Optimal Paths in a Constrained Image Plane for Purely Image-Based Parking

Paolo Salaris, Felipe Weileman Belo, Daniele Fontanelli, Luca Greco, Antonio Bicchi

To cite this version:

Paolo Salaris, Felipe Weileman Belo, Daniele Fontanelli, Luca Greco, Antonio Bicchi. Optimal Paths in a Constrained Image Plane for Purely Image-Based Parking. Int. Conf. on Intelligent Robots and Systems (IROS 2008), Sep 2008, Nice, France. pp.1673-1680. hal-01322840

HAL Id: hal-01322840

https://hal-centralesupelec.archives-ouvertes.fr/hal-01322840

Submitted on 14 Jan 2019
Optimal Paths in a Constrained Image Plane for Purely Image–Based Parking

Paolo Salaris, Felipe A. W. Belo, Daniele Fontanelli, Luca Greco, Antonio Bicchi

Abstract—This paper presents a correct solution to the optimal visual feedback control for a nonholonomic vehicle with limited field-of-view. Previous work on this subject [2] has shown that the search for a shortest path can be limited to simple families of trajectories. We preliminarily provide an extension of the alphabet of optimal control words, to cover some regions of the vehicle plane where the synthesis of [2] turns out to be suboptimal. The main contribution of this paper is an algorithm to translate the optimal synthesis to the image plane, thus enabling a purely image-based optimal control scheme. This allows better performance and increases the robustness of the overall process, avoiding the need of slowly-converging and error-prone parameter estimation algorithms. Simulations and experiments are reported which demonstrate the effectiveness of the proposed technique.

I. INTRODUCTION

One of the most important issues in mobile robotics concerns the trade-offs between sensorization capability, accuracy and cost. Recently, application of vision systems for vehicle guidance has gained increasing attention, both in estimation and control problems. Visual servoing techniques use visual information directly, by the computation of an image error signal, or indirectly, by the evaluation of the state of the system ([3], [4]). These two approaches, often referred to as Image–Based (IBVS) and Position–Based (PBVS) ([17]), can be regarded as the end-points of a range of different possibilities, whereby the raw sensorial information is gradually abstracted away to a more structured representation using some knowledge of the robot-environment model.

PBVS and in general higher-level control schemes have important, attractive features. Using the PBVS approach, for instance, the control law can be synthesized in the usual working coordinates for the robot, usually making the synthesis simpler ([6]). On the other hand, IBVS and other sensor-level control schemes have also several advantages, such as robustness (or even insensitivity) to modeling errors ([15]) and hence suitability to unstructured scenes and environments.

Thanks to well-established advances in point-feature extraction and tracking algorithms, such as the Scale Invariant Feature Transform proposed in [11], visual control is getting widespread in robotics. However, few practical problems still affect visual servoing approaches and depend on the particular available robotic set-up. One such issue arising with limited field-of-view cameras is that of keeping the features in view during the robot manoeuvres, which has been addressed at times using omni-directional cameras ([11]), image path planning ([15]), or switching visual servoing schemes ([5]). The limited field-of-view (FOV) constraint is most often neglected, even in the more intuitive IBVS approaches. In the context of mobile robotics, the FOV problem has been successfully solved for a unicycle–like vehicle in [16], [7].

An optimal solution for the visual-based parking problem of a differentially driven robot (essentially, a unicycle with no trajectory curvature bounds) has been provided very recently in the important paper [2]. The problem considered in that paper is equivalent to bringing the vehicle in a desired configuration, while keeping a specified feature in sight of a monocular, fixed camera. Based on the optimal (open-loop) paths thus obtained, a switched, homography–based, visual servoing scheme was later proposed in ([10]) to steer the vehicle along the planned trajectories in closed loop.

In this paper, we consider again the problem of finding shortest paths for a unicycle with a limited FOV camera to reach a desired configuration. With respect to [2], we describe optimal paths in the image space, so as to enable their execution by a purely IBVS controller, thus taking advantage of the robustness of the image-based approaches ([4], [7]).

In this paper we show that the optimal control synthesis presented in [2] is incomplete. Indeed, we introduce two new optimal control words, proving that there exist regions in the plane where they are the shortest ones. Unfortunately, it turns out that the analytic description of such regions in the 3D plane is not simple.

However, as we are mainly interested in an optimal synthesis in the image space, we translate all optimal 3-D paths in paths on the image plane, and provide a procedure to decide which is the optimal path to be applied for any given initial image. Feedback control along optimal paths in the image plane can then be obtained in a purely IBVS scheme, whose design relies on a set of Lyapunov controllers, each of which is in charge of a specific kind of maneuver. Experimental results on a real wheeled mobile robot equipped with a standard web–cam prove the validity of the proposed approach.

II. PROBLEM DEFINITION

Consider a unicycle–like nonholonomic mobile robot moving on a plane, where a right-handed reference frame \( \{W\} = \{O_w, X_w, Y_w, Z_w\} \) is fixed with the static environment. Without loss of generality, assume that the \( Z_w \) coordinate \( z(t) = 0 \), \( \forall t \), hence the state space of the mobile platform is \( \xi(t) = \begin{bmatrix} \end{bmatrix} \).
\((x(t), y(t), \theta(t))\), where the robot reference point \((x(t), y(t))\) is in the middle of the wheel axle and the robot orientation \(\theta(t)\) is zero when the vehicle heads to the \(X_o\) axis. Let the control inputs be \(u(t) = (v(t), \omega(t))\), where \(v(t)\) and \(\omega(t)\) are respectively the forward and angular velocities of the vehicle. Therefore, the system kinematic model is

\[
\dot{\xi} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ 0 \\ \omega \end{bmatrix} = f_v v + f_\omega \omega
\]

(1)

The mobile agent is equipped with a rigidly fixed pinhole camera with a reference frame \(\langle C \rangle = \{O_c, X_c, Y_c, Z_c\}\) such that the optical center \(O_c\) corresponds to the robot’s center \((x(t), y(t))^T\) and the optical axis \(Z_c\) is aligned with the robot’s forward direction. If the robot orientation is null \((\theta = 0)\), the \(Z_c\) axis is parallel to the \(X_w\) axis, with the same direction and the \(X_c\) axis is parallel to the \(Y_w\) axis, with opposite direction. Hence, giving the motionless feature coordinates in the fixed frame \(\langle W \rangle\) be expressed by \(wP = [x, y, z]^T\), letting \(H_w\) be the transformation matrix between \(\langle W \rangle\) and \(\langle C \rangle\) and assuming a pinhole camera model, the corresponding image features points will be:

\[
l_p = [x, y]^T = \frac{\alpha_x x}{T} \begin{bmatrix} \alpha_x & 0 & 0 & \alpha_y \end{bmatrix}^T
\]

(2)

where \([x, y, 1]^T\) are the feature coordinates in the camera frame, \(\alpha_x\) and \(\alpha_y\) are the focal lengths of the camera calibration matrix \(K_c = \text{diag}(\alpha_x, \alpha_y, 0)\), and \([x, y]^T\) are the features coordinates in the image frame (measured in pixels) \((9)\) Without loss of generality, consider a symmetric FOV, with characteristic angle \(\phi\):

\[
\phi = \arctan \left( \frac{x_h}{\alpha_x} \right)
\]

(4)

where \(x_h\) is the x image boundary.

The origin \(O_I\) of the image plane reference frame \(\langle I \rangle = \{O_I, X_I, Y_I\}\) is assumed to be coincident with the principal point — i.e. the intersection of the camera axis (or \(Z_c\)) with the image plane. This choice simplifies the mathematical definition of the shortest paths on the image plane and, consequently, the image control design. Nevertheless, it is not crucial and any another choice can be made.

In the visual servoing literature, whenever an eye–in–hand configuration is considered (as is a camera rigidly fixed on a moving platform), the objective of the control task is to stabilize the robot towards the desired position controlling the camera position \((3), (4), (13)\). More precisely:

**Definition 1:** Given the desired and the current robot positions, which correspond the desired \(\langle C_d \rangle = \{O_{cd}, X_{cd}, Y_{cd}, Z_{cd}\}\) and the current \(\langle C_c \rangle = \{O_{cc}, X_{cc}, Y_{cc}, Z_{cc}\}\) reference frames respectively, the stabilization in the desired position is accomplished if \(\langle C_c \rangle \equiv \langle C_d \rangle\) at the end of the control task.

Indeed, as is customary in the visual servoing literature, \(\langle W \rangle \equiv \langle C_d \rangle\) (in our case \(X_w = Z_{cd}, Y_w = -X_{cd}\) and \(Z_w = -Y_{cd}\)), hence stabilizing the robot in the desired position corresponds to \(\ddot{\xi}(t) \to 0\) as \(t \to +\infty\).

In this paper, full camera calibration is assumed. Moreover, the robot is considered stabilized if definition 1 holds. More in depth, definition 1 is substituted with:

**Definition 2:** Given \(n\) desired and current image feature positions, \(F_d = [x_{d1}, y_{d1}, x_{d2}, y_{d2}, \ldots, x_{dn}, y_{dn}]^T\) and \(F_c = [x_{c1}, y_{c1}, x_{c2}, y_{c2}, \ldots, y_{cn}]^T\) respectively, the servoing task is accomplished if at the end of the control task is \(F_d = F_c \Rightarrow x_{di} = x_{ci}\) and \(y_{dj} = y_{cj}, \forall i = 1, \ldots, n\).

The optimal trajectories of the vehicle can be determined by a set of admissible controls given a bounded velocity. Admissible controls are bounded Lebesgue measurable functions in a time interval \([0, T]\) in \(\mathbb{R}^2\). Similarly to \((2)\), the set of admissible controls considered here is

\[
U = \{(v, \omega) : |v| \leq 1, |\omega| \leq 1\}.
\]

III. SHORTEST PATHS

In \((2)\), it has been shown that shortest paths are comprised of three different kinds of maneuvers: rotations on the spot, straight lines (represented by the symbol \(SL\)) and right or left logarithmic spirals (referred as \(T_1\) and \(T_2\) curves, respectively). Due to the physical and geometrical constraints of this problem, the language of optimal paths does not include all words generated by the previous three “symbols”. Given a goal point \(P\), the words of the language of the shortest paths, induce a partition of the plane into regions. The word univocally associated to a region encodes the shortest path from any point \(Q\) in that region to the goal point. Table I and figure 1 reports all the admissible words and related regions according to \((2)\).

The previous taxonomy does not include the words \(SL - T1\times T2\) and \(SL - T2\times T1\) which, instead, achieve the shortest path in some regions of the plane.

Indeed, let us consider the geometrical configuration depicted in figure 2. The point \(Q\) is in Region IV, hence, according to table I, the shortest path from \(Q\) to \(P\) is supposed to be of type \(T2Q\times T1P\). In fact, however, we show below that a path of type \(T2Q\times T1 - SL\) proves to be shorter for some \(Q\) in that region. In particular, we will show that the path \(PMNR\) is shorter than \(PR\) (the spiral segment \(RQ\) is common to both candidate optimal paths). Assume to have a reference frame \(\{O, X, Y\}\), such that the \(X\) axis is on the
TABLE I
ADMISSIBLE WORDS AND RELATED REGIONS. The symbols "" and "'" denote a non-smooth and smooth transition between segments, respectively.

<table>
<thead>
<tr>
<th>Region</th>
<th>Type of path</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>SL</td>
</tr>
<tr>
<td>I'</td>
<td>SL</td>
</tr>
<tr>
<td>II</td>
<td>T1Q − SL</td>
</tr>
<tr>
<td>II'</td>
<td>T2Q − SL</td>
</tr>
<tr>
<td>III</td>
<td>SL − T2p</td>
</tr>
<tr>
<td>III'</td>
<td>SL − T1p</td>
</tr>
<tr>
<td>IV</td>
<td>T1Q * T2p</td>
</tr>
<tr>
<td>V</td>
<td>T2Q * T1p</td>
</tr>
</tbody>
</table>

Fig. 2. Example of new path of type T2Q * T1 − SL.

line for O and P. Using polar coordinates, the point P and the logarithmic spiral passing through P can be represented as follows:

\[ P : (p_P, 0) \quad T_P : (p_P e^{-\theta_t}, \theta) , \]

where \( t = \frac{\cos \phi}{\sin \phi} \). The intersection point \( R \) between \( T_P \) and \( T_R \) can be written as a generic point belonging to \( T_P \) as

\[ R : (p_R, \theta_R) = (p_P e^{-\theta_t'}, \theta_R) , \]

and

\[ T_R : (p_R e^{(\theta - \theta_W)t'}, \theta) . \]

\( C_M \) is one of the two arcs of circle, from P to O, bounding the Region I', namely the region where each point can be achieved from P by a straight line. The point M belongs to the segment of \( C_M \) bounded by P and W (intersection point of \( C_M \) and \( T_R \)):

\[ C_M : \left( \frac{p_P \sin(\phi - \theta)}{\sin \phi} , \theta \right) \quad \theta \in [0, \phi] \]

\[ M : (p_M, \theta_M) = \left( \frac{p_P \sin(\phi - \theta_M)}{\sin \phi} , \theta_M \right) \quad \theta_M \in [0, \theta_W] , \]

where \( \theta_W \) is such that the following intersection relation holds

\[ e^{-2\theta_t'} = e^{-\theta_W t' \frac{\sin(\phi - \theta_W)}{\sin \phi}} . \quad (6) \]

Fig. 3. Length comparison of the two paths PR and PMNR.

The logarithmic spiral passing through \( M \) is given by

\[ T_M : \left( \rho_M e^{(\theta_M - \theta_t')}, \theta \right) . \]

It intersects \( T_R \) in

\[ N : (\rho_N, \theta_N) = \frac{\rho_P e^{\theta_R'} e^{\theta_M} \sin(\phi - \theta_M) \sin \phi}{\sin \phi} \] \[ \theta_N = \theta_R + \frac{\theta_M}{2} + \frac{\sin \phi}{2 \cos \phi} \ln \left( \frac{\sin(\phi - \theta_M)}{\sin \phi} \right) . \]

We are now ready for the computation of all the lengths of the segments making up the path PMNR:

\[ \frac{PM}{\sin \phi} = \frac{\rho_M - \rho_N}{\cos \phi} \]

\[ \frac{NR}{\cos \phi} = \frac{\rho_R - \rho_N}{\cos \phi} . \]

It is worth noting that if \( \theta_M = 0 \), that is \( M = P \), the path PMNR degenerates into PR, hence \( \frac{PM}{NR} = 0 \) and \( \frac{MN}{PR} = \frac{\rho_P - \rho_N}{\cos \phi} . \)

A choice of the parameters compliant with the configuration shown in figure 2, is the following: \( \theta_R = \frac{\pi}{2}, \rho_P = 100, \phi = \frac{\pi}{2} \). It is apparent from figure 3, that any path PMNR for any choice of \( \theta_M \in (0, \theta_W) \), is shorter than the path PR. By solving numerically equation (6) with the previous parameters, we found \( \theta_W = 0.7224 \) rad. The same analysis holds true also for a starting point \( Q' \) symmetric to \( Q \) w.r.t. the straight line passing through the points \( P \) and \( O \). The same argument applies also to any point \( R \) belonging to the logarithmic spiral \( T_P \) out of the points \( O \) and \( P \) (i.e. for negative values of the angle \( \theta_W \)); this fact becomes apparent if one swaps the point \( P \) with the point \( R \).

The existence of shorter paths in region where \( T1Q * T_P \) or \( T1P * T_Q \) were assumed to be the shortest ones, calls for a finer partition of the motion plane. In particular, Regions II, II', IV, V have to be subdivided again. In particular, there exists a portion of Regions IV, V, where the shortest paths are of kind \( SL - T1 * T2p \) or \( SL - T2 * T1p \), and a portion of Regions II, II' where the shortest paths are of kind \( T1 * T2 - SL \) or \( T2 * T1 - SL \). We will refer to these regions as Region VI, VII, VIII, and IX, respectively.
Unfortunately, it turns out that the analytic description of the correct partition in the 3D plane is not simple. However, as we are mainly interested in an optimal synthesis in the image space, we translate all optimal 3-D paths in paths on the image plane, and provide a procedure to decide which is the optimal path to be applied for any given initial image.

IV. Trajectories on the Image Plane

The main objective of the proposed solution is to make a controlled parking of a nonholonomic mobile platform equipped with a limited FOV camera using only image information. Furthermore, the trajectories that guide the robot to the desired posture should be optimal, that is the shortest in the 3-D working space of the vehicle. As shown in the previous sections, the optimal paths are words in a certain alphabet, whose elements are the rotations on the spot, the straight line or the logarithmic spiral. The image-based control is then feasible once the words are translated from the 3-D world to the image space, i.e. when an equivalent function is further used, where

$$I_{\text{original}}(s) = I_{\text{image}}(s)$$

is further used, where $\omega(t)$ is the angular velocity of the unicycle and $\Delta$ is the orientation variation between initial and final positions. As it will be shown in the rest of the paper, $\Delta$ estimation is the choice of this paper, that can be directly measurable on the image plane using epipolar geometry. For, at least eight points in non singular configuration (e.g. with the same $x$ coordinate) are needed, even though, for the sake of robustness and precision of calculation, more features are desirable.

A. Pure Rotation

Consider the image Jacobian, setting $\nu = 0$ and constant angular velocity, $\omega = \bar{\omega}$. The image Jacobian becomes:

$$I_{\text{original}} = \begin{bmatrix} I_{x_1} & I_{y_1} \end{bmatrix}$$

where

$$I_{\text{original}} = [I_{x_1}, I_{y_1}]^T$$

is the current image feature position. Substituting the integral of (7) in (8) gives

$$I_{y_c} = \frac{I_{y_1} \cos \left( \arctan \left( \frac{I_{x_1}}{I_{y_1}} \right) \right)}{\cos \left( \arctan \left( \frac{I_{x_1}}{I_{y_1}} \right) \right)},$$

the equation of a conic (see figure 4, solid line), the intersection between the image plane and the cone with vertex in the camera center (optical center) and base circumference passing through the 3-D feature position, independent from the initial feature position $I_{\text{original}} = [I_{x_1}, I_{y_1}]^T$.

Consider now only (7), which, by separation, and integration between the initial time instant $t_i$ and the final time instant $t_f$ gives

$$\theta - \theta_0 = \arctan \left( \frac{I_{x_d}}{\alpha_0} \right) - \arctan \left( \frac{I_{x_i}}{\alpha_0} \right),$$

where $I_{\text{target}} = [I_{x_d}, I_{y_d}]^T$ is the desired (final) feature position, and $\theta_0$ is the initial value of $\theta$, the robot orientation. Therefore, the angle variation can be computed for pure rotations by means of (10).

B. Pure Translation

Consider again the image Jacobian, setting $\nu = 0$ and a generic constant $\nu = \nu$. The image Jacobian, after integration, yields

$$I_{x_c} = \frac{I_{x_1} \nu y_c}{r_{y_1}}, \quad I_{y_c} = -\frac{I_{x_1} \nu y_c}{r_{y_1}},$$

and, eliminating time dependence,

$$I_{y_c} = \frac{I_{y_1}}{I_{x_1}} I_{x_c}.$$  \hspace{1cm} (11)

Equation (11) describes a straight line passing through the initial position of the feature and the principal point (see figure 4, dashed line), trajectory of an image feature for pure translation.

C. Logarithmic Spiral

The logarithmic spiral is completely determined by its characteristic angle $\phi$, defined in equation (4). Since such an angle remains constant as the robot travels on the spiral, the coordinates of the image feature $I_{x_c}$ should be constant and equal to the image horizontal boundary $x_b$. Therefore, the image plane trajectory for the logarithmic spiral will be simply a straight line on the image (see figure 4, dash–dotted line).
V. IMAGE PLANE OPTIMAL PATHS

Once the "alphabet" has been defined, the rules to construct the optimal words and the choice of the correct path are needed. We remark that for a path to be feasible it is required that the feature never get out from the FOV during the motion of the vehicle. Whereas the definition of the optimal language for the 3–D paths induces a partition of the plane in optimal regions, an analogous partition on the image plane is not immediate. Hence, since we want to perform an image–based parking avoiding, as much as possible, the use of 3–D information, we need a different selection mechanism for the choice of the shortest path. The paths are divided in three groups hierarchical ordered as follow: SL, SL−T, and the third group comprising both T1∗T2 and SL−T1∗T2. The selection of a group is performed when any path of all the previous groups (previous w.r.t. the hierarchy) is not feasible. For the first and the second group, there exists only one path satisfying the constraint IV, hence a feasibility check is easily accomplished. Unfortunately, there exist more that one feasible path of type SL−T1∗T2 satisfying the constraint IV. Therefore, the choice of the shortest path among the feasible ones in the third group needs some additional information provided by the 3–D reconstruction (see section V-C).

A. Unconstrained Path

To realize an optimal image path for Region I or Region I’, we have to control features towards their final positions and to compensate for the angle between the current and desired images. Hence, all the features on the image have to move along: a piece of a conic passing through the initial positions, which corresponds to a pure rotation of θa; a piece of straight line passing through the principal point, which corresponds to a pure translation; and, finally, a piece of conic passing through the desired positions corresponding to a pure rotation of θb. Figure 5 shows a typical unconstrained optimal path for an image plane feature.

If Δ is the angle between the current and desired images, then

\[ \theta_a + \theta_b = \Delta = \left[ \arctan \left( \frac{t_3}{a_3} \right) - \arctan \left( \frac{t_3}{a_3} \right) \right] + \left[ \arctan \left( \frac{t_4}{a_4} \right) - \arctan \left( \frac{t_4}{a_4} \right) \right]. \]

(see figure 5). The unknown values are only \( t_1 \) and \( t_2 \), as \( x_b \), \( t_1 \) and \( t_2 \) are the x-axis coordinate of the image bound, the initial and the final feature’s position respectively. However, \( t_1 \) and \( t_2 \) are linearly dependent, as they belong to the same straight line. Moreover, \( t_2 \) is the x-axis coordinate of the intersection point between the straight line passing by \( t_1 \) and the conic passing by \( t_3 \). As a consequence, it is possible to express the last equation as a function of \( t_1 \) only, hence in a unique solution. Δ angle is directly computed by comparing the desired and current views. A common approach is to use the fundamental matrix, an important tool in epipolar geometry (see for details [9], [12], [14]).

In the case of multiple features, all feature trajectories are of the same category, even though some of them can get out from the FOV. Notice that, if all the features must be kept inside the FOV, region I and I’ can be evaluated as the intersection of all regions of type I and I’ simply computing the unconstrained path for all the features. Although desirable, this is not strictly needed in our implementation (as instead necessary in [10]).

B. Constrained optimal path: Region II, II’, III and III’

If the unconstrained optimal path is not feasible, , the robot is not in the Region I or Region I’. Note that regions close to Region I and I’ are respectively Region III or III’ and Region II and II’ (recall figure 1).

Similarly to the realization of the unconstrained optimal path and given the hypothesis that the robot position pertains to Region III, if Δ is the angle between the initial and final (or desired) image, then the path in figure 6 is determined solving the following equation:

\[ \Delta = \left[ \arctan \left( \frac{t_1}{a_1} \right) - \arctan \left( \frac{t_1}{a_1} \right) \right] + \left[ \arctan \left( \frac{t_2}{a_2} \right) - \arctan \left( \frac{t_2}{a_2} \right) \right] + \left[ \arctan \left( \frac{t_3}{a_3} \right) - \arctan \left( \frac{t_3}{a_3} \right) \right], \]

where the second addendum in the second member is the angle variation of the robot’s orientation θ during the spiral path. In this equation the unknown value are only \( t_1 \) and \( t_2 \), as \( x_b \), \( t_1 \) and \( t_2 \) are the x-axis coordinate of image bound, the initial and the final feature’s position respectively, whereas \( t_3 \) is the intersection between the conic passing by the feature final position and the image bound. However, \( t_1 \) and \( t_2 \) are x-axis and y-axis coordinates of two points on the same straight line, that is linearly dependent. As a consequence, it is possible to express the last equation as a function of \( t_1 \) (or \( t_2 \)), arbitrarily. It is worthwhile to note that the solution is, again, unique. For the other regions the trajectory of the feature is likewise calculated.

In the case of multiple features, this trajectory can be calculated for one particular feature (principal feature), that is closer to the image boundary, choosing the correct boundary through the sign of Δ. Using the principal feature path, the trajectories of all other features are computed by geometric reconstruction on the image plane.

C. Constrained optimal path: Region IV, V, VI and VII

If optimal paths of type SL−T2p, SL−T1p, T2−SL or T1−SL are not feasible on the image plane (e.g. the first rotation brings the feature out of the image plane), the vehicle is inside either region IV, V, VI or VII. Due to space
limitation, only the trajectory inside Region VII is analyzed (others are similar). If Δ is the angle between the initial and final (or desired) image, then the path in figure 7 is determined solving the following equation:

$$\Delta = \arctan\left(\frac{I_1 y}{x} - \arctan\left(\frac{I_2 y}{z}\right)\right) + \frac{\pi}{\alpha_4} \ln\left(\frac{I_2 y}{I_3 y}\right) + 2\left[\frac{\pi}{\alpha_4} \ln\left(\frac{I_2 y}{I_3 y}\right) + \arctan\left(\frac{I_2 y}{I_3 y}\right) - \arctan\left(\frac{I_1 y}{z}\right)\right].$$

In this case it is not possible to express the equation as a function of an unique unknown value, as $I_4 y$ is completely independent from all the other variables. More precisely, for each $I_4 y$, there exists a valid image path. Therefore, The choice of the optimal path in this last case can be done only by the use of its 3–D reconstruction (since both the path $T1 - T2p$ or $SL - T1 - T2p$ are feasible). It is worthwhile to note that such a reconstruction does not need to be exact, since a scaled one is sufficient. Hence, the optimal path is determined on a generic plane parallel to the plane of motion. Notice that while the controller is based on a single point feature trajectory, therefore it is purely image–based, such a trajectory need 3–D information to be computed (basically, the fact that the robot moves on a plane, the camera calibration and the angle scaled estimation).

In the case of multiple features, the image trajectories for all the features, except the principal feature path, get out of the image plane since the principal feature goes from one horizontal boundary (during the first spiral) to the other (for the second spiral). Although the robustness of the overall strategy may dramatically decrease, this is a side effect of choosing optimal trajectories. Nevertheless, the choice of an image based control law can still preserve convergence, while a position based strategy (as the homography based in [10]) is completely compromised (position reconstruction with only one feature is not possible at all).

D. Path Implementation

All the optimal paths reported previously are implemented via the fundamental matrix estimation. Therefore, at least eight features are needed, as in [10]. At the moment, the main difference between our method and the one reported in [10] is that in our case we need the angle estimation only in the initial position, while in [10] a continuous homography estimation is needed along the path. Hence, even though the feasibility is both related to the feature point richness, in this implementation only a subset (at least one) of the initially selected feature is needed to complete the path make it more suitable for practical application (a feature could be lost for tracking inefficiency, noise or light changing).

The proposed solution is an image feature planning that is computed once and for all and then the control is a pure image feature trajectory tracking. Hence, the overall control accuracy suffers of all the side effects of open loop schemes and relies on correct SIFT matching.

VI. EXPERIMENTAL RESULTS

A. Practical Implications

In order to implement a feature tracking based control, some practical issues should be taken into account on the feature motion. Recalling (2), $I_3 y \to 0$ if $c y \to 0$ or $c z \to +\infty$, Hence, for the time derivative of (2), i.e. the image jacobian, it can be shown that

$$\begin{align*}
\lim_{c y \to 0} [I_3 x, I_3 y] &= \begin{cases} I_3 x + \frac{I_3 y + \alpha c}{\alpha_4} \omega & \text{for } c y \to 0 \\
\lim_{c y \to 0} [I_3 x, I_3 y] &= \begin{cases} I_3 x + \frac{\alpha_2}{\alpha_4} \omega & \text{for } c z \to +\infty
\end{cases}
\end{cases}
\end{align*}$$

The same effects are also observed for quantization effects or noise in the feature extraction. Therefore:

- In (12), $v$ cannot be observed from the optical flow $[I_3 x, I_3 y]$;
- Regardless the current feature position $[I_3 x, I_3 y]$ or the current distance $c z$, a well established relation exists between the horizontal feature motion $I_3 x$ and the robot’s angular velocity $\omega$;
- Indirect measurements of $\omega$ through $I_3 y$ are instead more sensitive to noise or to the distance $c z$. This fact gets worse when $I_3 y \to 0$.

B. Implementation

A parking experiment was realized in order to validate the effectiveness of the proposed technique. The implementation consisted of the following steps:

- Feature detection: SIFT descriptors [11] are matched between initial and final images. An affine transformation model is fit for all matched points and the 8 ones
with the smallest residual error are chosen to further calculate the optimal trajectory. Features in which $|y| < 20$ were discarded for the practical reasons seen above.

- **Feature trajectory calculation:** the tracked feature is chosen and its path is computed as described in section III. If more than one feature can be used, the one with the smallest residual error is chosen;

- **Feature tracking and robot control:** the feature is tracked using its estimated position calculated using the affine transformation model given by the SIFT based recognition.

### C. Control Design

The technique was evaluated for initial configurations that correspond to Regions I, III and III'. The robot’s proportional control laws $u = (v, \omega)$ are obtained using a simple, quadratic Lyapunov based controller. For each image trajectory component a different Lyapunov function is chosen, in order to minimize the feature errors. More precisely, given the image jacobian
\[
\begin{cases}
  I_x = \frac{1}{c_2^2} v + \frac{r^2 + a_2^2}{a_1} \omega \\
  I_y = \frac{1}{c_2^2} v + \frac{r^2 + a_2^2}{a_1} \omega
\end{cases}
\]
the control laws will be

1) **Pure rotations:** since forward velocity $v = 0$ to enhance robustness, the Lyapunov function is
\[
\begin{align*}
  V(t_x) &= \frac{1}{2} (t_x - t_{x_d})^2 \\
  \dot{V}(t_x) &= (t_x - t_{x_d}) \frac{1}{2} (t_x - t_{x_d}) + \omega (t_x - t_{x_d}) \gamma(t_x) \omega
\end{align*}
\]
with $\gamma(t_x) > 0$, $\forall t_x$. Hence, choosing $\omega = -(t_x - t_{x_d})$ yields to $t_x \rightarrow t_{x_d}$ and $t_y \rightarrow t_y$ (see equation (9));

2) **Pure Translations:** since angular velocity $\omega = 0$ to enhance robustness, the Lyapunov function is
\[
\begin{align*}
  V(t_x, t_y) &= \frac{1}{2} (t_x - t_{x_d})^2 + \frac{1}{2} (t_y - t_{y_d})^2 \\
  \dot{V}(t_x, t_y) &= \frac{1}{2} [(t_x - t_{x_d})^2 + (t_y - t_{y_d})^2] v
\end{align*}
\]
Recalling (11), the time derivative of the Lyapunov function will be
\[
\dot{V}(t_x, t_y) = \frac{1}{c_2^2} t_z + \frac{t_x^2 + t_y^2}{c_2^2} (t_y - t_{y_d}) t_v.
\]
Since $t_{x_d}$ and $t_{y_d}$ are the initial feature positions and $\frac{1}{c_2^2} + \frac{1}{c_2^2} > 0$, choosing $v = -(t_y - t_{y_d}) t_v$ makes $\dot{V}(t_x, t_y) < 0$ except in $t_y = t_y_d$. Notice that $t_y = 0$ is not feasible for a straight line path of equation (11). $t_x \rightarrow t_{x_d}$ comes from (11).

As $t_y$ is less sensitive to $[v, \omega]$, and, more sensitive to noise, for the SL movement correspondent to Region I, the feature error is substituted by an image error given by the scale factor extracted from the affine transformation model between the actual image and the final one. The choice of the image scale is suitable since at the end of the manoeuvre, the robot reaches the desired position with only an orientation error. Hence: $[v, \omega] = [-\delta(F), 0]$ where $F$ is the set of the tracked feature points and $\delta(\cdot)$ is the affine transformation function. Instead, for the SL trajectory of Regions III and III', the robot ends the linear trajectory as soon as it reaches the image boundary, hence not in the desired position. As a matter of fact, the function $\delta$ is substituted with a the sum of cartesian distances between current and desired positions of $F$.

3) **Spiral trajectories:** the controller in the spiral case can be viewed as a combination of the two previously reported controllers since a spiral trajectory can be approximated by a set of linear trajectories in the 3–D robot space. For the sake of robustness, the feature error is substituted by an image error as in the previous case.

This way a feature based controller through via–points is defined. A major drawback of the proposed technique is that it completely relies on the correct SIFT matching.

### D. Specifications

The experimental setup was comprised of a Quickcam Ultravision camera mounted over the front-part of a K-team Koala vehicle. The controller is implemented under Windows XP on a 1.2 GHz Intel Core Single UMPC. The ERSP vision library is used to perform SIFT recognition ([8]) and Matlab libraries are used to compute the proposed algorithm. The image resolution was 320x240 pixels and the controller rate was around 7 Hz.

### E. Results

The first experiment related to an initial position of the robot in Region I is shown in figure 8. A second experiment related to Regions III and III', is shown in Fig. 9. Results are summarized in table II.

The method accomplished the task of driving the tracked feature through the planned path. It can be observed from the figure that the final positions of the features are almost coincident with the desired positions. The final feature position error is due to the tracking noise and to the estimation of the angle $\Delta$ (that determines the feature trajectories). Hence,
The robustness of the overall algorithm must be increased, nevertheless the IBVS approach seems to be promising for an effective application in real world environments. The adoption of robust recognition systems, performant feature trackers and feature estimation filters may increase the applicability of the proposed technique. Robustness to uncalibrated camera parameters should also be considered. Furthermore, the technique should be generalized to cope with known obstacles in the robot path.

VIII. ACKNOWLEDGMENTS

This work was supported by the PHRIENDS Specific Targeted Research Project, funded under the 6th Framework Programme of the European Community under Contract IST-045359. The authors are solely responsible for its content.

REFERENCES