

Robotic Visual Servoing of Moving Targets

Navid Shahriari, Silvia Fantasia, Fabrizio Flacco, Giuseppe Oriolo

Abstract—We present a new image-based visual servoing scheme for tracking moving targets. This is achieved with a twofold approach. First, we devise a straightforward adaptation of a previously proposed depth observer to account for the fact that the target is not stationary. Second, we estimate the disturbance on the visual feature dynamics due to the target motion, and we add a related compensation term to the visual controller. In particular, the target velocity components parallel to the image plane are reconstructed using a disturbance observer, whereas the orthogonal component is retrieved from the measurement of the Focus Of Expansion. Comparative experiments show that the proposed method can improve over classical visual servoing schemes by 50% or more.

I. INTRODUCTION

Image-Based Visual Servoing (IBVS) is a classical approach to improve the quality of task execution in robotic systems. The main idea is to define the task and accordingly control the robot using visual features, such as points or elementary shapes, defined within the image provided by an eye-in-hand camera.

The strength of the visual servoing approach results from the combination of an intuitive, human-oriented task description with the robustness and accuracy of the control action [1], [2]. In the classical IBVS scheme, however, the target object used to define the visual features must be stationary to guarantee convergence; moreover, the depth associated to these features must be known for computing the so-called interaction matrix. If these two conditions are violated, the performance of the classical scheme may deteriorate [3]. In particular, motion of the target will generate a tracking error, whereas imprecise knowledge of the feature depth reduces the convergence region. Instability may occur if either of these non-idealities is particularly severe.

A possible approach to estimate the depth of a feature is to increase the sensing capabilities of the robot by using stereo vision [4] or time-of-flight sensors [5]. These solutions may be inappropriate in service or industrial applications of visual servoing; for example, stereo cameras have a reduced field of view and usually need more mounting space with respect to monocular cameras, whereas time-of-flight sensors are sensitive to light conditions and surface properties. Other depth estimation methods, such as [6], [7], make use of highly structured visual features at the price of a reduced flexibility of the visual servoing system.

A conceptually different approach to depth estimation of fixed targets was proposed in [8] based on a nonlinear observer. The main advantage of this technique, which is that

neither special cameras nor structured features are required. In particular, the method works with monocular cameras and simple point features. Other subsequent works followed the same approach, including, e.g., [9]–[11].

As for the target motion, object tracking methods existing in the literature are based on Kalman filtering [12]–[15] or dedicated algorithms [16]. Another approach is to treat the motion of the target as a disturbance for the visual servoing task, see, e.g., [17]. Finally, a predictive controller is used in [18].

The objective of this paper is to bring together the two above issues — depth estimation and target motion — so as to improve the performance of classical IBVS schemes in the presence of a moving target. This is achieved with a twofold approach. First, we devise a straightforward adaptation of the depth observer of [8] to account for the fact that the target is not fixed. Second, we estimate the disturbance on the visual feature dynamics due to the target motion, and we add a related compensation term to the visual controller. In particular, the target velocity components parallel to the image plane are reconstructed using a disturbance observer, whereas the orthogonal component is retrieved from the measurement of the Focus Of Expansion.

The paper is organized as follows. First, the general structure of the proposed IBVS for the case of moving targets is discussed in Section II. Section III presents the adaptation to this case of the the depth observer presented in [8] for fixed visual targets. A two-phase technique is then described in Section IV for estimating the target velocity using a disturbance observer in combination with an optical flow estimator. Experimental results with a 6-DOF industrial manipulator carrying a monocular camera are presented in Section VI. Some future work is briefly mentioned in the concluding section.

II. VISUAL SERVOING WITH MOVING TARGET

Basic visual servoing (VS) schemes, either Position-Based (PBVS) or Image-Based (IBVS), assume that the target is stationary [1], [2]. If the target is moving, it is possible to modify such schemes so as to guarantee the same stability properties provided that its velocity is known. In particular, in this paper we will consider IBVS control of a single point feature associated to a 3D point P (extension to multiple point features is straightforward).

Denote by $\mathbf{f} = (f_u \ f_v)^T$ the point feature in the image plane, with coordinates expressed in centered pixel coordinates with respect to the camera principal point. If point P is stationary, the rate of variation of \mathbf{f} depends on the camera linear and angular velocities $\mathbf{v}_c, \boldsymbol{\omega}_c$ expressed in the camera

The authors are with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Via Ariosto 25, 00185 Rome, Italy. E-mail: {flacco,oriolo}@dis.uniroma1.it.

frame via the following relationship (see [19])

$$\dot{\mathbf{f}} = \mathbf{J}_f(\mathbf{f}, Z) \begin{pmatrix} \mathbf{v}_c \\ \boldsymbol{\omega}_c \end{pmatrix}, \quad (1)$$

where the interaction matrix \mathbf{J}_f depends on the depth Z of the point feature

$$\mathbf{J}_f(\mathbf{f}, Z) = \begin{pmatrix} -\frac{\lambda}{Z} & 0 & \frac{f_u}{Z} & \frac{f_u f_v}{\lambda} & -\left(\lambda + \frac{f_u^2}{\lambda}\right) & f_v \\ 0 & -\frac{\lambda}{Z} & \frac{f_v}{Z} & \lambda + \frac{f_v^2}{\lambda} & -\frac{f_u f_v}{\lambda} & -f_u \end{pmatrix} \quad (2)$$

and λ is the focal length of the calibrated camera.

Now assume that point P is moving with a certain world velocity \mathbf{v}_P , also expressed in the camera frame (note that P as a point does not have angular velocity). Then eq. (1) is modified as follows

$$\dot{\mathbf{f}} = \mathbf{J}_f(\mathbf{f}, Z) \begin{pmatrix} \mathbf{v}_c - \mathbf{v}_P \\ \boldsymbol{\omega}_c \end{pmatrix}. \quad (3)$$

Consider the following control law

$$\begin{pmatrix} \mathbf{v}_c \\ \boldsymbol{\omega}_c \end{pmatrix} = \mathbf{J}_f^\dagger(\mathbf{f}, Z) \mathbf{K} \mathbf{e} + \begin{pmatrix} \mathbf{v}_P \\ \mathbf{0} \end{pmatrix}, \quad (4)$$

where \mathbf{J}_f^\dagger is the pseudoinverse of \mathbf{J}_f , \mathbf{K} is a positive definite gain matrix, and $\mathbf{e} = \mathbf{f}_d - \mathbf{f}$ is the feature error w.r.t. the desired set-point \mathbf{f}_d (typically, one sets $\mathbf{f}_d = \mathbf{0}$ to drive the feature to the center of the image plane). By substitution of (4) in (3) it may be easily verified that the closed-loop error dynamics becomes $\dot{\mathbf{e}} = -\mathbf{K} \mathbf{e}$, i.e., global asymptotic stability is obtained.

The interpretation of (4) is obvious: since \mathbf{v}_P acts as an input disturbance on the feature dynamics, the controller simply compensates this disturbance. In general, however, the nominal control law (4) cannot be computed for two reasons:

- 1) The depth Z is unknown in a pure IBVS setting and therefore the pseudoinverse of the interaction matrix (2) cannot be computed.
- 2) The velocity of \mathbf{v}_P to be compensated is unknown.

The first is a well-known issue already in the case of stationary targets, and is usually solved by using a constant depth value \bar{Z} in place of the true Z ; global stability is lost but error convergence is still guaranteed provided that \bar{Z} is sufficiently close to the value of Z at the set-point. As shown in [8], a better solution is to use a time-varying estimate \hat{Z} produced by using a nonlinear observer, because this significantly increases the basin of attraction of the VS controller. We argue that this approach may be even more beneficial in the case of moving targets, where the feature depth can be expected to undergo larger variations.

The second problem can be addressed by using an estimate $\hat{\mathbf{v}}_P$ of the velocity of P . To preserve the pure visual nature of the control scheme, we would like to compute this estimate using image data only.

Wrapping up the above discussion, the implementable control law will be

$$\begin{pmatrix} \mathbf{v}_c \\ \boldsymbol{\omega}_c \end{pmatrix} = \mathbf{J}_f^\dagger(\mathbf{f}, \hat{Z}) \mathbf{K} \mathbf{e} + \begin{pmatrix} \hat{\mathbf{v}}_P \\ \mathbf{0} \end{pmatrix}, \quad (5)$$

with \hat{Z} produced by the depth observer and $\hat{\mathbf{v}}_P$ by the target velocity estimator.

The next section briefly describes the structure of the depth observer, including a straightforward modification to take into account the motion of the target. The procedure for estimating the target velocity is at the core of this paper and will be discussed at length in Section IV.

III. DEPTH OBSERVER WITH MOVING TARGET

Define the state vector $\mathbf{x} = (x_1 \ x_2 \ x_3)^T = (f_u \ f_v \ \frac{1}{Z})^T$ and the input vector $\mathbf{u} = (\mathbf{v}_c - \mathbf{v}_P \ \boldsymbol{\omega}_c)^T$. Following [8] the dynamical system that produces the point features f_u, f_v can be written as

$$\dot{\mathbf{x}} = \begin{pmatrix} -\lambda x_3 & 0 & x_1 x_3 & \frac{x_1 x_2}{\lambda} & -\left(\lambda + \frac{x_1^2}{\lambda}\right) & x_2 \\ 0 & -\lambda x_3 & x_2 x_3 & \lambda + \frac{x_2^2}{\lambda} & -\frac{x_1 x_2}{\lambda} & -x_1 \\ 0 & 0 & x_3^2 & \frac{x_2 x_3}{\lambda} & -\frac{x_1 x_3}{\lambda} & 0 \end{pmatrix} \mathbf{u} \quad (6)$$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \quad (7)$$

Consider the following observer of (6)

$$\dot{\hat{\mathbf{x}}} = \boldsymbol{\alpha}(\hat{\mathbf{x}}, \mathbf{y}) \mathbf{u} + \boldsymbol{\beta}(\hat{\mathbf{x}}, \mathbf{y}, \mathbf{u}), \quad (8)$$

where

$$\boldsymbol{\alpha}(\hat{\mathbf{x}}, \mathbf{y}) = \begin{pmatrix} -\lambda \hat{x}_3 & 0 & y_1 \hat{x}_3 & \frac{y_1 y_2}{\lambda} & -\left(\lambda + \frac{y_1^2}{\lambda}\right) & y_2 \\ 0 & -\lambda \hat{x}_3 & y_2 \hat{x}_3 & \lambda + \frac{y_2^2}{\lambda} & -\frac{y_1 y_2}{\lambda} & -y_1 \\ 0 & 0 & \hat{x}_3^2 & \frac{y_2 \hat{x}_3}{\lambda} & -\frac{y_1 \hat{x}_3}{\lambda} & 0 \end{pmatrix}$$

and

$$\boldsymbol{\beta}(\hat{\mathbf{x}}, \mathbf{y}, \mathbf{u}) = \begin{pmatrix} k_1 e_1 \\ k_2 e_2 \\ k_3 ((-\lambda u_1 + y_1 u_3) e_1 + (-\lambda u_2 + y_2 u_3) e_2) \end{pmatrix}$$

where $k_1, k_2, k_3 > 0$ and e_1, e_2 are the first two components of the observation error $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$, which can be directly measured.

The resulting error dynamics is

$$\begin{aligned} \dot{e}_1 &= -k_1 e_1 + (\lambda u_1 + y_1 u_3) e_3 \\ \dot{e}_2 &= -k_2 e_2 + (\lambda u_2 + y_2 u_3) e_3 \\ \dot{e}_3 &= -k_3 (-\lambda u_1 + y_1 u_3 - \lambda u_2 + y_2 u_3) \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} \\ &\quad + (\hat{x}_3^2 - \hat{x}_3^2) u_3 + \frac{y_2 u_4 - y_1 u_5}{\lambda} e_3 \end{aligned}$$

Under a *persistence of excitation* condition [8] on the camera velocity inputs, it is possible to prove local asymptotic stability for the observation error, i.e., convergence of \hat{Z} to Z provided that the observer initialization is sufficiently close to the true value. Asymptotic stability becomes global if u_3, u_4 and u_5 are guaranteed to be small enough; in a visual servoing loop, this can be achieved by suitable scaling of the control inputs.

Note that the observer (8) needs in principle the velocity \mathbf{v}_P of point P to reconstruct the input vector \mathbf{u} . In practice, we will use $\hat{\mathbf{v}}_P$ produced by the target velocity estimator in place of \mathbf{v}_P .

IV. TARGET VELOCITY ESTIMATION

We now describe a two-stage method for estimating the velocity of the target point P . In the following, we assume that the depth Z of P is known. This is needed to overcome the scale ambiguity present in any perspective system, i.e., the impossibility of distinguishing an object from the same object twice as large, twice as far and moving twice as fast. In practice, our target velocity estimator will use the \hat{Z} produced by the depth observer.

A. Disturbance observer

One way to estimate \mathbf{v}_P from visual data is to build a disturbance observer. To this end, consider the following disturbance-free copy of the feature dynamics (3)

$$\dot{\mathbf{w}} = \mathbf{J}_f(\mathbf{f}, Z) \begin{pmatrix} \mathbf{v}_c \\ \boldsymbol{\omega}_c \end{pmatrix} + \mathbf{H}(\mathbf{f} - \mathbf{w}), \quad (9)$$

with an additional forcing signal which depends on the mismatch between the actual features \mathbf{f} and their virtual counterparts \mathbf{w} through matrix $\mathbf{H} = h \cdot \mathbf{I}$, with $h > 0$. The dynamics of the error $\mathbf{r} = \mathbf{f} - \mathbf{w}$ is readily computed as

$$\dot{\mathbf{r}} = \mathbf{J}_f(\mathbf{f}, Z) \begin{pmatrix} -\mathbf{v}_P \\ \mathbf{0} \end{pmatrix} - \mathbf{H}\mathbf{r}$$

which, in the Laplace domain, becomes

$$\begin{aligned} \mathbf{r} &= (s\mathbf{I} + \mathbf{H})^{-1} \mathbf{J}_f(\mathbf{f}, Z) \begin{pmatrix} -\mathbf{v}_P \\ \mathbf{0} \end{pmatrix} \\ &= -(s\mathbf{I} + \mathbf{H})^{-1} \mathbf{J}_{f,v}(\mathbf{f}, Z) \mathbf{v}_P, \end{aligned}$$

where $\mathbf{J}_{f,v}$ denotes the submatrix obtained taking the first three columns of the interaction matrix \mathbf{J}_f . Thus, vector $\mathbf{H}\mathbf{r}$ is a low-pass filtered version of $-\mathbf{J}_{f,v}(\mathbf{f}, Z) \mathbf{v}_P$ with time constant h :

$$\mathbf{H}\mathbf{r} = -\mathbf{H}(s\mathbf{I} + \mathbf{H})^{-1} \mathbf{J}_{f,v}(\mathbf{f}, Z) \mathbf{v}_P$$

and note that $\mathbf{H}(s\mathbf{I} + \mathbf{H})^{-1} = \text{diag}\{h/(s+h), h/(s+h)\}$. One can then obtain an estimation $\hat{\mathbf{v}}_P$ of \mathbf{v}_P as

$$\hat{\mathbf{v}}_P = \mathbf{J}_{f,v}^\dagger(\mathbf{f}, Z) \mathbf{H}\mathbf{r}. \quad (10)$$

Simulations and experiments have shown this method reconstructs with reasonable accuracy the velocity of P in the x and y directions of the camera frame. However, the estimation of velocities along in the z direction is poor, essentially due to the lesser influence of such motions on the visual feature. For this reason, we will use a different approach for estimating the z component of \mathbf{v}_P .

B. Focus of expansion

When the relative motion between the camera and the target object is purely translational, it is possible to define the Focus of Expansion (FOE) as the point of the image plane where all feature velocity vectors associated to the target object tend to intersect (Figure 1). Its coordinates are

$$\mathbf{f}_{\text{FOE}} = \left(\lambda \frac{v_{R,x}}{v_{R,z}}, \lambda \frac{v_{R,y}}{v_{R,z}} \right), \quad (11)$$

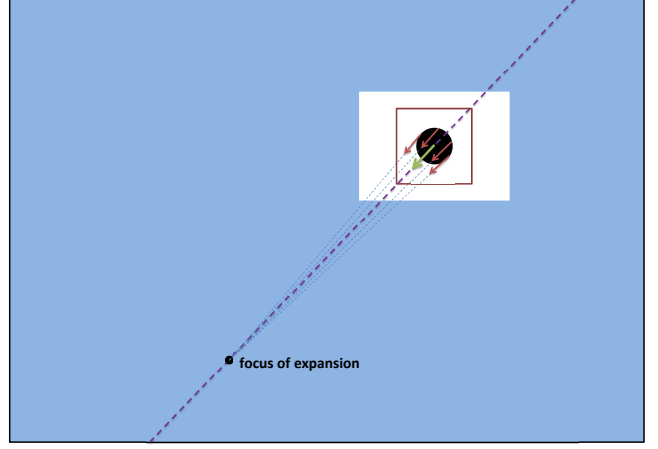


Fig. 1. The Focus of Expansion. The target object is in white, while the region of interest around the target point is the red square. Vectors in red belong to the optical flow, while the green vector is the velocity of the target point. The FOE is always located on the purple dashed line.

where $\mathbf{v}_R = (v_{R,x}, v_{R,y}, v_{R,z}) = \mathbf{v}_P - \mathbf{v}_c$ is the relative velocity between the camera and the target. Once the FOE associated to our target is measured from the image, it is possible to reconstruct an estimate $\hat{v}_{R,z}$ of $v_{R,z}$ using (11). Note that in this computation we use the estimates $\hat{v}_{P,x}$ and $\hat{v}_{P,y}$ produced by the disturbance observer, as well as the known camera velocity. From $\hat{v}_{R,z}$, one directly obtains $\hat{v}_{P,z}$.

A few remarks are in order concerning the computation of the FOE associated to the target motion. The first is that the velocity vectors of several feature points associated to the target object must be available. To this end, an appropriate region of interest centered at the target point is considered (see Figure 1) for computing the optical flow at each sampling instant.

Another issue is that — as already mentioned — the FOE is only defined for pure translational motions between the camera and the target object. Since the camera angular motion is commanded and thus exactly known, its undesired effect on the FOE computation can be removed using a *de-rotation* procedure [20]. As for the target object, however, we must assume pure translational motion at this stage of our research.

In our implementation, the FOE is obtained with the method of [21], which is based on optical flow and uses a matched filter. The optical flow is computed with Lucas-Kanade's method, see, e.g., [22]. The matched filter finds the FOE by searching for the pixel that minimizes a cost function related to optical flow. The time needed to actually come up with the optimal solution obviously depends on the size of the region of interest. In any case, the computation time may be reduced using a two-level search: a rough estimate of the FOE is first found using a lower resolution, and then refined increasing the resolution in a limited area around the estimate.

In our case, a further reduction of the time needed to compute the FOE can be obtained by searching only on the line passing through the target point feature and having the

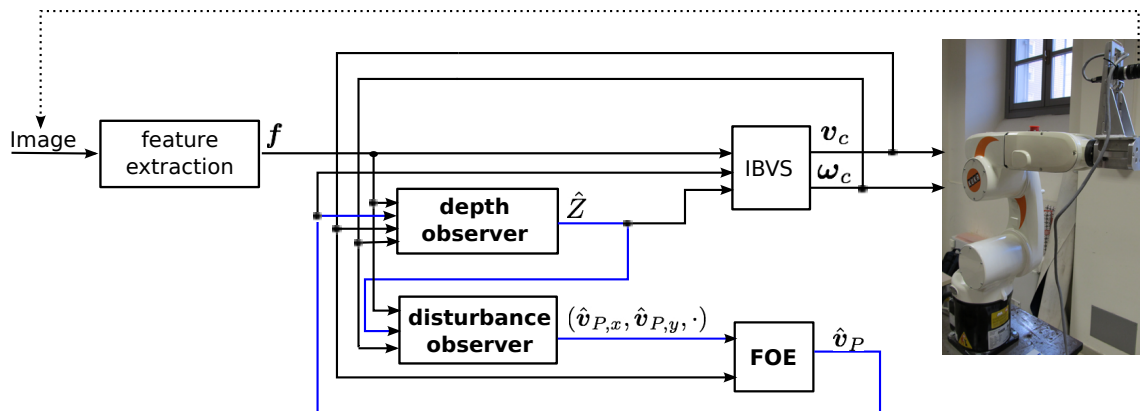


Fig. 2. Block diagram of complete visual servoing system.

same direction as its velocity (see Figure 1).

V. THE COMPLETE SYSTEM

Figure 2 shows the block diagram of the complete system. The depth observer is in charge of estimating the depth of the target point, whereas the disturbance observer and the FOE are used to reconstruct the target velocity. This information is then fed to the IBVS control law, which generates the motion of the eye-in-hand camera.

Note the presence of an algebraic loop between the depth and the disturbance observer. Namely, the depth observer needs to know the target velocity to account for it, while the disturbance observer uses the depth estimate to compute the interaction matrix. As a consequence, we can expect convergence of both observers only if the target depth is initialized at a value close to the actual one, or if the target velocity is small enough.

VI. EXPERIMENTAL RESULTS

To validate the effectiveness of the proposed approach, we performed experiments on a setup consisting of a 6-DOF industrial manipulator, (KUKA KR5 sixx R650), and a monocular CCD Camera (Point Grey Flea 2), illustrated in Figure 3. The target is attached to the end-effector of another robot (KUKA LWR IV), so that its motion is accurately known as a ground truth.

The visual servoing task is to track a point feature represented by the center of a black dot. Feature extraction is implemented using the Visual Servoing Platform (ViSP, see [23]), a library which allows fast development of visual tracking and servoing applications. The optical flow, needed to compute the FOE, is computed via the Open Source Computer Vision library (OpenCV, see [24]), which provides many real-time computer vision routines. Figure 4 shows a sample of the detected target and the associated optical flow.

To allow a comparative assessment, we report results of two experiments in which the target motion was the same; in particular, the target trajectory in Cartesian space is shown in Figure 5. The first experiment was run under the classical IBVS scheme which uses a constant depth for the target, while in the second experiment the proposed control law (5)

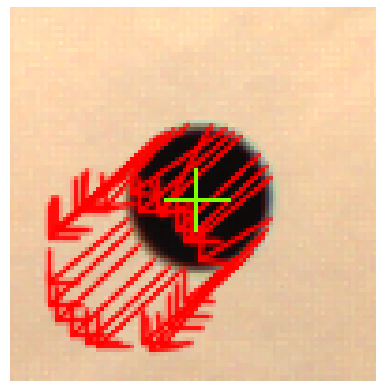


Fig. 4. A sample camera image with the target centroid (green cross) and associated optical flow (red arrows).

was applied. The initial depth of the target was $Z_0 = 1.226$ m. Both servoing schemes were correctly initialized with this depth and used a gain matrix $\mathbf{K} = \text{diag}\{1, 1\}$.

As shown by Figure 6, the norm of the visual error in the first experiment has a maximum of 94 pixels and a mean value of 54.74 pixel. In the second experiment, the maximum error is around 49 pixels and the mean value is 28.70. Overall, the improvement of the proposed IBVS controller over the classical scheme is close to 50%.

The significant improvement with the proposed visual servoing scheme is obviously due to the combined action of the depth and disturbance observers. Figures 7 and 8 show, respectively, the actual vs. estimated depth and the actual vs. estimated target velocity. The plots confirm the effectiveness of the designed observers.

Movie clips of both experiments are contained in the video attachment to the paper.

VII. CONCLUSIONS

In this paper, a new visual servoing scheme has been proposed for tracking moving point targets. A non-linear depth observer provides an estimate the depth of the point target, which is then used in the computation of the interaction matrix. In itself, this enlarges the region of convergence of the visual controller. In addition, the velocity of the target

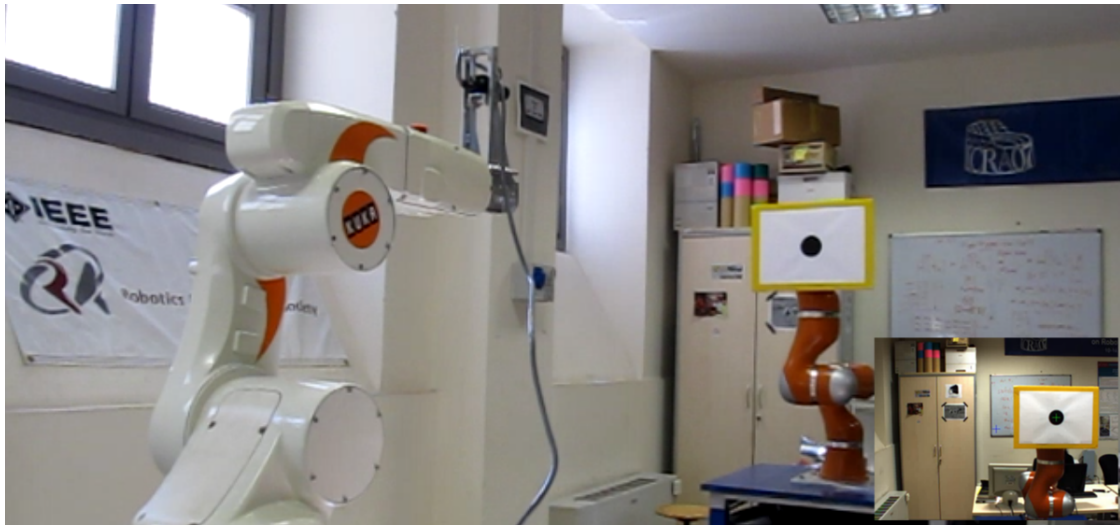


Fig. 3. Our experimental setup. The KUKA KR5 carrying the Point Grey Flea 2 camera is on the left. The visual target attached to the KUKA LWR is on the right. The camera image is shown in the bottom-right corner.

point is reconstructed with a two-phase technique, which makes use of a disturbance observer in combination with a FOE-based estimator. Such velocity is then used to compute a disturbance compensation term to be included in the VS controller. Comparative experiments have shown that the proposed method can improve over classical visual servoing schemes by 50% or more.

Future work will focus on reducing the limitation of the proposed system; in particular, we would like to allow rotational motions of the target object. We are also working towards an analytic proof of the overall stability of the proposed method.

REFERENCES

- [1] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.
- [2] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.
- [3] F. Chaumette and S. Hutchinson, "Visual servo control. II. Advanced approaches [tutorial]," *IEEE Robotics Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.
- [4] C. Shan-shan, Z. Wu-heng, and F. Zhi-lin, "Depth estimation via stereo vision using birchfield's algorithm," in *IEEE International Conference on Communication Software and Networks*, pp. 403–407, 2011.
- [5] M. Marshall, M. Matthews, A.-P. Hu, G. McMurray, and H. Lipkin, "Uncalibrated visual servoing for intuitive human guidance of robots," in *IEEE International Conference on Robotics and Automation*, pp. 4463–4468, 2012.
- [6] A. Torralba and A. Oliva, "Depth estimation from image structure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1226–1238, 2002.
- [7] M. Wahab, N. Sivadev, and K. Sundaraj, "Development of monocular vision system for depth estimation in mobile robot soccer," in *IEEE Conference on Sustainable Utilization and Development in Engineering and Technology*, pp. 36–41, 2011.
- [8] A. De Luca, G. Oriolo, and P. Robuffo Giordano, "Feature depth observation for image-based visual servoing: Theory and experiments," *International Journal of Robotic Research*, vol. 27, no. 10, pp. 1093–1116, 2008.
- [9] F. Morbidi, G. Mariottini, and D. Prattichizzo, "Observer design via immersion and invariance for vision-based leader-follower formation control," *Automatica*, vol. 46, no. 1, pp. 148–154, 2010.
- [10] M. Sassano, D. Carnevale, and A. Astolfi, "Observer design for range and orientation identification," *Automatica*, vol. 46, no. 8, pp. 1369–1375, 2010.
- [11] A. Dani, N. Fischer, and W. Dixon, "Single camera structure and motion," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 238–243.
- [12] P. Corke and M. Goods, "Controller design for high performance visual servoing," in *Proc. 12th World Congress IFAC*, pp. 395–398, 1993.
- [13] F. Bensalah and F. Chaumette, "Compensation of abrupt motion changes in target tracking by visual servoing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 181–187, 1995.
- [14] F. Miyazaki, M. Takeuchi, M. Matsushima, T. Kusano, and T. Hashimoto, "Realization of the table tennis task based on virtual targets," in *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3844–3849, 2002.
- [15] H. Nakai, Y. Taniguchi, M. Uenohara, T. Yoshimi, H. Ogawa, F. Ozaki, J. Oaki, H. Sato, Y. Asari, K. Maeda, H. Banba, T. Okada, K. Tatsuno, E. Tanaka, O. Yamaguchi, and M. Tachimori, "A volleyball playing robot," in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1083–1089, 1998.
- [16] P. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 2, pp. 152–165, 1993.
- [17] N. Papanikolopoulos, P. Khosla, and T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, pp. 14–35, 1993.
- [18] J. Gangloff and M. de Mathelin, "Visual servoing of a 6-dof manipulator for unknown 3-d profile following," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 511–520, 2002.
- [19] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing, Springer, 2011.
- [20] R. Nelson and J. Aloimonos, "Finding motion parameters from spherical motion fields (or the advantages of having eyes in the back of your head)," *Biological Cybernetics*, vol. 58, no. 1, pp. 261–273, 1988.
- [21] D. Sazbon, H. Rotstein, and E. Rivlin, "Finding the focus of expansion and estimating range using optical flow images and a matched filter," *Machine Vision Applications*, vol. 15, no. 4, pp. 229–236, 2004.
- [22] Y. Wu, "Optical flow and motion analysis," *Advanced Computer Vision Notes Series 6*.
- [23] E. Marchand, "Visp: a software environment for eye-in-hand visual servoing," in *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3224–3229, 1999.
- [24] "Opencv reference manual, <http://opencv.willowgarage.com/>,"

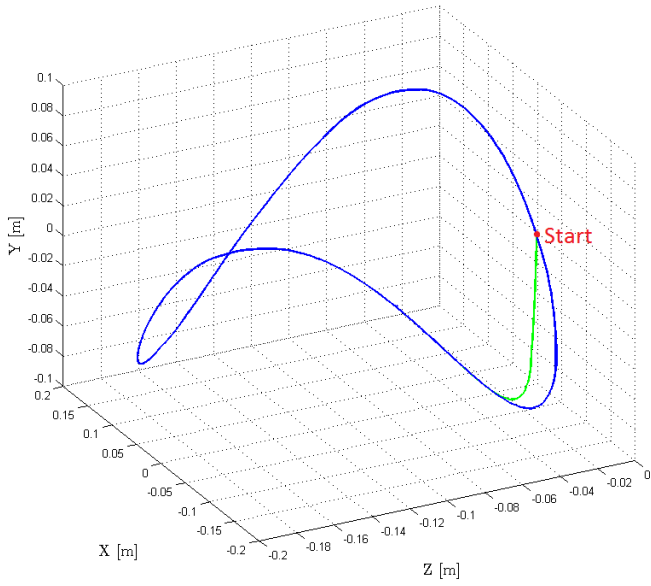


Fig. 5. The target motion in the experiments (red) with a short initial transient (green).

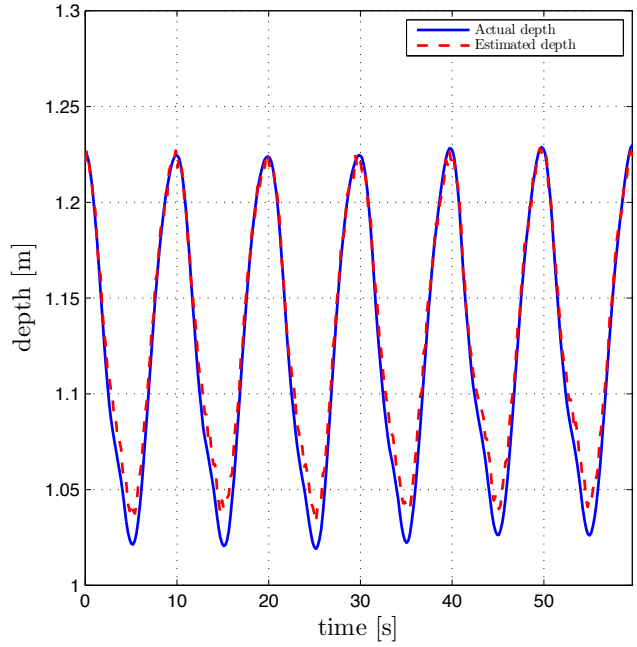


Fig. 7. Experiment 2: Performance of the depth observer. Actual target depth (solid-blue) vs. its estimate (dashed-red).

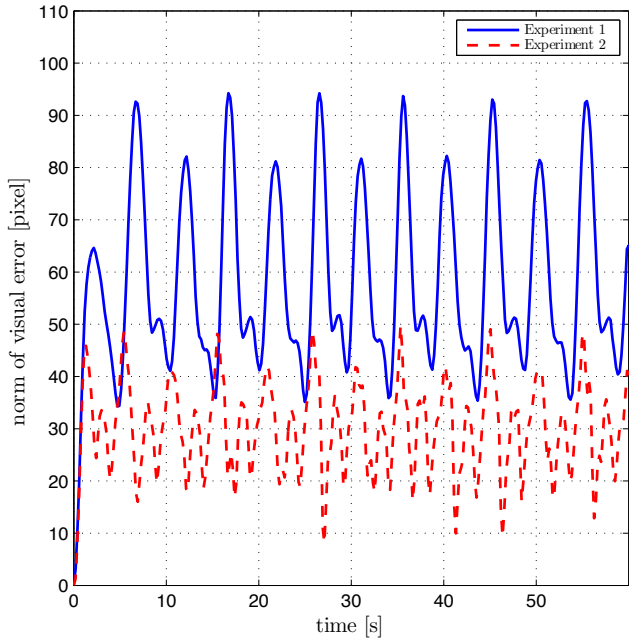


Fig. 6. Comparative results: Norm of visual error using classical IBVS (experiment 1) and the proposed method (experiment 2).

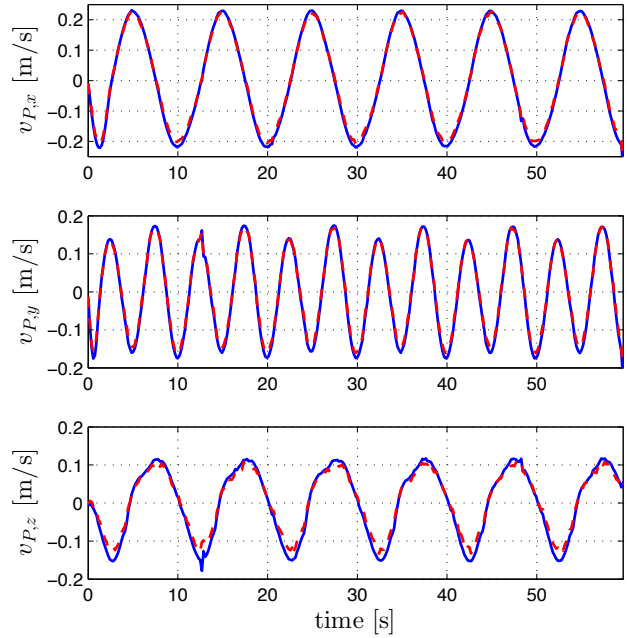


Fig. 8. Experiment 2: Performance of the disturbance observer. Actual target velocity in the camera frame (solid-blue) vs. its estimate (dashed-red).