf{0})$) as

$$\dot{\mathbf{e}} = \dot{\mathbf{s}} = \mathbf{J}_\omega \omega + \mathbf{J}_v v$$

where

$$\mathbf{J}_\omega = \begin{bmatrix} 1 + {x_f}^2 \\ \frac{\cos\theta_m}{h_1}(\cos\theta_1(w_1\rho_m - l_1) - \sin\theta_1(w_1 + l_1\rho_m)) + \rho_m \sin\theta_m \end{bmatrix} \tag{11}$$

and

$$\mathbf{J}_v = \begin{bmatrix} 0 \\ \frac{\cos\theta_m}{h_1}(\sin\theta_1 - \rho_m \cos\theta_1) \end{bmatrix}. \tag{12}$$

Following the same strategy as in [29], designing a locally asymptotically stable control scheme that tries to exponentially regulates $\mathbf{e}$ to 0 (i.e. such that $\dot{\mathbf{e}} = -\lambda\mathbf{e}$ where $\lambda > 0$) leads to

$$\omega = -\mathbf{J}_\omega^+(\lambda\mathbf{e} + \mathbf{J}_v v^*) \tag{13}$$

where $\mathbf{J}_\omega^+$ is the pseudo-inverse of $\mathbf{J}_\omega$ and v^* the constant forward velocity.

From the control law given in 13, one can observe that $\mathbf{J}_\omega$ is always of full rank, but for two singular degenerate configurations. They occur when the vanishing point lies at either side of the image plane (i.e if the wheelchair is perpendicular to the corridor wall), wherein $x_f = \pm\infty$ and the first row of $\mathbf{J}_\omega$ tends to $\pm\infty$. In such a case the robot cannot decide by itself where to move (i.e. the human should make the decision). Moreover, if $x_f = \infty$, the vanishing point will not be detected by the feature extraction process. Another important caveat to consider is that the control law aims to center the *camera* in the middle of the corridor. Therefore, in order to center the wheelchair in the middle of the corridor, a non-zero value should be tuned for the desired value of θ_m as soon as $w_1 \neq 0$.

3.3. Simulation Results

We report two simulated cases meant to illustrate the convergence of the control law and validate it as a suitable system for the task of corridor following. The two initial conditions assessed were as follows.

- Case I represents a case where initially $x_f = 0.15$ meters and $\theta_m = 0.55$ rad which represents a situation where the robot is closer to the right corridor wall while facing the right corridor wall. The extrinsic parameters l_1, w_1 were set to zero in order to asses the control law in an ideal case.

- Case II for which initially $x_f = -1$ meter and $\theta_m = -0.85$ rad which represents a case where the robot is closer to the left corridor wall and facing the left side. The extrinsic parameters were set at $l_1 = 0.5$m and $w_1 = 0.5$m. The desired value of θ_m was tuned to -0.3 rad so that the wheelchair is centered within the corridor. The performance of this case was compared to the performance when there exists some calibration errors with l_1 having an error of 0.15m, w_1 an error of -0.15m and h_1 an error of $+0.2$m.

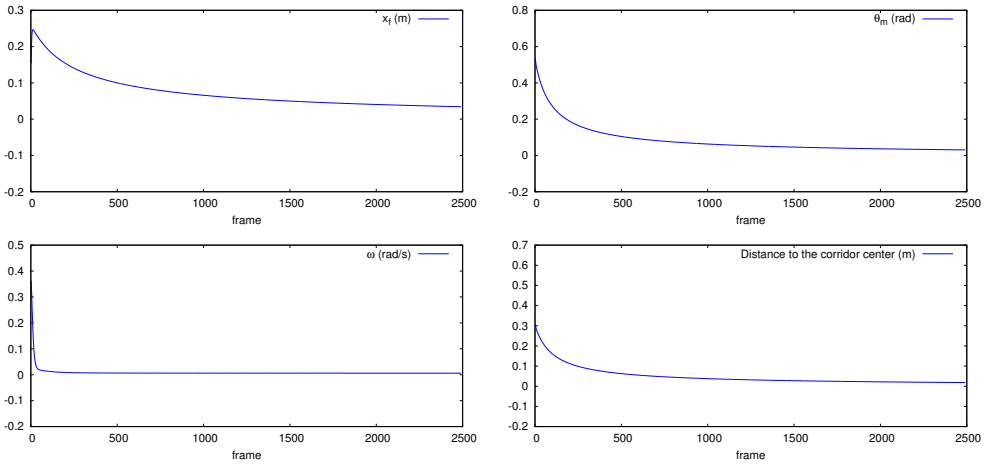


Figure 4: Evolution of x_f, θ_m, ω and camera distance to the corridor center - Case I

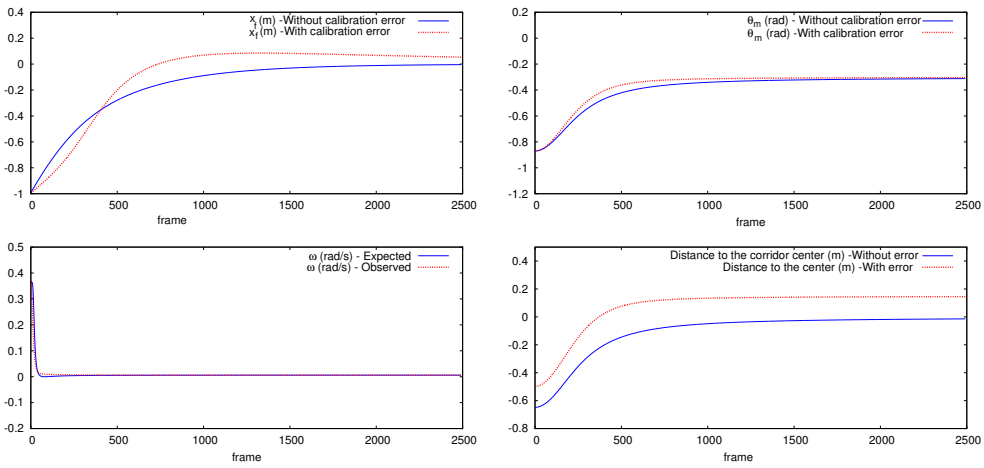


Figure 5: Evolution of x_f, θ_m, ω and camera distance to the corridor center - Case II with no calibration errors (blue) and with calibration errors (red)

The translational velocity v^* was set as 0.2 ms^{-1} while the height h_1 as 0.5m. The results of the simulation can be observed in Figures 4 and 5 where the evolution of the visual features x_f, θ_m, ω and also the distance of the camera from the center of the corridor are shown.

We can observe that the rotational velocity ω decreases in a pure exponential way and that the visual features converge to their desired values $x_f^* = 0$ m and $\theta_m^* = 0/0.3$ rad in both the cases. The convergence of x_f is not completely exponential owing to the fact that two visual features are controlled with 1 DOF only. We can see that x_f tends to a value > 0 or < 0 which in turn allows θ_m to converge. This data is corroborated by the fact that the wheelchair distance from the center of the corridor moves to zero as θ_m converges to θ_m^*.

Moreover the inclusion of calibration errors in the camera extrinsic parameters does not affect the convergence of the system as seen in Figure 5. The only effect such errors in modelling have is an offset in the position of the wheelchair in the corridor corresponding to the modelling error: the wheelchair naturally overshoots the center of the corridor by 0.15m due to the error in w_1.

Based on the above results, it can be concluded that the proposed formulation for corridor following in a wheelchair is a viable solution for implementation in a real-time system. Experimental results will be shown in Section 5.

4. Autonomous Doorway Passing

The task of autonomously passing through a doorway once it is detected is a non-trivial task owing to the fact that there are a lot of high level constraints to be taken care of. For example, whether the door is open or closed, and/or, is the width of the door sufficient enough to let the wheelchair through, etc remain unknown. However, these attributes either can be left for the user to set (i.e. human in the control loop), or left to a home automation system (which works directly with the wheelchair) that manages these high level parameters. Therefore, the aim here is to design a robust low-level controller that achieves an adequate wheelchair motion given a detected doorway.

Based on the above criteria, a novel Lyapunov-based control scheme is designed which exploits the position of the line feature representing the initial doorpost of the door in the image (i.e. the doorpost closer to the robot) as the visual feature.

4.1. Door recognition and tracking

There are several methods for detecting and representing doors in an indoor navigation construct [30] [31]. But, in the present case, the representations have to be simple enough so that effective features can be extracted for visual servoing. Also, detections have to be robust as well as fast enough to process frames in real time at a reasonable speed.

Consequently, we use a door detection and tracking framework specifically developed for indoor navigation tasks [24]. This framework uses a set of information including the vanishing point to estimate a 3-D geometrical structure of the corridor. This structure then defines a search space in which trapezoidal shapes representing doors are extracted. To this aim we consider that the doorposts are nearly vertical in the image. Furthermore, for tracking the detected doorposts, a 2-D edge tracker inspired from the *Moving Edges* (ME) algorithm [32] [33] is applied on the doorposts. The implementation of this framework on a generic corridor is shown in Figure 6.

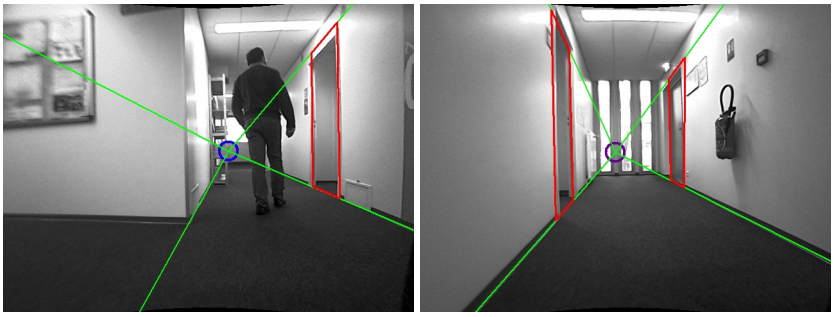


Figure 6: Door detection framework

It has to be noted that the doorpost at the far end of the door will not be present in the field of view of the camera as the wheelchair tries to position itself in front of the door. Consequently we have to design a solution which uses the position of a single (i.e. the initial) doorpost as the input to the system.

4.2. Visual Feature - Definition and Extraction

Considering Figure 7, if the foot of the doorpost is represented by the point D then $D = (x_d, h_2, z_d)$ in frame $\mathcal{F}_{c_2}$. In polar coordinates point D can be represented by

$$r = \sqrt{x_d^2 + z_d^2} \quad \text{and} \tag{14a}$$

$$\phi_d = \arctan(x_d/z_d). \tag{14b}$$

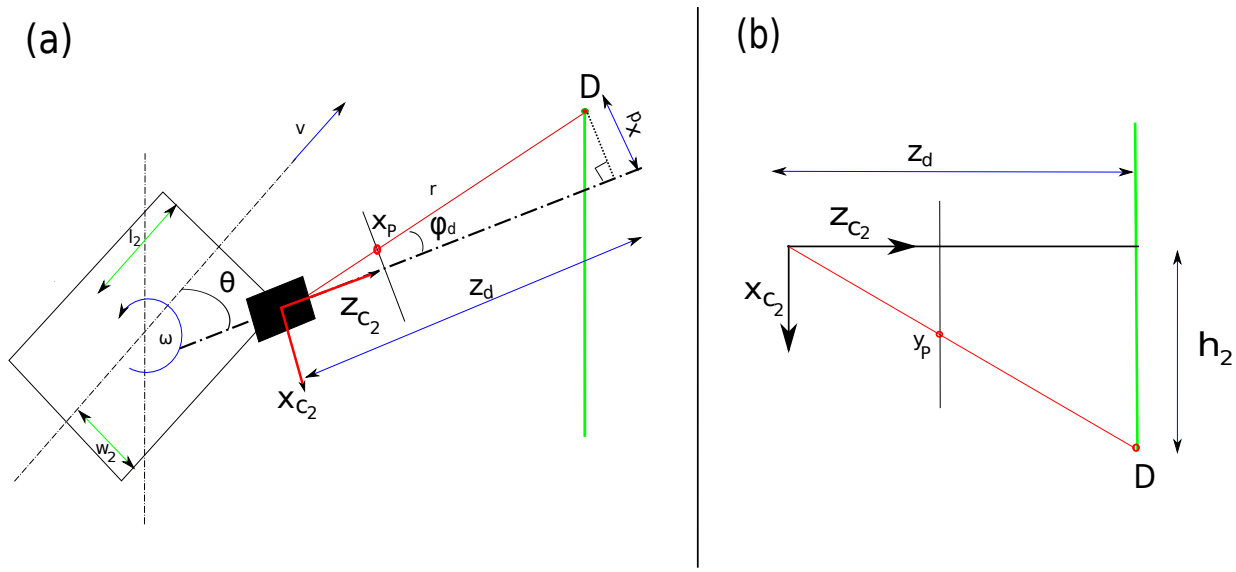


Figure 7: Geometrical constraints while considering doorway passing(top view (a), front view(b)

If the point D projects in the image at point $P = (x_P, y_P)$, then owing to a calibrated camera, the perspective projection equations reduce to

$$x_P = \frac{x_d}{z_d}$$

and

$$y_P = \frac{h_2}{z_d}.$$

To perform a successful doorway passing, the wheelchair needs to avoid the closest doorpost with a predefined margin m. To do so we formally define the visual feature as the angle ϕ_d, since it can be easily estimated from the measure of the position of the doorpost . Indeed from x_P coordinate, we immediately obtain using 14b.

$$\phi_d = \arctan(x_P).$$

Similarly from y_P coordinate of D, the distance r between the camera and the doorpost is given by

$$r = \frac{h_2}{y_P . \cos(\phi_d)}, \tag{15}$$

since we get $z_d = r\cos(\phi_d)$ from Figure 7b.

4.3. Lyapunov-based controller

The visual feature ϕ_d must evolve in a certain way so that the wheelchair takes up a trajectory that realises the task. To assign a desired value ϕ_d^* which must be achieved by ϕ_d for task completion, we have to first assess this trajectory.

4.3.1. Desired trajectory

We specify the desired trajectory that the wheelchair should follow for a successful doorway passage as presented in Figure 8. This trajectory has been chosen since the wheelchair must be able to position itself in front of the doorpost with a tolerance of m no matter what its starting position and orientation is. Consequently, the wheelchair must ideally take a tangential path towards an imaginary circle centred at the doorpost (with radius m) and when the camera distance to the doorpost r is equal to m, it must take up a smooth circular trajectory about the doorpost. Note that this trajectory implicitely allows avoiding collision with the walls of both sides of the corridor.

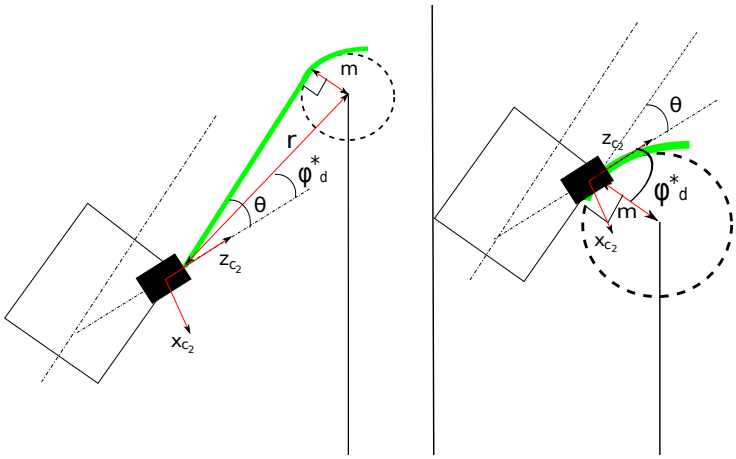


Figure 8: The desired camera trajectory shown in green. (a) when $r > m$ and (b) when $r < m$.

The trajectory can be decomposed into two parts with the first one containing the tangential motion towards the circle (when $r > m$) and the second containing the circular motion about the doorpost (when $r \leq m$). The characteristic of the tangential motion toward the circle is that ideally the value of $\theta - \phi_d$ (see Figure 8) should be equal to $\arcsin(\frac{m}{r})$. Since the angle θ and θ_2 are defined in the frames $\mathcal{F}_{c_2}$ and $\mathcal{F}_r$ respectively, and since θ corresponds to the same angle as θ_2 geometrically, we have $\theta = -\theta_2$. Therefore

$$\phi_d^* = \theta_2 + \arcsin(\frac{m}{r}) \quad \text{if} \quad r > m. \tag{16}$$

As r gets closer to m, the wheelchair must switch to the circular motion about the doorpost. For such case it is clear from Figure 8 that the desired value of the visual feature ϕ_d^* must be equal to $\frac{\pi}{2} - \theta$. Therefore, we can state that

$$\phi_d^* = \theta_2 + \frac{\pi}{2} \quad \text{if} \quad r \leq m. \tag{17}$$

4.3.2. Control law formulation

To design a globally asymptotically stable system, we select as Lyapunov candidate function

$$V = \frac{1}{2}(\phi_d - \phi_d^*)^2. \tag{18}$$

Of course we have $V = 0$ when $\phi_d = \phi_d^*$. Then, we have to design a control scheme so that $\dot{V} < 0$. We have

$$\dot{V} = (\phi_d - \phi_d^*)(\dot{\phi}_d - \dot{\phi}_d^*). \tag{19}$$

First by deriving 14b we obtain

$$\dot{\phi}_d = \frac{z_d \dot{x}_d - x_d \dot{z}_d}{x_d^2 + z_d^2}, \tag{20}$$

which can be re-written as (see Appendix),

$$\dot{\phi}_d = \omega + \frac{1}{r}\left[\sin(\phi_d - \theta_2)v + (l_2 \cos(\phi_d - \theta_2) - w_2 \sin(\phi_d - \theta_2))\omega\right]. \tag{21}$$

Then from 16 we obtain when $r > m$

$$\dot{\phi}_d^* = \frac{d}{dt}\left(\arcsin\left(\frac{m}{r}\right)\right) = \frac{-\dot{r}m}{r\sqrt{r^2 - m^2}} \tag{22}$$

which can be re-written as (see Appendix)

$$\dot{\phi}_d^* = \frac{\sin(\phi_d^* - \theta_2)\left[\cos(\phi_d - \theta_2)v - (l_2 \sin(\phi_d - \theta_2) + w_2 \cos(\phi_d - \theta_2))\omega\right]}{r\cos(\phi_d^* - \theta_2)} \tag{23}$$

Finally from 17 we have when $r \leq m$

$$\dot{\phi}_d^* = 0. \tag{24}$$

By substituting the expressions for $\dot{\phi}_d$ and $\dot{\phi}_d^*$ in 19, we obtain a form for the derivative of V as (see Appendix)

$$\dot{V} = \left(\phi_d - \phi_d^*\right)(vA(r, \phi_d) + \omega(1 + B(r, \phi_d))) \tag{25}$$

where, when $r > m$

$$\begin{cases} A(r, \phi_d) = \dfrac{\sin(\phi_d - \phi_d^*)}{r\cos(\phi_d^* - \theta_2)} \\ B(r, \phi_d) = \dfrac{l_2 \cos(\phi_d - \phi_d^*) - w_2 \sin(\phi_d - \phi_d^*)}{r\cos(\phi_d^* - \theta_2)}. \end{cases} \tag{26}$$

and when $r \leq m$ in which case we recall that $\dot{\phi}_d^* = 0$,

$$\begin{cases} A(r, \phi_d) = \dfrac{\sin(\phi_d - \theta_2)}{r} \\ B(r, \phi_d) = \dfrac{l_2 \cos(\phi_d - \theta_2) - w_2 \sin(\phi_d - \theta_2)}{r}. \end{cases} \tag{27}$$

Thus in both cases, if we choose ω such that

$$\omega = \frac{-k(\phi_d - \phi_d^*) - A(r, \phi_d)v^*}{1 + B(r, \phi_d)}, \tag{28}$$

where k is a positive gain factor, then $\dot{V} < 0$. This ensures that the system is globally asymptotically stable and the visual feature ϕ_d will converge asymptotically to the desired value ϕ_d^*. Furthermore, when $r > m$, it can also be shown that $\omega = 0$ when $\phi_d = \phi_d^*$ thus providing a straight tangential motion towards the circle. Finally, as soon as $\phi_d = \phi_d^* = \theta_2 + \frac{\pi}{2}$ (when $r \leq m$) from (27) we have $A = \frac{1}{r}$ and $B = -\frac{w_2}{r}$ from which we deduce $\omega = \frac{-v^*}{r - w_2}$. This naturally corresponds to a circular motion when $r = m$ whose radius is equal to $m - w_2$. Note also that the switching has to be performed just before $r = m$ so that A and B in (26) does not reach a very high value.

4.4. Simulation Results

To validate the framework formulated above, the results of two simulated cases are presented. The aim of the simulations were to assess the trajectory taken by the wheelchair during the doorway passage, particularly the control law switch which happens as $r \sim m$ which facilitates the wheelchair to take up a circular trajectory about the doorpost. For both cases the doorpost was fixed at the origin of a Cartesian door frame say $\mathcal{F}_d$ with the wheelchair starting at a relative position (x_{door}, y_{door}) with respect to $\mathcal{F}_d$ with $x_{door} = -1.5$m and $y_{door} = -1.5$m. The camera was initially aligned parallel to the corridor wall. The margin m was set at 0.2m, the height h_2 to 0.5m and v^* was set to 0.1 ms^{-1}. Finally the gain k was set to 2.

- Case I represents an ideal case where the factors l_2, w_2 and θ_2 representing the extrinsic parameters were set to 0.
- Case II represents a real world situation where the extrinsic parameters were set to non-zero so that $l_2 = 0.5$m, $w_2 = -0.5$m and the angle $\theta_2 = -0.8$rad. Moreover, this case was compared using calibration errors. Indeed, errors of 0.2m, −0.2m and −0.15m were respectively added to l_2, w_2 and h_2 respectively. Also an error of +0.2 rad was added on θ_2.

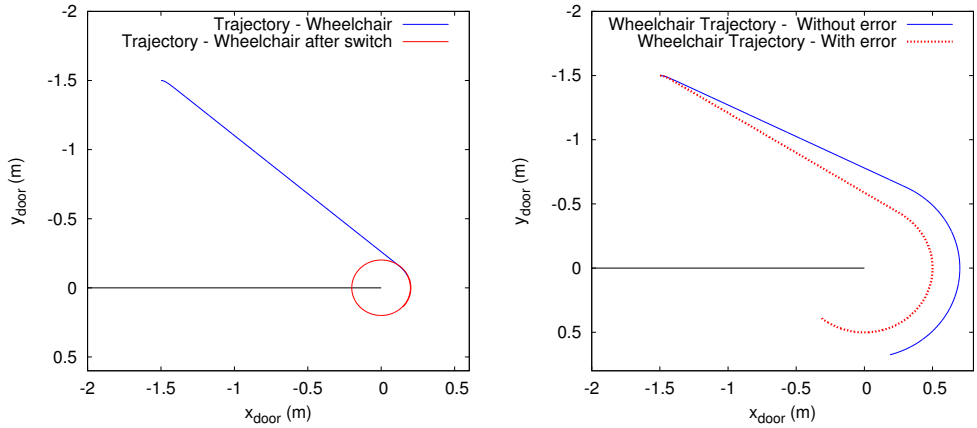


Figure 9: The trajectories: Case I (left) and Case II - without error in blue and with error in red (right)

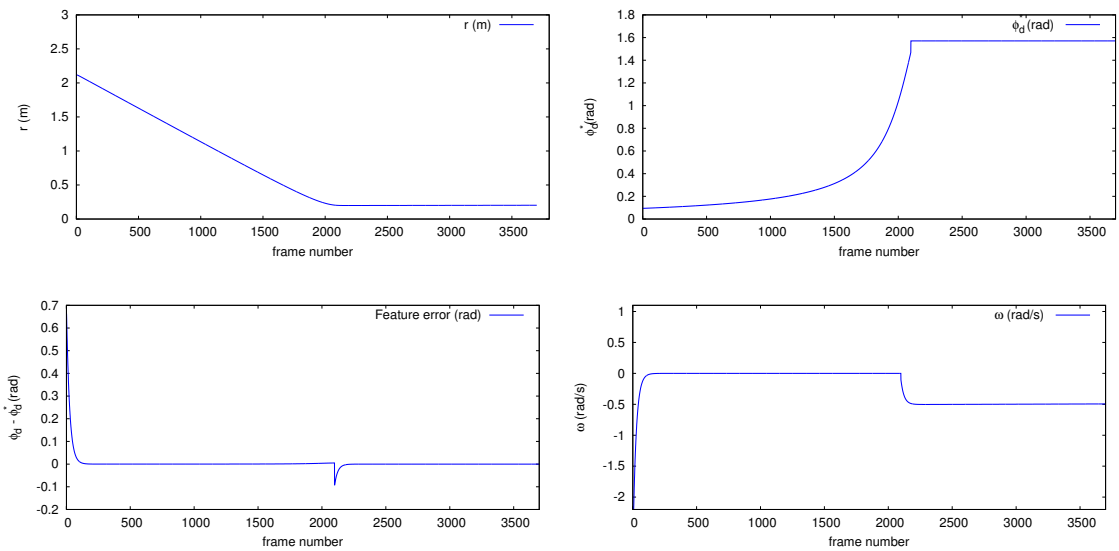


Figure 10: Evolution of r, ϕ_d^*, $\phi_d - \phi_d^*$ and ω - Case I

The trajectories taken by the wheelchair are shown in Figure 9. It can be observed that in both cases, the wheelchair takes up a straight line and tangential motion towards a circle centered at the doorpost and as $r \approx m$, the control law switch allows the robot to take up a circular trajectory around the doorpost. We can observe the evolution of r, ϕ_d^*, the feature error $\phi_d - \phi_d^*$ and ω in Figures 10 and 11. As for case I, the visual feature ϕ_d converges to ϕ_d^* in a few iterations as shown by the evolution of the feature error. As the the control law switches at $r \approx 0.2$m, the error increases due to the change in behaviour of ϕ_d^* but again moves to zero exponentially. Furthermore, it is clear that the distance from the camera to the doorpost r decreases constantly and stays near to the margin $m = 0.2$m. During the task, the rotational velocity ω is initially high but converges rapidly to 0. It then reacts to the control law switch and converges to the expected value for realizing the circular trajectory. In practice, in order to avoid a sudden increase in ω at the start, a method ensuring the continuity of the control law, such as the one proposed in [34] could be used.

The addition of the real world dimensions does not affect the system dynamics at all. As shown in Figure 9 the trajectory taken up by the wheelchair respects the dimensions of the wheelchair in the case with perfect calibration. Moreover, errors in calibration do not have an impact on the convergence of the control law as can be seen from the

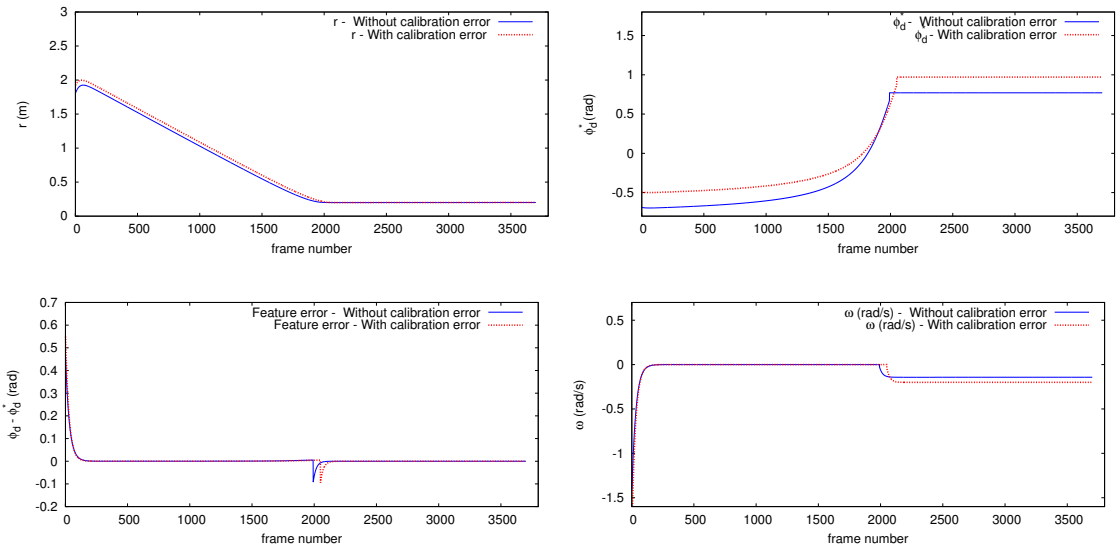


Figure 11: Evolution of r, ϕ_d^*, $\phi_d - \phi_d^*$ and ω - Case II

evolution of the feature error in Figure 11. But since there is an error in the angle θ_2, this affects the value of ϕ_d^*. Also, the error in w_2 naturally affects the trajectory of the wheelchair corresponding to the error since the radius of the circular motion is equal to $m - w_2$ according to the designed control law. We can however note that the feature error $\phi_d - \phi_d^*$ remains small, demonstrating the robustness of the control scheme with respect to large calibration errors.

5. Experiments and Results

The wheelchair robot used in this work is an off-the-shelf Penny and Giles wheelchair adapted to robotic use using ROS middleware [35]. The wheelchair is equipped with two Raspberry PI [36] camera modules with 100° field of view. The video stream from the camera runs at 15 frames per second and correspond to a frame width of 808 pixels and a frame height of 480 pixels. The *camera 1* was coarsely calibrated at $l_1 = 0.38$m, $w_1 = 0.32$m and $h_1 = 0.64$m while *camera 2* at $l_2 = 0.0$m, $w_2 = -0.32$m and $h_2 = 0.5$m with respect to $\mathcal{F}_r$. *Camera 2* was aligned at an angle of $\theta_2 = -50° \sim -0.88$rad with respect to the robot frame. Finally, the margin for doorway passing m was set at 0.2m.

Also, prior to the feature extraction, images are rectified against the distortions due to the camera lens. Extraction of the visual features as well as the computation of the control laws were performed using the ViSP software [37]. Computation is performed on a Core i7 laptop connected to the wheelchair using Ethernet network. The results presented here pertain to the one realised in one of the corridors of the Inria building 12C in Rennes, France. Human intervention is needed only for turning the visual servoing process on/off and to select a doorway to pass through.

1 The wheelchair starts at an unknown position in the corridor and the servoing process is activated in order to follow the corridor.

2 As soon as a turn arrives, the user interacts with the system to select the end of the corridor to turn about using the doorway passing control scheme.

3 After the wheelchair has turned about the corridor, the servoing process for following the new corridor switches on automatically when both $\phi_d^* = \frac{\pi}{2} + \theta_2$ and a vanishing point is detected.

4 While the wheelchair is still following the corridor the user interacts again, this time to select a doorway to pass through.

5 The servoing process is switched off once the wheelchair manages to pass through the specific doorway.

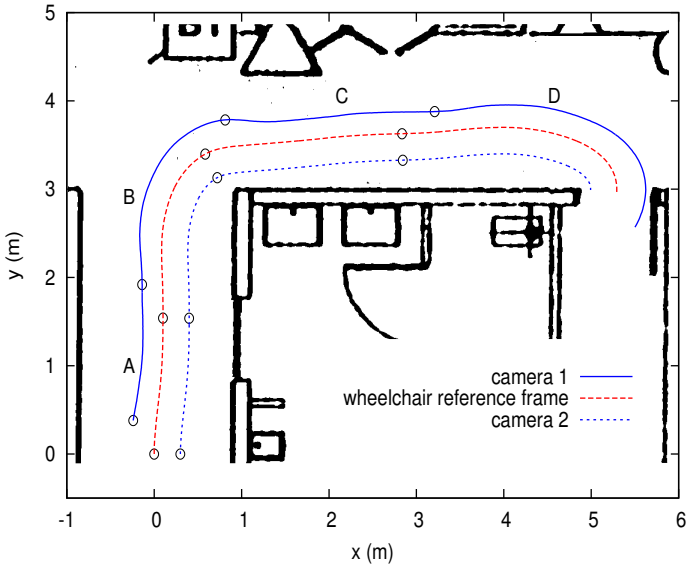
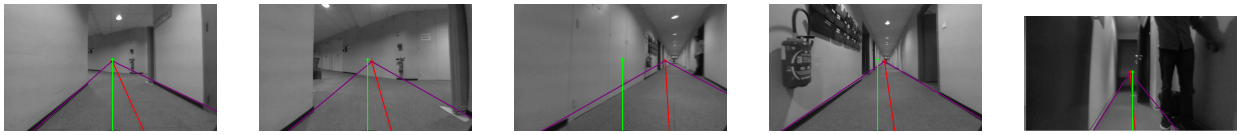


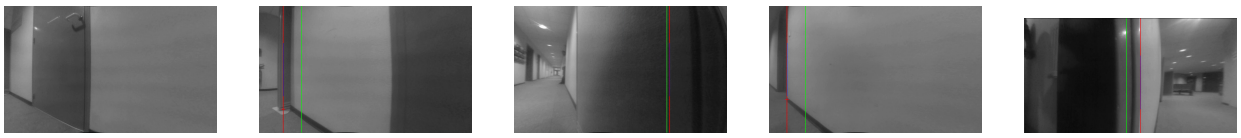
Figure 12: Trajectory taken by the wheelchair with respect to a frame fixed at the initial position

The video of the experiment can be downloaded from `http://www.irisa.fr/lagadic/team/MarieBabel/video/2014_AutonomousCorridorDoorPassing.mp4`.



(a) Frame 23 (b) Frame 111 (c) Frame 241 (d) Frame 392 (e) Frame 590

Figure 13: The *camera 1* frame at points A,B,C,D at the end of the navigation respectively.



(a) Frame 23 (b) Frame 111 (c) Frame 241 (d) Frame 392 (e) Frame 590

Figure 14: The *camera 2* frame at points A,B,C,D and at the end of the navigation respectively.

Figure 12 shows the trajectory taken by *camera 1*, *camera 2* and the wheelchair during the experiment process. It is reproduced in a global frame with the wheelchair starting position taken as the origin. Odometry has been used to reconstruct the trajectory on a map of the environment for visualization. It has to be noted that neither odometry nor the environment map is employed in the servoing process.

The trajectory is decomposed into four parts. During part A, the wheelchair performs corridor following in order to position itself in the middle of the corridor while controlling the two visual features x_f and θ_m. It can be noted that the wheelchair starts relatively in a central position and the correction is done as soon as the servoing starts. As

the wheelchair moves to the end of the corridor, the user selects the wall partition to turn right. Consequently, part B is activated where the wheelchair manages to judge the wall partition as a doorpost and turn about it. Thus we can observe that the framework for doorway passing can be employed for turning within corridors.

After the wheelchair successfully turns, the corridor following task (part C) is automatically activated once $\phi_d = \frac{\pi}{2} + \theta_2$ and a vanishing point is detected. Again the control scheme acts to position the wheelchair in the middle of the corridor. Finally, the user selects a doorway to pass through to start (part D) while the wheelchair manages to turn about the doorpost for a successful doorway passage. The servoing process is stopped as soon as the wheelchair is positioned in front of the doorway.

At critical points of the servoing process, the camera view can be seen in Figures 13 and 14 for *camera 1* and *camera 2* respectively. Figures 13a and 14a represent the start of the servoing process. The user selects the wall partition in the frames shown in Figures 13b and 14b which marks the transition from A to B. A new vanishing point is obtained in frame 241 (Figures 13c and 14c) with the visual feature $\phi_d \sim \frac{\pi}{2} + \theta_2$. Then frame 392 (Figures 13d and 14d) marks the start of doorway passing and a transition onto part D. Finally, frame 590 (Figures 13e and 14e) shows the frames at the end of the navigation.

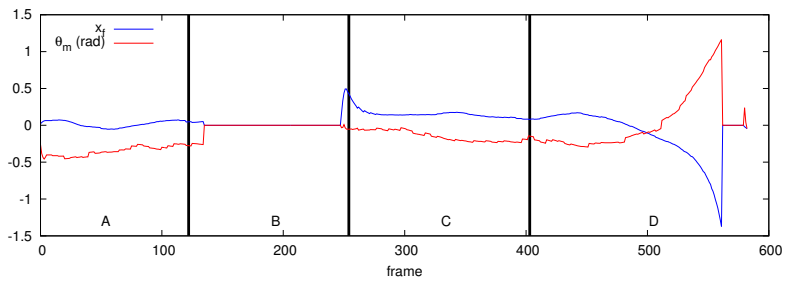


Figure 15: The evolution of the corridor visual features x_f and θ_m

Figure 15 shows the behaviour of visual features x_f and θ_m during the servoing process. Since the front facing *camera 1* is located at $w = 0.32$m from the centre of the wheelchair reference frame, the desired value of θ_m was set at -0.3rad. This was done in order to facilitate the wheelchair in the middle of the corridor rather than the camera. During part A, we can observe the convergence of the features to their desired values $x_f = 0$ and $\theta_m = -0.3$ rad as the wheelchair centres itself in the corridor. The features evolve till the vanishing point is not detected anymore during the corridor turning part B. As part C starts, the features are detected again and we can observe the exponential decrease of x_f to zero. Also, θ_m converges gradually to -0.3rad as observed in simulation. Again, the features are correctly extracted during part D owing to the fact that the vanishing point is detected until the wheelchair makes a complete turn. Thus, the behaviour of x_f and θ_m during parts A and C validates the control scheme for autonomous corridor following for use in a real time system.

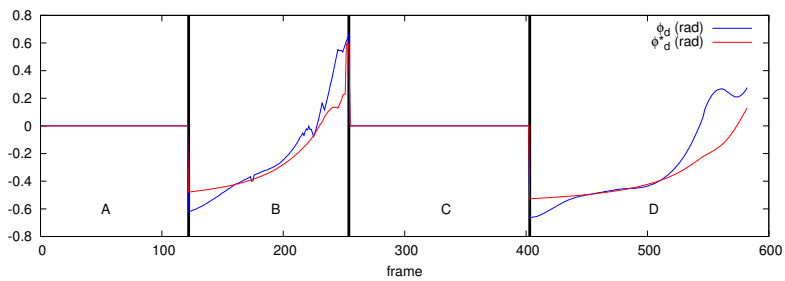


Figure 16: The evolution of the visual feature ϕ_d

Figures 16 and 17 present the evolution of the visual feature ϕ_d as well as the distance r from the camera to the doorpost/wall partition during the experiment. The values are naturally zero in part A and C as no servoing occurs for doorway passing. It is activated as soon as the user selects the wall partition to turn about. We can observe that

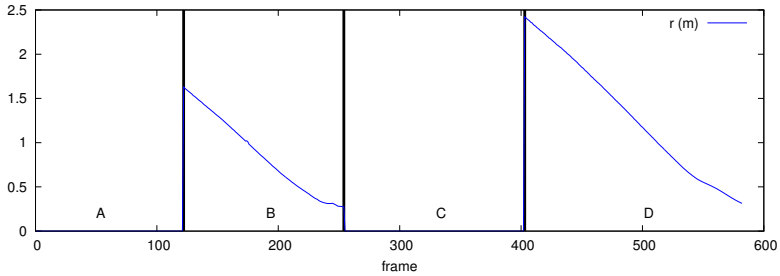


Figure 17: The evolution of the distance to the doorpost r

ϕ_d manages to follow ϕ_d^* very closely and reaches $\frac{\pi}{2} + \theta_2 = 0.69$ rad. The perturbations in ϕ_d can be attributed to the low dynamics of the wheelchair and to visual tracking errors. But we can observe that the control scheme manages to servo the feature to its desired value. A similar behaviour of ϕ_d can be reported while in part D. The visual feature manages to follow the desired value during the servoing process. There is a small discrepancy at the end which is again due to some tracking errors even though ϕ_d converges to ϕ_d^* at the end. The final desired value of 0.69 rad is not reached since the servoing is stopped as soon as the wheelchair is positioned in front of the doorway.

When considering r, we can observe an almost constant decrease to the margin $m = 0.2$m. Then the value of $r = m$ remains almost constant till the task switch occurs. This behaviour is critical for the purpose of doorpost collision avoidance and for the requirement that the wheelchair is able to position in front of the doorway no matter what its initial position.

A final important point to mention is that the cameras are coarsely calibrated with respect to the intrinsic parameters as well as their position with respect to the wheelchair. The robust performance of the controller in such a case demonstrates that the impact of modelling errors is minimal on the system.

6. Conclusion

In this work we have proposed a visual servoing approach for autonomously following a corridor as well as pass though doorways and turn about corridors in an electric wheelchair. The task of corridor following is realised by exploiting two visual features, namely the vanishing point and the angle obtained from the projection of the median line of the corridor onto the image. A control law was designed to servo the features onto their desired values by taking into account the kinematics of the wheelchair. The doorway passing (and corridor turning) task employs the position of a single doorpost in the image as an input to a Lyapunov-based control scheme which allows the wheelchair to take up a desired trajectory about the doorpost. This trajectory avoids collision with the wall and guarantees that the wheelchair positions itself in front of the doorway regardless of its initial position.

Results in simulation demonstrate the convergence and robustness of both control schemes. Experiments conducted on a robotic platform indicate the validity of applying the proposed low-level control system on an assitive wheelchair. Future works aim at enhancing the system to account for the human-in-the-loop condition where the wheelchair user would have considerable presence in the control loop thus modifying the system into a *semi-autonomous assitive system*.

Acknowledgement

This work is supported by the Inria large-scale initiative action Personally Assisted Living (PAL). The authors would like to thank Daniel Guillard and Luc Le Pape from Ergovie and Eric Bazin from INSA Rennes.

Appendix

We recall that (see 20)

$$\dot{\phi}_d = \frac{z_d \dot{x}_d - x_d \dot{z}_d}{x_d^2 + z_d^2}, \tag{29}$$

From the well known kinematics equation $\dot{\mathbf{x}} = -\boldsymbol{v} - [\boldsymbol{\omega}]_\times \mathbf{x}$, we deduce $\dot{x_d}$ and $\dot{z_d}$ as

$$\begin{cases} \dot{x_d} = & -{}^{c_2}v_x - z_d^{c_2}\omega_y \\ \dot{z_d} = & -{}^{c_2}v_z + x_d^{c_2}\omega_y \end{cases} \tag{30}$$

which using equation (5) leads to

$$\dot{x_d} = -\sin\theta_2 v + \omega(l_2\cos\theta_2 + w_2\sin\theta_2) + z_d\omega, \tag{31a}$$

$$\dot{z_d} = -\cos\theta_2 v - \omega(l_2\sin\theta_2 - w_2\cos\theta_2) - x_d\omega. \tag{31b}$$

Since $x_d = r\sin\phi_d$ and $z_d = r\cos\phi_d$, we obtain

$$\dot{\phi_d} = \frac{r\cos\phi_d[-\sin\theta_2 v + \omega(l_2\cos\theta_2 + w_2\sin\theta_2) + r\cos\phi_d\omega] - r\sin\phi_d[-\cos\theta_2 v - \omega(l_2\sin\theta_2 - w_2\cos\theta_2) - r\sin\phi_d\omega]}{r^2}, \tag{32}$$

which can be re-written as

$$\begin{aligned}\dot{\phi_d} = \frac{1}{r}\left[(v(-\cos\phi_d\sin\theta_2 + \sin\phi_d\cos(\theta_2))\right] \\ + \frac{1}{r}\left[\omega(l_2(\cos\phi_d\cos\theta_2 + \sin\phi_d\sin\theta_2) + w_2(\cos\phi_d\sin\theta_2 - \sin\phi_d\cos\theta_2)) + r\omega)\right].\end{aligned} \tag{33}$$

Using trigonometric relations we finally get,

$$\dot{\phi_d} = \omega + \frac{1}{r}\left[\sin(\phi_d - \theta_2)v + (l_2\cos(\phi_d - \theta_2) - w_2\sin(\phi_d - \theta_2))\omega\right]. \tag{34}$$

Similarly to obtain $\dot{r}$, in order to derive $\dot{\phi_d^*}$ (in equation 22), we deduce from 14a

$$\dot{r} = \frac{x_d\dot{x_d} + z_d\dot{z_d}}{\sqrt{x_d^2 + z_d^2}}. \tag{35}$$

Therefore using 31a and 31b, we can write

$$\dot{r} = \frac{r\sin\phi_d[-\sin\theta_2 v + \omega(l_2\cos\theta_2 + w_2\sin\theta_2) + r\cos\phi_d\omega] + r\cos\phi_d[-\cos\theta_2 v - \omega(l_2\sin\theta_2 - w_2\cos\theta_2) - r\sin\phi_d\omega]}{r}. \tag{36}$$

This gives,

$$\dot{r} = -v(\sin\phi_d\sin\theta_2 + \cos\phi_d\cos\theta_2) + \omega[l_2(\sin\phi_d\cos\theta_2 - \cos\phi_d\sin\theta_2) + w_2(\sin\phi_d\sin\theta_2 + \cos\phi_d\cos\theta_2)] \tag{37}$$

and using trigonometric formulae we finally have,

$$\dot{r} = -\cos(\phi_d - \theta_2)v + (l_2\sin(\phi_d - \theta_2) + w_2\cos(\phi_d - \theta_2))\omega. \tag{38}$$

Therefore $\dot{\phi_d^*}$ can be written as (from 22)

$$\dot{\phi_d^*} = \frac{m\left[\cos(\phi_d - \theta_2)v - (l_2\sin(\phi_d - \theta_2) + w_2\cos(\phi_d - \theta_2))\omega\right]}{r\sqrt{r^2 - m^2}} \tag{39}$$

From 16 and basic trigonometric relations, if we substitute $\frac{m}{r} = \sin(\phi_d^* - \theta_2)$ and $\frac{\sqrt{r^2-m^2}}{r} = \cos(\phi_d^* - \theta_2)$, we obtain

$$\dot{\phi_d^*} = \frac{\sin(\phi_d^* - \theta_2)\left[\cos(\phi_d - \theta_2)v - (l_2\sin(\phi_d - \theta_2) + w_2\cos(\phi_d - \theta_2))\omega\right]}{r\cos(\phi_d^* - \theta_2)} \tag{40}$$

Finally we derive the details to obtain 25. From 21 and 40 we can write $\dot{\phi}_d - \dot{\phi}_d^*$ as

$$\dot{\phi}_d - \dot{\phi}_d^* = \omega + \frac{1}{r}\left[\sin(\phi_d - \theta_2)v + (l_2\cos(\phi_d - \theta_2) - w_2\sin(\phi_d - \theta_2))\omega\right] - \frac{\sin(\phi_d^* - \theta_2)\left[\cos(\phi_d - \theta_2)v - (l_2\sin(\phi_d - \theta_2) + w_2\cos(\phi_d - \theta_2))\omega\right]}{r\cos(\phi_d^* - \theta_2)}. \quad (41)$$

We can rearrange the above equation as,

$$\dot{\phi}_d - \dot{\phi}_d^* = \frac{v}{r}\left[\sin(\phi_d - \theta_2) - \frac{\sin(\phi_d^* - \theta_2)\cos(\phi_d - \theta_2)}{\cos(\phi_d^* - \theta_2)}\right] + \omega\left[1 + \frac{l_2}{r}\left[\cos(\phi_d - \theta_2) + \frac{\sin(\phi_d - \theta_2)\sin(\phi_d^* - \theta_2)}{\cos(\phi_d^* - \theta_2)}\right] - \frac{w_2}{r}\left[\sin(\phi_d - \theta_2) - \frac{\sin(\phi_d^* - \theta_2)\cos(\phi_d - \theta_2)}{\cos(\phi_d^* - \theta_2)}\right]\right]. \quad (42)$$

and finally we can re-write the equation as

$$\dot{\phi}_d - \dot{\phi}_d^* = v\left[\frac{\sin(\phi_d - \phi_d^*)}{r\cos(\phi_d^* - \theta_2)}\right] + \omega\left[1 + \frac{l_2\cos(\phi_d - \phi_d^*) - w_2\sin(\phi_d - \phi_d^*)}{r\cos(\phi_d^* - \theta_2)}\right] \quad (43)$$

from which we deduce the form for $\dot{V}$ given in 25.

References

[1] M. Finlayson, T. van Denend, Experiencing the loss of mobility: perspectives of older adults with ms, Disability and Rehabilitation 25(20) 2003, pp 1168–1180.

[2] L. Iezzoni, E. McCarthy, R. Davis, H. Siebens, Mobility difficulties are not only a problem of old age, Journal of General Internal Medicine 16 (4) 2001, pp. 235–243.

[3] A. Murarka, S. Gulati, P. Beeson, B. Kuipers, Towards a safe, low-cost, intelligent wheelchair, in: Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV) 2009, pp. 42–50.

[4] H. S. Kaye, T. Kang, M. P. LaPlante, Mobility device use in the united states, Disability Statistics Report 14, 2000.

[5] H. Yanco, Integrating robotic research: A survey of robotic wheelchair development, in: AAAI Spring Symposium on Integrating Robotic Research, 1998.

[6] T. Gomi, A. Griffith, Developing intelligent wheelchairs for the handicapped, in: V. Mittal, H. Yanco, J. Aronis, R. Simpson (Eds.), Assistive Technology and Artificial Intelligence, Vol. 1458 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1998, pp. 150–178.

[7] S. P. Levine, D. A. Bell, L. A. Jaros, R. C. Simpson, Y. Koren, J. Borenstein, The navchair assistive wheelchair navigation system, IEEE Transactions on Rehabilitation Engineering 7, 1999, pp 443–451.

[8] E. Demeester, E. EB Vander Poorten, A. Hüntemann, J. De Schutter, Wheelchair navigation assistance in the fp7 project radhar: Objectives and current state, IROS Workshop on Progress, Challenges and Future Perspectives in Navigation and Manipulation Assistance for Robotic Wheelchairs, 2012.

[9] A. Kokosy, T. Floquet, G. Howells, H. Hu, M. Pepper, C. Donz, SYSIASS An Intelligent Powered Wheelchair, in: International Conference on Systems and Computer Science, 2012.

[10] N. Winters, J. Gaspar, G. Lacey, J. Santos-Victor, Omni-directional vision for robot navigation, Proceedings of IEEE Workshop on Omnidirectional Vision, 2000.

[11] R. Carelli, E. O. Freire, Corridor navigation and wall-following stable control for sonar-based mobile robots, Robotics and Autonomous Systems 45 (34) 2003, pp. 235 - 247.

[12] R. Carelli, C. Soria, O. Nasisi, R. Freire, Stable agv corridor navigation with fused vision-based control signals, in: IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the], Vol. 3, 2002, pp. 2433–2438.

[13] S. Patel, S.-H. Jung, J. Ostrowski, R. Rao, C. Taylor, Sensor based door navigation for a nonholonomic vehicle, Proceedings. of IEEE International Conference on Robotics and Automation (ICRA), Vol. 3, 2002, pp. 3081–3086.

[14] W. Chung, S. Kim, M. Choi, J. Choi, H. Kim, C. bae Moon, J.-B. Song, Safe navigation of a mobile robot considering visibility of environment, IEEE Transactions on Industrial Electronics, 56 (10), 2009, pp. 3941–3950.

[15] F. Chaumette, S. Hutchinson, Visual servo control, part i: Basic approaches, IEEE Robotics and Automation Magazine 13 (4), 2006, pp. 82–90.

[16] H. Hadj-Abdelkader, Y. Mezouar, P. Martinet, F. Chaumette, Catadioptric visual servoing from 3-d straight lines, IEEE Transactions on Robotics 24 (3), 2008, pp. 652–665.

[17] A. Faragasso, G. Oriolo, A. Paolillo, M. Vendittelli, Vision-based corridor navigation for humanoid robots, IEEE International Conference on Robotics and Automation (ICRA), 2013, pp. 3190–3195.

[18] J. Toibero, C. Soria, F. Roberti, R. Carelliz, P. Fiorini, Switching visual servoing approach for stable corridor navigation, International Conf. on Advanced Robotics, ICAR 2009, pp. 1–6.

[19] S. Blazic, A novel trajectory-tracking control law for wheeled mobile robots, Robotics and Autonomous Systems 59 (11) 2011, pp. 1001 - 1007.
[20] J. J. Park, B. Kuipers, A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment, in: IEEE Inter. Conf. on Robotics and Automation ICRA 2011, May, pp. 4896–4902.
[21] C. Rother, A new approach for vanishing point detection in architectural environments, in: In Proc. 11th British Machine Vision Conference, 2000, pp. 382–391.
[22] K. Boulanger, K. Bouatouch, S. Pattanaik, ATIP: A Tool for 3D Navigation inside a Single Image with Automatic Camera Calibration, EG UK conference, 2006.
[23] R. Grompone von Gioi, J. Jakubowicz,J.-M. Morel, G. Randall, LSD: a Line Segment Detector, Image Processing On Line, 2012.
[24] R. Sekkal, F. Pasteau, M. Babel, B. Brun, I. Leplumey, Simple Monocular door detection and tracking, IEEE Int. Conf. on Image Processing, ICIP, Melbourne, Australia, 2013.
[25] K. Ok, D.N. Ta, F. Dellaert, Vistas and wall-floor intersection features - enabling autonomous flight in man-made environments, Workshop on Visual Control of Mobile Robots, 2012.
[26] E. Delage, H. Lee, A. Ng, A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image, in: , IEEE Conf. on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 2418 – 2428.
[27] B. Espiau, F. Chaumette, P. Rives, A new approach to visual servoing in robotics, IEEE Transactions on Robotics and Automation 8 (3), 1992, pp. 313-326.
[28] J. R. Azinheira, P. Rives, Image-based visual servoing for vanishing features and ground lines tracking: Application to a uav automatic landing, International Journal of Optomechatronics 2 (3), 2008.
[29] A. Cherubini, F. Chaumette, G. Oriolo, Visual servoing for path reaching with nonholonomic robots, Robotica 29 (7), 2011, pp. 1037–1048.
[30] D. Anguelov, D. Koller, E. Parker, S. Thrun, Detecting and modeling doors with mobile robots, Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Vol. 4, 2004, pp. 3777–3784.
[31] Z. Chen, S. Birchfield, Visual detection of lintel-occluded doors from a single image, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2008, pp. 1–8.
[32] P. Bouthemy, A maximum likelihood framework for determining moving edges, IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 (5), 1989, pp. 499–511.
[33] S. Boukir, P. Bouthemy, F. Chaumette, D. Juvin, A local method for contour matching and its parallel implementation, Machine Vision and Applications 10 (5-6), 1998, pp. 321–330.
[34] N. Mansard, F. Chaumette, Task sequencing for high-level sensor-based control, IEEE Transactions on Robotics, 23 (1), 2007, pp. 60–72.
[35] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, Ros: an open-source robot operating system, ICRA Workshop on Open Source Software, 2009.
[36] R. pi fundation, Raspberry Pi An ARM GNU/Linux box for $25. Take a byte!, http://www.raspberrypi.org/, 2012.
[37] E. Marchand, F. Spindler, F. Chaumette, ViSP for visual servoing: a generic software platform with a wide class of robot control skills, IEEE Robotics and Automation Magazine 12 (4), 2005, pp. 40–52.