

An Experimental Study of Hybrid Switched System Approaches to Visual Servoing

Nicholas R. Gans and Seth A. Hutchinson

ngans@uiuc.edu, seth@uiuc.edu

Dept. of Electrical and Computer Engineering
 The Beckman Institute for Advanced Science and Technology
 University of Illinois at Urbana-Champaign
 Urbana, IL USA

Abstract—In the recent past, many researchers have developed control algorithms for visual servo applications. In this paper, we introduce a new hybrid switched system approach, in which a high-level decision maker selects between two visual servo controllers. We have evaluated our approach with simulations and experiments using three individual visual servo systems and three candidate switching rules. The proposed method is very promising for visual servo tasks in which there is a significant distance between the initial and goal configuration, or the task is one that can cause an individual visual servo system to fail.

I. INTRO

Visual servoing has proven to be a highly effective means to control a robot manipulator through the use of visual data. It provides a high degree of accuracy using even simple camera systems and robustness in the face of signal error and uncertainty of system parameters.

Visual servo methods have classically been divided into two camps, Position Based Visual Servoing (PBVS) and Image Based Visual Servoing (IBVS). There are extensive resources detailing these methods [1–4]. In the late nineties, Chaumette outlined a number of problems that cannot be solved using the traditional local linearized approaches to visual servo control [5]. This resulted in a variety of partitioned visual servo systems which used the image Jacobian linearization of IBVS for specific degrees of freedom, and 3D techniques exemplified in PBVS for the remainder [6–10].

Rather than combining systems, another approach is the use of hybrid switched systems, i.e., systems comprised of a set of continuous subsystems along with a discrete switching control [11, 12]. Hybrid switched systems can offer an increased region of stability and increased rate of convergence, and there exists the potential to switch between unstable systems in a pattern that makes the total system stable.

Section II will provide an introduction to the theory behind hybrid switched systems. In Section III, we discuss our individual visual servo systems. In Sections IV and V we will present the two switched systems along with their simulated and experimental results.

II. HYBRID SWITCHED SYSTEM CONTROL

The theory of hybrid switched control systems, i.e., systems that comprise a number of continuous subsystems and a discrete system that switches between them, has received notable attention in the control theory community [11–13]. In general, a hybrid switched system can be represented by the differential

equation

$$\dot{x} = f_\sigma(x) : \sigma \in \{1..n\} \quad (1)$$

where f_σ is a collection of n distinct functions. For our purposes, it is convenient to explicitly note that the switching behavior directly affects the choice of the control input u

$$\dot{x} = f_\sigma(x, u_\sigma) : \sigma \in \{1..n\}. \quad (2)$$

A useful interpretation is to consider σ to be a discrete signal, switching among discrete values in 1..n. The value σ at time t determines which function $f(x, u_\sigma)$ is used. The signal σ is typically classified as state-dependent or time-dependent, depending on whether switches are caused by the state of x or the time t , although overlap does exist between these classes. In our research we explored state-dependent switching contingent on the state of the image plane or camera pose, a time-dependent switch induced by a random variable, and a combined method where a random variable influenced by the state determined switches.

The systems we present are each comprised of two visual servo controllers; each visual servo controller provides a velocity screw, $u = [T_x, T_y, T_z, \omega_x, \omega_y, \omega_z]^T$, and a switching rule determines which is used as the actual control input at each control cycle.

The stability of a switched system is not insured by the stability of the individual controllers. Indeed, a collection of stable systems can become unstable when inappropriately switched. As an illustration, Figure 1 (from [13]) shows trajectories for two asymptotically stable subsystems in (a) and (b). A set of switches resulting in a stable system is shown in (c), while a series of switches resulting in an unstable system are shown in (d).

Stability of switched systems can be proven using Lyapunov's direct method [13, 14]. Generally this requires establishing a common Lyapunov function that works for all subsystems. Alternately, one can establish a family of Lyapunov functions for the systems such that at each switch, the value of the function at the end of that interval is less than the value of the function of the interval that proceeded it, as illustrated for a one dimensional family of two functions in Figure 2.

Stability of a switched system can be extremely difficult to prove. However, we have performed extensive empirical evaluations that demonstrate the efficacy of our approach. We will turn our attention to establishing stability in the near future.

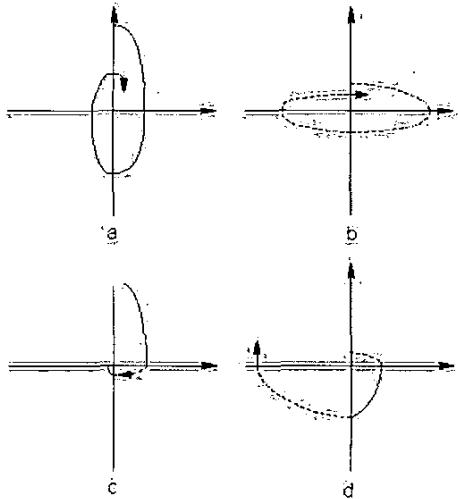


Fig. 1. trajectories of switched systems

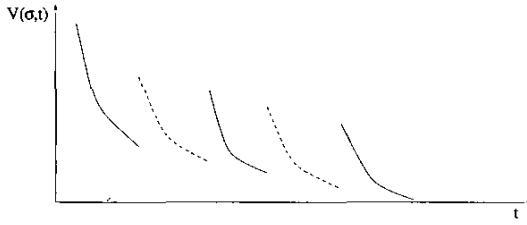


Fig. 2. stable family of Lyapunov function

III. THREE VISUAL SERVO CONTROLLERS

We present the three visual servo controllers used by our switched systems.

A. Homography Based Controller

The homography method exploits the epipolar constraints between two images of planar feature points. The homography matrix has been used previously for visual servoing in [6, 8] to control a restricted set of degrees of freedom. We however, use it to control all degrees of freedom.

Define \mathbf{f}^* , \mathbf{f} , as the homogeneous coordinates in two images of a set of 3D points lying on a plane π , where * denotes features in the goal image.. These are related by

$$\mathbf{f}^* = \mathbf{H}\mathbf{f} \quad (3)$$

where \mathbf{H} is the calibrated homography matrix. As shown in [15, 16], \mathbf{H} can be decomposed as

$$\mathbf{H} = \mathbf{R}(\mathbf{I}_3 - \frac{\mathbf{t}\mathbf{n}^T}{d}) \quad (4)$$

where \mathbf{I}_3 is a 3×3 identity matrix and \mathbf{R} and \mathbf{t} are the rotation matrix and translation vector, respectively, relating the two camera views. The parameter \mathbf{n} is the the normal of the plane π and describes the orientation of π with respect to the current

camera view; d is the distance from the current camera origin to the plane π . We calculate the vector $\mathbf{T} = [T_x, T_y, T_z]^T = \hat{\mathbf{d}}\mathbf{t}$, where $\hat{\mathbf{d}}$ is an estimate of d . Given knowledge of the geometry of the feature point locations it is possible to accurately estimate $\hat{\mathbf{d}}$ and so determine \mathbf{t} to the proper scale. From the rotation matrix \mathbf{R} , we extract the roll, pitch and yaw angles, $\omega_z, \omega_x, \omega_y$, obtaining the velocity screw $\mathbf{u} = k[T_x, T_y, T_z, \omega_x, \omega_y, \omega_z]$ in which k is a scalar gain constant, or a 6×6 gain matrix..

Of the numerous methods to calculate \mathbf{H} , we have used a linear solution since visual servoing, in general, requires quicker calculations than iterative methods may provide. Decomposing the homography as in (4) is not a trivial exercise and generally cannot be solved to a unique solution. Additional information pertaining to the use of the homography in visual servoing can be found in [6].

Since this method provides rotation and translation vectors to completely realize the camera's goal position from its current position, it shares many of the performance characteristics of PBVS systems. Namely this system will perform optimally in Cartesian space. The end effector will typically follow the shortest path to the goal position. This, however, can lead to large motions of the features in the image space. This can cause the feature points to leave the field of view, resulting in system failure. We will define system failure as any time that a system cannot zero the error within 250 iterations.

B. Affine-Approximation Controller

For camera motions that do not involve rotation about the camera x - or y - axes, the initial and goal images will be related by an affine transformation. While this is a constrained set of motions, it is common in many situations such as aligning camera with a component on a conveyor belt.

Define $\tilde{\mathbf{f}}^*$, $\tilde{\mathbf{f}}$, as the calibrated pixel coordinates of two points in the image plane. Again, * indicates the features are in the goal image. Then these points are related by the affine transformation

$$\begin{aligned} \tilde{\mathbf{f}}^* &= \mathbf{A}\tilde{\mathbf{f}} + \mathbf{b} \\ &= \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \begin{bmatrix} C_\theta & -S_\theta \\ S_\theta & C_\theta \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \end{aligned} \quad (5)$$

in which C_θ and S_θ denote respectively $\cos \theta$ and $\sin \theta$; f_x and f_y are image point coordinates; a , s_i , and θ describe the skew, scale, and rotation respectively; and \mathbf{b} is the translation. Both \mathbf{A} and \mathbf{b} can be obtained by solving a linear system of equations, and QR decomposition can then be used to determine a , s_i , and θ .

$$\tilde{\mathbf{f}}^* = \mathbf{R}\mathbf{Q}\mathbf{f} + \mathbf{b} = \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{bmatrix} \begin{bmatrix} q_{21} & q_{12} \\ q_{11} & q_{22} \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (6)$$

The \mathbf{Q} matrix is a permutation of the rotation matrix in (5), and rotation θ about the camera z -axis equals $\text{arcsine}(q_{21})$. During an affine transformation, the rotations about the x - and y -axes are, by definition, zero. The r_{11} and r_{22} of (6) respectively equal the s_1 and s_2 parameters of (5) and provide z -axis translation to scale. Translation along the x - and y - axes are defined to scale in the vector \mathbf{b} . Multiplying the scaled translations by a depth estimate will provide true values. Again, knowledge

of the feature point geometry will allow for the depth estimate to be accurately derived.

Given a, s_i, θ, t_x, t_y we again have the position and orientation relating the initial and camera goal positions. This controller provides the velocity screw $u = k[t_x, t_y, s_2, 0, 0, \theta]$ where k is a gain constant or matrix. Note that if there is no rotation about the x - or y -axes, we will have $s_1 = s_2$.

C. IBVS

There is a vast amount of literature regarding IBVS systems [4, 5, 17, 18]. In IBVS systems, the control exists in the image space. In the common case of a camera mounted on the robot end effector, the motion of a two-dimensional feature point $\mathbf{f} = [\mathbf{u}, \mathbf{v}]^T$ in the image is related to the velocity screw of the end effector $\dot{\mathbf{r}} = [T_x, T_y, T_z, \omega_x, \omega_y, \omega_z]^T$ by the relation

$$\dot{\mathbf{f}} = \mathbf{J}_{\text{im}}(\mathbf{u}, \mathbf{v}, \mathbf{z})\dot{\mathbf{r}}, \quad (7)$$

where \mathbf{J}_{im} is the image Jacobian [1, 2]. Given at least three feature points, it is possible to use (7) to construct the control law

$$\mathbf{u} = k\mathbf{J}_{\text{im}}^+(\mathbf{r})\dot{\mathbf{f}} \quad (8)$$

where \mathbf{u} will be the velocity screw and k is a scalar gain factor or a 6×6 gain matrix.

Under this control, feature points tend to move in straight lines to their goal positions. This provides desirable performance in the image space, but as first reported by Chaumette [5], it can lead to extraneous motions of the end effector in 3D Cartesian space. These motions can lead to singular positions for the robot or singularities in the image Jacobian, leading to task failure.

IV. A IBVS/HOMOGRAPHIC HYBRID SWITCHED SYSTEM

Our first switched system presented here uses the IBVS and homographic methods as sub-systems. A higher level decision maker determines which system to use at each iteration. This system was designed in hopes of maximizing the strengths IBVS and PBVS systems as discussed in Section III.

As noted in Section II, we explored three switching signals: **State-Dependent Switching**. We attempt to avoid the weaknesses of both systems by switching when the current system is approaching a problematic state. We determine a threshold level for how far the feature points will be allowed to stray from the center of the image, as well as a threshold on the distance we will allow the camera to move from the feature points. At each iteration, we compare each switching parameter to its threshold. If we are using IBVS and move past the threshold distance from the feature point plane we switch to PBVS to bring us towards the goal position and end camera retreat. If we are using PBVS and the feature points move outside the threshold distance to the image center we switch to IBVS to bring the image points towards their goal configuration, which is commonly centered. **Random Switching**. Random switching has been used in control systems for such tasks as task routing [19]. At each iteration, We use a binary random variable to select between the two systems with equal probability. This provides a strong test to stability under arbitrary switching, and can tell us whether an undesirable switching pattern may result in instability.

Biased Random Switching. The state-dependent levels discussed above are now inputs to a probabilistic function used to determine the next system used. The farther the camera is from the image plane, the more likely the system is to choose the PBVS method. Likewise, the farther the feature points are from the image center, the more likely the system is to choose to use IBVS. The probability will be unity at either threshold.

We first present a series of simulations to show performance under ideal conditions. Simulations were performed for an ideal camera with a 512×512 pixel array, with each pixel measuring $10\mu\text{m} \times 10\mu\text{m}$ and a focal length of 7.8mm. We allowed perfect depth estimation. Visual servoing was halted if the pixel error was reduced below 1 pixel, or had converged to steady state for ten iterations.

We simulate a goal image where the feature points are close to the image border, and an error image where the camera is rotated by 160° about the optical axis. This is an extremely difficult task for the individual subsystems. In our simulations, using only the PBVS method would result in a loss of the feature points, and using only the IBVS method induced a camera retreat of 10 meters. Either of these would likely cause failure in a physical system. All three methods of switching were successfully able to zero the error.

The first simulation was the state-dependent switching system. Figure 3 shows the feature point errors, the velocity screw, the value of our switching parameters and the feature point positions at each iteration. Tick marks at the bottom of the graphs show the system currently being used: black for IBVS, cyan for PBVS. The color of the position lines follow the same color scheme regarding which system is determining the motion.

The feature points begin far from the center of the image, so we begin in IBVS mode to bring the points towards the goal. Indeed, the maximum error decreases, along with a sharp increase in the distance of the camera from the feature points. We enforced a threshold of 1.75 meter for the camera distance, so the camera switches to PBVS when camera retreat reaches this distance. The camera retreat is corrected along with completion of the rotation.

Figure 4 shows results for the random switching method. The feature points are kept within the image, and the camera retreats less than under the state based switching method. This is due to the large number of switches; since the switching is entirely random, it is possible to select the IBVS method the majority time and experience extreme camera retreat, although this never happened during our simulations or experiments. It does take slightly longer in this case to zero the error than state based switching.

Our final simulation result is for probabilistic switching, shown in Figure 5. The feature point trajectories closely resemble those of the random method. The feature point excursion is kept low, and the camera distance is also lower than that experienced under either the state-dependent method or random method, and it is slightly faster than both other systems as well.

Our experiments were performed using a camera mounted on the end effector of a PUMA 560 robot. The camera is a Sony VFW-V500, which has a 640×480 color pixel display. The lens focal length is 14.4mm. The feature points consisted of four color dots on a black sheet. The image was thresholded in

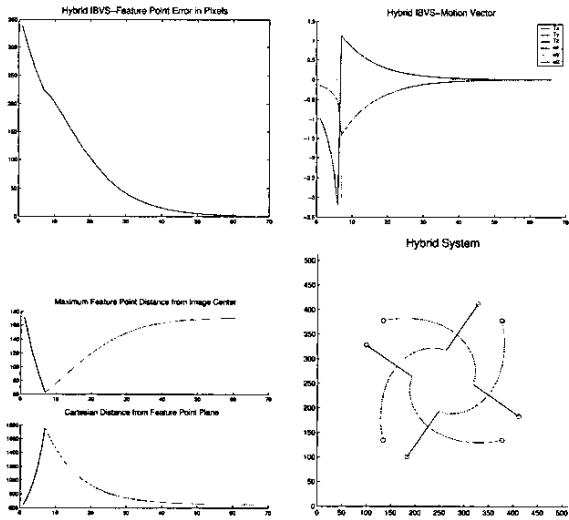


Fig. 3. State-Dependant Switching

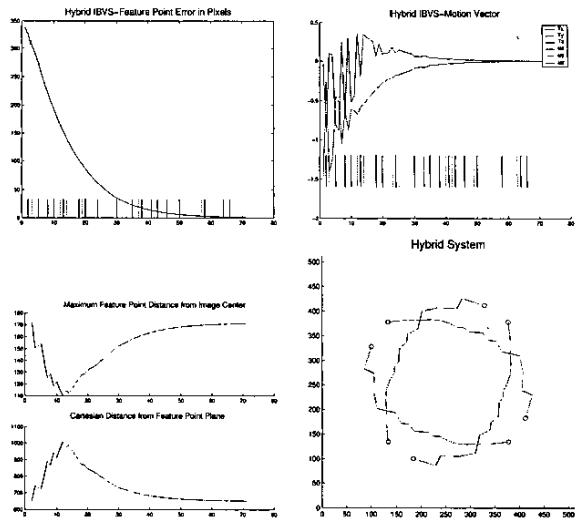


Fig. 5. Probabilistic Switching

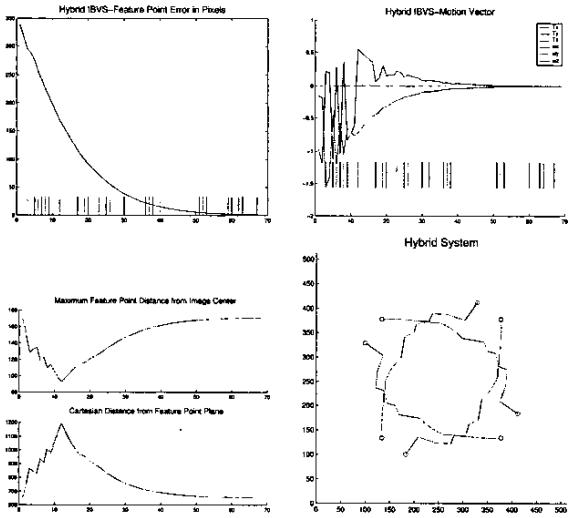


Fig. 4. Random Switching

RGB space to locate the center points of each dot. This provided 4 co-planar feature points.

The first experiments were very similar to the simulations. They involved a goal image with the camera very close to the feature points, and an initial offset consisting of a large rotation about the optical axis. Two such images can be seen in Figure 6.

Naturally, the individual systems did not perform ideally in our experiments due to the presence of camera calibration and depth estimation errors. For instance, the the feature point error did not strictly become smaller at every iteration under IBVS. The systems do well however, and did perform expected motions such as IBVS camera retreat during rotation. We feel that the fact that the systems worked well, even when the subsystems did not perform ideally, is a testament to the strength of the switched system.

During live experiments, both PBVS and IBVS systems failed

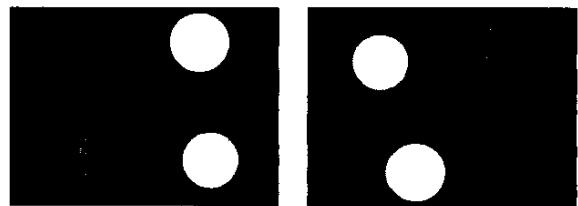


Fig. 6. Goal and Initial Image

this task if used individually. PBVS lost the feature points, and IBVS, experienced a great deal of camera retreat, losing focus of the image, and ultimately losing the feature points when making rotations. The figures show the same data we presented in the simulated results, with some minor changes. The graphs of both the feature point error and feature point trajectories have the color of the dot they correspond to, and trajectories with a black shadow indicate that PBVS was used to calculate that motion.

Figure 7 shows the results for the state-dependent switching method. Since the feature points are close to the image edge we are begin by using IBVS. As expected, the camera retreat rotates, bringing the the feature points towards the image center along with an increase in camera distance. Finally PBVS takes over and is able to reduce both. The system is unable to completely zero the feature point error, after 250 iterations when visual servoing was halted. Results for random selection are seen in Figure 8. The maximum feature distance from the image center is higher than seen in the state-dependent method, but the camera retreat is kept much lower. Due to the lower camera retreat, the system is able to zero the error faster than either the state-dependent or probabilistic methods. However, the maximum feature point distance is 250 pixels; clearly, the error must be along the horizontal image axis, or the feature points would have left the field of view and the system would have failed. This indicates a potential for system failure using the random method, though the system never failed during our experiments.

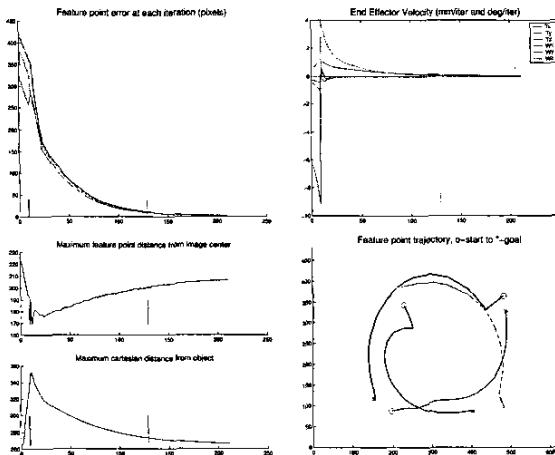


Fig. 7. State-Dependant Switching

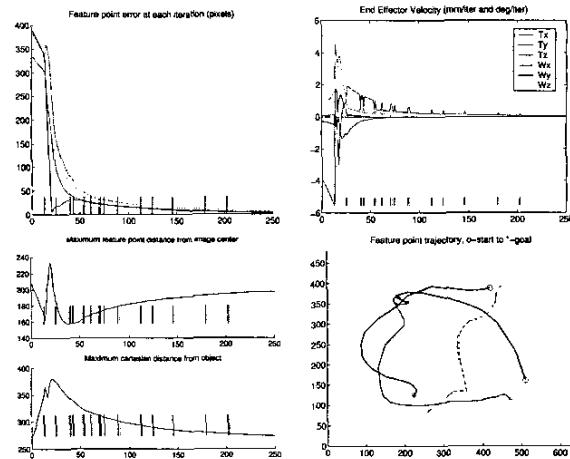


Fig. 9. Probabilistic Switching

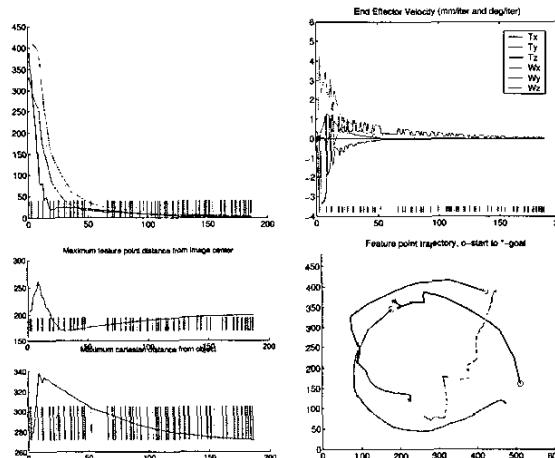


Fig. 8. Random Switching

The probabilistic method is presented in Figure 9. The method choices are identical to the state-dependent method for the first thirty iterations. After this point the switching does become fairly random, though the velocity screw shows much evidence of change due to switching. This system is also unable to completely zero the error after 250 iterations, due to remaining optical axis translation.

We repeated the experiments using an oblique view involving heavy rotation about the camera y-axis, moderate rotation about the camera z-axis, and translation along all axes. The goal and initial images are shown in figure 10.

For this task, both subsystems are capable of zeroing the feature point error, and camera retreat is not a dangerous factor here. Results can be seen in figures 11, 12 and 13. The state-dependent and probabilistic methods again perform very similarly. The random system is able to zero the error more quickly than the other two methods, but in general experiences a larger feature point error.

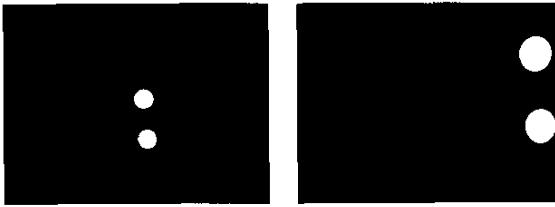


Fig. 10. Goal and Initial Image

V. AN AFFINE/HOMOGRAPHIC HYBRID SWITCHED SYSTEM

The major strength of the homography-based controller in this system is that it is the only controller capable of handling general motions which include rotation about the x and y -axes. If the camera motion does not involve such rotations, the two approaches have similar performance. However, in the presence of noise, the affine method is much more accurate. We conducted a series of Monte Carlo tests in which both systems performed

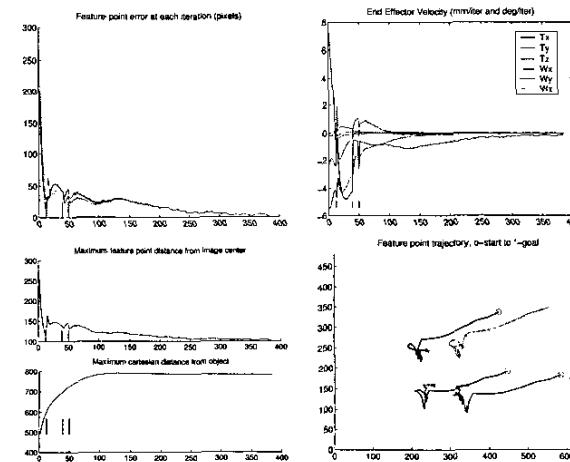


Fig. 11. State-Dependant Switching

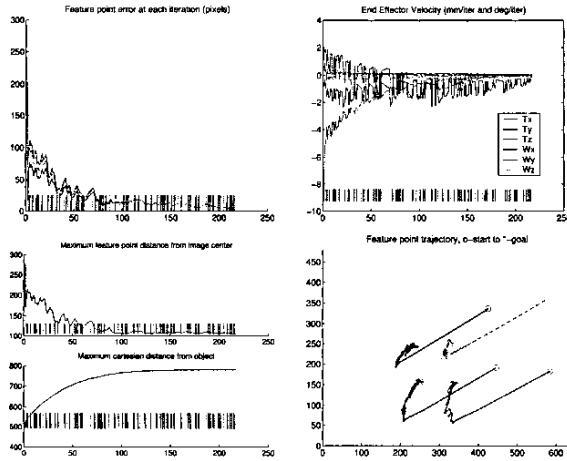


Fig. 12. Random Switching

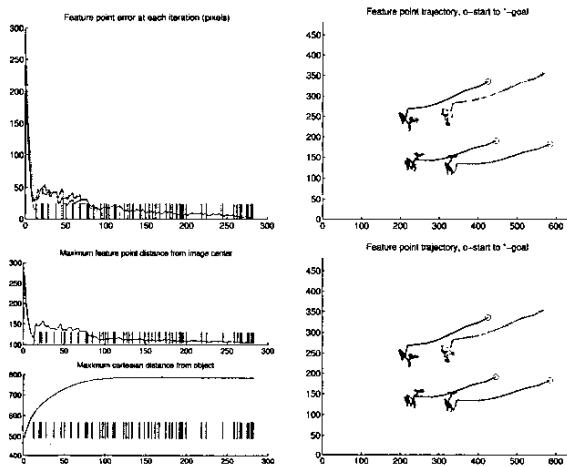


Fig. 13. Probabilistic Switching

an identical affine motion under the effects of increasing white noise. Under large amounts of noise, the homography-based method typically had an error in the pose estimation that was fifty times greater than the affine approach, and error in the total rotation was almost fifteen times greater.

We again explored three switching rules:

State-Dependent Switching. After solving the homography, we take the RMS value of the rotations about x - and y - and compare it to a threshold value, if the amount of rotation is less than this value, we select the affine solution. We have found that the affine method can successfully zero the error for any motion with a rotation about y or x less than 0.5° .

Random Switching. As discussed in Section IV.

Biased Random Switching. The current RMS of x - and y -rotation is used to determine the value of a random variable which selects the current system.

We performed simulations and experiments of a similar camera motion task, an oblique view of the image plane requiring motion along each degree of freedom to zero the error. The simulation and camera configurations are the same as discussed in

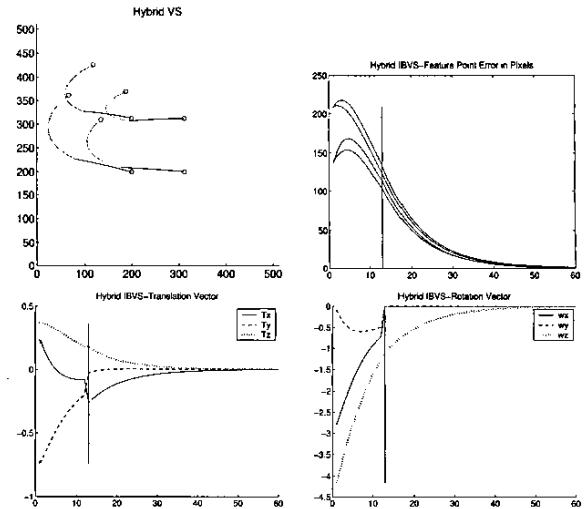


Fig. 14. General Motion, State-Dependant Switching

Section IV.

We first present the simulations using the state-dependent switching method. For the results in Figure 14 a rotation of thirty degrees of the feature point plane about both the world y -axes is followed by a moderate camera translation along all degrees of freedom and rotation about the optical axis. This causes a general motion involving all degrees of freedom.

The top left image shows the feature point trajectory; black line segments are motions induced by the affine method, while cyan portions are induced by the homographic method. We see the homographic method used for the first portion of the motion, with a switch to the affine method when the x and y rotation have been reduced. The upper right graph shows the pixel error for the four feature points. A black vertical line at the bottom of the graph indicates a switch to the affine method, a cyan line indicates a switch to the homographic system. Clearly a switch to the affine method occurred at about the twelfth iteration, and causes a slight incongruity in the velocities, though the error remains fairly smooth.

We repeated the previous test, but added a Gaussian random variable with variance 0.5 pixels to the feature point locations, simulating white noise. The results are seen in Figure 15. The system remains to the homographic method for almost all of the motion, briefly switching to the affine method at several points when the noise causes the $\text{RMS}(\omega_x, \omega_y)$ term to exceed the threshold. The system still zeros the error, though it takes longer than previous tests, over 75 iterations. The velocities are extremely rough in appearance due to the noise.

Figure 16 shows test results using the random switching method. The trajectory and motion vectors are irregular, though the feature point errors are smooth. It is worth noting that this system avoids the large feature point motions of the state-dependent switching system, which almost loses the feature points from the image plane.

We finally simulated the biased switching rule, shown in Figure 17. It has a similar appearance to the state-dependent switching system, remaining in the homography-based method for the

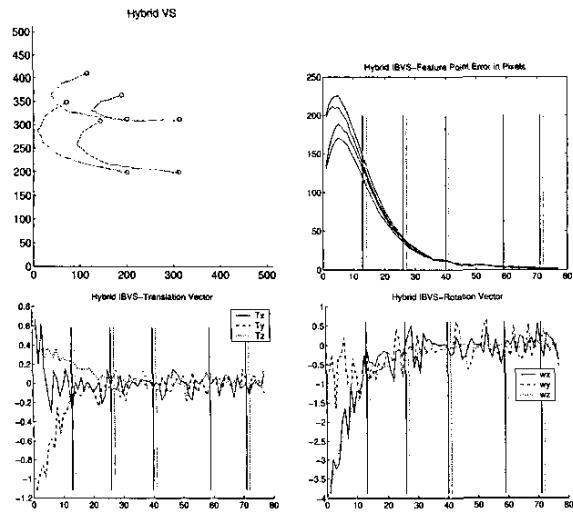


Fig. 15. General Motion with Noise, State-Dependant Switching

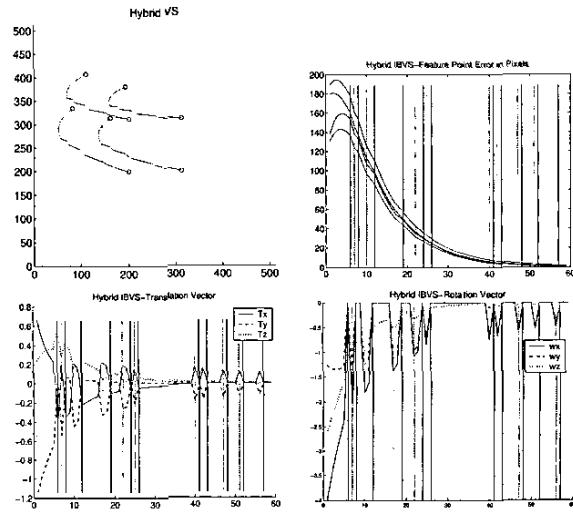


Fig. 17. General Motion, Biased Random Switching

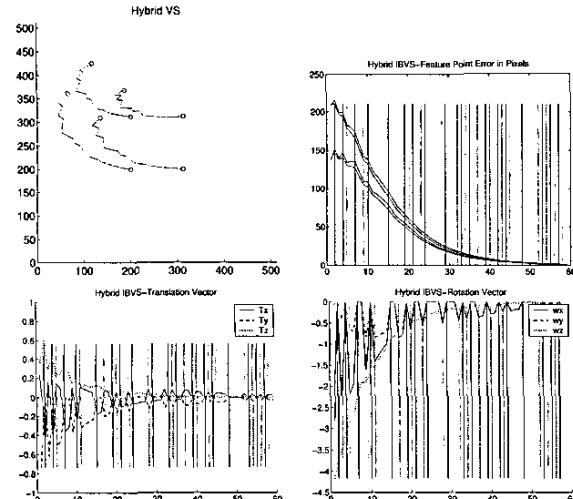


Fig. 16. General Motion, Random Switching

first portion, while performing x and y rotations. At this point the probabilistic effects begin to surface as the controller oscillates between methods, with a slight preference for the affine method. The error is zeroed in just under 60 iterations, and the error plots and trajectories are quite smooth while the velocities are irregular.

Simulations provided a good idea of the performance characteristics of our switched system controller. We then conducted experiments also involving an oblique view, as seen in Figure 10. The experimental setup was the same as that described in Section IV.

We first explored the state-dependent switching method. Since there is a great deal of rotation about that the camera y-axis, we expect that it will use the homographic method for the majority of iterations, and switch to using the affine method when the y axis rotation has become very small. Figure 18

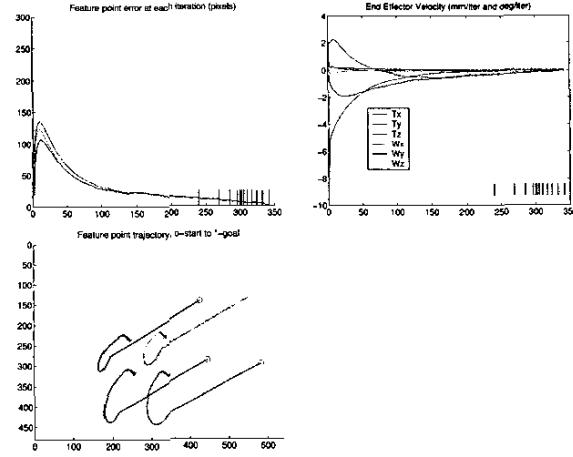


Fig. 18. Experiment Results State-Dependant Switching

shows the feature point error, the velocity screw of the vector and the recorded feature point position for each iteration. In the first two graphs, small lines on the bottom of the graph indicate a switch has taken place; a black line indicates the homography method is being used for the following iteration, while a cyan line indicated the affine method will be used. We do see the homographic method used for almost two thirds of the iterations, at which point it switches between the affine solution and the homographic method as the amount of y-axis rotation becomes negligible. The third graph shows the trajectory the point followed in the image plane. Portions of the lines with a black shadow indicate when the affine method is being used.

Figure 19 shows the results of random switching. All the measured values are much more chaotic. The feature point error tends to be greater, as does the magnitude of the velocity screw variables. However, the error is still zeroed in approximately the same amount of time and we also avoid the extremely large initial motion which the homographic method introduced in the

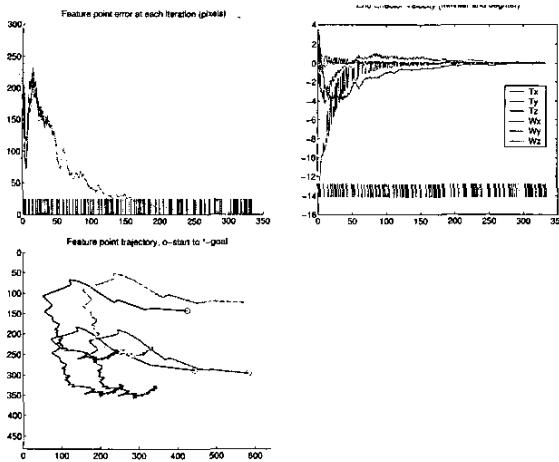


Fig. 19. Experiment Results Random Switching

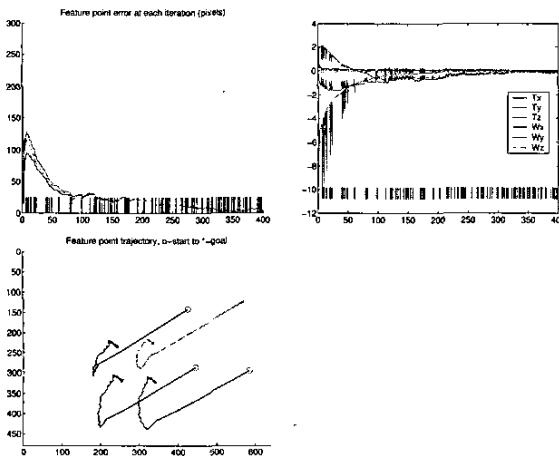


Fig. 20. Experiment Results Probabilistic Switching

state-dependent case.

Finally, Figure 20 shows the probabilistic system. The feature point error is similar to the state-dependent method, though it tends to be slightly smaller. Likewise the velocity screw tends to have similar shape and size to the state-dependent method. The system is, however, not able to zero the error as quickly as the other switching methods.

VI. CONCLUSION

We have presented two switched hybrid control visual servo systems. Each switched system is composed of two visual subsystems, and a decision maker that generates a signal to switch between them depending on the current state of the system, the time, or both. Simulation and experimental results are extremely promising. The IBVS/Homographic system showed a great deal of potential. It was ostensibly stable in all of our tests and successfully zeroed a task error that caused either individual subsystem to fail. The Affine/Homographic system also displayed stability in all of our tests, but the practicality of this system may be limited.

There remain future avenues to explore. There are more systems that could be integrated into a hybrid switched system framework, as well as the complication of including more than two continuous subsystems. There also remains the question of stability, which has not been resolutely established. Our experimental results are certainly compelling, and indicate this is a fruitful field for development.

REFERENCES

- [1] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE Journal of Robotics and Automation*, vol. RA-3, pp. 404–417, Oct. 1987.
- [2] J. Feddema and O. Mitchell, "Vision-guided servoing with feature-based trajectory generation," *IEEE Trans. on Robotics and Automation*, vol. 5, pp. 691–700, Oct. 1989.
- [3] P. Martinet, J. Gallice, and D. Khadraoui, "Vision based control law using 3d visual features," 1996.
- [4] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 12, pp. 651–670, Oct. 1996.
- [5] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The confluence of vision and control* (D. Kriegman, G. Hager, and S. Morse, eds.), vol. 237 of *Lecture Notes in Cont. and Info. Sci.*, pp. 66–78, Springer-Verlag, 1998.
- [6] E. Malis, F. Chaumette, and S. Bouabdellah, "2-1/2d visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 15, pp. 238–250, Apr. 1999.
- [7] G. Morel, T. Liebezeit, J. Szewczyk, S. Bouabdellah, 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.), vol. 250 of *Lecture Notes in Cont. and Info. Sci.*, pp. 99–108, Springer-Verlag, 2000.
- [8] K. Deguchi, "Optimal motion control for image-based visual servoing by decoupling translation and rotation," in *Proc. Int. Conf. Intelligent Robots and Systems*, pp. 705–711, Oct. 1998.
- [9] 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.
- [10] N. R. Gans, P. I. Corke, and S. A. Hutchinson, "Performance tests of partitioned approaches to visual servo control," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, 2002.
- [11] M. Branicky, V. Borkar, and S. Mitter, "A unified framework for hybrid control," in *Proc. of the 33rd IEEE Conf. on Decision and Control*, 1994.
- [12] R. W. Brockett, *Hybrid models for motion control systems*. 1993. H. L. Trentelman and J. C. Willems, Eds.
- [13] D. Liberzon and A. Morse, "Basic problems in stability and design of switched systems," *IEEE Control Systems Magazine* 19, 1999.
- [14] M. Branicky, "Multiple lyapunov functions and other analysis tools for switched and hybrid systems," in *IEEE Trans. Automat. Contr.*, 1998.
- [15] O. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, no. 3, pp. 485–508, 1988.
- [16] Z. Zhang and A. Hanson, "3d reconstruction based on homography mapping," in *ARPA Image Understanding workshop, Palm Springs, CA*, 1996.
- [17] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. on Robotics and Automation*, vol. 8, pp. 313–326, June 1992.
- [18] R. Kelly, R. Carelli, O. Nasini, B. Kuchen, and F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," in *IEEE Trans. on Mechatronics*, 2000.
- [19] R. Boel and J. van Schuppen, "Distributed routing for load balancing," in *Proceedings of the IEEE*, vol. 77, Iss. 1, 1989, pp. 210–221, Jan 1989.