

Visual Servoing: Reaching the Desired Location Following a Straight Line via Polynomial Parameterizations

Graziano Chesi, Domenico Prattichizzo and Antonio Vicino

Department of Information Engineering

University of Siena

Via Roma 56, 53100 Siena, Italy

{chesi,prattichizzo,vicino}@dii.unisi.it

Abstract—The problem of establishing if it is possible to reach the desired location keeping all features in the field of view and following a straight line is considered. The proposed approach is based on the polynomial parameterization of the camera path and allows one to find the path that follows a straight line and maximizes the distance of the image trajectories from the screen boundary.

Index Terms—Visual servoing, Point correspondences, Straight line, Path planning, Polynomial trajectory.

I. INTRODUCTION

The “teaching-by-showing” approach has gained the attention of several researchers in the last years due to its dynamic nature that allows for guiding a robot in partially or even completely unknown environments (see for example [1]–[3] for detailed classifications). Many methods for dealing with such a task have been proposed. Some examples are the position-based visual servoing (PBVS) where the feedback error is the camera pose (see for example [4], [5]), and the image-based visual servoing (IBVS) where the feedback error is the image error (see for example [6], [7]). In the 2 1/2 D visual servoing [8] the feedback error contains both the camera pose and the image error. In partitioning methods, the degrees of freedom of the camera are grouped according to some criteria and differently controlled (see for example [9]–[11]). Other methods, such as visual servoing based on navigation functions [12], path-planning techniques [13]–[15], visual servoing invariant to the intrinsic parameters [16], visual servoing based on image moments [17] and methods based on the generation of circular-like trajectories [18], have been proposed in the literature.

The aim of these methods is that of guaranteeing convergence, features in the field of view, robustness against uncertainty on calibration parameters and image noise, joints physical constraints matching and/or achievement of some performances. A problem of theoretical and practical interests that has not been completely investigated yet is to establish whether and how the camera may reach the desired location keeping all features in the field of view and following a straight line in the euclidean space. Indeed, methods as [4], [15], [19] steer the camera along a straight line but either they do not take into account the visibility constraint or they may fail in finding a solution even if it exists since they consider only a subclass of the camera paths. This problem is important because, by following

a straight line, the camera moves along the shortest path hence minimizing the probability of leaving the workspace.

In this paper, the problem of establishing if there exists a path in the rotational space that combined with a straight translational path allows the camera to reach the desired location keeping all features in the field of view is considered. Specifically, the aim consists of finding among such camera paths that maximizing the distance of the image trajectories from the screen boundary. To this end, a scaled euclidean reconstruction of the scene is computed from the current and desired views of the features. A parametrization of the camera trajectory is introduced by linearly parameterizing the translational path and polynomially parameterizing the rotational path through its Cayley parameter. The arbitrary degree that can be used to represent the rotational path makes complete the above parametrization in the sense that all possible solutions can be considered. Then, it is shown that the problem of calculating the distance of the image trajectories from the screen boundary for a fixed path can be solved by computing the real roots of some polynomials in a certain interval. The optimal path is hence computed by maximizing such a distance with respect to all possible paths. A suitable initialization for this optimization is found by solving a simplified problem which returns a path that keeps all points in front of the camera. Finally, the camera can be steered along the computed path by following the image trajectories through an IBVS controller.

II. PRELIMINARIES

A. Notation and problem formulation

Let \mathbf{I}_n denote the identity matrix $n \times n$, $\mathbf{0}_n$ the null vector $n \times 1$, $\mathbf{1}_n$ the vector $n \times 1$ with all elements equal to 1, \mathbf{e}_i the i -th column of \mathbf{I}_3 , and $[\mathbf{v}]_{\times}$ the skew-symmetric matrix of $\mathbf{v} \in \mathbb{R}^3$. Moreover:

- $\mathcal{F}, \mathcal{F}^*$: current and desired camera frames;
- $\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3$: rotation and translation of \mathcal{F}^* with respect to \mathcal{F} expressed in \mathcal{F} ;
- $\mathbf{A} \in \mathbb{R}^{3 \times 3}$: upper triangular intrinsic parameters matrix;
- $\mathbf{m}_i \in \mathbb{R}^3$ (resp., \mathbf{m}_i^*): image projections of the i -th point on \mathcal{F} (resp., \mathcal{F}^*) in normalized coordinates;
- $\mathbf{p}_i \in \mathbb{R}^3$ (resp., \mathbf{p}_i^*): image projections of the i -th point on \mathcal{F} (resp., \mathcal{F}^*) in pixel coordinates according to $\mathbf{p}_i = \mathbf{A}\mathbf{m}_i$ and $\mathbf{p}_i^* = \mathbf{A}\mathbf{m}_i^*$.

Let us suppose that a set $\mathcal{P} = \{(\mathbf{p}_i, \mathbf{p}_i^*), i = 1, \dots, n\}$ of n point correspondences between current and desired views is available. The problems dealt with in this paper are:

- to establish if it is possible to reach the desired location following a straight line in the euclidean space and keeping all features in the field of view;
- to find among such paths the one that maximizes the distance of the image trajectories from the screen boundary.

B. Cayley parameters

The proposed approach exploits the representation of rotation matrices through the Cayley parameters in order to obtain a formulation of the problem through polynomials. These parameters are defined as follows (see for example [20] for more details). Let \mathbf{R} be any matrix in $SO(3)$. Then, there exists $\mathbf{a} \in \mathbb{R}^3$ called *Cayley parameter*¹ of \mathbf{R} such that

$$\mathbf{R} = \frac{\mathbf{\Lambda}(\mathbf{a})}{1 + \|\mathbf{a}\|^2} \quad (1)$$

where

$$\mathbf{\Lambda}(\mathbf{a}) = \begin{bmatrix} 1 + (\mathbf{a}_1)^2 - (\mathbf{a}_2)^2 - (\mathbf{a}_3)^2 & 2((\mathbf{a}_1)\mathbf{a}_2 - (\mathbf{a}_3)) & 2((\mathbf{a}_1)\mathbf{a}_3 + (\mathbf{a}_2)) \\ 2((\mathbf{a}_1)\mathbf{a}_2 + (\mathbf{a}_3)) & 1 - (\mathbf{a}_1)^2 + (\mathbf{a}_2)^2 - (\mathbf{a}_3)^2 & 2((\mathbf{a}_2)\mathbf{a}_3 - (\mathbf{a}_1)) \\ 2((\mathbf{a}_1)\mathbf{a}_3 - (\mathbf{a}_2)) & 2((\mathbf{a}_2)\mathbf{a}_3 + (\mathbf{a}_1)) & 1 - (\mathbf{a}_1)^2 - (\mathbf{a}_2)^2 + (\mathbf{a}_3)^2 \end{bmatrix} \quad (2)$$

The relation is bijective. In fact, any $\mathbf{a} \in \mathbb{R}^3$ defines through the previous relation a matrix in $SO(3)$.

III. PATH PLANNING

A. Problem formalization

The first step for solving the considered problems is to compute \mathbf{m}_i and \mathbf{m}_i^* from \mathcal{P} and \mathbf{A} , and then the camera pose from these image points. Since \mathbf{t} can be recovered only up to a scale factor from image pairs, the achievable components are \mathbf{R} and $\mathbf{d} = \mathbf{t}/\|\mathbf{t}\|$. Their computation can be done through the essential matrix algorithm or the homography matrix algorithm in the case of non coplanar features supposing $n \geq 8$. If the features are known to be coplanar, the camera pose can be computed through the homography matrix algorithm supposing $n \geq 4$ (see [21] and [22] for details). From \mathbf{R} , \mathbf{d} and the image points, a scaled euclidean reconstruction of the set of 3D points can be obtained. Let us denote with $\mathbf{q}_i \in \mathbb{R}^3$ the i -th point of this reconstruction expressed in \mathcal{F} .

Let w be a real scalar in $[0, 1]$ and let us denote with $\mathbf{d}(w) = w\mathbf{d}$ the camera center in the scaled space along the line connecting the current location ($w = 0$) to the desired location ($w = 1$). Let $\mathbf{R}(w)$ and $\mathbf{m}_i(w, \mathbf{R}(w))$ be

the rotation matrix and the i -th image point along the path, and let us introduce the set of constraints

$$\begin{cases} \mathbf{R}(0) = \mathbf{I}_3, \mathbf{R}(1) = \mathbf{R} \\ x_m < (\mathbf{m}_i(w, \mathbf{R}(w)))_1 < x_M \quad \forall w \in [0, 1] \quad \forall i = 1, \dots, n \\ y_m < (\mathbf{m}_i(w, \mathbf{R}(w)))_2 < y_M \quad \forall w \in [0, 1] \quad \forall i = 1, \dots, n \end{cases} \quad (3)$$

where x_m, x_M, y_m and y_M are the screen limits in normalized coordinates. The set of feasible rotational paths is hence

$$\mathcal{R} = \{\mathbf{R}(w) \in SO(3) : (3) \text{ is satisfied}\}. \quad (4)$$

Let $\rho(\mathbf{R}(w))$ be the distance of the image trajectories from the screen boundary along the path defined as

$$\rho(\mathbf{R}(w)) = \inf_{w \in [0, 1], i=1, \dots, n} \rho_i(w, \mathbf{R}(w)) \quad (5)$$

$$\rho_i(w, \mathbf{R}(w)) = \inf \{(\mathbf{m}_i(w, \mathbf{R}(w)))_1 - x_m, x_M - (\mathbf{m}_i(w, \mathbf{R}(w)))_1, (\mathbf{m}_i(w, \mathbf{R}(w)))_2 - y_m, y_M - (\mathbf{m}_i(w, \mathbf{R}(w)))_2\}. \quad (6)$$

The problems stated at the end of Section II-A can be unified into the computation of the rotational path that achieves the maximum distance

$$\mu = \sup_{\mathbf{R}(w): \mathbf{R}(0)=\mathbf{I}_3, \mathbf{R}(1)=\mathbf{R}} \rho(\mathbf{R}(w)). \quad (7)$$

In fact, \mathcal{R} is not empty if and only if $\mu > 0$.

B. Polynomial approach

Computing the distance $\rho(\mathbf{R}(w))$ may be a difficult task depending on the dependency of $\mathbf{R}(w)$ from w . One way of simplifying such a task is to select a rational parameterization of $\mathbf{R}(w)$. This can be done by exploiting the Cayley parameters, in particular by *polynomially parameterizing* the Cayley parameter $\mathbf{a}(w)$ of $\mathbf{R}(w)$.

Hence, let us select $\mathbf{a}(w)$ polynomial in w of degree δ for some $\delta > 0$, and let us express $\mathbf{a}(w)$ as

$$\mathbf{a}(w) = \mathbf{D} [w^\delta, w^{\delta-1}, \dots, 1]^T \quad (8)$$

where $\mathbf{D} \in \mathbb{R}^{3 \times (\delta+1)}$ is a coefficient matrix. The constraints in (3) at the current and desired locations are satisfied if and only if

$$\mathbf{a}(0) = \mathbf{0}_3, \quad \mathbf{a}(1) = \mathbf{a}, \quad (9)$$

where \mathbf{a} is the Cayley parameter of \mathbf{R} . The matrix \mathbf{R} is supposed such that the norm of \mathbf{a} is finite². The constraints in (9) impose a certain structure to the matrix \mathbf{D} that can be parameterized as

$$\mathbf{D} = [\mathbf{C}, \mathbf{a} - \mathbf{C}\mathbf{1}_{\delta-1}, \mathbf{0}_3] \quad (10)$$

where $\mathbf{C} \in \mathbb{R}^{3 \times (\delta-1)}$ is a free parameter.

Let us denote by $\mathbf{a}(w, \mathbf{C})$ the Cayley parameter of the rotation matrix along the path parameterized by \mathbf{C} . In order

²As previously mentioned, the subset of $SO(3)$ where the norm of \mathbf{a} is infinite has null measure. However, the proposed technique can be extended to include such case by selecting $\mathbf{a}(w)$ rational in w with a pole in $w = 0$.

¹The norm of \mathbf{a} is infinite on a subset with null measure of $SO(3)$.

to take into account the visibility constraints, let us define the polynomials of degree $2\delta + 1$ in w

$$b_{i,j}(w, \mathbf{C}) = (\mathbf{q}_i - w\mathbf{d})^T \mathbf{\Lambda}(\mathbf{a}(w, \mathbf{C})) \mathbf{e}_j \quad (11)$$

for $i = 1, \dots, n$ and $j = 1, 2, 3$. We have that the i -th image point along the path is

$$\mathbf{m}_i(w, \mathbf{C}) = \left[\frac{b_{i,1}(w, \mathbf{C})}{b_{i,3}(w, \mathbf{C})}, \frac{b_{i,2}(w, \mathbf{C})}{b_{i,3}(w, \mathbf{C})}, 1 \right]^T. \quad (12)$$

In order to get rid of the denominator and manage polynomials instead of rational functions, let us introduce the index

$$\alpha(\mathbf{C}) = \min_{w \in [0,1], i=1, \dots, n} b_{i,3}(w, \mathbf{C}). \quad (13)$$

We have that the points lie in front of the camera along the trajectory if and only if $\alpha(\mathbf{C}) > 0$. Therefore, the distance of the image trajectories from the screen boundary along the path is

$$\rho(\mathbf{C}) = \begin{cases} \hat{\rho}(\mathbf{C}) & \text{if } \alpha(\mathbf{C}) > 0 \\ -\infty & \text{otherwise} \end{cases} \quad (14)$$

where

$$\hat{\rho}(\mathbf{C}) = \sup \varepsilon \quad (15)$$

subject to

$$\begin{cases} b_{i,1}(w, \mathbf{C}) - (x_m + \varepsilon)b_{i,3}(w, \mathbf{C}) > 0 \\ -b_{i,1}(w, \mathbf{C}) + (x_M - \varepsilon)b_{i,3}(w, \mathbf{C}) > 0 \\ b_{i,2}(w, \mathbf{C}) - (y_m + \varepsilon)b_{i,3}(w, \mathbf{C}) > 0 \\ -b_{i,2}(w, \mathbf{C}) + (y_M - \varepsilon)b_{i,3}(w, \mathbf{C}) > 0 \\ \forall w \in [0, 1] \quad \forall i = 1, \dots, n \end{cases} \quad (16)$$

Finally, the maximum distance achievable with a Cayley parameter of degree δ is

$$\mu_\delta = \sup_{\mathbf{C}} \rho(\mathbf{C}). \quad (17)$$

Clearly, it holds that $\mu_\delta \leq \mu$. Therefore, $\mu_\delta > 0$ implies that \mathcal{R} is not empty.

A method for efficiently computing $\rho(\mathbf{C})$ will be described in the next subsection. However, solving the optimization (17) may be a difficult task if the matrix \mathbf{C}_0 used to initialize the search for μ belongs to the interior of the set of matrices \mathbf{C} for which at least one point goes behind the camera along the path. In fact, in such a case, there are neighborhoods of \mathbf{C}_0 where the function $\rho(\mathbf{C})$ assumes a constant value, in particular $-\infty$, and the optimization algorithm used to solve (17) may not be able to get out from such neighborhoods. Therefore, an initialization \mathbf{C}_0 satisfying $\alpha(\mathbf{C}_0) > 0$ should be used. Such an initialization can be found by solving

$$\sup_{\mathbf{C}} \alpha(\mathbf{C}). \quad (18)$$

In particular, it is sufficient to run (18) until the current cost function $\alpha(\mathbf{C})$ becomes greater than zero. Notice that optimization (18) does not suffer of the initialization problem of (17).

C. Computation of the distance for a fixed path

Let us consider first the computation of $\alpha(\mathbf{C})$. This amounts to n computations of the minimum

$$\alpha_i(\mathbf{C}) = \min_{w \in [0,1]} b_{i,3}(w, \mathbf{C}) \quad (19)$$

and can be simply performed by evaluating $b_{i,3}(w, \mathbf{C})$ at the points where its derivative vanishes and at the extremes $w = 0$ and $w = 1$.

Let us consider now $\hat{\rho}(\mathbf{C})$. Clearly, this amounts to $4n$ computations (four for each point) of the form

$$\zeta = \sup \varepsilon \quad (20)$$

subject to

$$f_0(w) - \varepsilon f_1(w) > 0 \quad \forall w \in [0, 1] \quad (21)$$

where $f_1(w) > 0$ for all $w \in [0, 1]$ being $\alpha(\mathbf{C}) > 0$. The following properties hold:

- 1) there exists $\varepsilon > -\infty$ such that (21) is satisfied;
- 2) $\zeta < +\infty$;
- 3) let $\bar{\varepsilon}$ be such that $f_0(w) - \bar{\varepsilon} f_1(w) \leq 0$ for some $w \in [0, 1]$. Then, there does not exist any $\varepsilon > \bar{\varepsilon}$ such that (21) is satisfied.

Therefore, ζ can be computed as the smallest ε such that $f_0(w) - \varepsilon f_1(w) = 0$ has either one root in 0, or one root in 1, or a double root in the interval $(0, 1)$. Specifically:

$$\zeta = \min_{i=1,2,3} \varepsilon_i \quad (22)$$

where

$$\varepsilon_1 = \frac{f_0(0)}{f_1(0)} \quad (23)$$

$$\varepsilon_2 = \frac{f_0(1)}{f_1(1)} \quad (24)$$

$$\varepsilon_3 = \min \left\{ \frac{f_0(w)}{f_1(w)} : f_2(w) = 0, w \in (0, 1) \right\} \quad (25)$$

and $f_2(w)$ is the polynomial defined as

$$f_2(w) = \frac{df_0(w)}{dw} f_1(w) - \frac{df_1(w)}{dw} f_0(w). \quad (26)$$

Let us observe that the degree of $f_2(w)$ is 4δ since $f_0(w)$ and $f_1(w)$ are polynomials of degree $2\delta + 1$ and the coefficient of the monomial of degree $4\delta + 1$ in $f_2(w)$ is always zero.

Summarizing, the computation of $\alpha(\mathbf{C})$ amounts to n root-computations of polynomials of degree 2δ , and the computation of $\hat{\rho}(\mathbf{C})$ amounts to $4n$ root-computations of polynomials of degree 4δ .

D. Remarks

Following the planned path. Once the path has been computed, the camera can be steered to the desired location along such a path by using an IBVS controller that allows to follow image trajectories (see for example [13] for details about such a controller). The image trajectory of the i -th point corresponding to the matrix \mathbf{C} is given by

$$\mathcal{T}_i(\mathbf{C}) = \{\mathbf{m}_i(w, \mathbf{C}), w \in [0, 1]\}. \quad (27)$$

Effects of calibration errors. Calibration errors do affect the proposed method. Indeed, in presence of such uncertainties the estimated camera pose and scaled euclidean reconstruction are affected by errors. Consequently, by following the image trajectories $\mathcal{T}_i(\mathbf{C})$, the camera does not move along a straight line in the euclidean space but along a different curve whose distance from the sought line grows as the calibration errors increase.

IV. NUMERICAL EXAMPLES

The following numerical case has been considered: $\mathbf{A} = [400, 0, 300; 0, 400, 200; 0, 0, 1]$ pixels (supposed known), screen with limits $(0, 600)$ pixels for the x -axis and $(0, 400)$ pixels for the y -axis, $n = 8$. Problems (18) and (17) have been solved through the function FMINSEARCH of MATLAB by evaluating $\alpha(\mathbf{C})$ and $\hat{\rho}(\mathbf{C})$ as described in Section III-C. Problem (17) has been initialized with the solution of problem (18), and problem (18) has been initialized with $\mathbf{C} = \mathbf{0}_{3 \times (\delta-1)}$.

A. Example 1

Figure 1a shows the current and desired locations denoted by \mathcal{F} and \mathcal{F}^* , and the observed features corresponding to the biggest white circles. The current and desired image points are reported in Table I.

i	$(\mathbf{p}_i)_1$	$(\mathbf{p}_i)_2$	$(\mathbf{p}_i^*)_1$	$(\mathbf{p}_i^*)_2$
1	346.5	306.1	441.5	120.9
2	366.3	329.7	473.9	155.2
3	386.9	354.3	512.0	196.5
4	274.3	252.1	185.3	128.0
5	267.1	309.2	185.3	197.3
6	287.6	323.5	254.7	197.3
7	292.8	271.5	254.7	128.0
8	281.1	288.6	220.0	162.7

TABLE I
IMAGE POINTS FOR EXAMPLE 1.

We have initially selected $\delta = 1$ (corresponding to a linear trajectory in the Cayley parameter space and, hence, to an empty matrix \mathbf{C}) finding $\mu_1 = 0.0080$ that is a feasible path. However, the image trajectories are quite close to the screen boundary being the distance in pixels equal to $400\mu_1 = 3.2$. In order to increase this distance we can use larger values of δ . In particular, with $\delta = 2$ we find $\mu_2 = 0.1143$ whose corresponding path is shown in Figures 1a–1c.

B. Example 2

Table II reports the image points corresponding to Figure 2a.

This time $\delta = 1$ corresponds to a non feasible path being $\mu_1 = -0.0830$. With $\delta = 2$ we have found $\mu_2 = 0.2928$ whose corresponding path is shown in Figures 2b and 2c.

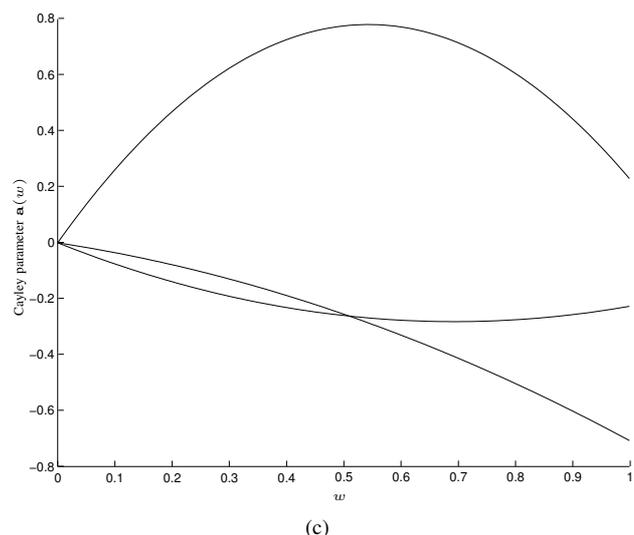
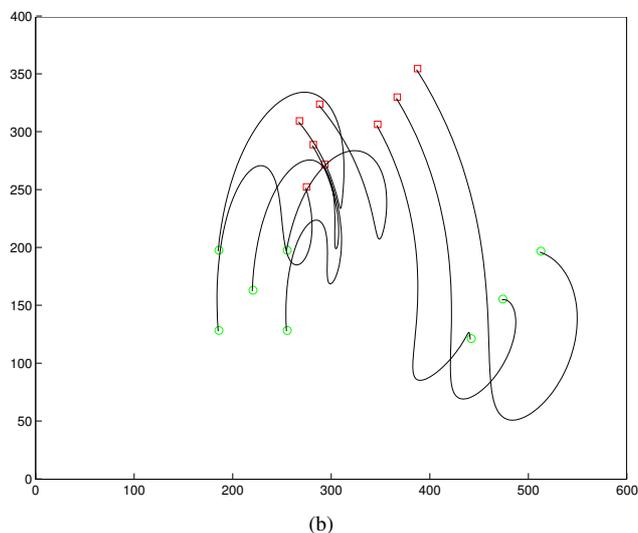
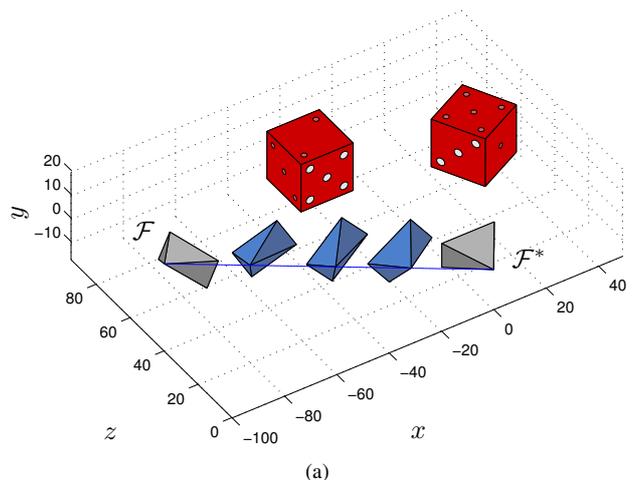
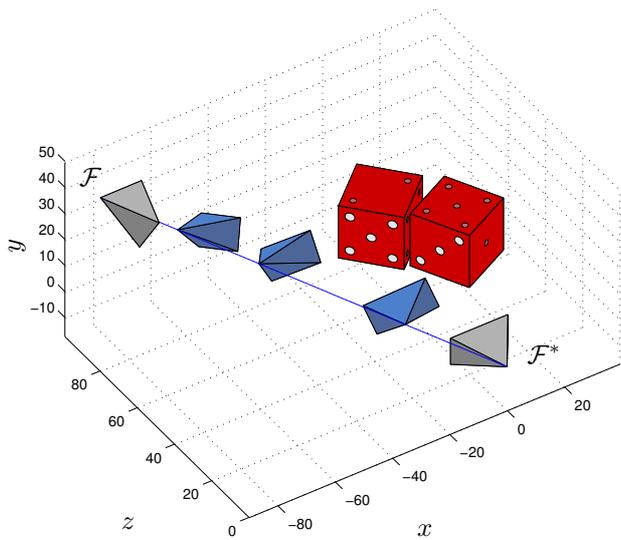
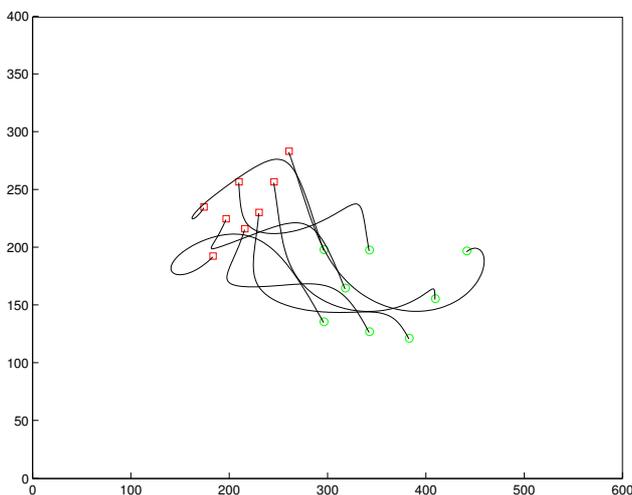


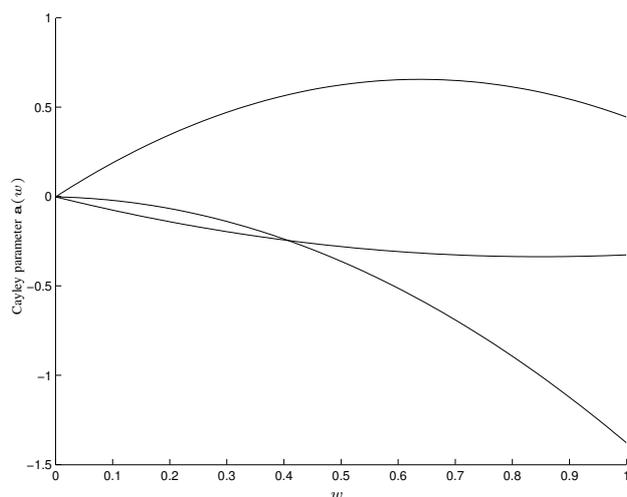
Fig. 1. (Example 1) Case $\delta = 2$: scene and intermediate locations along the computed path (a), image trajectories with current points in red and desired points in green (b), and Cayley parameter $\mathbf{a}(w)$ (c).



(a)



(b)



(c)

Fig. 2. Example 2.

i	$(\mathbf{p}_i)_1$	$(\mathbf{p}_i)_2$	$(\mathbf{p}_i^*)_1$	$(\mathbf{p}_i^*)_2$
1	230.2	229.8	382.9	120.9
2	245.4	256.3	409.4	155.2
3	260.7	282.9	441.5	196.5
4	183.3	192.0	296.2	135.1
5	174.2	234.5	296.2	197.6
6	209.7	256.3	342.4	197.3
7	215.8	215.7	342.4	126.5
8	196.6	224.4	317.9	164.3

TABLE II
IMAGE POINTS FOR EXAMPLE 2.

i	$(\mathbf{p}_i)_1$	$(\mathbf{p}_i)_2$	$(\mathbf{p}_i^*)_1$	$(\mathbf{p}_i^*)_2$
1	337.3	179.8	380.3	125.1
2	285.2	198.3	400.1	157.5
3	236.5	215.6	423.9	196.7
4	214.6	122.0	398.2	233.9
5	108.3	112.4	398.2	300.7
6	106.4	179.6	430.1	320.6
7	197.2	181.5	430.1	240.6
8	158.8	151.7	412.7	273.4

TABLE III
IMAGE POINTS FOR EXAMPLE 3.

C. Example 3

In this example we have considered a more difficult case by increasing the initial displacement in terms of rotation angles, see Figure 3a. The image points are reported in Table III.

With $\delta = 1$ and $\delta = 2$ we could not find any feasible path ($\mu_1 = -\infty$ and $\mu_2 = -0.0082$ respectively). With $\delta = 3$ we have found $\mu_3 = 0.0342$ whose corresponding path is reported in Figure 3.

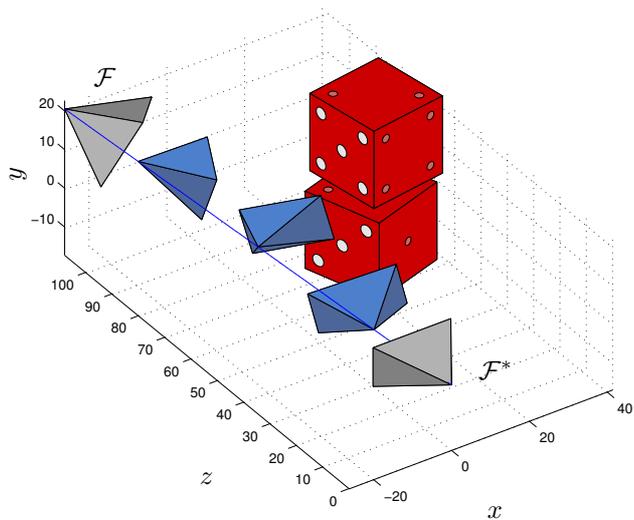
V. CONCLUSION

A problem not investigated yet is to establish if it is possible to reach the desired location keeping all features in the field of view and following a straight line in the euclidean space. The proposed approach deals with such problem by using polynomials to obtain a complete parameterization of the path. In particular, the computation of the camera path that follows a straight line and maximizes the distance of the image trajectories from the screen boundary is considered. The found path can then be followed by using an IBVS controller.

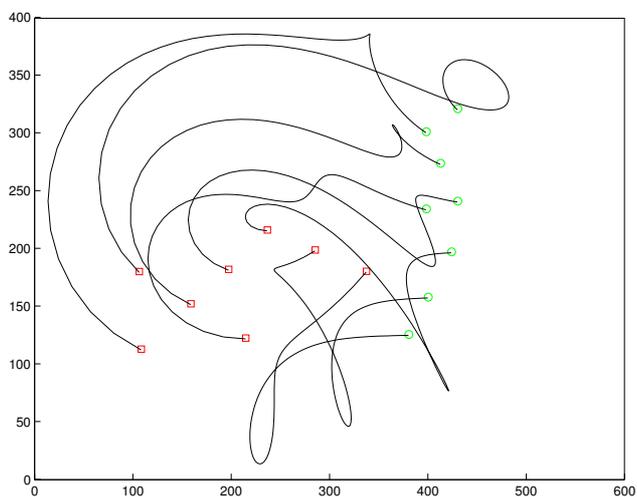
Future work will be devoted to obtain a convex formulation of the problem and will consider other kind of performances.

REFERENCES

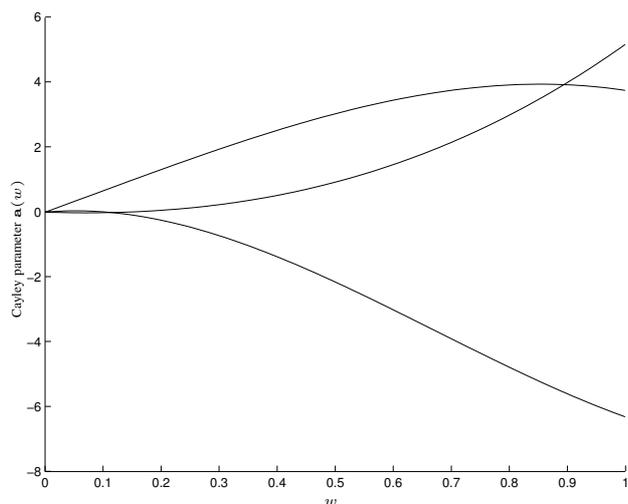
- [1] K. Hashimoto, *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*. Singapore: World Scientific, 1993.
- [2] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.
- [3] K. Hashimoto, "A review on vision-based control of robot manipulators," *Advanced Robotics*, vol. 17, no. 10, pp. 969–991, 2003.



(a)



(b)



(c)

Fig. 3. Example 3.

- [4] C. Taylor and J. Ostrowski, "Robust vision-based pose control," in *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, California, 2000, pp. 2734–2740.
- [5] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice, "Position based visual servoing: keeping the object in the field of vision," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington, D.C., 2002, pp. 1624–1629.
- [6] A. C. Sanderson, L. E. Weiss, and C. P. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE Trans. on Robotics and Automation*, vol. RA-3, pp. 404–417, 1987.
- [7] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1991, pp. 2267–2272.
- [8] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.
- [9] K. Deguchi, "Optimal motion control for image-based visual servoing by decoupling translation and rotation," in *Proc. Int. Conf. on Intelligent Robots and Systems*, 1998, pp. 705–711.
- [10] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 4, pp. 507–515, 2001.
- [11] P. Oh and P. Allen, "Visual servoing by partitioning degrees-of-freedom," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 1, pp. 1–17, 2001.
- [12] N. Cowan, J. Weingarten, and D. Koditschek, "Visual servoing via navigation functions," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 4, pp. 521–533, 2002.
- [13] Y. Mezour and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 4, pp. 534–549, 2002.
- [14] H. Zhang and J. Ostrowski, "Visual motion planning for mobile robots," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 2, pp. 199–208, 2002.
- [15] J. Park and M. Chung, "Path planning with uncalibrated stereo rig for image-based visual servoing under large pose discrepancy," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 2, pp. 250–258, 2003.
- [16] E. Malis, "Visual servoing invariant to changes in camera-intrinsic parameters," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 1, pp. 72–81, 2004.
- [17] O. Tahri and F. Chaumette, "Image moments: generic descriptors for decoupled image-based visual servo," in *Proc. IEEE Int. Conf. on Robotics and Automation*, New Orleans, Louisiana, 2004, pp. 1185–1190.
- [18] G. Chesi and A. Vicino, "A visual servoing technique for large displacements," in *IEEE Int. Conf. on Robotics and Automation 2004*, New Orleans, Louisiana, 2004, pp. 1855–1860.
- [19] E. Malis, "Vision-based control invariant to camera intrinsic parameters: stability analysis and path tracking," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington, D.C., 2002, pp. 217–222.
- [20] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton: CRC Press, 1994.
- [21] O. Faugeras and Q.-T. Luong, *The Geometry of Multiple Images*. Cambridge (Mass.): MIT Press, 2001.
- [22] E. Malis and F. Chaumette, "2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *Int. Journal of Computer Vision*, vol. 37, no. 1, pp. 79–97, 2000.