23rd International Symposium on Mathematical Theory of Networks and Systems
Hong Kong University of Science and Technology, Hong Kong, July 16-20, 2018

Acceleration of convergence while tracking planned feature trajectories for Visual Servoing

Tiantian Shen[1] and Zhiguang Feng[2]

Abstract— **This extended abstract illustrates mainly an idea for accelerating the convergence in eye-in-hand visual servoing applications. Path-planning techniques have been embedded into VS (Visual Servoing) to guide the sensor motion satisfying multiple constraints and limitations. Ahead of visual servoing, path-planning techniques can produce a series of feature trajectories, either projected from a target when the camera virtually moves along the planned camera path or directly planned in the image plane. These feature trajectories are then interpolated and tracked by an IBVS (Imaged Based VS) controller to conduct a converged VS process. In general, larger the interpolation interval or the controller gain, quicker the convergence and, unfortunately, probably larger the tracking error of the planned path and even divergence in the workspace due to image noises. This abstract aims to obtain an optimal interpolation interval under a certain interpolation policy and an associated controller gain under an upper bound of tracking error defined in the workspace to accelerate the global convergence of path-planning based VS applications.**

I. INTRODUCTION

In an eye-in-hand visual servoing system [2], the controller utilizes the desired feature set of a target as a reference to guide the camera towards its desired pose (the required converge point) that corresponds to the desired feature set. This is typically realized by iteratively reducing feature errors obtained from two views of the target. One view comes from a camera at an arbitrarily camera pose, the other is from the same camera at the desired pose. In traditional IBVS (Image-Based Visual Servoing) controller, as given in the following formula, the performance is mainly dependent on the selected feature set and a controller gain.

$$\mathbf{T} = -\lambda_1 \hat{\mathbf{L}}^{+}(\mathbf{s}(t) - \mathbf{s}^*). \tag{1}$$

Here, $\mathbf{T} = [\upsilon_x, \upsilon_y, \upsilon_z, \omega_x, \omega_y, \omega_z]^\top$ describes camera velocities in translation and rotation at time t, which decrease along with the falling trends of $|\mathbf{s}(t) - \mathbf{s}^*|$. $\mathbf{s}(t)$ holds current feature values at time t and $\mathbf{s}^*$ the desired feature values. $\hat{\mathbf{L}}^{+}$ is the pseudo-inverse of the estimated interaction matrix or image jacobian. λ_1 is a positive gain that controls the iteration progress.

Divergence of the servo path from the desired camera pose is possible, or a local minimum [1] of feature error $\mathbf{e} = \mathbf{s}(t) - \mathbf{s}^*$ might occur, especially when camera displacement is large. Global convergence in both feature space and workspace is necessary in successful VS applications and it can be achieved sometimes by appropriately selecting the feature set, like the case proposed for cylinders in [9]. However, guarantee of convergence is not our major concern here. Based on the assumption that global convergence is met, other constraints and limitations like continuous visibility of the selected features, collision avoidance in the workspace and etc. necessitate path-planning techniques as a preface [8], [4]. In optimization-based path-planning techniques [3], [7], a camera path in the Cartesian space is first planned satisfying multiple constraints and limitations. Along this camera path, image projection of the target will form an image trajectory and pertinent trajectories of features used in an IBVS controller. At last, these planned feature trajectories is tracked by interpolating several intermediate values as temporary desired features and then converge to them one by one through an IBVS controller.

We focus on the minimization of global convergence time under a tolerable tracking error defined in the workspace. In general, it depends on the interpolation policy and the controller gain. Larger the interpolation interval or the controller gain, quicker the convergence and, unfortunately, larger the tracking error. This abstract proposes to realize the acceleration of convergence by adjusting the interpolation and controller gain through a bounded minimization problem. The considered target will not only limited to typical feature points, but also include several geometrical primitives.

II. TRACKING PLANNED FEATURE TRAJECTORIES

This section first introduces how to obtain a series of planned feature trajectories, and then two tracking schemes, respectively, for feature points and moment-based features.

In optimization-based path-planning techniques, a virtual VS (VVS) process [10] can be used to estimate the camera displacement based on an approximated target position and model, similar to moving a virtual camera from one pose to the other with instant camera velocities computed as in (1). This VVS method is equivalent to nonlinear methods that consist in minimizing a cost function using iterative algorithms. The estimated displacement between the desired camera pose and the current one will serve as a prerequisite and a boundary condition in the following path-planning techniques.

A camera path is then modeled as seven polynomials on a common path parameter $w \in [0,1]$ with $w = 0$ and $w = 1$, respectively, indicates the beginning and the end of the path. Variables in polynomial coefficients are initialized and optimized to achieve a satisfactory camera path meeting not only boundary condition but also other constraints such as

[1]Tiantian Shen is with School of Information Science and Engineering, Central South University, Changsha 410083, P. R. CHINA tiantianshen@gmail.com

[2]Zhiguang Feng is with the College of Automation, Harbin Engineering University, Harbin 150001, P. R. CHINA congdian@gmail.com

camera field of view (FOV) limit of the target [8], [4]. Along this camera path, image projection of the target will form a planned image trajectory and pertinent trajectories of features used in an IBVS controller. We denote the planned feature trajectories as $\mathbf{s}_p(w)$ with $w \in [0,1]$. Such planned trajectories are interpolated into several segments, at the ends of which feature values in $\mathbf{s}_p(w)$ are tracked by an IBVS controller.

In the sequel, we recall two interpolation strategies for tracking planned feature trajectories that adhere to large camera displacement. Tracking performance will be shown by examples with some moment-based features of geometrical primitives.

A. Typical Feature Points

To chase planned feature trajectories $\mathbf{s}_p(w)$ that consist of pixel coordinates of some feature points, $\mathbf{s}_p(w)$ are interpolated with $w = 1 - e^{-t\lambda_2}$ and therefore generate several discrete feature values. These intermediate feature values are successively brought into the following modified IBVS controller to implement the whole tracking task.

$$\mathbf{T} = -\lambda_1 \hat{\mathbf{L}}^+(\mathbf{s}(t) - \mathbf{s}_p(w)) - \frac{\partial[\hat{\mathbf{L}}^+(\mathbf{s}(t) - \mathbf{s}_p(w))]}{\partial t}. \quad (2)$$

The first addend here is a classical IBVS controller in (1), where intermediate values in $\mathbf{s}_p(w)$ are substituted for the desired feature set. The larger the λ_1, the faster the end-effector/camera moves towards the desired location. The larger the positive gain λ_2, the faster for the value of $w = 1 - e^{-t\lambda_2}$ converges from 0 to 1. The second addend in (2) is a compensation item [4]. If the target is known to be motionless, we have the second addend to be rewritten as:

$$\mathbf{T} = -\lambda_1 \hat{\mathbf{L}}^+(\mathbf{s}(t) - \mathbf{s}_p(w)) + \hat{\mathbf{L}}^+ \frac{\partial \mathbf{s}_p(w)}{\partial t}, \quad (3)$$

with

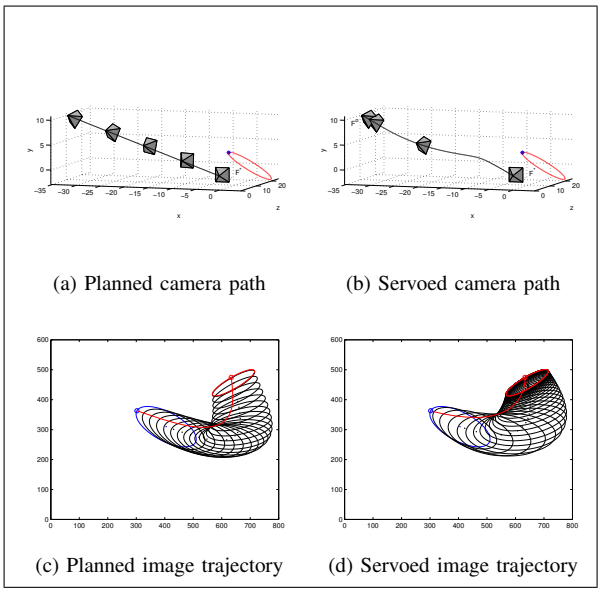
$$\frac{\partial \mathbf{s}_p(w)}{\partial t} = \frac{\partial \mathbf{s}_p(w)}{\partial w} \cdot \frac{dw}{dt} = \lambda_2 e^{-t\lambda_2} \frac{\partial \mathbf{s}_p(w)}{\partial w}. \quad (4)$$

This interpolation strategy and tracking scheme generally perform well for feature points by appropriately selecting values of λ_1 and λ_2 [5]. Optimal values of λ_1 and λ_2 for accelerating convergence is one thing. The other thing is that the development in (4) for moment-based features is tedious when some geometrical primitives are taken as a target, what's worse, the above interpolation strategy and tracking scheme does not apply very well in tracking these moment-based features.

B. Moment-based Features

Moment-based features can be constructed from projection area of some basic geometrical primitives, like spheres, circles, cylinders and etc.

Fig. 1 shows the tracking results when a circle and a point constitute a target. The left hand side displays a planned camera path in a synthetic scene and the associated image trajectory. When feature trajectories extracted from Fig. 1 (c) are followed by the controller in (3), it is obvious that large tracking errors appear in both the workspace and the image space, as shown in the right hand side of Fig. 1.



(a) Planned camera path (b) Servoed camera path

(c) Planned image trajectory (d) Servoed image trajectory

Fig. 1: Path-planning based visual servoing with a circle and a point on the circle [6].

To solve this problem, another interpolation strategy was utilized to track moment-based feature trajectories [7], [9]. Let us recall the modified IBVS controller:

$$\mathbf{T} = -\lambda_1 \hat{\mathbf{L}}^+(\mathbf{s}(t) - \mathbf{s}_i^*), \quad i = 1, \ldots, num\_seg. \quad (5)$$

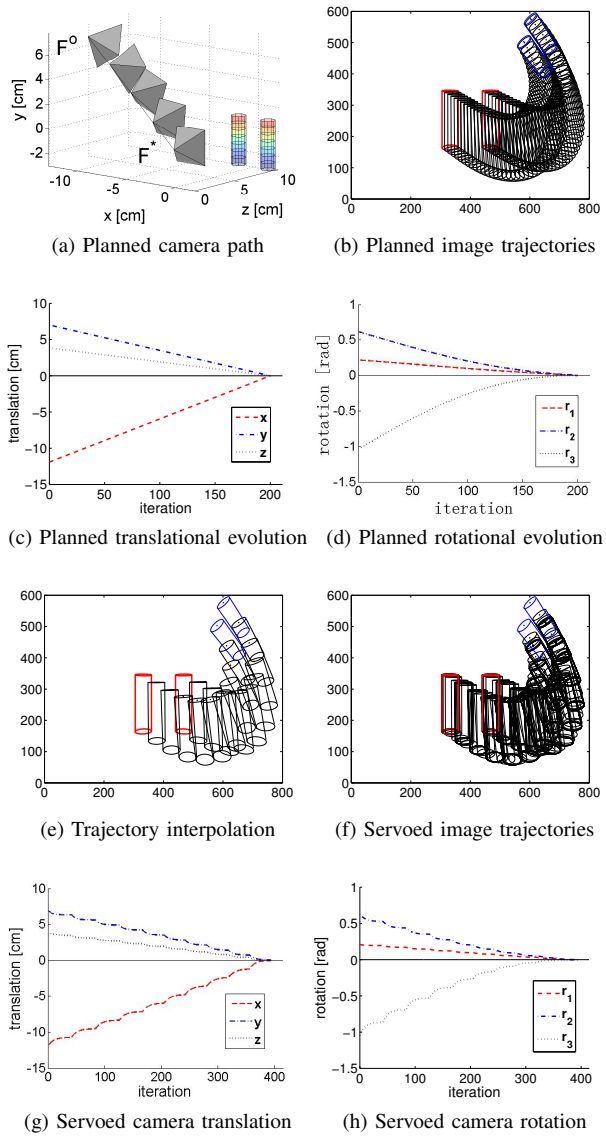
Here, λ_1 is a positive gain as the same as the one used in (1), usually $\lambda_1 = 0.1$. The desired feature values in $\mathbf{s}_i^*$ are values extracted from the planned image trajectory, like the plotted one in Fig. 1 (c). We equally divide the planned camera path into *num_seg* segments, and extract feature values from image projections at the ends of these segments and assign sequentially these values into $\mathbf{s}_i^*$. For every $\mathbf{s}_i^*$, controller (5) guides the camera motion with updated $s(t)$ until the largest element in feature error $\mathbf{s}(t) - \mathbf{s}_i^*$ is bellow a threshold, usually set as 1 pixel. In details, we compute the length of the planned camera path and denoted it as l_{path_length}, select a segment length to be denoted as

$$l_{segment\_length} = \lambda_3 l_{path\_length}, \quad \lambda_3 \in (0,1), \quad (6)$$

and then derive the value of *num_seg* in (5) as the nearest integer to the division of $l_{path\_length}/l_{segment\_length}$.

As a result, the planned camera path is interpolated with several intermediate values and then produce a few segments, the end of which corresponds to the path abscise value computed as $w_i = i/num\_seg$, $i = 1, \ldots, num\_seg$. At different stages of visual servoing, planned feature values at w_i are computed and treated as $\mathbf{s}_i^*$. Overall, the tracking performance is mainly dependent on the number of interpolated segments, that is actually determined by the value of λ_3, and the secondly important controller gain λ_1.

This tracking strategy has been applied to a path-planning based visual servoing application with two parallel cylinders. In this scenario, the tracking performance with $\lambda_3 = 0.1$ and $\lambda_1 = 0.1$ is displayed here in Fig. 2. It can be seen in Fig.



(a) Planned camera path (b) Planned image trajectories

(c) Planned translational evolution (d) Planned rotational evolution

(e) Trajectory interpolation (f) Servoed image trajectories

(g) Servoed camera translation (h) Servoed camera rotation

Fig. 2: Path-planning based visual servoing with two parallel cylinders [9].

2 (e) that there are totally 9 intermediate values and 10 segments to constitute a whole planned image trajectory. The tracking process by applying the visual servo controller in (5) is plotted in Fig. 2 (f)-(h).

Tracking errors during iterative servo process in both camera translational and rotational coordinates can be seen in Fig. 2 (g)-(h), taking the planned evolution in Fig. 2 (c)-(d) as a reference. It is also noticed that the maximum camera displacement is about 11.912 cm and 59.845 degrees around the z-axis. Not quite large a displacement, however, it takes nearly 400 iterations, the number of which is proportional to the whole convergence time. The tracking errors in camera translation are less than at least 0.5 cm estimated from Fig. 2 (g). Under a tolerable tracking error defined in the workspace, we aim at the minimization of the whole convergence time by increasing the interpolation interval and the controller gain.

III. CONCLUSIONS

The abstract, at this stage, describes the background and the motivated problem. It is expected, at the next stage, to solve this problem for generalized geometrical features (including feature points and moment-based features). Examples will be displayed to illustrate the contradiction between acceleration of convergence and a decline in the tracking error of the planned path. After the proposed idea is verified by variform static targets, our next objective is to extend the work to the case of a moving target with various shapes.

REFERENCES

[1] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G. Hager, and A. Stephen Morse, editors, *The confluence of vision and control*, volume 237 of *LNCIS*, pages 66–78. Springer, Heidelberg, 1998.

[2] F. Chaumette, S. Hutchinson, and P. Corke. Visual Servoing. In O. Khatib B. Siciliano, editor, *Handbook of Robotics, 2nd edition*, pages 841–866. Springer, 2016.

[3] G. Chesi and Y. S. Hung. Global path-planning for constrained and optimal visual servoing. *IEEE Trans. Robot.*, 23(5), 2007.

[4] Y. Mezouar and F. Chaumette. Path planning for robust image-based control. *IEEE Trans. on Robotics and Automation.*, 18(4):534–549, 2002.

[5] T. Shen and G. Chesi. Visual servoing path-planning for cameras obeying the unified model. *Advanced Robotics*, 26(8–9):843–860, 2012.

[6] T. Shen and G. Chesi. Following a straight line in visual servoing with elliptical projections. In *13th International Conference on Informatics in Control, Automation and Robotics*, pages 47–56, Lisbon, Portugal, 2016.

[7] T. Shen and G. Chesi. Visual servoing path-planning with elliptical projections. In Kurosh Madani, Dimitri Peaucelle, and Oleg Gusikhin, editors, *Informatics in Control, Automation and Robotics*, pages 30–54. Lecture Notes in Electrical Engineering, vol 430. Springer, Cham, 2018.

[8] T. Shen, S. Radmard, A. Chan, Elizabeth A. Croft, and G. Chesi. Optimized vision-based robot motion planning from multiple demonstrations. *Autonomous Robots*, 41(1):Online first, DOI: 10.1007/s10514–017–9667–4, 2017.

[9] T. Shen, J. Yang, Y. Cai, D. Li, and G. Chesi. Visual servoing with cylinders: Reaching the desired location following a straight line. In *36th Chinese Control Conference*, pages 11183–11188, Da Lian, China, 2017.

[10] O. Tahri, Y. Mezouar, F. Chaumette, and H. Araujo. Visual servoing and pose estimation with cameras obeying the unified model. In G. Chesi and K. Hashimotos, editors, *Visual Servoing via Advanced Numerical Methods*, pages 231–252. LNCIS Series, No 401, Springer-Verlag, 2010.