

Keeping Features in the Field of View in Eye-In-Hand Visual Servoing: A Switching Approach

Graziano Chesi, Koichi Hashimoto, Domenico Prattichizzo, and Antonio Vicino

Abstract—A visual servoing strategy for keeping features in the field of view is proposed which consists of a switching among position-based control strategies and backward motion. In the absence of uncertainty on the extrinsic parameters, all features are kept in the field of view. Moreover, if the intrinsic parameters are also known, the trajectory length is minimized in the rotational space and, for some cases, also minimized in the translational space. Simulation results also show a certain degree of robustness against uncertainty on the intrinsic parameters.

Index Terms—Field of view, point correspondences, switching control, visual servoing.

I. INTRODUCTION

In eye-in-hand visual servoing systems, a control law is computed on the basis of the view of an object acquired by a vision system, usually one camera, mounted on a robot end effector (see, e.g., [1] and [2]). As a consequence, a fundamental requirement on visual servoing algorithms is to keep the observed object in the field of view (FOV) during the robot control. However, this requirement is difficult to satisfy if a geometrical model of the object or the camera parameters are unknown.

In position-based visual servoing (PBVS) (see, e.g., [3] and [4]), it is possible that points leave the FOV since the camera is controlled in the three-dimensional (3-D) space and no control is performed in the image domain. Although in image-based visual servoing (IBVS) (see, e.g., [5]–[7]) the camera is controlled in the image domain, this problem can still be present and, moreover, local minima are present (see [8]).

Recently, several approaches have been developed to deal with the image constraint. In [9], a hybrid method based on the use of PBVS to control the rotation and IBVS to control the position of a reference point is proposed. In [10], a hybrid method which controls the position of an ellipse in the image including all points is presented. In [11] and [12], partitioning methods are used to steer the camera through decoupled control laws. In [13], navigation functions are used based on the knowledge of a geometrical model of the object. Approaches based on the use of potential fields for repelling points from the image boundary have been also proposed, as in the partitioning method [14] and as in the path-planning technique [15]. In [16], a motion planning approach

Manuscript received June 23, 2003; revised November 4, 2003. This paper was recommended by Associate Editor F. Chaumette and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported in part by the Japan Society for the Promotion of Science (JSPS) under Grant-in-Aid for Scientific Research 14205034, and in part by the Italian Space Agency (ASI) under Grant I/R/124/02 "TEMA: Team-based Exploration by Mobile Agents." This paper was presented in part at the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, R.O.C., 2003.

G. Chesi was with the Department of Information Physics and Computing, University of Tokyo, Tokyo 113-8656, Japan. He is now with the Department of Information Engineering, University of Siena, 53100 Siena, Italy (e-mail: chesi@dii.unisi.it).

K. Hashimoto was with PREST JST and the Department of Information Physics and Computing, University of Tokyo, Tokyo 113-8656, Japan. He is now with PREST JST and the Department of System Information Sciences, Graduate School of Information Sciences, Tohoku University, Miyagi 980-8579, Japan (e-mail: koichi@ic.is.tohoku.ac.jp).

D. Prattichizzo and A. Vicino are with the Department of Information Engineering, University of Siena, 53100 Siena, Italy (e-mail: prattichizzo@dii.unisi.it; vicino@dii.unisi.it).

Digital Object Identifier 10.1109/TRO.2004.829456

is proposed for mobile robots which allows one to take into account image constraints. In [17], a modified PBVS has been proposed which allows one to keep the object frame origin in the FOV. In [18], a visual servoing invariant to the intrinsic parameters has been proposed which allows one to enlarge the FOV by decreasing the focal lengths. However, in most cases it is not guaranteed either that all points remain in the FOV or that the camera converges to the desired posture from any feasible initial one.

In this paper, a new visual servoing approach for dealing with the FOV problem is proposed which consists of a switching among PBVS strategies and backward motion without requiring knowledge of the 3-D model of the object. Specifically, classical PBVS is used when all points lie in a subarea of the image. In the case that at least one point lies outside such a subarea, PBVS rotational control law, PBVS translational control law, or backward motion are selected based on image points' motion prediction and distance between points and image boundary.

In the absence of uncertainty of the extrinsic parameters, all features are kept in the FOV. Moreover, if the intrinsic parameters are also known, the trajectory length is minimized in the rotational space and, in some cases, including all of those in which no point goes out from the subarea, also minimized in the translational space. Simulation results also show a certain degree of robustness against uncertainty on the intrinsic parameters. Although at present a formal proof seems difficult to obtain, it is expected that the proposed strategy is globally convergent.

Some drawbacks are also present which restrict the applicability of the proposed strategy, specifically the presence of chattering and the possibility that the camera is excessively far from the object due to the backward motion.

II. 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{e}_1 , \mathbf{e}_2 , and \mathbf{e}_3 the column vectors of \mathbf{I}_3 , $\|\mathbf{A}\|$ the Frobenius norm of matrix \mathbf{A} , and $[\mathbf{v}]_{\times}$ the skew-symmetric matrix of vector $\mathbf{v} = [v_1, v_2, v_3]^T \in \mathbb{R}^3$ defined as $[\mathbf{v}]_{\times} = [0, -v_3, v_2; v_3, 0, -v_1; -v_2, v_1, 0]$. Moreover, we define the following:

| | |
|--|---|
| $\{\boldsymbol{\theta}, \mathbf{c}\}$ | current camera posture, where $\boldsymbol{\theta} \in \mathbb{R}^3$ is the orientation expressed in exponential coordinates and $\mathbf{c} \in \mathbb{R}^3$ is the camera center with respect to an absolute frame; |
| $\{\boldsymbol{\theta}^*, \mathbf{c}^*\} = \{\mathbf{0}_3, \mathbf{0}_3\}$ | desired camera posture coinciding with the absolute frame; |
| $\mathbf{r}, \mathbf{t} \in \mathbb{R}^3$ | rotational and translational components of the current camera posture with respect to the desired one according to $\mathbf{r} = -\boldsymbol{\theta}$ and $\mathbf{t} = -e^{[-\boldsymbol{\theta}]_{\times}} \mathbf{c}$; |
| $\mathbf{m}_i, \mathbf{m}_i^* \in \mathbb{R}^3$ | image projections (in normalized homogeneous coordinates) of the i th 3-D point; |
| $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ | intrinsic parameters matrix; |
| $\boldsymbol{\omega}, \mathbf{v} \in \mathbb{R}^3$ | rotational and translational camera velocities expressed in the current camera posture. |

The problem is stated as follows. A set of n 3-D points is observed, first, from the desired camera posture and, second, from an initial one $\{\boldsymbol{\theta}_{\text{init}}, \mathbf{c}_{\text{init}}\}$. The point correspondences between the two views are known. Then, the goal is to steer the camera ensuring that all points are kept in the FOV and that the desired camera posture is reached for any feasible initial one.

Before proceeding, it is worthwhile to remember that, by using point correspondences from two perspective views, it is possible to recover

the rotation \mathbf{r} and the normalized translation $\mathbf{t}/\|\mathbf{t}\|$ of the camera displacement (see, e.g., [19] and [20]).

III. PROPOSED STRATEGY

A. Main Idea

The main idea of our strategy consists of realizing a switching visual servoing where a position-based control law is selected when all points lie inside the image, and some rotational or translational control laws are selected when at least one point lies on the image boundary in order to push such a point inside the image and to guarantee the final convergence. In particular, at any iteration¹ in which all points lie inside the image, the camera velocities are selected according to the following standard PBVS:

$$\begin{cases} \boldsymbol{\omega} = \lambda_r \mathbf{r} \\ \mathbf{v} = \frac{\lambda_t \mu \mathbf{t}}{\|\mathbf{t}\|} \end{cases} \quad (1)$$

where λ_r and $\lambda_t \in \mathbb{R}$ are positive control gains and $\mu \in \mathbb{R}$ is an image error ensuring the stop condition

$$\mu = \sqrt{\frac{1}{n} \sum_{i=1}^n \|\mathbf{m}_i - \mathbf{m}_i^*\|^2}. \quad (2)$$

Then, at any iteration in which at least one point lies on the image boundary, the camera velocities are selected as follows.

- If the points lying on the image boundary move inside the image by applying only the rotational control law in (1), then

$$\begin{cases} \boldsymbol{\omega} = \lambda_r \mathbf{r} \\ \mathbf{v} = \mathbf{0}_3. \end{cases} \quad (3)$$

- Otherwise, if the points lying on the image boundary move inside the image by applying only the translational control law in (1), then

$$\begin{cases} \boldsymbol{\omega} = \mathbf{0}_3 \\ \mathbf{v} = \frac{\lambda_t \mu \mathbf{t}}{\|\mathbf{t}\|}. \end{cases} \quad (4)$$

- Otherwise, the camera is sent away from the observed object through a backward translational motion along the the optical axis by selecting

$$\begin{cases} \boldsymbol{\omega} = \mathbf{0}_3, \\ \mathbf{v} = -\lambda_b \mu \mathbf{e}_3 \end{cases} \quad (5)$$

where $\lambda_b \in \mathbb{R}$ is a positive quantity.

Observe that, by selecting (5), we are always able to push inside the image any point that has reached the image boundary. Nevertheless, we restrict the selection of (5) as a last resort by verifying first if it is possible to push such points inside the image by selecting (3) and, second, if it is possible by selecting (4),² in order to avoid the backward motion that produces a longer trajectory in the translational space.

This basic idea, however, cannot be directly implemented since it is not possible to guarantee that a point escaping from the FOV lies on the image boundary at an iteration before leaving the image. Moreover, the predicted direction along which the points move by selecting (3) or (4) depends on the available estimate of the intrinsic parameters, and, hence, the strategy would not be robust against calibration errors.

¹We use the term ‘‘iteration’’ since visual servo systems are practically discrete-time systems.

²The chosen priority between (3) and (4) is due to the fact that it is not possible to achieve the translational convergence with (4) without first achieving the rotational one with (3).

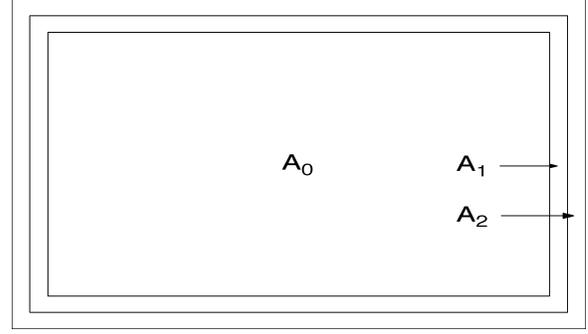


Fig. 1. Partition of the image in bands A_2 and A_1 and area A_0 .

TABLE I
SWITCHING VISUAL SERVOING ALGORITHM: CAMERA VELOCITIES
SELECTION IN FUNCTION OF THE CURRENT STATE

| algorithm state | camera velocities |
|-----------------|-------------------|
| S1 | select as in (1) |
| S2R | select as in (3) |
| S2T | select as in (4) |
| S3 | select as in (5) |

B. Algorithm

First, in order to deal with the problem introduced by the time discretization, we define a band along the image boundary for detecting points escaping from the FOV. In fact, while it is not possible to guarantee that an escaping point lies on the image boundary at any iteration, it is possible to guarantee that such point lies in a sufficiently large band along the image boundary. Second, in order to achieve a larger robustness degree against uncertainties on the intrinsic parameters, we do not select (3) and (4) if at least one point is too close to the image boundary. Hence, we define a second band surrounded by the previously defined one and select (5) if at least one point lies on the outer band, while we eventually select (3) and (4) if no point lies on the outer band and at least one lies on the inner band. Fig. 1 shows the image partitioned in band A_2 (outer band), band A_1 (inner band), and main area A_0 .

In order to formally state the proposed strategy, let us define the following conditions:

- condition CR: any point lying outside A_0 initially moves toward A_0 by selecting (3);
- condition CT: any point lying outside A_0 initially moves toward A_0 by selecting (4).

(Conditions CR and CT can be easily checked from \mathbf{r} and $\mathbf{t}/\|\mathbf{t}\|$ as shown in [21].) Then, let $\delta > 0$ be any constant and let us introduce the algorithm states defined as follows:

- state S1: (all points lie inside A_0) or (all points lie inside $A_0 \cup A_1$ and $\|\mathbf{r}\| \leq \delta$);
- state S2R: (S1 does not hold) and (all points lie inside $A_0 \cup A_1$ and (condition CR holds) and ($\|\mathbf{r}\| > \delta$);
- state S2T: (S1 and S2R do not hold) and (all points lie inside $A_0 \cup A_1$) and (condition CT holds) and ($\|\mathbf{r}\| > \delta$);
- state S3: S1, S2R, and S2T do not hold.

Therefore, the proposed strategy consists of selecting, at any iteration, the camera velocities $\boldsymbol{\omega}$ and \mathbf{v} , as shown in Table I. Constant δ is used to ensure that the system does not stall in state S2R and can be freely selected.

C. Remarks

In order to correctly use the proposed strategy, the following assumptions have to be satisfied:

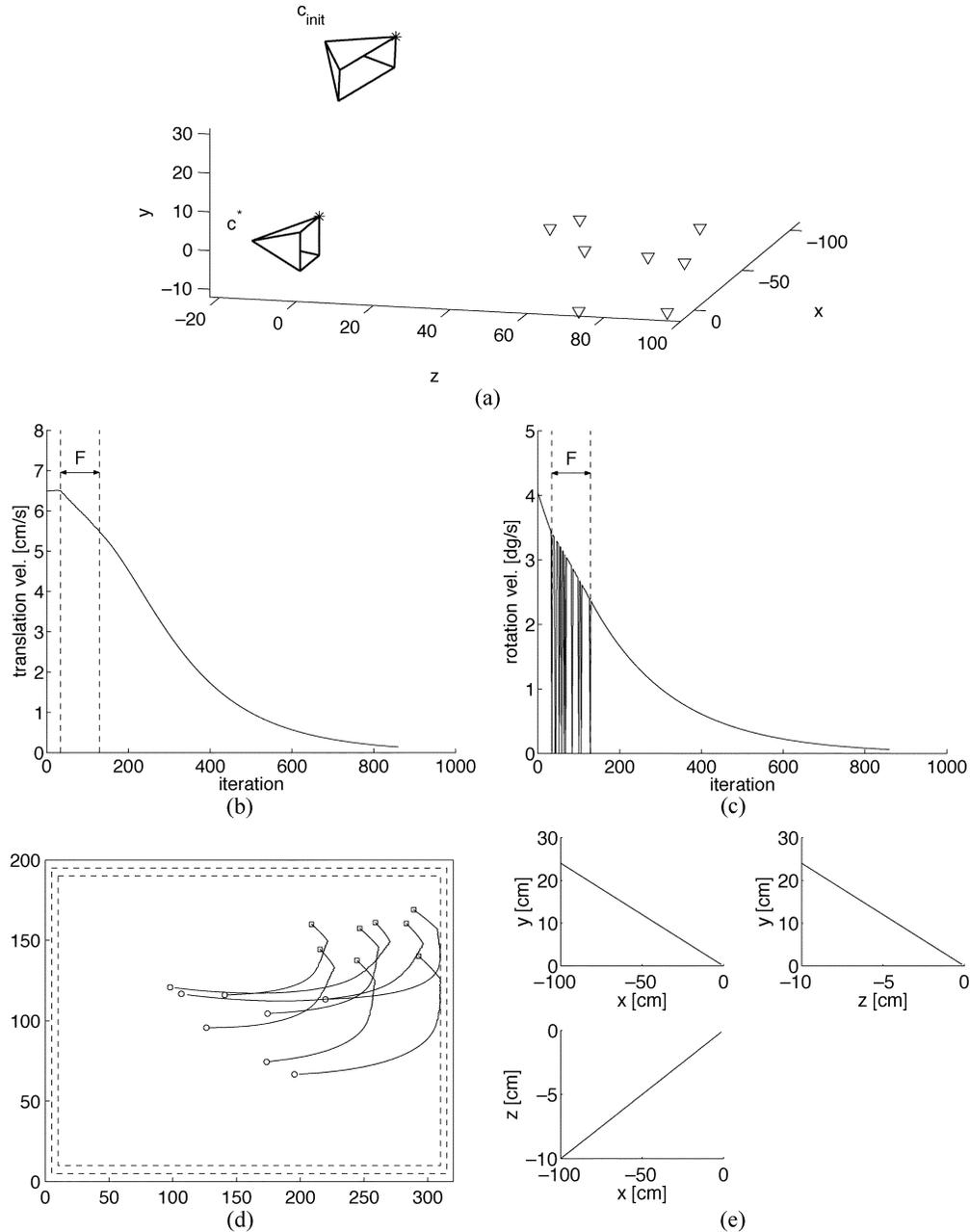


Fig. 2. Example 1. (a) Initial and desired postures. (b) Translation velocity $\|v\|$. (c) Rotation velocity $\|\omega\|$. (d) Camera view. (e) Translational trajectory.

- 1) the thickness of band A_2 is not less than the maximum shift of any image point between two sampling instants (since visual servoing is generally realized as a discrete-time systems);
- 2) all points lie inside A_0 in the desired view.

The first assumption ensures that no point lying in A_1 at any sampling instant may lie outside the image at the next sampling instant, hence allowing the backward motion to keep all features in the FOV in the absence of uncertainty on the extrinsic parameters. The second assumption is used to guarantee that close to the desired posture no switch is generated, and hence, convergence can be achieved.

It is easy to verify that, in the absence of calibration errors, the trajectory length is minimized in the rotational space and, in some cases, including all of those in which no point goes out from A_0 , minimized also in the translational space.

At present, it is not completely clear whether global convergence is achieved due to the difficulty of obtaining a formal proof for the

proposed strategy. However, no example falsifying the conjecture of global stability has been found yet.

It can be also shown that local stability, that is, convergence from sufficiently close initial camera postures, is guaranteed in the presence of sufficiently small calibration errors (the proof is based on the fact that the estimates of r , t and the points motion prediction used in establishing conditions CR and CT are continuous functions of the calibration errors).

Finally, some drawbacks are also present which restrict the applicability of the proposed approach, specifically the presence of chattering and the possibility that the camera is excessively far from the object due to the backward motion used to ensure the visibility constraint.

IV. RESULTS

In both simulation and experimental results, a configuration of $n = 8$ points is used to estimate the camera displacement through the essential matrix algorithm.

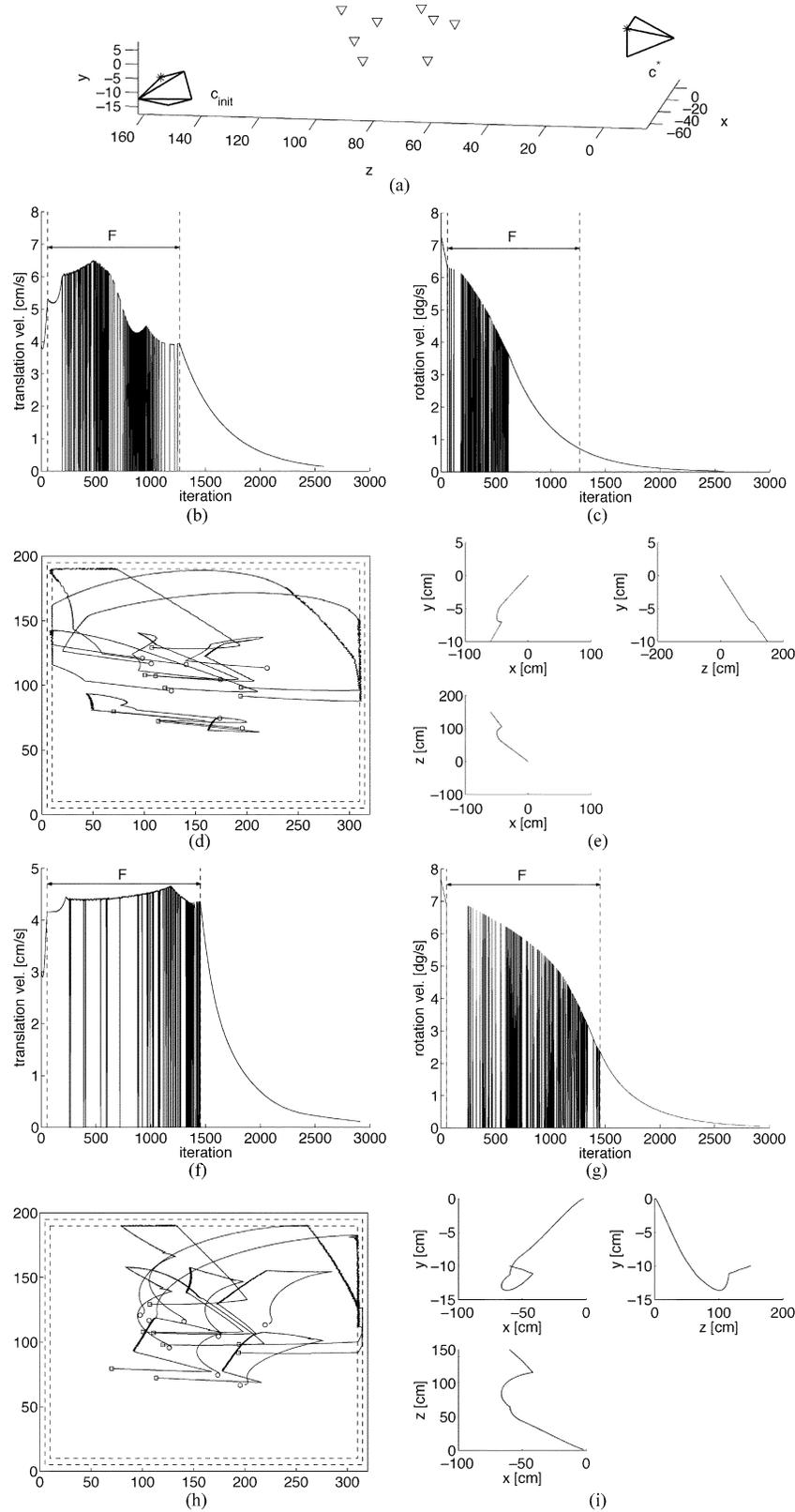


Fig. 3. Example 2. (a) Initial and desired postures. (b) Translation velocity $\|\mathbf{v}\|$. (c) Rotation velocity $\|\boldsymbol{\omega}\|$. (d) Camera view. (e) Translational trajectory. (f)–(i) The same quantities of (b)–(e) in the presence of uncertainty on the intrinsic parameters.

A. Simulations

The screen size is 320×200 pixels, bands A_1 and A_2 each have a thickness equal to five pixels, $\mathbf{K} = [320, 0, 160; 0, 200, 100; 0, 0, 1]$, $\delta = 0.1$ rad, and the sampling time is 50 ms.

Example 1: The initial camera posture is $\{\boldsymbol{\theta}_{init}, \mathbf{c}_{init}\} = \{[\pi/12, \pi/6, -\pi/8]^T \text{ rad}, [-100, 24, -10]^T \text{ cm}\}$ as shown in Fig. 2(a). Fig. 2(b)–(e) shows the results obtained in the absence of uncertainties. Since the intrinsic parameters are supposedly known,

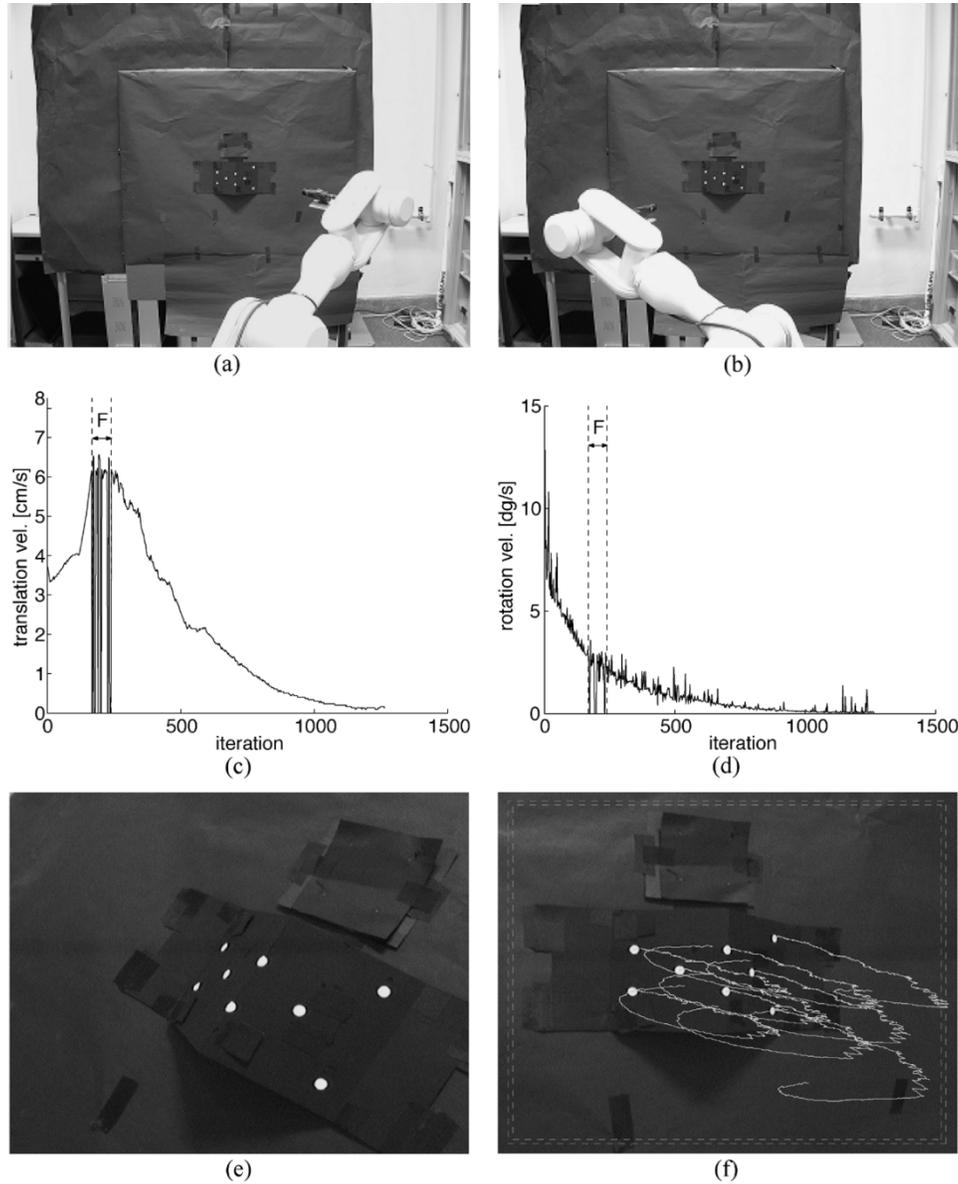


Fig. 4. Experiment. (a) Initial posture. (b) Desired posture. (c) Translation velocity $\|\dot{\mathbf{v}}\|$. (d) Rotation velocity $\|\dot{\boldsymbol{\omega}}\|$. (e) Initial view. (f) Desired view.

the trajectory length in the rotational space is minimized. Observe that, in this case, the trajectory length is minimized also in the translational space as indicated by Fig. 2(e), which shows that the camera center follows a straight line (that is, the algorithm never reaches state S3 corresponding to the backward motion). The algorithm state is S1 outside the interval F shown in Fig. 2(b) and (c), inside which state switches among S1, S2R, and S2T.

Example 2: In this example, a large initial camera displacement is considered, specifically $\{\theta_{\text{init}}, \mathbf{c}_{\text{init}}\} = \{[\pi/10, 4\pi/5, \pi/10]^T \text{ rad}, [-60, -10 \ 150]^T \text{ cm}\}$ corresponding to a main rotation angle of about 140° , as shown in Fig. 3(a). Fig. 3(b)–(e) shows the results obtained in the absence of uncertainties. Contrary to *Example 1*, the algorithm this time reaches state S3, and hence, the backward motion is used to keep points in the FOV (in interval-F state switches among S1, S2R, S2T, and S3). This simulation has been carried out also in the presence of large uncertainties, specifically by using the wrong estimate $\hat{\mathbf{K}} = [410, 0, 200; 0 \ 290 \ 130; 0, 0, 1]$ pixels. The results are shown in Fig. 3(f)–(i).

B. Experiments

Fig. 4(a) and (b) shows the robot manipulator (a six-degree-of-freedom industrial robot MITSUBISHI PA10) and the observed object used in the experiment carried out at the Ishikawa Hashimoto Laboratory. The screen size is 505×459 pixels, bands A_1 and A_2 have thicknesses equal to 5 and 12 pixels, respectively, $\delta = 0.1$ rad, and the sampling time is 50 ms. The extrinsic and intrinsic camera parameters are only approximately known. Fig. 4(c)–(f) shows the control velocities, the initial camera view, and the desired camera view with the points trajectory. Analogously to *Example 2*, the algorithm reaches state S3.

V. CONCLUSION

A visual servoing strategy for dealing with the problem of keeping the observed points in the FOV has been proposed. The approach consists of a switching among PBVS strategies and backward motion without requiring knowledge of the 3-D model of the object. In the

absence of uncertainty on the extrinsic parameters, all features are kept in the FOV. Moreover, if the intrinsic parameters are also known, the trajectory length is minimized in the rotational space and, for some cases, also minimized in the translational space. Simulation results have also shown a certain degree of robustness against uncertainty on the intrinsic parameters. Although at present a formal proof seems difficult to obtain, it is conjectured that the proposed strategy is globally convergent.

Some drawbacks are also present which restrict the applicability of the proposed strategy, specifically the presence of chattering and the possibility that the camera is excessively far from the object due to the backward motion. Future work will be devoted to solving these problems.

ACKNOWLEDGMENT

The authors would like to thank Dr. R. Mori and R. Shinohara for their precious help in the realization of the experiments and the reviewers of this paper for their constructive comments.

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. Robot. Automat.*, vol. 12, pp. 651–670, Oct. 1996.
- [3] R. Basri, E. Rivlin, and I. Shimshoni, "Visual homing: Surfing on the epipole," *Int. J. Comput. Vis.*, vol. 33, no. 2, pp. 22–39, 1999.
- [4] C. Taylor and J. Ostrowski, "Robust vision-based pose control," in *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, 2000, pp. 2734–2740.
- [5] A. Sanderson, L. Weiss, and C. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE J. Robot. Automat.*, vol. RA-3, pp. 404–417, Oct. 1987.
- [6] J. Feddema and O. Mitchell, "Vision-guided servoing with feature-based trajectory generation," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 691–700, Oct. 1989.
- [7] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1991, pp. 2267–2272.
- [8] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The Confluence of Vision and Control*, G. H. D. Kriegman and A. Morse, Eds. Berlin, Germany: Springer-Verlag, 1998, pp. 66–78.
- [9] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D visual servoing," *IEEE Trans. Robot. Automat.*, vol. 15, pp. 238–250, Apr. 1999.
- [10] G. Morel, T. Leibzeit, J. Szewczyk, S. Boudet, and J. Pot, "Explicit incorporation of 2D constraints in vision based control of robot manipulators," in *Experimental Robotics VI*, P. Corke and J. Trevelyan, Eds. Berlin, Germany: Springer-Verlag, 2000.
- [11] K. Deguchi, "Optimal motion control for image-based visual servoing by decoupling translation and rotation," in *Proc. Int. Conf. Intelligent Robots and Systems*, 1998, pp. 705–711.
- [12] P. Oh and P. Allen, "Visual servoing by partitioning degrees of freedom," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 1–17, Feb. 2001.
- [13] N. Cowan, J. Weingarten, and D. Koditschek, "Visual servoing via navigation functions," *IEEE Trans. Robot. Automat.*, vol. 18, pp. 521–533, Aug. 2002.
- [14] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 507–515, Aug. 2001.
- [15] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. Robot. Automat.*, vol. 18, pp. 534–549, Aug. 2002.
- [16] H. Zhang and J. Ostrowski, "Visual motion planning for mobile robots," *IEEE Trans. Robot. Automat.*, vol. 18, pp. 199–208, Apr. 2002.
- [17] 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. Robotics and Automation*, Washington, DC, 2002, pp. 1624–1629.
- [18] E. Malis, "Vision-based control invariant to camera intrinsic parameters: Stability analysis and path tracking," in *Proc. IEEE Int. Conf. Robotics and Automation*, Washington, DC, 2002, pp. 217–222.
- [19] O. Faugeras and Q.-T. Luong, *The Geometry of Multiple Images*. Cambridge, MA: MIT Press, 2001.
- [20] 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. J. Comput. Vis.*, vol. 37, no. 1, pp. 79–97, 2000.
- [21] G. Chesi, K. Hashimoto, D. Prattichizzo, and A. Vicino, "A switching control law for keeping features in the field of view in eye-in-hand visual servoing," in *Proc. IEEE Int. Conf. Robotics and Automation*, Taipei, Taiwan, R.O.C., 2003, pp. 3929–3934.

A Novel Filter for Terrain Mapping With Laser Rangefinders

Cang Ye and Johann Borenstein

Abstract—This paper introduces a novel filter for terrain mapping with a two-dimensional laser rangefinder. The filter, called the certainty-assisted spatial (CAS) filter, uses the physical constraints on motion continuity and spatial continuity to identify corrupted pixels and missing data in an elevation map. The filter removes the corrupted pixels, fills in the missing data, and leaves the uncorrupted pixels intact so as to preserve the details of a terrain map. Our extensive indoor and outdoor mapping experiments show the CAS filter's superior performance in erroneous data reduction and map detail preservation over conventional filters.

Index Terms—Certainty map, elevation map, filtering, laser rangefinder (LRF), mixed pixels, terrain mapping.

I. INTRODUCTION

Autonomous navigation of mobile robots on rugged terrain requires the capability to decide whether an obstacle should be traversed or circumnavigated. The ability to make this decision and to actually execute it is called "obstacle negotiation" (ON). A crucial issue involved in ON is terrain mapping. Research efforts on terrain mapping have been devoted to indoor environments [1], outdoor, off-road terrain [2]–[4], as well as planetary terrain [5]–[7]. Most of the existing methods employ stereovision [2], [3], [7], which is sensitive to environmental conditions (e.g., ambient illumination) and has low-range resolution and accuracy. As an alternative or supplement, three-dimensional (3-D) laser rangefinders (LRFs) have been employed since the early 1990s [5], [6], [8]. However, 3-D LRFs are usually costly, bulky, and heavy. Therefore, they are not suitable for small and/or expendable robots. Furthermore, most of them are designed for stationary use due to the slow frame rate.

A more feasible solution for lower-cost robots is a two-dimensional (2-D) LRF. Researchers at Carnegie Mellon University (CMU), Pittsburgh, PA, [1] used a SICK 2-D LRF looking *upward* to perform indoor 3-D mapping. The 3-D map is not suitable for ON since no ground map

Manuscript received December 27, 2002; revised July 3, 2003. This paper was recommended by Associate Editor G. Oriolo and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported in part by the Department of Energy under Award DE-FG04-86NE3796 and in part by the University of Michigan's Automotive Research Center (ARC) under a grant funded by TACOM.

The authors are with the Mobile Robotics Laboratory, Department of Mechanical Engineering, The University of Michigan, Advanced Technologies Laboratory, Ann Arbor, MI 48109-2110 USA (e-mail: yecang@engin.umich.edu; johannb@umich.edu).

Digital Object Identifier 10.1109/TRO.2004.829457