INTERNATIONAL JOURNAL OF RESEARCH IN MECHANICAL, MECHATRONICS AND AUTOMOBILE ENGINEERING (IJRMMAE)
ISSN: 2454-1435 (Print) | 2454-1443 (online)
Volume 3 Issue 2, July – September 2017 - www.ijrmmae.in – Pages 37-42

ADVANCED APPROACHES IN POSITION BASED VISUAL SERVO MECHANISM

R. Nandhakumar[1], D. Pinky[2], V.Kamalakannan[3]

Department of Mechatronics Engineering, Agni College of Technology, Chennai.

ABSTRACT

This chapter introduces visual servo control, using computer vision data in the servo loop to control the motion of a robot. We first describe the basic techniques that are by now well established in the field. We give a general overview of the formulation of the visual servo control problem, and describe the two archetypal visual servo control schemes: image-based and position-based visual servo control. We then discuss performance and stability issues that pertain to these two schemes, motivating advanced techniques. Of the many advanced techniques that have been developed

POSITION-BASED CONTROL SCHEMES

Position-based control schemes (PBVS) use the pose of the camera with respect to some reference coordinate frame to define $\boldsymbol{s}$. Computing this pose from a set of measurements in one image necessitates the camera intrinsic parameters and the 3-D model of the object observed to be known. This classic computer vision problem is called the 3-D localization problem. While this problem is beyond the scope of the present chapter, many solutions have been presented in the literature and its basic principles are recalled in Chap. 23. It is then typical to define $\boldsymbol{s}$ in terms of the parameterization used to represent the camera pose. Note that the parameters $\boldsymbol{a}$ involved in the definition of $\boldsymbol{s}$ are now the camera intrinsic parameters and the 3-D model of the object. It is convenient to consider three coordinate frames: the current camera frame Fc, the desired camera frame Fc∗ , and a reference frame Fo attached to the object. We adopt here the standard notation of using a leading superscript to denote the frame with respect to which a set of coordinates is defined.

Thus, the coordinate vectors c*t*o and c∗ *t*o give the coordinatesof the origin of the object frame expressed relative to the current camera frame, and relative to the desired camera frame, respectively. Furthermore, let $\boldsymbol{R}$= c∗ $\boldsymbol{R}$c be the rotation matrix that gives the orientation of the current camera frame relative to the desired frame. We can define $\boldsymbol{s}$ to be ($\boldsymbol{t}$, $\theta\boldsymbol{u}$), in which $\boldsymbol{t}$ is a translation vector, and $\theta\boldsymbol{u}$ gives the angle/axis parameterization for the rotation. We now discuss two choices for $\boldsymbol{t}$, and give the corresponding control laws. If $\boldsymbol{t}$ is defined relative to the object frame F0, we obtain $\boldsymbol{s}$ = (c*t*0, $\theta\boldsymbol{u}$), $\boldsymbol{s}$∗ = (c∗ *t*0, 0), and $\boldsymbol{e}$ = (c*t*0− c∗*t*0, $\theta\boldsymbol{u}$). In this case, the interaction matrix related to $\boldsymbol{e}$ is given

$$\boldsymbol{L}_e = \begin{pmatrix} -\boldsymbol{I}_3 & [{}^c\boldsymbol{t}_0]_x \\ \boldsymbol{0} & \boldsymbol{L}_{\theta u} \end{pmatrix}$$

in which ***I***3 is the 3×3 identity matrix and ***Lθu*** is given by

$$L_{\theta u} = I_3 - \frac{\theta}{2}[u]_x + \left(1 - \frac{sinc\theta}{sinc^2}\right)[u]_x^2$$

where sinc *x* is the sinus cardinal defined such that *x* sinc *x* = sin *x* and sinc 0 = 1. Following the development in Sect. 24.1, we obtain the control scheme

$$v_c = -\lambda \widehat{L_e^{-1}} e$$

since the dimension *k* of ***s*** is six, that is, the number of camera degrees of freedom. By setting

$$\widehat{L_e^{-1}} = \begin{pmatrix} -I_3 & [{}^c t_0]_x \, x \, L_{\theta u}^{-1} \\ 0 & L_{\theta u}^{-1} \end{pmatrix}$$

we obtain after simple developments:

$$\begin{cases} v_c = -\lambda[{}^{c^*}t_0 - {}^c t_0] + [{}^c t_0]_x \theta u \\ R_c = -\lambda \theta u \end{cases}$$

since ***Lθu*** is such that ***L***−1 *θ**u** θ**u*** = *θ**u***. Ideally, that is, if the pose parameters are perfectly estimated, the behavior of ***e*** will be the expected one ($\dot{e} = -\lambda e$). The choice of ***e*** causes the rotational motion to follow a geodesic with an exponential decreasing speed and causes the translational parameters involved in ***s*** to decrease with the same speed. This explains the nice exponential decrease of the camera velocity components. Furthermore, the trajectory in the image of the origin of the object frame follows a pure straight line (here the center of the four points has been selected as this rigin). On the other hand, the camera trajectory does not follow a straight line. Another PBVS scheme can be designed by using ***s*** = (*c*∗ ***t***c*,* *θ**u***). In this case, we have ***s***∗ = **0**, ***e*** = ***s***, and

$$L_e = \begin{pmatrix} R & 0 \\ 0 & L_{\theta u} \end{pmatrix}$$

Note the decoupling between translational and rotational motions, which allows us to obtain a simple control scheme

$$\begin{cases} v_c = -\lambda R^T \, {}^{c^*} t_c \\ \omega_c = -\lambda \theta u \end{cases}$$

In this case, if the pose parameters involved are estimated perfectly, the camera trajectory is a pure straight line, while the image trajectories are less satisfactory than before. Some particular configurations can even be found so that some points leave the camera field of view. The stability properties of PBVS seem quite attractive. Since $L_{\theta u}$ is nonsingular when *θ* is not equal to 2*kπ*, we obtain from the global asymptotic stability of the system since $L_e \widehat{L_e^{-1}} = I6$, under the strong hypothesis that all the pose parameters are perfect. This is true for both methods presented above, since the interactions matrices are full rank when $L_{\theta u}$ is nonsingular. With regard to robustness, feedback is computed using *estimated* quantities that are a function of the image measurements and the system calibration parameters. The interaction matrix given in corresponds to perfectly estimated pose parameters, while the real one is unknown since the estimated pose parameters may be biased due to calibration errors, or inaccurate and unstable due to noise. The true positivity condition should in fact be written:

$$\widehat{L_e L_e^{-1}} > 0$$

ADVANCED APPROACHES

Hybrid VS

Suppose we have access to a clever control law for ωc, such as the one used in PBVS:

$$\omega c = -\lambda\ \theta \boldsymbol{u}$$

How could we use this in conjunction with traditional IBVS? Considering a feature vector **s**t and an error **e**t devoted to control the translational degrees of freedom, we can partition the interaction matrix as follows

$$\begin{aligned} \dot{s_t} &= L\, s_t v_s \\ &= (L_v L_\omega)\begin{pmatrix} v_c \\ \omega_c \end{pmatrix} \\ &= L_v v_c + L_w \omega_c \end{aligned}$$

Now, setting ˙**e**t =−λ**e**t, we can solve for the desired translational control input as

$$-\lambda e t = \dot{e} t = \dot{s} t = L v v c + L \omega \omega c\ ,$$

$$\Rightarrow v c = -L^{+} v\ (\lambda e t + L \omega \omega c)\ .$$

We can think of the quantity (λ_{et}+$L_\omega \omega_c$) as a modified error term, one that combines the original error with the error that would be induced by the rotational motion due to ωc. The translational control input **v**c =−**L**+ **v** (λ_{et}+$L_\omega \omega_c$) will drive this error to zero. The method known as 2.5-D visual servo was the first to exploit such a partitioning in combining IBVS and PBVS. More precisely, s_t has been selected as the coordinates of an image point, and the logarithm of its depth, so that **Lv** is a triangular always invertible matrix. More precisely, we have **s**t = (**x**, log Z), **s**∗t= (**x**∗ , log Z∗), **e**t = (**x**−**x**∗ , log ρZ) where $\rho Z = Z/Z*$, and

$$L_v = \frac{1}{Z^* \rho z}\begin{pmatrix} -1 & 0 & x \\ 0 & -1 & y \\ 0 & 0 & -1 \end{pmatrix}$$

$$L_\omega = \begin{pmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \\ -y & x & 0 \end{pmatrix}$$

Note that the ratio ρZ can be obtained directly from the partial pose estimation algorithm If we come back to the usual global representation of visual servo control schemes, we have **e** = (**e**t, θ**u**) and

Le given by

$$L_e = \begin{pmatrix} L_v & L_\omega \\ 0 & L_{\theta u} \end{pmatrix}$$

from which we immediately obtain the control law. The behavior obtained using this choice for ***s***t is shown in Fig. 24.12. Here, the point that has been considered in ***s***t is the center of gravity ***x***g of the target. We note the image trajectory of that point,which is a straight line as expected, and the nice decreasing of the camera velocity components, which makes this scheme very near from the first PBVS one. As for stability, it is clear that this scheme is globally asymptotically stable in perfect conditions. Furthermore, thanks to the triangular form of the interaction matrix ***Le***, it is possible to analyze the stability of this scheme in the presence of calibration errors using the partial pose-estimation algorithm. Finally, the only unknown constant parameter involved in this scheme, that is $Z*$, can be estimated online using adaptive techniques. Other hybrid schemes can be designed, for instance, the third component of ***s***t is different and has been selected so that all the target points remain in the camera field of view as far as possible. Another example has been proposed in. In that case, ***s*** is selected as ***s*** = (c∗ ***t***c, ***x***g, θuz) which provides with a block-triangular interaction matrix of the form:

$$L_e = \begin{pmatrix} R & 0 \\ L'_v & L'_\omega \end{pmatrix}$$

where ***Lv*** and ***Lω*** can easily be computed. This scheme is such that, under perfect conditions, the camera trajectory is a straight line (since c∗ ***t***c is a part of ***s***), and the image trajectory of the center of gravity of the object is also a straight line (since ***x***g is also a part of ***s***). The translational camera degrees of freedom are devoted to realize the 3-D straight line, while the rotational camera degrees of freedom are devoted to realize the 2-D straight line and compensate also the 2-D motion of ***x***g due to the translational motion. This scheme is particularly satisfactory in practice. Finally, it is possible to combine 2-D and 3-D features in different ways. For instance, it has been proposed to use in ***s*** the 2-D homogeneous coordinates of a set of image points expressed in pixels multiplied by their corresponding depth: ***s*** = ($u1Z1$, $v1Z1$, $Z1$, · · · , $un\ Zn$, $vn\ Zn$, Zn). As for classical IBVS, we obtain in this case a set of redundant features, since at least three points have to be used to control the six camera degrees of freedom (here $k \geq 9$). However, it has been demonstrated, that this selection of redundant features is free of attractive local minima.

PARTITIONED VS

The hybrid visual servo schemes described above have been designed to decouple the rotational motions from the translational ones by selecting adequate visual features defined in part in 2-D, and in part in 3-D (which is why they have been called 2.5-D visual servoing). This work has inspired some researchers to find features that exhibit similar decoupling properties but using only features expressed directly in the image. More precisely, the goal is to find six features such that each is related to only one degree of freedom (in which case the interaction matrix is a diagonal matrix), the Grail is to find a diagonal interaction matrix whose elements are constant, as near as possible to the identity matrix, leading to a pure, direct, and simple linear control problem. The first work in this area partitioned the

INTERNATIONAL JOURNAL OF RESEARCH IN MECHANICAL, MECHATRONICS AND AUTOMOBILE ENGINEERING (IJRMMAE)
ISSN: 2454-1435 (Print) | 2454-1443 (online)
Volume 3 Issue 2, July – September 2017 - www.ijrmmae.in – Pages 37-42

interaction matrix to isolate motion related to the optic axis. Indeed, whatever the choice of ***s***, we have

$$\begin{aligned} \dot{s} &= L_s v_c \\ &= (L_{xy} v_{xy}) L_z v_z \\ &= \dot{s_{xy}} + \dot{s_z} \end{aligned}$$

in which ***Lxy*** includes the first, second, fourth, and fifth columns of ***Ls***, and ***Lz*** includes the third and sixth columns of ***Ls*** . Similarly, ***vxy*** = (*vx, vy, ωx, ωy*) and ***vz*** = (*vz, ωz*). Here, ˙***sz*** = ***Lzvz*** gives the component of ˙***s*** due to the camera motion along and rotation about the optic axis, while ˙***sxy*** = ***Lxyvxy*** gives the component of ˙***s*** due to velocity along and rotation about the camera *x* and *y* axes. Proceeding as above, by setting ˙***e***=−*λ****e*** we obtain

$$-\lambda e = \dot{e} = \dot{s} = L_{xy}v_{xy} + L_z v_z\,,$$

which leads to

$$v_{xy} = -L^{+}_{xy}\,[\lambda e(t) + L_z v_z]\,.$$

As before, we can consider [*λ****e***(*t*)+***Lzvz***] as a modified error that incorporates the original error while taking into account the error that will be induced by ***vz*** . Given this result, all that remains is to choose ***s*** and ***vz*** . As for basic IBVS, the coordinates of a collection of image points can be used in ***s***, while two new image features can be defined to determine ***vz***.

- Define *α*, with 0 ≤*α*<2*π*, as the angle between the horizontal axis of the image plane and the directed line segment joining two feature points. It is clear that *α* is closely related to the rotation around the optic axis.
- Define *σ*2 to be the area of the polygon defined by these points. Similarly, *σ*2 is closely related to the translation along the optic axis. Using these features, ***vz*** has been defined as

$$\begin{cases} v_z = \lambda_{v_z} \ln \dfrac{\sigma^*}{\sigma} \\ w_z = \lambda_{w_z} (\alpha^* - \alpha) \end{cases}$$

REFERENCES

[1] B. Espiau, F. Chaumette, P. Rives: A new approach to visual servoing in robotics, IEEE Trans. Robot. Autom. **8**, 313–326 (1992)
[2] S. Hutchinson, G. Hager, P. Corke: A tutorial on visual servo control, IEEE Trans. Robot. Autom. **12**, 651–670 (1996)
[3] L. Weiss, A. Sanderson, C. Neuman: Dynamic sensor-based control of robots with visual feedback, IEEE J. Robot. Autom. **3**, 404–417 (1987)
[4] J. Feddema, O. Mitchell: Vision-guided servoing with feature-based trajectory generation, IEEE Trans. Robot. Autom. **5**, 691–700 (1989)
[5] D. Forsyth, J. Ponce: Computer Vision: A Modern Approach (Prentice Hall, Upper Saddle River 2003)
[6] Y.Ma, S. Soatto, J. Kosecka, S. Sastry: An Invitation to 3-D Vision:From Images to Geometric Models (Springer, New York 2003)
[7] H. Michel, P. Rives: Singularities in the determination of the situation of a robot effector from the erspective view of three points. Tech. Rep. 1850, INRIA Res. Rep. (1993)

[8] M. Fischler, R. Bolles: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM **24**, 381–395 (1981)
[9] E. Malis: Improving vision-based control using efficient second-order minimization techniques, IEEE Int. Conf. Robot. Autom. (New Orleans 2004) pp. 1843–1848
[10] P. Corke, S. Hutchinson: A new partitioned approach to image-based visual servo control, IEEE Trans. Robot. Autom. **17**, 507–515 (2001)
[11] F. Chaumette: Potential problems of stability and convergence in image-based and position-based visual servoing. In: The Confluence of Vision and Control, LNCIS Series, Vol.237, ed. by D. Kriegman, G. Hager, S. Morse (Springer, Heidelberg 1998) pp. 66–78