Online Estimation of Image Jacobian by Taylor Polynomial using Evolutionary Strategy for Visual Servo Systems

Pariwat Keosatit, Prabhas Chongstitvatana

Department of Computer Engineering
Chulalongkorn University
Bangkok, 10330
Tel : (622) 218-6983
Fax: (662) 218-6955
e-mail : prabhas@chula.ac.th

Abstract

This paper proposed an online image Jacobian estimation technique for visual servo system. The technique estimate the Jacobian of the current robot position rather than use the estimated Jacobian of the last position for executing robot at the current position. Taylor polynomial is used to approximate image Jacobian function. Evolutionary strategy issue to find its coefficients. A task by a three–degree-of-freedom robot manipulator with visual feedback from stereo cameras is exemplified. The experiment is carried out by comparing with one offline method and the other two online methods under extensive simulations. The result shows that the proposed method when applied to adapt Jacobian performs the visual servoing task with smaller trajectory error than the other methods.

I. Introduction

Visual servo systems have been studied for several years. They have found limited use outside laboratories, partly because these systems require the complete information of the system model and the geometry of the robot workspace. This information is not precisely available in many applications. This is the motivation for investigating visual servo technique that requires less or no prior knowledge.

Most previous works on uncalibrated visual servoing focus on the image Jacobian-based scheme. The relationship given by image Jacobian describes how image feature parameters change with respect to changing manipulator pose. The image Jacobian was first introduced by Weiss et. al. [1], who referred to it as the feature sensitivity matrix. It is also referred to as the interaction matrix [2] and the B matrix [3], [4]. Other applications of the image Jacobian include [5], [6], [7], [8].

Researches on the estimation of the image Jacobian have been extensively studied. Some estimation methods depend on system configurations and tasks to be accomplished, e.g., in [9], [3]. Some methods use offline estimation which is correct only in a small region, e.g., in [10], [11], [12]. Some methods are online estimation but they use the estimated Jacobian of the last position for executing robot at the current position, e.g., in [13], [14].

In this paper, we propose a novel online method to estimate the image Jacobian matrix. This method does not need a prior knowledge of the kinematics structure or system parameters. It also estimates the Jacobian of the current robot position rather than uses the estimated Jacobian of the last position for executing robot at the current position.

This paper is organized as follows. The next section presents the proposed method in details. Section III describes the experiments. Section IV discusses the results. Finally we conclude the work in Section V.

II. Image Jacobian estimation

The image Jacobian relates the change of image feature parameters (e.g., the position of the robot's end effector in an image coordinate) Δf to the change of manipulator pose from a particular controller command Δr with the following relation:

$$\Delta f = J(r)\Delta r \qquad (1)$$

In most visual servoing work, a Jacobian has been either:

A. Derived analytically.

These techniques need prior knowledge of the kinematics structure or system parameters. They are used to calculate the Jacobian.

B. Derived partially analytically and partially estimated. (e.g., [9], [3])

These techniques still needs some prior knowledge of the kinematics structure or system parameters. They are used with some physical estimation to estimate the Jacobian.

C. Determined experimentally by physically executing a set of orthogonal calibration movements. (e.g., [10], [11], [12])

An initial estimation of the image Jacobian can be done by performing calibration motions as in [10]. They get the Jacobian estimation by test movements along the basis of the control variables. This constant Jacobian turns out to be accurate enough for subsequent control in their experiments.

However this Jacobian is correct only in a small region near the calibrated position of the robot.

D. Online estimated. (e.g., [13], [14])

Jagersand, Fuentes, and Nelson [13] used an online method, which estimates the Jacobian by just observing the process, without a priori models or introducing any extra calibration movements. In observing the last movement they obtain the changes in visual appearance $\Delta \mathrm{f}_{l\to(l+1)}$ corresponding to a particular controller command $\Delta \mathrm{r}_{l\to(l+1)}$. They want to update the Jacobian in such a way as to satisfy the most recent observation (Jacobian of the last position) which is the secant condition:

$$\Delta \mathrm{f}_{l\to(l+1)} = \mathrm{J}_l(\mathrm{r})\Delta \mathrm{r}_{l\to(l+1)} \tag{2}$$

The above condition is under-determined, and a family of Broyden updating formulas can be defined [17]. They choose the following asymmetric correction formula:

$$\mathrm{J}_l = \mathrm{J}_{l-1} + \frac{(\Delta \mathrm{f}_{l\to(l+1)} - \mathrm{J}_{l-1}\Delta \mathrm{r}_{l\to(l+1)})\Delta \mathrm{r}^T_{l\to(l+1)}}{\Delta \mathrm{r}^T_{l\to(l+1)}\Delta \mathrm{r}_{l\to(l+1)}} \tag{3}$$

Subsequently, Praditwong and Chongstitvatana [14] tried to adjust the Jacobian to satisfy (2) by using evolutionary strategy. They used evolutionary strategy to generate a number of variants of the Jacobian and used the observed changes in the last movement to select the best estimate among these variants. In other words, the best estimate is the one that, if it is used previously, the motion will be closer to the actual motion that is already known.

Note that the estimated Jacobian that fits the last movement (Jacobian of the last position) is used as the Jacobian for the current position. This is our motivation for developing the new method.

Our idea bases on the fact that image Jacobian is the function of the position. If we want to predict the Jacobian at the current position $l+1$, we can approximate the Jacobian functions by its Taylor polynomials around the position l and use them to find the Jacobian at $l+1$. We choose first-order Taylor polynomial. The Taylor polynomial is as follow:

$$f(x_1,x_2,\ldots,x_n) = f(a_1,a_2,\ldots,a_n) + \sum_{k=1}^{n}(x_k - a_k)\frac{\partial}{\partial x_k} f(a_1,a_2,\ldots,a_n) \tag{4}$$

Where $f(x_1,x_2,\ldots,x_n)$ is the function of $(x_1,x_2,\ldots,x_n)$ around point $(a_1,a_2,\ldots,a_n)$ and $\frac{\partial}{\partial x_k} f(a_1,a_2,\ldots,a_n)$ is Taylor's coefficient around point $(a_1,a_2,\ldots,a_n)$.

To have Taylor polynomial around the position l, we must find Taylor's coefficient around point l. Consider the following Taylor polynomial approximating j, an element of the Jacobian J at $(x_1,x_2,\ldots,x_n)$ around the position l:

$$\mathrm{j} = \mathrm{j}_l + \sum_{k=1}^{n}(x_k - l_k)\frac{\partial}{\partial x_k}\mathrm{j}_l \tag{5}$$

If we use (5) to evaluate j at $l-1$, we get

$$\mathrm{j}_{l-1} = \mathrm{j}_l + \sum_{k=1}^{n}((l-1)_k - l_k)\frac{\partial}{\partial x_k}\mathrm{j}_l \tag{6}$$

or

$$\mathrm{j}_l = \mathrm{j}_{l-1} - \sum_{k=1}^{n}((l-1)_k - l_k)\frac{\partial}{\partial x_k}\mathrm{j}_l \tag{7}$$

Evolutionary strategy is used to generate a number of variants of the Taylor's coefficient $\frac{\partial}{\partial x_k}\mathrm{j}_l$ and we select the variant which produce j_l that make J_l best fit the last movement datas. Once we have the Taylor's coefficient $\frac{\partial}{\partial x_k}\mathrm{j}_l$, we can use (5) to find j at $l+1$ and then we obtain J_{l+1}.

The J_l that best fits the last movement is the one that, if it is used previously, the motion will be closer to the actual motion that is already known. In other words, it is J_l that satisfied (2). Thus, we choose the following fitness function for the evolutionary strategy:

$$fitness = \left|\mathbf{\Delta r}_{l\to(l+1)} - \mathbf{J}_l^{-1}\mathbf{\Delta f}_{l\to(l+1)}\right| \tag{8}$$

Now, the evolutionary strategy used to find Taylor's coefficients will be discussed. Evolutionary strategy was developed by Rechenberg [15] and Schwefel [16] as an experimental optimization technique. $(1+1)-ES$ is adapted in our optimization. It works on the basis of two individuals only, i.e., one parent and one descendant per generation. The descendant is created by applying normally distributed variations with expectation zero and standard deviation σ to the parent (called mutation), and either the descendant becomes parent of the next generation, if it is better than its parent, or the parent "survives". The standard deviation σ is adjusted according to the 1/5-success rule [17]. This rule updates the standard deviation σ at each n-th generation, based on the measured relative frequency p of successful mutations:

$$\sigma = \begin{cases} \sigma / c & , \quad p > 1/5 \\ \sigma \cdot c & , \quad p < 1/5 \\ \sigma & , \quad p = 1/5 \end{cases} \tag{9}$$

A choice of c=0.817 derived by Schwefel [18] is used.

III. Experiment

In our simulation, we use MATLAB as the technical computing environment. The toolbox used to model robot is Robotics toolbox developed by Corke [19].

A three–degree-of-freedom robot manipulator is used with visual feedback from stereo cameras. We assume that the end effector and the target are visible in both cameras all the time.

We compare the trajectory error between: the method 1 where the Jacobian is determined experimentally by physically executing a set of orthogonal calibration moves [10], the method 2 -- online estimation method [14], the method 3 -- another online estimation method [13], and the method 4 -- the proposed method.

The metrics are 1) the number of moves to reach the target, and 2) the trajectory error, which is measured as deviation from the straight-line between the initial position of the end effector to the target. Because evolutionary strategy is non-deterministic algorithm, e.g., every run of the algorithm will give slightly difference results, we repeat the experiment 1000 times and the data is averaged from all runs. For deterministic method, one run is sufficient.

Six targets are randomly chosen. The step size used is 1/4. For evolutionary strategy parameters, we adjust σ every 20 generations and the maximum generation is 500. The initial value of σ is determined from the initial distance between the end effector and the target. The number of variances generated by evolutionary strategy is 100.

IV. Results

From Table 1, the numbers of moves of Method 1 are rather high. The other methods have similar numbers of moves. Table 2 shows the trajectory error. The trajectory error of Method 1 is highest because it uses constant Jacobian for every point in the path. The trajectory error of Method 2 and Method 3 are similar because of the same principle (Online estimation that Jacobian of the last position is used as the Jacobian for the current position). Our method (Method 4) has the lowest trajectory error since it tries to estimate the Jacobian of the current robot position rather than uses the estimated Jacobian of the last position for executing robot at the current position. This fact can be explained as follows. By solving (7) using evolutionary strategy, the Taylor's coefficient around the last position (the position l) is obtained. This coefficient is substituted into (5) to obtain the Jacobian of the current position. Hence our method obtains a more accurate Jacobian and uses it to calculate the motion to the next position. Fig. 1 shows typical trajectories of the compared methods in reaching a target.

V. Conclusion

We have shown that the proposed technique which uses Taylor polynomial with evolutionary strategy to approximate image Jacobian function performed better than the compared methods for the visual servoing task. The experiment demonstrates that this method can work very well. We are investigating higher-order Taylor polynomial and the other evolutionary strategies for coping with the higher-dimensional manipulation problem.

References

[1] A. C. Sanderson, L. E. Weiss, and C. P. Neuman, “Dynamic sensor-based control of robots with visual feedback control,” *IEEE Journal of Robotics and Automation*, Vol. 7, pp. 31-47, 1987.

[2] B. Espiau, F. Chaumette, and P. Rives, “A new approach to visual servoing in robotics,” *IEEE Transactions on Robotics and Automation*, Vol. 8, pp. 313-326, 1992.

[3] N. P. Papanikolopoulos and P. K. Khosla, “Adaptive robot visual tracking: Theory and experiments,” *IEEE Transactions on Automatic Control*, Vol. 38, pp. 429-445, 1993.

[4] N. P. Papanikolopoulos and P. K. Khosla, and T. Kanade, “Visual tracking of a moving target by a camera mounted on a robot: A combination of vision and control,” *IEEE Transactions on Robotics and Automation*, Vol. 9, pp. 14-35, 1993.

[5] J. Feddema and O. Mitchell, “Vision-guided servoing with feature-based trajectory generation,” *IEEE Transactions on Robotics and Automation*, Vol. 5, pp. 691-700, 1989.

[6] A. Castano and S. A. Hutchinson, “Visual compliance: Task-directed visual sservo control,” *IEEE Transactions on Robotics and Automation*, Vol. 10, pp. 334-342, 1994.

[7] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, “Manipulator control with image-based visual servo,” in *Proceedings of IEEE International Conference on Robotics and Automation*, Sacremento, CA, 1991, pp. 2267-2272.

[8] B. Yoshimi and P. K. Allen, “Active, uncalibrated visual servoing,” *Proceedings of IEEE International Conference on Robotics and Automation*, San Diego, CA, 1994, pp. 156-161.

[9] J. T. Feddema and G. C. S. Lee, “Adaptive Image Feature Prediction and Control for Visual Tracking with a Hand-Eye Coordinated Camera,” *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, pp. 1172-1183, 1990.

[10] A. Conkie and P. Chongstitvatana, “An Uncalibrated Visual Servo System,” in *Proceedings of British Machine Vision Conference*, Oxford, United Kingdom, 1990, pp. 277-280.

[11] N. Hollinghurst and R. Cipolla, “Uncalibrated Stereo Hand-Eye Coordination,” in *Proceedings of British Machine Vision Conference*, Surrey, England, 1993, pp. 389-398.

[12] G. Hager, “Calibration-Free Visual Control Using Projective Invariance,” in *Proceedings of the International Conference on Computer Vision*, Cambridge, MA, 1995, pp. 1009-1015.

[13] M. Jagersand, O. Fuentes, and R. Nelson, “Experimental evaluation of uncalibrated visual servoing for precision manipulation,” in *Proceedings of IEEE International Conference on Robotics and Automation*, Albuquerque, NM, 1997, pp. 2874-2880.

[14] K. Praditwong and P. Chongstitvatana, “An uncalibrated visual servoing using evolution strategies to estimate image jacobian,” in *Proceedings of the 2nd Asian Symposium on Industrial Automation and Robotics*, Bangkok, Thailand, 2001, pp. 60-63.

[15] I. Rechenberg, *Cybernetic Solution Path of an Experimental Problem*, Translated by B. F. Toms, Royal Aircraft Establishment Library Translation No. 1122, Ministry of Aviation, Royal Aircraft Establishment, 1965.

[16] H.-P. Schwefel, *Kybernetische Evolution als Strategie der experimentellen Forschung in der Stromungstechnik*, Diploma Thesis, Hermann-Fottinger Institute for Hydrodynamics, Technical University of Berlin, 1965.

[17] I. Rechenberg, *Evolutionsstrategie : Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*, Fromman--Holzboog Verlag, 1973.
[18] H.-P. Schwefel, *Evolution and Optimum Seeking*, Wiley,1995.
[19] P. I. Corke, “A robotics toolbox for MATLAB,” *IEEE Robotics and Automation Magazine*, Vol. 3, pp. 24-32, 1996.

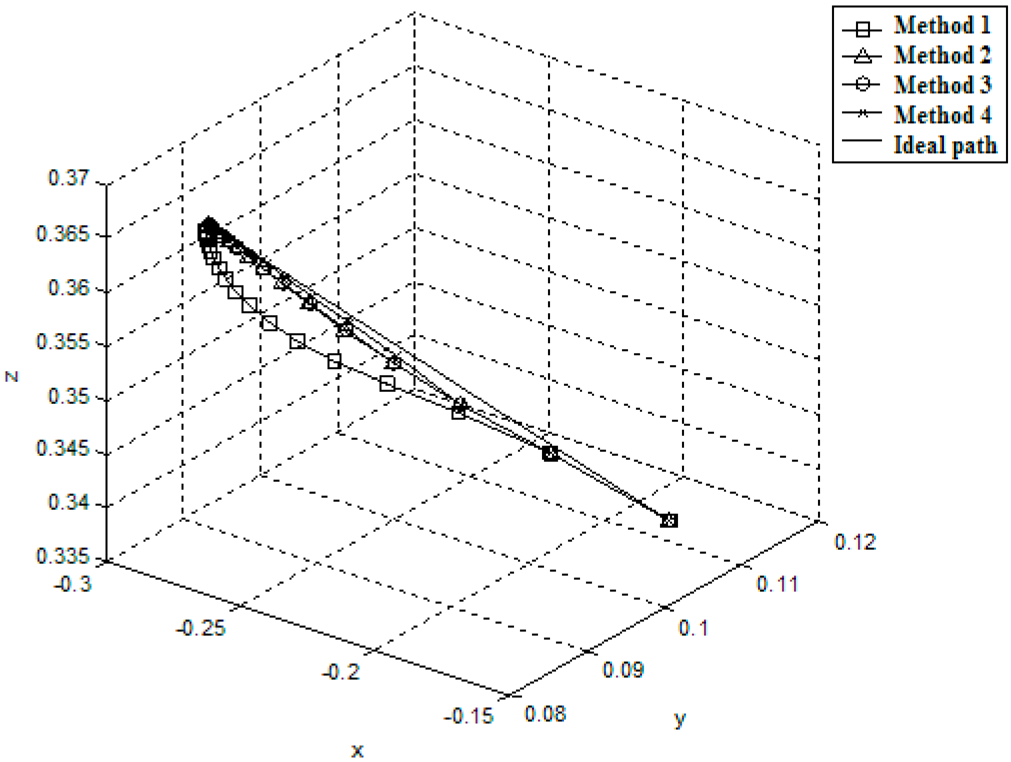

Fig. 1 The trajectories between the compared methods. **Method 1:** Determining experimentally by physically executing a set of orthogonal calibration method [10]. **Method 2:** Online estimation method that adjusts the Jacobian to satisfy the last movement condition by using evolutionary strategy [14]. **Method 3:** Online estimation method that uses Broyden updating formula to update the Jacobian in such a way as to satisfy the most recent observation condition [13]. **Method 4:** The proposed method -- Online estimation method that estimates the Jacobian of the current robot position.

TABLE 1
The number of moves

Path No.	**1**	**2**	**3**	**4**	**5**	**6**
Method 1	22	17	21	21	31	23
Method 2	17	16	19	19	21	20
Method 3	17	16	19	19	21	20
Method 4	17	16	19	19	20	20

TABLE 2
The trajectory error

Path No.	**1**	**2**	**3**	**4**	**5**	**6**
Method 1	4.84	2.87	7.85	5.21	17.00	20.59
Method 2	1.65	0.78	1.74	1.54	5.21	7.31
Method 3	1.44	0.75	2.40	1.14	5.61	6.37
Method 4	0.86	0.45	1.03	0.72	4.39	4.29